W251

Week 11 Homework

Travis Metz

July 2020

Results

With dense layers of 64 and 32, batch size of 256, unchanged epsilon decay of .995, and unchanged learning rate of .001:

- positive reward ~200 episodes (rolling 10)
- />200 reward ~380 (rolling 10)
- 100 test average 248.76

32, 16, 256

- positive reward ~ 240 (rolling 10)
- />200 reward ~460 (rolling 10)
- 100 test average 167.41

64, 32, 512

- positive reward ~250 (rolling 10)
- />200 reward ~420 (rolling 10)
- 100 test average 237.61

64, 32, 256, 0.99 decay

- positive reward ~170 (rolling 10)
- />200 reward ~240 (rolling 10)
- 100 test average 253.03
- 62.69 minutes to train

128, 64, 256, 0.99 decay

- positive reward ~110 (rolling 10)
- />200 reward ~210 (rolling 10)
- 100 test average - inadvertently stopped before finished
- 48 minutes to train (stopped at 320)

128, 64, 256, 0.995 decay

- positive reward ~ 280 (rolling 10)
- />200 reward ~ 420 (rolling 10)
- 100 test average - 252.90
- 97 minutes to train (stopped at 860)
- 60 minutes to test

256, 128, 256, 0.995 decay

- positive reward ~ 270 (rolling 10)
- />200 reward ~ 410 (rolling 10)
- 100 test average - 254.21
- 92 minutes to train (stopped at 1000)
- 61 minutes to test

512, 512, 256, 0.995 decay

- positive reward ~ 350 (rolling 10)
- />200 reward ~ 720 (rolling 10)
- 100 test average - 196.14
- 123 minutes to train (stopped at 1000)
- 71 minutes to test

256, 256, 256, 0.995 decay

- positive reward ~ 270 (rolling 10)
- />200 reward ~ 640 (rolling 10)
- 100 test average - 243.67
- 88 minutes to train (stopped at 740)
- 68 minutes to test

64, 64, 256, 0.995 decay

- positive reward ~ 280 (rolling 10)
- />200 reward ~ 440 (rolling 10)
- 100 test average - 244.97
- 86 minutes to train (stopped at 690)
- 64 minutes to test

128, 32, 256, 0.995 decay

- positive reward ~ 280 (rolling 10)
- />200 reward ~ 500 (rolling 10)
- 100 test average - 235.16
- 103 minutes to train (stopped at 1000)
- 86 minutes to test

128, 64, 256, 0.99 decay

- positive reward ~ 170 (rolling 10)
- />200 reward ~ 230 (rolling 10)
- 100 test average - 240.95
- 74 minutes to train (stopped at 1000)
- 93 minutes to test

Videos

I had to abandon my IBM Cloud account so have transitioned my classwork to AWS and am using an AWS S3 bucket.

(These run in Chrome but do not run on my local MP4 player.)

Training

https://w251-hw11-metz.s3.us-east-2.amazonaws.com/episode0.mp4

https://w251-hw11-metz.s3.us-east-2.amazonaws.com/episode100.mp4

https://w251-hw11-metz.s3.us-east-2.amazonaws.com/episode400.mp4

https://w251-hw11-metz.s3.us-east-2.amazonaws.com/episode600.mp4

Testing

https://w251-hw11-metz.s3.us-east-2.amazonaws.com/testing_run0.mp4

https://w251-hw11-metz.s3.us-east-2.amazonaws.com/testing_run20.mp4

Homework questions

1) What parameters did you change?

I changed the depth of first and second dense layers. I also experimented with different batch sizes. Also varied epsilon decay rate and tried different learning rates.

2) What values did you try?

See above.

3) Did you try any other changes that made things better or worse?

See above.

4) Did they improve or degrade the model? Did you have a test run with 100% of the scores above 200?

No.

5) Based on what you observed, what conclusions can you draw about the different parameters and their values?

There is very modest differences in model quality, even with models of meaningfully different sizes. The quality of the model is highly dependent on the epsilon decay level - the model needs time to explore random spaces. It also seems like the model is better with the second dense layer being smaller than first.

Lowering epsilon decay rate reduces the time for model to get 'acceptable' solutions (as seen in data above), but it also reduces the random space explored so reduces high end effectiveness of model.

6) What is the purpose of the epsilon value?

It is efffectively the percentage of times the model tries a random action rather than its view of the most rewarding action. Early in a model's development this figure needs to be high so that the model explores more space. It's decay allows the transition from explore to exploit as the model gets more sure of the solution.

7) Describe "Q-Learning".

Q-Learning is a process or algorithim of learning a model using random search and a reward structure. No pre-existing model of the procedure is required. The goal is to learn the action that generates the highest reward for each and every feasible state of the environment.

Other notes

In order to speed up, modified such that only showed graphical representation on every 100th episode during training, and printed results every 10th episode. Used jtop to monitor resource usage - not entirely clear if GPU being used for model (as opposed to the graphical representation). Modified to measure last 10 scores to get better sense of recent progress.

commands to build image

docker build -t hw11-image -f Dockerfile.agent . (TRM changed so that does not automatically run lander)

commands to run image etc

xhost + sudo jetson_clocks --store sudo jetson_clocks time docker run --name hw11 -it --rm --net=host --runtime nvidia -e DISPLAY=$DISPLAY -v /tmp/.X11-unix/:/tmp/.X11-unix:rw --privileged -v /home/trmetz/hw11-w251/:/tmp/ hw11-image

monitor gpu and system usage

sudo jtop