Reinforcement Learning for Games

Timothy Lillicrap & Blake Richards

Tutorial #2
Train a value function

What moves are good?

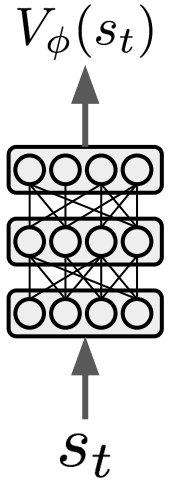
Goal: Learn how to train a value function from a dataset of games played by an expert.

Train a value function

Our value function $V(s_t)$ returns a number between -1 and 1, representing the predicted outcome of the game:
-1 is a loss, 1 is a win from the current player's perspective.

We put the linear output of the network through the tanh() function to bound it's outputs between [-1, 1].

We train via: $\nabla_\phi \mathcal{L} = \sum_{t=0}^{T} \nabla_\phi (R_t - V_\phi(s_t))^2$



Exercise

- Load a dataset of expert generated games.
- Train a network to minimize MSE for win/loss predictions given board states sampled from the dataset.

Tutorial #3
Play games using a value function

Beat the random player!

Goal: Learn how to use a value function in order to create a player that works better than random.

Use the value function to choose an action

Using a model of the environment, value functions can be used to rank potential actions. Then, actions can be chosen according to their rank. For example:

for i in 1 to k:

Choose a random legal action a^i for the current state s_t.
Step the environment: s_{t+1} = env.step(s_t, a^i)
Estimate the value: V(s_{t+1})
Build an array of [V(s_{t+1}), a^i] pairs.

To act, choose the action associated with the highest value.

Exercise

- Sample some random moves and use the value function to rank them.
- Choose the best move as the action and play it.
- Show that doing so beats the random player.