Handwritten Digit Recognition with MNIST Dataset

Mark McCauley

1 Code

```
import tensorflow as flow
import matplotlib.pyplot as plt

from keras.models import Sequential
from keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D

(x_train, y_train), (x_test, y_test) = flow.keras.datasets.mnist.load_data()

# Keras API only accepts 4 dimensional arrays so we must reshape data
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)

input_shape = (28, 28, 1)

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')

x_train /= 255
x_test /= 255

# set up model
model = Sequential()
model.add(Conv2D(28, kernel_size=(3,3), input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(128, activation=flow.nn.relu))
model.add(Dropout(0.2))
model.add(Dense(10, activation=flow.nn.softmax))

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x=x_train,y=y_train, epochs=10)
loss, acc = model.evaluate(x_test, y_test)
print(acc)
```

2 Testing and Results

This neural network was able to predict the digits with 97% accuracy on one epoch and above 98% with 10 epochs. These results are predictible as the MSNIST data set is reliable and simple to work with.

```
Using TensorFlow backend.
Epoch 1/10
2019-01-29 11:11:11.152783: I tensorflow/core/platform/cpu_feature_guard.cc:141]
60000/60000 [==============================]
- 39s 642us/step - loss: 0.2219 - acc: 0.9328
Epoch 2/10
60000/60000 [==============================]
- 37s 620us/step - loss: 0.0875 - acc: 0.9736
Epoch 3/10
60000/60000 [==============================]
- 38s 640us/step - loss: 0.0608 - acc: 0.9808
Epoch 4/10
60000/60000 [==============================]
- 36s 595us/step - loss: 0.0480 - acc: 0.9847
Epoch 5/10
60000/60000 [==============================]
- 36s 593us/step - loss: 0.0373 - acc: 0.9879
Epoch 6/10
60000/60000 [==============================]
- 38s 640us/step - loss: 0.0318 - acc: 0.9897
Epoch 7/10
60000/60000 [==============================]
- 39s 650us/step - loss: 0.0262 - acc: 0.9908
Epoch 8/10
60000/60000 [==============================]
- 37s 613us/step - loss: 0.0243 - acc: 0.9919
Epoch 9/10
60000/60000 [==============================]
- 37s 618us/step - loss: 0.0224 - acc: 0.9924
Epoch 10/10
60000/60000 [==============================]
 - 37s 619us/step - loss: 0.0199 - acc: 0.9936
10000/10000 [==============================] - 2s 184us/step
0.9861
```