An Efficient Method for Selecting the Optimal Structure of a Fuzzy Neural Network Architecture

Bojan Novak

University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor, Slovenia

The fusion of artificial neural networks with soft computing enables to construct learning machines that are superior compared to classical artificial neural networks, because knowledge can be extracted and explained in the form of simple rules. An efficient method for selecting the optimal structure of a fuzzy neural network architecture is developed. The Vapnik Chervonenkis (VC) dimension is introduced as a measure of the capacity of the learning machine. A prediction of the expected error on the yet unseen examples is estimated with the help of the VC dimension. The structural risk minimization principle is introduced for constructing the optimal architecture with the lowest expected error for the small data sets. A comparison between fuzzy neural network and the neural network ARX model is presented.

Keywords: soft computing, learning theory, neural networks

1. Introduction

In 1958, Rosenblatt developed a biologically inspired learning machine simulated on the computer. Its name was Perceptron and it was able to solve a simple pattern recognition task and was able to generalize. A whole new field of learning machines appeared with the common name artificial neural networks. An effective method to describe the general principle of inductive inference in different machines was developed by Vapnik and Chervonenkis at the end of the 1960s. It is known as the empirical risk minimization (ERM) principle. At the beginning the theory was developed for pattern recognition but was later extended for function approximation, regression estimation, estimating the values of function at given points, estimating the function on the basis of indirect measurements and similar. The necessary and sufficient conditions for consistency of the ERM principle were first developed for the indicator functions (having 0,1 values). During the learning process the empirical risk is minimized on these indicator functions. This is the first necessary step. Second is to define theoretically as accurate as possible the bounds on the probability of the test error on yet unseen examples for the function minimizing the empirical risk.

The application of the ERM principle generates the best possible solution with the increasing number of examples only in cases where the uniform law of large numbers applies. Uniform law of large numbers is defined: the frequency of an event converges to the probability of this event with the increasing number of observations over all sets of events defined by indicator functions implemented by the learning machine. In the late 1960s Vapnik and Chervonenkis defined the conditions where the uniform law of large numbers held for a given set of events and the bounds on the nonasymptotic rate of uniform convergence. They introduced a capacity concept for the set of indicator functions – the VC dimension, which characterizes the variability of the set of indicator functions. The maximum number of different binary (values 0 or 1) partitioning of k samples is 2^k. The growth function is defined as

$$G(k) \leq k \ln 2. \qquad (1.1)$$

The distribution-independent condition for ERM

to have fast convergence is

$$\lim_{k\to\infty} \frac{G(k)}{k} = 0. \qquad (1.2)$$

If, for an indicator function, the expression (1.1) is valid for any k, then such a function is able to split any sample of arbitrary size, in all possible ways, or it is able to fit any data set with zero error. Later on, the well-known problem of over-fitting arises. A requirement for an indicator function is that after some finite value of k its growth is less than $k \ln 2$. This values is the VC dimension – h (VC = Vapnik Chervonenkis) [Vapnik et al. (1996), Vapnik (1998)]. Then the growth function is logarithmically bounded

$$G^{\Lambda}(k) = \begin{cases} k \ln 2 & \text{if } k \le h \\ h\Big(1 + \ln \frac{k}{h}\Big) & \text{if } k > h \end{cases} \qquad (1.3)$$

where $G^{\Lambda}(k)$ is a growth function of a set of indicator functions $Q(x, \alpha)$, $\alpha \in \Lambda$ and α represent a capacity ability and $Q(x, \alpha)$ presents convex penalty term. For example, it could be the order of the polynomial chosen from the finite set of orders Λ. In the case of real functions the VC dimension is bounded.

2. Algorithm Description

For the given k observations each consisting of a pair: $\mathbf{x}_i$, y_i, where $\mathbf{x}_i \in R^n, i = 1, \ldots, k$ is the input vector and y_i is the associated output. The learning machine is actually building up a mapping ability $\mathbf{x} \to f(\mathbf{x}, \alpha)$ where the functions $f(\mathbf{x}, \alpha)$ themselves are labeled by adjustable parameters α. The expectation of the test error for the trained machine is

$$R[f] = \int \frac{1}{2} L(y - f(\mathbf{x}, \alpha) dP(\mathbf{x}, y)). \qquad (2.1)$$

$R[f]$ is the risk functional. P is the probability and L presents loss function (could be in the form such as that in the (2.2)). The mean error rate measured on the finite number of observations is the "empirical risk"

$$R_{emp}(\alpha) = \frac{1}{k} \sum_{i=1}^{k} (y - f(x_i, \alpha))^2. \qquad (2.2)$$

$R_{emp}(\alpha)$ is fixed for a particular choice of α and for a particular training set $\{\mathbf{x}_i, y_i\}$ and the probability is not included in the equation. The expression $(y_i - f(\mathbf{x}_i, \alpha))^2$ is the loss function. The empirical risk minimization does not imply a small error on the test set if the number of examples in the training data set is limited. The structural risk minimization is one of the new techniques for handling efficiently a limited amount of data. For a probability $1 - \eta$, $\eta : 0 \le \eta \le 1$ the bound holds that depends on the parameter Φ which is defined as

$$\Phi\Big(\frac{h}{k}, \frac{\log(\eta)}{k}\Big) = \sqrt{\frac{h\Big(\log \frac{2k}{h} + 1\Big) - \log\Big(\frac{\eta}{4}\Big)}{k}}$$

$$R(\alpha) \le \frac{R_{emp}(\alpha)}{\Big(1 - \Phi\Big(\frac{h}{k}, \frac{\log(\eta)}{k}\Big)\Big)}. \qquad (2.3)$$

$R(\alpha)$ is the actual error on the previously unseen examples. The parameter h is the VC dimension [Schölkopf et al. (1995), Vapnik et al. (1996), Vapnik (1998)]. It describes the capacity of a set of functions implemented on the learning machine.

According to the eq. (2.3) risk could be controlled by two quantities: $R_{emp}(\alpha)$ and $h(\{f(\mathbf{x}, \alpha) : \alpha \in k_{sub}\})$, where k_{sub} is some subset of the index set k. The empirical risk R_{emp} depends on the choice of the optimal function (α) applied in the learning machine. The VC dimension h depends on the set of functions $\{f(\mathbf{x}, \alpha) : \alpha \in k_{sub}\}$. The parameter h is controlled by introducing the structure of nested subsets $S_n := \{f(x, \alpha) : \alpha \in k_n\}$,

$$S_1 \subset S_2 \subset S_3 \subset \ldots \subset S_n \subset \ldots \qquad (2.4)$$

with the adequate VC dimensions satisfying

$$h_1 \le h_2 \le \ldots\ldots h_n \le \ldots\ldots$$

The structural minimization principle chooses the function $f(x, \alpha^*)$ in the subset $\{f(\mathbf{x}, \alpha) : \alpha \in k_n\}$ with the minimal right hand side of the eq. (2.3). The guaranteed risk bound is minimal. In the case of prediction, a nonlinear function $f(\alpha)$ has to be constructed that gives minimal error on the test data (data not from learning set). This is done through regression

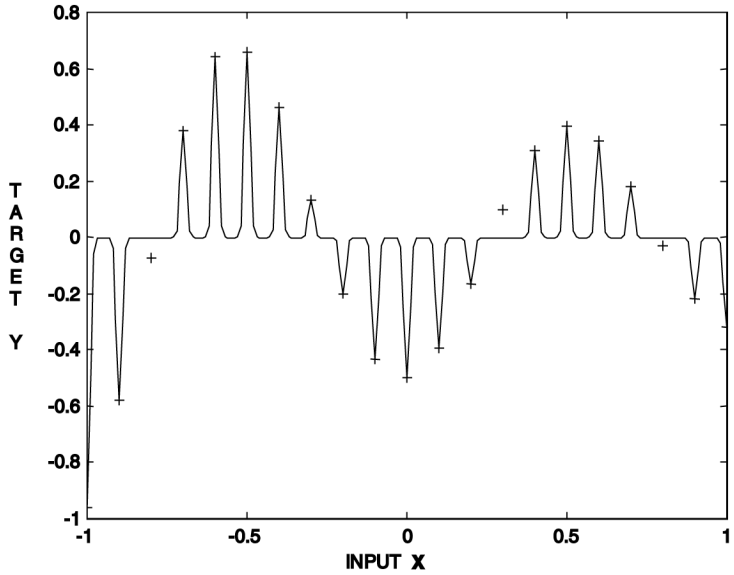

Fig. 2.1 Improper complexity – overfit.

procedure. Because of the simplicity the linear case will be explained first

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b. \tag{2.5}$$

The optimal parameter $\mathbf{w}$ and b could be calculated through empirical error minimization (2.2).

Complex functions can easy generate zero training errors. In the fig. 2.1 there is a simple example of regression function with high complexity. The function goes (almost) exactly through learning points (marked with an x), but for other points between them, it produces meaningless results. This effect is known also as overfit.

In practice, only limited amounts of data are available. That implies that any regression model will be inaccurate – biased. More complex functions require exponentially more data. This means that building regression model solely on the empirical risk minimization defined in (2.2) is inadequate. It has to be expanded by a term that forces optimal model by lowest possible complexity. This is achieved by regularized risk R_{reg}

$$R_{reg} = R_{emp} + \lambda \|\mathbf{w}\|_2^2. \tag{2.6}$$

The regularization parameter λ regulates influence of the penalization term. In the case of linear regression we would like to find the function (represented by coefficients $\mathbf{w}$) with the smallest steepness among the functions that minimize (2.2). There is always some imprecision in the data set – noise. Therefore some small tolerance in errors ε should be allowed and problem (2.6) can be solved as optimization problem

$$\min \frac{1}{2}\|\mathbf{w}\|_2^2 \tag{2.7}$$

with respect to the constraints

$$\begin{aligned} y_1 &= \mathbf{w}^T x_i - b \leq \varepsilon \\ \mathbf{w}^T x_i &+ b - y_i \leq \varepsilon. \end{aligned}$$

In the formulation (2.7) we rely on the assumption that the convex optimization problem is feasible. Sometimes this may not be the case and slack variables ξ are introduced to deal with otherwise infeasible constraints of the optimization problem (2.7)

$$\min \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{k}(\xi_i + \xi_i^*) \tag{2.8}$$

with respect to constraints

$$\begin{aligned} y_1 - \mathbf{w}^T x_i - b &\leq \varepsilon + \xi_i \\ \mathbf{w}^T x_i + b - y_i &\leq \varepsilon + \xi_i^* \\ \xi_i,\, \xi_i^* &\geq 0. \end{aligned}$$

The constant $C > 0$ regulates the trade off between the flatness of f and the amount up to

which deviations larger than ε are tolerated. The ξ insensitive loss function is defined

$$|\xi|_\varepsilon := \begin{cases} 0 & \text{if} \quad |\xi| \leq \varepsilon, \\ |\xi| - \varepsilon & \text{otherwise.} \end{cases} \tag{2.9}$$

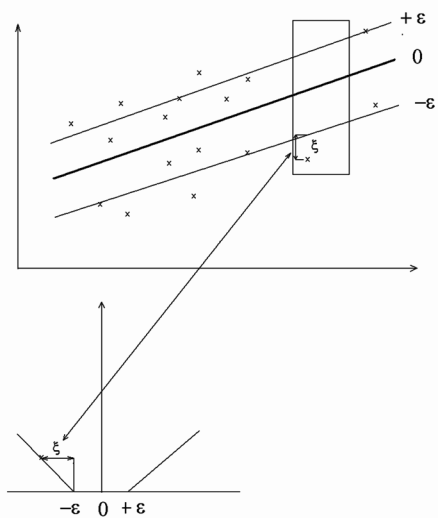

Fig. 2.2 ξ insensitive loss function.

Graphically the ε insensitive loss is presented in fig. 2.2. Above is input $\mathbf{x}$ on the abscise and $y = f(x)$ on the ordinate. Below is ε on the abscise and penalty value on the ordinate. Only the points outside $+\varepsilon$ and $-\varepsilon$ area contribute to the rise of the value in (2.8). For example, the point x marked with ζ_i, that is outside $\pm\varepsilon$ region (fig. 2.2) above, is linearly penalized by the amount shown below. All points inside $\pm\varepsilon$ region are not penalized.

The problem (2.8) can be handled more easily as the dual quadratic program. A Lagrange function is constructed from both objective function (it will be called primal objective function) and the corresponding constraints by introducing dual set of variables. It can be shown that this function has a saddle point with respect to the primal and dual variables at the optimal solution [Mangasarian (1969)]. The Lagrange function has the following form

$$\begin{aligned} L := & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{k}(\xi_i + \xi_i^*) \\ & - \sum_{i=1}^{k}\alpha_1(\varepsilon + \xi_i - y_i + \mathbf{w}^T\mathbf{x}_i + b) \\ & - \sum_{i=1}^{k}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \mathbf{w}^T\mathbf{x}_i - b) \\ & - \sum_{i=1}^{k}(\eta_i\xi_i + \eta_i^*\xi_i^*). \end{aligned} \tag{2.10}$$

The dual variables in (2.10) have to satisfy constraints $\alpha_i, \alpha_i^*, \eta_i, \eta_i^* \geq 0$. From the saddle point the condition follows

$$\partial_b L = \sum_{i=1}^{k}(\alpha_i^* - \alpha_i) = 0 \tag{2.11}$$

$$\partial_w L = \mathbf{w} - \sum_{i=1}^{k}(\alpha_i^* - \alpha_1)\mathbf{x}_i = 0 \tag{2.12}$$

$$\partial_{\xi^{(*)}} L = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0. \tag{2.13}$$

Substituting (2.11), (2.12) and (2.13) into (2.10) yields dual optimization problem [Vapnik 1998] where now the maximum is searched

$$\begin{aligned} \max & -\frac{1}{2}\sum_{i,j=1}^{k}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\mathbf{x}_i\mathbf{x}_j \\ & - \varepsilon\sum_{i,j=1}^{k}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{k} y_i(\alpha_i - \alpha_i^*) \end{aligned} \tag{2.14}$$

with respect to the constraints

$$\begin{aligned} & \sum_{i=1}^{k}(\alpha_i - \alpha_i^*) = 0 \\ & \alpha_i, \alpha_i^* \in [0, C]. \end{aligned}$$

Dual variables η_i, η_i^* are eliminated through condition (2.13) [Vapnik 1998]. As values can be above ε region or below $-\varepsilon$ region, variables are separated into α_i, η_i, ξ and α_i^*, η_i^*, ξ^*. From (2.12) $\mathbf{w}$ can be expressed

$$\mathbf{w} = \sum_{i=1}^{k}(\alpha_i^* - \alpha_i)\mathbf{x}_i$$

and (2.5) can be expressed in the form of

$$f(\mathbf{x}) = \sum_{i=1}^{SV}(\alpha_i^* - \alpha_i)\mathbf{x}_i\mathbf{x} + b. \tag{2.15}$$

The form (2.15) is the support vector expansion of the linear regression.

The coefficient b can be computed by exploiting Karush-Khun-Tucker conditions [Karush (1939)], [Kuhn et al. (1951)]. These state that at the optimal solution the product between dual variables and constraints has to vanish. In our case this means

$$\begin{aligned} \alpha_i(\varepsilon + \xi_i - y_i + \mathbf{w}x_i + b) = 0 \\ \alpha_i^*(\varepsilon + \xi_i^* + y_i - \mathbf{w}x_i - b) = 0 \end{aligned} \tag{2.16}$$

$$\begin{aligned} (C - \alpha_i)\xi_i = 0 \\ (C - \alpha_i^*)\xi_i^* = 0. \end{aligned} \tag{2.17}$$

From (2.16) it follows that only samples $(\mathbf{x}_i, y_i)$ with corresponding $\alpha_i^{(*)} = C$ lie outside $\varepsilon-$ insensitive tube around f. Situation $\alpha_i\alpha_i^* = 0$ means that both variables cannot be nonzero at the same time. This would require nonzero slack variables in both directions. For $\alpha_i^{(*)} \in [0, C]$ we have $\xi_i^{(*)} = 0$ and the second factor in (2.16) must vanish. It follows

$$\begin{aligned} b = y_i - \mathbf{w}x_i - \varepsilon = 0 \quad \text{for} \quad \alpha_i \in (0, C) \\ b = y_i - \mathbf{w}x_i + \varepsilon = 0 \quad \text{for} \quad \alpha_i^* \in (0, C). \end{aligned} \tag{2.18}$$

From (2.16) it follows that only for $|f(\mathbf{x}_i) - y_i| \geq \varepsilon$ the Lagrange multipliers may be nonzero. For all samples inside the $\varepsilon-$ tube α_i, α_i^* vanish. For $|f(\mathbf{x}_i) - y_i| < \varepsilon$ second factor is nonzero. Because of Karush-Khun-Tucker conditions α_i, α_i^* has to be zero. Therefore, we have sparse expansions of $\mathbf{w}$ in terms of $\mathbf{x}_i$. In other words, we do not need all $\mathbf{x}_i$ to describe $\mathbf{w}$. Examples with the vanishing coefficient are called support vectors.

The regression algorithm is made nonlinear by applying map $\Phi : X \to F$ and then the standard regression algorithm. An example of such map is $\Phi : \mathcal{R}^2 \to \mathcal{R}^3$

$$\Phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1, x_2, x_2^2). \tag{2.19}$$

Where subscripts in this case refer to the components of $\mathbf{x} \in \mathcal{R}^2$. For higher order maps analytical expressions become impossible. The work around is implicit mapping via kernels $K(\mathbf{x}, \mathbf{x}^T) := \Phi(\mathbf{x}) \bullet \Phi(\mathbf{x}^T)$ instead of using $\Phi(\mathbf{x})$ explicitly ($\bullet$ is dot product). Now we can rewrite (2.14) as follows

$$\begin{aligned} \max -\frac{1}{2}\sum_{i,j=1}^{k}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(\mathbf{x}_i\mathbf{x}_j) \\ -\varepsilon\sum_{i,j=1}^{k}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{k} y_i(\alpha_i - \alpha_i^*) \end{aligned} \tag{2. 20}$$

with respect to the constraints

$$\begin{aligned} \sum_{i=1}^{k}(\alpha_i + \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C]. \end{aligned}$$

The expansion of $\mathbf{w}$

$$\mathbf{w} = \sum_{i=1}^{k}(\alpha_i^* - \alpha_i)\Phi(\mathbf{x}_i)$$

and function f in (2.15) can be expressed in the form of

$$f(\mathbf{x}) = \sum_{i=1}^{k}(\alpha_i^* - *_i)K(\mathbf{x}_i\mathbf{x}) + b. \tag{2.21}$$

Optimization problem now corresponds to finding the flattest function in the feature space and not in the original input space.

The fusion of ANN with soft computing enables construction of a new learning machine that is superior compared to classical ANN, because knowledge can be extracted and explained in the form of simple rules [Zadeh (1997)]. The following new support vector FANN architecture can be defined as presented in the fig. 2.3.

Layer 1 calculates membership values. Layer 2 performs T norm operator (multiplication). Layer 3 derives the product of each rule's output. Layer 4 performs the kernel Gaussian radial basis operation (2.22). Layer 5 sums its inputs as the overall output where $\overline{\alpha} = \alpha_i^* - \alpha_i$.

In the model presented here, the membership functions are in the form of Gaussian functions:

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left[-\frac{(\mathbf{x} - \mathbf{x}_i)^2}{2\sigma^2}\right]. \tag{2.22}$$

The support vector technique places one local Gaussian function (2.22) in each support vector

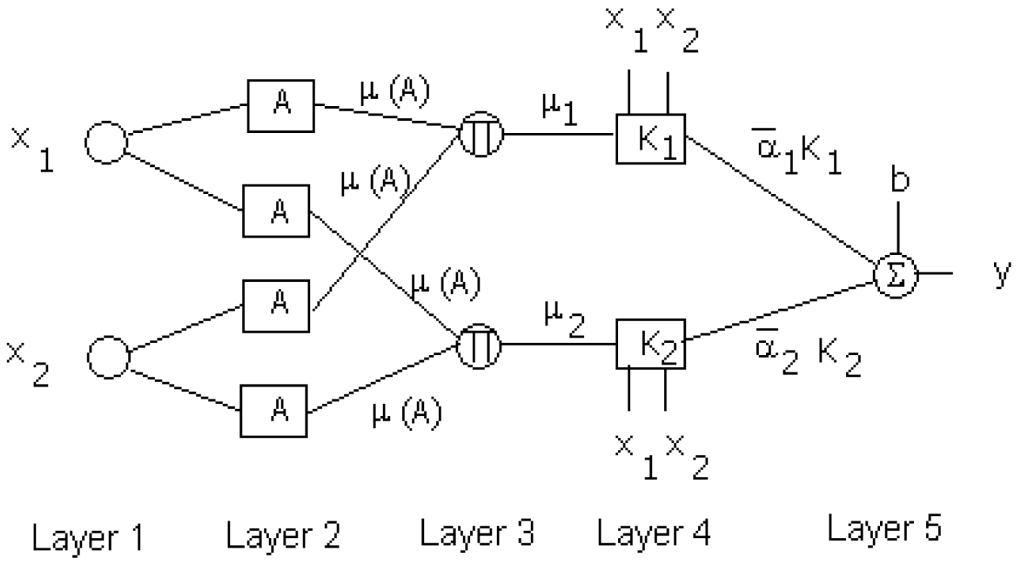

Fig. 2.3 Support vector FANN architecture.

so application of clustering methods [Bezdek et al. (1987), Chiu (1994), Yager et al. (1994)] is unnecessary. The basis width σ of (2.22) is selected by structural minimization principles (2.3) and (2.4). The output from the FANN is

$$f(\mathbf{x}) = \sum_{i=1}^{SV} (\alpha_i^* - \alpha_i) K(\mathbf{x}_i \mathbf{x}) + b \qquad (2.23)$$

where SV is the number of support vectors.

3. Case Study

An example of application of the above theory is shown with regard to the daily electrical energy consumption [Srinivasan et al. (1995)]. The data set that consists of meteorological data $(\mathbf{x}_i)$ and daily energy consumption (y_i). The data set is preprocessed to present only days from Monday to Friday, without holidays or any other outliers. The identification task is a multiple input single output type problem. The output was the prediction of daily energy consumption.

The idea applied here is to use only a part of the data set (a window) due to the nature of the problem. Therefore, only local data (around the date of prediction) from each year is taken as the learning set. This technique cannot be applied using the standard time series methods, because the data set in this case is too small. The structural risk minimization principle was applied to minimize expected risk (2.4). For each predicted point in fig. 3.1 optimization problem (2,20) in feature space is solved. Prediction is calculated with (2.23).

For comparison purpose two different approaches for nonlinear system identification were applied to the described problem. The first method is extension of the classical identification model ARX, where implementation is done with the help of neural networks (NNARX). This model does not include feedback and does not pose the stability problems, which can occur in recurrent networks such as NNARMAX and other similar techniques. The second method is a window based fuzzy identification method described above.

The task is to identify the shape of an energy consumption curve and afterwards make predictions (fig. 3.1). The database consists of the data for five years, where the last 30 days are removed for testing a prediction error.

In the system identification theory, different models exist for nonlinear identification. In our approach, we have chosen the ARX model [Ljung (1997)], modeled by regressor vector:

$$\Phi(t) = [y(t-1) \ldots y(t-n_a), \mathbf{u}(t-n_k) \ldots \mathbf{u}(t-n_b-n_k+1)]^T \qquad (3.1)$$

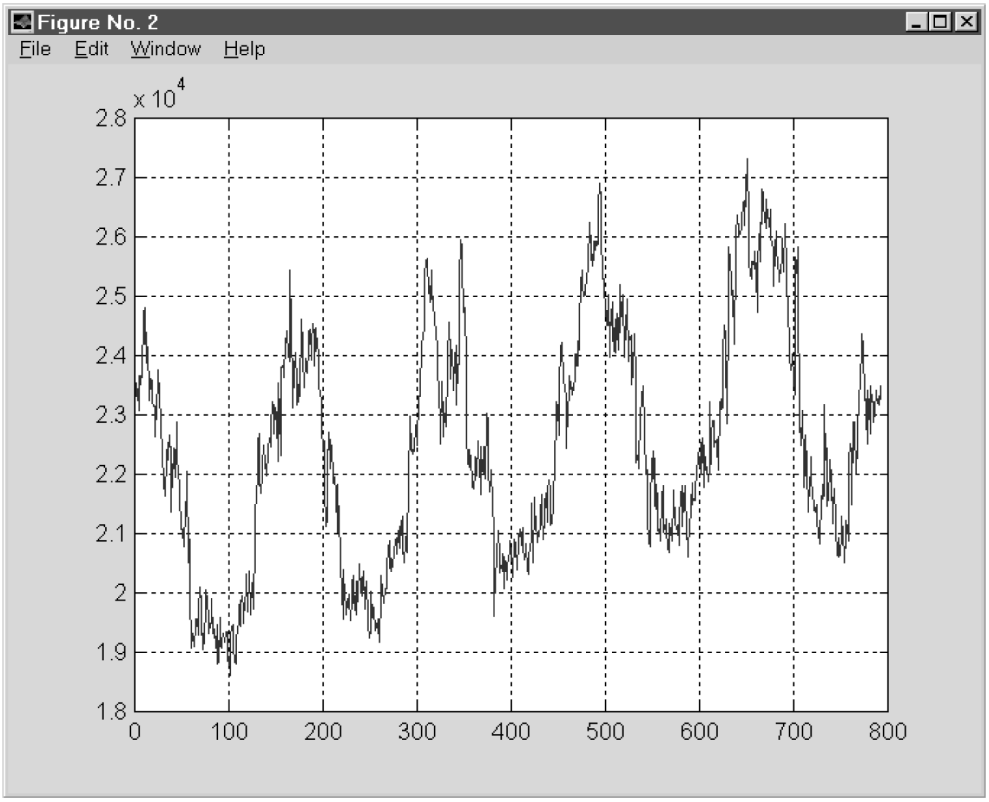

Fig. 3.1 Daily energy consumption curve for the work days.

$y(t) =$ the output at the sampling time t

$u(t) =$ the input at the sampling time t

$n_a =$ the number of the past outputs

$n_b =$ the number of the past inputs

$n_k =$ the time delay ($n_k = 1$ usually)

and the predictor:

$$\widehat{y}\,(t, \Theta) = (t|t-1, \Theta) = g(\varphi(t), \Theta) \quad (3.2)$$

$\widehat{y}\,(t, \Theta) =$ is the predicted output

$g(\varphi(t), \Theta) =$ is the function realized by the artificial neural network (ANN).

$\Theta =$ is a vector of weights of the ANN.

In our case study, the ANN is a multilayer perceptron (MLP) with one hidden layer. For the given training set:

$$Z^N = \{[\mathbf{u}(t), y(t)]|t = 1, \ldots, N\}, \quad (3.3)$$

training the MLP presents a mapping from the set of the training data to the set of possible weights such that

$$Z^N \rightarrow \widehat{\Theta}\ . \quad (3.4)$$

In addition, the network will produce output, which is given by

$$\widehat{y}\,(t, \Theta) \approx y(t, \Theta) \quad (3.5)$$

where the predicted value will be as close as possible to the true data y.

The prediction-error-approach is used to minimize the square of error

$$\min V_N(\Theta, Z^N) = \frac{1}{N}\sum_{i=1}^{N}(y(t) - \widehat{y}\,(t|\Theta))^2. \quad (3.6)$$

Minimization is done with the Levenberg-Marquardt method [Nørgaard (1997)]. However, this is not an easy task because of the following problems

1) selecting a proper model structure (complexity),

2) multiple minima exist in the error surface and the simulated annealing algorithm should be applied [Aarts et al. (1987)].

Performances of the FANN versus the NNARX are presented for the case of a daily electrical energy consumption prediction. The learning data set consists of meteorological data and the energy from the previous days. The data are preprocessed to present only the days from Monday to Friday, without holidays or any other outliers. The trend is obvious in fig. 3.1 and to make the curve stationary, it was removed before the learning phase.

Due to the nature of the problem, in the case of training the FANN only a part of the data set (a window around the date of the prediction) was used. This technique cannot be applied using the NNARX model, because the data set in this case is too small. For the NNARX the classical time series approach was used with the daily energy consumption from the previous days and meteorological data for the same days and predicted values (from the weather forecast) for the day of prediction.

The FANN structure enables extracting the rules in the "if – then" form from the positions and the width (σ) of the membership functions for the each input. By applying these rules the FANN can explain each particular prediction it has made. Standard ANNs are not able to explain their conclusions. Only limited information about their conclusion – making process can be devised from Hinton diagrams.

To achieve the best prediction accuracy for the NNARX model, optimal pruning method was applied. The result of optimization is given in the fig. 3.2, where unnecessary nodes were pruned. The pruning procedure used in this study is based on the modified method of Hansen and Pedersen [Hansen et al. (1994)]. This technique stems from the so-called optimal brain surgeon method developed by Hassibi and Stork [Hassibi et al. (1993)]. The Neural Network Based System Identification Toolbox developed by Magnus Nørgaard [Nørgaard (1997)] was applied for modeling and optimizing NNARX. In fig. 3.3 daily energy consumption predictions are presented, made by the FANN and the NNARX for the 30 days time period. The FANN window based method achieved about 10% better prediction accuracy.

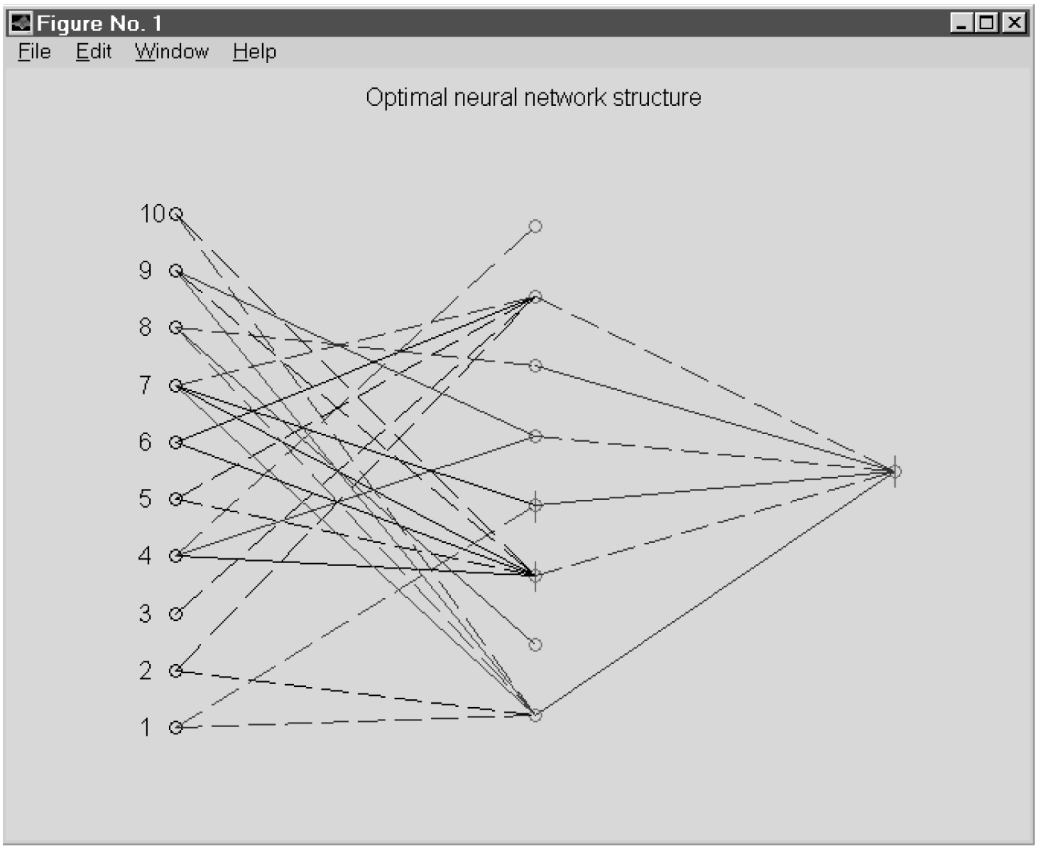

Fig. 3.2 Resulting NNARX after pruning.

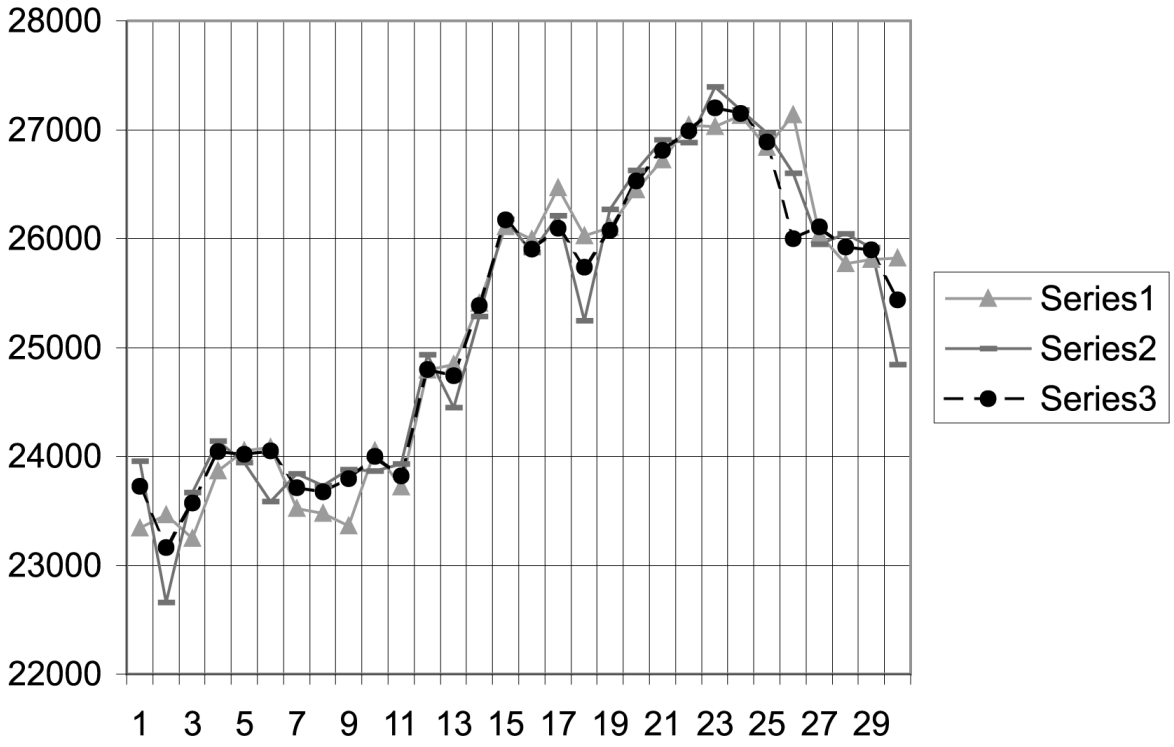

Fig. 3.3 Comparison of the results, 1 = NNARX, 2 = actual, 3 = FANN.

4. Conclusion

The fusion of artificial neural networks with soft computing enables to construct learning machines that are superior compared to classical artificial neural networks because knowledge can be extracted and explained in the form of simple rules. An efficient method for selecting the optimal structure of a fuzzy neural network architecture is developed and a new fuzzy neural architecture is introduced. The Vapnik Chervonenkis (VC) dimension is applied as a measure of the capacity of the learning machine. Prediction of the expected error on the yet unseen examples can be estimated with the help of the VC dimension. The structural risk minimization principle is introduced for constructing the machine with the lowest expected error.

Performances of the above theory are tested on the prediction of the daily electrical energy consumption. The data set consists of meteorological data and daily energy consumption. The idea applied here is to use only a part of the data set (a window) due to the nature of the problem. Therefore, only local data (around the date of prediction) from each year is taken as the learning set. This technique cannot be applied using standard time series methods because the data set in this case study is too small. The structure risk minimization principle was applied to minimize any expected error.

Performances of two different methodologies: FANN with windows and NNARX for the identification and the prediction are presented. The FANN has better performance due to the following properties

1) a well developed theoretical background for learning from a small data set,
2) transformation of the identification problem from nonlinear to linear feature space with the kernel method makes problem convex. Global optimum is granted and learning process is faster,
3) it requires a smaller data set than the NNARX so only local behavior of the data is exploited which is more consistent with the physical meaning of the process.

References

[1] AARTS E. AND LAARHOVEN P., *Simulated Annealing: Theory and Practice*, John Wiley and Sons, 1987.

[2] BEZDEK J., HATAWAY R., SABIN M. AND TUCKER W., Convergence Theory for Fuzzy C means: Counter Examples, and Repairs, *The Analysis of Fuzzy Information*, Bezdek J., editor, Vol. **3**, Chap. 8, CRC Press, 1987.

[3] CHIU S.L., Fuzzy Model Identification Based on Cluster Estimation, *Journal of Intelligent & Fuzzy Systems*, Vol. **2**, No. 3, Sept. 1994.

[4] HANSEN L. K. AND PEDERSEN M. W., Controlled Growth of Cascade Correlation Nets, *Proc. ICANN'94*, Sorrento, Italy, Eds. Marinaroand M., Morasso P.G., p. 797–800, 1994.

[5] HASSIBI B. AND STORK D. G., Second Order Derivatives for Network Prunning: Optimal Brain Surgeon, *NIPS 5*, Eds. Hanson S. J. and et al., San Mateo, p. 164, Morgan Kaufmann, 1993.

[6] KARUSH W., Minima of functions of several variables with inequalities as side constraints, *Master's thesis*, Dep. Of Mathematics, Univ. of Chicago, 1939.

[7] KUHN W. AND TUCKER A.W., Nonlinear Programming in Neymann J. (ed.), *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, CA, p. 481–492, 1951.

[8] LJUNG L., *System identification – Theory for the User*, Prentice Hall, 1997.

[9] MANGASARIAN O. L., *Nonlinear Programming*, McGraw-Hill, New York, NY, 1969.

[10] NØRGAARD M., *Neural Network Based System Identification Toolbox*, Department of Automation, TU-Denmark, 1997.

[11] SCHÖLKOPF B., BURGES C. AND VAPNIK V. N., Extracting Support Data for a Given Task, in Fayyad U.M. and Uthurusamy R., (eds.), *First International Conference on Knowledge Discovery and Data Mining, Proceedings*, AAAI Press, Menlo Park, CA, 1995.

[12] SRINIVASAN D., LIEW A. C. AND CHANG C. S., Application of Fuzzy Systems in Power Systems, *Electric Power System Research*, 35 p. 39–43, 1995.

[13] SRINIVASAN D., LIEW A. C. AND CHANG C. S., Demand Forecasting Using Fuzzy Neural Computation, with Special Emphasis on Weekend and Public Holiday Forecasting, *IEEE PES Winter Meeting*, New York, USA, paper No. 95 WM 158–6–PWRS, 1995.

[14] VAPNIK V. N., GOLOWICH S. E. AND SMOLA A., Support vector method for function approximation, regression and signal processing, *Advances in Neural Information Processing Systems*, Vol. **9** MIT Press, Cambridge MA., USA, 1996.

[15] VAPNIK V.N., Statistical Learning Theory, John Wiley and Sons, (1998).

[16] YAGER R. AND FILEV D., Generation of Fuzzy Rules by Mountain Clustering, *Journal of Intelligent & Fuzzy Systems*, Vol. **2**, No. 3, 209–219, 1994.

[17] ZADEH L. A., The Role of Fuzzy Logic and Soft Computing in the Conception, Design and Deployment of Intelligent Systems, *Proceedings of Computer Based Medical Systems 97*, Maribor, June 1997.

Received: October, 2000
Revised: April, 2001
Accepted: May, 2001

Contact address:
Bojan Novak
University of Maribor
Faculty of Electrical Engineering and Computer Science
Smetanova 17, 2000 Maribor
e-mail: novakb@uni-mb.si

BOJAN NOVAK received B. S. degree 1982, M. S. degree 1985 and Ph. D. degree 1990 all from Technical faculty of Maribor, University of Maribor. From 1984 to 1987 he was assistant, from 1987 to 1995 assistant profesor and from 1995–2000 associated profesor, all at the Technical faculty of Maribor, University of Maribor. His research interest are in the areas of soft computing, fuzzy logic, learning theory, neural networks, operational research, theory of forecasting and pattern recognition.