

57. Shallow Neural Network

```
In [4]: import matplotlib.pyplot as plt
import tensorflow as tf
from keras.utils import to_categorical
layers = tf.keras.layers

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

x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0

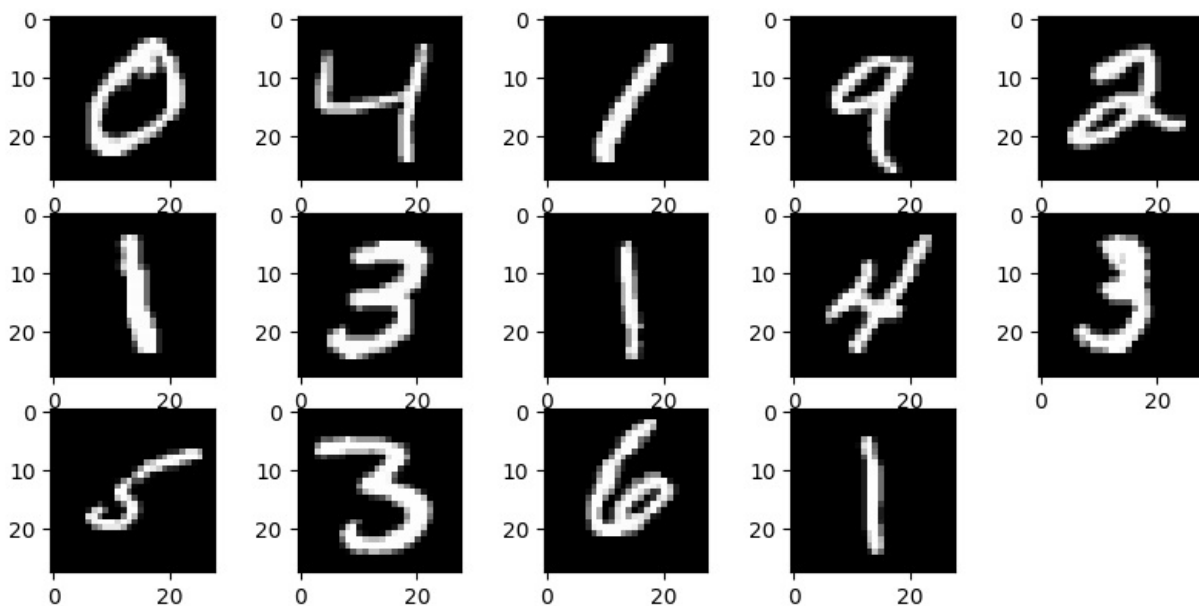
# one-hot encode the labels
y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)

plt.figure(figsize=(10,8))
for i in range(1, 15):
    # image in ith position of grid
    plt.subplot(5, 5, i)
    plt.imshow(x_train[i], cmap=plt.get_cmap('gray'))
plt.show()

model = tf.keras.Sequential()
model.add(layers.Flatten())
model.add(layers.Dense(5, activation='sigmoid'))
model.add(layers.Dense(10, activation='softmax'))

model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
model.fit(x_train, y_train, epochs=10, batch_size=32,
validation_data=(x_test, y_test))

loss, accuracy = model.evaluate(x_test, y_test)
print('\n\nTest loss', loss)
print('Test accuracy', accuracy)
```



```

Epoch 1/10
1875/1875 ————— 3s 976us/step - accuracy: 0.5176 - loss: 1.7324 - val_accuracy: 0.7813 - val_loss: 0.9753
Epoch 2/10
1875/1875 ————— 2s 903us/step - accuracy: 0.7840 - loss: 0.8945 - val_accuracy: 0.8119 - val_loss: 0.7155
Epoch 3/10
1875/1875 ————— 1s 780us/step - accuracy: 0.8161 - loss: 0.6839 - val_accuracy: 0.8385 - val_loss: 0.6063
Epoch 4/10
1875/1875 ————— 2s 1ms/step - accuracy: 0.8406 - loss: 0.5962 - val_accuracy: 0.8568 - val_loss: 0.5422
Epoch 5/10
1875/1875 ————— 2s 881us/step - accuracy: 0.8589 - loss: 0.5314 - val_accuracy: 0.8666 - val_loss: 0.5010
Epoch 6/10
1875/1875 ————— 2s 1ms/step - accuracy: 0.8656 - loss: 0.4915 - val_accuracy: 0.8698 - val_loss: 0.4773
Epoch 7/10
1875/1875 ————— 2s 980us/step - accuracy: 0.8682 - loss: 0.4770 - val_accuracy: 0.8729 - val_loss: 0.4646
Epoch 8/10
1875/1875 ————— 2s 911us/step - accuracy: 0.8740 - loss: 0.4535 - val_accuracy: 0.8756 - val_loss: 0.4492
Epoch 9/10
1875/1875 ————— 1s 752us/step - accuracy: 0.8783 - loss: 0.4393 - val_accuracy: 0.8770 - val_loss: 0.4385
Epoch 10/10
1875/1875 ————— 2s 806us/step - accuracy: 0.8791 - loss: 0.4334 - val_accuracy: 0.8815 - val_loss: 0.4312
313/313 ————— 0s 662us/step - accuracy: 0.8600 - loss: 0.4966

```

Test loss 0.4312390685081482
Test accuracy 0.8815000057220459

58. Deep Neural Network

```

In [9]: import matplotlib.pyplot as plt
import tensorflow as tf
from keras.utils import to_categorical
from keras.datasets import fashion_mnist

layers = tf.keras.layers
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
y_train = to_categorical(y_train, num_classes=10)
y_test = to_categorical(y_test, num_classes=10)

print('X_train: ' + str(x_train.shape))
print('Y_train: ' + str(y_train.shape))
print('X_test: ' + str(x_test.shape))
print('Y_test: ' + str(y_test.shape))

for i in range(1, 15):
    plt.subplot(5, 5, i)
    plt.imshow(x_train[i], cmap=plt.get_cmap('gray'))
plt.show()

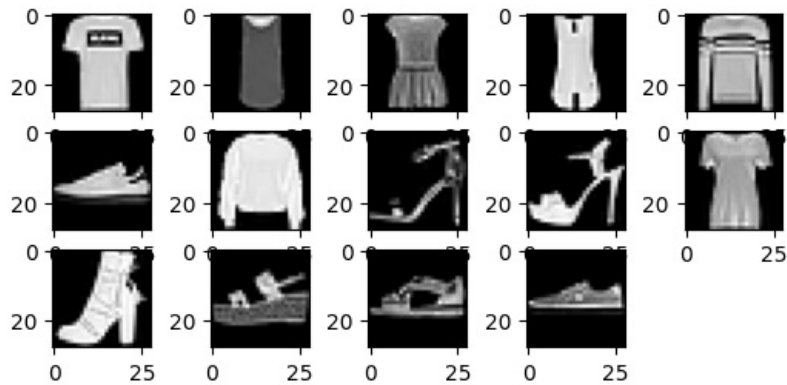
model = tf.keras.Sequential()
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(x_train, y_train, epochs=10)

loss, accuracy = model.evaluate(x_test, y_test)
print('\n\nTest loss', loss)
print('Test accuracy', accuracy)

```

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
 29515/29515 ————— 0s 1us/step
 Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
 26421880/26421880 ————— 4s 0us/step
 Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
 5148/5148 ————— 0s 1us/step
 Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
 4422102/4422102 ————— 2s 0us/step
 X_train: (60000, 28, 28)
 Y_train: (60000, 10)
 X_test: (10000, 28, 28)
 Y_test: (10000, 28, 28)



Epoch 1/10
 1875/1875 ————— 5s 2ms/step - accuracy: 0.6834 - loss: 2.1924
 Epoch 2/10
 1875/1875 ————— 3s 2ms/step - accuracy: 0.8180 - loss: 0.5118
 Epoch 3/10
 1875/1875 ————— 4s 2ms/step - accuracy: 0.8374 - loss: 0.4545
 Epoch 4/10
 1875/1875 ————— 3s 2ms/step - accuracy: 0.8456 - loss: 0.4332
 Epoch 5/10
 1875/1875 ————— 3s 2ms/step - accuracy: 0.8566 - loss: 0.3970
 Epoch 6/10
 1875/1875 ————— 3s 2ms/step - accuracy: 0.8610 - loss: 0.3891
 Epoch 7/10
 1875/1875 ————— 3s 2ms/step - accuracy: 0.8642 - loss: 0.3666
 Epoch 8/10
 1875/1875 ————— 3s 2ms/step - accuracy: 0.8726 - loss: 0.3566
 Epoch 9/10
 1875/1875 ————— 5s 1ms/step - accuracy: 0.8716 - loss: 0.3468
 Epoch 10/10
 1875/1875 ————— 3s 1ms/step - accuracy: 0.8741 - loss: 0.3429
 313/313 ————— 0s 883us/step - accuracy: 0.8602 - loss: 0.3830

Test loss 0.390420138835907
 Test accuracy 0.859000027179718

In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js