Armando Mendez

Deep Q-Network for Cartpole Report

The initial neural network (used in trials 1-7) is comprised of 3 layers, with 4 nodes on the input layer, 100 on the second, and 2 on the output. I tested with differing node amounts for the inner layer ranging from 50-150. In the end I decided on the following network architecture:

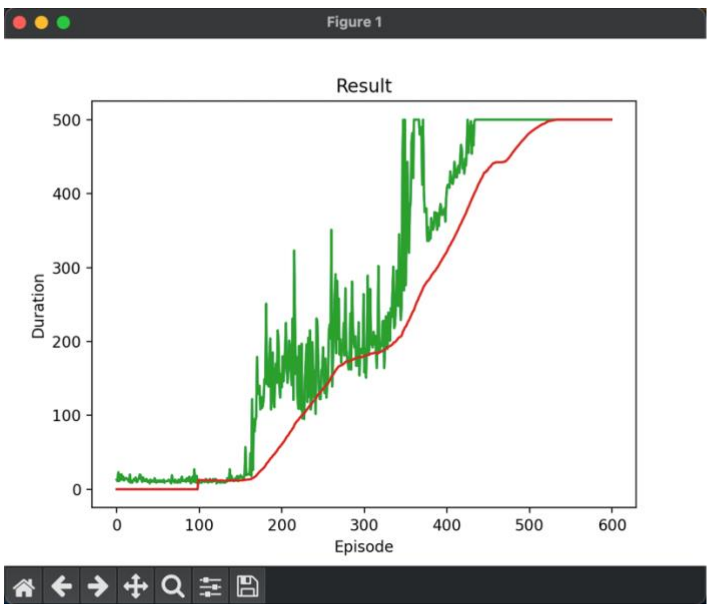
NN Architecture: layer 1: (4, 125) | layer2: (125,125) | layer3: (125,2)

After iterative updating, I settled on the following hyperparameters:

BATCH_SIZE = 125 | GAMMA = 0.99 | EPS_START = 0.9 | EPS_END = 0.05 | EPS_DECAY = 1000 | TAU = 0.005 | LR = 0.0001

I further tested different activation functions including sigmoid, leaky_relu, and relu. I ultimately went with the relu function as it had the highest accuracy out of the set. I do wonder how each would perform under optimized hyperparameters for the given activation function. This however would take lots and lots of testing/time.

Under the specified parameters and architecture, here is how the DQN agent performed:

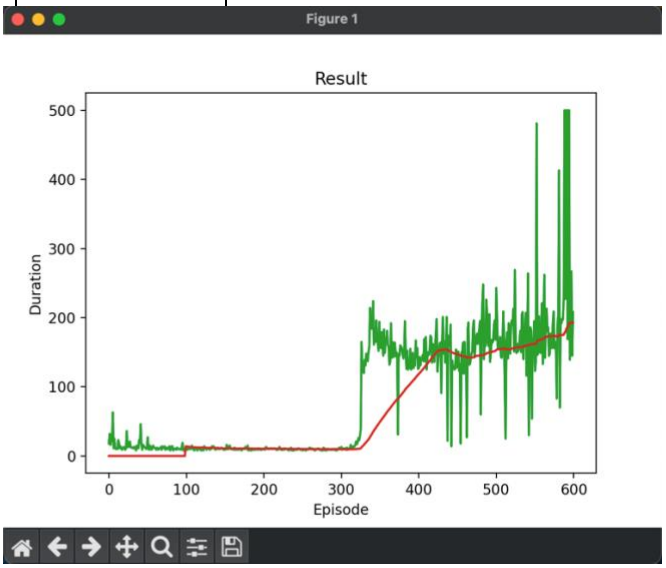


The agent took a little over 500 episodes to finish training, however once it did, it experienced no variance in future iterations. The agent was able to learn perform near perfect, with a consistent score of 475 (the max reward for cartpole game).

Armando Mendez

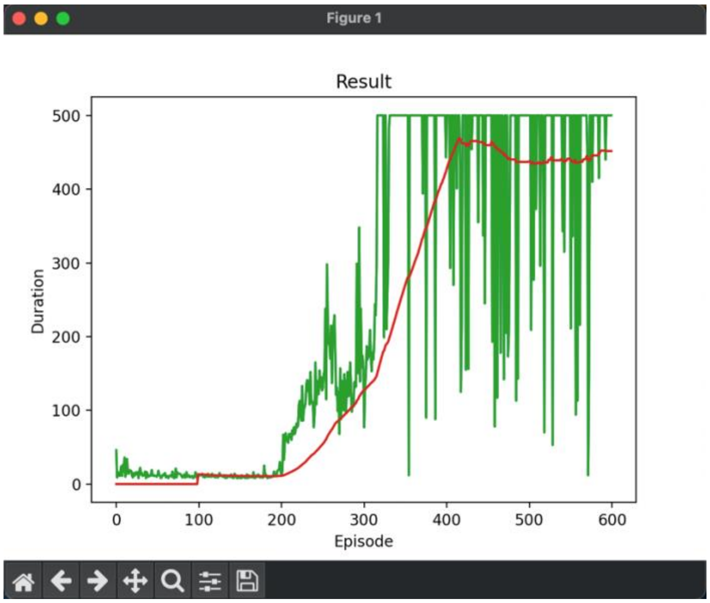
Hyperparameter Tests

Trial 1: BATCH_SIZE = 150 | GAMMA = 0.9 | EPS_START = 0.9 | EPS_END = 0.05 | EPS_DECAY = 1000 | TAU = 0.005 | LR = 0.001



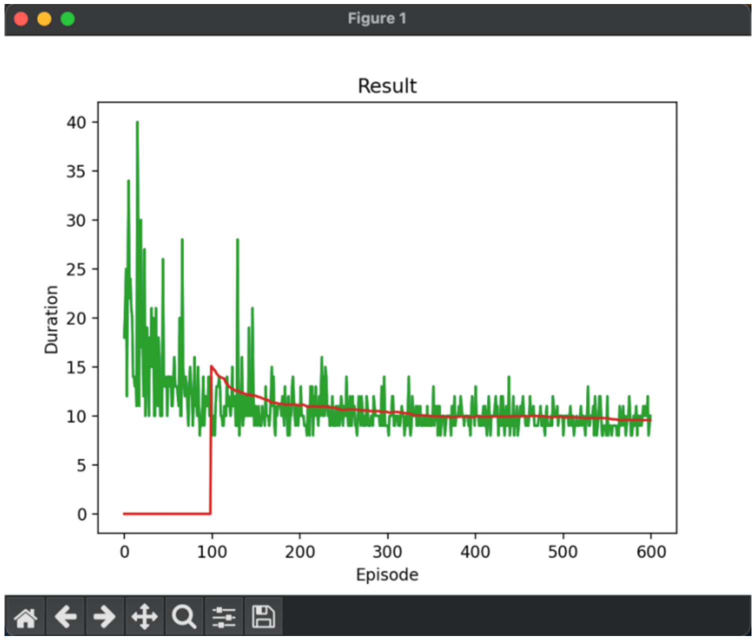
Increased the gamma to allow for greedier learning. Hopefully incentivizing the agent to perform actions which favor long-term rewards (and thus staying upwards for longer)

Trial 2: BATCH_SIZE = 150 | GAMMA = 0.99 | EPS_START = 0.9 | EPS_END = 0.05 | EPS_DECAY = 1000 | TAU = 0.005 | LR = 0.001



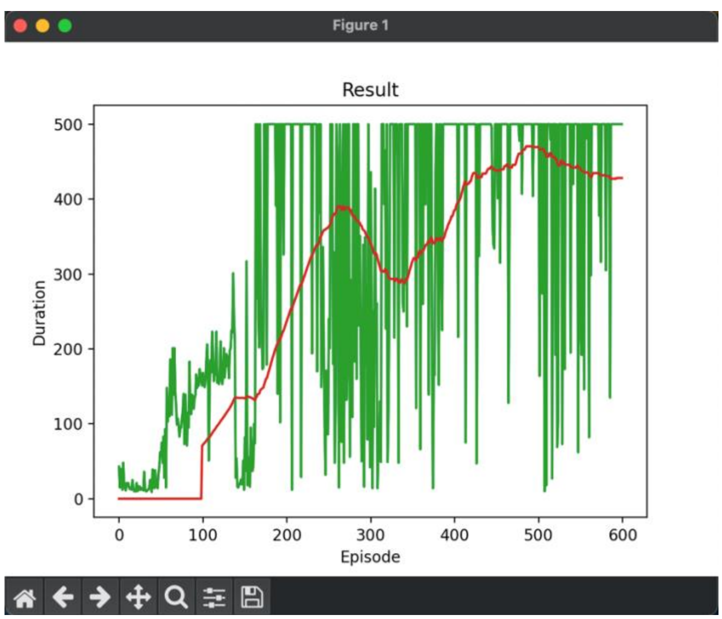
Changes to gamma worked, however now there seems to be high variance and still no convergence. Increase epsilon decay to reduce exploration/variance at the later episodes.

Trial 3: BATCH_SIZE = 150 | GAMMA = 0.99 | EPS_START = 0.9 | EPS_END = 0.05 | EPS_DECAY = 1500 | TAU = 0.005 | LR = 0.001



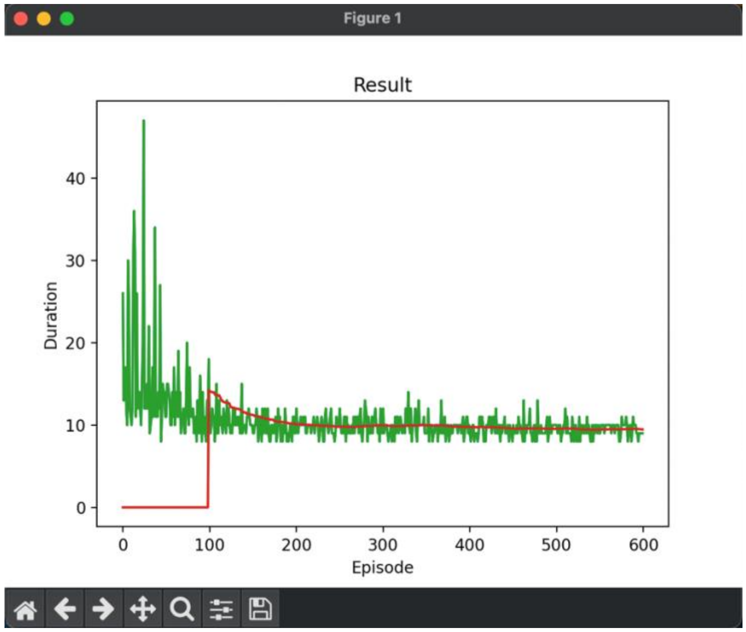
Agent was unable to converge accurately, however variance was reduced slightly. Will update epsilon decay to midway between 1000 and 1500.

Trial 4: BATCH_SIZE = 150 | GAMMA = 0.99 | EPS_START = 0.9 | EPS_END = 0.05 | EPS_DECAY = 1200 | TAU = 0.005 | LR = 0.001



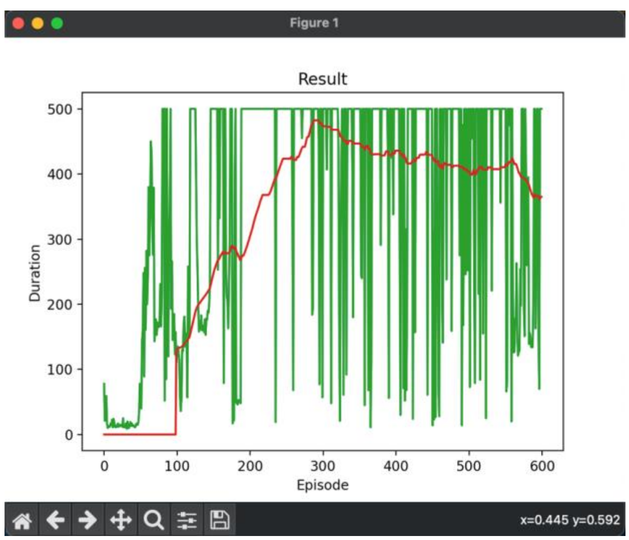
Agent seems more able to learn however still struggles with chaos in variance, will decrease learning rate to see effects.

Trial 5: BATCH_SIZE = 150 | GAMMA = 0.99 | EPS_START = 0.9 | EPS_END = 0.05 | EPS_DECAY = 1200 | TAU = 0.005 | LR = 0.0001



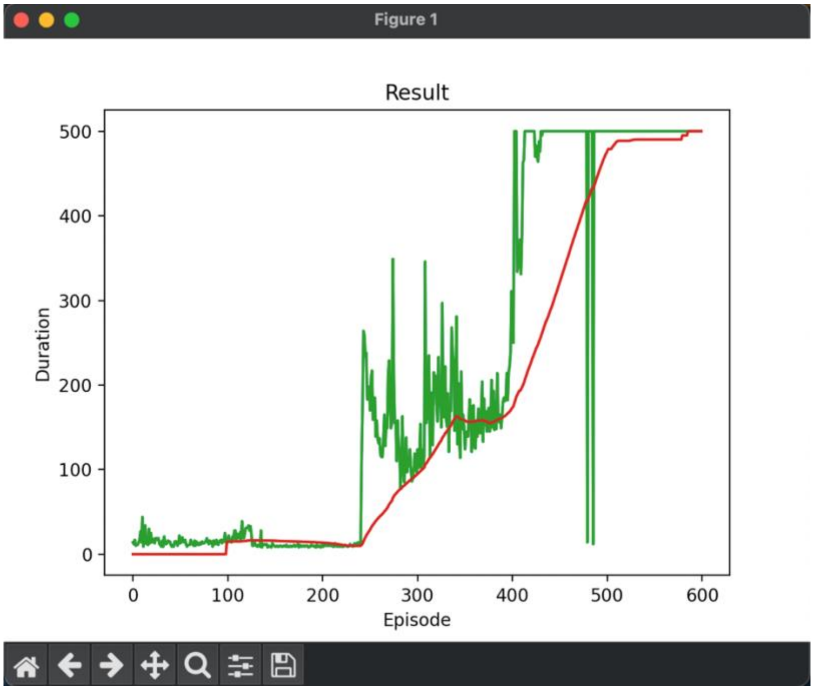
Similar issue to high variance, where agent fails to learn fast enough, will reset to eps_decay =1000 and LR =0.001, and test tweaking the batch_size. Hopefully with smaller batch sizes the agent will learn quicker before beginning to converge.

Trial 6: BATCH_SIZE = 125 | GAMMA = 0.99 | EPS_START = 0.9 | EPS_END = 0.05 | EPS_DECAY = 1000 | TAU = 0.005 | LR = 0.001



The smaller batch size worked, however agent fails to converge policy, will apply updates to learning rate and decay.

Trial 7: BATCH_SIZE = 125 | GAMMA = 0.99 | EPS_START = 0.9 | EPS_END = 0.05 | EPS_DECAY = 1000 | TAU = 0.005 | LR = 0.0001



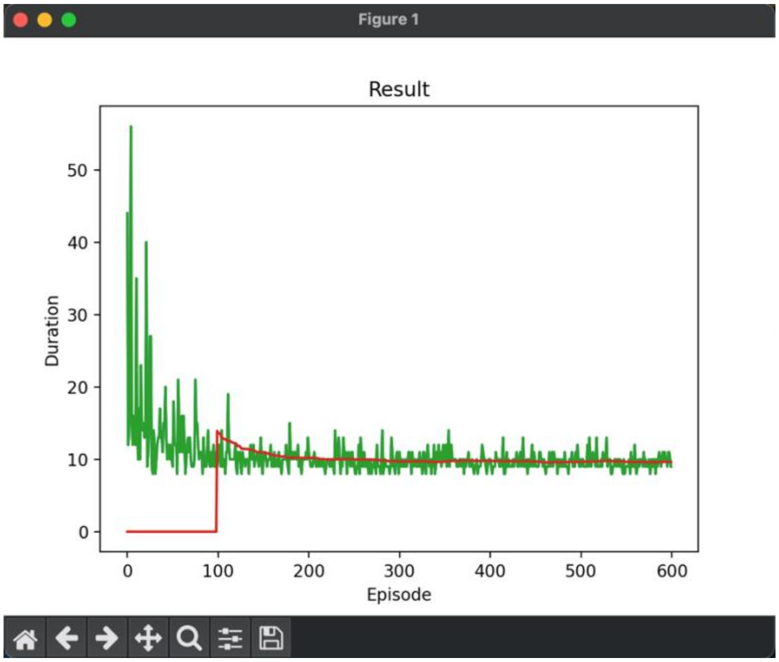
These are the hyper parameters I chose to go with, partly due to getting fed up with the iterative process (many trails not shown here), and because I wanted to work on the layer nodes. Agent seems to converge on an optimal policy with minor variance.

Armando Mendez

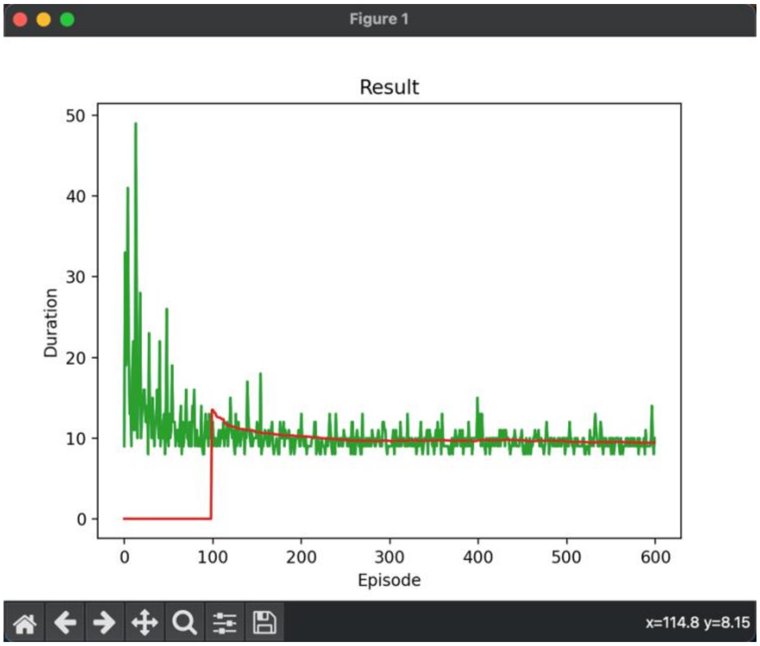
NN Architecture Tests

Hyperparameters: BATCH_SIZE = 125 | GAMMA = 0.99 | EPS_START = 0.9 | EPS_END = 0.05 | EPS_DECAY = 1000 | TAU = 0.005 | LR = 0.0001

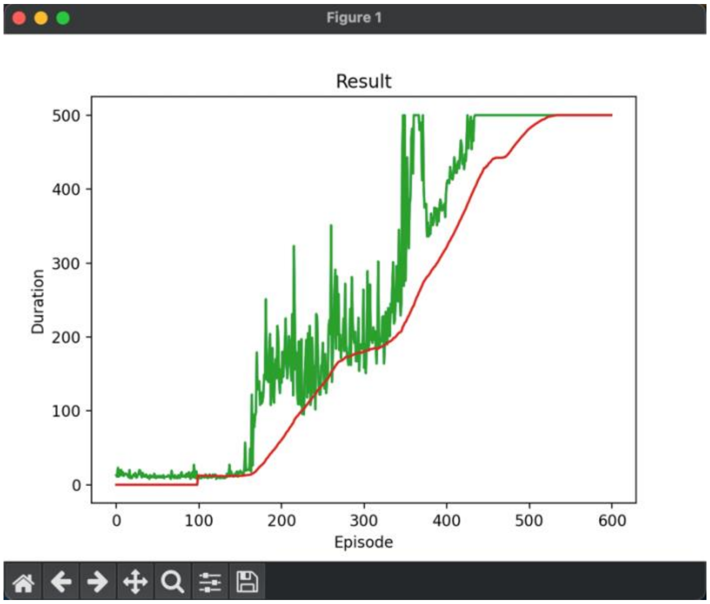
Trial 8: layer 1: (4, 50) | layer2: (50,50) | layer3: (50,2)



Trial 9: layer 1: (4, 75) | layer2: (75,75) | layer3: (75,2)

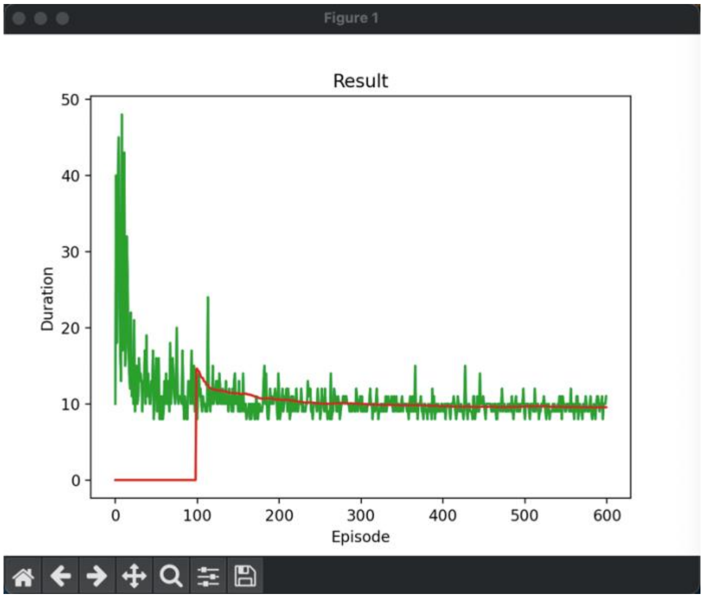


Trial 10: layer 1: (4, 125) | layer2: (125,125) | layer3: (125,2)



This seem to be the sweet spot

Trial 11: layer 1: (4, 150) | layer2: (150,150) | layer3: (150,2)

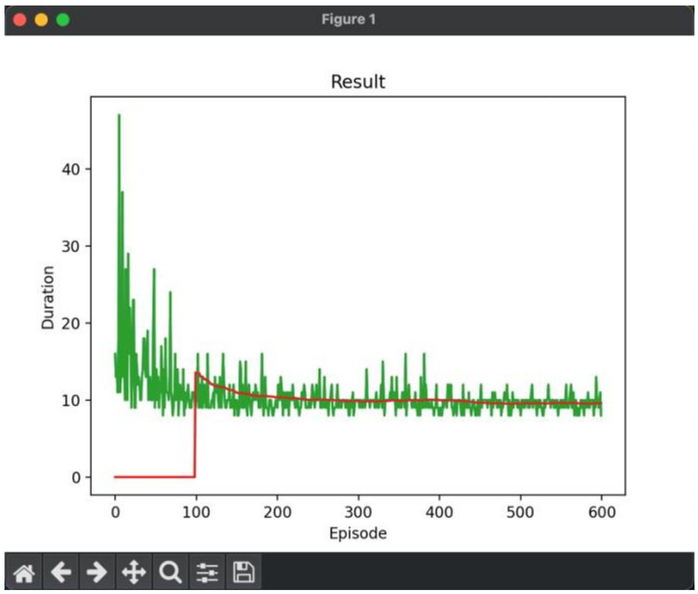


Armando Mendez

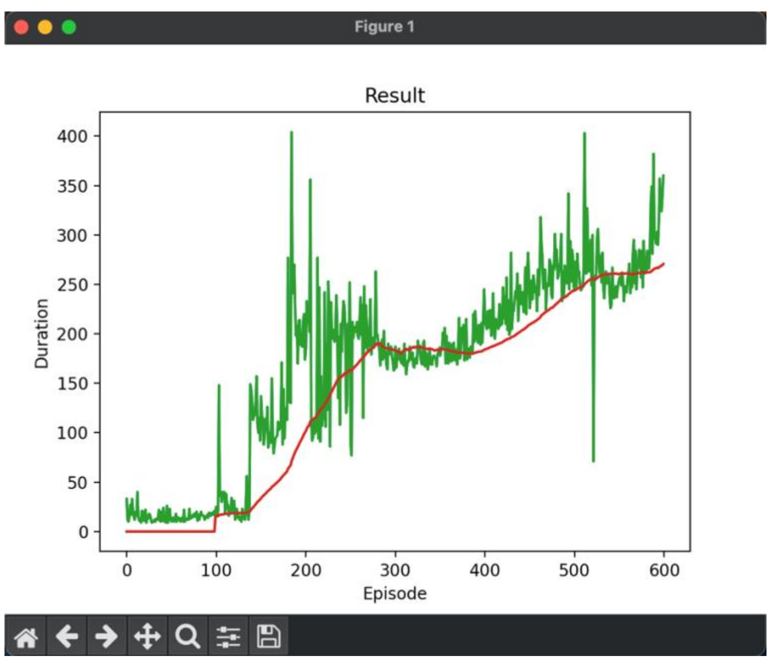
Activation Function Testing

BATCH_SIZE = 125 | GAMMA = 0.99 | EPS_START = 0.9 | EPS_END = 0.05 | EPS_DECAY = 1000 | TAU = 0.005 | LR = 0.0001
layer 1: (4, 125) | layer2: (125,125) | layer3: (125,2)

Trial 12: sigmoid



Trial 13: leaky Relu



Trial 14: Tanh