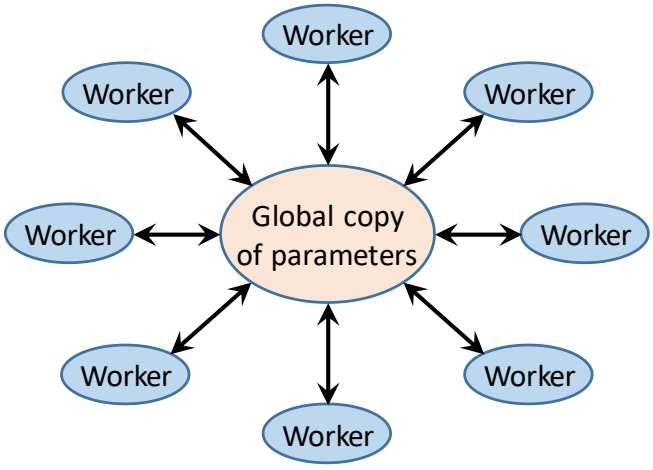
Predictive Analytics World / Deep Learning World Exercises – Deep RL

1. Create a graphical 7x7 maze RL environment with horizontal and vertical walls. Define the observation as a **list of four integers**, where a 1 indicates the presence of a wall in one of the four directions, and a 0 indicates the absence of a wall. Also include the previous action and reward in each observation. Define actions that try to move in each of the 4 directions. Let the user enter each action manually. Define one goal state. Moving away from the goal state should give a reward of 1.0, and should place the agent at a random location. Output each observation and reward.

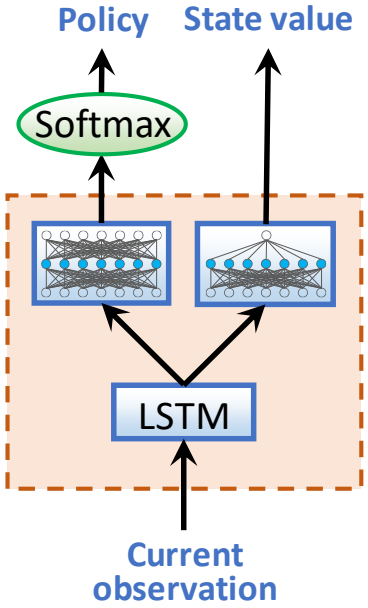
2. Connect a random agent to the environment. Output the mean reward received over 10,000 steps. Disable the graphical display for fast evaluation.

3. Modify the agent to use the A3C Policy Gradient method, as illustrated below. Reevaluate.

Asynchronous parameter updates



Each agent-worker network



4. Tune the following hyper-parameters to maximize total reward over 100,000 steps:

a. Size of hidden layer (default 64).
b. Number of LSTM units (default 128).
c. Backprop time window length (default 16).
d. Learning rate (default 0.001).
e. Discount rate (default 0.9).

5. Evaluate your best set of hyper-parameters on the random seed provided by the instructor.

6. When are neural networks needed in place of tables?

a. When the actions are continuous rather than discrete.
b. When the number of environment states is very large.
c. When the environment is partially observable.

7. When can RNNs (such as LSTMs) yield much better results than feed-forward NNs?

a. When the actions are continuous rather than discrete.
b. When the number of environment states is very large.
c. When the environment is partially observable.

8. When are policy gradient methods more appropriate than Q-learning?

a. When the actions are continuous rather than discrete.
b. When the number of environment states is very large.
c. When the environment is partially observable.