Elliptical Rule Extraction from a Trained Radial Basis Function Neural Network

Andrey Bondarenko, Arkady Borisov

DITF, Riga Technical University, Meza ¼, Riga, LV-1658, Latvia
Andrejs.bondarenko@gmail.com, arkadijs.borisovs@cs.rtu.lv

Abstract: *Currently knowledge management and discovery plays crucial role in different industries. Apart from that in many cases decisions made by companies in mission critical areas (like nuclear power industry, medicine and finance, to name a few) should be verified by domain expert and/or explained. This is needed to fulfill regulatory requirements, to verify correctness and / or discover new knowledge. Artificial neural networks are well known models that can be used to solve classification problems. Unfortunately they are "black box" models for end users, because classification process is fully hidden from the observer and cannot be easily mapped into formalized meaningful form that can be understood by the domain expert. There are multiple knowledge representations, and the one chosen to work with in this study is elliptical rules. A set of such rules can be used to determine the class of an input data point by the determination of ellipsoid that covers provided point. Current paper addresses the problem of elliptical rules (ER) extraction from trained artificial radial basis function neural network (RBFNN). This study uses RBFNN with tunable nodes - such networks are built using orthogonal forward feature selection algorithm and they usually have much less neurons (in comparison to RBFNNs containing neurons with fixed radii) while having comparable or even superior accuracy. The article poses non-convex optimization problem of finding multiple ellipsoids of largest volume inscribed into RBFNN decision boundary. Non-convexity arises due to complex nature of RBFNN decision boundary which serves as constraint. Further we describe an algorithm for extraction of ER from trained RBFNN. The provided experimental results show that few of the extracted elliptical rules have comparable and in some cases higher classification accuracy than original RBFNN. Although curse of dimensionality is applicable to the provided algorithm, experiments have shown that it can be readily used for relatively low-dimensional problems. Finally are described the possible future research directions.*

Keywords: radial basis function networks, knowledge acquisition, optimization.

Introduction

There are many problems for which artificial neural networks are still a preferable solution. They can be trained on data with outliers and are natural choice for multi classification problems, although the way in which classification is performed is a black-box algorithm for the end user. Due to specific requirements of different industries for clear and formal decision process description for validation and knowledge gathering purposes, knowledge extraction from trained neural network is a topical problem. In this paper we propose an algorithm for the extraction of elliptical rules from trained artificial radial basis function neural network with tunable nodes (Chen et al., 2009). Radial basis function neural networks are local in their nature - meaning each neuron is responsible for classification decision in specific region based on defined neuron radius and location parameters (and weight). Having that, it is possible to define optimization problem that would allow us to cover classification region with ellipsoids - hence giving us a formalized view of how classification is made. The structure of the paper is as follows: Chapter 2 provides background on neural networks, Chapter 3 poses optimization problem and algorithm, Chapter 4 shows experimental results and Chapter 5 concludes.

RBF Neural Network With Tunable Nodes

Radial basis function artificial neural network (Moody et al., 1989; Poggio et al., 1990) can be represented as follows:

$$(x) = \sum_{i}^{N} a_i \, p(\|x - c_i\|) \quad \textbf{(1)}$$

where a_i is the *i*-th neuron weight, c_i is the *i*-th neuron center, *N* is neurons count. The norm is usually taken to be Euclidean distance and the basis function *p* is taken to be Gaussian:

$$p(\|x - c_i\|) = \exp(-\beta \|x - c_i\|^2) \quad \textbf{(2)}$$

There are multiple strategies for training this kind of network. The network can have neurons with fixed radii, as this simplifies learning procedure. On the other hand, having neurons with different radii reduces the amount of neurons required to get necessary classification accuracy. But having neurons with varying radii requires a

specialized learning approach. We used the method described in (Chen et al., 2009). The proposed method allows us to build RBF networks with a small amount of neurons while preserving high accuracy.

Rule Extraction

Optimization problem

We can treat elliptical rule extraction from trained RBF neural network can as an optimization problem of finding ellipsoids of maximum volume inscribed into the space area(-s) defined by RFBNN decision boundary. It is possible to choose maximization criteria other than volume- like amount of points covered by newly shaped ellipsoid, but in this case we will not be able to use gradient-based solvers.

Let's denote ellipsoid as

$$\varepsilon = \{Bu + d \mid \|u\|_2 \} \quad \textbf{(3)}$$

a unit ball under affine transformations. As described in (Boyd and Vandenberghe, 2004) we say that *B* is n-length vector containing positive elements, so ellipsoid volume is proportional to *detB*. We can write down the following optimization problem:

$$\begin{aligned} &\max \log(\det B)) \\ &s.t. RBNN \supseteq \varepsilon \end{aligned} \quad \textbf{(4)}$$

This means – we are looking for ellipsoid of maximum volume fully inscribed into RBFNN defined decision boundary. Apart from that we have explicit constraint that ellipsoid should have non-negative radii – elements of vector B. The described problem allows us to find first ellipsoid inscribed into the RBFNN decision boundary. We should note that due to RBFNN nature, constraints of our problem are non-convex, thus the found ellipsoid can be a local solution. In most cases it will be insufficient to represent RBFNN with a single ellipsoid, thus we need to search for additional ellipsoids.

For that we need to apply the iterative approach. In contrast to recursive volume subdivision applied in (Nunez et al., 2002), we have chosen another strategy. Although recursive subdivision is also a feasible approach as it does not require objective function modification, on the other hand, it generates larger count of ellipsoids. Instead of space subdivision for searchable regions on each recursive iteration, we are just looking for inscribed ellipsoid with maximum volume not covered by already found ellipsoids, meaning the larger fraction of ellipsoid lies outside the region already covered by existing ellipsoids, the better it fits our needs:

$$\begin{aligned} &\max(\varepsilon_{vol} - E_{vol}) - P \\ &s.t. RBNN \supseteq \varepsilon \end{aligned} \quad \textbf{(5)}$$

here ε_{vol} denotes the volume of found ellipsoid and E_{vol} is the volume of already existing ellipsoids and *P* is penalty term. Modification in terms of volume calculation introduced large plateau areas with objective function equal to zero. To allow faster convergence of optimization procedure, penalty term *P* calculates minimal distance between candidate ellipsoid center and border of set formed by intersection of all previously found ellipsoids. Thus the 'further' the center of candidate ellipsoid contained within other found ellipsoids is, the larger penalty term *P* will be.

Such modifications allow us to ensure that, on each optimization iteration convergence will run faster and will allow us to find a new ellipsoid, which will cover most of not yet covered volume. Of course, there is no guarantee that the found ellipsoid will be a global solution.

Algorithm

Here we will describe an algorithm for elliptical rule extraction from a trained RBFNN. It is necessary to mention that extracted ellipsoids will be oriented parallel to coordinate axes, so original unit ball is deformed only by stretching, but not by rotation. On the very first iteration we need to find an inscribed ellipsoid of maximum volume that fits into the RBFNN decision boundary. In this case, objective function is defined as:

$$\log(prod(B)) \quad \textbf{(6)}$$

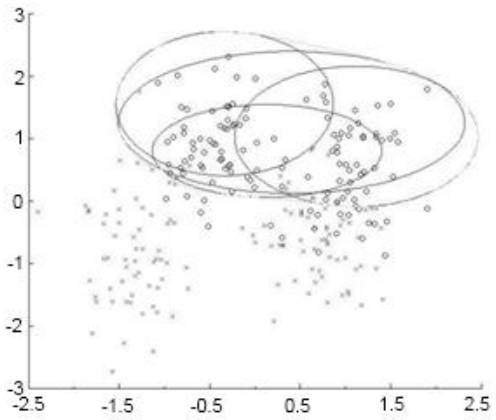

Fig. 1. **Decision boundary of RBFNN with 2 neurons and 2 extracted ellipsoids.**

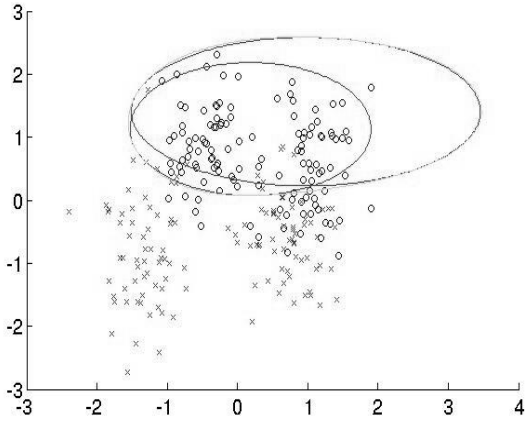

Fig. 2. **Decision boundary of RBFNN with 6 neurons and 4 extracted ellipsoids.**

The algorithm is listed below and has these inputs: *Data* - training input vectors, *C, R, w* - are parameters depicting RBFNN from which will be decomposed into the elliptical rules. The output of the algorithm is *Ellipses* set which contains ellipsoids. In the algorithm description *cRBF* – is handle for constraint function RBF which accepts *C, R* and *w* which are neurons centers, radii and weights respectively; *ub, lb* and *x0* – are the upper bound, lower bound and initial starting point for optimization. o*bjVol* – is objective function which accepts ellipsoid vector containing ellipsoid radii and ellipsoid center vector. cR*bfMod* and *objVolMod* are constraints and objective function modified versions. *U* – are points covered by RBFNN, but not covered by the supplied set of ellipsoids. *n* – is the amount of points (from *U*) that are not covered by newly found ellipsoid. If *n* is 0, then algorithm ads newly found ellipsoid and returns the result. This version of the algorithm is different than that provided in (Bondarenko and Borisov, 2012). Although it produces the same results in a two-dimensional case, it is capable of generating only a single inscribed ellipsoid. In *haberman* data set we stumbled on cases when a single ellipsoid was enough to cover data all points that are located inside RBF decision boundary. Thus we slightly modified the algorithm allowing it to terminate after acquisition of the first ellipsoid. Another point not depicted in the algorithm below is that in one of the two examined cases the algorithm produced four ellipsoids (this was set by maximum allowed ellipsoids count variable), but their manual clean-up revealed that only two of them are sufficient for covering all, but two points covered by RBFNN being decomposed. Such clean-up cannot be easily incorporated into the algorithm itself as each found ellipsoid directs global search into new not covered volumes, thus even if ellipsoid is not covering any new not yet covered points it should be preserved. In case when genetic algorithm or multi-start global search are used this should not be a problem.

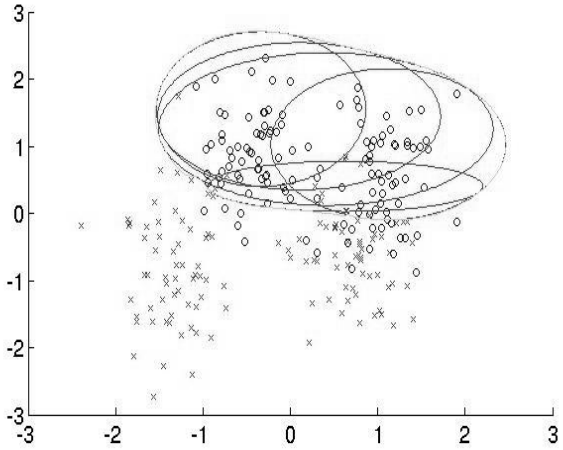

Fig. 3. **Decision boundary of RBFNN with 6 neurons and 5 extracted ellipsoids.**

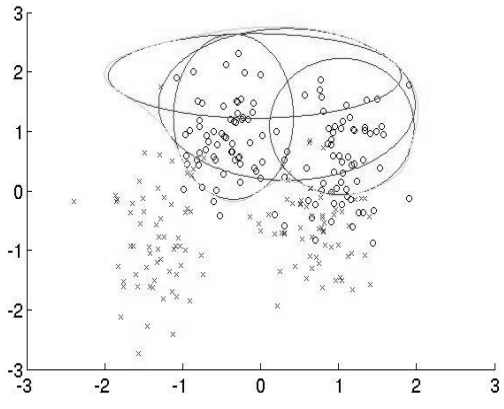

Fig. 4. **Decision boundary of RBFNN with 7 neurons and 4 extracted ellipsoids.**

```
IN:  ( maxEllipsoidsCount, Data, C, R, w )
OUT: ( Ellipsoids )

cRbf = @(x) constraintRbf(x, C, R, w);
objVol = @(x) objVolume(x);
e1 = solve(cRbf, objVol, ub, lb, x0);
Ellipsoids = e1;

U = uncoveredPoints(Data, Ellipsoids, C, R, w);
if (count(U) == 0)
   return e1;
end

i = 2;
while i < maxEllipsoidsCount
       i++;
       U = uncoveredPoints(Data, Ellipsoids, C, R, w);
       cRbfMod = @(x) constraintRbfModified(x, C, R, w);
       objVolMod = @(x) objVolumeModified(x);
       e = solve(cRbfMod, objVolMod, ub, lb, x0);
       n = size(uncoveredPoints(U, e, C, R, w), 1);
  Ellipsoids = Ellipsoids + e;
  if (n == 0)
     break;
  end
end
```

Table 1

RBFNN Accuracy, Extracted Ellipsoids Accuracy and Count

# of Neurons in RBFNN	RBNN Train Accuracy$_{(\text{std.dev.})}$	RBNN Test Accuracy$_{(\text{std.dev.})}$	Ellipsoids Train Accuracy$_{(\text{std.dev.})}$	Ellipsoids Test Accuracy$_{(\text{std.dev.})}$	Ellipsoids count $mean^{max}_{min}$
2 neurons	0.852	0.911	$0.84_{0.000}$	$0.887_{0.000}$	2^2_2
6 neurons	0.868	0.905	$0.868_{0.002}$	$0.9032_{0.005}$	4.4^4_5
7 neurons	0.876	0.905	$0.876_{0.000}$	$0.9031_{0.002}$	5.1^4_7
9 neurons	0.868	0.905	$0.8728_{0.005}$	$0.9039_{0.001}$	6.8^5_7

Experiments

We have created an algorithm supporting two and three dimensional input space. Furthermore, our final goal was to show that extracted rules are approximating RBFNN as close as possible. We set up experiments on synthetic two-dimensional Ripley data set, which can be found at (Frank and Asuncion, 2012) as well as haberman survival data set. Both of them have two classes. We run RBFNN construction algorithm described in (Chen et al., 2009) to construct several neural networks containing different amounts of neurons. Furthermore we observed only closed RBFNN defined classification boundaries which can be seen in figures. Looking at the algorithm one can notice *maxEllipsoidsCount* variable. We've initialized it with the number of neurons in the subject RBFNN, the only exception was a network with 9 neurons, for which maximum ellipsoids count to be extracted was set to seven as well as three dimensional data case where we set that value to three ellipsoids.
As it was already noted, the algorithm was not executed on open (not bounded areas meaning decision boundary lies outside the lower and upper bounds) decision areas, and RBFNN decision boundaries consisting of several separate space volumes (like two neurons forming two separate positive decision areas). Decision boundaries and extracted ellipses can be observed in Fig. 1-8 for two dimensional case and Fig. 9-10 for three-dimensional case. Experimental results can be observed in Table 1. One can notice that testing accuracies are higher than the training ones; this is due to the nature of training/testing data being used. We have not collected accuracies for three-dimensional cases apart the fact that ellipsoids built for the case depicted in Fig.9 left 3 uncovered points, while 2 ellipsoids depicted in Fig. 10 covered all data points that lied inside RBFNN decision boundary.

Overall computation time was partially an issue for us, finding the very first ellipsoid turned out to be rather quick operation while searching for subsequent ellipsoids was more CPU intensive task due to the amount of

computations needed to calculate modified objective function. Another point to mention is algorithms used in volume intersection and ellipsoid containment with RBFNN boundary calculations. In case of 2 dimensions we utilized mapping / clipping algorithms that allow us to acquire figure which corresponds to area belonging to candidate ellipse laying outside of already found ellipses. In case of three dimensions, we utilized an approach similar to the one described in (Persson and Strang, 2004). We defined distance-based volumes which were translated into three-dimensional meshes. This implied selection of appropriate mesh grid step for construction of isosurface for which volume can be calculated with appropriate precision. For checking whether ellipsoid is fully included within RBFNN decision boundary we created a set of points on its surface and checked each of points to be belonging to the required bounded volume. Although one can use any of the global optimization approaches, we believe GA is not a good option here, while search involving local solvers with different starting points proved to be the best in terms of computation time. Overall accuracy of the extracted rules – ellipses in 3 out of 4 cases lies within 1% which is a good result, taking into account the amount of rules being extracted.

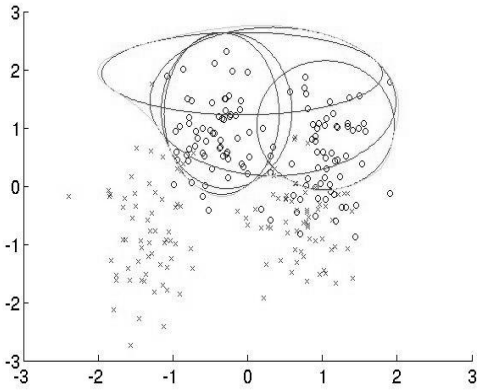

Fig. 5. **Decision boundary of RBFNN with 7 neurons and 5 extracted ellipsoids.**

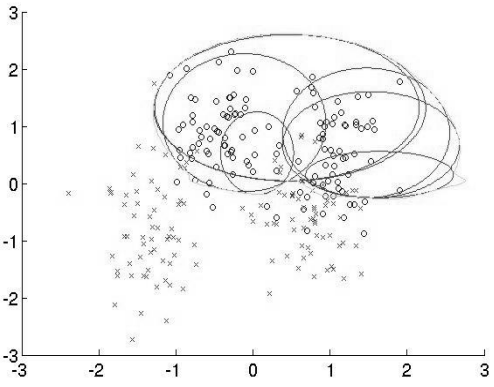

Fig. 6. **Decision boundary of RBFNN with 9 neurons and 7 extracted ellipsoids.**

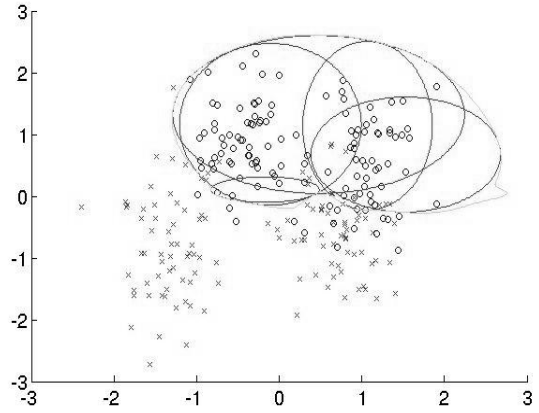

Fig. 7. **Decision boundary of RBFNN with 9 neurons and 5 extracted ellipsoids.**

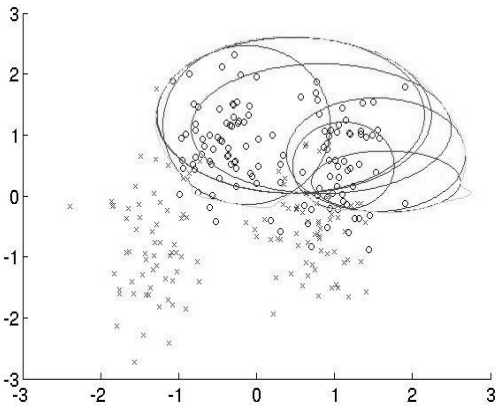

Fig. 8. **Decision boundary of RBFNN with 9 neurons and 7 extracted ellipsoids.**

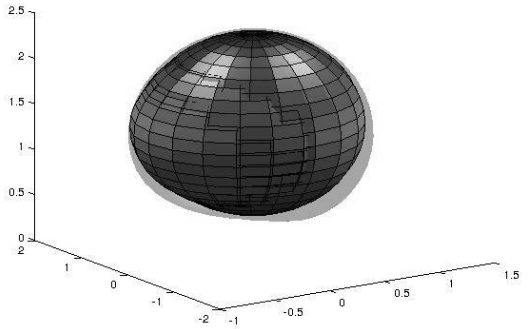

Fig. 9. **Three dimensional case. Decision boundary of RBFNN with 2 neurons and 3 extracted ellipsoids.**

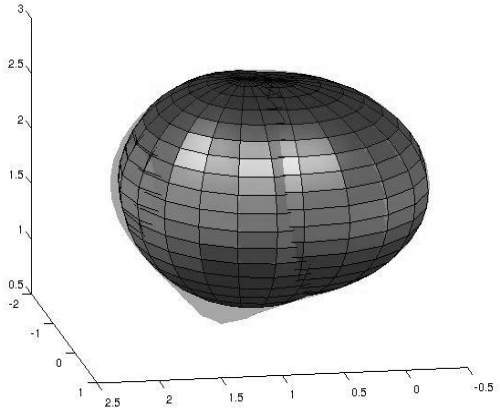

Fig. 10. **Three-dimensional case. Decision boundary of RBFNN with 2 neurons and 2 extracted ellipsoids.**

We believe it is possible to lower execution times by introducing heuristics for initial point selection. Thus basically we can deal with the acquisition of local solutions which can be better approach than the utilization of multistart optimization approach. In our algorithm we have used an ellipse fully residing inside RBFNN decision boundary without applying additional constraints; though constraints relaxation can potentially lower the count of extracted rules.

Conclusion

We have investigated the possibility of elliptical rule extraction from radial basis function neural networks. We developed an algorithm which was successfully applied to two and three-dimensional input data and radial basis function neural network trained on that data. The results observed indicate that the proposed algorithm can be successfully applied to low-dimensional problems. The experiments have shown that computation time can be a problem especially in case of larger RBF neural networks.

Apart from that feasible research direction is RBF structure exploiting to speed up constraints calculation, along with that algorithm is not tested on RBFNN decision boundary which covers open sets and several isolated space regions. Here by saying an open set we mean that classification boundary partially lies behind the upper and lower domain boundaries. This can be solved by extending boundaries further away or even eliminating them, although the effect of that needs to be tested and clearly finding appropriate solution depends on the optimization algorithm used. As it was noted in the Chapter Rule Extraction, one can utilize recursive space subdivision for subsequent searches although this can imply some limitations on fitness of found ellipsoids.

Overall amount of found ellipsoids along with the demonstrated accuracy shows that the proposed approach is feasible, especially for small radial basis function neural networks with low-to-moderately sized RBF layer.

References

Moody, J.and Darken, C.J., 1989. Fast learning in networks of locally tuned processing units, *Neural Computation*, 1, pp.281-294.

Poggio, T.and Girosi, F., 1990. Networks for approximation and learning, Proc. IEEE 78(9), pp.1484-1487.

Chen, S.X., Hong, Luk, B.L. and Harris, C.J., 2009. Construction of tunable radial basis function networks using orthogonal forward selection, *IEEE Trans. Systems, Man, and Cybernetics, Part B*, Vol.39, No.2, pp.457-466.

Bondarenko, A. and Jumutc, V., 2011. Extraction of Interpretable Rules from Piecewise-Linear Approximation of a Nonlinear Classifier Using Clustering-Based Decomposition, Cambridge, AIKED.

Nunez, H., Angulo, C., Catala, A., 2002. Rule extraction from support vector machines, Proceedings of the 10th European Symposium on Artifical Neural Networks, pp.335-343, Bruges, Belgium, April24-26.

Boyd, S., Vandenberghe, L., 2004. *Convex Optimization*, Cambridge University Press.

Bondarenko, A. and Borisov, A., 2012. Elliptical Rules Extraction from Trained Radial Basis Function Neural Network, Scientific Journal of Riga Technical University.

Frank, A.and Asuncion, A., 2012. *UCI Machine Learning Repository*, Available at: http://archive.ics.uci.edu/ml, 29, 2012. School of Information and Computer Science, Irvine, University of California.

Persson, P.O. and Strang., G.,2004. *A Simple Mesh Generator in Matlab*, SIAM Review, June, Volume 46(2), pp. 329-345.