

dl-prac2a

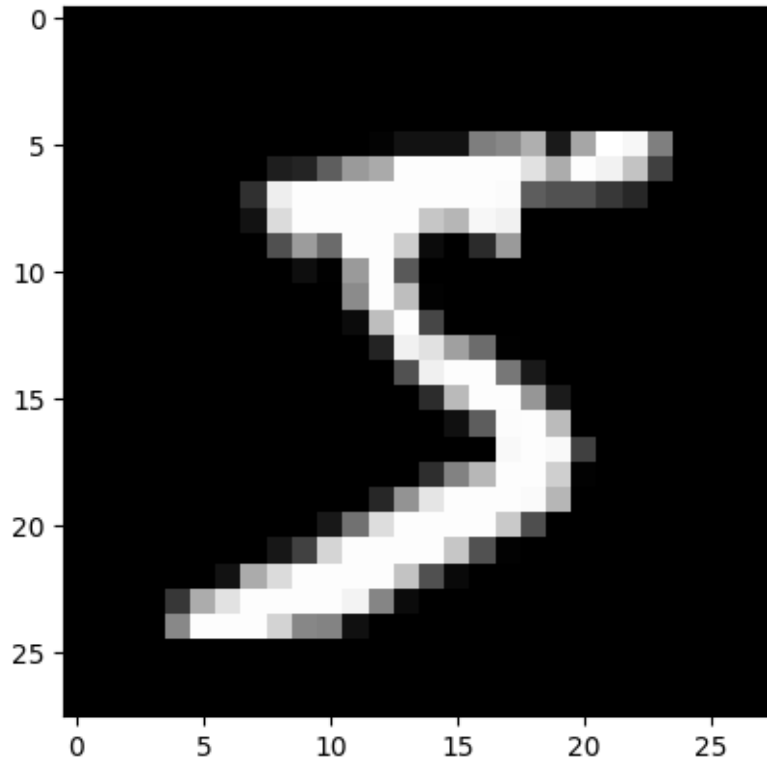
April 29, 2024

```
[1]: import numpy as np
      from tensorflow.keras.models import Sequential
      from tensorflow.keras.layers import Dense, Dropout
      from tensorflow.keras.optimizers import RMSprop
      from tensorflow.keras.datasets import mnist
      import matplotlib.pyplot as plt
      from sklearn import metrics
```

```
[2]: (x_train, y_train), (x_test, y_test) = mnist.load_data()
```

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

```
[3]: plt.imshow(x_train[0], cmap='gray')
      plt.show()
```



```
[4]: print(x_train[0])
```

```
[[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  3  18  18  18 126 136
 175 26 166 255 247 127  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 30 36 94 154 170 253 253 253 253 253
 225 172 253 242 195 64  0  0  0  0]
 [ 0  0  0  0  0  0  0 49 238 253 253 253 253 253 253 253 253 251
 93 82 82 56 39  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 18 219 253 253 253 253 253 198 182 247 241
  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 80 156 107 253 253 205 11  0 43 154
  0  0  0  0  0  0  0  0  0  0]
```

```

[ 0  0  0  0  0  0  0  0  0  0 14  1 154 253  90  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0 139 253 190  2  0  0  0
  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  11 190 253  70  0  0  0
  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  35 241 225 160 108  1
  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  81 240 253 253 119
 25  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  45 186 253 253
150 27  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  16  93 252
253 187  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 249
253 249 64  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  46 130 183 253
253 207  2  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  39 148 229 253 253 253
250 182  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0 24 114 221 253 253 253 253 201
 78  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0 23  66 213 253 253 253 253 198  81  2
  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0 18 171 219 253 253 253 253 195  80  9  0  0
  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  55 172 226 253 253 253 253 244 133  11  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0 136 253 253 253 212 135 132  16  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0]

```

```

[5]: print("x_train shape: ",x_train.shape)
      print("y_train shape: ",y_train.shape)
      print("x_test shape: ",x_test.shape)
      print("y_test shape: ",y_test.shape)

```

```

x_train shape: (60000, 28, 28)
y_train shape: (60000,)
x_test shape: (10000, 28, 28)
y_test shape: (10000,)

```

```
[6]: x_train = x_train.reshape(60000, 784)
      x_test = x_test.reshape(10000, 784)
      x_train = x_train.astype('float32')
      x_test = x_test.astype('float32')
      x_train = x_train/255 # Each image has Intensity from 0 to 255
      x_test = x_test/255
```

```
[7]: num_classes = 10
      y_train = np.eye(num_classes)[y_train]
      y_test = np.eye(num_classes)[y_test]
```

```
[8]: model = Sequential()
      model.add(Dense(512, activation='relu', input_shape=(784,)))
      model.add(Dropout(0.2))
      model.add(Dense(512, activation='relu'))
      model.add(Dropout(0.2))
      model.add(Dense(num_classes, activation='softmax'))
      model.compile(loss='categorical_crossentropy', optimizer=RMSprop(),
                    metrics=['accuracy'])
```

```
[9]: batch_size = 128
      epochs = 20
      history = model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs,
                          verbose=1, validation_data=(x_test, y_test))
```

```
Epoch 1/20
469/469 [=====] - 12s 24ms/step - loss: 0.2555 -
accuracy: 0.9218 - val_loss: 0.1091 - val_accuracy: 0.9640
Epoch 2/20
469/469 [=====] - 9s 19ms/step - loss: 0.1029 -
accuracy: 0.9686 - val_loss: 0.0783 - val_accuracy: 0.9771
Epoch 3/20
469/469 [=====] - 10s 21ms/step - loss: 0.0753 -
accuracy: 0.9777 - val_loss: 0.0698 - val_accuracy: 0.9778
Epoch 4/20
469/469 [=====] - 10s 21ms/step - loss: 0.0591 -
accuracy: 0.9813 - val_loss: 0.0787 - val_accuracy: 0.9757
Epoch 5/20
469/469 [=====] - 11s 23ms/step - loss: 0.0490 -
accuracy: 0.9843 - val_loss: 0.0630 - val_accuracy: 0.9818
Epoch 6/20
469/469 [=====] - 9s 19ms/step - loss: 0.0401 -
accuracy: 0.9872 - val_loss: 0.0627 - val_accuracy: 0.9830
Epoch 7/20
469/469 [=====] - 10s 21ms/step - loss: 0.0349 -
accuracy: 0.9892 - val_loss: 0.0710 - val_accuracy: 0.9808
Epoch 8/20
```

```

469/469 [=====] - 10s 22ms/step - loss: 0.0291 -
accuracy: 0.9905 - val_loss: 0.0668 - val_accuracy: 0.9837
Epoch 9/20
469/469 [=====] - 10s 22ms/step - loss: 0.0249 -
accuracy: 0.9918 - val_loss: 0.0699 - val_accuracy: 0.9824
Epoch 10/20
469/469 [=====] - 9s 18ms/step - loss: 0.0236 -
accuracy: 0.9925 - val_loss: 0.0601 - val_accuracy: 0.9860
Epoch 11/20
469/469 [=====] - 10s 21ms/step - loss: 0.0202 -
accuracy: 0.9933 - val_loss: 0.0677 - val_accuracy: 0.9849
Epoch 12/20
469/469 [=====] - 10s 21ms/step - loss: 0.0184 -
accuracy: 0.9941 - val_loss: 0.0732 - val_accuracy: 0.9827
Epoch 13/20
469/469 [=====] - 10s 21ms/step - loss: 0.0163 -
accuracy: 0.9947 - val_loss: 0.0722 - val_accuracy: 0.9848
Epoch 14/20
469/469 [=====] - 15s 33ms/step - loss: 0.0140 -
accuracy: 0.9953 - val_loss: 0.0710 - val_accuracy: 0.9848
Epoch 15/20
469/469 [=====] - 14s 31ms/step - loss: 0.0126 -
accuracy: 0.9959 - val_loss: 0.0721 - val_accuracy: 0.9855
Epoch 16/20
469/469 [=====] - 10s 20ms/step - loss: 0.0137 -
accuracy: 0.9958 - val_loss: 0.0704 - val_accuracy: 0.9857
Epoch 17/20
469/469 [=====] - 9s 19ms/step - loss: 0.0133 -
accuracy: 0.9957 - val_loss: 0.0733 - val_accuracy: 0.9856
Epoch 18/20
469/469 [=====] - 10s 21ms/step - loss: 0.0120 -
accuracy: 0.9959 - val_loss: 0.0756 - val_accuracy: 0.9844
Epoch 19/20
469/469 [=====] - 10s 21ms/step - loss: 0.0091 -
accuracy: 0.9971 - val_loss: 0.0848 - val_accuracy: 0.9843
Epoch 20/20
469/469 [=====] - 9s 19ms/step - loss: 0.0100 -
accuracy: 0.9965 - val_loss: 0.0917 - val_accuracy: 0.9836

```

```

[10]: score = model.evaluate(x_test, y_test, verbose=0)
      print('Test loss:', score[0])
      print('Test accuracy:', score[1])

```

```

Test loss: 0.09168839454650879
Test accuracy: 0.9836000204086304

```