UNIVERSITAT
ROVIRA i VIRGILI

Neural Evolution Computation

A1

David George Farouk Marei

November 22, 2021

Contents

1 Description of The implementation **2**

2 Data preparation **2**

3 Implementation Decision and hyper parameters **2**

4 Cross Validation **3**

5 Predictions **3**

1 Description of The implementation

The implemented Neural Network in this assignment consist of different parts. It has 3 type of layer. the first layer is input layer, which is equivalent to the number of input per pattern. next we have the hidden layer which is chosen by a parameter by the user (each hidden layer have neurons equivalent to the input layer) and finally the last layer is the output layer that consist of one neuron (regression model). the 2 main function of this network is forward propagation and back propagation. each one of those function had sub functions that helped to achieve the final goal. the tools that where used in this assignment in Visual Studio code. Python3.8 where the Libraries used are numpy, pandas, list,matplotlib, sklearn (minmax scalar, cross-validation, kfold and MLR) and Statistics. finally i used a virtual environment to install all those requirements on it.

2 Data preparation

for the data preparation i used the supplied data set from the professor and read it in python panda data frame and applied a min max scalar on it. the range of the data after this process is between 0 and 1. the data was split to input and output and the whole data was split to 80 percent for training and 20 percent for testing.For the cross validation we split to 4 different splits for training data, however the testing data was constant.

3 Implementation Decision and hyper parameters

Switched to a less dynamic hidden layer creating algorithm as my previous code was not learning. However, a good result was achieved with this structured. to test the assignment. a virtual environment is needed to be created and the requirements.txt file needed to be installed. pip vevenv create myprojectenv and then activate the environment. pip install -r requirements.txt. and then python neural-network.py or open it in Visual Studio Code. A some learning rate were tried, but 0.7 worked the best. 300 epochs for training our data. Four hidden layer and every hidden layer have number of neuron equivalent to the input layer. bias and momentum are randomly generated and integrated in neural network class.

4 Cross Validation

For the first data set a 4 split cross validation is used with constant testing input that have been used in the four training experience. in the next table, it will show the mean loss and the mean prediction error for each split.

	mean loss	mean error
cv1	0.0220567	0.0064571
cv2	0.0245465	0.0089850
cv3	0.0308356	0.0074987
cv4	0.0171546	0.0072034

i was not able to to the CV for the second detest my code kept crashing(maybe because i have no gpu) however the training and predicting went good.

5 Predictions

Prediction of each dataset will be attached with this file in csv file. comparing the prediction with the real data and MLR model.

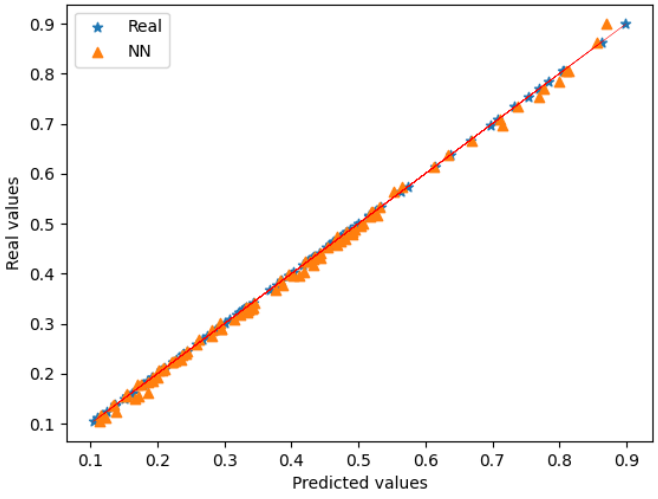

Figure 1: first data-set NN VS Real

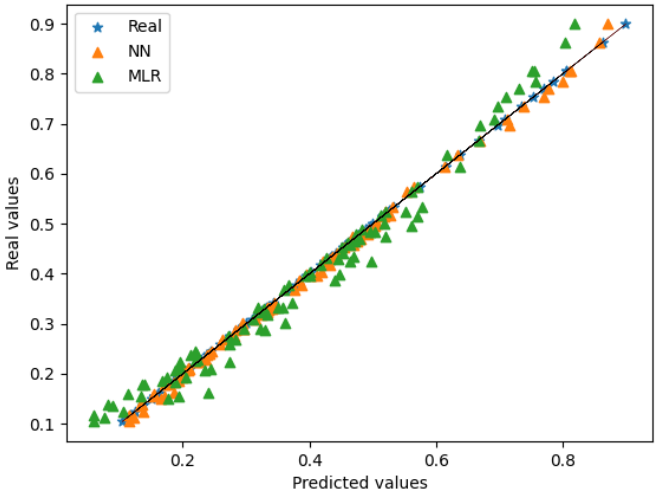

Figure 2: First data-set MLR VS NN VS Real

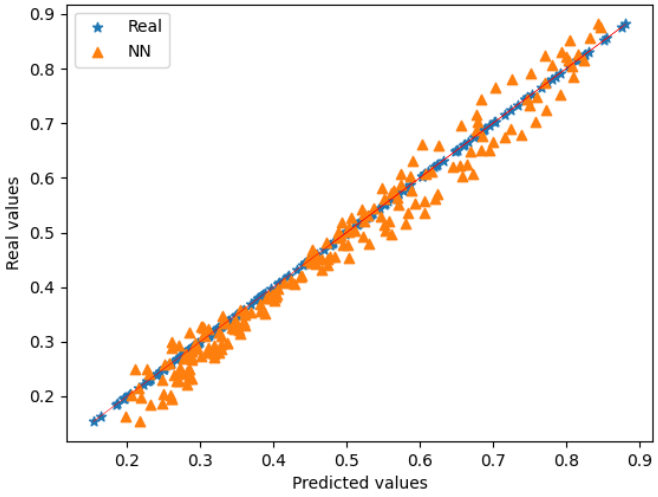

Figure 3: Second data-set NN VS Real

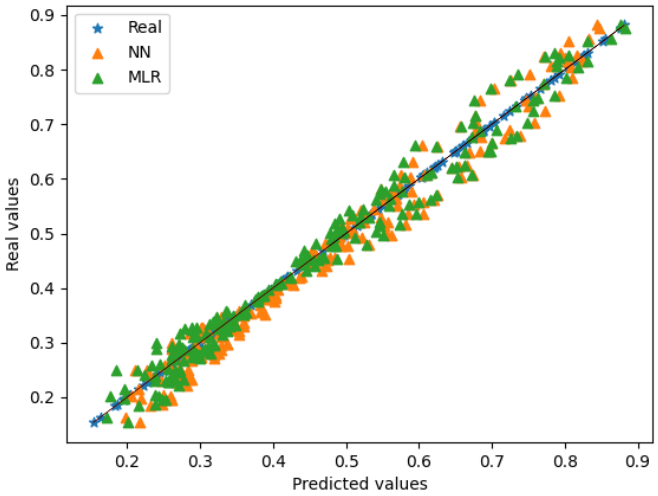

Figure 4: Second data-set MLR VS NN VS Real