CSE 564: Software Design

Fall 2018

Assignment 4: Neural Network Design and Training

Total Points: 100

Due date: 11/14/2018

- This is a two-person assignment. Each group of TWO students will be working on this assignment together and having one submission.
- Both students will receive the same grade for this assignment
- For input, you can use data from assignment 2 of either one of the two members, but not from a third person

In this assignment you will design and train a multi input, multi output neural network to find the correlation between the input file features vector and an the output file features vector. You need to use the input and output files that you produced in your assignment 2 as the input and output of this neural network.

Your input file has 10 features including:

Velocity	LanePos	Steer	SpeedLimit	Accel	Brake	LongAccel	LatAccel	HeadwayTime	HeadwayDist

Your output file has 4 features including:

Mode	Response Time	Mistakes	Steps

To get started, you can find a neural network code for regression in this URL:

https://www.tensorflow.org/tutorials/keras/basic_regression

It is a simple model for regression with neural network:

```
def build_model():
  model = keras.Sequential([
    keras.layers.Dense(64, activation=tf.nn.relu,
                       input_shape=(train_data.shape[1],)),
    keras.layers.Dense(64, activation=tf.nn.relu),
    keras.layers.Dense(1)
  ])
```

To have a successful design, you need to reach Mean absolute error of less than 0.4 of your network. For properly designing your neural network you need to try different combination of hyper-parameters including the number of hidden layers, the number of hidden neurons in each hidden layer, learning rate, batch size, activation function and loss function and the number of epochs.

Deliverables:

1. Code in python with some screenshots that shows training process and the error of the network during different epochs like the next figure. Add at least 3 screen shots from the first epochs, middle epochs and the final epochs.
2. A report that explains the effect of changing each hyper-parameters on your network final error
3. Provides a Table that shows all combination of hyper-parameters that you tried and the result of each combination. Each column of this table shows the values of one hyper-parameters and the last two columns show the train error and the test error.
4. You need to include all hyper-parameters you used in your design process, even if they negatively affected your training results and you had to revert to previous state
5. The mean absolute error of your test set should be less than 0.4
6. The hyper-parameters table need to have these columns: Number of hidden layers, number of hidden neurons in each hidden layer, learning rate, number of epochs, activation function, loss function, batch size, train mean absolute error and test mean absolute error

```
1606/1606 [==============================] - 933s 581ms/step - loss: 7.6376 - mean_absolute_error: 1.8111 - val_loss: 5.2309
- val_mean_absolute_error: 1.5297
Epoch 2/400
1606/1606 [==============================] - 1095s 682ms/step - loss: 5.6145 - mean_absolute_error: 1.5555 - val_loss: 4.9780
- val_mean_absolute_error: 1.4948
Epoch 3/400
1606/1606 [==============================] - 1141s 711ms/step - loss: 5.2460 - mean_absolute_error: 1.3773 - val_loss: 3.8804
- val_mean_absolute_error: 1.1574
Epoch 4/400
1606/1606 [==============================] - 1193s 743ms/step - loss: 4.0682 - mean_absolute_error: 1.2174 - val_loss: 3.7940
- val_mean_absolute_error: 1.1469
Epoch 5/400
1606/1606 [==============================] - 1216s 757ms/step - loss: 4.0311 - mean_absolute_error: 1.2002 - val_loss: 3.8343
- val_mean_absolute_error: 1.2069
Epoch 6/400
1606/1606 [==============================] - 1250s 778ms/step - loss: 3.9938 - mean_absolute_error: 1.2089 - val_loss: 3.9689
- val_mean_absolute_error: 1.1569
Epoch 7/400
1606/1606 [==============================] - 1260s 784ms/step - loss: 4.0560 - mean_absolute_error: 1.2004 - val_loss: 3.8496
```