Roshan Shetty 525

DML Mini Project

Aim: Handwritten digit recognition using mnist dataset

What is MNIST?

1. Set of 70,000 small images of digits handwritten by high school students and employees of the US causes Bureau.

2. All images are labeled with the respective digit they represent.

3. MNIST is the hello world of machine learning. Every time a data scientist or machine learning engineer makes a new algorithm for classification, they would always first check its performance on the MNIST dataset.

4. There are 70,000 images and each image has 28*28 = 784 features.

5. Each image is 28*28 pixels and each feature simply represents one-pixel intensity from 0 to 255. If the intensity is 0, it means that the pixel is white and if it is 255, it means it is black.

Code:

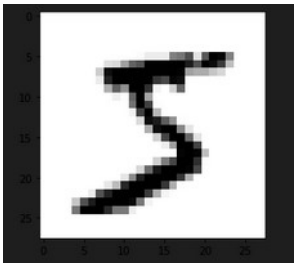
```
import tensorflow.keras as keras
import tensorflow as tf

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

import matplotlib.pyplot as plt
plt.imshow(x_train[0],cmap=plt.cm.binary)
plt.show()

print(y_train[0])

x_train = tf.keras.utils.normalize(x_train, axis=1)
x_test = tf.keras.utils.normalize(x_test, axis=1)

model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Flatten())

model.add(tf.keras.layers.Dense(128, activation=tf.nn.relu))
model.add(tf.keras.layers.Dense(128, activation=tf.nn.relu))
model.add(tf.keras.layers.Dense(10, activation=tf.nn.softmax))

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

model.fit(x_train, y_train, epochs=3)

val_loss, val_acc = model.evaluate(x_test, y_test)
print(val_loss)
```

Roshan Shetty 525

```
print(val_acc)

predictions = model.predict(x_test)

import numpy as np
print(np.argmax(predictions[16]))

plt.imshow(x_test[16],cmap=plt.cm.binary)
plt.show()
```

Output:

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11493376/11490434 [==============================] - 0s 0us/step
11501568/11490434 [==============================] - 0s 0us/step
```



```
Epoch 1/3
1875/1875 [==============================] - 19s 9ms/step - loss: 0.2571 - accuracy: 0.9234
Epoch 2/3
1875/1875 [==============================] - 10s 5ms/step - loss: 0.1036 - accuracy: 0.9679
Epoch 3/3
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0722 - accuracy: 0.9773

<keras.callbacks.History at 0x7f2c391b3d90>
```

```
313/313 [==============================] - 1s 2ms/step - loss: 0.1001 - accuracy: 0.9706
0.10010665655136108
0.9706000089645386
```