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

Go is an incurably hard game for a computer system to approach using traditional methods, the paper we are reviewing today proposes an new approach to address this issue. Go has an exponentially large search space to find moves and evaluate game strategy – The approach we are discussing today is based on Convolutional Neural Networks which utilizes the depth (game length) and breadth (number of legal moves per position) as opposed to current implementations that are based on on Monte Carlo search tree (MCST) that are enhanced by policies trained to predict human expert moves These policies are used to narrow the search space. This method produces decent games. However this produces shallow policies or value functions based on linear combinations of input features.

AlphaGo managed to address this by using deep neural networks and MTCS. The deep neural networks are composed of two parts, a policy network to handle move selections and a value network to evaluate board positions. To efficiently exploit MCTS with deep neural network AlphaGo uses CPUs and GPUs. The simulation are executed in an asynchronous multi-thread search on CPUs. The policy and value networks are computed in parallel GPUs. The final version of AlphaGo uses 40 search threads, 48 CPUs and 8 GPUs.

This implantation has proved to be superior the old approach – it also managed to beat top Go players in the world. The neural networks were trained from gameplay through general-purpose supervised and reinforcement learning method avoiding the construction of handcrafted evaluation function as for instance Deep Blue.