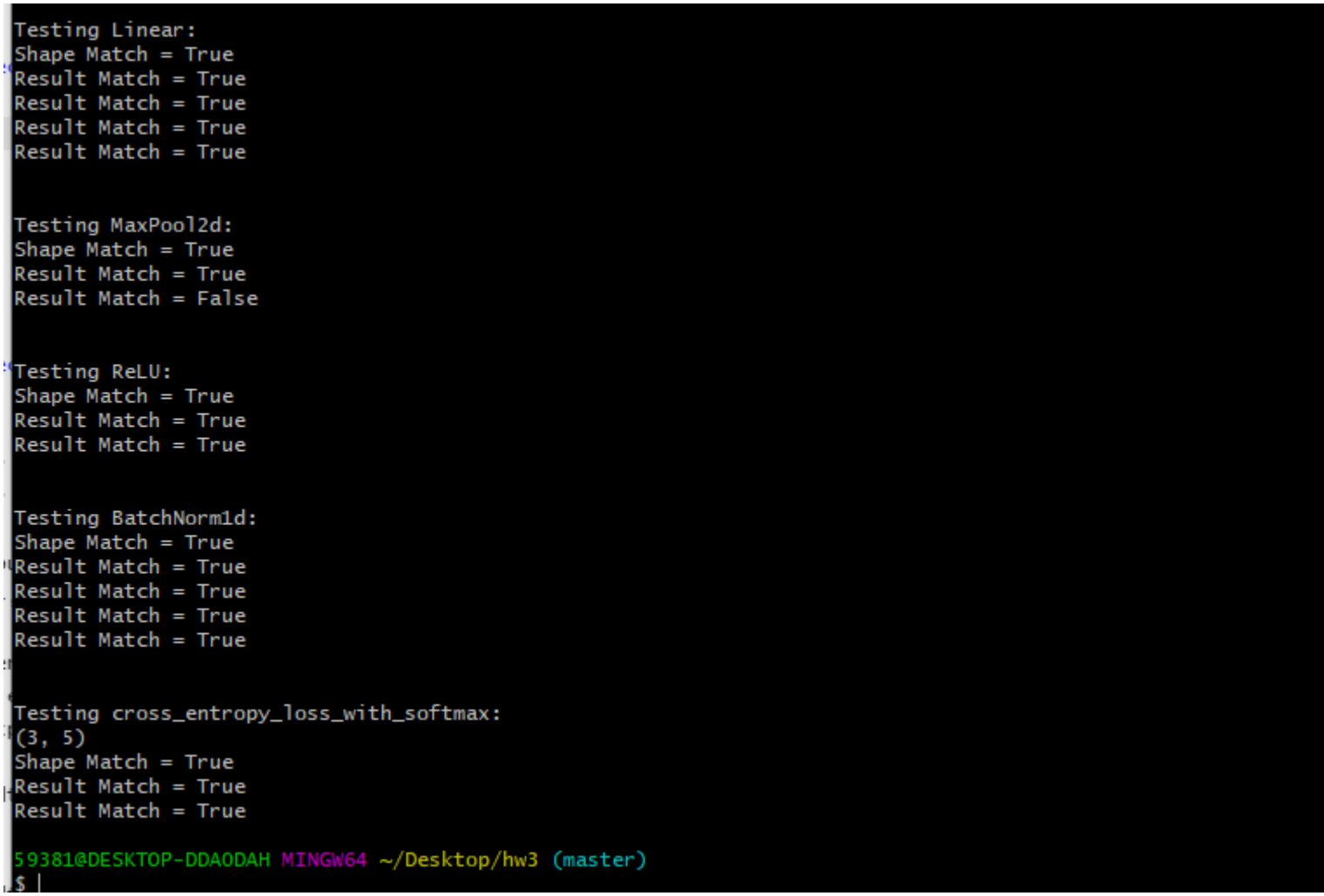
1 pynet screenshot



Brief description of Network

For the design of my simple network, I use nn.sequential() as a container for a series of layers.

Input tensor size: (64,3,32,32)

Therefore, I applied two similar series of Conv-ReLU-Maxpool2d sequence to extract the feature map. The first Conv layer takes in 3 channels and output 16 channels.(64,16,16,16)

Width and height after Conv:32-5+4+1 = 32

Width and height after maxpool:32-2/2 + 1 = 16

After first sequence output will be (64,16,32,32). An additional batchnorm layer was added to smooth the learning process.Similarly, the output of the second sequence will be (64,32,8,8).

For prediction of classes, I flatten each image map to be a single vector. And feed the output to a linear layer. I just use a linear layer, instead of a softmax layer to put each batch into 10 different classes(like a regression task).

For the back-prop and parameter updating, I use Adam optimization for a more stable convergence since we only have 10 epoch.

```
Epoch 9, Iteration 300, loss = 0.6657
Checking accuracy on validation set
Got 729 / 1000 correct (72.90)

Epoch 9, Iteration 400, loss = 0.4219
Checking accuracy on validation set
Got 720 / 1000 correct (72.00)

Epoch 9, Iteration 500, loss = 0.4702
Checking accuracy on validation set
Got 719 / 1000 correct (71.90)

Epoch 9, Iteration 600, loss = 0.4439
Checking accuracy on validation set
Got 707 / 1000 correct (70.70)

Epoch 9, Iteration 700, loss = 0.6935
Checking accuracy on validation set
Got 729 / 1000 correct (72.90)

Checking accuracy on test set
Got 7144 / 10000 correct (71.44)
[Finished in 1674.1s]
```

Line 234, Column 47