Functional link artificial neural network (FLANN).
Polynomial perceptron network (PPN).
Canonical piecewise-linear neural network (CPWLN).

Functional link artificial neural network (FLANN)

Functional link artificial neural network (FLANN) is one-layer network (there is no hidden layer), and owing to that its algorithm contains less number of transformations and thus, enables fast convergence to approximation problem solving in comparison with traditional neural networks.

The FLANN model has the form

$$y(n) = f\left(\sum_{i=1}^{G} w_i \varphi_i(\mathbf{X}(n))\right) = f\left(\mathbf{W}^t \cdot \mathbf{\Phi}(\mathbf{X}(n))\right), \qquad (6.1)$$

where $y(n)$ is the model output signal; f is the nonlinear activation function; $\mathbf{X}(n)$ is the excitation vector ($\mathbf{X}(n) = [x_1(n), x_2(n), ..., x_Q(n)]^t$); t is the sign of transposition; $\mathbf{W}$ is the network weights vector ($\mathbf{W} = [w_1, w_2, ..., w_G]^t$); $\mathbf{\Phi}(\mathbf{X}(n))$ is the vector of functions φ_i ($i = 1, 2, ..., G$):

$$\mathbf{\Phi}(\mathbf{X}(n)) = [\varphi_1(\mathbf{X}(n)), \varphi_2(\mathbf{X}(n)), ..., \varphi_G(\mathbf{X}(n))]^t.$$

The FLANN structure is depicted in fig. 6.1.

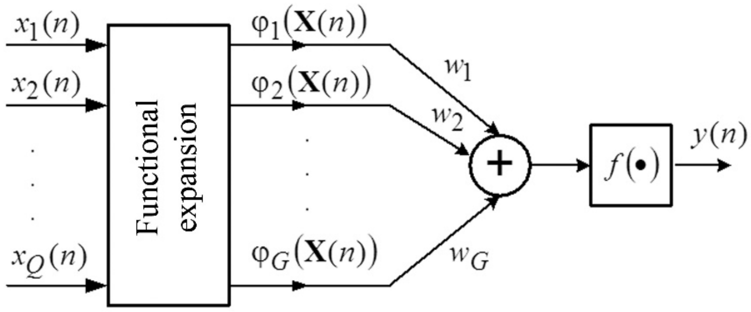

Fig. 6.1. The FLANN structure

Functions φ_i ($i = 1, 2, ..., G$) transform input signals into basis functions by, for instance, trigonometrical polynomial, Chebyshev polynomial, Legendre polynomial and perform the multi-dimensional transformation of basis functions. Basis functions are used in model (6.1) for lowering condition number when the problems of approximation with high nonlinearity degree are being solved.

Let us consider some examples of basis functions formation. Functional expansion of two-dimensional excitation $\mathbf{X}(n) = [x_1(n), x_2(n)]^t$ in the form of trigonometrical polynomials are obtained by vector

$$\mathbf{\Phi}(\mathbf{X}(n)) = [x_1(n), \cos(\pi x_1(n)), \sin(\pi x_1(n)), ..., \cos(2\pi x_1(n)), \sin(2\pi x_1(n)), ...,$$

$$x_2(n), \cos(\pi x_2(n)), \sin(\pi x_2(n)), ..., \cos(2\pi x_2(n)), \sin(2\pi x_2(n)), ..., x_1(n) x_2(n)]^t.$$

One-dimensional polynomials of Chebyshev orthogonal basis is formed from expression

$$T_p(x(n)) = \frac{p}{2} \sum_{k=0}^{p/2} (-1)^k \frac{(p-k-1)!}{k!(p-2k)!} (2x(n))^{p-2k}, \tag{6.2}$$

where $p \in [0, P]$ is the polynomial power. From (6.2) Chebyshev polynomials with low powers are written in the form:

$$\begin{aligned} T_0(x(n)) &= 1, \\ T_1(x(n)) &= x(n), \\ T_2(x(n)) &= 2x^2(n) - 1, \\ T_3(x(n)) &= 4x^3(n) - 3x(n). \end{aligned} \tag{6.3}$$

Functional expansion of vector excitation $\mathbf{X}(n) = [x_1(n), x_2(n), ..., x_Q(n)]^t$ by Chebyshev polynomials (6.3) produces vector in multi-dimensional space

$$\mathbf{\Phi}(\mathbf{X}(n)) = [1, T_1(x_1(n)), T_1(x_2(n)), ..., T_1(x_Q(n)),$$

$$T_1(x_1(n)) T_1(x_2(n)), ..., T_1(x_{Q-1}(n)) T_1(x_Q(n)),$$

$$T_2(x_1(n)), T_2(x_2(n)), ..., T_2(x_Q(n)), \ ...,$$

$$T_P(x_1(n)), T_P(x_2(n)), ..., T_P(x_Q(n))]^t. \qquad (6.4)$$

For example, using the expansion (6.4) for the power $P = 3$ and considering (6.3) we write

$$\mathbf{\Phi}(\mathbf{X}(n)) = [1, x_1(n), x_2(n), ..., x_Q(n), x_1(n)x_2(n), ..., x_{Q-1}(n)x_Q(n),$$

$$2x_1^2(n) - 1, ..., 2x_Q^2(n) - 1, 4x_1^3(n) - 3x_1(n), ..., 4x_Q^3(n) - 3x_Q(n)]^t.$$

FLANN containing Chebyshev basis functions (CFLANN) is called Chebyshev functional link artificial neural network (CFLANN). Let us note that CFLANN corresponds to two-layer perceptron neural network.

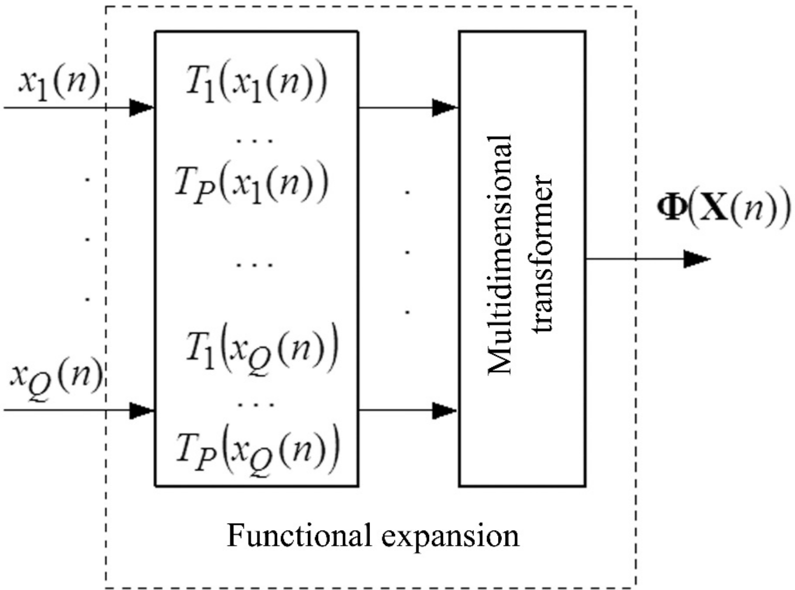

Fig. 6.2. The structure of the unit “Functional expansion” in FLANN

The unit “Functional expansion” included in FLANN scheme (see fig. 6.1) has the structure represented in fig. 6.2. This unit forms Chebyshev polynomials and implements their multi-dimensional transformation.

Polynomial perceptron network (PPN)

Polynomial perceptron network (PPN) is one-layer network (without hidden layer) and that enables algorithm simplicity of its learning and fast convergence to approximation problem solving.

The PPN structure is depicted in fig. 6.3.

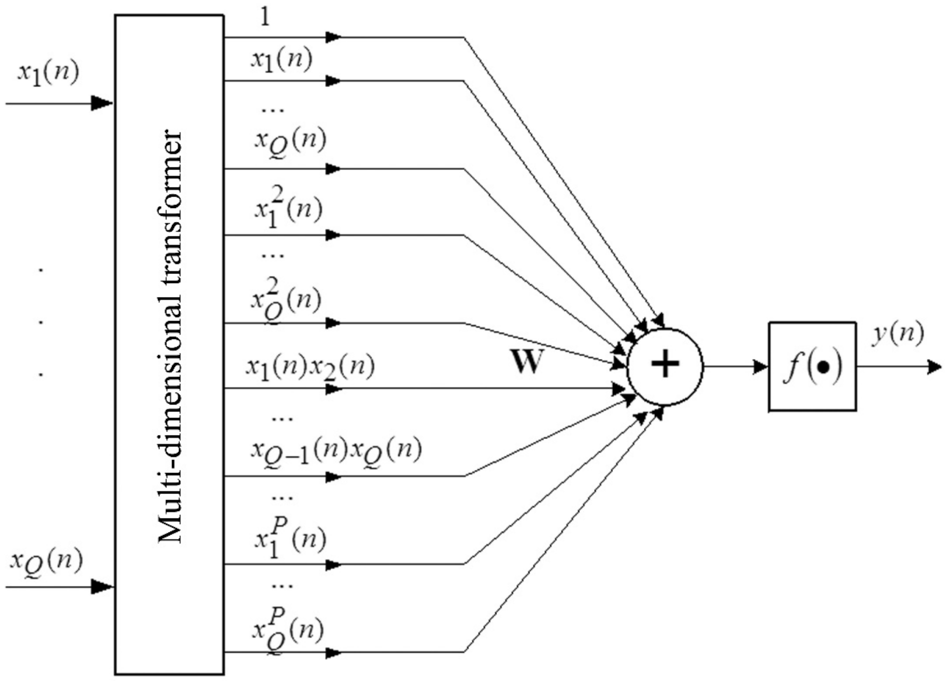

Fig. 6.3. The PNN structure

The PPN model has the form

$$y(n) = f\left(\mathbf{W}^t \cdot \mathbf{F}(\mathbf{X}(n))\right),$$

where $y(n)$ is the model output signal; f is the nonlinear activation function; $\mathbf{X}(n)$ is the excitation vector ($\mathbf{X}(n) = [x_1(n), x_2(n), ..., x_Q(n)]^t$); t is the sign of transposition; $\mathbf{W}(n)$ is the neural network weights vector ($\mathbf{W} = [w_1, w_2, ..., w_G]^t$); $\mathbf{F}(\mathbf{X}(n))$ is the functions vector containing elements with multi-dimensional transformation:

$$\mathbf{F}(\mathbf{X}(n)) = \lfloor 1, x_1(n), ..., x_Q(n), ..., x_1^2(n), ..., x_Q^2(n), ...,$$

$$x_1(n)x_2(n), ..., x_{Q-1}(n)x_Q(n), ..., x_1^P(n), ..., x_Q^P(n)\Big]^t$$

(P is the power of vector-function element).

As an example, we illustrate the structure of PPN with input signals vector $\mathbf{X}(n) = [x_1(n), x_2(n), x_3(n)]^t$ and the multi-dimensional transformer of the second power in fig. 6.4.

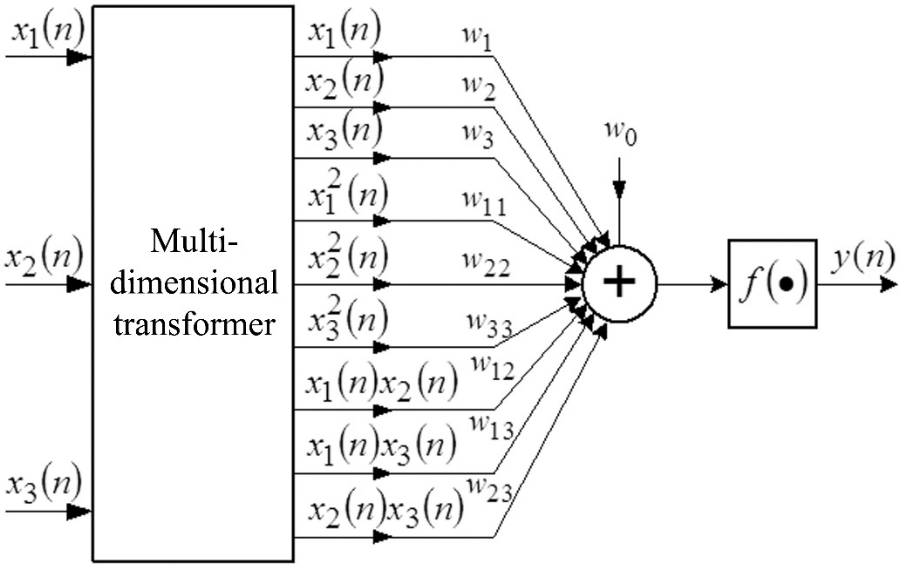

Fig. 6.4. The structure of PPN of the second power

The comparative analysis of PPN and FLANN demonstrates that PPN (fig. 6.3) is transformed into FLANN (see fig. 6.1) by additional transformation of input signals into basis functions on the base of Legendre, Chebyshev polynomials etc.

Canonical piecewise-linear neural network (CPWLN)

Canonical piecewise-linear neural network (CPWLN) together with input signal vector

$$\mathbf{X}(n) = [x_1(n), x_2(n), ..., x_m(n)]^t = [x(n), x(n-1), ..., x(n-m)]^t \qquad (6.5)$$

and output signal $y(n)$ has two (one hidden and output) layers. CPWLN comprises piecewise-linear activation functions.

The mathematical model of CPWLN is written in canonical form:

$$y(n) = a + \mathbf{D}^t \cdot \mathbf{X}(n) + \sum_{i=1}^{N} C_i |q_i(n)| = a + \mathbf{D}^t \cdot \mathbf{X}(n) + \sum_{i=1}^{N} C_i \left| \mathbf{L}_i^t \cdot \mathbf{X}(n) + b_i \right|, \quad (6.6)$$

where $|q_i(n)|$ is the i-th activation function, $q_i(n) = \mathbf{L}_i^t \cdot \mathbf{X}(n) + b_i$; $\mathbf{D} = [d_1, d_2, ..., d_m]^t$; $\mathbf{L}_i = [l_{i,1}, l_{i,2}, ..., l_{i,m}]^t$; b_i ($i = 1, 2, ..., N$) are the parameters of hidden layer; a, C_i ($i = 1, 2, ..., N$) are the parameters of output layer.

Equation (6.6) is represented in the form

$$y(n) = a + \mathbf{D}^t \cdot \mathbf{X}(n) + \sum_{i=1}^{N} C_i \operatorname{sgn}(q_i(n)),$$

where $\operatorname{sgn}(q_i(n)) = +1$, if $q_i(n) \geq 0$; $\operatorname{sgn}(q_i(n)) = -1$, if $q_i(n) < 0$.

The CPWLN structure is depicted in fig. 6.5.

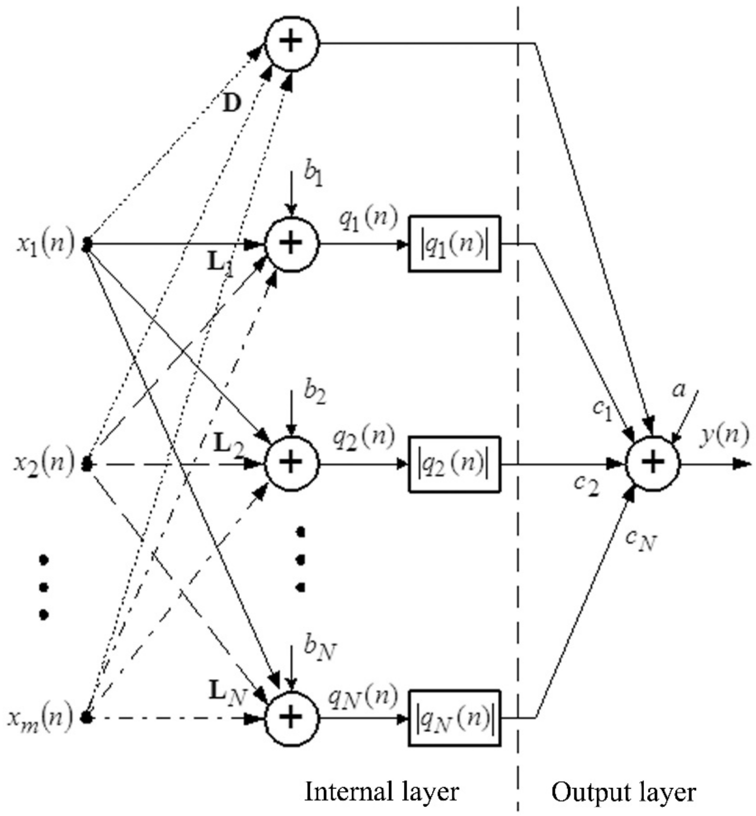

Fig. 6.5. The CPWLN structure

CPWLN advantages are the following:

– general (canonical form) piecewise-linear model application leads to reduction of determined parameters in comparison with separate linear simulation of space domains;

– for simulation of substantial non-linearity less amount of CPWLN parameters is required in comparison with Volterra polynomial;

– activation function presentation is suitable for realization with digital elementary basis use (continuous sigmoid functions are replaced by linear functions combinations).

CPWLN is constructed in the form of cascade connection of units having the structure depicted in fig. 6.5. The cascade structure of CPWLN is depicted in fig. 6.6.

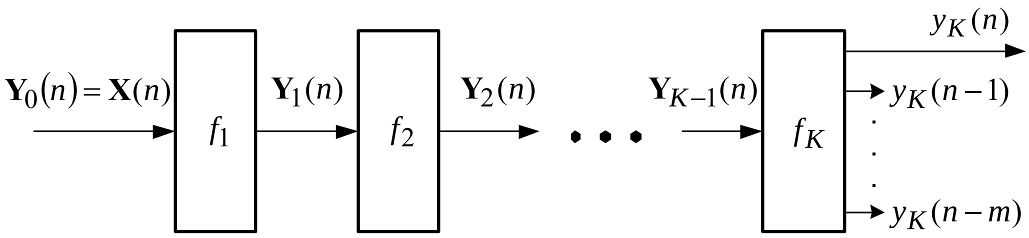

Fig. 6.6. The cascade CPWLN structure

Signal $y_K(n)$ obtained at one output of the last K-th unit is the output signal of cascade CPWLN. The output signal vector of the j-th unit is formed according to the following recurrent equation:

$$\mathbf{Y}_j(n) = a_j + \mathbf{D}_j^t \cdot \mathbf{Y}_{j-1}(n) + \sum_{i=1}^{N} C_{j,i} \left| \mathbf{L}_{j,i}^t \cdot \mathbf{Y}_{j-1}(n) + b_{j,i} \right|, \; j = 1, 2, ..., K, \quad (6.7)$$

where $\mathbf{Y}_0(n) = \mathbf{X}(n)$ is the from (6.5), $\mathbf{Y}_j(n) = \left[y_j(n), y_j(n-1), ..., y_j(n-m)\right]^t$; $\mathbf{D}_j = \left[d_{j,1}, d_{j,2}, ..., d_{j,m}\right]^t$; $\mathbf{L}_{j,i} = \left[l_{j,i,1}, l_{j,i,2}, ..., l_{j,i,m}\right]^t$.

Equation (6.7) is written in generalized form

$$\mathbf{Y}_j(n) = f_j\left(f_{j-1}(\ldots f_1(\mathbf{Y}_0(n))\ldots) \right),$$

where f_j is the piecewise-linear function in canonical form describing the j-th unit.

Recurrent canonical piecewise-linear function network (RCPWLN) as well as non-linear regression models is applied for random processes prediction, e.g. in blind equalizers.

The RCPWLN model has form

$$f\left(\mathbf{Y}(n), \mathbf{X}(n)\right) = \mathbf{A} + \mathbf{D}_0 \cdot \mathbf{Y}(n) + \mathbf{D}_1 f\left(\mathbf{Y}(n-1), \mathbf{X}(n-1)\right) + \mathbf{D}_2 \cdot \mathbf{X}(n), \quad (6.8)$$

where $\mathbf{X}(n)$, $\mathbf{Y}(n)$ are the input (expression (6.5)) and output vector signals of RCPWLN respectively; f is a nonlinear function.

The element $y_k(n)$ of vector $\mathbf{Y}(n)$ in (6.8) is described by the following expression

$$y_k(n) = a_k + \mathbf{G}_{1,k}^t \cdot \mathbf{Y}(n-1) + \mathbf{G}_{2,k}^t f\left(\mathbf{Y}(n-1), \mathbf{X}(n-1)\right) + \mathbf{G}_{3,k}^t \cdot \mathbf{X}(n) + \sum_{i=1}^{r} C_{k,i} \times$$

$$\times \left| \mathbf{L}_{1,k,i} \cdot \mathbf{Y}(n-1) + \mathbf{L}_{2,k,i} f\left(\mathbf{Y}(n-1), \mathbf{X}(n-1)\right) + \mathbf{L}_{3,k,i} \cdot \mathbf{X}(n) + b_{k,i} \right|. \quad (6.9)$$

Table 6.1

The dimensions of variables in RCPWLN models

Dimension	Variable
$N \times 1$	$\mathbf{Y}(n)$, $\mathbf{Y}(n-1)$, $\mathbf{G}_{1,k}$, $\mathbf{L}_{1,k,i}$
$m \times 1$	$\mathbf{X}(n)$, $\mathbf{G}_{3,k}$, $\mathbf{L}_{3,k,i}$
$P \times 1$	$f\left(\mathbf{Y}(n), \mathbf{X}(n)\right)$, $f\left(\mathbf{Y}(n-1), \mathbf{X}(n-1)\right)$, $\mathbf{A}$, $\mathbf{D}_1$, $\mathbf{G}_{2,k}$, $\mathbf{L}_{2,k,i}$
$P \times N$	$\mathbf{D}_1$
$P \times m$	$\mathbf{D}_2$
Scalar	a_k, $C_{k,i}$, $b_{k,i}$

The dimensions of variables in (6.8) and (6.9) are represented in table 6.1.