Locally recurrent neural networks
(locally recurrent networks with dynamic neurons,
block-oriented neural networks)

Locally recurrent neural networks.

Locally recurrent neural networks. There are feedbacks only inside neuron models. This means that there are neither feedback connections between neurons of successive layers nor lateral links between neurons of the same layer.

Among the locally recurrent networks, two types of structures are distinguished:

– the networks having a structure similar to static feedforward one, but consist of the so-called dynamic neurons;

– the block-oriented neural networks of Wiener, Hammerstein, Wiener-Hammerstein, etc.

Let us consider the above-mentioned structures and their corresponding non-linear models.

Locally recurrent networks with dynamic neurons.

In general, differences between dynamic neuron models depend on the localization of internal feedbacks.

Model with local activation feedback. This neuron model was studied by Frasconi, and may be described by the following equations:

$$x(n) = \sum_{i=1}^{m} w_i u_i(n) + \sum_{i=1}^{r} d_i x(n-i), \tag{5.1}$$

$$y(n) = F\left(x(n)\right),$$

where u_i, $i = 1, 2, ..., m$ are the external inputs to the neuron, w_i reflects the input weights, $x(\cdot)$ is the activation potential, d_i, $i = 1, 2, ..., r$ are the coefficients which determine feedback intensity of $x(n-i)$, F is a non-linear activation function.

With reference to Fig. 5.1, the input to the neuron can be a combination of input variables and delayed versions of the activation $x(\cdot)$.

Note that right-hand side summation in (5.1) can be interpreted as the Finite Impulse Response (FIR) filter. This neuron model has the feedback signal taken before the non-linear activation block (Fig. 5.1).

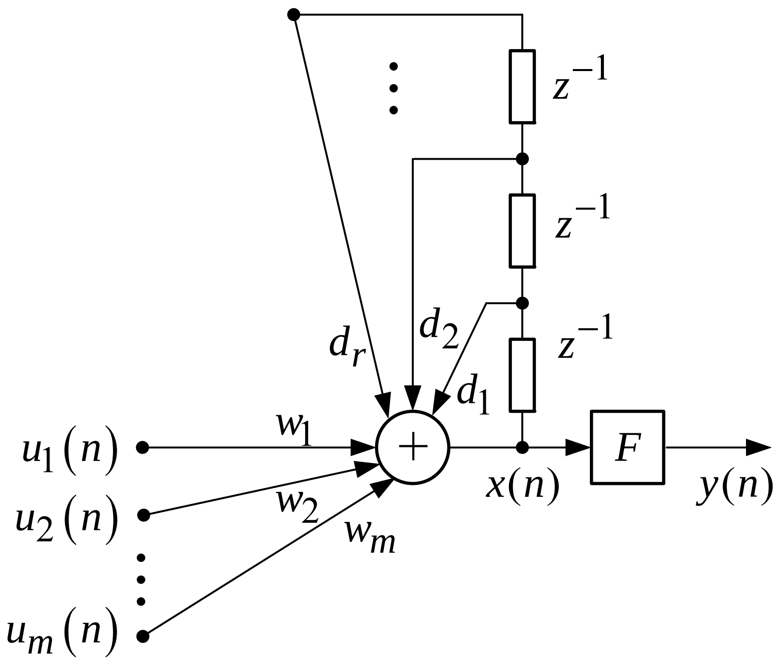


Fig. 5.1. Neuron architecture with local activation feedback

Model with local output feedback. Another dynamic neuron architecture was proposed by Gori (Fig. 5.2). In contrast to local synapse as well as local activation feedback, this neuron model takes feedback after the non-linear activation block. In a general case, such a model can be described as follows:

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

$$y(n) = F\left(x(n)\right),$$

where d_i, $i = 1, 2, ..., r$ are the coefficients which determine feedback intensity of the neuron output $y(n-i)$, $i = 1, 2, ..., r$.

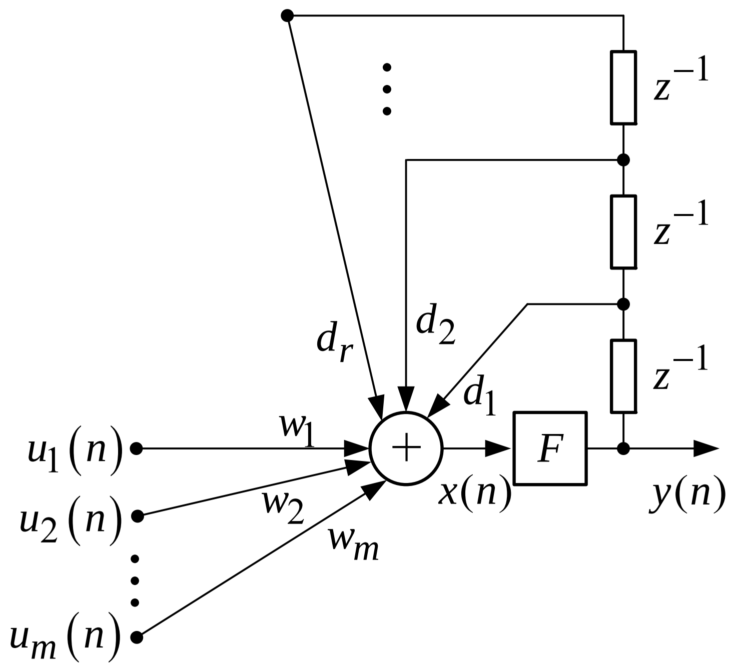


Fig. 5.2. Neuron architecture with local output feedback

In this architecture, the output of the neuron is filtered by the FIR filter, whose output is added to the inputs, providing the activation.

On the basis of dynamic neurons, the Hopfield network is usually constructed in the form of the structure depicted in Fig. 5.3. The input layer only collects and distributes feedback signals from the output layer.

The Hopfield model is the most popular dynamic model. **It is an interconnected recurrent network with McCulloch–Pitts neurons.**

The Hopfield model operates in an unsupervised manner. The dynamics of the network are described by a system of nonlinear ordinary differential equations (ODE). The Hopfield model is a non-linear dynamic devise model in state space.

The discrete form of the Hopfield model is defined by

$$x_i(n+1) = \sum_{j=1}^{r} d_{ij} y_j(n) + b_i,$$

$$y_i(n+1) = F\left(x_i(n+1)\right),$$

where $y_i(n)$ is the output of the i-th neuron, b_i is a bias to the neuron, F is a sigmoidal function, n is the normalized discrete time, taking values 0, 1, 2, …

Correspondingly, the continuous Hopfield model is given by

$$\frac{dx_i(t)}{dt} = \sum_{j=1}^{r} d_{ij} y_j(t) + b_i,$$

$$y_i(t) = F\left(x_i(t)\right),$$

where t denotes the continuous-time variable.

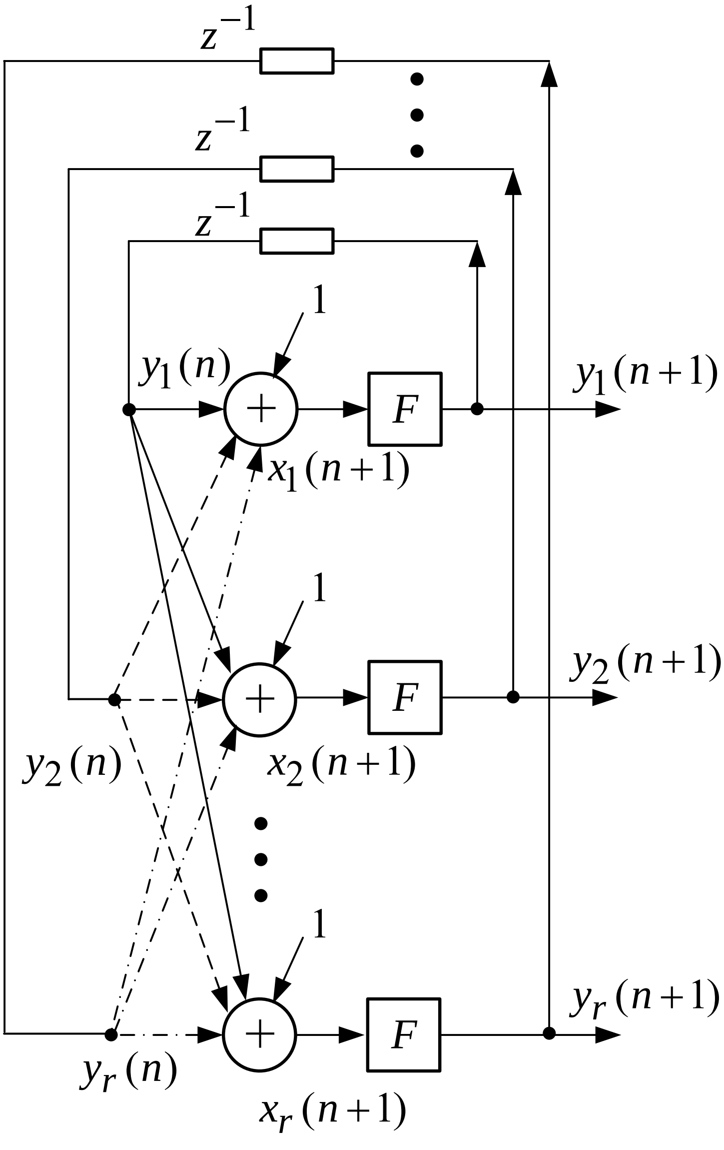


Fig. 5.3. The architecture of the Hopfield network

The Hopfield model is a network model that is most suitable for hardware implementation. Numerous circuits for the Hopfield model have been proposed, including analog VLSI and optical implementations.

Applications of the Hopfield Model

Due to recurrence, the Hopfield network remembers cues from the past and does not complicate the training procedure. The dynamics of the network are described by a system of ODEs and by an associated energy function to be minimized. This network is a stable dynamic system, which converges to a stable fixed point when implemented in asynchronous update mode.

The Hopfield network can be used for converting analog signals into the digital format, for associative memory, and for solving COPs (combinatorial optimization problems).

The Hopfield network can be used as an effective interface between analog and digital devices, where the input signals to the network are analog and the output signals are discrete values. The neural interface has the capability of learning. The neural-based analog-to-digital (A/D) converter adapts to compensate for initial device mismatches or long-term drifts. By adjusting the parameters of the neuron circuit with a learning rule, an adaptive network is generated.

Associative memory is a major application of the Hopfield network. The fixed points in the network energy function are used to store feature patterns. When a noisy or incomplete pattern is presented to the trained network, a pattern in the memory is retrieved. This property is most useful for pattern recognition or pattern completion.

The energy minimization ability of the Hopfield network is used to solve optimization problems.

Model with two loops of feedback. On the basis of the digital filter theory, the more complex structures of a dynamic neuron are constructed, one of which is shown in Fig. 5.4. In the structure, the activation signal and the output signal of the neuron are processed by different FIR filters in the respective feedback loops.

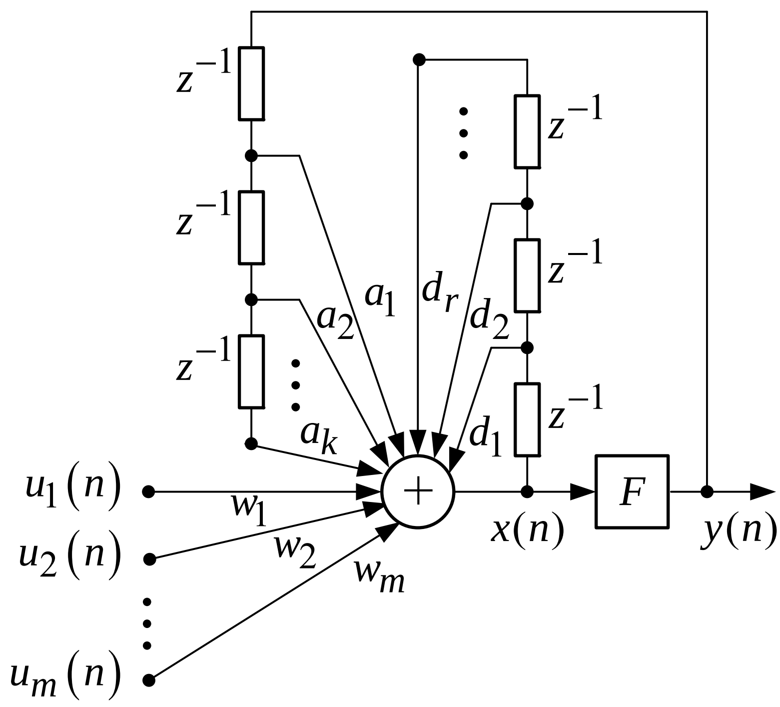


Fig. 5.4. Neuron architecture with two loops of feedback

The nonlinear model of the considered neuron has the form:

$$x(n) = \sum_{i=1}^{m} w_i u_i(n) + \sum_{i=1}^{r} d_i x(n-i) + \sum_{i=1}^{k} a_i y(n-i),$$

$$y(n) = F\left(x(n)\right),$$

where $u_i(n)$, $i = 1, 2, ..., m$ are the external inputs of the neuron; w_i, $i = 1, 2, ..., m$ are the synaptic weights of external input signals; $x(n)$ is the activation signal; d_i, $i = 1, 2, ..., r$ are the synaptic weights of the states of the activation signal in feedback; a_i, $i = 1, 2, ..., k$ are the synaptic weights of the states of the output signal in feedback; F is a nonlinear activation function; $y(n)$ is the output signal of the neuron.

Block-oriented neural networks.

The output signal of a block-oriented network is the combination of output signals of units (modules) of a network. Identifying the structure, evaluating parameters and verifying a neural network model using submodels of different blocks are often more convenient than similar operations performed with an entire network model.

A block-oriented network consists of nonlinear static and linear dynamic modules. Simple connections of such modules are the following:

- The Wiener structure depicted in Fig. 5.16, *a* is the cascade connection between linear dynamic module and nonlinear static (memoryless) one.

The Wiener model is written in the form

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

where $u(n)$, $y(n)$ are input and output signals correspondently; $H\left(q^{-1}\right)$ is the linear operator in rational form affected signal $u(n)$; q^{-1} is the operator of the time delay; f is the nonlinear activation function.

The linear operator is given as

$$H\left(q^{-1}\right) = \frac{B\left(q^{-1}\right)}{A\left(q^{-1}\right)}; \tag{5.2}$$

where $A\left(q^{-1}\right)$, $B\left(q^{-1}\right)$ are linear operators in formulas

$$A\left(q^{-1}\right) = 1 + a_1 q^{-1} + \ldots + a_{N_a} q^{-N_a}; \tag{5.3}$$

$$B\left(q^{-1}\right) = b_1 q^{-1} + \ldots + b_{N_b} q^{-N_b}; \tag{5.4}$$

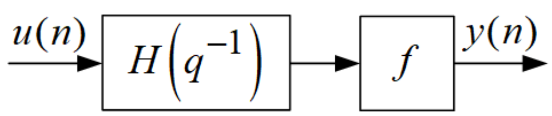
respectively.

- The Hammerstein structure shown in Fig. 5.16, b is the cascade connection between nonlinear static module and linear dynamic one.

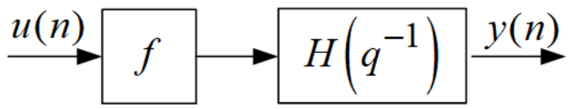
The Hammerstein model is of the form

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

According to this model, linear operator $H\left(q^{-1}\right)$ affects the signal resulted from processing of exciting signal $u(n)$ by nonlinear function f.



a



b

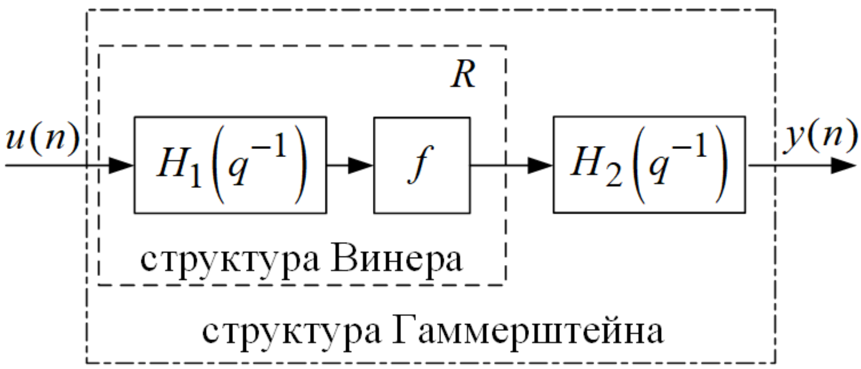
Fig. 5.16. Simple block-oriented structures:
a – the Wiener structure, *b* – the Hammerstein structure

The Wiener and Hammerstein structures are combined for building more complex connections. Let's consider some of these connections.

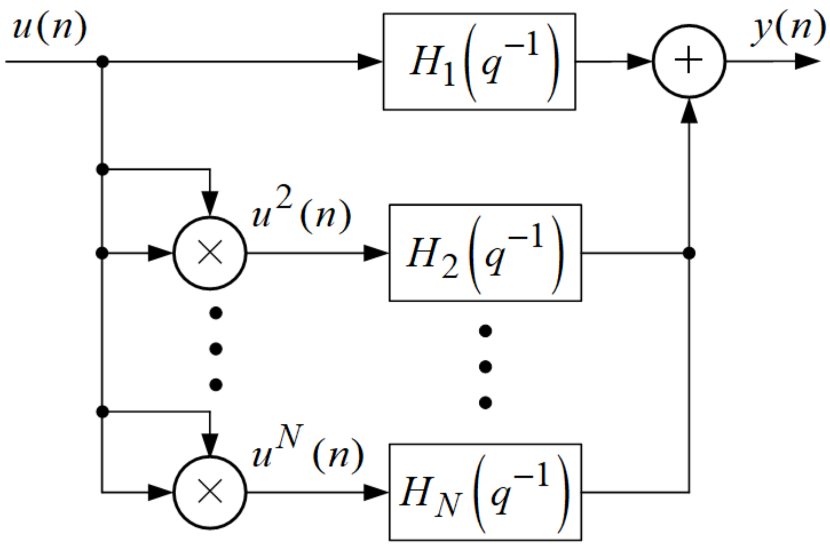
- The Wiener–Hammerstein structure depicted in Fig. 15.17, *a* consists of the Wiener structure (modules $H_1\left(q^{-1}\right)$ and f) as well as the Hammerstein structure (modules R and $H_2\left(q^{-1}\right)$).

The Wiener–Hammerstein model is written in the form of operator equation

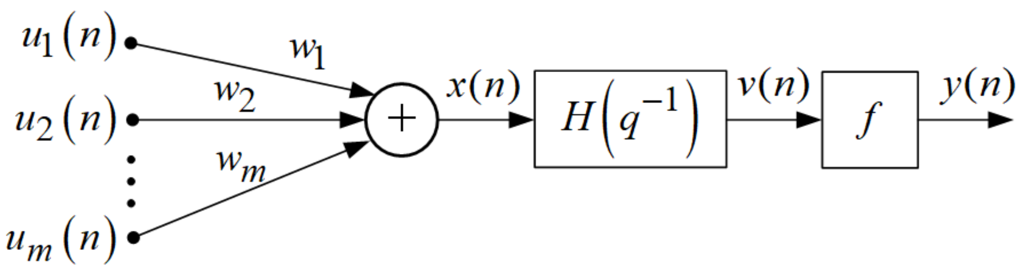
$$y(n) = H_2\left(q^{-1}\right)\left[f\left(H_1\left(q^{-1}\right)[u(n)]\right)\right].$$



a



b



c

Fig. 5.17. The network architectures
based on the Wiener and Hammerstein structures:
a – the Wiener–Hammerstein structure,
b – the generalized Hammerstein structure,
c – dynamic perceptron

• The generalized Hammerstein structure shown in Fig. 15.17, *b* is described by operator model:

$$y(n) = \sum_{i=1}^{N} H_i\left(q^{-1}\right)\left[u^i(n)\right].$$

• Dynamic perceptron shown in Fig. 15.17, *c* has the following operator model:

$$y(n) = f\left(H\left(q^{-1}\right)\left[\sum_{i=1}^{m} w_i u_i(n)\right]\right),$$

where $H\left(q^{-1}\right)$ is a linear operator from expression (5.2).

The set of difference equations:

$$x(n) = \sum_{i=1}^{m} w_i u_i(n),$$

$$v(n) = \sum_{i=1}^{N_b} b_i x(n-i) + \sum_{j=1}^{N_a} a_j v(n-j),$$

$$y(n) = f\left(v(n)\right),$$

where a_i and b_i are the parameters of linear operators $A\left(q^{-1}\right)$ and $B\left(q^{-1}\right)$ from (5.3) and (5.4) accordingly, corresponds to the operator model of dynamic perceptron.