ASSIGNMENT NO. 1

```
from sklearn.datasets import load_boston
import pandas as pd

boston_dataset = load_boston()

df = pd.DataFrame(boston_dataset.data,
columns=boston_dataset.feature_names)
df['MEDV'] = boston_dataset.target

df.head(n=10)
```

In [3]:

```
# If load_boston does not work then download the data and use this.
# Data :
https://github.com/afnan47/sem8/blob/master/DL/1_boston_housing.csv
import pandas as pd
df = pd.read_csv("./1_boston_housing.csv")
```

In [4]:

```
from sklearn.model_selection import train_test_split

X = df.loc[:, df.columns != 'MEDV']
y = df.loc[:, df.columns == 'MEDV']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=123)
```

In [5]:

```
from sklearn.preprocessing import MinMaxScaler
mms = MinMaxScaler()
mms.fit(X_train)
X_train = mms.transform(X_train)
X_test = mms.transform(X_test)
```

In [1]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

model = Sequential()

model.add(Dense(128, input_shape=(13, ), activation='relu',
name='dense_1'))
model.add(Dense(64, activation='relu', name='dense_2'))
model.add(Dense(1, activation='linear', name='dense_output'))

model.compile(optimizer='adam', loss='mse', metrics=['mae'])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_1 (Dense)             (None, 128)               1792

 dense_2 (Dense)             (None, 64)                8256

 dense_output (Dense)        (None, 1)                 65
```

```
=================================================================
Total params: 10,113
Trainable params: 10,113
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:

```
history = model.fit(X_train, y_train, epochs=100, validation_split=0.05,
verbose = 1)
```

```
Epoch 1/100
11/11 [==============================] - 2s 57ms/step - loss: 1468.4281 - m
ae: 32.0408 - val_loss: 104.8761 - val_mae: 6.1343
Epoch 2/100
11/11 [==============================] - 0s 9ms/step - loss: 291.9155 - mae
: 13.9797 - val_loss: 149.5561 - val_mae: 11.4423
Epoch 3/100
11/11 [==============================] - 0s 7ms/step - loss: 151.9729 - mae
: 10.3968 - val_loss: 141.8171 - val_mae: 7.9483
Epoch 4/100
11/11 [==============================] - 0s 8ms/step - loss: 87.9686 - mae:
7.0836 - val_loss: 96.4421 - val_mae: 8.3894
Epoch 5/100
11/11 [==============================] - 0s 10ms/step - loss: 66.2962 - mae
: 5.8493 - val_loss: 79.1745 - val_mae: 5.6716
Epoch 6/100
11/11 [==============================] - 0s 13ms/step - loss: 59.8653 - mae
: 5.6803 - val_loss: 81.4188 - val_mae: 5.8136
Epoch 7/100
11/11 [==============================] - 0s 7ms/step - loss: 57.8316 - mae:
5.3080 - val_loss: 78.1150 - val_mae: 6.1632
Epoch 8/100
11/11 [==============================] - 0s 8ms/step - loss: 55.0569 - mae:
5.0945 - val_loss: 78.3047 - val_mae: 6.4755
Epoch 9/100
11/11 [==============================] - 0s 4ms/step - loss: 56.2966 - mae:
5.2586 - val_loss: 78.7134 - val_mae: 6.3377
Epoch 10/100
11/11 [==============================] - 0s 4ms/step - loss: 56.3019 - mae:
5.3718 - val_loss: 85.3209 - val_mae: 7.5740
Epoch 11/100
11/11 [==============================] - 0s 4ms/step - loss: 65.1538 - mae:
6.0167 - val_loss: 82.1899 - val_mae: 5.8459
Epoch 12/100
11/11 [==============================] - 0s 4ms/step - loss: 60.3453 - mae:
5.9333 - val_loss: 90.0579 - val_mae: 6.0229
Epoch 13/100
11/11 [==============================] - 0s 8ms/step - loss: 59.1005 - mae:
5.7201 - val_loss: 93.0269 - val_mae: 6.0567
Epoch 14/100
11/11 [==============================] - 0s 8ms/step - loss: 55.1477 - mae:
5.3231 - val_loss: 78.0659 - val_mae: 6.2727
Epoch 15/100
11/11 [==============================] - 0s 11ms/step - loss: 49.6804 - mae
: 4.9658 - val_loss: 78.2121 - val_mae: 6.2786
Epoch 16/100
11/11 [==============================] - 0s 17ms/step - loss: 51.0314 - mae
: 5.1608 - val_loss: 82.2059 - val_mae: 6.0243
```

```
Epoch 17/100
11/11 [==============================] - 0s 12ms/step - loss: 50.8898 - mae
: 5.0943 - val_loss: 79.5624 - val_mae: 6.7917
Epoch 18/100
11/11 [==============================] - 0s 7ms/step - loss: 48.4888 - mae:
4.8631 - val_loss: 79.4000 - val_mae: 6.9262
Epoch 19/100
11/11 [==============================] - 0s 8ms/step - loss: 47.7700 - mae:
4.8788 - val_loss: 78.2341 - val_mae: 6.6031
Epoch 20/100
11/11 [==============================] - 0s 10ms/step - loss: 46.3437 - mae
: 4.8531 - val_loss: 79.5777 - val_mae: 6.2096
Epoch 21/100
11/11 [==============================] - 0s 15ms/step - loss: 46.8095 - mae
: 4.8446 - val_loss: 79.9989 - val_mae: 6.2106
Epoch 22/100
11/11 [==============================] - 0s 11ms/step - loss: 46.5442 - mae
: 4.8957 - val_loss: 80.0767 - val_mae: 6.3410
Epoch 23/100
11/11 [==============================] - 0s 5ms/step - loss: 46.3140 - mae:
4.7125 - val_loss: 79.3820 - val_mae: 6.3287
Epoch 24/100
11/11 [==============================] - 0s 4ms/step - loss: 45.1415 - mae:
4.6794 - val_loss: 78.1009 - val_mae: 6.8739
Epoch 25/100
11/11 [==============================] - 0s 5ms/step - loss: 46.8965 - mae:
4.8287 - val_loss: 77.1064 - val_mae: 6.9487
Epoch 26/100
11/11 [==============================] - 0s 4ms/step - loss: 45.7256 - mae:
4.8219 - val_loss: 80.3456 - val_mae: 7.5219
Epoch 27/100
11/11 [==============================] - 0s 3ms/step - loss: 45.4823 - mae:
4.8070 - val_loss: 86.8667 - val_mae: 7.9910
Epoch 28/100
11/11 [==============================] - 0s 4ms/step - loss: 52.8585 - mae:
5.2916 - val_loss: 75.9413 - val_mae: 6.9428
Epoch 29/100
11/11 [==============================] - 0s 4ms/step - loss: 43.7494 - mae:
4.7075 - val_loss: 75.8774 - val_mae: 6.8408
Epoch 30/100
11/11 [==============================] - 0s 4ms/step - loss: 41.9384 - mae:
4.5150 - val_loss: 74.4587 - val_mae: 6.8040
Epoch 31/100
11/11 [==============================] - 0s 4ms/step - loss: 42.0174 - mae:
4.6436 - val_loss: 72.5272 - val_mae: 6.0201
Epoch 32/100
11/11 [==============================] - 0s 3ms/step - loss: 43.1446 - mae:
4.7250 - val_loss: 73.3913 - val_mae: 6.0527
Epoch 33/100
11/11 [==============================] - 0s 5ms/step - loss: 42.5598 - mae:
4.5334 - val_loss: 70.9002 - val_mae: 6.6792
Epoch 34/100
11/11 [==============================] - 0s 4ms/step - loss: 44.9650 - mae:
4.9500 - val_loss: 70.4410 - val_mae: 6.3134
Epoch 35/100
11/11 [==============================] - 0s 4ms/step - loss: 44.2963 - mae:
4.7071 - val_loss: 76.1223 - val_mae: 5.9119
```

```
Epoch 36/100
11/11 [==============================] - 0s 4ms/step - loss: 44.2435 - mae:
4.7664 - val_loss: 78.8820 - val_mae: 5.6082
Epoch 37/100
11/11 [==============================] - 0s 4ms/step - loss: 39.0730 - mae:
4.4803 - val_loss: 69.7008 - val_mae: 6.2753
Epoch 38/100
11/11 [==============================] - 0s 4ms/step - loss: 40.0643 - mae:
4.4467 - val_loss: 70.4070 - val_mae: 5.9936
Epoch 39/100
11/11 [==============================] - 0s 3ms/step - loss: 39.7373 - mae:
4.5365 - val_loss: 70.5524 - val_mae: 5.7211
Epoch 40/100
11/11 [==============================] - 0s 4ms/step - loss: 40.1192 - mae:
4.3651 - val_loss: 68.0985 - val_mae: 6.1247
Epoch 41/100
11/11 [==============================] - 0s 4ms/step - loss: 39.6499 - mae:
4.6153 - val_loss: 67.4054 - val_mae: 6.3753
Epoch 42/100
11/11 [==============================] - 0s 4ms/step - loss: 41.4035 - mae:
4.6629 - val_loss: 67.4899 - val_mae: 5.8294
Epoch 43/100
11/11 [==============================] - 0s 4ms/step - loss: 40.0755 - mae:
4.5632 - val_loss: 68.8983 - val_mae: 5.8097
Epoch 44/100
11/11 [==============================] - 0s 4ms/step - loss: 39.9469 - mae:
4.5411 - val_loss: 67.7507 - val_mae: 6.1112
Epoch 45/100
11/11 [==============================] - 0s 4ms/step - loss: 42.2766 - mae:
4.8104 - val_loss: 76.0111 - val_mae: 7.6458
Epoch 46/100
11/11 [==============================] - 0s 4ms/step - loss: 41.7172 - mae:
4.8151 - val_loss: 67.7236 - val_mae: 5.6180
Epoch 47/100
11/11 [==============================] - 0s 4ms/step - loss: 36.0112 - mae:
4.1907 - val_loss: 65.1950 - val_mae: 5.4492
Epoch 48/100
11/11 [==============================] - 0s 5ms/step - loss: 34.7964 - mae:
4.2764 - val_loss: 73.0951 - val_mae: 5.5137
Epoch 49/100
11/11 [==============================] - 0s 6ms/step - loss: 39.2266 - mae:
4.4734 - val_loss: 65.3500 - val_mae: 5.4734
Epoch 50/100
11/11 [==============================] - 0s 4ms/step - loss: 36.9901 - mae:
4.4038 - val_loss: 65.8660 - val_mae: 6.4395
Epoch 51/100
11/11 [==============================] - 0s 4ms/step - loss: 36.3954 - mae:
4.3774 - val_loss: 66.5157 - val_mae: 5.5489
Epoch 52/100
11/11 [==============================] - 0s 3ms/step - loss: 35.5653 - mae:
4.3058 - val_loss: 65.4602 - val_mae: 5.6216
Epoch 53/100
11/11 [==============================] - 0s 4ms/step - loss: 34.9797 - mae:
4.3144 - val_loss: 71.0284 - val_mae: 5.6592
Epoch 54/100
11/11 [==============================] - 0s 4ms/step - loss: 34.9526 - mae:
4.3036 - val_loss: 79.4032 - val_mae: 5.6828
```

```
Epoch 55/100
11/11 [==============================] - 0s 4ms/step - loss: 35.1356 - mae:
4.1692 - val_loss: 63.7029 - val_mae: 6.1306
Epoch 56/100
11/11 [==============================] - 0s 4ms/step - loss: 35.1635 - mae:
4.3432 - val_loss: 64.8605 - val_mae: 5.7787
Epoch 57/100
11/11 [==============================] - 0s 4ms/step - loss: 34.1723 - mae:
4.2143 - val_loss: 73.1095 - val_mae: 5.7776
Epoch 58/100
11/11 [==============================] - 0s 4ms/step - loss: 34.2169 - mae:
4.0968 - val_loss: 75.7989 - val_mae: 6.0117
Epoch 59/100
11/11 [==============================] - 0s 6ms/step - loss: 37.3029 - mae:
4.4993 - val_loss: 69.9205 - val_mae: 5.8598
Epoch 60/100
11/11 [==============================] - 0s 5ms/step - loss: 36.1463 - mae:
4.3774 - val_loss: 93.1340 - val_mae: 6.6789
Epoch 61/100
11/11 [==============================] - 0s 3ms/step - loss: 35.3942 - mae:
4.1797 - val_loss: 69.6372 - val_mae: 6.9278
Epoch 62/100
11/11 [==============================] - 0s 3ms/step - loss: 44.9558 - mae:
5.2795 - val_loss: 74.5363 - val_mae: 5.8979
Epoch 63/100
11/11 [==============================] - 0s 4ms/step - loss: 40.2197 - mae:
4.6395 - val_loss: 67.6870 - val_mae: 5.7321
Epoch 64/100
11/11 [==============================] - 0s 4ms/step - loss: 35.6249 - mae:
4.4874 - val_loss: 63.1919 - val_mae: 6.0185
Epoch 65/100
11/11 [==============================] - 0s 5ms/step - loss: 33.5510 - mae:
4.1380 - val_loss: 75.4116 - val_mae: 5.7737
Epoch 66/100
11/11 [==============================] - 0s 4ms/step - loss: 33.3171 - mae:
4.3020 - val_loss: 65.2549 - val_mae: 5.6731
Epoch 67/100
11/11 [==============================] - 0s 4ms/step - loss: 30.4981 - mae:
3.9988 - val_loss: 65.9374 - val_mae: 6.0587
Epoch 68/100
11/11 [==============================] - 0s 4ms/step - loss: 31.4584 - mae:
4.0822 - val_loss: 70.9584 - val_mae: 5.9610
Epoch 69/100
11/11 [==============================] - 0s 4ms/step - loss: 32.0598 - mae:
4.0980 - val_loss: 65.5629 - val_mae: 6.0606
Epoch 70/100
11/11 [==============================] - 0s 4ms/step - loss: 30.0251 - mae:
3.9024 - val_loss: 66.6761 - val_mae: 5.7830
Epoch 71/100
11/11 [==============================] - 0s 4ms/step - loss: 30.8111 - mae:
4.1027 - val_loss: 73.0866 - val_mae: 5.8113
Epoch 72/100
11/11 [==============================] - 0s 3ms/step - loss: 32.4922 - mae:
4.1465 - val_loss: 65.9663 - val_mae: 5.8200
Epoch 73/100
11/11 [==============================] - 0s 6ms/step - loss: 29.3212 - mae:
4.0037 - val_loss: 68.6681 - val_mae: 5.7642
```

```
Epoch 74/100
11/11 [==============================] - 0s 4ms/step - loss: 28.9626 - mae:
3.9230 - val_loss: 70.9759 - val_mae: 5.7254
Epoch 75/100
11/11 [==============================] - 0s 4ms/step - loss: 27.2850 - mae:
3.7621 - val_loss: 65.1627 - val_mae: 5.7805
Epoch 76/100
11/11 [==============================] - 0s 4ms/step - loss: 29.2583 - mae:
3.9348 - val_loss: 63.4590 - val_mae: 5.9799
Epoch 77/100
11/11 [==============================] - 0s 6ms/step - loss: 31.6208 - mae:
4.1665 - val_loss: 71.6206 - val_mae: 5.7315
Epoch 78/100
11/11 [==============================] - 0s 5ms/step - loss: 29.2416 - mae:
3.9052 - val_loss: 63.1969 - val_mae: 5.9108
Epoch 79/100
11/11 [==============================] - 0s 4ms/step - loss: 28.9797 - mae:
3.9858 - val_loss: 79.6432 - val_mae: 6.1192
Epoch 80/100
11/11 [==============================] - 0s 4ms/step - loss: 30.1288 - mae:
4.0720 - val_loss: 64.5131 - val_mae: 5.7942
Epoch 81/100
11/11 [==============================] - 0s 3ms/step - loss: 27.6364 - mae:
3.8092 - val_loss: 65.6751 - val_mae: 5.5210
Epoch 82/100
11/11 [==============================] - 0s 4ms/step - loss: 28.2396 - mae:
3.9231 - val_loss: 91.4888 - val_mae: 6.3524
Epoch 83/100
11/11 [==============================] - 0s 4ms/step - loss: 40.4750 - mae:
4.8559 - val_loss: 60.2526 - val_mae: 5.8334
Epoch 84/100
11/11 [==============================] - 0s 4ms/step - loss: 37.2588 - mae:
4.8038 - val_loss: 68.2171 - val_mae: 5.4533
Epoch 85/100
11/11 [==============================] - 0s 4ms/step - loss: 30.8255 - mae:
3.9874 - val_loss: 67.0612 - val_mae: 5.5345
Epoch 86/100
11/11 [==============================] - 0s 4ms/step - loss: 27.5965 - mae:
3.8464 - val_loss: 63.5904 - val_mae: 5.8538
Epoch 87/100
11/11 [==============================] - 0s 4ms/step - loss: 27.9007 - mae:
3.9001 - val_loss: 79.7395 - val_mae: 5.8449
Epoch 88/100
11/11 [==============================] - 0s 3ms/step - loss: 29.3778 - mae:
4.0870 - val_loss: 64.1248 - val_mae: 6.0386
Epoch 89/100
11/11 [==============================] - 0s 4ms/step - loss: 28.6313 - mae:
3.9004 - val_loss: 64.7652 - val_mae: 5.6717
Epoch 90/100
11/11 [==============================] - 0s 4ms/step - loss: 26.2589 - mae:
3.7857 - val_loss: 73.6284 - val_mae: 5.6817
Epoch 91/100
11/11 [==============================] - 0s 4ms/step - loss: 25.5219 - mae:
3.6840 - val_loss: 61.2618 - val_mae: 6.2671
Epoch 92/100
11/11 [==============================] - 0s 4ms/step - loss: 37.5101 - mae:
4.7214 - val_loss: 70.1923 - val_mae: 5.4056
```

```
Epoch 93/100
11/11 [==============================] - 0s 4ms/step - loss: 30.1798 - mae:
4.1516 - val_loss: 59.7963 - val_mae: 6.0707
Epoch 94/100
11/11 [==============================] - 0s 4ms/step - loss: 29.6977 - mae:
4.0330 - val_loss: 57.3263 - val_mae: 5.6396
Epoch 95/100
11/11 [==============================] - 0s 4ms/step - loss: 27.2931 - mae:
3.9198 - val_loss: 77.2184 - val_mae: 5.6756
Epoch 96/100
11/11 [==============================] - 0s 4ms/step - loss: 30.7540 - mae:
4.0007 - val_loss: 63.6341 - val_mae: 6.7858
Epoch 97/100
11/11 [==============================] - 0s 5ms/step - loss: 30.7228 - mae:
4.1860 - val_loss: 69.4769 - val_mae: 5.6059
Epoch 98/100
11/11 [==============================] - 0s 4ms/step - loss: 26.2480 - mae:
3.8078 - val_loss: 65.4580 - val_mae: 5.5055
Epoch 99/100
11/11 [==============================] - 0s 4ms/step - loss: 25.7807 - mae:
3.6395 - val_loss: 67.9779 - val_mae: 5.6516
Epoch 100/100
11/11 [==============================] - 0s 4ms/step - loss: 25.8939 - mae:
3.7184 - val_loss: 63.8970 - val_mae: 5.7290
```

In [ ]:

```
mse_nn, mae_nn = model.evaluate(X_test, y_test)

print('Mean squared error on test data: ', mse_nn)
print('Mean absolute error on test data: ', mae_nn)
```

```
5/5 [==============================] - 0s 2ms/step - loss: 25.9594 - mae: 3
.8497
Mean squared error on test data:  25.959447860717773
Mean absolute error on test data:  3.8496510982513428
```

ASSIGNMENT NO.2

```
from tensorflow.keras.datasets import imdb
```

In [15]:

```
(train_data, train_label), (test_data, test_label) =
imdb.load_data(num_words = 10000)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-da
tasets/imdb.npz
17464789/17464789 [==============================] - 0s 0us/step
```

In [24]:

```
import numpy as np

def vectorize_sequences(sequences, dimensions = 10000):
  results = np.zeros((len(sequences), dimensions))
  for i,sequences in enumerate(sequences):
    results[i, sequences] = 1
  return results

x_train = vectorize_sequences(train_data)
y_train = vectorize_sequences(test_data)
```

In [25]:

```
y_train = np.asarray(train_label).astype('float32')
y_test = np.asarray(test_label).astype('float32')
```

In [31]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

In [19]:

```
model = Sequential()
model.add(Dense(16, input_shape=(10000, ), activation = "relu"))
model.add(Dense(16, activation = "relu"))
model.add(Dense(1, activation = "sigmoid"))
```

In [32]:

```
model.compile(optimizer='adam', loss = 'mse', metrics = ['accuracy'])
```

In [22]:

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 16)                160016

 dense_1 (Dense)             (None, 16)                272
```

```
 dense_2 (Dense)             (None, 1)                 17

=================================================================
Total params: 160,305
Trainable params: 160,305
Non-trainable params: 0
_________________________________________________________________
```

In [33]:

```
history = model.fit(x_train, y_train, validation_split = 0.3, epochs = 20,
verbose = 1, batch_size = 512)
```

```
Epoch 1/20
35/35 [==============================] - 2s 39ms/step - loss: 0.0041 - accu
racy: 0.9960 - val_loss: 0.1162 - val_accuracy: 0.8652
Epoch 2/20
35/35 [==============================] - 1s 35ms/step - loss: 0.0042 - accu
racy: 0.9959 - val_loss: 0.1171 - val_accuracy: 0.8648
Epoch 3/20
35/35 [==============================] - 1s 32ms/step - loss: 0.0043 - accu
racy: 0.9958 - val_loss: 0.1192 - val_accuracy: 0.8636
Epoch 4/20
35/35 [==============================] - 1s 31ms/step - loss: 0.0042 - accu
racy: 0.9959 - val_loss: 0.1210 - val_accuracy: 0.8619
Epoch 5/20
35/35 [==============================] - 1s 31ms/step - loss: 0.0041 - accu
racy: 0.9960 - val_loss: 0.1208 - val_accuracy: 0.8628
Epoch 6/20
35/35 [==============================] - 1s 42ms/step - loss: 0.0041 - accu
racy: 0.9959 - val_loss: 0.1210 - val_accuracy: 0.8619
Epoch 7/20
35/35 [==============================] - 2s 47ms/step - loss: 0.0040 - accu
racy: 0.9961 - val_loss: 0.1213 - val_accuracy: 0.8621
Epoch 8/20
35/35 [==============================] - 1s 31ms/step - loss: 0.0041 - accu
racy: 0.9960 - val_loss: 0.1214 - val_accuracy: 0.8620
Epoch 9/20
35/35 [==============================] - 1s 32ms/step - loss: 0.0041 - accu
racy: 0.9961 - val_loss: 0.1204 - val_accuracy: 0.8655
Epoch 10/20
35/35 [==============================] - 1s 32ms/step - loss: 0.0041 - accu
racy: 0.9961 - val_loss: 0.1215 - val_accuracy: 0.8633
Epoch 11/20
35/35 [==============================] - 1s 37ms/step - loss: 0.0040 - accu
racy: 0.9961 - val_loss: 0.1224 - val_accuracy: 0.8623
Epoch 12/20
35/35 [==============================] - 1s 32ms/step - loss: 0.0039 - accu
racy: 0.9961 - val_loss: 0.1223 - val_accuracy: 0.8617
Epoch 13/20
35/35 [==============================] - 1s 35ms/step - loss: 0.0039 - accu
racy: 0.9961 - val_loss: 0.1223 - val_accuracy: 0.8613
Epoch 14/20
35/35 [==============================] - 1s 36ms/step - loss: 0.0039 - accu
racy: 0.9961 - val_loss: 0.1223 - val_accuracy: 0.8616
Epoch 15/20
35/35 [==============================] - 1s 31ms/step - loss: 0.0039 - accu
racy: 0.9961 - val_loss: 0.1223 - val_accuracy: 0.8613
```

```
Epoch 16/20
35/35 [==============================] - 1s 38ms/step - loss: 0.0039 - accu
racy: 0.9961 - val_loss: 0.1224 - val_accuracy: 0.8611
Epoch 17/20
35/35 [==============================] - 2s 48ms/step - loss: 0.0039 - accu
racy: 0.9961 - val_loss: 0.1224 - val_accuracy: 0.8607
Epoch 18/20
35/35 [==============================] - 1s 37ms/step - loss: 0.0039 - accu
racy: 0.9961 - val_loss: 0.1224 - val_accuracy: 0.8608
Epoch 19/20
35/35 [==============================] - 1s 35ms/step - loss: 0.0039 - accu
racy: 0.9961 - val_loss: 0.1224 - val_accuracy: 0.8605
Epoch 20/20
35/35 [==============================] - 2s 60ms/step - loss: 0.0039 - accu
racy: 0.9961 - val_loss: 0.1224 - val_accuracy: 0.8609
```

ASSIGNMENT NO.3

```
from tensorflow.keras.datasets import fashion_mnist
```

In [38]:

```
(train_x, train_y), (test_x, test_y) = fashion_mnist.load_data()
```

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

In [ ]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, MaxPooling2D, Conv2D
```

In [50]:

```
model = Sequential()
```

In [51]:

```
model.add(Conv2D(filters=64,kernel_size=(3,3),activation='relu',input_shape
=(28, 28, 1)))

# Adding maxpooling layer to get max value within a matrix
model.add(MaxPooling2D(pool_size=(2,2)))

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

In [52]:

```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten_1 (Flatten)         (None, 784)               0

 dense_8 (Dense)             (None, 128)               100480

 dense_9 (Dense)             (None, 10)                1290

=================================================================
Total params: 101,770
```

```
Trainable params: 101,770
Non-trainable params: 0
_________________________________________________________________
```

In [56]:

```
model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy',
metrics = ['accuracy'])
```

In [85]:

```
model.fit(train_x.astype(np.float32), train_y.astype(np.float32), epochs =
5, validation_split = 0.2)
```

```
Epoch 1/5
1500/1500 [==============================] - 7s 5ms/step - loss: 0.4881 - a
ccuracy: 0.8302 - val_loss: 0.5287 - val_accuracy: 0.8273
Epoch 2/5
1500/1500 [==============================] - 6s 4ms/step - loss: 0.4893 - a
ccuracy: 0.8298 - val_loss: 0.5376 - val_accuracy: 0.8243
Epoch 3/5
1500/1500 [==============================] - 7s 5ms/step - loss: 0.4774 - a
ccuracy: 0.8342 - val_loss: 0.5451 - val_accuracy: 0.8282
Epoch 4/5
1500/1500 [==============================] - 6s 4ms/step - loss: 0.4751 - a
ccuracy: 0.8361 - val_loss: 0.5717 - val_accuracy: 0.8299
Epoch 5/5
1500/1500 [==============================] - 7s 5ms/step - loss: 0.4753 - a
ccuracy: 0.8363 - val_loss: 0.5278 - val_accuracy: 0.8255
```

Out[85]:

```
<keras.callbacks.History at 0x7fbcee0aceb0>
```

In [86]:

```
loss, acc = model.evaluate(test_x, test_y)
```

```
313/313 [==============================] - 1s 2ms/step - loss: 0.5724 - acc
uracy: 0.8171
```

In [87]:

```
labels = ['t_shirt', 'trouser', 'pullover', 'dress', 'coat', 'sandal',
'shirt', 'sneaker',   'bag', 'ankle_boots']
```

In [88]:

```
predictions = model.predict(test_x[:1])
```

```
1/1 [==============================] - 0s 73ms/step
```

In [90]:

```
import numpy as np
```

In [91]:

```
label = labels[np.argmax(predictions)]
```

In [93]:

```
import matplotlib.pyplot as plt
print(label)
plt.imshow(test_x[:1][0])
```

```
plt.show
```

```
ankle_boots
```

Out[93]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```