1.) Data Preprocessing

```
# Data Preprocessing
from keras.datasets import mnist

# Importing the dataset
(X_train,y_train),(X_test,y_test) = mnist.load_data()

#flatten the images
X_train=X_train.reshape((X_train.shape[0],784))
X_test=X_test.reshape((X_test.shape[0],784))
y_train=y_train.reshape((y_train.shape[0],1)) #fixing shape
y_test=y_test.reshape((y_test.shape[0],1)) #fixing shape

#Encoding of categorical data to one-hot format
from sklearn.preprocessing import OneHotEncoder
onehotencoder=OneHotEncoder(sparse=False)
y_train=onehotencoder.fit_transform(y_train)
y_test=onehotencoder.transform(y_test)

# Splitting the Training into the Training set and Validation set
from sklearn.model_selection import train_test_split
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train,
test_size = 0.175, random_state = 0)

#Feature Scaling
X_train = X_train / 255
X_val = X_val / 255
X_test = X_test / 255
```

```
Using TensorFlow backend.
```

2.) Build and Train ANN

```
#ANN

#import libraries
import keras
from keras.models import Sequential
from keras.layers import Dense,Activation,Dropout,BatchNormalization

#Build ANN model
model=Sequential()
model.add(Dense(256,kernel_initializer='he_normal',input_shape=(784,))
)
```

```
model.add(Activation('relu'))
model.add(Dropout(0.4))

model.add(Dense(256,kernel_initializer='he_normal'))
model.add(Activation('relu'))
model.add(Dropout(0.4))

model.add(Dense(10,activation='softmax',kernel_initializer='glorot_nor
mal'))

#Compiling model
model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.001),los
s='categorical_crossentropy',metrics=['accuracy'])

#Training model
history=model.fit(X_train,y_train,epochs=150,batch_size=512,validation
_data=(X_val,y_val),verbose=2,use_multiprocessing=True)

Train on 49500 samples, validate on 10500 samples
Epoch 1/150
 - 1s - loss: 0.7091 - accuracy: 0.7776 - val_loss: 0.2229 -
val_accuracy: 0.9326
Epoch 2/150
 - 1s - loss: 0.2741 - accuracy: 0.9191 - val_loss: 0.1526 -
val_accuracy: 0.9538
Epoch 3/150
 - 1s - loss: 0.2042 - accuracy: 0.9410 - val_loss: 0.1261 -
val_accuracy: 0.9621
Epoch 4/150
 - 1s - loss: 0.1684 - accuracy: 0.9493 - val_loss: 0.1071 -
val_accuracy: 0.9675
Epoch 5/150
 - 1s - loss: 0.1452 - accuracy: 0.9561 - val_loss: 0.0985 -
val_accuracy: 0.9705
Epoch 6/150
 - 1s - loss: 0.1256 - accuracy: 0.9631 - val_loss: 0.0925 -
val_accuracy: 0.9719
Epoch 7/150
 - 1s - loss: 0.1146 - accuracy: 0.9656 - val_loss: 0.0858 -
val_accuracy: 0.9743
Epoch 8/150
 - 1s - loss: 0.1031 - accuracy: 0.9687 - val_loss: 0.0809 -
val_accuracy: 0.9757
Epoch 9/150
 - 1s - loss: 0.0941 - accuracy: 0.9713 - val_loss: 0.0771 -
val_accuracy: 0.9763
Epoch 10/150
 - 1s - loss: 0.0869 - accuracy: 0.9732 - val_loss: 0.0723 -
val_accuracy: 0.9785
Epoch 11/150
```

```
 - 1s - loss: 0.0824 - accuracy: 0.9740 - val_loss: 0.0695 -
val_accuracy: 0.9805
Epoch 12/150
 - 1s - loss: 0.0748 - accuracy: 0.9766 - val_loss: 0.0692 -
val_accuracy: 0.9790
Epoch 13/150
 - 1s - loss: 0.0713 - accuracy: 0.9769 - val_loss: 0.0712 -
val_accuracy: 0.9787
Epoch 14/150
 - 1s - loss: 0.0668 - accuracy: 0.9792 - val_loss: 0.0652 -
val_accuracy: 0.9813
Epoch 15/150
 - 1s - loss: 0.0577 - accuracy: 0.9813 - val_loss: 0.0694 -
val_accuracy: 0.9791
Epoch 16/150
 - 1s - loss: 0.0573 - accuracy: 0.9814 - val_loss: 0.0649 -
val_accuracy: 0.9789
Epoch 17/150
 - 1s - loss: 0.0554 - accuracy: 0.9823 - val_loss: 0.0633 -
val_accuracy: 0.9803
Epoch 18/150
 - 1s - loss: 0.0525 - accuracy: 0.9828 - val_loss: 0.0646 -
val_accuracy: 0.9804
Epoch 19/150
 - 1s - loss: 0.0514 - accuracy: 0.9838 - val_loss: 0.0641 -
val_accuracy: 0.9810
Epoch 20/150
 - 1s - loss: 0.0472 - accuracy: 0.9850 - val_loss: 0.0657 -
val_accuracy: 0.9805
Epoch 21/150
 - 1s - loss: 0.0451 - accuracy: 0.9851 - val_loss: 0.0651 -
val_accuracy: 0.9806
Epoch 22/150
 - 1s - loss: 0.0454 - accuracy: 0.9848 - val_loss: 0.0648 -
val_accuracy: 0.9810
Epoch 23/150
 - 1s - loss: 0.0404 - accuracy: 0.9865 - val_loss: 0.0656 -
val_accuracy: 0.9802
Epoch 24/150
 - 1s - loss: 0.0386 - accuracy: 0.9873 - val_loss: 0.0681 -
val_accuracy: 0.9810
Epoch 25/150
 - 1s - loss: 0.0377 - accuracy: 0.9878 - val_loss: 0.0632 -
val_accuracy: 0.9818
Epoch 26/150
 - 1s - loss: 0.0372 - accuracy: 0.9879 - val_loss: 0.0637 -
val_accuracy: 0.9820
Epoch 27/150
 - 1s - loss: 0.0354 - accuracy: 0.9886 - val_loss: 0.0683 -
```

```
val_accuracy: 0.9814
Epoch 28/150
 - 1s - loss: 0.0346 - accuracy: 0.9886 - val_loss: 0.0655 -
val_accuracy: 0.9818
Epoch 29/150
 - 1s - loss: 0.0328 - accuracy: 0.9888 - val_loss: 0.0658 -
val_accuracy: 0.9824
Epoch 30/150
 - 1s - loss: 0.0335 - accuracy: 0.9889 - val_loss: 0.0666 -
val_accuracy: 0.9819
Epoch 31/150
 - 1s - loss: 0.0295 - accuracy: 0.9898 - val_loss: 0.0669 -
val_accuracy: 0.9818
Epoch 32/150
 - 1s - loss: 0.0307 - accuracy: 0.9900 - val_loss: 0.0672 -
val_accuracy: 0.9817
Epoch 33/150
 - 1s - loss: 0.0309 - accuracy: 0.9895 - val_loss: 0.0645 -
val_accuracy: 0.9836
Epoch 34/150
 - 1s - loss: 0.0304 - accuracy: 0.9901 - val_loss: 0.0664 -
val_accuracy: 0.9833
Epoch 35/150
 - 1s - loss: 0.0283 - accuracy: 0.9908 - val_loss: 0.0697 -
val_accuracy: 0.9813
Epoch 36/150
 - 1s - loss: 0.0295 - accuracy: 0.9899 - val_loss: 0.0673 -
val_accuracy: 0.9821
Epoch 37/150
 - 1s - loss: 0.0293 - accuracy: 0.9904 - val_loss: 0.0687 -
val_accuracy: 0.9818
Epoch 38/150
 - 1s - loss: 0.0269 - accuracy: 0.9910 - val_loss: 0.0691 -
val_accuracy: 0.9816
Epoch 39/150
 - 1s - loss: 0.0269 - accuracy: 0.9915 - val_loss: 0.0651 -
val_accuracy: 0.9825
Epoch 40/150
 - 1s - loss: 0.0246 - accuracy: 0.9919 - val_loss: 0.0647 -
val_accuracy: 0.9833
Epoch 41/150
 - 1s - loss: 0.0246 - accuracy: 0.9919 - val_loss: 0.0701 -
val_accuracy: 0.9829
Epoch 42/150
 - 1s - loss: 0.0260 - accuracy: 0.9913 - val_loss: 0.0737 -
val_accuracy: 0.9812
Epoch 43/150
 - 1s - loss: 0.0250 - accuracy: 0.9912 - val_loss: 0.0677 -
val_accuracy: 0.9833
```

```
Epoch 44/150
 - 1s - loss: 0.0245 - accuracy: 0.9912 - val_loss: 0.0678 -
val_accuracy: 0.9825
Epoch 45/150
 - 1s - loss: 0.0236 - accuracy: 0.9922 - val_loss: 0.0678 -
val_accuracy: 0.9827
Epoch 46/150
 - 1s - loss: 0.0233 - accuracy: 0.9921 - val_loss: 0.0701 -
val_accuracy: 0.9803
Epoch 47/150
 - 1s - loss: 0.0215 - accuracy: 0.9929 - val_loss: 0.0666 -
val_accuracy: 0.9830
Epoch 48/150
 - 1s - loss: 0.0215 - accuracy: 0.9929 - val_loss: 0.0671 -
val_accuracy: 0.9842
Epoch 49/150
 - 1s - loss: 0.0201 - accuracy: 0.9931 - val_loss: 0.0676 -
val_accuracy: 0.9826
Epoch 50/150
 - 1s - loss: 0.0205 - accuracy: 0.9929 - val_loss: 0.0701 -
val_accuracy: 0.9835
Epoch 51/150
 - 1s - loss: 0.0210 - accuracy: 0.9927 - val_loss: 0.0733 -
val_accuracy: 0.9820
Epoch 52/150
 - 1s - loss: 0.0226 - accuracy: 0.9925 - val_loss: 0.0655 -
val_accuracy: 0.9830
Epoch 53/150
 - 1s - loss: 0.0208 - accuracy: 0.9932 - val_loss: 0.0765 -
val_accuracy: 0.9812
Epoch 54/150
 - 1s - loss: 0.0202 - accuracy: 0.9931 - val_loss: 0.0680 -
val_accuracy: 0.9835
Epoch 55/150
 - 1s - loss: 0.0211 - accuracy: 0.9925 - val_loss: 0.0688 -
val_accuracy: 0.9832
Epoch 56/150
 - 1s - loss: 0.0218 - accuracy: 0.9927 - val_loss: 0.0713 -
val_accuracy: 0.9833
Epoch 57/150
 - 1s - loss: 0.0218 - accuracy: 0.9924 - val_loss: 0.0675 -
val_accuracy: 0.9833
Epoch 58/150
 - 1s - loss: 0.0193 - accuracy: 0.9935 - val_loss: 0.0724 -
val_accuracy: 0.9825
Epoch 59/150
 - 1s - loss: 0.0188 - accuracy: 0.9933 - val_loss: 0.0726 -
val_accuracy: 0.9833
Epoch 60/150
```

```
 - 1s - loss: 0.0181 - accuracy: 0.9938 - val_loss: 0.0724 -
val_accuracy: 0.9841
Epoch 61/150
 - 1s - loss: 0.0206 - accuracy: 0.9932 - val_loss: 0.0775 -
val_accuracy: 0.9827
Epoch 62/150
 - 1s - loss: 0.0195 - accuracy: 0.9929 - val_loss: 0.0758 -
val_accuracy: 0.9820
Epoch 63/150
 - 1s - loss: 0.0176 - accuracy: 0.9941 - val_loss: 0.0732 -
val_accuracy: 0.9841
Epoch 64/150
 - 1s - loss: 0.0189 - accuracy: 0.9937 - val_loss: 0.0738 -
val_accuracy: 0.9824
Epoch 65/150
 - 1s - loss: 0.0175 - accuracy: 0.9942 - val_loss: 0.0705 -
val_accuracy: 0.9824
Epoch 66/150
 - 1s - loss: 0.0169 - accuracy: 0.9944 - val_loss: 0.0695 -
val_accuracy: 0.9831
Epoch 67/150
 - 1s - loss: 0.0163 - accuracy: 0.9946 - val_loss: 0.0777 -
val_accuracy: 0.9827
Epoch 68/150
 - 1s - loss: 0.0185 - accuracy: 0.9940 - val_loss: 0.0735 -
val_accuracy: 0.9830
Epoch 69/150
 - 1s - loss: 0.0176 - accuracy: 0.9940 - val_loss: 0.0750 -
val_accuracy: 0.9826
Epoch 70/150
 - 1s - loss: 0.0193 - accuracy: 0.9937 - val_loss: 0.0734 -
val_accuracy: 0.9830
Epoch 71/150
 - 1s - loss: 0.0152 - accuracy: 0.9947 - val_loss: 0.0728 -
val_accuracy: 0.9828
Epoch 72/150
 - 1s - loss: 0.0160 - accuracy: 0.9946 - val_loss: 0.0761 -
val_accuracy: 0.9835
Epoch 73/150
 - 1s - loss: 0.0153 - accuracy: 0.9948 - val_loss: 0.0779 -
val_accuracy: 0.9832
Epoch 74/150
 - 1s - loss: 0.0170 - accuracy: 0.9941 - val_loss: 0.0793 -
val_accuracy: 0.9833
Epoch 75/150
 - 1s - loss: 0.0170 - accuracy: 0.9946 - val_loss: 0.0787 -
val_accuracy: 0.9834
Epoch 76/150
 - 1s - loss: 0.0166 - accuracy: 0.9945 - val_loss: 0.0775 -
```

```
val_accuracy: 0.9830
Epoch 77/150
 - 1s - loss: 0.0161 - accuracy: 0.9945 - val_loss: 0.0712 -
val_accuracy: 0.9842
Epoch 78/150
 - 1s - loss: 0.0149 - accuracy: 0.9951 - val_loss: 0.0744 -
val_accuracy: 0.9841
Epoch 79/150
 - 1s - loss: 0.0164 - accuracy: 0.9943 - val_loss: 0.0772 -
val_accuracy: 0.9839
Epoch 80/150
 - 1s - loss: 0.0171 - accuracy: 0.9942 - val_loss: 0.0778 -
val_accuracy: 0.9824
Epoch 81/150
 - 1s - loss: 0.0163 - accuracy: 0.9945 - val_loss: 0.0703 -
val_accuracy: 0.9828
Epoch 82/150
 - 1s - loss: 0.0135 - accuracy: 0.9956 - val_loss: 0.0750 -
val_accuracy: 0.9834
Epoch 83/150
 - 1s - loss: 0.0159 - accuracy: 0.9948 - val_loss: 0.0740 -
val_accuracy: 0.9833
Epoch 84/150
 - 1s - loss: 0.0144 - accuracy: 0.9951 - val_loss: 0.0704 -
val_accuracy: 0.9845
Epoch 85/150
 - 1s - loss: 0.0151 - accuracy: 0.9948 - val_loss: 0.0670 -
val_accuracy: 0.9834
Epoch 86/150
 - 1s - loss: 0.0141 - accuracy: 0.9951 - val_loss: 0.0743 -
val_accuracy: 0.9836
Epoch 87/150
 - 1s - loss: 0.0135 - accuracy: 0.9953 - val_loss: 0.0742 -
val_accuracy: 0.9839
Epoch 88/150
 - 1s - loss: 0.0142 - accuracy: 0.9954 - val_loss: 0.0710 -
val_accuracy: 0.9845
Epoch 89/150
 - 1s - loss: 0.0145 - accuracy: 0.9952 - val_loss: 0.0745 -
val_accuracy: 0.9840
Epoch 90/150
 - 1s - loss: 0.0153 - accuracy: 0.9949 - val_loss: 0.0811 -
val_accuracy: 0.9827
Epoch 91/150
 - 1s - loss: 0.0148 - accuracy: 0.9951 - val_loss: 0.0768 -
val_accuracy: 0.9833
Epoch 92/150
 - 1s - loss: 0.0153 - accuracy: 0.9948 - val_loss: 0.0776 -
val_accuracy: 0.9829
```

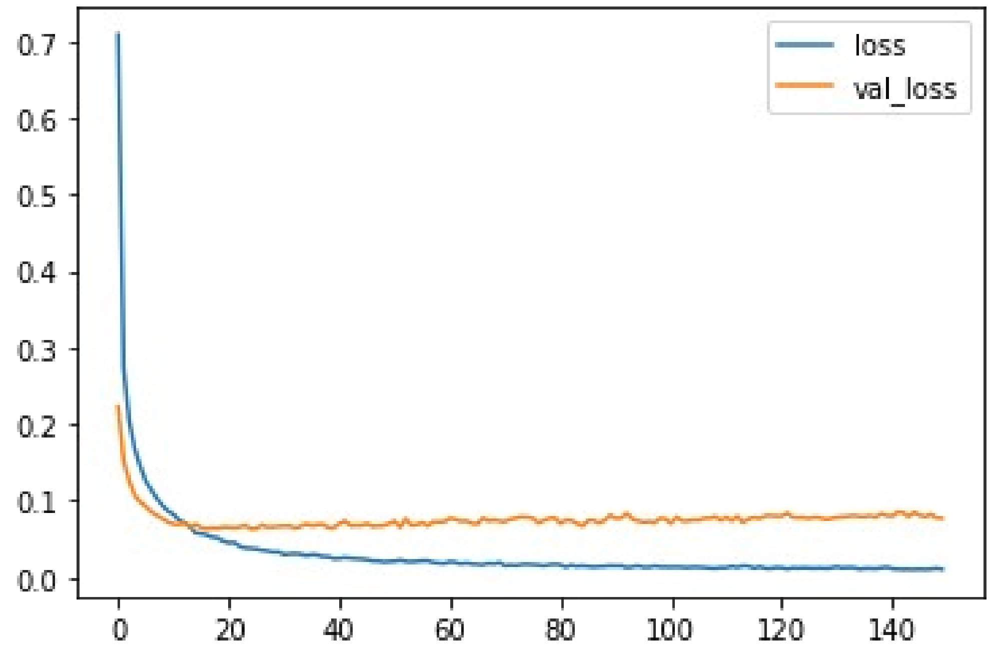
```
Epoch 93/150
 - 1s - loss: 0.0126 - accuracy: 0.9957 - val_loss: 0.0835 -
val_accuracy: 0.9830
Epoch 94/150
 - 1s - loss: 0.0138 - accuracy: 0.9956 - val_loss: 0.0764 -
val_accuracy: 0.9834
Epoch 95/150
 - 1s - loss: 0.0150 - accuracy: 0.9953 - val_loss: 0.0737 -
val_accuracy: 0.9834
Epoch 96/150
 - 1s - loss: 0.0133 - accuracy: 0.9953 - val_loss: 0.0722 -
val_accuracy: 0.9842
Epoch 97/150
 - 1s - loss: 0.0131 - accuracy: 0.9958 - val_loss: 0.0729 -
val_accuracy: 0.9835
Epoch 98/150
 - 1s - loss: 0.0149 - accuracy: 0.9948 - val_loss: 0.0714 -
val_accuracy: 0.9846
Epoch 99/150
 - 1s - loss: 0.0136 - accuracy: 0.9956 - val_loss: 0.0772 -
val_accuracy: 0.9837
Epoch 100/150
 - 1s - loss: 0.0136 - accuracy: 0.9955 - val_loss: 0.0763 -
val_accuracy: 0.9845
Epoch 101/150
 - 1s - loss: 0.0136 - accuracy: 0.9953 - val_loss: 0.0706 -
val_accuracy: 0.9841
Epoch 102/150
 - 1s - loss: 0.0131 - accuracy: 0.9957 - val_loss: 0.0788 -
val_accuracy: 0.9841
Epoch 103/150
 - 1s - loss: 0.0134 - accuracy: 0.9957 - val_loss: 0.0738 -
val_accuracy: 0.9849
Epoch 104/150
 - 1s - loss: 0.0126 - accuracy: 0.9955 - val_loss: 0.0737 -
val_accuracy: 0.9844
Epoch 105/150
 - 1s - loss: 0.0136 - accuracy: 0.9955 - val_loss: 0.0749 -
val_accuracy: 0.9832
Epoch 106/150
 - 1s - loss: 0.0132 - accuracy: 0.9956 - val_loss: 0.0744 -
val_accuracy: 0.9839
Epoch 107/150
 - 1s - loss: 0.0129 - accuracy: 0.9956 - val_loss: 0.0772 -
val_accuracy: 0.9842
Epoch 108/150
 - 1s - loss: 0.0119 - accuracy: 0.9960 - val_loss: 0.0772 -
val_accuracy: 0.9834
Epoch 109/150
```

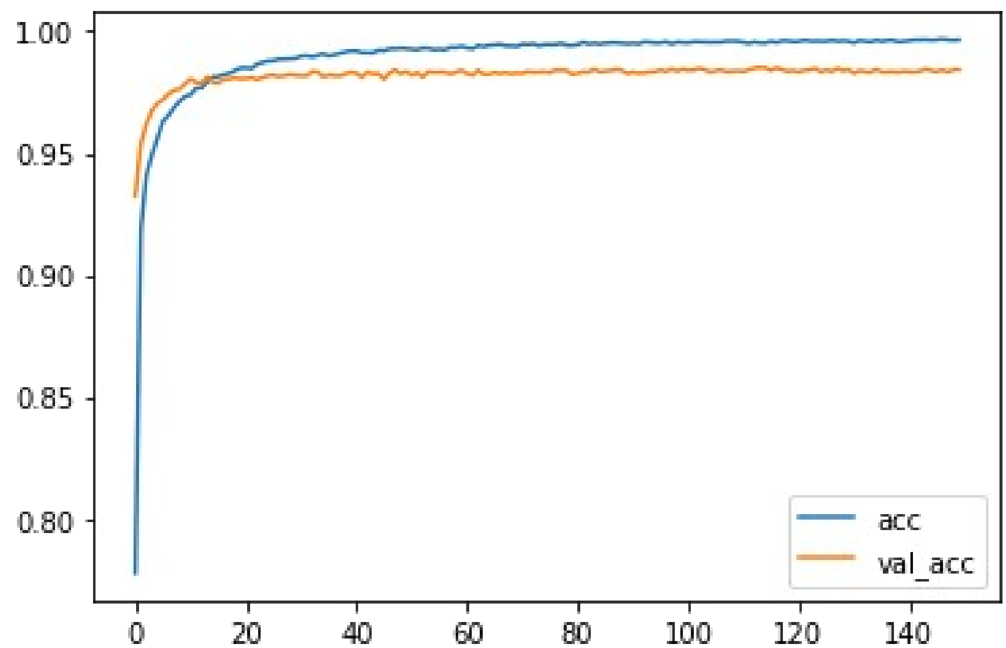
```
 - 1s - loss: 0.0117 - accuracy: 0.9960 - val_loss: 0.0797 -
val_accuracy: 0.9836
Epoch 110/150
 - 1s - loss: 0.0125 - accuracy: 0.9960 - val_loss: 0.0744 -
val_accuracy: 0.9838
Epoch 111/150
 - 1s - loss: 0.0125 - accuracy: 0.9955 - val_loss: 0.0789 -
val_accuracy: 0.9835
Epoch 112/150
 - 1s - loss: 0.0146 - accuracy: 0.9950 - val_loss: 0.0738 -
val_accuracy: 0.9842
Epoch 113/150
 - 1s - loss: 0.0129 - accuracy: 0.9957 - val_loss: 0.0800 -
val_accuracy: 0.9850
Epoch 114/150
 - 1s - loss: 0.0152 - accuracy: 0.9953 - val_loss: 0.0716 -
val_accuracy: 0.9852
Epoch 115/150
 - 1s - loss: 0.0145 - accuracy: 0.9952 - val_loss: 0.0764 -
val_accuracy: 0.9850
Epoch 116/150
 - 1s - loss: 0.0119 - accuracy: 0.9959 - val_loss: 0.0783 -
val_accuracy: 0.9840
Epoch 117/150
 - 1s - loss: 0.0140 - accuracy: 0.9954 - val_loss: 0.0768 -
val_accuracy: 0.9850
Epoch 118/150
 - 1s - loss: 0.0117 - accuracy: 0.9961 - val_loss: 0.0803 -
val_accuracy: 0.9844
Epoch 119/150
 - 1s - loss: 0.0115 - accuracy: 0.9961 - val_loss: 0.0805 -
val_accuracy: 0.9835
Epoch 120/150
 - 1s - loss: 0.0129 - accuracy: 0.9957 - val_loss: 0.0799 -
val_accuracy: 0.9839
Epoch 121/150
 - 1s - loss: 0.0125 - accuracy: 0.9959 - val_loss: 0.0801 -
val_accuracy: 0.9834
Epoch 122/150
 - 1s - loss: 0.0108 - accuracy: 0.9964 - val_loss: 0.0838 -
val_accuracy: 0.9840
Epoch 123/150
 - 1s - loss: 0.0123 - accuracy: 0.9960 - val_loss: 0.0785 -
val_accuracy: 0.9842
Epoch 124/150
 - 1s - loss: 0.0119 - accuracy: 0.9960 - val_loss: 0.0779 -
val_accuracy: 0.9849
Epoch 125/150
 - 1s - loss: 0.0124 - accuracy: 0.9960 - val_loss: 0.0785 -
```

```
val_accuracy: 0.9833
Epoch 126/150
 - 1s - loss: 0.0120 - accuracy: 0.9959 - val_loss: 0.0769 -
val_accuracy: 0.9842
Epoch 127/150
 - 1s - loss: 0.0117 - accuracy: 0.9964 - val_loss: 0.0787 -
val_accuracy: 0.9835
Epoch 128/150
 - 1s - loss: 0.0121 - accuracy: 0.9959 - val_loss: 0.0780 -
val_accuracy: 0.9849
Epoch 129/150
 - 1s - loss: 0.0116 - accuracy: 0.9961 - val_loss: 0.0755 -
val_accuracy: 0.9839
Epoch 130/150
 - 1s - loss: 0.0122 - accuracy: 0.9959 - val_loss: 0.0772 -
val_accuracy: 0.9843
Epoch 131/150
 - 1s - loss: 0.0139 - accuracy: 0.9953 - val_loss: 0.0785 -
val_accuracy: 0.9828
Epoch 132/150
 - 1s - loss: 0.0120 - accuracy: 0.9963 - val_loss: 0.0789 -
val_accuracy: 0.9839
Epoch 133/150
 - 1s - loss: 0.0128 - accuracy: 0.9959 - val_loss: 0.0797 -
val_accuracy: 0.9834
Epoch 134/150
 - 1s - loss: 0.0128 - accuracy: 0.9957 - val_loss: 0.0807 -
val_accuracy: 0.9832
Epoch 135/150
 - 1s - loss: 0.0106 - accuracy: 0.9964 - val_loss: 0.0795 -
val_accuracy: 0.9845
Epoch 136/150
 - 1s - loss: 0.0115 - accuracy: 0.9959 - val_loss: 0.0795 -
val_accuracy: 0.9834
Epoch 137/150
 - 1s - loss: 0.0116 - accuracy: 0.9963 - val_loss: 0.0805 -
val_accuracy: 0.9833
Epoch 138/150
 - 1s - loss: 0.0111 - accuracy: 0.9964 - val_loss: 0.0786 -
val_accuracy: 0.9835
Epoch 139/150
 - 1s - loss: 0.0122 - accuracy: 0.9957 - val_loss: 0.0831 -
val_accuracy: 0.9832
Epoch 140/150
 - 1s - loss: 0.0134 - accuracy: 0.9959 - val_loss: 0.0808 -
val_accuracy: 0.9836
Epoch 141/150
 - 1s - loss: 0.0111 - accuracy: 0.9963 - val_loss: 0.0799 -
val_accuracy: 0.9834
```

```
Epoch 142/150
 - 1s - loss: 0.0107 - accuracy: 0.9965 - val_loss: 0.0855 -
val_accuracy: 0.9833
Epoch 143/150
 - 1s - loss: 0.0101 - accuracy: 0.9965 - val_loss: 0.0845 -
val_accuracy: 0.9837
Epoch 144/150
 - 1s - loss: 0.0104 - accuracy: 0.9966 - val_loss: 0.0794 -
val_accuracy: 0.9847
Epoch 145/150
 - 1s - loss: 0.0101 - accuracy: 0.9964 - val_loss: 0.0852 -
val_accuracy: 0.9838
Epoch 146/150
 - 1s - loss: 0.0105 - accuracy: 0.9965 - val_loss: 0.0810 -
val_accuracy: 0.9835
Epoch 147/150
 - 1s - loss: 0.0099 - accuracy: 0.9969 - val_loss: 0.0793 -
val_accuracy: 0.9842
Epoch 148/150
 - 1s - loss: 0.0106 - accuracy: 0.9965 - val_loss: 0.0823 -
val_accuracy: 0.9833
Epoch 149/150
 - 1s - loss: 0.0118 - accuracy: 0.9964 - val_loss: 0.0777 -
val_accuracy: 0.9844
Epoch 150/150
 - 1s - loss: 0.0103 - accuracy: 0.9965 - val_loss: 0.0771 -
val_accuracy: 0.9843
```

```python
import matplotlib.pyplot as plt
plt.plot(history.history['loss'],label='loss')
plt.plot(history.history['val_loss'],label='val_loss')
plt.legend()
plt.show()
plt.plot(history.history['accuracy'],label='acc')
plt.plot(history.history['val_accuracy'],label='val_acc')
plt.legend()
plt.show()
```






```
train=model.evaluate(X_train,y_train)
print(train)

49500/49500 [==============================] - 4s 86us/step
[0.00023556755931090402, 1.0]

val=model.evaluate(X_val,y_val)
print(val)
```

```
10500/10500 [==============================] - 1s 87us/step
[0.07706455891983031, 0.9842857122421265]
```

```
print(100*(1-train[1]))
print(100*(train[1]-val[1]))#After tuning.....the measure of "variance" wasn't reducing after...~~ 1.5%
```

```
0.0
1.5714287757873535
```

```
model.save('mnist-ann.model')
```

3.) Evaluation on Test set

```
model.evaluate(X_test,y_test)
```

```
10000/10000 [==============================] - 1s 85us/step
```

```
[0.08582737032464338, 0.9842000007629395]
```