# Genetic Selection and Cloning in GMDH MIA Method

Marcel Jiřina[1], Marcel Jiřina, jr.[2]

[1] *Institute of Computer Science, Pod vodarenskou vezi 2, 182 07 Prague 8 – Liben, Czech Republic*

*2 Faculty of Biomedical Engineering, Czech Technical University in Prague, Zikova 4, 166 36, Prague 6, Czech Republic*

marcel@cs.cas.cz jirina@fbmi.cvut.cz

**Abstract.** *The GMDH MIA algorithm is modified by the use of selection procedure from genetic algorithms and including cloning of the best neurons generated to get even lesser error. 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 slight modification of parameters of copies of the best neuron, i.e. neuron with the largest fitness. We describe the algorithm and show that the procedure is relatively simple. The genetically modified GMDH network with cloning (GMC-GMDH) can outperform other powerful methods. It is demonstrated on some tasks from Machine Learning Repository.*

## Keywords

Inductive modeling, GMDH MIA algorithm, genetic algorithm, genetic selection, cloning, artificial neural networks.

## 1 Introduction

In this paper we solve difficult classification tasks by the use of GMDH MIA, i.e. group method data handling multilayer iterative adaptive method, genetically modified (GM) by selection algorithm common in genetic algorithms and by cloning the best neurons generated up to given a moment of learning process. We denote the genetically modified GMDH with cloning as GMC-GMDH.

The new GMC-GMDH algorithm has no layered structure. When a new neuron is added it is connected to two already existing neurons or to network inputs randomly with probability proportional to fitness. From the best neuron found the clones are derived to reach even better fitness.

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 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 $z$):

$$z = a_1 + a_2x + a_3y + a_4x^2 + a_5y^2 + a_6xy \,, \tag{1}$$

where coefficients *a*,..*f* are determined statistically and unique for each transfer function. The coefficients can be thought of as analogous to weights found in other types of neural networks.

The network of transfer functions is constructed 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 grow of a number of neurons in a layer, after finishing the layer, limited number best neurons are selected and the others removed from the network [3].

The work with populations of GMDH networks is a rather complex and time-consuming task. Very interesting and principally simple, perhaps the simplest approach published Hiassat and Mort in 2004 [8]. Their method does not work with population of networks but individual neurons are individuals and can be considered as a population. At the same time, they do not remove any neuron during learning and keep the layered structure. 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 genetic 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.

In difference to [3] and [8] in our new method there is only a single layer which grows during learning one neuron at a time. At the same 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 neuron which was found to be the best neuron up to now generated. Inexact copies follow from the fact that to have 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.

The new approach consists of five basic parts or steps, the standard quadratic neurons of standard GMDH MIA algorithm, a fitness function based on reciprocals of mean error, the selection procedure of choosing two parent neurons, adding a new neuron forming a single layer only, and finding its six parameters using minimal mean squared error algorithm. To stop the process of new neurons generation, some stopping rules are proposed.

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 Genetic selection in GMDH MIA

Here we describe the approaches we use for constructing our genetically modified GMDH network with cloning.

### 2.1 The learning set

We assume the *n*-dimensional real valued input and a one-dimensional real valued output. The learning set consists of $n + 1$ dimensional vectors $(x_i, y_i) = (x_{1i}, x_{2i}, \ldots x_{ni}, y_i)$, $i = 1, 2, \ldots N$ 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 tuning set), and $N_V$ is the number of rows used for error evaluation during learning (the verification set).

### 2.2 New genetically modified GMDH network algorithm

The standard quadratic neuron of the GM GMDH network is an individual. 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 of 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 probability to be a parent is proportional to 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 selection 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 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 GM GMDH is, in fact, no crossover in the sense combining two parts of parents' genomes. In our approach eq. (1) gives symmetrical procedure of mixing the parents' influence but not their features, parameters.

### 2.3 Selection procedure

In the 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 use $p_0 = 0.1$. 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:

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

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

The computation of six parameters of the new neuron is the same as in the GMDH MIA algorithm.

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

### 2.4 Best neuron

A new neuron added need not be better than all others. 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, i.e. the best network's output. After learning, this output is used as a network output in the recall phase.

### 2.5 Pruning

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

## 3 Immunity-based Models, Cloning

Note first that authors dealing with artificial immune systems, e.g. [12] use different terminology than 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 various mechanisms or processes in the immune system which are investigated in the development of artificial immune systems (AIS). A comprehensive summary can be found in [11].

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

```
BEGIN
      Given the Best GMDH Neuron with parents (i.e. input signals
      from) In1, In2 and with six parameters a, b, .. f.
      REPEAT
            Produce a copy of the Best Neuron. A copy has the same
            inputs In1 and In2 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 the 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 the Best Neuron.

The simplest approach spreads value of each parameter randomly around its value in the Best Neuron. For each parameter one can use normal distribution with mean as well as standard deviation equal to the original value of the parameter. In mean the spread would be –100, +100 percent for each parameter. It was found to be too large especially when changing all six parameters. Then we reduce

the spread, i.e. the standard deviation, to 1/6 of parameter's value and, at the same time, use a multiplicative constant, which a user can set up appropriately; default value is 1.

# 4 Error development and stopping rule

From the new strategy of network building in genetically modified GMDH method with cloning (GMC-GMDH) described above follows also stopping rule different from searching for minimal error on the validating set like in original GMDH MIA method. In our case error on the validating set for the best neuron monotonously decreases having no minimum. On the other hand, the indexes of the successive best neurons became rather distant. At the same time, the error of best neurons lessens by lesser and lesser amount. The process can be stopped either when very small change in error is reached, or too many new neurons are built without appearance of a new best neuron or when predefined large number of neurons is depleted.

# 5 Experiments

Experiments described below show that the GMC-GMDH approach outperforms 1-NN method in most cases, in many cases outperforms naïve Bayes method and also the *k*-NN method where *k* equals to the square root of the number of training set samples.

The classification ability of genetically modified GMDH algorithm with cloning was tested using real-life tasks from UCI Machine Learning Repository [13]. We do not describe these tasks in detail here as all can be found in [13]. For each task the same approach to testing and evaluation was used as described in [13]. In Table 1 the results are shown together with results for other standard methods. For running GMC-GMDH program default parameters were used as follows for all tasks: Stop computation after 10000 neurons was generated. Probability that new neuron's input is one of input signals is 10 %, probability that new neuron's input is one of already existing neurons is 90 %. Maximal number of clones generated from one parent neuron is limited to int(sqrt(No. of neurons generated up to now)). For all methods optimal threshold for minimal error was used.

Classification errors for four different methods including the GMC-GMDH are summarized in Fig 1. The methods for comparison are

1-NN – 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

Bay1 – the naïve Bayes method using ten bins histograms.

LWM1 – the learning weighted metrics method [14] modified with nonsmooth learning process.

GMC-GMDH – the new method proposed.

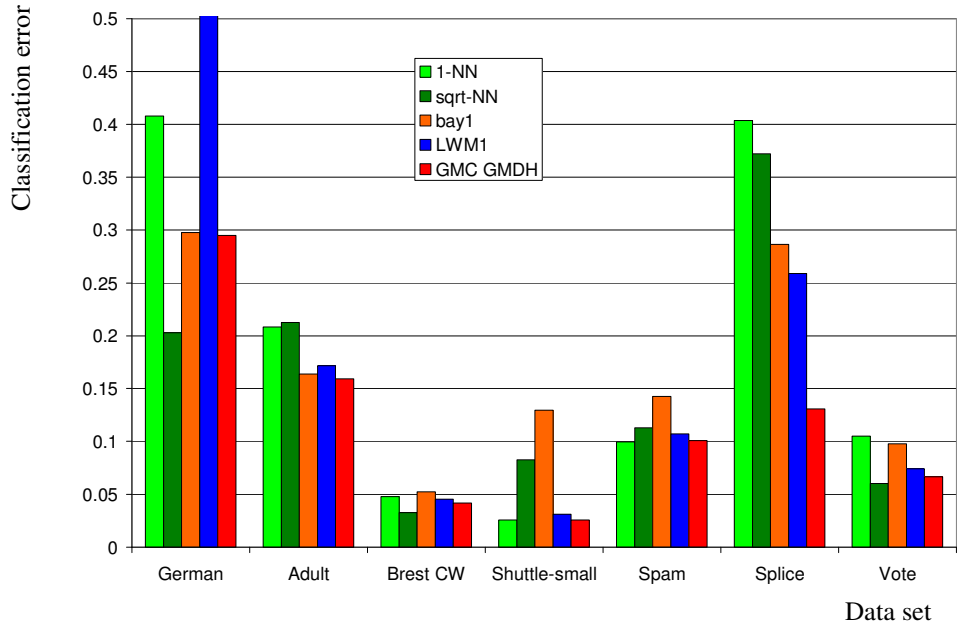


Fig. 1. Classification errors for four methods on some data sets from UCI MLR.
Note that for the task Shuttle-small all errors are ten times enlarged.
In the black and white print the items in the legend from top to bottom correspond to individual columns for each data set from left to right in the same order.

# 6 Conclusions

The Genetically modified GMDH method presented here is an elegant approach how to improve 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. The layered structure of the GMDH algorithm disappears as any new neuron can be connected to the output of any already existing neuron or even to any input of the network.

Moreover the idea of genetically modified GMDH neural network is extended by cloning. Clones are close but not identical copies of original individuals. An individual in our case is the best neuron just generated. Intuition behind says that even when 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 clones have mutated, i.e. slightly changed parameters of the parent individual, the best neuron. We found the cloning with large parameters changes has small effect, but with small changes a new best neuron often arises.

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, it has no critical parameters to be tuned, and its computational complexity governs before all linear regression with data of the learning set. As there is no searching or sorting like in nearest neighbor-based methods, the GMC-GMDH is much faster than methods mentioned especially for large learning sets.

Important advantage of genetically modified GMDH with cloning is even less control parameters for learning than in standard GMDH MIA algorithm, which is considered “parameter-less”, but one must set up number of best neurons selected in newly generated layer and thus indirectly control learning time and size of network. The only limitations of the GMC-GMDH method are learning time or memory size.

# 7 Acknowledgements

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. See also Farlow 1984 p. 225-241.
4. Ivakhnenko, A.G., Műller, 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., Műller, 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.
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 Degrese. 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/~mlearn/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.