p1

Sunday, February 28, 2021 4:54 PM

Architecture:

1 Hidden layer (256 units), Sigmoid Activation
Output Layer (10 neurons): Softmax Cross Entropy Loss

hiddens = [256]
lr = 0.05
num_epochs = 15
batch_size = 8

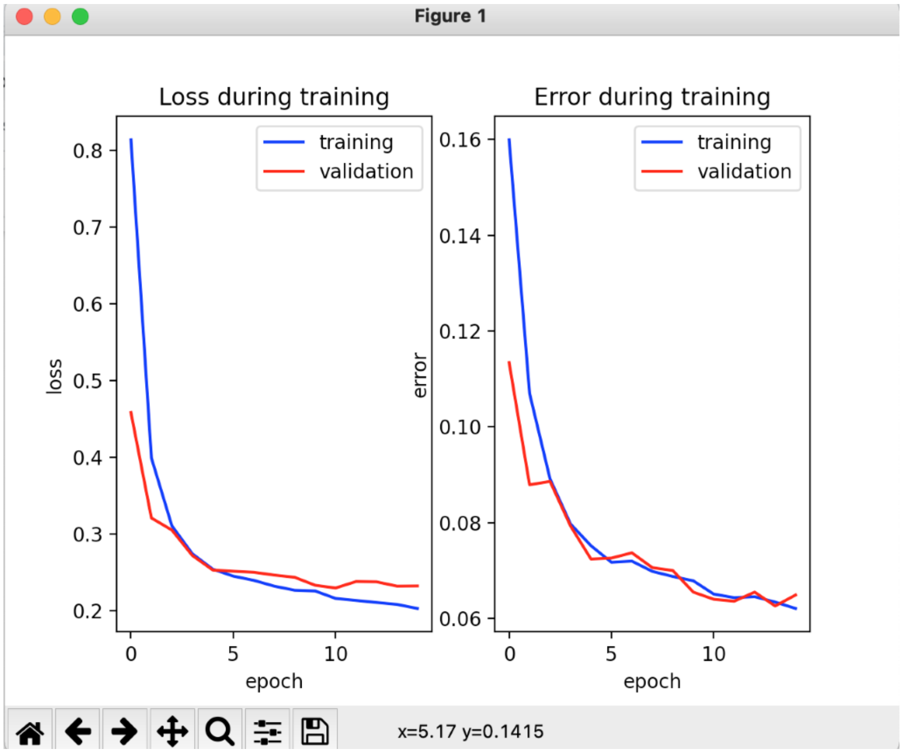
Test Error: 0.068

Reflection:

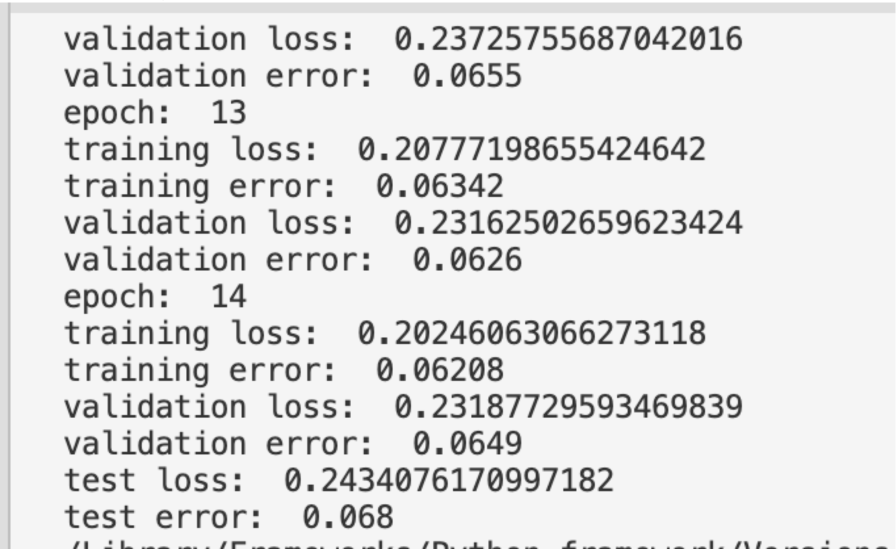
I noticed that using a smaller batch size like 8 may be slower to train since it needs to perform more weight updates per epoch, and the updates are also noisier (hence the 'zig zag' shapes in the graph). However, the validation loss decreases faster and converges towards the minimum faster.

I decided to use 1 Hidden layer with sigmoid activation, and softmax cross entropy loss to classify the input into K=10 output classes. I experimented with 2 Hidden layers with sigmoid activation, but found that it does not achieve a significantly lower test error, and takes more time to train.

When observing the graph, it seems that the graph overfits when the training error continues to decrease, but the validation error shoots back up. This encouraged me to lower the learning rate slightly. I found 0.05 to be a good balance; even though there is still some overshoot, it retains a relatively fast convergence rate compared to smaller learning rates.



Error of final epoch



```
validation loss:  0.23725755687042016
validation error:  0.0655
epoch:  13
training loss:  0.20777198655424642
training error:  0.06342
validation loss:  0.23162502659623424
validation error:  0.0626
epoch:  14
training loss:  0.20246063066273118
training error:  0.06208
validation loss:  0.23187729593469839
validation error:  0.0649
test loss:  0.2434076170997182
test error:  0.068
/Library/Frameworks/Python.framework/Versions
```