Problem Solving using Recurrent Neural Network based on the Effects of Gestures

Sanghun Bang and Charles Tijus
Laboratoire Cognitions Humaine et Artificielle (CHArt), University Paris 8, 2 rue de la Liberté, 93526 Saint-Denis, France

Keywords: Problem Solving, Neural Network, Recurrent Neural Network, Reinforcement Learning, Cognition, Embodied Cognition, Tower of Hanoi.

Abstract: 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 dicks 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 an approach for change of state sequences training, which combines Recurrent Neural Network and Reinforcement Learning methods.

1 INTRODUCTION

The theory of embodied cognition (Varela and Thompson, 1991; Barsalou, 2010) suggests that our body influences our thinking. Even an approximate and imprecise body motion can affect the way that we think about. 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, 2008), gestures are used as grounding for a mapping between thinking and real objects in the world, in order for the easy catching of meanings.

To analyze the effect of gestures on problem solving cognitive processes (learning, memorizing, planning, and decision-making), participants were asked to solve the puzzle of Tower of Hanoi (TOH). Classical puzzle-like problem, such as Tower of Hanoi puzzle and missionaries-cannibals received some attention 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., 1993).

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 carry 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 Recurrent Neural Network (RNN). State-of-the-art have recently demonstrated performance' RNN models across a variety of tasks in domains such as text (Sutskever et al., 2011), motion capture data (Sutskever et al., 2009), and music (Eck and Di Studi Sull Intelligenza, 2002). In particular, RNNs can be trained for sequence activation while processing real data sequences. Therefore, we modeled the Tower of Hanoi solving processes with the help of RNN method.

Bang, S. and Tijus, C.
Problem Solving using Recurrent Neural Network based on the Effects of Gestures.
DOI: 10.5220/0006931802110216
In *Proceedings of the 10th International Joint Conference on Computational Intelligence (IJCCI 2018)*, pages 211-216
ISBN: 978-989-758-327-8

The minimum number of moves needed to solve TOH with n disks is denoted by $2^n - 1$. In order to find what would be a participant minimum number of moves, we also propose a novel approach for sequence training, which combines Recurrent Neural Network and Reinforcement Learning (RL) method.

In RL model, the method is to evaluate and select the generated moves by comparing their results with the goal target state. This is accomplished by a reward mechanism where the favorable moves obtain the higher rewards and the unnecessary moves or repeated moves don't. In this article, the implementation of reinforcement learning is based on the Q-learning method

2 BACKGROUND AND RELATED WORK

2.1 Tower of Hanoi

The french mathematician Edouard Lucas introduced the Tower of Hanoi (TOH) puzzle in 1883 (Chan, 2007). Figure 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 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.

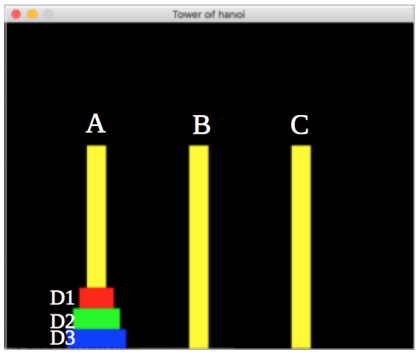

Figure 1: Three disks in Tower of Hanoi puzzle(initial state).

2.2 Recurrent Neural Network

Because the TOH solving process is a sequential process. In this work, we have used a simple recurrent neural network (RNN) which is based on Elman network (Elman, 1990). This network is made up of 3 layers : $\mathbf{x} = (x_1,...,x_T)$ a input sequence , output vector sequence denoted as $\mathbf{y} = (y_1,...,y_T)$, and $\mathbf{h} = (h_1,...,h_T)$ is the hidden vector sequence.(See Figure 2)

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

$$y_t = g(h_t V) \quad (2)$$

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.

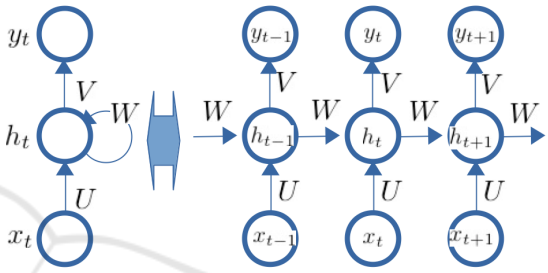

Figure 2: Simple Recurrent Neural Network Architecture.

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

$$\Pr(\mathbf{x}) = \prod_{t=1}^{T} \Pr(x_{t+1}|y_t) \quad (3)$$

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

$$L(\mathbf{x}) = \sum_{t=1}^{T} \log \Pr(x_{t+1}|y_t) \quad (4)$$

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 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 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 rewards. We can calculate the future rewards $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$,

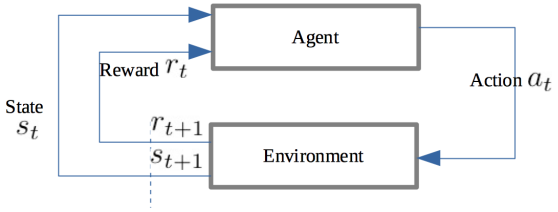

Figure 3: Model(Reinforcement Learning).

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:

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha\left(r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)\right) \quad (5)$$

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

3 MODEL

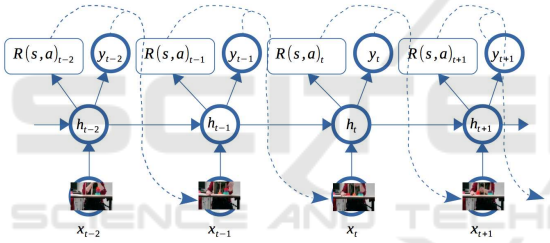

Figure 4: Solving of Tower of Hanoi puzzle with RL and RNN: x_t is the observation, h_t is the hidden state for RNN, y_t is the predicted observation for time $t+1$, $R(s,a)_t$ is the predicted reward.

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's take an example. We can have two possible next actions $\{G1, G3\}$ at time 0 (See Figure 5) according to the rules of TOH (Table 1).

Table 1: Rewards and possible actions at time 0.

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

Table 2 is achieved by looping an output of the network at time 0 with the input of the network. Therefore, base on the table 1 and 2, we choose the next action $G3$ (From Peg A to Peg C).

Table 2: Probabilities and possible actions at time 0.

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

Furthermore, table 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.

Table 3: Negative Reward and possible action.

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

Table 4: Probabilities and possible actions.

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

Based on the possible actions, we can predict possible action $G3$ in table 4. But in order to maximize its cumulative reward (See table 3), our RNN+RL model takes an action $G2$ instead of $G3$.

4 EXPERIMENTS

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 called this action as $G1$. We can encode all possible actions in the same way (See Figure 5). For example, Figure 6 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 $\{G0, G1, G3, G2\}$

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

Table 5: Solution acquired by a participant.

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

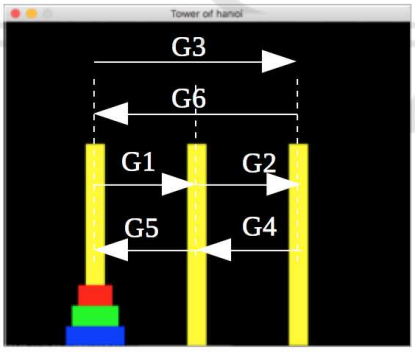

Figure 5: Coding - Move a disk from the left peg to the middle peg ($G1$). Move a disk from the middle peg to the right peg ($G2$). Move a disk from the left peg to the right peg ($G3$). Move a disk from the right peg to the middle peg ($G4$). Move a disk from the middle peg to the left peg ($G5$). Move a disk from the right peg to the left peg ($G6$).

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 6 shows results of Tower of Hanoi for all participants.

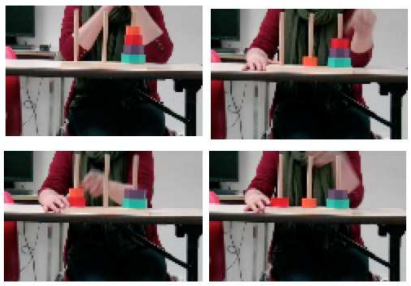

Figure 6: Clockwise from top left: Initial state($G0$), move a disk from peg A to peg B($G1$), move a disk from peg A to peg C($G3$) and move a disk from peg B to peg C ($G2$). We encode the sequence of these movements:$\{G0, G1, G3, G2\}$.

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 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 trough 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. $[r = .496, p < 0.0072]$

Table 6: 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		

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. 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 ran-

dom 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 RNN'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 RNN 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 RNN, 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 (5) and repeated the process until a terminal state was reached.

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.

Compared to the experimental results in table 6, this RNN model shows good performance improvement. Nevertheless, this result is not the fastest solution. According to the table 6, the first participant and the thirteenth participant find the fastest solution. On the other hand, from figure 8, we can see that this combined model(RNN+RL) improves the performance compared with RNN model.

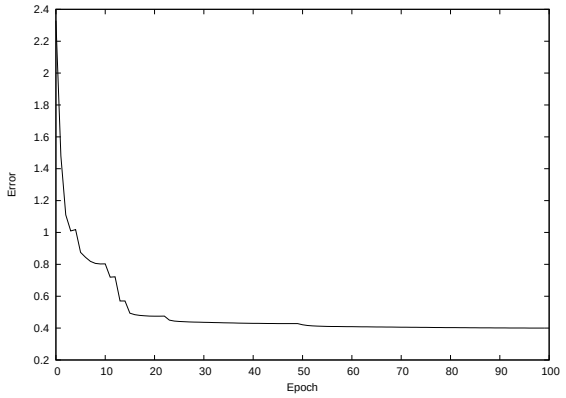

Figure 7: RNN train error.

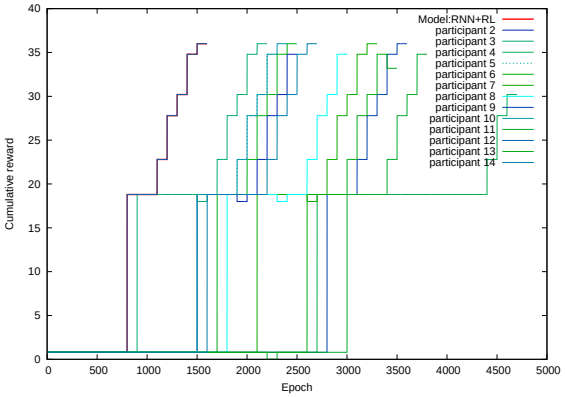

Figure 8: Cumulative reward graph for RNN+RL model .

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 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.

As is well known, the simplest RNN model has a vanishing gradient problem. That's why, we will implement the gated activation functions, such as the long short-term Memory(LSTM) (Hochreiter and Schmidhuber, 1997) and the gated recurrent unit (Cho et al., 2014) to overcome the limitations of our model.

ACKNOWLEDGEMENTS

We would like to thank Mathilde Malgrange and the participants for participating in the experiment. We also appreciate Maria Jose Escalona and Francisco

José DOMINGUEZ MAYO for our academic exchanges and cooperation.

REFERENCES

Barsalou, L. W. (2010). Grounded Cognition: Past, Present, and Future. *Topics in Cognitive Science*, 2(4):716–724.

Chan, T.-H. (2007). A statistical analysis of the towers of hanoi problem. *International Journal of Computer Mathematics*, 28(1-4):57–65.

Cho, K., van Merriënboer, B., Gülçehre, Ç., 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, Doha, Qatar. Association for Computational Linguistics.

Eck, D. and Di Studi Sull Intelligenza, J. S. I. D. M. (2002). A first look at music composition using lstm recurrent neural networks. *people.idsia.ch*.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Nathan, M. J. (2008). An embodied cognition perspective on symbols, gesture, and grounding instruction. *Symbols*.

RICHARD, J.-F., Poitrenaud, S., and Tijus, C. (1993). Problem-Solving Restructuration: Elimination of Implicit Constraints. *Cognitive Science*, 17(4):497–529.

Sutskever, I., Hinton, G. E., and Information, G. T. (2009). The recurrent temporal restricted boltzmann machine. *papers.nips.cc*.

Sutskever, I., Martens, J., and Conference, G. H. (2011). Generating text with recurrent neural networks.

Varela, F. and Thompson, E. (1991). *The embodied mind: Cognitive science and human experience*. Cambridge.