UDACITY

PROJECT

Your first neural network

A part of the Deep Learning Nanodegree Foundation Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

2 SPECIFICATIONS REQUIRE CHANGES

Good job in general! :)
You're super close to finishing this assignment, just take a look at the feedback I gave you, correct the issues and you'll be good to go! :)

Good luck! I look forward to see the amazing things you build next :)

Code Functionality

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

Good job! All unit tests passed! :)

The sigmoid activation function is implemented correctly

Well done! Great implementation of the activation function! :)

Forward Pass

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

Good job! :)

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

Good work! Great use of the activation function! :)

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

Good job! :)

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

Well done! In this case the activation function is f(x)=x :)

Backward Pass

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

Good job! :)

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

Goof job updating the weights! :)

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.

Good effort here!
I marked this one because it's hard to tell if your number of epochs is correct or not, the scale is too big.
your errors are in the scale of 0.1, it's very hard to know this way if it can improve further, I would suggest to improve the scaling.
You'll either way have to change it after changing the learn rate :)

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.

Great work! Good selection of a number of hidden units. :)
For future reference, here's a guide to determine a good number of hidden units - 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.

Goo effort here! :)
please note that the effective learn rate is self.lr / n_records and we're usually trying to work with a learn rate in the order of magnitude of 0.01. In your case the learn rate is 0.2 / 1440
(1440 is the number of records 60*24)
Please increase the learn rate.

☑ RESUBMIT

DOWNLOAD PROJECT