

# Genetically Modified GMDH Method with Cloning

Marcel Jiřina<sup>1</sup> and Marcel Jiřina, jr.<sup>2</sup>

<sup>1</sup> Institute of Computer Science, Pod vodarenskou vezi 2,  
182 07 Prague 8 – Liben, Czech Republic  
marcel@cs.cas.cz  
<http://www.cs.cas.cz/~marcel/>

<sup>2</sup> Faculty of Biomedical Engineering, Czech Technical University in Prague,  
Zikova 4, 166 36, Prague 6, Czech Republic  
jirina@fbmi.cvut.cz  
<http://www.fbmi.cvut.cz/kontakty/jirina-marcel/>

*(Paper received on June 11, 2007, accepted on September 1, 2007)*

**Abstract.** The GMDH MIA algorithm is modified by the use of the selection procedure from genetic algorithms and including cloning of the best neurons generated to get even lesser errors. The selection procedure finds parents for a new neuron among already existing neurons according to fitness and with some probability also from network inputs. The essence of cloning is a slight modification of parameters of copies of the best neuron, i.e. a neuron with the largest fitness. The genetically modified GMDH network with cloning (GMC GMDH) can outperform other powerful methods. It is demonstrated on some tasks from the Machine Learning Repository.

## 1 Introduction

In this paper we solve difficult classification tasks by the use of the GMDH MIA, i.e. group method data handling multilayer iterative adaptive method, modified by selection algorithm common in genetic algorithms and by cloning the best neurons generated up to a given a moment of the learning process. Thus we combine genetic algorithm from which we use selection operation and immune networks from which we use a notion of cloning.

The basis of our method is the standard GMDH MIA method described in many papers since 1971 by [1], [4], [5], [6], [7], [10] and many others.

The basic approach of the GMDH is that each neuron (node or unit) in the network receives input from exactly two other neurons with the exception of neurons representing the input layer. The two inputs,  $x$  and  $y$  are then combined to produce a partial descriptor based on the simple quadratic transfer function (The output signal is  $z$ ):

$$z = a_1 + a_2x + a_3y + a_4x^2 + a_5y^2 + a_6xy, \quad (1)$$

where coefficients  $a, \dots, f$  are determined by linear regression and are unique for each transfer function, i.e. each neuron. The coefficients can be thought of as analogous to weights found in other types of neural networks.

The network of neurons with transfer functions (1) is constructed as one layer at a time. The first network layer consists of functions of each possible pair of  $n$  input variables (zero-th layer) resulting in  $n(n-1)/2$  neurons. The second layer is created using inputs from the first layer and so on. Due to exponential growing of a number of neurons in a layer, after finishing the layer, the limited number of the best neurons is selected and the others removed from the network.

In [3], a selection procedure from genetic algorithms is used for finding two parents for a new neuron instead of the systematic procedure in the standard GMDH MIA algorithm with apparent layered structure. Our approach uses a selection procedure from genetic algorithms, but there is no explicit layered structure. The network grows during learning one neuron at a time. No neuron is deleted during the learning process and in the selection procedure its chance to become a parent for a new neuron is proportional to its fitness. If a new neuron appears to be the best, its clones are generated.

Clones are inexact copies of the parent neuron, which was found to be the best neuron up to now generated. Inexact copies follow from the fact that having exact copy has no sense in GMDH networks, so some mutation process must be applied to get clones a little bit different from the parent neuron.

It is shown here that a new algorithm, especially cloning, allows tuning the GMDH neural network more effectively than it is possible in genetically optimized GMDH networks.

## 2 Genetically modified GMDH

Here we describe approaches which result from our constructing of genetically modified GMDH network with cloning.

### 2.1 The learning set

We assume an  $n$ -dimensional real valued input and a one-dimensional real valued output. The learning set consists of  $n+1$  dimensional vectors  $(x_i, y_i) = (x_{1i}, x_{2i}, \dots, x_{ni}, y_i)$ ,  $i = 1, 2, \dots, N$  where  $N$  is the number of learning samples or examples. The learning set can be written in the matrix form

$$[X, Y].$$

The matrix  $X$  has  $n$  columns and  $N$  rows;  $Y$  is a column vector of  $N$  elements. In the GMDH the learning set is usually broken to two disjoint subsets, the construction (training) set or setup set and the so-called validation set. In the learning process the former one is used for setting up parameters of neurons of the newly created layer, the



latter for evaluation of an error of newly created neurons. Thus  $N = N_s + N_v$ , where  $N_s$  is the number of rows used for setting up the parameters of neurons (the training set), and  $N_v$  is the number of rows used for error evaluation during learning (the validation set).

## 2.2 Selection process in GMDH MIA

Hiasaat and Mort have published a very interesting and principally simple approach in 2004 [8]. Their method does not remove any neuron during learning. Thus it allows unfit individuals from early layers to be incorporated at an advanced layer where they generate fitter solutions. Secondly, it also allows those unfit individuals to survive the selection process if their combinations with one or more of the other individuals produce new fit individuals, and thirdly, it allows more implicit non-linearity by allowing multi-layer variable interaction. The GMDH algorithm is constructed in exactly the same manner as the standard GMDH algorithm except for the selection process. In order to select the individuals that are allowed to pass to the next layer, all the outputs of the GMDH algorithm at the current layer are entered as inputs in the algorithm. It was shown in [8] that this approach can outperform the standard GMDH MIA when used in the prediction of two daily currency exchange rates.

## 2.3 New genetically modified GMDH network algorithm

The standard quadratic neuron is an individual of the genetically modified GMDH network. Its parents are two neurons (or possibly one or two network inputs) from which two input signals are taken. A selection of one neuron or input as one parent and another neuron or input as the other parent can be made by the use of different criteria. In genetic algorithms in selection step there is a common approach that the probability to be a parent is proportional to the value of the fitness function. Just this approach is used here. The fitness is simply reciprocal of the mean error on the validation set.

Note that in the standard GMDH MIA algorithm all possible pairs of neurons from the preceding layer (or inputs when the first layer is formed) are taken as pairs of parents. The selection consists of taking a limited number of the best descendants, "children", while the others are removed after they have arisen and were evaluated. In this way, all variants of the GMDH MIA are rather ineffective as there are a lot of neurons generated, evaluated and then simply removed with no other use.

An operation of a crossover in the genetically modified GMDH is, in fact, no crossover in the sense of combining two parts of parents' genomes. In our approach eq. (1) gives an exactly symmetrical procedure of mixing the parents' influence but not their features, parameters.

## 2.4 Selection procedure

The initial state form  $n$  inputs only, there are no neurons.

If there are  $k$  neurons already, the probability of a selection from inputs and from neurons is given by

$$p_i = n/(n + k),$$

$$p_n = k/(n + k)$$

for  $n/(n + k) > p_0$ , where  $p_0$  is minimal probability that one of network inputs will be selected; we found  $p_0 = 0.1$  as optimal. Otherwise

$$p_i = p_0,$$

$$p_n = (1 - p_0).$$

The fitness function is equal to the reciprocal error on the verification set. Let  $\varepsilon(j)$  be the mean error of the  $j$ -th neuron on the validating set. The probability that neuron  $j$  will be selected is the following:

$$p_n(j) = (1 - p_n) \frac{1/\varepsilon(j)}{\sum_{s=1}^{N_r} 1/\varepsilon(s)}.$$

Moreover, it must be assured that the same neuron or the same input is not selected as a second parent of the new neuron.

After the new neuron is formed and evaluated it can become immediately a parent for another neuron. Thus the network has no explicit layered structure. Each new neuron can be connected to any input or up to now existing neurons.

The computation of six parameters  $a, \dots, f$ , see (1), of the new neuron is the same as in the GMDH MIA algorithm.

## 2.5 Best neuron

A new neuron added need not be better than any other. Therefore, the index and error value of the best neuron is stored as long as a better neuron arises. Thus every time there is information about the best neuron, which can act as network's output. After learning, this output is used as a network output in the recall phase.

## 2.6 Pruning

After learning, the best neuron and all its ancestors have their roles in the network function. All others can be removed.



### 3 Cloning

There are various mechanisms or processes in the immune system, which are investigated in the development of artificial immune systems. A comprehensive summary can be found in [11].

Note first that authors dealing with artificial immune systems, e.g. [12] use rather different terminology than that used in neural networks community and genetic algorithms community. So some translation or mapping is needed. Here especially, antibody – neuron, affinity – fitness.

#### 3.1 Cloning mechanisms

There are lots of ideas about approaches to cloning. From these ideas we use cloning in form close to the SIMPLE CLONALG algorithm [12] in this way:

```
BEGIN
  Given the Best GMDH Neuron with parents (i.e. input
  signals from)  $In_1$ ,  $In_2$  and with six parameters  $a$ ,  $b$ ,
  ..  $f$ .
  REPEAT
    Produce a clone of the Best Neuron. A clone
    has the same inputs  $In_1$  and  $In_2$  but mutated
    parameters  $a$ , ..  $f$ , i.e. parameters slightly
    changed (details see below).
    Evaluate fitness of this clone neuron.
    If this neuron happens to be better than the
    Best Neuron, break this clone generating cycle
    (and start this cloning algorithm from the
    beginning with new Best Neuron again).
  UNTIL a terminal criterion is satisfied or the
  maximum number of clones is reached.
END.
```

#### 3.2 Mutation

It has no sense for the clones to be exact copies of the Best Neuron. Therefore, some mutation must be in effect. The clone to be a true clone must have the same parents. So, basic parameters – the two parents are not changed. A problem is how to change parameters  $a$ , ..  $f$ . These changes should be small enough to keep sufficient similarity of clone to original individual (the Best Neuron) and, at the same time, sufficiently large to reach necessary changes for searching data space in the neighborhood of parameters of the Best Neuron.

## 2.4 Selection procedure

The initial state form  $n$  inputs only, there are no neurons.

If there are  $k$  neurons already, the probability of a selection from inputs and from neurons is given by

$$p_i = n/(n + k) ,$$

$$p_n = k/(n + k)$$

for  $n/(n + k) > p_0$ , where  $p_0$  is minimal probability that one of network inputs will be selected; we found  $p_0 = 0.1$  as optimal. Otherwise

$$p_i = p_0,$$

$$p_n = (1 - p_0).$$

The fitness function is equal to the reciprocal error on the verification set. Let  $\mathbb{E}(j)$  be the mean error of the  $j$ -th neuron on the validating set. The probability that neuron  $j$  will be selected is the following:

$$p_n(j) = (1 - p_n) \frac{1/\mathbb{E}(j)}{\sum_{s=1}^{N_n} 1/\mathbb{E}(s)}$$

Moreover, it must be assured that the same neuron or the same input is not selected as a second parent of the new neuron.

After the new neuron is formed and evaluated it can become immediately a parent for another neuron. Thus the network has no explicit layered structure. Each new neuron can be connected to any input or up to now existing neurons.

The computation of six parameters  $a, ..f$ , see (1), of the new neuron is the same as in the GMDH MIA algorithm.

## 2.5 Best neuron

A new neuron added need not be better than any other. Therefore, the index and error value of the best neuron is stored as long as a better neuron arises. Thus every time there is information about the best neuron, which can act as network's output. After learning, this output is used as a network output in the recall phase.

## 2.6 Pruning

After learning, the best neuron and all its ancestors have their roles in the network function. All others can be removed.



neurons generated for stopping computation was 10000. Probability that a new neuron's input is one of input signals was 10 %, probability that a new neuron's input is one of already existing neurons was 90 %. The maximal number of clones generated from one parent neuron is limited to  $\text{int}(\sqrt{\text{No. of neurons generated up to now}})$ . For all methods optimal threshold for a minimal error was used.

Classification errors for four different methods including the GMC GMDH are summarized in Table 1. In the second column the cross validation factor is given. The methods for comparison are 1-NN, the standard nearest neighbor method. Sqrt-NN, the  $k$ -NN method with  $k$  equal to the square root of the number of samples of the learning set. Bayes, the naïve Bayes method using ten bins histograms. LWM1, the learning weighted metrics method [14] modified with nonsmooth learning process.

**Table 1.** Classification errors for four methods on some data sets from UCI MLR.

Data set	Algorithm				
	1-NN	sqrt-NN	Bayes	LWM1	GMC GMDH
Shuttle-small	0.00259	0.00828	0.01294	0.00310	0.00259
Brest CW	0.04792	0.03255	0.05244	0.04538	0.04188
Vote	0.10526	0.06015	0.09774	0.07407	0.06667
Spam	0.09967	0.11270	0.14267	0.10735	0.10085
Adult	0.20826	0.21242	0.16370	0.17171	0.15923
Splice	0.40351	0.37206	0.28655	0.25874	0.13087
German	0.40767	0.20283	0.29768	0.72844	0.29465

In Table 1 and in Fig.1 it can be seen that the GMC GMDH method outperforms other methods in tasks Adult, Shuttle small, and Splice or nearly outperforms Brest CW, Spam, and Vote being the second best with very small difference with respect to the best method considered. It is the second best in task German.

## 5 Conclusions

The Genetically modified GMDH method is an elegant idea how to improve the efficiency of the popular GMDH MIA method. It is based on the usage of a selection principle of genetic algorithms instead of systematic assignment of all pairs formed by neurons of the last layer. Thus all neurons once generated remain at least potential parents for new neurons during the whole learning process. Also each input signal may be used with some probability as a parent signal for a new neuron. Thus the layered structure of the GMDH algorithm disappears because any new neuron can be connected to the output of any already existing neuron or even to any input of the network.

The target of this paper is to simplify the idea of a genetically modified GMDH neural network and to extend it by cloning. Clones are close but not identical copies of original individuals. An individual in our case is the best neuron. Intuition behind says

that even when the parameters of the best neuron were set up by linear regression, i.e. with a minimal mean squared error, due to nonlinearity of the problem as well as the GMDH network, the statistical normality assumptions are not met. Thus true minimum may lie somewhere in the neighborhood of parameters of the best neuron. Therefore the clones have mutated, i.e. slightly changed parameters of the parent individual, the best neuron. We found that the cloning with large parameters changes has small effect, but with small changes a new best neuron often arises.

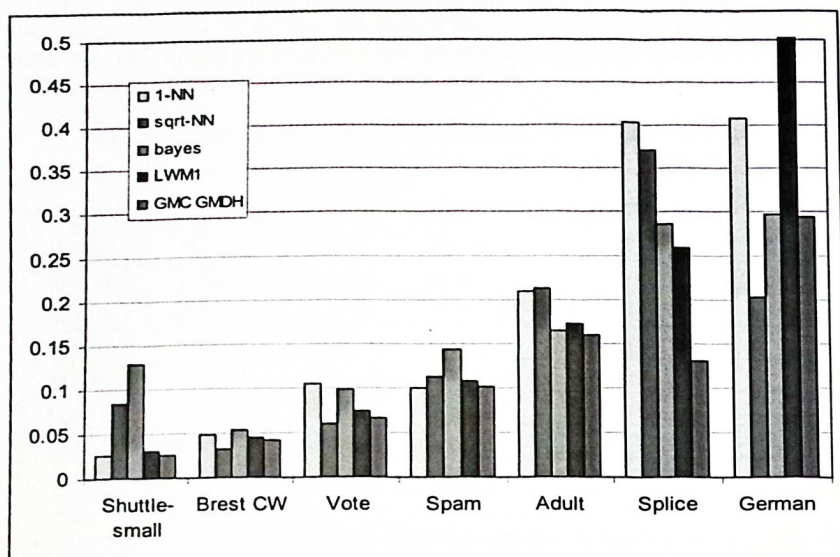


Fig. 2. Classification errors for four methods on some data sets from the UCI MLR. Note that for Shuttle small data the errors are ten times enlarged in this graph.

The genetically modified GMDH method with cloning (GMC GMDH) presented here appears to be a relatively simple and efficient method giving reliably good results better or comparable with the best results obtained by other methods. The new method behaves rather well and it has no critical parameters to be tuned. As there is no searching or sorting like in the nearest neighbor-based methods, the GMC GMDH is much faster than methods mentioned especially for large learning sets.

Essential advantage of the genetically modified GMDH with cloning is that one need not set up the number of the best neurons selected in the newly generated layer and thus indirectly control learning time and size of the network as in a standard GMDH MIA algorithm. Only the limitations of the genetically modified GMDH method with cloning are learning time or memory size.



## Acknowledgement

This work was supported by the Ministry of Education of the Czech Republic under project Center of Applied Cybernetics No. 1M0567, and project No. MSM6840770012 Transdisciplinary Research in the Field of Biomedical Engineering II.

## References

1. Ivakhnenko, A.G.: Polynomial Theory of Complex System. IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-1, No. 4, Oct. 1971, pp. 364-378.
2. Farlow, S.J.: Self-Organizing Methods in Modelling. GMDH Type Algorithms. Marcel Dekker, Inc., New York, 1984.
3. Tamura, H., Kondo, T.: Heuristics-free group method of data handling algorithm of generating optimal partial polynomials with application to air pollution prediction. Int. J. Systems Sci., 1980, vol. 11, No. 9, pp. 1095-1111.
4. Ivakhnenko, A.G., Müller, J.A.: Present State and New Problems of Further GMDH Development. SAMS, Vol. 20, 1995, pp. 3-16.
5. Ivakhnenko, A.G., Ivakhnenko, G.A., Müller, J.A.: Self-Organization of Neural Networks with Active Neurons. Pattern Recognition and Image Analysis, Vol. 4, No. 2, 1994, pp. 177-188.
6. Ivakhnenko, A.G., Wunsch, D., Ivakhnenko, G.A.: Inductive Sorting/out GMDH Algorithms with Polynomial Complexity for Active neurons of Neural network. IEEE 6/99, 1999, pp. 1169-1173.
7. Nariman-Zadeh, N. et al.: Modelling of Explosive Cutting process of Plates using GMDH-type neural network and Singular value Decomposition. Journ. of material processes technology Vol. 128, 2002, No. 1-3, pp. 80-87.
8. Hiassat, M., Mort, N.: An evolutionary method for term selection in the Group Method of Data Handling. Automatic Control & Systems Engineering, University of Sheffield, [www.maths.leeds.ac.uk/statistics/workshop/lasr2004/Proceedings/hiassat.pdf](http://www.maths.leeds.ac.uk/statistics/workshop/lasr2004/Proceedings/hiassat.pdf).
9. Oh, S.K., Pedrycz, W.: The Design of Self-organizing Polynomial Neural Networks. Information Sciences (Elsevier), Vol. 141, Apr. 2002, No. 3-4, pp 237-258.
10. F.Hakl, M.Jirina, E.Richter-Was: Hadronic tau's identification using artificial neural network. ATLAS Physics Communication, ATL-COM-PHYS-2005-044, last revision: 26 August 2005, <http://documents.cern.ch/cgi-bin/setlink?base=atlnot&categ=Communication&id=com-phys-2005-044>.
11. Negative Selection Algorithms: From the Thymus to V-Detector. Dissertation Presented for the Doctor of Philosophy Degree. The University of Memphis, August, 2006.
12. Guney K., Akdagli A., Babayigit B.: Shaped-beam pattern synthesis of linear antenna arrays with the use of a clonal selection algorithm. Neural Network World, Volume 16 (2006), pp. 489-501.
13. Merz, C.J., Murphy, P.M., Aha, D.W.: UCI Repository of Machine Learning Databases. Dept. of Information and Computer Science, Univ. of California, Irvine, <http://www.ics.uci.edu/~mllearn/MLrepository.html>, 1997.
14. R. Paredes, E. Vidal, Learning Weighted Metrics to Minimize Nearest-Neighbor Classification Error. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 7, July 2006, pp. 1100-1110.

