In [1]:
```
#Pra-02
# Multiclass classification using Deep Neural Networks: Example: Use the
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
```

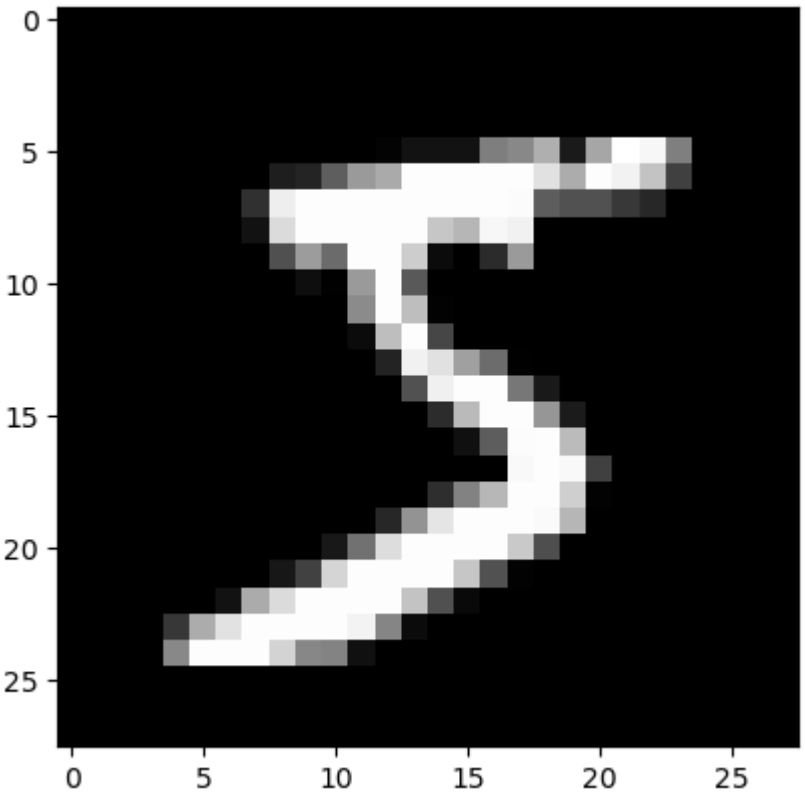
```
WARNING:tensorflow:From C:\Users\Ekata\AppData\Roaming\Python\Python310
\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_soft
max_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_
softmax_cross_entropy instead.
```

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

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-kera
s-datasets/mnist.npz (https://storage.googleapis.com/tensorflow/tf-kera
s-datasets/mnist.npz)
11490434/11490434 [==============================] - 4s 0us/step
```

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

Out[3]:
```
<matplotlib.image.AxesImage at 0x1d95f130fa0>
```



In [4]:
```
plt.show()
```

In [5]:
```
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   3  18  18  18 126 1
36
  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 2
53
  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 2
51
   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 2
41
    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 1
54
    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   0 139 253 190   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  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  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  81 240 253 253 1
19
   25   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 2
53
  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  16  93 2
52
  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 2
49
  253 249  64   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0  46 130 183 2
53
  253 207   2   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0  39 148 229 253 253 2
53
  250 182   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 2
```

```
01
   78   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  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  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   0 136 253 253 253 212 135 132  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   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0]]
```

In [6]:
```
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,)
```

In [8]:
```
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
num_classes = 10
y_train = np.eye(num_classes)[y_train]
y_test = np.eye(num_classes)[y_test]
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'])
```

```
WARNING:tensorflow:From C:\Users\Ekata\AppData\Roaming\Python\Python310
\site-packages\keras\src\backend.py:873: The name tf.get_default_graph
is deprecated. Please use tf.compat.v1.get_default_graph instead.
```

In [10]:
```
batch_size = 128
epochs = 20
history = model.fit(x_train, y_train, batch_size=batch_size,epochs=epoch
```

```
Epoch 1/20
WARNING:tensorflow:From C:\Users\Ekata\AppData\Roaming\Python\Python310
\site-packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.Ragg
edTensorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTenso
rValue instead.

WARNING:tensorflow:From C:\Users\Ekata\AppData\Roaming\Python\Python310
\site-packages\keras\src\engine\base_layer_utils.py:384: The name tf.ex
ecuting_eagerly_outside_functions is deprecated. Please use tf.compat.v
1.executing_eagerly_outside_functions instead.

469/469 [==============================] - 9s 14ms/step - loss: 3.9472
- accuracy: 0.8554 - val_loss: 0.2690 - val_accuracy: 0.9321
Epoch 2/20
469/469 [==============================] - 6s 13ms/step - loss: 0.5301
- accuracy: 0.9030 - val_loss: 0.2441 - val_accuracy: 0.9359
Epoch 3/20
469/469 [==============================] - 6s 13ms/step - loss: 0.4023
- accuracy: 0.9215 - val_loss: 0.2189 - val_accuracy: 0.9529
Epoch 4/20
469/469 [==============================] - 6s 13ms/step - loss: 0.3488
- accuracy: 0.9323 - val_loss: 0.1946 - val_accuracy: 0.9573
Epoch 5/20
469/469 [==============================] - 6s 13ms/step - loss: 0.3153
- accuracy: 0.9386 - val_loss: 0.2362 - val_accuracy: 0.9532
Epoch 6/20
469/469 [==============================] - 6s 13ms/step - loss: 0.2819
- accuracy: 0.9452 - val_loss: 0.2043 - val_accuracy: 0.9615
Epoch 7/20
469/469 [==============================] - 6s 13ms/step - loss: 0.2812
- accuracy: 0.9453 - val_loss: 0.1916 - val_accuracy: 0.9615
Epoch 8/20
469/469 [==============================] - 6s 13ms/step - loss: 0.2726
- accuracy: 0.9508 - val_loss: 0.2300 - val_accuracy: 0.9612
Epoch 9/20
469/469 [==============================] - 6s 13ms/step - loss: 0.2575
- accuracy: 0.9515 - val_loss: 0.2092 - val_accuracy: 0.9661
Epoch 10/20
469/469 [==============================] - 6s 13ms/step - loss: 0.2439
- accuracy: 0.9536 - val_loss: 0.2204 - val_accuracy: 0.9623
Epoch 11/20
469/469 [==============================] - 6s 13ms/step - loss: 0.2532
- accuracy: 0.9527 - val_loss: 0.2577 - val_accuracy: 0.9586
Epoch 12/20
469/469 [==============================] - 7s 15ms/step - loss: 0.2468
- accuracy: 0.9554 - val_loss: 0.2403 - val_accuracy: 0.9649
Epoch 13/20
469/469 [==============================] - 7s 14ms/step - loss: 0.2435
- accuracy: 0.9567 - val_loss: 0.2647 - val_accuracy: 0.9618
Epoch 14/20
469/469 [==============================] - 11s 24ms/step - loss: 0.2398
- accuracy: 0.9570 - val_loss: 0.2467 - val_accuracy: 0.9692
Epoch 15/20
469/469 [==============================] - 13s 27ms/step - loss: 0.2378
- accuracy: 0.9588 - val_loss: 0.2518 - val_accuracy: 0.9618
Epoch 16/20
469/469 [==============================] - 7s 15ms/step - loss: 0.2325
- accuracy: 0.9580 - val_loss: 0.2768 - val_accuracy: 0.9603
Epoch 17/20
469/469 [==============================] - 7s 15ms/step - loss: 0.2402
- accuracy: 0.9575 - val_loss: 0.3136 - val_accuracy: 0.9645
```

```
Epoch 18/20
469/469 [==============================] - 6s 14ms/step - loss: 0.2265
- accuracy: 0.9597 - val_loss: 0.2834 - val_accuracy: 0.9683
Epoch 19/20
469/469 [==============================] - 6s 13ms/step - loss: 0.2383
- accuracy: 0.9603 - val_loss: 0.2875 - val_accuracy: 0.9662
Epoch 20/20
469/469 [==============================] - 6s 13ms/step - loss: 0.2345
- accuracy: 0.9592 - val_loss: 0.3816 - val_accuracy: 0.9669
```

In [11]:
```
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Test loss: 0.381622850894928
Test accuracy: 0.9668999910354614
```

In [ ]:
```

```