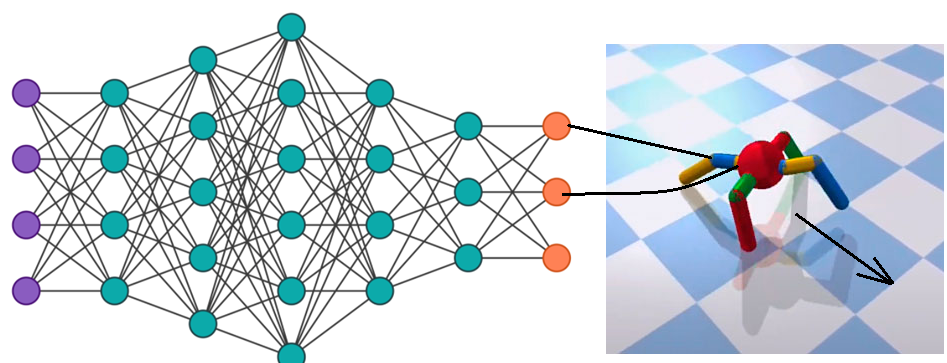
**Introduction**

In this homework you will work with TRPO and PPO algorithms for the sake of training a robot to walk. In fact, algorithms do not care about environment and robot structure. So they could be used to train wide variety of robots. Policy function receives observation as an array of numbers as input and produces actions as output. So virtual environment generates training data on the fly. And there is no need to store a training database.



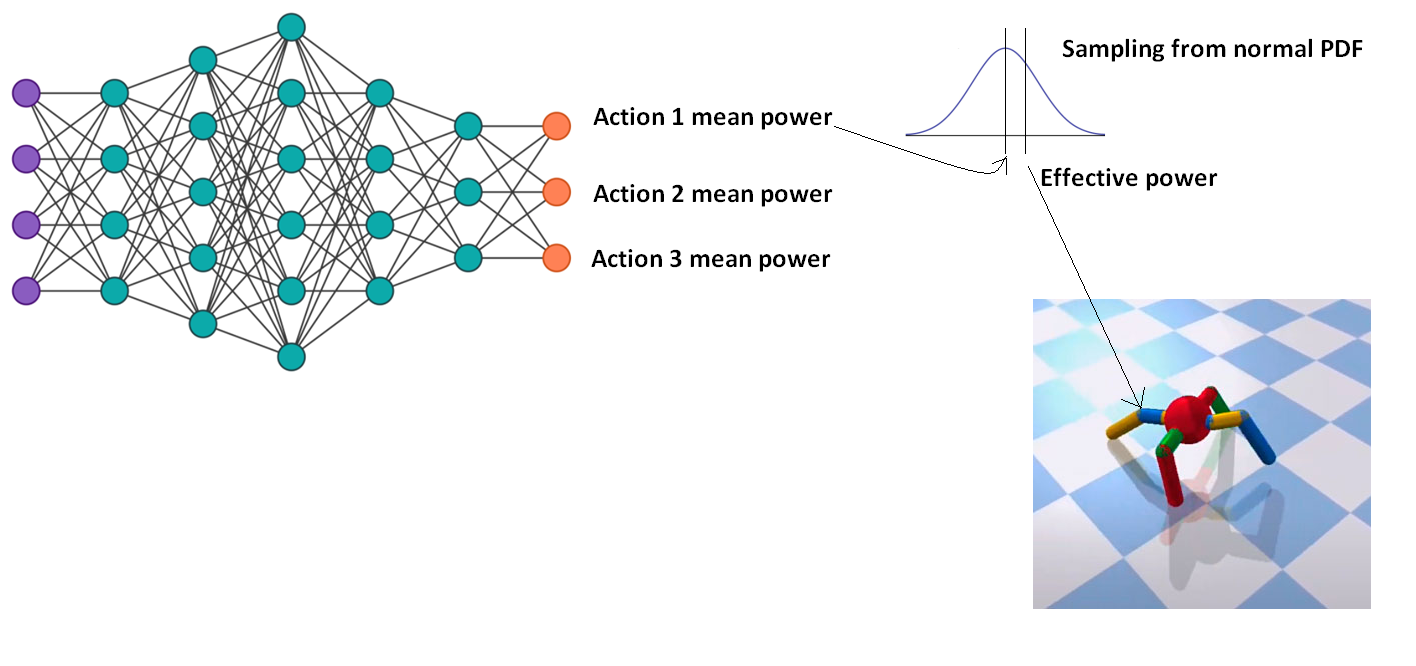
The Ant has 8 joints/actions and 28 observations. Observation includes ant’s position, velocity and torque. The reward function has several terms like forward velocity and control cost. It has a positive reward signal for the pelvic velocity, a negative reward signal for the effort of the current action state, and a negative signal if the joints are at their limit.

You may refer source code in order to get more details https://github.com/openai/gym/blob/master/gym/envs/mujoco/ant.py

**Action sampling.**

Policy function outputs just mean power values. Effective power values obtained from there by sampling from normal probability distribution (PDF). So effective values are assigned using random numbers. This step emulates vibrations in whole system. Vibrations also help to jump over local maximums during gradient descent. So this noise improves the convergence process.

Along with mean value the Normal PDF has also a dispersion value. Dispersion appears either to vary or to be constant during code execution. Its depends on implementation. You may investigate source code in order to catch this point. Just investigate source code and answer if dispersion if constant or not.



**Policy update**

Common algorithms obtain trajectories (array of states and actions) using current policy. Next it makes gradient steps in order to strengthen good actions and to weaken bad actions. So during the next iterations good actions become more probable and bad actions become less probable and average reward goes up.

**Running the notebook**

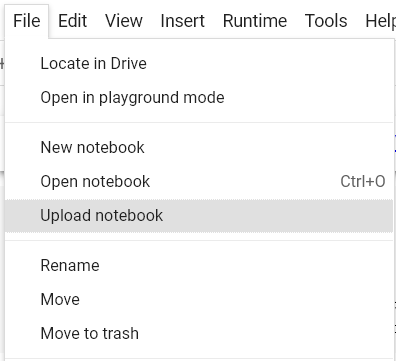
You may use one of the following ways to run the notebook

1) use the following url in order to access this notebook

<https://colab.research.google.com/drive/1hqjxgn6cXwq6e8Ndi9VFozZlE1A31Mg4?usp=sharing>

2) upload attached .ipynb file into the google collab and execute it

1. Open archive
2. Extract .ipynb file
3. Open google collab
4. Upload file to the google collab using the following menu item:



**3) open file in local Jupiter**

1. For windows just run install\_and\_run\_for\_windows.bat

**Scope of work**

1. Make the notebook runnable and edit input parameters at the beginning.
2. Investigate TPRO. It is too complex to understand. So you could only review it.
3. Add code to PPO. It is affordable to understand and modify it.

**Completion criteria**

Attain reward at least 600 and ensure that robot goes forward.