Week 6-1: Paper Summaries

CE-510 Seminar: Social Media Mining

Student Name: Jiaqi Guo NetID: JGR9647

■ ImageNet Classification with Deep Convolutional Neural Networks

In this paper, the author proposes a groundbreaking structure, AlexNet, which contains five convolutional layers and three fully connected layers. Its characteristics can be summarized as follows:

1. **Convergence/training was accelerated** with the use of linear correction unit, **ReLU**. ReLU is a kind of nonlinear neuron (when input x is greater than 0, output x; The input x is less than 0, then the output 0) with faster calculation speed and faster convergence than the previous activation function.

2. **Data augmentation:** This ensures the diversity of data and makes the model fit better, the data augmentation methods used by the author include flipping the image horizontally and changing the pixel intensity value of RGB on the image

3. **Dropout:** The dropout model training will soon overfit, adding a double to the iteration times.

4. **Muti GPU training:** The author divided the network into two parts, divided it into two Gpus, and solved the problem by parallel computing

In addition, this paper also proves the importance of data and model size, that is, to solve the task of thousands classification level, a sufficiently large model is required, and a sufficiently large model requires enough data with a certain diversity for fitting. According to the experiment, even removing a single convolutional layer will have a huge impact on the final classification result.

Possible Improvement Directions:

1. The model has 60 million parameters, which requires a lot of calculation. Future developments could be aimed at reducing computational complexity

■ Mastering the game of Go with deep neural networks and tree search

Go has long been considered the most challenging classic game in the field of artificial intelligence because of its vast search space and the difficulty of evaluating moves and moves. Here, the author of this paper introduces a new approach to go calculation, which uses a "value network" to evaluate the game and a "strategy network" to select positions. The chess games played by go masters are supervised to learn, and the chess games played by themselves are strengthened to learn, and the two are combined to train the deep neural network.

In AlphaGo, the author represented the checkerboard position as 19x19 image and used convolutional neural network to construct position representation. At the same time, the authors use neural networks to reduce the effective depth and breadth of the search tree. The training process can be summarized as follows:

1. First, supervised learning (SL) is used to train a strategy network P_σ, and the training data is human expert knowledge. The supervised learning method makes the gradient of high quality and fast update.

2. Secondly, they train a fast strategy network P_π with shallow network depth, which can quickly select behaviors in Monte Carlo Rollouts.

3. A reinforcement learning (RL) strategy network P_ρ is trained to improve the SL strategy network by optimizing the results of **selfplay**, which adjusts the strategy to win the game rather than maximize the prediction accuracy.

4. Finally, a value network V_θ is trained to predict the winner of the game between RL strategy network and itself.

Possible Improvement Directions:

1. To evaluate the situation of Go is to regard it as a statistical problem. When the computer evaluates a situation, it makes random moves until the end of the game, and then repeats the game over and over again to see what the winning rate is. If the winning rate is high, the situation is considered good. In other words, it considers a good situation to have a high probability of winning a simulated game.

   But in some cases, that's not true. If you've ever played an endgame of chess, you'll know that there's usually only one move you can win, all the other moves you can lose, and the odds are very low, but you can win if you make the right move. Therefore, a low win percentage does not mean a bad situation, and a high win percentage does not mean a good situation. This is why the Monte Carlo method has loopholes. Moreover, the problem will not fundamentally change as the number of simulated games increases.

   AlphaGo's method of simulating matches is based on the Monte Carlo method, but it is improved. Instead of making random shots, AlphaGo makes them in better positions, but it still uses the odds to judge a situation. It has been argued that a high winning rate does not equal a good situation. So we can optimize the settlement method by improving the algorithm's evaluation strategy for a game