In [50]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
```

In [51]:
```
train = pd.read_csv(r'/Users/kvno1ahmednagar/Desktop/DATASETS/fashionmnist-2/fashion-mnist_train.csv')
test = pd.read_csv(r'/Users/kvno1ahmednagar/Desktop/DATASETS/fashionmnist-2/fashion-mnist_test.csv')
```

In [52]:
```
from sklearn.model_selection import train_test_split
```

In [53]:
```
train.head()
```

Out[53]:

	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel775	pixel776	pixel777	pixel778	pixel77
0	2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
1	9	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
2	6	0	0	0	0	0	0	0	5	0	...	0	0	0	30	4
3	0	0	0	0	1	2	0	0	0	0	...	3	0	0	0	
4	3	0	0	0	0	0	0	0	0	0	...	0	0	0	0	

5 rows × 785 columns

In [54]:
```
train.shape
```

Out[54]:
```
(60000, 785)
```

In [55]:
```
test.shape
```

Out[55]:
```
(10000, 785)
```

In [56]:
```
train_data = np.array(train,dtype = 'float32')
test_data = np.array(test,dtype = 'float32')
```

In [57]:
```
X_train = train_data[:,1:]
Y_train = train_data[:,0]
```

In [13]:
```
X_train.head()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-13-73020798007e> in <module>
----> 1 X_train.head()

AttributeError: 'numpy.ndarray' object has no attribute 'head'
```

In [58]:
```
X_train.shape
```

Out[58]:
```
(60000, 784)
```

In [59]:
```
Y_train.shape
```

Out[59]:
```
(60000,)
```

In [60]:
```
X_test = test_data[:,1:]
Y_test = test_data[:,0]
```

In [61]:
```
X_test.shape
```

Out[61]:
```
(10000, 784)
```

In [62]:
```
Y_test.shape
```

Out[62]:
```
(10000,)
```

In [63]:
```
X_train, X_valid , Y_train, Y_valid = train_test_split(X_train,Y_train, test_size = 0.2)
```

In [64]:
```
X_train.shape
```

Out[64]:
```
(48000, 784)
```

In [65]:
```
Y_train.shape
```

Out[65]:
```
(48000,)
```

In [66]:
```
X_valid.shape
```

Out[66]:
```
(12000, 784)
```

In [67]:
```
Y_valid.shape
```

Out[67]:
```
(12000,)
```

In [68]:
```
rows = 28
cols = 28
im_shape = (rows,cols,1)


X_train = X_train.reshape(X_train.shape[0],*im_shape)
X_test = X_test.reshape(X_test.shape[0],*im_shape)
X_valid = X_valid.reshape(X_valid.shape[0],*im_shape)
```

In [69]:
```
X_train.shape
```

Out[69]:
```
(48000, 28, 28, 1)
```

In [70]:
```
from keras.utils import np_utils
```

In [76]:
```
Y_train = np_utils.to_categorical(Y_train,num_classes = 10)
Y_valid = np_utils.to_categorical(Y_valid,num_classes = 10)
Y_test = np_utils.to_categorical(Y_test,num_classes = 10)
```

In [71]:
```
import keras
from keras.models import Sequential
from keras.layers import Conv2D,MaxPooling2D,BatchNormalization,Flatten,Dense,Dropout
from keras.optimizers import Adam
```

In [72]:
```
from keras.regularizers import l2
```

In [73]:
```
model = Sequential()
model.add(Conv2D(filters = 64, kernel_size = (3,3), strides = (1,1),activation = 'relu', input_shape = (28,28,1)))
model.add(MaxPooling2D(pool_size = (3,3), strides = (1,1)))
model.add(Dropout(0.25))

model.add(Conv2D(filters = 32, kernel_size = (3,3), strides = (1,1),activation = 'relu'))
model.add(MaxPooling2D(pool_size = (2,2), strides = (1,1)))
model.add(Dropout(0.15))

model.add(Conv2D(filters = 128, kernel_size = (5,5), strides = (1,1),activation = 'relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size = (3,3), strides = (1,1)))
model.add(Dropout(0.20))

model.add(Flatten())
model.add(Dense(128,activation = 'relu'))
model.add(Dense(10,activation = 'softmax'))
```

In [74]:
```
model.compile(loss = 'categorical_crossentropy', optimizer = Adam(lr = 0.001), metrics = ['accuracy'])
```

In [77]:
```
model.fit(X_train, Y_train, batch_size = 128, epochs = 15, verbose = 1, validation_data = [X_valid,Y_valid])
```

```
WARNING:tensorflow:From /opt/anaconda3/lib/python3.7/site-packages/keras/backend/tensorflow_backend.py:422: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

Train on 48000 samples, validate on 12000 samples
Epoch 1/15
48000/48000 [==============================] - 1091s 23ms/step - loss: 1.8568 - accuracy: 0.7469 - val_loss: 1.5247 - val_accuracy: 0.7204
Epoch 2/15
48000/48000 [==============================] - 923s 19ms/step - loss: 0.4347 - accuracy: 0.8491 - val_loss: 0.5780 - val_accuracy: 0.8310
Epoch 3/15
48000/48000 [==============================] - 1407s 29ms/step - loss: 0.3520 - accuracy: 0.8740 - val_loss: 0.5425 - val_accuracy: 0.8340
Epoch 4/15
48000/48000 [==============================] - 536s 11ms/step - loss: 0.3233 - accuracy: 0.8816 - val_loss: 0.3778 - val_accuracy: 0.8747
Epoch 5/15
48000/48000 [==============================] - 561s 12ms/step - loss: 0.3002 - accuracy: 0.8916 - val_loss: 0.3413 - val_accuracy: 0.8820
Epoch 6/15
48000/48000 [==============================] - 2268s 47ms/step - loss: 0.2767 - accuracy: 0.8994 - val_loss: 0.3778 - val_accuracy: 0.8746
Epoch 7/15
48000/48000 [==============================] - 637s 13ms/step - loss: 0.2601 - accuracy: 0.9034 - val_loss: 0.3369 - val_accuracy: 0.8877
Epoch 8/15
48000/48000 [==============================] - 1037s 22ms/step - loss: 0.2529 - accuracy: 0.9086 - val_loss: 0.3056 - val_accuracy: 0.8931
Epoch 9/15
48000/48000 [==============================] - 1553s 32ms/step - loss: 0.2353 - accuracy: 0.9130 - val_loss: 0.3707 - val_accuracy: 0.8860
Epoch 10/15
48000/48000 [==============================] - 554s 12ms/step - loss: 0.2223 - accuracy: 0.9183 - val_loss: 0.2741 - val_accuracy: 0.9053
Epoch 11/15
48000/48000 [==============================] - 620s 13ms/step - loss: 0.2200 - accuracy: 0.9194 - val_loss: 0.3559 - val_accuracy: 0.8822
Epoch 12/15
48000/48000 [==============================] - 589s 12ms/step - loss: 0.2072 - accuracy: 0.9231 - val_loss: 0.2797 - val_accuracy: 0.9045
Epoch 13/15
48000/48000 [==============================] - 548s 11ms/step - loss: 0.2010 - accuracy: 0.9263 - val_loss: 0.3091 - val_accuracy: 0.9003
Epoch 14/15
48000/48000 [==============================] - 548s 11ms/step - loss: 0.1951 - accuracy: 0.9286 - val_loss: 0.2799 - val_accuracy: 0.9109
Epoch 15/15
48000/48000 [==============================] - 544s 11ms/step - loss: 0.1887 - accuracy: 0.9295 - val_loss: 0.3239 - val_accuracy: 0.8982
```

Out[77]:
```
<keras.callbacks.callbacks.History at 0x64780bed0>
```

In [78]:
```
score = model.evaluate(X_test,Y_test, verbose = 0)
print('Test Loss: {:.4f}'.format(score[0]))
print('Test Acc: {:.4f}'.format(score[1]))
```