Research review summary - AlphaGo

This article is going to have a brief summary about one of the most famous artificial intelligence(a.i.) implementation developed by Google DeepMind, AlphaGo. Here is the related research paper:
https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf
It is always giving a feeling to layman that artificial intelligence is something that untouchable or impossible to understand. Here, I am going to have a brief review of the concept from AlphaGo.

To begin with, if we are talking about artificial intelligence implementation, some of the topics we cannot escape from, such as game tree, search or pruning. Actually, game tree is kind of simulating what will the situation be if we make a move. For example, we might want to imagine how the Go (chess game) will be if you or your opposite make a move. Therefore, it is useful to build a game tree to simulate the game in order to have the final win. After simulating all the possible situation, search function will take in charge to help us finding out which move(s) might lead us to the final win. However, it is really time consuming if we search through every possible moves. Therefore, pruning function will come up to help us deleting some obvious useless or helpless move.

Unfortunately, the possible outcomes are still too many. We can try to imagine that there is a 17 x 17 or 19 x 19 grid in a Go game. We might need to spend few days or a year to go through each final possible outcome in order to make a single move, although we have a intel i-7, one of the fastest cpu, in our pc.

Based on this information, it is interesting to know what AlphaGo can make the ‘correct’ decision quickly. Rather than only apply some typically searching function, like Min Max or Alpha Beta, the clever developers applied a supervised machine learning and rollout knowledge inside in order to cut down some useless moves in calculation. The development team uses ‘value networks’ to evaluate board positions and ‘policy networks’ to select moves. These deep neural networks are trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play. Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs that simulate thousands of random games of self-play. We also introduce a new search algorithm that combines Monte Carlo simulation with value and policy networks.

Therefore, the great AlphaGo can make the ‘correct’ choice fastly and defeat lots of expert as a result. Finally, rather than appreciate how AlphaGo win the game, this might also provide an information to explain a question we might ask.
Why AlphaGo loses one of the game with Korean player.