UDACITY

PROJECT

Your first neural network

A part of the Deep Learning Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Code Functionality

All the code in the notebook runs in Python 3 without failing, and all unit tests pass.

👍

The sigmoid activation function is implemented correctly

Nice usage of the python lambda functionality! You can read about common activation functions here http://cs231n.github.io/neural-networks-1/#actfun

All unit tests must be passing

😎

Forward Pass

The forward pass is correctly implemented for the network's training.

👍

The run method correctly produces the desired regression output for the neural network.

The input to the output layer is implemented correctly in both the train and run methods.

The output of the network is implemented correctly in both the train and run methods.

Great job here 👍👍👍!!. A lot of students get tripped up here and use the sigmoid function on the output nodes.
Tip: In other nets you can include a bias layer. Read up if you're interested with this great post http://stackoverflow.com/questions/2480650/role-of-bias-in-neural-networks

Backward Pass

The network correctly implements the backward pass for each batch, correctly updating the weight change.

Updates to both the input-to-hidden and hidden-to-output weights are implemented correctly.

Perfect work on the back prop. This can be tricky when using 2 types of activation functions but you nailed it. It's really exciting when you finally get your NN training and starting to predict accurately. If you ever need a quick refresher I found this to be a great video https://www.youtube.com/watch?v=GlcnxUlrtek

If you're interested in other perspectives on back propagation check out this awesome video by Prof Winston. https://www.youtube.com/watch?v=uXt8qF2Zzfo

Or this very informative post https://medium.com/@karpathy/yes-you-should-understand-backprop-e2f06eab496b#.pqvqa3xf3

This is also a great video by our course teacher: https://www.youtube.com/watch?v=p69khggr1Jo

Hyperparameters

The number of epochs is chosen such the network is trained well enough to accurately make predictions but is not overfitting to the training data.

This is great! Essentially, you'll want to start your epochs low and increase until your error loss curves flatten out
Here's a good resource on training
http://neuron.csie.ntust.edu.tw/homework/94/neuron/Homework3/M9409204/discuss.htm

The number of hidden units is chosen such that the network is able to accurately predict the number of bike riders, is able to generalize, and is not overfitting.

Nice! See this post about choosing the number of hidden nodes
https://www.quora.com/How-do-I-decide-the-number-of-nodes-in-a-hidden-layer-of-a-neural-network

The learning rate is chosen such that the network successfully converges, but is still time efficient.

Nice!

There are MANY ways to tune this net. You found one.

Epoch 2000, 25, .01 is another. Check that out and compare your results!

You can read up more on learning rates in this excellent post http://cs231n.github.io/neural-networks-3/

DOWNLOAD PROJECT

RETURN TO PATH

Student FAQ