Transfer Learning Approach for Fast Convergence of Deep Q Networks in Game Pong

Baomin Shao
Department of Computer Science and Technology, Shandong University of Technology
No. 266, Xincun Xi Road, Zibo, Shandong, China

Xue Jiang
Department of Computer Science and Technology, Shandong University of Technology
No. 266, Xincun Xi Road, Zibo, Shandong, China

Qiuling Li
Department of Computer Science and Technology, Shandong University of Technology
No. 266, Xincun Xi Road, Zibo, Shandong, China

ABSTRACT
By simulating the psychological and neurological system, deep reinforcement learning method has been playing an important role in the development and application of artificial intelligence with the help of the powerful feature representation capability of deep neural networks. The deep Q network which improves traditional RL methods by breaking out the learning mechanism of value function approximation and policy search based on shallow structure, has the capabilities of hierarchical feature extraction and accurate Q value approximation in various high-dimensional sensing environments.

In this paper, DQN was adapted into Game Pong playing, however, it was found that by adjusting hyperparameters (network architecture, exploration, learning rate), the Q-values could not converge easily. The lacking convergence of the Q-loss might be the limiting factor for better game playing results. A transfer learning approach has been adopted for fast convergence of DQN in game Pong, several measure standards was used as rewards to train DQN, experiments showed that this approach can get fast convergence of DQN training, and DQN network play good performance on game Pong.

General Terms
Deep Reinforce Learning, Digital Image Processing.

Keywords
DQN; Transfer Learning, Game Pong, Image Evaluation

1. INTRODUCTION
Pong is one of the earliest arcade video games. It is a "tennis like" game featuring simple two-dimensional graphics which was originally manufactured by Atari. There are two paddles and a ball on the stage, the goal is to move two paddles to keep a ball in play to defeat the opponent by being the first one to gain certain points, a player gets a point once the opponent misses a ball. There are three actions that each agent can take: move up, move down, stay at the same place. The game ends when one team reach the maximum score. It has been studied in a variety of contexts as an interesting RL domain. In pong game, agents can easily last thousands of time steps, which is far more time consuming compared with other domains [1]. On the other side, its observations are also complex, containing the comprehension of players' score and side walls, which information need to be extracted from raw game scene image.

By simulating the psychological and neurological system, deep reinforcement learning method has been playing an important role in the development and application of artificial intelligence with the help of the powerful feature representation capability of deep neural network[2-4]. In recent years, the progress reports on RL agents have had tremendous influence in the field of machine learning and artificial intelligence. In this area the deep Q network (DQN) proposed by Google DeepMind[5], represents a major technical step forward in the quest for general AI, it demonstrated a general-purpose agent that is able to continually adapt its behavior without any human intervention. Inspired by the features and advantages of DQN, different background applications have been developed: using DQN to character segmentation of license plate images[6], DQN for financial signal representation and trading[7], using DQN in text-based games for language understanding[8]. Many studies showed that game playing agents trained by DQN was able to surpass the overall performance of a professional human reference player, that shows that DQN has the capability of scene value evaluation in various high-dimensional sensing environments. Besides the application development combined with different research areas, many researchers dedicated to optimize the structure of DQN for a better and more stable performance, and numerous algorithms were proposed[9-11], however, The powerful feature representation and function approximation of deep neural networks often mean a high cost of computation, it's found that in many applications the Q-values can't converge easily and the algorithm consumes much more time for training. Furthermore, the agent in game normally needs to constantly interact with the environment to make decisions. Thus, the process takes even longer time.

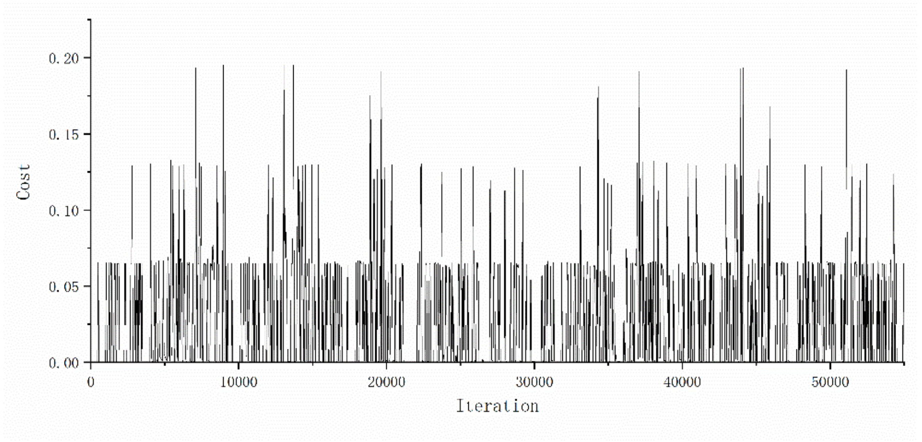

Figure 1 The plot shows cost function changes per iteration, there is no sign of convergence in the first 50000 iterations

Although transfer RL is not a new research area, few studies focus on deep RL algorithms, such as transfer DQN. Compared with traditional back propagation networks, radial basis function networks, and MDP, a deep neural network has more layers, more parameters, and longer training time. To accelerate the process of learning, it is necessary to study the method of transfer learning in DQN. If the errors from action estimation cannot be theoretically eliminated, the transfer process of DQN may accumulate these errors to cause the poor performance of the algorithm. Therefore, a double DQN is adopted in this paper to remove the influence of action overestimation.

The rest of this paper is organized as follows: Section II introduces several background techniques of DQN in our experiment and out model and algorithm in detail. In Section III, experiment results and discussion were given, some conclusion were made in this section.

2. PROPOSED METHOD

2.1 DQN

Deep reinforce learning method considers tasks as an agent training procedure interacts with certain environment ε , which is composed with a sequence of actions, observations and rewards. The agent selects an action a_t from the action set of game environment, $\mathrm{A}=\{1...\mathrm{N}\}$. The selected action was used to play the game and get a new observation. Normally the observation is represented as an image that is a vector of raw pixel value. The agent also receives a reward value r_t to record the human understanding of the observation. In Pong game, the reward can only be received when a ball was missed, maybe thousands of time-steps have elapsed. Because the agent can only get the observation of current stage, sometimes some frames before current stage can be buffered, it's less realistic to totally understand the situation of current observation and give it an evaluation value. So a sequence of actions $x_1, a_1, x_2, a_2, ..., a_{t-1}, x_t$ and observations was considered to learn dependencies upon this sequence and learn game playing strategy. The goal of the agent is to select actions by maximize reward of next frame, and interact with the game, repeat this process and get a good final reward. An optimal action-value function $Q^*(s,a)$ was defined as the maximum expected value after one moving strategy, $Q^*(s,a) = \max_\pi \mathrm{E}[R_t | s_t = s, a_t = a, \pi]$, where π is a policy distributions over actions. The basic idea of DQN is maximizing the expected value of $r + \gamma Q^*(s', a')$, future rewards are assumed to discount by this factor γ . $Q^*(s,a) = E_{s' \sim \varepsilon}[r + \gamma \max_{a'} Q^*(s', a') | s, a]$. DQN uses a non-linear neural network function as Q evaluation network, this network was trained by minimizing the loss function $L_i(\theta_i)$ in each iteration i ,

$$L_i(\theta_i) = E_{s,a \sim \rho(.)}[(y_i - Q(s,a;\theta_i))^2]$$

Where y_i is the target for iteration i and $\rho(s,a)$ is a probability distribution over sequence and actions. it is often computationally expedient to optimize the loss function by stochastic gradient descent., The parameters from the previous iteration are held fixed when optimizing the loss function.

2.2 Transfer Learning

If knowledge from related tasks, such as weights, value functions, and policies, can be transferred to target tasks, the learning difficulties can be sharply reduced. Many researchers have tried to resolve this problem by three approaches: model transfer, sample transfer, and feature transfer. Model transfer is the simplest and most direct approach. Fachantidis et al [12]. transferred the state transition and reward function models of source tasks to target tasks. In this way, they realize the transfer from 2-D to 3-D in the mountain car task. However, the algorithm performance will be affected by the model dependence between related tasks. Compared with model transfer, sample transfer is more general. It is reported in[13] that even if there are obvious model differences between source and target tasks, the sample transfer from source tasks can still help the target task reduce requirements for samples.

When directly using pong score as the loss function computing parameters, as is shown in Figure 1, the cost function is difficult to convergence, this is because only the left paddle touches the ball score value will change, and in other time changing frames, the score doesn't change, this leads to less training data, it is difficult to achieve the purpose reinforcement learning training. The experimental results also prove that the operation of the paddle is essentially moving randomly, the moving has no relationship with better score. In view of this, the parameters that can measure the change of the environment in the time sequence were introduced, the following different distances are used to measure the value of different frames, and a new reward function based on these distances was built.

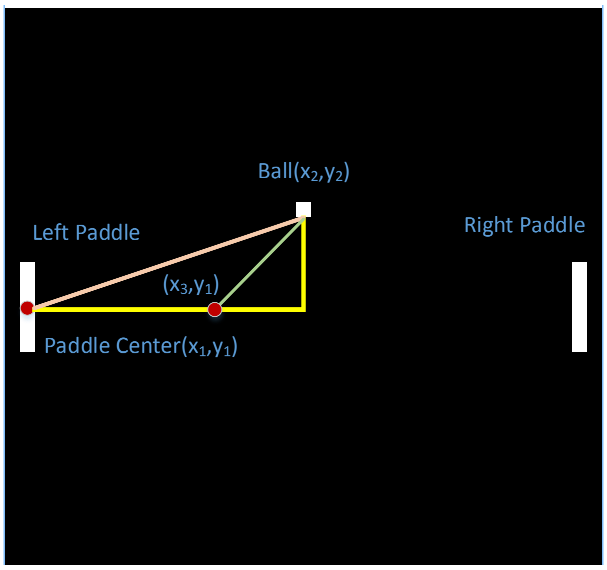

Figure 2 Pong environment and different evaluation methods, these methods use different distance between position of ball and the center point of left paddle that agent controls

The most commonly used distance is the Euclidean distance. In an image space, the Euclidean distance is the straight-line distance between two pixels. The distance between two points p and q is defined as the square root of the sum of the squares of the differences between the corresponding coordinates of the points. The two-dimensional Euclidean geometry, the Euclidean distance between two points paddle center (x_1, y_1) and ball (x_2, y_2) is defined as:

$$d_E = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

This distance is shown as an orange line in Figure 2.

The manhattan distance between two points in an image space is based on a strictly horizontal and/or vertical path as opposed to the diagonal. It's the simple sum of the horizontal and vertical components, it is also called the L1 distance of two points, then the manhattan distance between two points paddle center (x_1, y_1) and ball (x_2, y_2) is given by

$$d_{MH} = |x_1 - x_2| + |y_1 - y_2|$$

The manhattan distance is shown as the yellow line in Figure 2.

The chessboard distance metric measures the path between the pixels based on an 8-connected neighborhood. Pixels whose edges or corners touch are 1 unit apart. The chessboard distance is a metric defined on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension. In two dimensions, i.e. image space of Pong frame, their chessboard distance is $d_{chess} = \max(|x_1 - x_2|, |y_1 - y_2|)$, in Figure 2 it corresponds with the bottom yellow line. If the maximum function in chessboard distance is changed into minimum function, then city block distance can be calculated.

3. EXPERIMENTS AND RESULTS

3.1 Deep Neural Network Architecture

The DQN network model had three convolutional layers followed by three fully connected layers. In the input layer, there was one channel getting the sequential frames. One frame was an 84x84 vector. In the first convolutional layer, there were 32 filters. The filter size was 3x3 with stride 4. In the second convolutional layer, there were 32 filters with filter size 3x3 with stride 2. The third convolutional layer had 32 filters with filter size 3x3 with stride 1. After three convolutional layers, there were one flatten layer and three fully connected layers. The output dimension was equal to the number of actions in the Game Pong DQN agent. In Game Pong, the total number of actions for each character was 3. These actions were the moving directions of left paddle. (Up, Down, and Hold-on). As is shown in Figure 3. The target net had the same structure with the evaluation net, in the training period, the parameters in the target net were fixed, and the

parameters in the evaluation net were updated using gradient descent method, whose loss function could be changed with different evaluation standards. After numerous iteration of training, the parameters in the evaluation net were copied to target net.

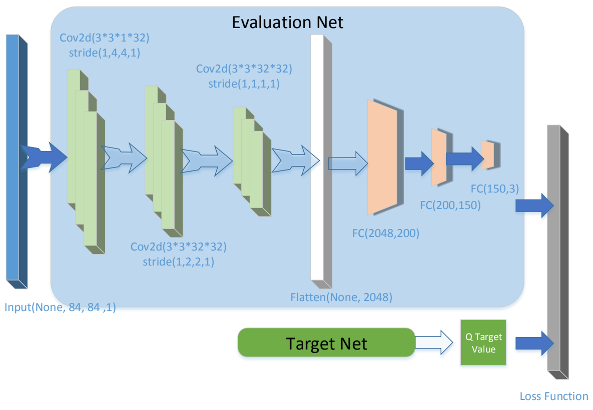

Figure 3 The structure of DQN for Game Pong

Transfer learning significantly improved the performance of our network. When the four measuring methods were adopted in the network, its performance was improved more quickly than the network trained from scratch on the game score. As is shown in Figure 4, the above four distance measurement of scenes were respectively adopted as the parameters of DQN training. Compared with the initial score parameters, all of them could converge better. Moreover, the experimental scenario also showed that a better play could be achieved. Due to different parameters, the city block distance converges the fastest, and after 10,000 iterations the game could achieve a better playing scene. The first manhattan distance is more stable, it almost has the same performance as the agent using priori knowledge. When the other two distances were adopted, paddles could hit the ball when moving, but when the distance was far away, there would be shock, which might be because the direction of movement could not be determined during training.

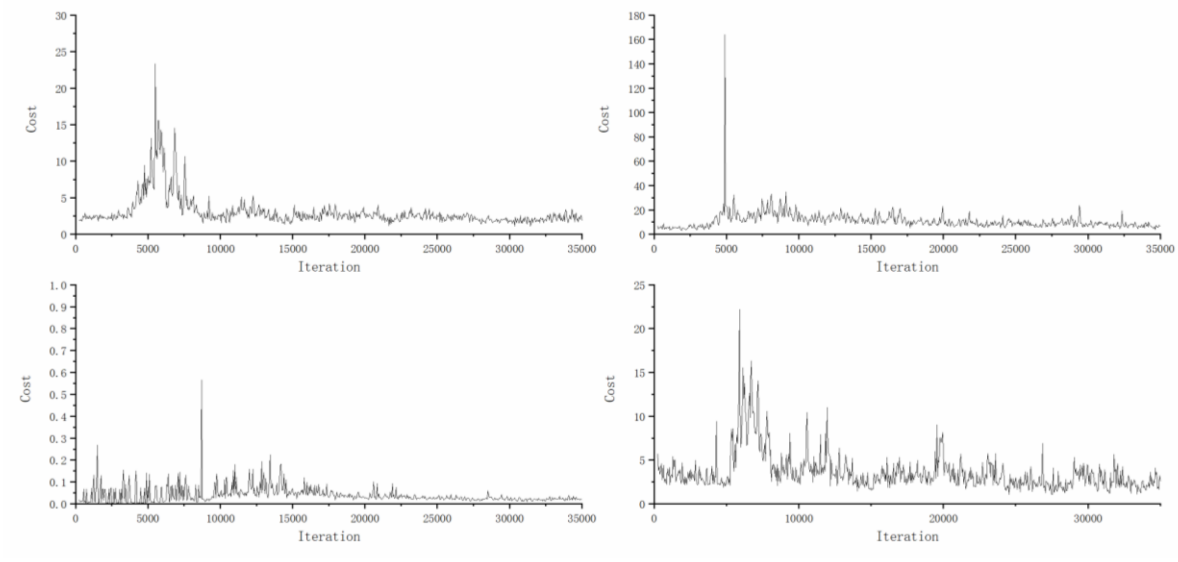

Figure 4 Transfer learning performance after adapting four kinds of evaluation distances, the four methods are manhattan distance, city block distance, Euclidean distance and chessboard distance

The reason for this improved performance is that the network likely learned many relevant features for Game Pong during its training, each action could give the training effective metric, the rewards were not smooth and strongly encouraged local optima. After training, the agent switched to the game score it only had to reweight the network parameter, and it converged very fast.

In conclusion, the experiment results indicated that transfer learning was a viable strategy for training an AI for Game Pong, the introduction of additional parameters based on the scene information of the game can improve the training process of DQN and achieve a relatively stable execution result. However, the game is relatively simple and the parameter selection is easy to achieve convergence. If there are more parameters in the complex scene, more analysis and comparison are required.

4. REFERENCES

[1] M. G. Bellmare, Y. Naddaf, J. Veness and M. Bowling, The Arcade learning environment: an evaluation platform for general agents, Journal of Artificial Intelligence Research, 47, pp.253–279, 2013

[2] D. Zhao and Y. Zhu, MEC-a near-optimal online reinforcement learning algorithm for continuous deterministic systems, IEEE Trans. Neural Netw. Learn. Sys., vol. 26, no. 2, pp. 346–356, Feb. 2015.

[3] B. Piot, M. Geist, and O. Pietquin, Bridging the gap between imitation learning and inverse reinforcement learning, IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 8, pp. 1814–1826, Aug. 2017.

[4] J. Li, H. Modares, T. Chai, F. L. Lewis, and L. Xie, Off-policy reinforcement learning for synchronization in multiagent graphical games, IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 10, pp. 2434–2445, Oct. 2017.

[5] V. Mnih et al., Human-level control through deep reinforcement learning, Nature, vol. 518, pp. 529–533, 2015.

[6] F. Abtahi, Z. Zhu, and A. M. Burry, A deep reinforcement learning approach to character segmentation of license plate images, in Proc. IAPR Int. Conf. Mach. Vis. Appl., Jul. 2015, pp. 539–542.

[7] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, Deep Direct Reinforcement Learning for Financial Signal Representation and Trading. IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 3, pp. 653–664, Mar. 2017.

[8] K. Narasimhan, T. Kulkarni, and R. Barzilay, Language understanding for text-based games using deep reinforcement learning, in Proc. Conf. Empir. Methods Nature Lang. Process., Sep. 2015, pp. 1–11.

[9] H. Y. Ong, K. Chavez, and A. Hong. (2015). Distributed deep Q learning. [Online]. Available: https://arxiv.org/abs/1508.04186

[10] M. E. Taylor, G. Kuhlmann, and P. Stone, Accelerating search with transferred heuristics, in Proc. ICAPS Workshop AI Planning Learn., 2007.

[11] M. Riedmiller, Neural fitted Q iteration—First experiences with a data efficient neural reinforcement learning method, in Proc. Eur. Conf. Mach. Learn., Oct. 2005, pp. 317–328.

[12] A. Fachantidis, I. Partalas, G. Tsoumakas, and I. Vlahavas, Transferring models in hybrid reinforcement learning agents, in Proc. IFIP Adv. Inf. Commun. Technol., Sep. 2011, pp. 162–171.

[13] A. Lazaric, M. Restelli, and A. Bonarini, Transfer of samples in batch reinforcement learning, in Proc. 25th Int. Conf. Mach. Learn., Jul. 2008, pp. 544–551.