keras_Testing

April 30, 2020

```
In [9]: import pickle
        import numpy as np
        import tensorflow as tf

        # Load pickled data
        with open('small_train_traffic.p', mode='rb') as f:
            data = pickle.load(f)
```

```
In [10]: # Split the data
         X_train, y_train = data['features'], data['labels']
```

```
In [11]: # Setup Keras
         from keras.models import Sequential
         from keras.layers.core import Dense, Activation, Flatten, Dropout
         from keras.layers.convolutional import Conv2D
         from keras.layers.pooling import MaxPooling2D
```

```
In [12]: # TODO: Build the Final Test Neural Network in Keras Here
         model = Sequential()
         model.add(Conv2D(32, kernel_size = (3, 3), activation = 'relu', input_shape = (32, 32,
         model.add(MaxPooling2D(pool_size = (2,2)))
         model.add(Dropout(rate = 0.5))
         model.add(Flatten())
         model.add(Dense(128, activation = 'relu'))
         model.add(Dense(5, activation = 'softmax'))
```

```
In [13]: # preprocess data
         X_normalized = np.array(X_train / 255.0 - 0.5 )

         from sklearn.preprocessing import LabelBinarizer
         label_binarizer = LabelBinarizer()
         y_one_hot = label_binarizer.fit_transform(y_train)
```

```
In [14]: # compile and fit the model
         model.compile('adam', 'categorical_crossentropy', ['accuracy'])
         history = model.fit(X_normalized, y_one_hot, epochs=6, validation_split=0.2)
```

```
Train on 80 samples, validate on 20 samples
Epoch 1/6
```

```
80/80 [==============================] - 0s 5ms/step - loss: 1.4797 - acc: 0.3250 - val_loss: 0.
Epoch 2/6
80/80 [==============================] - 0s 3ms/step - loss: 0.8430 - acc: 0.5375 - val_loss: 0.
Epoch 3/6
80/80 [==============================] - 0s 3ms/step - loss: 0.6627 - acc: 0.7375 - val_loss: 0.
Epoch 4/6
80/80 [==============================] - 0s 3ms/step - loss: 0.5091 - acc: 0.7375 - val_loss: 0.
Epoch 5/6
80/80 [==============================] - 0s 3ms/step - loss: 0.4301 - acc: 0.8125 - val_loss: 0.
Epoch 6/6
80/80 [==============================] - 0s 3ms/step - loss: 0.3621 - acc: 0.8125 - val_loss: 0.
```

```
In [15]: # evaluate model against the test data
         with open('small_test_traffic.p', 'rb') as f:
             data_test = pickle.load(f)

         X_test = data_test['features']
         y_test = data_test['labels']

         # preprocess data
         X_normalized_test = np.array(X_test / 255.0 - 0.5 )
         y_one_hot_test = label_binarizer.fit_transform(y_test)

         print("Testing")

         metrics = model.evaluate(X_normalized_test, y_one_hot_test)
         for metric_i in range(len(model.metrics_names)):
             metric_name = model.metrics_names[metric_i]
             metric_value = metrics[metric_i]
             print('{}: {}'.format(metric_name, metric_value))
```

```
Testing
20/20 [==============================] - 0s 693us/step
loss: 0.3711778223514557
acc: 0.949999988079071
```

```
In [16]: ### DON'T MODIFY ANYTHING BELOW ###
         ### Be sure to run all cells above before running this cell ###
         import grader

         try:
             grader.run_grader(metrics)
         except Exception as err:
             print(str(err))
```

```
Nice, accuracy was 0.949999988079
Good Job, accuracy was above 90%
```

```
In [ ]:
```