Introduction to computational models

Lab Assignment 3: Radial basis functions neural networks

Grado en Ingeniería Informática

Escuela Politécnica Superior de Córdoba

Universidad de Córdoba

Sergio Lucena Téllez - 49505734M

i72lutes@uco.es

Índex

Description of the steps of the RBF training

In this lab assignment we are developing a RBF. The neuronal network RBF is based on local functions instead of global functions as we did in the last assignment. A function is said to be RBF if its output depends on the distance between the input vector and a centre. Each RBF stores a centre as a reference, and every time a new pattern is presented to it, the distance to this centre is calculated and will be activated if the distance is small and not activated if the distance is high. These distances will be considered small or high depending on the radius, if the radius is small, activation will only occur when the pattern is very close to the centre, and if the radius is large, the activation will take place at greater distances.

The steps of the RBF training are:

1.- First, we read the training datasets and test datasets in the .csv format. Then we proceed to transform them into matrices to use them as input and output values in our RBF.

2.- Second, we initialise the centres, for classification problems, the centroid initialisation will be random and stratified and, for regression problems, will be randomly selected. After initialising, we use KMeans class with only one centroid initialisation and a maximum of 500 iterations.

3.- Third, we calculate the radius for each neuron in a simple way, we will take half of the diameter of the average distance to the rest of centroids.

4.- We will adjust the weights of the output layer in two ways depending if it is classification problem or regression problem. If the problem is classification we will use logistic regression and if the problem is regression we will use pseudo-inverse Moore-penrose.

5.- Finally, we will print the Training MSE and Test MSE, and if the problem is classification, we will also print Training CCR and Test CCR for comparisons.

Experiments

Descriptions of datasets used

- Datasets: The algorithm will work with a training and a testing set.
    - Sine function: this dataset is composed of 120 training patterns and 41 testing patterns. It has been obtained adding some random noise to the sine function.
    - Quake dataset: this dataset is composed of 1633 training patterns and 546 testing patterns. It corresponds to a database in which the objective is to find out the strength of an earthquake. As input variables, we use the depth of focus, the latitude at which it occurs and the longitude.
    - Parkinson dataset: this dataset is composed of 4406 training patterns and 1469 testing patterns. It contains, as inputs or independent variables, a series of clinical data from patients with Parkinson's disease, including biometric measurement data from their voice. Furthermore, as output or dependent variables, it includes the motor value and the UPDRS.
    - ILPD dataset: ILPD contains 405 training patterns and 174 test patterns. The dataset was collected from northeast Andhra Pradesh, India8. The class label is used to divide patients into two groups (liver or non-liver patients). 441 records correspond to men, while 142 correspond to women. Any patient whose age exceeds 89 years appears as age "90.
    - noMNIST dataset: The dataset is composed of 900 training patterns and 300 test patterns. It includes a set of letters written with different typologies or symbols. They are adjusted to a squared grid of 28 × 28 pixels. The images are in grey scale in the interval [−1,0; +1,0]10.Each of the pixels is an input variable (with a total of 28 × 28 = 784 input variables) and the class corresponds to a written letter (a, b, c, d, e y f , with a total of 6 classes).

Descriptions of the values parameters considered

In the experiments, we will realise four different configurations to see how values improve the RBF and each configuration has different parameters considered.

- In the first configuration, for all datasets, we consider a number of hidden neurons equal to the 5%, 15%, 25% and 50% of the total number of patterns of the dataset. For regression, we will use L1 regularisation and eta value 1e-2, and for classification, we will use L1 and eta value 1e-1.
- In the second configuration, for classification problems, with the best architecture from the last configuration, we will change the eta values from 1 to 1e-10, along with the two types of regularisation L2 and L1. In this configuration, we will use a ratio RBF of 10%.
- In the third configuration, for all datasets, we compare the results obtained using the different algorithm of *sklearn.cluster.KMeans*. In this configuration, we will use the best architecture and the best configuration for the logistic regression.
- In the fourth configuration, for classification datasets, we will consider the datasets as regression datasets. The values considered are ratio RBF of 10%, L1 regularisation and eta value of 1e -2.

Results

First configuration

In this configuration, for all datasets, we consider a number of hidden neurons equal to the 5%, 15%, 25% and 50% of the total number of patterns of the dataset. For regression, we will use L1 regularisation and eta value 1e-2, and for classification, we will use L1 and eta value 1e-1.

Sine

Table 1

Values of the Training MSE and Test MSE using different ratios RBF in the sine dataset.

RBF	Training MSE	Test MSE
0.05	0.013800 +- 0.000108	0.022301 +- 0.000292
0.15	0.011193 +- 0.000017	0.376057 +- 0.021240
0.25	0.010361 +- 0.000002	2.200944 +- 0.299373
0.5	0.010359 +- 0.000003	2.367383 +- 0.370299

Quake

Table 2

Values of the Training MSE and Test MSE using different ratios RBF in the quake dataset.

RBF	Training MSE	Test MSE
0.05	0.028221 +- 0.000047	0.028686 +- 0.000103
0.15	0.025509 +- 0.000064	0.049683 +- 0.003434
0.25	0.022101 +- 0.000099	1.763458 +- 0.424248
0.5	0.017839 +- 0.000046	1346.316325 +- 720.222555

Parkinson

Table 3

Values of the Training MSE and Test MSE using different ratios RBF in the Parkinson dataset.

RBF	Training MSE	Test MSE
0.05	0.022140 +- 0.000067	0.025374 +- 0.000169
0.15	0.014995 +- 0.000085	0.021660 +- 0.000584
0.25	0.011313 +- 0.000086	0.024062 +- 0.002811
0.5	0.005571 +- 0.000044	0.112474 +- 0.018874

iLDP

Table 4

Values of the Training MSE, Test MSE, Training CCR and Test CCR using different ratios RBF in the ILDP dataset.

RBF	Training MSE	Test MSE	Training CCR	Test CCR
0.05	0.163233 +- 0.000957	0.163233 +- 0.000957	73.43% +- 0.92%	69.20% +- 1.07%
0.15	0.144008 +- 0.002410	0.194082 +- 0.005762	76.74% +- 0.61%	69.54% +- 1.21%
0.25	0.130400 +- 0.000923	0.214510 +- 0.005466	78.86% +- 0.46%	66.55% +- 1.80%
0.5	0.115953 +- 0.000761	0.226564 +- 0.002309	80.99% +- 0.54%	64.94% +- 1.15%

noMNIST

Table 5

Values of the Training MSE, Test MSE, Training CCR and Test CCR using different ratios RBF in the noMNIST dataset.

RBF	Training MSE	Test MSE	Training CCR	Test CCR
0.05	0.029709 +- 0.000607	0.028893 +- 0.000492	87.78% +- 0.45%	88.33% +- 0.82%
0.15	0.008265 +- 0.000473	0.027938 +- 0.002418	97.73% +- 0.34%	89.60% +- 1.08%
0.25	0.002297 +- 0.000115	0.002297 +- 0.000115	99.78% +- 0.07%	87.53% +- 0.88%
0.5	0.000635 +- 0.000035	0.028867 +- 0.001283	100.00% +- 0.00%	89.93% +- 0.57%

In general, we can observe how increasing the number of hidden neurons makes the neural network perform worse. The reason behind this is because we are overtraining the neuronal network and we are adapting it to the dataset instead of the problem, so in the Test MSE and Test CCR we obtain worse results. We have an exception which is the Parkinsons datasets. In this dataset we have a high number of patterns and they are different between each other, so increasing the number of neurons trains the network better, without reaching the overtraining.

Second configuration

In this configuration, for classification problems, with the best architecture from the last configuration, we will change the eta values from 1 to 1e-10, along with the two types of regularisation L2 and L1. In this configuration, we will use a ratio RBF of 10%.

ILDP

Table 6

Values of the Training MSE, Test MSE, Training CCR and Test CCR using different eta values and L1 regularisation in the ILDP dataset.

η	Training MSE	Test MSE	Training CCR	Test CCR
1	0.170082 +- 0.000361	0.183159 +- 0.000909	72.30% +- 0.50%	71.15% +- 0.43%
1e-1	0.158375 +- 0.000819	0.181106 +- 0.000785	75.21% +- 0.77%	69.54% +- 1.09%
1e-2	0.144008 +- 0.002410	0.194082 +- 0.005762	76.74% +- 0.61%	69.54% +- 1.21%
1e-3	0.151270 +- 0.001650	0.195996 +- 0.005138	75.70% +- 0.40%	67.24% +- 1.50%
1e-4	0.151058 +- 0.001841	0.196986 +- 0.005099	75.56% +- 0.60%	67.47% +- 1.29%
1e-5	0.150952 +- 0.001928	0.197440 +- 0.004921	75.65% +- 0.55%	67.36% +- 1.68%
1e-6	0.150948 +- 0.001927	0.197477 +- 0.004925	75.65% +- 0.55%	67.36% +- 1.68%
1e-7	0.150948 +- 0.001927	0.197478 +- 0.004925	75.65% +- 0.55%	67.36% +- 1.68%
1e-8	0.150948 +- 0.001927	0.197478 +- 0.004925	75.65% +- 0.55%	67.36% +- 1.68%
1e-9	0.150948 +- 0.001927	0.197478 +- 0.004925	75.65% +- 0.55%	67.36% +- 1.68%
1e-10	0.150948 +- 0.001927	0.197478 +- 0.004925	75.65% +- 0.55%	67.36% +- 1.68%

Table 7

Values of the Training MSE, Test MSE, Training CCR and Test CCR using different eta values and L2 regularisation in the ILDP dataset.

η	Training MSE	Test MSE	Training CCR	Test CCR
1	0.170034 +- 0.000363	0.179722 +- 0.000180	71.85% +- 0.22%	71.03% +- 0.28%
1e-1	0.162851 +- 0.000429	0.179679 +- 0.000497	73.38% +- 0.57%	72.99% +- 1.31%
1e-2	0.157845 +- 0.000565	0.182769 +- 0.001319	75.51% +- 0.61%	69.66% +- 1.11%
1e-3	0.154159 +- 0.001120	0.187749 +- 0.003925	76.00% +- 0.53%	69.20% +- 2.01%
1e-4	0.151314 +- 0.002074	0.194060 +- 0.006074	75.85% +- 0.82%	67.70% +- 1.83%
1e-5	0.150360 +- 0.002290	0.199069 +- 0.005922	75.56% +- 0.56%	67.59% +- 1.73%
1e-6	0.150234 +- 0.002224	0.201704 +- 0.005597	75.26% +- 0.69%	66.78% +- 0.99%
1e-7	0.150230 +- 0.002233	0.202325 +- 0.005667	75.21% +- 0.69%	66.67% +- 1.03%
1e-8	0.150233 +- 0.002219	0.201642 +- 0.006696	75.31% +- 0.66%	66.67% +- 1.15%
1e-9	0.150265 +- 0.002238	0.201369 +- 0.005560	75.21% +- 0.71%	67.13% +- 1.23%
1e-10	0.150226 +- 0.002219	0.201966 +- 0.006738	75.31% +- 0.68%	66.67% +- 1.36%

noMNIST

Table 8

Values of the Training MSE, Test MSE, Training CCR and Test CCR using different eta values and L1 regularisation in the noMNIST dataset.

η	Training MSE	Test MSE	Training CCR	Test CCR
1	0.046967 +- 0.000425	0.044060 +- 0.000314	83.87% +- 0.52%	87.67% +- 0.56%
1e-1	0.026650 +- 0.000312	0.026671 +- 0.000615	89.84% +- 0.41%	90.33% +- 0.30%
1e-2	0.017565 +- 0.000471	0.027529 +- 0.001808	93.27% +- 0.48%	89.67% +- 0.60%
1e-3	0.013026 +- 0.001128	0.034028 +- 0.003271	95.58% +- 0.74%	87.40% +- 0.83%
1e-4	0.011976 +- 0.001132	0.036576 +- 0.003398	96.13% +- 0.61%	86.53% +- 0.83%
1e-5	0.011889 +- 0.001156	0.037038 +- 0.003599	96.16% +- 0.60%	86.67% +- 0.87%
1e-6	0.011831 +- 0.001162	0.011831 +- 0.001162	96.16% +- 0.60%	86.60% +- 0.57%
1e-7	0.011830 +- 0.001162	0.037060 +- 0.003669	96.16% +- 0.60%	86.60% +- 0.57%
1e-8	0.011830 +- 0.001162	0.037060 +- 0.003669	96.16% +- 0.60%	86.60% +- 0.57%
1e-9	0.011830 +- 0.001162	0.037060 +- 0.003669	96.16% +- 0.60%	86.60% +- 0.57%
1e-10	0.011830 +- 0.001162	0.037060 +- 0.003669	96.16% +- 0.60%	86.60% +- 0.57%

Table 9

Values of the Training MSE, Test MSE, Training CCR and Test CCR using different eta values and L2 regularisation in the noMNIST dataset.

η	Training MSE	Test MSE	Training CCR	Test CCR
1	0.053236 +- 0.000293	0.051622 +- 0.000448	83.22% +- 0.25%	85.27% +- 0.77%
1e-1	0.034945 +- 0.000276	0.032655 +- 0.000325	87.13% +- 0.20%	90.27% +- 0.25%
1e-2	0.025894 +- 0.000232	0.025901 +- 0.000226	90.02% +- 0.29%	90.60% +- 0.25%
1e-3	0.019614 +- 0.000248	0.025923 +- 0.001000	92.51% +- 0.46%	90.13% +- 0.40%
1e-4	0.015329 +- 0.000723	0.029489 +- 0.002489	94.09% +- 0.65%	89.00% +- 0.52%
1e-5	0.012738 +- 0.001107	0.035231 +- 0.003244	95.73% +- 0.72%	87.33% +- 0.89%
1e-6	0.011279 +- 0.001397	0.038896 +- 0.003327	96.33% +- 0.65%	85.73% +- 0.71%
1e-7	0.010852 +- 0.001667	0.039597 +- 0.003308	96.64% +- 0.78%	85.73% +- 0.49%
1e-8	0.010828 +- 0.001455	0.039457 +- 0.002956	96.64% +- 0.69%	85.80% +- 0.50%
1e-9	0.010681 +- 0.001655	0.039855 +- 0.003315	96.60% +- 0.78%	85.67% +- 0.76%
1e-10	0.010748 +- 0.001736	0.039674 +- 0.003379	96.71% +- 0.88%	85.60% +- 0.65%

With this configuration we can notice how with a really small eta value the Test CCR are pretty similar. It is because we are learning so little that it won't change the weights in a noticeable way. In the case of eta value 1, we obtain worse results because the eta value is too high and we are overtraining the neural network.

Third configuration

In this configuration, for all datasets, we compare the results obtained using the different algorithm of *sklearn.cluster.KMeans*. We will use the best architecture and the best configuration for the logistic regression.

Sine

Table 9

Values of the Training MSE and Test MSE comparing the different initialisation in the sine dataset.

KMeans	Training MSE	Test MSE
random	0.013817 +- 0.000189	0.022182 +- 0.000245
k-means++	0.013800 +- 0.000108	0.022301 +- 0.000292

Quake

Table 10

Values of the Training MSE and Test MSE comparing the different initialisation in the quake dataset.

KMeans	Training MSE	Test MSE
random	0.028397 +- 0.000056	0.028377 +- 0.000262
k-means++	0.028221 +- 0.000047	0.028686 +- 0.000103

Parkinson

Table 11

Values of the Training MSE and Test MSE comparing the different initialisation in the Parkinson dataset.

KMeans	Training MSE	Test MSE
random	0.013982 +- 0.000276	0.019696 +- 0.000442
k-means++	0.014995 +- 0.000085	0.021660 +- 0.000584

ILDP

Table 12

Values of the Training MSE, Test MSE, Training CCR and Test CCR comparing the different initialisation in the ILDP dataset.

KMeans	Training MSE	Test MSE	Training CCR	Test CCR
random	0.159361 +- 0.000742	0.181354 +- 0.000959	74.81% +- 0.88%	70.34% +- 0.86%
k-means++	0.162851 +- 0.000429	0.179679 +- 0.000497	73.38% +- 0.57%	72.99% +- 1.31%

noMNIST

Table 13

Values of the Training MSE, Test MSE, Training CCR and Test CCR comparing the different initialisation in the noMNIST dataset.

KMeans	Training MSE	Test MSE	Training CCR	Test CCR
random	0.025886 +- 0.000775	0.026072 +- 0.001578	90.22% +- 0.66%	91.13% +- 0.98%
k-means++	0.025894 +- 0.000232	0.025901 +- 0.000226	90.02% +- 0.29%	90.60% +- 0.25%

Normally, k-means++ will perform better than random, but it doesn’t mean k-means++ initialisation is always the best. K-means++ try to select initial cluster centroids using sampling based on an empirical probability distribution of the points’ contribution to the overall inertia. This technique speeds up convergence, and is theoretically proven to be optimal.

In our results, we can perceive k-means++ give us better results (ILDP dataset), same results (Quake datasets) and worse results (noMNIST dataset).

Fourth configuration

In this configuration, for classification datasets, we will consider the datasets as regression datasets. The values considered are ratio RBF of 10%, L1 regularisation and eta value of 1e -2.

ILDP

Table 14

Values of the Training MSE, Test MSE, Training CCR and Test CCR if it is considered a regression problem in the ILDP dataset.

Considered as	Training MSE	Test MSE	Training CCR	Test CCR
Regression	0.157274 +- 0.001710	0.190624 +- 0.002509	75.90% +- 0.40%	68.62% +- 1.84%
Classification	0.162851 +- 0.000429	0.179679 +- 0.000497	73.38% +- 0.57%	72.99% +- 1.31%

noMNIST

Table 15

Values of the Training MSE, Test MSE, Training CCR and Test CCR if it is considered a regression problem in the noMNIST dataset.

Considered as	Training MSE	Test MSE	Training CCR	Test CCR
Regression	0.624652 +- 0.021231	0.756030 +- 0.033329	61.00% +- 1.40%	59.00% +- 3.06%
Classification	0.025886 +- 0.000775	0.026072 +- 0.001578	90.22% +- 0.66%	91.13% +- 0.98%

As we can see, regression gives us a really bad result in noMNISTT and a similar result in ILDP. The reason for that difference is because the noMNIST dataset has more output classes and regression assumes a distance between each of the classes and an order between these, so regression is inconsistent for this kind of dataset. In the ILDP, there are only two classes and the difference between them is not that high, so regression obtains similar results with approximation of the CCR values to the closer integer.

Now, at the end of the lab report we are going to talk about fairness because it is not implemented in the code, but it is important to, at least, talk about the false negative rate focused on the ildp dataset. This dataset presents a class imbalance problem, as there are 167 patients with liver disease and 416 healthy patients. Moreover, this dataset has 441 men and 142 women. For that reason, statistically the majority of the liver disease will be men, and the model tends to predict that men have liver disease and women do not. To solve this problem we can use the false negative rate which is the percentage of actual positives incorrectly predicted as negative. In other words, it can be interpreted as the percentage of people who have wrongfully not benefited from the model, in our case, it would be the percentage of people who should have but were not diagnosed with liver disease.

Bibliography

https://www.districtdatalabs.com/fairness-and-bias-in-algorithms

https://towardsdatascience.com/analysing-fairness-in-machine-learning-with-python-96a9ab0d0705

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

https://stackoverflow.com/questions/3518778/how-do-i-read-csv-data-into-a-record-array-in-numpy