CS427:Reinforcement Learning **(Date: 13-Apr-2022)**

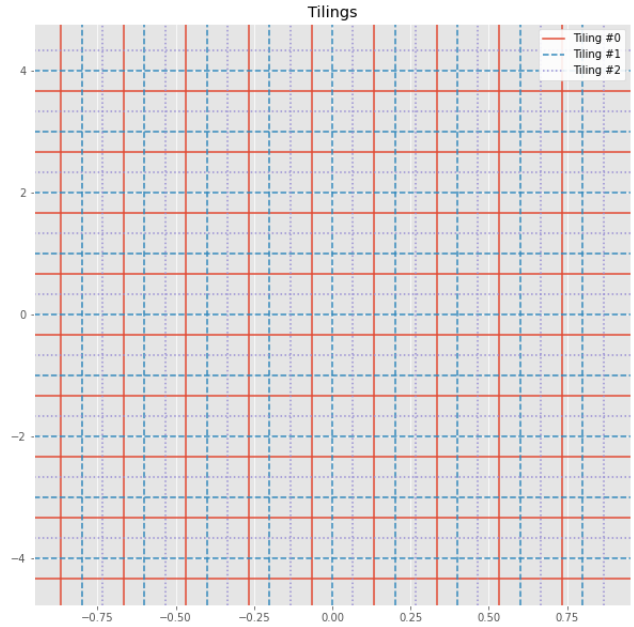
Assignment 5

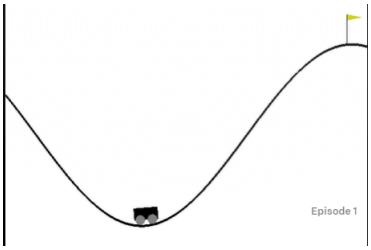
Instructor: Dr. Prabuchandran K J *Rollno.:* 211022005, 211022001

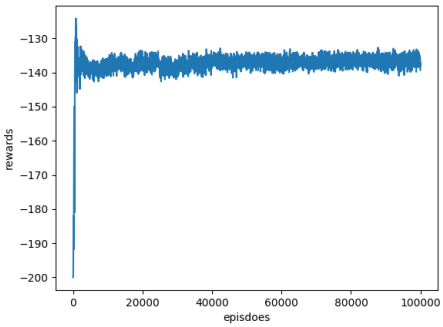
1 Multi-Tile Coding

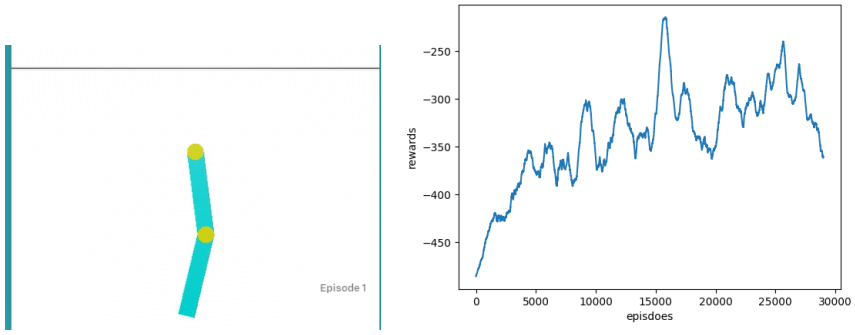
Environments Tested: MountainCar, Acrobot

Multi-Tile Coding with offset: This is a method for discretizing the state space. It involves taking multiple grids over the continuous state space with each of them offset by certain amount. Depending on where the continuous state falls on multiple grids, cells of each grids are activated. By taking multiple grids we take lower dimension state space to higher dimension, due to this regression task could be simplified. In our algorithm we took 2 grids each of 10×10 with an offsets $(-0.066, -0.33), (0, 0), (0.066, 0.33)$ along it's dimension with $low = (-1, -5)$ and $high = (1, 5)$.



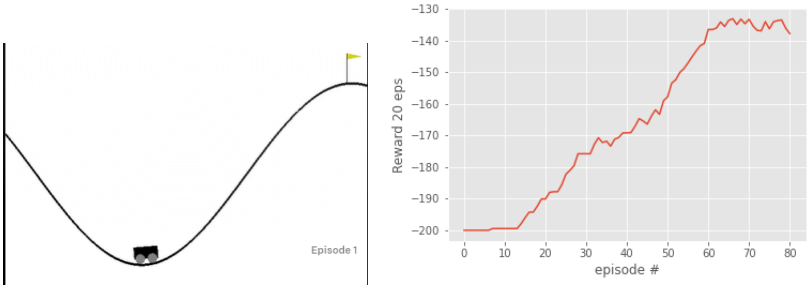




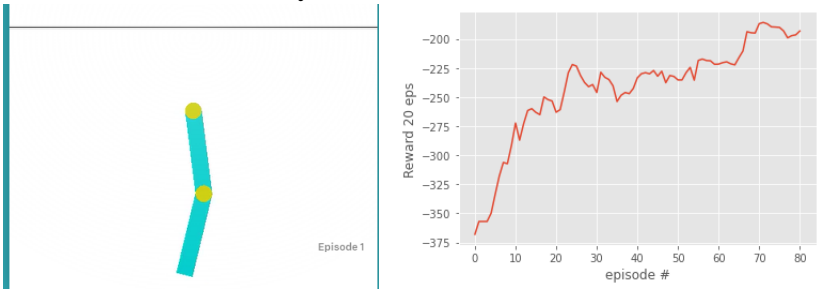


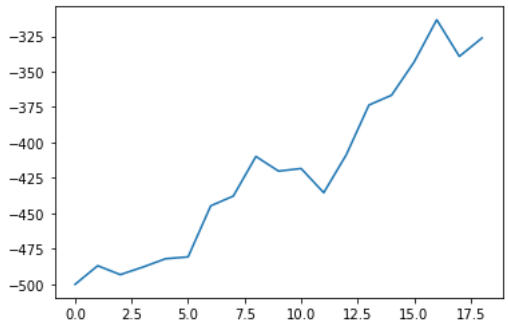
2 Radial Basis Function

Environments: Mountain Car, Acrobot
In Radial Basis Function (RBF), One or more Gaussian kernels are used to sample points for a single data point all with certain weights. And after learning the weights regression problem is solved. **Specs:** We used 4 RBF kernels with $\gamma = 5, 2, 1, 0.5$ and for each original data 100 samples are sampled. Before fitting an RBF Kernel, data was standard-scaled such that mean is zero and variance is unity.
Following are average rewards vs episodes in Mountain car and Acrobot respectively.
Mountain-Car RBF Q learning



ACROBOT RBF Q Learn





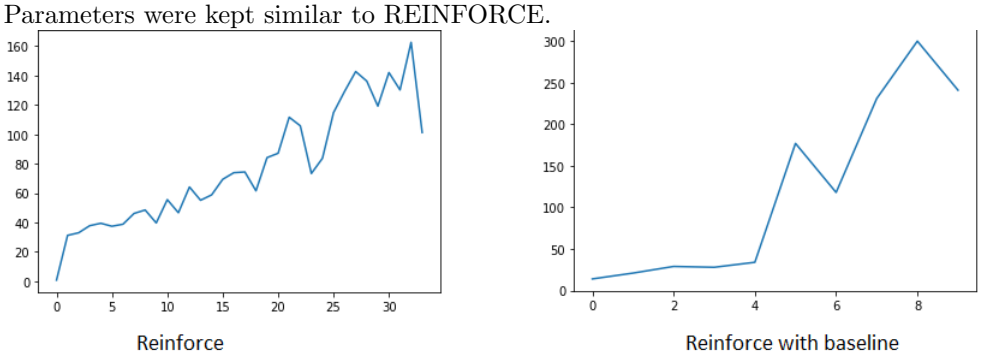
Acrobot Reinforce

3 REINFORCE

Environments: CartPole, Acrobot In this algorithm there is a neural network trained to give best policy. Neural Network consisted of input-layer, single hidden layer followed by output layer. The number of inputs and outputs depends on observation space and action space of environment. **Hyper-Parameters:** $\gamma = 0.99$, $\alpha = 0.001$ **Loss for s,a pair:**$-log(\pi_\theta(a|s)) * r(s,a)$. **Rewards have been averaged over an interval of 50 episodes**

Acrobot Problem The acrobot system includes two joints and two links, where the joint between the two links is actuated. Initially, the links are hanging downwards, and the goal is to swing the end of the lower link up to a given height.
The observation is a numpy array‘ with shape ‘(6,)‘ that provides information about the two rotational joint angles as well as their angular velocities. The Action space is to apply ±1 or 0 torque. Reward for reaching the top is 0, and penalised by -1 for every step not reaching. Terminal reward is -100.

REINFORCE with Baseline Environments: CartPole In this algorithm two neural networks are trained to estimate policy and value function. It is supposed to be faster than reinforce. Parameters were kept similar to REINFORCE.



Cartpole Comparison

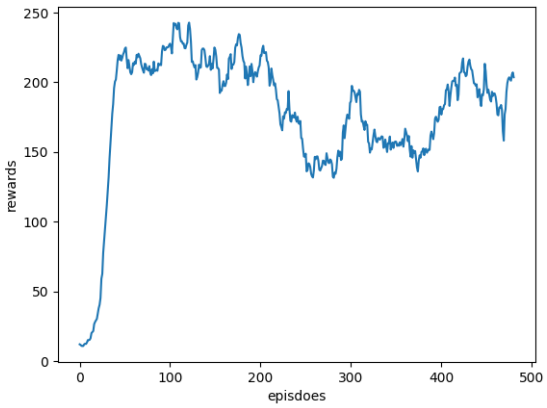
4 Deep Q Network

Env: Cartpole

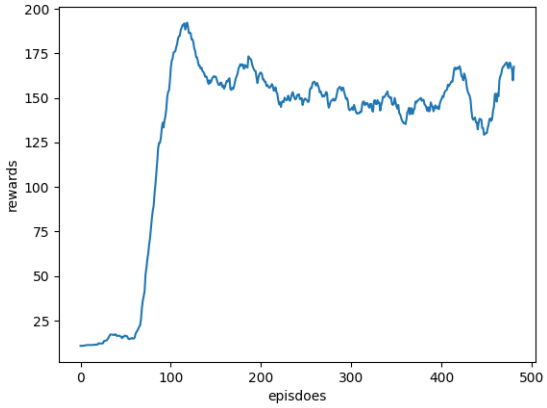
Hyperparameters:

- Optimizer: Adam
- Relu activated but output Linear activated
- Loss: MSE
- Learning rate:0.01
- discount: 0.9
- epsilon: 0.3 with decay 0.99

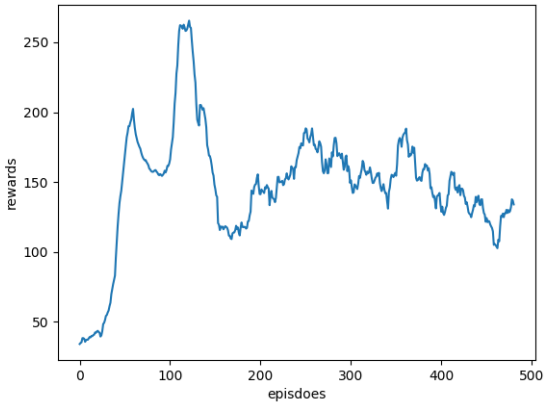
64 Nodes



24 Nodes



8 Nodes



We compared DQN with 64,24,8 hidden nodes on cartpole environment with 500 episodes.

5 Actor Critic

Environment: Cartpole
Actor Hyperparameters

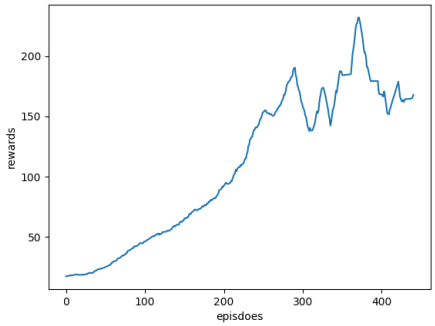
- Optimizer: Adam
- Relu activated but output softmax activated
- Loss: crossentropy
- Learning rate:0.001
- discount: 0.99

Critic Hyperparameters

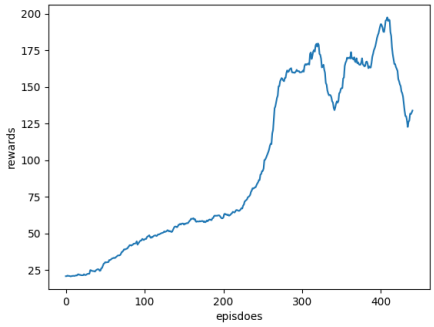
- Optimizer: Adam
- Relu activated but output linear activated
- Loss: MSE
- Learning rate:0.005
- discount: 0.99

Experimented on number of nodes in hidden layer Hypermaters: Actor's $\alpha = 0.001$**, Critic's** $\alpha = 0.005$.
Actor loss: $-\sum_{t=1}^{T} log\pi_\theta(a_t|s_t)[G_w(s_t, a_t) - V_\theta(s_t)]$
Critic Loss: $(G_w - V_\theta^\pi)^2$

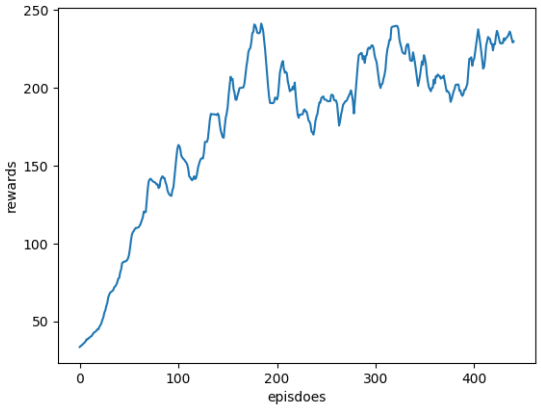
24 Nodes



8 Nodes



64 Nodes



Observation

It looks like A2C converges faster compared to DQN in these tasks.
For the given Environment(CartPole) the model with 64 neurons in hidden layer performs better than 24 and 8 but training is slow.