Artificial Neural Networks and Deep Architectures, DD2437

Short report on lab assignment 2
Radial basis functions, competitive learning and self-organisation maps

Balint Kovacs (balintk@kth.se), Victor Castillo (vcas@kth.se),
Jun Zhang (junzha@kth.se)

February 7th, 2019

1 Main objectives and scope of the assignment

Our major goals in the assignment were

- know how to build the structure and perform training of an RBF network for either classification or regression purposes
- be able to comparatively analyze different methods for initializing the structure and learning the weights in an RBF network
- know the concept of vector quantization and learn how to use it in NN context
- be able to recognize and implement different components in the SOM algorithm
- be able to discuss the role of the neighborhood and analyze its effect on the self-organization in SOMs
- know how SOM-networks can be used to fold high-dimensional spaces and cluster data

2 Methods

Python, numpy, matplotlib

3 Results and Discussion - Part I: RBF networks and Competitive Learning

3.1 Function approximation with RBF networks

Results obtained by varying the number of units to get the absolute residual error below 0.01 and 0.001.

- Sin(2x) function: 4, and 6 nodes respectively

- Square(2x) function: 32 nodes, while 0.001 residual was not achieved changing the number of nodes

We could reduce the error in the Square function by transforming the results, gating intermediate values to the closest value (-1 or 1), thresholding at 0 and making the model a binary classifier.

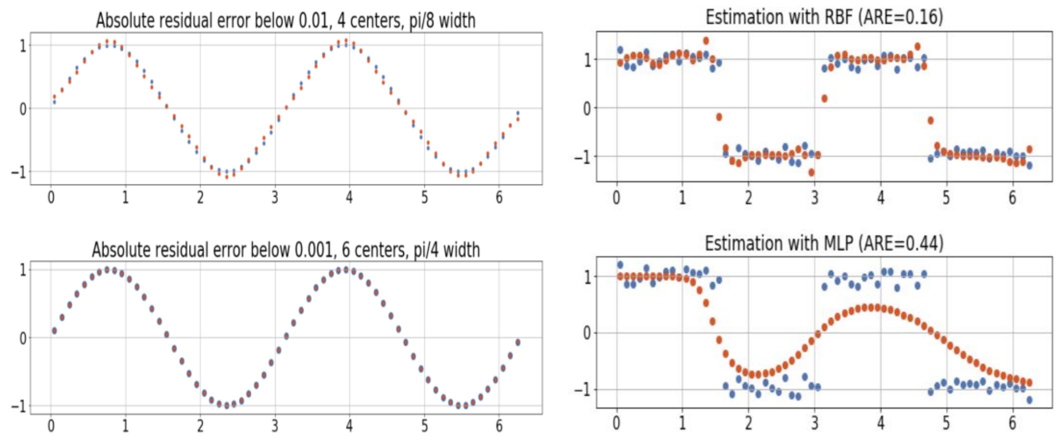

Figure 1: Comparing error estimations for Sin(2x), Square(2x), using RBF and MLP

As for the width parameter, when it is too small, it can be local to the current point (as can be seen on the last plot above on the left side of **Figure 2**), or when it is too big, it does not have enough effect on the points nearby. The number of units needs to be chosen according to the complexity of the problem. The number of local minima and maxima is sufficient for the sine function for example.
The learning rate is not used in the least squares algorithm. In the case of the delta rule, if the learning rate is too large, then the outputs will be too big and it is possible that it will never achieve convergencee. If the learning rate is too small, the outputs stay close to zero.
Reproducing the same results as the least square estimate on the original data is a matter of finding the right parameters for learning rate and the number of epochs, since the least square estimate has the advantage of finding the exact solution. But with the right parameters, it is possible.

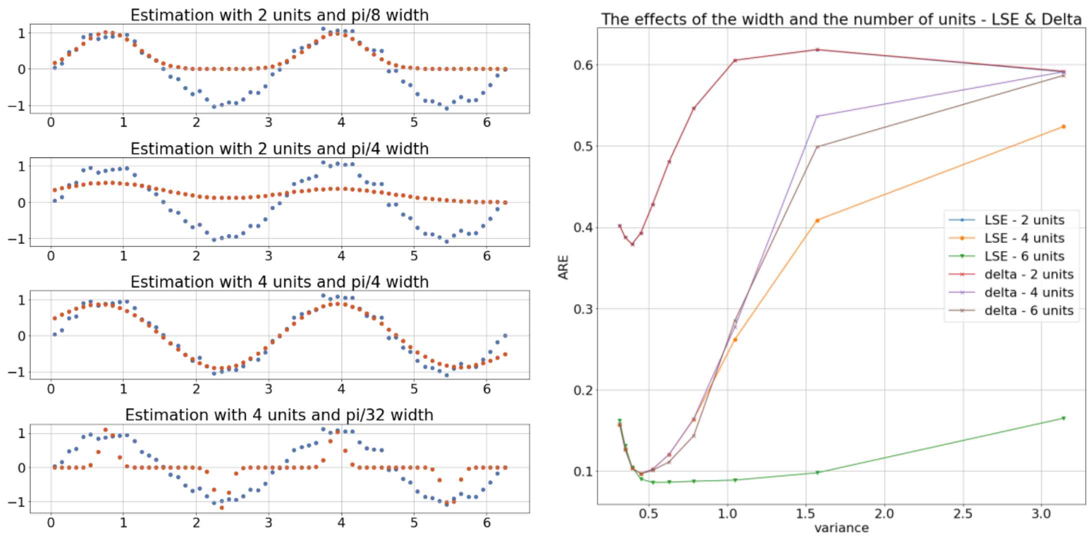

Figure 2: Comparing the performance of RBF networks with different structures

Randomly positioning the centroids has a negative effect on the performance. RBF networks are sensitive for the choice of centroids, so they have to be chosen either by hand, or preferably a learning algorithm.

When comparing the RBF network with MLP, the former performs significantly better on both data sets. In theory, estimating the square function should be easier to approximate with sigmoids, but the algorithm probably stucks at local minimas.

3.2 Competitive learning for RBF unit initialization

Since the training samples are uniformly distributed, the centroids cannot estimate any clusters. They will be close to distributed normally. The generalization performance is worse than picking them manually, but better than random.

Dead units can be avoided by updating more than a single node, in the proportion of the distance from the winner node. The distance function can be a gaussian for example. One different approach would be re-initializing the location of nodes that did not win in the last 'n'.

On the ballistic dataset, we reached the best performance on both the training and the test set by using 100 nodes. However, this is largely overfitting the noisy samples we have. For the best generalization performance, around 10 nodes should be enough(**Figure 3**).

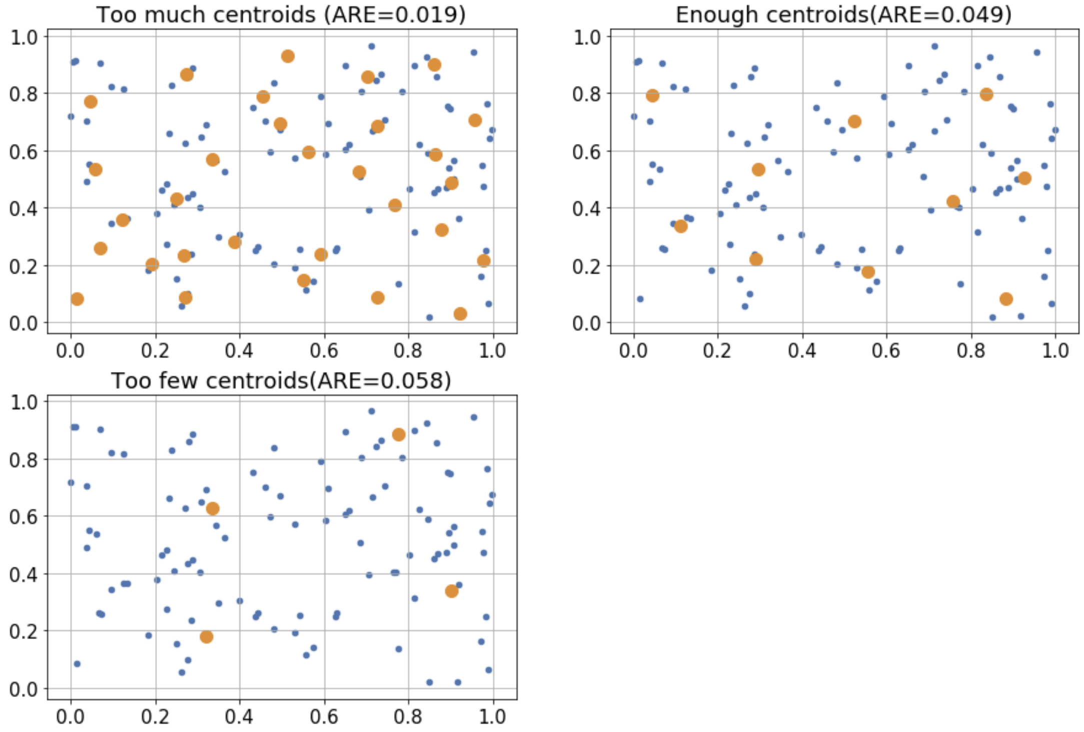

Figure 3: Comparing different number of centroids choices

4 Results and discussion - Part II: Self-organising maps

4.1 Topological ordering of animal species

The transformation from the 32 × 84 dimensional input to a 100-dimensional space, with neighboring nodes being closely related in this mapping. The animals that have more similarity (taxonomically in a way, but not necessarily using the correct characteristics from the field) are assigned close to each other. The insects, mammals, birds are grouped with others of their kind.

```
["'camel'" "'giraffe'" "'horse'" "'pig'" "'antelop'" "'kangaroo'"
 "'skunk'" "'dog'" "'hyena'" "'bear'" "'rabbit'" "'rat'" "'ape'" "'cat'"
 "'lion'" "'bat'" "'elephant'" "'walrus'" "'crocodile'" "'seaturtle'"
 "'frog'" "'penguin'" "'ostrich'" "'duck'" "'pelican'" "'spider'"
 "'beetle'" "'grasshopper'" "'dragonfly'" "'butterfly'" "'housefly'"
 "'moskito'"]
```

The neighborhood is initially defined to be the 50 nearest nodes to the winning node, but are reduced as the epochs iterate in this manner:

```
#  update rule based on epochs
if i < 4:
    neighbourhood -= 10
elif i<9:
    neighbourhood = 5
elif i<14:
    neighbourhood = 3
else:
    neighbourhood = 1
```

4.2 Cyclic tour

In this case, by mapping the cities to the unit nodes, we are able to obtain to the optimal route (shortest distance) between the two nearest points. (**Figure 4**). The approximation is not perfect, but close to the optimal one: if two input patterns are the nearest to the same node, then their position is chosen at random.

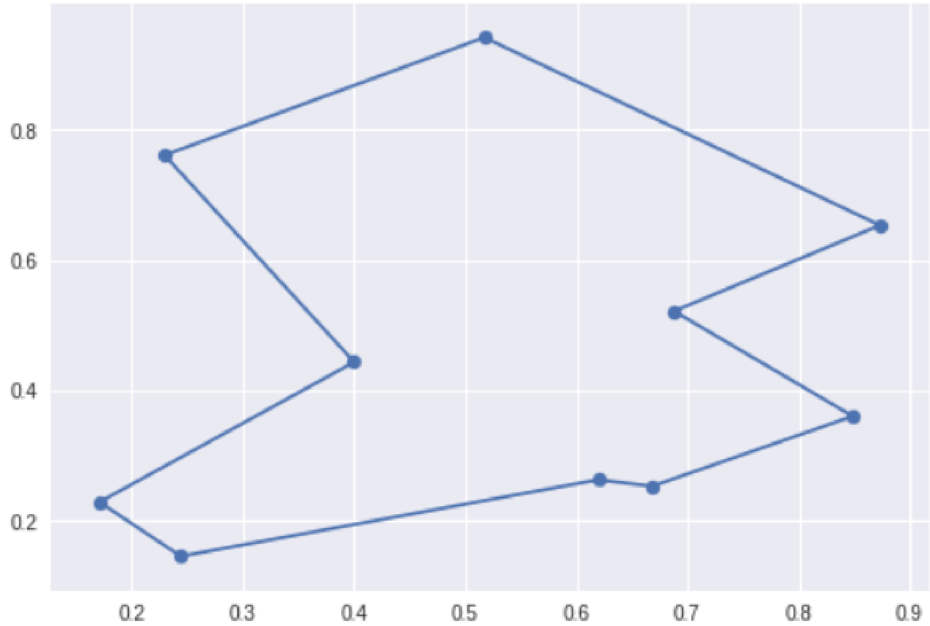

Figure 4: Optimal route between two cities

4.3 Clustering with SOM

From the generated clustering, one could conclude several things, like: The MP without a party is closest to the leftists, the moderates are opposite from the Left party in one dimension, as are Liberals and Social Democrats in the other and the center party, the Christian Democrats and the Green party are close, in between the others. (**Figure 5**). Also, gender and voting patterns are uncorrelated (**Figure 6**), as is mostly the case with the Districts (**Figure 7**), except for one case, by the bottom right.

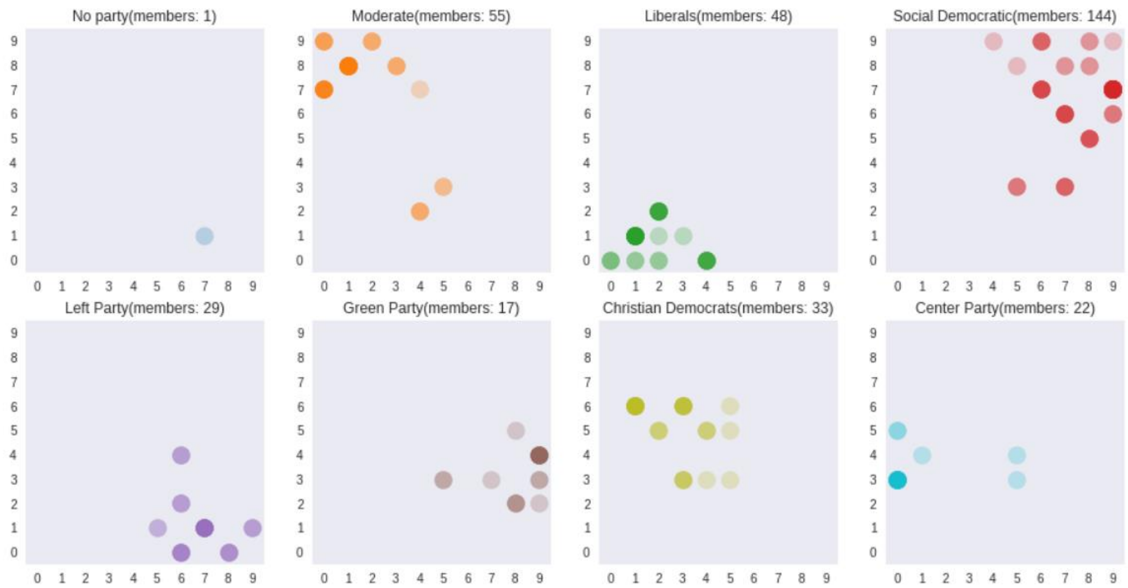

Figure 5: MPs clustered votes by Parties

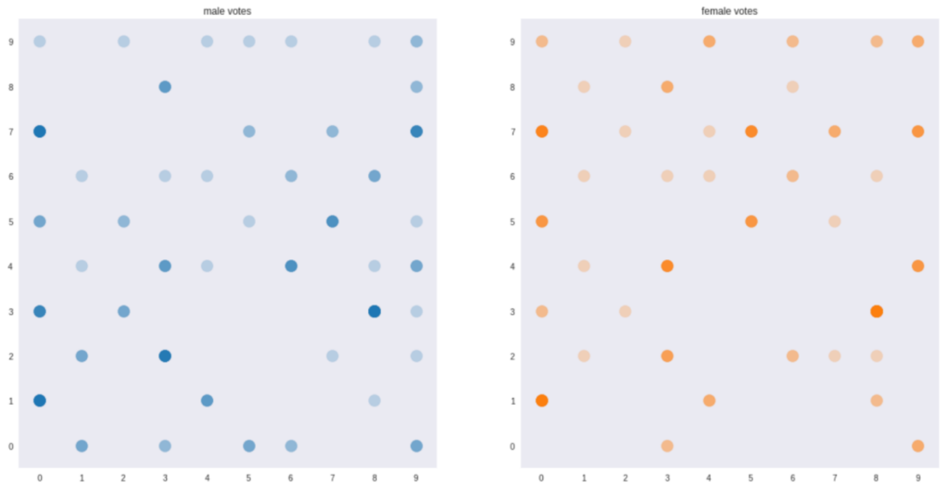

Figure 6: MPs clustered votes by gender (Male / Female)

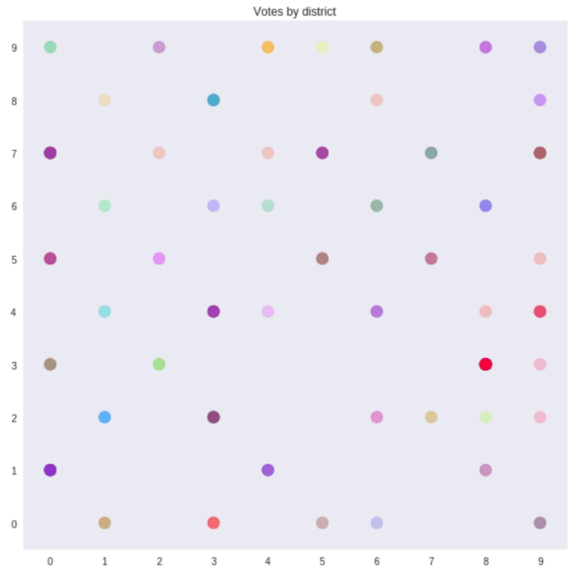

Figure 7: MPs clustered votes by District

5 Final remarks

RBF allows approximating very well functions that may be complex in other cases, such as the sine wave.

SOM is useful for exploratory analysis and find meaning in the data, allowing to discover the underlying structure of unlabelled data in an unsupervised matter.