In [1]:
```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import time

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input dir
ectory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# Any results you write to the current directory are saved as output.
```

```
/kaggle/input/Kannada-MNIST/Dig-MNIST.csv
/kaggle/input/Kannada-MNIST/train.csv
/kaggle/input/Kannada-MNIST/test.csv
/kaggle/input/Kannada-MNIST/sample_submission.csv
```

In [2]:
```
train_df = pd.read_csv("/kaggle/input/Kannada-MNIST/train.csv")
#test_df = pd.read_csv("/kaggle/input/Kannada-MNIST/Dig-MNIST.csv")
```

In [3]:
```
x_train, y_train = np.asarray(train_df.iloc[:,1:]), np.asarray(train_df.iloc[:,0])
#x_test, y_test = np.asarray(test_df.iloc[:,1:]), np.asarray(test_df.iloc[:,0])
```

In [4]:
```
# Set numeric type to float32 from uint8
x_train = x_train.astype(np.float32)
#x_test = x_test.astype(np.float32)

# Normalize value to [0, 1]
x_train /= 255
#x_test /= 255

# Reshape the dataset into 4D array
x_train = np.asarray(x_train).reshape(x_train.shape[0], 28, 28, 1)
#x_test = np.asarray(x_test).reshape(x_test.shape[0], 28, 28, 1)
```

In [5]:
```
import tensorflow.keras as keras
from keras import layers
from keras.models import Sequential

model = Sequential()

model.add(layers.Conv2D(6, kernel_size=(5, 5), strides=(1, 1), activation='tanh', input_shape=(28,28,1), padd
ing='same'))#1
model.add(layers.AveragePooling2D(pool_size=(2, 2), strides=(2, 2), padding='valid'))#2
model.add(layers.Conv2D(16, kernel_size=(5, 5), strides=(1, 1), activation='tanh', padding='valid'))#3
model.add(layers.AveragePooling2D(pool_size=(2, 2), strides=(2, 2), padding='valid'))#4
model.add(layers.Conv2D(120, kernel_size=(5, 5), strides=(1, 1), activation='tanh', padding='valid'))#5
model.add(layers.Flatten())
model.add(layers.Dense(84, activation='tanh'))#6
model.add(layers.Dense(10, activation='softmax'))

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

```
Using TensorFlow backend.

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_1 (Conv2D)            (None, 28, 28, 6)         156       
_________________________________________________________________
average_pooling2d_1 (Average (None, 14, 14, 6)         0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 10, 10, 16)        2416      
_________________________________________________________________
average_pooling2d_2 (Average (None, 5, 5, 16)          0         
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 1, 1, 120)         48120     
_________________________________________________________________
flatten_1 (Flatten)          (None, 120)               0         
_________________________________________________________________
dense_1 (Dense)              (None, 84)                10164     
_________________________________________________________________
dense_2 (Dense)              (None, 10)                850       
=================================================================
Total params: 61,706
Trainable params: 61,706
Non-trainable params: 0
_________________________________________________________________
```

In [6]:
```
start = time.time()
hist = model.fit(x=x_train,y=y_train, epochs=30, batch_size=16, validation_split=0.2, verbose=1)
end = time.time()
print('Elapsed time: {:.1f} second(s)'.format(end - start))
```

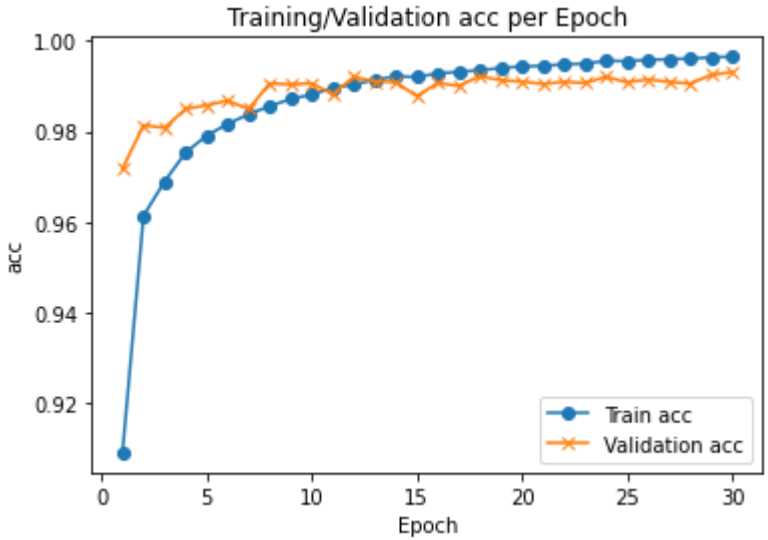
```
Train on 48000 samples, validate on 12000 samples
Epoch 1/30
48000/48000 [==============================] - 26s 550us/step - loss: 0.3857 - accuracy: 0.9091 - val_loss:
0.0992 - val_accuracy: 0.9718
Epoch 2/30
48000/48000 [==============================] - 26s 542us/step - loss: 0.1333 - accuracy: 0.9614 - val_loss:
0.0645 - val_accuracy: 0.9813
Epoch 3/30
48000/48000 [==============================] - 26s 537us/step - loss: 0.1044 - accuracy: 0.9689 - val_loss:
0.0620 - val_accuracy: 0.9808
Epoch 4/30
48000/48000 [==============================] - 25s 524us/step - loss: 0.0853 - accuracy: 0.9753 - val_loss:
0.0522 - val_accuracy: 0.9849
Epoch 5/30
48000/48000 [==============================] - 26s 538us/step - loss: 0.0717 - accuracy: 0.9790 - val_loss:
0.0460 - val_accuracy: 0.9858
Epoch 6/30
48000/48000 [==============================] - 25s 527us/step - loss: 0.0617 - accuracy: 0.9816 - val_loss:
0.0400 - val_accuracy: 0.9868
Epoch 7/30
48000/48000 [==============================] - 26s 541us/step - loss: 0.0542 - accuracy: 0.9838 - val_loss:
0.0446 - val_accuracy: 0.9849
Epoch 8/30
48000/48000 [==============================] - 25s 529us/step - loss: 0.0478 - accuracy: 0.9855 - val_loss:
0.0300 - val_accuracy: 0.9906
Epoch 9/30
48000/48000 [==============================] - 25s 526us/step - loss: 0.0427 - accuracy: 0.9871 - val_loss:
0.0323 - val_accuracy: 0.9903
Epoch 10/30
48000/48000 [==============================] - 26s 534us/step - loss: 0.0391 - accuracy: 0.9881 - val_loss:
0.0299 - val_accuracy: 0.9906
Epoch 11/30
48000/48000 [==============================] - 26s 532us/step - loss: 0.0356 - accuracy: 0.9895 - val_loss:
0.0342 - val_accuracy: 0.9882
Epoch 12/30
48000/48000 [==============================] - 27s 553us/step - loss: 0.0330 - accuracy: 0.9903 - val_loss:
0.0263 - val_accuracy: 0.9920
Epoch 13/30
48000/48000 [==============================] - 26s 542us/step - loss: 0.0307 - accuracy: 0.9914 - val_loss:
0.0275 - val_accuracy: 0.9910
Epoch 14/30
48000/48000 [==============================] - 28s 576us/step - loss: 0.0287 - accuracy: 0.9921 - val_loss:
0.0273 - val_accuracy: 0.9908
```

```
Epoch 15/30
48000/48000 [==============================] - 26s 543us/step - loss: 0.0273 - accuracy: 0.9920 - val_loss:
0.0338 - val_accuracy: 0.9877
Epoch 16/30
48000/48000 [==============================] - 26s 551us/step - loss: 0.0254 - accuracy: 0.9928 - val_loss:
0.0280 - val_accuracy: 0.9907
Epoch 17/30
48000/48000 [==============================] - 25s 530us/step - loss: 0.0240 - accuracy: 0.9931 - val_loss:
0.0285 - val_accuracy: 0.9900
Epoch 18/30
48000/48000 [==============================] - 25s 525us/step - loss: 0.0227 - accuracy: 0.9935 - val_loss:
0.0234 - val_accuracy: 0.9922
Epoch 19/30
48000/48000 [==============================] - 26s 544us/step - loss: 0.0217 - accuracy: 0.9939 - val_loss:
0.0255 - val_accuracy: 0.9912
Epoch 20/30
48000/48000 [==============================] - 26s 536us/step - loss: 0.0208 - accuracy: 0.9943 - val_loss:
0.0264 - val_accuracy: 0.9910
Epoch 21/30
48000/48000 [==============================] - 26s 543us/step - loss: 0.0197 - accuracy: 0.9944 - val_loss:
0.0265 - val_accuracy: 0.9904
Epoch 22/30
48000/48000 [==============================] - 26s 533us/step - loss: 0.0187 - accuracy: 0.9948 - val_loss:
0.0270 - val_accuracy: 0.9909
Epoch 23/30
48000/48000 [==============================] - 26s 534us/step - loss: 0.0178 - accuracy: 0.9949 - val_loss:
0.0280 - val_accuracy: 0.9907
Epoch 24/30
48000/48000 [==============================] - 25s 527us/step - loss: 0.0170 - accuracy: 0.9955 - val_loss:
0.0211 - val_accuracy: 0.9920
Epoch 25/30
48000/48000 [==============================] - 25s 526us/step - loss: 0.0165 - accuracy: 0.9955 - val_loss:
0.0245 - val_accuracy: 0.9908
Epoch 26/30
48000/48000 [==============================] - 26s 539us/step - loss: 0.0155 - accuracy: 0.9957 - val_loss:
0.0231 - val_accuracy: 0.9914
Epoch 27/30
48000/48000 [==============================] - 25s 525us/step - loss: 0.0152 - accuracy: 0.9959 - val_loss:
0.0246 - val_accuracy: 0.9909
Epoch 28/30
48000/48000 [==============================] - 26s 537us/step - loss: 0.0144 - accuracy: 0.9960 - val_loss:
0.0288 - val_accuracy: 0.9905
Epoch 29/30
```

```
48000/48000 [==============================] - 25s 531us/step - loss: 0.0138 - accuracy: 0.9963 - val_loss:
0.0219 - val_accuracy: 0.9924
Epoch 30/30
48000/48000 [==============================] - 26s 532us/step - loss: 0.0132 - accuracy: 0.9965 - val_loss:
0.0201 - val_accuracy: 0.9931
Elapsed time: 773.5 second(s)
```

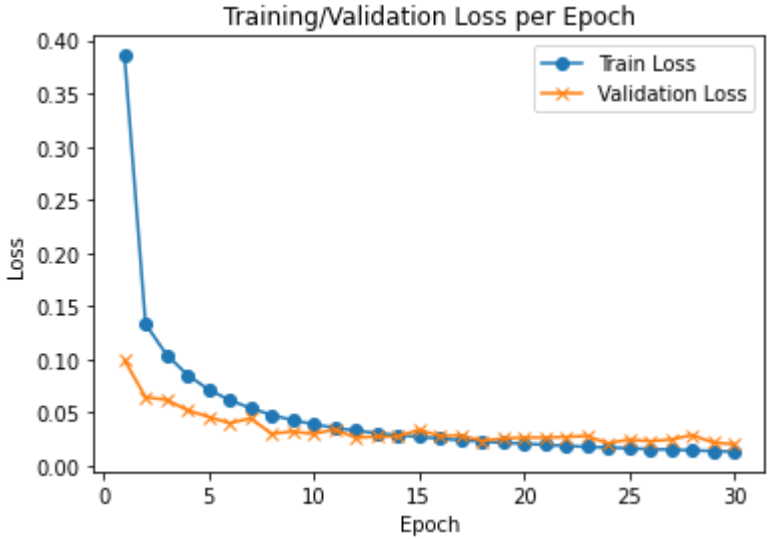
In [7]:
```
import matplotlib.pyplot as plt
f, ax = plt.subplots()
ax.plot([None] + hist.history['accuracy'], 'o-')
ax.plot([None] + hist.history['val_accuracy'], 'x-')
# Plot legend and use the best location automatically: loc = 0.
ax.legend(['Train acc', 'Validation acc'], loc = 0)
ax.set_title('Training/Validation acc per Epoch')
ax.set_xlabel('Epoch')
ax.set_ylabel('acc')
```

Out[7]:
```
Text(0, 0.5, 'acc')
```



In [8]:
```
import matplotlib.pyplot as plt
f, ax = plt.subplots()
ax.plot([None] + hist.history['loss'], 'o-')
ax.plot([None] + hist.history['val_loss'], 'x-')
# Plot legend and use the best location automatically: loc = 0.
ax.legend(['Train Loss', 'Validation Loss'], loc = 0)
ax.set_title('Training/Validation Loss per Epoch')
ax.set_xlabel('Epoch')
ax.set_ylabel('Loss')
```

Out[8]: Text(0, 0.5, 'Loss')



In [9]:
```
start = time.time()
hist2 = model.fit(x=x_train,y=y_train, epochs=30, batch_size=128, validation_split=0.2, verbose=1)
end = time.time()
print('Elapsed time: {:.1f} second(s)'.format(end - start))
```

```
Train on 48000 samples, validate on 12000 samples
Epoch 1/30
48000/48000 [==============================] - 15s 318us/step - loss: 0.0114 - accuracy: 0.9971 - val_loss:
0.0237 - val_accuracy: 0.9918
Epoch 2/30
48000/48000 [==============================] - 16s 324us/step - loss: 0.0111 - accuracy: 0.9973 - val_loss:
0.0230 - val_accuracy: 0.9918
Epoch 3/30
48000/48000 [==============================] - 15s 316us/step - loss: 0.0110 - accuracy: 0.9973 - val_loss:
0.0261 - val_accuracy: 0.9909
Epoch 4/30
48000/48000 [==============================] - 16s 327us/step - loss: 0.0109 - accuracy: 0.9974 - val_loss:
0.0241 - val_accuracy: 0.9916
Epoch 5/30
48000/48000 [==============================] - 15s 318us/step - loss: 0.0109 - accuracy: 0.9974 - val_loss:
0.0245 - val_accuracy: 0.9912
Epoch 6/30
48000/48000 [==============================] - 15s 319us/step - loss: 0.0108 - accuracy: 0.9974 - val_loss:
0.0238 - val_accuracy: 0.9917
Epoch 7/30
48000/48000 [==============================] - 15s 315us/step - loss: 0.0108 - accuracy: 0.9975 - val_loss:
0.0238 - val_accuracy: 0.9915
Epoch 8/30
48000/48000 [==============================] - 16s 324us/step - loss: 0.0107 - accuracy: 0.9974 - val_loss:
0.0234 - val_accuracy: 0.9916
Epoch 9/30
48000/48000 [==============================] - 15s 319us/step - loss: 0.0106 - accuracy: 0.9975 - val_loss:
0.0234 - val_accuracy: 0.9916
Epoch 10/30
48000/48000 [==============================] - 15s 316us/step - loss: 0.0105 - accuracy: 0.9975 - val_loss:
0.0249 - val_accuracy: 0.9913
Epoch 11/30
48000/48000 [==============================] - 15s 318us/step - loss: 0.0105 - accuracy: 0.9975 - val_loss:
0.0227 - val_accuracy: 0.9923
Epoch 12/30
48000/48000 [==============================] - 16s 330us/step - loss: 0.0105 - accuracy: 0.9974 - val_loss:
0.0238 - val_accuracy: 0.9919
Epoch 13/30
48000/48000 [==============================] - 15s 318us/step - loss: 0.0104 - accuracy: 0.9975 - val_loss:
0.0239 - val_accuracy: 0.9915
Epoch 14/30
48000/48000 [==============================] - 15s 316us/step - loss: 0.0104 - accuracy: 0.9975 - val_loss:
0.0235 - val_accuracy: 0.9917
```

```
Epoch 15/30
48000/48000 [==============================] - 15s 318us/step - loss: 0.0103 - accuracy: 0.9976 - val_loss:
0.0230 - val_accuracy: 0.9915
Epoch 16/30
48000/48000 [==============================] - 16s 326us/step - loss: 0.0103 - accuracy: 0.9977 - val_loss:
0.0236 - val_accuracy: 0.9918
Epoch 17/30
48000/48000 [==============================] - 15s 318us/step - loss: 0.0102 - accuracy: 0.9976 - val_loss:
0.0233 - val_accuracy: 0.9918
Epoch 18/30
48000/48000 [==============================] - 15s 315us/step - loss: 0.0101 - accuracy: 0.9976 - val_loss:
0.0228 - val_accuracy: 0.9921
Epoch 19/30
48000/48000 [==============================] - 15s 318us/step - loss: 0.0101 - accuracy: 0.9977 - val_loss:
0.0249 - val_accuracy: 0.9913
Epoch 20/30
48000/48000 [==============================] - 16s 327us/step - loss: 0.0101 - accuracy: 0.9975 - val_loss:
0.0232 - val_accuracy: 0.9922
Epoch 21/30
48000/48000 [==============================] - 15s 317us/step - loss: 0.0099 - accuracy: 0.9976 - val_loss:
0.0248 - val_accuracy: 0.9914
Epoch 22/30
48000/48000 [==============================] - 15s 318us/step - loss: 0.0100 - accuracy: 0.9975 - val_loss:
0.0234 - val_accuracy: 0.9916
Epoch 23/30
48000/48000 [==============================] - 15s 318us/step - loss: 0.0099 - accuracy: 0.9977 - val_loss:
0.0233 - val_accuracy: 0.9919
Epoch 24/30
48000/48000 [==============================] - 16s 329us/step - loss: 0.0098 - accuracy: 0.9977 - val_loss:
0.0236 - val_accuracy: 0.9922
Epoch 25/30
48000/48000 [==============================] - 15s 318us/step - loss: 0.0098 - accuracy: 0.9977 - val_loss:
0.0238 - val_accuracy: 0.9921
Epoch 26/30
48000/48000 [==============================] - 15s 318us/step - loss: 0.0098 - accuracy: 0.9978 - val_loss:
0.0236 - val_accuracy: 0.9921
Epoch 27/30
48000/48000 [==============================] - 15s 322us/step - loss: 0.0097 - accuracy: 0.9977 - val_loss:
0.0230 - val_accuracy: 0.9919
Epoch 28/30
48000/48000 [==============================] - 16s 325us/step - loss: 0.0097 - accuracy: 0.9977 - val_loss:
0.0246 - val_accuracy: 0.9916
Epoch 29/30
```

```
48000/48000 [==============================] - 15s 315us/step - loss: 0.0096 - accuracy: 0.9977 - val_loss:
0.0249 - val_accuracy: 0.9917
Epoch 30/30
48000/48000 [==============================] - 15s 318us/step - loss: 0.0096 - accuracy: 0.9978 - val_loss:
0.0224 - val_accuracy: 0.9917
Elapsed time: 460.6 second(s)
```

In [10]:
```
start = time.time()
hist3 = model.fit(x=x_train,y=y_train, epochs=30, batch_size=4, validation_split=0.2, verbose=1)
end = time.time()
print('Elapsed time: {:.1f} second(s)'.format(end - start))
```

```
Train on 48000 samples, validate on 12000 samples
Epoch 1/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0226 - accuracy: 0.9927 - val_loss: 0.0
406 - val_accuracy: 0.9855
Epoch 2/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0172 - accuracy: 0.9946 - val_loss: 0.0
450 - val_accuracy: 0.9872
Epoch 3/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0145 - accuracy: 0.9957 - val_loss: 0.0
337 - val_accuracy: 0.9902
Epoch 4/30
48000/48000 [==============================] - 57s 1ms/step - loss: 0.0129 - accuracy: 0.9957 - val_loss: 0.0
359 - val_accuracy: 0.9885
Epoch 5/30
48000/48000 [==============================] - 53s 1ms/step - loss: 0.0122 - accuracy: 0.9960 - val_loss: 0.0
320 - val_accuracy: 0.9893
Epoch 6/30
48000/48000 [==============================] - 53s 1ms/step - loss: 0.0103 - accuracy: 0.9968 - val_loss: 0.0
348 - val_accuracy: 0.9900
Epoch 7/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0090 - accuracy: 0.9974 - val_loss: 0.0
293 - val_accuracy: 0.9906
Epoch 8/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0076 - accuracy: 0.9979 - val_loss: 0.0
265 - val_accuracy: 0.9913
Epoch 9/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0069 - accuracy: 0.9979 - val_loss: 0.0
243 - val_accuracy: 0.9922
Epoch 10/30
48000/48000 [==============================] - 53s 1ms/step - loss: 0.0063 - accuracy: 0.9983 - val_loss: 0.0
257 - val_accuracy: 0.9923
Epoch 11/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0063 - accuracy: 0.9981 - val_loss: 0.0
237 - val_accuracy: 0.9932
Epoch 12/30
48000/48000 [==============================] - 51s 1ms/step - loss: 0.0058 - accuracy: 0.9983 - val_loss: 0.0
326 - val_accuracy: 0.9910
Epoch 13/30
48000/48000 [==============================] - 51s 1ms/step - loss: 0.0047 - accuracy: 0.9987 - val_loss: 0.0
235 - val_accuracy: 0.9927
Epoch 14/30
48000/48000 [==============================] - 51s 1ms/step - loss: 0.0039 - accuracy: 0.9990 - val_loss: 0.0
330 - val_accuracy: 0.9916
```

```
Epoch 15/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0037 - accuracy: 0.9990 - val_loss: 0.0
232 - val_accuracy: 0.9935
Epoch 16/30
48000/48000 [==============================] - 64s 1ms/step - loss: 0.0039 - accuracy: 0.9989 - val_loss: 0.0
271 - val_accuracy: 0.9923
Epoch 17/30
48000/48000 [==============================] - 51s 1ms/step - loss: 0.0033 - accuracy: 0.9992 - val_loss: 0.0
331 - val_accuracy: 0.9910
Epoch 18/30
48000/48000 [==============================] - 53s 1ms/step - loss: 0.0028 - accuracy: 0.9992 - val_loss: 0.0
286 - val_accuracy: 0.9926
Epoch 19/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0028 - accuracy: 0.9994 - val_loss: 0.0
282 - val_accuracy: 0.9915
Epoch 20/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0025 - accuracy: 0.9995 - val_loss: 0.0
320 - val_accuracy: 0.9912
Epoch 21/30
48000/48000 [==============================] - 51s 1ms/step - loss: 0.0023 - accuracy: 0.9995 - val_loss: 0.0
260 - val_accuracy: 0.9930
Epoch 22/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0023 - accuracy: 0.9994 - val_loss: 0.0
259 - val_accuracy: 0.9928
Epoch 23/30
48000/48000 [==============================] - 51s 1ms/step - loss: 0.0018 - accuracy: 0.9996 - val_loss: 0.0
286 - val_accuracy: 0.9925
Epoch 24/30
48000/48000 [==============================] - 51s 1ms/step - loss: 0.0019 - accuracy: 0.9996 - val_loss: 0.0
336 - val_accuracy: 0.9903
Epoch 25/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0016 - accuracy: 0.9997 - val_loss: 0.0
258 - val_accuracy: 0.9926
Epoch 26/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0015 - accuracy: 0.9997 - val_loss: 0.0
310 - val_accuracy: 0.9923
Epoch 27/30
48000/48000 [==============================] - 52s 1ms/step - loss: 0.0014 - accuracy: 0.9997 - val_loss: 0.0
322 - val_accuracy: 0.9915
Epoch 28/30
48000/48000 [==============================] - 54s 1ms/step - loss: 0.0015 - accuracy: 0.9996 - val_loss: 0.0
257 - val_accuracy: 0.9931
Epoch 29/30
```

```
48000/48000 [==============================] - 53s 1ms/step - loss: 0.0012 - accuracy: 0.9997 - val_loss: 0.0
311 - val_accuracy: 0.9922
Epoch 30/30
48000/48000 [==============================] - 52s 1ms/step - loss: 9.6999e-04 - accuracy: 0.9999 - val_loss:
0.0288 - val_accuracy: 0.9925
Elapsed time: 1577.9 second(s)
```

In [11]:
```
start = time.time()
hist4 = model.fit(x=x_train,y=y_train, epochs=30, batch_size=4096, validation_split=0.2, verbose=1)
end = time.time()
print('Elapsed time: {:.1f} second(s)'.format(end - start))
```

```
Train on 48000 samples, validate on 12000 samples
Epoch 1/30
48000/48000 [==============================] - 14s 289us/step - loss: 7.9271e-04 - accuracy: 0.9999 - val_los
s: 0.0288 - val_accuracy: 0.9925
Epoch 2/30
48000/48000 [==============================] - 14s 286us/step - loss: 7.8528e-04 - accuracy: 0.9999 - val_los
s: 0.0287 - val_accuracy: 0.9925
Epoch 3/30
48000/48000 [==============================] - 13s 281us/step - loss: 7.7865e-04 - accuracy: 0.9999 - val_los
s: 0.0287 - val_accuracy: 0.9926
Epoch 4/30
48000/48000 [==============================] - 14s 284us/step - loss: 7.7247e-04 - accuracy: 0.9999 - val_los
s: 0.0287 - val_accuracy: 0.9926
Epoch 5/30
48000/48000 [==============================] - 14s 296us/step - loss: 7.6673e-04 - accuracy: 0.9999 - val_los
s: 0.0287 - val_accuracy: 0.9926
Epoch 6/30
48000/48000 [==============================] - 14s 284us/step - loss: 7.6138e-04 - accuracy: 0.9999 - val_los
s: 0.0286 - val_accuracy: 0.9926
Epoch 7/30
48000/48000 [==============================] - 14s 283us/step - loss: 7.5642e-04 - accuracy: 0.9999 - val_los
s: 0.0286 - val_accuracy: 0.9926
Epoch 8/30
48000/48000 [==============================] - 14s 281us/step - loss: 7.5166e-04 - accuracy: 0.9999 - val_los
s: 0.0286 - val_accuracy: 0.9926
Epoch 9/30
48000/48000 [==============================] - 14s 295us/step - loss: 7.4735e-04 - accuracy: 0.9999 - val_los
s: 0.0286 - val_accuracy: 0.9926
Epoch 10/30
48000/48000 [==============================] - 14s 285us/step - loss: 7.4314e-04 - accuracy: 0.9999 - val_los
s: 0.0286 - val_accuracy: 0.9926
Epoch 11/30
48000/48000 [==============================] - 13s 281us/step - loss: 7.3952e-04 - accuracy: 0.9999 - val_los
s: 0.0286 - val_accuracy: 0.9926
Epoch 12/30
48000/48000 [==============================] - 14s 282us/step - loss: 7.3583e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 13/30
48000/48000 [==============================] - 14s 288us/step - loss: 7.3233e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 14/30
48000/48000 [==============================] - 14s 291us/step - loss: 7.2906e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
```

```
Epoch 15/30
48000/48000 [==============================] - 13s 280us/step - loss: 7.2605e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 16/30
48000/48000 [==============================] - 14s 284us/step - loss: 7.2313e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 17/30
48000/48000 [==============================] - 14s 282us/step - loss: 7.2037e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 18/30
48000/48000 [==============================] - 14s 292us/step - loss: 7.1786e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 19/30
48000/48000 [==============================] - 14s 285us/step - loss: 7.1540e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 20/30
48000/48000 [==============================] - 14s 287us/step - loss: 7.1308e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 21/30
48000/48000 [==============================] - 14s 283us/step - loss: 7.1095e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 22/30
48000/48000 [==============================] - 14s 296us/step - loss: 7.0891e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 23/30
48000/48000 [==============================] - 14s 284us/step - loss: 7.0680e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 24/30
48000/48000 [==============================] - 14s 284us/step - loss: 7.0494e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 25/30
48000/48000 [==============================] - 13s 280us/step - loss: 7.0307e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 26/30
48000/48000 [==============================] - 14s 288us/step - loss: 7.0120e-04 - accuracy: 0.9999 - val_los
s: 0.0285 - val_accuracy: 0.9926
Epoch 27/30
48000/48000 [==============================] - 14s 290us/step - loss: 6.9963e-04 - accuracy: 0.9999 - val_los
s: 0.0284 - val_accuracy: 0.9926
Epoch 28/30
48000/48000 [==============================] - 14s 284us/step - loss: 6.9795e-04 - accuracy: 0.9999 - val_los
s: 0.0284 - val_accuracy: 0.9926
Epoch 29/30
```

```
48000/48000 [==============================] - 14s 281us/step - loss: 6.9644e-04 - accuracy: 0.9999 - val_los
s: 0.0284 - val_accuracy: 0.9926
Epoch 30/30
48000/48000 [==============================] - 14s 284us/step - loss: 6.9492e-04 - accuracy: 0.9999 - val_los
s: 0.0284 - val_accuracy: 0.9926
Elapsed time: 411.5 second(s)
```