TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2020

AlphaZero

The Value and Policy Networks

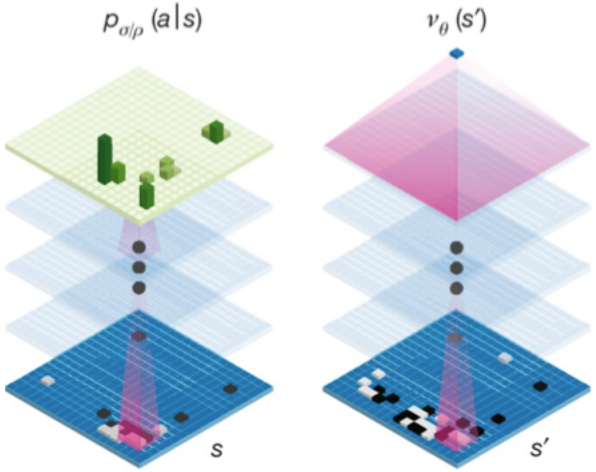
The major innovation of AlphaGo and AlphaZero is to use CNNs as evaluation functions.

We have a policy network computing $\pi_\Phi(a|s)$ and a value network computing $V_\Phi(s)$.

The Value and Policy Networks

$\pi_\Phi(a|s)$ Policy network

$V_\Phi(s)$ Value network



In AlphaZero the networks are either 20 block or 40 block ResNets and either separate networks or one dual-headed network.

Top Level Algorithm

To play a game (against yourself) select a move at each position.

Each move selection involves growing a tree of possible future positions.

The tree is "sparse" in the sense that the children of a node are a (possibly empty) subset of the possible next positions.

The tree is grown by a series of "simulations".

Each simulation starts at the root and recursively selects a move at each node until the selected move adds a new node to the tree.

Simulations

During a simulation moves are selected by maximizing an upper value as in the UCT algorithm.

Each simulation adds one new node to the tree.

Each simulation returns the value $V_\Phi(s)$ (from the value network) for the newly generated node.

The Data Structures

Each node in the tree data structrure stores the following information which is initialized by running the value and policy networks on state s.

- $V_\Phi(s)$ — the value network value for the position s.
- The policy probabilities $\pi_\Phi(a|s)$ for each legal action a.
- The number $N(s, a)$ of simulations that have tried move a from s. This is initially zero.
- The average $\hat{\mu}(s, a)$ of $V_\Phi(s)$ and the values of the simulations that have tried move a from position s.

Simulations and Upper Confidence Bounds

In descending into the tree, a simulation selects the move $\text{argmax}_a U(s,a)$ where we have

$$U(s,a) = \hat{\mu}(s,a) + \lambda_u \, \pi_\Phi(a|s)/(1+N(s,a))$$

We have that $U(s,a)$ will typically decrease as $N(s,a)$ increases.

We can think of $U(s,a)$ as an upper confidence bound in the UCT algorithm.

Root Action Selection

When the search is completed, we must select a move from the root position to make actual progress in the game. For this we use a post-search stochastic policy

$$\pi_{s_{\text{root}}}(a) \propto N(s_{\text{root}}, a)^{\beta}$$

where β is a temperature hyperparameter.

Constructing a Replay Buffer

We run a large number of games.

We construct a replay buffer of triples (s, π_s, R) where

- s is a position encountered in a game and hence a root position of a tree search.
- π_s is the distribution on a defined by $P(a) \propto N(s,a)^{\beta}$.
- $R \in \{-1, 1\}$ is the final outcome of the game for the player whoes move it is at position s.

The Loss Function

Training is done by SGD on the following loss function.

$$\Phi^* \;=\; \underset{\Phi}{\operatorname{argmin}}\; E_{(s,\pi,R)\sim\text{Replay},\, a\sim\pi} \begin{pmatrix} (V_\Phi(s) - R)^2 \\ -\lambda_\pi \log \pi_\Phi(a|s) \\ +\lambda_R \,||\Phi||^2 \end{pmatrix}$$

The replay buffer is periodically updated with new self-play games.

The AlphaZero Algorithm

The AlphaZero algorithm is a general RL learning method. It can be applied to any problem of sequential decision making.

In principle, the AlphaZero algorithm could be applied to the problem of direct optimization of the BLEU score in machine translation.

END