UDACITY

PROJECT

Your first neural network

A part of the Deep Learning Nanodegree Foundation Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Great work! The network is implemented perfectly and the hyperparameters are producing great results.

There is some subjectivity involved in model tuning, and choosing optimal hyperparameters is a huge area of research. I was reading through your reviewer history and noticed you received a lot of feedback about the hyperparameters, so hopefully you picked up some good knowledge from the process 😃 Just keep in mind the tradeoffs involved with adjusting each value and the impacts they have on the model

I like your answer to the optional question also. It shows that you clearly understand what is going on and have ideas to improve the model. This is really important and a valuable skill to keep practicing 👏

I hope you enjoyed this project! Best of luck in the future and keep up the fantastic work! 💯

Code Functionality

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

Great work! The code is running and passing all unit tests 👍

The sigmoid activation function is implemented correctly

You got it!

Forward Pass

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

Good job using `np.dot()`. It comes in handy a lot when working with matrices

The output of the hidden layer is implemented correctly in both the `train` and `run` methods.

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.

Awesome work on the forward pass! Perfection 💯

Backward Pass

The network output error is implemented correctly

Updates to both the weights are implemented correctly.

Great job here! The backward pass is complicated, so congratulations on learning this! ⭐

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.

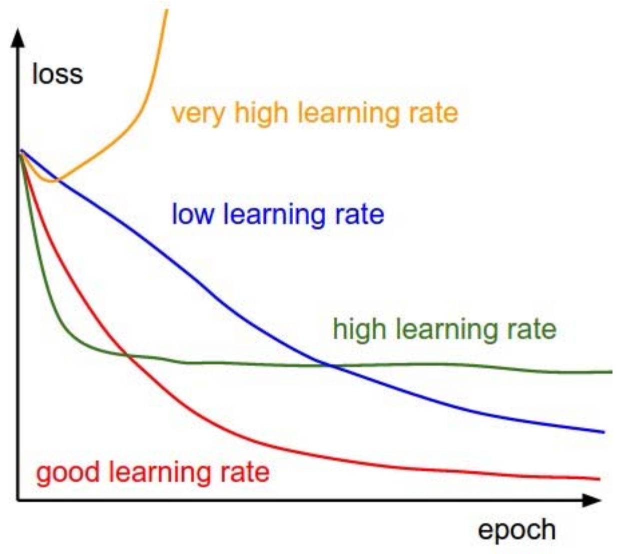
Great! The network is fitting the data very accurately and generalizing well. The validation loss has plateaued, indicating the network has converged. In general, 12000 epochs is a pretty large value for a dataset this size; I added some comments about experimenting with the learning rate and epochs below that you are welcome to try

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.

Fantastic! It looks like you know how to tune the hidden units well ✅ Those loss functions are beautiful

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

Great work! This is a good hyperparameter to experiment with. In this implementation, we are dividing by `n_records=128` at each iteration in the batch learning process, which effectively reduces the `learning_rate` value you are setting. In practice, `learning_rates` tend to be pretty low (in the 0.001-0.1 range). You can experiment with increasing the learning rate and decreasing the epochs to try to maintain accuracy and decrease the training time.



DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

Student FAQ