

```

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Activation
from keras.optimizers import SGD
from sklearn.model_selection import train_test_split
import time
from keras.datasets import mnist;
from keras.utils import to_categorical
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
from keras import models
from keras import layers

drive.mount('/content/drive')

x_train = train_images.reshape((60000,28,28,1))
x_test = test_images.reshape((10000,28,28,1))
x_train = x_train.astype('float32')/255
x_test = x_test.astype('float32')/255
y_train = to_categorical(train_labels)
y_test = to_categorical(test_labels)

model = models.Sequential()
model.add( layers.Conv2D(32, (3,3), activation='relu',
input_shape=(28,28,1)) )
model.add( layers.MaxPooling2D((2,2)) )
model.add( layers.Flatten() )
model.add( layers.Dense(64,activation='relu') )
model.add( layers.Dense(10,activation='softmax') )
model.compile( optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'] )

history = model.fit(x_train, y_train, epochs=5, batch_size=4, verbose=1)

```

---

```

predict_x = model.predict(x_test)
y_result = np.argmax(predict_x,axis=1)
bledy = 0
for i in range(len(y_test)):
    if y_test[i][y_result[i]]!=1:
        bledy += 1

print("Liczba bledow ciagu testowego: ", bledy,"odsetek źle rozpoznanych",bledy/len(y_test)*100)

predict_x = model.predict(x_train)
y_result = np.argmax(predict_x,axis=1)
bledy = 0
for i in range(len(y_train)):
    if y_train[i][y_result[i]]!=1:
        bledy += 1
    img = train_images[i].reshape(28,28)
    plt.imshow( img, cmap='Greys')
    plt.show()

print("Liczba bledow ciagu treningowego: ", bledy,"odsetek źle rozpoznanych",bledy/len(y_train)*100)

```

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz>  
11490434/11490434 [=====] - 2s 0us/step  
Epoch 1/5  
15000/15000 [=====] - 65s 4ms/step - loss: 0.1307 - accuracy: 0.9607  
Epoch 2/5  
15000/15000 [=====] - 52s 3ms/step - loss: 0.0487 - accuracy: 0.9847  
Epoch 3/5  
15000/15000 [=====] - 52s 3ms/step - loss: 0.0313 - accuracy: 0.9898  
Epoch 4/5  
15000/15000 [=====] - 52s 3ms/step - loss: 0.0212 - accuracy: 0.9934  
Epoch 5/5  
15000/15000 [=====] - 52s 3ms/step - loss: 0.0158 - accuracy: 0.9949  
313/313 [=====] - 1s 2ms/step  
Liczba błędów ciągu testowego: 137 odsetek źle rozpoznanych 1.37  
1875/1875 [=====] - 3s 2ms/step

