313326985 Shahar Shcheranski
206172686 Sarit Hollander

Assignment 3

Meta and Transfer Learning

Section 1 – Training individual networks

CartPole

Here we ran the CartPole environment again using 0 to pad the input (to get input size of 6) and "empty" actions (to get action space of 3) to make the transfer learning task easier.

We implement the actor-critic method to solve the problem. For the actor network we used a discrete policy estimator that learned a vector of size of the action space. We sampled an action from the Softmax function. The policy estimator is a MLP with one hidden layer with 12 units and sampling head. For the critic network we used a regression network with one hidden layer with 32 units. We used learning rate of 0.0004 for the actor network and learning rate of 0.005 for the critic network.

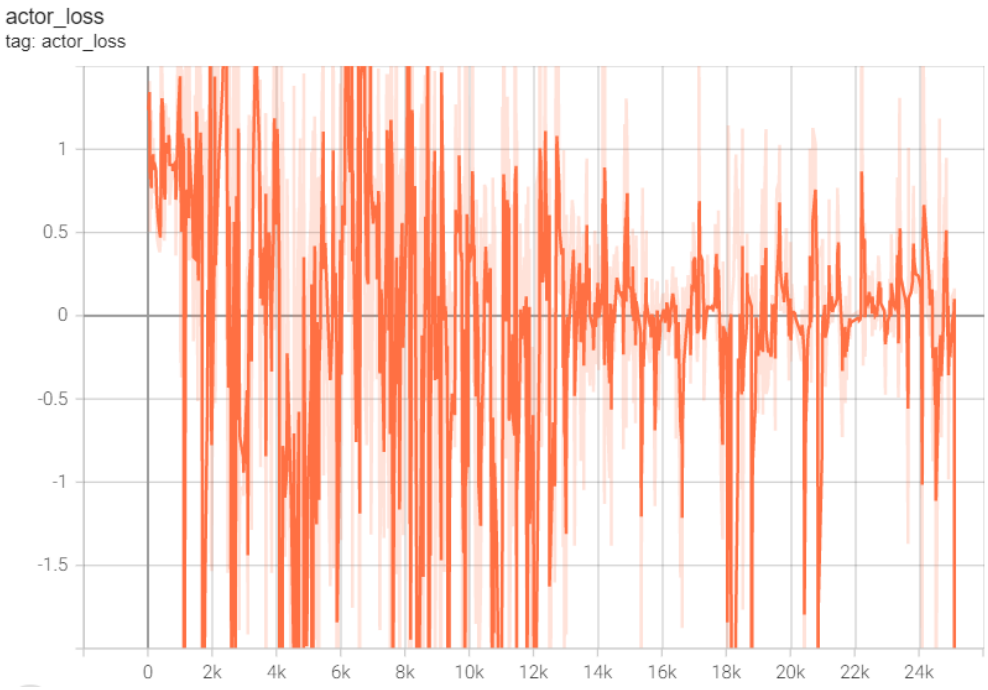
For evaluation, each time the agent achieved an episode with a total reward greater than 475, we run our model for 100 episodes. Our criterion for solving the problem is when the agent achieved an average reward of 475 or greater in the evaluation. Our model successfully finished its first evaluation run with average reward of 500.

Once the algorithm converged, we saved the weights of the network for the later sections.

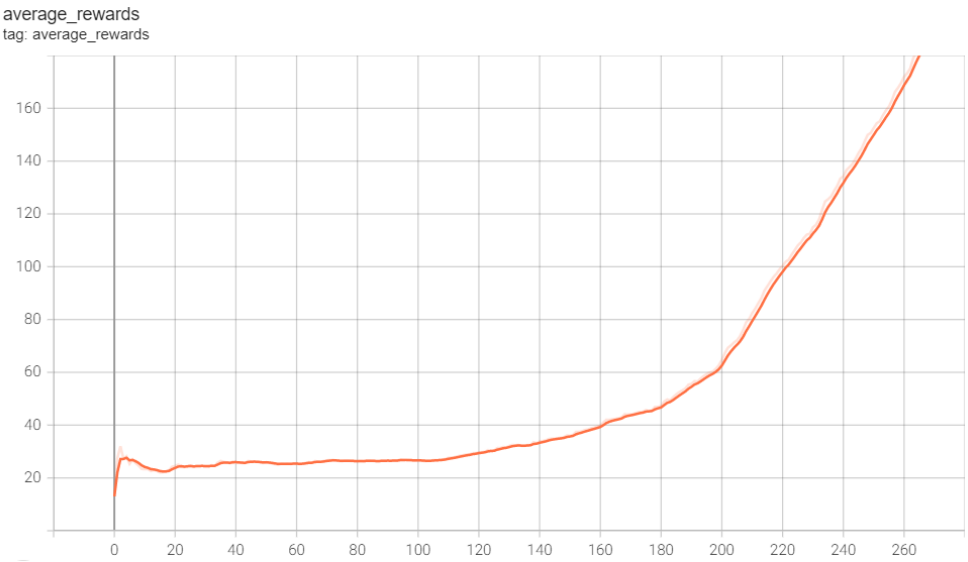
Statistics

Running time: 82.109 seconds.

Number of training iterations for converge: 267 episodes.



313326985 Shahar Shcheranski
206172686 Sarit Hollander



Acrobot

Here we ran the Acrobot environment.

We implement the actor-critic method to solve the problem. For the actor network we used a discrete policy estimator that learned a vector of size of the action space. We sampled an action from the Softmax function. The policy estimator is a MLP with one hidden layer with 12 units and sampling head. For the critic network we used a regression network with one hidden layer with 32 units. We used learning rate of 0.0004 for the actor network and learning rate of 0.005 for the critic network.

For evaluation, each time the agent achieved an episode with a total reward greater than -90, we run our model for 15 episodes. Our criterion for solving the problem is when the agent achieved an average reward of -90 or greater in the evaluation.

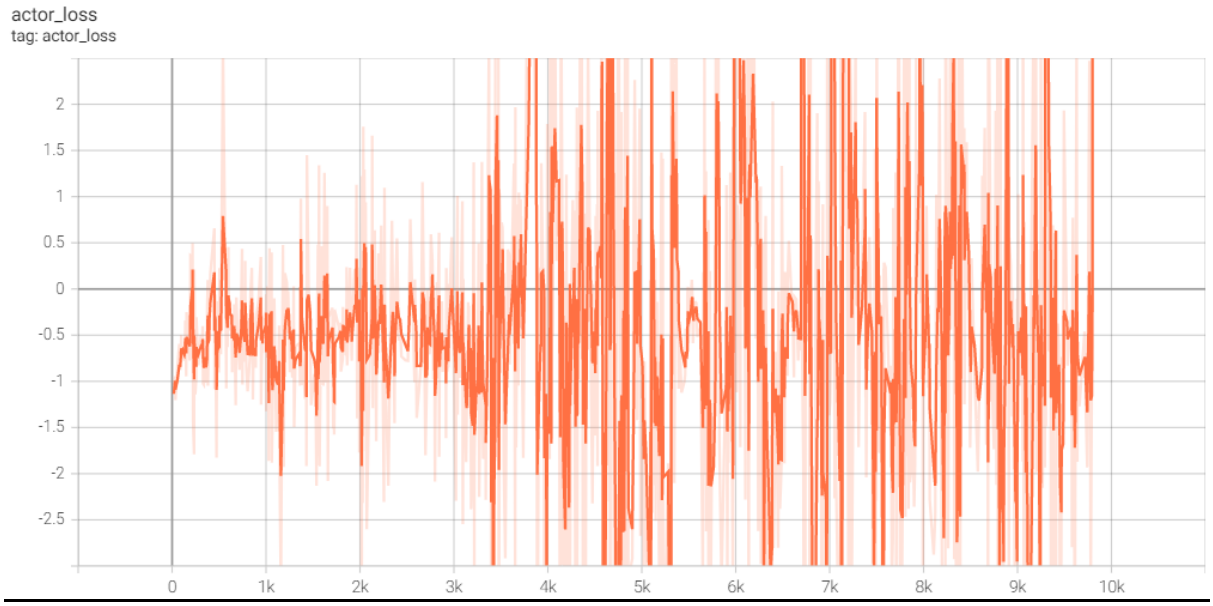
Once the algorithm converged, we saved the weights of the network for the later sections.

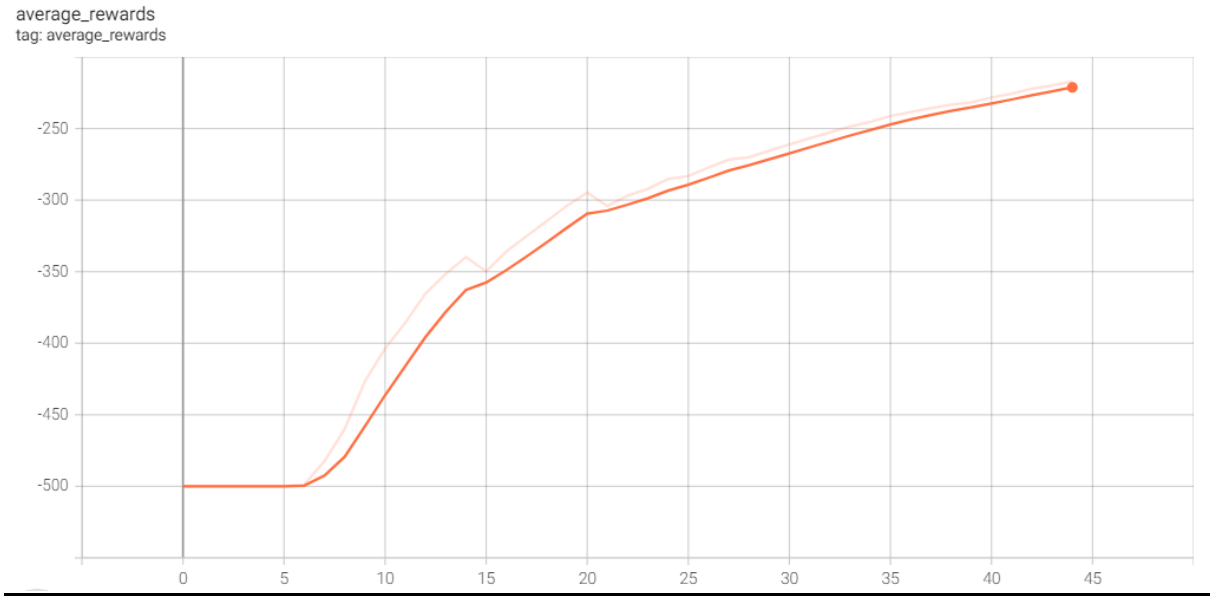
Statistics

Running time: 24.780 seconds.

Number of training iterations for converge: 44 episodes.

313326985 Shahar Shcheranski
206172686 Sarit Hollander





313326985 Shahar Shcheranski
206172686 Sarit Hollander

Mountain Car Continuous

Here we ran the Mountain Car Continuous environment. During the ran we used 0 to pad the input (to get input size of 6) to make the transfer learning task easier.

We implement the actor-critic method to solve the problem. For the actor network we used a continuous policy estimator that learned two parameters μ and σ. We sampled an action from the Normal distribution. The policy estimator is a MLP with two hidden layers with 40 units and sampling head. For the critic network we used a regression network with two hidden layers with 400 units. Also, we replaced the ReLu activation with the ELU activation.

In addition, we standardized the environment using StandardScaler by sampling 10,000 observations from the task space.

We used learning rate of 0.00002 for the actor network and learning rate of 0.001 for the critic network.

For evaluation, each time the agent achieved an episode with a total reward greater than 90, we run our model for 15 episodes. Our criterion for solving the problem is when the agent achieved an average reward of 90 or greater in the evaluation.

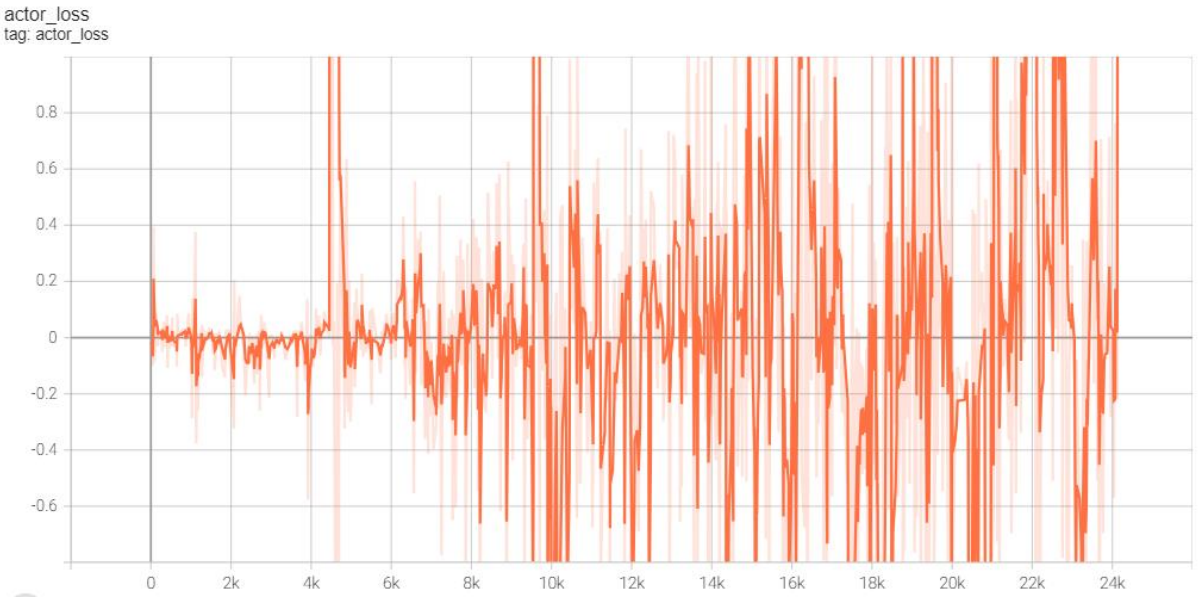
Once the algorithm converged, we saved the weights of the network for the later sections.

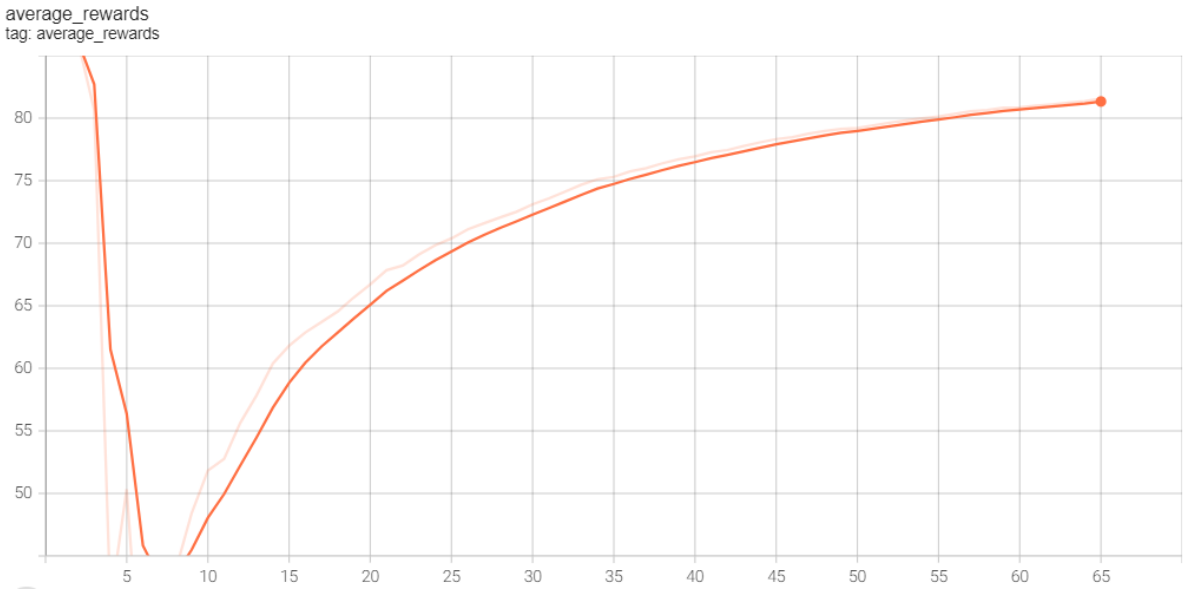
Statistics

Running time: 174.633 seconds.

Number of training iterations for converge: 65 episodes.

313326985 Shahar Shcheranski
206172686 Sarit Hollander





As we can see in the training, our model started with high rewards and then very low rewards and then the model start to improve slowly and get better rewards from episode to episode.

313326985 Shahar Shcheranski
206172686 Sarit Hollander

Section 2 – Fine-tune an existing model

Acrobot -> CartPole

We implement the actor-critic method to solve the problem. For the actor network we used a discrete policy estimator that learned a vector of size of the action space. We sampled an action from the Softmax function. The policy estimator is a MLP with one hidden layer with 12 units initialized with the weights from the hidden layer of the Acrobot model, Dropout layer, another hidden layer with 40 units initialized with GlorotNormal distribution and sampling head. For the critic network we used a regression network with one hidden layer with 32 units initialized with the weights from the hidden layer of the Acrobot model.

In addition, we standardized the environment using StandardScaler by sampling 10,000 observations from the task space.

We used learning rate of 0.00002 for the actor network and learning rate of 0.001 for the critic network.

For evaluation, each time the agent achieved an episode with a total reward greater than 475, we run our model for 100 episodes. Our criterion for solving the problem is when the agent achieved an average reward of 475 or greater in the evaluation.

Unfortunately, we had a really hard time with this task, using the weights from the Acrobot model as a starting point destroyed our model and we didn't converge at all. It could be that we were biased towards values that were implemented and could not learn new values as well as new policies.

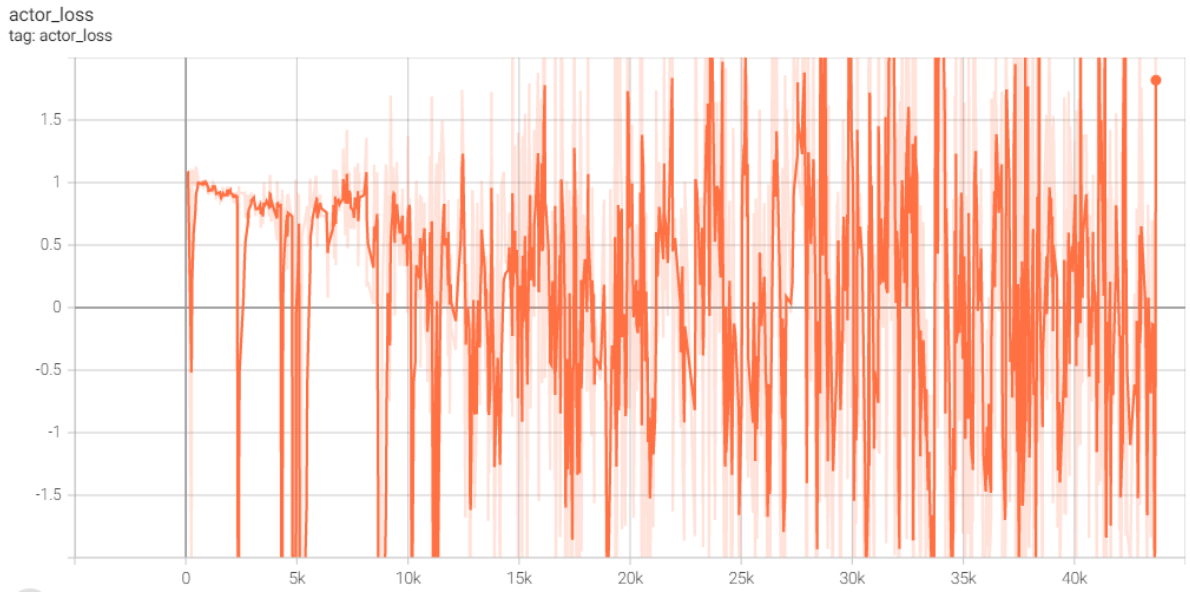
Statistics

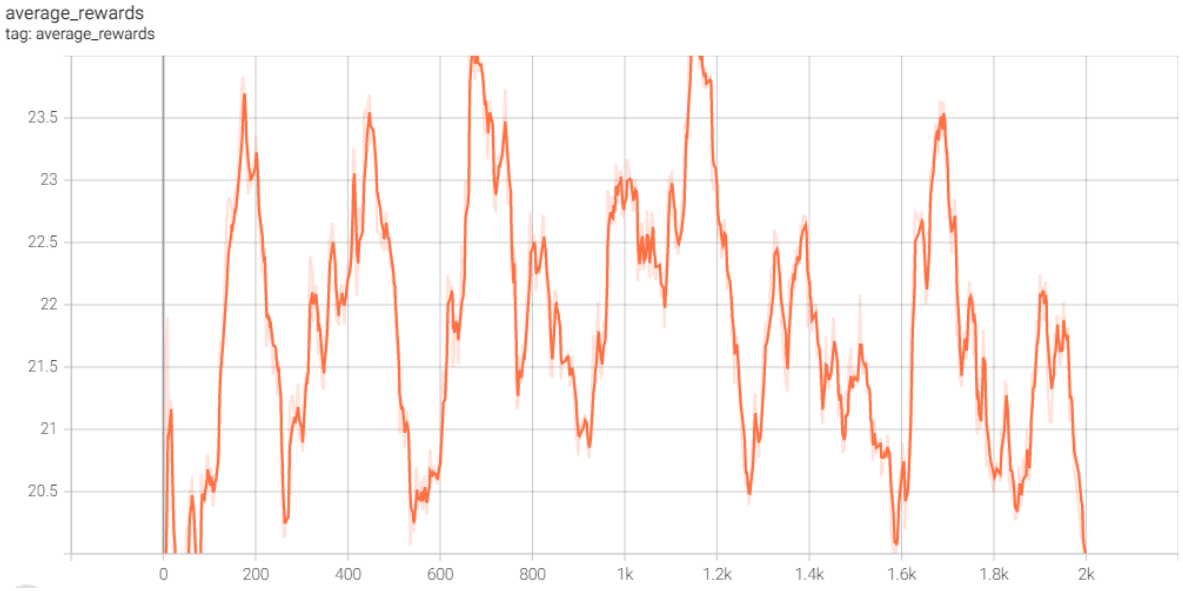
Running time: 118.588 seconds.

Number of training iterations for converge: we did not success to converge, we ran our model for 2000 episodes of training.

	Section 1	**Section 2**
Running time	82.109 seconds	118.588 seconds
Number of training iterations for converge	267 episodes	2000 episodes (no converge)

313326985 Shahar Shcheranski
206172686 Sarit Hollander





CartPole -> Mountain Car Continuous

We implement the actor-critic method to solve the problem. For the actor network we used a continuous policy estimator that learned two parameters μ and σ. We sampled an action from the Normal distribution. The policy estimator is a MLP with one hidden layer with 12 units initialized with the weights from the hidden layer of the CartPole model and sampling head. For the critic network we used a regression network with one hidden layer with 32 units initialized with the weights from the hidden layer of the CartPole model.

We used learning rate of 0.00008 for the actor network and learning rate of 0.004 for the critic network.

For evaluation, each time the agent achieved an episode with a total reward greater than 90, we run our model for 15 episodes. Our criterion for solving the problem is when the agent achieved an average reward of 90 or greater in the evaluation.

313326985 Shahar Shcheranski
206172686 Sarit Hollander

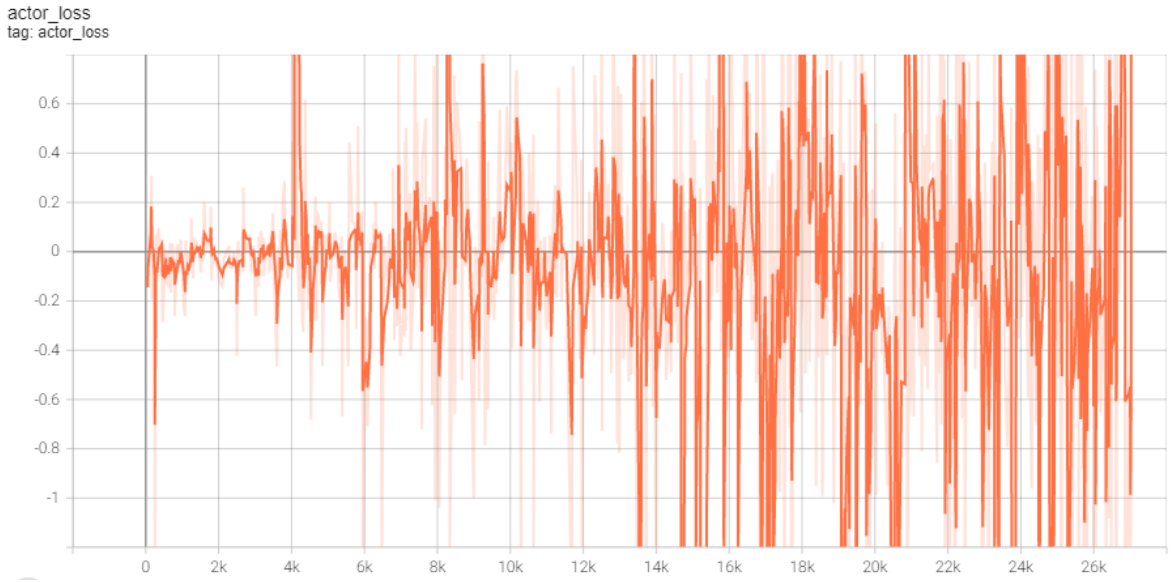
Statistics

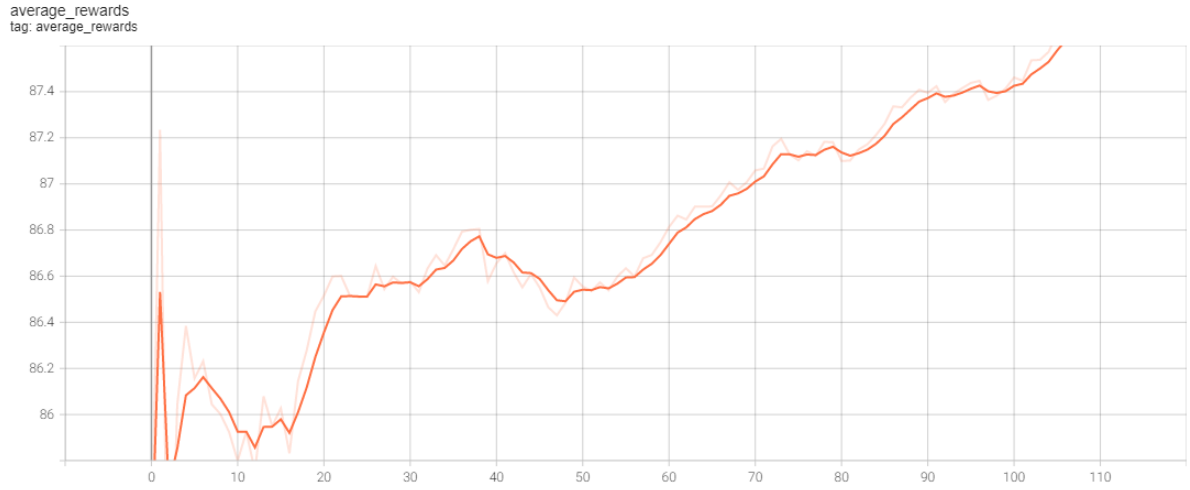
Running time: 190.927 seconds.

Number of training iterations for converge: 109 episodes.

As we can see, comparing the results to those in section 1, using the weights from the CartPole model as a starting point didn't help us to improve the converge time. We think that it took more time, because the CartPole model from section 1 is much simpler than this model.

	Section 1	**Section 2**
Running time	174.633 seconds	190.927 seconds
Number of training iterations for converge	65 episodes	109 episodes





313326985 Shahar Shcheranski
206172686 Sarit Hollander

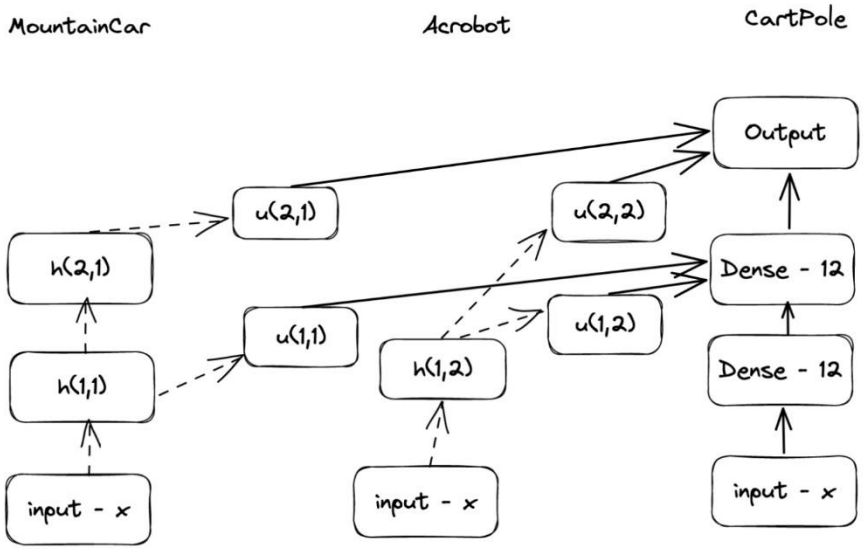
Section 3 – Transfer learning

Acrobot + Mountain Car Continuous -> CartPole

We implemented the actor-critic method to solve this problem. For the actor network we used a discrete policy estimator that learned the size of the action space. We sampled an action from the induced Softmax function. The policy network is built with Progressive Neural Networks, where the two frozen previous models from section 1 are actor network of Acrobot and actor network of MountainCarContinuous. The MountainCarContinuous has two hidden layers which we connected to the two hidden layer output of our actor network using 4 U adapter dense layers, all with 12 units output and ReLU. We saw that the activation that coming out of the adapters are huge, so we manually reduced the signal by multiplying the activations by 0.001. In addition, we added a sampling head. We used a regression network for the critict network with 2 hidden layers of 16 and 8 units with ReLU as activation function. We implemented a Progress Neural Network, where the two frozen models from section 1 are critic network of Acrobot and critic network of CartPole, multiplying the outputs by 0.001. Then we used a dense layer with output of size 1.

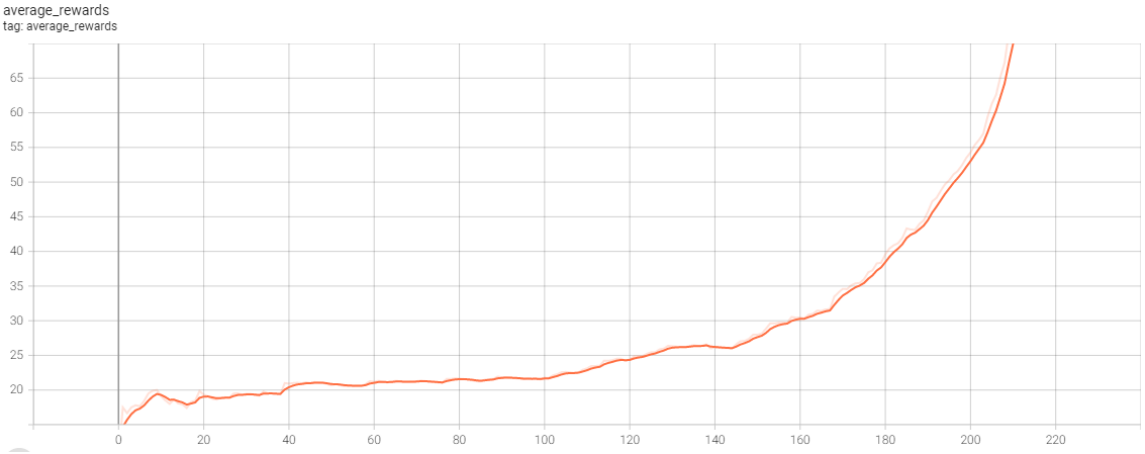
In order to freeze a graph node in TensorFlow version 1.x we used the function stop_gradient which means that if we apply it to a tensor its treated as a constant. So the dense layers that are connected with the dashed line in the actor progressive network figure are donsidered as frozen.

The actor progressive neural network for cartpole:



We used learning rate of 0.0002 for the actor network and learning rate of 0.01 for the critic network.

313326985 Shahar Shcheranski
206172686 Sarit Hollander



Statistics

Running time: 159.082 seconds.

Number of training iterations for converge: 221 episodes.

	Section 1	**Section 2**	**Section 3**
Running time	82.109 seconds	118.588 seconds	159.082 seconds
Number of training iterations for converge	267 episodes	2000 episodes (no converge)	221 episodes

The agent solve the task faster than the 267 episodes it took the original network to solve the task.

CartPole + Acrobot -> Mountain Car Continuous

We implement the actor-critic method to solve the problem. For the actor network we used a continuous policy estimator that learned two parameters μ and σ. We sampled an action from the Normal distribution. The policy network is build using Progressive Neural Networks, where the two frozen previous models are Acrobot actor network from section 1, and CartPole actor network from section 1. Both models have only one hidden layer which we connected to the first hidden layer output of our actor network using two U adapter dense layers all with 40 units output and ELU. We saw that the activation that coming out of the adapters are very big, so we manually reduced the signal by multiplying the activations by 0.001. On top of that we added a sampling head. For the

313326985 Shahar Shcheranski
206172686 Sarit Hollander

critic network we used a regression network with 2 hidden layers of 400 units with ELU as activation function. Same as with the actor network we implemented a Progress Neural Network, with two frozen models Acrobot critic network from section 1, and CartPole critic network from section 1 and multiplied the outputs of the adapter with 0.01. Then a dense layer with output of size 1.

We used learning rate of 0.00001 for the actor network and learning rate of 0.0005 for the critic network.

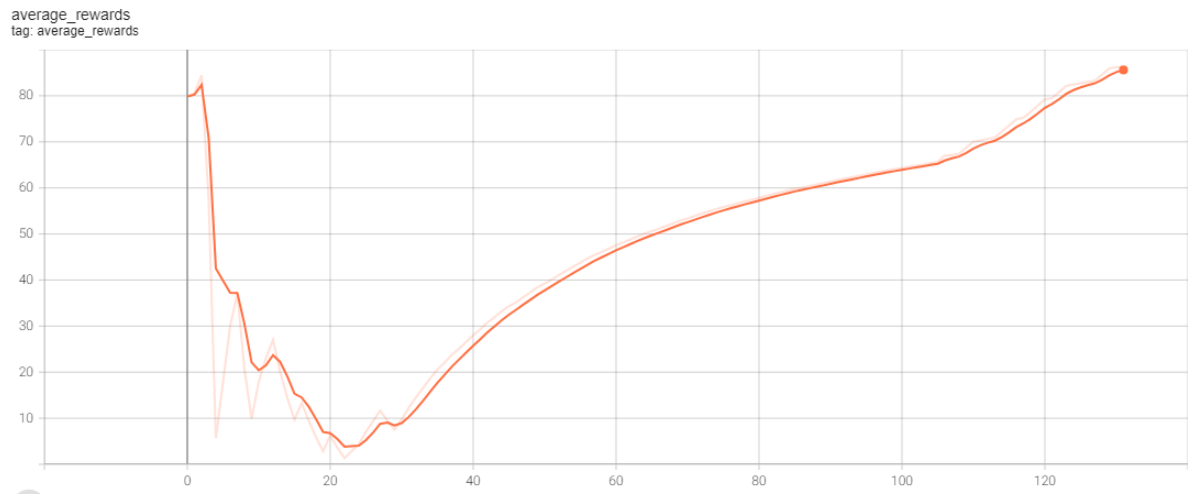
Statistics

Running time: 1147.414 seconds.

Number of training iterations for converge: 131 episodes.

	Section 1	**Section 2**	**Section 3**
Running time	174.633 seconds	190.927 seconds	1147.414 seconds
Number of training iterations for converge	65 episodes	109 episodes	131 episodes

As we can see, comparing the results to those in section 2, the transfer learning didn't help us to improve the converge time.



313326985 Shahar Shcheranski
206172686 Sarit Hollander