Aim: Convolutional Neural Network (CNN)

In [1]:
```
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow import keras
import numpy as np

(x_train, y_train), (x_test, y_test) = keras.datasets.fashion_mnist.load_data
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-dat
asets/train-labels-idx1-ubyte.gz (https://storage.googleapis.com/tensorflow/
tf-keras-datasets/train-labels-idx1-ubyte.gz)
29515/29515 [==============================] - 0s 2us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-dat
asets/train-images-idx3-ubyte.gz (https://storage.googleapis.com/tensorflow/
tf-keras-datasets/train-images-idx3-ubyte.gz)
26421880/26421880 [==============================] - 176s 7us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-dat
asets/t10k-labels-idx1-ubyte.gz (https://storage.googleapis.com/tensorflow/t
f-keras-datasets/t10k-labels-idx1-ubyte.gz)
5148/5148 [==============================] - 0s 0s/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-dat
asets/t10k-images-idx3-ubyte.gz (https://storage.googleapis.com/tensorflow/t
f-keras-datasets/t10k-images-idx3-ubyte.gz)
4422102/4422102 [==============================] - 8s 2us/step
```

In [2]:
```
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0

x_train = x_train.reshape(-1, 28, 28, 1)
x_test = x_test.reshape(-1, 28, 28, 1)
```

In [3]:
```
x_train.shape
```

Out[3]:
```
(60000, 28, 28, 1)
```

In [4]:
```
x_test.shape
```

Out[4]:
```
(10000, 28, 28, 1)
```

In [5]:
```
y_train.shape
```

Out[5]:
```
(60000,)
```

In [6]:
```
y_test.shape
```

Out[6]:
```
(10000,)
```

In [7]:
```
model = keras.Sequential([
    keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(28,28,1)),


    keras.layers.MaxPooling2D((2,2)),

    keras.layers.Dropout(0.25),

    keras.layers.Conv2D(64, (3,3), activation='relu'),

    keras.layers.MaxPooling2D((2,2)),

    keras.layers.Dropout(0.25),

    keras.layers.Conv2D(128, (3,3), activation='relu'),

    keras.layers.Flatten(),
    keras.layers.Dense(128, activation='relu'),

    keras.layers.Dropout(0.25),
    keras.layers.Dense(10, activation='softmax')
])
```

In [8]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        320       
                                                                 
 max_pooling2d (MaxPooling2D  (None, 13, 13, 32)       0         
 )                                                               
                                                                 
 dropout (Dropout)           (None, 13, 13, 32)        0         
                                                                 
 conv2d_1 (Conv2D)           (None, 11, 11, 64)        18496     
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 5, 5, 64)         0         
 2D)                                                             
                                                                 
 dropout_1 (Dropout)         (None, 5, 5, 64)          0         
                                                                 
 conv2d_2 (Conv2D)           (None, 3, 3, 128)         73856     
                                                                 
 flatten (Flatten)           (None, 1152)              0         
                                                                 
 dense (Dense)               (None, 128)               147584    
                                                                 
 dropout_2 (Dropout)         (None, 128)               0         
                                                                 
 dense_1 (Dense)             (None, 10)                1290      
                                                                 
=================================================================
Total params: 241,546
Trainable params: 241,546
Non-trainable params: 0
_________________________________________________________________
```

In [10]:
```
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metri

history = model.fit(x_train, y_train, epochs=3, validation_data=(x_test, y_te
```

```
Epoch 1/3
1875/1875 [==============================] - 46s 24ms/step - loss: 0.4153 -
accuracy: 0.8474 - val_loss: 0.3430 - val_accuracy: 0.8802
Epoch 2/3
1875/1875 [==============================] - 46s 24ms/step - loss: 0.3342 -
accuracy: 0.8773 - val_loss: 0.3052 - val_accuracy: 0.8883
Epoch 3/3
1875/1875 [==============================] - 45s 24ms/step - loss: 0.3038 -
accuracy: 0.8875 - val_loss: 0.2720 - val_accuracy: 0.8999
```

In [11]:
```python
test_loss, test_acc = model.evaluate(x_test, y_test)

print('Test accuracy:', test_acc)
```

```
313/313 [==============================] - 2s 7ms/step - loss: 0.2720 - accu
racy: 0.8999
Test accuracy: 0.8999000191688538
```