From Reinforcement Learning to Generated Advertised Network

Jialu Wang
Shanghai Jiaotong University
Shanghai, China
faldict@sjtu.edu.cn

Abstract

In this paper, I tells from the basic theories and the change of my inner mind. After the exploration of DRL, I came up with a few interesting ideas and made efforts to find answers on myself.

1. Introduction

Reinforcement learning explicitly considers the whole problem of a goal-directed agent interacting with an uncertain environment. It faces the challenge of the trade-off between exploration and exploitation[1]. Generated adversarial networks is considered as a zero-sum game between generator and discriminator. In this paper, I give an explanation of these theory in a beginner level and try to express my comments.

2. Reinforcement Learning

The basic idea of reinforcement learning is simply to capture the most important aspects of the real problem facing a learning agent interacting with its environment to achieve a goal. At every moment, the agent observes the environment and makes a decision. Then the agent executes an action and influence the environment, while the environment emits a reward signal to the agent. As the process loops over and over, the agent senses the state of environment and always try to make decisions to acquire maximum long term rewards.

2.1. Four Elements of Reinforcement Learning

As you see forward, there are four main subelements of a reinforcement learning system beyond the agent and the environment: a policy, a reward signal, a value function and a model of the environment.

A policy defines the learning agent's way of behaving at a given time. It is a mapping from perceived states of the environment to actions to be taken when in those states.

A reward signal defines the goal in a reinforcement learning problem. On each time step, the environment sends to the agent a single number, a reward. The agent's goal is to maximize the total reward in the long run, thus the reward signals discriminate what is good and what is bad for the agent at the moment.

Whereas the reward signal indicates what is good in an immediate sense, a value function specifies what is good in the long run. It is the accumulation of the expected reward starting from that state over the future.

The last element is a model of the environment. I think it is a definition of what environment will behave as a reaction of the agent's state and action. Models are used for planning, by which agent might predict the resultant next state and next reward to make a precise decision.

2.2. Two Categories of Reinforcement Learning

Reinforcement learning can be divided into two categories: model-based and model-free. The former, considered as planning methods, require a model of the environment, such as planning methods, while the latter, considered as learning methods, that can be used without a model, such as Monte Carlo and temporal-difference methods[1].

The model-based algorithms contain a generative model. Generative models of time-series data can be used to simulate possible futures and could be used for planning and for reinforcement learning in a variety of ways. A generative model used for planning can learn a conditional distribution over future states of the world, given the current state of the world and hypothetical actions an agent might take as input. The agent can query the model with different potential actions and choose actions that the model predicts are likely to yield a desired state of the world. Another way that generative models might be used for reinforcement learning is to enable learning in an imaginary environment, where mistaken actions do not cause real damage to the agent. Generative models can also be used to guide exploration by keeping track of how often different states have been visited or different actions have been attempted previously. Generative models, and especially GANs, can also be used for

inverse reinforcement learning.

3. Deep Reinforcement Learning

Deep reinforcement learning is considered as the future direction of deep learning. However, in my opinion, DRL is still weak and needs more strong application to prove its importance. So what on earth is deep reinforcement learning? DeepMind have trained a DQN to play Atari game, regarded as the start of deep reinforcement learning. Frankly speaking, deep reinforcement learning is just a combination of deep learning and reinforcement learning. To give more details, it uses deep network to represent value function, policy or even model, and optimizes them end-to-end by using stochastic gradient descent. For example, in the Atari game, the deep network receives every frames of the game UI as input and outputs vectors as the state and the value of the reinforcement learning agent, which we can called AI. Then AI use Q-Learning to make a decision and behave an action. Do you still think DRL something complicated or unreadable?

In my words, deep learning could be the "eyes" of AI, and reinforcement learning is exactly the "brain". Deep learning "sees" the images of the game, while reinforcement learning decides how to play it. The training process is that AI tunes its policy to play according its experience of success and failure. As a normal human, AI also needs a "hand" to operate actions. In Atari Game, the "hand" hides behind the interface of OpenAI's gym library so we don't need to care about it. In other real world problems, such as self-driving, machinery is exactly acting as the role of "hands". The cooperation of "eyes", "brain" and "hands" is the key of AI, and the "brain" algorithm counts most important.

Now please permit me to express myself, something seemed stupid. I think the difference between machine visions and humans is not the "brain" to understand the image, but the "eyes" to see an object. Specifically, we can watch the world by our two eyes, while machine can only read the input photos.The time we have a 3D sense, the machines only understand the 2D world. Though just lack of one dimension, a large amount of knowledge has lost. I have tried to image myself as a machine, and when I have a shot of pictures, I would get confused sometimes. So I think training machines to understand higher dimension world directly is not the future of computer vision. The right direction is to generate enough knowledge first for machines to learn and understand. GAN seem to be one method, and this is the reason why I choose this direction and take more interests in it.

4. Generated Adversarial Networks

As I said before, I took some time to study GANs rather than delve deeper into DRL. In this section, I will introduce a little what I have learned, and these have ever surprised me more than DRL.

4.1. basic idea

To totally understand generative models, maximum likelihood estimation maybe a good start. Because of the limited time and space, I have to jump it and talk about GANs directly. As the name of "Adversary", the basic idea of GANs is to set up a game between two players[2]. One is the generator, who creates samples that are intended to come from the same distribution as the training data, the other player is the discriminator, who examines samples that are come from real or fake and learns using traditional supervised learning techniques. The generator always try to generate data like the real world data and treat the discriminator, while the discriminator always try to check whether the data come from the generator or the real world. Thus the problem become a game theory problem. GANs try to find a balance between generator and discriminator.

4.2. DCGAN

Today most GANs are based on the DCGAN architecture[3]. I would like to simply introduce some insights of it. The first is using the all convolutional net which replaces deterministic spatial pooling functions (such as maxpooling) with strided convolutions, allowing the network to learn its own spatial downsampling. It removes fully connected layers for deeper architectures. Next is the trend towards eliminating fully connected layers on top of convolutional features. Third is using batch normalization layers[4] in most layers of both the discriminator and the generator, with the two minibatches for the discriminator normalized separately. Finally, it uses Adam optimizer[5] instead of SGD.

4.3. Ideas to combine GAN with RL

In this part, I would propose two measures to make a combination of GAN and RL. These guesses are just my ideas in a low level awareness, so they still need to design experiments to validate.

First, it is necessary to balance the two players to prevent one from overpowering the other. And I guess whether we can let RL play a role of judger to make a balance between generator and discriminator. Obviously it is a dynamic programming problem, and a comparison between adjusting the optimizer of generator and discriminator by different algorithms, such as greedy and DP, is worthwhile to be done.

Next, I have asked myself what the growth and evolution of generator and discriminator look like. Finally I come

up with the coevolution between humans and environments. Then I take the generator as the agent while take the discriminator as the environment in a reinforcement learning system. The generator's action is to generate the sample data and the environment's reward is the discriminator's output. Also, discriminator acts as the model of environments. The only difficulty is how to define the policy functions. After all, it is just my guess and needs to try.

Above is my ideas and maybe I need more time to think of those.

5. Conclusion

In summary, I have walked through the world of RL and GAN and made a varieties of ideas. In this training stage, I have greatly developed my skills of reading papers and writing codes. Last but not the least, I have found my directions and interests. In the next stage, I would like to delve deeper and do something amazing.

References

[1] Richard S. Sutton, Andrew G. Barto (2016). Reinforcement Learning: An Introduction. https://webdocs.cs.ualberta.ca/ sutton/book/bookdraft2016sep.pdf

[2] Ian Goodfellow. NIPS 2016 Tutorial: Generative Adversarial Networks. arXiv preprint arXiv:1701.00160

[3] Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434

[4] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep net- work training by reducing internal covariate shift

[5] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980