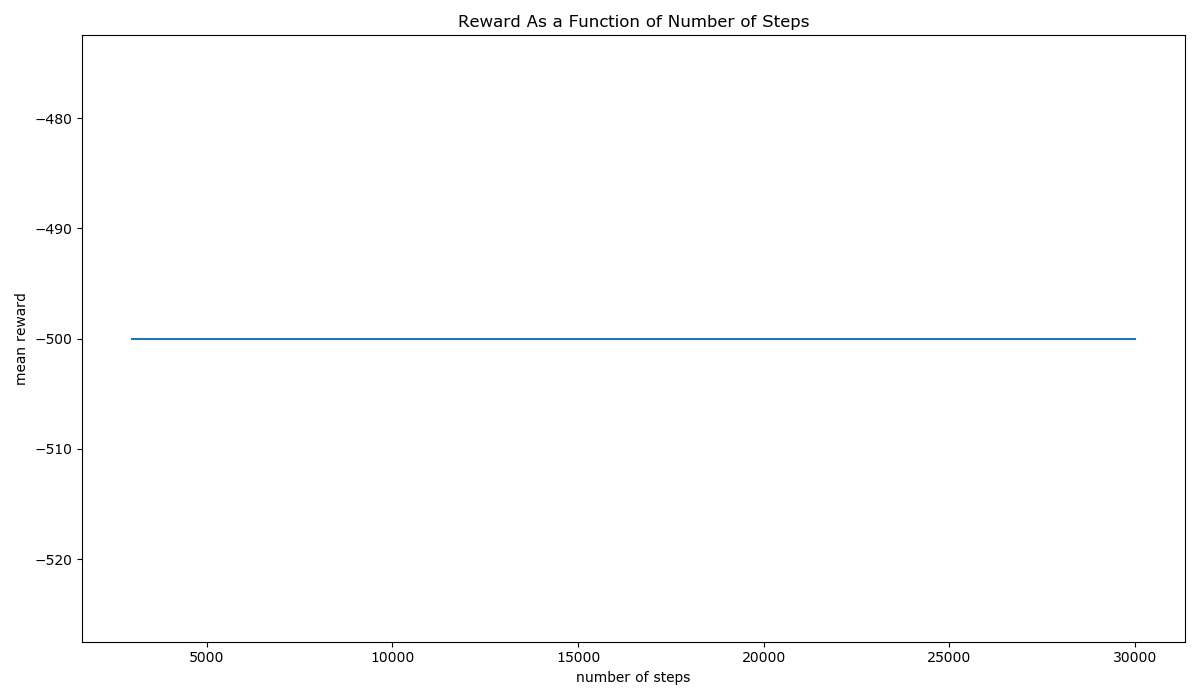
**Reinforcement Learning Assignment 4 Report**

**Students:** Itay Osovlanski (311129274)  
 Tomer Laor (206783813)

The code is solving the Mountain Car issue using Actor-Critic and function approximation.  
The features are 2D radial basis functions.  
It runs a simulation using a pre-computed weights, then it computes weights for the current run and outputs a plot of the reward as function on the steps.

**Main methods:**  
actor\_critic: This function returns W, the best weights that it found, and policy evaluations along checkpoints.  
This method is implemented using the description of Actor-Critic in David Silver's lecture (lecture 7 slide number 25).  
For each episode we initialize the state, get the features for them and pick a first action. Then, for each episode we sample a reward and the next state from doing the action that we previously picked. We sample a new action from phi\_theta(s, a), which is a softmax policy, and compute delta.  
We compute theta using policy gradient and update the weights accordingly.  
softmax\_policy: Returns the softmax policy of s, a. (computes phi\_theta(s, a))  
get\_features\_actor: Returns phi(s, a)  
get\_features\_critic: Returns phi(s)



Unfortunately, it didn't learn how to reach the top. We would be happy to get feedback about what we did wrong.  
We've also tried to use the example in Tensorflow's documentation for Actor-Critic to solve the problem, and it didn't learn how to reach the top as well. The notebook of our try is here: (This code is not a part of our submission, but it's a way for us to show that we've tried different things) <https://colab.research.google.com/drive/1olkTg2hN5pch2NzulbjwORmjYSqY3ZK9?usp=sharing> and the link to TF's documentation is: <https://www.tensorflow.org/tutorials/reinforcement_learning/actor_critic?hl=en>