Transactions of the VŠB – Technical University of Ostrava, Mechanical Series
No. 2, 2013, vol. LIX
article No. 1965

Martin MARIŠKA[*], Petr DOLEŽEL[**]

PIECEWISE-LINEAR NEURAL NETWORK – POSSIBLE TRAINING ALGORITHMS EFFICIENCY COMPARISON

PO ČÁSTECH LINEÁRNÍ NEURONOVÁ SÍŤ – POROVNÁNÍ EFEKTIVITY TRÉNOVACÍCH ALGORITMŮ

Abstract

In this article, a benchmark of algorithms for training of piecewise-linear artificial neural networks is introduced. At first, motivation of this article is described for a special topology of the neural network is used. This topology can be advantageously used in system control design, but it is difficult problem to train it. In this article, there is described a set of possible training algorithms, these algorithms are tested and evaluated. Benchmarking data are based on real problems.

Abstrakt

V článku je představen benchmark několika trénovacích algoritmů pro učení umělé neuronové sítě s po částech lineárními aktivačními funkcemi. V první části článku je představena použitá topologie neuronové sítě a její využití, dále jsou pak popsány možné algoritmy učení a tyto algoritmy jsou pak testovány a porovnány. K testování jsou použita data reálných procesů.

Keywords

piecewise-linear neural network, benchmark, machine-learning algorithms, process control

1 INTRODUCTION

An artificial neural network (ANN) is an adaptive mathematical structure that reorganizes and changes its structure based on external or internal information that flows through the network. Nowadays, it is especially used for modeling of complex nonlinear relationships between input and output datasets or decision making tools. The ANN is now widespread through plenty of scientific domains. The ANN models have been found useful and efficient, particularly in problems for which the characteristics of the processes are difficult to describe by physical equations.

2 MOTIVATION

A special topology used for linearization of the nonlinear model exists for the ANN. This approach can be used for process control and detailed methodology is described in [1]. The topology itself is defined in following way: suppose feed-forward ANN with one hidden layer that can have only one neuron in the output layer. Besides, it has linear saturated activation functions in hidden layer and linear activation function in output layer (see Fig. 1). Once any nonlinear problem is modeled by this kind of network, it is possible to divide it into a set of linear subproblems where each of them can be solved by some effective algorithm.

[*] Ing., Department of Process Control, Faculty of Electrical Engineering and Informatics, University of Pardubice, Náměstí Čs. legií 565, Pardubice, e-mail mariska.martin@gmail.com
[**] Ing., Ph.D., Department of Process Control, Faculty of Electrical Engineering and Informatics, University of Pardubice, Náměstí Čs. legií 565, Pardubice, e-mail petr.dolezel@upce.cz

However, the methodology does not have any recommendation about the machine-learning algorithms. Methodology only describes that approximation quality of the topology is given by quality of training. Therefore, the problem is in speed and performance of the machine-learning algorithm. The purpose of this contribution is to identify the fastest general purpose algorithm that can be used for training of the ANN with linear saturated activation functions in hidden layer.

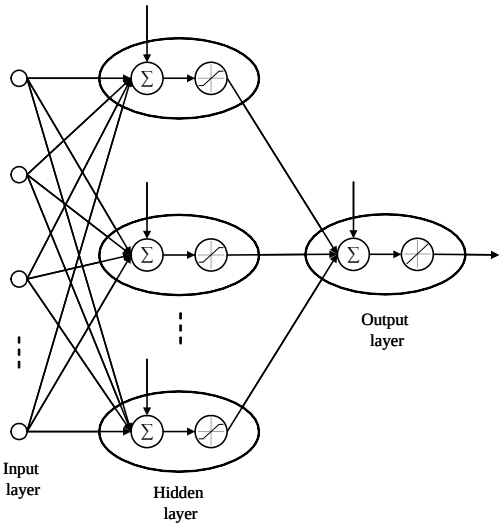

Fig. 1 Typical structure of piecewise–linear neural network

3 ALGORITHMS

For training of the ANN, the supervised machine-learning algorithms are selected because the input and output data are always known. Most of the machine-learning algorithms use some gradient-based optimization technique. Thus, these algorithms require analytical derivative of the activation functions. The ANN topology uses the linear saturated activation functions in hidden layer due to piecewise-linear modeling. The linear saturated function is not differentiable at starting point of saturation so the derivative function is replaced in following tests by derivative function of hyperbolic tangent function because of their similar course. Brief information about selected benchmarked algorithms is below:

- Levenberg–Marquardt (LM) – the algorithm that combines the advantages of gradient-descent and Gauss-Newton methods. Algorithm is described in [2], [3] and in this implementation of the algorithm, there is is added Bayesian regularization to overcome the problem in interpolating noisy data [4].
- Scaled Conjugate Gradient (SCG) – the algorithm based on conjugate directions but it does not perform a line search at each iteration. For more details see [5].
- Resilient Propagation (RPROP) – the algorithm based only on change of the sign of the partial derivative over all patterns (not the magnitude), and it acts independently on each "weight". See [6], [7].
- Quick Propagation (QP) – the algorithm based loosely on Newton's method but fundamentally it is more heuristic than formal. It makes two risky assumptions. At first, the error vs. weight curve for each weight can be approximated by a parabola whose arms are opened upward. At second, the slope change of the error curve, as seen by each weight, is not affected by all other weights which are changing at the same time. More in [8].

4 BENCHMARK

For measuring performance, the Caliper was used (Caliper is Google's open-source framework for java). Framework handles a lot of inconveniences and inaccuracies. The main idea is to measure the speed of the training in time units. Training speed in time units depends on the speed of convergence, computational demands or other performance characteristics. For comparative reasons the algorithm that defines same rules for all measurements is constructed (see Fig. 2).

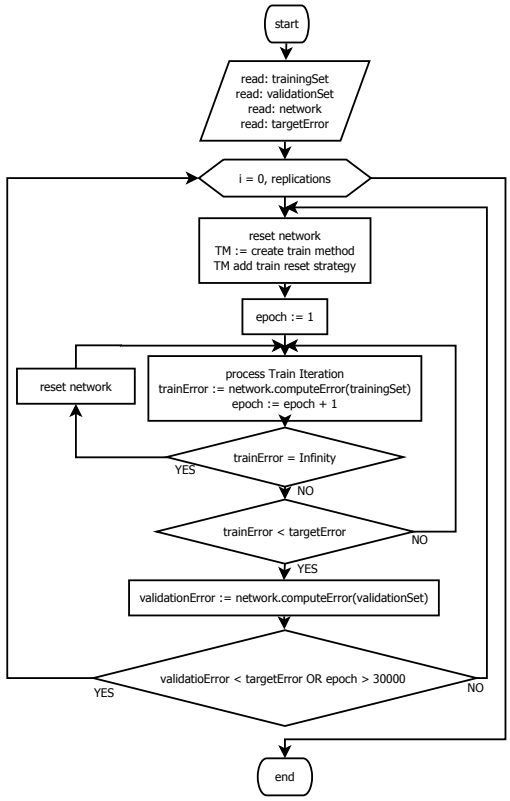

Fig. 2 Algorithm in flowchart for measure function

One of the most important parameters is target error. The target error determines escape condition for training. If calculated error from validation set is less than target error, the training will end. Measure algorithm uses important strategy (this strategy is not explicitly mentioned in flowchart, but it is a part of benchmarked function). If training error is not less than target error after 500 iterations (this is coefficient in benchmark), network weights are reset to new values. This strategy helps to set the appropriate initial values of weights and helps to speed up training.

In flowchart, there is some operation that needs an explanation. The operation "reset network" means resetting the weight matrix and the bias values by Nguyen – Widrow randomizer [9]. Input parameters for measure are training and validation data, network, targeted error and number of replications.

The standard Mean Square Error (MSE) is used to determine error (1).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(i_i - a_i)^2, \tag{1}$$

where:

MSE – mean square error,

i_i – expected value,

a_i – actual value,

n – number of outputs,

i – index of output value.

5 BENCHMARK DATA SETS

Most contributions present the performance results of the algorithms only for a very small number of problems. In most cases, less than three problems are presented and one or several of these problems are meaningless synthetic problems. One of the reasons could be that it is difficult to get data for real problems. For this paper a subset of benchmark problems from Proben1is used. The Proben1 is a set of standard datasets for the ANN evaluation based on real problems. Brief explanation of chosen datasets is below:

- Cancer (classification problem) – Diagnosis of breast cancer. Try to classify a tumor as either benign or malignant based on cell descriptions gathered by microscopic examination.
- Glass (classification problem) – Classify glass types. The results of a chemical analysis of glass splinters (percent content of 8 different elements) plus the refractive index are used to classify the sample to be either float processed or non float processed building windows, vehicle windows, containers, tableware, or head lamps.
- Heart (classification problem) – Predict heart disease. Decide whether at least one of four major vessels is reduced in diameter by more than 50%. The binary decision is made based on personal data such as age, sex, smoking habits, subjective patient pain descriptions, and results of various medical examinations such as blood pressure and electro cardiogram results.
- Thyroid (classification problem) – Diagnose thyroid hyper- or hypofunction. Based on patient query data and patient examination data, the task is to decide whether the patient's thyroid has overfunction, normal function, or underfunction.
- Flare (approximation problem) – Prediction of solar flares. Try to guess the number of solar flares of small, medium, and large size that will happen during the next 24-hour period in a fixed active region of the sun surface. Input values describe previous activity and the type and history of the active region.

The topology requires only one neuron in output layer. Some datasets have more than one output. In these cases outputs are transformed to only one value by (2).

$$output = \sum_{i=1}^{n} ideal_i \cdot 10^i, \tag{2}$$

where:

$output$ – output layer value,

$ideal_i$ – output ideal value,

n – number of ideals outputs,

i – index of ideal output.

Where n is the number of ideal outputs and ideal is the appropriate dataset output. For each input values normalization to interval <–1, 1> is used. For more detailed information about datasets see [10].

6 RESULTS

Each experiment is usually measured in 7 trials and if needed for required standard deviation accuracy 5 more trials can be additionally measured. The threshold of standard deviation is at least one digit place lower than measured result. The trials are executed with replication parameter. The number of replications changes from 500 to 1000 and it depends on the speed of executed code in benchmark function.

All results are introduced in Tab 1 and they are presented in milliseconds. All results are measured on computer with this hardware configuration: Intel Core i5 2.53GHz, 4GB RAM, Windows 7 x64 and model: Acer Aspire 5820TG.

Some train methods converge slowly, therefore values above the threshold of 2000 milliseconds are rather approximate to real values for they cannot be measured with required accuracy because of time and performance issues. This problem appears especially in HEART and THYROID datasets.

Tab. 1 Benchmark results

Parameters		**Results [ms]**			
Dataset	**Target Error**	**RPROP**	**QP**	**SCG**	**LM**
CANCER	0,15	2,42	2,08	40,7	539.0
	0,10	2,89	2,47	64,7	628.0
	0,05	3,95	3,39	91,4	889,2
GLASS	0,15	0,84	0,47	0,80	196,0
	0,10	0,92	0,46	15,46	297,9
	0,05	1,35	1,30	27,8	805,7
FLARE	0,15	4,34	1,18	3,02	2094
	0,10	5,38	1,18	3,13	2036
	0,05	7,79	1,41	3,7	2016,7
THYROID	0,15	39,78	78,59	2866,3	3128,1
	0,10	42,04	89,29	2830,7	3342,1
	0,05	63,83	79,75	4427,1	83617
HEART	0,15	15,65	10722	18,6	103788
	0,10	22,67	19586	2055,1	206405
	0,05	42,97	57324	8716,4	186146

QP has best results in three types of datasets (cancer, glass, flare). On the other hand, it has notably bad results in heart dataset. If we consider that the Rprop is nearly as fast as QP and faster in results from heart and thyroid, it could be considered as the best general purpose algorithm. Rprop requires less adjustment of parameters than QP and hence Rprop is more stable than QP.

7 CONCLUSIONS

The article is focused on indentifying the best machine-learning algorithms for feed forward artificial neural network with linear saturated activation functions in hidden layer. Benchmark's results prove and show that the best training method for this type of the ANN's topology is the Quick Propagation but it is not applicable for all types of datasets. The best general purpose training method seems to be the Resilient Propagation. In addition, the benchmark shows that normally efficient LM algorithm is computationally more demanding than others. In benchmark comparison, the LM is significantly slower than training algorithms preferred by this contribution.

REFERENCES

[1] DOLEŽEL, P. ; TAUFER, I. Piecewise-linear artificial neural networks for PID controller tuning. *Acta Montanistica Slovaka*, 2012, XVII, Nr. 3, pp. 224-233. ISSN 1335-1788.

[2] LEVENBERG K., A method for the solution of certain problems in least squares. *Quart. Appl. Math.*, II, pp. 164–168, 1944.

[3] MARQUARDT D. W., An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math.*, XI, pp. 431–441, 1963. ISSN 0368-4245.

[4] FORESEE D. F., HAGAN M.T., Gauss-newton approximation to bayesian learning, In *International Conference on Neural Networks.* New Jersey : IEEE, 1997, pp. 1930-1935. ISBN 0-7803-4122-8.

[5] MOLLER M. F., A Scaled conjugate gradient algorithm for fast supervised learning, *Neural Networks*, 1993, VI, Nr. 4, pp. 525-533. ISSN 0893-6080.

[6] RIEDMILLER M., BRAUN H., A direct adaptive method for faster backpropagation learning: the RPROP algorithm, In *International Conference on Neural Networks.* New Jersey : IEEE, 1993, pp. 586-591. ISBN 0-7803-0999-5.

[7] IGEL C., HUSKEN M., Improving the Rprop learning algorithm, In *Proceedings of the Second International Symposium on Neural Computation*, Berlin : ICSC Academic Press, 2000, pp. 115-121. ISBN 3-906454-21-5.

[8] FAHLMAN S. E., *An Empirical study of learning speed in back-propagation networks*, Technical report CMU-CS-88-162, Pittsburgh : Carnegie Mellon University, 1988.

[9] NGUYEN D., WIDROW B., Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights, In *Proceedings of the International Joint Conference on Neural Networks*, Stanford : Stanford University, 1990, pp. 21-26. ISBN 0-8058-0775-6.

[10] PRECHELT L., A set of neural network benchmark problems and benchmarking rules, Technical Report 21/94, Karlsruhe : University of Karlsruhe, 1994.