Brief Papers

An Assessment of Qualitative Performance of Machine Learning Architectures: Modular Feedback Networks

Mo Chen, Temujin Gautama, and Danilo P. Mandic

***Abstract*—A framework for the assessment of *qualitative* performance of machine learning architectures is proposed. For generality, the analysis is provided for the modular nonlinear pipelined recurrent neural network (PRNN) architecture. This is supported by a sensitivity analysis, which is achieved based upon the prediction performance with respect to changes in the nature of the processed signal and by utilizing the recently introduced delay vector variance (DVV) method for phase space signal characterization. Comprehensive simulations combining the quantitative and qualitative analysis on both linear and nonlinear signals suggest that better quantitative prediction performance may need to be traded in order to preserve the nature of the processed signal, especially where the signal nature is of primary importance (biomedical applications).**

***Index Terms*—Delay vector variance, nonlinearity, pipelined recurrent neural networks (PRNNs), qualitative performance, sensitivity.**

I. Introduction

Most real-world signals contain both linear and nonlinear, as well as deterministic and stochastic components. It is, therefore, essential to characterize the signal nature before the actual processing approach is applied. In addition, a change in the linear, nonlinear, deterministic, or stochastic nature of a signal can convey important information[1] about the underlying signal generation mechanism. In such applications, it is important not only to obtain a good quantitative performance for the signal under study, but also that the signal nature is retained.

To establish a general framework for the quantitative performance analysis of machine learning algorithms and architectures, for generality, we will focus on a modular network, with feedback and nonlinear processing elements. One such architecture is the pipelined recurrent neural network (PRNN), which consists of a number of small-scale recurrent neural networks (RNNs), and maintains a relatively low computational complexity considering the entire number of neurons in its architecture [2]. In a previous study, we have analytically described the core features of the PRNN for the prediction application [3]; based on those quantitative results, we will use the PRNN as a convenient computational model for the analysis of the qualitative performance.

This way, we will combine our recent results on signal modality characterization [4], [5] and machine learning [6] to provide insights into the changes of the nature of the processed signals during online learning.

Manuscript received November 29, 2006; revised May 6, 2007; accepted May 12, 2007.

M. Chen and D. P. Mandic are with the Department of Electrical and Electronic Engineering, Communication and Signal Processing, Imperial College London, London SW7 2BT, U.K. (e-mail: mo.chen02@imperial.ac.uk; d.mandic@imperial.ac.uk).

T. Gautama is with the Philips Leuven, Leuven B-3001, Belgium (e-mail: temujin.gautama@philips.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2007.902728

[1]In the electrocardiogram (ECG) and heart rate variability (HRV) analysis, where the change in the signal nature from the linear stochastic to nonlinear deterministic, provides an indication of health hazard [1].

II. Background

A. Nature of a Signal and Surrogate Data

By the signal "nature," we adhere to a number of signal properties described in [4]: linear, nonlinear, deterministic, and stochastic signal behavior. A linear signal is generated by a linear time-invariant system, driven by white Gaussian noise, measured by a static, monotonic, and possibly nonlinear observation function; otherwise, the signal is considered to be nonlinear. A signal is considered deterministic if it can be precisely described by a set of equations; otherwise, it is considered as stochastic.

Surrogate time series, or "surrogate" for short, is a nonparametric randomized linear version of the original data which preserves the linear properties of the original data. In our experiments, we choose iterative amplitude adjust Fourier transform (iAAFT) method to generate surrogates, as it preserves the amplitude distribution of the original data and yields reliable results [7].

B. Quality Assessment Tool—Delay Vector Variance Method

Several methods for detecting nonlinear nature of a signal have been proposed over the past few years [8], [9]. For our purpose, it is desirable to have a method which is straightforward to visualize, and which facilitates the analysis of predictability, which is a core notion in online learning. One such method is our recently proposed delay vector variance (DVV) method [4], based upon examining the predictability of a signal in the phase space, and examining simultaneously the determinism and nonlinearity within a signal. This method can be summarized as follows. For an optimal[2] embedding dimension m, we have the following.

- Generate delay vectors (DVs): $\mathbf{x}(k) = [x_{k-m}, \ldots, x_{k-1}]^T$ and the corresponding target x_k.
- The mean μ_d and standard deviation σ_d are computed over all pairwise Euclidean distances between delay vectors (DVs), $\|\mathbf{x}(i) - \mathbf{x}(j)\|$ $(i \neq j)$.
- The sets $\Omega_k(r_d)$ are generated such that $\Omega_k(r_d) = \{\mathbf{x}(i) \,|\, \|\mathbf{x}(k) - \mathbf{x}(i)\| \leq r_d\}$, i.e., sets which consist of all DVs that lie closer to $\mathbf{x}(k)$ than a certain distance r_d, taken from the interval $[\max\{0, \mu_d - n_d\sigma_d\}; \mu_d + n_d\sigma_d]$, where n_d is a parameter controlling the span over which to perform the DVV analysis.
- For every set $\Omega_k(r_d)$, the variance of the corresponding targets $\sigma_k^2(r_d)$ is computed. The average over all sets $\Omega_k(r_d)$, normalized by the variance of the time series σ_x^2 yields the "target variance" $\sigma^{*2}(r_d) = (1/K)\sum_{k=1}^{K} \sigma_k^2(r_d)/\sigma_x^2$, where K is the total number of the sets $\Omega_k(r_d)$.

To illustrate the meaning of "signal nature," consider a linear benchmark signal [AR(4)] [6]

$$x(k) = 1.79x(k-1) - 1.85x(k-2) + 1.27x(k-3) - 0.41x(k-4) + n(k) \quad (1)$$

[2]We adopt Cao's method [10] for choosing the optimal embedding dimension in all of our simulations, which yields four for the linear benchmark signal (1) and two for the nonlinear benchmark signal (2).

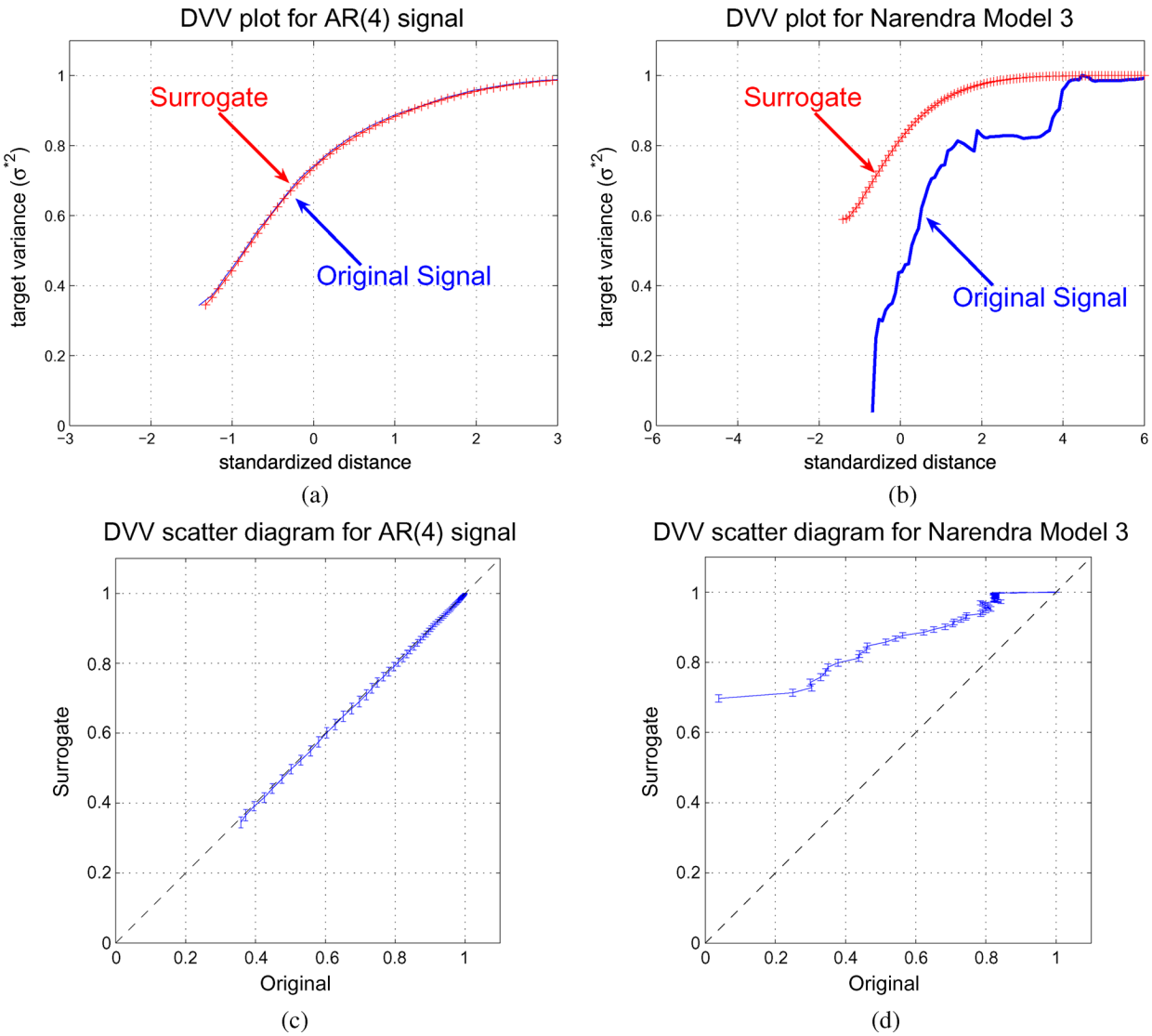

Fig. 1. Nonlinear and deterministic nature of signals. Diagrams (a) and (b) are DVV plots for the AR(4) signal and Narendra model 3, where the light line with crosses denotes the DVV plot for the average of 25 iAAFT-based surrogates while the dark solid line denotes that for the original signal. Diagrams (c) and (d) denote the DVV scatter diagrams for those two signals, where error bars denote the standard deviation of the target variances of surrogates.

and a nonlinear benchmark signal (a Narendra model 3 realization), given by [6]

$$z(k) = \frac{z(k-1)}{1+z^2(k-1)} + x^3(k) \tag{2}$$

where $x\{k\}$ denotes the AR(4) signal (1) and $\{n(k)\}$ is white Gaussian noise $n(k) \in \mathcal{N}(0,1)$.

In the following step, the linear or nonlinear nature of the time series is examined by performing DVV analyses on both the original and 25 of surrogate time series. The DVV plot (target variance, $\sigma^{*2}(r_d)$) is a function of the standardized distance $(r_d - \mu_d)/\sigma_d$. At the extreme right, the DVV plots smoothly converge to unity, since for maximum spans, all DVs belong to the same set, and the variance of the targets is equal to the variance of the time series, illustrated in Fig. 1(a) and (d). The minimal target variance, e.g., the lowest point of the curve, is a measure for the amount of noise which is present in the time series. DVV scatter diagram is constructed in the way where the horizontal axis corresponds to the DVV plot of the original time series, and the vertical to that of the surrogate time series. A linear signal will have similar DVV plot as its surrogate, resulting in the DVV scatter diagram coinciding with the bisector line as illustrated in Fig. 1(c), whereas a nonlinear signal will result in a deviation of DVV scatter diagram from the bisector line as illustrated in Fig. 1(d). This provides a very convenient tool for the qualitative analysis in machine learning since the deviation from bisector line in the DVV scatter diagram can be used to indicate the changes in the signal nature before and after processing.

III. PRNN Architecture

The PRNN is a modular neural network that consists of M nested recurrent neural networks (RNNs) [11] as its modules, with each module consisting of N neurons. All the modules operate using the same weight matrix $\mathbf{W}$, as shown in Fig. 2. Besides its modularity, it is not immediately obvious that the PRNN performs *nesting* [12] of its constituting modules, and at the same time *data-reusing* [13].

In our analysis, the PRNN is utilized to perform one-step-forward prediction on two benchmark signals. In the experiments, the real-time recurrent learning (RTRL) algorithm [2] was used to train RNNs within the PRNN, and the activation function of a neuron was chosen to be the

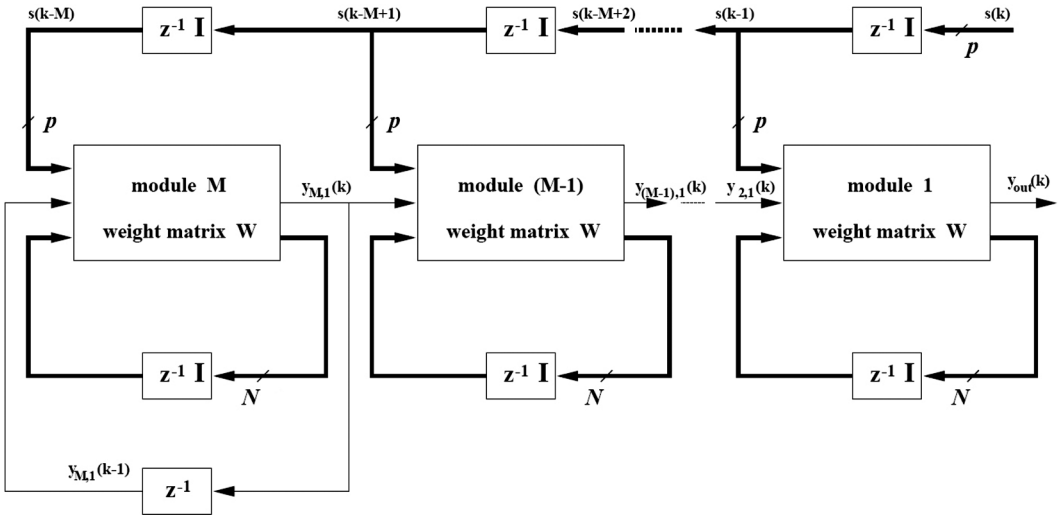

Fig. 2. PRNN.

logistic function $\Phi(v) = 1/(1+e^{-\beta v})$, where $\beta = 1$ and the number[3] of neurons $N = 2$. The synaptic weights of the PRNN were generated from an uniform distribution and rescaled between 0 and 0.01, whereas the bias was set to unity. The forgetting factor (λ), number of external inputs (p), learning rate (η) as well as the number of modules (M) were chosen depending on the individual case and will be addressed later.

To evaluate the prediction performance of PRNN in the sensitivity analysis, or, the *quantitative* performance, we adopt the standard prediction gain [3], e.g., $R_p = 10\log_{10}(\sigma_s^2/\sigma_e^2)[\mathrm{dB}]$, which is a logarithmic ratio between the signal variance σ_s^2 and prediction error variance σ_e^2.

To assess the *qualitative* performance, that is, a possible change in the signal nature introduced by the PRNN, besides comparing the similarity of the DVV scatter diagram of the one-step-forward prediction with that of the original signal, we introduce a metric to measure the degree of such similarity, e.g., $\varepsilon = \sqrt{\langle(\sigma^{*2}_{\mathrm{original}}(r_\mathrm{d}) - \sigma^{*2}_{\mathrm{prediction}}(r_\mathrm{d}))^2\rangle_{\mathrm{valid}\ r_\mathrm{d}}}$, where $\sigma^{*2}_{\mathrm{original}}(r_\mathrm{d})$ denotes the target variance at span r_d for the original signal whereas $\sigma^{*2}_{\mathrm{prediction}}(r_\mathrm{d})$ denotes that for the prediction. For robustness, all the results were obtained by averaging 100 independent trials. Thus, if the PRNN yields high-prediction gain (R_p), the quantitative performance is judged to be "good." As for the qualitative performance, the smaller ε, namely, the more similar the DVV scatter diagram for the filtered signal after being processed by the PRNN to that for the original signal is, the better the qualitative performance becomes.

IV. Analysis of the Performance of the PRNN

A. Quantitative Performance and Sensitivity Analysis

There are four parameters[4] which we need to choose within the PRNN. For convenience, we will vary M and η and fix λ and p at a time. Experiments were repeated four times, each time with a different value of λ and p for two benchmark signals.

Fig. 3 illustrates the overall performance of PRNN on a linear benchmark signal (1) in terms of prediction gain (R_p). Fig. 3(a) and (c) represents the case where p was set to two whereas Fig. 3(b) and (d) represents p set to eight. Fig. 3(a) and (b) illustrates the cases where λ was set to 0.5, and Fig. 3(c) and (d) illustrates the cases when λ set to 0.99. The learning rate (η) was varied from 0.001 to 5.0 at different step size whereas the number of modules (M) varied from 2 to 20. From Fig. 3, the PRNN shows tendency to reach its optimal prediction performance and then to start perform poorly on predicting the AR(4) signal with the increase of η. When the forgetting factor λ was increased, the PRNN obtained its best performance with a decreasing η. The explanation for this phenomenon lies in the weight update equation[5] for the PRNN [14]. From this equation, the whole update for $\mathbf{W}$ is inevitably affected by the increase of λ or η, which leads to the fast converging of the weights. However, when such an increase exceeds certain threshold value, the weights diverge and the learning algorithm will start to become unstable [illustrated in Fig. 3(d)] when $\eta > 2$.

Besides, when λ increases, the total number of modules (M) begins to play a more prominent role in the prediction performance of the PRNN. The reason for is that with the increase of λ, the value of the coefficient λ^{i-1} $(i = 1, \ldots, M)$ in weight update equation becomes significant, which will also accelerate the weight convergence. However, as both λ and M increase, it is not recommended to increase η at the same time.

Furthermore, for the same setting of λ, when p grows, the PRNN obtained its best performance at a rather small learning rate, which can be understood in the following context: As p grows, adjacent modules within PRNN will share more and more the same information and the PRNN will obtain more and more information about the signal. Therefore, the weights converge quicker and the PRNN itself obtains its best performance faster. However, large p also brings a problem of so-called "over-fitting," which leads to one side effect—although the weights converge faster, the redundant weights in modeling AR(4) signal make the PRNN unstable and its performance is inconsistent.

Table I illustrates the maximum prediction gain and the parameter settings. From Table I, when p was fixed, with the increase of forgetting factor (λ), the PRNN achieved its best performance on predicting

[3]Since there is limited improvement in the prediction gain (R_p) with the increase of N and the computational complexity was exponentially increased, therefore, we chose to set N to two.

[4]We can also consider β in the activation function of RNN modules as another parameter. However, it can be shown that there exists a strict relationship between β and learning rate η [6]. Therefore, we will restrict ourselves to varying η only, as this helps reduce computational complexity and will not bring bias into our sensitivity analysis.

[5]At time instant k, the incremental update for the lth weight of neuron n is [2]: $\Delta w_{n,l}(k) = -2\eta\sum_{i=1}^{M}\lambda^{i-1}e_i(k)(\partial e_i(k)/\partial w_{n,l}(k))$.

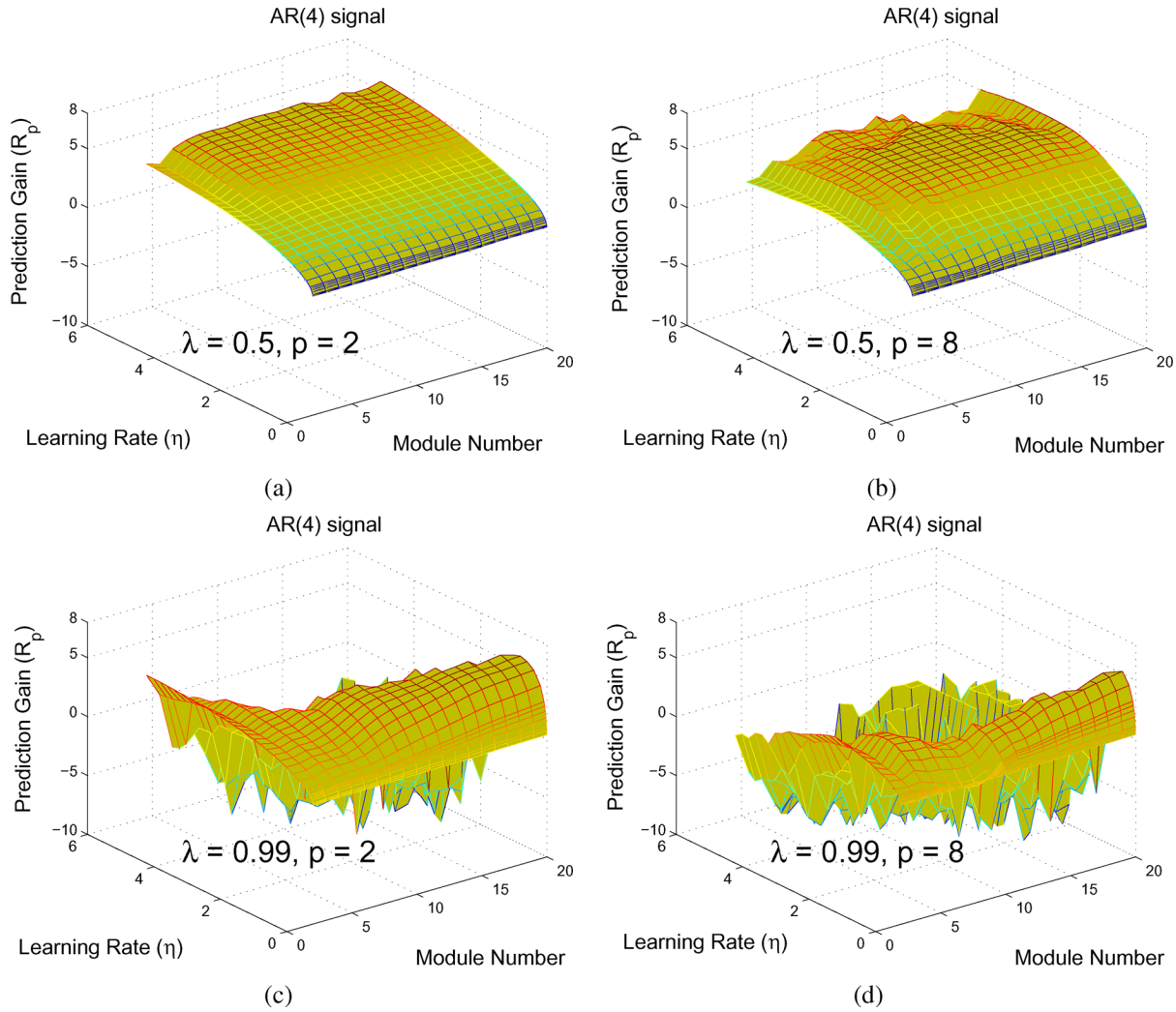

Fig. 3. Prediction performance of PRNN on a linear benchmark AR(4) signal.

TABLE I
PARAMETER SETTINGS OF THE PRNN AT THE MAXIMAL PREDICTION GAIN FOR AR(4) AND NARENDRA MODEL 3 SIGNAL

		AR(4)			Narendra Model 3		
p	λ	M	η	R_p^{max} (dB)	M	η	R_p^{max} (dB)
2	0.50	14	5.0	5.83297	20	3.6	1.31530
2	0.99	19	1.0	6.08877	20	0.4	1.34499
8	0.50	14	3.4	5.71395	20	2.0	1.45836
8	0.99	20	0.4	5.20809	20	0.2	1.00176

the AR(4) signal ($R_p^{\max}$) with a larger number of modules at smaller η. When λ was fixed, with the increase of p, PRNN achieved its best performance at smaller η. These findings are consistent with the previous analysis and discussion.

We next conducted a similar set of experiments on a nonlinear benchmark signal (2). From Table I, when p was fixed, with the increase of λ, the PRNN achieved its best performance at smaller η. When λ was fixed, with the increase of p, PRNN achieved its best performance at smaller η. At all the settings, PRNN always had its highest R_p for a larger number of modules.

B. Assessment of the Qualitative Performance of the PRNN

Next, in order to ascertain whether high quantitative performance yields good qualitative performance, we examined the possible change in the nature of a signal processed by the PRNN. The experimental setting was the same as the one used previously.

The left two diagrams in Fig. 4 illustrate the DVV scatter diagram for the PRNN consisting of five modules, evaluated on the prediction on the Narendra model 3 signal for different forgetting factors (λ). From Fig. 4, R_p decreased with the increase in λ whereas the qualitative performance improved as indicated by the decrease in ε (the DVV scatter diagrams of the original and predicted signal being closer). This can be explained in the following context: with fewer modules, the PRNN cannot capture the full dynamics of the Narendra model 3, but as the forgetting factor λ increases, the remote modules start to contribute more significantly to the weight update. Thus, the PRNN will learn more about the dynamics of the signal being predicted, which contributes to the improvement in preserving the signal nature. As for the

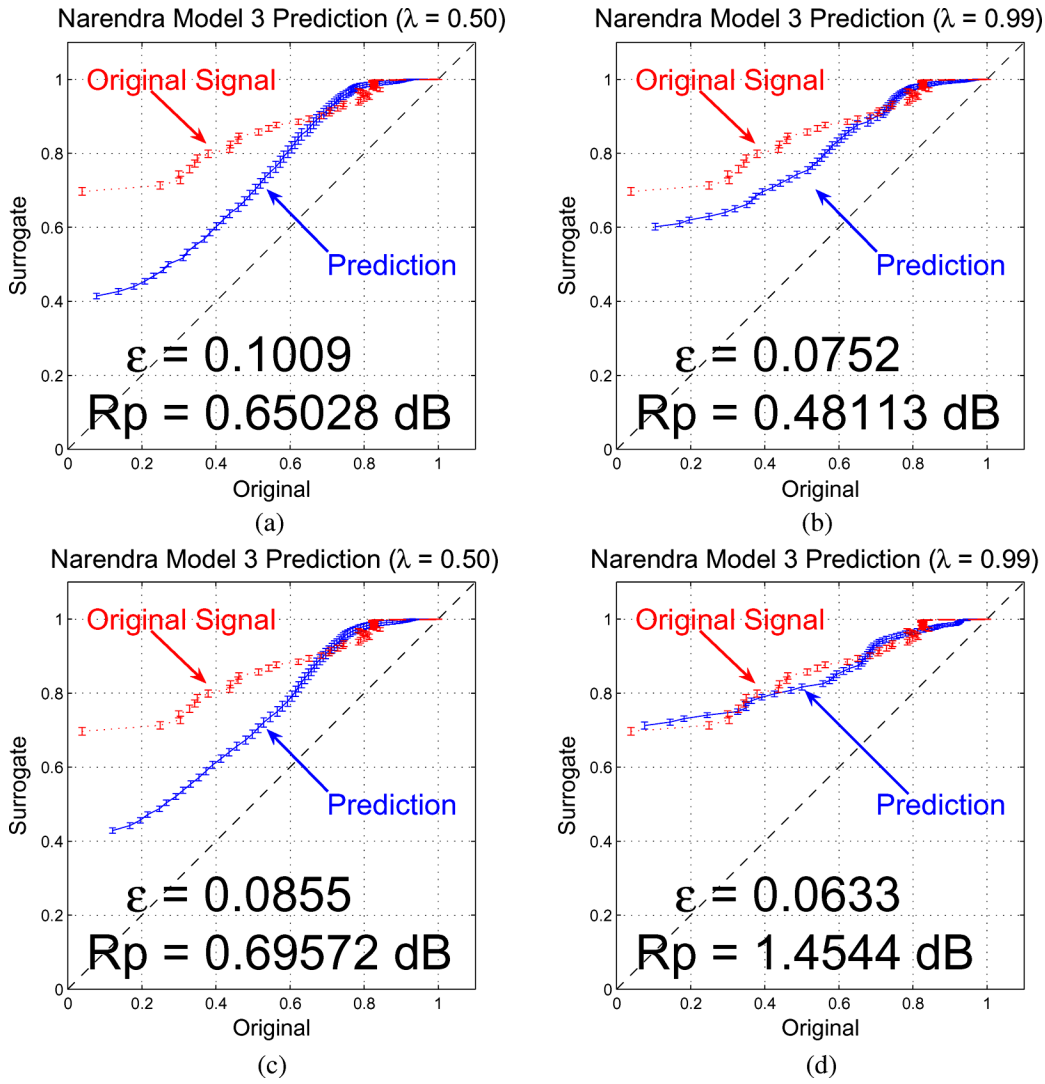

Fig. 4. *Qualitative* and *quantitative* comparison of the PRNN with five modules for the one-step-ahead prediction of the nonlinear benchmark signal (Narendra model 3) with different λ and M. (a) $M = 5$ and $\lambda = 0.5$. (b) $M = 5$ and $\lambda = 0.99$. (c) $M = 10$ and $\lambda = 0.5$. (d) $M = 10$ and $\lambda = 0.99$.

TABLE II
PREDICTION GAIN (R_p) OF INDIVIDUAL MODULES FOR THE PRNN WITH FIVE MODULES ON PREDICTING THE NARENDRA MODEL 3 SIGNAL, λ VARIES FROM 0.5 TO 0.99. THE 1ST MODULE IS THE OUTPUT OF THE PRNN

λ	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	(Module)
0.5	0.6503 dB	2.946 dB	4.749 dB	4.266 dB	3.227 dB	
0.99	0.4811 dB	2.688 dB	4.462 dB	5.561 dB	5.085 dB	

decrease in the prediction gain, this is mainly due to the fact that since using five modules is not enough to capture the nonlinear nature of the signal, emphasizing the contribution of remote modules in fact reduces the portion of the contribution of the first module, which is the main drive for the weight update. That is why for the Narendra model 3, the PRNN with five modules shows an increasingly better nature-preserving capability while it has worse prediction gain with the increase in λ, as illustrated by a decrease in ε. This also demonstrates that a high quantitative performance is not necessarily followed by a high qualitative performance. As the number of modules M increased from five to ten, the PRNN obtained much more information about the signal in hand, which not only caused the prediction gain to increase, but also the nature of the processed signal was better preserved, as illustrated in Fig. 4(c) and (d), where the DVV scatter diagrams become very close.

In the next experiment, we investigated the prediction performance of individual modules. Table II illustrates the prediction performance for individual modules of a PRNN with five modules on predicting Narendra model 3 signal. Observe that when $\lambda = 0.5$, the third module yields the highest R_p; when $\lambda = 0.99$, it was the fourth module. In our previous findings [13], the most remote module, in this case, the fifth module, will exhibit most pronounced "data-reusing" effect, and is supposed to have the highest R_p. In fact, it is the competition of the "forgetting" and "data-reusing" effect that determines the prediction performance of the individual module.

We will now finally analyze the qualitative performance for PRNN on prediction of the linear benchmark signal (1). From Fig. 5, the PRNN was able to preserve the nature of the processed signal, as illustrated by the fact that all the DVV scatter diagrams coincided with the bisector

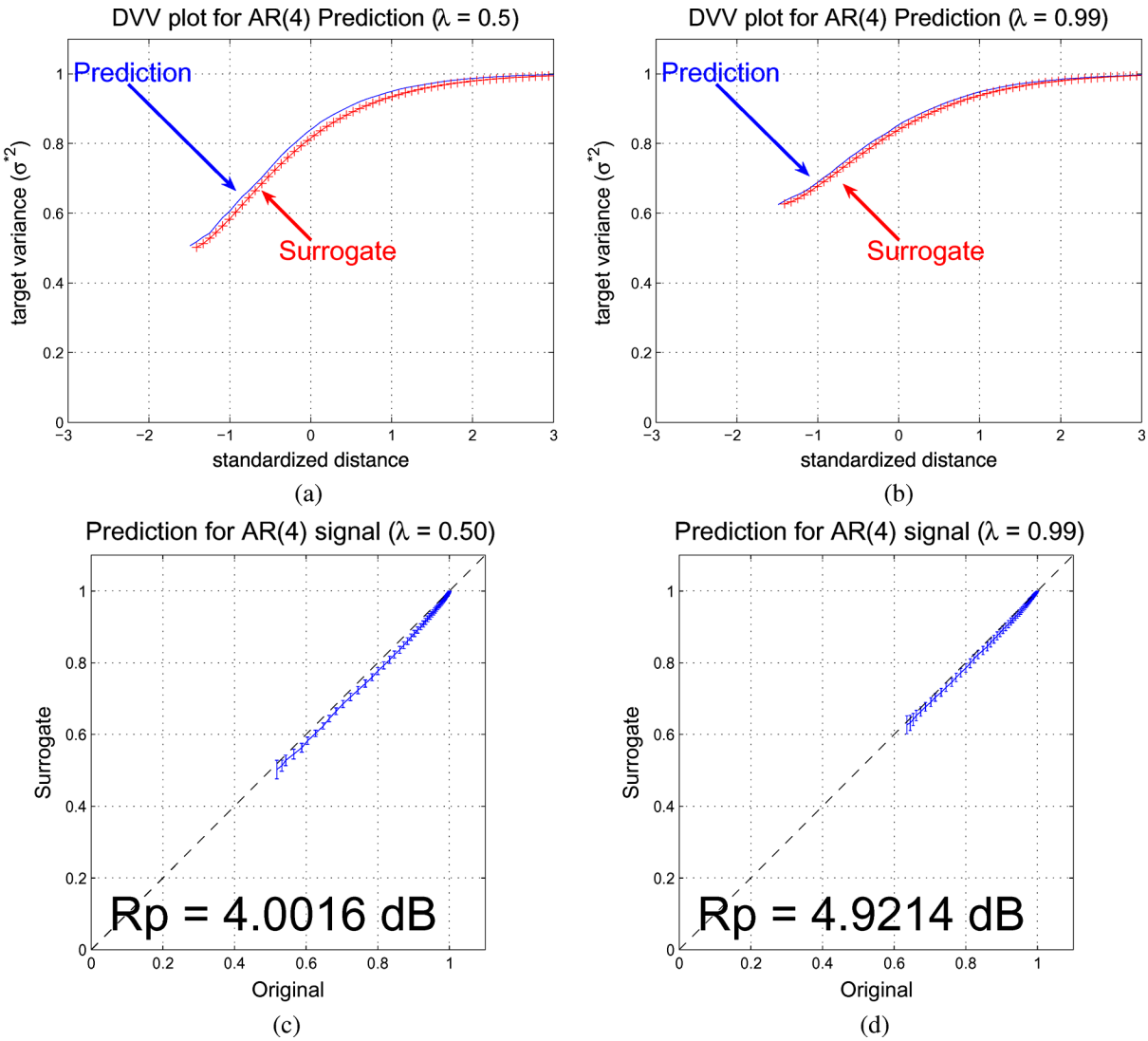

Fig. 5. *Qualitative* and *quantitative* comparison of the PRNN for the prediction of the linear benchmark AR(4) signal. λ varies from 0.5 to 0.99.

line, which indicates the preservation of the linear nature of the original signal. However, the noise level increases with the increase in the forgetting factor, illustrated by the leftmost point (minimal target variance, σ^{*2}) in the DVV plots starting further right. This is again a demonstration that a large forgetting factor makes the weights jitter, which, in turn, introduces stochastic noise into the predicted signal.

V. Conclusion

In this letter, we have provided new insight into the performance of online machine learning architectures. For generality, this is achieved for the PRNN, and the analysis is conducted based upon examining one-step-forward prediction on both linear and nonlinear benchmark signals and for a range of parameter settings. The qualitative performance assessment is achieved by examining whether there is a change in the nature of the processed signal, based upon the recently proposed "delay vector variance" (DVV) method for signal modality characterization. The qualitative performance analysis of simpler architectures follows naturally, by using only one module, or by canceling feedback, or by using linear neurons. It has been shown that there is a need for a tradeoff between the qualitative and quantitative performance index especially when the nature of a signal conveys some important information, for instance, health hazards in medical engineering.

References

[1] C. S. Poon and C. K. Merrill, "Decrease of cardiac chaos in congestive heart failure," *Nature*, vol. 389, pp. 492–495, 1997.

[2] S. Haykin and L. Li, "Nonlinear adaptive prediction of nonstationary signals," *IEEE Trans. Signal Process.*, vol. 43, no. 2, pp. 526–535, Feb. 1995.

[3] D. P. Mandic and J. A. Chambers, "On the choice of parameters of the cost function in nested modular RNNs," *IEEE Trans. Neural Netw.*, vol. 11, no. 2, pp. 315–322, Mar. 2000.

[4] T. Gautama, D. P. Mandic, and M. M. Van Hulle, "The delay vector variance method for detecting determinism and nonlinearity in time series," *Physica D*, vol. 190, no. 3–4, pp. 167–176, 2004.

[5] T. Gautama, D. P. Mandic, and M. M. Van Hulle, "Indications of nonlinear structures in brain electrical activity," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 67, pp. 046204-1–046204-5, 2003.

[6] D. P. Mandic and J. A. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. New York: Wiley, 2001.

[7] T. Schreiber and A. Schmitz, "Improved surrogate data for nonlinearity tests," *Phys. Rev. Lett.*, pp. 635–638, 1996.

[8] M. C. Casdagli and A. S. Weigend, "Exploring the continuum between deterministic and stochastic modeling," in *Time Series Prediction: Forecasting the Future and Understanding the Past*, A. S. Weigend and N. A. Gershenfeld, Eds. Reading, MA: Addison-Wesley, 1994, pp. 347–367.

[9] D. Kaplan, "Exceptional events as evidence for determinism," *Physica D*, vol. 73, no. 1, pp. 38–48, 1994.

[10] L. Cao, "Practical method for determining the minimum embedding dimension of a scalar time series," *Physica D*, vol. 110, pp. 43–50, 1997.
[11] A. Savran, "Multifeedback-layer neural network," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 373–384, Mar. 2007.
[12] Z. Hou, M. Gupta, P. Nikiforuk, M. Tan, and L. Cheng, "A recurrent neural network for hierarchical control of interconnected dynamic systems," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 466–481, Mar. 2007.
[13] M. Chen, T. Gautama, M. M. Van Hulle, and D. P. Mandic, "On nonlinear modular neural filters," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2005, vol. 5, pp. 317–320.
[14] D. P. Mandic, J. Baltersee, and J. A. Chambers, "Non-linear adaptive prediction of speech with a pipelined recurrent neural network and advanced learning algorithms," in *Signal Analysis and Prediction*, A. Prochazka, J. Uhlir, P. W. Rayner, and N. G. Kingsbury, Eds. Boston, MA: Birkhauser, 1998, vol. 5.

Recursive Support Vector Machines for Dimensionality Reduction

Qing Tao, Dejun Chu, and Jue Wang

***Abstract*—The usual dimensionality reduction technique in supervised learning is mainly based on linear discriminant analysis (LDA), but it suffers from singularity or undersampled problems. On the other hand, a regular support vector machine (SVM) separates the data only in terms of one single direction of maximum margin, and the classification accuracy may be not good enough. In this letter, a recursive SVM (RSVM) is presented, in which several orthogonal directions that best separate the data with the maximum margin are obtained. Theoretical analysis shows that a completely orthogonal basis can be derived in feature subspace spanned by the training samples and the margin is decreasing along the recursive components in linearly separable cases. As a result, a new dimensionality reduction technique based on multilevel maximum margin components and then a classifier with high accuracy are achieved. Experiments in synthetic and several real data sets show that RSVM using multilevel maximum margin features can do efficient dimensionality reduction and outperform regular SVM in binary classification problems.**

***Index Terms*—Classification, dimensionality reduction, feature extraction, projection, recursive support vector machines (RSVMs), support vector machines (SVMs).**

I. Introduction

Dimensionality reduction is an important preprocessing step in many applications of data mining, machine learning, and pattern recognition, due to the so-called curse of dimensionality [1], [2]. Now, principal component analysis (PCA, [3]) and linear discriminant analysis (LDA, [4]) are regarded as the most fundamental and powerful tools of dimensionality reduction for extracting effective features from high-dimensional vectors of input data. From the point of view of mathematics, PCA is an orthogonal transformation of the coordinate system in which we describe our data. The new coordinate values by which we represent the data are called principal components. Usually, a small number of principal components is sufficient to account for most of the structure in the data. From the viewpoint of pattern recognition, LDA aims to find the optimal discriminant vectors (and then, an orthogonal transformation) by maximizing the ratio of the between-class distance to the within-class distance, thus achieving the maximum class discrimination. LDA is the benchmark for the linear discrimination between two classes in multidimensional space. One of the most obvious differences between PCA and LDA is that the former does not employ the labels of all samples while the latter does.

Around 1997, several comparative studies between LDA and PCA on the face recognition problems were reported independently by numerous authors [5], [6], in which LDA outperformed PCA significantly. So far, LDA has proven to be a more efficient approach for extracting features for many pattern classification problems as compared to PCA. However, there exists a serious limitation for using LDA to solve high-dimensional recognition with finite samples. Usually, LDA requires the so-called total scatter matrix to be nonsingular. In many applications, especially in face recognition, all scatter matrices in question can be singular since the data points are from a very high-dimensional space and, in general, the sample size does not exceed this dimensionality. This is known as the *singularity* or *undersampled problem* [8] and inevitably gives rise to a problem of unstable numerical computation. In recent years, many approaches have been proposed to deal with such high-dimensional undersampled problems, including null space LDA and orthogonal LDA, and their detailed computational and theoretical analysis can be seen in [9]. Recently, a recursive LDA for calculating the discriminant features was suggested in [10]. This new algorithm incorporates the same fundamental idea behind LDA of seeking the projection that best separates the data corresponding to different classes, while in contrast to regular LDA, the features are obtained recursively and the number of features that may be derived is independent of the number of the classes to be recognized. Extensive experiments of comparing the recursive LDA algorithm with the traditional approaches have been carried out on face recognition problems, in which the resulting improvement of the performances by the new feature extraction scheme is significant. Obviously, how to employ the recursive idea to get a dimensionality reduction approach without undersampled problems is very interesting.

In the last few years, there have been very significant developments in the understanding of support vector machines (SVMs) and statistical learning theory [11]–[13]. In appearance, the geometric interpretation of a linear SVM, known as the maximum margin algorithm, is very clear. In theory, increasing margin has been shown to improve the generalization performance. In [14], an SVM-like framework was established for LDA and it was proved that the general framework of LDA is based on the simplest and most intuitive LDA with zero within-class variance. Further, it can be found that LDA and SVM are closely related. Commonly, they all try to seek the projection that best separates the data in terms of a specific objective function. Along the former direction, the within-class variance is minimized while between-class variance is maximized. Along the latter, the between-class distance is maximized with the large margin while within-class distance is not considered. Since the usual dimensionality reduction technique in supervised learning is mainly based on using a small number of orthogonal

Manuscript received November 11, 2006; revised February 7, 2007; accepted July 2, 2007. This work was supported by the National Basic Research Program 2004CB318103 and the National Science Foundation of China under Grant 60575001.

Q. Tao is with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, P.R. China and also with the New Star Research Institute of Applied Technology, Hefei 230031, P.R. China (e-mail: qing.tao@mail.ia.ac.cn; taoqing@gmail.com).

J. Wang is with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, P.R. China (e-mail: jue.wang@mail.ia.ac.cn).

D. Chu is with the New Star Research Institute of Applied Technology, Hefei, 230031 P.R. China.

Digital Object Identifier 10.1109/TNN.2007.908267