Artificial Neural Networks (Gerstner). Exercises for week 12

Model-Based Deep Reinforcement Learning

Exercise 1. Exploration in MCTS

In this exercise we will have a closer look at the exploration term in Monte Carlo Tree Search. Assume that in state s_0 there are two actions a_1 and a_2. Their true values would be $\mu(s_0, a_1) = 0.8$ and $\mu(s_0, a_2) = 2/3$, but this is unknown to the learner. So far, in the expansion of the tree in MCTS, each action was taken 5 times. By an unlucky coincidence, action a_1 always led to a loss, whereas action a_2 always led to a win, i.e. the current Q-values are $Q(s_0, a_1) = 0$ and $Q(s_0, a_2) = 1$.

a. Assume that the exploration term is of the form $\sqrt{\frac{2}{N(s,a)}}$, i.e. it decreases with visitation counts but it does not increase with the total number of MCTS iterations. Show that in this case the action selection of MCTS $a^* = \arg\max_a Q(s,a) + \sqrt{\frac{2}{N(s,a)}}$ never considers action a_1 anymore and thus fails to discover that action a_1 would actually be better.

b. Let us now use the UCB1 exploration term $\sqrt{\frac{2\log(N(s))}{N(s,a)}}$ and assume that every third time a_2 is taken, a loss results and otherwise a win. After how many iterations will MCTS explore again action a_1?

Exercise 2. AlphaZero.

In this exercise you will manually compute some steps in one iteration of AlphaZero's Monte Carlo Tree Search. Assume that from the previous Monte Carlo Tree Search you have already a tree that starts at state s_0 with $N(s_0, a_1) = 8, W(s_0, a_1) = 4, P(s_0, a_1) = 0.2$, $N(s_0, a_2) = 24, W(s_0, a_2) = 16, P(s_0, a_2) = 0.8$. We fix $C(s) = 1$.

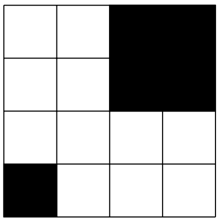
a. Determine the action (a_1 or a_2) that AlphaZero would take in the selection step of MCTS in state s_0.

b. Is it a greedy or an exploratory action that was taken in a?

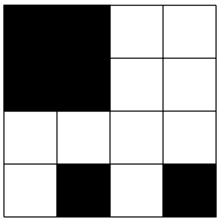
c. Update N, P, W and Q (if it changes in the backpropagation step of MCTS) under the assumption that the expansion step led to $v = 0.7$.

d. Compute the probability that AlphaZero would take the actual action a_1 now.

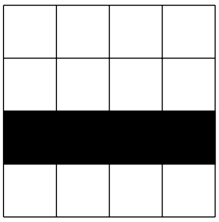
Exercise 3. MuZero



$o_1, \dots, o_{16}$



$o_{17}, \dots, o_{32}$



$o_{33}, \dots, o_{48}$

Above you see some example images of an environment, where states $o_1, \dots, o_{16}$ always have the black rectangle at the top right, while each pixel in the bottom row can be randomly 0 or 1. States $o_{17}, \dots, o_{32}$ have a similar pattern with a black rectangle at the top left and random pixels in the bottom row. States $o_{33}, \dots, o_{48}$ have a similar pattern with a bar in the second row from the bottom and random pixels in the bottom row. Assume the actual state transitions are $P^{a_1}_{o_i \to o_j} = 1/16$ for $i \in \{1, \dots, 16\}, j \in \{17, \dots, 32\}$, $P^{a_1}_{o_i \to o_j} = 1/16$ for $i \in \{17, \dots, 32\}, j \in \{33, \dots, 48\}$, $P^{a_1}_{o_i \to o_j} = 1/16$ for $i \in \{33, \dots, 48\}, j \in \{1, \dots, 16\}$ and $P^{a_2}_{o_i \to o_j} = P^{a_1}_{o_j \to o_i}$. The rewards are $R^{a_1}_{o_i \to o_j} = 1$ for $i \in \{1, \dots, 16\}, j \in \{17, \dots, 32\}$, all other rewards are zero. Episodes start in a random state and end after 10 actions have been taken.

a. How many bits does the latent representation s_t need to have at least for a model-based RL method that relies on a auto-encoder approach, where o_t has to be reconstructable from s_t?

b. How many bits does the latent representation s_t need to have at least for a model-based RL method like MuZero, where the latent state only needs to be sufficient for predicting the immediate reward, the value and the policy?

c. Can MuZero still find the optimal policy, if $P^{a_2}_{o_i \to o_j} = 1/32$ for $i \in \{17, \dots, 32\}, j \in \{1, \dots, 16, 33, \dots 48\}$?