UDACITY

< Return to Classroom

Predicting Bike-Sharing Patterns

REVIEW

CODE REVIEW

HISTORY

Requires Changes

2 specifications require changes

A great improvement and a wonderful job! Awesome work! Just a bit of tuning and you will have an awesome model.
A wonderful submission! Your code runs really well! 👏🏻
Just a bit of tuning required and your model will make awesome predictions!
Once you tune your model you will find that the model predicts perfectly for most of the days except for the last few days of the year. Try to imagine why this is so.. what can be done about this? Will more data help? Will feature engineering help? This is really important insight required for machine learning practise! Have fun! All the best! 👋🏻

Code Functionality

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

Correct!

The sigmoid activation function is implemented correctly

Correct!

Forward Pass

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

Correct!

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

Correct!

Backward Pass

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

Correct!

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

Correct!

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.

Correct!

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.

Correct!

Try to have the number of hidden nodes at least 8.

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

```
Not all of the tests passed.


**********************************************************************
                        Test Failure Feedback
**********************************************************************
```

```
Failed Test: The learning_rate is reasonable
----------------------------------------------------------------------
AssertionError: False is not true



**********************************************************************
                        Test Result Summary
**********************************************************************

The learning_rate is reasonable                                    F
```

Please remember there is a trade-off between LR and the Iterations. If you choose smaller LR you will need more iterations to get the model converge and the other way around. Our job is to find the most efficient combination where the training process is time efficient and at the same time is adequately accurate.

Your learning rate seems to be a bit low to start with, try choosing a learning rate that gets the ratio of learning rate/the number of records(128) to be around 0.01, try out values such as [0.7,0.8,0.9]. A good learning rate is key to making the gradient descent converge in reasonable amount of time. Choosing a low learning rate will mean that the weight updates will be slower and hence the model will take longer to converge, choosing a high learning rate will over shoot the gradient as it takes bigger steps and probably never converge to minima. So it is crucial to tune the learning rate.

The number of output nodes is properly selected to solve the desired problem.

Correct!

The training loss is below 0.09 and the validation loss is below 0.18.

You need to make sure to follow the suggestions and tune your model to get training loss below 0.09 and validation loss below 0.18 to pass the rubric.

RESUBMIT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

RETURN TO PATH