Information Statistics II

Lecture 11. Outline of artificial neural networks

Layered artificial neural network

— Pattern-to-pattern transformer by multiple connections between layers of neurons.

— A pattern is expressed by a set of values of neurons in a layer. Neurons in one layer are not connected to each other; but those in adjacent layers are connected.

— A pattern in a layer is transformed to the next layer. The value of a neuron in the next layer is determined by weighted sum of the values of connected neurons in the previous layer. The weight coefficient is assigned to each connection.

— Learning ability. Network is optimized by appropriately modifying the weight coefficients using errors between temporal outputs and ideal outputs.

Perceptron (simple perceptron with thresholding function)

— Early neural network model. Learning is applied to only the last two layers.

— Learning is achieved by tuning the weight coefficients to compenssate the error between the ideal output and a temporal output. The amount of correction is proportional to the value of neuron in the input layer (δ-rule).

— Limitation of expressible functions (only *linear-separable* functions can be achieved by two layers)

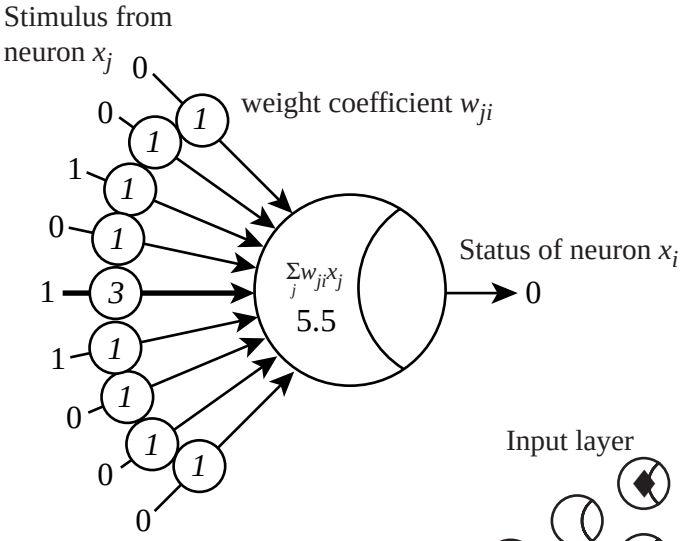


Fig. 1. artificial neuron model.

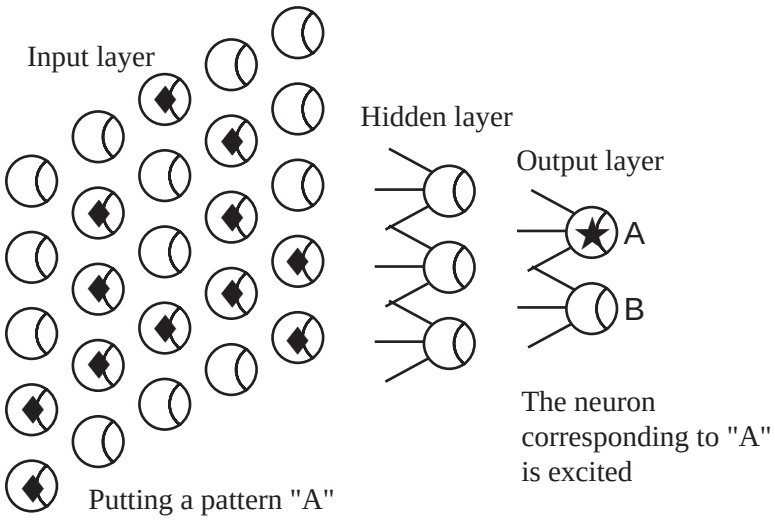
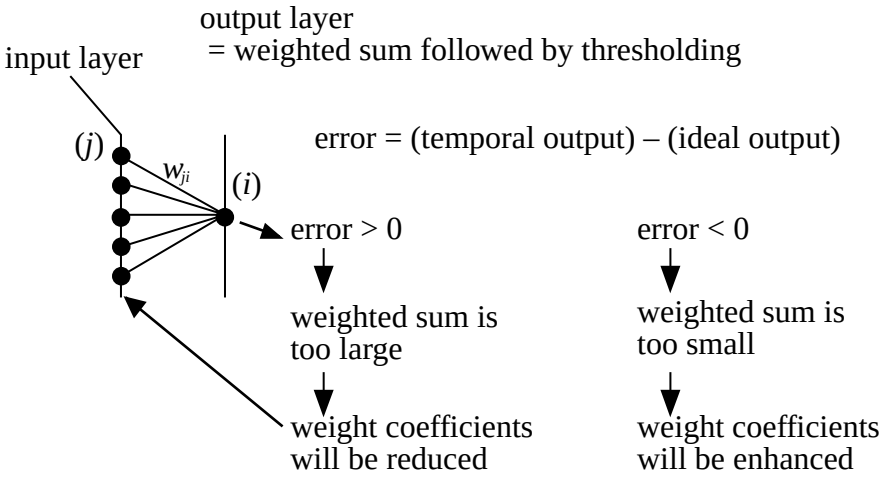


Fig. 2. pattern recognition by layered neural network.



How much reduced or enhanced?

$$w_{ji}(new) = w_{ji}(old) - \varepsilon \frac{([temporal]_i - [ideal]_i)}{\text{error}} [input]_j$$

Fig. 3. Schematic illustration of the concept of δ-rule.

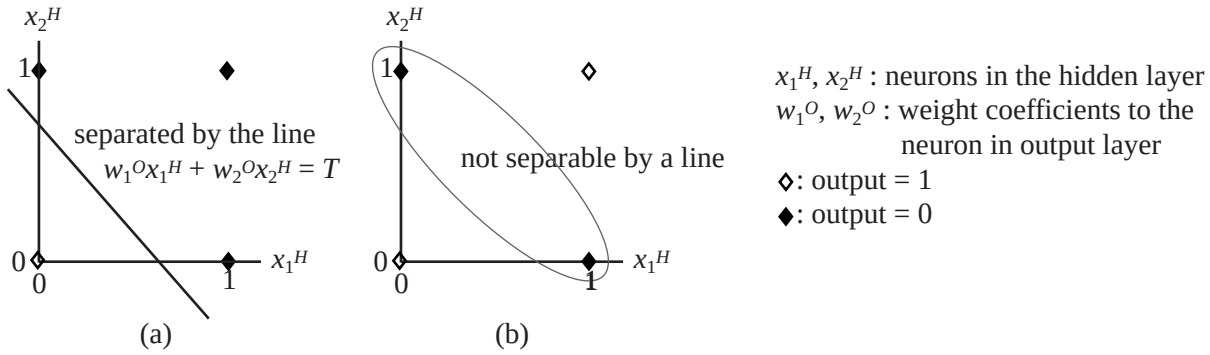


Fig. 4. Linearly unseparable function.

Error Back Propagation (EBP) algorithm

— Learning algorithm for multilayer neural network

— Errors in the one layer are shared to neurons in the preceding layers

— Criteria: Errors at one neuron in the output layer is shared to errors in the (n–1)th layer as:

[1] Sharing more errors to neurons whose connection weight to the output neuron is larger: since the error collection at the (n–1)th layer will be more effective at the output layer.

[2] the status of a neuron in the (n–1)th layer is determined by the value of a threshold-like function of the sum of stimuli from the (n–2)th layer. Sharing more errors to "sensitive" neurons, that is, ratio of the increment of status of neuron to that of the sum of stimuli is larger, since the error collection at the (n–1)th layer will be also more effective at the output layer.

— The learning method by the above criteria achieves the steepest decent of the error.

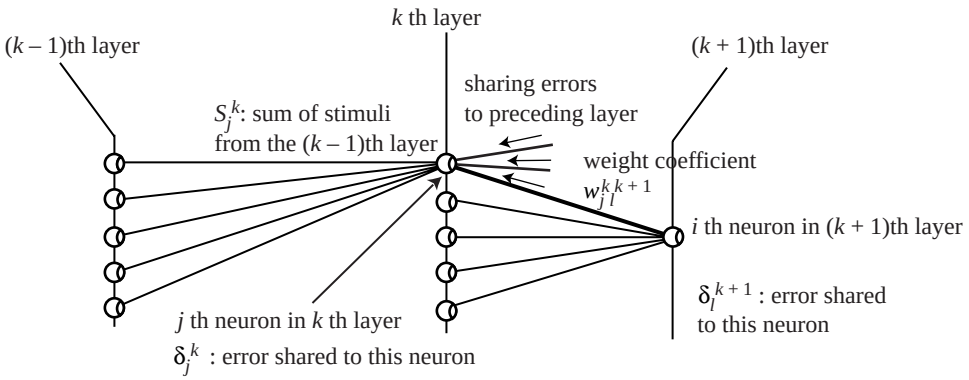


Fig. 5. Error back propagataion.

— Method:

Let δ_j^k be the error shared to the j th neuron in the k th layer, and δ_l^{k+1} be the error shared to the l th neuron in the (k+1)th layer. Let $w_{jl}^{k\,k+1}$ be the weight coefficient of the connection from the j th neuron in the k th layer to the l th neuron in the (k+1)th layer, and x_j^k be the status of the j th neuron in the k th layer. Using these values, "the sum of stimuli from the (k – 1)th layer to the j th neuron in the k th layer is as follows:

$$S_j^k = \sum_i w_{ij}^{k-1\,k} x_i^{k-1} \ . \tag{1}$$

The criterion [1] means that the error propagated from the l th neuron in the (k+1)th layer to the j th neuron in the k th layer is proportional to $w_{jl}^{k\,k+1}$. The criterion [2] means that this valuse is also proportional to dx_j^k / dS_j^k . Since such kind of error is propagated from each neuron in the (k+1)th layer, and its sum is δ_j^k ,

$$\delta_j^k = \sum_l \left(w_{jl}^{k\,k+1} \frac{dx_j^k}{dS_j^k} \right) \delta_l^{k+1} \ . \tag{2}$$

Let the threshold-like nonlinear function be f( ). Since

$$x_j^k = f(S_j^k) \tag{3}$$

we get

$$\frac{dx_j^k}{dS_j^k} = f'(S_j^k) \tag{4}$$

and Eq. (2) will be

$$\delta_j^k = \sum_l \left(w_{jl}^{k\,k+1} f'(S_j^k) \right) \delta_l^{k+1}$$
$$= \left[\sum_l (w_{jl}^{k\,k+1} \delta_l^{k+1}) \right] f'(S_j^k) \,. \tag{5}$$

By the error shared by each neuron in the k th layer, Equation (5) determines the error shared by the j th neuron in the k th layer. Since the error is the difference between the ideal output and the current output in case $k = n$, i.e. in the output layer, that is

$$\delta_l^n = (x_l^n - y_l) f'(S_l^n) \tag{6}$$

where y is the ideal output of the l th neuron in the output layer.

Using these errors on each neuron on each layer, the modified connection wetight coefficient $w'^{k-1\,k}_{jl}$ will be determined by d-rule as in Fig. 3, as follows:

$$w'^{k-1\,k}_{jl} = w^{k-1\,k}_{jl} - \varepsilon \delta_l^k x_j^{k-1} \,. \tag{7}$$

Appendix 1: Steepest descent method

Let $f(x, y, z)$ be a potential field and $(x(s), y(s), z(s))$ be a curve on xyz-coordinate. A tangent vector $\boldsymbol{b}$ is defined as

$$\boldsymbol{b} = \left(\frac{dx}{ds}, \frac{dy}{ds}, \frac{dz}{ds}\right) . \tag{8}$$

Let $D_{\boldsymbol{b}} f$ be the differentiation of f in the direction of $\boldsymbol{b}$. We get

$$\begin{aligned} D_{\boldsymbol{b}} f = \frac{\partial f}{\partial s} &= \frac{\partial f}{\partial x} \cdot \frac{dx}{ds} + \frac{\partial f}{\partial y} \cdot \frac{dy}{ds} + \frac{\partial f}{\partial z} \cdot \frac{dz}{ds} \\ &= \left(\frac{dx}{ds}, \frac{dy}{ds}, \frac{dz}{ds}\right) \cdot \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right) \\ &= \boldsymbol{b} \cdot \mathrm{grad}\ f \end{aligned} \tag{9}$$

We get from Eq. (9) that, if $\boldsymbol{b}$ is anti-parallel to grad f, the differential is negative and its absolute value is the maximum. This means that –grad f indicates the direction of the curve of steepest descent, or the curve where the decrement of f is maximum.

Appendix 2: Proof that EBP achieves the steepest descent

The squared sum of the error between the ideal output and the current output (status of a neuron in the n th layer) is defined as follows:

$$E = \frac{1}{2} \sum_l (x_l^n - y_l)^2 \,. \tag{A1}$$

In this case, E is a function of the weight coefficients.

The differential of E along the direction of $w_{jl}^{k-1\,k}$ is

$$\frac{\partial E}{\partial w_{jl}^{k-1\,k}} = \frac{\partial E}{\partial S_l^k}\frac{\partial S_l^k}{\partial w_{jl}^{k-1\,k}}\,, \tag{A2}$$

and from the definition of S_l^k in Eq. (1), we get

$$\begin{aligned}\frac{\partial S_l^k}{\partial w_{jl}^{k-1\,k}} &= \frac{\partial}{\partial w_{jl}^{k-1\,k}}\sum_i w_{il}^{k-1\,k}x_i^{k-1} \\ &= x_j^{k-1}\end{aligned}\,. \tag{A3}$$

Here we define

$$\delta_l^k = \frac{\partial E}{\partial S_l^k}\,, \tag{A4}$$

and insert Eqs. (A3)(A4) into Eq. (A2), then we get

$$\frac{\partial E}{\partial w_{jl}^{k-1\,k}} = \delta_l^k x_j^{k-1}\,. \tag{A5}$$

Comparing this to

$$w'^{k-1\,k}_{jl} = w_{jl}^{k-1\,k} - \varepsilon\delta_l^k x_j^{k-1}\,, \tag{7}$$

which defines how to modify the weight coefficients, Eq. (7) modifiies the weight coefficients along the direction of $\partial E / \partial w_{jl}^{k-1\,k}$, i. e. the direction of – grad E. Thus the modification by Eq. (7) achieves the steepest descent of E.

Now we prove that δ in Eq. (A4) is equivalent to what is derived from the criteria [1][2] in the body of this document. In case $k \neq n$,we get

$$\frac{\partial E}{\partial S_j^k} = \sum_l \frac{\partial E}{\partial S_l^{k+1}}\cdot\frac{\partial S_l^{k+1}}{\partial x_j^k}\cdot\frac{dx_j^k}{dS_j^k}\,. \tag{A6}$$

On the other hand,

$$\frac{dx_j^k}{dS_j^k} = f'(S_j^k)\,. \tag{4}$$

Since we get by replacing the suffixes of Eq. (1)

$$S_l^{k+1} = \sum_i w_{il}^{k\,k+1}x_i^k\,, \tag{A8}$$

$$\frac{\partial S_l^{k+1}}{\partial x_j^k} = w_{jl}^{k\,k+1}\,. \tag{A9}$$

Insertion of Eqs. (4)(A9) into Eq. (A6) yields

$$\frac{\partial E}{\partial S_j^k} = \sum_l \frac{\partial E}{\partial S_l^{k+1}}\cdot w_{jl}^{k\,k+1}\cdot f'(S_j^k)\,. \tag{A10}$$

From the definition of δ in Eq.(A4), we get

$$\delta_j^k = \frac{\partial E}{\partial S_j^k},\ \delta_l^{k+1} = \frac{\partial E}{\partial S_l^{k+1}}\ . \tag{A11}$$

Insertion of this into Eq. (A10) yields

$$\begin{aligned}\delta_j^k &= \sum_l \delta_l^{k+1} \cdot w_{jl}^{k\,k+1} \cdot f'(S_j^k) \\ &= \left[\sum_l (w_{jl}^{k\,k+1}\delta_l^{k+1})\right] f'(S_j^k)\ ,\end{aligned} \tag{A12}$$

i. e. the same as in the body of this document.

In case $k = n$,we get

$$\frac{\partial E}{\partial S_l^n} = \frac{\partial E}{\partial x_l^n} \cdot \frac{dx_l^n}{dS_l^n}\ . \tag{A13}$$

From Eq. (A1), we get

$$\begin{aligned}\frac{\partial E}{\partial x_l^n} &= \frac{\partial}{\partial x_l^n}\left\{\frac{1}{2}\sum_l (x_l^n - y_l)^2\right\} \\ &= x_l^n - y_l\end{aligned}\ . \tag{A14}$$

Insertion of this into Eq. (A13) yields

$$\frac{\partial E}{\partial S_l^n} = (x_l^n - y_l) \cdot \frac{dx_l^n}{dS_l^n}\ . \tag{A15}$$

From the definition of δ in Eq.(A4), we get

$$\delta_l^n = \frac{\partial E}{\partial S_l^n}\ , \tag{A16}$$

and from Eq. (10) we get

$$\frac{dx_l^n}{dS_l^n} = f'(S_l^n)\ , \tag{A17}$$

and insertion of Eqs. (A16)(A17) into (A15) yields

$$\delta_l^n = (x_l^n - y_l) f'(S_l^n)\ , \tag{A18}$$

i. e. the same as in the body of this document.