Yusuke Kominami

Bachelor 3rd, Kyoto University.

Hitachi Kyoto University Laboratory

November 20, 2018

Machine Learning

1. My experience of Programming, Planning or Designing

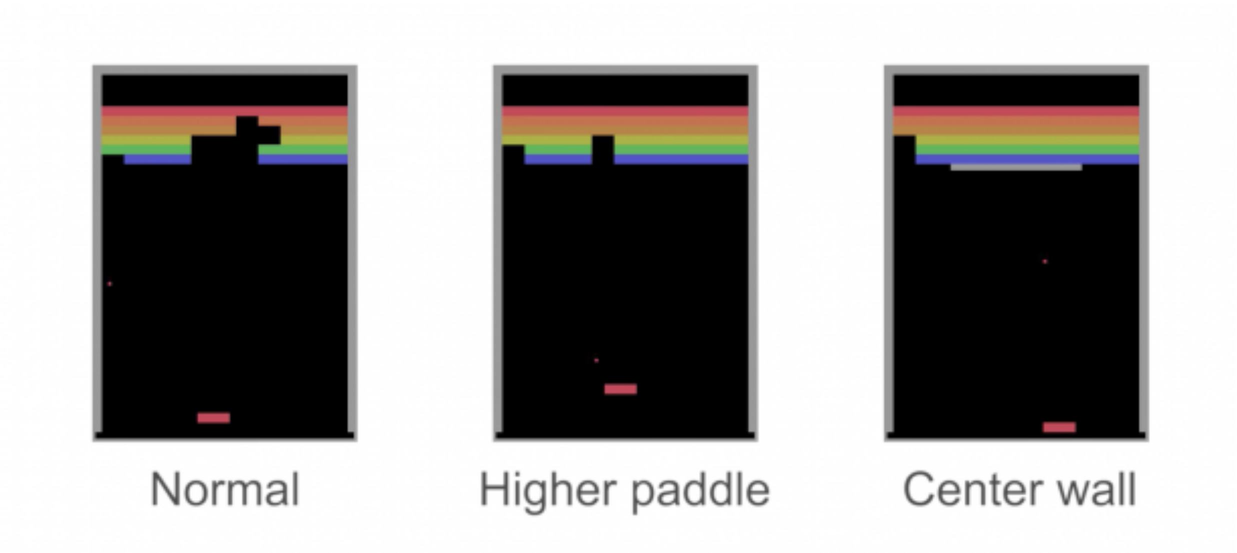
Time	Team	Contents
May, 2017 - June, 2017	Rist, Inc.	• Construct image recognition model • Use Python in constructing model, and PyTorch as framework • Use C++ language and OpenCV to preprocess images • This image recognition model is to check car images, recognize the damaged areas and annotate them.
April, 2018 - July, 2018	University course of Data Assimilation	• Make team of 3 persons and make the same data assimilation model • Make Kalman Filter and ensemble Kalman Filter • Analyze data and explain it. • Use Python and Jupyter notebook to share progress easily • Play a role of the team leader and manager.
June, 2017 - July, 2018	Hitachi, Ltd. Hitachi Kyoto University lab.	• Research on deep learning, mainly deep reinforcement learning. • Read papers • Construct proposed algorithms • Discuss and pursue another better idea • Make presentation

2. My experience in Hitachi : Research on Reinforcement Learning

The purpose of our lab was to make research on evolutionary computation and artificial intelligence. There were some student researchers, and each was assigned a main area. And my area was reinforcement learning.

There were many challenges in deep reinforcement learning, and I focused on one challenge, generalization.

What is generalization ? For example, Deep Q Network (shortly, DQN) is the first successful reinforcement learning model proposed in 2013, and this DQN has outperformed human in Breakout, Pong or maze game. This seemed to be the birth of Artificial Intelligence. However, DQN had some big problems. One of them was “catastrophic forgetting”. In Breakout, Catastrophic forgetting happens when the environment changes (for example, higher paddle or center wall). And then DQN does not score like before learning.



Why does such a thing happens ? This is an easy question; DQN does not think logically but DQN acts just reflexively. This is the completely different point from human. Then I thought model which had acquired generalization should be able to adapt the change of the environment. This is my notion about the challenge of reinforcement learning.

My approach to this challenge was to make the model predict future environment. When humans, for example, play a RPG game, they will consider which action will be the best to achieve the goal. Like this, I thought that reinforcement learning model would be able to predict future environment by using neural network.

Reinforcement learning is based on Bellman equation.

$$Q^{\pi}(s_t, a_t) = \sum_{s_{t+1} \in \mathcal{S}} P(s_{t+1} \mid s_t, a_t)(r_t + \gamma \sum_{a_{t+1} \in \mathcal{A}} \pi(a_{t+1} \mid s_{t+1}) Q^{\pi}(s_{t+1}, a_{t+1}))$$

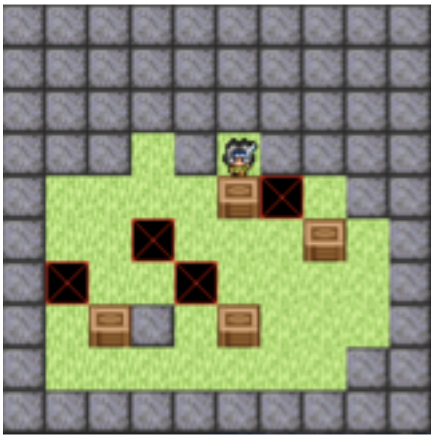
And formula can be introduced by the definition of Marcov Chain. According to this formula means the value of current state is based on only next value. So, to predict far future, we have to design the model to consider more future state in evaluating current state and action value. Then we can write down new formula (in easy style):

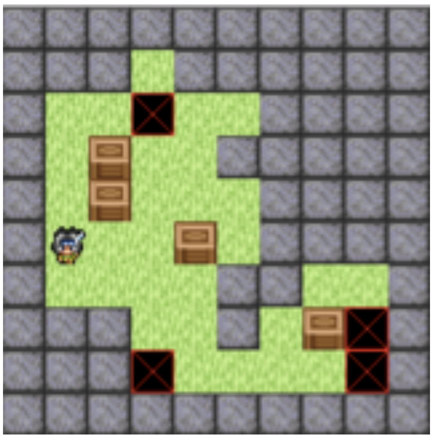
$$Q(s_t, a_t) = \sum_{n=0}^{k-1} \gamma^n r_{t+n} + \gamma^k Q(s_{t+k}, a_{t+k})$$

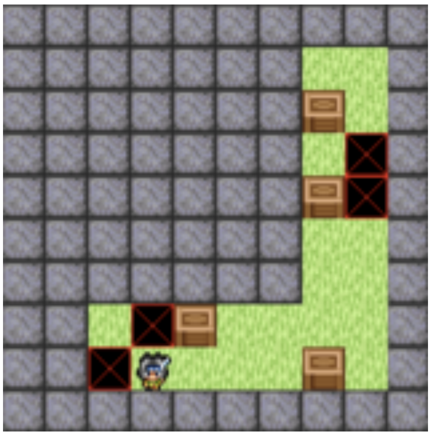
Finally my idea could be wrote down. Then I applied this to neural network.

In my experiment, I made a new environment: Sokoban. Sokoban is a popular video game in Japan. I made this with PyGame. Then, inputs to neural network were action and state images, and output was future state.

The result was successful. See below images.







These images are the predicted states by the model. And this reinforcement learning outperformed DQN in Sokoban.

My hypothesis for generalization of reinforcement learning resulted in success.