

UNIVERSITE PARIS VIII - VINCENNES-SAINT-DENIS

Ecole doctorale "Cognition, Langage, Interaction"

Laboratoire Cognitions Humaine et Artificielle

Development framework for language acquisition based on the embodied cognition: case of problem solving and gesture

Thèse de l'Université Paris 8 présentée et soutenue publiquement

Sanghun BANG

En vue de l'obtention du grade de Docteur en Psychologie Cognitive

le vendredi 15 janvier 2021

Jury

M. Youssef CHAHIR, Professeur, Université de Caen Normandie, Rapporteur

Mme Laure LEGER CHORKI, Maîtresse de Conférences-HDR, Université Paris Nanterre, Rapporteur

Mme Maria José ESCALONA CUARESMA, Professeure, Université de Séville, Examinatrice

Mme Sandra JHEAN-LAROSE, Professeure, Université d'Orléans, Examinatrice

Mme Olga MEGALAKAKI, Professeure, University d'Orléans, Examinatrice

M. Charles TIJUS, Professeur, Université Paris 8, Directeur

Mme Isis TRUC, Professeure, Université Paris , Présidente

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Prof. Charles Tijus for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study. I am grateful to the University of Seville's Maria José Escalona and Francisco José for their continued research collaboration and advice. As a co-supervisor, Prof. Maria José Escalona helped me a lot with my thesis.

Besides my advisor, I would like to thank the rest of my thesis committee in University Paris 8 and Ecole Pratique des Hautes Etudes. They asked me insightful questions about my thesis every year and gave generous advice.

We also thank our colleagues at the laboratory LUTIN (Le Laboratoire des Usages en Technologies d'information Numériques). In particular, engineer Tissier Geoffrey and Dr. Ventalon Geoffrey have always provided generous support and assistance. In particular, Mathilde Malgrange, a PhD student at the University Paris 8, successfully performed the Hanoi Tower experiment. The availability of Mathilde's data allowed me to finish my thesis. I would like to sincerely thank Mathilde and hope that her thesis goes well. I would like to express my gratitude to Dain JUNG, who helped correct my English writing.

I would like to thank my parents, BANG kukdong and SEO namsuk, who gave birth to me and raised me so far. It is all thanks to my parents that I can be here today. Especially, I cannot forget the encouragement and consideration of the father-in-law, KIM youngsu and KIM soonil. Once again, I would like to thank my father-in-law for taking this opportunity.

Finally, there is only one person that cannot be left out. I would like to express my greatest gratitude to my beloved wife, KIM eunsik. My wife always cared about and encouraged me for my studies even in the difficult process of raising a child after marriage. My wife's dedication and consideration allowed me to complete my thesis.

There are a lot of people to be grateful for and people to say sorry for, but I close my acknowledgement with the excuse of the limitations of the space.

Résumé

La cognition incarnée est fondée sur le fait que notre système cognitif n'est pas seulement connecté au corps, mais qu'il sert aussi de mode de communication du corps à l'environnement. Le cerveau vivant et actif, qui interagit avec l'environnement à travers notre corps vivant, a un impact majeur sur notre pensée, en particulier sur ses processus et mécanismes. Cela signifie que nos expressions communicatives comprennent les composantes corporelles de nos systèmes de capture sensorielle et les composantes émotionnelles qui sont liées à notre sensorimotricité et à la réalisation de nos séquences d'actions planifiées. À partir de ce cadre théorique, nos recherches ont eu pour but de tester l'hypothèse que les capacités cognitives humaines telles que le raisonnement, la résolution de problèmes accompagnée de l'apprentissage de phrases peuvent être impactées par notre gestuelle corporelle. Pour atteindre cet objectif de recherche, nous avons recueilli et analysé les données de résolution de problème en considérant le Deep Learning et l'apprentissage associatif comme modèles pour la cognition incarnée. Les données sont celles d'expériences menées auprès de participants aveugles et de participants voyants qui ont comme tâche la résolution du problème de la tour de Hanoi ; l'objectif étant d'examiner comment la résolution de problèmes peut être capturée en termes de cognition incarnée. L'intérêt pour l'intelligence artificielle, qui pourrait ainsi modéliser et mettre en œuvre les capacités cognitives humaines, a également été un point important de nos recherches. Dans un second temps, les données expérimentales ont été utilisées pour instancier et simuler des modèles cognitifs à l'aide de réseaux de neurones et d'apprentissage par renforcement. En déduisant la corrélation entre deux actions successives, nous avons implanté un raisonnement probabiliste associé à un modèle associatif cognitif pour générer l'action suivante. En particulier, nous avons conçu un modèle de réseau neuronal récurrent de résolution du problème de la tour de Hanoi à partir du traitement et de l'analyse d'images des actions consécutives réalisées par les participants. Enfin, nous avons essayé de modéliser les verbalisations des participants en les mappant au processus d'apprentissage récurrent du réseau neuronal basé sur la classification par étiquettes multiples. Il s'agit d'un prototype du modèle d'expressions langagières reflétant la planification et la réalisation du geste cognitif incarné qui pourrait accompagner l'apprentissage à partir des objets et de leurs relations. À travers cela, nous avons exploré comment reconnaître les objets et leur

transformation dans l'environnement par le geste corporel et associer les expressions verbales à ce processus d'apprentissage.

Mots-clés: Cognition incarnée, Tour de Hanoi, Apprentissage automatique, Réseau neuronal convolutif, Intelligence artificielle, Résolution de problèmes, Développement cognitif, intelligence sensori-motrice

Abstract

The idea of embodied cognition is based on the fact that our brain is not only a living organ which is connected to our body, but also the mode that the body communicates with our environment. The alive and active brain, which interacts with the environment through our living body, has a major impact on our thinking, especially cognitive thinking. This means that our communicative expressions are composed of bodily component following our sensor-motor systems and emotional components which concerns realization of our sensor-motor actions and of planned actions. Therefore, our research attempted to test the hypothesis that human cognitive abilities such as reasoning, problem solving accompanied by sentences learning can be deeply impacted by our body gesture. To achieve this research purpose, we have analyzed and explored embodied cognition, deep learning, and reinforcement learning, which are the theoretical backgrounds of the thesis. And then, we conducted experiments with blind and sighted participants to actually solve the Tower of Hanoi problem. The purpose of this study was to examine how problem solving can be captured in terms of embodied cognition. Interest in artificial intelligence, which can model and implement human cognitive abilities, was also the important point of our research.

Therefore, all these experimental data were then used to train and simulate cognitive models using neural networks and reinforcement learning. By inferring the correlation between two successive actions, we made a probabilistic reasoning and cognitive associative model to find the next behavior. In particular, we designed a Recurrent Neural Network model and solved the Tower of Hanoi problem with the help of analysis of consecutive images. Finally, we tried to understand natural language by mapping learning sentences to Recurrent Neural Network learning process based on multiple label classification. This is a prototype of the embodied cognitive language model that can learn natural language from objects and their relationships. Through this, we explored how to recognize things from the environment and develop them into language learning.

Keywords: embodied cognition, Tower of Hanoi, Machine learning, Convolutional Neural Network, Artificial Intelligence, Problem solving, Cognitive development, sensorimotor intelligence

Table of contents

List of figures	xiii
List of tables	xvii
1 Introduction	1
1.1 Introduction	1
1.2 Thesis organization	2
I Background Theory	5
2 Embodied cognition and Artificial Intelligence	7
2.1 Introduction	7
2.2 The new challenge of cognitive science	9
2.2.1 Cognitive revolution and Emergence of embodied cognition theory	9
2.2.2 Cognitive research based on Embodied cognition	10
2.3 Four aspects of embodied cognition theory	11
2.3.1 Embodied cognition	12
2.3.2 Embedded Cognition	13
2.3.3 Enactive Cognition	14
2.3.4 Extended cognition	15
2.4 Combination of Artificial Intelligence and Embodied Cognition	18
2.4.1 Classical Artificial Intelligence and Connectionism	19
2.4.2 Embodied Artificial Intelligence	20
2.5 Conclusions and the future	22
3 Neural Networks and Deep Learning	23
3.1 Introduction	24
3.2 The simple perceptron	25

3.3	Backpropagation algorithm	29
3.4	Recurrent Neural Networks	32
3.5	Convolutional Neural Network(CNN)	36
3.6	Conclusion and the future	42
4	Deep Reinforcement Learning	45
4.1	Introduction	46
4.2	Reinforcement Learning and Markov Decision Process	47
4.2.1	Basic concept	47
4.2.2	Value function	50
4.3	Monte Carlo Method and Temporal Difference Algorithm	51
4.4	State-Action-Reward-State-Action and Q-Learning	54
4.5	Deep Q-Learning	55
4.6	Conclusion	58
II	Experiment and Simulation	61
5	The effects of gesture in Tower of Hanoi	63
5.1	Introduction	63
5.2	Experiment: Observing blind and sighted participants solving the Tower of Hanoi problem	64
5.2.1	Participants	64
5.2.2	Procedure	65
5.2.3	Coding	65
5.3	Method and Analysis	67
5.3.1	The effect of deictic gesture	68
5.3.2	The effect of gesture interaction	69
5.3.3	Discussion	71
5.4	General discussion and conclusion	73
6	Solving the Tower of Hanoi using Neural Network approach	75
6.1	Introduction	75
6.2	Reasoning model	76
6.2.1	Multi-Layer Perceptron Model	76
6.2.2	Neural Association Models	78
6.2.3	Relational Network Model	80
6.3	Simulation	82

Table of contents

6.3.1	Experiment: Multi-Layer Perceptron Model	82
6.3.2	Experiment: Neural Association Models	86
6.3.3	Experiment: Relational Network Model	89
6.4	Conclusion	90
7	Problem solving using Recurrent Neural Network based on the effects of gestures	93
7.1	Introduction	93
7.2	Background and Related Work	95
7.2.1	Tower of Hanoi	95
7.2.2	Recurrent Neural Network	95
7.2.3	Reinforcement Learning	96
7.2.4	Tower of hanoi and Reinforcement Learning	97
7.3	Model	99
7.4	Experiments	101
7.4.1	Coding	101
7.4.2	Experiment: Tower of Hanoi	101
7.4.3	Experiment: RNN+RL model	103
7.4.4	Results	105
7.5	Conclusions	105
8	Using Convolutional Neural Networks and Recurrent Neural Network for Human Gesture Recognition and Problem Solving	109
8.1	Introduction	109
8.2	Background and Related Work	111
8.2.1	Tower of Hanoi	111
8.2.2	Recurrent Neural Network	111
8.2.3	Convolutional Neural Networks	113
8.2.4	Reinforcement Learning	113
8.3	Model	114
8.3.1	CNN Multi-label categorization	114
8.3.2	Combined Recurrent Neural Network and Reinforcement Learning	115
8.4	Experiments	121
8.4.1	Experiment: Multi-label categorization	121
8.4.2	Experiment: LSTM+RL	124
8.4.3	Results	126
8.5	Conclusions	126

9 Understanding Natural Language with Convolutional Neural Networks for Problem solving	129
9.1 Introduction	129
9.2 Background and related work	130
9.2.1 Cognitive Development Theory and Embodied cognition	130
9.2.2 Convolutional Neural Network in Natural Language Processing	131
9.3 Model	134
9.4 Experiments	137
9.5 Conclusion and the future	141
10 Discussion and Conclusion	143
References	147
Appendix A Results of Tower of Hanoi experiments on blind and sighted groups.	157
A.1 Results of Tower of Hanoi experiments for sighted group	157
A.2 Results of Tower of Hanoi experiments for the blind group	159
A.2.1 Visualization of the correlation matrix	161
A.3 Summary Two-Way ANOVA and Interaction Effects Results	163
A.3.1 Interaction effects for deictic gesture	163
A.3.2 Interaction effects for gesture interaction	164
Appendix B Tower of Hanoi Dataset	165
B.1 Tower of Hanoi Resolution Dataset for MLP	166
B.1.1 Movement encoding	166
B.2 Tower of Hanoi Resolution Dataset for Recurrent Neural Network	172
B.2.1 Movement encoding	172
B.2.2 Sequential data for the solution of TOH	173
B.3 Sequential images and coding	175
B.3.1 Sample Coding for Convolutional Neural Network	175
B.3.2 Sample Coding for Natural Language Processing	178
Appendix C Convolutional Neural Network - Tower of Hanoi	181
C.1 Image processing - kernel	181

List of figures

2.1	A brain in a vat that believes it is walking	12
2.2	An example of coupling arguments - Tetris Game	17
2.3	DyRET: Dynamic Robot for Embodied Testing	21
3.1	An artificial neuron	26
3.2	A biological neuron	26
3.3	Geometric representation of Perceptrons	27
3.4	A single Perceptron's Limitations	28
3.5	Multilayer perceptron	29
3.6	Recurrent Neural Network Architecture(RNN)	33
3.7	LSTM Memory cell	35
3.8	Hubel and Wiesel Experiment	37
3.9	Convolutional neural network structure	38
3.10	Convolution and subsampling	39
3.11	Comparison of fully connected layer, regionally connected layer, and convolutional layers.	39
3.12	Convolutional Neural Networks for image classification	42
4.1	Tolman's Maze	46
4.2	Basic concept of Reinforcement Learning	48
4.3	Markov Decision Process	48
4.4	Model-free Model-based Reinforcement Learning	54
4.5	State-Action Pair	54
4.6	Sarsa algorithm	56
4.7	Q-Learning algorithm	56
4.8	Schematic illustration of the convolutional neural network	57
4.9	Deep Q-Learning with Experience Replay Pseudo Algorithm	60
5.1	A participant performing Tower of Hanoi with disk	65

5.2	A participant performing Tower of Hanoi without disk	66
5.3	Effect of deictic gesture between group and condition	70
5.4	Effect of gesture interaction between group and condition1	72
6.1	Association Between raining and umbrella	76
6.2	Model of Multilayer perceptron for Tower of Hanoi	77
6.3	Structure of Neural Association Model	79
6.4	Association in Deep Neural Networks	79
6.5	Rule of game of the Tower of Hanoi	81
6.6	Model of Relational Network	81
6.7	Model loss according to the learning rate of multilayer perceptron model(12 input layers, 5 hidden layers and 7 output layers)	83
6.8	Model loss according to the learning rate of multilayer perceptron model(12 input layers, 5 hidden layers and 7 output layers)	84
6.9	ROC curves according to the learning rate of multilayer perceptron model(12 input layers, 5 hidden layers and 7 output layers)	85
6.10	Model accuracy according to the learning rate of multilayer perceptron model(12 input layers, 10 hidden layers and 7 output layers)	87
6.11	Model loss according to the learning rate of multilayer perceptron model(12 input layers, 10 hidden layers and 7 output layers)	88
6.12	Solution acquired by human	90
6.13	Reasoning model with different number of neurons was trained to find the rule of TOH.	90
7.1	Tower of Hanoi puzzle (initial state)	95
7.2	Simple Recurrent Neural Network Architecture	96
7.3	Model of Reinforcement Learning	97
7.4	The state-transition diagram corresponding to the 4-disk structure	98
7.5	Four disks Hanoi Tower Solution with Reinforcement Learning	100
7.6	Solving of Tower of Hanoi puzzle with RL and RNN	101
7.7	Coding for all possible actions	102
7.8	Sample coding from the acquired image	102
7.9	Comparing Results with Participants and the RNN Model	106
7.10	RNN train error at each epoch	106
7.11	Cumulative reward graph for RNN+RL model	107
8.1	Four disks in Tower of Hanoi puzzle	112
8.2	The Deep Convolutional Neural Network for multi-label classification	113

List of figures

8.3	Model(Reinforcement Learning) extracted from [5]	114
8.4	The per-class barplots of input images.	115
8.5	Labelling for the required images.	116
8.6	Solving of Tower of Hanoi puzzle with CNN and LSTM-RL	117
8.7	Coding - Move a disk from the peg	120
8.8	The predicted result based on Convolution Neural Network(Participant1)	121
8.9	The predicted result based on Convolution Neural Network(Participant2)	122
8.10	The predicted result based on Convolution Neural Network(Participant6)	123
8.11	The predicted result based on Convolution Neural Network (Participant1-second image)	123
8.12	Accuracy comparison for a convolutional neural network (CNN) with layers and epochs.	124
8.13	Accuracy comparison for a convolutional neural network (CNN) with layers and epochs.	126
8.14	Loss comparison for a convolutional neural network (CNN) with layers and epochs.	127
8.15	LSTM train error at each epoch	127
8.16	Cumulative reward graph for LSTM+RL model	128
9.1	Chimpanzee Ayumu performing the masking task from Inoue and Matsuzawa [64]	132
9.2	Convolutional Neural Networks for Sentence Classification	135
9.3	A Combined Convolutional Neural Network for Natural Language Processing	136
9.4	A Combined CNN-CNN Model architecture for Natural Language Processing	136
9.5	Accuracy and loss comparison for a natural language processing - In the case of left Peg	139
9.6	Accuracy and loss comparison for a natural language processing - In the case of middle Peg	139
9.7	Accuracy and loss comparison for a natural language processing - In the case of right Peg	140
9.8	Accuracy and loss comparison for a natural language processing - In the case of taking disk	140
9.9	Accuracy and loss comparison for a natural language processing - In the case of putting down disk	141
A.1	Correlation Matrix Type1	161
A.2	Correlation Matrix Type 2	162

C.1	Image processing - Sharpen operation	181
C.2	Image processing - Edge detection operation 1	182
C.3	Image processing - Edge detection operation 2	182
C.4	Image processing - Edge detection operation 3	182
C.5	Image processing - Box blur operation	183
C.6	Image processing - Gaussian blur 5×5	183

List of tables

5.1	Deictic gesture between group and condition (Average)	69
5.2	Tukey post-hoc comparison for the effect of deictic gestures (Try2)	69
5.3	Gesture interaction between group and condition (Average)	70
5.4	Tukeys post-hoc comparison for the effect of interactive gestures (Try1)	71
6.1	One example of training data set for multi-layer neural network model	77
6.2	Confusion Matrix	78
6.3	Model performance with the Multi-Layer Perceptron Model (12 input layers, 5 hidden layers and 7 output layers)	82
6.4	Model performance with the Multi-Layer Perceptron Model (12 input layers, 10 hidden layers and 7 output layers)	86
6.5	Coding for the Model of Relational Network	89
6.6	Model performance for Tower of Hanoi with 3 disks (training) and 4 disks (prediction)	91
7.1	Rewards and possible actions at time 0.	99
7.2	Probabilities and possible actions at time 0.	99
7.3	Solution acquired by a participant	103
7.4	Results of Tower of Hanoi for 15 participants	104
7.5	State Transition and Probabilistic Selection of RNN Models	105
8.1	The summary of Our Convolutional Neural Networks model	118
8.2	Sequential images and probabilities	119
8.3	Rewards and possible actions at time 0 extracted from ([5]).	120
8.4	Probabilities and possible actions at time 0 extracted from ([5]).	121
8.5	Solution acquired by a participant extracted from [5].	122
8.6	Negative Reward and possible action extracted from [5].	123
8.7	Probabilities and possible actions extracted from [5].	123
8.8	Results on performance metrics for all labels.	125

9.1	Example of data preprocessing	137
9.2	The summary of Our Convolutional Neural Networks model for Natural Language Processing	138
9.3	CNN results for the location of the disk and the participants' actions	138
A.1	Results of Tower of Hanoi experiments for sighted group	158
A.2	Results of TOH experiments for the blind group	160
A.3	Summary Two-Way ANOVA and Interaction Effects Results for Number of deictic gesture on first try by condition and group	163
A.4	Summary Two-Way ANOVA and Interaction Effects Results for Number of deictic gestures on second try by condition and group	163
A.5	Summary Two-Way ANOVA and Interaction Effects Results for Number of gesture interaction on first try by condition and group	164
A.6	Summary Two-Way ANOVA and Interaction Effects Results for Number of gesture interaction on second try by condition and group	164

Chapter 1

Introduction

1.1 Introduction

Many scientists have sought to explain physically the phenomena and the workings of the mind in humans. With the advent and development of information technology, we have been using this information technology to study the mind, especially to realize human cognitive abilities. In recent years, computers are replacing people when it comes to intelligent work. Information technology learns the search patterns of computer users and presents possible search results in advance. In addition, information technology recognizes human voices and attempts to communicate with humans or strengthens the security system through facial recognition. The basis of all these user convenience and technological advances is the breakthrough in artificial intelligence technology. From a technical engineering point of view, the work of modern artificial intelligence can be the concept of weak artificial intelligence that helps people work. However, recently, a model for human intelligence has been proposed beyond this simple technical aspect. Therefore, the starting point of our research began with an interest in artificial intelligence that can imitate and implement human cognitive abilities. So our research began by asking the question of how on earth are human intelligence and cognitive ability formed ?

We send messages of love to our children. We laugh at the children, give them a warm touch, and give them an angry expression when they do something wrong. Before children know the word love, they understand the meaning through visual information and actions of parents. We also say that while looking at the fluctuations of the stock market, prices rise and fall like a seesaw game. There is currently a video coming into our minds. This is an example of knowing that information of play from childhood experiences reconstructs their meanings through language. Therefore, we have been paying attention to the embodied cognitive theory that is receiving new attention in psychology. It is the theory that human

cognitive processes and decisions are influenced by sensory movements such as touch and smell experienced in daily life. The purpose of our study is to realize human cognitive ability based on embodied cognition. We used artificial intelligence to analyze information received through sensorimotor organs. Since then, we have tried to implement artificial intelligence that can infer and judge based on the analyzed data, furthermore, can learn sentences. Our study attempted to verify the hypothesis that human language recognition and cognitive ability, such as sentence learning and reasoning, are possible by a lot of information received through our sensory motors. In the future, our research will expand further through the combination of robot and artificial intelligence as a tool for acquiring information through sensory movements such as tactile and visual.

1.2 Thesis organization

To achieve this research purpose, in Chapters 2 to 4, we have analyzed and explored embodied cognition, deep learning, and reinforcement learning, which are the theoretical backgrounds of the thesis. In Chapter 5, we conducted experiments with the blind and the sighted participants to actually solve the Tower of Hanoi problem. The purpose of this study was to examine how subjects approach problem solving in terms of embodied cognition. In particular, in order to find out how the gesture affects cognition in the problem-solving process, we first asked to solve the Tower of Hanoi Puzzle with four disks and secondly to solve the problem without any disk. In this case, however, the subject explained the solution process to the supervisor.

All these experimental data were used to simulate cognitive models using neural networks and reinforcement learning. The purpose of the simulation of this cognitive model is to develop an embodied cognitive model similar to humans after observing the subject's problem solving process. Subjects constantly observe and interact with surrounding objects (disc, pillars, etc.) during the Hanoi Tower problem solving process. Subject tries to infer problem solving using gestures or touching objects. We simulated this diverse rapport and reasoning process using various deep learning technologies.

In Chapter 6, the traditional multilayer perceptron was first simulated with a three-disk Hanoi tower problem. We then modeled if the Hanoi Tower problem with four disks could be solved through inference without learning the data. We understand the rules of the game that you should put small disks on a large disk one at a time, and know the rules for moving disks on the left side to the right side in the shortest time. Therefore, since we know the rules, it is possible to solve problems with 4 or 5 disks. Of course, it will take a lot of trial and error to solve the optimal solution. However, in Chapter 6, we simulated how to solve four disks

without learning data about four disks using inference model. We have deduced the solution using a probabilistic reasoning model via deep learning. This is a model of inference based on the data learned from the relationship between objects.

In Chapter 7, we focused on a series of data on problem solving. Therefore, based on this series of data, we designed the Recurrent Neural Network model and solved the Hanoi Tower problem. However, if most of the data did not provide the shortest path, the Recurrent Neural Network model had the problem that it could never find the shortest path. This is because the number with the greatest probability is suggested. Therefore, in order to solve this problem, the model was modified and simulated to find the shortest path by applying reinforcement learning theory.

We coded a series of data on the whole process of problem solving in Chapter 7. For example, if the disk was moved from left to center, G1 was coded, and if it was moved from center to right, G2 was coded. Therefore, using this method, each movement was coded. On the other hand, in Chapter 8, we recorded a video of the subject's problem-solving process and divided the video into consecutive images. Each image was classified into multiple labels. Based on the classification of multiple labels, we designed an RNN model and performed simulations. By inferring the correlation between two successive images, we made a cognitive model to infer the next behavior. Finally, Chapter 9 can be seen as an extension of Chapter 8. We tried to understand natural language by mapping learning sentences to RNN learning process based on multiple label classification. It is a prototype of the embodied cognitive language model that can learn natural language from objects and their relationships. Through this, we explored how to recognize things from the environment and develop them into language learning.

Part I

Background Theory

Chapter 2

Embodied cognition and Artificial Intelligence

2.1 Introduction

Embodied Cognition is a theory of cognitive science stating that people's cognitive processes and decisions are influenced by sensory movements such as touch and smell that are experienced in everyday life. In other words, human thinks not only with the brain, but also with a body with sensory motor skills. According to a research paper, "Experiencing Physical Warmth Promotes Interpersonal Warmth", by psychologist John Bargh at Yale University(Williams and Bargh [141]), if someone is holding a hot cup of coffee, he/she tends to judge other people to be more generous and caring people than others who are holding a cold cup of coffee. He concludes that people holding a hot cup of coffee evaluated strangers as warm-hearted, kind and sensitive people whereas people holding a cold cup of coffee showed a tendency of looking at strangers as uncomfortable to approach and silent people. As this indicates, physical touch affects making decisions unconsciously.

In the From the play "Macbeth" by William Shakespeare, Lady Macbeth conspires with her husband to kill the king. After committing the murder, she mutters, rubbing her hand."Out, out, damn spot". Even though there was no blood on her hands, she would have thought that washing her hands would wash away her guilt. This is the so-called effect of Lady Macbeth.

Let us take another example. "Parasite", which won the Palme d'Or at the 2019 Cannes Film Festival, deals with the theme of smell. The boss of a company with a lot of wealth has a discreet conversation with his wife. He says that their chauffeur smells unpleasant, like the people with lower social status. In this film, the director differentiates people by associating

smell with social class. The film was looking for the problem of class from smell. The mind is affected by the body and the sensory organs to understand abstract concepts such as social problems and ethics.

Recently, psychology is receiving a new interest in embodied cognitive theory. According to Descartes, cognitive processes take place within individuals isolated from the outside world. And because of this, the representation that has meaning is seen as the essence of the mind. Contrary to the Descartes's point of view, recent psychology states that primary intelligence developed based on the behavior of the body, and then intelligence developed into abstract concepts and languages in the course of evolution. The embodied cognitive approach has rediscovered sensory mechanisms, the importance of the environment, and the relationship between cognition and the environment. So embodied cognition appeals to the idea that cognition depends not only on the agent's body, but also on the surrounding.

In the 17th century, Spinoza explains that the body and mind are not separate existence but are the same reality but contain two very different aspects. Nietzsche rejects the mental superiority over the body in the philosophy of life (*Lebensphilosophie*). Later, through Heidegger, through the French philosopher Merleau-Ponty, the phenomenological stance on the body and mind is clearly declared. Finally, epistemological thinking about the body begins to appear. The paradigm of cognitive science called embodied cognition has brought about a second cognitive revolution since the 1980s. Embodied cognition can be said to be an attempt to unravel the human cognitive process not only through the human body as a biological object, but also through the brain. In addition, embodied cognition is not limited to cognitive science but expands to the paradigm with artificial intelligence. It is a challenge as an embodied artificial intelligence to interact with the environment, and humans acting autonomously through sensors in the real world.

This chapter will examine the emergence of cognitive science and embodied theory called the cognitive science revolution of the 20th century as well as the theoretical background related to embodied cognition. In addition, along with the rapid development of artificial intelligence in recent years, we will examine how embodied perception will change the paradigm of artificial intelligence.

2.2 The new challenge of cognitive science

2.2.1 Cognitive revolution and Emergence of embodied cognition theory

In the end of the 1950s, psychology attempts to reconstruct the natural notion of mind. As an attempt to reintroduce empirical research and to introduce a new approach, it is considered a paradigm shift that has been approached in a new conceptual framework of philosophy of mind that has been dealt with in philosophical traditions for a long time. We call this attempt a cognitive revolution which is considered as the scientific revolution of the 20th century.

By this cognitive shift of the 1950s, the mind's operation was recognized as the same as that of computer information processing. Thus, the mind's work, which was excluded from previous cognitive behaviorism, emerges as a scientific research subjects. The mind was viewed as a calculable object. We looked at the human brain as the hardware of a computer and looked at the human mind as software.

Early cognitive science focused on human cognitive function. Cognition is a recurring mental process in which a person perceives stimuli and information, encodes them in various formats, stores them in memory, and later reproduces them. Therefore, because of the complex cognitive function, cognitive science requires cooperation between the interdisciplinary disciplines such as philosophy, psychology, linguistics, anthropology, neuroscience, and artificial intelligence. In addition, cognitive science presupposes the mind as a sign system, so the mind manipulates signs in various cognitive processes such as thinking, perception, and memory. The process by which the mind manipulates symbols is called calculation. The goal of cognitive science, therefore, is to uncover computational theories that explain how the mind works. The method of studying the mind is divided into top-down and bottom-up methods. Top-down sees the whole as determining parts, while bottom-up sees the actions of parts as determining the whole. In the case of cognitive science, if the cognitive activity performed by the brain corresponds to the upper level, the electrochemical phenomenon occurring inside the nervous system of the brain falls into the lower part.

It was divided into a cognitive psychology of a top-down approach but since the 1980s, a neuroscience of a bottom-up approach has been paid attention. In the early days of cognitive science, the role of the body in information processing was not so essential. In order to transmit information to the brain, the body was considered as just an input and output device. But in the late 1980s, embodied cognitive theory emerged that the senses and actions of the body affect the cognitive function of the mind.

2.2.2 Cognitive research based on Embodied cognition

At the beginning of the 20th century, positivism and psychology were rampant. Positivism regarded the objectivity of natural science as omnipotent, and psychology tended to stick to human subjective experience. Husserl questioned the dichotomy itself to distinguish objectivity and subjectivity. Husserl draws on the term 'intentionalität'. Intentionality means that our consciousness always means 'consciousness about what'. The notion of 'Orientation of consciousness' is only possible in relation to the object.

Merleau-Ponty, succeeding Edmund Husserl's theory, studied existential phenomenology. Merleau-Ponty rejected dualistic divisions of subjectivity and objectivity, nature and spirit. For him, human beings are physical beings and rooted in the world through the body. He says that the world is the natural background and environment of all my thoughts and all my obvious perceptions (1945; *Phenomenology of Perception*). Mind and body are not two, but one. It suggests that human language and thought are embodied through the body.

Descartes doubted and doubted everything in order to gain self-evident truth (or axiom). For Descartes, the starting point of philosophy had to be clear and certain. Descartes used methodological skepticism to reach certainty

"I have a complete understanding of what a body is when I think that it is merely something having extension, shape and motion, and I deny that it has anything which belongs to the nature of a mind. Conversely, I understand the mind to be a complete thing, which doubts, understandings, wills, and so on, even though I deny that it has any of the attributes which are contained in the idea of a body. This would be quite impossible if there were not a real distinction between the mind and the body. (Descartes 1641/1984, 86)"

Johnson and Lakoff in their book 'Philosophy in the Flesh' insist on embodied cognitive theory based on Experientialism. The argument is based on the theory that 1) the mind is inherently incarnated, 2) human thinking is almost unconscious, and 3) human thinking has a metaphoric nature. The physical experience, in particular the mind is constituted by the sensory-motor experience and is a natural metaphorical thinking based on the body. Thus, as mentioned in representationalism or classical cognitivism, the 'Multiple Realizability Thesis' that claims to be feasible on a computer is not an appropriate approach to mind research (Lakoff and Johnson [75]).

Embodied cognitive theory not only extends the scope of cognitive processes that were confined to the brain to the body, but also extends cognition from the human body to the environment and world in which humans live.

2.3 Four aspects of embodied cognition theory

This theory considers the cognitive process to be done by the environment, that is, it depends on the environment. Wilson's work, which characterizes embodied cognition, is frequently cited to understand the environment and the situation. Wilson defines six aspects of embodied cognition (Wilson [143]) :

1. Cognition is situated.
2. Cognition is time pressured.
3. We off-load cognitive work onto the environment.
4. The environment is part of the cognitive system.
5. Cognition is for action.
6. Off-line cognition is body based.

In order to understand the concept of environment and situation in embodied cognition, we need to check the claims from 1 to 4. Claim 3 reveals the role of the environment in the context of cognitive work. It means that our cognitive burden is distributed to the environment when we do cognitive tasks in the current situation. In claim 4 suggests that the cognitive system extends not only to the brain and body but also to the environment. The claims 1 and 2 emphasize that cognition occurs in any time under certain circumstances.

Wilson says in claim 1, that situated cognition is cognition that takes place in the context of task-relevant inputs and outputs. In addition, claim 2 means that the time pressure is the cognitive agents must cope with the constraints of "real-time" or "runtime". This is a criticism of the classical Artificial Intelligent model, which underestimates the constraints on a cognitive context and assumes a cognitive model that forms and manipulates internal representations of situations with time.

Wilson says, "Cognitive activity takes place in the context of a real-world environment, and it inherently involved perception and action". Situated cognition seems to be regarded as perceptions in which perception and action are inherently involved in the context of a real world environment.

2.3 Four aspects of embodied cognition theory

Rowlands describes four main types of embodied cognition (Embodied cognition, Embedded cognition, Enactive cognition, and extended cognition) based on embodiment (Rowlands [125]). The embodied cognitive theory not only extends the scope of cognitive processes

that were confined to the brain to the body, but also extends cognition from the human body to the environment and the world in which humans live. More specifically, let's look at the classification of embodied cognition.

2.3.1 Embodied cognition

Embodied cognitive theory is a theory that studies the mind embodied in the body (Lakoff and Johnson [75]). Early cognitive science studied the human mind based on representationalism and computationalism. Thus, the study basically followed the de-embodied point of view. Cognitive scientists believe that human thinking can be sustained by the signaling system that is transmitted to the brain due to the fact that the body does not have a hard time thinking even when the body is incomplete. He thought that if there is a brain without a body, human thinking can be maintained. More specifically, let us take an example of brain-in-the-vat thinking experiment (Figure 2.1).

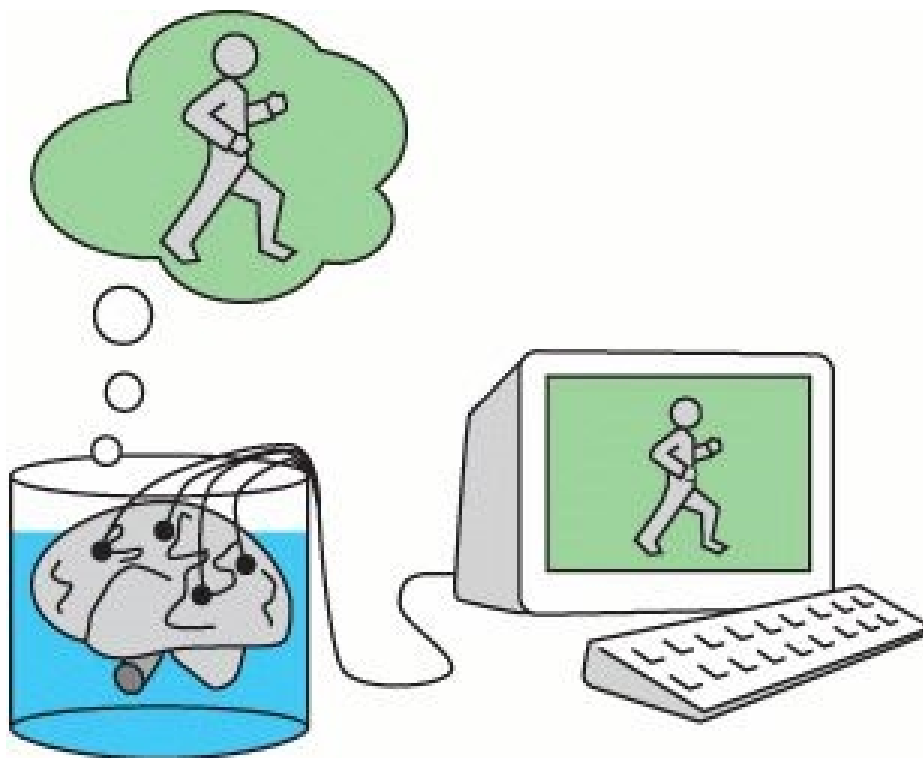


Fig. 2.1 A brain in a vat that believes it is walking.
(source: <https://extimatrix.wordpress.com/brain-in-a-vat-2/>)

Let us suppose a neuroscientist separates a brain from someone's body and keeps it separate. And suppose you've made a vat that connects neurons to wires and receives all the signals that humans have received. The question 'I am the brain in the vat?' presupposes

2.3 Four aspects of embodied cognition theory

that I am not the brain in the vat (Putnam [117]). Indeed, the brain in the vat only receives signals, not having any self-consciousness of who it is.

However, embodied cognitivists pay more attention to the meaning of vat in this thought experiment. Specifically, the prerequisites required for embodied cognitivists are as follows. First, the brain is alive and working well, and secondly, all epibiotic stimuli are replicated, and thirdly, all endotrophic self-organization activities are maintained without problems. In embodied cognitive scientists, the minimum sufficient condition to enable consciousness is that the entire body including the brain, must be involved. The brain is a body organ, but it cannot be an organism. The organism has to make a conscious approach to the world.

This vat is understood as a kind of substitute body because it satisfies the body's attributes and functions. The embodied cognitive theory thus argues that it is certain that our cognitive experiences are formed or constructed by the somatized brain, whether or not the brain in vat is actually possible. Zajac suggests that body movement is not entirely determined at the level of the brain but rather is reorganized by the design and flexibility of muscles and tendons, relationships with other muscles and joints, and previously activated memories (Zajac [147]).

2.3.2 Embedded Cognition

Embedded cognition sees that the source of cognitive process is rooted from the environment and world rather than human subject (Beer [13], Rowlands [125], Uexküll [138]). It is a more radical theory of four embodied cognitive theories.

Suppose you asked people how to get to the Louvre museum. Some people use Google Maps and others will ask for help from the people around them. People solve problems in ways that are appropriate for them in a given situation. Embedded cognition emphasizes that human intellectual activity depends not only on the individual's cognitive representation but also on the clues of tools and environments, and on interactions with people (Clancey [29]).

When looking at something, our sensory information does not accurately interpret visual information. This problem is called the problem of under-determination. To solve this problem, cognitive scientists introduced the unconscious inference method. The hypothesis of this method is that if we use perception as a cognition and use existing knowledge, the ambiguity inherent in the input sensory stimulus is resolved. The mental model completed through unconscious inference is the basis for planning and taking action. This reasoning soon becomes an idea and is placed between the senses and the actions (Hurley [63]).

However, in the theory of embodied cognition, the sensory-thinking-action cycle is replaced by the cycle of sensory-action. In the cycle of sensory-thinking-action, an agent passively accepts information. But in the cycle of sensory-action, the agent constantly

explores the world and actively receives information (Clark [30], Brooks [21], Pfeifer and Scheier [110]).

For example, not only bats can detect moving objects during rapid flight in tight spaces, but also they can respond to changes in their position in seconds, due to interactions in dynamic environments.

Agents change the world through actions, and this changed world actively responds to the environment through feedback that affect the future of the agent (Beer [13], Clancey [29], Pylyshyn [118]). Gibson proposes the ecological theory of perception, suggesting that the purpose of perception is to produce affordance (Bornstein and Gibson [17]). Affordance affects the characteristics of the agent's body to produce an unique response, which means that it is important to consider the characteristics of the body to comprehend context-dependent information. Uexküll, on the other hand, created the term 'Umwelt(From the german Umwelt meaning environment or surroundings)' to represent the original way in which individuals relate to the world through perception (Uexküll [138]) .

To describe Umwelt, you must describe both the world and the characteristics of the agent. As mentioned above, feedback depends on the characteristics of the body. Therefore the body of the agent constrains the environmental world that it experiences and the affordance as a way of reacting to the world. The greater the degree of embodied, the more feedback the body receives, and the size of embodied depends on how much the environment can change .

Conventional cognitive science has assumed that the mechanism for controlling behavior is within human beings, but in embedded cognition, the mechanism that underlies behavior is derived from the outside (Dawson [38]). Approached cognitive theory sees that the root of human cognitive process is planted in the environment and world rather than human beings.

2.3.3 Enactive Cognition

The word "enaction" means to perform or carry out an action. The fact that cognition was enacted shows that cognition is formed through the interaction between the organism and the environment. It means that cognition actively creates experiences through the brain's neural processes. If we use the term embodied as radically as possible, we can see that cognition is constitutively dependent on our living body. Based on the creation behavior approach, the living body is understood as an autonomous system (Varela et al. [139]).

The key features in this theory are the body and autonomy. One of the fundamental characteristics of the living body is self-individuation which allows the body to distinguish itself from the surroundings through the individualization process. In other words, self-generation or autopoiesis (self-production) maintains self through structural and functional endless changes of self-individuation. The enactive cognitive theory focuses on how the body

2.3 Four aspects of embodied cognition theory

self-individualizes in this way. It is distinguished from other embodied cognitive theories in that it emphasizes autonomy. Autonomy here is based on Maturana and Varela's theory of self-generation (Maturana and Varela [83]). This concept is called autopoietic enactivism (Campbell [22]).

Self-generation is a special aspect of living organism tissue. In other words, living organisms interrelate with each other during the ongoing process of exchanging energy with the world, the process of complete internal change of metabolism and metabolic processes. In spite of any possible variations, the same organization is constantly being recreated through their own activities. Living cells are an example of an operationally closed network (Di Paolo and Thompson [39]).

A closed, incomplete and endangered operating system makes itself possible. This is in part to support oneself through the activities of his own construction process. Moreover, in this process the system naturally decays. The system, in spite of its natural inclination, needs fundamentally restless energy, matter, and relations with the outside world to sustain itself.

Enactive cognition emphasizes the dynamic interaction between the organism and the environment above all else. It is argued that through this process cognition is formed and finally manifested as action. Thus, the theory of self-generation produced by Varela et al. Does not limit the concept of an organism to humans but to animals and even bacteria. To some extent, depending on the species, they emphasize that they create meaning through interactions with the environment in which they are communicating and situated in.

2.3.4 Extended cognition

In extended cognition the agent's mind is neither skull-bounded nor body-bounded. Rather, it is extended into the world in which the actor lies and lives. The extended cognitive theory goes a step further in asserting that cognitive and environmental processes lie in causal dependency relations. It is assumed that cognitive processes lie in constitutive dependency relations with brain, body, and environmental processes. That is to say, in this theory, the cognitive process is placed in the realized proceeding not only by the brain but also by the rest of the body and the world (Clark [31], J Chalmers [65]).

Clark and Chalmers present cognitive equivalence arguments and coupling arguments as the basis for extended cognitive theory. Cognitive equivalence arguments suggest that cognition expands when processes in the brain, body, and world are very similar to processes in the brain. The coupling arguments insist that human cognitive processes can be realized by the processes of the brain, body and the world, provided that the two processes, physical or environmental processes and cognitive processes, are causally and correctly connected.

Cognitive Equivalence arguments

Inga-Otto thought experiments are a good example of explaining cognitive equivalence arguments. Today, Inga and Otto receives a news of art exhibition taking place at The Grand Palais des Champs-Élysées. Inga left right away because she remembered the location of the museum. Otto, on the other hand, has early stages of Alzheimer's disease. So he always recorded information on his computer in order to don't forget something. Otto moved to the showroom after checking the information from the computer. In both cases, the only difference is whether the information is recorded in the brain or on the hard disk. The essential causal dynamics relations between two pieces of information are exactly the same.

Certainly, Inga's information(Brainy) contains non-derived and original contents, while Otto's information(documentary) contains derived and unoriginal content. Therefore, they are criticized for not being equal. Derived contents come from social practices such as red traffic lights and white flags. It is therefore argued that it is not important to distinguish between derivatives in the cognitive state.

Furthermore, we will look at two types of memory: the phenomenon of negative transfer and the generation effect. In the first phase of the phenomenon of negative transfer, subjects see a pair, such as the name of a man (A, Vincent) and his wife's name (B, Clothilde), and then practice asking them to answer B when asked about A. The next step is to transfer the pair of A to C(C, Jean-Christophe) rather than B. Subjects find it difficult to respond to paired A-C that is not significantly related than paired A-B, which is relatively important. According to this hypothesis, Inga shows performance similar to that of other subjects, while Otto does not. Otto recorded the A-B pair in the first step and the pair A-C in the next step. Knowing the pair A-B stored on the computer does not interfere with the process of knowing the pair A-C. Unlike the subjects and Inga, Otto does not exhibit the phenomenon of negative transfer.

On the other hand, we will perform the task of memorizing pair A-B to carry generation effect experiment out. One group memorizes the word itself, such as Monkey-Banana, while the other group memorizes a pair of words in a sentence, such as Monkey eats a banana. As expected, the task performance increases when memorizing word pairs in a sentence. Let's apply this to Inga and Otto. Inga does a better job in memorizing pairs of words in sentences whereas Otto does not. Otto does not show any difference in the performance of the task due to the setting conditions of the generation effect experiment. In the above two experiments, the claim that Inga and Otto's information is the same in importance or relevance may be somewhat exaggerated or misleading. Thus, extended cognitive advocates suggest that they do not necessarily have to assert strong positions. If Inga and Otto suggests that they are 'somewhat' identical, the cognitive process can be extended to some extent.

Coupling Arguments

Let us look at the coupling arguments. As if we use pencil and paper to solve complex multiplication problems, people tend to rely on the tools from the surroundings. Because human organisms are connected to external entities. In other words, if process X has a reliable causal effect on cognitive process Y, one can infer that the whole X-Y is legitimately cognitive. In other words, the binding action in the brain as well as the binding action in the environment can be reliable.

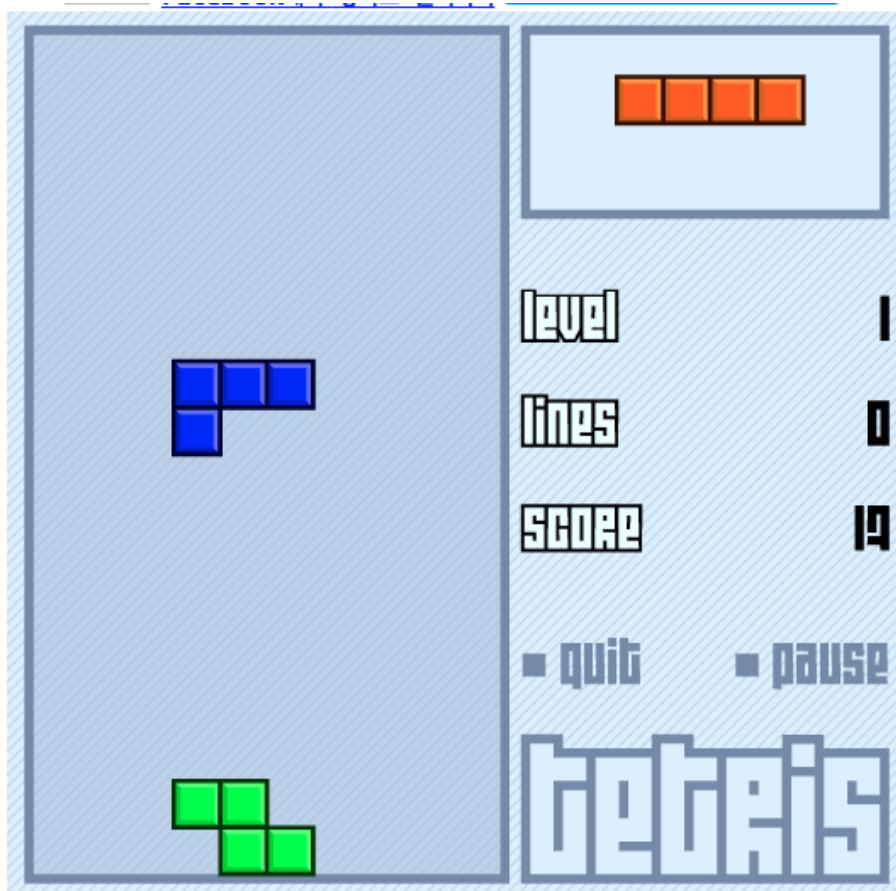


Fig. 2.2 An example of coupling arguments - Tetris Game

Let us take the famous game Tetris as an example (Figure 2.2). The next Tetris shape should be rotated 90 degrees to fill in the empty space on the horizontal line. And we have to think about where to put the next shape. We can imagine in our head what the shape would look like if we rotate it, but we can also rotate it and see it on the screen.

In the extended cognitive theory, there is no difference in cognition between the perception directly into the head and the game of watching the rotation pattern while checking on the

computer screen. Problem solving with the help of a computer screen is also a cognitive process.

2.4 Combination of Artificial Intelligence and Embodied Cognition

Artificial Intelligence(AI) emerged with the advent of computers in the 1950s and has continued to evolve. In the early thirty years, artificial intelligence was philosophically based on rationalism. AI tried to implement intelligence through knowledge program (Charniak and McDermott [25]). This is called the symbolic AI paradigm.

Then from 1990 to the present we call the second Connectionist AI paradigm. This period is philosophically based on empiricism. It took a methodology to implement intelligent systems by learning from data. This second paradigm is developing rapidly, especially in recent years, through Deep learning. Deep learning has the advantage of solving complex problems well, but it requires a lot of training data and is difficult to interpret the model. Symbolic Artificial intelligence(AI) model, on the other hand, is easy to interpret but has the disadvantage of poor learning.

Artificial intelligence(AI) technologies such as machine learning and deep learning may be merely the ability to classify after pattern analysis. But Artificial intelligence(AI) technologies quickly classify and analyze massive amounts of data that humans cannot handle. Computer computing power never lags behind the human brain. But we need an algorithm that translates this computing power into intelligence. Artificial intelligence(AI) is part of the effort to find this algorithm.

In the age of symbolic AI, artificial intelligence was developed by human logic, while in the age of connectionism AI, AI algorithms were automatically developed from data by machine learning. Connectionism is a way for machines to learn data passively, but its intelligent is limited to a certain level. Therefore, the third AI paradigm should be an artificial intelligence as a brain-like cognitive system. It is a reinterpretation of artificial intelligence within the theoretical frame of embodied cognition that appeared in the 1980s and 90s.

Rodney Brooks, a pioneer of embodied artificial intelligence, said that by excluding internal representation, he could build a faster and more robust robot. Brooks studied robots that can move and navigate similarly to insects that exclude internal representation. According to Brooks, intelligence is defined by the dynamics of interaction with the world.

The third cognitive AI is an autonomous cognitive system in which a machine has sensors and motors and interacts with the environment to create learning data itself. This means

that there is no limit to intelligence enhancement, and it provides the basis for reaching true human-level artificial intelligence or artificial general intelligence.

2.4.1 Classical Artificial Intelligence and Connectionism

We buy products we want on the Amazon website and ask for today's weather via voice recognition Siri. Amazon kindly presents similar products and interesting shopping lists at the bottom of the screen, and the smart Siri also notices if my voice cracks because of cold. AI is now deeply embedded in our daily lives. In addition, artificial intelligence adopting machine learning is being evolved by applying to natural language processing, speech recognition, image recognition, etc.

Artificial Intelligence (AI) is a technology that enables machines to learn from experience, to adjust existing knowledge based on new inputs, and to perform tasks in a human-like way. In other words, the ideal artificial intelligence can be said to be the implementation of human intelligence. AI does not have to resemble human intelligence, but human intelligence is still regarded as an ideal model to overcome its weaknesses and limitations. The symbolic AI mentioned above has a top-down approach. The symbolic AI is the main research subject at the high-level to clarify the cognitive function and its associated algorithm.

Symbolic AI is considered to have a constituent structure in which cognitive states are associated in part-whole. Symbolic AI, for example, sees our mental states organized in part-whole relations, from simple mental states to complex mental states, in a way that corresponds to the logical combination of sentences. It has syntactic properties which are logical or formal features that form the basis of inference between mental states, forming part-whole relationships. However, this classical artificial intelligence contributed to explaining higher cognitive capacities such as propositional inference, but showed a limitation in explaining lower cognitive capacities such as pattern recognition. That's why Connectionist AI was sought.

The connectionist AI paradigm takes empirical methodology. In particular, we will comprehend cognition based on neurophysiology based on the specific understanding of the human brain. Just as our cognitive system improves cognitive ability through repeated experiences and learning, connectionism tries to raise cognitive ability through repeated learning, constructing artificial neural networks similar to the human brain. In classical AI, there is a part-whole structural association between cognitive states. This structural association determines the inferential association between cognitive states. However, connectionism denies this structural connection between representations. Each cognitive and representational state is only one physical state, which is composed of a combination of neurons.

Reasoning in connectionism is a causal process. This process is described in a mechanical way of electrochemical reactions between neurons. Also, in connectionism, learning and the subsequent improvement in ability are explained that they are caused by the connection strength of neurons. Connectionism has since been studied in machine learning methods such as decision trees and Bayesian nets and has evolved into deep learning.

2.4.2 Embodied Artificial Intelligence

Rodney Brooks, one of the pioneers of embodied intelligence, developed 'Ghengis', a robot that moves and navigates like an insect. He introduced the concept of action-based subsumption architecture. Ghengis was working to prove the subsumption architecture. Ghengis' central processing unit does not control all actions.

The ant-shaped Ghengis has six legs and a sensor attached to each leg. These sensors cause other legs to react. In other words, even without a central processing unit acting as a brain, it can elicit a response to a specific action. These activities have become an important research area for sensory-motor intelligence at the low level. Thus, the embodied approach is based on constructivist methodologies that form development and learning through interaction with the environment rather than assuming a given one.

This constructivist methodology is in line with the embodied cognitive system theory that emphasizes the agent's interaction with the environment with the body (Varela et al. [139], Barsalou [10]) and with the perceptual behavior system theories which states that cognition is based on the behavior of the environment (Prinz et al. [116]). In this sense, the concept of free will and self can be interpreted as being sociocultural through interaction with the environment and others (Prinz [115]). The objectives of the embodied approach are firstly based on understanding biological systems, secondly by attempting to abstract intelligent behavior into general laws, and thirdly by the application of knowledge to create robots or general intelligence devices.

First, let us look at the development of robots using the embodied approach. Robotics research has an extensive sensory motor repertoire, and has the advantage of studying more and more interrelationships with the outside world.

For example, Let us consider a recent study of humanoid walking robots. Even though it is slower than humans and there are many unnatural aspects of working styles, it has been a remarkable achievement (Research on humanoid robots such as Honda's Asimo and Kawada's HRP). Furthermore, the University of Oslo in Norway recently introduced the Dynamic Robot for Embodied Testing (DyRET) using embodied cognition. They are inspired by four-legged mammals, therefore they are designed to move on their own.



Fig. 2.3 DyRET: Dynamic Robot for Embodied Testing
(source:<https://www.digitaltrends.com/cool-tech/dyret-robot-learns-to-walk/>)

Of course, the artificial intelligence we ultimately want to realize is high-level intelligence. Rather than low-level intelligence based on behaviors under the sensory motor system, higher levels of intelligence, such as problem solving, reasoning, natural language understanding, emotional processing, and perception, should be realized. The question then is whether artificial intelligence can process symbols like humans. How can organisms acquire meaning from the world? How can they create symbolic grounding problems based on what they have learned? Where can I find the answer to these questions? We must find the answer not only in terms of learning but also in terms of embodiment.

We will look at the case of image recognition using visual information. As in the case of objects or facial recognition, we have the ability to very accurately perceive a distorted image despite light, distance, and direction very accurately. Even under the shining sun or at dark nights, we can accurately identify the faces of our families. This is because the agent does not simply accept the input vector as it is. Changing stream of sensory stimulation is closely related to the agent's current behavior (Pfeifer and Iida [109]). Through interaction with the physical environment, the agent derives and demonstrates sensory simulation (Pfeifer and Scheier [110]). In this way, embodied artificial intelligence will be able to solve and recognize problems in the "real world" by placing the interaction between objects and the environment at the core. Embodied AI is an AI paradigm that can deal with real-time online situations in which reasoning and learning are cyclically linked.

2.5 Conclusions and the future

We have previously examined the philosophical and theoretical background of embodied cognition. We also compared the four theories of embodiment (embodied cognition, embedded cognition, enactive cognition, extended cognition) with each other to see how embodied cognition would be implemented in artificial intelligence. The human cognitive process has only been understood as a mechanism for acting on a given object, but the new cognitive science has proposed the theory of enaction through embodiment. Since human beings are self-generate cognitively, they will be able to demonstrate better their cognitive creativity on the basis of embodiment.

In this paper we will examine the correlation between cognition and gestures in the problem solving process. We will apply the theory of embodied cognition to simulate this problem solving process. In particular, I would like to examine the reasoning in the process of problem solving, visual information and gestures, and understanding of language from this perspective. We would like to apply deep learning methods from an embodied perspective.

Chapter 3

Neural Networks and Deep Learning

In the previous chapter, we introduced the concept of embodied cognition. In this and the next chapter, we will look at how humanized cognition can be combined with artificial intelligence to understand human inference and cognition.

In particular, according to the cognitive embodied approach, the cognitive process is deeply related to the body's interaction with the environment, which has a great influence on our thinking, reasoning and decision making. We intend to simulate this process of inference through neural networks and deep learning.

In Chapter 5, we are going to do an experiment to solve the Tower of Hanoi puzzle problem to observe the effects of behavior in the cognitive process. And based on this experimental data, We will try to solve the puzzle problem using neural network modeling in chapter 6. First of all, we will solve the problem with typical multi-layer perceptron modeling. And then, we will present an inference solution that can figure the rules of the game.

In chapters 7 and 8, we want to focus on the behavior of the actor to find the solution of the puzzle. To do this, we intend to approach problem solving by using a Recurrent Neural Network and Convolutional Neural Network. Lastly, the chapter 9 is intended to simulate how language learning is possible based on the recognized objects and rules. Also, on the contrary, based on language learning, we also want to realize whether it is possible to learn a cognition and rules of games. Therefore, in this chapter, we would like to mention the neural network and deep learning which are the entire theoretical backgrounds.

3.1 Introduction

We remember that the AlphaGo program developed by Google's DeepMind had a historic victory against South Korean Go grandmaster Lee Sedol in April 2016. This event was a landmark achievement for an Artificial Intelligence.

AlphaGo used machine learning to learn Go's rules and strategies. It is known that the number of choices in the game of Go is much higher than the atoms in the universe. So in the case of one's playing, one's intuition plays a significant role. After the intuitive judgment, the Go players are searching for the best strategy. AlphaGo reduced the number of cases using Tree search. Also in order to find the optimal solution, AlphaGo used 1202 CPUs and 176 GPUs. AlphaGo also used two artificial neural networks to think like humans: Policy network and value network. Now, AlphaGo has gone a step further to develop universal artificial intelligence after confronting humans. As such, artificial intelligence is approaching the human realm, and more than ever, interest in artificial intelligence and deep learning has become more active.

As we saw in the second chapter, connectionism is an approach to the study of human cognition that explicitly employs some of the mechanisms, known as connectionist networks or neural network models (Adorf [1]). The neural network models, starting from the anatomy and physiology of the nervous system, are able to learn and their behavior improves with training or experience.

The history of artificial neural network climbs back to the 1940s. Neuroscientists learn that what neurons do is similar to the binary circuits that make up a computer, and they begin to design artificial nerves called perceptrons (Rosenblatt [124]). This perceptron is the origin of neural networks (deep learning). In 1958, the New York Times published an article about a world where robots would soon be able to walk, speak, and recognize self. ¹

Contrary to expectations, perceptron does not solve most problems, and neural network technology is in a recession for some time. But with the advent of Multilayer Perceptron technology and the back propagation algorithm, artificial neural network technology is facing a new phase.

Marvin Minsky shows that we can solve the XOR problem by constructing a neural network with a Multilayer Perceptron(MLP). However, it was impossible to calculate the weight of hidden layers in this model. In other words, there was no way to learn Multilayer

¹ The New York Times
Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser(July 7, 1958)
... Dr. Rosenblat, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptron might be fired to the planets as mechanical space explorers. The Navy said the perceptron would be the first non-living mechanism « capable of receiving, recognizing and identifying its surroundings without any human training or control » ...

3.2 The simple perceptron

Perceptron at that time. To solve this problem, the concept of back propagation of error can be introduced to train the Multilayer Perceptron(MLP).

However, the learning of MLP in a multi-layer structure can be learned through this Backpropagation. Experience has shown that MLP requires more hidden layers of artificial neural networks for better results. However, the higher the number of hidden layers make you are faced with gradient vanishing, which makes it impossible to calculate the weight due to Backpropagation.

We then proved that it is possible to learn MLP, no matter how many layers of the neural network are, if we choose the initial input well. The methodology suggests that even complex problems can be solved with a large number of hidden neural networks. Such a neural network with a large number of layers is called a deep neural network, and a method of learning a deep neural network is called Deep Learning.

With the development of big data in the 2000s, the study of artificial neural networks became more vibrant. Computing resources are getting better, data is more abundant, and there are ways to solve the problems that arise when learning. It is also becoming possible to solve real-world problems such as recognizing, speaking, or translating objects. This chapter examines the perceptron and neural network related theories that are the foundation of deep learning.

3.2 The simple perceptron

The field of neural network was firstly initiated as mathematical models of the information processing by investigating networks of interconnected neurons, which was based on real biological neurons. In particular, the simple perceptron has a set of inputs and an output layer, connected to each other by weighted connections. With regard to the biological neuron (Figure 3.2), the input and output layer represent neurons, and the weighted connections represent the synapses between the neurons. Therefore, we can define this model as follows

Consider a linear threshold unit with n inputs $x_1, \dots, x_n \in \mathbb{R}$ and $n+1$ weights w_0, w_1, \dots, w_n . Letting $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ and $\mathbf{w} = (w_1, w_2, \dots, w_n)^T \in \mathbb{R}^n$, we have

$$y = \phi(\mathbf{w}^T \mathbf{x} + w_0) = \begin{cases} 1 & \text{if } \mathbf{w}^T \mathbf{x} + w_0 \geq 0 \\ -1 & \text{if } \mathbf{w}^T \mathbf{x} + w_0 < 0 \end{cases}$$

The goal of the simple perceptron is to decide whether a set of input belongs to one of two classes, \mathcal{C}_∞ or \mathcal{C}_ϵ . If the output is 1, the pattern will be assigned to the class \mathcal{C}_∞ . If the output is -1 , the pattern will be assigned to \mathcal{C}_ϵ . The two classes \mathcal{C}_∞ and \mathcal{C}_ϵ are separated by

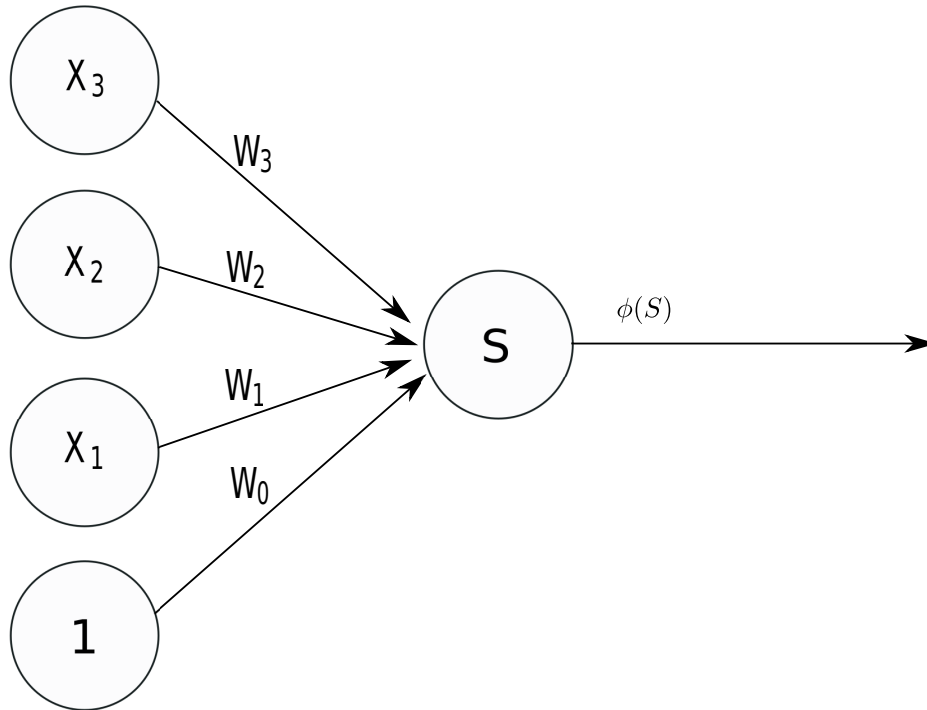


Fig. 3.1 Model of the neuron (perceptron)

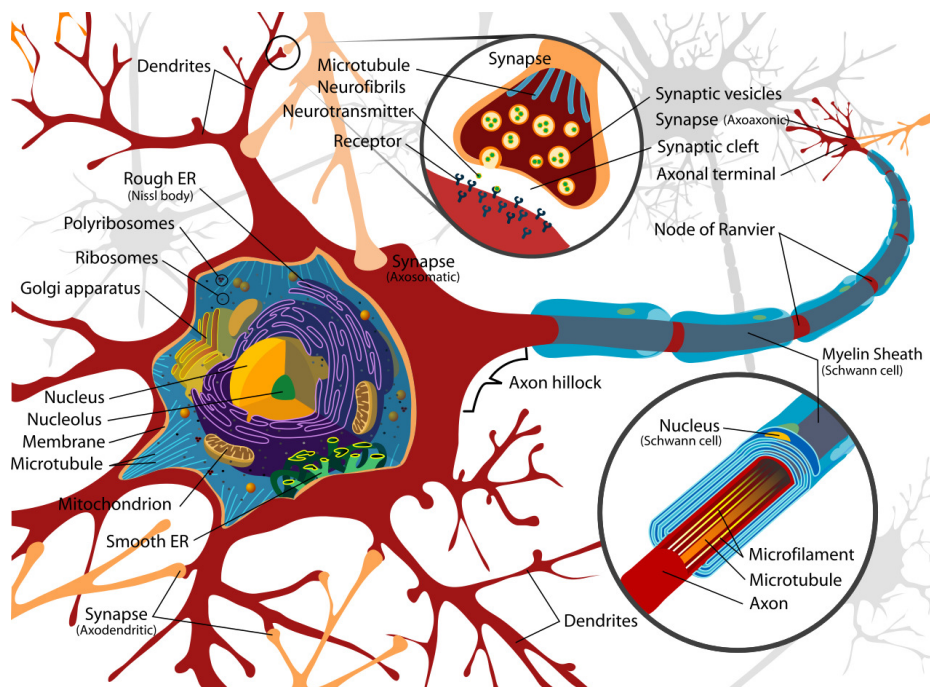


Fig. 3.2 Biological neuron (source: www.wikipedia.org)

3.2 The simple perceptron

a straight line. Let inputs $\mathbf{x} = (x_1, x_2)$ and weights $\mathbf{w} = (w_0, w_1, w_2)$. separation between the two classes is given by the equation:

$$w_1 \times x_1 + w_2 \times x_2 + w_0 = 0 \quad (3.1)$$

$$x_2 = \frac{w_1}{w_2}x_1 - \frac{w_0}{w_2} \quad (3.2)$$

We draw a geometric representation of perceptron from equation (3.2). We know that the weights changes the slope of the line and the bias determines the 'offset'.

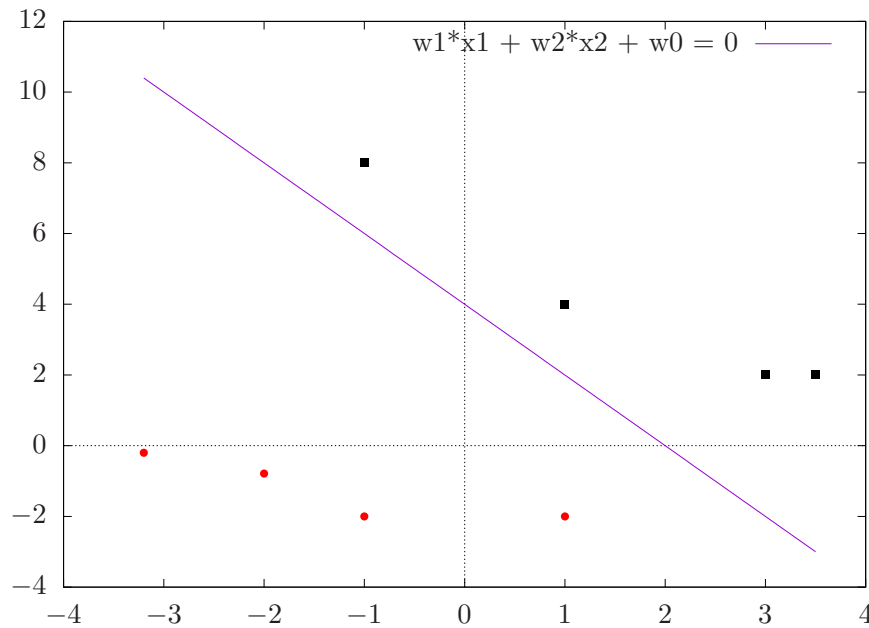


Fig. 3.3 Geometric representation of Perceptrons

In fact, the deep learning we use today has the same fundamental structure as Rosenblatt's basic perceptron. The only difference is that it extends to multiple nodes and multiple layers. Frank Rosenblatt's perceptron at that time received much of the same expectations and attention from academics and the media as today's deep learning. Many hoped that Perceptron could create artificial human-like intelligence. But in 1969, Marvin Minsky and Seymour Papert mathematically proved the limitations of Perceptron in a book called "Perceptrons: an introduction to computational geometry" (Minsky and Papert [91]). Since then, the expectation and enthusiasm for artificial intelligence have rapidly decayed. Minsky and Papert pointed out that Perceptron is a simple linear classifier and cannot perform even simple XOR classification (See Figure 3.4).

Perceptron enabled the implementation of gates such as AND, OR, and NAND. However, in the case of XOR gates, the perceptron is impossible to implement by alone itself. Percep-

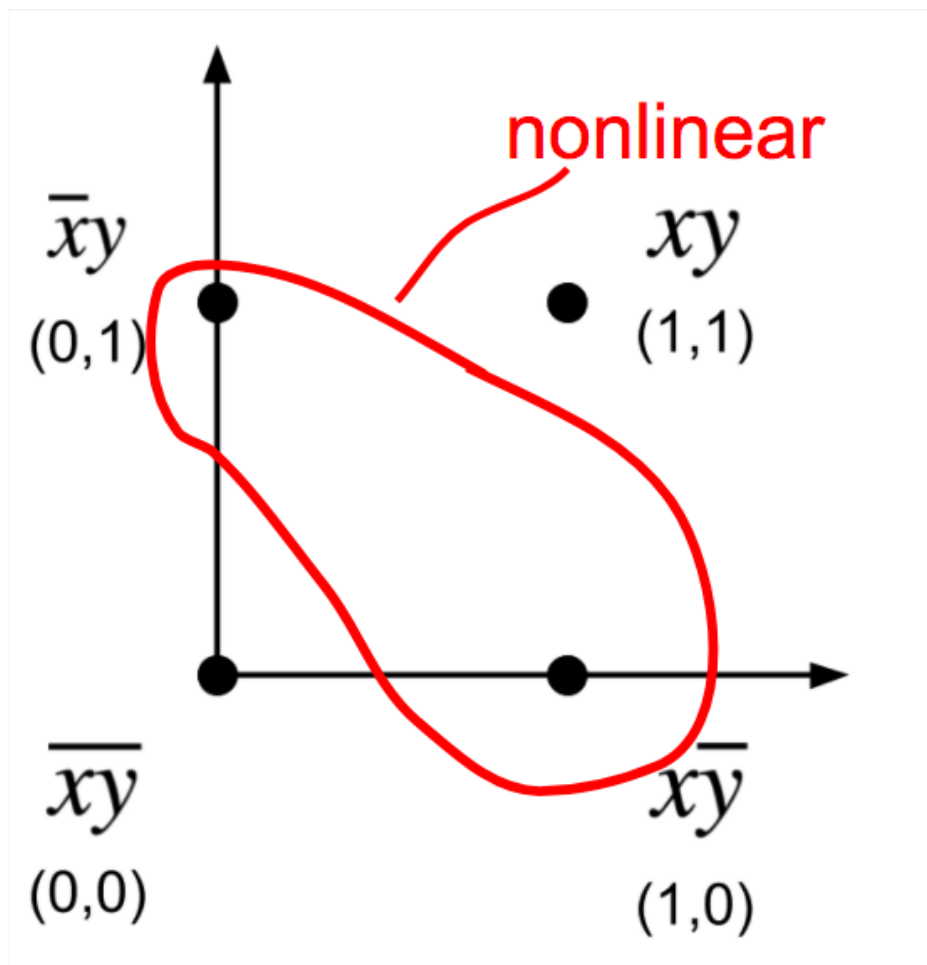


Fig. 3.4 A single Perceptron's Limitations

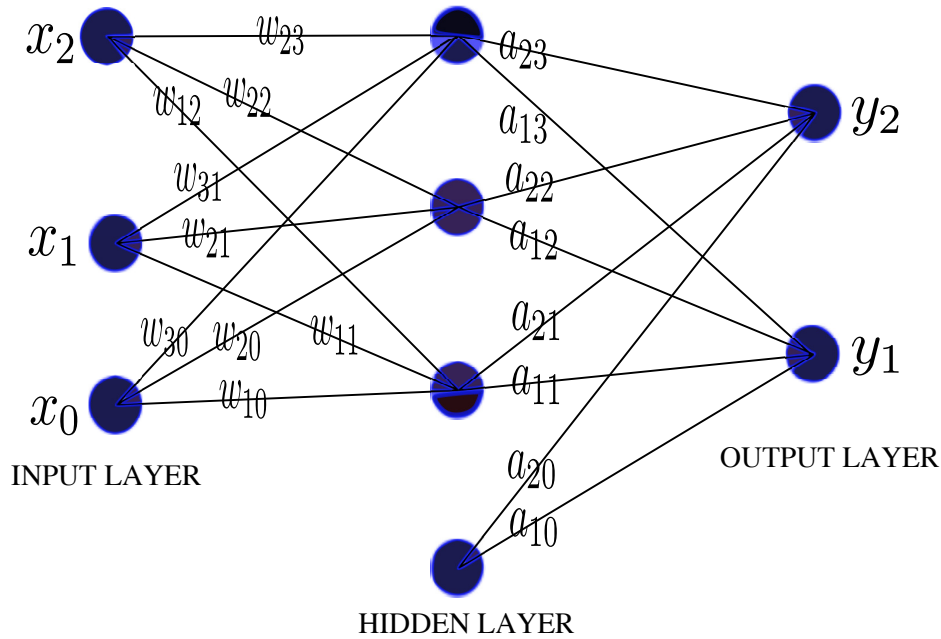


Fig. 3.5 Multilayer perceptron

tron divides the area by a straight line because the XOR gate does not divide the area by a straight line.

3.3 Backpropagation algorithm

In 1986 Geoffrey E. Hinton et al. succeeded in learning multilayer perceptrons with error backpropagation algorithms. The simple perceptron could calculate the error from the desired result and adjust the weight proportionately. In contrast, multilayer perceptrons have multiple hidden layers between the input and output layers. The outputs of nodes belonging to these layers do not know how to adjust the weights because there is no basis for measuring the error. However, the backpropagation algorithm is able to readjust the weights between the nodes of the hidden layer while sending the error occurring at the output layer from the output layer to the input layer. Backpropagation algorithms have proven to be able to train multi-layered perceptrons with many hidden layers. This is the basis of the algorithm for learning neural networks. We will briefly look at the error backpropagation algorithm below.

The model in Figure 3.5 is a simple feedforward neural network with an input layer on the left side, a hidden layer and an output layer. In the figure, x_i is the input of the model, z_i is the output of the node in the hidden layer, and y_k is the output. x_0 and z_0 are biases of the input layer and the hidden layer, respectively. w_{kj} means a weight connecting node j and

node k . Firstly, when the pattern is an input to the model in the figure 3.5, the output value of k of the output node (y_k) is expressed as follows.

$$y_k = \sigma \left(\sum_{j=0}^M w_{kj} h \left(\sum_{i=0}^D w_{ji} x_i \right) \right) \quad (3.3)$$

In this equation, $\sigma(\cdot)$ and $h(\cdot)$ are activation functions at each node and sigmoid function is used. However, for the output layer, the softmax function is used. The inputs to the hidden and output layer nodes are called a_j and a_k . The formula from input to output is as follows.

$$a_j = \sum_{i=0}^D w_{ji} x_i \quad (3.4)$$

$$z_j = h(a_j) \quad (3.5)$$

$$a_k = \sum_{j=0}^D w_{kj} z_j \quad (3.6)$$

$$y_k = \sigma(a_k) \quad (3.7)$$

Now, to train the neural network, we set the error function as follows.

$$E(W) = \frac{1}{2} \sum_{n=1}^N \|y_n - t_n\|^2 \quad (3.8)$$

where

y_n : output vector of neural network,

t_n : target vector of neural network, N : number of learning patterns

The goal of the backpropagation algorithm is to calculate the partial derivative of the cost function E with respect to the weight w or bias b in the network. Since $E(W)$ is nonlinear in the extreme multidimensional space, there can be numerous points where the first derivative is zero. In order to optimize this case, the method of iteratively improving the w vector which reduces the value of the error function is as follows.

$$W^{(\tau+1)} = W^{(\tau)} + \Delta W^{(\tau)} \quad (3.9)$$

The superscript τ in the equation represents the iteration step and the improved value k to the next step depends on the algorithm. The simplest form is to use a partial derivative called

3.3 Backpropagation algorithm

gradient descent or steepest descent. This method means that small steps in the gradient direction increase the error and steps in the opposite direction to reduce the error. It is expressed by the following equation.

$$W^{(\tau+1)} = W^{(\tau)} - \eta \nabla E(W^{(\tau)}), \text{ where } \eta : \text{learning rate, } \eta > 0 \quad (3.10)$$

Now let's look at the error backpropagation algorithm. Considering an error function of one pattern among N input patterns, it is as follows.

$$E_n = \frac{1}{2} \sum_{k=1}^C (y_k - t_k)^2 \quad (3.11)$$

In the figure, w_{kj} is the weight connecting node k of the output layer and node j of the hidden layer. The partial derivative of w_{kj} in $E(W)$ is as follows.

$$\frac{\partial E_n}{\partial w_{kj}} = \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial w_{kj}} \quad (3.12)$$

The above equation shows the partial derivative as the product of two partial derivatives using the chain rule. Since a_k is the input value of node k of the output layer, it can be expressed as follows by equations 3.7 and 3.11.

$$\frac{\partial E_n}{\partial a_k} = (y_k - t_k) \sigma'(a_k) \quad (3.13)$$

Then let us rewrite the equation by 3.7.

$$\frac{\partial a_k}{\partial w_{kj}} = z_j \quad (3.14)$$

$$\delta_k \equiv \frac{\partial E_n}{\partial a_k} \quad (3.15)$$

$$\frac{\partial E_n}{\partial w_{kj}} = \delta_k z_j \quad (3.16)$$

Now let's look at the partial derivative for w_{ji} , the weight that connects node j of the hidden layer with node i of the input layer. Therefore, the equation can be expressed as follows by 3.12 and 3.16.

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} = \delta_j z_i \quad (3.17)$$

However, δ_j in 3.17 has the same meaning as δ_k defined in 3.15, but the form is more complicated. Therefore, if δ_j is also applied to the chain rule, the equation is expressed as follows.

$$\delta_j = \frac{\partial E_n}{\partial a_j} = \sum_k \frac{E_n}{\partial a_k} \frac{\partial a_k}{\partial a_j} = \sum_k \frac{\partial E_n}{\partial a_k} \left(\frac{\partial a_k}{\partial z_j} \frac{z_j}{\partial a_j} \right) * / \quad (3.18)$$

Finally, the equation 3.18 can be expressed as follows by using 3.14, 3.15, and 3.15

$$\delta_j = h'(a_j) \sum_k w_{kj} \delta_k \quad (3.19)$$

The backpropagation algorithm computes the partial derivative of E_n by continuing to pass δ_j in this way. The backpropagation algorithm is now summarized as follows.

1. Input pattern x_n is put in the neural network's input layer and the value is forwarded according to 3.4 and 3.5. In the process, it stores z_j which is an output value of each node.
2. δ_k for the node of the output layer is calculated using 3.15.
3. Calculate δ_j for nodes in the hidden layer by forwarding the delta backwards.
4. Compute the partial derivative using 3.19.

As shown in Figure 3.5, MultiLayer perceptron (MLP) is a feedforward neural network. The units in every layer (Input, Hidden and Output) are connected via feeding forward from one layer to the next. This model is trained with the backpropagation learning algorithm. MLP is widely used for speech recognition, pattern classification, and so on.

3.4 Recurrent Neural Networks

Recurrent neural network(RNN) models are suitable for processing sequential information, like time series data such as stock market and macroeconomic analysis, as well as language and speech recognition.

The feedforward neural network discussed above was a neural network flowing in only one direction from the input layer to the output layer. In contrast, RNN (circulating neural network) is similar to feedforward neural network, but the output is written again as an input data as shown below (Figure 3.6). This is called a recurrent connection, which is very interesting.

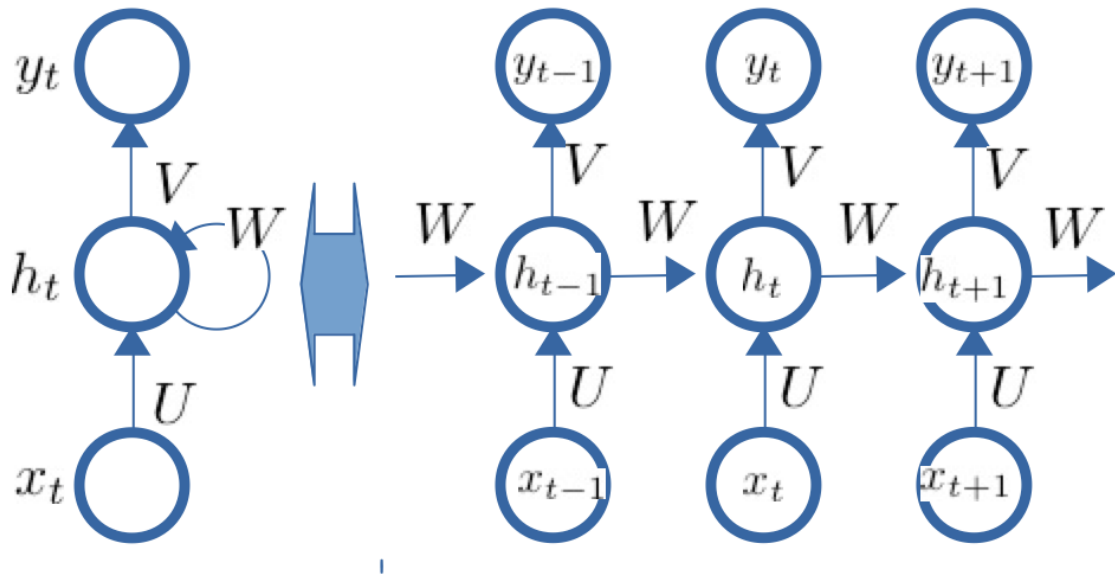


Fig. 3.6 Recurrent Neural Network Architecture(RNN)

For a better understanding, let us look at an example from Goodfellow et al (Goodfellow et al. [52]). Here are two sentences: 'I went to Nepal in 2009' and 'In 2009 I went to Nepal'. We want to use machine learning to find out which year I traveled to Nepal. Whether or not the year 2009 is at the end of the sentence, we know that we traveled in 2009. The feedforward model learns differently depends on the year position. Unlike the feedforward model, the RNN model shares the same weight across multiple time steps. Thus, the RNN model does not matter whether the year 2009 is in the head or the back of sentences.

RNN is a model on the left side of the figure 3.6. Given the input data(x_t) of t time, the RNN is multiplied by the weight(U) and input to the internal node. The internal node creates a state value by adding information which has previously been stored. The state value h_t of the internal node at t time is expressed as the following expression.

$$h_t = f(Ux_t + Wh_{t-1}) \quad (3.20)$$

The function f uses sigmoid or ReLu as the activation function. The output of the model at t time is generated by multiplying the internal node's state value by V and in some cases, applying the softmax function.

The model on the left side of the figure 3.6 looks simple at first glance, but the re-entry of previous information from the internal node to the loop is different from the forward neural network, which complicates the operation of the model. Let us draw the RNN as it progresses over time to make the model easier to comprehend. That is the model on the right side of

the picture. If you unfold the model like this, you can say that it is a forward neural network with three inner layers (h_{t-1} , h_t , h_{t+1}).

The difference is that the input data is put in as different internal nodes and the weight W connecting the internal layers is the same. In the expanded structure on the right side of the figure 3.6, the number of internal nodes is depicted as three, but the number depends on the number of inputs. For example, if you enter 10 words in a sentence, 10 internal nodes will be created.

RNN frequently uses cross-entropy error functions. If the target value at time t is t_t and the output of the model is a_t , the error value at t time is: The error is $\sum_t E_t$.

$$E_t(t_t, y_t) = -t_t \log y_t \quad (3.21)$$

The method of learning weights in RNN is to apply error backpropagation algorithm to the unfolded model. However, it is necessary to add a partial function for overlapping weights. This method is also called Backpropagation through time (BPTT). Now let's look at the calculation method the first-way derivative for W , the weight that connects the inner layers. The linear derivative function for W across the model is as follows.

$$\frac{\partial E}{\partial W} = \sum_{t=1}^T \frac{\partial E_t}{\partial W} \quad (3.22)$$

Applying the chain rule using the partial derivative of t time output (y_t) in the expanded model of the figure, the first order of E_t for W that leads to the internal node expressed as h_t is as follows. The partial derivatives are

$$\frac{\partial E_t}{\partial W} = \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial W} \quad (3.23)$$

On the other hand, since W must go back to the internal node of the first stage, the first derivative of E_t for W which is connected to h_1 is:

$$\frac{\partial E_t}{\partial W} = \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_{t-1}} \cdots \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W} \quad (3.24)$$

3.23 and 3.24 can be added together by calculating the partial derivatives of W in the middle stage and using 3.22.

$$\frac{\partial E}{\partial W} = \sum_{t=1}^T \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \left(\prod_{j=k}^{t-1} \frac{\partial h_{j+1}}{\partial h_j} \right) \frac{\partial h_k}{\partial W} \quad (3.25)$$

Similarly, the partial derivatives for U and V can be calculated. However, in the process of calculating the partial function using 3.25, a multiplication of the partial function occurs.

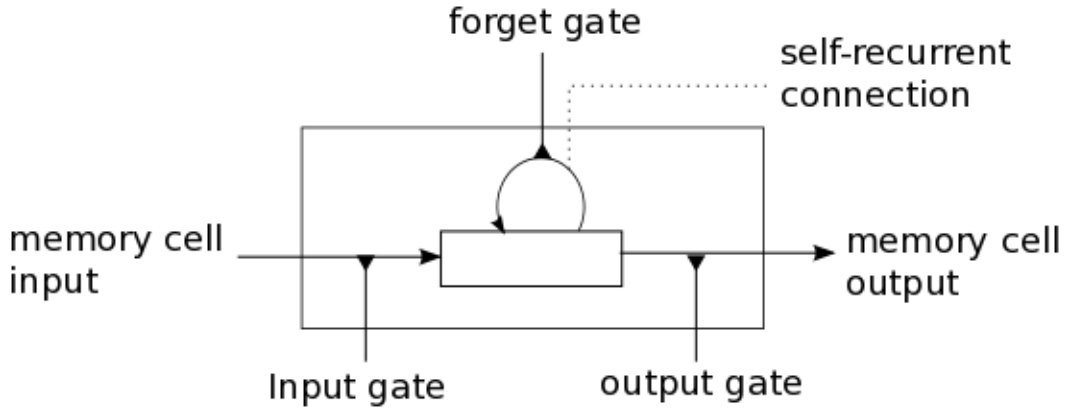


Fig. 3.7 LSTM Memory cell

As the number of sentences input and the number of internal nodes increase, the number of multiplications naturally increases. Multiplying so many values can cause the partial derivatives to be close to zero (vanishing gradient) or very large (exploding gradient).

This can be solved relatively easily by limiting the value of the partial derivatives to be very large, but it is not easy to find a case near zero, which causes the learning speed of the weight to slow down or stop. In addition to the sigmoid function, ReLU can be used as an activation function to reduce the value of the partial derivatives, but a model to solve the vanishing gradient problem was proposed in 1997, which is a long short-term memory (Hochreiter and Schmidhuber [59]).

LSTM changed the internal node of the RNN in the figure 3.7 into a complex structure called a memory cell.

The LSTM memory cell has four components: an internal memory node with cyclic inputs and three switches. Each switch is a value between 0 and 1, such as a logistic function, that controls the value passing through it. The formula to modify the state value of the LSTM memory cell at t time is as follows. Firstly, if i_t is the input gate value and \tilde{C}_t is the input value of the internal memory node, the value is calculated as follows.

$$i_t = \sigma(U_i x_t + W_i s_{t-1} + b_i) \quad (3.26)$$

$$\tilde{C}_t = \tanh(U_c x_t + W_c s_{t-1} + b_c) \quad (3.27)$$

Now, if f_t is the value of forget gate, the new state value of the internal memory node is calculated as follows.

$$f_t = \sigma(U_f x_t + W_f s_{t-1} + b_c) \quad (3.28)$$

$$C_t = i_t * \tilde{C}_t + f_t * C_{t-1} \quad (3.29)$$

Using the new state value, we calculate the value of the output gate o_t and the memory cell output as follows.

$$o_t = \sigma(U_o x_t + W_o s_{t-1} + b_0) \quad (3.30)$$

$$s_t = o_t * \tanh(C_t) \quad (3.31)$$

In the above equation, W , U and V are the weights at the corresponding gate and b is the bias. The type of input/output data of a RNN is generally a sequential model. However, in some cases, non-sequential data may be input or output. Various applications are possible depending on the type of input/output data. When the input / output data is sequential, it is a case where the machine translation or annotation is applied to the video. If the output data is out of order, it can be applied to emotion analysis. The case where the input data is out of order is applied to annotations for pictures.

3.5 Convolutional Neural Network(CNN)

David H. Hubel and Torsten Wiesel conducted a cat experiment that provided crucial insights into the structure of the visual cortex (Hubel and Wiesel [62]).

They have shown that many neurons in the visual cortex have small local receptive fields. This means that neurons respond only to visual stimuli that are within some range of vision.

Receptive fields of neurons can overlap one another, and these overlapping receiving areas form the entire field of view. Additionally, some neurons responded only to images of vertical lines, while others responded to lines of different angles. They also demonstrated that some neurons with large receptive field respond to complex patterns(texture,object) by combining low-level patterns (edge,blob). This observation led to the idea that high-level neurons are based on the output of neighboring low-level neurons.

Receptive zones mean that external stimuli affect only a specific region rather than the entire region. You may find that when you poke through various parts of your body with your fingers, you have a limited range of feeling. Also, the size of the area you feel depends on where you poke.

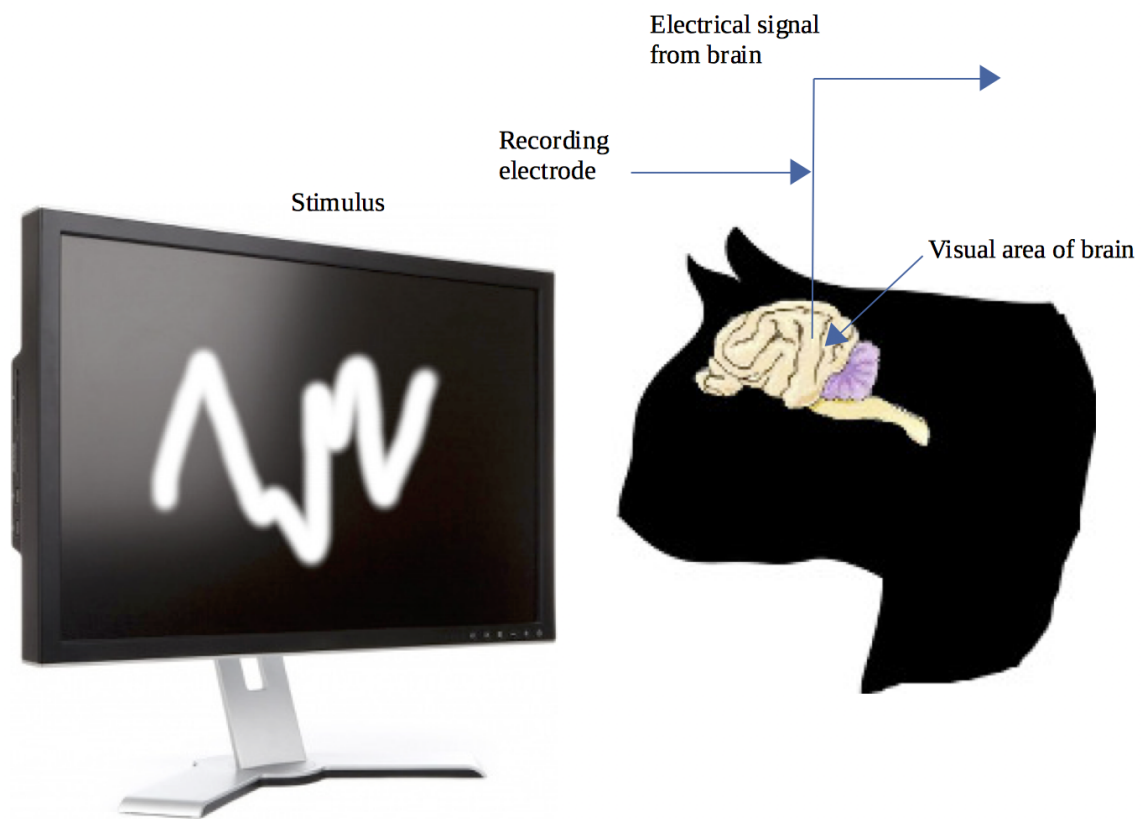


Fig. 3.8 Hubel and Wiesel Experiment

Likewise, pixels at a specific location in the image have a high correlation with only some of the pixels around them. As the distance from the pixels increases, the influence decreases.

If you want to use this algorithm to interpret the video or image (recognition algorithm), you do not need to give the same importance to the entire image. Instead, if you process only a specific range, you will be able to interpret the image more effectively.

This idea has been developed into a convolutional neural network (CNN). Yann Lecun et al. presents a practical paper that combines MLP and Backpropagation Algorithm with Convolution to apply MNIST problems (Le Cun et al. [78]).

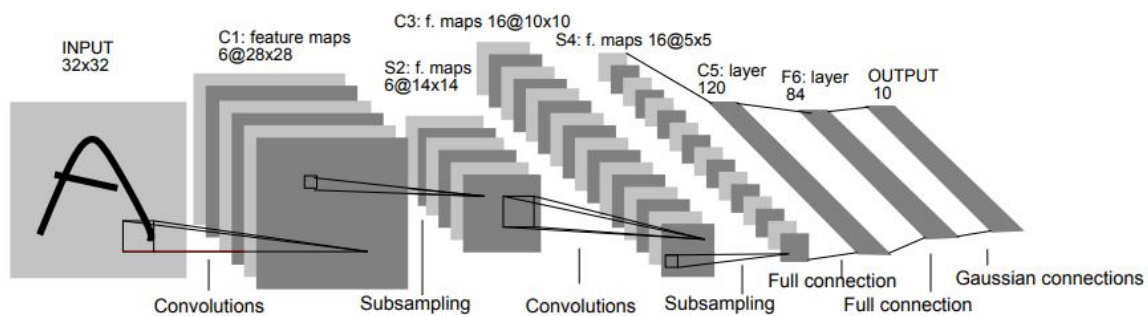


Fig. 3.9 Convolutional neural network structure (LeCun et al. [79])

Then, in 1998, LeCun et al. proposed the Neural Networks structure, which is the origin of the modern Convolutional Neural Networks (CNNs) called LeNet-5 (See Figure 3.9). The LeNet-5 is a practical application in the post office’s check recognition system, demonstrating the strong power of artificial neural networks (LeCun et al. [79]).

More specifically, let’s take a look at how CNN works. CNN can be seen as an integrated model that combines the two steps, feature extraction and classification, into one step in the existing pattern recognition method. The CNN basically consists of a plurality of convolution layers and subsampling layers.

The figure 3.10 describes the key ideas of convolution and subsampling. The convolutional layer generates a feature map by applying various convolution kernels to the input. Convolution acts as a template for extracting features from high-dimensional input images. One convolution targets partial inputs and is repeated several times, changing positions, scanning the entire image.

That is, one convolution has a small number of connecting lines, and the weight of the connection is shared even when the position is changed. This allows convolution to extract features regardless of location. For a convolution, sharing weights despite of change of position, is one of the key ideas to reduce the complexity of the problem by reducing the total number of parameters to learn.

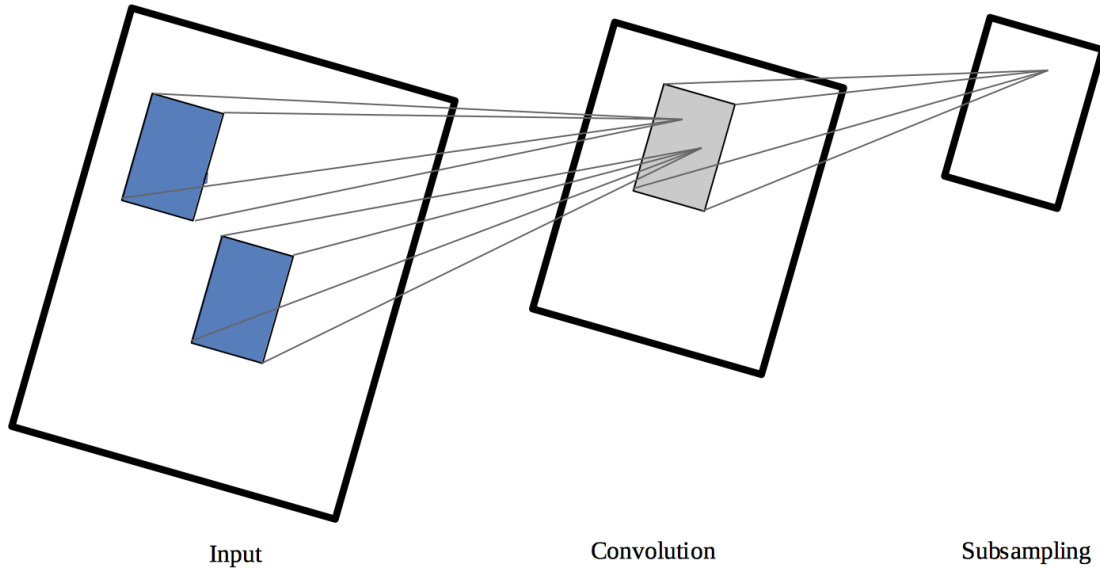


Fig. 3.10 Convolution and subsampling

The subsampling layer is a neuron layer with reduced spatial resolution for the created feature map. Subsampling is the process of reducing dimensions, which also reduces the complexity of the problem. As the subsampling operator, the maxpooling operator that takes the maximum value for the target neurons or the average pooling operator that takes the average value is mainly used. Pooling not only reduces dimensions but also makes the feature map robust to shift and distortion.

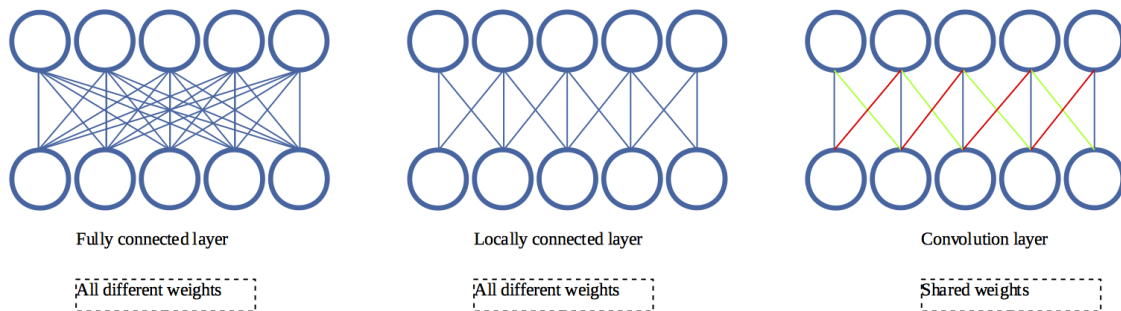


Fig. 3.11 Comparison of fully connected layer, regionally connected layer, and convolutional layers.

The figure 3.11 also illustrates the idea of convolution. We describe how the convolutional layer differs from the fully connected layer and the locally connected layer. In the fully connected layer, there is a connection line between neurons in a neighboring layer. All neurons in one layer and all neurons in a neighboring layer are connected. In other words,

if the number of neurons in each of the two layers is five, there are $5 \times 5 = 25$ connecting lines and 25 weights between the two layers. In contrast, using a locally connected layer, neurons in the upper layer are only connected to some neurons in the lower layer. (See the center figure in Figure 3.11). In the middle figure, the five neurons in the upper layer are only connected to 2–3 neurons in the lower layer, each with $2 + 3 + 3 + 2 = 13$ connections.

The connecting lines of each neuron have different weights. The convolutional layer (right side of Figure 3.11) has the same partial connectivity as the locally connected layer, but upper layer neurons share weights with each other. That is, five neurons in each upper layer share these 3 weights. Therefore, although there are 13 connecting lines, the weight is 3. By doing this we reduce the number of parameters we need to learn. Using such a connection structure, it is possible to extract a portion which has a similar feature with a different position, so that the feature can be extracted regardless of the movement of the object from side to side in the input image.

As a concrete example, let's look at the structure of LeNet5 (Figure 3.9). The model takes a 32×32 pixel image as input, undergoes three convolutions and two subsamplings, and finally determines the output through a fully connected multilayer network.

Convolution uses a 5×5 matrix and subsampling is $1/2$. The overall structure can be described as follows, where C_n represents the n th convolutional layer and s_m represents the m th subsampling layer.

- C1: Constructs 6 feature maps of shape 28×28 using a convolution matrix of 5×5 from a 32×32 image.
- S2: Constructs 6 subsampling map of shape 14×14 using $1/2$ subsampling from each 28×28 feature map in C1.
- C3: Again, we construct 16 feature maps of shape 10×10 using a 5×5 convolution matrix.
- S4: Again, use $1/2$ subsampling to construct 16 subsampling maps of shape 5×5 .
- C5: Constructs 120 convolutions from the result of S4.
- F6: 84 subsampling is done using the fully connected layer.
- RBF: Finally, the output is calculated from 10 RBF neurons.

All neurons before the F6 layer have sigmoid activation functions and are defined as follows.

$$f_i = f(v_i) = Atanh(Sv_i) \quad (3.32)$$

S and A are parameters that determine the shape of the activation function, usually using constants. In the last layer, the output value of the CNN is determined by the output feature map and the image map as follows.

$$y_j = \sum_{i=1}^{84} (f_i - W_{ij})^2, \text{ where } j = 0, \dots, 9 \quad (3.33)$$

f_i are output image feature values and w_{ij} are target image feature values. In this design, the total number of connections is 187000. This corresponds to the number of weights 187384 in a multilayer neural network with a fully connected $(28 \times 28) \times 236 \times 10$ structure.

However, the number of variable weights that are actually learned is 14000, which is much smaller than the total number of connections (10% or less). The structure of the LeNet CNN is not larger than the fully connected multilayer neural network MLP. However, convolution at run time is slow because it takes 67 percent of the time. The Cnn described above uses three convolution layers, so execution is three times slower than the same size mlp. Previously, the results were reported to go through a radical basis function (RBF). Recently, however, the use of rectified linear unit (ReLU) neurons as an active function can alleviate the vanishing gradient problem.

Such CNN has been successfully applied to text recognition and to various types of image recognition. Especially in the fields of speech recognition, document reading, and handwriting recognition, it has been applied since the early 1990s. However, CNN has become a major concern in the academic world since the image recognition model (Krizhevsky et al. [72]) released in 2012 has achieved remarkable results.

This model is responsible for classifying as many as one million color images into a thousand classes. The overall model is shown in Figure 3.9. As shown in Figure 3.9, since this model is implemented by two GPUs, the inner layer is divided into two parts except the input layer and the output layer.

There are three input panels because the color image must be recognized. In Figure 3.9, the first inner layer on the right side of the input layer performs convolution and max-pooling, which consists of 96 panels. The second inner layer to the right performs convolution and max-pooling as well, with a total of 256 panels. The next two inner layers consist of convolutions without max-pooling, and the fifth inner layer consists of 256 panels that perform convolutions and max-pooling.

The sixth and seventh inner layers are all connected to the nodes of the lower layer, and the final output layer consists of 1000 nodes. The model has a weight of approximately 60

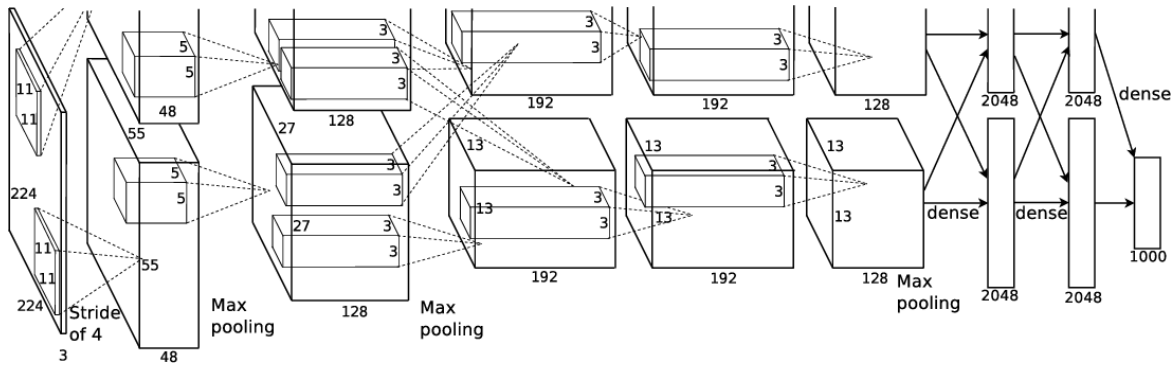


Fig. 3.12 Convolutional Neural Networks for image classification (Krizhevsky et al. [72])

million and took six to seven days using a GTX 580 3GB GPU to learn. This model uses ReLU ($\max(0, x)$) as an activation function to reduce learning time and to prevent over fitting in a dropout manner. In addition, the recognition rate of the model is increased by transforming the learning image to increase the number of input patterns. As a result, we achieved an error rate close to half of the recognition error rate of other models at that time. For that reason, many current CNN models have 10-20 ReLU layers, billions of node-to-node connections, and train hundreds of millions of weights. Of course, with the development of learning algorithms and the development of GPU technology, the learning speed has been reduced by several hours.

3.6 Conclusion and the future

In this chapter, we looked at the emergence of artificial neural networks and their evolution. We also reviewed the RNN and CNN theories that are representative supervised learning of artificial neural networks. RNN means that in sequential data, it remembers all the past information and also predicts the next behavior. The RNN algorithm also supplements the vulnerability through LSTM - This algorithm does not consider information that can be forgotten or ignored. As if humans don't need to remember any information, LSTM treats sequential data faster and more efficiently through information selection. Nowadays, interactive artificial intelligence based on RNN is emerging one after another.

Convolutional neural networks that construct neurons by mimicking animal visual cortex are actively used for speech recognition or image analysis. Especially, CNN of computer vision and RNN for natural language processing are combined to build translation module between image and language. In Chapters 7 and 8, we will use CNN and RNN together to simulate language in terms of embodied cognition. In Chapter 6, we will use a traditional multilayer perceptron to simulate the problem solving. We also try to figure out how to make

3.6 Conclusion and the future

inferences when we are faced with unlearned situations. To do this, we tried to infer by comparing and analyzing information about agent's behavior, object's state and location.

Deep learning uses more advanced neural network algorithms to implement faster, emotional, human-like computer programs. Here we would like to see the hypothesis presented by the mathematician Laplace. If technology is advanced, computer's power is incomparable, and all possible variables can be set, we hypothesize that we can predict the past, present and future.

This, of course, is akin to reckless optimism. But with the advent of big data and advances in computer technology, more active unsupervised learning is combined with existing supervised learning that requires human intervention. The AI technology that computers can learn for themselves like humans may come in the near future. However, there is criticism that deep learning is only a buzzword for neural networks. It is so-called the problem of black-box, which is always a problem in neural network theory research. This is how we can reuse the acquired knowledge itself in neural networks. In the end, we should start thinking about the cognitive system that makes use of the empirical methodology and symbolism artificial intelligence.

Chapter 4

Deep Reinforcement Learning

The neural network and deep learning we presented in Chapter 3 were a learning model planned by a supervisor. The reinforcement learning presented in this chapter presents a mental model of the world as an unsupervised learning model. In other words, based on the current state and behavior of the surroundings, we will create a model that makes predictions about the next state and behavior.

Psychologist Edward Tolman conducted an experiment in which a rat went through a maze (See Figure 4.1) In the early learning period, that is, the latent learning stage, it was observed that the mice draw their own cognitive maps while exploring the maze and learning the structure without being rewarded.

Once rewarded, the rats showed that they could use the cognitive maps they had already created to get through the maze faster. This behavioral psychology's Operant conditioning resembles artificial intelligence's reinforcement learning.

Let us Consider the process of actors reaching their goals. If actors are not given enough information about a given environment, they will have to perform a search for every possible cases. However, if the actor has a cognitive model of the environment, it is not necessary to explore possible behaviors individually. You can approach problem solving simply by inference.

In this chapter, we will examine the theoretical background of reinforcement learning called compensation through the Operant conditioning. In addition, I would like to consider how to solve the problem by simply inferring the number of cases without repeating them. We will simulate the application and consideration of this reasoning method in chapters 7 and 8.

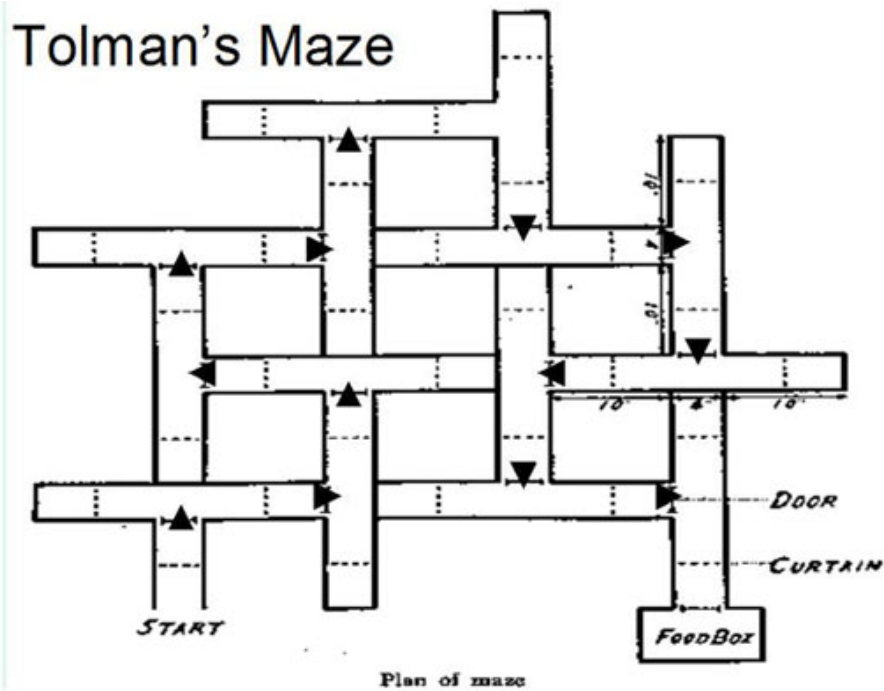


Fig. 4.1 Tolman's Maze (source: <https://berkeleysciencereview.com/2014/11/man-maze/>)

4.1 Introduction

In the previous sections we reviewed supervised learning such as Multilayer Perceptron(MLP), Recurrent Neural Network(RNN), and Convolutional Neural Network(CNN). In this chapter, we will learn about Reinforcement Learning. Reinforcement learning deals with sequential decision problems in which the order of observation of the data affects the results. In reinforcement learning, the agent interacting with the environment chooses the behavior in its current state. The result is a transition state that is passed back to the agent through a series of Markov Decision Processes(MDP) that receives rewards. In supervised learning, Agent received passive feedback from the teacher on the given input and the correct answer (guidance signal). Reinforcement learning, however, requires learners to generate behavior dynamically, with only a reward as feedback from the environment.

As we are well aware, humans and animals have advanced brain structures that allow them to process visual information quickly. This structure allows visual information to be shown as a small representation of essential dimensions, and humans and other animals appear to behave in conjunction with this process of reinforcement learning and similar hierarchical sensory processing systems (Mnih et al. [93]). Recently, the deep reinforcement learning is used as the input of the pixels of the image of the video game to enable the learning the operation of the human-level direct controller. Deep Q-Learning(DQN) has shown that

human-level manipulation can be achieved using deep learning and reinforcement learning algorithms. This chapter examines the theory related to the basic concepts of reinforcement learning and the overall situation leading to the recent achievement of DQN.

4.2 Reinforcement Learning and Markov Decision Process

4.2.1 Basic concept

Reinforcement learning is a learning method that deals with the process by which agents interact with their environment and achieve their goals (Figure 4.2). The agent is in state s_t at time t and selects action a_t to move to the next state s_{t+1} . At this time, the agent is rewarded with r_{t+1} . Then the agent repeats the sequential decision process, selecting the next action a_{t+1} from the state s_{t+1} . The agent has a policy function π that selects the action at the state s_t .

$$\phi : S \rightarrow A$$

$$\phi(s_t) = a_t$$

The reward function is defined as following :

$$R : S \times A \times S \rightarrow \mathbb{R}$$

$$r(s_t, a_t, s_{t+1}) = r_{t+1}$$

The sum of future rewards R_t is defined as follow:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

The value $V^\pi(s)$ of the policy is defined as the expected value of the sum R_t of future reward for taking action according to policy π in the state s .

$$V^\pi(s) = E_\pi[R_t | S_t = s]$$

The goal of learning is to find the optimal policy π^* that maximizes value.

One way to solve the classic problem, the Markov Decision Process(MDP), is the reinforcement learning theory. Unlike supervised learning in the previous chapter, Reinforcement

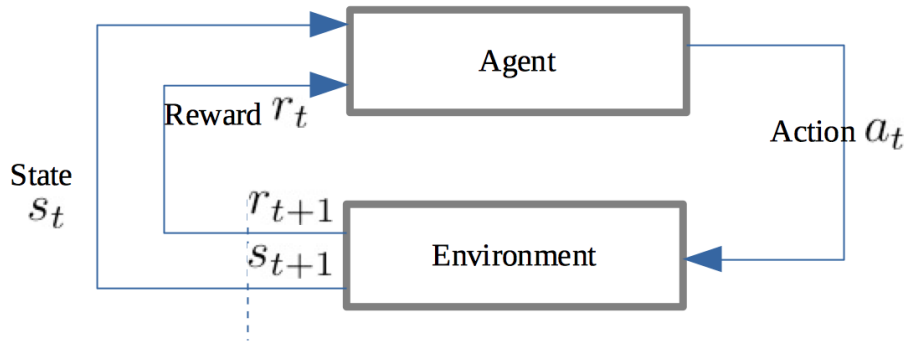


Fig. 4.2 Basic concept of Reinforcement Learning

Learning does not receive any instructions from the teacher. The learner performs only certain actions in a given situation and receives only the rewards. This reward will not only have an immediate effect on the current state, but also will eventually affect all rewards. Therefore, trial-and-error search and delayed reward are important features of reinforcement learning.

Also, exploration and exploitation are other important features of reinforcement learning. The agent chooses the most efficient of the actions(Exploitation) performed in the past to obtain as many rewards as possible. However for even better and more versatile choices, agents can also perform actions(Exploration) they've never tried before.

All of this series of reinforcement learning attempts to solve the Markov decision process. The Markov decision process is a random process that states that the values and decisions of each state exist in an environment where all states have Markov properties. The random process means that there is no need to remember the past other than the current state, and that each state is stochastically represented by a random variable. Mathematically, MDP is defined as $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ (See Figure 4.4). First, let's see what each element means.

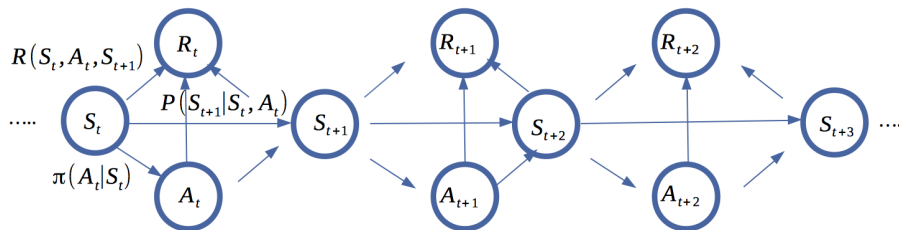


Fig. 4.3 Markov Decision Process

First of all, \mathcal{S} is a set of states. State is information that an agent can observe about the environment. This information can be defined in various ways. In other words, this may be

various signals at low level observed through the sensory organs of the agent, or it may be a representation of the senses expressed in higher dimensions. These states usually have some information to maximize the environment itself or the reward, depending on the problem. If a state has all the information needed to make the next decision, we call it the Markov state or Markov property. The mathematical expression is as follows.

$$P(S_{t+1}|S_t) = P(S_{t+1}|S_1, \dots, S_t)$$

When we observe a Markov state, the decision of the behavior for that observation and the prediction of the next state is the same as the optimal result after observing all the past. Given a state at time t called s_t , the agent takes some action a_t and receives the reward r_t from the environment for the previous actions. This sequence of $s_t, a_t, r_t, \dots, r_1, s_0, a_0$ is named as history up to t . In general, it is assumed that this complete historical information is required to obtain a complete probability distribution of the following states s_{t+1} and reward r_{t+1} , but if the state signal has a Markov property, This probability distribution can be determined exactly by knowing the state s_t and the behavior a_t .

Let us take the example of Go. If we know the placement of black and white stones at some point t and how much of the opponent's stone we obtained, we do not need any previous information on the order in which the games were played in order to place the next stone. Considering each state of Go as Markov, we can immediately find the optimal choice without past information, information flow, and memory. In other words, it can be seen that in the Markov's environment, no matter how the current environment represents the current state, it does not have a difference.

Just as we review all past information and make the best choice, the assumption of the Markov state is that we can make the same best choice by observing the previous state. Thus it fits well with our intuition. However, while we agree that the theory is concise and easy to implement, this assumption is often inconsistent with the real-world environment and the agent's ability to observe. However, Markov's assumptions help us with better understanding of reinforcement learning algorithms. The Markov assumption is the algorithmic basis for extending the more complex and practical Non-Markov cases(Sutton and Barto [134]).

\mathcal{A} is the set of actions. The category of behavior can be discrete or continuous, and these behaviors must be designed to suit the nature of the problem. \mathcal{P} is the state transition matrix. This gives the probability of a random variable for each next state given the current state and behavior, which is expressed as $\mathcal{P}_{SS'}^a = P(S_{t+1} = s' | S_t = s, A_t = a)$.

\mathcal{R} is the reward function, defined as xy_t , the average of the rewards to be received next step. γ is a discount element, given as a real number between 0 and less than 1. This discount factor determines how much of each reward to be taken at each time step. Animals including

humans are sensitive to time, even if they have the same total, and tend to judge their value differently. Thus, uncertainty about the future causes us to seek immediate gains. This philosophy is expressed in the definition of return G_t , which is the sum of all discounted rewards since the t point.

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (4.1)$$

Looking at the definition above, as the γ value approaches 0, the rewards for the near future are reflected, and as it gets closer to 1, the gamma values reflect the rewards for the distant future. When the status does not end, the discount element γ is considered to play a role that does not diverge the sum of rewards. On the other hand, policy is defined as the probability distribution of the agent's behavior for a given state and is expressed as follows.

$$\pi(a|s) = P(A_t = a | S_t = s)$$

With this definition, the transition matrix when following any policy, and the next reward function, can be represented by $\mathcal{P}_{SS'}^{\pi} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}_{SS'}^a$ and $\mathcal{R}_S^{\pi} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}_S^a$, respectively.

So far, we have briefly looked at the basic concepts of agents. Next, let's take a look at the value function, which is a state measurement function that aims at the goal when an agent conducts reinforcement learning.

4.2.2 Value function

The value function expresses the long-term value (return) of the state. The state-valued function $V^{\pi}(s)$ represents the average return $V^{\pi}(s) = \mathbb{E}[G_t | S_t = s]$ for the state when the policy is followed. Behavior-value function $q_{\pi}(s)$ is similar to state-value function. This function shows the average return $q_{\pi}(s) = \mathbb{E}[G_t | S_t = s, A_t = a]$ for state s and action a when the policy is followed. Substituting these two value functions into the above-mentioned identical equation for return (equation 4.1) yields the Bellman expectation equation.

$$V(s) = \mathbb{E}[R_{t+1} + \gamma V^{\pi}(S_{t+1}) | S_t = s]$$

$$Q_{\pi}(s, a) = \mathbb{E}[R_{t+1} + \gamma Q^{\pi}(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$$

Since the Bellman expectation formula can recursively represent the value of the model for the policy, a direct solution of the value function can be obtained linearly. The reinforcement

learning model takes an approach that uses the linearity of these Bellman's expectation formulas.

The primary purpose of reinforcement learning is to seek the behavior of the agent with the most rewards. When the rewards are ordered by the value of the value function, there are policies with the maximum value of this function, which we define as optimal policy. The optimal policy is usually expressed with symbols such as π^* . The optimal value function is defined as the return when the optimal policy is followed. This value can be said to be the maximum performance of MDP. The optimal state-value function is the value representing the largest transformation of all policies for a state. It is defined as follows.

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

The optimal action-value function is also expressed as follow:

$$Q^*(s) = \max_{\pi} Q^{\pi}(s, a)$$

Similarly, the corresponding identical equation is called Bellman optimality equation and is expressed as follow.

$$V^*(s) = \max_a R_s^a + \gamma \sum_{s' \in S} \mathcal{P}_{SS'}^a V^*(s')$$

$$Q^*(s, a) = R_s^a + \gamma \sum_{s' \in S} \mathcal{P}_{SS'}^a \max_{a'} Q^*(s', a')$$

The Bellman's equation is a nonlinear equation because the maximum value must be found, as shown in the equation. This Bellman's best solution is diverse. Reinforcement learning algorithms are still being developed to solve the MDP using this Bellman's equation efficiently.

4.3 Monte Carlo Method and Temporal Difference Algorithm

As discussed in the previous section, reinforcement learning problems are defined as being solved by obtaining optimal policies in the Markov decision process. Let's look at the reinforcement learning method based on the well-known value function. The value function is defined as the expectation of rewards in the future for continuing to follow the policy π in the current state $S_t = s$.

$$V_{\pi}(s) = \mathbb{E}[R_{t+1} + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots | S_t = s]$$

To get the value function according to the definition of this formula, we need to calculate the expectation. Depending on how to solve these expectations, reinforcement learning can be divided into two categories: planning and learning. Planning is a probabilistic approach to calculate expectations through explicit calculations.

Learning expectations are approximately solved using the Monte Carlo method of statistics. This "Planning" method has the big disadvantage that it is necessary to find out the transition probability of the environment first to get the expected value because it is very difficult to find out the transition probability in advance. The "Learning" method, on the other hand, does not require this probability of transition, but requires some practical experience to approximate the Monte Carlo method. That is, the agent solves approximately this expectation with a sequence of past and rewards gained from interacting with the environment. The mathematical expression of the Monte Carlo method is as follows.

$$G(s) = (R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s)$$

$$V_{\pi}(s) = \mathbb{E}[G_1(s) + \gamma G_2(s) + \gamma^2 G_3(s) + \dots | S_t = s]$$

In order for the expected approximation to be accurate, the larger the number of Sample G, the more accurate value function can be found. However, the previous value function is not enough to calculate the optimal policy. We can see the value of the current state s , but to find out what to do with it, we have to use transition probability to search for the next state s_{t+1} . However, since we do not use transition probabilities here, we need to get the behavior value function, not the state value function. Fortunately, however, the behavior value function can be easily computed by simply averaging the return values G of the action a in the state s rather than the state s .

$$G(s, a) = (R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s, A_t = a)$$

$$q_{\pi}(s, a) = \mathbb{E}[G_1(s, a) + \gamma G_2(s, a) + \gamma^2 G_3(s, a) + \dots | S_t = s, A_t = a]$$

If we can get the behavior value function, we can figure out a policy by choosing the most valuable behavior a given a state s .

$$\pi(s) = \arg \max_a Q(s, a)$$

4.3 Monte Carlo Method and Temporal Difference Algorithm

Using the Monte Carlo method, we can get the behavioral value function for a given policy. However, in order to find the optimal solution, it is necessary to reach the optimal solution by altering the behavioral value function and policy through the Generalized Policy Iteration.

$$\pi_0 \xrightarrow{E} q_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} q_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \dots \xrightarrow{I} \pi_* \xrightarrow{E} q_*$$

E is an approximation of the behavioral value function from the current policy through the Monte Carlo method, and the step I , indicated by the arrow, is the step of finding a better policy from the current behavioral value function.

The Monte Carlo method is a good example of reinforcement learning that is very intuitive to understand. However, this method has the disadvantage of waiting until the end of the episode to get the return value G . In the meantime, we can't do any learning. To solve this problem, Temporal Difference algorithm has been proposed. The amount of time to wait can be controlled with the parameter λ . For TD(0), where $\lambda = 0$, we can wait for one step and start learning immediately. In the Monte Carlo method, the value function is defined by the following formula:

$$V_{\pi}(s) = \mathbb{E}[G_1(s) + G_2(s) + G_3(s) + \dots | S_t = s]$$

In TD(0), however, the value function is computed using a single compensation value through a trick called bootstrapping.

$$V(s) = \mathbb{E}[R_{t+1} + \gamma V_{\pi}(S_{t+1}) | S_t = s]$$

This definition and the definition of the value function used in Monte Carlo are exactly the same. However, in calculating the expected value, the former has to wait until all the rewards are obtained, whereas the latter is online learning which can calculate the value function as soon as one of the rewards comes out. This is a very important advantage because the Monte Carlo method is difficult to apply to an infinitely long or very long problem. In addition, TD(0) has a very good ability to learn more quickly from fewer data than the Monte Carlo method.

Monte Carlo's dynamic programming method requires a state transition probability. This method is called model-based RL because state transition probability acts as a model to describe the nature of the problem. The TD method, on the other hand, is a model-free RL that does not require transition probability. Another difference is that the Monte Carlo method is an offline learning method and the TD method is an online learning method.

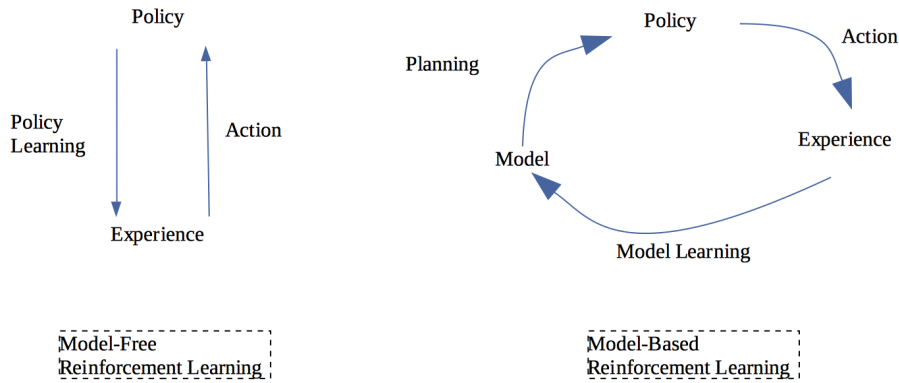


Fig. 4.4 Model-free Model-based Reinforcement Learning

4.4 State-Action-Reward-State-Action and Q-Learning

In reinforcement learning, there are On-policy control and Off-policy control according to the agent’s control method using optimal policy. There is a difference in terms of whether the policy of the agent is updated while executing the policy or whether the policy is updated by sampling the policies in the existing episode. Therefore, we describe two algorithms that are the core of each control method in reinforcement learning, State-Action-Reward-State-Action(Sarsa) and Q-Learning. Sarsa is a representative on-policy control algorithm. This chapter introduces the method using TD. In the control algorithm, the agent aims to learn the behavior value function. In the case of the on-policy algorithm, we must infer $Q^\pi(s, a)$ according to the behavioral policy π that follows all states s and behaviors a . The learning method of Q finds the optimal value function by TD in the existing value function clause. You can get it in the same way. For example, episodes consist of ordered state-behavior pairs.

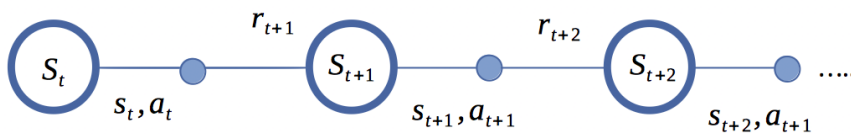


Fig. 4.5 State-Action Pair

The control algorithm learns the value of a state-behavior pair by considering the probability of transition from one state-behavior pair to another. This can be applied in the same way as the convergence of the behavioral value of TD(0) described in the following equation.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

This update algorithm is applied every time a transition occurs, unless it is in the terminating state. If s_{t+1} is the final state, the Q value is defined as 0. The expression must contain five elements of $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$. Basically, on-policy control algorithm using Sarsa continuously infers Q^π through behavior policy π and updates the behavior policy π with greediness. The Sarsa algorithm is summarized as follows.

When using the ϵ – greedy policy, all state-action pairs converge if they repeat infinitely. Next, let's look at Q-learning, a representative off-policy control algorithm. The Q learning method was proposed by Watkins(Watkins and Dayan [140]) in 1989 and the formula is as following:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (4.2)$$

As in the equation 4.2, the algorithm shows that the Q value that is learned is always approximated directly to Q^* , even if the agent behaves with a different policy. It can be understood that a policy converges to an optimal value as long as it visits all state-behavior pairs in the environment. Under these assumptions, Q learning always converges to Q^* (Watkins and Dayan [140]). The basic Q-learning algorithm is as follows.

4.5 Deep Q-Learning

The visual signals coming through the sensory organs that humans deal with every moment are very large data. In order to treat such data rapidly, humans and animals have developed a structure of the brain that can treat visual information quickly. This structure allows visual information to be represented as a small representation of essential dimensions, and humans and other animals appear to behave in conjunction with this process of reinforcement learning and similar hierarchical sensory processing systems (Mnih et al. [93]). This section deals with the Deep Q-network(DQN), one of the applications of reinforcement learning using deep learning. This model was initially proposed to apply reinforcement learning algorithms to real-time video game problems.

The problems addressed by DQN are the video games found on the Atari-2600 home game console in the late 1970s. DQN aims to score simple games such as Breaking Bricks and Pac-Man in the same way that humans enjoy the game with 8 directions and one button on the game controller. The purpose of this study is to learn the manipulation of a human-level direct controller using only the pixels of an image in a video game which requires deep reinforcement learning. DQN trained deep convolutional neural networks with Q-Learning to approximate behavior-state functions to learn the behavior of game controller manipulation.

```

Initialize  $Q(s, a)$  arbitrarily;
repeat(for each episode)
    Initialize  $s$ ;
    Choose  $a$  from  $s$  using policy derived from  $Q$  (e.g.,  $\epsilon$  – greedy);
    repeat(for each step of episode)
        Take action  $a$ , observe  $r, s'$ ;
        Choose  $a'$  from  $s'$  using policy derived from  $Q$  (e.g.,  $\epsilon$  – greedy);
         $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$ ;
         $s \leftarrow s'; a \leftarrow a'$ ;
    until  $s$  is terminal;
until;

```

Fig. 4.6 Sarsa algorithm

```

Initialize  $Q(s, a)$  arbitrarily;
repeat(for each episode)
    Initialize  $s$ ;
    repeat(for each step of episode)
        Choose  $a'$  from  $s'$  using policy derived from  $Q$  (e.g.,  $\epsilon$  – greedy);
         $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$ ;
         $s \leftarrow s'$ ;
    until  $s$  is terminal;
until;

```

Fig. 4.7 Q-Learning algorithm

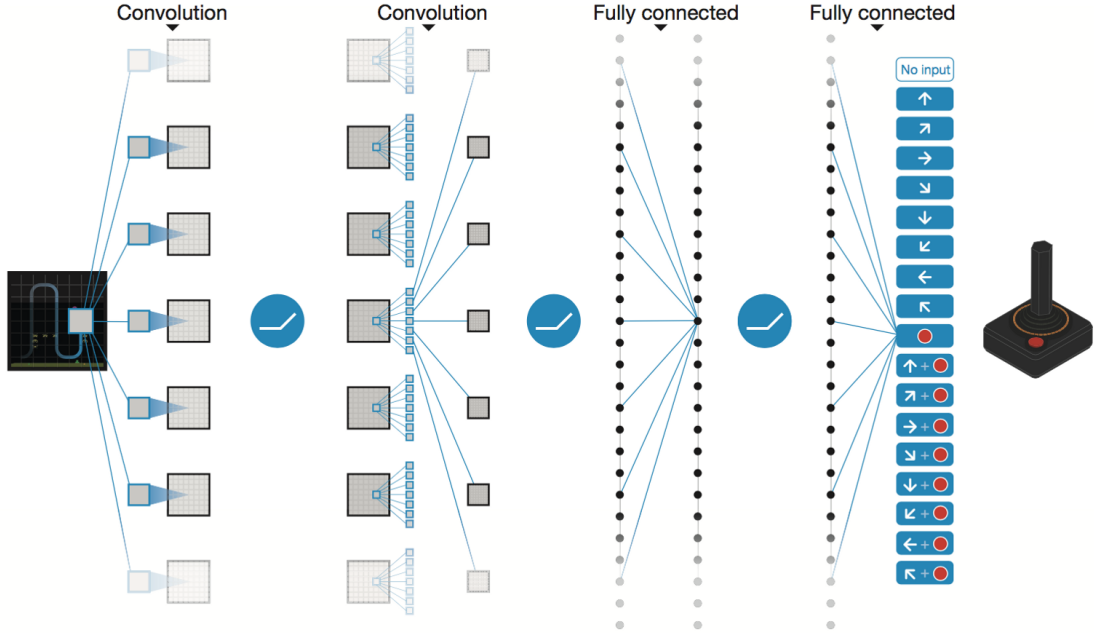


Fig. 4.8 Schematic illustration of the convolutional neural network from (Mnih et al. [94])

The structure of the DQN is shown in the figure. First, the network inputs are preprocessed images in black and white corresponding to the recent four frames of $4 \times 84 \times 84$. In this way, DQN’s input processing method is to view four black and white images of the present and the third previous frame as a Markov state representing all the information of the game. The DQN then takes the pixels of this image, obtains a generalized representation of the input through several convolutional layers, and then decides which action to perform with the controller through several closely coupled layers. All activation functions are rectified linear units(ReLU), and the error function of Q-Learning in this network is as follows.

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s') \sim U(D)} = \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i) \right)^2 \right] \quad (4.3)$$

In the above formula 4.3, θ_i is a parameter of the current DQN. θ_i^- is the target network that DQN is trying to reach and is assigned to θ_i at certain stages for convergence. For exploration and usage, we used an ϵ – greedy method that takes random behavior instead of the highest Q function with a probability as high as ϵ and gradually close to 0.

One of the features of DQN is that it uses heuristic replay. This saves the current state, behavior, compensation, and transitions of the next state in large memory, then randomly selects samples of data for min-batch in the learning phase. There are many reasons for doing an experience replay. The first reason is that the experience of each step is potentially used

for many weight updates, which can increase the efficiency of the data. The second reason is that learning directly from consecutive samples is inefficient because the correlation between the samples is strong. Finally, using experiential replays, the distribution of behaviors is averaged over many previous states, making learning smooth. The overall algorithm is as follows.

The algorithm is described in detail as follows. After initializing the memory and parameters containing the replay experience, the game is run with a game emulator. Every hour, we assign the current action a_t to ϵ – greedy, which is the highest Q function value in DQN. After this is done, the current state, behavior, reward, and next state are stored in replay memory. On the other hand, learning is performed by randomly extracting a mini-batch from the replay memory and updating it using Q-learning, and allocating a target network to the current network with constant cycle.

As a result, DQN reported successful performance in games that require immediate situational judgment and manipulation. In particular, when tested in 49 real-world Atari-2600 games without changing the network structure and learning style, DQN outperformed the existing algorithms in all 49 games and exceeded the human level in 29 games. In this way, DQN has demonstrated that human-level manipulation can be achieved using deep learning and reinforcement learning algorithms.

However, if you need some knowledge of the symbols which are commonly used, if you have short-term and long-term goals that do not hold MDP assumptions, and if there are games that are difficult to explore with a simple ϵ – greedy, it is known that the DQN algorithm does not work well.

4.6 Conclusion

Let us review reinforcement learning again. At lunch time we go out to eat. The sandwich shop where I usually go to is closed for construction, and the small brasserie, which is famous for its deliciousness, cannot go in because there are too many customers. While hesitating, time passes and we get more and more impatient. We check today's recommendations and prices at a Lebanese restaurant located in a side street just one block away. Then we search this restaurant on TripAdvisor and check the ratings and comments before making a decision. We use two methods of communicating with the environment: observation and action. Observation is looking at the environment through the sense perception. After the information of the closed sandwich shop and the brasserie filled with guests is delivered to the brain, we revise the original plan. We explore the back streets anew but new information causes interference in the brain's cognitive system. However, the final decision is made

4.6 Conclusion

after analyzing the reward effects (price, rating, valuation) of the new challenge. We learn the environment based on incomplete observations. Once a model of the environment has been created, it should be used to select optimized behaviors that are as close to the target as possible from the various possible methods in the current situation. Once you've decided what to do, it's time to execute it and to repeat the changes in real time.

Reinforcement learning relies on behavioral rewards and changes in state, but linked with embodied cognition. Unlike conventional information processing approaches, it is to see the recognition as a dynamic system that changes with time without intervening symbolic units or representations. Reinforcement learning collects and learns data on its own as agents interact with the environment. Thus, reinforcement learning is closer to human cognitive behavior than other supervised or unsupervised learning. In this paper, we will simulate the process of finding the optimal solution using reinforcement learning in Chapters 7 and 8.

```

Initialize replay memory  $D$  to capacity  $N$ ;
Initialize action-value function  $Q$  with random weights;
for  $episode=1, M$  do
    Initialize state  $s_t$ ;
    for  $t=1, T$  do
        With probability  $\epsilon$  select a random action  $a_t$ ;
        otherwise select  $a_t = \max_a Q^*(s_t, a; \theta)$ ;
        Execute action  $a_t$  and observe reward  $r_t$  and state  $s_{t+1}$ ;
        Store transition  $(s_t, a_t, r_t, s_{t+1})$  in  $D$ ;
        Set  $s_{t+1} = s_t$ ;
        Sample random minibatch of transitions  $(s_t, a_t, r_t, s_{t+1})$  from  $D$ ;
        Set  $y_j = \begin{cases} r_j, & \text{for terminal } s_{t+1}. \\ r_j + \gamma \max_{a'} Q(s_{t+1}, a'; \theta), & \text{for non-terminal } s_{t+1}. \end{cases}$ ;
        Perform a gradient descent step on  $(y_j - Q(s_t, a_j; \theta))^2$ ;
    end
end

```

Fig. 4.9 Deep Q-Learning with Experience Replay Pseudo Algorithm

Part II

Experiment and Simulation

Chapter 5

The effects of gesture in Tower of Hanoi

In this chapter, we will emphasize the embodied perspective by exploring the role of gestures in problem solving. According to our hypothesis, although the use of gestures and gesture strategies are rough and inaccurate, gestures affect memory and planning of learning process. In other words, the more the individual interacts with the environment through gestures, the more the individual's cognitive interaction ability is improved. To evaluate this hypothesis, we studied the influence of gestures on cognition in solving the tower of Hanoi problem. Participants are consisted of the blinds sighted people. Participants are given the task of solving the problem of the Tower of Hanoi with four discs. The first experiment had to solve the Tower of Hanoi with four discs, and the other had to solve the Tower of Hanoi without the disc.

In our experiments, we obtained the following data: total duration, number of gestures, number of deictic gestures, as well as data on the number of illegal gestures, and so on. According to the data, when solving the Tower of Hanoi without any disk, sighted participants used deictic gestures to recall the process of solving with disc in the first experiment. For blind participants, gesture interactions played a key role in solving the TOH problem. ¹

5.1 Introduction

According to the embodied approach of cognition (Anderson [2], Lakoff and Johnson [75], Wilson [143], Barsalou [8]), cognitive processes are deeply related to physical interactions that affect our thinking, reasoning, and decision making. As the body is part of the causal relationship that enables perceptual interaction, the main goal of the study of embodied

¹Sanghun, B., Mathilde, M., François, J. and Charles, T. The effects of gesture in Tower of Hanoi. In the 2nd Eurasian conference on educational innovation 2019 (ECEI-2019): Papers submitted to this conference. Adopted but not participated.

cognition is to understand the intellectual behavior of cognition and behavior related to the movement of the body. Embodied cognition seems to have a lot of contributions to our understanding of the nature and of the influence of gestures. In the literature on embodied cognition, gestures are often used as a grounding for a mapping between idea and an object in the world, to make the meaning clearer (Harnad [56], Lakoff and Johnson [75], Lakoff and Nuñez [76]).

As the recent research shows, the effect of the gesture is to increase the problem solving ability by decreasing the working memory (Cook et al. [36]). Let us take another example. We often use our hands when we talk. This is a cognitive action to convey what the speaker is trying to say more clearly (Hostetter et al. [61]). In addition, gestures in infancy are used to convey concepts which are not yet implemented in language (Singer and Goldin-Meadow [130]).

Therefore, in this chapter, we asked participants to solve the Hanoi Tower puzzle to observe the effect of gestures on cognitive processes (learning, memory, planning, decision making, etc.). The rules of the Tower of Hanoi puzzle are simple. There are three rods and a number of disks in different sizes. Disks are stacked on one rod. The goal is to move all the disks from one rod to the other.

We can only move one disk at a time and we can't put a large disk on a small disk. We also asked participants to solve the Tower of Hanoi problem without disks. Participants have to solve the problem without disk by imagining moving the disk in their heads. In this case, the participants should verbally explain the solution to the supervisor. The Tower of Hanoi problem solving depends on the presence of the disk is to compare and analyze gestural effect in correlation between visual effect and cognition. In this chapter, we also compared planning performance depending on the presence of a disk.

5.2 Experiment: Observing blind and sighted participants solving the Tower of Hanoi problem

5.2.1 Participants

We recruited 14 participants (average age 41, standard deviation 8.51). The blind group and the sighted group consisted of six females and one male, respectively. The average age of blind group was 39 years with a standard deviation of 6.65. The average age of sighted group is 43 years with the standard deviation 10.30.

5.2 Experiment: Observing blind and sighted participants solving the Tower of Hanoi problem

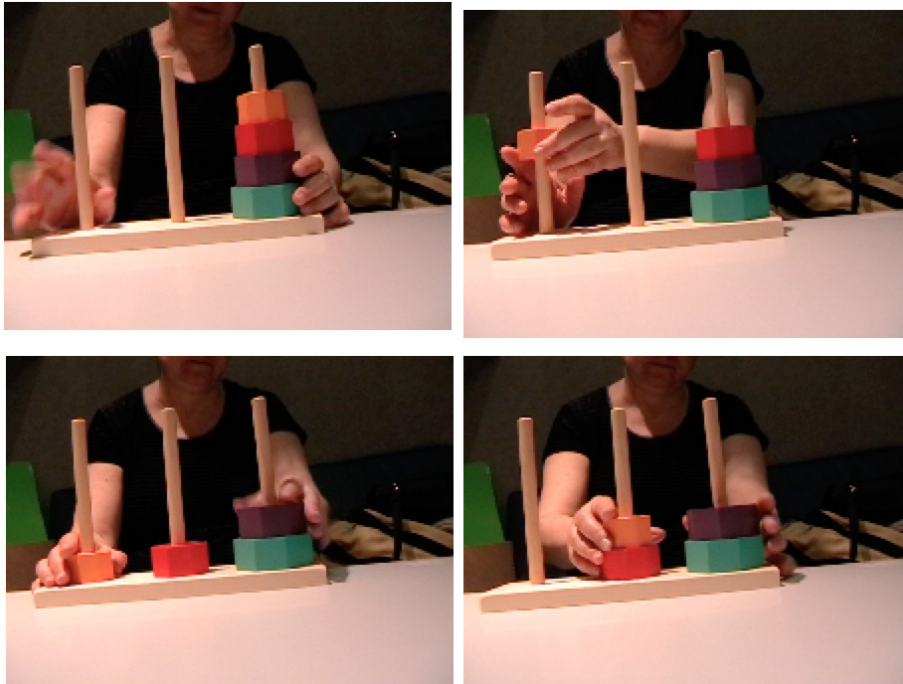


Fig. 5.1 A participant performing Tower of Hanoi with disk

5.2.2 Procedure

We tested each participant individually. Before the game began, participants were asked to learn the rules and then solved the problem. The blind participant was given the same explanation as the sighted participant but gave additional explanation for problem solving. Participants were asked to resolve a TOH with 4 disks and we recorded everything in the process (Fig. 5.1). After the first experiment, the participants were asked to solve the TOH problem without the disk. As in the first experiment, all problem solving procedures were recorded. In the second experiment (Fig. 5.2), the basic premise was that although the disks were not visible, all four disks were stacked on the left disk. When the participants moved the disk, we asked them to explain which disk they were grabbing and where to move it.

5.2.3 Coding

With information about the recorded gesture behavior, we classified the experimental data. Appendix A summarizes the experimental results for the blind group and the sighted group. As shown in tables A.1 and A.2, the data consists of two qualitative and 16 quantitative variables. Two qualitative variables consist of conditions and groups. The group variable has

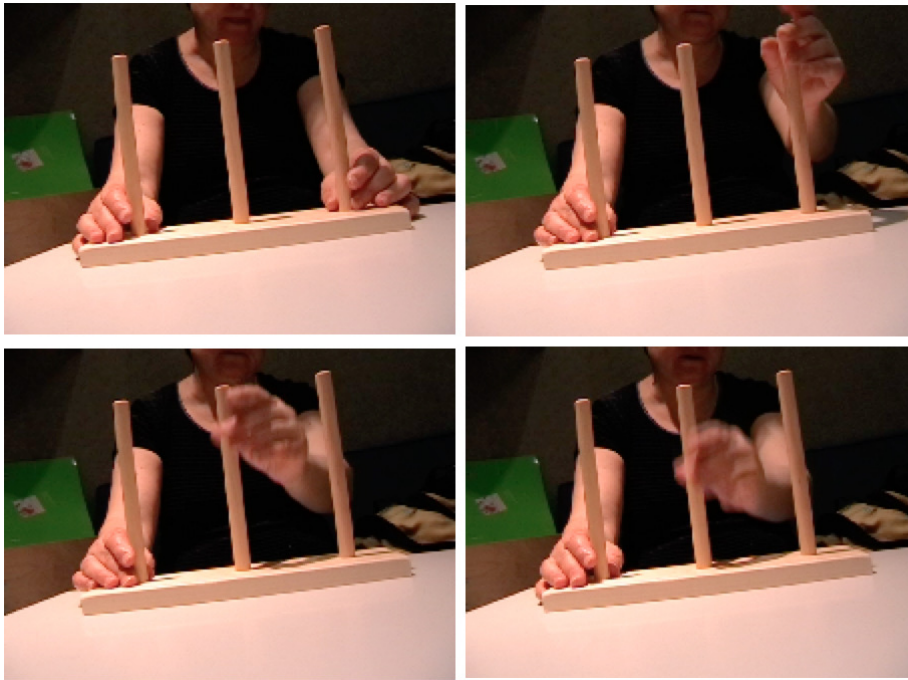


Fig. 5.2 A participant performing Tower of Hanoi without disk

already been mentioned above, and the condition variable is divided into whether the disk is used or not. Here we classify 16 quantitative variables as follows:

- Average time between two gestures in the first experiment (A)
- Average time between two gestures in the second experiment (B)
- Average gesture time in the first experiment (C)
- Average gesture time in the second experiment (D)
- Total time in first experiment (E)
- Total time in second experiment (F)
- Number of strokes in the first experiment (G)
- Number of strokes in the second experiment (H)
- Number of deictic gestures in the first experiment(I)
- Number of deictic gestures in the second experiment(J)
- Number of gestures in the first experiment(K)

- Number of gestures in the second experiment(L)
- Number of interaction gestures in the first experiment(M)
- Number of interaction gestures in the second experiment(N)
- Number of incorrect gestures in the first experiment(O)
- Number of incorrect gestures in the second experiment(P)

Among the classified data from above, we would like to know the classification criteria of 'deictic gesture' and 'interactive gesture'. Deictic gesture is a co-verbal gesture with the hand pointing to the object. It helps us to understand the participants' attention towards a specific referent in the proximal or distal environment (Cochet and Vauclair [32]). Deictic gestures and their meanings are affected by physical appearance, movement, and the distance between the gesture and the object. We often understand deictic gesture as what is commonly referred to as hand.

We called interaction gesture, a gesture with the hand comprising the gesture of moving disk as well as the gesture of touching. We suppose that the interaction gestures are influenced by experience and learning.

5.3 Method and Analysis

ANOVA is used to analyze the data obtained from the experimental results of the blind and sighted groups. This method is useful for investigating the significance of the effects of factors that can influence the Hanoi Tower problem solving. The two-factor ANOVA model is written as follows:

$$X_{ijk} = \bar{y} + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk} \quad (5.1)$$

where

- \bar{y} , overall mean response (Grand mean)
- α_i , the treatment effect for the i-th category of the row variable
- β_j , the treatment effect for the j-th category of the column variable
- $(\alpha\beta)_{ij}$, the interaction effect for the combination of the i-th row category and the j-th column category

Therefore, to investigate the effect of gesture between groups or conditions, a two-way analysis of variance (ANOVA) was performed. ANOVA included two conditions (with disk and without disk) and two groups (sighted and blind participants) as factors. The dependent variables included reaction time taken at first attempt (Trial 1), time taken at second attempt (Trial 2), number of deictic gestures on first attempt (Trial 1), number of deictic gestures on second attempt (Trial 2), number of gesture interaction on first attempt (Trial 1), and number of gesture interaction on second attempt (Trial 2).

5.3.1 The effect of deictic gesture

First of all, we applied two-way analysis of variance (ANOVA) to measure the relevance of the deictic gesture by factors (Condition and Group) when applicants solve the Hanoi Tower puzzle for the first time. Let us examine the results for the number of deictic gestures by the factors in our first attempt at solving the Tower of Hanoi. In the participants' first attempts at solving TOH, the conditional factor had a significant impact on the deictic gesture $F(1, 24) = 5.8161, p = 0.02389$.

In order to verify the interaction on the effect of deictic gesture on the first attempt, we set the null hypothesis for the interaction of group and condition to be "there is no interaction between the condition level and group". The alternative hypothesis, on the other hand, is that "there is an interaction between the condition level and the group". The test statistic is $F(1, 24) = 0.65$ and the p.value is larger than 0.05 ($p = 0.4294$). Therefore, at the significance level of 5%, we accept the null hypothesis and conclude that there is not a significant interaction between the condition level and the group for the effect of deictic gesture on the first try.

Now, let us examine the results for the number of deictic gestures by the factors in our second attempt at solving the Tower of Hanoi. In the second trial, the results of the ANOVA analysis on the number of deictic gestures showed a significant effect not only on the conditions $F(1, 24) = 9.3364, p = 0.005436$, groups $F(1, 24) = 9.3364, p = 0.005436$, but also on the condition-group interaction.

In order to verify the interaction on the effect of deictic gesture on the second attempt, we set the null hypothesis for the interaction of group and condition to be "there is no interaction between the condition level and group". The alternative hypothesis, on the other hand, is that "there is an interaction between the condition level and the group". The test statistic is $F(1, 24) = 0.65$ and the p.value is less than 0.05 ($p = 0.4294$). Therefore, at the significance level of 5%, we reject the null hypothesis and conclude that there is a significant interaction between the condition level and the group for the effect of deictic gesture on the second try.

5.3 Method and Analysis

Table 5.1 Deictic gesture between group and condition (Average)

Group	Number of deictic gestures(Trial 1)			Number of deictic gestures(Trial 2)		
	Overall	With disk	Without disk	Overall	With disk	Without disk
Sighted	6.5714	0	13.1429	6.7857	0	13.5714
Blind	3.2857	0	6.5714	1.1429	0	2.2857

More specifically, a Tukey post-hoc analysis was performed to determine which mean pairs were significant. The figure 5.3 shows Post-hoc t-test's result. This result shows the significant effect of dictic gestures, as revealed by two-way ANOVA followed by a Tukeys post hoc comparison.

Table 5.2 Tukey post-hoc comparison for the effect of deictic gestures (Try2)

Groups being compared	difference in means	Lower confidence interval	Upper confidence interval	Adjusted p-value
1:4-2:4	2.285714e+00	-7.837285	12.408713	0.9237264
2:3-2:4	1.903239e-16	-10.122999	10.122999	1.0000000
1:3-2:4	1.357143e+01	3.448429	23.694428	0.0057843
2:3-1:4	-2.285714e+00	-12.408713	7.837285	0.9237264
1:3-1:4	1.128571e+01	1.162715	21.408713	0.0249215
1:3-2:3	1.357143e+01	3.448429	23.694428	0.0057843

1: Without disk, 2: Disk, 3: Sighted participants, 4: Blind participants

The results presented in the table 5.2 give the difference in mean, confidence level, and adjusted p-values for all possible pairs. The interactions post hoc tests compare each pair of combinations. This shows that there is a significant difference ($p = 0.0058$) between participants with the sighted participants solving diskless problems and those with the blind participant solving problems with disks. It also shows that there is a significant difference ($p = 0.025$) when the sighted participants solve the diskless Hanoi Tower problem and when visually impaired participants solve the Diskless Hanoi Tower problem. In addition, for the sighted participants, there is a significant difference ($p = 0.0058$) depending on the presence of a disk.

5.3.2 The effect of gesture interaction

As in the previous section, we applied two-way ANOVA in order to analyze the effects of interactive gestures. Let us examine the consequences of the effect of interactive gestures on the first attempt at problem solving in the Tower of Hanoi. When participants first solved the Tower of Hanoi problem, conditional factors $F(1, 24) = 21.32, p = 0.0001$ and condition-

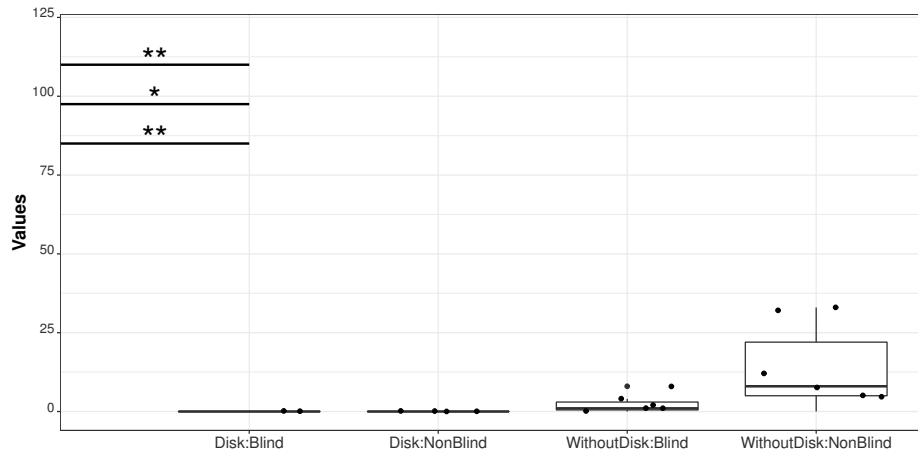


Fig. 5.3 Effect of deictic gesture between group and condition. ** $p < 0.01$: Significant effect of deictic gesture (Sighted participant without disk versus Blind participant with disk) as revealed by two-factor ANOVA followed by Tukey’s post hoc comparisons. ** $p < 0.01$: Significant effect of deictic gesture (Sighted participant without disk versus sighted participant with disk) as revealed by two-factor ANOVA followed by Tukey’s post hoc comparisons. * $p < 0.05$: Significant effect of deictic gesture (Sighted participant without disk versus blind participant without disk) as revealed by two-factor ANOVA followed by Tukey’s post hoc comparisons.

Table 5.3 Gesture interaction between group and condition (Average)

Group	Number of gesture interaction(Trial 1)			Number of gesture interaction(Trial 2)		
	Overall	With disk	Without disk	Overall	With disk	Without disk
Sighted	6.5	9.42858	3.5714	5.2143	7.8571	2.5714
Blind	10.5	20.1429	0.8571	7.7143	14.4286	1

group interactions factor $F(1, 24) = 6.08, p = 0.02$ had a significant impact on interaction gestures.

In order to verify the effect of the interaction gesture on the first attempt, we set the null hypothesis for the interaction of group and condition: "There is no interaction between the condition level and the group". The alternative hypothesis, on the other hand, states that there is an interaction between the condition level and the group. The test statistic is $F(1, 24) = 6.08$ and the P value $p = 0.0212$ is less than 0.05. Therefore, at the significance level of 5%, we reject the null hypothesis and conclude that there is a significant interaction between the condition level and the group.

More specifically, a Tukey post-hoc analysis was performed to determine which mean pairs were significant. The figure 5.4 shows Post-hoc t-test’s result. This result shows the

5.3 Method and Analysis

significant effect of interactive gestures, as revealed by two-way ANOVA followed by a Tukeys post hoc comparison.

Table 5.4 Tukeys post-hoc comparison for the effect of interactive gestures (Try1)

Groups being compared	difference in means	Lower confidence interval	Upper confidence interval	Adjusted p-value
1:4-2:4	-19.285714	-29.906584	-8.66484502	0.0002240
2:3-2:4	-10.714286	-21.335155	-0.09341645	0.0474722
1:3-2:4	-16.571429	-27.192298	-5.95055931	0.0013050
2:3-1:4	8.571429	-2.049441	19.19229784	0.1447657
1:3-1:4	2.714286	-7.906584	13.33515498	0.8941129
1:3-2:3	-5.857143	-16.478012	4.76372641	0.4409476

1: Without disk, 2: Disk, 3: Sighted participants, 4: Blind participants

Regarding the effects of interactive gestures, it can be seen that there are significant differences ($p = 0.0002$) between the blind participants in solving the tower of Hanoi with disk and without disk. There is also an important difference ($p = 0.0474$) between the case where the sighted participants solve a problem with a disk and the blind participants solve a problem with a disk. Finally, it can be seen that there are significant differences ($p = 0.0013$) between the sighted participants solving without a disk and the blind participants solving without a disk.

Let us examine the consequences of the effects of interactive gestures in the second attempt at solving the Tower of Hanoi. In this case, the results of the ANOVA analysis with respect to the number of interaction gestures showed only significant effects about the condition $F(1, 24) = 14.88, p = 0.0008$.

Therefore, the second attempt concludes that there is no significant interaction between the condition level and the group with respect to the interaction gesture at the significance level of 5%.

5.3.3 Discussion

When the sighted participants solved the Tower of Hanoi problem without disk, they made use of their deictic gestures regardless of the number of attempts. Some participants had difficulty remembering the process of solving the Hanoi Tower problem without a disk. As can be seen from the experiments, participants who had difficulty finding answers tried to communicate with the environment (disks, columns) through their gestures. Therefore in other cases such as solving the Hanoi Tower problem without the disks, we have to imagine that they are in front of me like a mental representation. As if we imagine something that

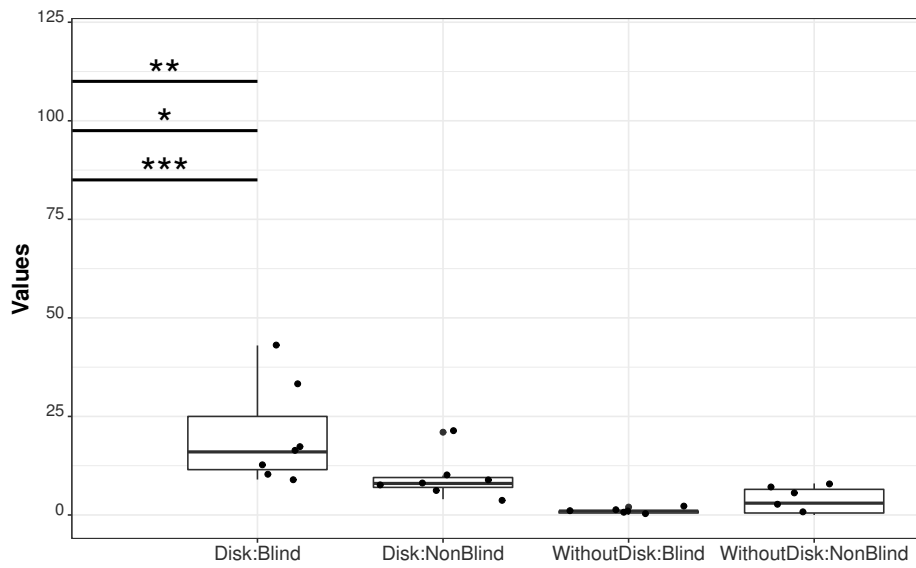


Fig. 5.4 Effect of gesture interaction between group and condition.*** $p < 0.001$: Significant effect of gesture interaction (Blind participant without disk versus Blind participant with disk) as revealed by two-factor ANOVA followed by Tukey's post hoc comparisons. ** $p < 0.01$: Significant effect of deictic gesture (Sighted participant without disk versus Blind participant with disk) as revealed by two-factor ANOVA followed by Tukey's post hoc comparisons. * $p < 0.05$: Significant effect of deictic gesture (Sighted participant with disk versus blind participant with disk) as revealed by two-factor ANOVA followed by Tukey's post hoc comparisons.

5.4 General discussion and conclusion

doesn't actually exist in our senses. Deictic gestures are then used as grounding for mapping between imagined objects and actions. Therefore, this deictic gesture helps you remember what they did in previous attempts.

But does using deictic gestures help solve the Hanoi Tower problem faster? Of course not. This is because the number of deictic gestures used in the first trial is correlated with the total time spent in the first trial [$r = .44, p < 0.019$]. Deictic gestures helped to find previous memories but did not reduce total time. We can also see that when we first solve the Hanoi Tower problem, the number of directed gestures has a weak correlation [$r = .298, p > 0.1$] to the total time required when we solve the puzzle for the second time. When the participants solved the problem for the second time, this gesture helped a little, but it did not have a big correlation overall.

On the other hand, the blind participants interact with the environment not with their eyes, but with their hands. Interactive gestures that touch or rotate disks provide action information to their mental representations. For this group, the number of interaction gestures in the second attempt correlates with the total duration in the second attempt [$r = .496, p < 0.0072$].

In this study, we found that the sighted participants use more deictic gestures, while the blind participants use more interactive gestures. The blind participants can see that they form their mental representation by touching objects with their hands. On the other hand, the sighted participants can refer to objects by hand to maximize their visual effects and form a mental representation. What was unsatisfactory in this experiment was the difficulty in drawing conclusions due to the small sample size. In addition, the Hanoi Tower problem was solved twice, with and without a disk, but it would have been nicer if we tried more attempts.

5.4 General discussion and conclusion

When we solve a problem or a task, our brain captures states across modalities and integrates them with a multimodal representation. Our brain then retrieves and executes the memory associated with the previous simulation. In this section, we performed two experiments (Tower of Hanoi problem with or without disk) to test the effect of the gesture on the planning. Participants were asked to solve the Tower of Hanoi without disk after solving the Tower of Hanoi with disk.

Our results show that in the absence of discs, participants have difficulty visual simulations. Since visually perceived objects automatically trigger simulations of functional actions, such as grabbing or moving objects. As participants have a difficulty triggering simulations without visual objects, deictic gestures of the sighted participants play a central role in generating visual inferences. Meanwhile, the blind participants have difficulties

solving Tower of Hanoi because of the lack of tactile objects. The interactive gestures for these participants play an important role in building their mental representation.

In spite of the lack of samples, the more the user interacts with the environment through gestures, the more the individual's cognitive ability increases. In the introduction we tried to understand the embodied perspective by exploring the role of gestures in problem solving. Then, I would like to reconstruct this experiment from the embodied point of view. It is also worth considering an experiment to see the effect of vision and touch on problem solving by changing the color of the disk or by changing the sense of touch depends on the size of the disk.

Chapter 6

Solving the Tower of Hanoi using Neural Network approach

6.1 Introduction

In this chapter, we propose a reasoning solution to infer the rules of a game such as the Tower of Hanoi. Neural networks require a large amount of data to improve their performance. But humans don't need much data for learning. If we can solve the Tower of Hanoi with four discs, the ten discs can be easily solved as well. It is because we know the rules. But if Artificial Intelligence doesn't know the rules of the game, it will take a lot of effort and time to learn the Tower of Hanoi with 10 disks. Therefore, the goal of this chapter is to find out how to solve the Tower of Hanoi without much learning.

From a psychological point of view, the reasoning of human being can be considered as the process of drawing conclusions. Human beings' endeavors to solve problems or to make decisions are goal-oriented (Leighton and Sternberg [81]). Reasoning has long been regarded as exclusive domain of the human being. It was one of the most difficult areas to implement mechanically. Because we need to understand contextual meanings, such as texts and images, and we have to consider contextual relationships that change depending on the situation, even with the same information. Therefore, reasoning has received a lot of attentions in the last few decades (Minsky [90], Mueller [96]). In recent years, advances in deep learning, especially the evolution of learning algorithms, are leading to rapid research and development of artificial intelligence in the field of reasoning (Kumar and others [74]).

We will present three models for reasoning in this chapter. First, we would like to present the process of solving the Hanoi Tower problem using a multi-layer perceptron neural network, one of the artificial neural network algorithms. After that, we will use neural

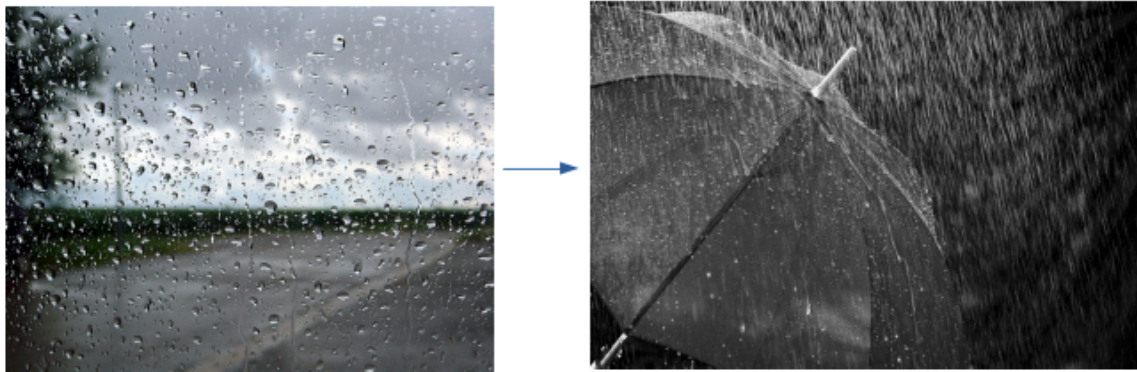


Fig. 6.1 Association Between rain and umbrella

association model and relational network model which are nonlinear models for probabilistic reasoning. The neural association model measures the probability of association of event 2 (E2) occurring under the conditions of event 1 (E1), $P(E2|E1)$. For example, we look at the picture 6.1 below. We can infer that event E2, which can be deduced in case of rain event E1, is most likely an umbrella. Based on these probabilistic inferences, we would like to deduce the Hanoi Tower solution. Meanwhile, the relationship network model will also utilize a learning neural network to infer about different objects and their relationships.

Publications related to this chapter are :

1. Sanghun, B. Solving the Tower of Hanoi using Neural Network approach. In International Conference on Machine Learning and Soft Computing (ICMLSC 2017), 2017. Ho chi minh, Viet Nam
2. Bang, S., Tijus, C. Neural Network-Based Reasoning for Solving the Tower of Hanoi. In the Proceedings of The Eleventh International Conference on Advanced Cognitive Technologies and Applications COGNITIVE 2019, 2019. Venice, Italy.

6.2 Reasoning model

6.2.1 Multi-Layer Perceptron Model

As shown in the figure 6.2, the simple form of the feed-forward neural network was a simple form with one input layer on the left and one inner layer and one output layer.

Like the simple perceptron, the multilayer perceptron consists of a hidden layer, an output layer, a weight, and biases, but the number of hidden layers has two or more.

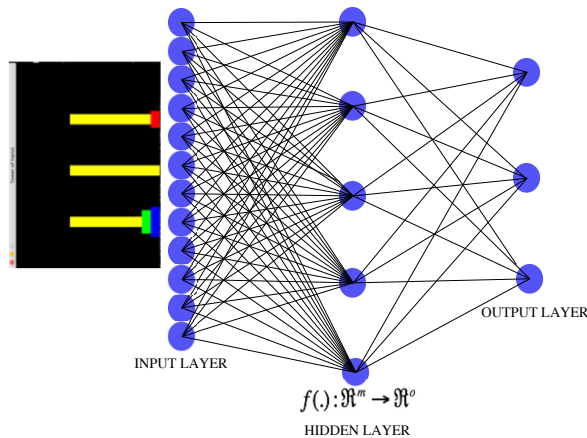


Fig. 6.2 Model of Multilayer perceptron for Tower of Hanoi

First, in order to build the Hanoi Tower Learning Model, we placed the data to be trained on the input layer, and performed iterative learning to output the most suitable value for the desired result by adjusting the weight and bias values in the hidden layer.

We used 28 sample training sets with 4 disks to solve the Tower of Hanoi puzzle. This sample data was obtained from the participants in the previous chapter. Each sample consists of 12 elements of vectors. It consists of three discs and three columns. The first input vector (1111 0000 0000) means that all four disks are in the first peg (Table 6.1). The leftmost 1 is the disk located at the top of peg A. If this input vector is presented, the model of multilayer perceptron will learn the output vector (100 010) for the initial input vector. What this output vector means is to move the disk located at the top of the left peg to the middle peg. This model trains repeatedly the pairs of input and output practice data. This repetitive neural network learning is a model to find the rule of The Tower of Hanoi.

Table 6.1 One example of training data set for multi-layer neural network model

	Input	Output
1	(1111 0000 0000)	(100 010)
2	(0111 0001 0000)	(100 001)
3	(0011 0001 0001)	(010 001)
4	(0011 0000 0011)	(100 010)
5	(0001 0001 0011)	(001 100)
6	(0011 0001 0001)	(001 010)
7	(0011 0011 0000)	(100 010)
8	(0001 0111 0000)	(100 001)
9	(0000 0111 0001)	(010 001)
10	(0000 0011 0011)	(010 100)

Table 6.2 Confusion Matrix

Predicted Values	Actual Values	
	True	False
True	True Positive	False Positive
False	True Negative	False Negative

We performed the accuracy and ROC(Receiver Operation Characteristic) analysis of each model to test the model suitability of the multi-layer perceptron artificial neural network. Accuracy refers to the ratio of how many samples predicted that my algorithm is the correct answer among the total number of samples are included. Accuracy analysis is to measure the actual predictive accuracy of our model.

In the end, the factor that evaluates the model can be defined as the relationship between the answer presented by the model and the actual answer. If the correct answer is divided into True and False, and the classification model also provides a value of True false As shown in Table 6.2 below, the case can be divided into 2×2 matrix.

Now if we look at each case:

- True Positive (TP): Predict the true answer as true (correct answer)
- False Positive (FP): Predict the correct answer that is actually false as True (Incorrect answer)
- False Negative (FN): Predict the correct answer that is true as false (incorrect answer)
- True Negative (TN): Predict the correct answer that is actually false as false (correct answer)

6.2.2 Neural Association Models

Neural relationship models also focus on the association with objects. Take the example of the image mentioned earlier. If we see rain through the window, the probability of carrying an umbrella increases as we go out. In this case, the keywords of the relationship become rain. In the previous section, let us say event E1 is three disks being stacked on the left at the initial state $t = 0$. Let us say E2 is the case that lifts the smallest disk and moves it to the right. peg1 contains disks, and the smallest disk is on the second smallest disk, indicating the relationship between events e1 and e2. Given a pair of inputs, $E1 = (e_i, r_k)$, we find the conditional probability $Pr(E2|E1)$ to find $E2 = (e_j)$ (Figure 6.3). In other words, when the triplet $X_n = (e_i, r_k, e_j)$ is given, the prediction value y_n is obtained.

6.2 Reasoning model

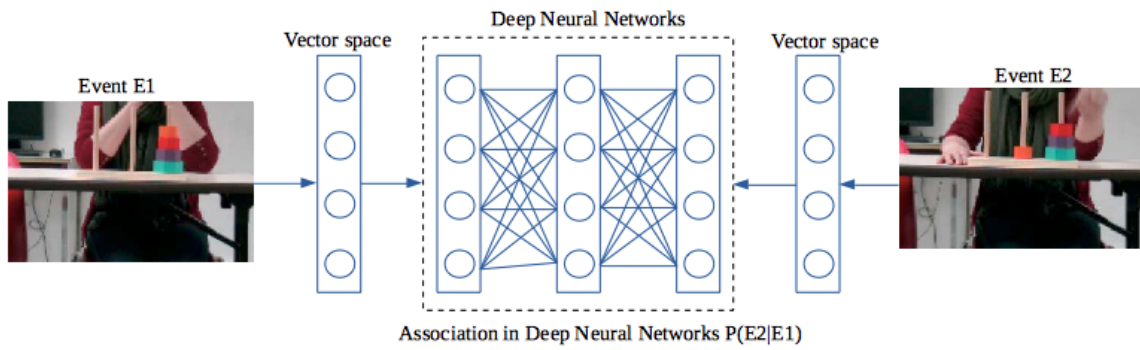


Fig. 6.3 Structure of Neural Association Model

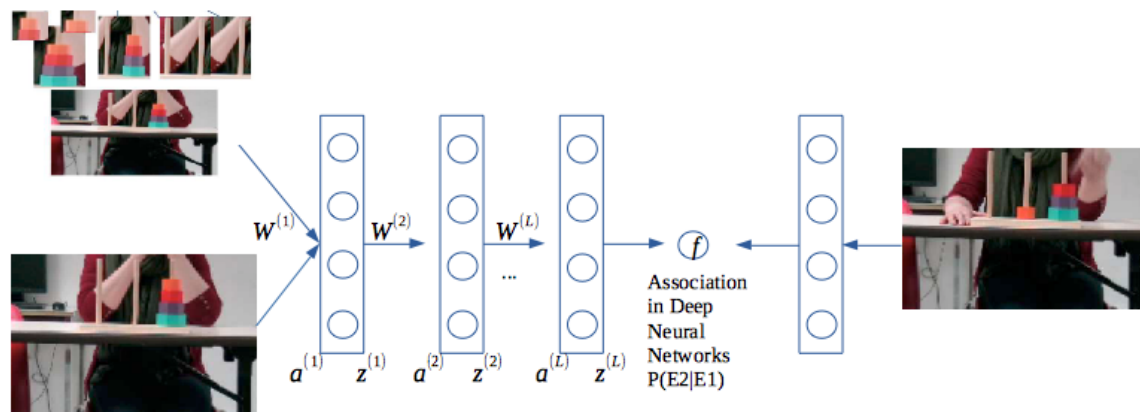


Fig. 6.4 Association in Deep Neural Networks

Let us look at Figure 6.4 for a closer look. Given a triple $X_n = (e_i, r_k, e_j)$, the vectors $v_i^{(1)} (\in V^{(1)})$ and $v_j^{(2)} (\in V^{(2)})$ are embedding the entities e_i and e_j respectively. The relation r_k is represented by a vector $c_k \in C$ (all relations). The input is $z^{(0)} = [v_i^{(1)}, c_k]$. During the process, we have:

$$a^{(l)} = W^{(l)}z^{(l-1)} + b^l (l = 1, \dots, L)$$

$$z^{(l)} = h(a^{(l)}) = \max(0, a^{(l)}) (l = 1, \dots, L)$$

where $W^{(l)}$: Weight matrix and b^l : bias for layer l

We calculate a sigmoid score for each triple x_n :

$$f(x_n; \Theta) = \sigma(z^{(L)} \cdot v_j^{(2)}), \text{ Where } \sigma(x) = 1/(1 + e^{-x})$$

All network parameters of this structure is represented as $\Theta = \{W, V^{(1)}, V^{(2)}, C\}$

Our hypothesis is that we can infer the rules of the game not through the description of rules of the game, but through the movement of the hand the disc. Therefore, we study this problem solving through the action of others and the change of states. To solve the problem of tower of Hanoi, we make use of machine learning methodology. We develop an intelligence that learns and infers the rules of the game by supervised learning for neural network. Based on the understanding of rules, the intelligence finds also an optimal solution by means of reinforcement learning method. Finally, we show that the artificial intelligence leans the problem of Tower of Hanoi from the movement of body, behaviors, and the change of states. ¹

6.2.3 Relational Network Model

As mentioned in the previous chapter, only one disk can be moved at a time, and the disk can only be moved if it is the top disk on a pile (See Figure 6.5). Also a larger disk cannot be placed on a smaller disk. From the rules of the game, we can infer the behavior of the participants. How can we make such an inference without any information about the rule of game? The relational model we present was inspired by Interaction Networks (INs) (Battaglia et al. [12]) and Relation Networks (RNs) (Raposo and others [120]). These models, which have a neural network architecture, are very helpful for inference learning by observing objects and their relationships and behaviors.

In a similar way to the relational network (Raposo and others [120]) model, our model infers the rules of the game based on the relationship \mathcal{R} between objects. Based on the data

¹Sanghun, B. Solving the Tower of Hanoi using Neural Network approach. In International Conference on Machine Learning and Soft Computing (ICMLSC 2017), 2017.

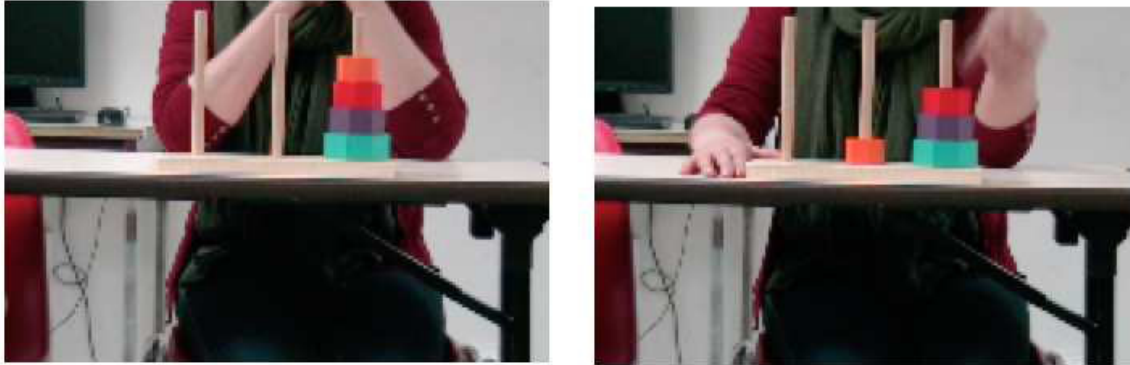


Fig. 6.5 The smallest disk in the four is at the top, and participant grabs it and moves it.

we obtained, we constructed the relationship between objects (including \mathcal{R} , r elements) to the location of the objects and the inclusion relations between them. For example, in the initial state of $t = 0$, the smallest disk is in contact with the second smallest disk, the smaller disk is above the larger disk, and the larger disk is below. The Hanoi Tower with three disks has encoding properties such as object location (bottom and top) and containment relationships. Thus, six objects (three disks and three pegs) each have three encoding attributes.

We can express the smallest disk like this, $o_1 = (o_1^{beneath}, o_1^{on}, o_1^{inclusion})$. We are generally interested in models defined by the form of function composition $f \cdot g$, where f is a function that returns the predicted value y . The function g_ψ is defined to operate on a particular factorization of D (ex: $g_\psi(D) \equiv g_\psi(o_1^1, o_1^2, \dots, o_i^1, \dots, o_m^n)$).

As shown in the figure 6.6, the second smallest disk is in contact with the smallest disk. In this case, it is associated with the second smallest disk below the smallest disk. In other words, $g_\psi(o_1, o_2) = beneath(o_1, o_2)$. However, no disc is on top of the smallest disc. For inclusion, we define that peg1 contains four disks, and peg2 and peg3 do not contain any disks.

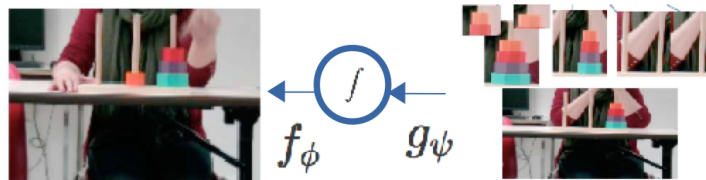


Fig. 6.6 Model of Relational Network

After all, the model we want to predict is given by $y = f_\phi(\sum_{i,j} g_\psi(o_i, o_j))$, where f_ϕ And g_ψ is a multi-layered perceptron. We train to optimize the model. Finally, our model gives

the probability of whether the disk can be moved in different situations. In other words, we want to train the model with three disks and test the prediction and accuracy of four disks.

6.3 Simulation

6.3.1 Experiment: Multi-Layer Perceptron Model

The multilayer perceptron model is the most basic implementation of the Depp neural network. This is presented as one of the most basic modeling methods for learning the rules of Tower of Hanoi. Participants' learning data were extracted from the previous experiment. As mentioned in the section 6.2.1, data coding was performed to learn a multilayer neural network model. (Full data are listed in the appendix) Of these data, 80% were reserved for training and 20% were classified as data for prediction and evaluation.

In this simulation, a trial and error method was applied to calculate the number of hidden layers and the number of nodes. We used the ReLU function for the activation functions of the input and hidden layers, and the sigmoid function and the tanh function for the output layer. Previously, sigmoid and tanh activation functions were used extensively for all layers. However, in recent AI implementations, we are using the ReLU activation function. Since we were able to get better performance with this activation function ReLU.

Table 6.3 Model performance with the Multi-Layer Perceptron Model (12 input layers, 5 hidden layers and 7 output layers)

Index	Net.name	Training accuracy	Test accuracy	Area Under ROC	learning rate
1	MLP 12-5-7	0.875444889	0.866359413	0.799283154	0.1
2	MLP 12-5-7	0.874936461	0.875575960	0.788530465	0.2
3	MLP 12-5-7	0.871377826	0.867895544	0.799283154	0.3
4	MLP 12-5-7	0.873411298	0.870967686	0.795698924	0.4
5	MLP 12-5-7	0.874936521	0.870967686	0.792114695	0.5
6	MLP 12-5-7	0.874936521	0.861751020	0.799283154	0.6
7	MLP 12-5-7	0.873919725	0.870967686	0.777777778	0.7
8	MLP 12-5-7	0.872902870	0.860214949	0.781362007	0.8
9	MLP 12-5-7	0.871377826	0.858678937	0.784946236	0.9

The first MLP (See Table 6.3) was used by employing 12 input layers, 5 hidden layers and 7 output layer and the second (See Table 6.4) used 12 input layers, 10 hidden layers and 7 output layer. For each model, the accuracy and AUROC values according to the change in learning rate were compared and analyzed.

6.3 Simulation

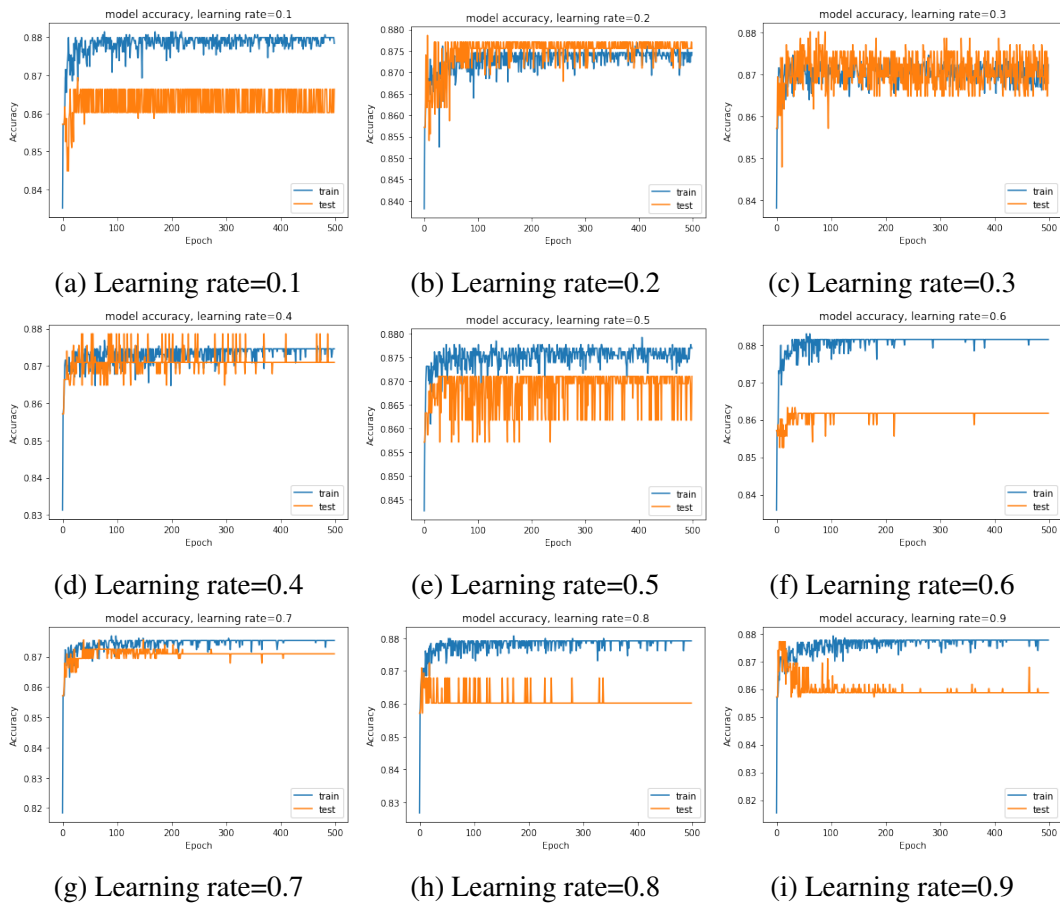


Fig. 6.7 Model loss according to the learning rate of multilayer perceptron model(12 input layers, 5 hidden layers and 7 output layers)

In the case of the first model, Figures ?? and 6.8 show the model's accuracy and loss when the number of training is set to 500. On average, the training accuracy is 87 percent and the test data accuracy is 86 percent. When the learning rate is 0.1, it converges at loss 0.35, and when the learning rate is 0.7, it converges at 0.30. In the case of accuracy and loss, it can be seen that when the number of learning reaches 30-40 times, the accuracy and loss of the data do not increase any more and stabilize.

We also presented the results of the ROC analysis in Figure 6.9. The larger the value of the area under the curve, the better the model performance. Therefore, in the case of the first MLP model, 12-5-7, it can be seen that the lower the learning rate, the higher the model performance.

In the case of the second model, MLP12-10-7, like the first model, the training accuracy is 87% and the accuracy of the test data reaches 86%. Also, the graph of accuracy and loss follows roughly the same increasing trend and stabilization. However, in the case of ROC

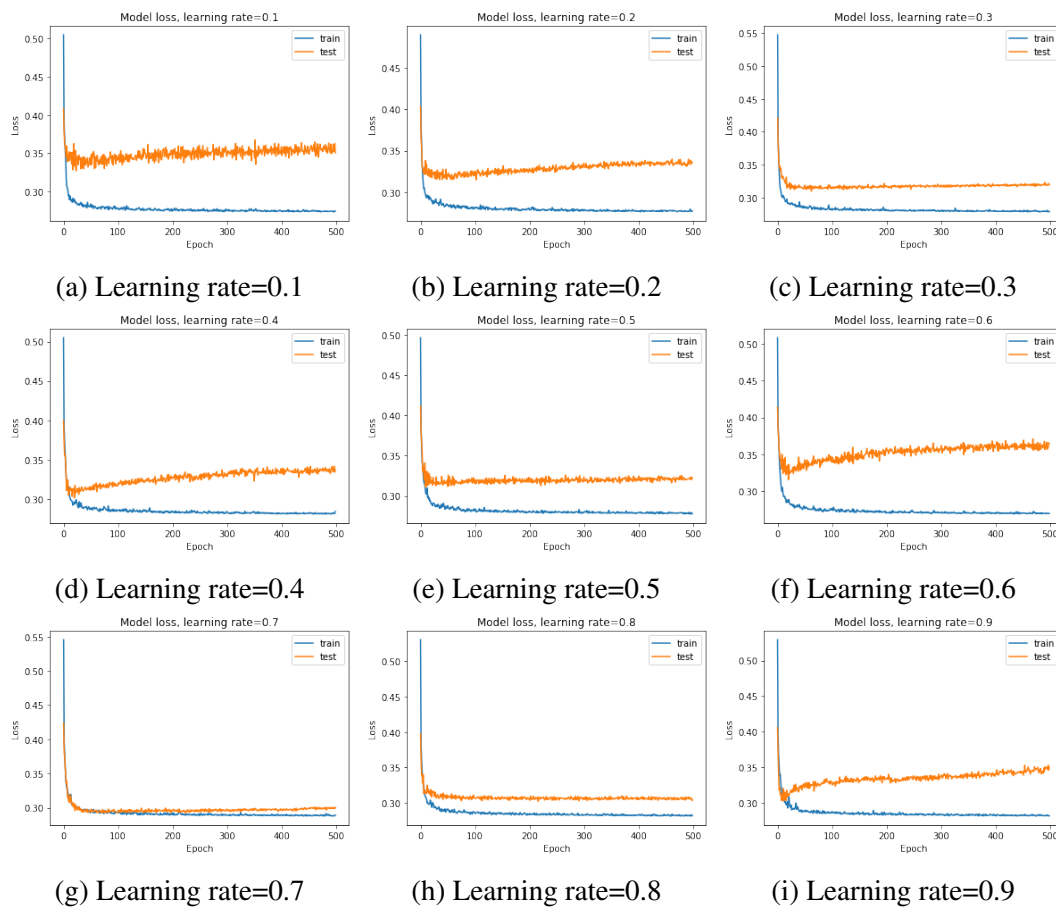


Fig. 6.8 Model loss according to the learning rate of multilayer perceptron model(12 input layers, 5 hidden layers and 7 output layers)

6.3 Simulation

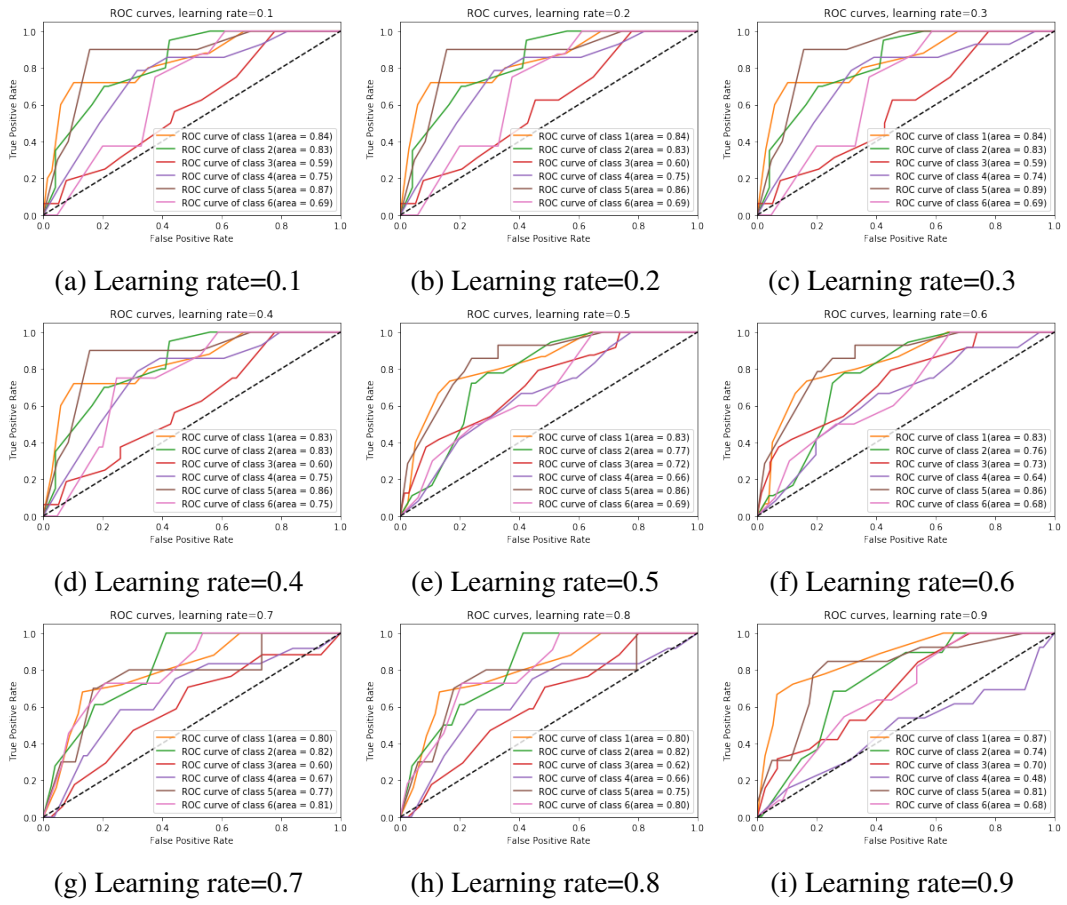


Fig. 6.9 ROC curves according to the learning rate of multilayer perceptron model(12 input layers, 5 hidden layers and 7 output layers)

analysis, it can be seen that the performance of the model is superior to that of the first model. In general, there is no significant difference in the error range, but the performance of the model is better than the case of 5 cases with 10 hidden layers. It can be seen that the smaller the learning rate, the better the performance.

Table 6.4 Model performance with the Multi-Layer Perceptron Model (12 input layers, 10 hidden layers and 7 output layers

Index	Net.name	Training accuracy	Test accuracy	Area under ROC	learning rate
1	MLP 12-10-7	0.875444889	0.875576078	0.826164875	0.1
2	MLP 12-10-7	0.873411357	0.874039947	0.827956989	0.2
3	MLP 12-10-7	0.875444829	0.861751079	0.806451613	0.3
4	MLP 12-10-7	0.870869398	0.867895543	0.813620072	0.4
5	MLP 12-10-7	0.871886194	0.869431614	0.788530466	0.5
6	MLP 12-10-7	0.868835747	0.835637509	0.811827957	0.6
7	MLP 12-10-7	0.872394562	0.867895364	0.790322581	0.7
8	MLP 12-10-7	0.866293788	0.846390187	0.781362007	0.8
9	MLP 12-10-7	0.872394502	0.867895483	0.826164875	0.9

6.3.2 Experiment: Neural Association Models

Neural Association Model(NAM) measures the probability of correlation with event2 occurring under the condition of event1. Consider the case of the Tower of Hanoi, which has three disks. In the initial state, there are three disks in peg A The largest disk is at the bottom and the smallest disk is at the top. These conditions can be assumed to be event1. Under these conditions of event1, a correlation with event2 is formed, which means that the smallest disk can be caught and moved to other empty pegs. The path of movement may be peg B or peg C. This model has constructed a model that can predict the next behavior by selecting the event with the highest probability in the case of various selections. This model extends a Multi-Layer Perceptron model to a Deep Neural Network model so that the correlation between events can be inferred and predicted. The NAM model was trained using the stochastic gradient descend algorithm, and the hidden layers were tested between 1 and 5. The head vector represents the state of the disk, the relation vector represents the resulting behavior, and the relation vector configures the relationship between the two vectors. The average training precision was 91% and the average test data precision was 82%.

6.3 Simulation

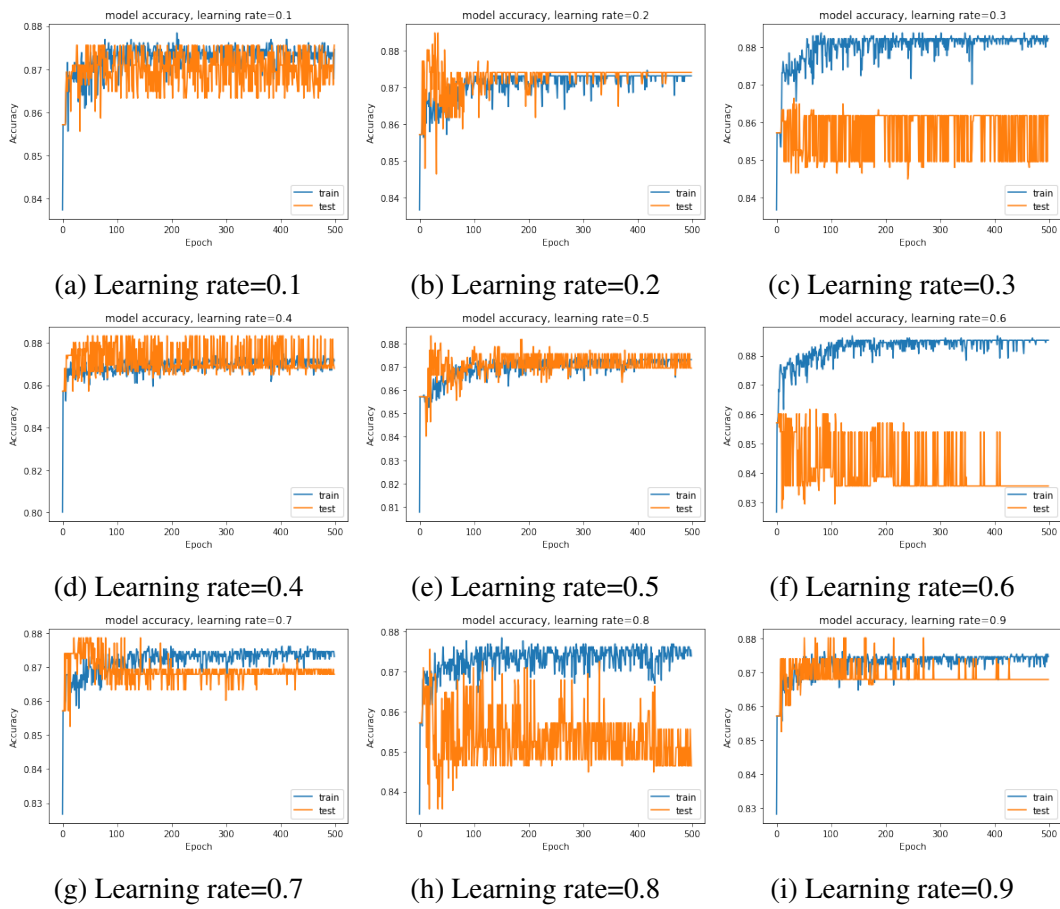


Fig. 6.10 Model accuracy according to the learning rate of multilayer perceptron model(12 input layers, 10 hidden layers and 7 output layers)

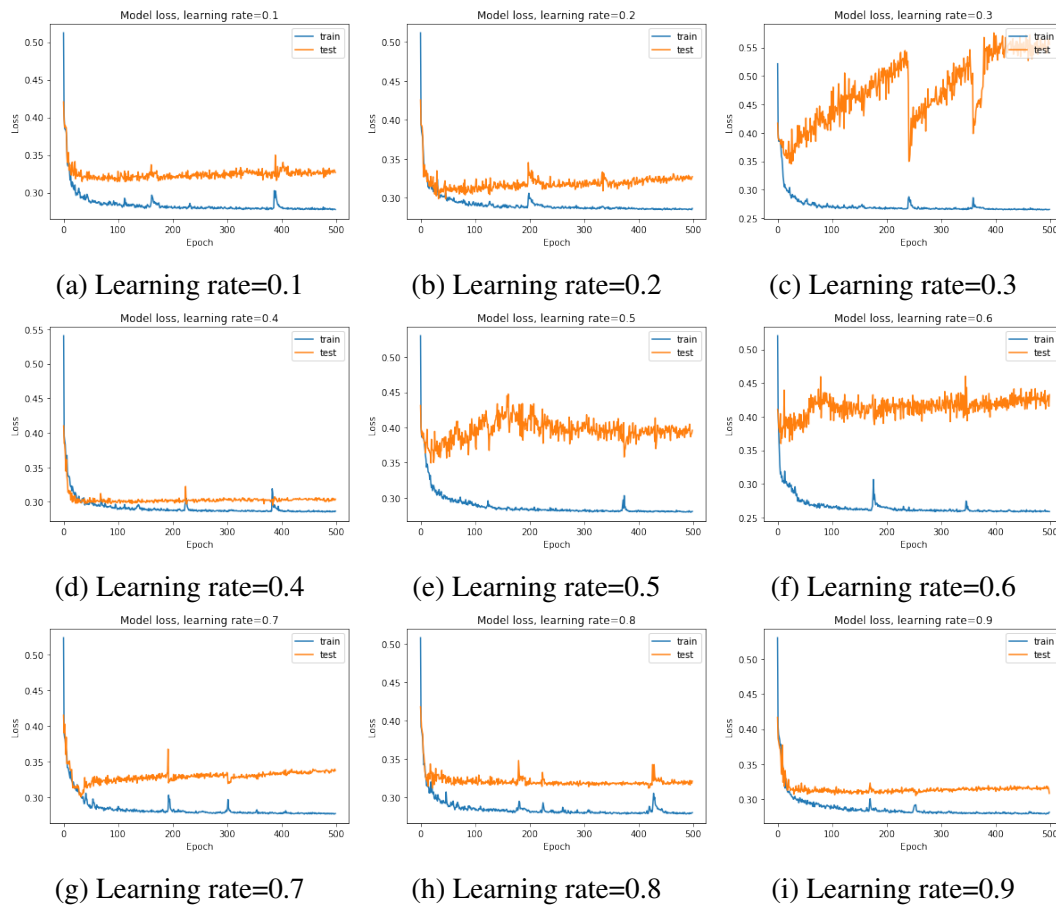


Fig. 6.11 Model loss according to the learning rate of multilayer perceptron model(12 input layers, 10 hidden layers and 7 output layers)

Table 6.5 Coding for the Model of Relational Network

Index	Disk	Relationship		Inclusion			Action
		Beneath	On(=Over)	Peg A	Peg B	Peg C	
1	d_1	1	0	3	0,	0	1
2	d_2	1	1	3	0	0	0
3	d_3	1	1	3	0	0	0
4	d_1	1	0	2	1	0	1
5	d_2	1	0	2	1	0	1
6	d_3	1	1	2	1	0	0

6.3.3 Experiment: Relational Network Model

In the previous section, we trained the Hanoi Tower with 4 disks through the MLP model. We tested the accuracy of the model with random test data and found that the model's performance reached a fairly high level. However, this model was a supervisor training model.

If we want to solve the Tower of Hanoi problem with 5 or 10 disks, we have to code the data again. Therefore, this Relational Network Model is a model that started with the idea of how to infer the rules of the game without much coding of data. It is a model that infers the rules of the game according to the relationship between objects. This makes inferences from trained data when new training data are presented. As mentioned earlier, it is to infer possible behavior through the location of objects and inclusion relationships between objects.

First, we coded the Tower of Hanoi with three disks. As shown in the table 6.5, index 1 is the initial state, and disk 1 is above disk 2, and it is in peg A with other disks. As expressed in section 6.2.2, this state can be expressed as a relationship to an object as follows. : $d_1 = (d_1^{beneath}, d_1^{on}, d_1^{inclusion})$.

Therefore, since the object d2 exists under d1, the value of "relationship" becomes true. On the other hand, since there is no object above d_1 , the characteristic value of "on" is false. And since peg A contains three disks, the value is 3, and since peg B and C do not contain any disks, the value is 0. Therefore, in this state, the value of the action for disk1 is true. In this way, the number of possible cases was coded and learned.

The model has 5 input units, one or two hidden layers and one output unit. There are different neurons (5,10) in each layers. Rectified linear activation functions are used in each hidden layer and a sigmoid activation function is used in the output layer, for binary classification. Reasoning model with different architectural variations was trained to find the rule of Tower of Hanoi. The model with different number of neurons reaches a cross

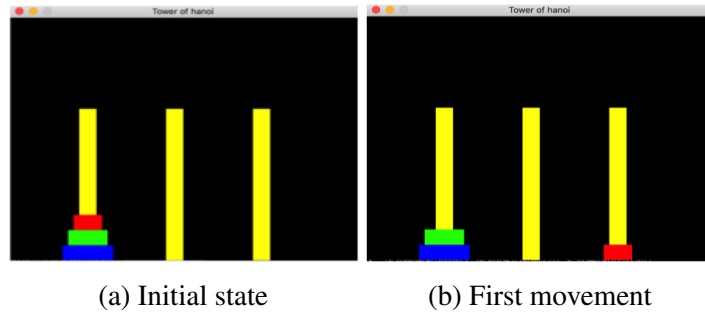


Fig. 6.12 Solution acquired by human

entropy loss below 0.6. However, the fourth model (MLP:5 inputs, two hidden layers with 10 neurons, 1 output) performed well as compared to other models.

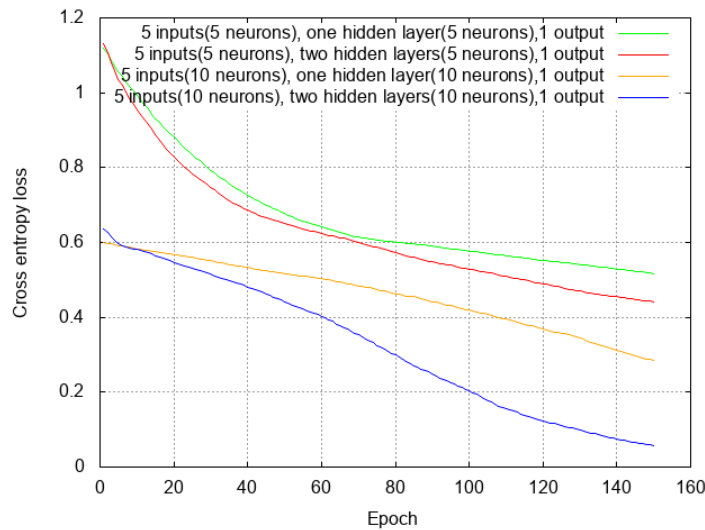


Fig. 6.13 Reasoning model with different number of neurons was trained to find the rule of TOH.

As we can see in Figure 6.13 and Table 6.6, the test accuracy with 3 disks for the first and fourth model are 95.24% and 100%, respectively. When we try to infer the rule of game for The tower of Hanoi with 4 disks, we find our reasoning model achieves 75% prediction accuracy for the first model and 96.67% for the fourth model.

6.4 Conclusion

As we experience, human reasoning and thinking do not require much learning data. Anyone who can learn four discs does not go through tens of thousands of trials and errors to solve

6.4 Conclusion

Table 6.6 Model performance for Tower of Hanoi with 3 disks (training) and 4 disks (prediction)

Model	Accuracy	
	training(%)	prediction(%)
Five inputs (5 neurons), one hidden layer (5 neurons)	95.24	75
Five inputs (5 neurons), two hidden layers (5 neurons)	90.48	76.67
Five inputs (10 neurons), one hidden layer (10 neurons)	90.48	71.67
Five inputs (10 neurons), two hidden layers (10 neurons)	100	96.67

the Hanoi disc problem with 10 discs. We can generally apply other games as well. In games like chess, Othello, Go, etc. we can infer some of the rules of the game just by watching. In order to understand the rules of the game in more detail, we can improve our understanding by getting feedback through conversations with supervisors or opponents. Of course, for game efficiency and victory, strategic considerations are necessary. This is related to the reinforcement learning we will learn in the next chapter. What we've noticed in this chapter is to look at the properties, relationships, and behavior of objects to infer the laws of the game just by looking at them. In this chapter, we performed the simulation of the Tower of Hanoi using the relational network model and neural linkage model. Although the accuracy and prediction did not reach 100 percent, we were able to reason about the rules with a fairly high probability. In this chapter, the reasoning model is studied based on the simulation model derived from such cognitive recognition. In the following chapters, we approach the solution based on the more specific and extended cognitive model.

Chapter 7

Problem solving using Recurrent Neural Network based on the effects of gestures

Models of puzzle problem solving, such as Tower of Hanoi, are based on moves analysis. In a grounded and embodied based approach of cognition, we thought that gestures made to take the discs to one place and place them in another place could be beneficial to the learning process, as well as to the modeling and simulation. Gestures comprise moves, but in addition they are also prerequisites of moves when the free hand goes in one location to take a disc. Our hypothesis is that we can model the solving of the Tower of Hanoi through observing the actions of the hand with and without objects. We collected sequential data of moves and gestures of participants solving the Tower of Hanoi with four dics and, then, train a Recurrent Neural Network model of Tower of Hanoi based on these data in order to find the shortest solution path. In this paper, we propose a approach for change of state sequences training, which combines Recurrent Neural Network and Reinforcement Learning methods.

Publications related to this chapter is :

1. Sanghun, B. and Charles, T. Problem solving using Recurrent Neural Network based on the effects of gestures. 10th International Joint Conference on Computational Intelligence (IJCCI2018), 2018.

7.1 Introduction

Embodied cognition (Varela et al. [139]) deals with the study of thought and behavior. Embodied cognition seems to have a lot of contributions to our understanding of the nature and influence of gestures. Frequently in the literature on embodied cognition (Nathan [97]), gestures are used as a grounding for a mapping between idea and an object in the

world, in order to make meaning easier to understand. The embodied cognitive approach helps to understand the intellectual behavior of perception and behavior based on the body's movements, even if the body's movements are inaccurate or rough.

In this chapter, participants were asked to solve the Tower of Hanoi puzzle (TOH) to observe how gestures affect our perception process (learning, memorization, planning, decision making). As like the Tower of Hanoi puzzle and missionaries-cannibals game, these classic puzzle-solving problems have received plenty of attentions because they do not include domain-specific knowledge. Therefore, it can be used to investigate basic cognitive mechanisms such as search mechanisms and decision making (RICHARD et al. [122]).

Our hypothesis is that we can solve the Tower of Hanoi game not through description of rules of the games, but through observing the movement of the hand and objects. We conduct research experiments to validate this hypothesis. The participants were given two successive tasks : to solve the three-disk Tower of Hanoi task, then to solve this problem with four disks. Through these research experiments, we have obtained the sequential data concerning about the solution of Tower of Hanoi. And then, we have showed how the Recurrent Neural Network(RNN) model can be used to solve the problem of TOH.

The RNN model has recently demonstrated state-of-the-art performance in operations in various domains such as text (Sutskever et al. [133]), motion capture data (Sutskever et al. [132]), and music. (Eck and Schmidhuber [42]). RNN can generate a series of outputs given a series of inputs at the same time, as used for time series forecasts such as stock prices. Therefore, when a continuous state value is presented, a continuous output value can be obtained. Also, successive input values are entered, but all outputs can be ignored and only the last output value can be taken. For example, as we saw in Chapter 6, the output does not change when an instructional or interactive gesture is presented as an input. This is because, despite the process of reasoning, it may not be a state change that has a decisive effect on the next action.

As mentioned earlier, the RNN model has a problem that it will never find the shortest path unless most of the data suggests the shortest path to problem solving. Therefore, we propose a new sequence training method that combines the Recurrent Neural Network and the Reinforcement Learning (RL) method to find this minimal movement.

In RL model, we evaluate the generated moves by comparing these with the target state. This is accomplished by reward mechanism where the favorable moves obtain the higher rewards and the unnecessary moves or repeated moves doesn't gain the higher rewards. To realize the reinforcement learning Q-learning method is used in this chapter.

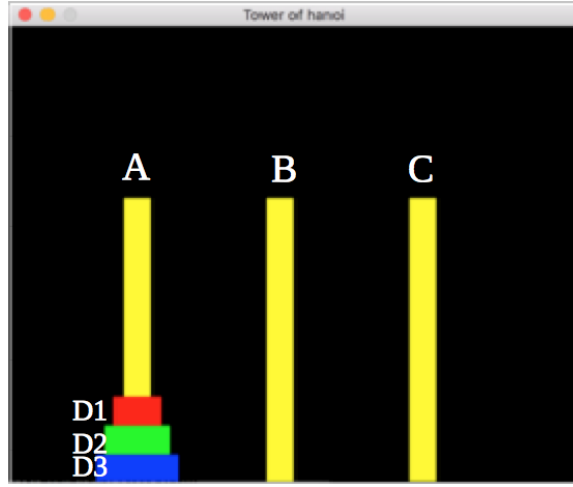


Fig. 7.1 Tower of Hanoi puzzle (initial state)

7.2 Background and Related Work

7.2.1 Tower of Hanoi

The french mathematician Edouard Lucas introduced the Tower of Hanoi (TOH) puzzle in 1883 (Chan [24]). Figure 7.1 shows a standard example of TOH. There are three pegs, A, B, and C. There are three disks (D_1, D_2, D_3) on peg A. The largest disk is at the bottom of peg A and the smallest at the top. The purpose of TOH is to move the entire stack of disks from the initial source peg A to a destination peg C. But there are two constraints: One disk at a time should be moved and a large disk cannot be placed on top of a smaller one.

7.2.2 Recurrent Neural Network

As in previous section of Tower of Hanoi, we know that the problem solving of Tower of Hanoi is considered as a sequential process. In this work, we have used a simple recurrent neural network (RNN) which is based on Elman network ([43]). This network is made up of 3 layers : $\mathbf{x} = (x_1, \dots, x_T)$ a input sequence , output vector sequence denoted as $\mathbf{y} = (y_1, \dots, y_T)$, and $\mathbf{h} = (h_1, \dots, h_T)$ is the hidden vector sequence.(See Figure 7.2)

$$h_t = f(x_t U + h_{t-1} W) \quad (7.1)$$

$$y_t = g(h_t V) \quad (7.2)$$

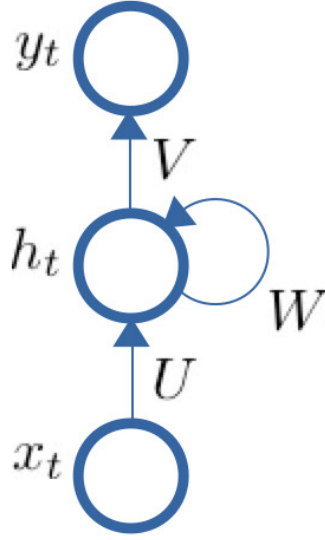


Fig. 7.2 Simple Recurrent Neural Network Architecture

where the U is the weight at the input neuron, W is the weight matrix at the recurrent neuron, $f(z) = \frac{1}{1+\exp^{-z}}$ is sigmoid activation function and $g(z_m) = \frac{\exp^{z_m}}{\sum_k \exp^{z_k}}$ is softmax function.

This model generates one output. The output vector y_t is fed back to the model as a new input. The probability given by the network to the input sequence \mathbf{x} is

$$Pr(\mathbf{x}) = \prod_{t=1}^T Pr(x_{t+1}|y_t) \quad (7.3)$$

and the sequence loss $L(\mathbf{x})$ used to train the network is the negative logarithm of $Pr(x)$:

$$L(\mathbf{x}) = \sum_{t=1}^T \log Pr(x_{t+1}|y_t) \quad (7.4)$$

7.2.3 Reinforcement Learning

An environment takes the agent's current state s_t at time t and action a_t as input, and returns the agent's reward $r(s_t, a_t)$ and next state s_{t+1} (See Figure 7.3). The agent's goal is to maximize the expected cumulative reward over a sequence of action.

An agent interacts with an environment. Given the state (s_t) of the environment at time t , the agent takes an action a_t according to its policy $\pi(a_t|s_t)$ and receives a reward $r(s_t, a_t)$ (Figure 7.3). The objective of Tower of Hanoi is to find the solution in a way that is the shortest possible movement. To do this, we take actions that maximize the future discounted

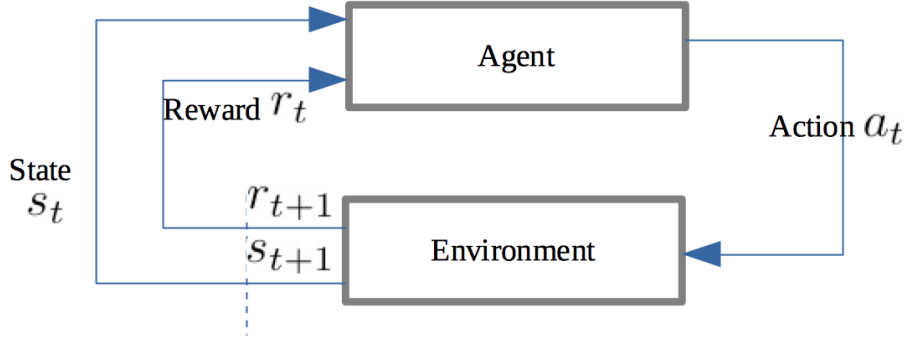


Fig. 7.3 Model of Reinforcement Learning

rewards. We can calculate the future rewards $R_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'}$, where γ is a discount factor for future rewards. In this article, the way of optimal solution is taken by the maximum action-value function:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t) \right) \quad (7.5)$$

7.2.4 Tower of hanoi and Reinforcement Learning

Figure 7.4 shows the state transition graph for the Tower of Hanoi puzzle. This shows the state transition as the disk moves according to the rules of the puzzle. We know that the minimum number of moves required to solve the Tower of Hanoi puzzle is $2^N - 1$, where N is the number of disks. Therefore, 4 disks can be solved with 7 moves. Therefore, the shortest path of the 4 Disk Hanoi Tower Puzzle follows the path of the following state transition graph : (0000)→(0001)→(0021)→ (0022)→(0122)→(0120)→(0110)→(0111)→(2111) →(2112)→(2102)→(2100)→(2200)→(2201)→(2221)→(2222).

Before explaining our learning model, let us see how to use reinforcement learning to solve the Hanoi Tower problem. First of all, we create the reward matrix R . The compensation matrix of the Hanoi Tower for four disks is represented by the following matrix 7.6. The columns of the matrix represent states and the rows represent actions. Compensation for movement from the state of (0,0,1,2) to the state (0,0,2,2) is indicated by zero. The transition from the state of (0,0,1,2) to the state (0,1,0,0) violates the rule and is referred to as $-\infty$. And when moving to the target state (2,2,2,2), the compensation is set to 100.

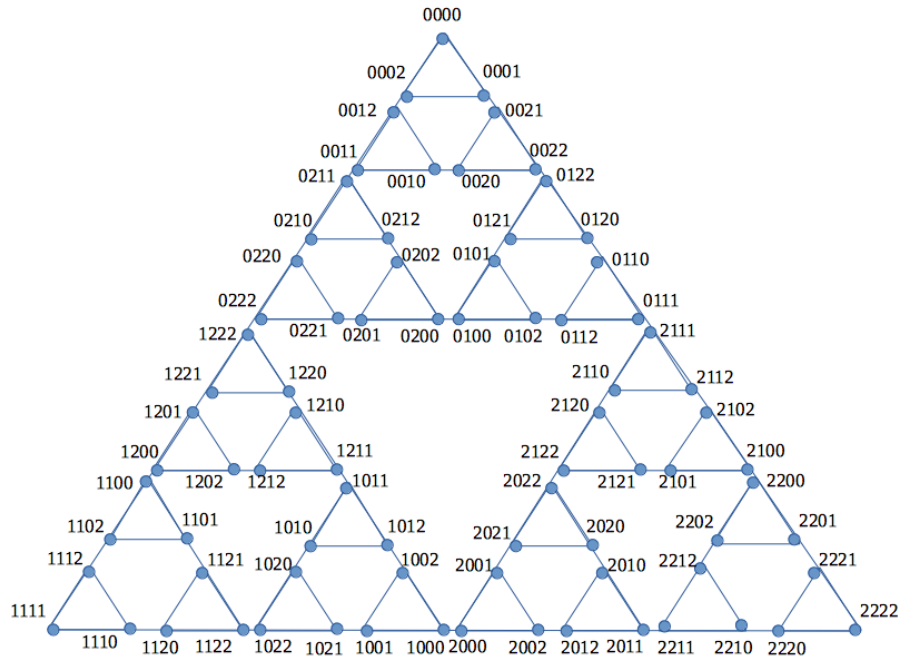


Fig. 7.4 The state-transition diagram corresponding to the 4-disk structure

$$\mathbf{R} = \begin{matrix} & (0000) & (0001) & (0002) & \dots & (2220) & (2221) & (2222) \\ \begin{matrix} (0000) \\ (0001) \\ (0002) \\ \vdots \\ (0222) \\ (1000) \\ (1222) \\ \vdots \\ (2222) \end{matrix} & \begin{pmatrix} -\infty & -\infty & -\infty & \dots & -\infty & -\infty & -\infty \\ -\infty & -\infty & 0 & \dots & -\infty & -\infty & -\infty \\ -\infty & 0 & -\infty & \dots & -\infty & -\infty & -\infty \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ -\infty & -\infty & -\infty & \dots & -\infty & -\infty & 100 \\ 0 & -\infty & -\infty & \dots & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & \dots & -\infty & -\infty & 100 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ -\infty & -\infty & -\infty & \dots & -\infty & -\infty & -\infty \end{pmatrix} & (7.6) \end{matrix}$$

The Q value is then initially set to zero and assigned a value for possible movement in each state. As in Equation 7.5, we maximize the values to find the optimal solution. Figure 7.5 shows the result of simulation with discount factor (gamma) 0.9 and learning factor(alpha) 0.001, 0.01, and 0.1, respectively. An interesting feature of learning is that in

all cases learning is slow initially, and at some point learning reaches the 'inflection point' and then converges to the optimal number of movements at an accelerated rate.

7.3 Model

RNN models can be optimized to predict observations and immediate rewards. On the other hand, RL models can be trained to maximize long-term rewards. We can calculate the probabilities distribution of the observation over all possible actions. These calculated probabilities are helpful for determining the next action. Let us take an example. We can have two possible next actions $\{G1, G3\}$ at time 0 (See Figure 7.7) according to the rules of TOH (Table 7.1).

Table 7.1 Rewards and possible actions at time 0.

<i>G1</i>	<i>G2</i>	<i>G3</i>	<i>G4</i>	<i>G5</i>	<i>G6</i>
1	0	1	0	0	0

The table 7.2 shows the number of probabilistic cases of possible choices given the input at the initial time $t = 0$. The two tables 7.1 and 7.2 show that the next action we choose is $G3$ (From Peg A to Peg C).

Table 7.2 Probabilities and possible actions at time 0.

<i>G1</i>	<i>G2</i>	<i>G3</i>	<i>G4</i>	<i>G5</i>	<i>G6</i>
0.35	0.003	0.60	0.00	0.001	0.001

In the previous section, we found answers to four disks with reinforcement learning theory. In the previous compensation matrix, we set an infinite value when the rule was violated and set a value of 0 when the rule was not violated. If we moved some disks, we didn't break the rules. But consider the case where the shift is the number of cases where the optimal value is not found. Table 7.3 shows how one participant solves the Tower of Hanoi with four discs. If you look at the sixth and seventh lines, you can see that this participant moves the same disc twice in succession. This does not violate the rules, but it is not the optimal solution. So in this case we set negative compensation -1. With this adjusted compensation matrix, we built a new model that combines the calculated probabilities with the adjusted compensation relationship to find the optimal solution. Initially, set the Q matrix value to 0 and update the Q value while performing RNN. Compare the probability value and the Q value of the RNN for the following possible behaviors to ensure that you select the most optimal value.

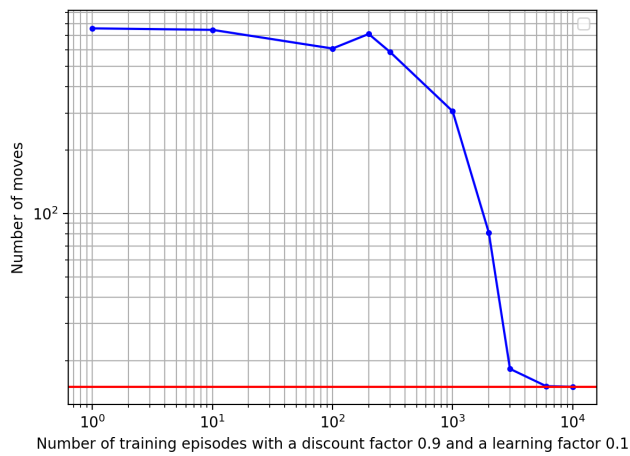
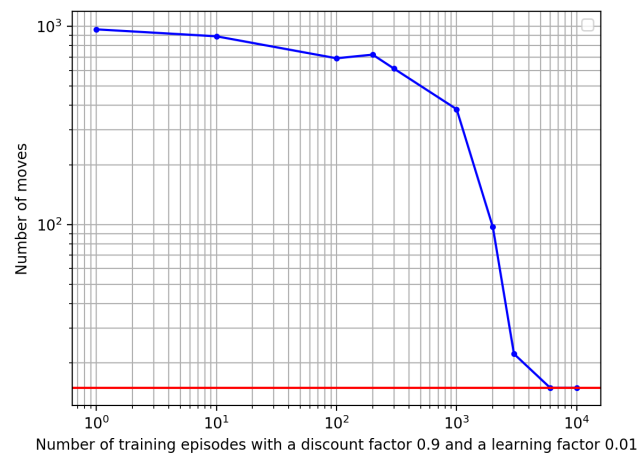
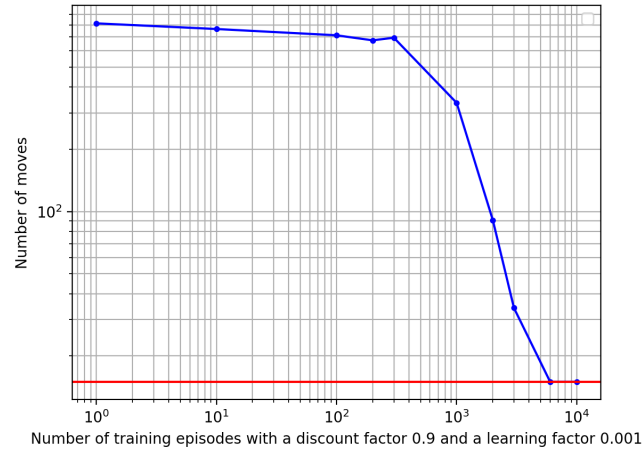


Fig. 7.5 Four disks Hanoi Tower Solution with Reinforcement Learning

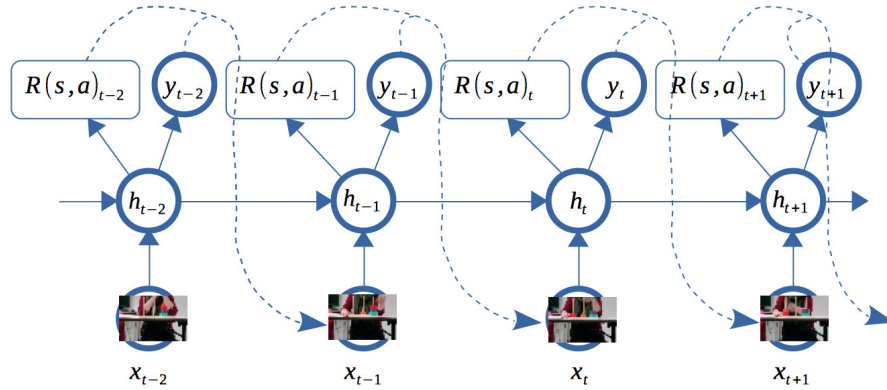


Fig. 7.6 Solving of Tower of Hanoi puzzle with RL and RNN: x_t is the observation, h_t is the hidden state for RNN, y_t is the predicted observation for time $t + 1$, $R(s, a)_t$ is the predicted reward

7.4 Experiments

7.4.1 Coding

We encoded participant's actions from the sequences of observations. If we move a disk from peg A to peg B, we then encoded as $G1$. We can encode all possible actions in the same way (See Figure 7.7). For example, Figure 7.8 illustrates an example of sequence of solution. From initial state($G0$), we moved a disk from peg A to peg B($G1$) and then moved another disk from peg A to peg C($G2$). Next, we decided to take the disk from peg B and put it on another disk in peg C($G2$). In this case, we encode this sequence as $\{G0, G1, G3, G2\}$

7.4.2 Experiment: Tower of Hanoi

We recruited 14 participants (Average age 41, Standard Deviation=8.51). The blind group consisted of 6 women and 1 man (Average age 39, Standard Deviation=6.65). The sighted group consisted of 6 women and 1 man (Average age 43, Standard Deviation=10.30). Sitting down at the table, the participants were then given four disks of the Tower of Hanoi that they had to solve. The instructions were given to the participants. The participants were requested to solve the four disks TOH as we collected their gesture. Through these research experiments, we have obtained the sequential data concerning about the solution of Tower of Hanoi. Table 7.4 shows results of Tower of Hanoi for all participants.

More specifically, the sighted participants made use of their deictic gesture which is used as grounding for a mapping between the object imagined and action. The deictic gesture forces them to remember what they have done in previous attempt. The result

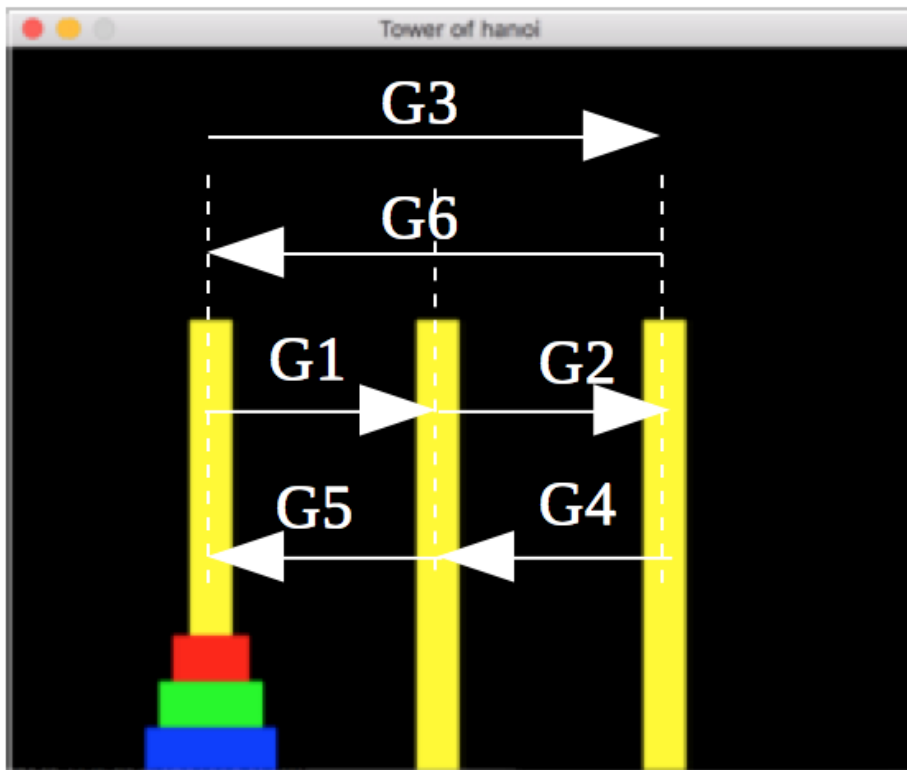


Fig. 7.7 Coding for all possible actions



Fig. 7.8 Clockwise from top left: Initial state(G_0), move a disk from peg A to peg B(G_1), move a disk from peg A to peg C(G_3) and move a disk from peg B to peg C (G_2). We encode the sequence of these movements : $\{G_0, G_1, G_3, G_2\}$

Table 7.3 Solution acquired by a participant

Number	States			Rewards
	Peg A	Peg B	Peg C	
1	$d_1/d_2/d_3/d_4$			0
2	$d_2/d_3/d_4$		d_1	1
3	d_3/d_4	d_2	d_1	1
4	d_3/d_4	d_1/d_2		1
5	d_4	d_1/d_2	d_3	1
6	d_4	d_2	d_1/d_3	1
7	d_1/d_4	d_2	d_3	-1
8	d_1/d_4		d_2/d_3	1
9	d_4		$d_1/d_2/d_3$	1
10		d_4	$d_1/d_2/d_3$	1
11	d_1	d_4	d_2/d_3	1
12	d_1	d_2/d_4	d_3	1
13		$d_1/d_2/d_4$	d_3	1
14	d_3	$d_1/d_2/d_4$		1
15	d_3	d_2/d_4	d_1	1
16	d_2/d_3	d_4	d_1	1
17	$d_1/d_2/d_3$	d_4		1
18	$d_1/d_2/d_3$		d_4	10
19	d_2/d_3		d_1/d_4	10
20	d_3	d_2	d_1/d_4	10
21	d_3	d_1/d_2	d_4	10
22		d_1/d_2	d_3/d_4	15
23	d_1	d_2	d_3/d_4	15
24	d_1		$d_2/d_3/d_4$	18
25			$d_1/d_2/d_3/d_4$	20

shows that the number of deictic gestures for this group is correlated with the total duration [$r = .44, p < 0.019$]. Meanwhile, the blind people build their mental representation with their hands through touch. For the blind participants, the gestures added action information to their mental representation of the tasks by touching the disk or rotating it. the number of gesture for the blind people is correlated with the total duration. [$r = .496, p < 0.0072$]

7.4.3 Experiment: RNN+RL model

Our hypothesis is to solve the Tower of Hanoi puzzle through observing the movement of the hand and objects. To do this, we conducted experiment by training our combined model(RNN+RL) on the sequential data obtained in previous experiments on TOH solution.

Table 7.4 Results of Tower of Hanoi for 15 participants

Participant	Number of moves	Participant	Number of moves
1	15	9	44
2	25	10	35
3	21	11	23
4	48	12	38
5	23	13	15
6	24	14	34
7	32	15	26
8	30		

First of all, we evaluate and compare the performance of RNN model on this sequential data. And then the combined model (RNN+RL) is carried out.

The weights of all networks are initialized to random values uniformly distributed in the interval from $[-1/\sqrt{n}, 1/\sqrt{n}]$, where n is the number of incoming connections. To train our model we minimize the loss function for our training data.

RNN model

Codings based on the acquired data are listed in Appendix B.1 Tower of Hanoi Resolution Dataset for Recurrent Neural Network. Based on these data, the results of simulating the RNN model are shown in the table 7.5. In state (0000) there is a 59% chance of choosing G1 and a 36% chance of choosing G3. The RNN model makes the choice with the highest probability of a possible choice. In this way, the most optimal solution chosen by our model proceeds to the selection indicated in red in the table, and the result of the state transition is as follows. (0000), (0001), (0021), (0022), (0122), (0121), (0120), (0110), (0111), (2111), (2112), (2102), (2101), (2121), (2120), (2121), (2101), (2100), (2201), (2201), (2221), (2222)

Comparing this result with the minimum paths of participants in 7.4, the total number of disk movements is 21. It is more than the minimum path 15 of participants 1 and 13, but it is the third fastest path (See Figure 7.9). To find the minimum path 15 in the RNN model, more data is needed. In particular, it is possible to have a lot of data about the minimum path. Then, we applied the RNN-RL model to find the minimum path without additional training data.

Table 7.5 State Transition and Probabilistic Selection of RNN Models

State	G3	G1	G2	G4	G6	G5	Choice
(0000)	0.364	0.591	0.002	0.0	0.002	0.0	G1
(0001)	0.946	0.009	0.001	0.002	0.003	0.0	G3
(0021)	0.0	0.003	0.493	0.01	0.397	0.008	G2
(0022)	0.008	0.902	0.014	0.0	0.0	0.001	G1
(0122)	0.0	0.002	0.009	0.619	0.299	0.0	G4
(0121)	0.031	0.0	0.001	0.011	0.096	0.824	G5
(0120)	0.005	0.0	0.065	0.874	0.0	0.013	G4
(0110)	0.057	0.894	0.001	0.0	0.004	0.015	G1
(0111)	0.97	0.0	0.0	0.003	0.002	0.002	G3
(2111)	0.0	0.001	0.95	0.016	0.001	0.004	G2
(2112)	0.004	0.002	0.001	0.001	0.009	0.938	G5
(2102)	0.001	0.0	0.002	0.947	0.019	0.0	G4
(2101)	0.892	0.001	0.0	0.0	0.003	0.055	G3
(2121)	0.0	0.004	0.003	0.003	0.021	0.905	G5
(2120)	0.025	0.861	0.048	0.004	0.0	0.0	G1
(2121)	0.02	0.0	0.0	0.039	0.892	0.004	G6
(2101)	0.027	0.0	0.02	0.044	0.01	0.765	G5
(2100)	0.001	0.058	0.89	0.001	0.0	0.0	G2
(2201)	0.018	0.802	0.0	0.0	0.06	0.0	G1
(2201)	0.748	0.001	0.0	0.019	0.001	0.0	G3
(2221)	0.0	0.001	0.97	0.019	0.001	0.0	G2

7.4.4 Results

After training for the model RNN, we obtained the following shortest path : G1,G3,G2,G1,G4,G5,G4, G1,G3,G2,G5,G4,G3,G5,G1,G6,G5,G2,G1,G3,G2(21 movements). The training error is shown in figure 7.10.

Compared to the experimental results in table 7.4, this RNN model shows good performance improvement. Nevertheless, this result is not the fastest solution. According to the table 7.4, the first participant and the thirteenth participant find the fastest solution. On the other hand, from figure 7.11, we can see that this combined model(RNN+RL) improves the performance compared with RNN model.

7.5 Conclusions

As participants have a difficulty triggering simulations with visual object, deictic gesture for the sighted participants plays a central role in generating visual inferences. Meanwhile, the blind participants have a difficulty solving TOH because of the lack of tactile object. The

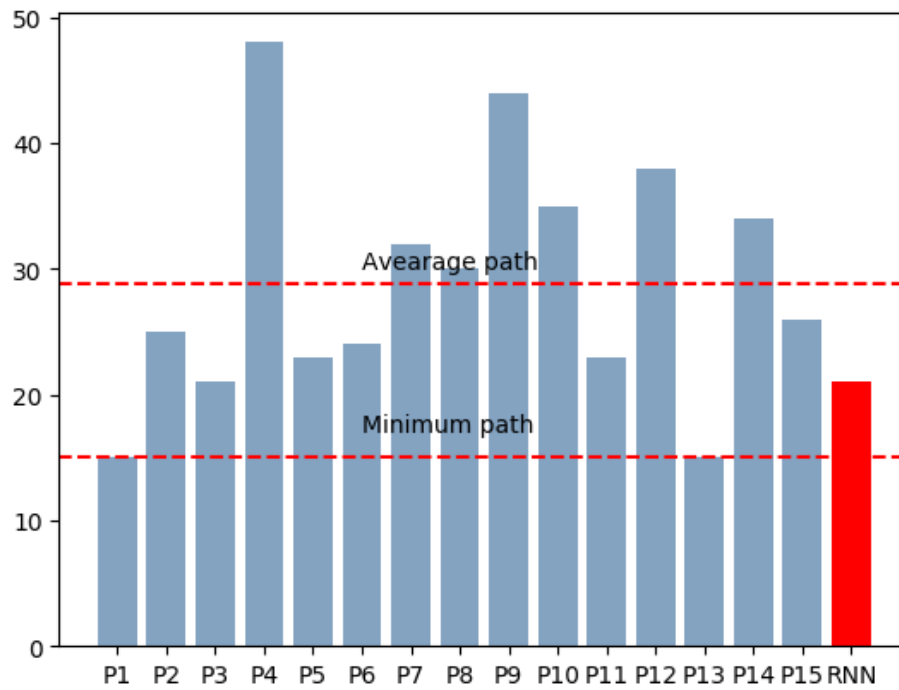


Fig. 7.9 Comparing Results with Participants and the RNN Model

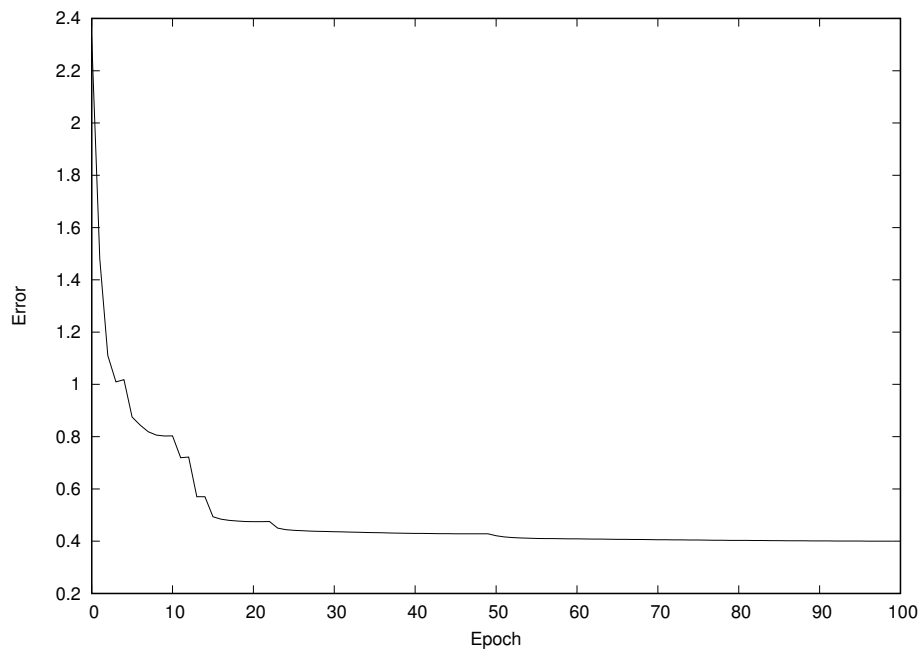


Fig. 7.10 RNN train error at each epoch

7.5 Conclusions

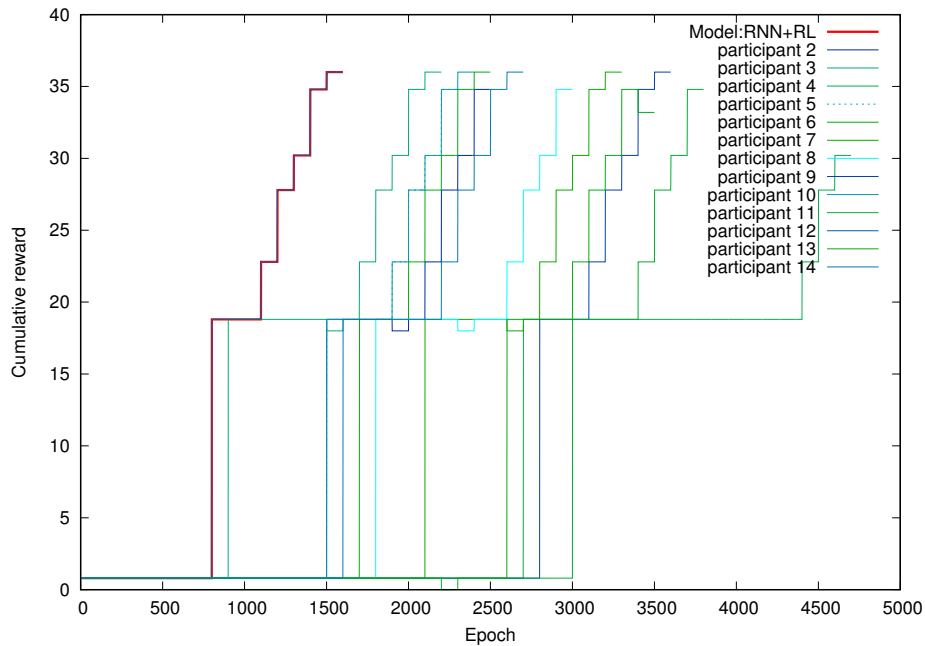


Fig. 7.11 Cumulative reward graph for RNN+RL model

interaction gestures for these participants play an important role in building their mental representation.

Base on this experiment, we conducted an experiment (Tower of Hanoi task) in order to test the effects of gesture. In this work, we propose a new approach that combines recurrent neural network and reinforcement learning to solve the TOH task through observing the movement of the hand and objects. Our RNN+RL model finds the optimal solution for TOH. However, although our sequential data comprises the movement action on disk, this was not enough to describe the reasoning process for deictic gestures and interaction gestures, including touching the disk and rotating it. Later, we will implement more sophisticated modeling to understand the TOH problem solving reasoning processes.

Chapter 8

Using Convolutional Neural Networks and Recurrent Neural Network for Human Gesture Recognition and Problem Solving

Problem solving, such as Tower of Hanoi, is a mental process that involves identifying problems, developing strategies and organizing knowledge. We have acquired a series of data from participants solving the Tower of Hanoi problem to understand human behavior and cognitive reasoning in the problem-solving process. These data have a lot of semantic information, including the objects around participants, actions and interactions with objects. Therefore, given a series of data that are all labeled with a multiple categories, we have predicted these categories for a novel set of test data with the help of convolutional neural networks (CNNs). In addition to category inference, we use the Long Short-Term Memory(LSTM) model to do behavioral reasoning from sequential data. Papers submitted related to this chapter is :

1. Sanghun, B. and Charles, T. Using Convolutional Neural Networks and Recurrent Neural Networks for Humain Gesture Recognition and Problem Solving, 2018. Computational Intelligence, IJCCI 2018, Revised Selected papers

8.1 Introduction

The theory of embodied cognition (Varela et al. [139], Barsalou [11]) suggests that our body influences our thinking. Embodied cognition approaches made contributions to our

understanding of the nature of gestures and how they influence learning. Frequently in the literature on embodied cognition (Nathan [97]), gestures are used as grounding for a mapping between thinking and real objects in the world, in order for the easy catching of meanings.

Classical puzzle-like problem, such as Tower of Hanoi puzzle and missionaries-cannibals have received a lot of attentions because they do not involve domain-specific knowledge and can, therefore, be used to investigate basic cognitive mechanisms such as search and decision-making mechanisms (RICHARD et al. [122]). To analyze the effect of gestures on problem solving cognitive processes (learning, memorizing, planning, and decision-making), we recruited participants who were asked to solve the puzzle of Tower of Hanoi (TOH).

Our hypothesis is that we can model the solving processes of the Tower of Hanoi, not simply through the description of the disks' moves according to the rules, but through observing the movements of the solver's hand with or without the disks. In order to test this hypothesis, we carried out an experiment for which participants were given two successive tasks: to solve the three-disk Tower of Hanoi task, then to solve this problem with four disks.

We investigated how gestures ground the meaning of abstract representations used in this experiment. The gestures added action information to their mental representation. The deictic gesture used in this experiment forces the participants to remember what they have done in previous attempt. The purpose of our research through this experiment is to infer the problem solving or the rules of game through modeling of human behavior. We collected all of the sequential gestures data that bring reaching the goal. These data were used to model and simulate how to solve the problem of TOH with Convolutional Neural Networks and Long Short-Term Memory (LSTM).

Recently, Deep CNNs (LeCun et al. [79], Szegedy et al. [135]) have been very successful on single-label object classification, i.e., ImageNet Large Scale Visual Recognition Challenge (Szegedy et al. [135], Russakovsky et al. [126]). These algorithms have been very successful for a variety of tasks including image classification (Krizhevsky et al. [73], Oquab et al. [103], Sharif Razavian et al. [129]), object detection (Girshick et al. [47], Sermanet et al. [128]), and others (Cho et al. [26]). In this paper, we pay our attention to the multi-label image classification to understand the Tower of Hanoi solution. The objects around us, surrounding scenes, actions, and interactions with objects all contain semantic information as like our everyday life. The images obtained from real world have many and complex categories. Likewise, our collected sequential images also contain various semantic information: The objects around participant, their actions, and interactions with objects. Therefore, the task with multi-label image classification helps to understand more complex semantic information.

State-of-the-art have recently demonstrated performance' LSTM models across a variety of tasks in domains such as text (Sutskever et al. [133]), motion capture data (Sutskever et al. [132]), and music (Eck and Schmidhuber [42]). In particular, LSTM can be trained for sequence activation while processing real data sequences. Therefore, we modeled the Tower of Hanoi solving processes with the help of LSTM method.

8.2 Background and Related Work

8.2.1 Tower of Hanoi

The french mathematician Edouard Lucas introduced the Tower of Hanoi (TOH) puzzle in 1883 (Chan [24]). Figure 8.1 shows a standard example of TOH. There are three pegs, A, B, and C. There are four disks (d_1, d_2, d_3, d_4) on peg A. The largest disk is at the bottom of peg A and the smallest at the top. The goal of TOH is to move the whole stack of disks from the initial source peg A to a destination peg C. There are three rules as constraints: One disk at a time should be moved, in a location, the smallest disk is the one to take and a large disk cannot be placed on top of a smaller one. The minimum number of moves needed to solve TOH with n disks is denoted by $2^n - 1$.

8.2.2 Recurrent Neural Network

In our previous work (Bang and Tijus [5]), we used a simple Recurrent Neural Network which is based on Elman network (Elman [43]). As is well known, the simplest RNN model has a vanishing gradient problem. That's why, we implemented the gated activation functions, such as the long short-term Memory(LSTM) (Hochreiter and Schmidhuber [60]) to overcome the limitations of our model. Because the TOH solving process is a sequential process. In this work, we have used a LSTM-RNN network which computes a mapping from an $\mathbf{x} = (x_1, \dots, x_T)$ a input sequence to an output vector sequence denoted as $\mathbf{y} = (y_1, \dots, y_T)$ by calculating the network unit activation using the following equations :

$$i_t = \delta(W_{ix}x_t + U_i h_{t-1} + b_i) \quad (8.1)$$

$$f_t = \delta(W_{fx}x_t + U_f h_{t-1} + b_f) \quad (8.2)$$

$$o_t = \delta(W_{ox}x_t + U_o h_{t-1} + b_o) \quad (8.3)$$

$$c_t = f_t \circ c_{t-1} + i_t \gamma(W_{cx}x_t + U_c h_{t-1} + b_c) \quad (8.4)$$

$$h_t = o_t \circ \delta(c_t) \quad (8.5)$$

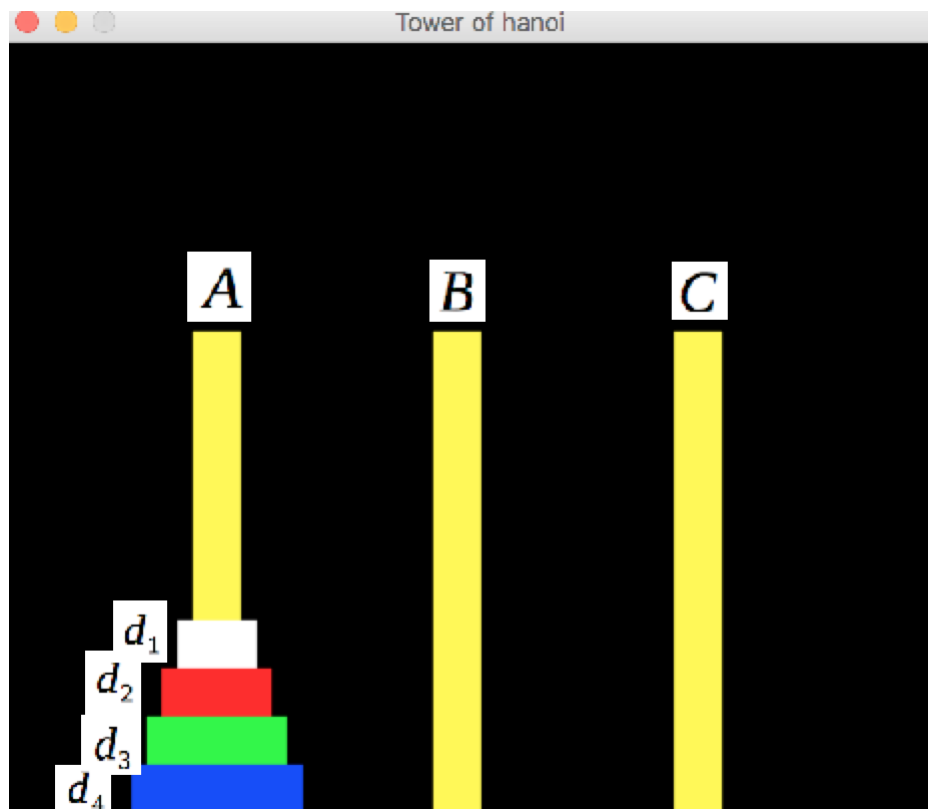


Fig. 8.1 Four disks in Tower of Hanoi puzzle

where W and U terms denote weight matrices, the b terms denote bias vectors(ex: b_i is the input gate bias vector), σ and γ denote activation functions, x_t is input vector, f_t is forget gate's activation vector, i_t is input gate's activation vector, h_t is hidden state vecor, and o_t denotes output gate's activation vector.

8.2.3 Convolutional Neural Networks

Convolutional neural networks(CNNs) are a type of artificial neural network designed to be easy to apply to images. CNNs were first proposed in 1998 by Lecun et al (LeCun et al. [79]) and consists of a convolution layer and a pooling layer, unlike the structure used in general multilayer perceptron (See Figure 8.2). Convolutional Neural Networks are a very famous neural network model used of image classification. In particular, We were interested in work of Gong et. al (Gong et al. [51]) on the recent multi-label deep convolutional ranking net. Because we should take into account various situations and behaviors in the problem solving process.

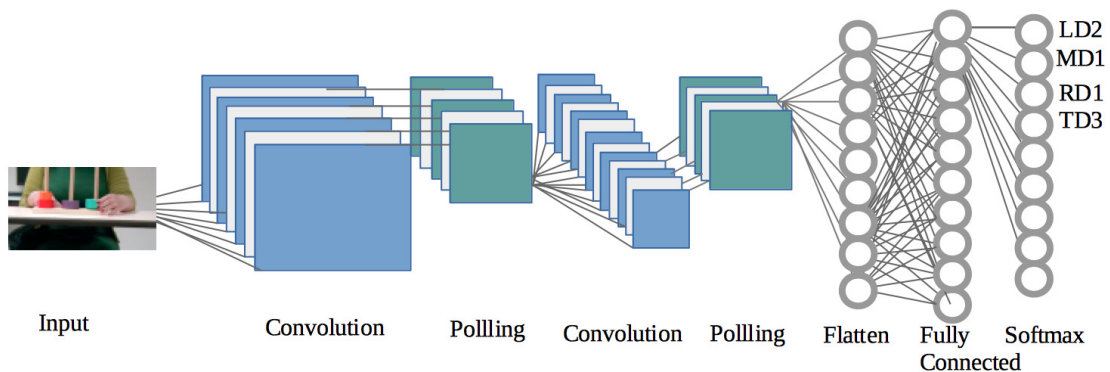


Fig. 8.2 The Deep Convolutional Neural Network for multi-label classification

8.2.4 Reinforcement Learning

An environment takes the agent's current state s_t at time t and action a_t as input, and returns the agent's reward $r(s_t, a_t)$ and next state s_{t+1} (See Figure 3). The agent's goal is to maximize the expected cumulative reward over a sequence of action.

An agent interacts with an environment. Given the state(s_t) of the environment at time t , the agent takes an action a_t according to its policy $\pi(a_t|s_t)$ and receives a reward $r(s_t, a_t)$ (See Figure 8.3). The objective of Tower of Hanoi is to find the solution in a way that is the shortest possible movement. To do this, we take actions that maximize the future discounted

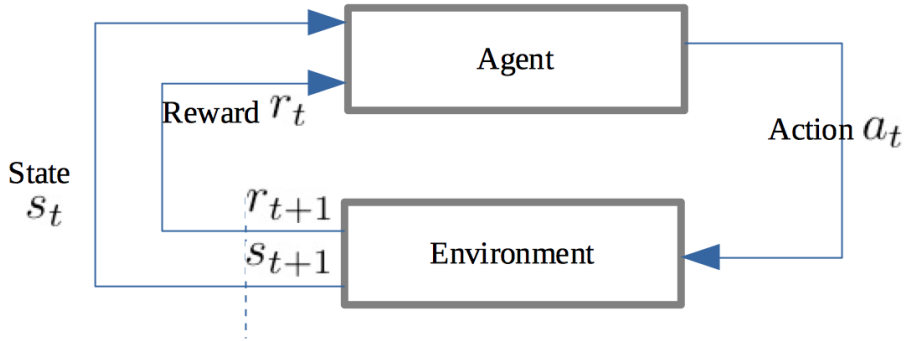


Fig. 8.3 Model(Reinforcement Learning) extracted from [5]

rewards. We can calculate the future rewards $R_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'}$, where γ is a discount factor for future rewards. In this article, the way of optimal solution is taken by the maximum action-value function:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right) \quad (8.6)$$

where $\alpha \in [0, 1)$ is the learning rate sequence, and γ is the discount factor.

8.3 Model

In this chapter, CNN and LSTM-RL methods are introduced. Our method is divided into two parts. The first method consists of extracting the features and predicting the multi-label by means of Convolutional Neural Networks method. The first model doesn't just mean classification. In the next chapter, it will be extended to a model for examining the relationship between natural language processing and images. This will allow not only to classify the elements in the photo, but also to be able to explain. The second method combines Long Short-Term Memory Recurrent Neural Network and Reinforcement Learning methods to obtain the shortest solution path.

8.3.1 CNN Multi-label categorization

The video data was converted into a series of images. In addition, for each image, a multi-label classification was performed for all the information on the participant's actions, hand movements, holding the disk, and changing the location. Sample coding data for continuous images are presented in Appendix B.3.1 Convolutional Neural Network. Let us take a more specific example. If the participant's image is given, as shown in Figure 8.5, the image is labeled LD1, MD1, RD2, and TD1. LD1 means one disk is on the left and MD1 means the

disk is in the middle. RD2 means two disks on the right. Finally, in the case of TD1, it means to grab disk 1. If you describe multiple categories in this way, you can summarize them as follows:

LD1-LD4(There are one to four disks on the left side), MD1-MD4(There are one to four disks in the middle), RD1-RD4(There are one to four disks on the right), TD1-TD4(Take disk 1, disk 2, disk 3, disc 4), PD1-PD4(Put disk 1, disk 2, disk 3, disk 4). The per-class precision is shown in figure 8.4.

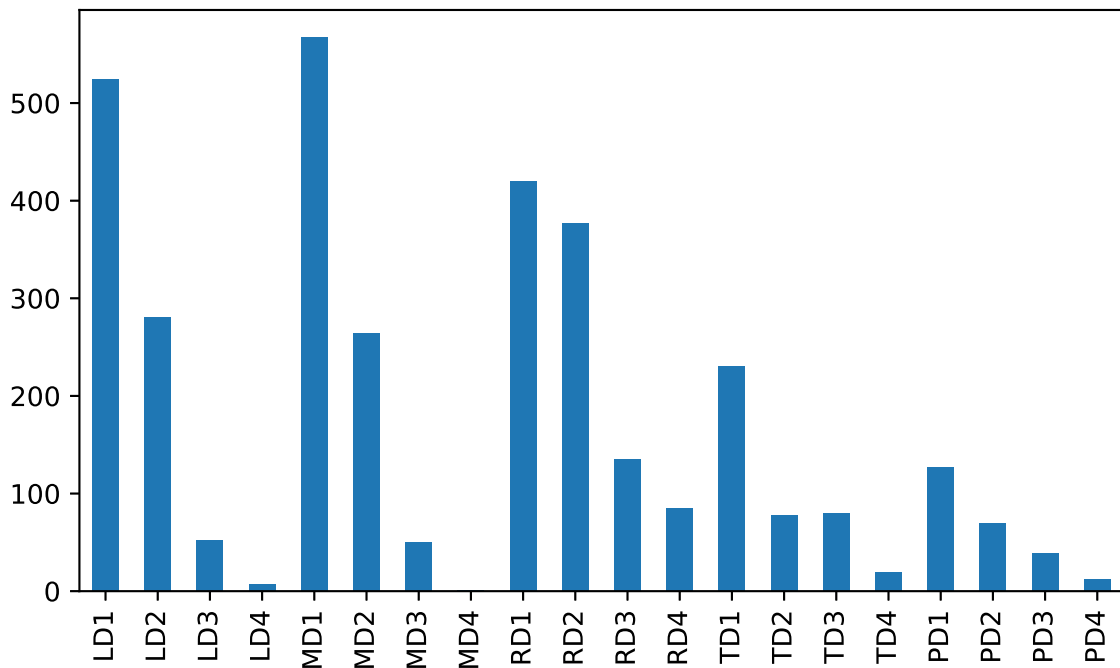


Fig. 8.4 The per-class barplots of input images. A description of the multi-category is as follows: LD1-LD4(There are one to four disks on the left side), MD1-MD4(There are one to four disks in the middle), RD1-RD4(There are one to four disks on the right), TD1-TD4(Take disk 1, disk 2, disk 3, disc 4), PD1-PD4(Put disk 1, disk 2, disk 3, disk 4).

Table 8.1 shows our proposed architecture. This model with 4 layers takes a input image $80 \times 80 \times 3$ (3:color image,1:black & white image) and predicts a 20 dimensional tag vector. And our CNN model has 869,748 trainable parameters.

8.3.2 Combined Recurrent Neural Network and Reinforcement Learning

In the previous model, if the learning was performed in a static environment given data, Reinforcement Learning proceeds with the agent taking some action for the given environment



Fig. 8.5 Labelling for the required images: There is one disk on the left (LD1) and the middle peg (MD1), and two disks on the right peg (RD2). The participant is holding d_1 (TD1).

and getting some reward from it. At this time, the agent learns to maximize the reward. That is, reinforcement learning is an algorithm that encompasses the process of collecting data in a kind of dynamic environment. Based on the multi-label classification of CNN, we predicted possible action between sequential images (See Figure 8.6).

After learning of the image through CNN is completed, we present the image with all four disks on the right. The composite model of reinforcement learning and recurrent neural networks presents probability values for possible next actions for the presented image. In particular, LSTM was applied when using a recurrent neural network in this model. Table 8.2 shows the successive images and predictable probabilities for each classification. For the first image, there is a 100 percent chance that there are four disks on the right, and a 45 percent chance that the participant will grab disk 1. Likewise, there are subtle differences between the 2nd and 5th images, but our model predicts the same result as the first image. Therefore, our model gives the command to catch disk 1 as the predicted value for the next action. In the seventh figure, the probability value of holding disk 1 decreases, and the probability value of dropping disk 1 increases to 14 percent. We know intuitively that we're putting down Disc 1 because we caught Disc 1 in the previous action. However, the probability value is low for our model to make a definitive decision. After that, the probability of catching Disc 1 drops to 17 %, and the probability of dropping Disc 1 increases to 27 %. At this time, our model

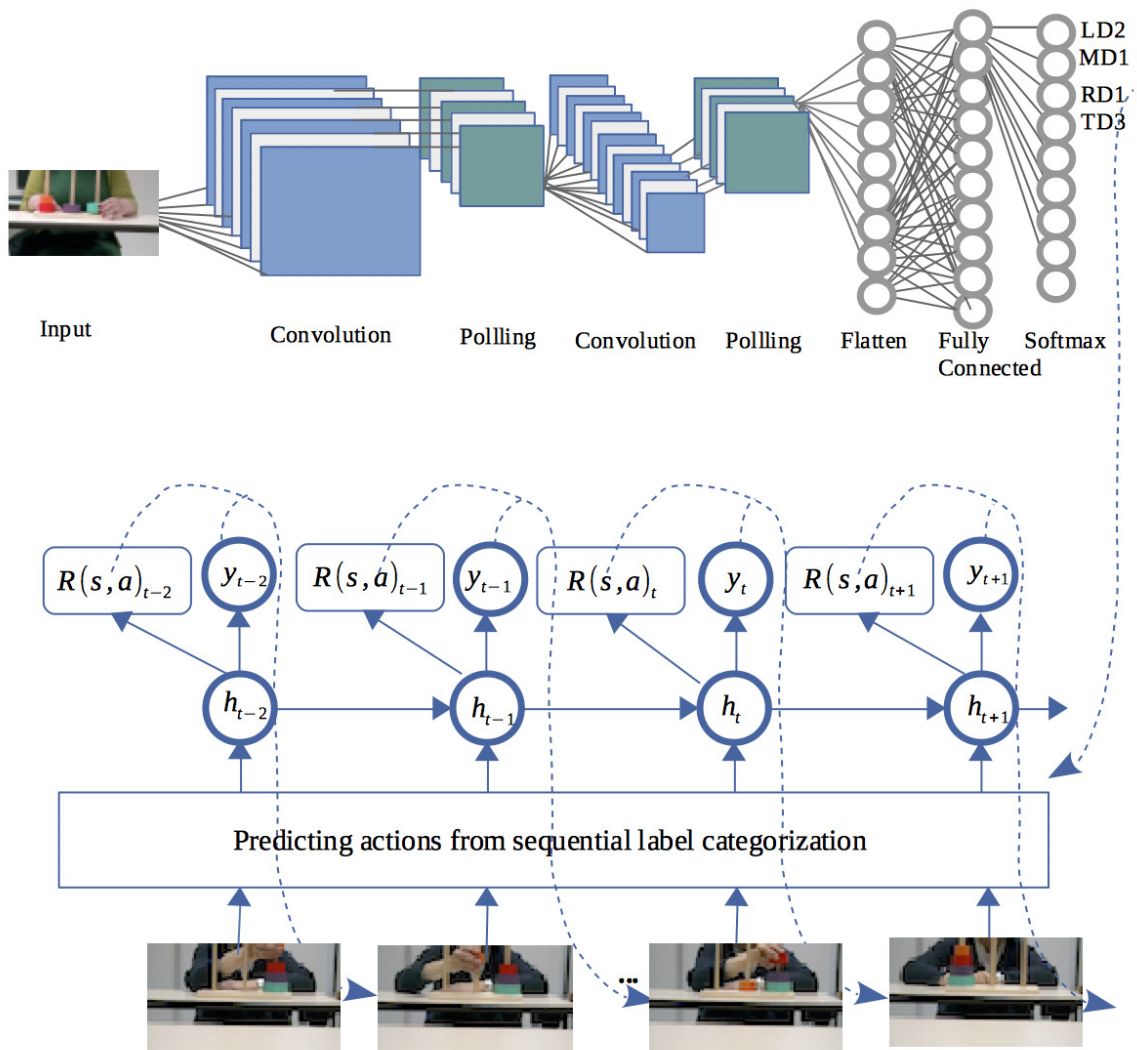


Fig. 8.6 Solving of Tower of Hanoi puzzle with CNN and LSTM-RL: x_t is the observation, h_t is the hidden state for LSTM, y_t is the predicted observation for time $t + 1$, $R(s, a)_t$ is the predicted reward

Table 8.1 The summary of Our Convolutional Neural Networks model

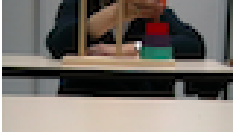






Layer (type)	Output Shape	Param
conv2d ₁ (Conv2D)	(None, 76, 76, 16)	1216
max pooling2d 1 (MaxPooling2)	(None, 76, 76, 16)	0
dropout 1 (Dropout)	(None, 76, 76, 16)	0
conv2d 2 (Conv2D)	(None, 72, 72, 32)	12832
max pooling2d 2 (MaxPooling2)	(None, 36, 36, 32)	0
dropout 2 (Dropout)	(None, 36, 36, 32)	0
conv2d 3 (Conv2D)	(None, 32, 32, 64)	51264
max pooling2d 3 (MaxPooling2)	(None, 16, 16, 64)	0
dropout 3 (Dropout)	(None, 16, 16, 64)	0
conv2d 4 (Conv2D)	(None, 12, 12, 128)	204928
max pooling2d 4 (MaxPooling2)	(None, 6, 6, 128)	0
dropout 4 (Dropout)	(None, 6, 6, 128)	0
flatten 1 (Flatten)	(None, 4608)	0
dense 1 (Dense)	(None, 128)	589952
dropout 5 (Dropout)	(None, 128)	0
dense 2 (Dense)	(None, 64)	8256
dropout 6 (Dropout)	(None, 64)	0
dense 3 (Dense)	(None, 20)	1300
Total params: 869,748		
Trainable params: 869,748		
Non-trainable params: 0		

chooses to put down the disk out of two actions (hold the disk and put it down). This will match G4. As show in fig 8.7 , it means "move a disk from right peg to the middle peg".

After predicting actions from sequential images, LSTM models can be optimized to predict observations. In this way, the next action is predicted, and when the next image is presented, it is predicted again by the trained first model CNN, and the next action is decided. On the other hand, RL models can be trained to maximize long-term rewards. We can calculate the probabilities distribution of the observation over all possible actions. These calculated probabilities are helpful for determining the next action. As shown in the figure 8.7, we can have two possible next actions $\{G1, G3\}$ at time 0 according to the rules of TOH (See Table 8.3).

Table 8.4 is achieved by looping an output of the network at time 0 with the input of the network. Therefore, base on the table 8.3 and table 8.4, we choose the next action G3 (From Peg A to Peg C).

Table 8.2 Sequential images and probabilities

Sequential image	Multi-label and probability	Predictive Action
	['RD4', 'TD1', 'TD2', 'TD3', 'PD1', 'PD2', 'MD1', 'LD1'] ['1.00', '0.45', '0.01', '0.00', '0.00', '0.00', '0.00', '0.00']	TD1
	['RD4', 'TD1', 'TD2', 'TD3', 'PD1', 'PD2', 'MD1', 'LD1'] ['1.00', '0.45', '0.01', '0.00', '0.00', '0.00', '0.00', '0.00']	TD1
	['RD4', 'TD1', 'TD2', 'TD3', 'PD1', 'PD2', 'MD1', 'LD1'] ['1.00', '0.44', '0.01', '0.00', '0.00', '0.00', '0.00', '0.00']	TD1
	['RD4', 'TD1', 'TD2', 'TD3', 'PD1', 'PD2', 'MD1', 'LD1'] ['1.00', '0.44', '0.01', '0.00', '0.00', '0.00', '0.00', '0.00']	TD1
	['RD4', 'TD1', 'TD2', 'TD3', 'PD1', 'PD2', 'MD1', 'LD1'] ['1.00', '0.44', '0.01', '0.00', '0.00', '0.00', '0.00', '0.00']	TD1
	['RD3', 'TD1', 'PD1', 'TD2', 'LD1', 'MD1', 'PD2', 'TD3'] ['0.98', '0.26', '0.14', '0.11', '0.04', '0.04', '0.01', '0.01']	TD1
	['RD3', 'MD1', 'PD1', 'TD1', 'TD2', 'PD2', 'TD3', 'TD4'] ['0.99', '0.92', '0.27', '0.17', '0.15', '0.02', '0.01', '0.01']	PD1 G4

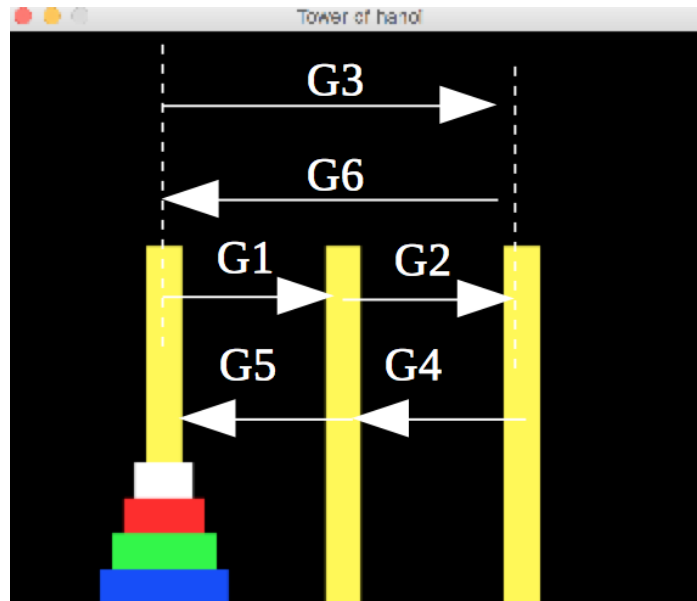


Fig. 8.7 Coding - Move a disk from the left peg to the middle peg ($G1$). Move a disk from the middle peg to the right peg ($G2$). Move a disk from the left peg to the right peg ($G3$). Move a disk from the right peg to the middle peg ($G4$). Move a disk from the middle peg to the left peg ($G5$). Move a disk from the right peg to the left peg ($G6$) extracted from [5]

Table 8.3 Rewards and possible actions at time 0 extracted from ([5]).

$G1$	$G2$	$G3$	$G4$	$G5$	$G6$
1	0	1	0	0	0

Furthermore, table 8.5 is a result acquired by a participant. This participant moved the same disk in a row (Between 6th and 7th line). In this case, the agent receives a negative rewards (-1) because this is not the optimal solution. Thus, in order to find the optimal solution, we modify selection algorithm by combining the calculated probabilities and rewards for possible action.

Based on the possible actions, we can predict possible action $G3$ in table 8.7. But in order to maximize its cumulative reward (See table 8.6), our LSTM+RL model takes an action $G2$ instead of $G3$.

Table 8.4 Probabilities and possible actions at time 0 extracted from ([5]).

$G1$	$G2$	$G3$	$G4$	$G5$	$G6$
0.35	0.003	0.60	0.00	0.001	0.001

8.4 Experiments

8.4.1 Experiment: Multi-label categorization

The classifier was optimized by splitting 1186 original images into 9% training images (1067 images) and 10% validation images(119 images). We then tested the accuracy of the model with 300 test data.

As shown in previous section, We have acquired a series of data from participants solving the Tower of Hanoi problem. These data contain semantic information, including actions and interactions with objects. In this experiment, given a series of data that are all labeled with a multiple categories, we have predicted these categories for a novel set of test data. For the evaluation metrics, precision/recall/F1 score have been used to evaluate the performance of a classification problems.

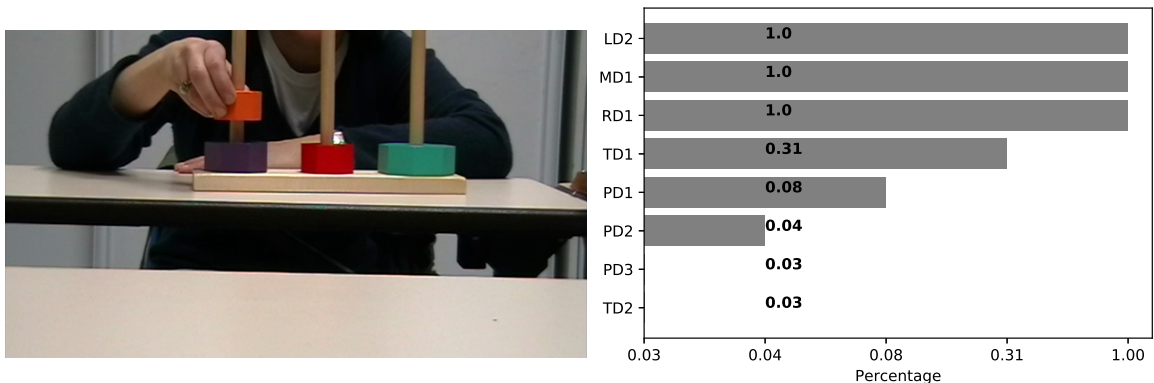


Fig. 8.8 The predicted result based on Convolution Neural Network(Participant1)

For precision/recall/F1 score, if the estimated label confidence for a label is greater than 0.5, the label is expected to be positive. However, for the estimated label confidence for the labels (TD1-4 and PD1-4), only positive label were assigned to the variable with the highest estimated label confidence of the eight labels. As Based on the trained CNN model, we predicted the multi-label categorization for the test image. Figure 8.11 shows an example of the testing result.

According to the table 8.8, the resulting F1 values for status labels(LD1-4,MD1-4,RD1-4) , which represented the model accuracy for multi-label prediction of CNN model, were

Table 8.5 Solution acquired by a participant extracted from [5].

States			Rewards
Peg A	Peg B	Peg C	
$d_1/d_2/d_3/d_4$			0
$d_2/d_3/d_4$		d_1	1
d_3/d_4	d_2	d_1	1
d_3/d_4	d_1/d_2		1
d_4	d_1/d_2	d_3	1
d_4	d_2	d_1/d_3	1
d_1/d_4	d_2	d_3	-1
d_1/d_4		d_2/d_3	1
d_4		$d_1/d_2/d_3$	1
	d_4	$d_1/d_2/d_3$	1
d_1	d_4	d_2/d_3	1
d_1	d_2/d_4	d_3	1
	$d_1/d_2/d_4$	d_3	1
d_3	$d_1/d_2/d_4$		1
d_3	d_2/d_4	d_1	1
d_2/d_3	d_4	d_1	1
$d_1/d_2/d_3$	d_4		1
$d_1/d_2/d_3$		d_4	10
d_2/d_3		d_1/d_4	10
d_3	d_2	d_1/d_4	10
d_3	d_1/d_2	d_4	10
	d_1/d_2	d_3/d_4	15
d_1	d_2	d_3/d_4	15
d_1		$d_2/d_3/d_4$	18
		$d_1/d_2/d_3/d_4$	20

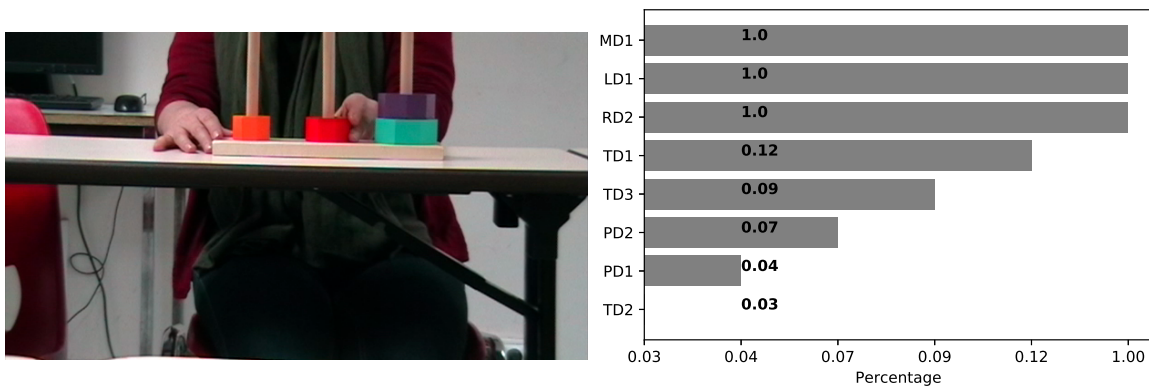


Fig. 8.9 The predicted result based on Convolution Neural Network(Participant2)

8.4 Experiments

Table 8.6 Negative Reward and possible action extracted from [5].

$G1$	$G2$	$G3$	$G4$	$G5$	$G6$
-1	1	-1	0	0	0

Table 8.7 Probabilities and possible actions extracted from [5].

$G1$	$G2$	$G3$	$G4$	$G5$	$G6$
0.0002	0.001	0.83	0.13	0.0004	0.002

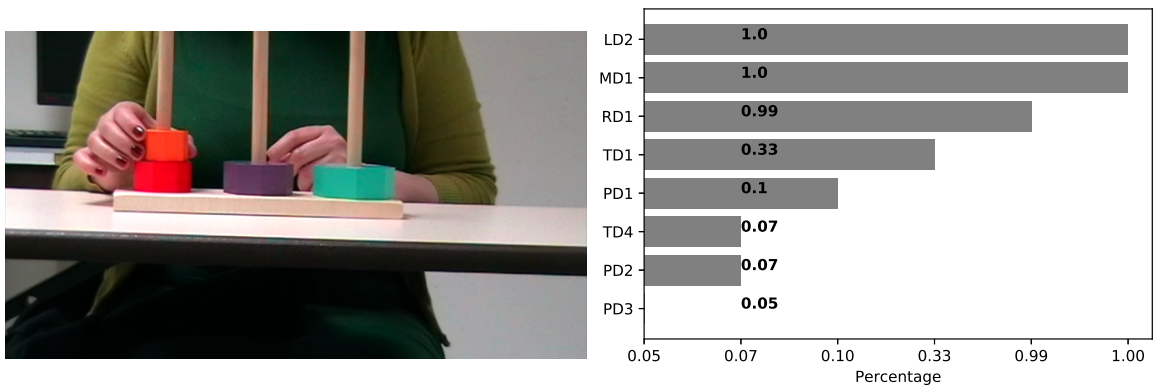


Fig. 8.10 The predicted result based on Convolution Neural Network(Participant6)

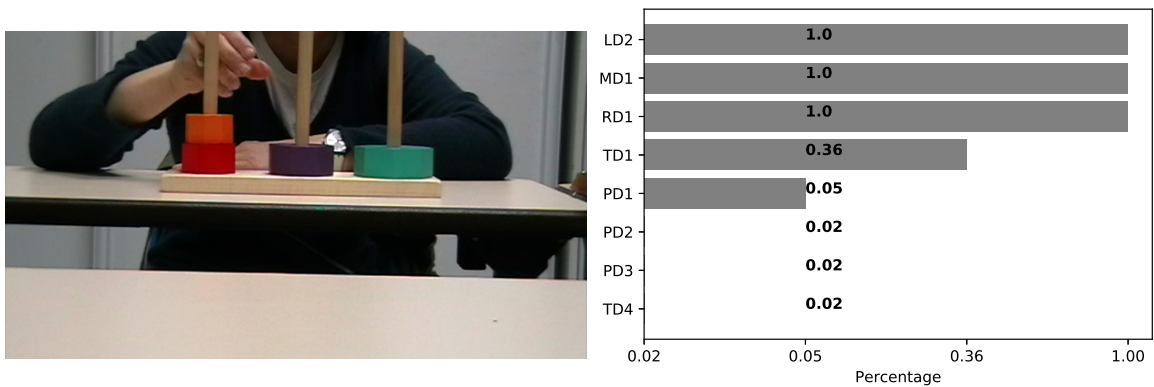


Fig. 8.11 The predicted result based on Convolution Neural Network(Participant2-second image)

high(>0.90) with the exception of LD3,LD4, and RD1 (0.8570,0.4979, and 0.8650, respectively). For the action's labels(TD1-4,PD1-4), a low degree of F1 value was found for TD4,PD3, and PD4 (0.4936, 0.4958, and 0.4979, respectively). We present visualizations of the learning curves (See Figures 8.13 and 8.14). Overfitting occurred in the early 10 to 50 iterations. After 50 iterations, accuracy increasing with the number of iterations and training accuracy above the validation accuracy.

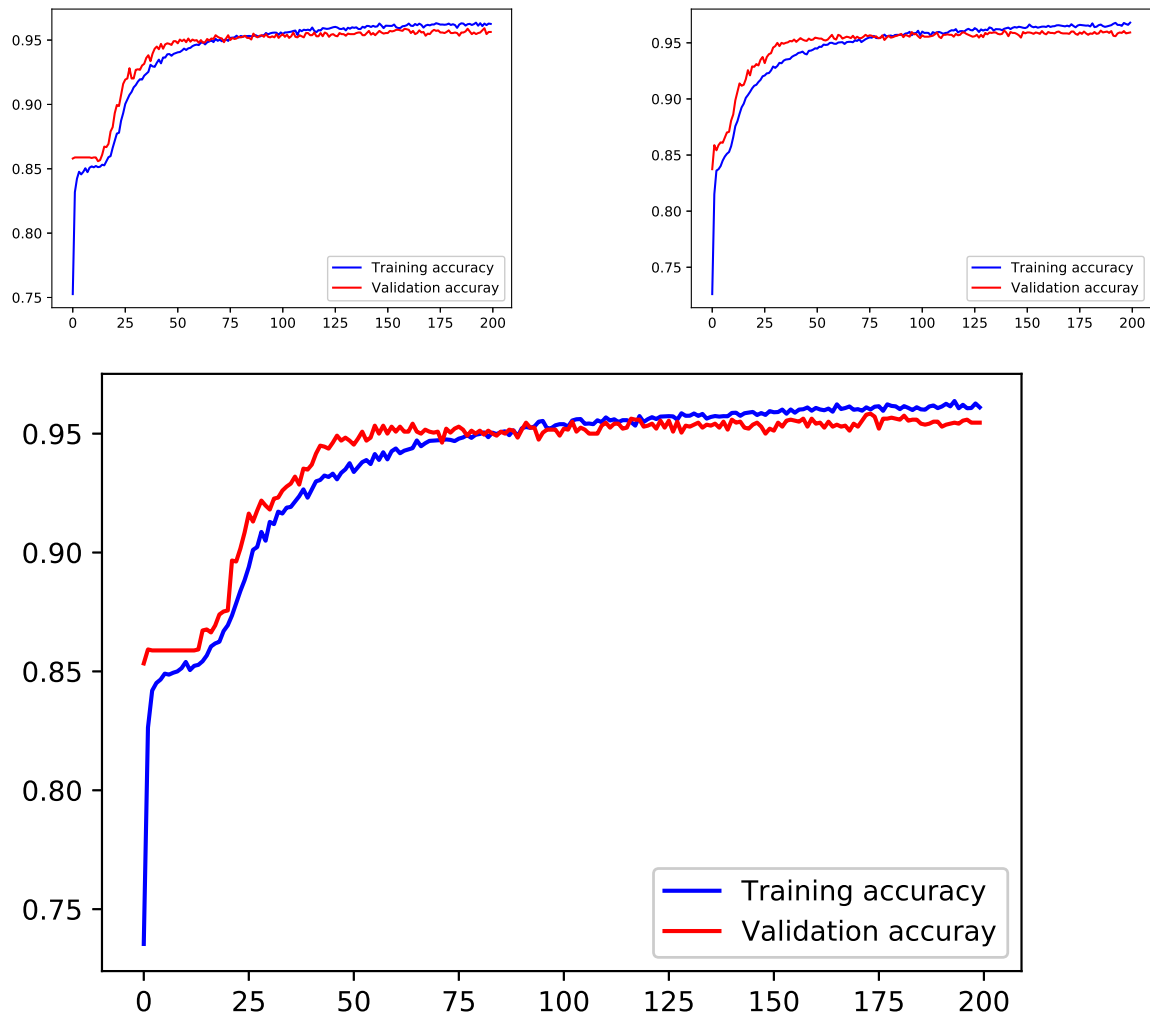


Fig. 8.12 Accuracy comparison for a convolutional neural network (CNN) with layers and epochs. The image on the left are 2 convolutional layers with 200 epochs. The middle is 3 convolutional layers with 200 epochs. The last image on the right is 4 convolutional layers with 200 epochs.

8.4.2 Experiment: LSTM+RL

The output vector obtained from CNNs contains semantic information. We feed this output vector into the RNN as an initial state for label prediction. And then, our model predict a possible action between sequential action. To do this, we conducted experiment by training our combined model(LSTM+RL) on the sequential data obtained in previous experiments on TOH solution. First of all, we evaluate and compare the performance of LSTM model on this sequential data. And then the combined model (LSTM+RL) is carried out.

Table 8.8 Results on performance metrics for all labels.

Label	Precision	Recall	F1
One disk on the left(LD1)	0.9389	0.9435	0.9406
Two disks on the left(LD2)	0.9610	0.9716	0.9662
Three disks on the left(LD3)	0.8912	0.8289	0.8570
Four disks on the left(LD4)	0.4958	0.5000	0.4979
One disk in the center(MD1)	0.9242	0.9225	0.9233
Two disks in the center(MD2)	0.9588	0.9706	0.9645
Three disks in the center(MD3)	1.0000	1.0000	1.0000
Four disks in the center(MD4)	1.0000	1.0000	1.0000
One disk on the right(RD1)	0.8684	0.8618	0.8650
Two disks on the right(RD2)	0.9552	0.9490	0.9520
Three disks on the right(RD3)	0.9333	0.9906	0.9595
Four disks on the right(RD4)	0.9955	0.9444	0.9683
Take d_1 (TD1)	0.5457	0.5535	0.5495
Take d_2 (TD2)	0.5276	0.5454	0.5364
Take d_3 (TD3)	0.5580	0.7201	0.5825
Take d_4 (TD4)	0.4874	0.5000	0.4936
Put d_1 (PD1)	0.5753	0.6312	0.5877
Put d_2 (PD2)	0.4906	0.5127	0.5014
Put d_3 (PD3)	0.4916	0.5000	0.4958
Put d_4 (PD4)	0.4958	0.5000	0.4979

The weights of all networks are initialized to random values uniformly distributed in the interval from $[-1/\sqrt{n}, 1/\sqrt{n}]$, where n is the number of incoming connections. To train our model we minimize the loss function for our training data.

To train the combined model we first initialized all of the LSTM's parameters with the uniform distribution between -0.1 and 0.1. We used stochastic gradient descent, with a fixed learning rate of .5. After 5 epochs, if loss increased in every epoch, we adjusted the learning rate. This LSTM model made predictions representing probabilities of the next action. To train the Q-function we initialized all Q-values of all state-action pairs to zero and initialized the states with their given rewards. Based on all possible actions obtained by LSTM, we measured a reward value for each possible action. If an action has the highest probability and reward, we can choose this as next sequential action. Otherwise we search another possible action with the highest reward value as next sequential action. Then, we updated the Q-value according to the equation (8.6) and repeated the process until a terminal state was reached.

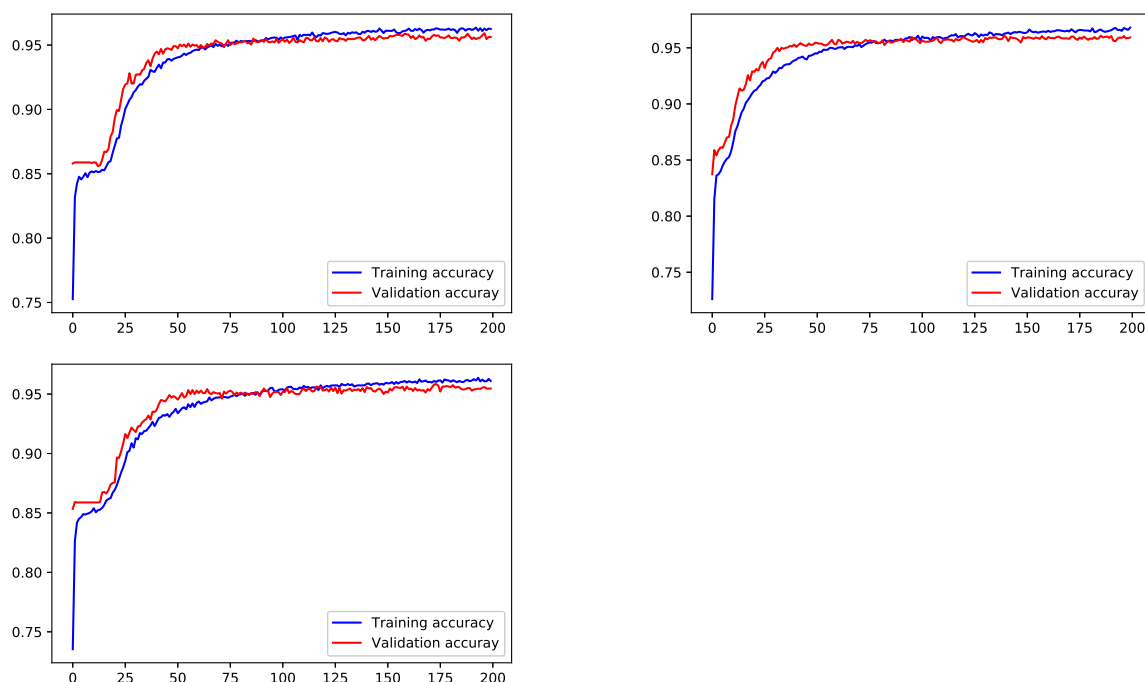


Fig. 8.13 Accuracy comparison for a convolutional neural network (CNN) with layers and epochs. The image on the left are 2 convolutional layers with 200 epochs. The middle is 3 convolutional layers with 200 epochs. The last image on the right is 4 convolutional layers with 200 epochs.

8.4.3 Results

Similar to the results in Chapter 7, the results of the simulation were the same. After training for the model RNN, we obtained the following shortest path : G1,G3,G2,G1,G4,G5,G4, G1,G3,G2,G5,G4,G3,G5,G1,G6,G5,G2,G1,G3,G2(21 movements). The training error is shown in figure 8.15.

Compared to the experimental results in table 7.4, this RNN model shows good performance improvement. Nevertheless, this result is not the fastest solution. According to the table 7.4, the first participant and the thirteenth participant find the fastest solution. On the other hand, from figure 8.16, we can see that this combined model(LSTM+RL) improves the performance compared with LSTM model.

8.5 Conclusions

The images obtained from real world have many and complex categories. The objects around us, surrounding scenes, actions, and interactions with objects all contain semantic

8.5 Conclusions

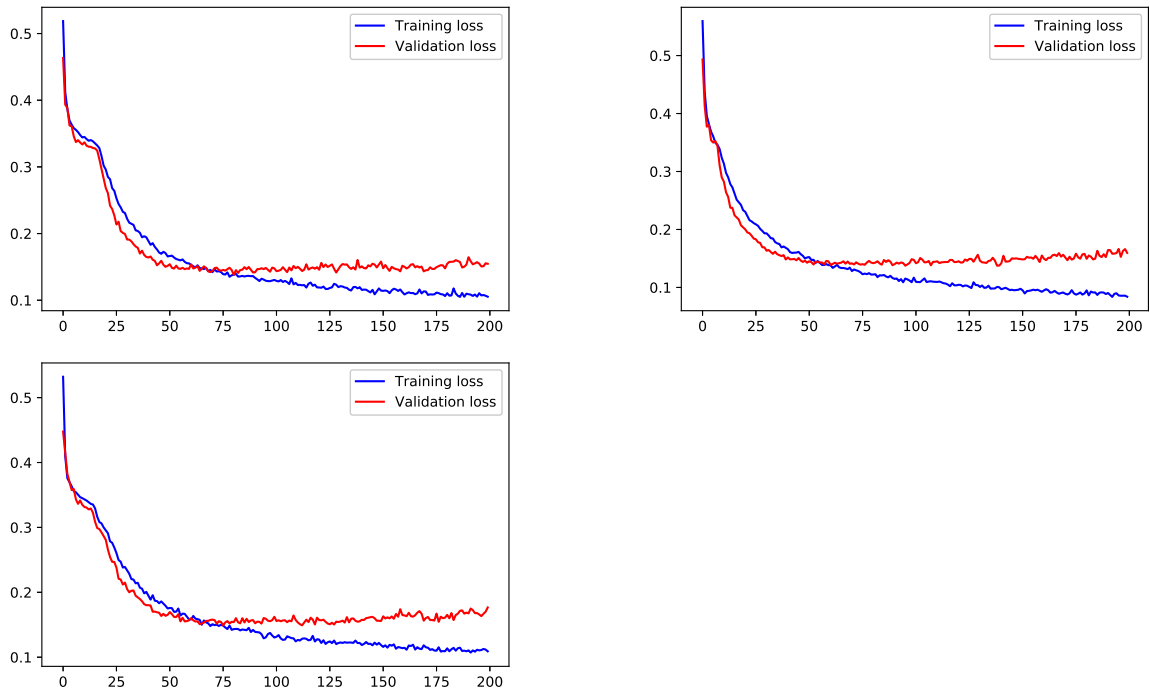


Fig. 8.14 Loss comparison for a convolutional neural network (CNN) with layers and epochs. The image on the left is 2 convolutional layers with 200 epochs. The middle image is 3 convolutional layers with 200 epochs. The last image on the right is 4 convolutional layers with 200 epochs.

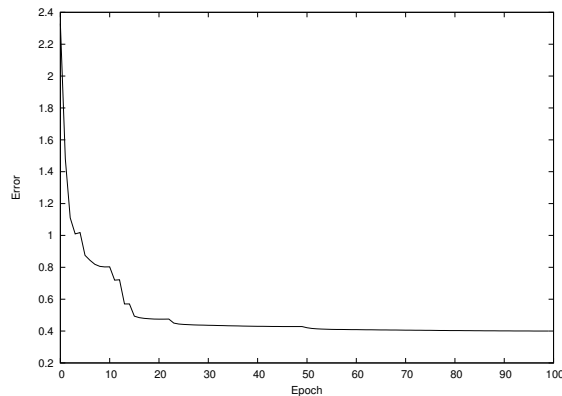


Fig. 8.15 LSTM train error at each epoch

information. In this paper, we focused on the semantic information we encountered in the problem-solving process. We firstly explored multi-label image classification in order to analyze these semantic information. In addition, we proposed a convolutional neural network(CNN) and LSTM-RL based method that is capable of efficiently extracting features

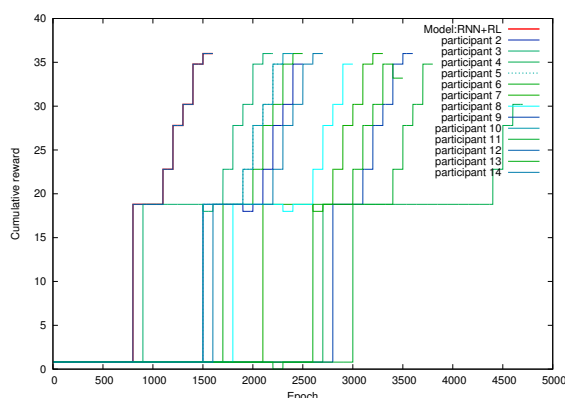


Fig. 8.16 Cumulative reward graph for LSTM+RL model

and finding solution of Tower of Hanoi. The proposed model has achieved the state-of-art performance. It can be seen that our proposed model can be a good choice to find a solution of Tower of Hanoi.

Further, there are two things that need to be studied in depth. First of all, as participants have a difficulty triggering simulations with visual object, deictic gesture for the sighted participants plays a central role in generating visual inferences. Meanwhile, the blind participants have a difficulty solving TOH because of the lack of tactile object. The interaction gestures for these participants play an important role in building their mental representation. Based on these observations, what we are wondering is how artificial intelligence constructs mental expressions like human beings and how to understand the meaning of such hand motions. The second thing to study is extremely simple. In the end, in order to find the solution of problem as like human beings, we must understand the rule of game accurately by understanding natural language. One disk at a time should be moved, in a location, a large disk cannot be placed on top of a smaller one. When someone plays the game directly by explaining the rules of the game as above, we can understand the rule of game while mapping languages and images. Likewise, we will do research to understand game rules through natural language processing and image analysis.

Chapter 9

Understanding Natural Language with Convolutional Neural Networks for Problem solving

9.1 Introduction

When we have conversations with children of the age of 2 or so, we can see that they try to communicate with only two or three words without considering the existing grammar framework. Children recognize objects through sensory movements and try to communicate based on them. It doesn't matter any language and no grammatical disruption matters. Think about it, We can see them playing with multinational kids in Crash too. Children can fully communicate with each other even though they are not yet proficient in French. It can be seen that the reason is related to dialogue based on actions and emotions, in other words, based on sensory movements. Early childhood children directly connect verbal forms to concepts and categories that have already been established in the process of non-verbal cognitive development.

It can be assumed that CNN's multi-classification method is concepts and categories learned in the process of embodies. Therefore, we studied whether language learning is possible through mapping of longer sentences based on the multi-category classification in the preceding chapter and the Hanoi Tower problem inference through RNN. In this chapter, AI will try to talk directly about the next action after observing the current state and the participants' actions. For example, after confirming that the disks are stacked on the left, we will verbally command the participant to move the topmost disk to the center or to the right. In this way, the participant seeks to solve the Tower of Hanoi problem only by following the

verbal instructions of artificial intelligence. In addition, when a supervisor, that is, a human gives verbal instructions to artificial intelligence, we want to implement a simulation that can understand the verbal instructions and solve puzzles. The purpose of this simulation is to make it possible to understand language and make inferences by mapping images, or, more precisely, images of states and actions surrounding us with language.

9.2 Background and related work

9.2.1 Cognitive Development Theory and Embodied cognition

In our study, Piaget's theory of cognitive development serves as a starting point. According to the Piaget's theory of cognitive development, infancy depends on behavior and perception, and in childhood, children understand observable aspects of reality. As a child becomes a young man, he becomes increasingly aware of abstract rules and principles. These stages of cognitive development involve many aspects of mental development, including reasoning, language, morals and memory. Piaget believed that children can build knowledge when they actively participate in cognitive development and interact with the world (Piaget, Jean [112], Piaget and Cook [111]).

Therefore, the research in this chapter is based on Piaget's theory of cognitive development. In particular, we carefully looked at the sensorimotor stage, which is an early cognitive development stage. This is because infants acquire a basic understanding of the world around them through sensory and motor skills. Infants' innate abilities (sight, hearing, smell, taste and touch) are closely coupled with constantly developing bodily functions. Infants acquire cognitive abilities about themselves and their surroundings through constant touch, grasping, and tasting. They experience the world and gain knowledge through their senses and movements. In addition, through repeated trial and error, children discover more about the world around them.

The main debates in cognitive development are nature and nurture, i.e. birth theory and empirical theory. However, many agree that this is the wrong dichotomy. Much evidence from life sciences and behavioral sciences shows that from the very beginning of development, the activity of genes interacts with events and experiences in the environment. Another issue is whether cultural and social experiences relate to the evolutionary change of thinking. Another question concerns the systematic relationship with non-human animals. Most learning and cognition are similar in humans and animals.

The Kyoto University research team conducted a memory test to show that chimpanzees have better photo memory than humans (Inoue and Matsuzawa [64]). First, the numbers

1 through 9 appear and disappear quickly on the computer screen. Then the participants (chimpanzee and students) remembered the numbers and pressed the screen where the numbers were placed in order. Chimpanzee and college students were tested at different speeds respectively.

Surprisingly, chimpanzee Ayumu ordered numbers faster and more accurately than college students. Through experiments, common sense overturned the notion that human cognitive abilities are better than those of animals. In the case of humans, the research team says that these photographic memories exist when they are young and gradually deteriorate as they grow. This can be fully understood under the assumption that our cognitive abilities are not only created by our brain, but also our body and environment.

In the embodied cognitive theory, it is said that cognitive ability primarily develops based on body movements, and later develops into abstract concepts and language in the process of evolution. Therefore, it can be seen that chimpanzees are superior to humans in low-level intelligence tests, such as the photo memory test, which requires the connection of hand gestures and visual stimuli.

Let us look at another case. Let's take a look at the research of Professor Goldin-Meadow's team at the University of Chicago Department of Psychology. According to their research, a study found that children who solve problems by moving their hands freely showed much better grades (Goldin-Meadow et al. [50]). According to the published thesis, students who move their hands freely when solving a problem had a 1.5 times higher rate of correct answers than those who did not. It is proving that speaking and thinking while actively using the body is more helpful in improving cognitive abilities.

In addition, a new approach has been proposed in the field of cognitive linguistics, which has recently developed a theory related to conceptual fusion centering on the metaphorical meaning of language. Considering that the basis of human language lies in the sensorimotor activity of the body, the existing Chomsky linguistics-centered approach should be modified and the weight of cognitive linguistics should be increased (Knott [71], Chomsky et al. [27]).

9.2.2 Convolutional Neural Network in Natural Language Processing

In the past decades, many research methods have been proposed for natural language processing. In this chapter, we will deal with the problem of natural language processing using convolutional neural networks. First, let's look at word embedding, a traditional method well known in natural language processing. Let us take an example. Suppose you have three words: France, Paris and the Eiffel Tower in order. In that order, numbers 1, 2, and 3 are assigned respectively. France can be expressed in the form of a vector of (1,0,0) Paris (0,1,0)

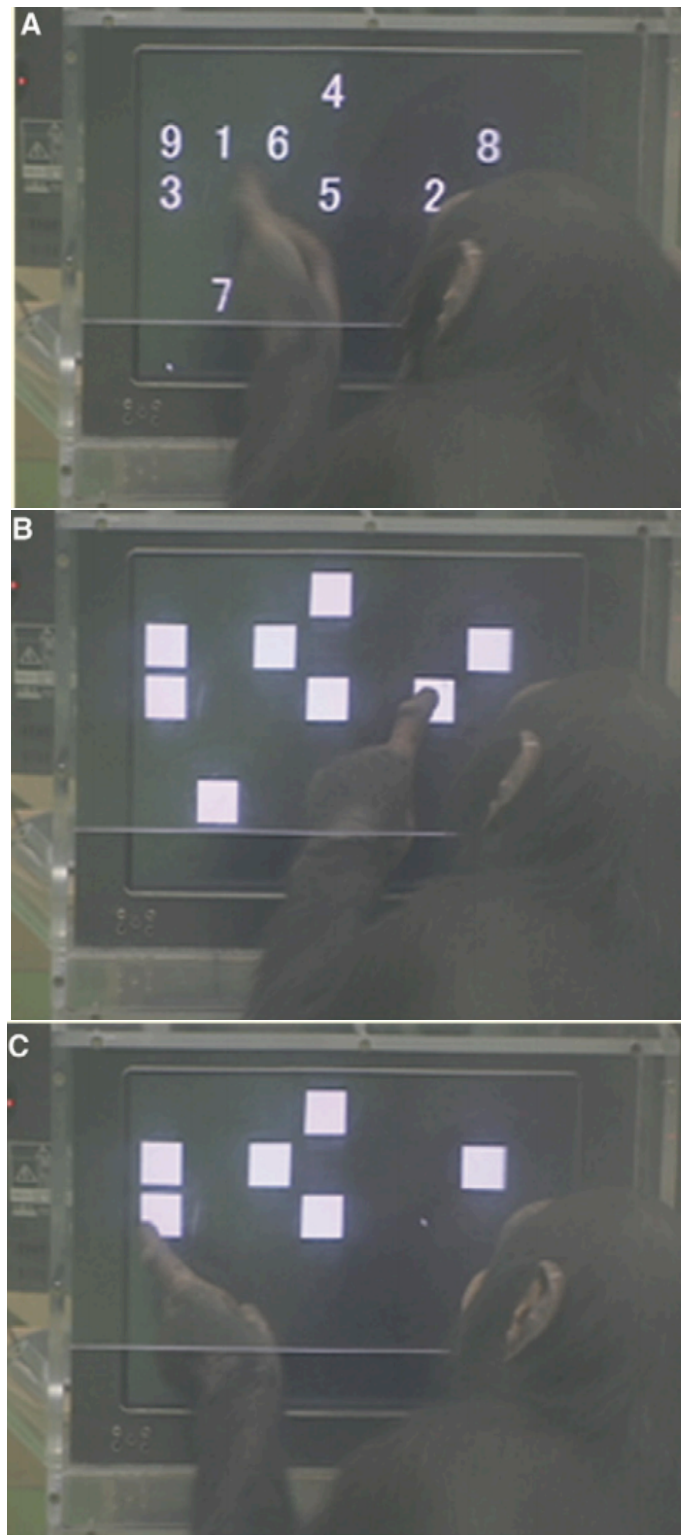


Fig. 9.1 Chimpanzee Ayumu performing the masking task from Inoue and Matsuzawa [64]

and Eiffel Tower (0,0,1). This is called one-hot encoding. Each row of the word document matrix, i.e., the vector representing the document, is equal to the sum of all the word vectors constituting the document. However, this is an old way of not considering the meaning of the word at all. Therefore, in order to be able to understand the meaning of words better, an attempt was made to express words of more dimensions as vectors.

These word embeddings have led to cutting-edge results in natural language processing. For example, word embedding is used in the stacked denoising autoencoder model for emotional classification according to domain characteristics (Glorot et al. [49]), or combinatorial categorical autoencoders method is used to learn sentence synthesis using word embedding (Hermann and Blunsom [58]). This embedding model has shown great results in processing natural language. Of course, there was a disadvantage that the data space became too large because all words had to be expressed in the vector space. As a result, there is a problem that the performance of the analysis technique is weakened. Therefore, natural language processing researchers further expand the concept of word2vec, where learning is performed by converting words into vectors and using a neural network structure to input one hot vector of the word and predict the one-hot vector value of the surrounding words (Mikolov et al. [84]).

In this way, word embedding showed great results in various natural language processing such as emotion analysis, summarization, and QA. This embedding model was a simple neural network with few hidden layers. However, in parallel with the development of deep learning, a new consideration of the embedding method was needed. Therefore, to output multi-category prediction results, multi-task learning using a convolution neural network was used. (Collobert and Weston [33]) In order to improve the performance of word embedding, there is a need for an efficient function that extracts high-level features from n-grams. In addition, the convolutional neural network had the advantage of being able to extract key n-gram features from input sentences. Since these abstracted features could be very useful in problems such as summarization, sentiment analysis, machine translation, and QA, natural language processing through the use of a convolutional neural network (CNN) becomes more and more utilized. In outputting multi-category prediction results, convolutional modeling was used to solve natural language processing problems such as part of speech tagging, entity name recognition, semantic inversion, and language modeling. Therefore, in this chapter, we want to simulate language recognition based on cnn. In particular, learning of concepts and categories was studied in the previous chapter. We also want to compare and analyze how previous learning is helpful for long sentence learning.

9.3 Model

The CNN model architecture for natural language learning was based on the architecture of Colobert and Kim (Mikolov et al. [84], Kim [69]). In order to find the sentence that matches the image, we will construct a model like the figure 9.2. As we mentioned in Chapter 3 (Neural Networks and Deep Learning) and Chapter 8 (Using Convolutional Neural Networks and Recurrent Neural Network for Human Gesture Recognition and Problem Solving), CNN is an architecture designed for image processing.

As shown in the figure 3.10, CNN is learned in the form of extracting and preserving regional information of an image. If the CNN filter for image processing plays a role of extracting regional information of the image, the text CNN filter preserves the local information of the text, that is, context information.

Word	V_1	V_2	V_3	V_4	...	V_{p-2}	V_{p-1}	V_p
W_1
W_2
W_3
...
W_{n-2}
W_{n-1}
W_n

The total number of words per sentence is n . Each of these words is composed of p -dimensional vectors. The size of the filter is variously adjusted to 1,2,3, and local information of the sentence is sequentially preserved in the order of the words appearing in the sentence. If there is one filter, it is called Uni-gram, if there are two, it is called Bi-gram, and if there is three, it is called Tri-gram. By adjusting the size of the filter in this way, various n -gram models have been created. A sentence corresponding to each image was embedded. In order to embed words into vectors, distributed representations such as Word2Vec and GloVe are used, but our model randomly gave the initial value of the word vector and then used it by gradually updating it in the learning process like other parameters. As in the figure 9.2, we match the feature map as many as the number of filters, and then go through the max-pooling process and output the score as the number of classes.

In Chapter 8, we multilabeled the acquired images. In this chapter, we tried to combine the CNN that learned multi-labeling and the CNN that learned sentences (Figure 9.3. Through this, we are trying to prove our hypothesis that sentence learning could be done more effectively if sentence learning was performed from cnn, which has already learned the

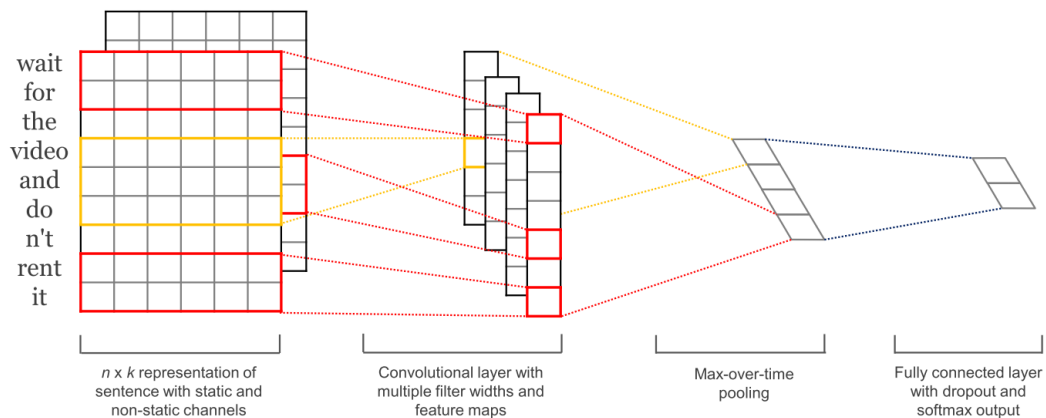


Fig. 9.2 Convolutional Neural Networks for Sentence Classification

categories of images. Finally, we compared and analyzed what kind of result would be obtained when the sentence was immediately learned without multi-labeling from the image.

As in the figure 9.3, we split the CNN a little more to make sentence analysis more effective. We designed the results of the divided CNNs to be summed and released. In the data pre-processing process, we classified the disk into 5 completely different classes according to the shape of the disk and the behavior of the participants.

For example, if there is no disk on the left, it has a value of 0 and is classified as 1 for one, 2 for two, 3 for three, and 4 for four. Similarly, the center and the right were also classified into 5 classes. Also, the action of the participant was classified into 5 classes according to which disk was held and which disk was placed. Given a sentence in this way, five convolutional neural network learning simulations were executed (Figure 9.4)

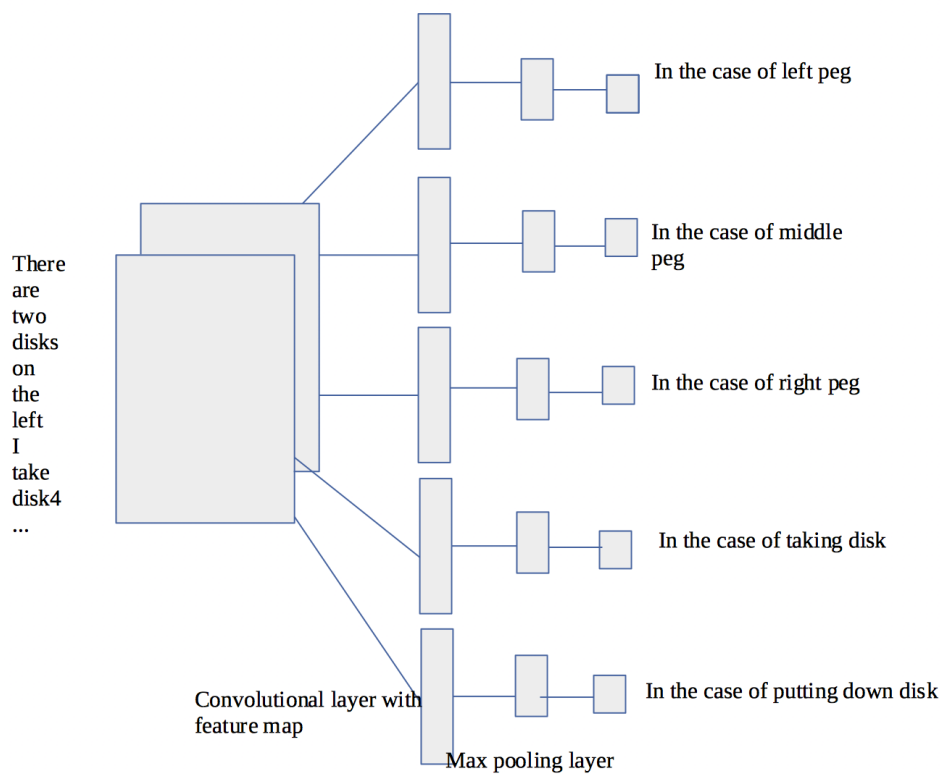


Fig. 9.3 A Combined Convolutional Neural Network for Natural Language Processing

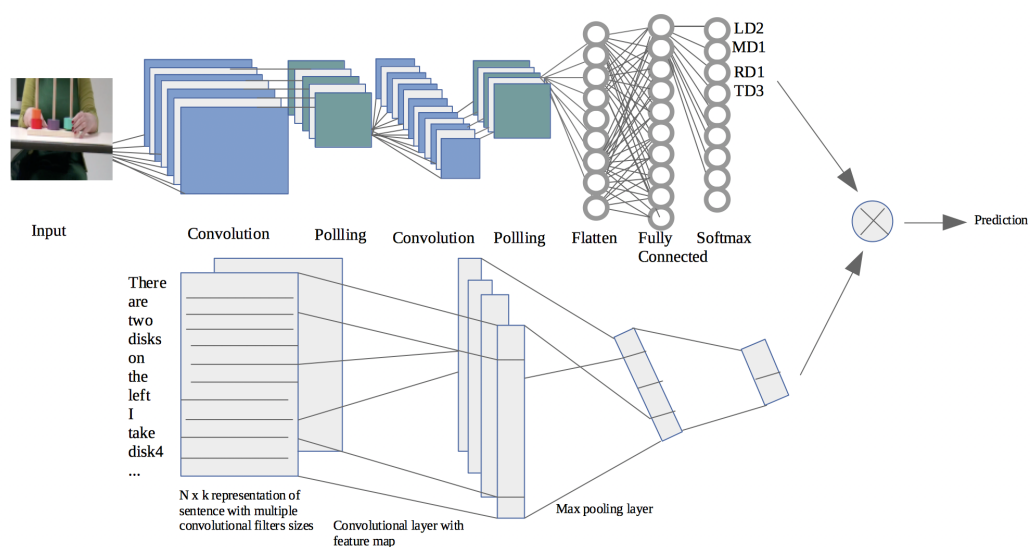


Fig. 9.4 A Combined CNN-CNN Model architecture for Natural Language Processing

9.4 Experiments

In the data preprocessing process, it was classified into 5 classes. The preprocessing data for sentence learning were classified as shown in the table 9.1.

Table 9.1 Example of data preprocessing

Sentence	Ld	Md	Rd	Td	Pd
There are four disks on the right. I take disk1.	0.0	0.0	4.0	1.0	0.0
There is one disk in the middle. There are three disks on the right. I put down disk1.	0.0	1.0	3.0	0.0	1.0
There is one disk in the middle. There are three disks on the right. I take disk2.	0.0	1.0	3.0	2.0	0.0
There is one disk in the middle. There are two disks on the right.	0.0	1.0	2.0	0.0	0.0
There is one disk on the left. There is one disk in the middle.	1.0	1.0	2.0	0.0	2.0
There are two disks on the right. I put down disk2.					
There is one disk on the left. There is one disk in the middle.	1.0	1.0	2.0	1.0	0.0
There are two disks on the right. I take disk1.					

Ld: Disk on the left (LdCategory), Md: Disk in the center (Md-Category)

Rd: Disk on the right (RdCategory), Td: Take disk (TdCategory)

Pd: Put disk (PdCategory)

As mentioned in the previous section, we did not use distributed representation methods such as Word2Vec in this chapter. However, we chose a method of randomly initializing the word vector and updating it in the learning process. In the natural language processing procedure using computers, it is virtually impossible to process natural language by accepting sentences as they are. Therefore, when applying a sentence to the natural language processing mode, we divide the sentence into units called tokens. Also, when passing these token values to a computer, the token values are converted into numbers and transmitted to facilitate calculation. This is in the same context as in the previous chapter when we were working on an image and inputting the input value as a pixel value.

After these data preprocessing and tokenizers, we implemented a CNN model for disk location and actor actions. The architecture of this model is presented in table 9.2.

Given a sentence, our model showed 94.12 % accuracy with the disk on the left. It also showed a high probability of accuracy in learning language related to the participants' behavior. The probability of judging to grasp the disc was 68.91 % and the accuracy of the disc releasing was 71.43 %. However, when the disk is located in the center and on the right, it did not show such high probability of accuracy (Table 9.3). In addition, the accuracy of

Table 9.2 The summary of Our Convolutional Neural Networks model for Natural Language Processing

Layer (type)	Output Shape	Param
input 3 (InputLayer)	(None, 52)	0
embedding 3 (Embedding)	(None, 52, 300)	6000000
reshape 3 (Reshape)	(None, 52, 300, 1)	0
conv2d 7 (Conv2D)	(None, 50, 1, 100)	90100
conv2d 8 (Conv2D)	(None, 49, 1, 100)	120100
conv2d 9 (Conv2D)	(None, 48, 1, 100)	150100
max pooling2d 7 (MaxPooling2D)	(None, 1, 1, 100)	0
max pooling2d 8 (MaxPooling2D)	(None, 1, 1, 100)	0
max pooling2d 9 (MaxPooling2D)	(None, 1, 1, 100)	0
concatenate 3 (Concatenate)	(None, 3, 1, 100)	0
flatten 3 (Flatten)	(None, 300)	0
dropout 3 (Dropout)	(None, 300)	0
dense 3 (Dense)	(None, 5)	1505
Total params: 6,361,805		
Trainable params: 6,361,805		

the model and the graph of the model loss according to the epoch can be seen in the figures 9.5, 9.6, 9.7, 9.8 and 9.9.

Table 9.3 CNN results for the location of the disk and the participants' actions

Category	Precision
Disk on the left (LdCategory)	94.12 %
Disk in the center (MdCategory)	42.02 %
Disk on the right (RdCategory)	38.66 %
Take disk (TdCategory)	68.91 %
Put disk (PdCategory)	71.43 %

Finally, we tried to map the results of the five text CNNs thus obtained and the data obtained from the images of the previous chapter. This methodology calculated the correlation between event 1 and event 2 as a probability measure, as in the Neural Association Model presented in Section 6.2.2. E1 is the predicted value based on the sentence analysis model, and E2 is the predicted value based on the image analysis model. Therefore, we tried to find a probabilistic correlation between the two $Pr(E2|E1)$. The data predicting multi-category from the test data images were compared with the values of five text CNNs. When the values match, a predictable sentence was derived from the test image. Our learning model showed 91.6 % accuracy in predicting the learned sentences.

9.4 Experiments

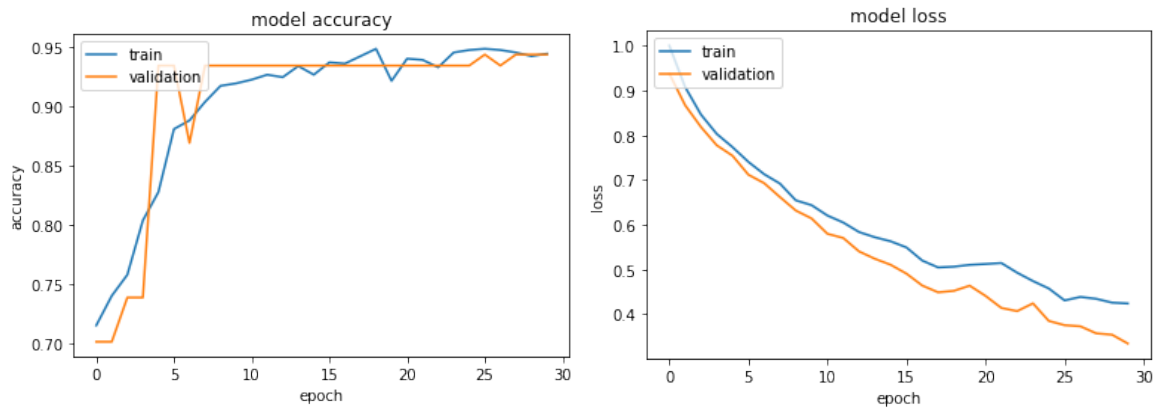


Fig. 9.5 Accuracy and loss comparison for a natural language processing - In the case of left Peg

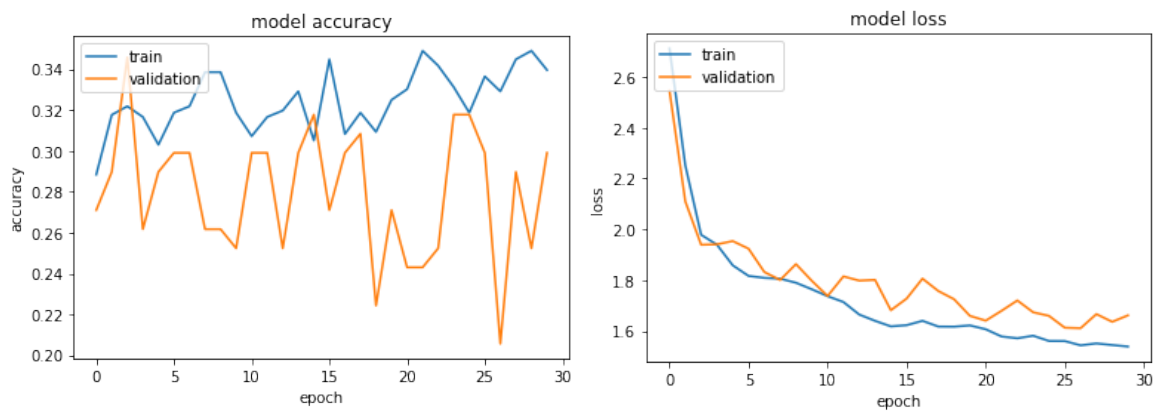


Fig. 9.6 Accuracy and loss comparison for a natural language processing - In the case of middle Peg

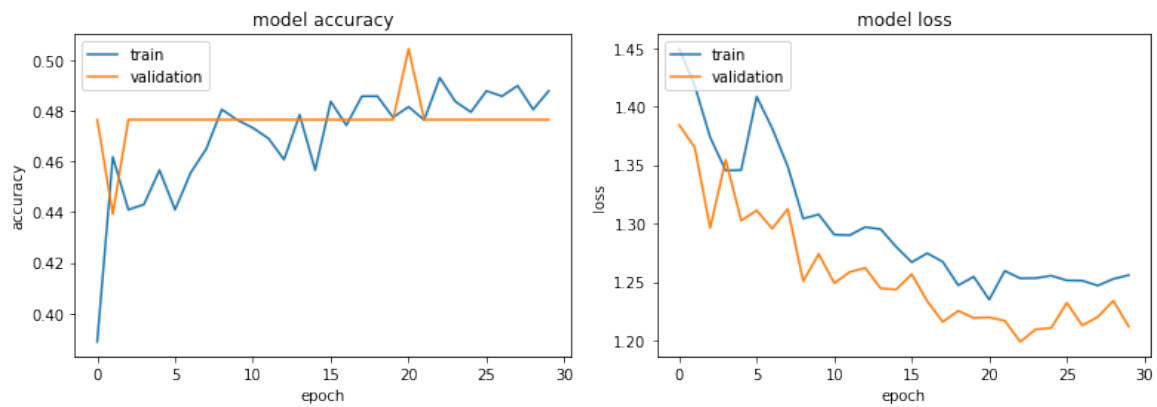


Fig. 9.7 Accuracy and loss comparison for a natural language processing - In the case of right Peg

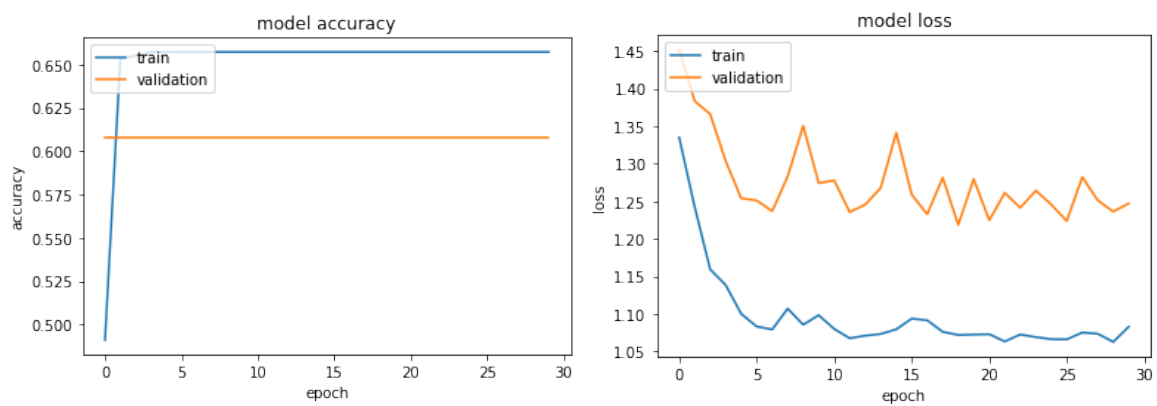


Fig. 9.8 Accuracy and loss comparison for a natural language processing - In the case of taking disk

9.5 Conclusion and the future

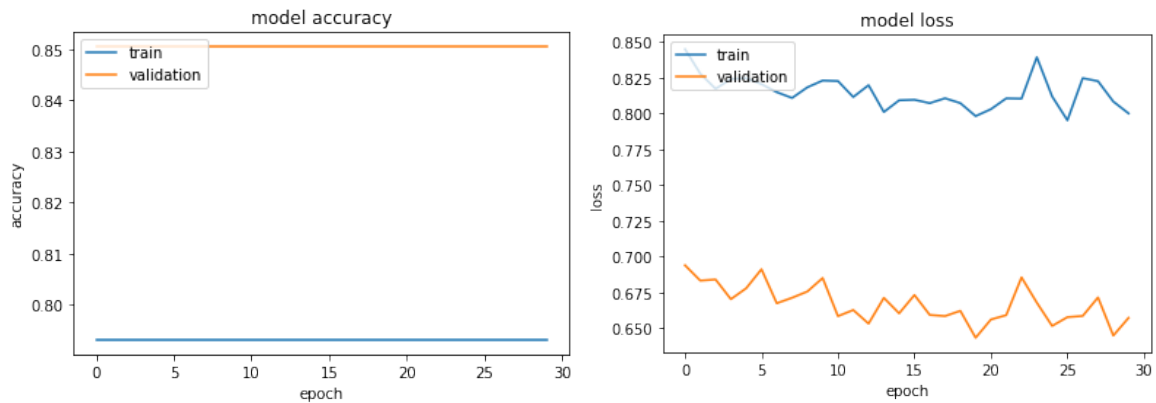


Fig. 9.9 Accuracy and loss comparison for a natural language processing - In the case of putting down disk

9.5 Conclusion and the future

As the participants solved the Tower of Hanoi puzzle, we paid attention to their actions and the locations of the disks. In this way, we trained the image based on the objects around the participant and the information acquired from the surrounding scenes. We also tried to learn the corresponding long sentences from this learned information. This is not a simple multi-category learning, but a natural language understanding from the learning of images. Therefore, an element of sentence learning using CNN was added to the learned image.

Before mapping the CNN results obtained by learning multi-category from images, we investigated what result values are derived only with sentence learning itself. Given a sentence, the CNN model for sentence learning was able to derive good results when inferring the location of the disk and the participant's behavior. After that, we combined the CNN with multi-category image classification and the CNN with sentence learning. This tested whether the corresponding long sentence could be inferred when the test image was presented, and we were able to see interesting possibilities. In the future, we want to research various models for language learning as a cognitive process. In particular, in solving the problem of the Tower of Hanoi, we intend to utilize more sophisticated modeling of language, image, and behavior of participants.

Chapter 10

Discussion and Conclusion

We introduced embodied cognitive theory as an alternative approach to human cognition. Cognition relies on experiences that manifested through the body with various sensorimotor abilities. Sensorimotor processes and perceptual activities cannot be fundamentally separated from living cognition (Varela et al. [139]). Therefore, in order to understand the complex and rich interactions of the body, brain, and world, new concepts, tools, and methods suitable for studying creative, decentralized and self-organizing phenomena are needed (Clark [30, 31]). Therefore, it was considered that embodiedism can be constructed in the following methodological dimension.

- Cognition is an activity in the world based on the structure and ability of the body.
- Cognition produces meaning in the process of contacting the world through the body.
- Perception cannot be explained independently of the structure and function of the body.
- In order to properly understand and explain cognition, the structure and function of the body must be considered.

In this way, in order to better understand cognition, we saw that it was necessary to pay attention to the connection between the sensory movements surrounding our body and the world. Therefore, we tried to simulate how language learning is possible based on embodied cognition in this paper.

Deep learning, which imitates the human brain, has continued to develop over the past decade. In particular, artificial intelligence algorithms are immediately solving difficult problems that humans cannot solve. Artificial intelligence is proving its usefulness in data mining, voice recognition, face recognition, banking software, and search engines. We focused on deep learning to implement human cognitive ability based on an embodied

cognitive approach. To this end, we collected experience data of participants solving the Hanoi Tower puzzle. Based on this data, various simulations were performed to make problem inference.

First, in Chapter 5, we collected the experience data of participants solving the Hanoi Tower puzzle. In the case of sighted participants, the gestures played a very important role in generating visual reasoning in the process of problem solving. On the other hand, in the case of blind participants, a touch plays a very important role in the process of problem solving. Therefore, we sought to understand the embodied perspective by exploring the role of gestures in problem solving. We performed various simulations to infer the problem based on the data obtained from the participants. However, in order to examine how visual and tactile senses affect problem solving in the future, we would like to perform various experiments such as changing the color of the disc or preparing a disc with a different tactile sense. In particular, in order to measure the effect of tactile sensation on cognitive abilities, we think it is necessary to connect with robots that can realize the tactile sensation of the hand.

In Chapter 6, we tried to solve the Tower of Hanoi problem through neural network models. The supervisor training model, Multi-layer Perceptron Model, showed excellent accuracy in predicting the following behavior. However, the supervisor model has a problem of continuing coding of data as the number of disks increases. As we know well, human reasoning and thinking do not require very much learning data. Intuitive reasoning plays a very important role in human cognitive abilities. Therefore, in order to overcome the limitations of the supervisor model, we created two inference models. Human cognitive ability is able to infer the rules of the game simply by watching the game without any explanation of rules. Therefore, the relational network model for understanding the rules of the game led to very meaningful results. This is a methodology that considers the relationship between objects, and it could lead to remarkable results in terms of embodied cognition. We are going to challenge more diverse research tasks from this chapter 6. For example, we did not consider the size and color of objects in this study. When our model tries to infer the rules of the game in consideration of the relationship with the object, we would like to simulate in the future what results will be obtained when the conditions for the properties of these objects are presented together.

In problem solving such as the Tower of Hanoi, the simulations in our study were based on analysis of participants' movements. In the embodied method of perception, we tried to model the problem solving of Tower of Hanoi by observing the gestures of the experimental participants in Chapter 7. In this chapter, we simulated the Hanoi Tower inference model by applying the Recurrent Neural Network(RNN) model, which is mainly used for natural

language processing. We obtained sequential data on participants' movements and gestures when participants solved the Tower of Hanoi problem. And then, we trained a Recurrent Neural Network model based on this data. However, the RNN model had a problem that the shortest path could never be found unless a large number of subjects' data suggests the shortest path to problem solving. Therefore, by using the Reinforcement Learning method to compensate for this, an optimal solution model was developed and good results were obtained. In the case of the Recurrent Neural Network model, coding was performed according to the location and movement of the disk. However, in Chapter 7, no coding was made for the participants' directive gestures, the act of touching the disk, and the various movements of their hands when they think. Participants' directing actions or touching objects play a very important role in motivating logical reasoning and inference of the next action. Therefore, in the future, more complex coding will have to be performed based on this model.

In addition, the topic of language learning, which was most interested in our thesis, was simulated using the Convolutional Neural Network (CNN) in Chapters 8 and 9. The Objects around us, surrounding scenes, actions, and interactions with objects contain a variety of semantic information, just like in everyday life. Images obtained in the real world have many and complex categories. Likewise, the sequential images we collect contain a variety of semantic information: objects around the participant, their actions, and interactions with the objects. Therefore, the multi-label image classification task in Chapter 8 helped to understand more complex semantic information. We extracted a series of images from the video data of participants who participated in the Hanoi Tower problem solving, classified each image, and then trained it through CNN. In this simulation, the CNN characterizes the image. If we change the color of the disk, we need to measure what we want to get. This will have to be reviewed in the future if the properties of the object change under the conditions of the same game, e.g. color or shape, what kind of result it will lead to. We also simulated to find the optimal solution of Tower of Hanoi by combining CNN and RNN models.

In Chapter 9, we simulated how to understand the language given a complete sentence based on the multi-label image learned in Chapter 8. After that, we mapped the results of multi-label CNN and text analysis CNN to analyze the effect of sentence learning from images. According to our results, given an image, the probability of predicting the correct sentence was very high. This proved that in the case of a model in which concepts and categories were learned, long sentences could be better understood. In the case of this natural language analysis, there were not many types of sentences used because it was limited to solving the Hanoi Tower problem. In the case of presenting sentences in various languages such as English, French, Korean, etc., we try to find out what understanding the sentences

will result in. In the future, I would like to proceed with research on interesting topics that things do.

I set up a thought experiment. Imagine a child and a parent standing in front of a traffic light. Currently, the traffic light is red. Parents, who are dragging a stroller, stop and stand on their way. The child learns the present situation through the sense organs. Parents do not forget to explain that they must stop when the traffic lights turn red. A child who has experienced countless times of repetitive situations will learn one rule that we must stop in front of a red light. It begins with the learning of the image received through the sense organs, and has insight that the situation can be recognized more clearly than when the rules in language are delivered. However, when crossing the street despite the red light, there are also problems of various choices, such as whether to follow a human-made legal decision or a voluntary judgment of artificial intelligence.

Based on these sensory motor organs, multiple sensors are required to acquire information through interaction with the environment. This eventually becomes a subject directly connected to the development of cognitive robots that acquire information through sensory organs. Therefore, in the future, I would like to participate in a multi-sensor-based cognitive robot research that has an architecture of a perceptual behavior cycle model based on embodied cognitive theory. Cognitive behavioral intelligence technology, which is the basis for the development of such cognitive robots, can process multi-sensor data acquired in the real world without imitating intelligence only by formal symbolic manipulation as in conventional artificial intelligence. Multi-sensor-based cognitive robots require dynamic and flexible new intelligent technology that must act immediately based on information input in real time. In the conceptual framework of embodied cognition, as the mind, brain, body, and environment are conceptualized as a coherent gestalt that does not separate from each other, the exploration and application of human cognition through artificial intelligence will also be considered as embodied approaches. As such, cognitive information processing technology based on embodied cognition will be a new and big challenge for me.

References

- [1] Adorf, H. M. (1989). Connectionism and neural networks. *Knowledge-Based Systems in Astronomy*. Editor: A. Heck, 329(Chapter 12):215–245.
- [2] Anderson, M. L. (2003). Embodied Cognition: A field guide. *Artificial Intelligence*, 149(1):91–130.
- [3] Arbib, M. A. (2003). *The Handbook of Brain Theory and Neural Networks*. MIT Press.
- [4] Ashby, F. G. and Maddox, W. T. (2005). Human category learning. *Annual Review of Psychology*, 56:149–178.
- [5] Bang, S. and Tijus, C. (2018). Problem Solving using Recurrent Neural Network based on the Effects of Gestures. In *Proceedings of the 10th International Joint Conference on Computational Intelligence (IICCI)*, pages 211–216.
- [6] Barsalou, L. W. (1999a). Perceptions of perceptual symbols. *Behavioral and Brain Sciences*, 22(4):637–660.
- [7] Barsalou, L. W. (1999b). Perceptual symbol systems. *The Behavioral and brain sciences*, 22(4):577–609– discussion 610–60.
- [8] Barsalou, L. W. (2007). Grounded Cognition. *dx.doi.org*, 59(1):617–645.
- [9] Barsalou, L. W. (2008). Cognitive and Neural Contributions to Understanding the Conceptual System. *Current directions in psychological science*, 17(2):91–95.
- [10] Barsalou, L. W. (2010a). Grounded Cognition: Past, Present, and Future. *Topics in Cognitive Science*, 2(4):716–724.
- [11] Barsalou, L. W. (2010b). Grounded Cognition: Past, Present, and Future: Topics in Cognitive Science. *Topics in Cognitive Science*, 2(4):716–724.
- [12] Battaglia, P., Pascanu, R., Lai, M., Rezende, D. J., and Kavukcuoglu, K. (2016). Interaction Networks for Learning about Objects, Relations and Physics. pages 4502–4510.
- [13] Beer, R. D. (2016). The Dynamics of Active Categorical Perception in an Evolved Model Agent. *Adaptive Behavior*, 11(4):209–243.
- [14] Beilock, S. L. and Goldin-Meadow, S. (2010). Gesture Changes Thought by Grounding It in Action. *Psychological Science*, 21(11):1605–1610.

-
- [15] Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A Neural Probabilistic Language Model. *Journal of Machine Learning Research* (), pages 1137–1155.
- [16] Billard, A. and Siegwart, R. (2004). *Robot learning from demonstration*. Robotics and Autonomous Systems.
- [17] Bornstein, M. H. and Gibson, J. J. (1980). The Ecological Approach to Visual Perception. *The Journal of Aesthetics and Art Criticism*, 39(2):203.
- [18] Botvinick, M., Barrett, D. G. T., Battaglia, P., de Freitas, N., Kumaran, D., Leibo, J. Z., Lillicrap, T., Modayil, J., Mohamed, S., Rabinowitz, N. C., Rezende, D. J., Santoro, A., Schaul, T., Summerfield, C., Wayne, G., Weber, T., Wierstra, D., Legg, S., and Hassabis, D. (2017). Building Machines that Learn and Think for Themselves: Commentary on Lake et al., Behavioral and Brain Sciences, 2017. *arxiv.org*.
- [19] Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. (2012). Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription.
- [20] Bril, B. (2002). L'apprentissage de gestes techniques : ordre de contraintes et variations culturelles réflexions méthodologiques et anthropologiques. *Le gest technique. Réflexions méthodologiques et anthropologiques. Revue d'Anthropologie des Connaissances*, 14(2):113–149.
- [21] Brooks, R. A. (1991). Intelligence Without Reason. *IJCAI*.
- [22] Campbell, D. (2013). Radicalizing Enactivism: Basic Minds without Content By DANIEL F. HUTTO and ERIK MYIN. *Analysis*, 74:174–176.
- [23] Chahir, Y., Bouziane, A., Mostefai, M., and al alwani, A. (2014). Weakly supervised learning from scale invariant feature transform keypoints: an approach combining fast eigendecomposition, regularization, and diffusion on graphs. *Journal of Electronic Imaging*, 23(1):013009.
- [24] Chan, T.-H. (2007). A statistical analysis of the towers of hanoi problem. *International Journal of Computer Mathematics*, 28(1-4):57–65.
- [25] Charniak, E. and McDermott, D. (1985). *Introduction to Artificial Intelligence*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [26] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [27] Chomsky, N., Poeppel, D., Churchland, P., and Newport, E. L. (2011). Governing Board Symposium The Biology of Language in the 21st Century. *CogSci*.
- [28] Christopher R Madan, A. S. (2012). Using actions to enhance memory: effects of enactment, gestures, and exercise on human memory. *Frontiers in psychology*, 3:507.

References

- [29] Clancey, W. J. (1993). Situated action: A neuropsychological interpretation (Response to Vera and Simon). Technical report.
- [30] Clark, A. (1997). *Being There: Putting Mind, Body, and World Together Again*. The MIT Press.
- [31] Clark, A. J. (1999). An embodied cognitive sciences. *Trends in Cognitive Sciences*, 3:345–351.
- [32] Cochet, H. and Vauclair, J. (2014). Deictic gestures and symbolic gestures produced by adults in an experimental context: Hand shapes and hand preferences. *Laterality: Asymmetries of Body, Brain and Cognition*, 19(3):278–301.
- [33] Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing - deep neural networks with multitask learning. *ICML*, pages 160–167.
- [34] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. P. (2011). Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.*
- [35] Cook, S. W. and Tanenhaus, M. K. (2009). Embodied communication: Speakers’ gestures affect listeners’ actions. *Cognition*, 113(1):98–104.
- [36] Cook, S. W., Yip, T. K., and Goldin-Meadow, S. (2012). Gestures, but not meaningless movements, lighten working memory load when explaining math. *Language and Cognitive Processes*, 27(4):594–610.
- [37] Cun, Y. L., Boser, B., Denker, J. S., Howard, R. E., Habbard, W., Jackel, L. D., and Henderson, D. (1990). Advances in Neural Information Processing Systems 2. pages 396–404. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [38] Dawson, M. (2014). *Embedded and situated cognition*. The Routledge handbooks of embodied cognition, Routledge Handbooks.
- [39] Di Paolo, E. and Thompson, E. (2014). The enactive Approach. In Shapiro, L., editor, *The Routledge Handbook of Embodied Cognition*. Dordrecht, Holland ; Boston : D. Reidel Pub. Co.
- [40] Dominey, P. F., Hoen, M., and Inui, T. (2006). A Neurolinguistic Model of Grammatical Construction Processing. *Cognitive Neuroscience, Journal of*, 18(12):2088–2107.
- [41] Du, Y., Wang, W., and Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. *CVPR*, pages 1110–1118.
- [42] Eck, D. and Schmidhuber, J. (2002). A first look at music composition using lstm recurrent neural networks. Technical report.
- [43] Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science*, 14(2):179–211.
- [44] Escalona, M. J., Mayo, F. D., Majchrzak, T. A., and Monfort, V., editors (2019). *Web Information Systems and Technologies*. Springer International Publishing.

-
- [45] Fischer, M. H. and Zwaan, R. A. (2008). Embodied language: A review of the role of the motor system in language comprehension. *The Quarterly Journal of Experimental Psychology*, 61(6):825–850.
- [46] Gallese, V., Fadiga, L., Fogassi, L., and Rizzolatti, G. (1996). Action recognition in the premotor cortex. *Brain*, 119:593–609.
- [47] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. pages 580–587.
- [48] Glenberg, A. B. and Robertson, D. A. (2000). Symbol Grounding and Meaning: A Comparison of High-Dimensional and Embodied Theories of Meaning. *Journal of Memory and Language*, 43:379–401.
- [49] Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain Adaptation for Large-Scale Sentiment Classification - A Deep Learning Approach. *ICML*.
- [50] Goldin-Meadow, S., Cook, S. W., and Mitchell, Z. A. (2009). Gesturing Gives Children New Ideas About Math. *Psychological Science*, 20(3):267–272.
- [51] Gong, Y., Jia, Y., Leung, T., Toshev, A., and Ioffe, S. (2013). Deep Convolutional Ranking for Multilabel Image Annotation. *arXiv.org*, page arXiv:1312.4894.
- [52] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. The MIT Press.
- [53] Graves, A. and Jaitly, N. (2014). Towards End-To-End Speech Recognition with Recurrent Neural Networks. *ICML*, pages 1764–1772.
- [54] Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Networks. *arXiv.org*, page arXiv:1303.5778.
- [55] Griffiths, T. L., Chater, N., Kemp, C., Perfors, A., and Tenenbaum, J. B. (2010). Probabilistic models of cognition: exploring representations and inductive biases. *Trends in Cognitive Sciences*, 14(8):357–364.
- [56] Harnad, S. (1990). The Symbol grounding problem. *Physica D: Nonlinear Phenomena*.
- [57] Hayes, G. M. and Demiris, J. (1994). A robot controller using learning by imitation.
- [58] Hermann, K. M. and Blunsom, P. (2013). The Role of Syntax in Vector Space Models of Compositional Semantics. *ACL*.
- [59] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.
- [60] Hochreiter, S. and Schmidhuber, J. (2006). Long Short-Term Memory. *dx.doi.org*, 9(8):1735–1780.
- [61] Hostetter, A. B., Alibali, M. W., and Kita, S. (2007). I see it in my handsâ eye: Representational gestures reflect conceptual demands. *Language and Cognitive Processes*, 22(3):313–336.

References

- [62] Hubel, D. H. and Wiesel, T. N. (1959). Receptive Fields of Single Neurons in the Cat's Striate Cortex. *Journal of Physiology*, 148:574–591.
- [63] Hurley, S. (2001). Perception And Action: Alternative Views. *Synthese*, 129:3–40.
- [64] Inoue, S. and Matsuzawa, T. (2007). Working memory of numerals in chimpanzees. *Current Biology*, 17(23):R1004–R1005.
- [65] J Chalmers, D. (2010). The Singularity: A Philosophical Analysis. *Journal of Consciousness Studies*, 17:7–65.
- [66] Jaques, N., Gu, S., Turner, R. E., and Eck, D. (2016). Tuning Recurrent Neural Networks with Reinforcement Learning. *arXiv.org*.
- [67] Jouen, F. and Molina, M. (2005). Exploration of the newborn's manual activity: A window onto early cognitive processes. *Infant Behavior and Development*, 28(3):227–239.
- [68] Kahneman, D. (2011). *Thinking, fast and slow*. Farrar, Straus and Giroux, New York.
- [69] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *EMNLP*, pages 1746–1751.
- [70] Kiverstein, J. and Miller, M. (2015). The embodied brain: towards a radical embodied cognitive neuroscience. *Frontiers in Human Neuroscience*, 9.
- [71] Knott, A. (2011). Chomsky and embodied cognition - A sensorimotor interpretation of Minimalist logical form. *CogSci*.
- [72] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012a). ImageNet Classification with Deep Convolutional Neural Networks. *papers.nips.cc*, pages 1097–1105.
- [73] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012b). Imagenet classification with deep convolutional neural networks. In *NIPS*.
- [74] Kumar, A. and others (2015). Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. *arxiv.org*, page arXiv:1506.07285.
- [75] Lakoff, G. and Johnson, M. (1999). *Philosophy in the flesh: the embodied mind and its challenge to western thought*. New York: Basic books.
- [76] Lakoff, G. and Nuñez, R. (2001). *Where mathematics come from: how the embodied mind brings mathematics into being*. Basic Books, [S.l.].
- [77] Lawrence, S., Giles, C. L., and Fong, S. (2000). Natural language grammatical inference with recurrent neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 12(1):126–140.
- [78] Le Cun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Handwritten Digit Recognition with a Back-propagation Network. In *Proceedings of the 2Nd International Conference on Neural Information Processing Systems*, pages 396–404, Cambridge, MA, USA. MIT Press.

-
- [79] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324.
- [80] Léger, L., Rouet, J.-F., Ros, C., and Vibert, N. (2012). Orthographic versus semantic matching in visual search for words within lists. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, 66(1):32–43.
- [81] Leighton, J. P. and Sternberg, R. J. (2004). *The nature of reasoning*. Cambridge University Press.
- [82] Liu, Q., Jiang, H., Evdokimov, A., Ling, Z.-H., Zhu, X., Wei, S., and Hu, Y. (2016). Probabilistic Reasoning via Deep Learning: Neural Association Models. *arXiv.org*, page arXiv:1603.07704.
- [83] Maturana, H. R. and Varela, F. J. . (1980). *Autopoiesis and cognition : the realization of the living*. Dordrecht, Holland ; Boston : D. Reidel Pub. Co.
- [84] Mikolov, T., 0010, K. C., Corrado, G., and Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. *CoRR*, cs.CL:arXiv:1301.3781.
- [85] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. *INTERSPEECH*, pages 1045–1048.
- [86] Mikolov, T., Kombrink, S., Burget, L., Cernocký, J., and Khudanpur, S. Extensions of recurrent neural network language model. In *ICASSP 2011 - 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531. IEEE.
- [87] Mikolov, T., Kombrink, S., Burget, L., Signal, S., , and 2011. Extensions of recurrent neural network language model. *ieeexplore.ieee.org*.
- [88] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed Representations of Words and Phrases and their Compositionality. pages 3111–3119.
- [89] Mikolov, T. and Zweig, G. (2012). Context dependent recurrent neural network language model. *SLT*, pages 234–239.
- [90] Minsky, M. (1986). *The society of mind*. New York : Simon and Schuster.
- [91] Minsky, M. and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, USA.
- [92] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.
- [93] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. *arXiv.org*, page arXiv:1312.5602.

References

- [94] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.
- [95] Morsella, E. and Krauss, R. M. (2004). The Role of Gestures in Spatial Working Memory and Speech. *The American journal of psychology*, 117(3):411.
- [96] Mueller, E. T. (2015). *Commonsense reasoning: an event calculus based approach; 2nd ed.* Morgan Kaufmann, Waltham, MA.
- [97] Nathan, M. J. (2008). An embodied cognition perspective on symbols, gesture, and grounding instruction. *Symbols*.
- [98] Nicolescu, M. and Mataric, M. J. (2005). Task learning through imitation and human-robot interaction. *Models and mechanisms of imitation and social . . .*
- [99] Niedenthal, P. M. (2007). Embodying emotion. *Science (New York, N.Y.)*, 316(5827):1002–1005.
- [100] Nilsson, N. J. (1980). *Principles of Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [101] Nilsson, N. J. (1998). *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [102] Nyamapfene, A. (2011). Towards Understanding Child Language Acquisition: An Unsupervised Multimodal Neural Network Approach. *J. Inf. Sci. Eng. ()*, 27(5):1613–1639.
- [103] Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. pages 1717–1724.
- [104] Parikh, A. P., Täckström, O., Das, D., and Uszkoreit, J. (2016). A Decomposable Attention Model for Natural Language Inference.
- [105] Peng, B., Lu, Z., Li, H., and Wong, K.-F. (2015). Towards Neural Network-based Reasoning. *arXiv.org*, page arXiv:1508.05508.
- [106] Pezzulo, G., Barsalou, L. W., Cangelosi, A., Fischer, M. H., McRae, K., and Spivey, M. J. (2012). Computational Grounded Cognition: a new alliance between grounded cognition and computational modeling. *Frontiers in psychology*, 3:612.
- [107] Pezzulo, G., Barsalou, L. W., Cangelosi, A., Fischer, M. H., Spivey, M., and McRae, K. (2011). The Mechanics of Embodiment: A Dialog on Embodiment and Computational Modeling. *Frontiers in psychology*, 2.
- [108] Pezzulo, G. and Calvi, G. (2011). Computational explorations of perceptual symbol systems theory. *New Ideas in Psychology*, 29(3):275–297.

-
- [109] Pfeifer, R. and Iida, F. (2004). Embodied Artificial Intelligence: Trends and Challenges. In *Embodied Artificial Intelligence*, pages 1–26. Springer, Berlin, Heidelberg, Berlin, Heidelberg.
- [110] Pfeifer, R. and Scheier, C. (1999). *Understanding Intelligence*. MIT Press, Cambridge, MA, USA.
- [111] Piaget, J. and Cook, M. (1952). The origins of intelligence in children.
- [112] Piaget, Jean (1971). The theory of stages in cognitive development. In *Measurement and Piaget.*, pages ix, 283–ix, 283. McGraw-Hill SN -, New York, NY, US.
- [113] Poitrenaud, S. (1995). The PROCOPE semantic network: an alternative to action grammars. *International Journal of Human-Computer Studies*, 42(1):31–69.
- [114] Poitrenaud, S., RICHARD, J.-F., and Tijus, C. (2005). Properties, categories, and categorisation. *Thinking and Reasoning*, 11(2):151–208.
- [115] Prinz, W. (2012). *Open Minds: The Social Making of Agency and Intentionality*. The MIT Press.
- [116] Prinz, W., Beisert, M., and Herwig, A., editors (2013). *Action Science: Foundations of an Emerging Discipline*. The MIT Press.
- [117] Putnam, H. (1981). *Reason, Truth and History*. Cambridge University Press.
- [118] Pylyshyn, Z. W. (2000). Situating vision in the world. *Trends in Cognitive Sciences*, 4(5):197–207.
- [119] Ramanathan, V., Li, C., Deng, J., Han, W., Li, Z., Gu, K., Song, Y., Bengio, S., Rosenberg, C., and Li, F.-F. (2015). Learning semantic relationships for better action retrieval in images. *CVPR*, pages 1100–1109.
- [120] Raposo, D. and others (2017). Discovering objects and their relations from entangled scene representations. *CoRR*, abs/1702.05068.
- [121] RICHARD, J.-F. (1997). La résolution de problèmes. *Recherche en soins infirmiers*, (50).
- [122] RICHARD, J.-F., Poitrenaud, S., and Tijus, C. (1993). Problem-Solving Restructuration: Elimination of Implicit Constraints. *Cognitive Science*, 17(4):497–529.
- [123] RICHARD, J.-F. and VERSTIGGEL, J.-C. (1990). La représentation de l’action dans les processus de compréhension. *Langages*, (100):115–126.
- [124] Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. *Psychological Review*, pages 65–386.
- [125] Rowlands, M. (2010). *The New Science of the Mind: From Extended Mind to Embodied Phenomenology*. A Bradford Book. MIT Press.

References

- [126] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252.
- [127] Russell, S. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition.
- [128] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013). OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks.
- [129] Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. pages 806–813.
- [130] Singer, M. A. and Goldin-Meadow, S. (2005). Children Learn When Their Teacher’s Gestures and Speech Differ. *Psychological Science*, 16(2):85–89.
- [131] Smith, L. and Gasser, M. (2006). The Development of Embodied Cognition: Six Lessons from Babies. *dx.doi.org*, 11(1-2):13–29.
- [132] Sutskever, I., Hinton, G., and Taylor, G. (2008). The Recurrent Temporal Restricted Boltzmann Machine. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, pages 1601–1608, USA. Curran Associates Inc.
- [133] Sutskever, I., Martens, J., and Hinton, G. (2011). Generating Text with Recurrent Neural Networks. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1017–1024, USA. Omnipress.
- [134] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. The MIT Press, second edition.
- [135] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going Deeper With Convolutions. *cv-foundation.org*, pages 1–9.
- [136] Thomaz, A. L., Hoffman, G., and Breazeal, C. (2006). Reinforcement Learning with Human Teachers - Understanding How People Want to Teach Robots. *RO-MAN*.
- [137] Truck, I. (2015). Comparison and links between two 2-tuple linguistic models for decision making. *Knowledge-Based Systems*, 87:61–68.
- [138] Uexküll, J. v. (2001). An introduction to Umwelt. *Semiotica*, 2001(134).
- [139] Varela, F. J., Thompson, E. T., and Rosch, E. (1992). *The Embodied Mind: Cognitive Science and Human Experience*. The MIT Press, new edition edition.
- [140] Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3-4):279–292.
- [141] Williams, L. E. and Bargh, J. A. (2008). Experiencing Physical Warmth Promotes Interpersonal Warmth. *Science (New York, N.Y.)*, 322(5901):606–607.

-
- [142] Wilson, A. D. and Golonka, S. (2013). Embodied cognition is not what you think it is. *Frontiers in Psychology*, 4.
- [143] Wilson, M. (2002). Six views of embodied cognition. *Psychonomic Bulletin and Review*, 9(4):625–636.
- [144] Yacoubi, A. and Sabouret, N. (2018). Expression verbale des tendances à l’action en langue française (verbal expression of action tendencies in french). In *CNIA+RJICA*.
- [145] Yamins, D. L., Hong, H., Cadieu, C., and DiCarlo, J. J. (2013). Hierarchical Modular Optimization of Convolutional Networks Achieves Representations Similar to Macaque IT and Human Ventral Stream. pages 3093–3101.
- [146] Yao, K., Zweig, G., Hwang, M.-Y., Shi, Y., and Yu, D. (2013). Recurrent neural networks for language understanding. *INTERSPEECH*, pages 2524–2528.
- [147] Zajac, F. E. (1993). Muscle coordination of movement: A perspective. *Journal of Biomechanics*, 26:109–124.
- [148] Zajonc, R. B. (2016). Mere Exposure: A Gateway to the Subliminal:. *Current directions in psychological science*, 10(6):224–228.
- [149] Zanga, A., RICHARD, J.-F., and Tijus, C. (2004). Implicit learning in rule induction and problem solving. *Thinking and Reasoning*, 10(1):55–83.

Appendix A

Results of Tower of Hanoi experiments on blind and sighted groups.

A.1 Results of Tower of Hanoi experiments for sighted group

Table A.1 Results of Tower of Hanoi experiments for sighted group

Participants	Condition	Group	A ¹	B ¹	C ¹	D ¹	E ²	F ²	G ³	H ³	I ⁴	J ⁴	K ⁵	L ⁵	M ⁶	N ⁶	O ⁷	P ⁷
P1DB	Disk	NonBlind	1	0	2	2	99	106	15	25	0	0	29	50	6	6	0	0
P2DB	Disk	NonBlind	17	2	17	1	284	371	21	66	0	0	43	126	9	19	2	2
P3DB	Disk	NonBlind	1	0	2	1	169	74	27	24	0	0	50	49	8	4	4	1
P4DB	Disk	NonBlind	1	0	2	1	200	106	50	32	0	0	105	66	10	6	1	0
P5DB	Disk	NonBlind	1	0	1	1	91	29	23	15	0	0	40	21	4	2	0	0
P6DB	Disk	NonBlind	2	1	2	1	284	116	34	26	0	0	73	53	21	9	0	0
P7DB	Disk	NonBlind	1	2	1	1	133	186	30	38	0	0	55	68	8	9	3	0
P1WB	WithoutDisk	NonBlind	1	0	3	1	182	42	25	15	2	0	54	30	6	3	0	0
P2WB	WithoutDisk	NonBlind	9	6	2	1	604	401	38	29	10	5	56	64	7	10	0	0
P3WB	WithoutDisk	NonBlind	6	1	2	1	401	71	21	16	35	33	54	34	1	0	0	34
P4WB	WithoutDisk	NonBlind	3	2	1	1	193	379	27	53	3	8	53	117	0	0	1	0
P5WB	WithoutDisk	NonBlind	2	1	1	1	79	60	16	17	0	12	26	30	3	1	0	0
P6WB	WithoutDisk	NonBlind	2	3	1	1	152	146	20	19	3	5	40	41	8	4	0	0
P7WB	WithoutDisk	NonBlind	4	7	1	1	247	361	20	27	39	32	53	45	0	0	0	0

¹ A: Average time between 2 gestures - T1, B:Average time between 2 gestures - T2, C:Average gesture time -T1, D: Average gesture time - T2

² E:Total duration - T1, F:Total duration - T2

³ G:Number of strokes - T1, H:Number of strokes - T2

⁴ I:Number of deictic gestures - T1, J:Number of deictic gestures - T2

⁵ K:Number of gestures - T1, L:Number of gestures - T2

⁶ M:Number of gesture interaction - T1,N:Number of gesture interaction - T2

⁷ O:Number of illicite gestures - T1, P:Number of illicite gestures - T2

A.2 Results of Tower of Hanoi experiments for the blind group

Table A.2 Results of TOH experiments for the blind group

Participants	Condition	Group	A ¹	B ¹	C ¹	D ¹	E ²	F ²	G ³	H ³	I ⁴	J ⁴	K ⁵	L ⁵	M ⁶	N ⁶	O ⁷	P ⁷
P1DN	Disk	Blind	2	1	3	3	489	97	36	17	0	0	63	27	43	9	3	0
P2DN	Disk	Blind	2	0	1	1	313	54	47	20	0	0	88	41	17	7	6	0
P3DN	Disk	Blind	3	0	2	1	981	1325	125	150	0	0	209	230	33	30	4	7
P4DN	Disk	Blind	0	0	2	1	58	37	15	17	0	0	29	29	10	4	0	0
P5DN	Disk	Blind	2	8	3	2	229	1076	27	66	0	0	45	112	13	25	3	2
P6DN	Disk	Blind	1	1	2	1	250	245	48	62	0	0	83	97	16	22	12	7
P7DN	Disk	Blind	1	1	2	1	76	186	17	21	0	0	32	41	9	4	0	0
P1WN	WithoutDisk	Blind	8	5	1	1	346	187	22	23	9	1	38	30	0	0	1	0
P2WN	WithoutDisk	Blind	2	0	1	1	198	199	30	36	0	4	58	79	1	4	2	1
P3WN	WithoutDisk	Blind	7	0	1	1	1126	1356	83	150	37	0	147	230	0	0	4	7
P4WN	WithoutDisk	Blind	0	0	1	1	35	27	15	15	0	0	28	24	2	1	0	0
P5WN	WithoutDisk	Blind	11	2	2	1	503	72	23	15	0	1	41	24	1	0	0	0
P6WN	WithoutDisk	Blind	3	4	1	0	145	363	18	40	0	2	37	74	1	1	0	1
P7WN	WithoutDisk	Blind	3	1	1	1	328	447	40	57	0	8	80	124	1	1	0	1

¹ A: Average time between 2 gestures -T1, B:Average time between 2 gestures -T2, C:Average gesture time -T1, D: Average gesture time - T2

² E:Total duration - T1, F:Total duration - T2

³ G:Number of strokes - T1, H:Number of strokes - T2

⁴ I:Number of deictic gestures - T1, J:Number of deictic gestures - T2

⁵ K:Number of gestures - T1, L:Number of gestures - T2

⁶ M:Number of gesture interaction - T1,N:Number of gesture interaction - T2

⁷ O:Number of illicite gestures - T1, P:Number of illicite gestures - T2

A.2.1 Visualization of the correlation matrix

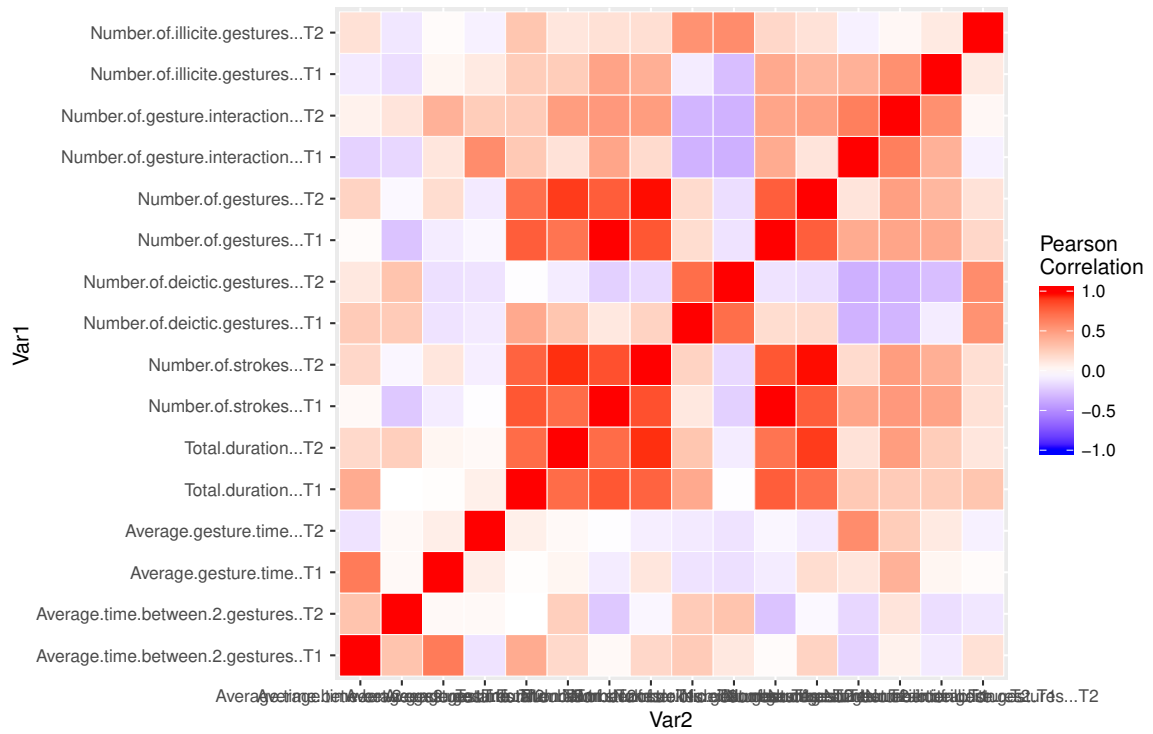


Fig. A.1 Correlation Matrix Type1

Results of Tower of Hanoi experiments on blind and sighted groups.

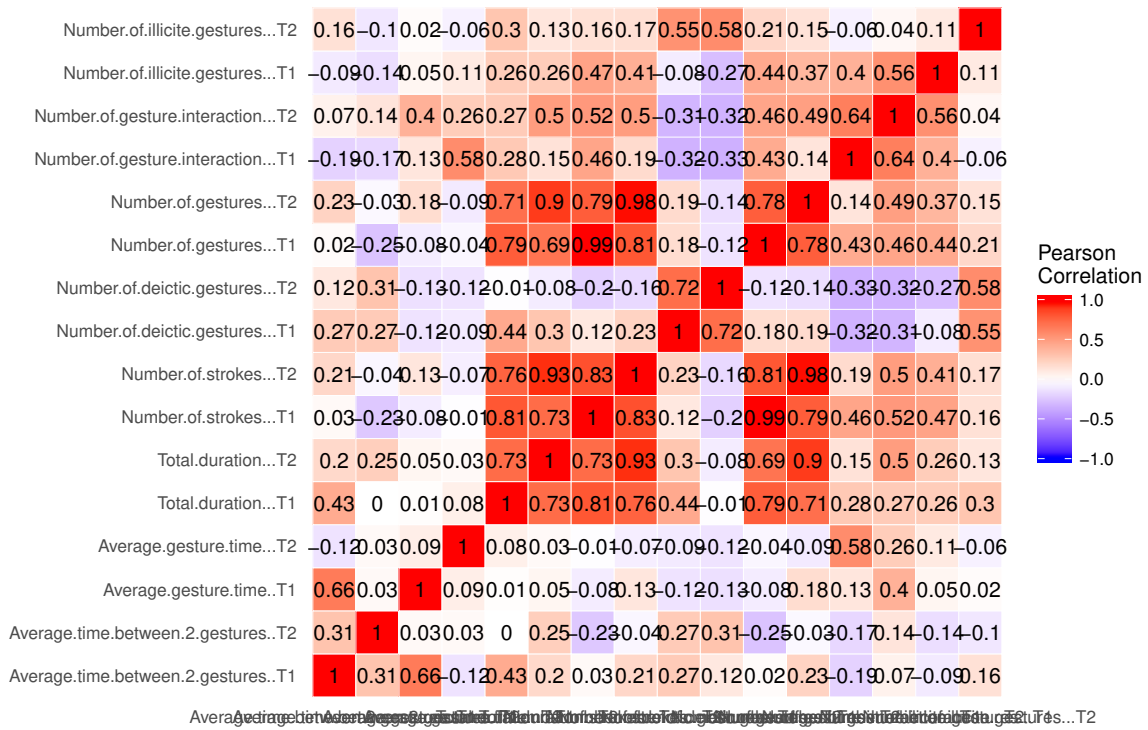


Fig. A.2 Correlation Matrix Type 2

A.3 Summary Two-Way ANOVA and Interaction Effects Results

A.3.1 Interaction effects for deictic gesture

Table A.3 Summary Two-Way ANOVA and Interaction Effects Results for Number of deictic gesture on first try by condition and group

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Condition	1	680.14	680.14	5.82	0.0239
Group	1	75.57	75.57	0.65	0.4294
Condition:Group	1	75.57	75.57	0.65	0.4294
Residuals	24	2806.57	116.94		

Table A.4 Summary Two-Way ANOVA and Interaction Effects Results for Number of deictic gestures on second try by condition and group

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Condition	1	440.04	440.04	9.34	0.0054
Group	1	222.89	222.89	4.73	0.0397
Condition:Group	1	222.89	222.89	4.73	0.0397
Residuals	24	1131.14	47.13		

A.3.2 Interaction effects for gesture interaction

Table A.5 Summary Two-Way ANOVA and Interaction Effects Results for Number of gesture interaction on first try by condition and group

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Condition	1	1106.29	1106.29	21.32	0.0001
Group	1	112.00	112.00	2.16	0.1547
Condition:Group	1	315.57	315.57	6.08	0.0212
Residuals	24	1245.14	51.88		

Table A.6 Summary Two-Way ANOVA and Interaction Effects Results for Number of gesture interaction on second try by condition and group

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Condition	1	612.89	612.89	14.88	0.0008
Group	1	43.75	43.75	1.06	0.3129
Condition:Group	1	116.04	116.04	2.82	0.1062
Residuals	24	988.29	41.18		

Appendix B

Tower of Hanoi Dataset

B.1 Tower of Hanoi Resolution Dataset for MLP

B.1.1 Movement encoding

Num.	Coding for 4 disks	Num.	Coding for 4 disks
1	1,1,1,1,0,0,0,0,0,0,0,0,1,0,0,0,1,0	2	0,1,1,1,0,0,0,1,0,0,0,0,1,0,0,0,0,1
3	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1	4	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0
5	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,1,0,0	6	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,0,1,0
7	0,0,1,1,0,0,1,1,0,0,0,0,1,0,0,0,1,0	8	0,0,0,1,0,1,1,1,0,0,0,0,1,0,0,0,0,1
9	0,0,0,0,0,1,1,1,0,0,0,1,0,1,0,0,0,1	10	0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,0,0
11	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,1,0,0	12	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1
13	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0	14	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,0,1
15	0,0,0,0,0,0,0,1,0,1,1,1,0,1,0,0,0,1	16	1,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1
17	0,1,1,1,0,0,0,0,0,0,0,1,1,0,0,0,1,0	18	0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,0,0,1
19	0,0,0,1,0,0,0,1,0,0,1,1,0,1,0,0,0,1	20	0,0,0,1,0,0,0,0,0,1,1,1,0,0,1,0,1,0
21	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,0,1	22	0,0,0,0,0,0,0,1,0,1,1,1,0,1,0,1,0,0
23	0,0,0,1,0,0,0,0,0,1,1,1,0,0,1,1,0,0	24	0,0,1,1,0,0,0,0,0,0,1,1,0,0,1,0,1,0
25	0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,0,0,1	26	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,1,0
27	0,0,0,0,0,0,1,1,0,0,1,1,0,0,1,0,1,0	28	0,0,0,0,0,1,1,1,0,0,0,1,0,1,0,1,0,0
29	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,0,0,1	30	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,0,1

B.1 Tower of Hanoi Resolution Dataset for MLP

Num.	Coding for 4 disks	Num.	Coding for 4 disks
31	0,0,0,0,0,0,0,1,0,1,1,1,0,0,1,0,1,0	32	0,0,0,0,0,0,0,1,1,0,0,1,1,0,0,1,1,0,0
33	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,1,0,0	34	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1
35	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0	36	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,0,1
37	0,0,0,0,0,0,0,1,0,1,1,1,0,1,0,0,0,1	38	1,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0
39	0,1,1,1,0,0,0,1,0,0,0,0,1,0,0,0,0,1	40	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1
41	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0	42	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,0,1,0
43	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,1,0,0	44	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,0,1,0
45	0,0,1,1,0,0,1,1,0,0,0,0,1,0,0,0,1,0	46	0,0,0,1,0,1,1,1,0,0,0,0,1,0,0,0,0,1
47	0,0,0,0,0,1,1,1,0,0,0,1,0,1,0,0,0,1	48	0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,0,0
49	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,0,1,0	50	0,0,0,1,0,0,1,1,0,0,0,1,1,0,0,0,0,1
51	0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,0,0	52	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,1,0
53	0,0,0,0,0,0,1,1,0,0,1,1,0,0,1,1,0,0	54	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,1,0,0
55	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1	56	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0
57	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,0,1	58	0,0,0,0,0,0,0,1,0,1,1,1,0,1,0,0,0,1
59	1,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0	60	0,1,1,1,0,0,0,1,0,0,0,0,1,0,0,0,0,1
61	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,1,0,0	62	0,1,1,1,0,0,0,1,0,0,0,0,0,1,0,0,0,1
63	0,1,1,1,0,0,0,0,0,0,0,1,1,0,0,0,1,0	64	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,1,0,0
65	0,1,1,1,0,0,0,1,0,0,0,0,0,1,0,0,0,1	66	0,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1
67	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0	68	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,0,1,0
69	0,0,0,1,0,0,1,1,0,0,0,1,0,0,1,1,0,0	70	0,0,1,1,0,0,1,1,0,0,0,0,0,1,0,0,0,1
71	0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,0,1,0	72	0,0,0,1,0,0,1,1,0,0,0,1,0,0,1,0,1,0
73	0,0,0,1,0,1,1,1,0,0,0,0,1,0,0,0,0,1	74	0,0,0,0,0,1,1,1,0,0,0,1,0,1,0,1,0,0
75	0,0,0,1,0,0,1,1,0,0,0,1,1,0,0,0,0,1	76	0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,0,0
77	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,1,0,0	78	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1
79	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0	80	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,0,1
81	0,0,0,0,0,0,0,1,0,1,1,1,0,1,0,0,0,1	82	1,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1

Num.	Coding for 4 disks	Num.	Coding for 4 disks
83	0,1,1,1,0,0,0,0,0,1,0,0,1,0,1,0	84	0,1,1,1,0,0,0,1,0,0,0,0,0,1,0,0,0,1
85	0,1,1,1,0,0,0,0,0,0,1,1,0,0,0,1,0	86	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,0,1,0
87	0,0,1,1,0,0,1,1,0,0,0,0,1,0,0,0,0,1	88	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,1,0,0
89	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1	90	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,1
91	0,0,0,1,0,0,0,0,0,1,1,1,1,0,0,0,1,0	92	0,0,0,0,0,0,0,1,0,1,1,1,0,0,1,1,0,0
93	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,0,1,0	94	0,0,0,1,0,0,1,1,0,0,0,1,1,0,0,0,0,1
95	0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,0,0	96	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,1,0
97	0,0,0,0,0,0,1,1,0,0,1,1,0,0,1,0,1,0	98	0,0,0,0,0,1,1,1,0,0,0,1,0,0,1,1,0,0
99	0,0,0,1,0,1,1,1,0,0,0,0,0,1,0,0,0,1	100	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,1,0,0
101	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,1,0,0	102	0,1,1,1,0,0,0,1,0,0,0,0,0,1,0,0,0,1
103	0,1,1,1,0,0,0,0,0,0,0,1,1,0,0,0,1,0	104	0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,0,0,1
105	0,0,0,1,0,0,0,1,0,0,1,1,0,1,0,0,0,1	106	0,0,0,1,0,0,0,0,0,1,1,1,1,0,0,0,1,0
107	0,0,0,0,0,0,0,1,0,1,1,1,0,0,1,0,1,0	108	0,0,0,0,0,0,1,1,0,0,1,1,0,0,1,1,0,0
109	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,1,0,0	110	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1
111	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0	112	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,0,1
113	0,0,0,0,0,0,0,1,0,1,1,1,0,1,0,0,0,1	114	1,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1
115	0,1,1,1,0,0,0,0,0,0,0,1,1,0,0,0,1,0	116	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,0,1,0
117	0,0,1,1,0,0,1,1,0,0,0,0,1,0,0,0,0,1	118	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,1,0,0
119	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1	120	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,1
121	0,0,0,1,0,0,0,0,0,1,1,1,1,0,0,0,1,0	122	0,0,0,0,0,0,0,1,0,1,1,1,0,0,1,1,0,0
123	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,0,1,0	124	0,0,0,1,0,0,1,1,0,0,0,1,1,0,0,0,1,0
125	0,0,0,0,0,1,1,1,0,0,0,1,0,0,1,1,0,0	126	0,0,0,1,0,1,1,1,0,0,0,0,0,1,0,1,0,0
127	0,0,1,1,0,0,1,1,0,0,0,0,0,1,0,0,0,1	128	0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,0,1,0
129	0,0,0,1,0,0,1,1,0,0,0,1,0,0,1,1,0,0	130	0,0,1,1,0,0,1,1,0,0,0,0,0,1,0,1,0,0
131	0,1,1,1,0,0,0,1,0,0,0,0,0,1,0,0,0,1	132	0,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0,1,0
133	0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,0,0,1	134	0,0,0,1,0,0,0,1,0,0,1,1,0,1,0,0,0,1

B.1 Tower of Hanoi Resolution Dataset for MLP

Num.	Coding for 4 disks	Num.	Coding for 4 disks
135	0,0,0,1,0,0,0,0,0,1,1,1,1,0,0,0,1,0	136	0,0,0,0,0,0,0,0,1,0,1,1,1,0,0,1,0,1,0
137	0,0,0,0,0,0,1,1,0,0,1,1,0,0,1,1,0,0	138	0,0,0,1,0,0,1,1,0,0,0,1,1,0,0,0,1,0
139	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,1,0,0	140	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1
141	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0	142	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,0,1
143	0,0,0,0,0,0,0,1,0,1,1,1,0,1,0,0,0,1	144	1,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1
145	0,1,1,1,0,0,0,0,0,0,0,1,1,0,0,0,1,0	146	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,0,1,0
147	0,0,1,1,0,0,1,1,0,0,0,0,1,0,0,0,0,1	148	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,0,0,1
149	0,0,0,1,0,0,0,1,0,0,1,1,0,1,0,1,0,0	150	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0
151	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,0,1,0	152	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,1,0,0
153	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1	154	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,1
155	0,0,0,1,0,0,0,0,0,1,1,1,1,0,0,0,1,0	156	0,0,0,0,0,0,0,1,0,1,1,1,0,0,1,0,1,0
157	0,0,0,0,0,0,1,1,0,0,1,1,0,0,1,1,0,0	158	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,1,0,0
159	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,0,1,0	160	0,0,1,1,0,0,1,1,0,0,0,0,0,1,0,0,0,1
161	0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,0,1,0	162	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,0,0,1
163	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,1,0	164	0,0,0,0,0,0,1,1,0,0,1,1,0,0,1,0,1,0
165	0,0,0,0,0,1,1,1,0,0,0,1,0,0,1,1,0,0	166	0,0,0,1,0,1,1,1,0,0,0,0,0,1,0,1,0,0
167	0,0,1,1,0,0,1,1,0,0,0,0,0,1,0,0,0,1	168	0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,0,1,0
169	0,0,0,1,0,0,1,1,0,0,0,1,0,0,1,1,0,0	170	0,0,1,1,0,0,1,1,0,0,0,0,0,1,0,1,0,0
171	0,1,1,1,0,0,0,1,0,0,0,0,0,1,0,0,0,1	172	0,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1
173	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0	174	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,0,1,0
175	0,0,0,1,0,0,1,1,0,0,0,1,1,0,0,0,0,1	176	0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,0,0
177	0,0,0,1,0,0,0,1,0,0,1,1,0,1,0,0,0,1	178	0,0,0,1,0,0,0,0,0,1,1,1,1,0,0,0,0,1
179	1,1,1,1,0,0,0,0,0,0,0,0,1,0,0,0,1,0	180	0,1,1,1,0,0,0,1,0,0,0,0,1,0,0,0,0,1
181	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1	182	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0
183	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,1,0,0	184	0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,0,0,1
185	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,1,0,0	186	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,0,1,0

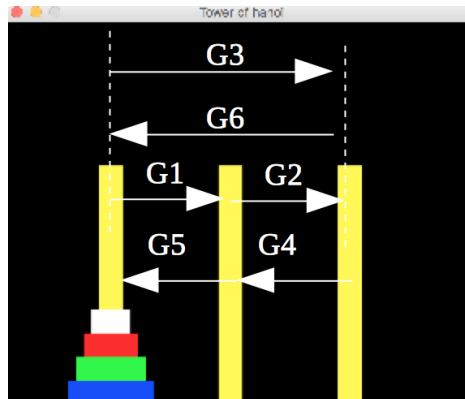
Num.	Coding for 4 disks	Num.	Coding for 4 disks
187	0,0,1,1,0,0,1,1,0,0,0,0,1,0,0,0,0,1	188	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,1,0,0
189	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,0,1,0	190	0,0,1,1,0,0,1,1,0,0,0,0,0,1,0,0,0,1
191	0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,0,1,0	192	0,0,0,1,0,0,1,1,0,0,0,1,0,0,1,0,1,0
193	0,0,0,1,0,1,1,1,0,0,0,0,1,0,0,0,0,1	194	0,0,0,0,0,1,1,1,0,0,0,1,0,1,0,0,0,1
195	0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,0,0	196	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,0,1,0
197	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,1,0,0	198	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1
199	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0	200	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,0,1
201	0,0,0,0,0,0,0,1,0,1,1,1,0,1,0,0,0,1	202	1,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0
203	0,1,1,1,0,0,0,1,0,0,0,0,1,0,0,0,0,1	204	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1
205	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0	206	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,1,0,0
207	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,0,1,0	208	0,0,1,1,0,0,1,1,0,0,0,0,1,0,0,0,1,0
209	0,0,0,1,0,1,1,1,0,0,0,0,1,0,0,0,0,1	210	0,0,0,0,0,1,1,1,0,0,0,1,0,1,0,0,0,1
211	0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,0,0	212	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,1,0,0
213	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1	214	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0
215	0,0,0,1,0,0,0,1,0,0,1,1,1,0,0,0,0,1	216	0,0,0,0,0,0,0,1,0,1,1,1,0,1,0,0,0,1
217	1,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1	218	0,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0,1,0
219	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,1,0,0	220	0,1,1,1,0,0,0,1,0,0,0,0,0,1,0,0,0,1
221	0,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0,1,0	222	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1
223	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0	224	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,1,0,0
225	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,0,1,0	226	0,0,1,1,0,0,1,1,0,0,0,0,1,0,0,0,1,0
227	0,0,0,1,0,1,1,1,0,0,0,0,1,0,0,0,0,1	228	0,0,0,0,0,1,1,1,0,0,0,1,0,1,0,1,0,0
229	0,0,0,1,0,0,1,1,0,0,0,1,1,0,0,0,0,1	230	0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,0,0
231	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,0,1,0	232	0,0,0,1,0,0,1,1,0,0,0,1,1,0,0,0,0,1
233	0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,0,0	234	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,0,1,0
235	0,0,0,1,0,0,1,1,0,0,0,1,1,0,0,0,1,0	236	0,0,0,0,0,1,1,1,0,0,0,1,0,1,0,1,0,0
237	0,0,0,1,0,0,1,1,0,0,0,1,1,0,0,0,0,1	238	0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,0,0

B.1 Tower of Hanoi Resolution Dataset for MLP

Num.	Coding for 4 disks	Num.	Coding for 4 disks
239	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,1,0,0	240	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1
241	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0	242	0,0,0,1,0,0,0,1,0,0,1,1,0,1,0,0,0,1
243	0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,0,0,1	244	1,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0
245	0,1,1,1,0,0,0,1,0,0,0,0,1,0,0,0,0,1	246	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1
247	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0	248	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,0,1,0
249	0,0,0,1,0,0,1,1,0,0,0,1,0,0,1,1,0,0	250	0,0,1,1,0,0,1,1,0,0,0,0,0,1,0,0,0,1
251	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,1,0,0	252	0,1,1,1,0,0,0,1,0,0,0,0,0,1,0,0,0,1
253	0,1,1,1,0,0,0,0,0,0,0,1,1,0,0,0,1,0	254	0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,0,0,1
255	0,0,0,1,0,0,0,1,0,0,1,1,0,1,0,0,0,1	256	0,0,0,1,0,0,0,0,0,1,1,1,1,0,0,0,1,0
257	0,0,0,0,0,0,0,1,0,1,1,1,0,0,1,0,1,0	258	0,0,0,0,0,0,1,1,0,0,1,1,0,0,1,1,0,0
259	0,0,0,1,0,0,1,1,0,0,0,1,0,1,0,1,0,0	260	0,0,1,1,0,0,0,1,0,0,0,1,0,0,1,0,1,0
261	0,0,1,1,0,0,1,1,0,0,0,0,1,0,0,0,0,1	262	0,0,0,1,0,0,1,1,0,0,0,1,1,0,0,0,1,0
263	0,0,0,0,0,1,1,1,0,0,0,1,0,0,1,1,0,0	264	0,0,0,1,0,1,1,1,0,0,0,0,1,0,0,0,0,1
265	0,0,0,0,0,1,1,1,0,0,0,1,0,0,1,0,1,0	266	0,0,0,0,1,1,1,1,0,0,0,0,0,1,0,1,0,0
267	0,0,0,1,0,1,1,1,0,0,0,0,0,1,0,0,0,1	268	0,0,0,1,0,0,1,1,0,0,0,1,1,0,0,0,0,1
269	0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,0,0	270	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,0,1,0
271	0,0,0,1,0,0,1,1,0,0,0,1,0,0,1,1,0,0	272	0,0,1,1,0,0,1,1,0,0,0,0,0,1,0,1,0,0
273	0,1,1,1,0,0,0,1,0,0,0,0,0,1,0,0,0,1	274	0,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0,1,0
275	0,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,0,1	276	0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0
277	0,0,0,1,0,0,0,1,0,0,1,1,0,0,1,0,1,0	278	0,0,0,1,0,0,1,1,0,0,0,1,1,0,0,0,0,1
279	0,0,0,0,0,0,1,1,0,0,1,1,0,1,0,1,0,0	280	0,0,0,1,0,0,0,1,0,0,1,1,0,1,0,0,0,1
281	0,0,0,1,0,0,0,0,0,1,1,1,1,0,0,0,0,1		

B.2 Tower of Hanoi Resolution Dataset for Recurrent Neural Network

B.2.1 Movement encoding



- G0: Initial state
- G1: Move a disk from peg A to peg B
- G2: Move a disk from peg B to peg C
- G3: Move a disk from peg A to peg C
- G4: Move a disk from peg C to peg B
- G5: Move a disk from peg B to peg A
- G6: Move a disk from peg C to peg A

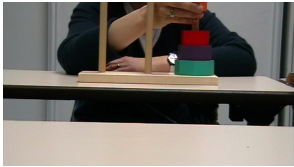
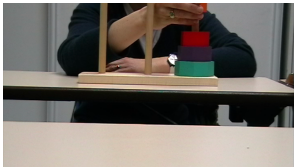
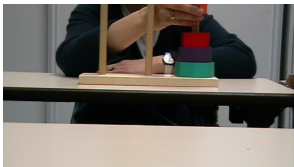
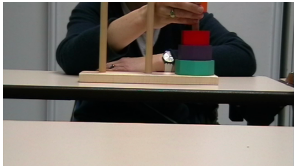
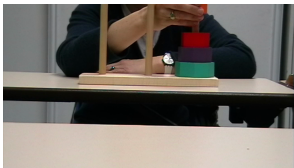
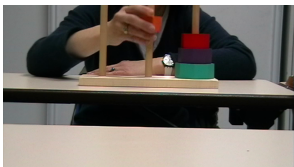
B.2.2 Sequential data for the solution of TOH

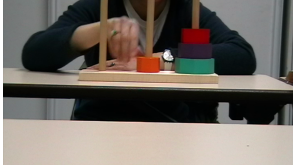
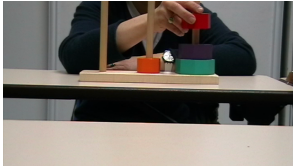
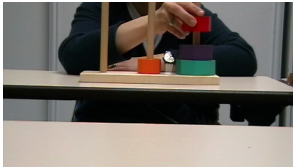
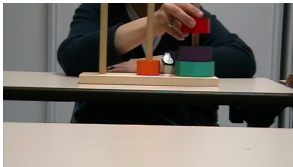
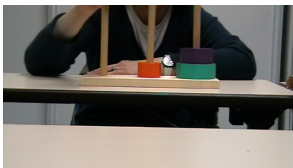
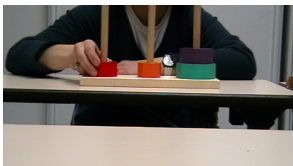
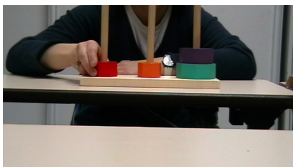
Num.	Sequential data for the solution of TOH
1	G0 G1 G3 G2 G1 G6 G4 G1 G3 G2 G5 G6 G2 G1 G3 G2.
2	G0 G1 G3 G2 G1 G6 G4 G1 G3 G2 G5 G6 G2 G1 G3 G2.
3	G0 G3 G1 G4 G3 G2 G4 G5 G3 G5 G6 G4 G3 G1 G4 G3 G5 G2 G3 G4 G6 G5 G2 G1 G3 G2.
4	G0 G1 G3 G2 G1 G4 G5 G4 G1 G3 G2 G5 G4 G3 G5 G1 G6 G5 G2 G1 G3 G2.
5	G0 G3 G1 G4 G3 G5 G3 G4 G6 G2 G5 G4 G2 G1 G4 G3 G6 G2 G5 G1 G6 G2 G3 G1 G6 G4 G1 G3 G5 G2 G3 G5 G4 G2 G1 G6 G4 G2 G3 G5 G4 G6 G2 G1 G4 G3 G5 G2 G3.
6	G0 G1 G3 G6 G2 G1 G6 G2 G3 G1 G4 G6 G2 G1 G4 G3 G5 G3 G5 G6 G2 G1 G3 G2.
7	G0 G3 G1 G4 G3 G2 G6 G2 G3 G1 G6 G4 G1 G6 G2 G5 G6 G2 G3 G1 G4 G3 G5 G2 G3.
8	G0 G3 G1 G4 G3 G6 G2 G4 G3 G2 G5 G4 G3 G2 G1 G4 G5 G4 G3 G5 G6 G3 G1 G4 G6 G2 G5 G6 G2 G3 G4 G3 G6 G2 G4 G3 G2 G1 G4 G6 G5 G2 G1 G3 G2.
9	G0 G3 G4 G2 G1 G4 G3 G5 G2 G3 G1 G6 G4 G3 G5 G1 G4 G6 G2 G5 G6 G2 G1 G3 G2 G1 G4 G6 G5 G2 G1 G3 G2.
10	G0 G3 G1 G4 G3 G5 G2 G3 G1 G6 G4 G1 G6 G5 G2 G1 G6 G5 G2 G1 G3 G2 G1 G4 G6 G2 G6 G2 G1 G3 G2.

Num.	Sequential data for the solution of TOH
11	G0 G3 G1 G4 G3 G2 G5 G1 G4 G5 G2 G3 G1 G4 G6 G5 G4 G2 G1 G2 G1 G4 G6 G5 G2 G1 G6 G5 G2 G3 G1 G4 G3 G5 G2 G3.
12	G0 G1 G3 G2 G1 G6 G3 G6 G4 G3 G5 G4 G2 G1 G4 G3 G2 G5 G4 G5 G2 G1 G3 G2.
13	G0 G1 G3 G2 G1 G6 G4 G1 G3 G2 G5 G6 G2 G1 G3 G2.
14	G0 G1 G3 G2 G1 G4 G6 G5 G2 G3 G1 G6 G2 G1 G2 G1 G6 G1 G5 G4 G1 G6 G5 G3 G5 G6 G2 G1 G2 G1 G4 G3 G5 G2 G1.
15	G0 G3 G1 G4 G3 G5 G2 G3 G1 G6 G4 G1 G6 G2 G5 G6 G2 G1 G3 G6 G2 G1 G4 G3 G5 G2 G3.
16	G0 G3 G1 G6 G2 G1 G2 G1 G6 G4 G1 G3 G5 G3 G5 G4 G3 G5 G4 G1 G5 G3 G5 G6 G2 G1 G3 G2.
17	G0 G1 G3 G2 G1 G4 G6 G2 G6 G2 G1 G3 G2 G1 G4 G6 G5 G4 G3 G1 G6 G3 G4 G5 G2 G3 G5 G4 G6 G5 G2 G1 G2 G1 G4 G3 G5 G2 G3.

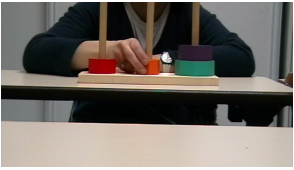
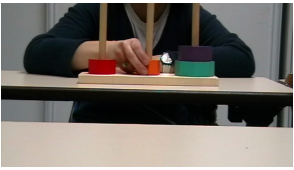
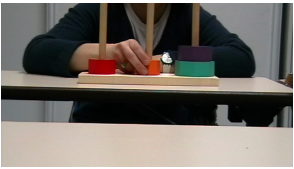
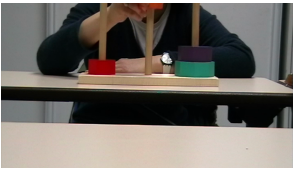
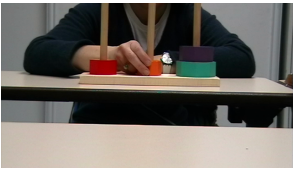
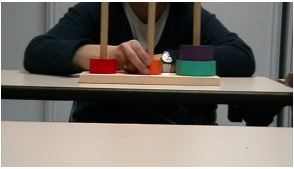
B.3 Sequential images and coding

B.3.1 Sample Coding for Convolutional Neural Network

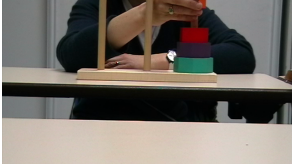
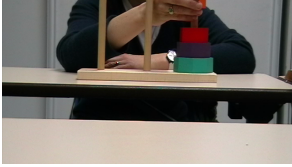
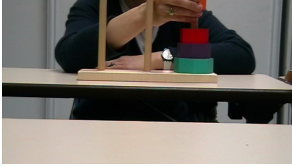
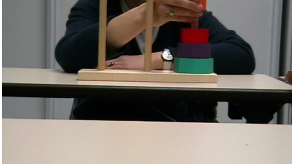
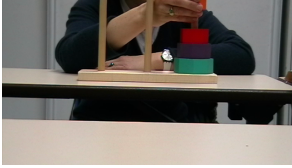
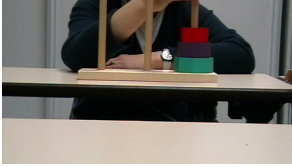
Filename	Image	Class
sujet1 000001.png		['RD4', 'TD1']
sujet1 000002.png		['RD4', 'TD1']
sujet1 000003.png		['RD4', 'TD1']
sujet1 000004.png		['RD4', 'TD1']
sujet1 000005.png		['RD4', 'TD1']
sujet1 000006.png		['RD3']
sujet1 000007.png		['MD1', 'RD3', 'PD1']

Filename	Image	Class
sujet1 000008.png		['MD1', 'RD3']
sujet1 000009.png		['MD1', 'RD3', 'TD2']
sujet1 000010.png		['MD1', 'RD3', 'TD2']
sujet1 000011.png		['MD1', 'RD3', 'TD2']
sujet1 000012.png		['MD1', 'RD2']
sujet1 000013.png		['LD1', 'MD1', 'RD2', 'PD2']
sujet1 000014.png		['LD1', 'MD1', 'RD2', 'PD2']

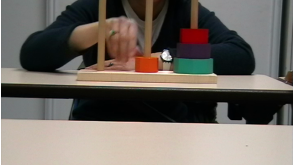
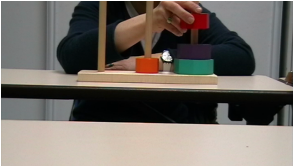
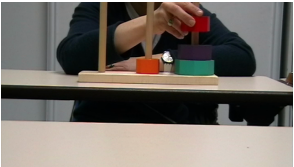
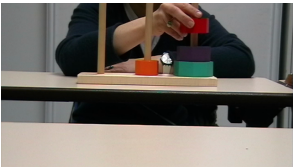
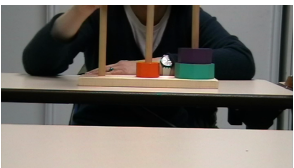
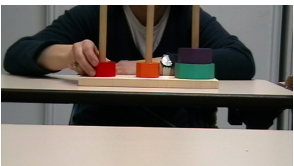
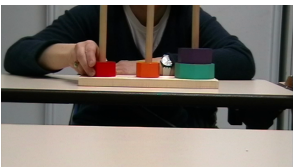
B.3 Sequential images and coding

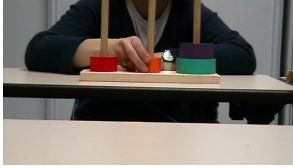
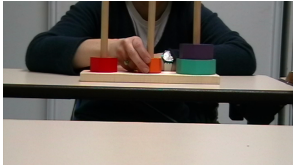
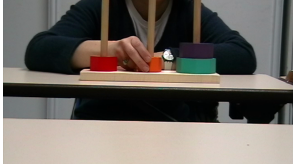
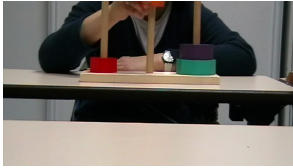
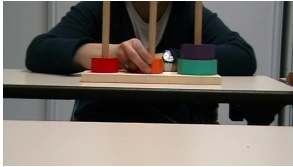
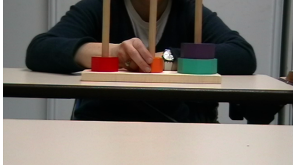
Filename	Image	Class
sujet1 000015.png		['LD1', 'MD1', 'RD2', 'TD1']
sujet1 000016.png		['LD1', 'MD1', 'RD2', 'TD1']
sujet1 000017.png		['LD1', 'MD1', 'RD2', 'TD1']
sujet1 000018.png		['LD1', 'MD1', 'RD2', 'TD1']
sujet1 000019.png		['LD1', 'MD1', 'RD2', 'TD1']
sujet1 000020.png		['LD1', 'MD1', 'RD2', 'TD1']

B.3.2 Sample Coding for Natural Language Processing

Filename	Image	Sentence
sujet1 000001.png		There are three disks on the right. I hold disk1
sujet1 000002.png		There are three disks on the right. I hold disk1
sujet1 000003.png		There are three disks on the right. I hold disk1
sujet1 000004.png		There are three disks on the right. I hold disk1
sujet1 000005.png		There are three disks on the right. I take disk1
sujet1 000006.png		There are three disks on the right

B.3 Sequential images and coding

Filename	Image	Sentence
sujet1 000008.png		There are three disks on the right. One disk is in the middle.
sujet1 000009.png		There are three disks on the right. One disk is in the middle. I take disk2
sujet1 000010.png		There are three disks on the right. One disk is in the middle. I take disk2
sujet1 000011.png		There are three disks on the right. One disk is in the middle. I take disk2
sujet1 000012.png		There are two disks on the right. One disk is in the middle.
sujet1 000013.png		There are two disks on the right. One disk is in the middle. One disk is on the left. I put down disk2
sujet1 000014.png		There are two disks on the right. One disk is in the middle. One disk is on the left. I put down disk2

Filename	Image	Sentence
sujet1 000015.png		There are two disks on the right. One disk is in the middle. One disk is on the left. I take disk1
sujet1 000016.png		There are two disks on the right. One disk is in the middle. One disk is on the left. I take disk1
sujet1 000017.png		There are two disks on the right. One disk is in the middle. One disk is on the left. I take disk1
sujet1 000018.png		There are two disks on the right. One disk is in the middle. One disk is on the left. I take disk1
sujet1 000019.png		There are two disks on the right. One disk is in the middle. One disk is on the left. I take disk1
sujet1 000020.png		There are two disks on the right. One disk is in the middle. One disk is on the left. I take disk1

Appendix C

Convolutional Neural Network - Tower of Hanoi

C.1 Image processing - kernel

$$\begin{pmatrix} 0 & -0.5 & 0 \\ -0.5 & 3 & -0.5 \\ 0 & -0.5 & 0 \end{pmatrix}$$

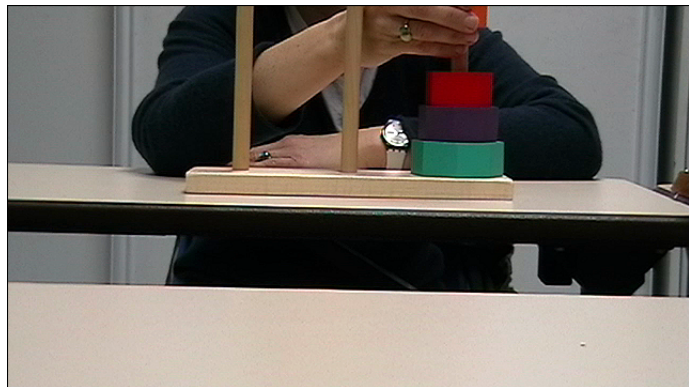


Fig. C.1 Sharpen operation

$$\begin{pmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

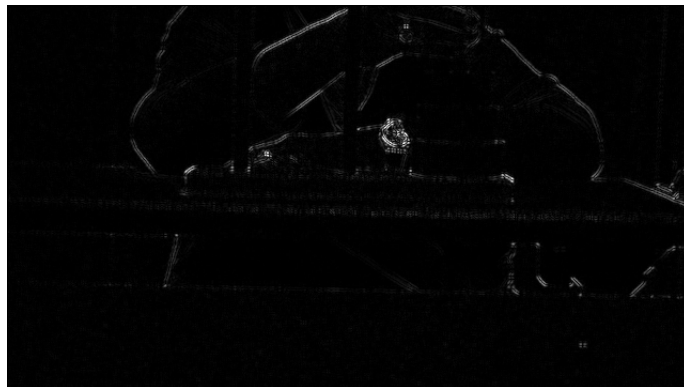


Fig. C.2 Edge detection operation 1

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

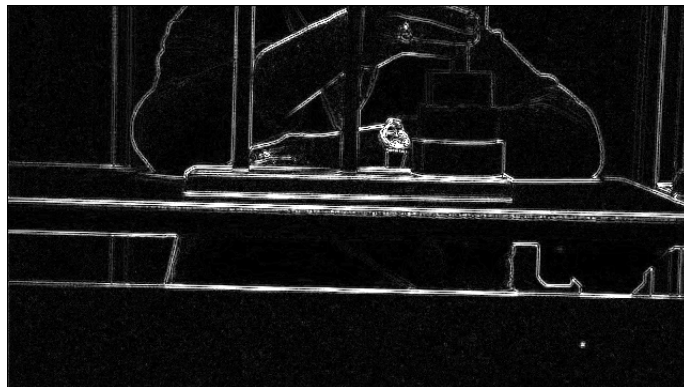


Fig. C.3 Edge detection operation 2

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

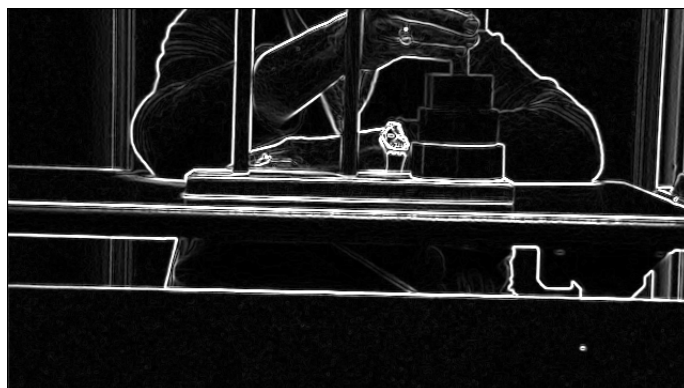


Fig. C.4 Edge detection operation 3

$$\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

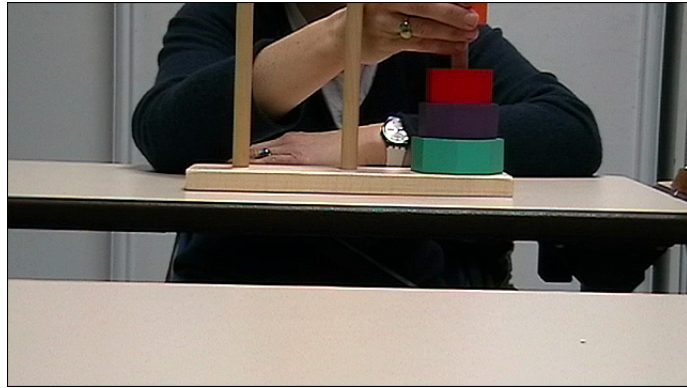


Fig. C.5 Box blur operation

$$\frac{1}{256} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

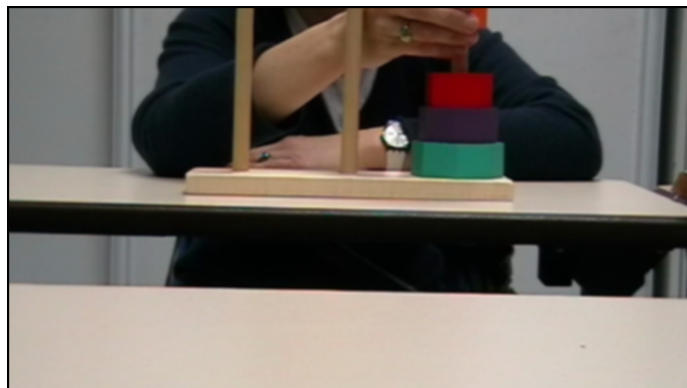


Fig. C.6 Gaussian blur 5×5

