INSTITUTO SUPERIOR TÉCNICO

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Merit Prize Report

LEIC-T

David Toma **ist1106160**

Tiago Costa **ist1106897**

Grupo **30**

29/11/2024

Background

We start by providing a brief theoretical background as a foundation for the decisions and conclusions taken on the implemented RBF network.

The goal of an RBF neural network is to provide a solution to the Multivariate Interpolation problem using radial basis functions. The architecture is a 3-layer neural network. The hidden layer contains radial basis functions. A connection between input and hidden nodes yields the value of the radial basis function applied to that node.

The association between the first two layers can be represented as a matrix A where the number of rows is the number of input instances (nodes), and the number columns is the number of radial basis functions in the hidden layer. The RBF type used in this implementation is the Gaussian function

$$\phi(||x - c||) = \exp\left(-\frac{||x - c||^2}{2s^2}\right)$$

where c is the basis (center) and s the spread [1].

Following the optimal value of clusters from 2), the hidden layer of the network will enclose two radial basis functions whose centers correspond to the centers of the two clusters. This means matrix A will have the following shape.

$$A = \begin{bmatrix} \phi(||x_1 - c_1||) & \phi(||x_1 - c_2||) \\ \phi(||x_2 - c_1||) & \phi(||x_2 - c_2||) \\ \vdots & \vdots \\ \phi(||x_m - c_1||) & \phi(||x_m - c_2||) \end{bmatrix}$$

The connections between the hidden and the output layer are the weights that the network needs to learn. The neural network can be represented as the following equation

$$Aw = t$$

With t being a vector containing all m labels of the input instances. The weights of the output layer (one for each RBF) can be calculated with

$$w = A^{-1} t$$

and since A is not square

$$w = A^{+} t$$

The problem is now reduced to simple linear optimization problem solved by the pseudo-inverse technique. [1]

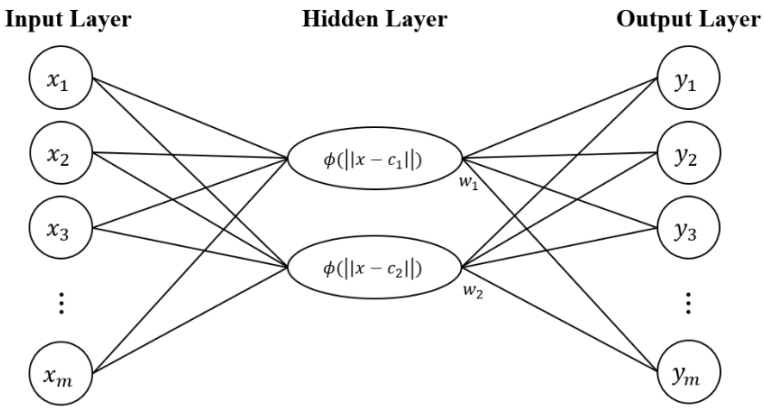


Figure 1: Diagram of the RBF network implemented (c_1 and c_2 represent the cluster centers while w_1 and w_2 are the output weights assigned to each RBF)

1. Spread value heuristic was obtained by computing the mean value of the distance between the two cluster centers.

There was an additional step implemented proposed on [2] where a third learning step was added.

After performing EM clustering to learn the centers of the RBFs in an unsupervised way and solving the linear pseudo-inverse problem to yield the weights, backpropagation is used to fine-tune the weights and the centers to find a configuration between both that improves the model's predictive accuracy. The goal here is to provide some supervised background to adjust better the cluster centers.
The learning rules derived from the network structure and using sum-of-squares error function are the following:

$$w_j = w_j + \eta \cdot \sum_{i=1}^{m} \phi_j(x_i) \cdot (t_i - y_i)$$

for the weights, and

$$c_j = c_j + \eta \cdot \sum_{i=1}^{m} \phi_j(x_i) \cdot \frac{x_i - c_j}{s^2} \cdot \sum_{k=1}^{2} w_k.(t_i - y_i)$$

for the centres, where η is the learning rate.

Dataset Structural Analysis

We get a strong inverse correlation for the number of clusters and respective silhouettes, with the two-cluster configuration getting the highest silhouette score (1).

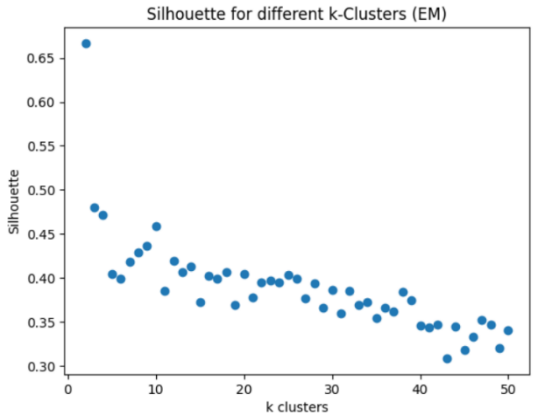
We plotted the relationships between the number of clusters, silhouette and accuracy of logistic regression over the test set mapped with cluster probabilities (2, 3).

From (2) we can observe that a lower number of clusters leads to higher accuracy and the tendency is for it to decrease with the increase of clusters (Strong inverse correlation as can be deduced from the Pearson r value).

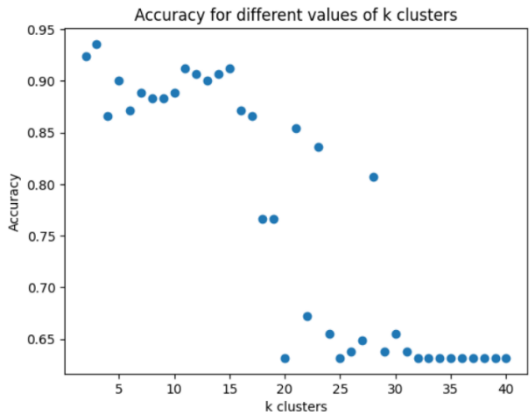
We can also observe (3) that the highest accuracy values were obtained for the clustering configurations with silhouettes above 0.40.
When silhouette is below 0.40 the model performs worse.
Values with silhouette around 0.40 are very inconstant in terms of accuracy, with it balancing between high and low values.

From (1) we can deduce that the points with silhouette around 0.40 with high accuracies have a lower number of clusters than the points with similar silhouette value but with lower accuracy. We can prove it by plotting a 3-D graph considering all three variables.

1.



2.



3.

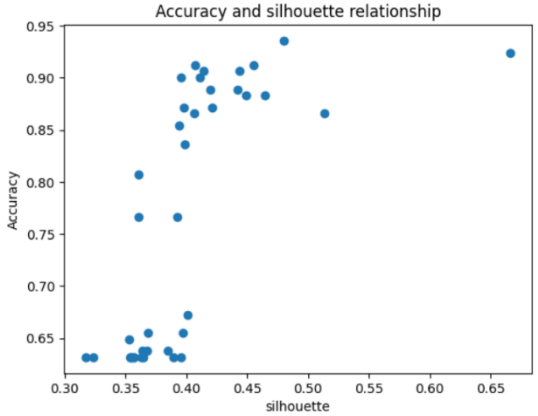


Figure 2: 2-D Relationships between number of clusters, silhouette and accuracy

By not considering the number of clusters on each point we get some inconsistencies, making accuracy and silhouette only moderately directly correlated.

The conclusion that can be made from these relationships is that accuracy is higher for clustering configurations with higher silhouette, where the number of clusters is lower.
The higher silhouette tells that the points lie better in the regions identified by the EM algorithm, making these patterns more likely to represent better the hidden structure of the data. The lower number of clusters is the final tiebreaker, telling us that the real hidden structure of the data is represented by a smaller number of those well-defined regions.

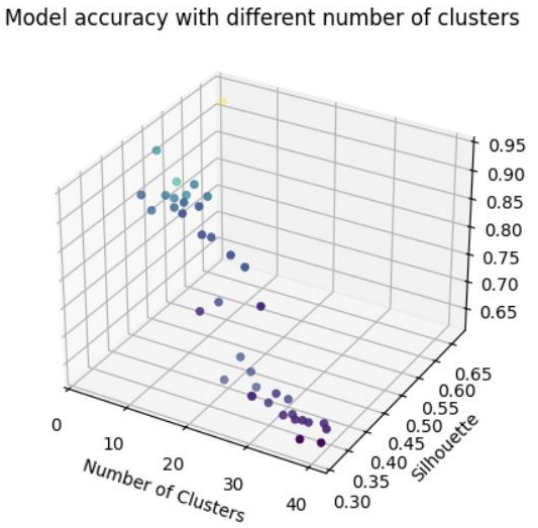


Figure 3: 3-D Relationship between number of clusters, silhouette and accuracy

RBF Neural Network

The RBF network implemented scored worse than Logistic Regression. We start by gaining some insights behind the way this model sees the data, to then formulate our hypothesis and conclusions.

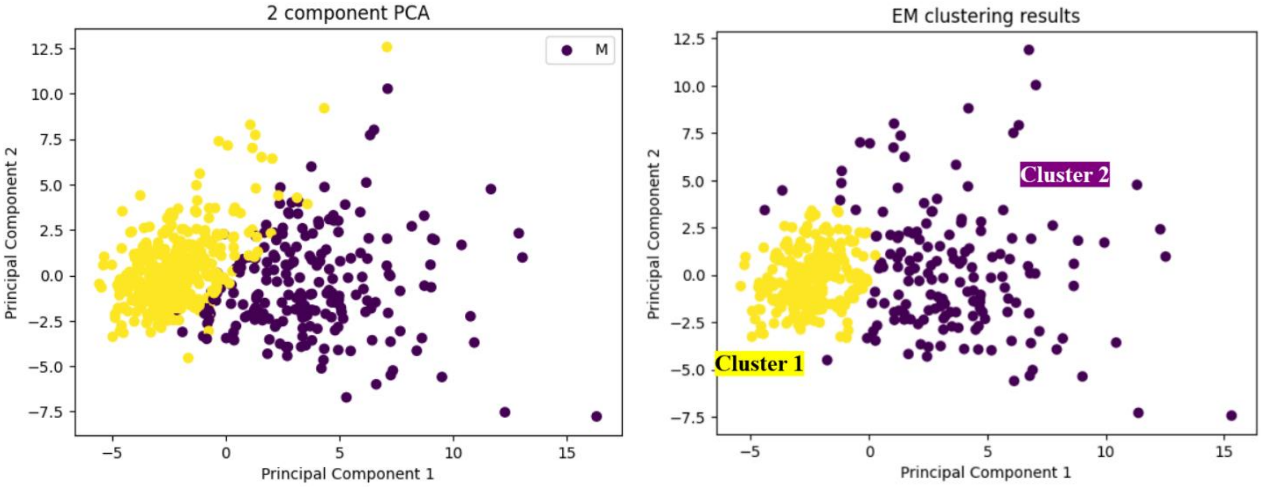


Figure 4: Class distribution of the dataset mapped in the two Principal Components (left) and EM clustering results (right)

After performing PCA with 2 components, we noticed a generally good division between label classes. The “Benign” class points (yellow) are very close to each other with only a few values being far away. This big aggregation of points, if seen as a cluster, could potentially lead to good results using an RBF network.

When performing EM clustering with optimal value for k (k = 2), we obtain the clusters on the right. The big aggregation of points from class “Benign” falls in one cluster. The points of class “Benign” far from that aggregation fall on the other cluster. By comparing the two previous plots we see that the regions EM identified have some relationship with the actual classes.

The values of the two RBFs for each data point (hidden layer) are directly correlated with the distance of that data point to each cluster center (Gaussian function has its maximum at the center). In terms of the model, this value can be translated as the influence each cluster will have on the final classification of that point. The weights of

each RBF will tell us the significance each one of these regions (clusters) has in the general classification performed by the model.

Cluster 1 is much denser than Cluster 2. A point to fall inside this Cluster 1 must be much closer to its center than it must be from the center of Cluster 2 to be classified in that cluster. It means that Cluster 2 is more tolerant to higher distances from its center than Cluster 1. By being so, only the values close enough to Cluster 1 will have that cluster playing a significant part in classification.

This assumption could be the reason that compromised the accuracy of the RBF model when compared to Logistic Regression, since the points with target "Benign" that are outside Cluster 1 will be less influenced by that cluster and more by the other cluster, making them more prone to bad classification.

We can fundament this hypothesis by plotting a confusion graph.

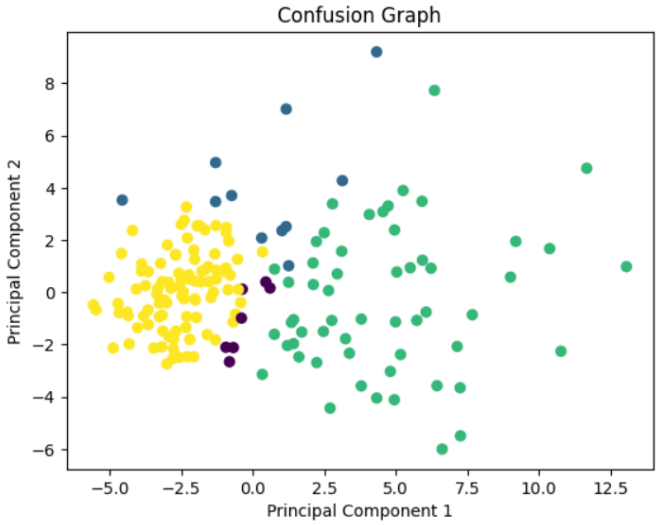


Figure 5: (Confusion) Graph with the result of the classification. Yellow – True Benign; Green – True Malign; Purple – False Benign; Blue – False Malign

As we can observe, the points from class "Benign" wrongly classified are those away from the center of Cluster 1, thus proving our assumption.

The rest of the data points wrongly classified are points of class "Malign" on the overlapping region between classes. These points are at a distance close enough from Cluster 1 to be more influenced by it than by Cluster 2. Since most of the points in Cluster 1 belong to class "Benign", the weights learned by the model will more likely predict as "Benign" the data points that are more influenced by this cluster.

This cluster configuration with Cluster 1 being much denser than cluster 2 can be seen as the biggest limitation of the model.

The backpropagation implemented after calculating the weights comes to counter the limitation provided by the initial clustering. Now we give the algorithm behind center selection some supervised background to adjust on the centers and weights to potentially increase the classification accuracy.

The backpropagation learning phase added comes with the assumption that the model is in a good starting point and only needs to be fine-tuned. This assumption can overcome the local minimum problem on purely backpropagating networks where initial weights are randomly set.

The idea of using backpropagation in this implementation comes as a hypothesis that it's possible to improve the accuracy of the model. From our results we see a slight improvement in some cases when using backpropagation. This improvement could be bigger with the right learning rate setting. Exploring techniques to find the right learning rate can be a subject to subsequent works.

Another limitation that played an important role in compromising the performance of the model when compared to logistic regression is the use of the same spread value on both RBFs (the two clusters have different densities). Additionally, backpropagation for computing the spreads could be implemented if each RBF had its own spread [2].

We finish this analysis with the hypothesis that a bigger hidden layer could capture the data characteristics in a more complex way and overcome the underfitting of the two-RBF solution where the data structure was simply represented by the big aggregation of class "Benign" points and "the rest of the dataset".

We fundament this hypothesis with the results obtained from performing logistic regression with clustering probabilities, where there were other clustering configurations with $k > 2$ and lower silhouette scores that also got very high accuracy scores (accuracy for $k = 3$ was even higher than for $k = 3$).

References

[1] D. S. Broomhead and D. Lowe, "Multivariable function interpolation and adaptive networks," Complex Systems, Vol. 2, pp. 321–355 (1988).

[2] F. Schwenker, H. A. Kestler, and G. Palm, "Three learning phases for radial-basis-function networks," *Neural Networks* 14, 439–458 (2001).