Recurrent neural networks as approximators of non-linear filters operators

E Solovyeva
Theoretical Electrical Engineering Department, St. Petersburg Electrotechnical University, 5 Prof. Popov St., St. Petersburg, 197376
E-mail: selenab@hotbox.ru

Abstract. In cases that the mathematical model of a device is complicated or it can not be constructed because of the absence of sufficient information about a researched object, the approach based on the description of unique relationship between the sets of input and output signals is used. The point is the approximation of a non-linear operator, establishing the unique input-output mapping, by mathematical constructions (multidimensional polynomials, regression models, neural networks). Demand to reach the high accuracy of approximation often arises in practice. Recurrent neural networks possessing the properties of dynamics and nonlinearity are considered as mathematical models within the framework of the input-output approach. The non-linear filters synthesis is the constructing of mathematical models using the available sets of input and output signals. Non-linear filters serve for cancelling non-Gaussian noise from distorted signals. As an example, images distorted by the impulse noise is recovered with the help of non-linear filtering performed by different kinds of neural networks.

1. Introduction
The mathematical modelling of non-linear dynamic systems is based on the “black box” approach, when the component descriptions of dynamic systems are too complicated or don’t contain sufficient information for constructing mathematical models. Within the framework of this approach, a non-linear object is represented as “black box”, and its behavioural model is built accounting for the sets of input and output signals [1]–[3]. One of the behavioural model forms is a neural network.

Neural networks with one or more feedback connections are referred to as recurrent networks. Feedback gives opportunity to accumulate information and use it later [3]–[6].

In view of the possible feedback location, recurrent networks can be divided into two groups:
- Globally recurrent networks with the feedback allowed between neurons of one layer or different layers. Such networks incorporate a static multi-layer perceptron or parts of it. Moreover, they apply the non-linear mapping capability of the multi-layer perceptron. Basically, one can be distinguished four kinds of globally recurrent networks:
  - Fully recurrent networks (the Real Time Recurrent Network (RTRN));
  - Partially recurrent networks (the Elman structure, the Jordan structure, the Recurrent Multi-Layer Perceptron Network (RMLPN));
  - State-space networks;
  - Cellular neural networks (CNN).
Locally recurrent networks with the feedback situated only inside neurons. This means there are no feedback loops between the neurons of successive layers and between the neurons of the same layer. Among locally recurrent networks one can be distinguished the following structures:

- The networks consisting of static feedforward and, so-called, dynamic neurons;
- The block-oriented neural networks of Wiener, Hammerstein, Wiener-Hammerstein and etc.

The major properties of various kinds of recurrent neural networks are described below.

2. **Fully recurrent networks**

The most general architecture of globally recurrent neural networks is often called the Real Time Recurrent Network (RTRN), because it has been designed for the real time signal processing. Any connections between neurons are allowed, that is why a fully connected neural architecture is obtained. The RTRN known as the Williams-Zipser network [3], [5] is shown in figure 1.

![Figure 1. The RTRN structure.](image)

The RTRN consists of external and feedback inputs, outputs, as well as a processing layer. At instant $n$ for the $i$-th neuron, its weights form the weight vector $W_i^T(n) = [w_{i,1}(n), ..., w_{i,p+N+1}(n)]$. 
where $p$ is the number of external inputs, $N$ is the number of the feedback loops, $T$ denotes the transpose operation. One additional element of the weight vector $W_i^T(n)$ is the bias input weight. Feedback consists of delayed output signals.

The RTRN depicted in figure 1 is described by the following equations:

$$y_i(n) = F(v_i(n)), \quad i = 1, 2, ..., N,$$

$$v_i(n) = \sum_{l=1}^{p+N+1} w_{i,l}(n)u_l(n),$$

$$U_i^T(n) = [s(n-1), ..., s(n-p), 1, y_1(n-1), y_2(n-1), ..., y_N(n-1)],$$

where the vector $U(n)$ comprises both external and feedback inputs to the network, as well as the bias input that is unity valued constant.

The fundamental advantage of these networks is the possibility to approximate a wide class of dynamic relations.

Such a kind of networks, however, exhibits some well-known disadvantages. First of them is large structural complexity. In addition, the network training is usually complicated and slowly convergent. There are problems with keeping the network stability. Moreover, the fixed relation between the number of states and the number of neurons does not allow to adjust the dynamics order and non-linear model properties separately. Generally, fully recurrent networks are too complex for applying [5].

3. Partially recurrent networks

Partially recurrent networks have less general character. Contrary to fully recurrent networks, the architecture of partially recurrent networks is based on the multi-layer feedforward perceptron with the additional layer of units that is called the context layer. The neurons of this layer serve as the internal states of a model.

Among many well-known structures, three partially recurrent networks have attracted considerable attention: the Elman structure, the Jordan structure and the Recurrent Multi-Layer Perceptron Network (RMLPN) [3]–[5].

Partially recurrent networks surpass fully recurrent networks in advantages, namely, their recurrent links are more structured and it leads to faster training and fewer problems of stability. Nevertheless, the number of states is still strongly related to the number of hidden neurons (the Elman network) or output neurons (the Jordan network), that severely restricts the flexibility of networks.

3.1. The Elman network

The Elman network [5] is probably the best-known example of a partially recurrent neural network. The implementation of these networks is considerably less expensive than of the multi-layer perceptron with the tapped delay lines.

This network possesses one hidden layer with the feedback loop from the hidden layer to the additional input called the context layer.

The Elman network spreads to control systems of moving objects, to detecting the change in signal characteristics [5].

3.2. The Jordan network

The Jordan network [5] solves the same task class like the Elman network, however, it has better approximating and predictive properties due to deeper memory and the additional layer of non-linear activation functions.
This network consists of the multilayer perceptron with one hidden layer and the feedback loop from the output layer to the additional input called the context layer. In the context layer, there are self-recurrent loops with the forgetting positive coefficient, which is less unit.

The Jordan network has been successfully applied to recognize and differentiate various output time-sequences or to classify various sequences [5].

3.3. The Recurrent Multi-Layer Perceptron Network

The Recurrent Multi-Layer Perceptron Network (RMLPN) [4], [5] is based on the multi-layer perceptron network with adding delayed links between neighbouring units of the same hidden layer (recurrent links) and with including unit feedback on itself (cross-talk links). The RMLPN is depicted in figure 2.

![Figure 2. The RMLPN structure.](image)

Empirical evidence indicates that the RMLPN is enable to simulate a large class of non-linear dynamic systems by means of using delayed recurrent and cross-talk weights.

The feedforward part of the network still maintains the well-known curve-fitting properties of the multi-layer perceptron, while the feedback part provides its dynamic character. Moreover, the usage of the past process observations is not necessary, because their effect is captured by internal network states.

The RMLPN has been successfully used as a model for the dynamic system identification. However, the drawback of this dynamic structure is strictly dependence of the network complexity on the number of hidden neurons and the training time [4], [5].

4. State-space networks

Another kind of globally recurrent neural networks known as a state-space neural network [4], [5] is shown in figure 3. The output of a hidden layer is fed back to the input through the bank of unit delays. The number of unit delays determines the system order. A user can choose how many neurons are applied in feedback.
Let $U(n) \in \mathbb{R}^d$ be the input vector, $X(n) \in \mathbb{R}^d$ be the output of the hidden layer at instant $n$, $Y(n) \in \mathbb{R}^N$ be the output vector. Then, the state-space model, whose block scheme is depicted in figure 3, is described by equations

$$\begin{align*}
X(n+1) &= F(X(n), U(n)), \\
Y(n) &= CX(n),
\end{align*}$$

where $F(\cdot)$ is the vector of non-linear functions characterizing the hidden layer, $C$ is the matrix of synaptic weights between hidden and output neurons.

Contrary to fully and partially recurrent networks, state-space networks possess a number of advantages [4]:

– state-space models can describe a wide class of non-linear dynamic systems;
– the number of states (the model order) is independent of the number of hidden neurons. In this way, only those neurons that feed their outputs back to the input layer through delays are responsible for defining the network state. Consequently, output neurons are excluded from the state definition;
– since the model states feed the network input, they are easily accessible from outside environment. This property can be useful in case that the state measurements (for instance, initial conditions) are available at some instants.

The state-space model includes several recurrent structures as special cases [4].

Although state-space neural networks seem to be more promising than fully or partially neural networks, one can be encountered a lot of difficulties [4]:

– approximation is only valid on the compact subsets of the state-space and within finite time intervals, thus, interesting dynamic characteristics are sometimes not reflected;
– wrong initial conditions can worsen performance, especially if the short data sets are used for the network training;
– training can become unstable;
– a model after training can be unstable.

In particular, these drawbacks appear in cases that state measurements and initial conditions are not available.

5. **Cellular neural networks**

The cellular neural network consists of neurons, which are locally connected, and dynamics is identical for each node. These neurons are commonly called cells. The connection with the cells outside the $r$-neighbourhood is enabled by the propagation effect of the network dynamics [6], [7]. Each cell in the CNN has an input, internal state and output. Any one cell is connected only to its
neighbouring cells. The cell located at the point \((i, j)\) of the two-dimensional area \(M \times N\) is denoted by \(C_{ij}\), and its \(r\)-neighbourhood \(N_{ij}^r\) is defined by

\[
N_{ij}^r = \{ C_{kl}, \max(|i-k|,|j-l|) \leq r, 1 \leq k \leq M; 1 \leq l \leq N \},
\]

where \(r\) is the size of a neighbourhood (a positive integer number). A set \(N_{ij}^r\) is sometimes referred to as the \((2r+1) \times (2r+1)\) neighbourhood. For the \(3 \times 3\) neighbourhood, \(r\) should be 1. Thus, parameter \(r\) controls the connectivity of the cell, i.e. the number of active synapses that connects the cell with its immediate neighbours.

The CNN is entirely characterized by the set of non-linear differential equations associated with cells [6], [7].

The mathematical model for the state equation of single cell \(C_{ij}\) is given by the following relation

\[
\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + \sum_{kl \in N_{ij}^r} A_{ij,kl}y_{kl}(t) + \sum_{kl \in N_{ij}^r} B_{ij,kl}u_{kl}(t) + I_{ij},
\] (1)

where \(x_{ij}(t)\) denotes the state of cell \(C_{ij}\); \(t\) is a continuous time; \(y_{kl}(t)\), \(u_{kl}(t)\) denote respectively the output and input of cells \(C_{kl}\) located in the influence sphere with radius \(r\); \(A_{ij,kl}\) and \(B_{ij,kl}\) are the feedback and feedforward templates, respectively; \(I_{ij}\) is the bias term.

In many applications, the CNN is isotropic, that is space-invariant [6], [7]. Isotropic network is characterized by the parameters of the above-mentioned equation, which are fixed for the entire neural network. For the isotropic CNN, for instance, under \(r = 1\), the terms of equation (1) are represented as

\[
\sum_{kl \in N_{ij}^r} A_{ij,kl}y_{kl}(t) = \sum_{|k-i| \leq 1} \sum_{|l-j| \leq 1} A(i-k,j-l)y_{kl}(t),
\]

\[
\sum_{kl \in N_{ij}^r} B_{ij,kl}u_{kl}(t) = \sum_{|k-i| \leq 1} \sum_{|l-j| \leq 1} B(i-k,j-l)u_{kl}(t).
\]

The output signal of cell \(C_{ij}\) is given by the following equation

\[
y_{ij}(t) = f(x_{ij}(t)),
\]

where \(y_{ij}(t)\) denotes the output signal of cell \(C_{ij}\), \(f(\cdot)\) is a non-linear function.

Generally, \(f(\cdot)\) can be any non-linear continuous function. The non-linear function is specified as the unity gain piecewise linear saturation function, the signum function (so-called hard limiter), the step function, the sigmoid function [3]–[7].

Cellular neural networks are very suited for the high-speed parallel signal processing, as well as solving partial differential equations [6], [7].

6. Locally recurrent networks
Locally recurrent networks are divided into two kinds. Including dynamic neurons is characteristic of one kind [5] and block-oriented structure is characteristic of another kind [8]. Both kinds of locally recurrent networks are represented below.
6.1. Locally recurrent networks with dynamic neurons
In locally recurrent networks, neurons described by dynamic models are connected by static links. Differences between the dynamic neuron models are determined by the localization of the feedback loops.

6.1.1. Dynamic neuron model with output feedback. The dynamic neuron has a local feedback from the neuron output to its input [5]. The output signal of the neuron is processed by a filter with the finite impulse response (FIR filter), the output signals of the filter are summed with weighted external signals impacting on the neuron. Then, the total signal is treated by the activation function.

The model of this dynamic neuron is described by the following equations [5]

\[ x(n) = \sum_{i=1}^{m} w_i u_i(n) + \sum_{i=1}^{r} d_i y(n-i), \]
\[ y(n) = f(x(n)), \]  

(2)

where \( u_i(n), \ i = 1, 2, \ldots, m \) are external signals; \( w_i, \ i = 1, 2, \ldots, m \) are synaptic weights for external signals; \( d_i, \ i = 1, 2, \ldots, r \) are synaptic weights for the feedback signals; \( x(n) \) is the activation signal; \( f \) is the non-linear activation function; \( y(n) \) is the neuron output signal.

The Hopfield network is known to be based on the mentioned dynamic neuron [4], [5]. The Hopfield network is used as associative memory, the apparatus for combinatorial optimization, as well as an interface between analog and digital devices. The physical nature of the analog Hopfield network topology causes the possibility of its VLSI implementation [4], [5].

6.1.2. Dynamic neuron model with activation feedback. The dynamic neuron has a local feedback from the activation unit to input [5]. External signals and the states of the activation signals are fed to the neuron input. The sum of the weighted input signals is transformed into the neuron output signal by means of the activation function.

The non-linear model of this neuron is described by two equations. One of them is equation (2) for the neuron output signal \( y(n) \), another is of form of

\[ x(n) = \sum_{i=1}^{m} w_i u_i(n) + \sum_{i=1}^{r} d_i x(n-i), \]  

(3)

where \( u_i(n), \ i = 1, 2, \ldots, m \) are external signals; \( w_i, \ i = 1, 2, \ldots, m \) are synaptic weights for external signals; \( d_i, \ i = 1, 2, \ldots, r \) are synaptic weights for the feedback signals; \( x(n) \) is the activation signal.

The summation in the second term on the right-hand side of expression (3) can be interpreted as the processing of the activation signal by the FIR filter.

6.1.3. Dynamic neuron model with two feedback loops. The activation and output signals of the neuron with two feedback loops are transformed by different FIR filters mapping the respective feedback loops.

The non-linear model of this neuron is described by two equations. One of them is equation (2), another is written as

\[ x(n) = \sum_{i=1}^{m} w_i u_i(n) + \sum_{i=1}^{r} d_i x(n-i) + \sum_{i=1}^{k} a_i y(n-i), \]
where \( u_i(n) \), \( i = 1, 2, ..., m \) are external signals; \( w_j \), \( i = 1, 2, ..., m \) are synaptic weights for external signals; \( x(n) \) is the activation signal; \( d_i \), \( i = 1, 2, ..., r \) are synaptic weights for the states of the activation signal; \( a_i \), \( i = 1, 2, ..., k \) are synaptic weights for the states of the output signal.

6.2. Block-oriented recurrent networks

The output signal of a block-oriented recurrent network is the combination of the output signals of the network units (blocks). Identifying the structure, evaluating the parameters and verifying the model of the neural network using the submodels of separate units are often more convenient than similar operations performed in case of the complete network model.

The block-oriented network consists of non-linear static and linear dynamic units [8]. The simple connections of such units are the following:

- the Wiener structure, in which a linear dynamic unit is followed by a memoryless (inertialess) nonlinearity. The operational Wiener model is written as

\[
y(n) = f(\{H(q^{-1})[u(n)]\}),
\]

where \( u(n) \), \( y(n) \) are the input and output signals, respectively; \( H(q^{-1}) \) is a linear operator affecting the input signal \( u(n) \); \( q^{-1} \) is the time-delay operator for one step; \( f \) is the non-linear activation function.

The linear operator is described by formula

\[
H(q^{-1}) = B(q^{-1})/A(q^{-1}),
\]

where \( A(q^{-1}) \), \( B(q^{-1}) \) are linear operators represented respectively by expressions

\[
A(q^{-1}) = 1 + a_1q^{-1} + ... + a_Nq^{-N},
\]

\[
B(q^{-1}) = b_1q^{-1} + ... + b_Nq^{-N};
\]

- the Hammerstein structure, in which a memoryless nonlinearity is followed by a linear dynamic unit. The operational Hammerstein model is given by

\[
y(n) = H(q^{-1})[f(u(n))].
\]

In this model, linear operator \( H(q^{-1}) \) affects a signal that is the result of processing the input signal \( u(n) \) by the non-linear activation function \( f \).

The Wiener and Hammerstein structures are combined to create more general block-oriented recurrent networks [8].

7. The modelling of non-linear filters

The problem of filtering non-Gaussian noise, for instance, the impulse noise is often effectively solved within the framework of the “black box” approach [9], [10]. According to this approach the mathematical filter model describes a relationship between the sets of input and output signals.

Let’s consider the filters synthesis on the class of bit-map (dot element) half-tone images under resolution measured by 256 grey levels, i.e., an image is the matrix of integers (the elements of
brightness, pixels) in interval [0; 255]. The pixel format is unit8. The impulse noise represents switched on and switched off pixels (white and black dots in a picture), whose emergence does not depend on the presence of noise spikes at adjacent dots.

On the bases of the input-output relationship, non-linear filters are built as the cascade connections of the following units [10]:

– the median filter and the discrete-time cellular neural network including five cells with the unity gain piecewise linear saturation functions (this filter is referred to as the combined discrete-time cellular neural network, CDTCNN);

– the median filter and the two-layer perceptron network including five neurons with the hyperbolic tangent functions in a hidden layer (this filter is called the combined two-layer perceptron network, CTLPN);

– the median filter and the Volterra filter of the second degree [1]–[3] (this filter is referred to as the combined Volterra filter, CVF).

The parameters of the above mentioned filters are defined on solving the approximation problem in the mean-square norm and using the learning image with the size of 220x148 pixels.

The mean-square errors of filtration of impulse noise with density amounted to 0.5 are summarized in table 1. “Tigers”, “Building” and “Fence” are the names of learning and two test images, correspondently. All the images have the size of 220x148 pixels. Non-linear filters were synthesised on the basis of designing corresponding algorithms in MATLAB.

|        | CDTCNN | CTLPN | CVF  |
|--------|--------|-------|------|
| Tigers | 771    | 841   | 1206 |
| Building| 1083   | 1186  | 1712 |
| Fence  | 1558   | 1800  | 2214 |

One can see from table 1, the offered CDTCNN yields higher filtration precision than the CTLPN and the CVF. It should be observed that the investigated filters provide different accuracy at the nearly equal complexity of them (56 parameters of the CDTCNN, the CTLPN and 54 parameters of the CVF).

8. Conclusion

Recurrent neural networks are represented as the behavioral models of non-linear dynamic systems, by means of which the non-linear operators of these systems are approximated. A non-linear operator establishes unique correspondence between the sets of the input and output signals of a dynamic system.

Depending on the feedback location, neural networks are divided into globally and locally recurrent ones. Inside these groups, the kinds of recurrent networks are distinguished, for which approximating properties, advantages, disadvantages, as well as the application fields are considered.

Information about recurrent neural networks is important for choosing the shape of the device model a priori. The model form is known to influence the accuracy and computational cost of modelling. This aspect is characteristic of the “black box” approach.

Taking into account the example of filtering out the impulse noise from distorted images, it is shown that the model of a non-linear filter based on the cellular neural network relating to globally recurrent neural networks provides higher accuracy of the image restoring in comparison with the multilayer perceptron network and the polynomial model.

References

[1] Ogunfunmi T 2007 Adaptive Nonlinear System Identification. The Volterra and Wiener Model Approaches (New York: Springer Science+Business Media, LLC) p 229
[2] Janczak A 2005 *Identification of Nonlinear Systems Using Neural Networks and Polynomial Models. A Block-oriented Approach* (Berlin: Springer-Verlag Berlin Heidelberg) p 197

[3] Bichkov U A, Inshakov U M, Solovyeva E B and Scherbakov S A 2017 *Analysis of mathematical models of continuous and discrete nonlinear systems* (St Petersburg: Saint Petersburg Electrotechnical University “LETI” Press) p 174

[4] Haykin S 2009 *Neural Networks and Learning Machines* (New York: Pearson Education Inc.) p 906

[5] Patan K 2008 *Artificial Neural Networks for The Modelling and Fault Diagnosis of Technical Processes* (Berlin: Springer-Verlag Berlin Heidelberg)

[6] Chen W K 2009 *Feedback, nonlinear and distributed circuits* (New York: Taylor & Francis Group, LLC.)

[7] Yalçın M E, Suykens J A K and Vandewalle J P L. 2005 *Cellular neural networks, multi-scroll chaos and synchronization* (Singapore: World Scientific Publishing Co. Pte. Ltd.)

[8] Mandic D P and Chambers J A 2001 *Recurrent neural networks for prediction: learning algorithms, architectures and stability* (New York: John Wiley & Sons, Inc.)

[9] Solovyeva E 2017 A split signal polynomial as a model of an impulse noise filter for speech signal recovery *J. Phys.: Conf. Ser.* **803**(1) 012156

[10] Solovyeva E 2016 Nonlinear models of digital filters and their application fields *Proc. Int. Symp. on Problems of Redundancy in Information and Control Systems* (St. Petersburg, 26–29 September 2016) (St. Petersburg: State University of Aerospace Instrumentation) pp 150–4