doi: 10.2478/v10313-012-0027-2

The Extraction of Elliptical Rules from the Trained Radial Basis Function Neural Network

Andrey Bondarenko[1], Arkady Borisov[2], [1-2]*Riga Technical University*

Abstract **– The paper describes an algorithm for approximation of trained radial basis function neural network (RBFNN) classification boundary with the help of elliptic rules. These rules can later be translated into IF-THEN form if required. We provide experimental results of the algorithm for a two-dimensional case. Currently, neural networks are not widely used and spread due to difficulties with the interpretation of classification decision being made. The formalized representation of decision process is required in many mission critical areas, such as medicine, nuclear energy, finance and others.**

Keywords **– radial basis function networks, knowledge acquisition, optimization**

I. Introduction

There are many problems to which artificial neural networks are still a preferable solution. They can be trained on data with outliers and are naturally chosen for multiple classification problems. However, the way, in which classification is made, is a black-box algorithm for the user. Due to specific requirements of different industries for clear and formal decision process description for validation and knowledge gathering purposes, knowledge extraction from the trained neural network is a topical problem. In the current paper, we propose the algorithm for extraction of elliptical rules from the trained artificial radial basis function neural network. Radial basis function neural networks are local in their nature, i.e., each neuron is responsible for classification decision in a specific region based on defined neuron radius and location parameters (and weight). Thus, it is possible to define an optimization problem that will allow us to cover a classification region with ellipsoids, hence, giving us a formalized view of how classification is made. The structure of the paper is as follows: Section 2 provides the background information on a neural network, Section 3 poses an optimization problem and algorithm, Section 4 shows experimental results, and Section 5 draws conclusions.

II. RBF Neural Network With Tunable Nodes

Radial basis function artificial neural network [1,2] can be represented as follows:

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

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

$$p(\|x - c_i\|) = \exp\left[-\beta \|x - c_i\|^{\,2}\right] \tag{2}$$

There are multiple strategies for training such a network. Network can have neurons with fixed radii, as it simplifies a learning procedure. On the one hand, neurons with different radii reduce the number of neurons required to get necessary classification accuracy [3]. On the other hand, neurons with varying radii require a specialized learning approach. We have used the method described in [3]. The proposed method allows us to build RBF networks with a small number of neurons and at the same time preserve high accuracy.

III. Rule Extraction

A. Optimization Problem

The extraction of elliptical rules from the trained RBF neural network can be treated 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 other maximization criteria; instead of volume it can be the number of points covered by a newly shaped ellipsoid.

Let us denote an ellipsoid as:

$$\varepsilon = \left\{ |\, Bu + d \,|\, \|u\|_2 \right\}. \tag{3}$$

a unit ball under affine transformation. As described in [5], we say that *B* is the n-length vector containing positive elements, so ellipsoid volume is proportional to *detB*. We can write next optimization problem:

$$\begin{array}{l} \max\log(\det B)) \\ s.t.\, RBNN \supseteq \varepsilon \end{array} \tag{4}$$

i.e., to find the ellipsoid of maximum volume fully contained within RBFNN defined decision boundary. Apart from that we have explicit constraint that the ellipsoid should have non-negative elements in its radius vector *B*. The described problem allows us to find the 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 this reason, we need to apply an iterative approach. In contrast to the recursive volume subdivision applied in [6, 7], we have chosen another strategy. Although the recursive subdivision is a feasible approach as it does not require the objective function modification, it generates a large number of ellipsoids. Instead of space subdivision for searchable regions on each subsequent iteration, we just look for the inscribed ellipsoid with maximum volume not covered by the found ellipsoids:

$$\begin{gathered} \max\left(\varepsilon_{vol} - E_{vol}\right) - P \\ s.t.\, RBNN \supseteq \varepsilon \end{gathered} \tag{5}$$

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

In this way we can ensure that on each iteration faster convergence will allow us to find a new ellipsoid, which will cover most of the area not yet covered. Of course, there is no guarantee that the found ellipsoid will be a global solution.

B. Algorithm

In this section we will describe the algorithm for elliptical rule extraction from trained RBFNN. We should mention that the extracted ellipsoids will be oriented parallel to coordinate axes. On the first iteration we need to inscribe the ellipsoid of maximum volume into the RBFNN decision boundary. The objective function is computed as follows:

```
log(prod(B)))
```

In general, the algorithm listed below has the following inputs: *Data* – training input vectors, *C*, *R*, *w* – parameters depicting RBFNN, from which knowledge in the form of elliptical rules will be extracted. Output of the algorithm is *Ellipse* set, which contains ellipsoids. In the algorithm description *cRBF* – is the handle for constraint function RBF, which accepts *C*, *R* and *w*, which are neuron centres, radii and weights, respectively. *ub*, *lb* and *x0* are upper bound, lower bound and initial starting point for optimization. o*bjVol* is the objective function, which accepts the ellipsoid vector containing ellipsoid radii and ellipsoid centre vector. cR*bfMod* and *objVolMod* are constraints and objective function modified versions. *U* – points covered by RBFNN, but not covered by the supplied set of ellipsoids. *n* is a number of points (from *U*) that are not covered by a newly found ellipsoid. If *n* is 0, then the algorithm adds a newly found ellipsoid and returns the result.

```
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;
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 I

RBFNN ACCURACY. EXTRACTED ELLIPSOID ACCURACY AND NUMBER

# of Neurons in RBFNN	RBNN Train Accuracy$_{(std.dev.)}$	RBNN Test Accuracy$_{(std.dev.)}$	Ellipsoid Train Accuracy$_{(std.dev.)}$	Ellipsoid Test Accuracy$_{(std.dev.)}$	Ellipsoid number $mean_{min}^{max}$
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_{5}^{4}
7 neurons	0.876	0.905	$0.876_{0.000}$	$0.9031_{0.002}$	5.1_{7}^{4}
9 neurons	0.868	0.905	$0.8728_{0.005}$	$0.9039_{0.001}$	6.8_{7}^{5}

IV. Experiments

We have created the algorithm supporting two-dimensional input vectors. Furthermore, it has not been our final goal to verify best possible classification accuracy (which directly correlates to RBFNN accuracy), but to show that the extracted rules are approximating RBFNN as close as possible. We have conducted experiments on synthetic two-dimensional Ripley dataset, which can be found in [8]. We have implemented RBFNN construction algorithm described in [3] to construct several neural networks containing a variable number of neurons. Furthermore, we have observed only closed RBFNN defined classification boundaries, which may be seen in figures. Looking at the algorithm, one can notice *maxEllipsoidsCount* variable. We have initialized it with a number of neurons in the subject RBFNN, the only exception is a network with 9 neurons, for which the maximum number of ellipsoids to be extracted has been set to 7.

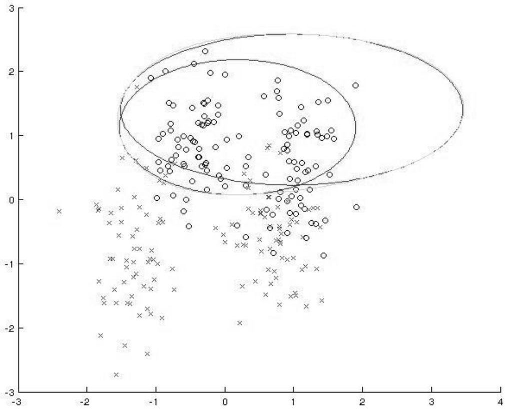

Fig. 1. Decision boundary of RBFNN with 2 neurons and 2 extracted ellipsoids

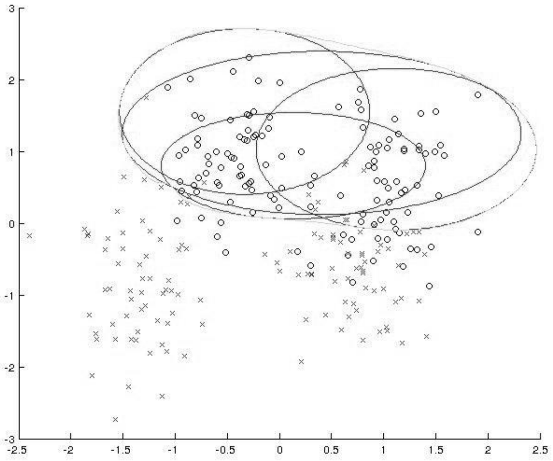

Fig. 2. Decision boundary of RBFNN with 6 neurons and 4 extracted ellipsoids

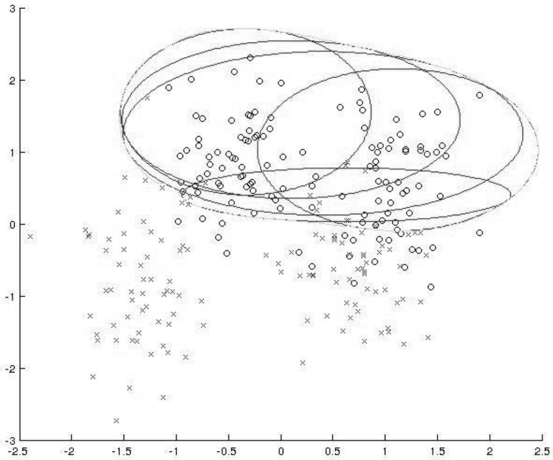

Fig. 3. Decision boundary of RBFNN with 6 neurons and 5 extracted ellipsoids

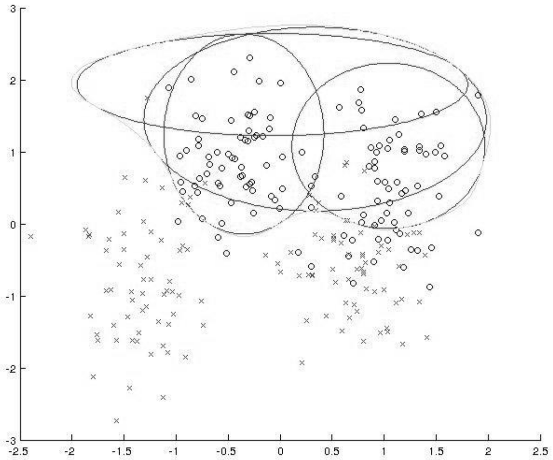

Fig. 4. Decision boundary of RBFNN with 7 neurons and 4 extracted ellipsoids

As already mentioned, you may notice that the algorithm has not been executed in open (unbounded areas, i.e., the decision boundary lies outside lower and upper bounds) decision areas, and RBFNN decision boundaries consist of several separate space volumes (like two neurons forming two separate positive decision areas). Decision boundaries and extracted ellipses can be observed in Fig. 1-9. Experimental results can be observed in Table III. One can notice that testing accuracies are higher than training ones due to nature of training/testing data being used.

Another point to mention is algorithms used in volume intersection and ellipsoid containment with RBFNN boundary calculations. To check whether an ellipsoid fully lies within RBFNN decision boundary, we have created a set of points on its surface and checked each of points to belong to a required set.

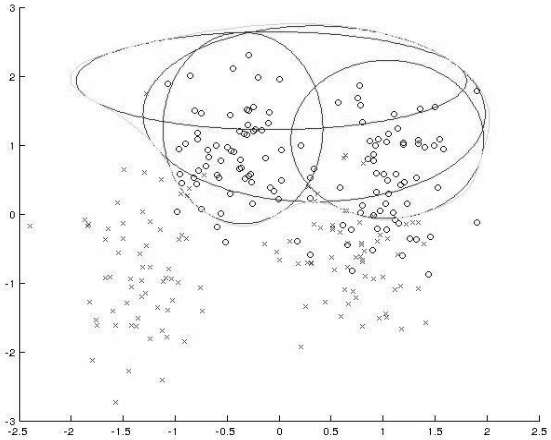

Fig. 5. Decision boundary of RBFNN with 7 neurons and 4 extracted ellipsoids

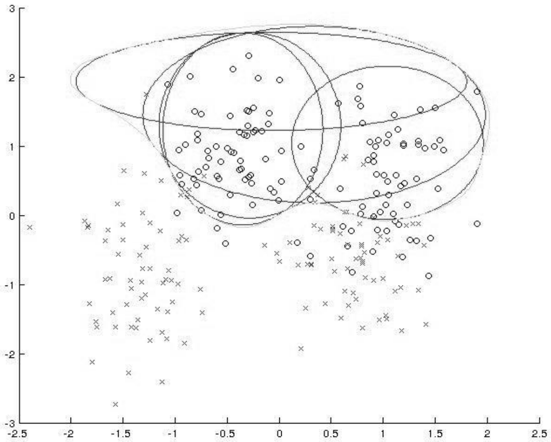

Fig. 6. Decision boundary of RBFNN with 7 neurons and 5 extracted ellipsoids

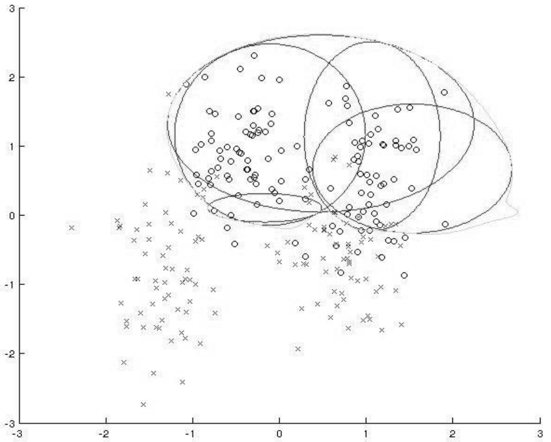

Fig. 7. Decision boundary of RBFNN with 7 neurons and 5 extracted ellipsoids

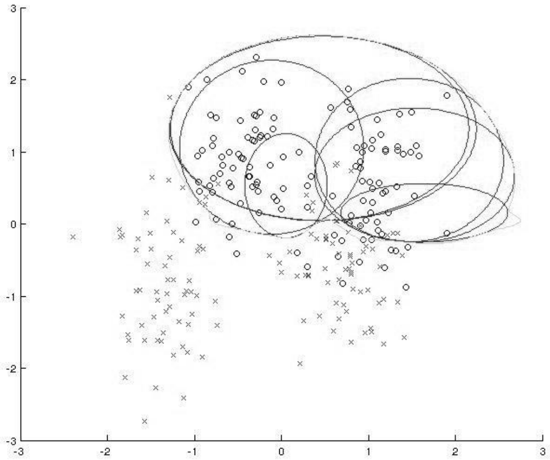

Fig. 8. Decision boundary of RBFNN with 9 neurons and 7 extracted ellipsoids

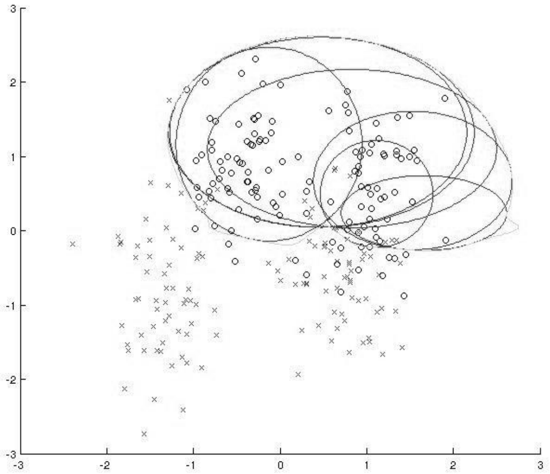

Fig. 9. Decision boundary of RBFNN with 9 neurons and 7 extracted ellipsoids

Overall computation time was partially an issue for us, because the process of finding very first ellipsoid turned out to be a rather quick operation while the process of searching for subsequent ellipsoids was more CPU intensive task due to the number of computations needed to calculate the modified objective function. Although one can use any of the global optimization approaches, we believe GA is not a good option here, while the search involving local solvers with different starting points has shown itself to be the best in terms of computation time. Overall accuracy of extracted rules – ellipses in 3 out of 4 cases – lies within 1%, which is a good result taking into account the number of rules being extracted.

We believe it is possible to lower execution times by introduction of heuristics for initial point selection. In our algorithm, we have used an ellipse fully residing inside RBFNN decision boundary without applying additional constraints.

V. Conclusions

We have investigated the possibility of elliptical rule extraction from radial basis function neural networks. We have developed the algorithm successfully applied to two-dimensional input data and radial basis function neural network trained on the data. The observed results indicate that the proposed algorithm can be successfully applied to low dimensional problems; thus, further research directions are related to the way of dealing with this problem.

Apart from that, feasible research direction is RBF structure exploiting to speed up constraint calculation, along with that the 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 upper and lower boundaries. This can be solved by extending boundaries further away or even eliminating them; however, the effect needs to be tested, and it is clear that it depends on optimization algorithms used. As mentioned in Section 3, one can utilize recursive space subdivision for subsequent searches; however, it can imply some limitations to the found ellipsoids.

Overall number of found ellipsoids along with demonstrated accuracy proves the proposed approach to be feasible, especially for low radial basis function neural networks with low-to-medium sized input layer.

References

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

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

[3] S. Chen, X. Hong, B.L. Luk and C.J. Harris, 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, 2009.

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

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

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

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

Andrey Bondarenko received the B.Sc. and Ms.Sc. degrees in 2004 and 2006 from the University of Latvia and Transport and Communication Institute in Riga, Latvia. Since 2010 he has been studying at Riga Technical University to obtain a doctoral degree in Computer Science. Currently major field of study is: extraction of concise rules from the trained artificial neural network. Previous publications: *Polytope Classifier: A Symbolic Knowledge Extraction from Piecewise-Linear Support Vector Machine*. LNCS 2011, Volume 6881/2011, pp.62-71. Research interests include: machine learning, computer vision, cognition. E-mail: andrejs.bondarenko@gmail.com

Arkady Borisov is Professor of Computer Science at the Faculty of Computer Science and Information Technology, Riga Technical University. He holds a degree of Doctor of Technical Sciences in Control of Technical Systems and a habilitation degree in Computer Science. His research interests include fuzzy sets, fuzzy logic, computational intelligence and bioinformatics. He has 220 publications in the fields of computer science and information technology. Contact information: 1 Kalku Street, Riga, LV-1658, phone: +371 67089530, e-mail: arkadijs.borisovs@cs.rtu.lv.

Andrejs Bondarenko, Arkādijs Borisovs. Eliptisko lēmumu izvilkšana no apmācīta radiālo bāzes funkciju neironu tīkla

Šobrīd plaši pielietot neironu tīklus dažādās nozarēs traucē to pieņemto klasifikācijas risinājumu necaurredzamība. Paskaidrojumi ir nepieciešami gan pieņemtā risinājuma validācijai, gan jaunu zināšanu par ieejas datos esošajām likumsakarībām izgūšanai. Šajā rakstā ir piedāvāts algoritms eliptisku likumu izgūšanai. Šeit eliptiskie lēmumi ir definēti kā *x^2/a^2 +y^2/b^2 <=1*, kur *a* un *b* ir elipses rādiusi. Domēna eksperts var viegli izanalizēt šādus likumus, kā arī tos var viegli transformēt uz IF-THEN-ELSE vai citu likumu/zināšanu formu no apmācīta mākslīgā neironu tīkla ar radiālo bāzes funkciju. Tiek aplūkots neizliekts optimizācijas uzdevums, kas tiek risināts ar globālās pārmeklēšanas palīdzību. Piedāvātais algoritms ir realizēts un pielietots RBF neironu tīklam ar ieejas slāni, kura izmērs ir divi. Tā kā pētījuma mērķis bija algoritma iespēju izpēte, nevis klasifikācijas precizitāte, tika izmantota viena sintētiska datu kopa, ar kuras palīdzību tika apmācīts RBF tīkls, izmantojot ortogonālās pazīmju izvēles algoritmu. Šis algoritms ļauj veidot RBF tīklus ar minimālu neironu skaitu ar dažādiem rādiusiem (kas orientēti paralēli koordinātu asīm). Rezultātā iegūtie neironu tīkli uzrāda augstu klasifikācijas līmeni. Rakstā aprakstīti eksperimentu rezultāti un vizualizēti izgūtie eliptiskie likumi, kas tika izgūti no tīkliem ar diviem, sešiem, septiņiem un deviņiem neironiem. Maksimālais likumu skaits, kas izgūti no tīkla, ir septiņi. Eksperimentu rezultāti liecina, ka trijos (no četriem) gadījumos eliptiskie likumi uzrādīja klasifikācijas kļūdu, kura salīdzinājumā ar RBF tīkla klasifikācijas kļūdu ir sliktāka par mazāk nekā vienu procentu. Piedāvāti arī turpmākie pētījuma virzieni.

Андрей Бондаренко, Аркадий Борисов. Извлечение эллиптических правил из обученной искусственной нейронной сети на радиальных базисных функциях

На сегодняшний день широкому применению искусственных нейронных сетей в различных отраслях препятствует непрозрачность принимаемого ими классификационного решения. Объяснение требуется как для валидации принятого решения, так и для извлечения новых знаний о взаимосвязях во входных данных. В данной статье предлагается алгоритм извлечения эллиптических правил вида *x^2/a^2 +y^2/b^2 <=1*, где *a* и *b* суть радиусы эллипса. Подобные правила могут быть относительно легко проанализированы доменным экспертом, либо трансформированы в IF_THEN_ELSE или другие виды правил/знаний из обученной искусственной нейронной сети на радиальных базисных функциях. Приводится постановка невыпуклой оптимизационной задачи, которая решается путем глобального поиска. Предложенный алгоритм реализован и применен к РБФ нейронной сети с размером входного слоя, равным двум. Поскольку целью исследования было изучение возможностей алгоритма, а не точности классификации, был использован один искусственный набор данных, на котором была обучена РБФ сеть с применением алгоритма ортогонального выбора признаков. Данный алгоритм позволяет строить РБФ сети с минимальным набором нейронов с различными радиусами (ориентированными параллельно осям координат). В результате полученные нейронные сети показывают высокий уровень классификации. В статье приведены результаты экспериментов, а также визуализированы извлеченные эллиптические правила. Проведено извлечение правил из сетей с двумя, шестью, семью и девятью нейронами. Максимальное количество правил, извлеченных из сети, равно семи. Результаты экспериментов показывают минимальную потерю точности в классификации между извлеченными эллиптическими правилами и аппроксимируемой искусственной нейронной РБФ сетью (в 3 из 4 случаев разница менее 1%). Предложены направления для дальнейших исследований.