University of Córdoba

Computer engineering
Computer Science Engineering Degree
4th course

Introduction to computational models

Lab Assignment 3: Radial basis functions neural networks

Valentín Gabriel Avram Aenachioei
03524931C
p92avavv@uco.es

Academic year 2022-2023
Córdoba, June 9, 2023

Contents

List of Tables

In this paper, I will set out how I have done the third lab assignment of the subject, the Radial basis functions Neural Network, explaining the Neural Network model used, its architecture and layer organization and how its algorithm works. As a conclusion, I will show, as experiments, ways to improve the error values given as a guideline, using different architectures for each dataset, and testing different experiments, analyzing their results.

The main goal of this lab assignment is the subsequent analysis, emphasising this over the implementation of the code.

1 Description of the model

The neural network model implemented is a Radially based function, explained in the theoretical classes. The RBF Neural Networks are based on local approach. Its output depends on the distance between the input vector and a vector stored in the centre.

Each RBF stores a centre as a reference, and the RBF activates depending on distance between the centre and the new pattern. This activation is controled by a radius, an hyperparameter which controlls how sensible is a RBF to its activation.

The architecture of the network is variable, it changes depending of the problem given. The general architecture is a three-layer network, an input layer, a hidden layer with the RBF function nuerons, and an output layer, which will apply a linear function in the case of regression problems, or a softmax function in the case of classification problems.

The training process of the RBF Neural Network will be divided in three steps:

1. Applicattion of a clustering algorithm which will be used to establish the centres of each RBF (input to hidden layer weights). For clustering, we will use the K-means method.

2. The radii are adjusted by the average distance to the rest of the centers (heuristic adjustment). The radium of the j-th neuron will be:
$$\sigma_j = \frac{1}{2(n_1-1)} \sum ||\mathrm{c}_j - c_i||$$

3. To adjust the hidden to output layer weights:

    - For regression problems, we will use the Moore-Penrose pseudo-inverse:
    $$\beta^T_{(n_1+1)xk} = (R^+)_{(n_1-1)xN} Y_{(Nxk)}$$
    - For classification problems, we will use a logistic regression linear model.

As hyperparameters, we will use the learning rate (η) as we did in the previous lab assignments. We will also use the radius of RBF neurons with respect to the total number of patterns in training. This hyperparameter will set the number of RBF neurons used. As other parameters, we will use the classification mode, the fairness metrics for the dataset already prepared for it, the type of regularization mode, and the number of output coulumns of the dataset.

2 Experiments

In this section, different architectures will be tested, looking to improve the error given as a guideline for each possible dataset. Once found the best architecture architecture, we will repeat the experiments, initializing the KMeans function with "k-means ++" while clustering.

We will also try different types of regularisation for classification problems, and last, we will try to run the network as a regression problem for classification problems.

2.1 Sine dataset

This dataset is composed by 120 training patterns and 41 testing patterns. Has been added some random noise to simulate a real sine function.

The objective is to improve the MSE value given as guideline:

$$MSE_{train} = 0.029687 \text{ and } MSE_{test} = 0.036366$$

The first step is to find the best possible architecture. The table 1 shows the results of the given architectures:

r value	**Nº RBFs**	MSE_{train}	MSE_{test}
0.05	*6*	*0.01381*	*0.02218*
0.15	*18*	*0.01121*	*0.36983*
0.25	*30*	*0.01033*	*3.10849*
0.5	*60*	*0.01035*	*2.079403*

Table 1: Sin dataset architectures

Once we have the best architecture, in this case, using 6 RBFs if we compare by the test MSE, We will compare the results obtained for the best architecture using the "k-means ++" initialization when clustering.

The table 2 shows the results of the experiment:

r value	Nº RBFs	MSE_{train}	MSE_{test}
0.05	*6*	*0.01380*	*0.02230*
0.15	*18*	*0.01119*	*0.37605*
0.25	*30*	*0.01036*	*2.2174*
0.5	*60*	*0.01035*	*2.37308*

Table 2: Sin dataset k-means ++ experiment

2.2 Quake dataset

This dataset is composed by 1633 training patterns and 546 testing patterns. Corresponds to a database in wich the objective is to find out the strength of an earthquake measured on the Richter scale.

The objective is to improve the MSE value given as guideline:

$$MSE_{train} = 0.030206 \text{ and } MSE_{test} = 0.027324$$

The first step is to find the best possible architecture. The table 3 shows the results of the given architectures:

r value	Nº RBFs	MSE_{train}	MSE_{test}
0.05	*81*	*0.028397*	*0.028377*
0.15	*244*	*0.025373*	*0.040940*
0.25	*408*	*0.022197*	*0.947957*
0.5	*816*	*0.018453*	*155.770134*

Table 3: Quake dataset architectures

Once we have the best architecture, in this case, using 81 RBFs if we compare by the test MSE, we will compare the results obtained for the best architecture using the "k-means ++" initialization when clustering.

The table 4 shows the results of the experiment:

r value	Nº RBFs	MSE_{train}	MSE_{test}
0.05	*81*	*0.02822*	*0.02868*
0.15	*244*	*0.02551*	*0.05018*
0.25	*408*	*0.02210*	*1.76308*
0.5	*816*	*0.01783*	*1449.328390*

Table 4: Quake dataset k-means ++ experiment

2.3 Parkinson dataset

This dataset is composed by 4406 training patterns and 1469 testing patterns. It contains a series of clinical data from patients with Parkinson's disease.

The objective is to improve the MSE value given as guideline:

$$MSE_{train} = 0.04339 \text{ and } MSE_{test} = 0.04635$$

The first step is to find the best possible architecture. The table 5 shows the results of the given architectures:

r value	Nº RBFs	MSE_{train}	MSE_{test}
0.05	*220*	*0.001670*	*0.028377*
0.15	*660*	*0.000751*	*0.001291*
0.25	*1101*	*0.000413*	*0.001052*
0.5	*2203*	*0.000134*	*0.001282*

Table 5: Parkinson dataset architectures

Once we have the best architecture, in this case, using 1101 RBFs if we compare by the test MSE, we will compare the results obtained for the best architecture using the "k-means ++" initialization when clustering.

The table 6 shows the results of the experiment:

r value	Nº RBFs	MSE_{train}	MSE_{test}
0.05	*220*	*0.001617*	*0.001899*
0.15	*660*	*0.000795*	*0.001333*
0.25	*1101*	*0.000460*	*0.001180*
0.5	*2203*	*0.000142*	*0.001692*

Table 6: Parkinson dataset k-means ++ experiment

2.4 ILDP dataset

This is a classification dataset. This dataset is composed by 405 training patterns and 174 test patterns. It contains a series of clinical data from two groups, liver and non-liver patients. This dataset is also prepared to use the fairness metrics.

The objective is to improve the CCR value given as guideline:

$CCR_{train} = 72.3457$ % and $CCR_{test} = 72.4138$ %

The first step is to find the best possible architecture. The table 7 shows the results of the given architectures:

r value	Nº RBFs	CCR_{train}	CCR_{test}
0.05	*20*	*74.02%*	*67.70%*
0.15	*60*	*77.23%*	*65.63%*
0.25	*101*	*82.81%*	*63.22%*
0.5	*202*	*87.51%*	*63.68%*

Table 7: ILDP dataset architectures

Once we have the best architecture, in this case, using 20 RBFs if we compare by the test MSE. Now, we will compare the results obtained for the best architecture using the "k-means ++" initialization when clustering.

The table 8 shows the results of the experiment:

The ildp dataset is a classification problem, we will repeat the experiments using different values for η , and for both L1 and L2 regularisation. For this

r value	**Nº RBFs**	CCR_{train}	CCR_{test}
0.05	*20*	*74.02%*	*67.70%*
0.15	*60*	*77.23%*	*65.63%*
0.25	*101*	*82.81%*	*63.22%*
0.5	*202*	*87.51%*	*63.68%*

Table 8: ILDP dataset k-means ++ experiment

experiments, we are using the best architecture, which is using 20 RBFs for this dataset.

The table 9 shows the results of the experiment:

As a last experiment, we will try to run the network for this dataset as a regression problem instead of a classification problem. The table 10 shows the results of the experiment:

The ildp dataset is already prepared to realise an algorithmical bias analysis, using *fairlearn metrics*. For this, we are going to divide the data of the dataset, class 0 for men and class 1 for women. We want to visualize the false positive ratio for both classes, and observe if there is a difference between classes, in this case, genders. As results:

In training, the false positive ratio for the Men class is around 9.21%, and for the Women class is around 11.75%.

In testing, the false positive ratio for the Men class is around 16.31%, and for the Women class is around 31.43%.

η **value**	**Regularisation**	CCR_{train}	CCR_{test}
10^{-1}	*L1*	*72.94%*	*70.80%*
10^{-1}	*L2*	*72.89%*	*70.00%*
10^{-2}	*L1*	*73.04%*	*69.31%*
10^{-2}	*L2*	*72.35%*	*70.80%*
10^{-3}	*L1*	*74.02%*	*67.82%*
10^{-3}	*L2*	*73.58%*	*68.05%*
10^{-4}	*L1*	*73.98%*	*67.70%*
10^{-4}	*L2*	*73.98%*	*67.93%*
10^{-5}	*L1*	*74.02%*	*67.70%*
10^{-5}	*L2*	*74.02%*	*67.59%*
10^{-6}	*L1*	*74.02%*	*67.70%*
10^{-6}	*L2*	*73.98%*	*67.82%*
10^{-7}	*L1*	*74.02%*	*67.70%*
10^{-7}	*L2*	*73.98%*	*67.82%*
10^{-8}	*L1*	*74.02%*	*67.70%*
10^{-8}	*L2*	*73.98%*	*67.82%*
10^{-9}	*L1*	*74.02%*	*67.70%*
10^{-9}	*L2*	*73.98%*	*67.82%*
10^{-10}	*L1*	*74.02%*	*67.70%*
10^{-10}	*L2*	*73.98%*	*67.82%*

Table 9: ILDP dataset η experiment

r value	Nº RBFs	MSE_{train}	MSE_{test}
0.05	*20*	*0.169868*	*0.185369*
0.15	*60*	*0.154626*	*0.188893*
0.25	*101*	*0.140403*	*0.201377*
0.5	*202*	*0.098872*	*0.757918*

Table 10: ILDP dataset as regression

2.5 noMNIST dataset

This dataset is composed by 900 training patterns and 300 test patterns. Is composed of a set of letters written with different typologies or symbols. The dataset is already normalized

The objective is to improve the CCR value given as guideline:

$CCR_{train} = 80.4444$ % and $CCR_{test} = 82.6667$ %

The first step is to find the best possible architecture. The table 11 shows the results of the given architectures:

r value	Nº RBFs	CCR_{train}	CCR_{test}
0.05	*45*	*87.93%*	*88.53%*
0.15	*135*	*99.93%*	*86.60%*
0.25	*225*	*100.00%*	*87.33%*
0.5	*450*	*100.00%*	*87.87%*

Table 11: noMNIST dataset architectures

Once we have the best architecture, in this case, using 45 RBFs if we compare by the test MSE. Now, we will compare the results obtained for the best architecture using the "k-means ++" initialization when clustering.

The table 12 shows the results of the experiment:

The noMNIST dataset is a classification problem, we will repeat the experiments using different values for η , and for both L1 and L2 regularisation.

r value	Nº RBFs	CCR_{train}	CCR_{test}
0.05	*45*	*87.93%*	*88.53%*
0.15	*135*	*99.93%*	*86.60%*
0.25	*225*	*100.00%*	*87.33%*
0.5	*450*	*100.00%*	*87.87%*

Table 12: noMNIST dataset k-means ++ experiment

For this experiments, we are using the best architecture, which is using 45 RBFs for this dataset.

The table 13 shows the results of the experiment:

As a last experiment, we will try to run the network for this dataset as a regression problem instead of a classification problem. The table 14 shows the results of the experiment:

2.6 Experimental conclusions

We can analyze the conclusions for each experiment as a whole, and focusing on the result of some specific datasets if its necessary.

For all the datasets, the first step was to find the best possible architecture, based on the number of RBFs of the network. For all the dataset, the network gave us the best possible results, using a $r = 0.05$ value, except for the Parkinsons dataset, which works better with bigger networks, in this case, using a $r = 0.25$ value.

For both classification datasets, L1 provide sligthly better results than L2 regularisation. For $\eta = 10^{-}5$ and lower η values, the difference on the results obtained with both regularisation types is close to zero. For the noMNIST dataset, the L2 regularization provides slightly better results, but the difference is practically imperceptible.

While comparing the initialisation of the K-Means algorithm, there is a difference between regression and classification problems. In regression, for their best architecture, the random initialisation provides better results for all 3 datasets. For classification, again for their best architecture, there is no difference between initialisation type, the results are exactly the same.

eta **value**	**Regularisation**	CCR_{train}	CCR_{test}
10^{-1}	*L1*	*80.49%*	*82.80%*
10^{-1}	*L2*	*77.67%*	*77.47%*
10^{-2}	*L1*	*86.42%*	*89.13%*
10^{-2}	*L2*	*83.36%*	*86.40%*
10^{-3}	*L1*	*87.84%*	*88.87%*
10^{-3}	*L2*	*86.22%*	*88.93%*
10^{-4}	*L1*	*87.87%*	*88.73%*
10^{-4}	*L2*	*87.42%*	*89.07%*
10^{-5}	*L1*	*87.93%*	*88.53%*
10^{-5}	*L2*	*87.82%*	*88.67%*
10^{-6}	*L1*	*87.93%*	*88.53%*
10^{-6}	*L2*	*87.96%*	*88.53%*
10^{-7}	*L1*	*87.93%*	*88.53%*
10^{-7}	*L2*	*87.98%*	*88.53%*
10^{-8}	*L1*	*87.93%*	*88.53%*
10^{-8}	*L2*	*88.00%*	*88.60%*
10^{-9}	*L1*	*87.93%*	*88.53%*
10^{-9}	*L2*	*88.02%*	*88.60%*
10^{-10}	*L1*	*87.93%*	*88.53%*
10^{-10}	*L2*	*88.00%*	*88.60%*

Table 13: noMNIST dataset η experiment

r value	**Nº RBFs**	CCR_{train}	CCR_{test}
0.05	*45*	*87.93%*	*88.53%*
0.15	*135*	*99.93%*	*86.60%*
0.25	*225*	*100.00%*	*87.33%*
0.5	*450*	*100.00%*	*87.87%*

Table 14: noMNIST dataset as regression

If we try to run the neural network as regression for a classification problem dataset, we can see that the provided results are bad, the $MSE > 0.8$ in every case.