ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 285-290

Parameter–Estimation–Based Learning for Feedforward Neural Networks: Convergence and Robustness Analysis

A. Alessandri [1], M. Maggiore [2], M. Sanguineti [3]

[1] Naval Automation Institute, CNR–IAN National Research Council,
Via De Marini 6, 16149 Genova, Italy
[2] The Ohio State University, Department of Electrical Engineering,
330 Caldwell Lab., 2015 Neil Av., Columbus, OH 43210-1272, USA
[3] Department of Communications, Computer, and System Sciences
DIST–University of Genoa, Via Opera Pia 13, 16145 Genova, Italy

Abstract. In this paper, a new algorithm for the learning of feedforward neural networks is presented. Stating the learning process for feedforward neural networks as a parameter estimation problem, we develop a new algorithm and provide an analysis of its convergence and robustness properties. The simulation results, for both classification and function approximations problems, confirm the effectiveness of the proposed algorithm, whose behaviour with respect to error back–propagation and extended Kalman filter–based learning are discussed.

1. Introduction

In the literature on neural networks, many efforts have been reported to improve the performances of learning algorithms: nonlinear optimization techniques for computing search directions more effective than the steepest descent, algorithms using also the second derivatives, adding noise during the training, heuristic acceleration techniques, etc. By considering a feedforward neural network as a nonlinear system having a layered structure, its learning algorithm can be regarded as a parameter estimation problem for such a system. Following this approach, training algorithms based on the extended Kalman filter (EKF) have been developed [1]. They show faster convergence than backpropagation (BP) and avoid tuning parameters, at the expense of increased computational burden (matrix inversions are needed) and large amount of memory, required for storing the covariance matrix.

In this paper, after stating the learning of feedforward neural networks as a parameter estimation problem, we develop a new batch learning algorithm, and demonstrate its convergence and robustness properties. Unlike BP, the algorithm shows fast convergence and, unlike the EKF–based training, it does not require successive linearizations. The algorithm works according to a sliding–window scheme (only a portion of the data is processed at every time, and the past is summarized in one prediction) is computationally tractable.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 285-290

2. The learning algorithm

We consider multilayer feedforward neural networks containing only one hidden layer, composed of ν neural units, with the hyperbolic tangent as activation function; the output layer is linear. The function implemented by such a network is denoted by $\underline{\gamma}(\underline{w}, \underline{u})$, where $\underline{w}$ and $\underline{u}$ are, respectively, the weights and the input vector. The data set consists of P input/output pairs $(\underline{u}_t, \underline{y}_t)$, $t = 0, 1, \ldots, P-1$, where $\underline{u}_t \in \Re^m$ and $\underline{y}_t \in \Re^p$ represent, respectively, the input and the desired output to the network. We let the weights of the network constitute the state of the following nonlinear system ($t = 0, 1, \ldots P-1$):

$$\begin{cases} \underline{w}_{t+1} = \underline{w}_t \\ \underline{y}_t = \underline{\gamma}\left(\underline{w}_t, \underline{u}_t\right) + \underline{\eta}_t \end{cases} \tag{1}$$

where $\underline{w}_0 = \underline{w}_1 = \ldots = \underline{w}_{P-1} \triangleq \underline{w}$, $\underline{\eta}_t \in \Re^p$ is a random noise whose statistics are unknown, and it is supposed $\underline{\eta}_t \in K \subset \Re^p$, where K is a compact set. We assume that $\underline{w} \in W$ and $\underline{u}_t \in U$, where W and U are compact sets. If we consider the data set $(\underline{u}_t, \underline{y}_t)$, $t = 0, 1, \ldots, P-1$, as generated by a process governed by an unknown function $\underline{f} : \Re^m \longrightarrow \Re^p$, i.e., $\underline{y}_t = \underline{f}(\underline{u}_t)$, then $\underline{\eta}_t$ is the error achieved by the approximator $\underline{\gamma}$ in the approximation of $\underline{f}$, in correspondence of the value $\underline{u}_t$. The possibility of writing (1) is guaranteed by the universal approximation properties of feedforward neural networks, provided that the unknown function $\underline{f}$ be sufficiently smooth (for instance, [2]).

The algorithm we propose for the network training is based on a sliding-window state estimator for system (1) [3]. According to this, we estimate the constant parameters vector $\underline{w}$ in such a way as to minimize the cost

$$J_t \triangleq \mu \|\hat{\underline{w}}_t - \bar{\underline{w}}_t\|^2 + \sum_{i=t-N}^{t} \left\|\underline{y}_i - \underline{\gamma}\left(\hat{\underline{w}}_t, \underline{u}_i\right)\right\|^2 \quad t = N, N+1, \ldots, P-1 \tag{2}$$

where the integer N is the dimension of the sliding-window, $\hat{\underline{w}}_t$ is the estimate of the weights vector $\underline{w}$ at the time t and $\bar{\underline{w}}_t$ is the "a priori" information on the value of $\hat{\underline{w}}_t$. Minimization of J_t at each temporal stage leads to a sequential state estimator; the optimal estimate $\hat{\underline{w}}_t^\circ$ of $\underline{w}$ at time t, given the "information set" $I_t^N \triangleq \left\{\underline{y}_{t-N}, \cdots, \underline{y}_t, \underline{u}_{t-N}, \cdots \underline{u}_t, \bar{\underline{w}}_t\right\}$, is $\hat{\underline{w}}_t^\circ \triangleq \operatorname{argmin}_{\underline{\hat{w}}_t} J_t\left(\hat{\underline{w}}_t, I_t^N\right)$, where $\bar{\underline{w}}_t \triangleq \hat{\underline{w}}_{t-1}^\circ$, and $\bar{\underline{w}}_N$ is the initial prediction. Let W_t° be the sets of the optimal estimates that minimize the cost (2), $Y \triangleq \underline{\gamma}(W, U)$, $Y \subset \Re^p$ and $\underline{H}\left(\underline{w}, \underline{u}_{t-N}^t\right) \triangleq \operatorname{col}\left[\underline{\gamma}\left(\underline{w}, \underline{u}_\tau\right),\ \tau = t-N, \ldots, t\right]$, where $\underline{u}_{t-N}^t \triangleq \operatorname{col}\left(\underline{u}_{t-N}, \underline{u}_{t-N+1}, \cdots, \underline{u}_t\right)$.

Assumption A. There exists a compact set $\tilde{W}$ such that $\tilde{W} \supseteq W \bigcup \left(\bigcup_{t=0}^{+\infty} W_t^\circ\right)$.

Assumption B. There exists an integer N such that, for any $\underline{u}_{t-N}^t \in U^{N+1}$, the mapping $\underline{H}\left(\underline{w}, \underline{u}_{t-N}^t\right) : \mathcal{W} \longrightarrow \Re^{p(N+1)}$ is an injective immersion (i.e., the Jacobian matrix must have rank n), where $\mathcal{W}$ is the closed convex hull of $\tilde{W}$ (such a $\tilde{W}$ exists in virtue of Assumption A).

Given a symmetric positive definite matrix A, we denote by $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ its minimum and maximum eigenvalue, respectively; for a generic matrix B, $\|B\|_{\max} \triangleq \|B\| = \sqrt{\lambda_{\max}(B^T B)}$ and $\|B\|_{\min} \triangleq \sqrt{\lambda_{\min}(B^T B)}$. Moreover, let $D(\underline{w}, \underline{u}_{t-N}^t) \triangleq \frac{\partial \underline{H}}{\partial \underline{w}} \in \Re^{p(N+1)\times n}$, $\underline{w} \in \mathcal{W}$ and $\Delta \triangleq \max_{\underline{w}\in\mathcal{W}, \underline{u}_{t-N}^t \in U^{N+1}} \|D(\underline{w}, \underline{u}_{t-N}^t)\|$, $\delta \triangleq \min_{\underline{w}\in\mathcal{W}, \underline{u}_{t-N}^t \in U^{N+1}} \|D(\underline{w}, \underline{u}_{t-N}^t)\|$. Let $r_\eta \triangleq \max_{\underline{\eta}_{t-N}, \cdots, \underline{\eta}_t \in E^{N+1}} \|\operatorname{col}(\underline{\eta}_{t-N}, \ldots, \underline{\eta}_t)\|$

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 285-290

and $\bar{k} \triangleq k\sqrt{N+1}$, where $k \in \Re^+$ is a suitable scalar such that $\left\| \frac{\partial \underline{\gamma}(\underline{w},\underline{u})}{\partial \underline{w}}(\underline{w}',\underline{u}) - \frac{\partial \underline{\gamma}(\underline{w},\underline{u})}{\partial \underline{w}}(\underline{w}'',\underline{u}) \right\| \leq k \, \|\underline{w}' - \underline{w}''\| \; \forall \underline{w}', \underline{w}'' \in \mathcal{W}, \forall \underline{u} \in U$. The following theorem guarantees, under suitable hypotheses, the boundedness of the estimation error (for the proof, see [4]).

Theorem 1. *Suppose that Assumptions A and B are verified. Let* $\underline{\hat{e}}_t \triangleq \underline{w} - \underline{\hat{w}}_t^\circ$ *and consider the largest closed ball* $N(\hat{r}_e)$ *with radius* $\hat{r}_e$ *and center in the origin such that* $\underline{\hat{e}}_N \in N(\hat{r}_e)$. *If there exists a choice of* μ, *for which the inequality*

$$\left(\delta^4 - 8\bar{k}\Delta^2 r_\eta\right)\mu + \delta^6 > 0 \tag{3}$$

is satisfied, the second-order equation $(2\Delta\bar{k}\mu^2)\xi^2 + [\mu(\mu+\delta^2)^2 - (\mu+\delta^2)^3 + 4\bar{k}\Delta^2\mu r_\eta]\xi + 2\bar{k}\Delta^3 r_\eta^2 + \Delta r_\eta(\mu+\delta^2)^2 = 0$ *has the two real positive roots* ξ^- *and* ξ^+, *with* $\xi^- < \xi^+$.

Then, if the choice of μ *yields also the fulfillment of the inequality*

$$\mu^2 + 2\left(\delta^2 - \bar{k}\,\Delta\,\xi^+\right)\mu + \left(\delta^4 - 2\bar{k}\,\Delta^2\,r_\eta\right) > 0 \tag{4}$$

we have $\lim_{t\to+\infty} \|\underline{\hat{e}}_t\| \leq \xi^-, \quad \forall \hat{r}_e < \xi^+$

Moreover, considering the sequence ξ_t *generated from the discrete-time system* $(\mu+\delta^2)^3\xi_t = (2\Delta\bar{k}\mu^2)\xi_{t-1}^2 + [\mu(\mu+\delta^2)^2 + 4\bar{k}\Delta^2\mu r_\eta]\xi_{t-1} + 2\bar{k}\Delta^3 r_\eta^2 + \Delta r_\eta(\mu+\delta^2)^2$, *where* $\xi_N \triangleq \|\underline{\hat{e}}_N\|$, *we have* $0 \leq \|\underline{\hat{e}}_t\| \leq \xi_t, \forall t \geq N$, *i.e., the sequence* ξ_t *constitutes an upper bound to the norm of the error dynamics.*

□

For large r_η, (3) imposes an upper bound to μ, whereas (4) is very likely to impose a lower bound. In other words, the theorem guarantees the boundedness of the estimation error $\forall \mu \in (\mu^-, \mu^+)$, where μ^- and μ^+ are suitable scalars. Assumption B is strictly related to an important property of networks. For $\underline{H}(\underline{w}, \underline{u}_{t-N}^t)$ to be an immersion, the Jacobian matrix $D(\underline{w}, \underline{u}_{t-N}^t)$ must have rank n, $\forall \underline{u}_{t-N}^t \in U^{N+1}$. Since $D\left(\underline{w}, \underline{u}_{t-N}^t\right) \in \Re^{p(N+1)\times n}$, the bigger $p(N+1)$ is, the easier is for the above matrix to have rank n. According to this, we require that $p(N+1) \gg n$, i.e., we choose a very large window size, compared to the size of the weights vector. This is related to the estimation error of a network: since it is decreasing with P and, unlike the approximation error, increasing with n [2], in order to minimize it we would require $P \gg n$: this follows from $p(N+1) \gg n$, noticing that $P \geq N+1$ (the choice of N influences the condition on the rank of $D\left(\underline{w}, \underline{u}_{t-N}^t\right)$). As regards the injectivity of $\underline{H}$, we observe that the mapping $\underline{\gamma} : \mathcal{W} \times U \to Y$ is not injective, due to the invariance of $\underline{\gamma}(\underline{w}, \underline{u})$ to certain permutations of the weights [5]. Since we only require the estimator to find one of the weights vectors that minimize the cost (2) (considering as equivalent all those obtained through permutations), we can neglect the injectivity assumption.

The minimization of (2) can be carried out using a nonlinear programming iterative algorithm that asymptotically converges to $\underline{\hat{w}}_t^\circ$. By considering only a finite number of iterations we find an approximate value of $\underline{\hat{w}}_t^\circ$, that we denote by $\underline{\tilde{w}}_t$. Moreover, let $\underline{\varepsilon}_t \triangleq \underline{\hat{w}}_t^\circ - \underline{\tilde{w}}_t$ and $\bar{\varepsilon} \triangleq \max_{t\in\{N,N+1,\ldots,P-1\}}(\|\underline{\varepsilon}_t\|)$. The next theorem concerns the robustness of the proposed training algorithm: it states that, if the error in the estimate of the weights vector is suitably bounded, then the boundness of the estimate is preserved (for the proof, see [4]).

Theorem 2. *Suppose that Assumptions A and B are verified. Let* $\underline{\tilde{e}}_t \triangleq \underline{w} - \underline{\tilde{w}}_t$ *and consider the largest closed ball* $N(\tilde{r}_e)$ *with radius* $\tilde{r}_e$ *and center in the origin such that* $\underline{\tilde{e}}_N \in N(\tilde{r}_e)$. *If there exists a choice of* μ, *for which the inequality*

$$[-16\bar{k}\Delta\,\bar{\varepsilon}]\,\mu^2 + (\delta^4 - 8\bar{k}\Delta^2 r_\eta - 16\bar{k}\Delta\delta^2\bar{\varepsilon})\,\mu + \delta^6 > 0 \qquad (5)$$

is satisfied, the second–order equation $(2\bar{k}\Delta\mu^2)\hat{\xi}^2 + \left[\mu(\mu+\delta^2)^2 - \mu(\mu+\delta^2)^3 + 4\bar{k}\Delta^2\mu r_\eta + \left(8\bar{k}\Delta\mu(\mu+\delta^2) + 4\bar{k}\Delta\mu^2\right)\bar{\varepsilon}\right]\hat{\xi} + 2\bar{k}\Delta^3 r_\eta^2 + \bar{k}(\mu+\delta^2)^2 r_\eta + [8\bar{k}\Delta(\mu+\delta^2)^2 + 2\bar{k}\Delta\mu^2 + 8\bar{k}\Delta\mu(\mu+\delta^2)]\bar{\varepsilon}^2 + [8\bar{k}\Delta^2(\mu+\delta^2) + 4\bar{k}\Delta^2\mu]\bar{\varepsilon}r_\eta + [(\mu+\delta^2)^3 + \mu(\mu+\delta^2)^2]\bar{\varepsilon} = 0$ *has the two real positive roots* $\hat{\xi}^-$ *and* $\hat{\xi}^+$, *with* $\hat{\xi}^- < \hat{\xi}^+$. *Then, if the choice of* μ *yields also the fulfillment of the inequality*

$$\mu^2 + 2\left(\delta^2 - \bar{k}\Delta\,\hat{\xi}^+ - 3\bar{k}\Delta\bar{\varepsilon}\right)\mu + \left(\delta^4 - 2\bar{k}\Delta^2\,r_\eta - 4\bar{k}\Delta\delta^2\,\bar{\varepsilon}\right) > 0 \qquad (6)$$

we have $\lim_{t\to+\infty} \|\tilde{\varepsilon}_t\| \le \hat{\xi}^-, \quad \forall\, \tilde{r}_e < \hat{\xi}^+$

□

3. Simulation results and conclusions

The proposed algorithm, called WEST (Weights ESTimator), has been compared, for both classification and function approximation, with the following learning algorithms: BPX (error back-propagation, with momentum and adaptive learning rate), LM (Levenberg-Marquardt optimization [6]), EKF (iterative extended Kalman filter-based training). The simulations have been performed with the Neural Network Toolbox of MATLAB [7]. For BPX and LM we have used, respectively, the functions *trainbpx* and *trainlm* of MATLAB; EKF and WEST have been properly implemented. To perform the minimization of the cost function (2), we have used the LM algorithm of MATLAB.

As regards *classification* problems, as testbed for our algorithm we have chosen a generalized 2-input XOR in the domain $[0,1]\times[0,1]$. We have used a network with 2 hidden neurons, trained through 2000 points uniformly distributed in the domain. The output corresponding to a class has been set to 0.9 when the input vector belongs to that class, -0.9 otherwise. For each algorithm 10 trials have been performed, with different random initializations of the weights; the final results are an average of those obtained in the runs that reached a global minimum, where a global minimum is considered reached when the mean squared error is under an heuristically fixed threshold (otherwise, we say that the network occurred in a local minimum). The test set consists of 11×11 uniformly distributed points; the temporal window N for WEST has been chosen equal to 99. As can be seen in Figure 1, the best error convergence is for LM and WEST (note that, to reduce the computational effort for EKF, the MSE has been calculated only every 200 patterns). If we consider the computational burden and the memory requirements of LM [7], it turns out that WEST is preferable; moreover, note that our algorithm is the only one that never stops at a local minimum.

As regards function approximation, we have trained a network with 30 hidden units to approximate the function $f(x,y) = \frac{\sqrt{x^2+y^2}}{1+\frac{1}{3}(x^2+y^2)}$ in the domain $[-10,10]\times[-10,10]$ (see Figure 2).
The training set has been generated considering 2000 input/output pairs uniformly distributed in the domain; for the window size of WEST we have chosen $N = 249$. In Figure 3 (a) to (d), we show the approximation of the function obtained through BPX, LM, EKF, and WEST training, respectively. Note that only LM and WEST succeed in detecting the minimum of f; the behavior of the MSE networks error, not shown owing to the lack of space, is analogous to that presented for the XOR problem ([4]). The lowest value of the MSE on the test set is for LM (see the table),

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 285-290

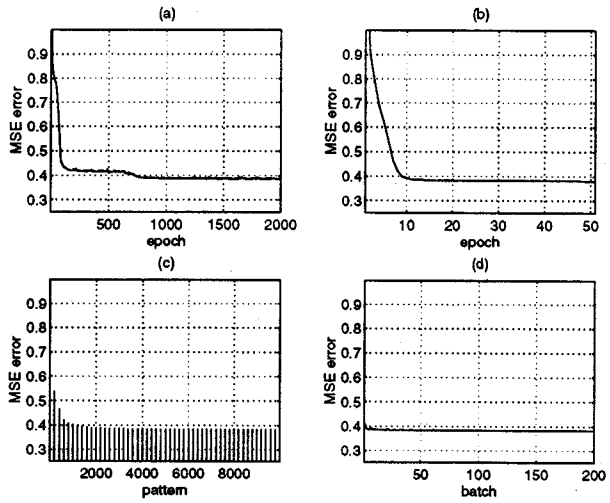

Figure 1: mean-squared error for BPX (a), LM (b), EKF (c), and WEST (d) training algorithms, in the classification problem.

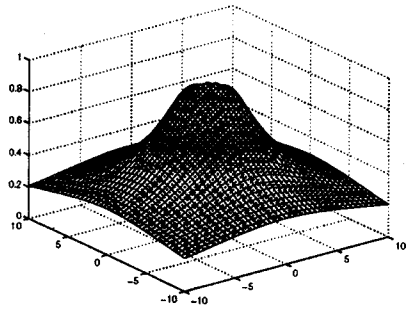

Figure 2: the function to be approximated

but WEST is very near; moreover, it provides an approximating function as good as the one of LM, but obtained with a reduced computational load and memory requirements (at each iteration, it deals with a 121×250 Jacobian matrix, whereas LM with a 121×2000 one).

type of training	XOR		approximation
	MSE	*local minima*	*MSE*
BPX	3.18	3/10	0.0013
LM	3.14	4/10	0.0007
EKF	3.11	5/10	0.0011
WEST	3.16	0/10	0.0008

To conclude, we summarize the most interesting features of the proposed new training algorithm: bounded network error (see Theorem 1), robustness (see Theorem

2), reduced computational burden, no need for successive linearization.

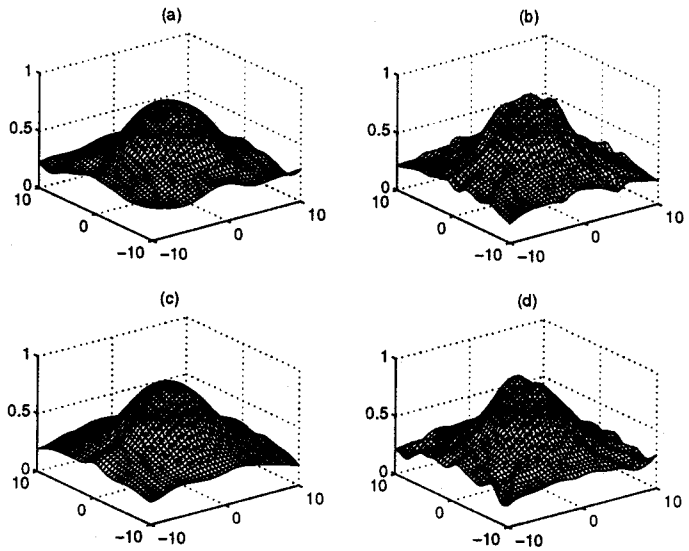

Figure 3: approximation of the function obtained through BPX (a), LM (b), EKF (c), and WEST (d) training algorithms.

References

[1] S. Singhal and L. Wu, "Training multylayer perceptrons with the extended Kalman algorithm", in *Advances in Neural Information Processing Systems 1*, D. S. Touretzky, Ed., 1989, pp. 133–140.

[2] A. R. Barron, "Approximation and estimation bounds for artificial neural networks", *Machine Learning*, vol. 14, pp. 115–133, 1994.

[3] A. Alessandri, M. Maggiore, T. Parisini, and R. Zoppoli, "Neural approximators for nonlinear sliding-window state observers", in *Proceedings of the 35th IEEE Conference on Decision and Control*, 1996, pp. 1461–1463.

[4] A. Alessandri, M. Maggiore, and M. Sanguineti, "A new learning algorithm with bounded error for feedforward neural networks", Tech. Rep. 97/1, DIST Tech. Report, 1997.

[5] H. J. Sussmann, "Uniqueness of the weights for minimal feedforward nets with a given input–output map", *Neural Networks*, vol. 5, pp. 589–593, 1992.

[6] J. J. More, "The Levenberg-Marquardt algorithm: Implementation and theory", in *Numerical Analysis*, G. A. Watson, Ed., Lecture Notes in Mathematics 630, pp. 105–116. Springer-Verlag, 1977.

[7] H. Demuth and M. Beale, *Neural Network Toolbox - For Use With MATLAB - User's Guide*, The Math Works, Inc., 1995.