5th International Conference
″Computational Mechanics and Virtual Engineering ″
COMEC 2013
24- 25 October 2013, Braşov, Romania

OPTIMIZATION OF AN ARTIFICIAL NEURAL NETWORK USED FOR THE PROGNOSTIC OF CANCER PATIENTS

Liviu GAIŢĂ[1], Manuella MILITARU[1]
University of Agronomic Sciences and Veterinary Medicine of Bucharest, Romania, clinica@ortovet.ro

Abstract: *An artificial neural network model was developed for providing an estimation of the survival time for cancer patients. Data from 31 dogs and cats treated as oncologic patients was used for training and validating the network which used 28 elementary predictors selected from clinical and para-clinical, macroscopic pathology, and histology information. The network was optimized to avert the overfitting, which blocks the learning process. Among the techniques tested and illustrated are: random noise in synapses, automatic cloning of well performing neurons, extended learning with/without jitter, jogging of connection weights, freezing of weights and biases, pruning of less performing nodes, cross-validation and bootstrapping methods for the selection of training and validation data sets. Once overfitting is avoided, the model provides not only reliable predictions, but also an identification of the most effective predictors.*
Keywords: *pathology, neural networks, oncology, survival time, veterinary*

1. INTRODUCTION

Neural networks (NN), in the current acceptance of the term, were first proposed by Rosenblatt in 1958, who described several variants of a *perceptron*, able to learn to solve some very simple classification tasks. In 1969, Marvin Minsky and Seymour Papert made a substantiated argument for the weaknesses of the perceptron model and hence the research on the subject fel into disgrace for the following 20 years. It was not before 1986 that David Rumelhart, Geoffrey Hinton, and Ronald Williams re-vitalised the idea that NN have a good potential to solve real problems. They proved the effectiveness of error back-propagation training algorithms for multilayer feedforward networks.

Masters described [1] some features of the problems for which neural networks are prone to outperform other tools:
- when the input data are, to a significant degree, „fuzzy" or subject to possibly large error: human opinions, ill-defined categories
- when patterns searched for are deeply hidden in a large amount of data
- when unpredictable nonlinearity is present
- when chaotic (in the mathematical sense) dynamics is affecting to a significant extend the analysed data.

The last 20 years brought a large amount of research on foundation and developments on NN [2] along with practical applications that made them regular tools in some areas: *recognition and refinement of patterns*, forms, objects from noisy data, *classification tasks* allowing diagnostic and alert in medical pathology and in law enforcement electronic monitoring work, in satellite mapping, in radar and sonnar scanning, *prediction* for medical prognostic, for meteorology, capital markets and other economic models, for educational and other social behaviour models.
Medical applications of NN are continuously reported [3], lately in dedicated forums as well. The impressive development of customised statistical tools for medical applications is being paralleled with various means to make the accumulated information readily available in practical situations. The Evidence Based Medicine approach creates a strong demand for such means and NN provide outstanding capacity to respond to such expectations [4].
Very little was reported on the use of NN in veterinary pathology. This study provides details on a phase in the development of a comprehensive NN model for the prognostic of cancer patients treated in a clinic for small animals. The main obstacle in making the model operational was the occurrence of over-fitting. This paper presents the work done to overcome that situation.

2. MATERIALS AND METHODS

The NN model was based on a training and validating data set describing 39 cancer patients, dogs and cats, treated in ORTOVET veterinary clinic in 2004-2005. Histological diagnostic was done in the Pathology Laboratories of FMV Bucharest and of "Dimitrie Gerota" Hospital in Bucharest. The tumor types in the group were [5] mammary gland tumours (18 cases), along with osteosarcoma, cutaneous lymphoma, haemangioma, ovarian tumours, malignant melanoma, histiocitoma, and carcinoma of various types and locations. 28 predictors were quantified, by a 0 to 4 score, for each patient:

-*clinical and paraclinical criteria*: tumour growth dynamics, post-surgery tumour-free lifetime, life span post-diagnostic, impact on blood work, resolution time for the surgical wound, age and TNM grade when the treatment started

-*macroscopic pathology criteria*: size of tumour at diagnostic time, hardness, presence of blood vessels, ulcerations

-*histology pathology criteria* at tumour level and at regional lymph node level

-*therapy criteria*: presence/absence of surgical treatment, chemotherapy, immune support therapy, hormonal therapy, NSAID therapy.

The EasyNN Plus v. 15.0a software, from Neural Planner Software was used to setup and train a neural network aiming at providing an estimation of the survival time. Hence survival time was set as output and all the other predictors as inputs.

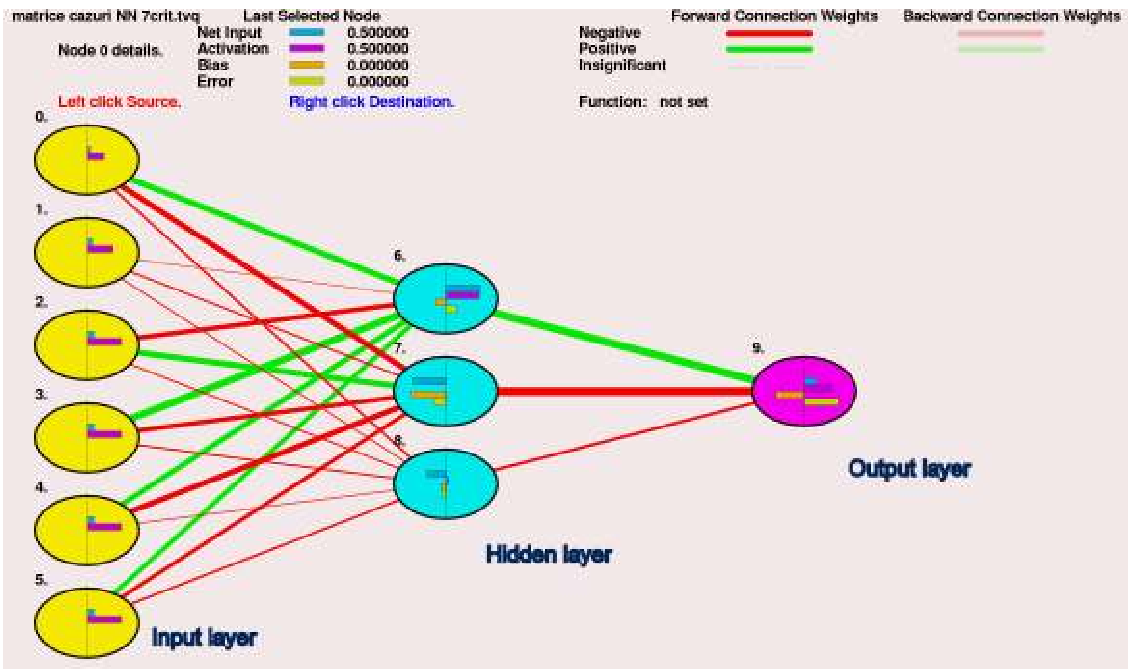

Figure 1: Basic configuration of a neural network

An illustration of the basic configuration of the NN is presented in Figure 1. The model is very flexible, allowing inter alia backward connections and manual and automatic set up of each weight w_{ij} from unit j to unit i as well as the possibility to add variable level of random noise to the activity function of each neuron:

$$y_i = f_i(\sum_{j \in A_i} w_{ij} y_j + b_i) \quad (1)$$

where:

$$A_i = \{j : \exists w_{ij}\} \quad (2)$$

is the set of nodes anterior to unit i and b_i is the so called bias in the i node. The basic form of f_i is **tanh**.

The set of examples is divided in a large set of learning (training) cases and a limited number of validating cases. Learning is essentially done by back-propagating the error at the level of the output nodes to each of the nodes in the hidden layers and then optimizing the weights to minimize error. Validation is used to test the performance of the network on cases outside the learning set of cases and make changes of the design of the network

(adding/pruning weights or nodes, adding noise to the activation function in hidden or input nodes, changing biases and so on) to improve that performance.
As all literature on the matter indicates, overfitting was the main problem for our NN model as well, as shown in Figure 2: learning error gets down towards zero but the validation error fails to improve, even diverge.

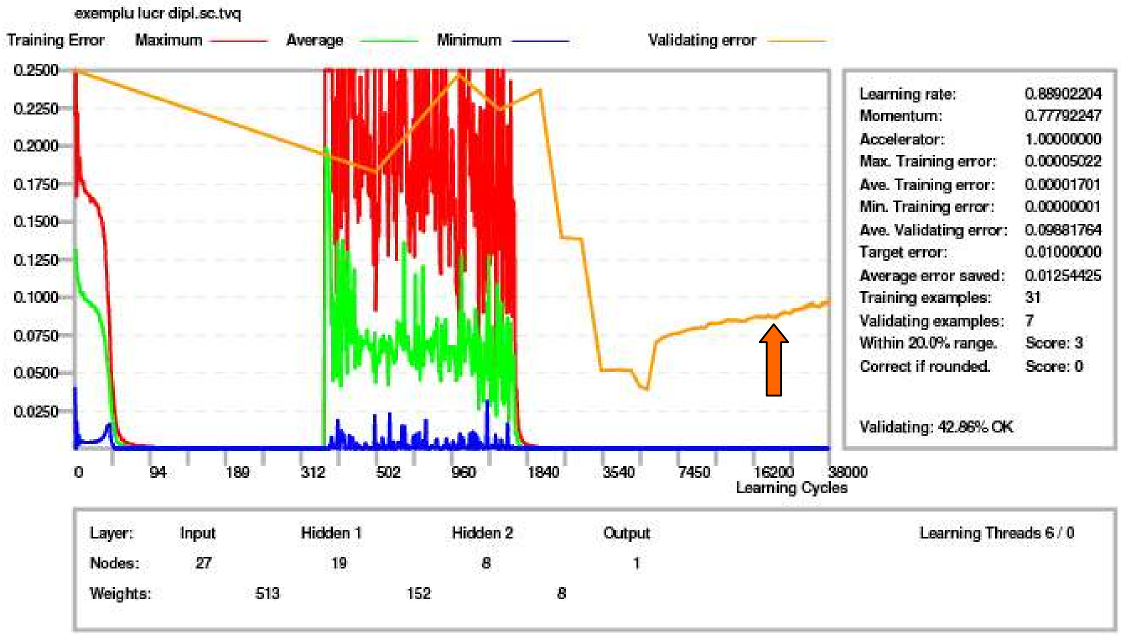

Figure 2: Overfitting is expressed by a low learning error but an increasing validating error

We present here some of the techniques tested for overcoming this situation. There are many theoretical attempts to develop effective strategies for sorting out the matter, but none proved to have universal application. Therefore, the use of various tools is, to a large extend, an empirical trial and error endeavor [8].

3. RESULTS AND DISCUSSION

Initial experience gathered in using some of tools proposed in the literature for sorting out the overfitting problem were mentioned in [6]. Further systematic work has been carried out on the matter and some initial assessments had to be adjusted, while others were confirmed.

3.1 Cloning of nodes

When, during the learning process, the learning error fails to decrease and complexity of the dataset is suspected as cause, cloning of well-performing hidden nodes can be considered. After evaluating the performance of the nodes, the hidden node contributing to learning and with greatest net input is cloned, then frozen. Learning is resumed and the freeze level is progressively decreased, with the aim of having both the original and the clone contributing to learning. The procedure is repeated for other hidden nodes selected by the same criteria.
Cloning proved to be very effective in improving the performance of the network and avoiding overfitting. It was most effective when combined with pruning underperforming weights and nodes. **Figure 3** illustrates how the reduced network in **Figure 1** developed to a more complex one, which avoids overlearning. The network is a variant of our model with only 6 predictors.

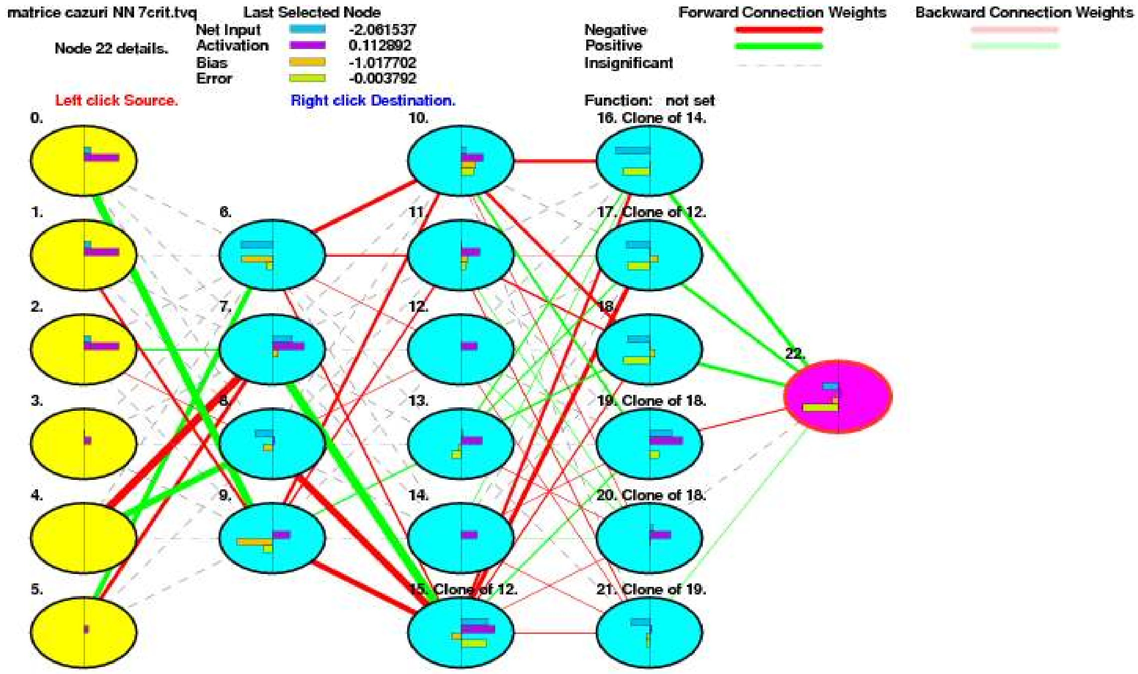

Figure 3: Development of a well performing network by cloning and prunning

3.2 Random noise

Noise, as a small random perturbation, can be added to activation both to the input nodes (usually called *jitter*) and to the hidden nodes. Using noise in the learning process has as theoretical basis the condition of the feedforward NN approximation function to be piecewise continuous (although it can accommodate discontinuities). For jitter, the argument is intuitive: the same target cases may be used for training not only starting from the known point in the input space, but also from many others in its neighbourhood. How close that neighbourhood is chosen is critical to make jitter help and not destroy the learning process. On the very irregular surface of the cost function, added noise can be useful in avoiding trapping in local minima or plateaus, but can also lead to bouncing around and missing the global minimum, if that is on the bottom of a steep depression.
In our model, noise proved useful only to overcome stalling during learning. It never helped improve validation and most often, regardless its level, increased the validation errors.

3.3 Bootstrapping and cross validation

We tested K-fold cross-validation, leave-one-out cross-validation, and bootstrapping as methods for resampling the available data, which needed to be used both for training the network and for validating its performance. In spite of what is suggested in [7], with solid theoretical substantiation [7], in our limited number of tests we found no significant benefit from bootstrapping over the other methods.

3.4 Confirmed performance

Once the overfitting problem is sorted out, the neural network has reasonable prediction accuracy and flexibility (**Figure 4**) to accommodate new data sets quite different from the training set, even incomplete data sets as input for enquiry.
Information from cases used initially for an enquiry on the network on the estimated survival time, turns later, when the survival time becomes a known fact, into an additional case in the learning set for the model. The model is dynamic and its use leads to changes in quality of performance, as reflected in the learning error and in the validation error.

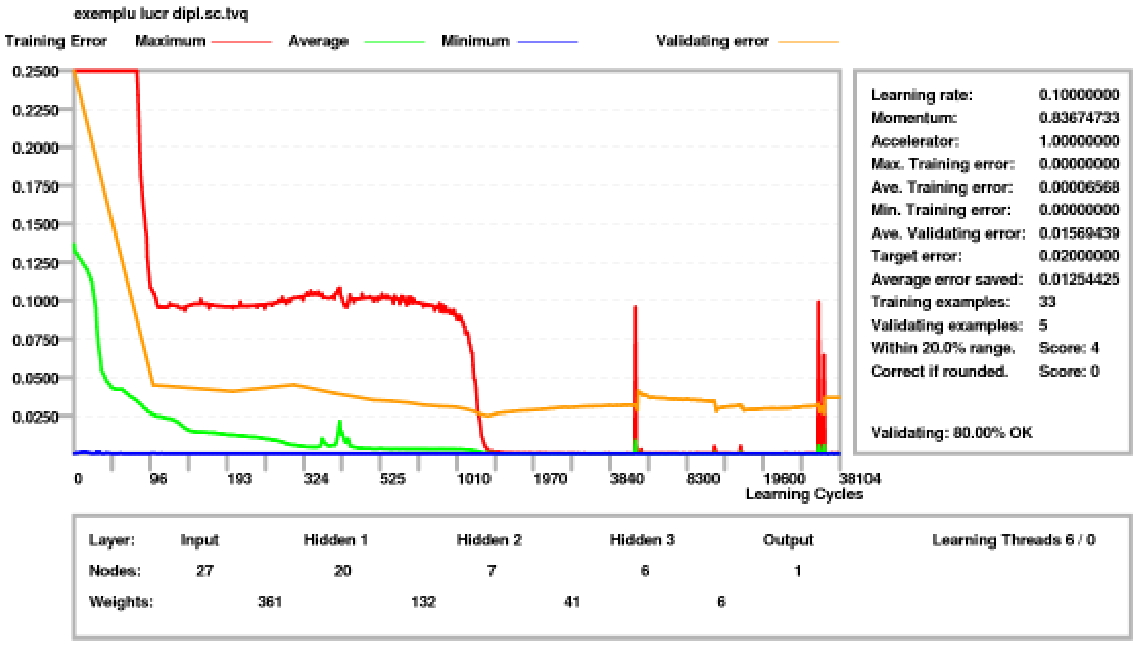

Figure 4: Optimized network: excellent training error and acceptable validating error

One consequence of the dynamics of the model is that some useful information, which can be extracted from the network beyond the output prediction, such as the relative importance of predictors, can change in time with the progress of learning. As compared to [6], a slightly different set of most important predictors emerged (**Table 1**). Basically, the importance reflects the weights emerging from the respective input node.

Table 1: Relative importance of predictors (top 5)

Rank of importance	Predictor	Relative importance
1	Tumor-free time post-surgery	9.220023
2	TNM score	6.516533
3	Tumour size	5.900695
4	Deviation of complete blood count from normal range	4.80634
5	Sclerosis/conjunctive invasion of tumour	4.441382

Another set of information, which reflects the characteristics of the model but has also significance for the medical practitioner using the model is the sensitivity of the outcome, i.e. the survival time, to each predictor (**Table 2**). It is critical to remember, though, that the output of the NN is a nonlinear and even non-monotone function of the inputs, so the sensitivity results are a global measure of the entire training data set that can obscure important local variations.

Table 2: Relative sensitivity of the survival time to predictor (top 5)

Rank	Predictor	Relative sensitivity
1	Tumour-free time post-surgery	0.923891
2	Tumour size	0.796161
3	TNM score	0.793433
4	Deviation of complete blood count from normal range	0.718756
5	Imunostimulating treatment	0.657432

4. CONCLUSIONS

Overfitting was confirmed as being the main difficulty in making operational a neural network model developed for estimating the prognostic of cancer patients in a small animal veterinary clinic. Among the techniques tested for overcoming overfitting, cloning of hidden nodes combined with adaptive freezing of weights and pruning of underperforming nodes and synapses proved to be the most effective tools. Random noise in activation functions of input and hidden nodes, along with jogging of connection weights were useful only for overcoming lack of progress in the learning process, but that was found to be a rare occurrence.

REFERENCES

[1] Masters T., Practical Neural Network Recipes in C++, Academic Press, 1993

[2] Pourret O., Naim P., Marcot B., Bayesian Networks, A Practical Guide to Applications, John Wiley & Sons, 2008

[3] Dayhoff, J, DeLeo J, Artificial Neural Networks – Opening the Black Box, Cancer, Vol. 91, No. S8, pp 1615-1635, 2001

[4] Sordo M., Introduction to Neural Networks in Healthcare, OpenClinical, 2002

[5] Gaiţă L., Militaru M., Lihet M., Pilot cancer prognostic model based on pathology laboratory results processed in artificial neural networks, Contributions of scientific research to the progress of veterinary medicine, Simposium FMV-USAMV Bucharest, 2011

[6] Gaiţă L., Militaru M., Pathology and Clinical Data Integrated in Artificial Neural Networks for Providing Prognosis in Cancer, Journal of Comparative Pathology, Vol. 148, Issue 1, January 2013

[7] Ye F., Zhao A., What You See May Not be What You Get – A Brief Introduction to Overfitting, Cancer Biostatistic Workshop, Vanderbilt-Ingram Cancer Center, April 16th 2010

[8] Panchal G., Ganatra A., Shah P., Panchal D., Determination of Over-Learning and Over-Fitting Problem in Back Propagation Neural Network, International Journal on Soft Computing, Vol.2, No.2, May 2011