UDACITY

< Return to "Deep Learning" in the classroom

DISCUSS ON STUDENT HUB

Predicting Bike-Sharing Patterns

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Congratulations!

You have successfully passed the project.

Code Functionality

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

All the code files and unit test functions are running flawlessly.

The sigmoid activation function is implemented correctly

Implementation of sigmoid activation function is all correct. You can also consider using this expit().

Forward Pass

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

Good Job!

The forward propagation is well coded. It's computing the hidden inputs, hidden outputs, final inputs and final outputs correctly.
It's great that you have understood the fact that we need not to use any activation function with the final layer.

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

The run() method is well coded. It's returning the final outputs correctly.

Backward Pass

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

Perfect!

The backward propagation is well coded. It's computing the errors, gradient steps and delta weight update steps. If there's still some doubt with the backward propagation, I encourage you to go through this blog post: https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

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

The final weight update steps are well coded. It's great that you've normalized 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.

Great Job!

The number of iterations as 8000 is a great choice. These many iterations are good enough to train the model and help it learn the patterns. It should keep decreasing as the network is overfit to the training data.

Basically students should choose the number of epochs such that the loss on the training set is low and the loss on the validation set isn't increasing.

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.

The number of hidden nodes as 15 is a great choice. This helps to build a reasonably complex model that can learn the patterns and predict great results.

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

The learning rate as 0.25 is acceptable. This should be chosen so as to let the weights converge and reach minima. I encourage you to try using a bit higher learning rate too like 0.75, 0.9, 1 etc.

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

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

Awesome!

The training-validation losses are converging and are in acceptable range. ⭐

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review