DNN Literature Review-10307389

Introduction

Deep neural networks are the foundation of many artificial intelligence applications currently (LeCun et al., 2015). Due to the breakthrough development of DNN in speech recognition (Deng et al., 2013) and image recognition (Krizhevsky et al.,2012), the application of DNN has grown significantly. DNN is deployed in a variety of applications, from natural language processing to vision or image processing to medicine to complex games (Silver et al., 2016), and in many areas, DNN can surpass human accuracy. The purpose of this report is to explore the application of DNN in the field of Atari games and compare the different methods.

In 1976 Atari launched the Atari 2600 game console in the United States, the first real home video game console system in history. The classic games in this game console include Breakout, Adventure, Bumper, Blasting Comet, Pac-Man, etc. The Arcade learning environment (ALE) is a common evaluation platform for artificial intelligence researchers, which is similar to the ImageNet challenge in image classification (Bellemare et al., 2013). This environment provides an Atari 2600 emulator which contains more than hundreds of games and also designed as a machine to fight human players. These games are built on a 160×210 pixels screen with 128 colours per pixel. Each game has a different motion space, including up to 18 possible actions.

Review of literature

To play Atari games well, a successful algorithm needs to solve both the game state representation and the strategy choice. The classical reinforcement learning algorithm assumes that the Value function can be represented by a table. Watkins and Dayan (1992) propose the Q-learning algorithm which can

record the value of each state and action pairing through a table. For example, in a linear function approximation, the value function of an action is represented as:

$$Q(s, a; \theta) = \theta^{\mathrm{T}} \varphi(s, a)$$

Where θ is a parameter that can be learned, and φ (·) is a feature function defined on the pairing of state and action.

This linear function approximation method can extract artificially designed features and learns the value functions in a linear combination of these features from recent game frames. Bellemare et al. (2013) of the University of Alberta first used the linear function approximation SARSA algorithm on ALE and designed four new general artificial feature sets. Although this algorithm has a good performance on Atari games, the reinforcement learning algorithm based on linear value function approximation is far weaker than human players in general.

Another training algorithm is oriented to transform reinforcement learning into supervised learning. The University of Michigan's Guo et al. (2014) use slow Monte Carlo tree search to generate small amounts of data to train fast convolutional neural networks, this neural network can mimic the behaviour of Monte Carlo search through data aggregation. Schulman et al. (2015) from Berkeley University use the trust region policy optimization to enable deep network learning strategies directly.

In 2013, a small company in London called DeepMind published a groundbreaking paper (Mnih et al., 2013) on how to get computers to learn to play Atari 2600 games. What's striking about this result is that the computer only looks at the screen pixels and receives rewards as the game scores increase. The same model architecture does not need to be modified and can be used to learn seven different games, three of which play better than humans.

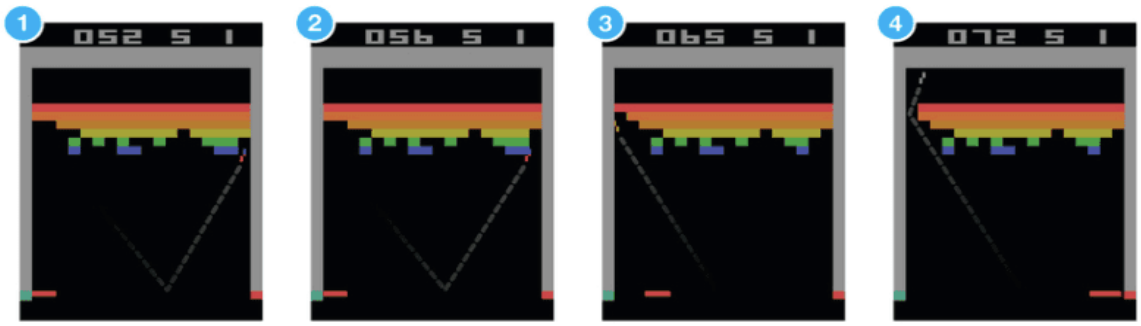
Obviously, deep neural networks can use nonlinear functions to efficiently represent Q (s, a; θ), thus greatly improving the ability to express the evaluation

function. Deep Q network (DQN) is an improvement of deep neural networks which is a combination of experience replay (Long-Ji Lin,1993) and target Q-separation. It trains the neural networks through Q-learning and uses uniform distribution to extract training samples. DeepMind's Schaul et al. (2016) optimized the sampling method for experience replay to improve the DQN performance on Atari games. Their idea is to more randomly select the state and action pairing with a larger temporal difference error.

DQN model in Breakout game

DQN is a classical algorithm of Deep Reinforcement Learning, which is between supervised learning and unsupervised learning. It has very few tags (rewards) and these tags are delayed. Through these rewards, the model constantly learns behaviour in the environment.

Take the game Breakout as an example to understand Deep Q network.



In this game, the player is required to control the tablet at the bottom of the screen to bounce the ball on the screen. There are a lot of bricks at the top of the screen. When the ball hits the brick, it will be crushed and the player will be rewarded.

In the neural network, the current screenshot state is input, the output is the action serve, left or right. According to the Q learning principle, the action with the maximum reward is directly selected as the next action to be performed. We can imagine that the neural network accepts external information, which is equivalent to collecting information from the nose, ears and eyes, then

outputting the value of each action through brain processing, and finally selecting the action by reinforcement learning.

The main factors that make DQN superior to other reinforcement learning algorithms are the use of Experience replay and Fixed Q-targets (Mnih et al., 2013). DQN has a memory library for learning previous experiences. Q learning is an off-policy offline learning method that learns from current experiences and learns from past experiences and even experiences from others. Fixed Q-targets is also a mechanism to disrupt correlation. If fixed Q-targets is used, we will use two neural networks with the same structure but different parameters in DQN. The neural network predicting Q estimation has the latest parameters. The parameters used by the neural network to predict Q reality are a long time ago. With these two enhancements, DQN can surpass humans in some games.

Conclusion

The game has always been an important branch of artificial intelligence. The success of deep learning in other fields has given the game artificial intelligence unprecedented inspiration. Although Atari games are no longer popular today, they provide an excellent soil and evaluation platform for deep learning in the game. Many of these ideas and techniques are immediately applied to other game problems such as poker, Go and even large real-time strategy game StarCraft.

References

A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in NIPS, 2012

Bellemare M, Naddaf Y, Veness J, Bowling M. The arcade learning environment: an evaluation platform for general agents. Journal of Artificial Intelligence Research, 2013, 47: 253−279

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of Go with deep neural networks and tree search,” Nature, vol. 529, no. 7587, pp. 484–489, Jan. 2016

Guo X X, Singh S, Lee H, Lewis R, Wang X S. Deep learning for real-time ATARI game play using offline Monte-Carlo tree search planning. In: Proceedings of the 2014 Advances in Neural Information Processing Systems. Cambridge: The MIT Press, 2014.

L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams et al., “Recent advances in deep learning for speech research at Microsoft,” in ICASSP, 2013.

Long-Ji Lin. Reinforcement learning for robots using neural networks. Technical report, DTIC Document, 1993.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. [online] arXiv.org. Available at: https://arxiv.org/abs/1312.5602 [Accessed 24 Mar. 2019].

Schulman J, Levine S, Moritz P, Jordan M, Abbeel P. Trust region policy optimization. In: Proceedings of the 32nd International Conference on Machine Learning. Lille, France: ICML, 2015.

Watkins C J H, Dayan P. Technical note: Q-learning. Machine Learning, 1992, 8(3−4): 279−292

Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, May 2015.