DL Assignment 4

Ankita Singh || Roll no: 57

Problem Statement: Convolutional neural network (CNN) Use MNIST Fashion ataset and create a classifier to classify fashion clothing into categories.

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

In [2]:
```
fashion_mnist = tf.keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515 [==============================] - 0s 3us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880 [==============================] - 3s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 [==============================] - 0s 0s/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 [==============================] - 1s 0us/step
```

In [3]:
```
class_names = ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ank
```

In [7]:
```
# training data
train_images.shape
```

Out[7]:
```
(60000, 28, 28)
```

In [8]:
```
test_images.shape
```

Out[8]:
```
(10000, 28, 28)
```

In [9]:
```
train_labels.shape
```

Out[9]:
```
(60000,)
```

In [10]:
```
test_labels.shape
```

Out[10]:
```
(10000,)
```

In [11]:
```
print(train_images[0])
```

```
[[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   1   0   0  13  73   0
    0   1   4   0   0   0   0   1   1   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   3   0  36 136 127  62
   54   0   0   0   1   3   4   0   0   3]
 [  0   0   0   0   0   0   0   0   0   0   0   0   6   0 102 204 176 134
  144 123  23   0   0   0   0  12  10   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0 155 236 207 178
  107 156 161 109  64  23  77 130  72  15]
 [  0   0   0   0   0   0   0   0   0   0   0   1   0  69 207 223 218 216
  216 163 127 121 122 146 141  88 172  66]
 [  0   0   0   0   0   0   0   0   0   1   1   1   0 200 232 232 233 229
  223 223 215 213 164 127 123 196 229   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0 183 225 216 223 228
  235 227 224 222 224 221 223 245 173   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0 193 228 218 213 198
  180 212 210 211 213 223 220 243 202   0]
 [  0   0   0   0   0   0   0   0   0   1   3   0  12 219 220 212 218 192
  169 227 208 218 224 212 226 197 209  52]
 [  0   0   0   0   0   0   0   0   0   0   6   0  99 244 222 220 218 203
  198 221 215 213 222 220 245 119 167  56]
 [  0   0   0   0   0   0   0   0   0   4   0   0  55 236 228 230 228 240
  232 213 218 223 234 217 217 209  92   0]
 [  0   0   1   4   6   7   2   0   0   0   0   0 237 226 217 223 222 219
  222 221 216 223 229 215 218 255  77   0]
 [  0   3   0   0   0   0   0   0   0  62 145 204 228 207 213 221 218 208
  211 218 224 223 219 215 224 244 159   0]
 [  0   0   0   0  18  44  82 107 189 228 220 222 217 226 200 205 211 230
  224 234 176 188 250 248 233 238 215   0]
 [  0  57 187 208 224 221 224 208 204 214 208 209 200 159 245 193 206 223
  255 255 221 234 221 211 220 232 246   0]
 [  3 202 228 224 221 211 211 214 205 205 205 220 240  80 150 255 229 221
  188 154 191 210 204 209 222 228 225   0]
 [ 98 233 198 210 222 229 229 234 249 220 194 215 217 241  65  73 106 117
  168 219 221 215 217 223 223 224 229  29]
 [ 75 204 212 204 193 205 211 225 216 185 197 206 198 213 240 195 227 245
  239 223 218 212 209 222 220 221 230  67]
 [ 48 203 183 194 213 197 185 190 194 192 202 214 219 221 220 236 225 216
  199 206 186 181 177 172 181 205 206 115]
 [  0 122 219 193 179 171 183 196 204 210 213 207 211 210 200 196 194 191
  195 191 198 192 176 156 167 177 210  92]
 [  0   0  74 189 212 191 175 172 175 181 185 188 189 188 193 198 204 209
  210 210 211 188 188 194 192 216 170   0]
 [  2   0   0   0  66 200 222 237 239 242 246 243 244 221 220 193 191 179
  182 182 181 176 166 168  99  58   0   0]
 [  0   0   0   0   0   0   0  40  61  44  72  41  35   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   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 [12]:
```
train_labels[0]
```

Out[12]:
```
9
```

In [13]:
```
train_images = train_images / 255
test_images = test_images / 255
```

In [14]:
```
train_images[0]
```

Out[14]:
```
array([[0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
```

```
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.00392157, 0.        , 0.        ,
 0.05098039, 0.28627451, 0.        , 0.        , 0.00392157,
 0.01568627, 0.        , 0.        , 0.        , 0.        ,
 0.00392157, 0.00392157, 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.01176471, 0.        , 0.14117647,
 0.53333333, 0.49803922, 0.24313725, 0.21176471, 0.        ,
 0.        , 0.        , 0.00392157, 0.01176471, 0.01568627,
 0.        , 0.        , 0.01176471],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.02352941, 0.        , 0.4       ,
 0.8       , 0.69019608, 0.5254902 , 0.56470588, 0.48235294,
 0.09019608, 0.        , 0.        , 0.        , 0.        ,
 0.04705882, 0.03921569, 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.60784314,
 0.9254902 , 0.81176471, 0.69803922, 0.41960784, 0.61176471,
 0.63137255, 0.42745098, 0.25098039, 0.09019608, 0.30196078,
 0.50980392, 0.28235294, 0.05882353],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.00392157, 0.        , 0.27058824, 0.81176471,
 0.8745098 , 0.85490196, 0.84705882, 0.84705882, 0.63921569,
 0.49803922, 0.4745098 , 0.47843137, 0.57254902, 0.55294118,
 0.34509804, 0.6745098 , 0.25882353],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.00392157,
 0.00392157, 0.00392157, 0.        , 0.78431373, 0.90980392,
 0.90980392, 0.91372549, 0.89803922, 0.8745098 , 0.8745098 ,
 0.84313725, 0.83529412, 0.64313725, 0.49803922, 0.48235294,
 0.76862745, 0.89803922, 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.71764706, 0.88235294,
 0.84705882, 0.8745098 , 0.89411765, 0.92156863, 0.89019608,
 0.87843137, 0.87058824, 0.87843137, 0.86666667, 0.8745098 ,
 0.96078431, 0.67843137, 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.75686275, 0.89411765,
 0.85490196, 0.83529412, 0.77647059, 0.70588235, 0.83137255,
 0.82352941, 0.82745098, 0.83529412, 0.8745098 , 0.8627451 ,
 0.95294118, 0.79215686, 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.00392157,
 0.01176471, 0.        , 0.04705882, 0.85882353, 0.8627451 ,
 0.83137255, 0.85490196, 0.75294118, 0.6627451 , 0.89019608,
 0.81568627, 0.85490196, 0.87843137, 0.83137255, 0.88627451,
 0.77254902, 0.81960784, 0.20392157],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.02352941, 0.        , 0.38823529, 0.95686275, 0.87058824,
 0.8627451 , 0.85490196, 0.79607843, 0.77647059, 0.86666667,
 0.84313725, 0.83529412, 0.87058824, 0.8627451 , 0.96078431,
 0.46666667, 0.65490196, 0.21960784],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.01568627,
 0.        , 0.        , 0.21568627, 0.9254902 , 0.89411765,
 0.90196078, 0.89411765, 0.94117647, 0.90980392, 0.83529412,
 0.85490196, 0.8745098 , 0.91764706, 0.85098039, 0.85098039,
 0.81960784, 0.36078431, 0.        ],
[0.        , 0.        , 0.00392157, 0.01568627, 0.02352941,
 0.02745098, 0.00784314, 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.92941176, 0.88627451, 0.85098039,
 0.8745098 , 0.87058824, 0.85882353, 0.87058824, 0.86666667,
 0.84705882, 0.8745098 , 0.89803922, 0.84313725, 0.85490196,
 1.        , 0.30196078, 0.        ],
[0.        , 0.01176471, 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.24313725,
 0.56862745, 0.8       , 0.89411765, 0.81176471, 0.83529412,
 0.86666667, 0.85490196, 0.81568627, 0.82745098, 0.85490196,
 0.87843137, 0.8745098 , 0.85882353, 0.84313725, 0.87843137,
 0.95686275, 0.62352941, 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.07058824,
 0.17254902, 0.32156863, 0.41960784, 0.74117647, 0.89411765,
 0.8627451 , 0.87058824, 0.85098039, 0.88627451, 0.78431373,
 0.80392157, 0.82745098, 0.90196078, 0.87843137, 0.91764706,
```

```
         0.69019608, 0.7372549 , 0.98039216, 0.97254902, 0.91372549,
         0.93333333, 0.84313725, 0.        ],
        [0.        , 0.22352941, 0.73333333, 0.81568627, 0.87843137,
         0.86666667, 0.87843137, 0.81568627, 0.8       , 0.83921569,
         0.81568627, 0.81960784, 0.78431373, 0.62352941, 0.96078431,
         0.75686275, 0.80784314, 0.8745098 , 1.        , 1.        ,
         0.86666667, 0.91764706, 0.86666667, 0.82745098, 0.8627451 ,
         0.90980392, 0.96470588, 0.        ],
        [0.01176471, 0.79215686, 0.89411765, 0.87843137, 0.86666667,
         0.82745098, 0.82745098, 0.83921569, 0.80392157, 0.80392157,
         0.80392157, 0.8627451 , 0.94117647, 0.31372549, 0.58823529,
         1.        , 0.89803922, 0.86666667, 0.7372549 , 0.60392157,
         0.74901961, 0.82352941, 0.8       , 0.81960784, 0.87058824,
         0.89411765, 0.88235294, 0.        ],
        [0.38431373, 0.91372549, 0.77647059, 0.82352941, 0.87058824,
         0.89803922, 0.89803922, 0.91764706, 0.97647059, 0.8627451 ,
         0.76078431, 0.84313725, 0.85098039, 0.94509804, 0.25490196,
         0.28627451, 0.41568627, 0.45882353, 0.65882353, 0.85882353,
         0.86666667, 0.84313725, 0.85098039, 0.8745098 , 0.8745098 ,
         0.87843137, 0.89803922, 0.11372549],
        [0.29411765, 0.8       , 0.83137255, 0.8       , 0.75686275,
         0.80392157, 0.82745098, 0.88235294, 0.84705882, 0.7254902 ,
         0.77254902, 0.80784314, 0.77647059, 0.83529412, 0.94117647,
         0.76470588, 0.89019608, 0.96078431, 0.9372549 , 0.8745098 ,
         0.85490196, 0.83137255, 0.81960784, 0.87058824, 0.8627451 ,
         0.86666667, 0.90196078, 0.2627451 ],
        [0.18823529, 0.79607843, 0.71764706, 0.76078431, 0.83529412,
         0.77254902, 0.7254902 , 0.74509804, 0.76078431, 0.75294118,
         0.79215686, 0.83921569, 0.85882353, 0.86666667, 0.8627451 ,
         0.9254902 , 0.88235294, 0.84705882, 0.78039216, 0.80784314,
         0.72941176, 0.70980392, 0.69411765, 0.6745098 , 0.70980392,
         0.80392157, 0.80784314, 0.45098039],
        [0.        , 0.47843137, 0.85882353, 0.75686275, 0.70196078,
         0.67058824, 0.71764706, 0.76862745, 0.8       , 0.82352941,
         0.83529412, 0.81176471, 0.82745098, 0.82352941, 0.78431373,
         0.76862745, 0.76078431, 0.74901961, 0.76470588, 0.74901961,
         0.77647059, 0.75294118, 0.69019608, 0.61176471, 0.65490196,
         0.69411765, 0.82352941, 0.36078431],
        [0.        , 0.        , 0.29019608, 0.74117647, 0.83137255,
         0.74901961, 0.68627451, 0.6745098 , 0.68627451, 0.70980392,
         0.7254902 , 0.7372549 , 0.74117647, 0.7372549 , 0.75686275,
         0.77647059, 0.8       , 0.81960784, 0.82352941, 0.82352941,
         0.82745098, 0.7372549 , 0.7372549 , 0.76078431, 0.75294118,
         0.84705882, 0.66666667, 0.        ],
        [0.00784314, 0.        , 0.        , 0.        , 0.25882353,
         0.78431373, 0.87058824, 0.92941176, 0.9372549 , 0.94901961,
         0.96470588, 0.95294118, 0.95686275, 0.86666667, 0.8627451 ,
         0.75686275, 0.74901961, 0.70196078, 0.71372549, 0.71372549,
         0.70980392, 0.69019608, 0.65098039, 0.65882353, 0.38823529,
         0.22745098, 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.15686275, 0.23921569, 0.17254902,
         0.28235294, 0.16078431, 0.1372549 , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 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 [15]:
```
plt.figure(figsize =(10, 10))
```

Out[15]:
```
<Figure size 1000x1000 with 0 Axes>
```

```
<Figure size 1000x1000 with 0 Axes>
```

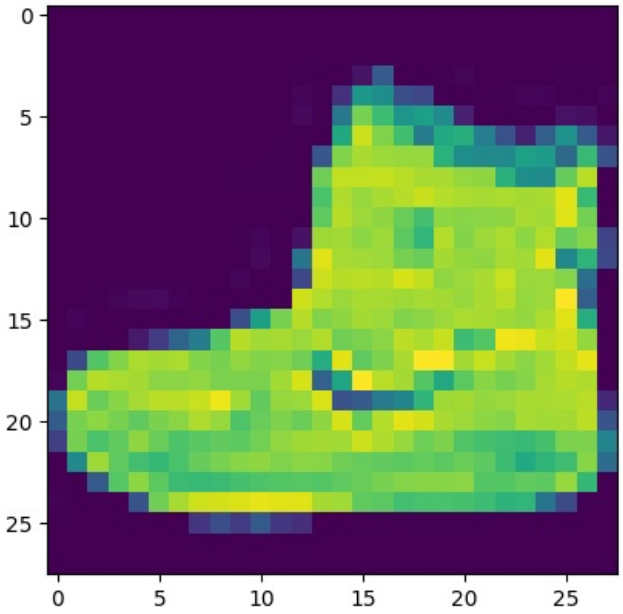
In [18]:
```
for i in range(25):
    plt.subplot(5,5, i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(train_images[i], cmap = plt.cm.binary)
    plt.xlabel(class_names[train_labels[i]])

plt.show
```

Out[18]: <function matplotlib.pyplot.show(close=None, block=None)>

In [19]:
```
plt.imshow(train_images[0])
```

Out[19]: <matplotlib.image.AxesImage at 0x24639862ec8>



In [20]:
```
train_images = train_images.reshape(-1, 28, 28, 1)
```

In [21]:
```
test_images = test_images.reshape(-1, 28, 28, 1)
```

In [22]:
```
train_images.shape
```

Out[22]: (60000, 28, 28, 1)

In [23]:
```
test_images.shape
```

Out[23]: (10000, 28, 28, 1)

In [24]:
```
train_labels.shape
```

Out[24]: (60000,)

In [25]:
```
test_labels.shape
```

Out[25]: (10000,)

In [27]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import (Conv2D, Dense, Flatten, Dropout, MaxPool2D)
```

In [28]:
```
model = Sequential()
```

In [29]:
```
model.add(Conv2D(32, (3,3), activation="relu", input_shape=(28, 28, 1)))
```

In [29]:
```
model.add(Conv2D(32,(3,3), activation= 'relu' , input_shape=(28, 28, 1)))
```

In [30]:
```
model.add(MaxPool2D((2,2)))
model.add(Dropout(0.25))
model.add(Conv2D(64, (3,3), activation = "relu"))
model.add(MaxPool2D((2,2)))
model.add(Dropout(0.25))
model.add(Conv2D(128, (3,3), activation = "relu"))
model.add(Flatten())
model.add(Dense(units = 128, activation = "relu"))
model.add(Dropout(0.25))
model.add(Dense(units=10, activation = "softmax"))
```

In [31]:
```
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 [32]:
```
model.compile(optimizer = "adam", loss = "sparse_categorical_crossentropy", metrics = ["accuracy"])
```

In [34]:
```
history = model.fit(train_images, train_labels, epochs=5, verbose=1, validation_data=(test_images, test_labels))
```

```
Epoch 1/5
1875/1875 [==============================] - 35s 19ms/step - loss: 0.3677 - accuracy: 0.8664 - val_loss: 0.3274
- val_accuracy: 0.8796
Epoch 2/5
1875/1875 [==============================] - 34s 18ms/step - loss: 0.3199 - accuracy: 0.8839 - val_loss: 0.2909
- val_accuracy: 0.8936
Epoch 3/5
1875/1875 [==============================] - 35s 19ms/step - loss: 0.2966 - accuracy: 0.8905 - val_loss: 0.2793
- val_accuracy: 0.8957
Epoch 4/5
1875/1875 [==============================] - 37s 19ms/step - loss: 0.2792 - accuracy: 0.8967 - val_loss: 0.2609
- val_accuracy: 0.9071
Epoch 5/5
1875/1875 [==============================] - 36s 19ms/step - loss: 0.2655 - accuracy: 0.9010 - val_loss: 0.2648
- val_accuracy: 0.9001
```

In [35]:
```
test_loss, test_acc = model.evaluate(test_images, test_labels)
```

```
313/313 [==============================] - 1s 5ms/step - loss: 0.2648 - accuracy: 0.9001
```

In [36]:
```
print('test_loss = ' , test_loss)
```

```
test_loss =  0.2648467421531677
```

In [37]:
```
print('test_acc = ', test_acc)
```

```
test_acc =  0.9000999927520752
```

In [ ]: