1 Select a Game-Playing paper from the following list or another of your choosing:

- Game Tree Searching by Min / Max Approximation by Ron Rivest, MIT (Fun fact, Ron Rivest is the R is in the RSA cryptographic protocol).
- Deep Blue by the IBM Watson Team (Fun fact, Deep Blue beat Gary Kasparov in Chess in one of the most famous AI spectacles of the 20th century).
- AlphaGo by the DeepMind Team.

2 Write a simple one page summary of the paper covering the following:

- A brief summary of the paper's goals or techniques introduced (if any).
- A brief summary of the paper's results (if any).

I choose DeepMind Team's paper "Mastering the game of Go with deep neural networks and tree search", published on Nature 529,484–489 (28 January 2016).

why Go board game is difficult?

The computation complexity is d^d where b is breadth and d is depth. For a full size Go board, the typical value is (b=250, d=150), compared to chess (b=35, d= 80). So exhaustive search of every possible move in Go game is infeasible.

In the past, the effective search space is reduced by 2 general principles:

1. reduce the search depth by position evaluation. Truncate the search tree at state s and replace the subtree below s by an approximate value function. But it was believed to be intractable in Go due to the complexity of the game.
2. reduce the search breadth by sampling actions from a policy p(a|s) that is a probality distribution over possible moves a in position s. For example, MonteCarlo rollouts search to maximum depth without branching at all.

In this paper, a combination of 3 techniques are used:

1. **Monte Carlo tree search** (MCTS): fast rollout policy, sample long sequences of actions and estimate the value of each state in a search tree.
2. **policy network**: pass the board position as 19x19 image and use convolutional layers to construct a representation of the position. **supervised learning** from expert human moves, which provides

fast, efficient learning updates with immediate feedback and high-quality gradient. It has 13 layers. Then improve the network by **reinforcement learning** under self-play.

3. **value network**: evaluate board position. Train the value function using the reinforcement learning policy network in the previous step.

Results: It turns out a combination of these 3 techniques performed best, winning > 95% of games against other variants. It defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-size game of Go.

What I learned? People learn from the end results, not from the empty, random thoughts. A policy can be made from experience to make wise choice. The value is based on top of the choice. **In short, end goal first, then choice, then value.**