Proiect RNA

Baza de date aleasa : CrabAgePrediction

1. Descrierea problemei: Aflarea varstei crabului

Pentru un fermier de crabi, cunoasterea varstei potrivite a crabului il ajuta sa decida daca si cand sa recolteze crabii.

2.Descrierea bazei de date:

* Sursa: <https://www.kaggle.com/datasets/sidhus/crab-age-prediction?resource=download>
* Descrierea parametrilor: Sex(Gender – Male , Female, Indeterminate) , Lungimea crabului , Diametrul crabului, Inaltimea crabului, Greutatea crabului, Greutatea fara carapace, Greutatea materiei moale in cavitatea viscerala a crabului, Greutatea carapacei, Varsta
* Unitatea de masura a lor : Lungimea, Diametrul si Inaltimea – in picioare ( in feet) , Restul de greutati in uncii ( ounces) , Varsta ( in luni)
* 1 Pound = 16 ounces ---//--- 1 foot = 30.48 cms
* Numarul de inregistrari din baza de date: 3893

3. Care este variabila dependenta? Care sunt varibilele independente?

* Variabile independente : Lungime, Diametrul, Inaltimea si celelalte greutati
* Variabila dependenta: Varsta

4. Detalii privind prelucrarea initiala a bazei de date:

* Nu sunt valori lipsa
* Am eliminat campul Sex(Gender)
* Daca se afla valori nule acestea se vor completa cu “?”

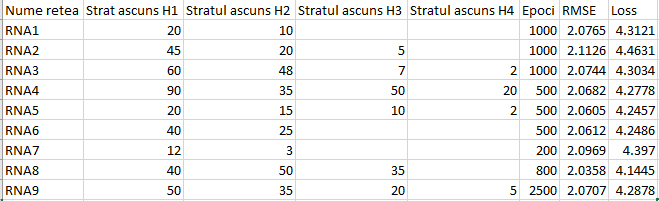
6. Justificarea tipului modelului: de ce este de regresie?

Este regresie doarece la sfarsit exista un numar ca date de iesire

7. Ce parametru/parametri statistici ati ales pentru evaluarea modelului?

Loss si RMSE

*8. Tabel centralizator in care se prezinta rezultatele experimentale pentru determinarea unei arhitecturi potrivite ale retelei neuronale artificiale.*



9. Codul sursa cu arhitectura finala

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

*import* pandas *as* pd

*from* scipy.stats *import* zscore

*from* keras.models *import* Sequential

*from* keras.layers.core *import* Dense, Activation

*import* os

*import* numpy *as* np

*from* sklearn *import* metrics

*from* keras.callbacks *import* EarlyStopping

*from* keras.callbacks *import* ModelCheckpoint

*import* matplotlib.pyplot *as* plt

path = "./data/"

save\_path = "./data/"

filename\_read = os.path.join(path,"CrabAgePrediction.csv")

df = pd.read\_csv(filename\_read,na\_values=['NA','?'])  # *se pune automat "?" in spatii libere*

crab = df['Sex']

df.drop('Sex',1,inplace=True) # *dam drop la coloana neinteresata*

dataset=df.values

x=dataset[:,0:6]    # *au ramas 8 coloane deoarece am dat drop la coloana Sex*

y=dataset[:,7]      # *coloana 8 este Age*

# *Split into train/test*

x\_train, x\_test, y\_train, y\_test = train\_test\_split( x, y, test\_size=0.25, random\_state=42)

model = Sequential()

model.add(Dense(40, input\_dim=x.shape[1], activation='relu')) # *Stratul ascuns 1*

model.add(Dense(50)) # *Stratul ascuns 2*

model.add(Dense(35)) # *Stratul ascuns 3*

model.add(Dense(1)) # *output*

model.compile(loss='mean\_squared\_error', optimizer='adam')

model.fit(x\_train,y\_train,validation\_data=(x\_test,y\_test),verbose=2,epochs=800)

# *Predict*

pred = model.predict(x\_test)

# *Measure MSE error*

score = metrics.mean\_squared\_error(pred,y\_test)

print("Final score (MSE): {}".format(score))

# *Measure RMSE error*

score = np.sqrt(metrics.mean\_squared\_error(pred,y\_test))

print("Final score (RMSE): {}".format(score))

for i in range(30):

print("{}. Sex: {}, Age: {}, predicted Age: {}".format(i+1,crab[i],y[i],pred[i]))

#*--------------------------------------------------------------------------*

# *Grafic de regresie*

def *chart\_regression*(pred, y, sort=True):

    t = pd.DataFrame({'pred': pred, 'y': y.flatten()})

*if* sort:

        t.sort\_values(by=['y'], inplace=True)

    plt.plot(t['y'].tolist(), label='expected')

    plt.plot(t['pred'].tolist(), label='prediction')

    plt.ylabel('output')

    plt.legend()

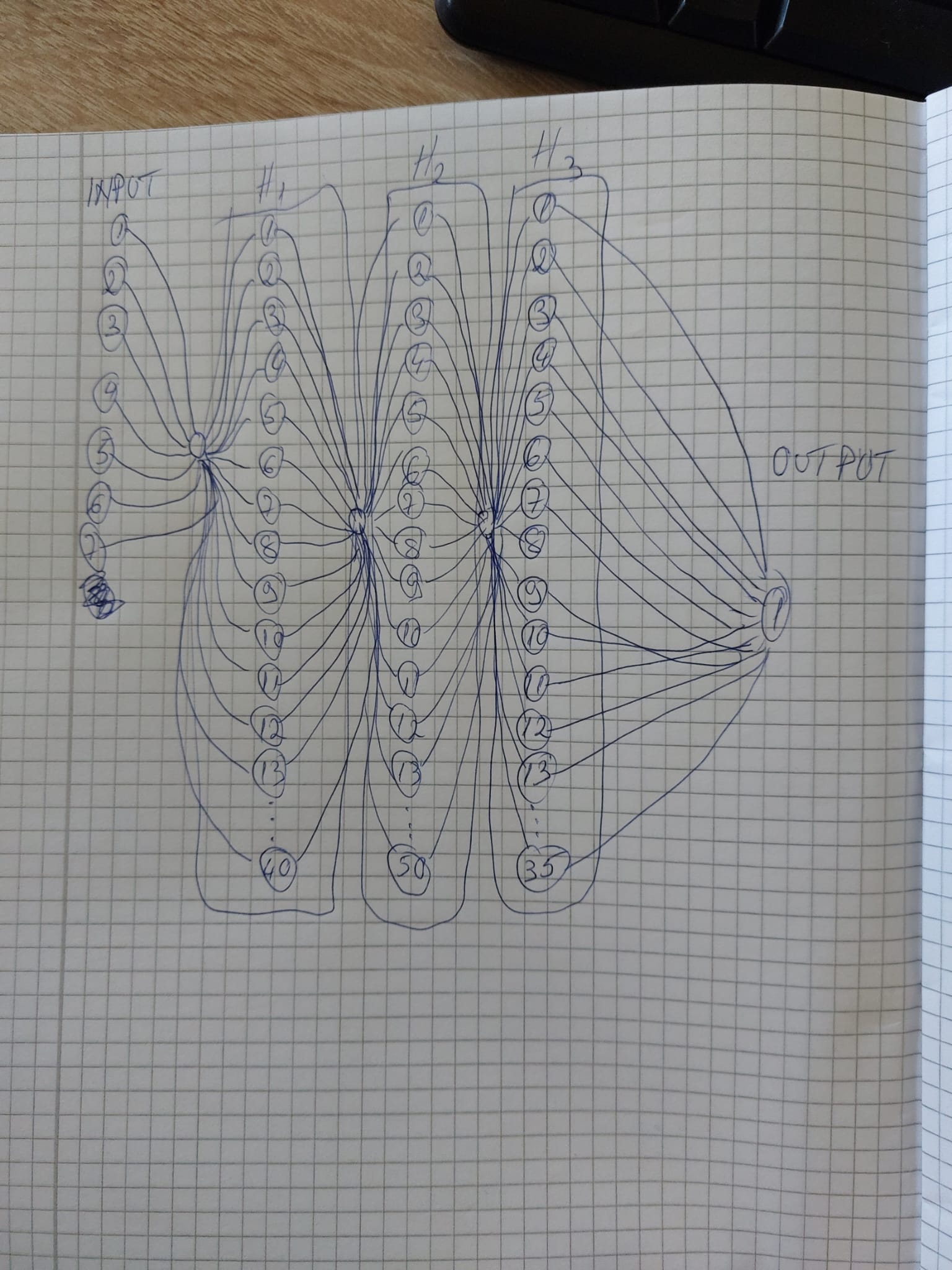
    plt.show()

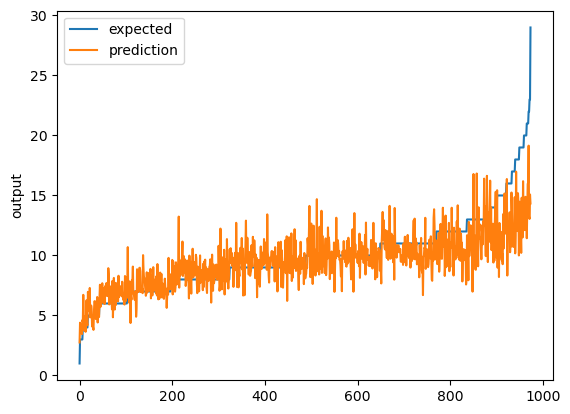
# *Trasează graficul*

chart\_regression(pred.flatten(),y\_test)

# *Salveaza toata reteaua in HDF5*

model.save(os.path.join(save\_path,"network\_fine.h5"))





10. Concluzie finala

Am folosit modelul de antrenare/validare la care am facut diverse experimente pentru a afla care parametrii sunt cei mai potriviti pentru aceasta problema.

RNA 8 este valoarea cea mai mica in aceste experimente facute si putem spune ca este cea mai potrivita arhitectura pentru regresie.

Student:

Neagu Ionut Adrian