ON-LINE LEARNING IN RECURRENT NEURAL NETWORKS USING NONLINEAR KALMAN FILTERS

Branimir Todorović[1]*, *Miomir Stanković*[1], *Claudio Moraga*[2]

[1] Faculty of Occupational Safety, University of Niš, 18000 Niš, Serbia and Montenegro
E-mail: bssmtod@EUnet.yu
[2] Department of Computer Science, University of Dortmund, Germany
E-mail: moraga@cs.uni-dortmund.de

ABSTRACT

The extended Kalman filter has been successfully applied to the feedforward and the recurrent neural network training. Recently introduced derivative-free filters (Unscented Kalman Filter and Divided Difference Filter) outperform the extended Kalman filter in nonlinear state estimation. In the parameter estimation of the feedforward neural networks UKF and DDF are comparable or slightly better than EKF, with a significant advantage that they do not demand calculation of the neural network Jacobian. In this paper, we consider the application of EKF, UKF and DDF to the recurrent neural network training. The class of non-linear autoregressive recurrent neural networks with exogenous inputs is chosen as a basic architecture due to its powerful representational capabilities

1. INTRODUCTION

The Extended Kalman Filter (EKF) has been accepted as effective and easy to implement method for state and parameter estimation. It has been applied with success to the feedforward neural network training [1,6] as well as to the recurrent neural network training[11]. It was shown that statistics estimated by the EKF can be used to sequentially estimate the structure (number of hidden neurons and connections) and parameters of feed-forward [6] and recurrent [7] Radial Basis Function (RBF) networks.

Estimators like the Unscented Kalman Filter (UKF) [2,3] and the Divided Difference Filter (DDF) [5], have been introduced recently as an outperforming alternative to EKF for nonlinear state estimation. In parameter estimation of the feedforward neural networks UKF and DDF are shown to be comparable or slightly better than EKF [8], with a significant advantage that they do not demand calculation of the neural network Jacobian.

In this paper we shall consider the training of a Non-linear AutorRegressive with eXogenous inputs (NARX) recurrent neural networks, using DDF, UKF and EKF. The class of NARX recurrent neural ntworks is chosen since it is shown in [4] that they outperform classical, fully connected, recurrent neural networks in tasks that involve long term dependencies for which the desired output depends on inputs presented at times far in the past.

2. NARX RECURRENT NEURAL NETWORK

A NARX model of a dynamic system is given by:

$$s_k = f(s_{k-1},...,s_{k-\Delta_s}, u_{k-1},...u_{k-\Delta_u}) \qquad (1)$$

where s_k corresponds to the true (noiseless) output of the system, u_k is the known input at time step k, Δ_u and Δ_s are the input and the output order, and $f(\cdot)$ is a non-linear function. We shall consider a NARX model for which f is implemented either using a Multilayer Perceptron (we shall name it NARX Recurrent Multilayer Perceptron – NARX_RMLP) or using a radial basis function network (NARX Recurrent Radial Basis Function network – NARX_RRBF). For comparison purposes, we shall assume that both models have two layers of neurons (Figure 1), with an output layer having a linear activation function.

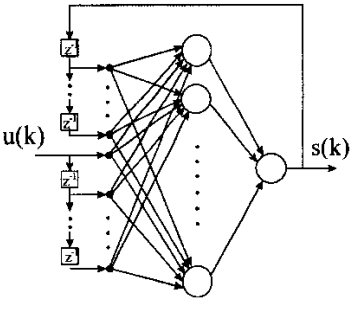


Figure 1: A NARX recurrent nerual network

The output of the i-th hidden neuron of a NARX_RMLP network is given by:

$$\phi_i(\mathbf{s}_{k-1}, \mathbf{u}_{k-1}, \mathbf{b}_i) = \tanh\left\{ b_{i0} + \sum_{l=1}^{\Delta_s} b_{il} s_{k-l} + \sum_{j=1}^{n_u} \sum_{\tau=1}^{\Delta_u} b_{ij\tau} u_{j,k-\tau} \right\} \qquad (2)$$

where $\mathbf{s}_{k-1} = [s_{k-1} \cdots s_{k-\Delta_s}]^T_{1\times\Delta_s}$ denotes the vector of previous network outputs, $\mathbf{u}_{k-1} = [u_{k-1}^T \cdots u_{k-\Delta_u}^T]^T_{1\times\Delta}$ is the vector of previous inputs and $\mathbf{b}_i = [b_{i0} b_{i1} ... b_{i\Delta_s} b_{i11} ... b_{in_u\Delta_u}]^T$ denotes the vector of hidden neuron weights.

* The work of B. Todorović was supported by a Scholarship of the German Academic Exchange Service (DAAD) under the Stability Pact for South East Europe.

The output of the i-th hidden neuron of a NARX_RRBF network is given by:

$$\phi_i(\mathbf{s}_{k-1},\mathbf{u}_{k-1},\mathbf{b}_i,\mathbf{m}_i)=\exp\left\{-\sum_{l=1}^{\Delta_s}(b_{il}(s_{k-l}-m_{il}))^2 - \sum_{j=1}^{n_u}\sum_{\tau=1}^{\Delta_u}(b_{ij\tau}(u_{j,k-\tau}-m_{ij\tau}))^2\right\} \quad (3)$$

where $\mathbf{m}_i=[m_{i0}m_{i1}...m_{i\Delta_s}m_{i11}...m_{in_u\Delta_u}]^{\mathrm{T}}$ denotes a vector of hidden neuron centers.

The network output is given by:

$$f(\mathbf{s}_{k-1},\mathbf{u}_{k-1},\mathbf{w})=a_0+\sum_{i=1}^{n_H}a_i\phi_i(\mathbf{s}_{k-1},\mathbf{u}_{k-1},\mathbf{b}_i\{,\mathbf{m}_i\}) \quad (4)$$

where $\mathbf{a}=[a_0\ a_1...a_{n_H}]$ are the output weights, w denotes the n_w dimensional vector of unknown network weights, $\mathbf{w}=[\mathbf{a}^{\mathrm{T}}\mathbf{b}^{\mathrm{T}}\{\mathbf{m}^{\mathrm{T}}\}]^{\mathrm{T}}$; n_H is the number of hidden neurons.

2. PARAMETER AND STATE ESTIMATION

2.1. State space model of the NARX recurrent network

Estimation of recurrent neural network parameters can be put in the framework of nonlinear state estimation by defining the state space model of network dynamics. The state vector x is obtained by augmenting the base state s, which is in our case defined as the previous Δ_s outputs of the recurrent network, with the vector of network parameters w.

$$x_k=\Phi(x_{k-1},\mathbf{u}_{k-1})+d_k,\ d_k\sim N(0,Q_k), \quad (5a)$$

$$y_k=H_kx_k+v_k,\ v_k\sim N(0,R_k) \quad (5b)$$

$$x_k=\begin{bmatrix}s_k\\ s_{k-1}\\ \vdots\\ s_{k-\Delta_s+1}\\ \mathbf{w}_k\end{bmatrix},\ \Phi(x_k,\mathbf{u}_k)=\begin{bmatrix}f(\mathbf{s}_k,\mathbf{u}_k,\mathbf{w}_k)\\ s_{k-1}\\ \vdots\\ s_{k-\Delta_s+1}\\ \mathbf{w}_k\end{bmatrix},\ d_k=\begin{bmatrix}d_{s,k}\\ 0\\ \vdots\\ 0\\ d_{w,k}\end{bmatrix},$$

$$Q_k=\begin{bmatrix}Q_{s,k} & 0\\ 0 & Q_{w,k}\end{bmatrix}\ \text{and}\ H=[1\ \ 0\ \ \cdots\ \ 0]_{1\times(\Delta_s+n_w)}.$$

Equation (5a) describes time evolution of the augmented state x, while the observation equation (5b) selects the current output of the network as the observation. The process noise d_k and observation noise v_k are assumed to be mutually independent, white, and Gaussian with known covariances Q_k and R_k respectively.

2.1. Minimum Mean Squared Error estimation

A Minimum Mean Squared Error (MMSE) estimate of the state x_k of a nonlinear discrete time system (5) is such that the estimation error $\tilde{x}_k=x_k-\hat{x}_k$ is unbiased ($E[\tilde{x}_k]=0$) and orthogonal to the observation y_k ($E[\tilde{x}_k y_k^T]=0$). Filters considered in this paper (EKF, DDF and UKF) provide a MMSE estimate of augmented state x_k using "predictor-corrector" scheme.

Given the estimate of the state $\hat{x}_{k-1}$ and its covariance $P_{x,k-1}$, obtained for the set of observations up to the time step $k-1$: $y_{1:k-1}=\{y_i, i=1,...,k-1\}$, the filter predicts the future state using the process model and the knowledge about the process noise distribution. Predicted mean and covariance are ideally:

$$\hat{x}_k^-=E[x_k/y_{1:k-1}] \quad (6a)$$

$$P_{x,k}^-=E[(x_k-\hat{x}_k^-)(x_k-\hat{x}_k^-)^{\mathrm{T}}/y_{1:k-1}] \quad (6b)$$

The estimate $\hat{x}_k$ and its covariance $P_{x,k}$ are obtained by updating (correcting) the state prediction ($\hat{x}_k^-, P_{x,k}^-$) with the current observation y_k:

$$\hat{x}_k=\hat{x}_k^-+K_k(y_k-\hat{y}_k^-) \quad (7a)$$

$$K_k=P_{xy,k}P_{y,k}^{-1} \quad (7b)$$

$$P_{x,k}=P_{x,k}^--K_kP_{y,k}^{-1}K_k^T \quad (7c)$$

$\hat{y}_k^-=E[y_k/y_{1:k-1}]$, $P_{y,k}=E[(y_k-\hat{y}_k^-)(y_k-\hat{y}_k^-)^T/y_{1:k-1}]$ are observation prediction and its covariance, and $P_{xy,k}=E[(x_k-\hat{x}_k^-)(y_k-\hat{y}_k^-)^T/y_{1:k-1}]$ is the cross-correlation. These equations depend on predicted values of the first two moments of x_k and y_k, given set of observations $y_{1:k-1}$. Due to the linearity of the observation equation (5b) we have:

$$\hat{y}_k^-=H_k\hat{x}_k^- \quad (8a)$$

$$P_{y,k}=H_kP_{x,k}^-H_k^{\mathrm{T}}+R_k \quad (8b)$$

$$P_{xy,k}=P_{x,k}^-H_k^{\mathrm{T}} \quad (8c)$$

and the problem is reduced to the propagation of a state x_{k-1} through the nonlinear dynamic equation (5a) in order to obtain prediction ($\hat{x}_k^-, P_k^-$).

3. DERIVATIVE-FREE NONLINEAR FILTERS

In this section, we shall consider three different approaches to nonlinear state estimation and apply them to the estimation of the NARX recurrent neural networks. As we saw in the previous section, the problem that remains to be solved is the estimation of a statics of a random variable propagated trough the nonlinear transformation. Let us define the problem in a general form. Suppose that x is a random variable with mean $\hat{x}$ and covariance P_x. A random variable y is related to x through the nonlinear function $y=f(x)$. We wish to calculate the mean $\hat{y}$ and covariance P_y of y. (Note that the derived solutions could be easily be applied to state prediction (6) by introducing substitutions $x\rightarrow x_{k-1}$ and $y\rightarrow x_k$.)

Extended Kalman filter is based on multidimensional Taylor series expansion of $f(x)$. We shall consider only the first order EKF, obtained by excluding nonlinear terms of Taylor series expansion:

$$f(x)=f(\hat{x}+\Delta x)\approx f(\hat{x})+f_x'(\hat{x})\Delta x \quad (9)$$

where $f'_x(\hat{x}) = \partial f / \partial x\big|_{x=\hat{x}}$ and Δx is zero mean radnom variable with covariance P_x. In that case we have:

$$\hat{y} \overset{\Delta}{=} E[f(x)] \approx f(\hat{x}) \tag{10a}$$

$$P_y \overset{\Delta}{=} E[(f(x)-\hat{y})(f(x)-\hat{y})^{\mathrm{T}}] \approx f'_x(\hat{x}) P_x (f'_x(\hat{x}))^{\mathrm{T}} \tag{10b}$$

3.1 Divided difference filter

In [5] Nørgaard et al. proposed a new set of estimators based on polynomial approximation of nonlinear transformations using multidimensional extension of Stirling's interpolation formula. Stirling interpolation formula is particularly simple if only first and second order polynomial approximation are considered:

$$f(x) \approx f(\hat{x}) + \tilde{D}_{\Delta x} f + \tilde{D}^2_{\Delta x} f \tag{11}$$

Divided difference operators are defined by:

$$\tilde{D}_{\Delta x} f = \frac{1}{h}\left(\sum_{p=1}^{n} \Delta x_p \mu_p \delta_p\right) f(\bar{x}) \tag{12a}$$

$$\tilde{D}^2_{\Delta x} f = \frac{1}{h^2}\left(\sum_{p=1}^{n} \Delta x_p^2 \delta_p^2 + \sum_{p=1}^{n}\sum_{\substack{q=1,\\ q\neq p}}^{n} \Delta x_p \Delta x_q (\mu_p \delta_p)(\mu_q \delta_q)\right) f(\bar{x}) \tag{12b}$$

where δ_p is a "partial" difference operator:

$$\delta_p f(\hat{x}) = f(\hat{x} + 0.5 \cdot h \cdot e_p) - f(\bar{x} - 0.5 \cdot h \cdot e_p) \tag{13}$$

and μ_p is an average operator:

$$\mu_p f(\hat{x}) = 0.5 \cdot (f(\hat{x} + 0.5 \cdot h \cdot e_p) + f(\hat{x} - 0.5 \cdot h \cdot e_p)). \tag{14}$$

and e_p is the pth unit vector.

Applying a stochastic decoupling of the variables in x by the following transformation $z = S_x^{-1} x$, (S_x is the Cholesky factor of the covariance matrix $P_x = S_x S_x^T$), Nørgaard et al. derived approximation of mean and covariance of $y = f(x)$ [5]:

$$\hat{y} = \frac{h^2 - n}{h^2} f(\bar{x}) + \frac{1}{2h^2}\sum_{p=1}^{n} (f(\bar{x} + h s_{x,p}) + f(\bar{x} - h s_{x,p})) \tag{15a}$$

$$P_y = \frac{1}{4h^2}\sum_{p=1}^{n} (f(\bar{x} + h s_{x,p}) - f(\bar{x} - h s_{x,p})) \cdot (f(\bar{x} + h s_{x,p}) - f(\bar{x} - h s_{x,p}))^{\mathrm{T}} + \frac{h^2 - 1}{4h^4}\sum_{p=1}^{n} (f(\bar{x} + h s_{x,p}) + f(\bar{x} - h s_{x,p}) - 2f(\bar{x})) \cdot (f(\bar{x} + h s_{x,p}) + f(\bar{x} - h s_{x,p}) - 2f(\bar{x}))^{\mathrm{T}} \tag{15b}$$

Interval length h is set equal to the kurtosis of the prior random variable x. For Gausians, $h^2 = 3$.

3.1 Unscented Kalman filter

Julier and Uhlman proposed the Unscented Transformation (UT) [2,3] in order to calculate the statistics of a random variable x propagated through nonlinear function $y = f(x)$. The n_x dimensional continuous random variable x with mean $\hat{x}$ and covariance P_x is approximated by $2n_x + 1$ sigma points X_p with corresponding weights ω_p, $p = 0,1,...,2n_x$:

$\mathsf{X}_0 = \hat{x}$, $\omega_0 = \lambda/(n+\lambda)$, $\lambda = \alpha^2(n_x + \kappa) - n_x$ f

or $p = 1,2...,n$

$$\mathsf{X}_p = \hat{x} + \sqrt{n+\lambda}\cdot s_{x,p} \qquad \omega_p = 0.5/(n+\lambda)$$
$$\mathsf{X}_{p+n_x} = \hat{x} - \sqrt{n+\kappa}\cdot s_{x,p} \quad \omega_{p+n_x} = 0.5/(n+\lambda) \tag{16}$$

where α determines the spread of the sigma points around $\hat{x}$ (usually $1.e-4 \le \alpha \le 1$) and $\kappa \in \Re$ is the scaling parameter, usually set to 0 or $3 - n_x$ [3]. $s_{x,p}$ is the pth row or column of the matrix square root of P_x

Each sigma point is instantiated through the function $f(\cdot)$ to yield the set of transformed sigma points $\mathsf{Y}_i = f(\mathsf{X}_i)$, and the mean $\hat{y}$ of a transformed distribution is estimated by:

$$\hat{y} = \sum_{p=0}^{2n_x} \omega_p \mathsf{Y}_p = \frac{\lambda}{n+\lambda} f(\hat{x}) + \frac{1}{2(n+\lambda)}\sum_{i=1}^{n} (f(\hat{x} + \sqrt{n+\lambda}\cdot s_{x,i}) + f(\hat{x} - \sqrt{n+\lambda}\cdot s_{x,i})) \tag{17}$$

The covariance estimate obtained by unscented transform is:

$$P_y = \sum_{p=0}^{2n_x} \omega_p (\mathsf{Y}_p - \hat{y})(\mathsf{Y}_p - \hat{y})^{\mathrm{T}} = \frac{\lambda}{n+\lambda}(f(\hat{x}) - \hat{y}) f(\hat{x}) - \hat{y})^{\mathrm{T}} + \frac{1}{2(n+\lambda)}\sum_{p=1}^{n} (f(\hat{x} + \sqrt{n+\lambda}\cdot s_{x,p}) - \hat{y}) f(\hat{x} + \sqrt{n+\lambda} s_{x,p}) - \hat{y})^{\mathrm{T}} + \frac{1}{2(n+\lambda)}\sum_{p=1}^{n} (f(\hat{x} - \sqrt{n+\lambda} s_{x,p}) - \hat{y}) f(\hat{x} - \sqrt{n+\lambda} s_{x,p}) - \hat{y})^{\mathrm{T}} \tag{18}$$

Estimation of states and parameters of NARX recurrent networks (state space model given by (5)) using unscented Kalman filter, consists in applying unscented transformation to a dynamic equation (5a) in order to obtain prediction ($\hat{x}_k^-$, P_k^-). Predicted statistics are updated with the current observation y_k applying equations (7).

4. EXPERIMENTS

In this section, we shall give the results of time series prediction using NARX recurrent neural networks trained using EKF, DDF and UKF. The time series is obtained from the well-known Mackey-Glass equation:

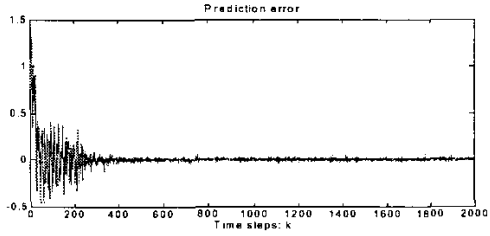
$$\dot{x}(t) = -bx(t) + \frac{ax(t-\Delta)}{1 - x(t-\Delta)^{10}} \tag{19}$$

with parameters $a = 0.2$, $b = 0.1$, $\Delta = 30$, initial conditions $x(t) = 0.9$, and sampling rate $\tau = 6$.

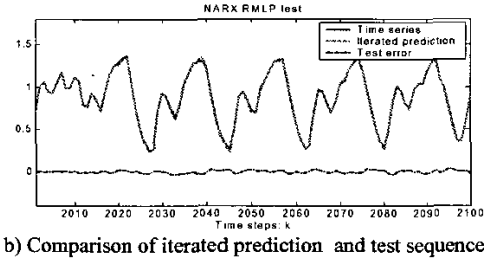
After sequential adaptation on 2000 consecutive samples (presented only once), networks were iterated for next N=100 samples.Table 1, compares the means and variances of *NRMSE* of iterated prediction obtained for NARX_RMLP (6 recurrent inputs, 10 hidden and one output unit), trained using DDF, UKF and EKF for 30 independent runs (different inital values of the network parameters).

Table 1. Normalized mean squared test error of NARX_RMLP

	DDF	UKF	EKF
mean(NRMSE)	0.1238	0.2602	0.3209
var(NRMSE)	4.1e-3	5.58e-2	9.91e-2



a) Prediction error during sequential training



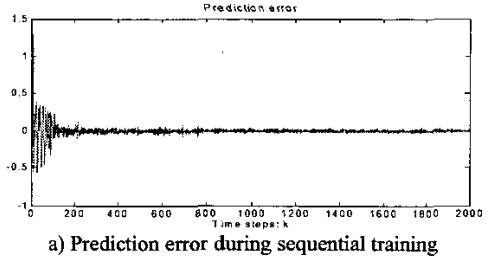
b) Comparison of iterated prediction and test sequence

Fig. 2 . NARX_RMLP training using DDF (NRMSE=5.98e-2)

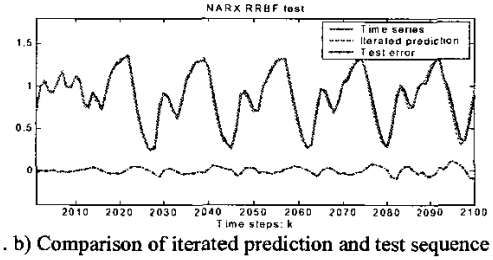
The *NRMSE* means and variances of iterated prediction obtained for NARX_RRBF with 8 hidden units, are given in Table 2.

Table 2. Normalized mean squared test error of NARX_RRBF

	DDF	UKF	EKF
mean(NRMSE)	0.1587	0.209	0.206
var(NRMSE)	3.83e-4	6.05e-3	7.49e-3



a) Prediction error during sequential training



. b) Comparison of iterated prediction and test sequence

Fig. 3. NARX_RRBF training using DDF (NRMSE=1.482e-1)

From Tables 1 and 2, we can see that DDF and UKF produced networks with better generalization capabilities then networks trained by EKF. Lower variances of NRMSE show that DDF and UKF were also less sensitive to initial values of parameters. Since EKF is based on linear approximation of dynamic equation (5a), and DDF uses nonlinear (second order) approximation of (5a) these results were expected.

5. CONCLUSIONS

In this paper we have discussed the application of tree filters: EKF, UKF and DDF to nonlinear parameter and state estimation of a NARX recurrent neural networks. DDF and UKF produced networks with lower generalization error, and are less sensitive to initial parameter values than EKF. Another significant advantage of these filters over EKF is that they do not demand the calculation of the neural network Jacobian, therefore they could be applied in training networks with non-differentiable neuron activation functions.

6. REFERENCES

1. de Freitas, J. F. G., Niranjan, M. and Gee, A.H.: "Hierarchical Bayesian-Kalman models for regularization and ARD in sequential learning," Technical Report CUED/F-INFENG/TR 307, Cambridge University., 1997.
2. Julier, S. J., & Uhlmann, J. K., A new extension of the Kalman filter to nonlinear systems. Preoceedings of AeroSense: The 11th international symposium on aerospace/defence sensing, simulation and controls, Orlando, FL, 1997.
3. Julier, S. J., & Uhlmann, J. K., The Scaled Unscented Transformation , Proceedings of the IEEE American Control Conference, 8-10 May, 2002
4. Lin, T., Horne, B. G., Tino, P., Lee Giles, C., Learning long-term dependencies in NARX recurrent neural networks, IEEE Transactions on Neural Networks, vol. 7. no 6, 1996
5. Nørgaard, M., Poulsen, N. K., & Ravn, O., Advances in derivative free state estimation for nonlinear systems, Technical Report, IMM-REP-1998-15, Department of Mathematical Modelling, DTU, revised April 2000.
6. Todorović, B., Stanković, M, Todorović-Zarkula, S.: Structurally adaptive RBF network in non-stationary time series prediction, In *Proc. IEEE AS-SPCC*, Lake Louise, Alberta, Canada, Oct. 1-4 (2000) pp. 224-229
7. Todorović, B., Stanković, M., Moraga, C.: "Extended Kalman Filter trained Recurrent Radial Basis Function Network in Nonlinear System Identification," *Proc. of ICANN 2002*, Spain, LNCS 2415, pp 819-824, Springer, August 2002
8. van der Merwe, R. & Wan, E. AS., Efficient Derivative-Free Kalman Filters for Online Learning, Proceedings of ESSAN, Bruges, Belgium, April 2001.
9. Williams, R.J.: "Some observations on the use of the extended Kalman filter as a recurrent network learning algorithm," Technical Report NU_CCS_92-1. Boston: Northeastern University, College of Computer Sci., 1992