

```
pip install tensorflow
```

```

↳ Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Requirement already satisfied: tensorflow in /usr/local/lib/python3.7/dist-packages (1.15.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.7/dist-packages (1.1.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (21.3)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.7/dist-packages (1.6.3)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages (3.1.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (59.0.1)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages (1.1.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.7/dist-packages (1.0.0)
Requirement already satisfied: flatbuffers<2,>=1.12 in /usr/local/lib/python3.7/dist-packages (1.12.0)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.7/dist-packages (1.21.0)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in /usr/local/lib/python3.7/dist-packages (3.19.6)
Requirement already satisfied: tensorflow-estimator<2.10.0,>=2.9.0rc0 in /usr/local/lib/python3.7/dist-packages (2.9.0rc0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.7/dist-packages (0.23.1)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.7/dist-packages (13.0.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.7/dist-packages (1.24.3)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-packages (1.12.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-packages (0.1.1)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.7/dist-packages (3.7.4)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages (2.3.2)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /usr/local/lib/python3.7/dist-packages (0.4.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages (1.11.0)
Requirement already satisfied: tensorboard<2.10,>=2.9 in /usr/local/lib/python3.7/dist-packages (2.9.0rc0)
Requirement already satisfied: keras<2.10.0,>=2.9.0rc0 in /usr/local/lib/python3.7/dist-packages (2.9.0rc0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.7/dist-packages (0.23.0)
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-packages (1.5.2)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.7/dist-packages (2.27.1)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages (2.6.8)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.7/dist-packages (1.0.1)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.7/dist-packages (0.4.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.7/dist-packages (1.6.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/dist-packages (1.6.3)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.7/dist-packages (0.6.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist-packages (0.2.1)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-packages (3.1.4)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (4.1.0)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist-packages (0.7.0)
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-packages (4.4.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (0.5.2)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist-packages (0.4.8)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (3.0.2)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (2022.9.24)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (1.25.11)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (2.5)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages (3.0.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (2.4.7)

```

```

import tensorflow as tf
import tensorflow.keras as keras
from tensorflow.keras import layers
keras.backend.set_image_data_format('channels_last')

```

```
import matplotlib.pyplot as plt
```

```

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D, Flatten

```

```

import numpy as np
import pandas as pd

```

```

train = pd.read_csv("/content/drive/MyDrive/mnist dataset/train.csv")
test = pd.read_csv("/content/drive/MyDrive/mnist dataset/test.csv")

```

```
train.head()
```

	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	..
0	1	0	0	0	0	0	0	0	0	0	..
1	0	0	0	0	0	0	0	0	0	0	..
2	1	0	0	0	0	0	0	0	0	0	..
3	4	0	0	0	0	0	0	0	0	0	..
4	0	0	0	0	0	0	0	0	0	0	..

5 rows × 785 columns



```
test.head()
```

	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	.
0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	

5 rows × 784 columns



```

image_size = 28*28
image_size

```

784

```
X_train = train.drop("label", axis = 1).copy()
X_test = test.copy()
Y_train = train["label"].copy()
```

```
X_train.describe()
```

	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8
count	42000.0	42000.0	42000.0	42000.0	42000.0	42000.0	42000.0	42000.0	42000.0
mean	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
std	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
min	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25%	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50%	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
75%	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
max	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

8 rows × 784 columns



```
#Normalize values
```

```
X_train = X_train/255.0
```

```
X_test = X_test/255.0
```

```
#Reshape to 28 * 28 so that we can see the images
```

```
X_train = X_train.values.reshape(-1, 28, 28, 1)
```

```
X_test = X_test.values.reshape(-1, 28, 28, 1)
```

```
import random
```

```
no_images=len(X_train)
```

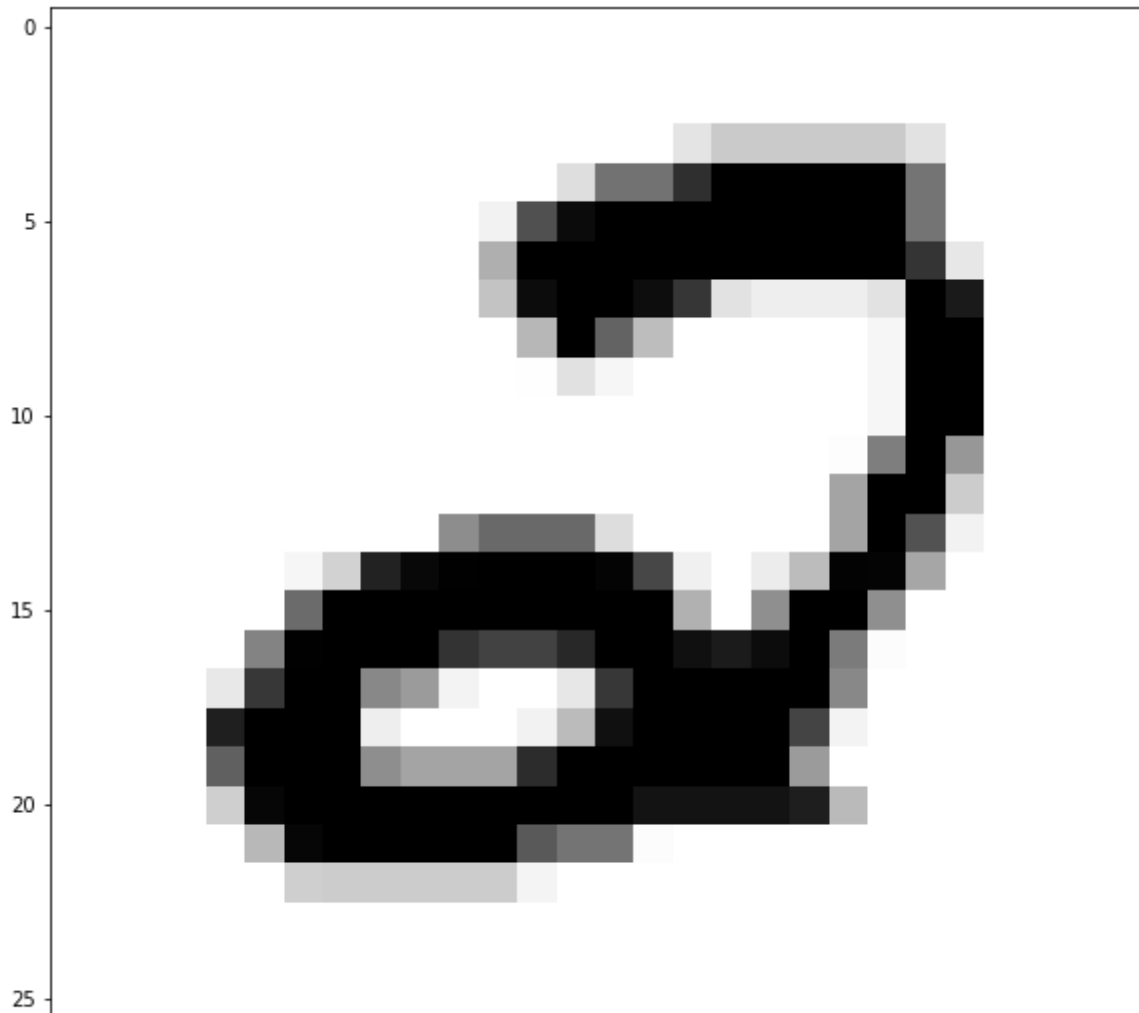
```
# Display random Image
```

```
fig, ax = plt.subplots(figsize=(10, 10))
```

```
plt.imshow(X_train[random.randint(0,no_images), :, :, 0], cmap='Greys', interpolation='nearest')
```

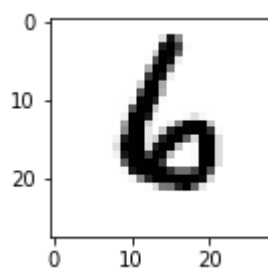
```
# replace random.randint(0,no_images) in code above with a number if you want to see specific image
#This displays a random image each time
```

```
plt.show()
```



```
fig, ax = plt.subplots(figsize=(2,2))
```

```
plt.imshow(X_train[random.randint(0, no_images), :, :, 0], cmap = "Greys", interpolation =  
plt.show())
```



```
validation_size = 0.2
```

```
X_train, X_val, Y_train, Y_val = train_test_split(X_train, Y_train, test_size = validation
```

```
#Get one-hot encoding
```

```
Y_train = keras.utils.to_categorical(Y_train, num_classes = 10)
```

```
Y_val = keras.utils.to_categorical(Y_val, num_classes = 10)
```

```
def CNN():
```

```
    model = Sequential()
```

```

model.add(Conv2D(filters = 32, kernel_size = (4,4), input_shape = (28,28,1), activation = 'relu'))
model.add(MaxPool2D(pool_size = (2,2)))

model.add(Flatten())

model.add(Dense(128, activation = "relu"))
model.add(Dense(256, activation = "relu"))
model.add(Dense(512, activation = "relu"))

model.add(Dense(10, activation = "softmax"))

```

```

return model

```

```

model = CNN()

```

```

model.compile(optimizer = "Adam", loss = "CategoricalCrossentropy", metrics = "accuracy")

```

```

model.summary()

```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 25, 25, 32)	544
max_pooling2d (MaxPooling2D)	(None, 12, 12, 32)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 128)	589952
dense_1 (Dense)	(None, 256)	33024
dense_2 (Dense)	(None, 512)	131584
dense_3 (Dense)	(None, 10)	5130
=====		
Total params: 760,234		
Trainable params: 760,234		
Non-trainable params: 0		

```

from tensorflow.keras.callbacks import EarlyStopping

```

```

early_stop = EarlyStopping(monitor = "val_loss", patience = 1)

```

```

training = model.fit(X_train, Y_train, validation_data = (X_val, Y_val), batch_size = 64,

```

```

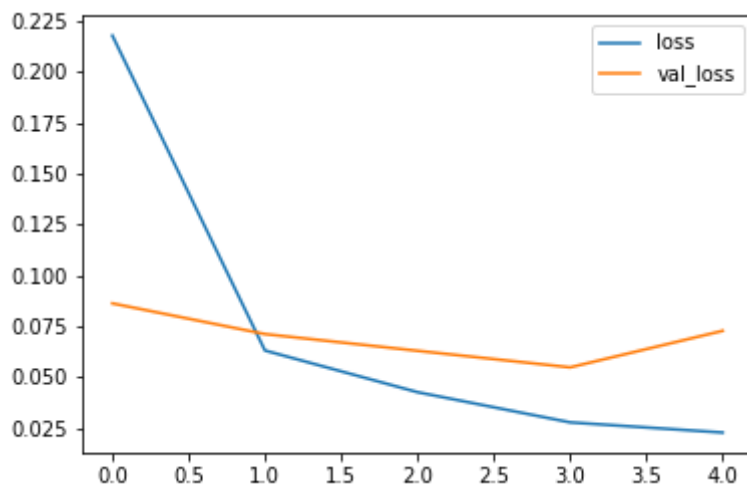
Epoch 1/50
525/525 [=====] - 30s 54ms/step - loss: 0.2180 - accuracy: 0.9320
Epoch 2/50
525/525 [=====] - 29s 54ms/step - loss: 0.0631 - accuracy: 0.9800
Epoch 3/50
525/525 [=====] - 26s 49ms/step - loss: 0.0426 - accuracy: 0.9850
Epoch 4/50
525/525 [=====] - 21s 40ms/step - loss: 0.0277 - accuracy: 0.9900
Epoch 5/50
525/525 [=====] - 24s 45ms/step - loss: 0.0227 - accuracy: 0.9920

```

```
metrics = pd.DataFrame(model.history.history)
```

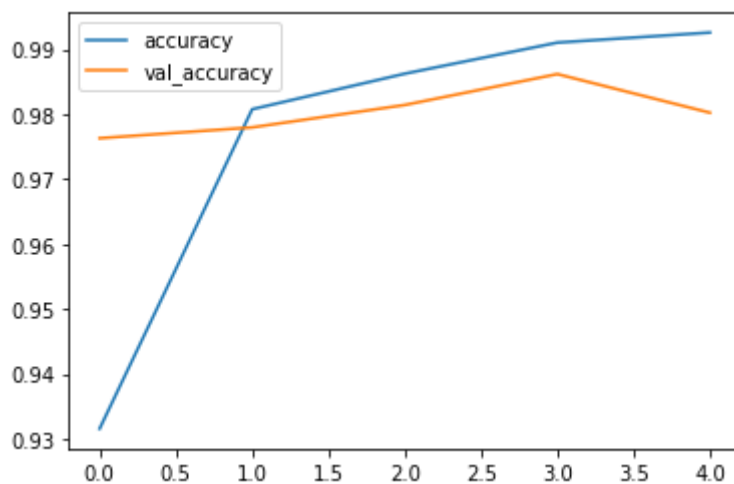
```
metrics[["loss", "val_loss"]].plot()
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f963fdf09d0>



```
metrics[["accuracy", "val_accuracy"]].plot()
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f96432e5f50>



```
model.metrics_names
```

```
['loss', 'accuracy']
```

```
from sklearn.metrics import classification_report, confusion_matrix
```

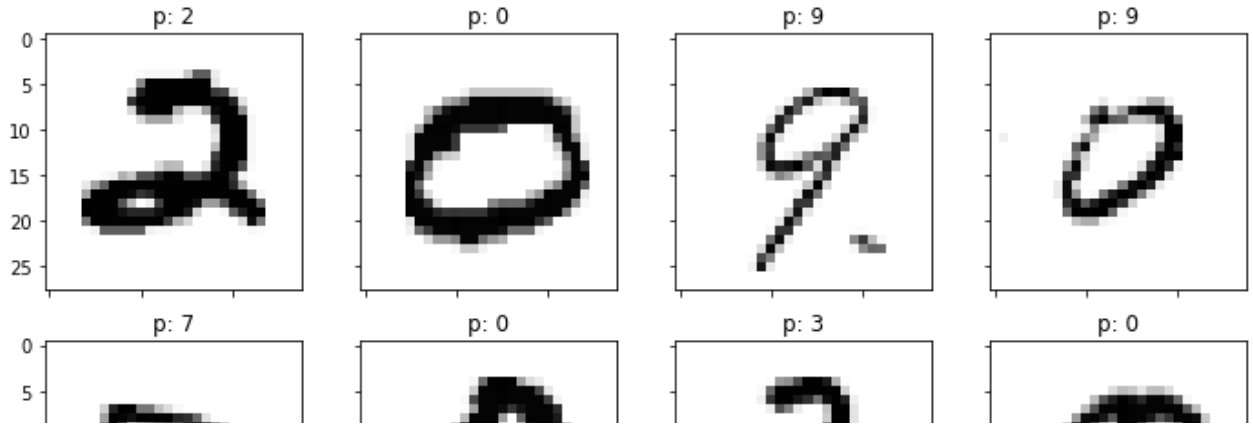
```
Y_val.shape
```

```
(8400, 10)
```

```
def predict(model, X, imgs):  
    s = int(np.sqrt(imgs))  
    fig, ax = plt.subplots(s, s, sharex=True, sharey=True, figsize=(15, 15))  
    ax = ax.flatten()  
    preds = model.predict(X[:imgs])  
    for i in range(imgs):  
        y_pred = np.argmax(preds[i])  
        img = X[i].reshape(28, 28)  
        ax[i].imshow(img, cmap='Greys', interpolation='nearest')  
        ax[i].set_title(f'p: {y_pred}')
```

```
predict(model, X_test, 25)
```

1/1 [=====] - 0s 117ms/step

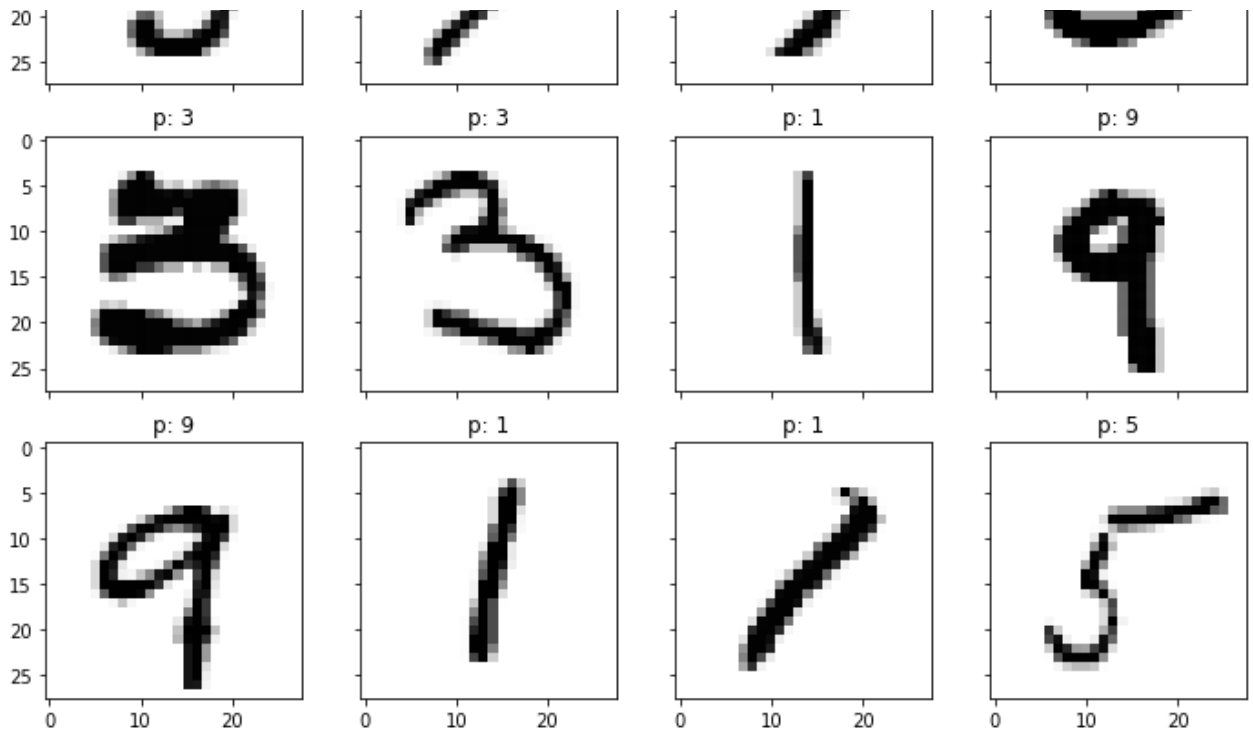


```
y_pred = model.predict(X_test)
y_pred = np.argmax(y_pred, axis=1)
```

875/875 [=====] - 7s 8ms/step

n: 5 n: 7 n: 4 n: 0

```
file_name = "submission.csv"
y_pred = pd.Series(y_pred, name='Label')
sub = pd.concat([pd.Series(range(1, 28001), name="ImageId"), y_pred], axis=1)
sub.to_csv(file_name, index=False)
```



✓ 0s completed at 11:50

● ×