Deep and Reinforced Learning

Alexander Shieh

INFOR, Taipei Chien Kuo High School

1. Deep Learning

Deep Learning refers to the field of machine learning that uses artificial neural network as its main model. Artificial neural networks are universal classifiers and regressers, although it is flexible to different problem schemes, it is relatively much harder to train compared to other machine learning models. Advances in Deep Learning like CNNs and RNNs have been prominent models in tasks like image classification and sentimental analysis.

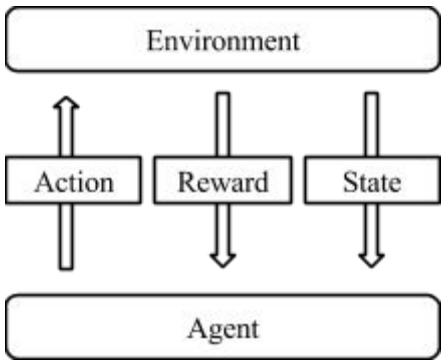
2. Reinforcement Learning

Reinforcement Learning is a specialized field in machine learning which was inspired by behaviorist psychology. It is centered around the problem of an agent learning specific tasks in an environment. The environment of Reinforcement Learning problems is carefully defined as Markov Decision Processes. Recent advances in reinforcement learning utilized both CNN[1] and RNN[2] to enhance move evaluation and policy search.

2.1. Markov Decision Process

A typical Markov Decision Process (MDP) describes an agent learning the optimal policy to gain maximal long-term reward from an environment. Elements of MDPs are as follows:

1. Set of states (S) of an environment
2. Set of actions (A) of a state
3. Probability distribution ($P(s,a,s')$) of state transition
4. Reward ($R(s,a,s')$) for taking an action and make a transition into another state
5. Policy (π), selected actions for the agent to take in each state



In real world Markov Decision Process problems, the probability distribution $P(s,a,s')$ and reward function $R(s,a,s')$ is unkown to the agent, this kind of situation is called Partially Observed Markov Decision Process (POMDP). As a result, the agent must learn them from observations or use model-free algorithms instead.[3]

2.2. Q-Learning

One of the simplest and most widely used model-free Reinforcement Learning Algorithms is the Q-Learning Algorithm. The Q-Learning Algorithm aims to find the best actions that holds

the best Q value, with the Q value representing the estimated long-term reward for taking an action in a state.

$$\pi^*(s) = argmax_a\, Q^*(s,a)$$

$$Q^*(s,a) = \sum_{s'} P(s,a,s')[R(s,a,s') + \gamma max_{a'}\, Q^*(s',a')]$$

The coefficient γ determines the proportion of the optimal Q value from the next state in current Q value. This coefficient alters the dependency of long-term rewards. To estimate the Q values in order to make it model-free, we can use the following formula to sample actual data from the environment and update previous predictions of Q values.

$$Q_i^*(s,a) = Q_{i-1}^*(s,a) + \alpha[R(s,a,s') + \gamma max_{a'}\, Q_{i-1}^*(s',a') - Q_{i-1}^*(s,a)]$$

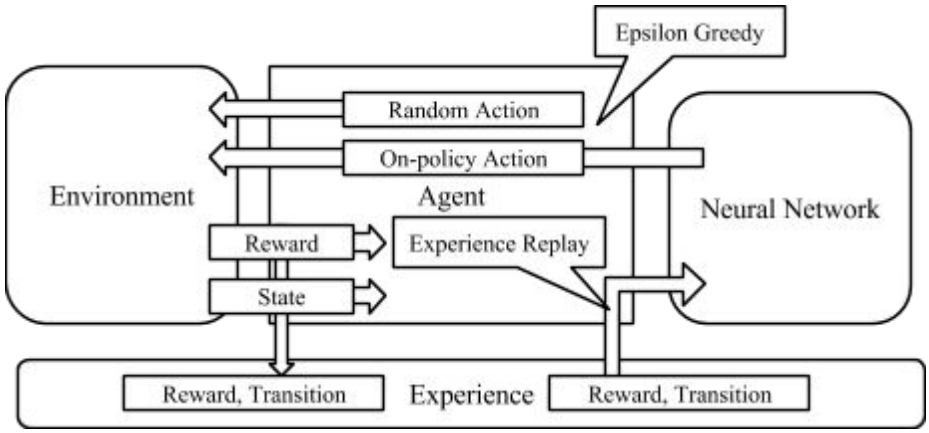
Learning rate coefficent α (usually pretty small) determines the amount of difference between the sample which will be added into the new Q value. From the above formula, we can see how Q values can be updated through iterations and trials by the agent over the environment until convergence.

Q-Learning has the advantage of not only being model-free, but also off-policy. This means no matter which sequence of actions the agent take to explore the environment won't affect the optimal policy learned. The Q-Learning algorithm often comes with ε-greedy exploration which allows the agent to try random actions while remain on the best policy for most of the time, in order not to stray away from optimal policy too much or lingering around the same states over and over.[4]

Though Q-Learning proposed a novel approach to MDP problems, problems still lies ahead because of the dimensionality of states. Most previous researches applied feature selection methods or used hand-built features to represent states and used other machine learning models like linear regression to learn and predict Q values. Another approach is using a neural network to do both feature selection and predicting together.[5] This will be the core of the latter introduced Deep Q-Learning algorithm.

3. Deep Q-Learning

A new approach called Deep Q-Learning using a technique called Experience Replay was proposed by Google Deep Mind in 2013[6]. This technique stores consecutive transitions (experience) taken over previous episodes and pick one of these transitions randomly to apply update to a neural network that predicts the Q values for each action of an input state. At the mean time, the agent will continue to do ε-greedy Q-Learning and store the sampled transition.



Work flow of the Deep Q-Learning Algorithm

The benefits of this approach is that it avoids sampling being dominated by on-policy decisions, which means it is more robust and balanced than using ε-greedy solely. But the disadvantages is once an important experience was erased, it becomes hard for the agent to replay that experience. Improvements are suggested in that paper with prioritized erasing.In order to make this approach feature-free, it uses a convolutional neural network to handle the state input (which is down sampled image) from games and outputs the Q value for each action of that state. This approach performed well with several atari games, and even exceeded human player level in two of seven games tested.

References

1. C. Maddison, A. Huang, I. Sutskever, D. Silver. Move Evaluation in Go Using Deep Convolutional Neural Networks. *ICLR 2015 (arXiv:1412.6564)*.
2. V.Mnih, N. Heess, A. Graves, K. Kavukcuoglu. Recurrent Models of Visual Attention. *NIPS 2014 (arXiv:1406.6247)*.
3. A. Ng. Shaping and Policy Search in Reinforcement Learning. *Doctoral Dissertation*, 2003.
4. Pieter Abbeel. UC Berkeley CS 188 Artificial Intelligence, 2011.
5. R. Sutton and A. Barto. Reinforcement Learning: An Introduction. *MIT Press*, 1998.
6. V. Mnih, K. Kavukcuoglu, D. Silver and A. Graves and Ioannis Antonoglou. Playing Atari with Deep Reinforcement Learning. *NIPS Deep Learning Workshop 2013 (arXiv:1312.5602)*.