TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2020

What About alpha-beta?

Grand Unification

AlphaZero unifies chess and go algorithms.

This unification of intuition (go) and calculation (chess) is surprising.

This unification grew out of go algorithms.

But are the algorithmic insights of chess algorithms really irrelevant?

Chess Background

The first min-max computer chess program was described by Claude Shannon in 1950.

Alpha-beta pruning was invented by various people independently, including John McCarthy in the late 1950s.

Alpha-beta has been the cornerstone of all chess algorithms until AlphaZero.

Alpha-Beta Pruning

```
def MaxValue(s,alpha,beta):
    value = alpha
    for s2 in s.children():
      value = max(value, MinValue(s2,value,beta))
      if value >= beta: break()
    return value

def MinValue(s,alpha,beta):
    value = beta
    for s2 in s.children():
      value = min(value, MaxValue(s2,alpha,value))
      if value <= alpha: break()
    return value
```

Strategies

An optimal alpha-beta tree is the union of a root-player strategy and an opponent strategy.

A strategy for the root player is a selection of a single action for each root-player move and a response for each possible action of the opponent.

A strategy for the opponent is a selection of a single action for each opponent move and a response for each possible action of the root player.

Proposal

Simulations should be divided into root-player strategy simulations and opponent strategy simulations.

A root-player strategy simulation is optimistic for the root player and pessimistic for the opponent.

An opponent strategy simulation is optimistic for the opponent player and pessimistic for the root-player.

Proposal

$$U(s,a) = \begin{cases} \lambda_u \, \pi_\Phi(s,a) & \text{if } N(s,a) = 0 \\ \hat{\mu}(s,a) + \lambda_u \, \pi_\Phi(s,a)/N(s,a) & \text{otherwise} \end{cases} \quad (1)$$

λ_u should be divided into λ_u^+ and λ_u^- with $\lambda_u^+ > \lambda_u^-$.

Simulations should be divided into two types — optimistic and pessimistic.

In optimistic simulations we use λ_u^+ for root-player moves and λ_u^- for opponent moves.

In pessimistic simulations we use λ_u^- for root-player moves and λ_u^+ for opponent moves.

END