Erick Stone
Homework 6
CS6600

I very quickly discovered that more nodes would be needed in each layer. The reason for this is that the q table has a state mapping to an action. The only problem is that the state is a real number meaning there is no limit to the number of states available. A couple tricks you can perform is to round to the nearest some decimal point on the state prior to storing it into the q_table. This can drastically shrink the q table.

I reduced the number of states by rounding the state value going into the q table to the nearest hundredth. This limits it to only 100 states per 1 whole number. I tried variations using 2^n values from 64 to 1024. 2^n values work well with the GPU and CPU communications.

I found that larger networks didn't seem to train at all. At least, not without a huge number of epochs. Networks that were smaller than 3 nodes didn't seem to stabilize. This means that they just might have not been powerful enough to handle this problem. I also tried increasing and decreasing the number of layers. The most rewarding structure found was 3 layers of 512. An hour glass shaped network also seemed to work well. Having a network with steadily increasing networks size as well as steadily decreasing size both seemed to not be very stable.

Title	Network	Variation Description
2	512x512x512	Begins training earlier, significant Variance, steady increase, moving towards stability
3	256x512x1024	Similar stability and growth curve only starting later than 2.
4	1024x512x256	Extreme variance, late growth start, no real indicator of approaching stability
5	512x256x512	Better stability than 2, later growth than 2

	50-Reward	50-Position
2	-175	0.1
3	-200	-0.1
4	-200	-0.3
5	-200	-0.2

	300-Reward	300-Position
2	600	0.48
3	700	0.4
4	50	0.2
5	200	0.48