

ON NONLINEAR MODULAR NEURAL FILTERS

Mo Chen, Danilo P. Mandic

Imperial College London
Department of Electrical and Electronic
Engineering,
Exhibition Road, London, SW7 2BT, U.K.
E-mail: {mo.chen, d.mandic}@imperial.ac.uk

Temujin Gautama and Marc M. Van Hulle

Laboratorium voor Neuro- en
Psychofysiologie,
K.U.Leuven, Campus Gasthuisberg,
Herestraat 49, B-3000 Leuven, BELGIUM
E-mail: {temu, marc}@neuro.kuleuven.ac.be

ABSTRACT

An assessment of the performance of the pipelined recurrent neural network (PRNN) is provided from two aspects, a *quantitative* one based on the prediction gain and a *qualitative* one based on examining the changes in the nature of the processed signal. This is achieved by means of the recently introduced ‘Delay Vector Variance’ (DVV) method for phase space signal characterisation. A comprehensive analysis of this approach on both linear and nonlinear benchmark signals suggests that the PRNN not only outperforms a single recurrent neural network (RNN) in terms of the prediction gain but also has better or similar performance in terms of preserving the nature of the processed signal.

1. INTRODUCTION

Pipelined recurrent neural networks¹ [1] (PRNNs) have been recently proposed for adaptive signal processing applications [2] and have been shown to be able to deal with nonlinear as well as non-stationary signals. This is achieved with fairly high computational efficiency as compared to a single network with the same number of neurons. Considerable research has been conducted towards improving the performance and computational efficiency of these networks, mostly concentrating on devising novel learning algorithms for the nonlinear predictor within the PRNN architecture, namely for modular nested recurrent neural networks [3, 4].

1.1. Some Background Notions

By the signal ‘nature’, we refer to the linear², nonlinear, deterministic³ and stochastic signal behaviour. For simplicity, we shall restrict ourselves only to the first two properties. Characterisation of signals based on their nature is still emerging, however, in some modern machine learning and signal processing applications, it is essential to characterise the signal behaviour, since the linear, nonlinear, deterministic or stochastic nature of a signal can convey important information about the underlying signal generation mechanism. For instance, in the electrocardiogram (ECG)

¹PRNN is a modular neural network, and consists of a certain number M of nested recurrent neural networks (RNNs) as its modules, with each module consisting of N neurons. All the modules operate using the same weight matrix \mathbf{W} , as shown in Figure 1.

²A linear signal is the one that is generated by a linear time-invariant system, driven by white Gaussian noise. A signal that cannot be generated this way is considered nonlinear [5].

³A signal is considered deterministic if it can be precisely described by a set of equations. Otherwise, it is stochastic.

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 [6]. Similar phenomena have been reported in the analysis of air pollutants [7] and brain electrical activity [8].

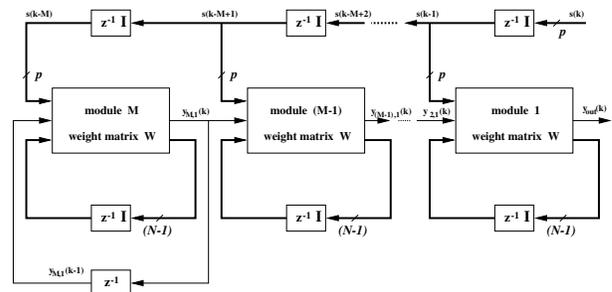


Fig. 1. Pipelined recurrent neural networks.

1.2. The PRNN Architecture

Besides its modularity, it is not immediately obvious that the PRNN performs nesting of its constituting modules, and at the same time data-reusing. Nesting can be functionally described by the PRNN output function⁴

$$y_{1,1}(k) = \Phi(s(k-1), \Phi(s(k-2), \dots, \Phi(s(k-M), \dots, y_{M,1}(k-M)))) \dots) \quad (1)$$

where Φ is the nonlinear activation function which is identical for all the modules, $y_{M,1}(k)$ is the output of the M th module of the PRNN and \mathbf{s} is the external input vector to the neurons. This nesting property (1) gives the PRNN its enhanced computing power as compared to the conventional RNN [4] and its increased ability to model the nonlinearity within the signals. The cost function for the PRNN is a weighted sum of squared errors at the output of every module of the PRNN. Hence, at time instant k , the incremental update for the l th weight of neuron n is as follows [1]:

$$\Delta w_{n,l}(k) = -2\eta \sum_{i=1}^M \lambda^{i-1} e_i(k) \frac{\partial e_i(k)}{\partial w_{n,l}(k)} \quad (2)$$

where η is the learning rate. To understand the operation of the PRNN, it is important to notice that, at time instant k :

⁴For the sake of simplicity, the functional dependence of the weight matrix \mathbf{W} to the nested nonlinearity was omitted.

- The output of the first module, $y_1(k)$, is in fact the prediction of $s(k)$. The prediction error $e_1(k) = s(k) - y_1(k)$ only partially contributes to the weight update (2).
- The output of the i th module, $y_i(k)$ is in fact the prediction of $s(k - i + 1)$, thus $e_i(k) = s(k - i + 1) - y_i(k)$ again partially contributes to the weight update (2).

In this way, the time sample $s(k)$ influences the weight update (2) exactly M times, inducing an effect similar to that of *a posteriori*, that is, data-reusing algorithms [9, 10] when applied to standard filters. Furthermore, module M will benefit most from this data-reusing behaviour, as it receives the highest number (M) of data-reusing ‘iterations’ before predicting $s(k)$.

Previously, we have investigated the qualitative performance of hybrid filtering architectures [11] as well as data-reusing (DR) algorithms [12], where we found that in hybrid filtering architectures a high prediction gain could bring the side effect of a change in the nature of the processed signal and that data-reusing algorithms not only exhibit an advantage in the performance but also better preserve the nature of the processed signal than the standard algorithms. Notice that data-reusing within the PRNN is embodied within its architecture, rather than through iterations on the current weight update. At the same time, the PRNN, as originally introduced in [1], together with a tapped-delay-line (TDL) filter when applied, represents a hybrid filter, which consists of a nonlinear adaptive filter followed by a linear filter. Therefore, there is a need to address this issue in this modular nested hybrid architecture. The modular hybrid PRNN was introduced with the idea that the nonlinear part of the network would perform nonlinear adaptive prediction whereas the linear part would perform linear prediction. We therefore employ the novel results in signal nature characterisation to investigate the way this hybrid modular nonlinear filter operates. *Qualitative* analysis of the change in the nature of the processed signal, both along the modules and for PRNNs of different size is performed. This will shed new light onto the performance of hybrid modular filters with implicit data-reusing capacities.

2. QUALITY ASSESSMENT TOOL – DELAY VECTOR VARIANCE METHOD

Several methods for detecting the nature of a signal have been proposed over the past few years, which include the *Deterministic versus Stochastic* (DVS) plots [13], the Correlation Exponent and δ - ε method [14]. The recently introduced DVV method [5] is shown to be well suited for applications in the signal processing context, since it simultaneously examines both the nonlinear and deterministic nature of a signal. This method investigates the local predictability of a signal in the phase space and can be summarised as follows:

For a given embedding dimension m (which, for convenience, was set to 2 in all of our simulations):

- Generate delay vectors (DVs): $\mathbf{x}(k) = [x_{k-m}, \dots, x_{k-1}]^T$ and the corresponding target x_k ,
- The mean μ_d and standard deviation σ_d are computed over all pairwise Euclidean distances between DVs, $\|\mathbf{x}(i) - \mathbf{x}(j)\|$ ($i \neq j$),

- The sets $\Omega_k(r_d)$ are generated such that $\Omega_k(r_d) = \{\mathbf{x}(i) \mid \|\mathbf{x}(k) - \mathbf{x}(i)\| \leq r_d\}$, *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 $\Omega_k(r_d)$, normalised by the variance of the time series, σ_x^2 , yields the ‘target variance’, $\sigma^{*2}(r_d)$:

$$\sigma^{*2}(r_d) = \frac{1}{N} \sum_{k=1}^N \frac{\sigma_k^2(r_d)}{\sigma_x^2} \quad (3)$$

As r_d increases, the target variance smoothly converges to unity. This is because all DVs start to belong to the same universal set, and the variance of targets is equal to the variance of the time series.

To illustrate the meaning of ‘signal nature’, consider a benchmark linear signal (AR(4)), given by [15]

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

and a benchmark nonlinear signal (x-component of Henon map), given by [16]

$$\begin{aligned} x_n &= 1 - a x_{n-\tau}^2 + y_{n-\tau} \\ y_n &= b x_{n-\tau} \end{aligned} \quad (5)$$

where τ is the time lag which was set to unity, and parameters a and b were set to 1.4 and 0.3, respectively. We will refer to this signal as the Henon map though only the x-component of Henon map is used in the following simulations.

To indicate nonlinearity within a signal, the DVV test is also per-

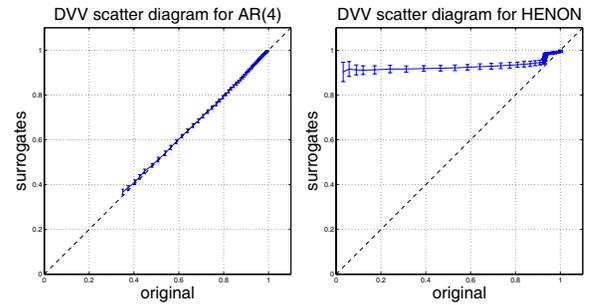


Fig. 2. Nonlinear nature of signals. Left: linear signal (4). Right: nonlinear signal (5). Error bars denote the standard deviation of the target variances of surrogates.

formed on a set of surrogate data⁵ using the same parameter setting as for the original signal. Next, the DVV plot (target variance σ^{*2} versus standardised distance $\frac{r_d - \mu_d}{\sigma_d}$) of the original time series is plotted against the average of that for surrogate data⁶. These plots

⁵Surrogate data are nonparametric linear versions of the original data.

⁶In fact, target variance (σ^{*2}) of the original data is plotted against the mean of the target variance of N surrogate data, for all corresponding distances ($\frac{r_d - \mu_d}{\sigma_d}$).

can be conveniently combined within a *scatter diagram*, where the horizontal axis corresponds to the target variance of the original time series, and the vertical axis to that of the surrogate time series. If the surrogate time series yield target variance similar to that of the original time series, the ‘DVV scatter diagram’ coincides with the bisector line, and the original time series is judged to be linear, as shown in the left diagram of Figure 2 (for the linear signal (4)). If not, then the original time series is judged to be nonlinear, as depicted in the right diagram of Figure 2 (for the nonlinear signal (5)).

3. THE PERFORMANCE OF THE PRNN

In this section, we investigate the *quantitative* and *qualitative* performance of the PRNN, based on prediction of two benchmark signals (4) and (5). To assess the *quantitative* performance, we use the common one-step forward prediction gain R_p , defined as

$$R_p = 10 \log \left(\frac{\hat{\sigma}_s^2}{\hat{\sigma}_e^2} \right) \text{dB} \quad (6)$$

which is a logarithmic ratio between the estimated signal variance $\hat{\sigma}_s^2$ and estimated prediction error variance $\hat{\sigma}_e^2$. As for assessing the *qualitative* performance, that is, the possible change in signal nature, we compare DVV scatter diagrams of the outputs of the PRNN with those of the original signal. This is achieved by averaging 100 independent trials. In the experiments, the real time recurrent learning (RTRL) algorithm, was used to train RNNs within the PRNN, and the activation function of a neuron was chosen to be the logistic function

$$\Phi(v) = \frac{1}{1 + e^{-bv}} \quad (7)$$

with β set to 1.0, forgetting factor (λ) set to 0.99 and the number of neurons (N) set to 5 as well as the tap length of external input (p) set to 5, whereas the number of modules (M) varies with individual case and will be addressed later.

Figure 3 illustrates both the *quantitative* (value of R_p) and *qual-*

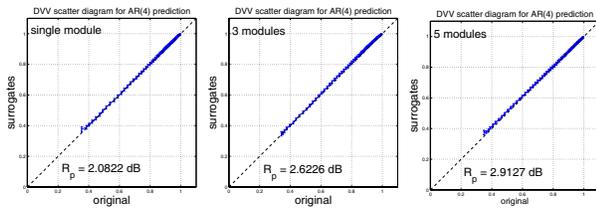


Fig. 3. *Qualitative* and *quantitative* comparison of the PRNN with different number of modules for the one-step ahead prediction of the linear benchmark signal (4). M varies from 1 (left diagram) to 3 (middle diagram) and 5 (right diagram).

itative (DVV scatter diagram) performance of the PRNN applied for prediction of linear benchmark signal (4). The tile diagram in Figure 3 illustrates the DVV scatter diagrams for outputs of PRNNs with different number of modules (1, 3, 5)⁷. From the Figure, in all the cases, the nature of the AR(4) signal were preserved. This is illustrated by the fact that all the DVV scatter diagrams in Figure 3 lie on the bisector line, which was also the case for the original signal (4) (left diagram in Figure 2). In terms of

⁷These cases were chosen to illustrate that the PRNN performs better than a single RNN. There is not much improvement in the prediction gain when the number of modules is further increased.

the prediction gain, PRNN with 5 modules performed best⁸, and as expected, prediction gain R_p increases with the increase in the number of modules increases, which is the evidence that PRNN performed better than a single RNN.

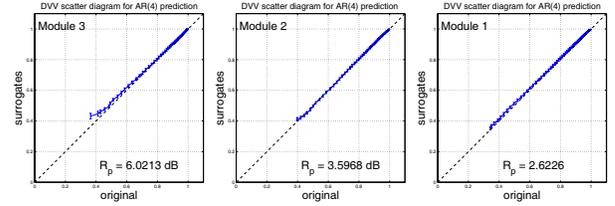


Fig. 4. *Qualitative* and *quantitative* comparison of individual modules in the PRNN of 3 modules for the one-step forward prediction of the linear benchmark signal (4).

Figure 4 illustrates more detailed results for a single PRNN consisting of three modules (module 3, module 2, module 1 from left to right in Figure 1, respectively predicting $s(k-2)$, $s(k-1)$, $s(k)$) on the performance of predicting the linear benchmark signal (5). In Figure 4, the left part was the DVV scatter diagram for the third module of PRNN (leftmost module in Figure 1), the middle one was that for the second module and the right one was that for the first module, namely, the actual output of the PRNN (rightmost module in Figure 1). From the Figure, the signal nature was well preserved by all three modules. However, when compared with the DVV scatter diagram for the original signal (left diagram in Figure 2), there is evidence of some nonlinearity in the left two diagrams, judged by the fact that DVV scatter diagrams for these two modules were not as perfectly on the bisector line as the that for the first module. As far as prediction gain is concerned, module 3 performed best and module 1 the worst. This can be explained in the context of data-reusing property of the PRNN architecture, which has been described in the previous section: due to the fact that all the modules share the same weight matrix, before module i was trying to predict $s(k-i+1)$ at time instant k , the PRNN system already had some information and weights were already optimised to a certain extent for the forthcoming prediction. That is why the distant modules from the output of the PRNN have better prediction gain.

Figure 5 illustrates a similar experiment conducted on the predic-

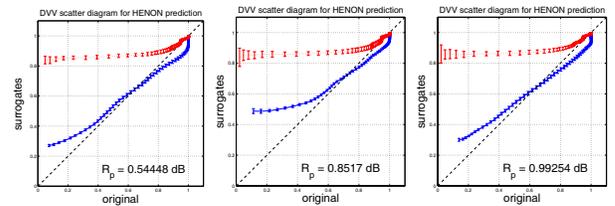


Fig. 5. *Qualitative* and *quantitative* comparison of the PRNN with different number of modules for the one-step forward prediction of the nonlinear benchmark signal (5). M varies from 1 (left diagram) to 3 (middle diagram) and 5 (right diagram). Dotted line above indicates the DVV scatter diagram for the original signal, whereas solid line below indicates that for the filtered one.

⁸The PRNN performance depends also on the value of N and p , which in this case were chosen to be the same for all three situations.

tion of the nonlinear benchmark signal (5) with the same setting as in the previous experiment. From the Figure, in all the cases, the filters were not able to properly capture the nature of the Henon map. This is illustrated by the fact that the DVV scatter diagram for the filtered signal (solid line) did not get close enough to that for the original signal (dotted line). However, we can still see some nonlinearity in the filtered signal. In terms of the prediction gain, PRNN with 5 modules performed best, and as expected, prediction gain R_p increased with the increase in the number of modules, which again proved that PRNN performed overall better than a single RNN.

In the light of our recent work on the quality assessment of hy-

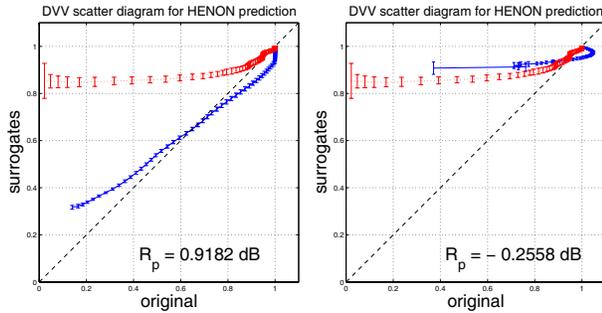


Fig. 6. Qualitative and quantitative comparison of PRNN without the FIR filter (left) and with the subsequent FIR filter (right) for the one-step forward prediction of the nonlinear benchmark signal (5). Dotted line indicates the DVV scatter diagram for the original signal, whereas solid line indicates that for the filtered one.

brid filters [11], we conducted one more experiment on the hybrid PRNN, realised by cascading a linear finite impulse response (FIR) filter after the PRNN. Figure 6 illustrates the results of such an experiment. The standard least mean square (LMS) was used to train the linear filter with tap length set to 8. In the Figure, the qualitative performance of PRNN without the subsequent linear FIR filter is shown in the left diagram while that of the hybrid filter is shown in the right diagram. It is clear that after filtering by the FIR filter, the nature of the Henon map time series was better preserved, as indicated by the fact that two curves, namely two DVV scatter diagrams for the original signal and the output of the PRNN, in the right diagram became closer to each other. However, in terms of the prediction gain, a stand-alone PRNN outperformed the hybrid PRNN filter, which conforms to our previous observation that in some cases a high prediction gain brings the side effect of linearising the signal nature. The reason for the weak prediction performance is that Henon map is very nonlinear, seen from fact that the DVV scatter diagram deviates from the bisector line greatly, which makes it difficult to model. Moreover, we chose this signal to show the principles and did not optimise the PRNN parameters.

4. CONCLUSIONS

In this paper, we have shown that the pipelined recurrent neural network (PRNN) outperforms single recurrent neural network (RNN) on both linear and nonlinear benchmark signals with higher prediction gain and possibly better qualitative performance in terms of signal nature preserving. Due to the data-reusing effect embodied in the PRNN architecture, it is shown for the distant modules from the output of PRNN to have higher prediction gain than the

output module. Besides, as an extension to our previous findings on hybrid filters, we have also conducted an experiment on the PRNN followed by a linear FIR filter, and the result is consistent with our previous findings that a quest for higher prediction gain can sometimes have an effect of linearising the signal nature. This is not preferred in the environments where the nature of a signal conveys some important information, e.g., about health hazards.

5. REFERENCES

- [1] S. Haykin and L. Li, "Nonlinear adaptive prediction of nonstationary signals," *IEEE Transactions on Signal Processing*, vol. 43, no. 2, pp. 526–535, 1995.
- [2] P. Chang and J. Hu, "Optimal nonlinear adaptive prediction and modeling of mpeg video in atm networks using pipelined recurrent neural networks," *IEEE Journal On Selected Areas In Communications*, vol. 15, no. 6, pp. 1087–1100, 1997.
- [3] J. Baltersee and J. A. Chambers, "Non-linear adaptive prediction of speech signals using a pipelined recurrent neural network," *IEEE Transactions on Signal Processing*, vol. 46, no. 8, pp. 2207–2216, 1998.
- [4] D. P. Mandic and J. A. Chambers, "Towards a PRNN based a posteriori non-linear predictor," *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1435–1442, 1999.
- [5] 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.
- [6] C. S. Poon and C. K. Merrill, "Decrease of cardiac chaos in congestive heart failure," *Nature*, vol. 389, pp. 492–495, 1997.
- [7] S. R. Dorling, R. J. Foxall, D. P. Mandic, and G. C. Cawley, "Maximum likelihood cost functions for neural network models of air quality data," *Atmospheric Environment*, vol. 37, pp. 3435–3443, 2003.
- [8] T. Gautama, D. P. Mandic, and M. M. Van Hulle, "Indications of nonlinear structures in brain electrical activity," *Phys. Rev. E*, vol. 67, pp. 046204–1–046204–5, 2003.
- [9] S. Roy and J. J. Shynk, "Analysis of the data-reusing LMS algorithm," in *Proceedings of the 32nd Midwest Symposium on Circuits & Systems*, 1989, vol. 2, pp. 1127–1130.
- [10] D. P. Mandic, S. Goh, and A. I. Hanna, "Data-reusing algorithms for complex valued adaptive filters," in *Complex Valued Neural Networks - Theories and Applications, Series on Innovative Intelligence*, A. Hirose, Ed., vol. 5, pp. 131–157. World Scientific, Singapore, 2003.
- [11] M. Chen and D. P. Mandic, "Quality assessment of hybrid filters," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2004, Montreal*, vol. 5, pp. 785–788, 2004.
- [12] M. Chen, T. Gautama, M. M. Bozic, D. P. Mandic, and M. M. Van Hulle, "Does iterative nonlinear neural adaptive filtering affect the nature of the processed signal?," *accepted by the Seventh Seminar on Neural Network Applications in Electrical Engineering, NEUREL 2004, Belgrade*, 2004.
- [13] 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., pp. 347–367. Reading, MA: Addison-Wesley, 1994.
- [14] D. Kaplan, "Exceptional events as evidence for determinism," *Physica D*, vol. 73, no. 1, pp. 38–48, 1994.
- [15] D. P. Mandic and J. A. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*, John Wiley & Sons, 2001.
- [16] M. Henon, "A two-dimensional mapping with a strange attractor," *Commun. Math. Phys.*, vol. 50, pp. 69–77, 1976.