# A RECURRENT MULTISCALE ARCHITECTURE FOR LONG-TERM MEMORY PREDICTION TASK

*Stefano Squartini*(*), *Amir Hussain*(**) *and Francesco Piazza*(*)

(*) Dipartimento di Elettronica e Automatica, University of Ancona, 60131 Ancona, Italy
sts@ea.unian.it, upf@ea.unian.it
(**) Dept of Computing Science, University of Stirling, Stirling FK9 4LA, U.K.
ahu@cs.stir.ac.uk

## ABSTRACT

In the past few years, researchers have been extensively studying the application of recurrent neural networks (RNNs) to solving tasks where detection of long term dependencies is required. This paper proposes an original architecture termed the *Recurrent Multiscale Network, RMN*, to deal with these kinds of problems. Its most relevant properties are concerned with maintaining conventional RNNs' capability of information storing whilst simultaneously attempting to reduce their typical drawback occurring when they are trained by gradient descent algorithms, namely the *vanishing gradient effect*. This is achieved through RMN which preprocesses the original signal separating information at different temporal scales through an adequate DSP tool, and handling each information level with an autonomous recurrent architecture; the final goal is achieved by a nonlinear reconstruction section. This network has shown a markedly improved generalization performance over conventional RNNs, in its application to time series prediction tasks where long range dependencies are involved.

## 1. INTRODUCTION

Recurrent neural networks (RNNs) have been widely used recently to deal with many dynamical and non-linear problems, such as time series forecasting and identification of general systems. In order to handle the temporal characteristics of such data, any good system solution should let a past input exercise its effect on a successive time instant, i.e. a sort of memorization mechanism must be provided in order to capture contextual information. RNNs are well suited for this as they have long-term memory embedded: the cycles in the graph of a recurrent network (represented by its feedback synapses) allow it to keep information about past inputs for an amount of time that is not fixed a-priori (*information latching* property), but rather depends on its weights and on the input data. In contrast, static networks (with no recurrent connections), even if they include delays (such as Time Delay Neural Networks), have a finite impulse response and cannot store a bit of information for an indefinite time.

Hence, RNNs are the best candidates to store information for an arbitrary duration, but such a capability is offset by the *vanishing gradient* problem. This effect takes place when a RNN is trained by learning algorithms based on computing the gradient of a cost function with respect to the weights of the network [1], such as in the back propagation through time (BPTT) or real time recurrent learning (RTRL). It has been analytically proven [2], [3] that either the system gets information latching being resistant to noise or, alternatively, it is efficiently trainable by gradient descent learning algorithm, but not both. The former case coincides with the occurrence of vanishing gradient effect that leads to poor generalization performances.

Several solutions have been proposed to mitigate this effect: some methods use alternative learning algorithms, or special architectures (additional memory for example) or a combination of these [2]. Also particular methods based on conceiving learning as a refinement process from a-priori knowledge (symbolic analysis) have been developed [2]. However almost all of these methods tend to not preserve the original simplicity and low computational cost of RNNs trained by BPTT, which is most widely used among gradient descent based learning algorithms. As a consequence, a new original architecture is proposed in this paper, in order to improve long-term dependencies detecting capability when common RNNs and BPTT are used together as the core of a hybrid learning system, resulting in a reduction of the vanishing gradient effect. The three-stage solution implemented is based on preprocessing the input signal through a multi-band decomposition technique and processing each obtained sequence separately through different RNNs. In principle, all information quantity associated to a particular temporal scale of the starting input can be learnt by a single recurrent structure more easily than in the case of detecting the same quantity from the entire original signal. Moreover, an appropriate reconstructing operation is provided, in order to process all elaborated sequences together, delivering the final result as an imitating attempt of the original target.

## 2. DISCRETE WAVELET DECOMPOSITION

The chosen multi-band decomposition technique is the discrete wavelet decomposition, which can be seen as an octave band filter bank, with its analysis section (Discrete Wavelet Transform (DWT)) and its synthesis section (inverse DWT (IDWT)) [4]. Looking at the decomposition part, it can be observed that the original signal $\{x_n\}$ is processed through filtering and down-sampling operations resulting in different sequences

$$w_j(k) = w_k^j = \left\langle x_n, \tilde{h}'^{j}_{n-2^j k} \right\rangle, \quad j = 1,\dots,J$$
$$v_J(k) = v_k^J = \left\langle x_n, \tilde{g}'^{J}_{n-2^J k} \right\rangle \tag{1}$$

where $\tilde{g}'^j = \left(\tilde{\mathbf{G}}'\uparrow\right)^{j-1}\tilde{g}'$, $\tilde{h}'^j = \left(\tilde{\mathbf{G}}'\uparrow\right)^{j-1}\tilde{h}'$ and the following notation equalities hold:

$$\left(\uparrow x\right)_{2n} = x_n, \left(\uparrow x\right)_{2n+1} = 0, \qquad \textit{upsampling}$$
$$\left(\downarrow x\right)_n = x_{2n} \qquad \textit{downsampling}$$
$$\mathbf{G}\,(g_n\ \textit{low}-\textit{pass filter})\ \left(\mathbf{G}x\right)_n = \sum_k x_k g_{n-k} \tag{2}$$
$$\mathbf{H}\,(h_n\ \textit{high}-\textit{pass filter})\ \left(\mathbf{H}x\right)_n = \sum_k x_k h_{n-k}$$
$$\tilde{g}_n = g^*_{-n} \qquad \textit{paraconjugate operator}$$

Each of these sequences is relative to a precise part of spectrum of the signal and has different length than the others; in more specific words, it means that their scale and resolution values are divided by 2 at each decomposition level, consequently reducing by the same factor the sequence length and the part of spectrum they represent. This fact is directly related to the coverage of time/frequency plane existing in continuous wavelet transform (CWT), confirming the analytical link between DWT and CWT.

The signal can be reassembled from the coefficients through filtering and up-sampling operation:

$$x(n) = x_n = \sum_{j=1}^{J}\sum_k w_k^j h^j_{n-2^j k} + \sum_k v_k^J g^J_{n-2^J k} \tag{3}$$

where $g^j = \left(\mathbf{G}\uparrow\right)^{j-1} g$, $h^j = \left(\mathbf{G}\uparrow\right)^{j-1} h$. However, since this filter bank is critically sampled, used filters are constrained to satisfy the following condition to achieve perfect reconstruction (without delay), here valid in case of simple two-band filter bank:

$$x - \mathbf{G}\uparrow\downarrow\mathbf{G}'x = \mathbf{H}\uparrow\downarrow\mathbf{H}'x \tag{4}$$

It can be easily extended to *J*-level decomposition case.

## 3. RECURRENT MULTISCALE NETWORK

The aforementioned idea of mitigating the vanishing gradient effect is now implemented through the architectural solution named Recurrent Multiscale Network (RMN). The name reflects the fact that the network combines information about the two main tools considered: *recurrent neural networks* and *multiscale decomposition*.

Such a structure is composed of three stages. As shown in Fig.1, each of them is well separated from the other ones, i.e. they work independently and perform in sequence during RMN simulations. Our RMN recalls the architectures proposed in [5], [6] which present common feed-forward neural networks (FFNN) in their second stage. Therefore their performances can not be seen under the point of view of vanishing gradient problem, since no kind of recurrent structures occur in these architectures.

### 3.1. DWT section

The first section, called *DWT section* (DWTsec in Fig.1), fulfils the function of preprocessing the signal, giving as outputs the coefficient sequences resulting from Discrete Wavelet Decomposition. The output dimensionality of this section depends on the chosen decomposition level *J* (equal to *J*+1). Original signal is processed in a 'batch' way: the complete signal is decomposed in order to have the corresponding coefficient sequences with different length (*direct decomposition*). In particular, short sequences are related to low frequencies and to the long term history of the signal; in contrast, long sequences describe high frequency components and short term history of the signal.

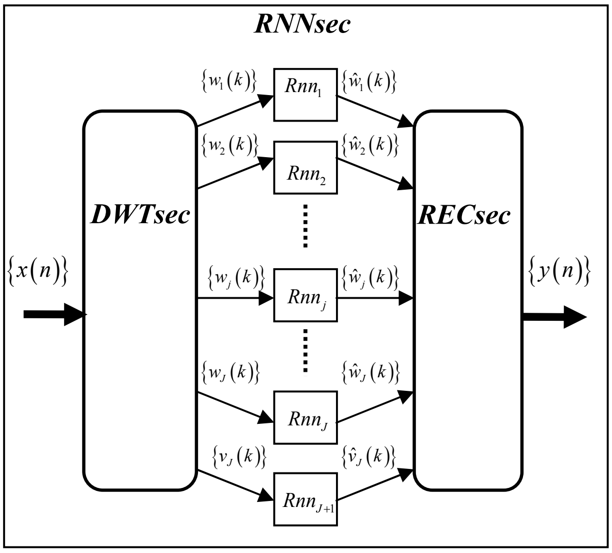

**Figure 1.** Recurrent Multiscale Network. It is composed of 3 independently working stages.

Another procedure, not here performed but employed in [5], [6], for preparing input data of second stage can be considered (*indirect decomposition*): it is based on transforming a segment of the original time series each time instant and retaining the last coefficient for each wavelet coefficient. This preprocessing solution lets all coefficient sequences have the same length, but does not preserve the advantages of temporal resolution differentiation at each decomposition level, that is inversely proportional to frequency (due to combination of filtering and decimating operations in DWT).

### 3.2. RNN section

The second section, RNN section (RNNsec in Fig.1), consists of a set of independent RNNs. There are as many RNNs as the number of output lines in the first stage. In fact each network has one of the coefficient sequences as its only input; consequently RNNs and output lines of DWT section are strictly associated. The target for each network is obtained by decomposing the original target through DWT and taking relative coefficient sequences. All networks are trained in a 'batch' and completely autonomous way. This is a benefit of RMN: in fact all networks can work in parallel.

Finally, it has to be noted that only *globally* (or *fully*)-RNNs (gRNNs) have been considered in this work. The learning algorithm typically used is BPTT (its 'batch' version, namely Epoch-wise BPTT).

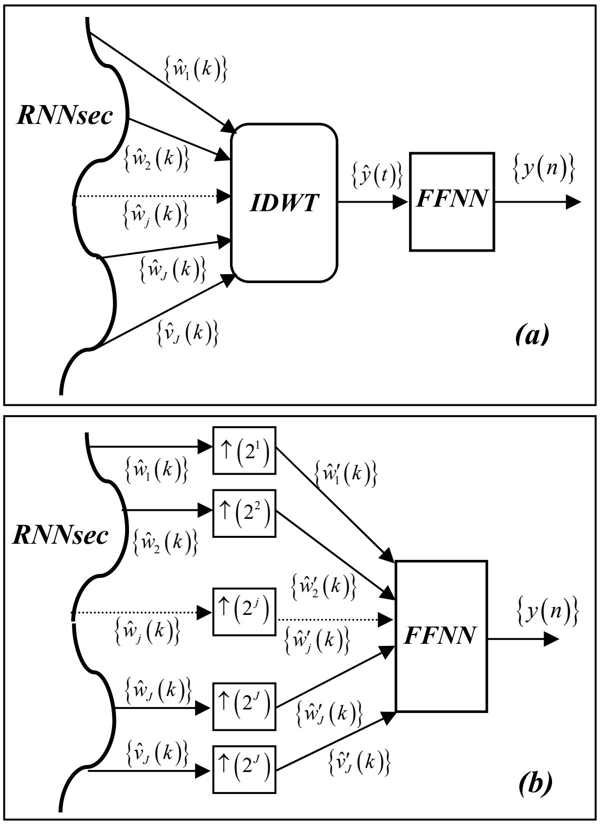

**Figure 2**. Two architectural implementations of third RMN stage. (a) IDWT-FFNN chain based solution. (b) intBANK-FFNN chain based solution.

### 3.3. REC section

In third section, *REC section*, (RECsec in Fig.1), the final result is delivered in the same domain of the original input signal. Sequences coming from the second stage are the inputs of this section. Two possible solutions have been developed for this RECsec. The first one is the *IDWT-FFNN chain*, whereas the second one is *intBANK-FFNN chain*, where an interpolation bank is used to get same length input signals for FFNN. In the former structure, reconstruction is performed by IDWT, while nonlinear links among sequences handled in the second stage are recovered through FFNN. The role of feed-forward neural network (generally trained through Levenberg Marquadt learning algorithm) is fundamental to achieve a good final result, since IDWT is not able to guarantee perfect reconstruction alone for the presence of nonlinear operations before itself. The latter solution has the neural network as reconstruction operator, allowing not to choose analysis and synthesis filters constrained by perfect reconstruction conditions.

Preprocessing operation performed by interpolation bank is needed to get suitable inputs for FFNN training; in this case the input dimensionality of FFNN (without considering the presence of input memory) coincides with the number of output lines of first stage, while in IDWT-FFNN chain solution it is equal to 1. It has to be observed that a cascade of interpolation filter and FFNN input memory for each single RNNsec's output can be seen as an adaptive FIR filter. Many parameters are user defined, and can concern type of filtering operation in each intBANK item (as ideal hold, zero padding or wavelet filters), type and depth of input memory. Such choices are strictly related to the handled task and can relevantly influence the right working of the reconstruction stage and, consequently, the global generalization performance.

Finally, in case of indirect decomposition no kind of interpolation filter set is needed, as all sequences have the same length. Anyway, for the same reason, IDWT-FFNN chain based solution can not be used as here described.

### 4. LONG-MEMORY TIME SERIES PREDICTION TASK

RMN needs to be applied to adequate tasks in order to assess the effectiveness of the overall structure on reducing vanishing gradient effect, in relation to the behavior of common gRNNs. It means that the considered task has to show the occurrence of the studied problem and a comparison between performances of RMN and those ones of gRNNs must be carried out. The chosen one deals with particular types of time series having *long range dependencies* (*LRD*), that have recently attracted much attention over a wide range of engineering applications where need of modeling various non-stationary phenomena occurs. Many fractal processes, named long-memory processes, have been used for this purpose. Such processes can be formally defined [7], together with the short-memory processes, generally used for stationary modeling. They can be essentially differentiated for their autocorrelation function (ACF), that, given a discrete time process $\{x_n\}$, are respectively:

$$C_{x,long}(t) \approx B|t|^d, \quad 0<d<1 \quad d = \textit{long-memory parameter} \quad (5)$$

$$C_{x,short}(t) \approx B\rho^{|t|}, \quad 0<\rho<1 \quad \rho = \textit{short-memory parameter} \quad (6)$$

Moving from these equations, time series with controllable memory measured by ACF have been generated. In the case of short-memory processes, the whitening inverse filter method has been used to get a time series characterized by $C_{x,short}(t) \approx Be^{-t/\tau}$. Moreover, the Cholesky factorization method has been developed to generate a long-memory time series whose ACF is imposed to be the same as that of a fractal process, namely the fractionally differenced Gaussian noise. The closer to one long/short-memory parameter values are, the wider the shape of the relative ACF and longer the temporal dependencies contained in $\{x_n\}$ will be.

A single step prediction task on this kind of series has been performed. The true aim consists of showing that it can represent a valid task to observe poor generalization behavior of full-band gRNNs when vanishing gradient occurs, and that prediction errors can be reduced by using RMN as learning system. Learning and testing time series are 1000-samples long. All prediction errors have been calculated over 10 runs. Depths of input memory are identical in all full-band gRNNs, and longer than in single-band gRNNs. Simulation results (Table 1) show that:

- Vanishing gradient effect has been occurred in full-band gRNNs (varying long-memory parameter $d$ ).
- It has not happened in case of short-memory temporal series (varying short-memory parameter $\rho$ ).

- Applying RMN to the task has led to a relevant improvement of generalisation performances, especially when long memory parameter approaches to high values, for both types of reconstruction section. Two levels of decomposition have been considered (1 and 2).

| *Long/short-memory parameter* | **Learning MSE** | | **Testing MSE** | |
|---|---|---|---|---|
| | **Mean** | **Std** | **Mean** | **Std** |
| *d=0.55* | 0.1003 | $1.4*10^{-4}$ | 0.1031 | $1.3*10^{-4}$ |
| *d=0.65* | 0.0849 | $7.8*10^{-5}$ | 0.1035 | $4.5*10^{-5}$ |
| *d=0.8* | 0.0797 | $2.1*10^{-4}$ | 0.0907 | $2.8*10^{-4}$ |
| *d=0.95* | 0.0317 | $5.1*10^{-5}$ | **0.5894** | 0.2425 |
| $\tau = 5$ | 0.0338 | $5.2*10^{-5}$ | 0.0390 | $3.9*10^{-5}$ |
| $\tau = 15$ | 0.0164 | $1.3*10^{-5}$ | 0.0168 | $5.5*10^{-5}$ |
| $\tau = 30$ | 0.0101 | $1.5*10^{-6}$ | 0.0106 | $9.5*10^{-6}$ |
| $\tau = 100$ | 0.0050 | $4.2*10^{-5}$ | 0.0060 | $6.8*10^{-7}$ |

***(a)***

| *Long-memory parameter* | **1levelRECsec MSE** | | **2levelRECsec MSE** | |
|---|---|---|---|---|
| | **Mean** | **Std** | **Mean** | **Std** |
| *d=0.55 learning* | 0.0941 | 0.0217 | 0.0921 | 0.0101 |
| *testing* | 0.0913 | 0.0166 | 0.0960 | 0.0121 |
| *d=0.65 learning* | 0.0916 | 0.0359 | 0.0716 | 0.0116 |
| *testing* | 0.0904 | 0.0165 | 0.1017 | 0.0087 |
| *d=0.8 learning* | 0.0608 | 0.0188 | 0.0794 | 0.0122 |
| *testing* | 0.0924 | 0.0355 | 0.0822 | 0.0099 |
| *d=0.95 learning* | 0.0850 | 0.0732 | 0.0862 | 0.0820 |
| *testing* | **0.1065** | 0.0314 | **0.1186** | 0.0435 |

***(b)***

| *Long-memory parameter* | **1levelRECsec MSE** | | **2levelRECsec MSE** | |
|---|---|---|---|---|
| | **Mean** | **Std** | **Mean** | **Std** |
| *d=0.55 learning* | 0.0782 | 0.0064 | 0.0727 | 0.0037 |
| *testing* | 0.0895 | 0.0200 | 0.0883 | 0.0203 |
| *d=0.65 learning* | 0.0728 | 0.0068 | 0.0810 | 0.0170 |
| *testing* | 0.1101 | 0.0140 | 0.0818 | 0.0122 |
| *d=0.8 learning* | 0.1026 | 0.0218 | 0.0748 | 0.0139 |
| *testing* | 0.0984 | 0.0232 | 0.0679 | 0.0096 |
| *d=0.95 learning* | 0.0896 | 0.0731 | 0.0629 | 0.0137 |
| *testing* | **0.1784** | 0.2172 | **0.0854** | 0.0201 |

***(c)***

| *Long-memory parameter* | **RNNsec-1 MSE** | | **RNNsec-2 MSE** | |
|---|---|---|---|---|
| | **Mean** | **Std** | **Mean** | **Std** |
| *d=0.55 learning* | 0.0815 | 0.0134 | 0.0849 | 0.0181 |
| *testing* | 0.0793 | 0.0213 | 0.0940 | 0.0245 |
| *d=0.65 learning* | 0.0664 | 0.0034 | 0.0894 | 0.0147 |
| *testing* | 0.0921 | 0.0240 | 0.1080 | 0.0169 |
| *d=0.8 learning* | 0.0486 | 0.0097 | 0.0925 | 0.0208 |
| *testing* | 0.0664 | 0.0247 | 0.1111 | 0.0640 |
| *d=0.95 learning* | 0.0238 | 0.0056 | 0.1097 | 0.0156 |
| *testing* | 0.1713 | 0.1140 | 0.1439 | 0.0270 |

***(d)***

**Table 1.** (a) Learning/testing performances of a full-band gRNN. (b) RMN learning/testing performances: *intBANK-FFNN chain* case. (c) RMN learning/testing performances: *IDWT-FFNN chain* case. (d) Learning/testing performances of RMN's second stage (relative to level 1 decomposition and valid for both types of reconstruction section).

Further simulations about multi-step prediction have been carried out, and results as good as in single-step prediction, obtained. More tests should be done by using long-memory time series long more than 1000 samples: different generating techniques should be involved to reduce the computational cost of methods here proposed.

Moreover, it has to be pointed out that wavelet decomposition is well suited to deal with $1/f$ processes, as it tends to de-correlate all coefficient sequences [8], thus reducing the amount of long-memory information they contain. This aspect has been confirmed by good single prediction performances of each single-band gRNN in the second stage (as shown in Table 1(d)) then resulting in the global improved behavior; consequently, it has allowed demonstrating the effectiveness of idea which RMN is based on.

## 5. CONCLUSIONS

In this preliminary work, an original architecture termed the RMN is presented, which is composed of different stages and is based on multi-band preprocessing operation of the input signals. The network has been applied to a sample task to assess its effectiveness on reducing the vanishing gradient effect. The selected task, namely time series prediction, has shown to be very useful to stress the occurrence of this typical RNN problem. Relevant results have been obtained, showing how DWT lets single band gRNNs in second stage of RMN be learned only on a particular temporal scale of original signal, leading to a general improvement of global results. Further studies on simulation sets and implementation tools in all stages of RMN could help to understand better RMN behavior concerning the vanishing gradient problem. For example, different decomposition techniques as Pyramid Transform or Wavelet Packet could be tested and their usefulness assessed for every selected task.

## 6. REFERENCES

[1] R.J. Williams, and D. Zipser, "Gradient-based learning algorithms for recurrent networks and their computational complexity", in *Back-propagation, Theory, Architectures and Applications*, Y. Chauvin, and D.E. Rumelhart eds., Hillsdale, N.J.: Lawrence Erlbaum Publishers, 1995, pp. 433-486.

[2] J.F. Kolen, and S.C. Kremer, *A Field Guide to Dynamical Recurrent Networks*, IEEE Press, 2001.

[3] Haykin, S., *Neural networks – A Comprehensive foundation*, Prentice Hall, Englewood Cliffs, NJ, 1999.

[4] M. Vetterli, and J. Kovacevic, *Wavelets and Subband Coding,* Prentice Hall, Englewood Cliffs, NJ, 1995.

[5] A.B. Geva, "Scale-Net – Multiscale Neural-Network Architecture for Time Series Prediction", *IEEE Transactions on Neural Networks,* vol. 9, no. 5, pp. 1471-1482, 1998.

[6] B.L. Zhang, R. Coggins, M.A. Jabri, D. Dersch, and B. Flower. "Multiresolution Forecasting for Futures Trading Using Wavelet Decompositions", *IEEE Transactions on Neural Networks*, vol.12, no.4, pp. 765-775, 2001.

[7] R.J. Bhansali, and P.S. Kokoszka, "Prediction of Long-Memory Time Series: an Overview", *Estadistica*, to be published.

[8] A.H. Tewfik, and M. Kim, "Correlation Structure of the Discrete Wavelet Coefficients of Fractional Brownian Motion", *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 904-909, 1992.