THE LEAST SQUARES SUPPORT VECTOR MACHINE BASED ON A NEO-FUZZY NEURON

Yevgeniy Bodyanskiy, Oleksii Tyshchenko, Daria Kopaliani

***Abstract**: The paper presents a fuzzy least squares support vector machine (LS-FSVM) which is implemented with the help of neo-fuzzy neurons (NFN) and which is essentially a zero order Takagi-Sugeno fuzzy inference system. The proposed LS-FSVM-NFN is numerically simple because it's generated with NFNs, it also has a small number of adjustable parameters and high speed associated with the possibility of applying the second order optimization learning procedures to process data in an online-mode.*

***Keywords**: Fuzzy support vector machine, neo-fuzzy neuron, learning procedure, time series.*

***ACM Classification Keywords**: 1.2.6 [Artificial Intelligence]: Learning – Connectionism and neural nets.*

Introduction

Currently, artificial neural networks (ANNs) are widely used for solving Data Mining, intelligent control, forecasting, pattern recognition tasks, etc. under uncertainty conditions, nonlinearity, stochasticity, randomness, various types of disturbances and noise, thanks to its universal approximating abilities and learning opportunities based on experimental data characterizing functioning of the investigated object [Haykin, 1999; Du, 2014].

ANNs' learning process is usually based on the use of the criterion optimization procedure, the convergence speed of this procedure can be quite low, especially while training multi-layer networks such as a multilayer perceptron (MLP), which creates a number of problems when a training sample is fed to the system in the form of an observations' sequence in an online mode, for example, adaptive control of non-stationary objects, Web Mining etc.

To accelerate the learning process in neural networks whose output signal is linearly dependent on adjustable synaptic weights is possible, for example, using radial basis (RBFN), normalized radial basis (NRBFN), polynomial (PNN) and the GMDH-neural networks (GMDH-ANN), however, their use is often complicated by the so-called "curse of dimensionality." The issue is not only in arising computational difficulties, but the reason is the available experimental data may be not enough for estimating a large number of synaptic weights.

An alternative to the optimization-based learning is the memory-based learning [Nelles, 2001] which is associated with a concept "neurons in the data points" [Zahirniak, 1990]. The most typical representative of neural networks whose training is based on this concept are generalized regression neural network (GRNN), but they solve the problem of interpolation and not approximation which complicates greatly their use while processing "noisy" data.

A hybrid of different neural networks, whose training is based both on optimization and memory, are support vector machines (SVM) [Vapnik, 1974; Vapnik, 1979; Cortes, 1995; Vapnik, 1995]. Their architecture coincides with RBFN and GRNN, synaptic weights are determined as a result of solving a nonlinear programming problem, and activation functions' centers are set according to the concept "neurons in the data points."

Thus, this network is a network with direct information transmission, which are generalizations of such popular constructions as MLP, RBFN, GRNN, which implement an empirical risk minimization method [Vapnik, 1974; Vapnik, 1979]. They have been widely applied to solving identification, pattern recognition and neurocontrol problems [Haykin, 1999; Du, 2014]. Although SVM-networks have a number of unquestionable advantages, their training is quite time consuming from a computational point of view, as it has to do with solving nonlinear programming problems of high dimensionality.

In this regard, least squares support vector machines (LS-SVM) [Suykens, 2002] were proposed as an alternative to the ordinary SVM, whose training is reduced to solving systems of linear equations. That's much easier from a computational point of view.

Neuro-fuzzy systems (NFS) [Jang, 1997] have more features compared to neural networks with their learning capability, approximation and linguistic interpretation of the results. Here, ANFIS [Jang, 1993] and TSK-systems [Takagi, 1985] are the most widely used systems, whose output layer is adjusted with the help of linear learning algorithms. It should be mentioned that the majority of neuro-fuzzy systems is trained with the help of optimization procedures.

A fuzzy analogue of a traditional SVM is a fuzzy support vector machine (FSVM) [Lin, 2002], where multidimensional kernel activation functions are replaced with one-dimensional bell-shaped membership functions. In [Abe, 2003], a least squares fuzzy support vector machine (LS-FSVM) was introduced to solve the tasks of pattern recognition based on binary training signals.

Although FSVM has a great potential compared to a traditional SVM, a training procedure is rather cumbersome from a computational point of view due to its implementation, which naturally limits its ability to solve real-time tasks.

It is advisable to develop rather simple neuro-fuzzy systems to realize the learning idea based on the empirical risk minimization when information is processed in an online mode.

A neo-fuzzy neuron [Yamakawa, 1992; Uchino, 1997; Miki, 1999] can be used as a basic element of such systems, which is characterized by high approximating properties, its simplicity and speed learning.

A Neo-Fuzzy Neuron

A neo-fuzzy neuron (NFN) is a nonlinear system with multiple inputs and a single output having the following mapping

$$\hat{y} = \sum_{i=1}^{n} f_i\left(x_i\right)$$

where x_i is the $i-$th component of a $n-$dimensional vector of input signals $x = \left(x_1,\ldots,x_i,\ldots,x_n\right)^T \in R^n$, $\hat{y}$ is a scalar NFN output. Structural units of the neo-fuzzy neuron are nonlinear synapses NS_i which transform the $i-$th input signal in the following way

$$f_i\left(x_i\right) = \sum_{l=1}^{h} w_{li}\mu_{li}\left(x_i\right)$$

where w_{li} is the $l-$th adjustable synaptic weight of the $i-$th nonlinear synapse, $l=1,2,\ldots h-$ the total quantity of synaptic weights and, respectively, membership functions $\mu_{li}\left(x_i\right)$ in the same nonlinear synapse. In this way transformation carried out by the NFN can be written as

$$\hat{y}=\sum_{i=1}^{n}\sum_{l=1}^{h}w_{li}\mu_{li}\left(x_i\right) \tag{1}$$

and the fuzzy inference carried out by the same NFN has a form of

IF x_i IS X_{li} THEN THE OUTPUT IS w_{li}

which means that actually a nonlinear synapse implements a fuzzy zero-order Takagi-Sugeno reasoning [Takagi, 1985].

Authors of the neo-fuzzy neuron [Yamakawa, 1992; Uchino, 1997; Miki, 1999] used traditional triangular constructions meeting the conditions of unity partitioning as membership functions:

$$\mu_{li}\left(x_i\right)=\begin{cases}\dfrac{x_i-c_{l-1,i}}{c_{li}-c_{l-1,i}},\ if\ x_i\in\left[c_{l-1,i},c_{li}\right],\\ \dfrac{c_{l+1,i}-x_i}{c_{l+1,i}-c_{li}},\ if\ x_i\in\left[c_{li},c_{l+1,i}\right],\\ 0,\ otherwise\end{cases}$$

where c_{li} are relatively arbitrarily chosen (usually evenly distributed) centers of membership functions over the interval $\left[0,1\right]$ where, naturally, $0\le x_i\le 1$.

This choice of membership functions ensures that the input signal x_i activates only two neighboring membership functions, and their sum is always equal to 1 which means that

$$\mu_{li}\left(x_i\right)+\mu_{l+1,i}\left(x_i\right)=1$$

and

$$f_i\left(x_i\right)=w_{li}\mu_{li}\left(x_i\right)+w_{l+1,i}\mu_{l+1,i}\left(x_i\right).$$

Of course, other types of membership functions (except triangular) can be used like cubic and B-splines, polynomials, harmonic and orthogonal functions, wavelets etc. It should be noticed that the NFN contains nh membership functions and the same amount of adjustable synaptic weights.

Introducing a $\left(nh\times 1\right)-$vector of membership functions

$$\mu\left(x\left(k\right)\right)=\left(\mu_{11}\left(x_1\left(k\right)\right),\ldots,\mu_{h1}\left(x_1\left(k\right)\right),\mu_{12}\left(x_2\left(k\right)\right),\ldots,\mu_{li}\left(x_i\left(k\right)\right),\ldots,\mu_{hn}\left(x_n\left(k\right)\right)\right)^T$$

(here $k=1,2,\ldots,N$ is a number of the vector observation $x\left(k\right)$ in a training sample or current discrete time) and a corresponding vector of NFN synaptic weights

$$w=\left(w_{11},\ldots,w_{h1},w_{12},\ldots,w_{h2},\ldots,w_{li},\ldots,w_{hn}\right)^T,$$

the transformation (1) carried out by the NFN can be rewritten in the form

$$\hat{y}(k) = w^T \mu(x(k)).$$

The NFN authors used a gradient learning procedure

$$w_{li}(k) = w_{li}(k-1) + \eta e(k)\mu_{li}(x_i(k)) = w_{li}(k-1) + \eta(y(k) - \hat{y}(k))\mu_{li}(x_i(k)) =$$
$$= w_{li}(k-1) + \eta(y(k) - w^T(k-1)\mu(x(k)))\mu_{li}(x_i(k))$$

where $y(k)$ is a reference signal, η is a learning rate parameter.

In [Bodyanskiy, 2003], a learning algorithm was proposed that posses both tracking (non-stationary cases) and filtering («noisy» data) properties:

$$\begin{cases} w(k) = w(k-1) + r^{-1}(k)e(k)\mu(x(k)), \\ r(k) = \alpha r(k-1) + \|\mu(x(k))\|^2, 0 \le \alpha \le 1, \end{cases} \tag{2}$$

when $\alpha = 0$, the algorithm (2) coincides with the optimal Kaczmarz-Widrow-Hoff learning algorithm.

Basically, to set the NFN lots of other learning algorithms and identification [Nelles, 2001; Ljung, 1999] can be used including the standard least squares method

$$w(N) = \left(\sum_{k=1}^{N} \mu(x(k))\mu^T(x(k))\right)^{-1} \sum_{k=1}^{N} \mu(x(k))y(k) \tag{3}$$

and also his recurrent and exponentially-weighted versions.

The NFN training based on the empirical risk minimization

Training the NFN with the help of the least squares support vector machine approach (LS-SVM-NFN) leads to the quadratic criterion optimization

$$E(N) = \frac{1}{2}\|w\|^2 + \frac{\gamma}{2}\sum_{k=1}^{N} e^2(k) \tag{4}$$

within the constraints as a system of N linear equations

$$y(k) = w^T \mu(x(k)) + e(k) \tag{5}$$

where $\gamma > 0$ is a regularization parameter (a momentum term).

The criterion optimization (4) without the constraints (5) leads to the expression

$$w(N) = \left(\sum_{k=1}^{N} \mu(x(k))\mu^T(x(k)) + \gamma^{-1}I\right)^{-1} \sum_{k=1}^{N} \mu(x(k))y(k)$$

which is rather close to (3) and which is essentially a ridge estimator, where $I - (nh \times nh)$ is an identity matrix.

Let's introduce a Lagrange function to take into account the constraints' system (5)

$$L(w, e(k), \lambda(k)) = E(k) + \sum_{k=1}^{N} \lambda(k)(y(k) - w^T \mu(x(k)) - e(k)) =$$
$$= \frac{1}{2}w^T w + \frac{\gamma}{2}\sum_{k=1}^{N} e^2(k) + \sum_{k=1}^{N} \lambda(k)(y(k) - w^T \mu(x(k)) - e(k))$$

(here $\lambda(k)$ stands for N undetermined Lagrange multipliers) and the Karush-Kuhn-Tucker system of equations

$$\begin{cases} \nabla_w L\left(w,e(k),\lambda(k)\right) = w - \sum_{k=1}^{N} \lambda(k)\mu\left(x(k)\right) = \vec{0}_N, \\ \dfrac{\partial L\left(w,e(k),\lambda(k)\right)}{\partial e(k)} = \gamma e(k) - \lambda(k) = 0, \\ \dfrac{\partial L\left(w,e(k),\lambda(k)\right)}{\partial \lambda(k)} = y(k) - w^T \mu\left(x(k)\right) - e(k) = 0 \end{cases} \tag{6}$$

where $\vec{0}_N - (N \times 1)$ is a vector formed with zeros.

The solution of the equation system (6) is:

$$\begin{cases} w(N) = \sum_{k=1}^{N} \lambda(k)\mu\left(x(k)\right), \\ \lambda(k) = \gamma e(k), \\ y(k) = w^T(N)\mu\left(x(k)\right) + e(k) \end{cases} \tag{7}$$

or in a matrix form

$$\left(\gamma^{-1} I_{NN} + \Omega_{NN}\right)\begin{pmatrix} \lambda(1) \\ \vdots \\ \lambda(N) \end{pmatrix} = \begin{pmatrix} y(1) \\ \vdots \\ y(N) \end{pmatrix}$$

(here $I_{NN} - (N \times N)$ is an identity matrix)

or

$$\left(\gamma^{-1} I_{NN} + \Omega_{NN}\right)\Lambda_N = Y_N$$

(here $\Omega_{NN} = \left\{\Omega_{pq} = \mu^T\left(x(p)\right)\mu\left(x(q)\right)\right\}, p = 1,2,\ldots,N; q = 1,2,\ldots,N$), whence

$$\Lambda_N = \left(\gamma^{-1} I_{NN} + \Omega_{NN}\right)^{-1} Y_N. \tag{8}$$

Then an output NFN signal

$$\hat{y}(x) = w^T(N)\mu(x)$$

for an arbitrary input signal x taking into account (7), (8) can be written in the form

$$\hat{y}(x) = \left(\sum_{k=1}^{N} \lambda(k)\mu\left(x(k)\right)\right)^T \mu(x). \tag{9}$$

If the processing data are consecutively supplied, the training process of the LS-SVM-NFN should be fulfilled in an online mode. Thus when a pair of $x(N+1), y(N+1)$ comes to the system, the expression (9) takes the form

$$\hat{y}(x)=\left(\sum_{k=1}^{N}\lambda(k)\mu\left(x(k)\right)+\lambda(N+1)\mu\left(x(N+1)\right)\right)^{T}\mu(x)$$

or in a matrix form

$$\left(\gamma^{-1}I_{N+1,N+1}+\Omega_{N+1,N+1}\right)\begin{pmatrix}\lambda(1)\\ \vdots\\ \lambda(N)\\ -----\\ \lambda(N+1)\end{pmatrix}=\begin{pmatrix}y(1)\\ \vdots\\ y(N)\\ -----\\ y(N+1)\end{pmatrix}$$

or

$$\begin{pmatrix}\Omega_{NN} & | & \omega_{N+1}\\ -- & - & --\\ \omega_{N+1}^{T} & | & \gamma^{-1}\end{pmatrix}\begin{pmatrix}\Lambda_{N}\\ -----\\ \lambda(N+1)\end{pmatrix}=\begin{pmatrix}Y_{N}\\ -----\\ y(N+1)\end{pmatrix} \tag{10}$$

where $\omega_{N+1}=\left(\mu^{T}\left(x(1)\right)\mu\left(x(N+1)\right),\mu^{T}\left(x(2)\right)\mu\left(x(N+1)\right),\ldots,\mu^{T}\left(x(N)\right)\mu\left(x(N+1)\right)\right)^{T}$.

It comes from the expression (10) that

$$\Lambda_{N+1}=\begin{pmatrix}\Lambda_{N}\\ -----\\ \lambda(N+1)\end{pmatrix}=\begin{pmatrix}\Omega_{NN} & | & \omega_{N+1}\\ -- & - & --\\ \omega_{N+1}^{T} & | & \gamma^{-1}\end{pmatrix}^{-1}\begin{pmatrix}Y_{N}\\ -----\\ y(N+1)\end{pmatrix}. \tag{11}$$

Using the Frobenius formula in the form of [Gantmacher, 2000]

$$M=\begin{pmatrix}A & | & B\\ - & - & -\\ C & | & D\end{pmatrix},\quad |D|\neq 0,$$

$$M^{-1}=\begin{pmatrix}A & | & B\\ - & - & -\\ C & | & D\end{pmatrix}^{-1}=\begin{pmatrix}K^{-1} & | & -K^{-1}BD^{-1}\\ ------ & - & -----------\\ -D^{-1}CK^{-1} & | & D^{-1}+D^{-1}CK^{-1}BD^{-1}\end{pmatrix},$$

$$K=A-BD^{-1}C$$

where taking into consideration (11)

$$K=\Omega_{NN}-\omega_{N+1}\gamma\omega_{N+1}^{T},\ K^{-1}=\left(\Omega_{NN}-\gamma\omega_{N+1}\omega_{N+1}^{T}\right)^{-1}$$

one can easily calculate the $(N+1)-$th Lagrange multiplier with the help of the expression

$$\lambda(N+1)=-\gamma\omega_{N+1}^{T}K^{-1}Y_{N}+\gamma\left(1+\gamma\omega_{N+1}^{T}K^{-1}\omega_{N+1}\right)y(N+1).$$

Then using the Sherman-Morrison formula of matrices inversion [Gantmacher, 2000], we finally get

$$\begin{cases}K^{-1}=\Omega_{NN}^{-1}+\dfrac{\Omega_{NN}^{-1}\omega_{N+1}\omega_{N+1}^{T}\Omega_{NN}^{-1}}{1-\omega_{N+1}^{T}\Omega_{NN}^{-1}\omega_{N+1}},\\ \lambda(N+1)=1+\gamma\omega_{N+1}^{T}K^{-1}\left(\omega_{N+1}-Y_{N}\right).\end{cases}$$

Conclusion

The paper presents a fuzzy least squares support vector machine (LS-FSVM) which is implemented with the help of neo-fuzzy neurons (NFN) and which is essentially a zero order Takagi-Sugeno fuzzy inference system. The proposed LS-FSVM-NFN is numerically simple because it's generated with NFNs, it also has a small number of adjustable parameters and high speed associated with the possibility of applying the second order optimization learning procedures to process data in an online-mode.

Bibliography

[Abe, 2003] S. Abe, D. Tsujinishi. Fuzzy Least Squares Support Vector Machines for multiclass problems. Neural Networks, 2003, №16, P. 785-792.

[Bodyanskiy, 2003] Ye. Bodyanskiy, I. Kokshenev, V. Kolodyazhniy. An adaptive learning algorithm for a neo-fuzzy neuron. Proc. 3rd Int. Conf. of European Union Soc. for Fuzzy Logic and Technology (EUSFLAT 2003), Zittau, Germany, 2003, P. 375-379.

[Cortes, 1995] C. Cortes, V. Vapnik. Support vector networks. Machine Learning, 1995, №20, P. 273-297.

[Du, 2014] K.-L. Du, M.N.S. Swamy. Neural Networks and Statistical Learning, London: Springer-Verlag, 2014, 816p.

[Gantmacher, 2000] F.R. Gantmacher. The Theory of Matrices, AMS Chelsea Publishing: Reprinted by American Mathematical Society, 2000, 660p.

[Haykin, 1999] S. Haykin. Neural Networks. A Comprehensive Foundation, Upper Saddle River, N.J.: Prentice Hall, 1999, 842 p.

[Jang, 1993] J.-S. R. Jang. ANFIS: Adaptive-network-based fuzzy inference systems. IEEE Trans. Syst., Man., and Cybern., 1993, № 23, P. 665-685.

[Jang, 1997] J.-S. R. Jang, C. T. Sun, E. Mizutani. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, N.J.: Prentice Hall, 1997, 640 p.

[Lin, 2002] Ch.-F. Lin, Sh.-D. Wang. Fuzzy Support Vector Machines. IEEE Trans. on Neural Networks, 2002, №13, P. 646-671.

[Ljung, 1999] L. Ljung. System Identification: Theory for the User, N.Y.: Prentice-Hall, 1999, 519p.

[Miki, 1999] T. Miki, T. Yamakawa. Analog implementation of neo-fuzzy neuron and its on-board learning. Computational Intelligence and Applications, ed. by N. E. Mastorakis, Piraeus: WSES Press, 1999, P. 144-149.

[Nelles, 2001] O. Nelles. Nonlinear System Identification, Berlin: Springer, 2001, 785p.

[Suykens, 2002] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle. Least Squares Support Vector Machines, Singapore: World Scientific, 2002, 294p.

[Takagi, 1985] T. Takagi, M. Sugeno. Fuzzy identification of systems and its application to modelling and control. IEEE Trans. Syst., Man., and Cybern., 1985, №15, P. 116-132.

[Uchino, 1997] E. Uchino, T. Yamakawa. Soft computing based signal prediction, restoration and filtering. Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks and Genetic Algorithms, ed. by Da Ruan, Boston: Kluwer Academic Publisher, 1997, P. 331-349.

[Vapnik, 1974] V.N. Vapnik, A.Ya. Chervonenkis. Pattern Recognition Theory (statistical learning problems), M.: Nauka, 1974, 416p. (in Russian)

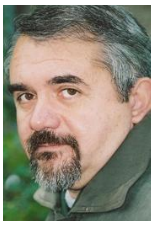
[Vapnik, 1979] V.N. Vapnik, A.Ya. Chervonenkis. Empirical data dependency restoration, M.: Nauka, 1979, 448p. (in Russian)

[Vapnik, 1995] V.N. Vapnik. The Nature of Statistical Learning Theory, N.Y.: Springer, 1995, 188p.

[Yamakawa, 1992] T. Yamakawa, E. Uchino, T. Miki, H. Kusanagi. A neo fuzzy neuron and its applications to system identification and prediction of the system behavior. Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks "IIZUKA-92", Iizuka, Japan, 1992, P. 477-483.

[Zahirniak, 1990] D. Zahirniak, R. Chapman, S.K. Rogers, B.W. Suter, M. Kabrisky, V. Pyati. Pattern recognition using radial basis function network. Application of Artificial Intelligence Conf., Dayton, OH, 1990, P. 249-260.

Authors' Information



Yevgeniy Bodyanskiy – Professor, Dr. – Ing. habil., The Head of Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, 14 Lenin Ave., Office 511, 61166 Kharkiv, Ukraine; e-mail: bodya@kture.kharkov.ua

Major Fields of Scientific Research: Artificial neural networks, Fuzzy systems, Hybrid systems of computational intelligence, Cascade neuro-fuzzy systems

Oleksii Tyshchenko - Ph.D., Senior Researcher at Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, 14 Lenin Ave., Kharkiv, 61166, Ukraine; e-mail: lehatish@gmail.com

Major Fields of Scientific Research: Artificial neural networks, Hybrid systems of computational intelligence, Reservoir Computing, Cascade neuro-fuzzy systems

Daria Kopaliani – PhD student, Kharkiv National University of Radio Electronics, 14 Lenin Ave., Office 517, 61166 Kharkiv, Ukraine; e-mail: daria.kopaliani@gmail.com

Major Fields of Scientific Research: Artificial neural networks, Cascade neuro-fuzzy systems