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

You did a great job incorporating the changes from previous review. The submission you made is almost perfect!

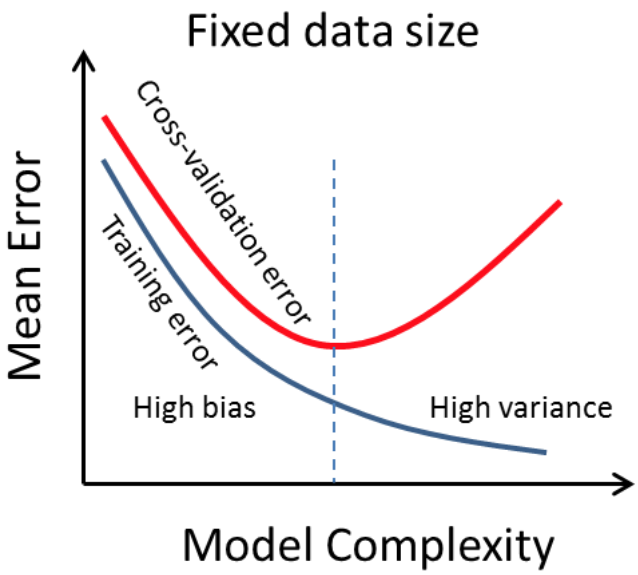
Here's some suggestion on correctly tuning the parameters :

Suggestion

You may have done this already, but one idea here is to initially try to "overshoot" a little:

- find a set of parameters that lead to a high variance model (i.e., one for which training loss is significantly lower than testing loss)
- then "dial it back" and modify the parameters so as to get the model to present both low bias and low variance (that is, a model with low training loss, low testing loss, and similar values for both losses)

Make sure you get as close as possible to the dotted line in the image below :



Code Functionality

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

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

The sigmoid activation function is implemented correctly

Good job implementing the `lambda` function.

Forward Pass

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

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.

Excellent work implementing the forward pass correctly

Backward Pass

The network output error is implemented correctly

Well done!

Updates to both the weights are implemented correctly.

Well done here! The backward pass is correctly implemented. Please remove the unnecessary comments

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.

Looks good

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.

Seems appropriate

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

Well done!

DOWNLOAD PROJECT

Have a question about your review? Email us at review-support@udacity.com and include the link to this review.

RETURN TO PATH

Rate this review

Student FAQ