

```

import csv
import os
import pickle
import csv
import pandas as pd
import numpy as np
from numpy import array
import pydot
from numpy import loadtxt
from sklearn import datasets
import tensorflow as tf
from tensorflow.keras.models import load_model, save_model
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Dense, Dropout, Flatten
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
# Make numpy values easier to read.
np.set_printoptions(precision=3, suppress=True)

#read the input data from cvs
img_train = pd.read_csv("/Users/sherry/Desktop/nn-assignment/trainset.csv")
img_test = pd.read_csv("/Users/sherry/Desktop/nn-assignment/testset.csv")

#slice data and label
train_label = img_train.values[:,0]
train_data = np.asarray(img_train.values[:,1:785])
test_data = np.asarray(img_test.values[0:14002,:])

# Scale images to the [0, 1] range
train_data = train_data.astype("float32") / 255
test_data = test_data.astype("float32") / 255

#split string into list of matrix
img_rows, img_cols = 28, 28
n = train_data.shape[0] #28000
m = train_data.shape[1] #748
train_data = tf.reshape(train_data, [-1,28,28,1])
test_data = tf.reshape(test_data, [-1,28,28,1])
train_label = tf.reshape(train_label, [28000,1])
#print(train_label.shape, train_data.shape)

input_shape = (28, 28, 1)
num_classes = 10

model = keras.Sequential(
    [
        keras.Input(shape=input_shape),
        layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Flatten(),
        layers.Dropout(0.5),
        layers.Dense(num_classes, activation="softmax"),
    ]
)

```

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

batch_size = 128
epochs = 15
model.fit(train_data, train_label, batch_size=batch_size, epochs=epochs,
validation_split=0.2)

predictions = model.predict(test_data)
#print(predictions)

classes = np.argmax(predictions, axis = 1)

#print('\n'.join(map(str, classes)))

submit = map(str, classes)
list1=list(range(1,14001))
list2=list(submit)
test1=pd.DataFrame(columns=['ImageId'],data=list1)
test2=pd.DataFrame(columns=['Label'],data=list2)
test = pd.concat([test1,test2],axis=1)
test.to_csv('desktop/submission.csv',index = False)
```