#قمت بإستخدام الداتا التي تحوي على أرقام في عمود LETTER

لعد القدرة على تحويلها :

import pandas as pd

df = pd.read\_csv('/content/drive/MyDrive/الداتا بدل حروف أرقام.csv')

dataset = df.values

dataset

# لمعرفة أنواع الداتا :

df.dtypes

#هذه الدالة تقوم بتحويل الداتا من

Str To Float

df['letter']=df['letter'].astype(float)

df.dtypes

#هنا تم تحديد قيم X

من عمود 0 ..... عدد الأعمدة 16

وكذلك y = col [16]

وفي النهاية طلبنا عرض بيانات X

X = dataset[:,0:16]

Y = dataset[:,16]

from sklearn import preprocessing

min\_max\_scaler = preprocessing.MinMaxScaler()

X\_scale = min\_max\_scaler.fit\_transform(X)

print(X\_scale[0][15])

#هنا تم تقسيم البيانات للتدريب والاختبار والتحقق

from sklearn.model\_selection import train\_test\_split

X\_train, X\_val\_and\_test, Y\_train, Y\_val\_and\_test = train\_test\_split(X\_scale, Y, test\_size=0.25)

X\_val, X\_test, Y\_val, Y\_test = train\_test\_split(X\_val\_and\_test, Y\_val\_and\_test, test\_size=0.5)

X\_train

هذه مكتبات تم أستدعائها من اجل بناء الشبكة العصبية #

from keras.models import Sequential

from keras.layers import Dense

import numpy as np

from keras.layers import Dropout

قمت ببناء الشبكة العصبية وهنالك شرح بسيط في العرض التقديمي

(ANN (30.11))

للدوال والعوامل المستخدمة ويقال لها المعاملات العليا #

from keras import initializers

initializer = initializers.GlorotNormal()

model= Sequential([

Dense(32, activation='sigmoid', input\_shape=(16,)),

  Dropout(0.2),

Dense(512, activation='relu'),

  Dropout(0.2),

Dense(256, activation='sigmoid'),

  Dropout(0.5),

Dense(256, activation='tanh'),

  Dropout(0.2),

Dense(128, activation='relu'),

  Dropout(0.5),

Dense(600, activation='LeakyReLU'),

  Dropout(0.3),

Dense(400, activation='relu'),

  Dropout(0.2),

Dense(365, activation='LeakyReLU'),

  Dropout(0.4),

Dense(370, activation='relu'),

Dense(250, activation='relu'),

Dense(125, activation='LeakyReLU'),

  Dropout(0.2),

Dense(215, activation='relu'),

Dense(128, activation='LeakyReLU'),

  Dropout(0.2),

Dense(1, activation='LeakyReLU'),

])

model.compile(optimizer='Adadelta', loss='MeanAbsoluteError', metrics=['accuracy'])

#في الطبقات أعلاه قمت بتجربة كل المعاملات العليا وتغير عدد الطبقات بشل غير متناهي ولكن الدقة لم تتحسن

فيها عدد الفترات التدريبية وكذلك دفعات تدريب الداتا #

hist = model.fit(xx, yy,batch\_size=8, epochs=5, validation\_data=(x\_v, y\_v))

عرض الدقة الكلية#

model.evaluate(X\_test, Y\_test)[1]

عرض model less #

import matplotlib.pyplot as plt

plt.plot(hist.history['loss'])

plt.plot(hist.history['val\_loss'])

plt.title('Model loss')

plt.ylabel('Loss')

plt.xlabel('Epoch')

plt.legend(['Train', 'Val'], loc='upper right')

plt.show()

عرض model الدقة

plt.plot(hist.history['accuracy'])

plt.plot(hist.history['val\_accuracy'])

plt.title('Model accuracy')

plt.ylabel('Accuracy')

plt.xlabel('Epoch')

plt.legend(['Train', 'Val'], loc='lower right')

plt.show()