The Principles Of A.I Alphago

YinChen Wu

Dr. Hubert Bray

Duke Summer Session

20 july 2017

Introduction

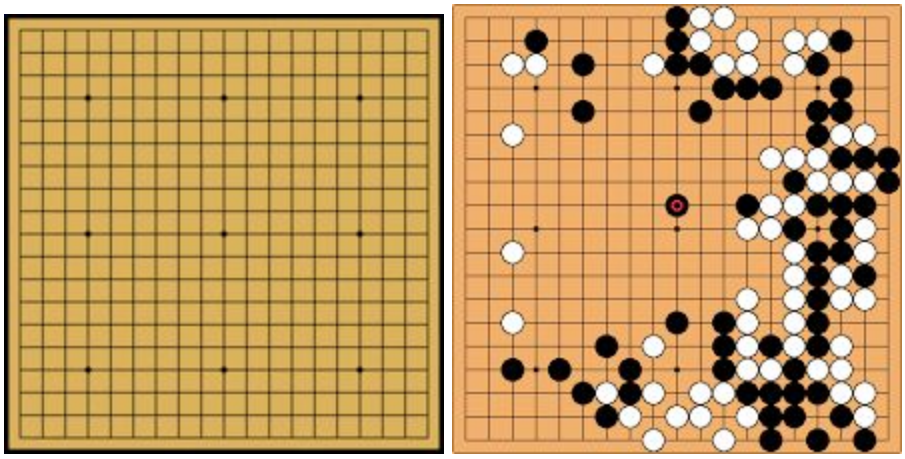
Go, a traditional Chinese board game, is a remarkable work of art which has been invented for more than 2500 years. With the blending of the chess pieces in the checkerboard, also comes the spark of the wisdom, the profoundness, and the special sensitivity of art. In the past, I believed that this talent of playing Go is exclusively owned by humans, that only we have the intuition and the sensitivity to touch the soul of Go. However, on 27 May 2017, the artificial intelligence, called “Alphago”, has successfully won all three games against Ke Jie, who has been ranked 1st among all human players worldwide since late 2014[1]. Does the success of the Alphago mean that the techniques of A.I has already been well developed that the machine is now able to overpower humans even in the fields like Go where the intuition plays a great role? There is no answer unless we take a deep look at how the Alphago operates. Therefore, let’s try to comprehend the principles and concepts of the Alphago in this paper.

What is Go

Before we begin to investigate the abstract principles underneath the Alphago, we first need to understand the rules of Go. The game of Go usually starts with an empty board. Two players have unlimited supply of pieces (called stone) and each turn put one stone on the board. The winner is decided by whose stones have surrounded the bigger territories, and if a stone is completely surrounded by opponent's stones, the opponent can capture the stone. The stone has to be put on the intersection of lines

[1] Source from Wikipedia

rather than on the squares.[2] Below are two games from the game of Go. You can see that the board is formed by 19x19 lines and the stones are only put on the intersections of lines.



from wiki

Although the rules of Go are much simpler than other board games, it is extremely hard and time-consuming. Usually, it takes an adult 2 months to be familiar with the game and 1-2 years to get the rank of 1 dan. Take an example of me. I studied the game of Go from 1st grade to 5th grade in the elementary school, but only got the rank of 3 dan in the end. Here are some pictures to help you get the better sense how hard Go is.

Ranking System in Kendo	
Unranked	New students
Kyu Grades	
6th Kyu	
5th Kyu	
4th Kyu	Begin bogu practice
3rd Kyu	
2nd Kyu	
1st Kyu	~1 year (minimum)
Dan Level	
1st Dan	Beginner phase
2nd Dan	
3rd Dan	
4th Dan	Intermediate phase
5th Dan	Sensei, ~15 years
6th Dan	
7th Dan	Advanced phase
8th Dan	

IKKF KYU / DAN LEVEL STRUCTURE

Kyu/Dan Level	Belt	Belt Designation
10th Kyu	White	White Belt - plain
9th Kyu	White	White Belt - 1 green stripe
8th Kyu	White	White Belt - 2 green stripes
7th Kyu	White	White Belt - 3 green stripes or Yellow Belt
6th Kyu	Green	Green Belt - 1 brown stripe
5th Kyu	Green	Green Belt - 2 brown stripes
4th Kyu	Green	Green Belt - 3 brown stripes
3rd Kyu	Brown	Brown Belt - 1 black stripe
2nd Kyu	Brown	Brown Belt - 2 black stripes
1st Kyu	Brown	Brown Belt - 3 black stripes
1st Dan	Black	Black Belt
2nd Dan	Black	Black Belt
3rd Dan	Black	Black Belt
4th Dan	Black	Black Belt
5th Dan	Black	Black Belt
6th Dan	Black	Black Belt
7th Dan	Black	Black Belt or Red & White Belt
8th Dan	Black	Black Belt or Red & White Belt
9th Dan	Black	Black Belt or Red Belt
10th Dan	Black	Black Belt or Red Belt

from wiki

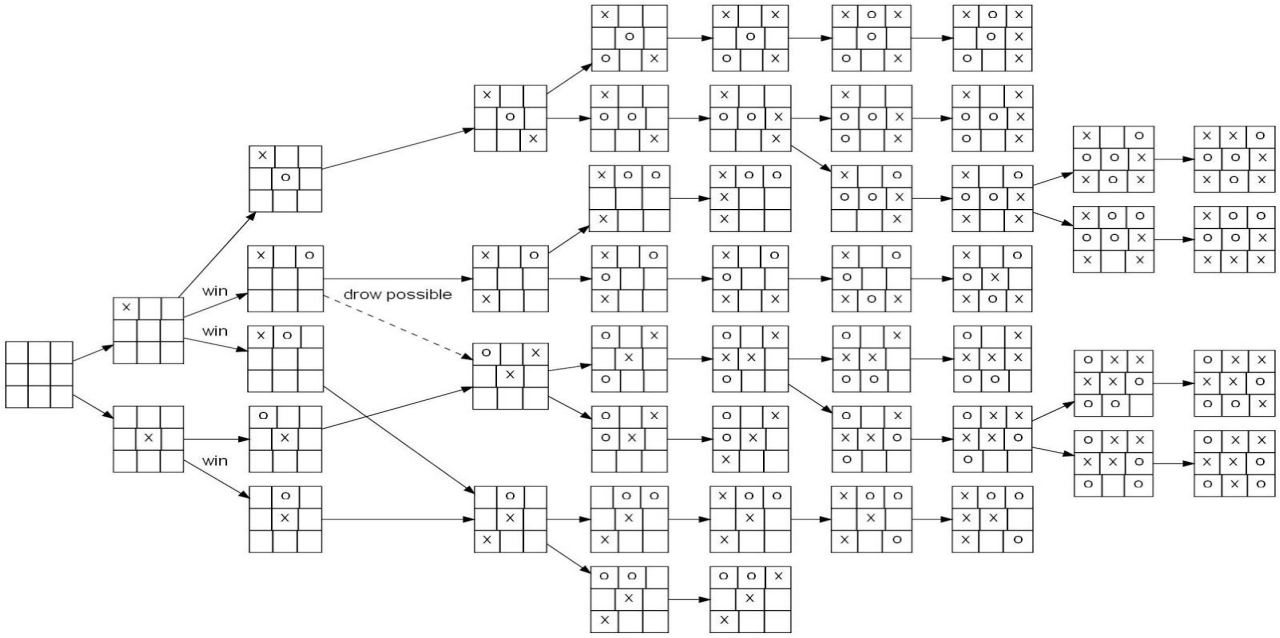
[2] Information referenced from britgo.org

The difficulty of Go is not only attributed to the simpleness of the rules of Go that players can play whatever ways they want, but also can be ascribed to the infinite possibilities of the game. In the game of Go, a player each turn is generally faced with a choice of a greater number of possible moves compared to chess(about 250 in Go vs. 35 in chess). Therefore, the total number of possible games of Go is estimated at 10 to the power of 761, compared to the 10 to the 120 for chess. [3] This is also a reason why it is extremely difficult for A.I to beat human in the field of Go. If A.I wants to play well in Go, it needs to calculate all the possible outcomes of the game and pick the best action that minimizes the worst possible cases it may suffer, which is a tremendous burden for the calculator to calculate. Even though in the game of chess, the A.I designer still has paid a lot of attention to reduce the amount of calculation for calculator. For example, in the deep blue[4], the calculator only requires to search the possible outcomes in the depth of six moves, and then using an evaluation function to compare which move is the best one. By only searching to six moves and replacing the outcomes with a single value summarizing, Deep Blue successfully minimized the algorithm. Now if Alphago wants to beat human, it definitely cannot use the traditional way of calculating all the possibilities, it should try something new.

Below is an example of all the possible games in the Tic-Tac-Toe. You can see for a game simple as Tic-Tac-Toe, there are only a few possibilities and therefore really easy for A.I to play.

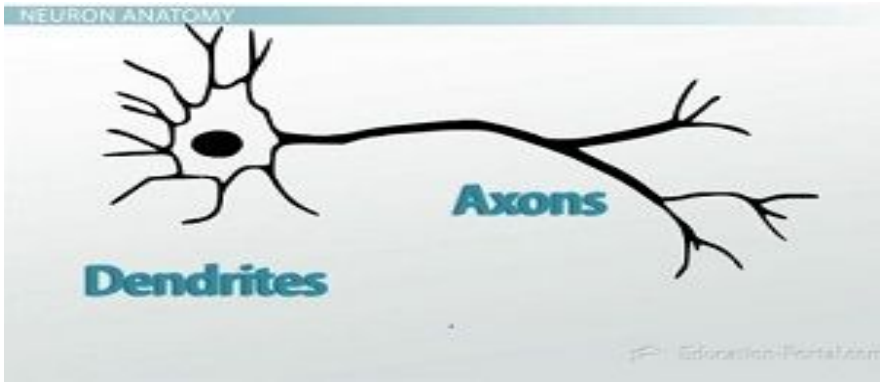
---

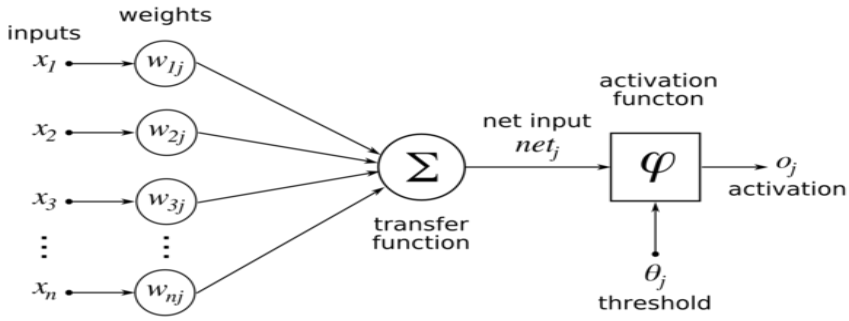
[3] Information from wikipedia
[4] The A.I that beats Kasparov in chess in 1997



Artificial Neural Network

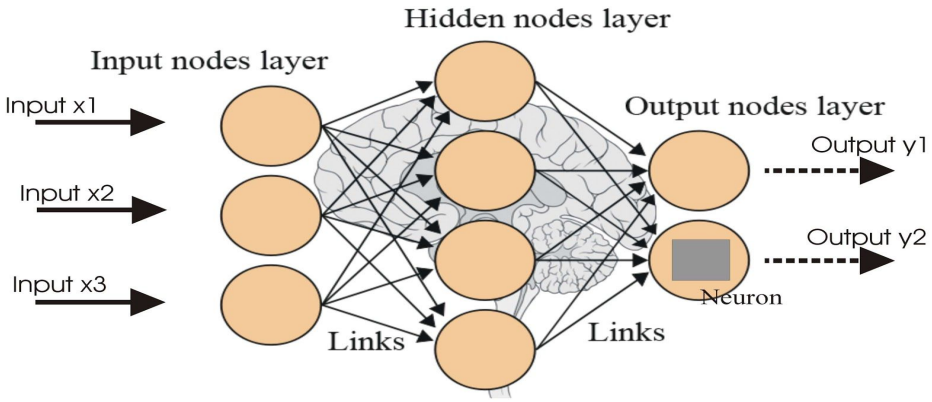
One of the techniques that Alphago applies to play Go is the “Artificial Neural Network”. In fact, Alphago does not directly use the technique of the ANN, but uses a way that’s developed from the ANN. In the ANN, the “neurons” is for gathering the inputs or data to summarize(or weight) them(like the dendrites in the brain), and then transferring those inputs to outputs through the activation functions(like the axons in the brain). There are two pictures of the real neurons and the neurons in the ANN.





Source both from Wikipedia

And the effect of the Artificial Neural network is connecting those neurons. There are the input node layer, the output node layer, and the hidden layer which is used to increase the complexity of the neurons to simulate the more complex functions.

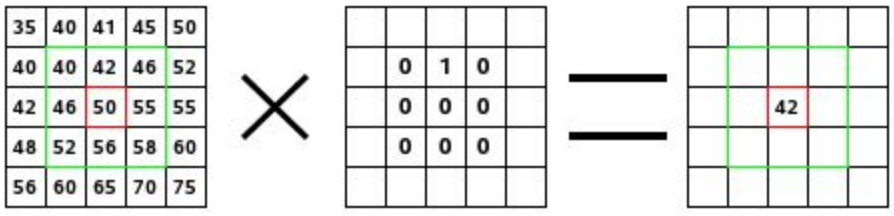


from wiki

In the base of Artificial Neural Network, people soon develop a technique called “Convolutional Neural Network”[5], which is also the one that the Alphago has used though this network is generally used to the image identification. Now imagine you want to put an image of 100x100 data in the Artificial Neural Network, if the number of neurons in the hidden node layer is equal to the neurons in the input node layer, you need to calculate a weighting[6] of 10^8, which is an immensely enormous number. Here the Convolutional Neural comes up with two crucial ideas.

The first is to just connect the data with the adjacent data to decrease the amount of calculation to the links between neurons. For example, if a neuron only requires to build up links with the adjacent 10*10 data, for an image of 100x100 data, what we need to calculate is merely 100*100*(10*10) = 10^6.

Another idea is using the convolution. As the picture shows, here we use many 3x3 kernels transfer the input of 5x5 image to an output of 3x3 convolved feature. The kernel is a feature, or a 3x3 matrix if you wanna see it as. In the network, the kernels gradually convolute around the input data, starting from the 3x3 square with the center at 40, then the square center 42, and then 46, until the last square with center at 58.

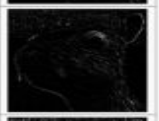


from wiki

[5]According to Wiki, “a **convolutional neural network** is a class of deep, feed-forward artificial neural network that have successfully been applied to analyzing visual imagery”

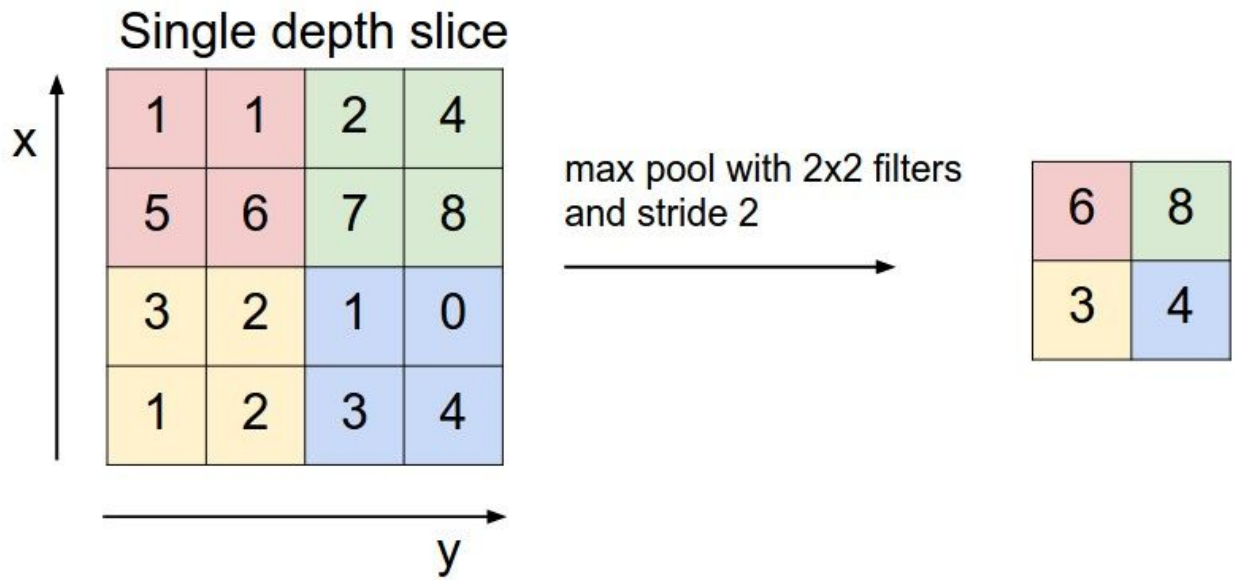
[6] Weight here means the strength or amplitude of a connection between two nodes

At last, a 3x3 convolved picture is formed(the green square on the left). Each kernel represents a typical feature, and the convolved picture is actually the original picture when highlighting a certain feature. In a Convolutional Neural Network, there are many different kernels to capture different features. Below is a picture using different kernels, or filters someone may call, to capture different features in an image of a dog.

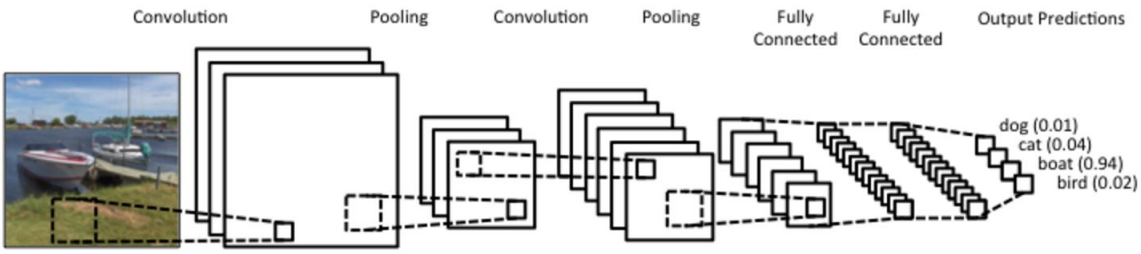
Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

source from zhi hu

However, if the network of the Convolutional Neural Network is exceedingly complex which may have tons of hidden layers and kernels, while the inputs are relatively small, it may cause the overfitting. Therefore, people then introduced the concepts of pooling or subsampling in another word, to highlight the feature in a certain square by summarizing all the data in this certain square. Here is a picture for the pooling.



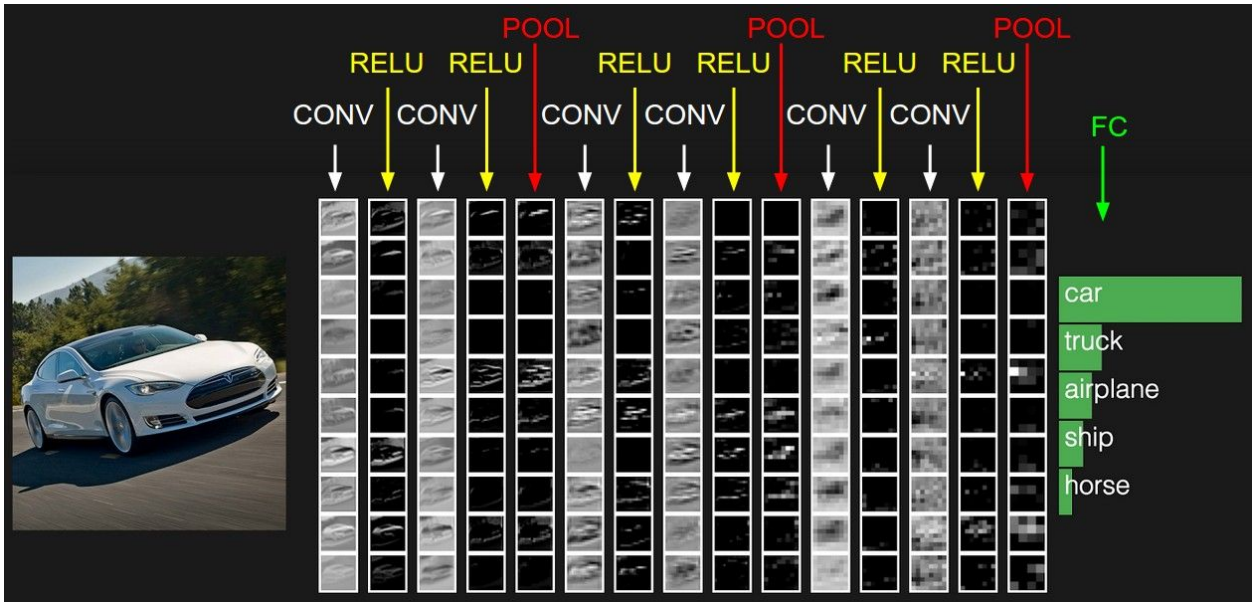
Generally, the image identification just uses the combination of both convolution and pooling. Below is right the procedure of the image identification.



source from wiki

In the past, if we wanna let computer itself to distinguish different pictures, we need to first find the feature of the picture ourselves to give the neural network to study. However, with the Convolutional Neural Network, we can just directly give a bunch of

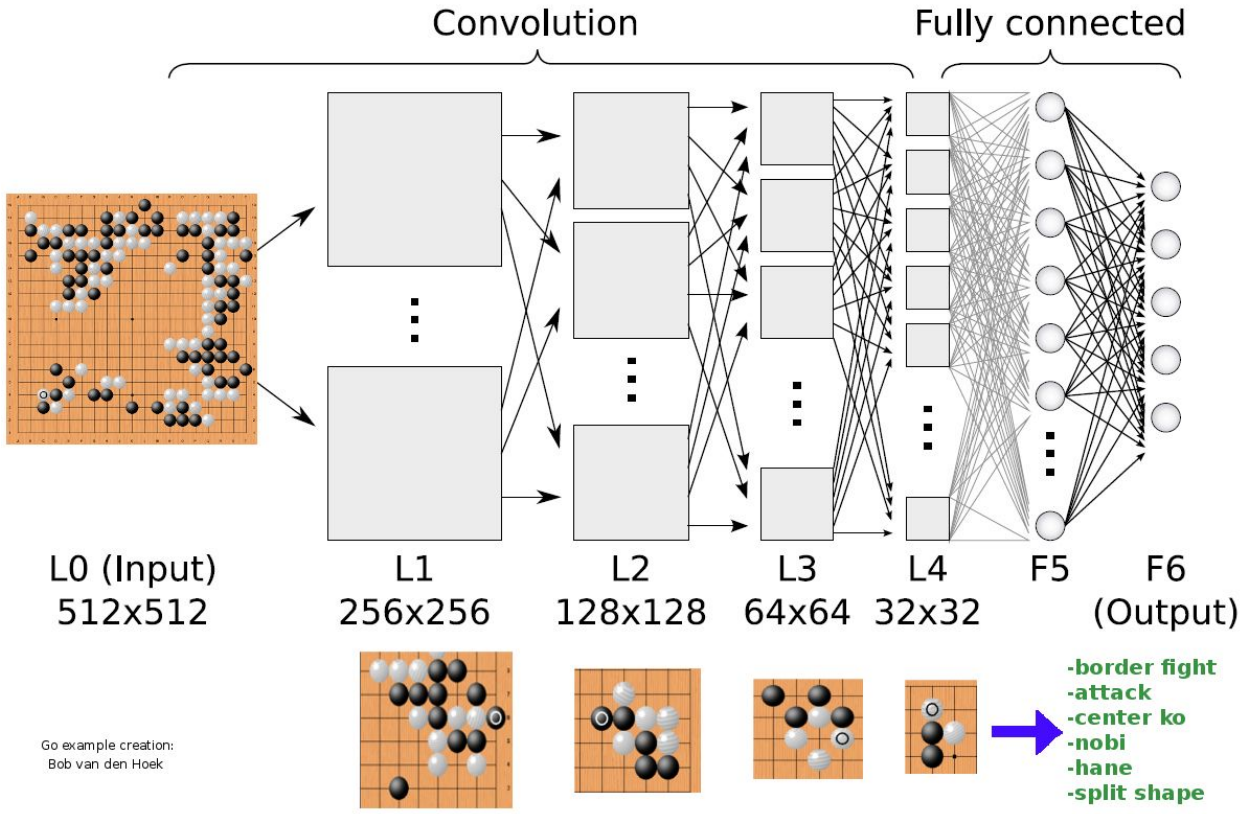
data to the neural network, and through the filters(kernels) the neural network can find the features itself. The more the kernels the Convolutional Neural Network has, the more advanced and abstract the feature can be. Therefore, with this technique, we don't need to practice the neural network, giving them the feature to distinguish between cars and trucks. What we need to do is merely giving it numerous images of cars and trucks, and it can find the abstract definition of cars or trucks itself.



Source from wiki

Until now, do you find that the image identification by Convolutional Neural Network is extremely similar to the game of Go? Go, like a 19x19 image, is a 19x19 square, and its rules are also not as exactly clear as chess, requiring certain intuition to play. Therefore, by the way of Convolutional Neural Network, the A.I designer doesn't need to teach Alphago the rules of Go, but gives the Alphago numerous Go games played by players and the results of those games, the Alphago itself can find the abstract concepts and

logics of Go through the bunch of games. Below is a picture analyzing a game of Go using the Convolutional Neural Network.

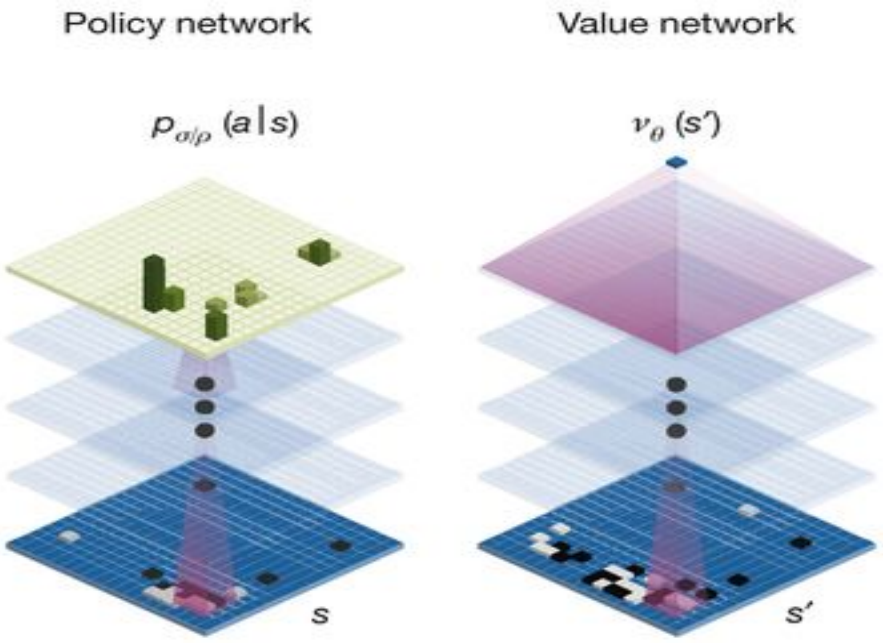


Alphago

Similar to Deep Blue, which primarily relies on the brute-force approach and the evaluation function, Alphago also has applied two techniques: The Convolutional Network as I mentioned before, and a tree search procedure. The Convolutional Network plays a role like the evaluation function, but the Convolutional network is learned by A.I itself and not created by designers. The tree search procedure can be

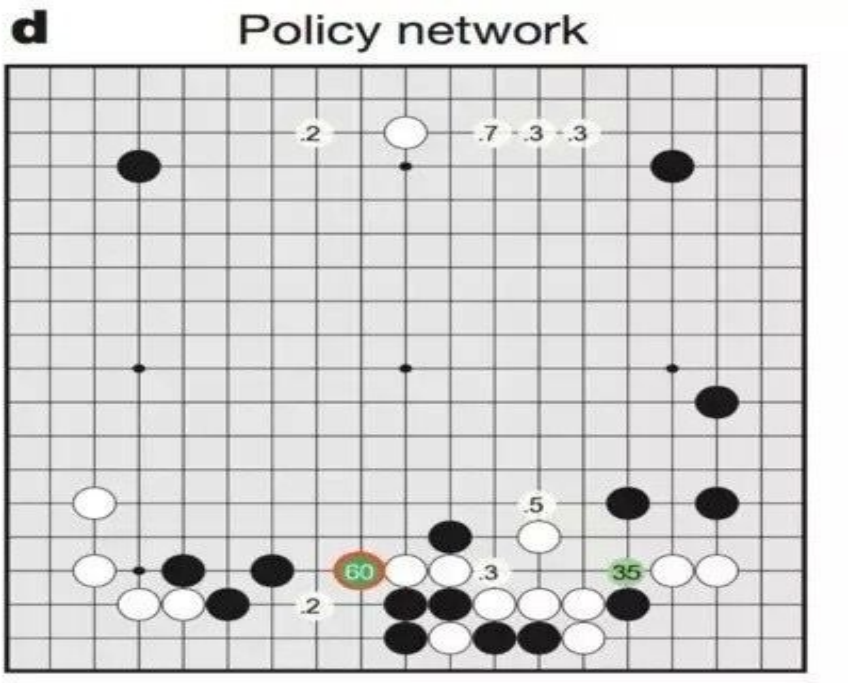
considered as the reflection in the gameplay, whereas the Convolutional Network acts as the intuition.

Alphago possesses with three convolutional networks, of two different kinds: the policy network and the value network. Both networks are basically similar to the convolutional networks used to image identification, with the only difference that the input is rather the coordinates of the stones put on the board.



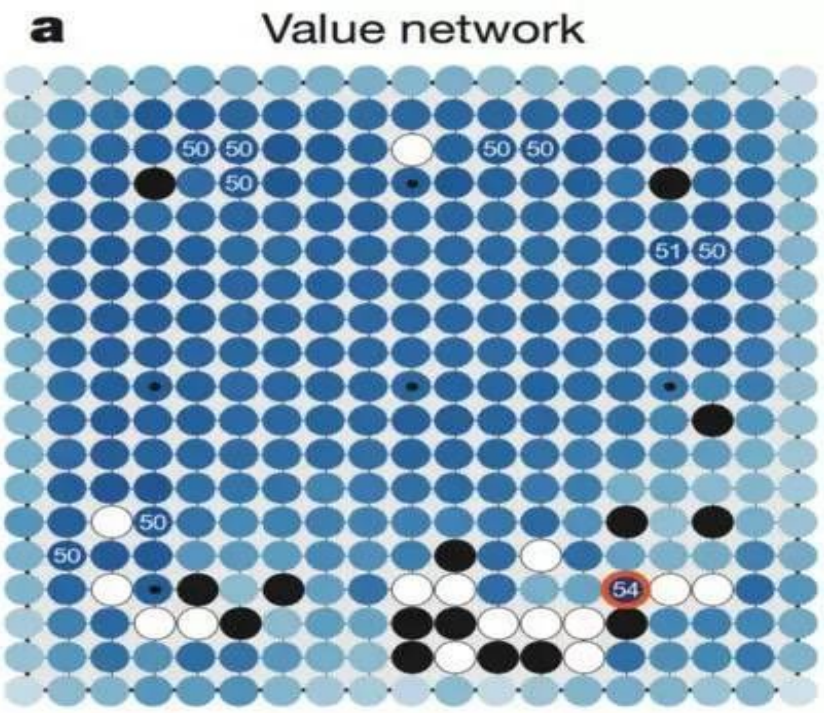
The policy network is used to predict where the opponent will put the stone. Designers have input countless game positions played by the professional players into the policy network, enlarging the database so that the network can predict the most possible position of next move. However, predicting human moves is not what Alphago is supposed to do. Rather, what Alphago should do is optimize the chance to win. Therefore, the designers then developed a method called “Deep Reinforcement

Learning” to improve the policy network. A fundamental policy network is created on the basis of part of the total game positions(about half of the all thirty millions different games played by players), fighting against a complete policy network build according to all the game positions(or game records). In the process of self-fighting, the fundamental policy network can soon be familiar with the possibilities where the complete policy network may put the stone, and there create the new database on the basis of that. After creating the new database, the fundamental policy network has been improved, and therefore can help the complete policy network built another newer database. And it forms a circle, gradually enhancing Alphago’s ability to find which move is most likely to win.



The value network estimates the value of each move, given the current state of game. The input here is the whole board of the game, while the output is the possibility of winning for each move. Like the policy network, the value network is also provided with

an overwhelming amount of games played by professional players. According to those data of games, Alphago can build up a sense to evaluate the chance to win of each move. However, we cannot assure that those games played by professional players are all completely accurate. For example, a professional player put a stone in one place is not because he thinks putting there is the most correct choice, but he knew that his opponent is a cautious person that he can take advantage of it by putting the stone there. And the case like that is really common in the matches of Go, though many people just are not aware of it. For this problem, designers set up two Alphago to fight against each other. Because they are both A.I, and their skills of Go are equal, the cases like that certainly would not happen. Therefore, through that Alphago can soon build up the correct database of evaluation on the basis of the outcomes played by two Alphago.



Putting all pieces together: Tree search

The final step of techniques behind Alphago is the tree search, Monte Carlo tree search as the full name. As the picture shows, there are four stages within: Selection, Expansion, Evaluation, and the Backup.

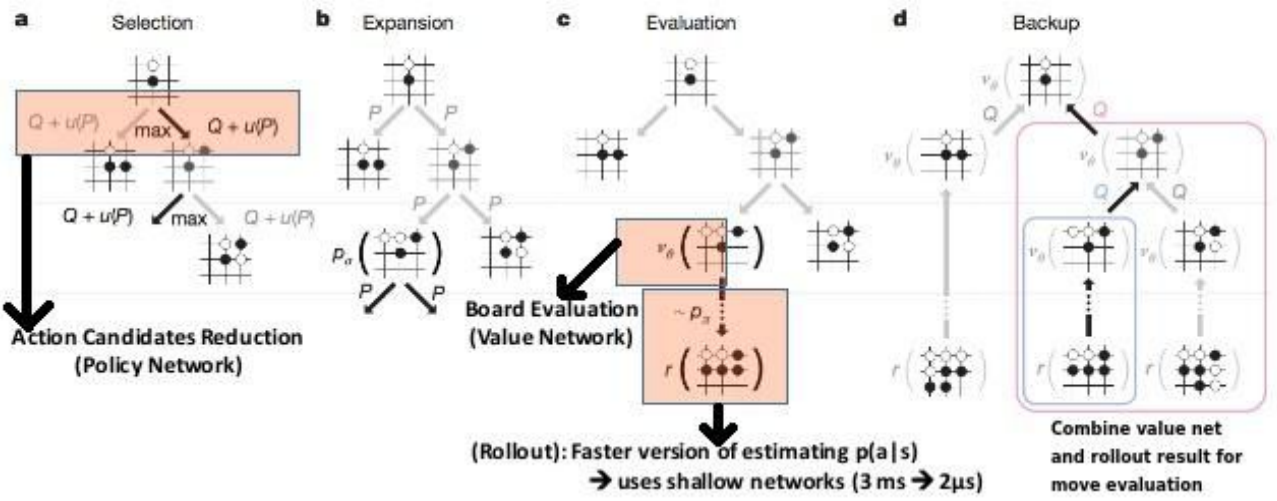
1. Selection: Given the current state of game, select some possible moves

2. Expansion: According to those possible moves, expand the one that Alphago has the most chance to win.

3. Evaluation: There are two ways to evaluate where Alphago should put the stone: one is directly using the Value Network to predict, another is to continue expanding to predict the possibilities in the further moves. Note that here Alphago uses another smaller policy network to expand. Though its accuracy is relatively lower, it is much faster(1500 times faster from 3 ms to 2 us) . The mixing rate of the value network and the expanding designers declare is 50% by each.

4. Backup: After deciding the best move, Alphago then begins to predict the possible move for the opponent, and afterward starts to calculate the moves further behind.

Looking ahead (w/ Monte Carlo Search Tree)



Conclusion

In conclusion, even though the technique of Alphago is essentially different with Deep Blue, where techniques in Alphago is all learned by Alphago itself using the Convolutional Network and the skills for Deep Blue is all designed by human, Alphago is still not able to comprehend the tactics and beauty of Go, what it does is merely using two powerful functions to determine where it should put the stone. Therefore, although indeed the winning of Alphago in the field of Go is a landmark symbol of the brilliant development in A.I techniques, there is still a long way which requires countless people's diligent mind and assiduous working before the A.I really be advanced and developed.

Bibliography

1.“Google DeepMind's AlphaGo: How It Works.... : Hacker News: Newest - howlDb." Google DeepMind's AlphaGo: How It Works.... : Hacker News: Newest - howlDb. Accessed July 24, 2017. http://howldb.com/p/google-deepmind-s-alphago-how-it-works-0sffkq

2."AlphaGo." DeepMind. Accessed July 24, 2017. https://deepmind.com/research/alphago/

3."Unsupervised learning." Wikipedia. July 13, 2017. Accessed July 24, 2017. https://en.wikipedia.org/wiki/Unsupervised_learning

4.OpenAI. "Unsupervised Sentiment Neuron." OpenAI Blog. May 22, 2017. Accessed July 24, 2017. https://blog.openai.com/unsupervised-sentiment-neuron/.

5."How do artificial neural networks work? - Quora." Accessed July 24, 2017. https://www.bing.com/cr?IG=EFE26B3F715E4801B9F5ABCE9A3E6C47&CID=3100DAB54B0663F21073D07C4A006227&rd=1&h=JnOoqfHIarVW-7SAc_h-Fo64ni7Rn51EJPBzf6z35q4&v=1&r=https%3a%2f%2fwww.quora.com%2fHow-do-artificial-neural-networks-work&p=DevEx,5382.1

6.Artificial neural network." Wikipedia. July 20, 2017. Accessed July 24, 2017. https://en.wikipedia.org/wiki/Artificial_neural_network

7.Convolutional neural network." Wikipedia. July 22, 2017. Accessed July 24, 2017. https://en.wikipedia.org/wiki/Convolutional_neural_network.

8.Monte Carlo tree search." Wikipedia. July 04, 2017. Accessed July 24, 2017. https://en.wikipedia.org/wiki/Monte_Carlo_tree_search

9.Brute-force search." Wikipedia. June 27, 2017. Accessed July 24, 2017. https://en.wikipedia.org/wiki/Brute-force_search.

10.Innovations of AlphaGo." DeepMind. Accessed July 24, 2017. https://deepmind.com/blog/innovations-alphago/.

11."Neural Networks and their Application to Go - ETH Z." Accessed July 24, 2017. https://www.bing.com/cr?IG=A92B0AE83C9747278F05B021DF6672FE&CID=25F778385A956F3C0EE472F15B936E43&rd=1&h=cMIckWaajssBQgalYAXQfwJWc3z7617-HYomIn1sDgI&v=1&r=https%3a%2f%2fstat.ethz.ch%2feducation%2fsemesters%2fss2016%2fseminar%2ffiles%2fslides%2ftalk11_Neural_Networks.pdf&p=DevEx,5343.1

12.Nielsen, Michael A. "Neural Networks and Deep Learning." Neural networks and deep learning. January 01, 1970. Accessed July 24, 2017. http://neuralnetworksanddeeplearning.com/chap1.htm