

Computer Science

Jerzy Świątek

Systems Modelling and Analysis

Choose yourself and new technologies

L.6. Choice of the best model



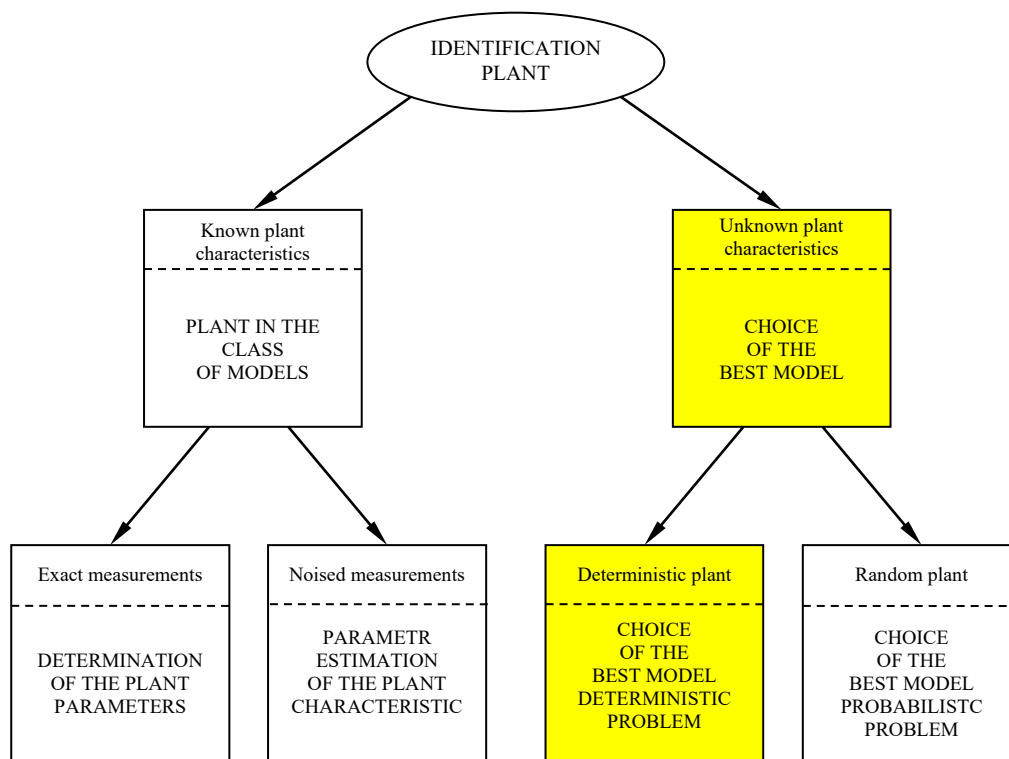
Wrocław University of Technology



Project co-financed from the EU European Social Fund



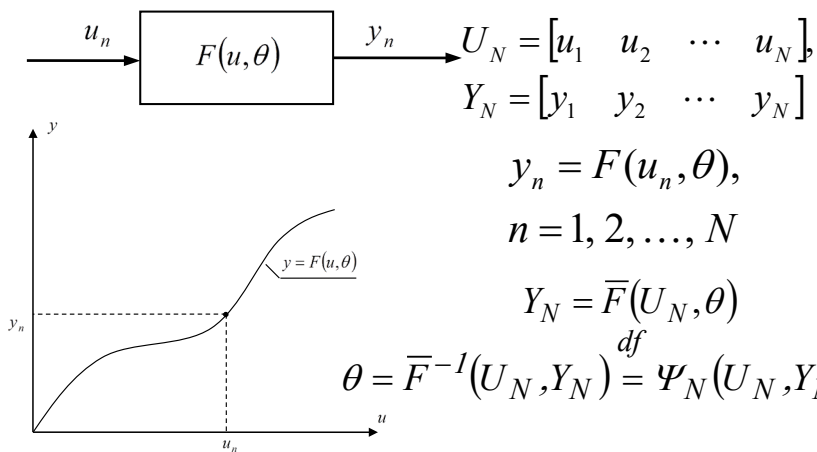
Typical identification tasks



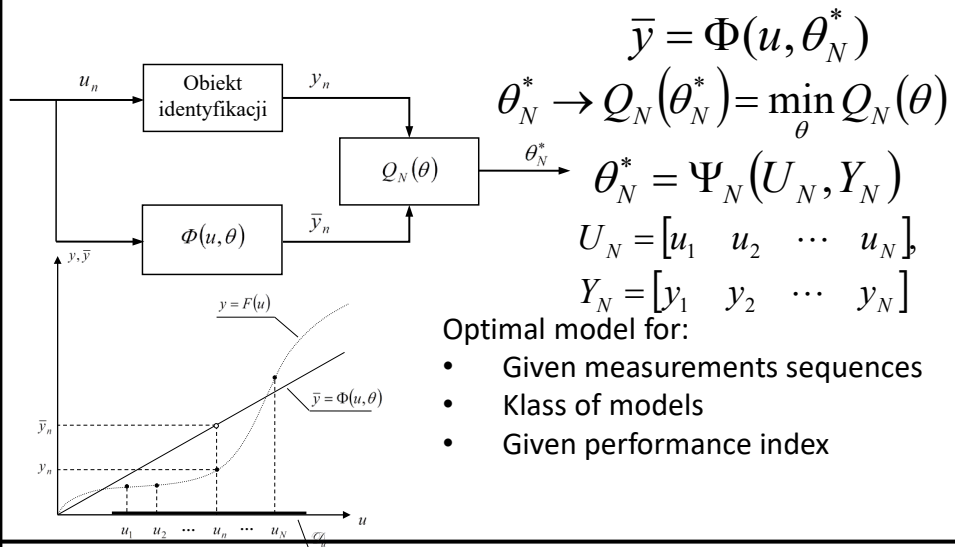


Plant in the class of model

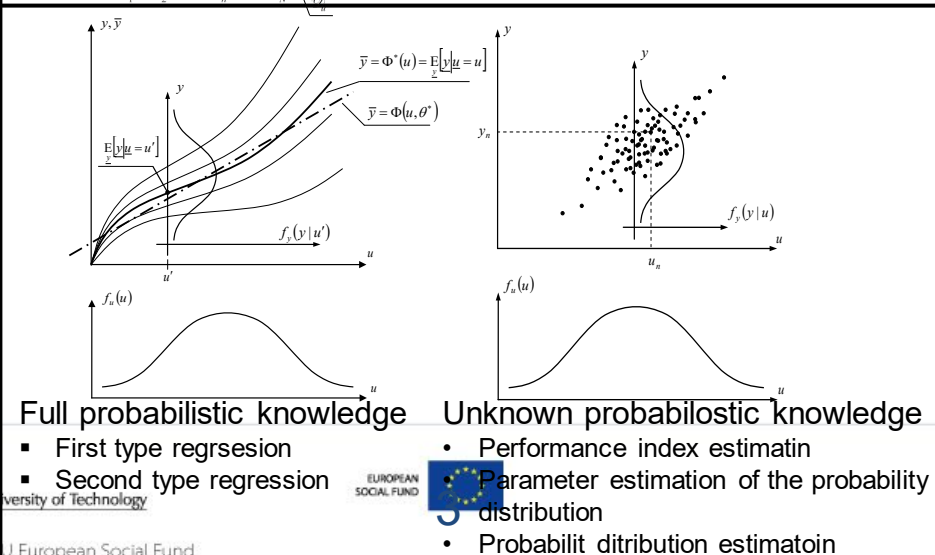
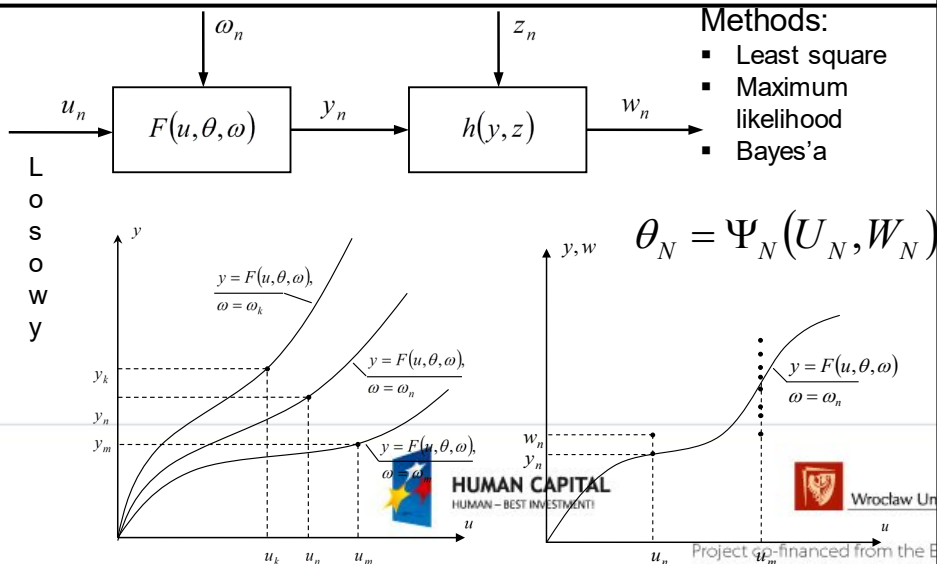
Deterministyczny



Choice of the best model



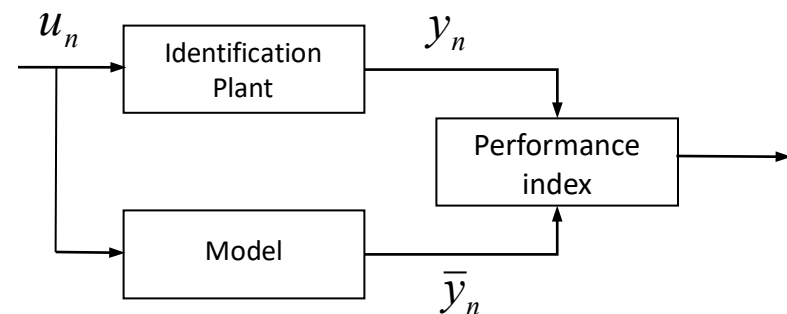
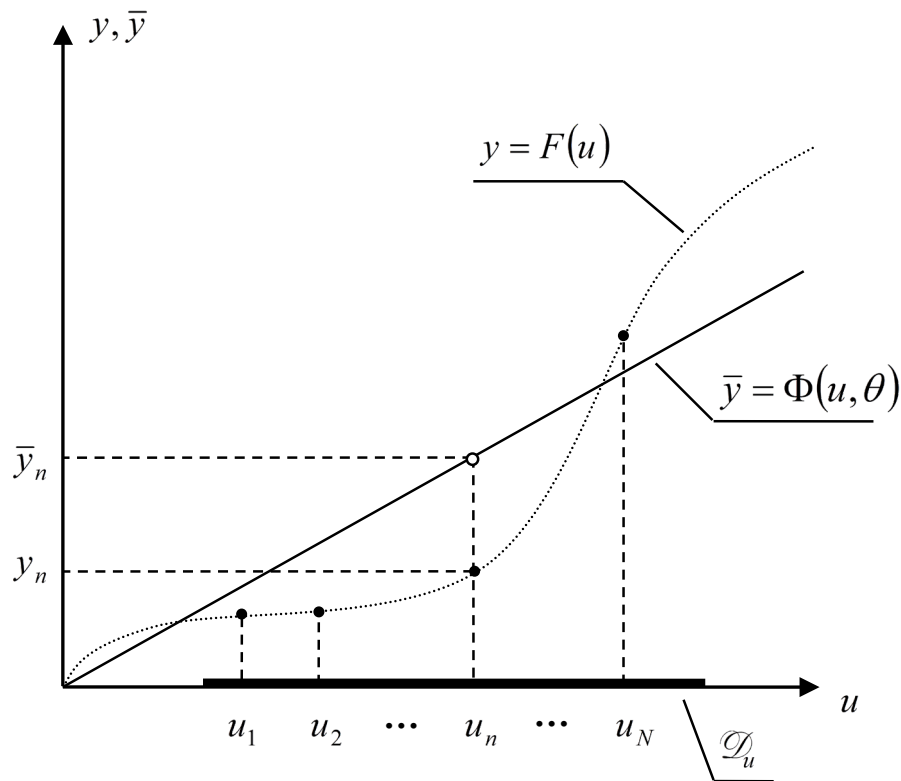
Losowy





Choice of the best model

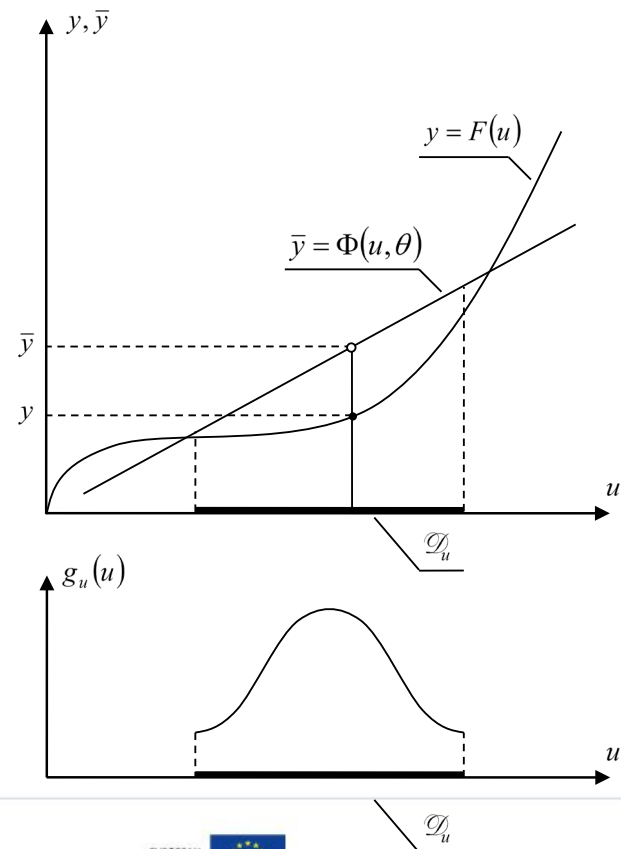
Deterministic problem





Approximation of static plant characteristic

- Problem formulation



Weight function: $g_u(u)$





Approximation of static plant characteristic

- Problem formulation

Static plant characteristic: $y = F(u)$

F – known function

u – input vector $u \in \mathcal{U} \subseteq \mathbb{R}^S$ $\mathcal{U}$ – input domain

y – output vector $y \in \mathcal{Y} \subseteq \mathbb{R}^L$ $\mathcal{Y}$ – output domain

Approximation function (model): $\bar{y} = \Phi(u, \theta)$

Φ – arbitrary given function

$\bar{y}$ – model output vector $\bar{y} \in \overline{\mathcal{Y}} \subseteq \mathbb{R}^L$ $\mathcal{D}_u$ – subset of input domain

θ – vector of model parameters $\theta \in \Theta \subseteq \mathbb{R}^R$ $\mathcal{Y}$ – output domain



Approximation of static plant characteristic

- Problem formulation

Measure of difference: $\forall u \in \mathcal{D}_u \quad q(y, \bar{y}) = q(F(u), \Phi(u, \theta))$

where e. g.: $q(y, \bar{y}) = \sum_{l=1}^L (y^{(l)} - \bar{y}^{(l)})^2$ $q(y, \bar{y}) = \sum_{l=1}^L |y^{(l)} - \bar{y}^{(l)}|$ L – dimensional output

$$q(y, \bar{y}) = (y - \bar{y})^2 = (F(u) - \Phi(u, \theta))^2$$

one dimensional output

$$q(y, \bar{y}) = |y - \bar{y}| = |F(u) - \Phi(u, \theta)|$$

($L = 1$)



Approximation of static plant characteristic

- Problem formulation

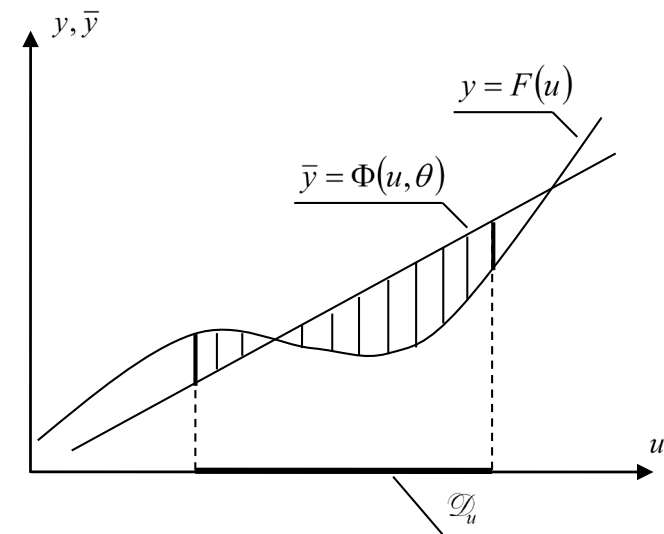
Performance index – a measure of the difference between function $F(u)$ and $\Phi(u, \theta)$

$$Q(\theta) = \|F(u) - \Phi(u, \theta)\|_{\mathcal{Q}_u}$$

$$Q(\theta) = \int_{\mathcal{Q}_u} q(F(u), \Phi(u, \theta)) g_u(u) du$$

e. g. :

$$Q(\theta) = \int_{\mathcal{Q}_u} |F(u) - \Phi(u, \theta)| g_u(u) du$$





Approximation of static plant characteristic

- Problem formulation

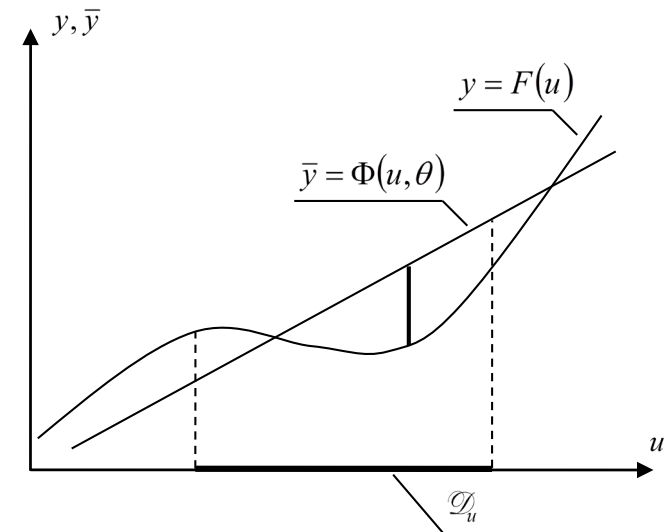
Performance index – a measure of the difference between function $F(u)$ and $\Phi(u, \theta)$

$$Q(\theta) = \|F(u) - \Phi(u, \theta)\|_{\mathcal{Q}_u}$$

$$Q(\theta) = \max_{u \in \mathcal{Q}_u} \{q(F(u), \Phi(u, \theta)) g_u(u)\}$$

e. g. :

$$Q(\theta) = \max_{u \in \mathcal{Q}_u} \{|F(u) - \Phi(u, \theta)| g_u(u)\}$$



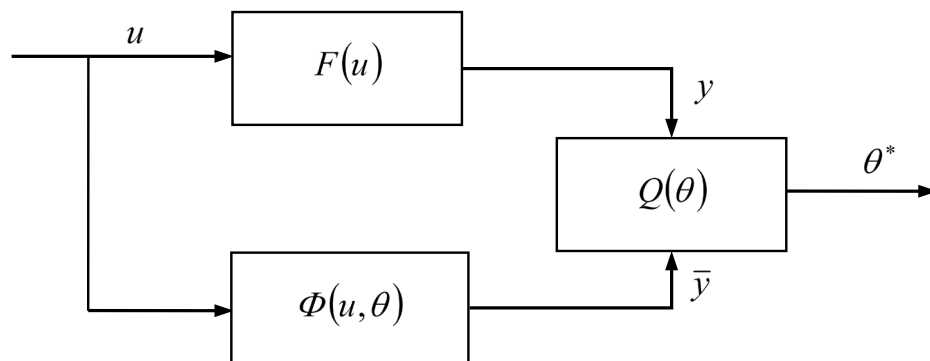


Approximation of static plant characteristic

- Problem formulation

Optimal model: $\bar{y} = \Phi(u, \theta^*)$

where θ^* – optimal model parameters: $\theta^* \rightarrow Q(\theta^*) = \min_{\theta \in \Theta} Q(\theta)$





Linear case – problem formulation

Model: $\bar{y} = \Phi(u, \theta) = \theta^T \varphi(u)$

where

$$\theta \stackrel{df}{=} \begin{bmatrix} \theta^{(1)} \\ \theta^{(2)} \\ \vdots \\ \theta^{(R)} \end{bmatrix} \quad \varphi(u) \stackrel{df}{=} \begin{bmatrix} \varphi_1(u) \\ \varphi_2(u) \\ \vdots \\ \varphi_R(u) \end{bmatrix} \quad \text{e. g. : } \varphi(u) = \begin{bmatrix} 1 \\ u \\ \vdots \\ u^{R-1} \end{bmatrix}$$

Performance index:

$$Q(\theta) = \int_{\mathcal{D}_u} (F(u) - \theta^T \varphi(u))^2 g_u(u) du$$



Linear case - solution

Optimality condition for the vector θ :

$$\text{grad}_{\theta} Q(\theta) = -2 \left[\int_{\mathcal{D}_u} (F(u) - \theta^{*T} \varphi(u)) \varphi(u) g_u(u) du \right] = 0_R,$$

$$0_R = \left\{ \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\} R$$

The solution is:

$$\theta^* = \left[\int_{\mathcal{D}_u} \varphi(u) \varphi^T(u) g_u(u) du \right]^{-1} \int_{\mathcal{D}_u} F(u) \varphi(u) g_u(u) du$$





Linear case - solution

Note that:

$$\int_{\mathcal{U}} \varphi(u) \varphi^T(u) g_u(u) du =$$

$$\begin{bmatrix} \int_{\mathcal{U}} \varphi_1(u) \varphi_1(u) g_u(u) du & \int_{\mathcal{U}} \varphi_1(u) \varphi_2(u) g_u(u) du & \cdots & \int_{\mathcal{U}} \varphi_1(u) \varphi_R(u) g_u(u) du \\ \int_{\mathcal{U}} \varphi_2(u) \varphi_1(u) g_u(u) du & \int_{\mathcal{U}} \varphi_2(u) \varphi_2(u) g_u(u) du & \cdots & \int_{\mathcal{U}} \varphi_2(u) \varphi_R(u) g_u(u) du \\ \vdots & \vdots & \ddots & \vdots \\ \int_{\mathcal{U}} \varphi_R(u) \varphi_1(u) g_u(u) du & \int_{\mathcal{U}} \varphi_R(u) \varphi_2(u) g_u(u) du & \cdots & \int_{\mathcal{U}} \varphi_R(u) \varphi_R(u) g_u(u) du \end{bmatrix}$$



Linear case - solution

Let us choose $\varphi_i(u)$ such that:

$$\int_{\mathcal{U}} \varphi_i(u) \varphi_j(u) g_u(u) du = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}, \quad i, j = 1, 2, \dots, R$$

Then:
$$\int_{\mathcal{U}} \varphi(u) \varphi^T(u) g_u(u) du = \mathbf{I}$$

and approximation algorithm has the form:

$$\theta^* = \int_{\mathcal{U}} F(u) \varphi(u) g_u(u) du$$



Linear case - example

Plant characteristic: $y = F(u) = u^2$

Model: $\bar{y} = \Phi(u, \theta) = \theta^T \varphi(u) = \theta^{(1)} + \theta^{(2)}u$ where: $\varphi(u) = \begin{bmatrix} 1 \\ u \end{bmatrix}$, $\theta = \begin{bmatrix} \theta^{(1)} \\ \theta^{(2)} \end{bmatrix}$

Interval: $\mathcal{D}_u = \{u \in \mathcal{R} : 0 \leq u \leq 1\}$

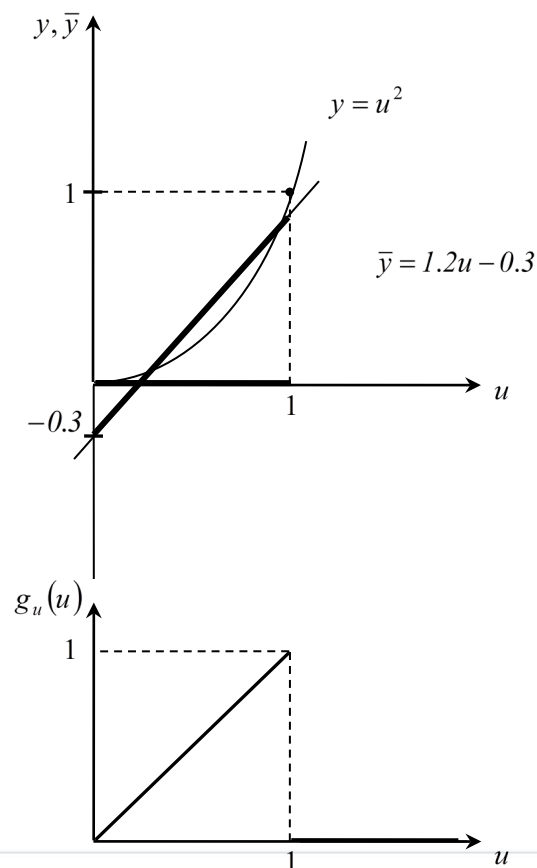
Weight function: $g_u(u) = \begin{cases} u & \text{for } u \in [0, 1] \\ 0 & \text{for } u \notin [0, 1] \end{cases}$

Performance index: $Q(\theta) = \int_{\mathcal{D}_u} (F(u) - \theta^T \varphi(u))^2 g_u(u) du$



Linear case - example

Graphical interpretation:





Linear case - example

Performance index has the form:

$$Q(\theta) = \int_{\mathcal{D}_u} (F(u) - \theta^T \varphi(u))^2 g_u(u) du = \int_0^1 \left(u^2 - [\theta^{(1)} \ \theta^{(2)}] \begin{bmatrix} 1 \\ u \end{bmatrix} \right)^2 u du$$

Optimality condition has the form:

$$\text{grad}_{\theta} Q(\theta) = -2 \left[\int_0^1 \left(u^2 - [\theta^{(1)*} \ \theta^{(2)*}] \begin{bmatrix} 1 \\ u \end{bmatrix} \right) \begin{bmatrix} 1 \\ u \end{bmatrix} u du \right] = 0_2$$

Solution has the form:

$$\theta^* = \left[\int_0^1 \begin{bmatrix} 1 \\ u \end{bmatrix} [1 \ u] u du \right]^{-1} \int_0^1 u^2 \begin{bmatrix} 1 \\ u \end{bmatrix} u du = \begin{bmatrix} \int_0^1 u du & \int_0^1 u^2 du \\ \int_0^1 u^2 du & \int_0^1 u^3 du \end{bmatrix}^{-1} \begin{bmatrix} \int_0^1 u^3 du \\ \int_0^1 u^4 du \end{bmatrix}$$





Linear case - example

Consequently:

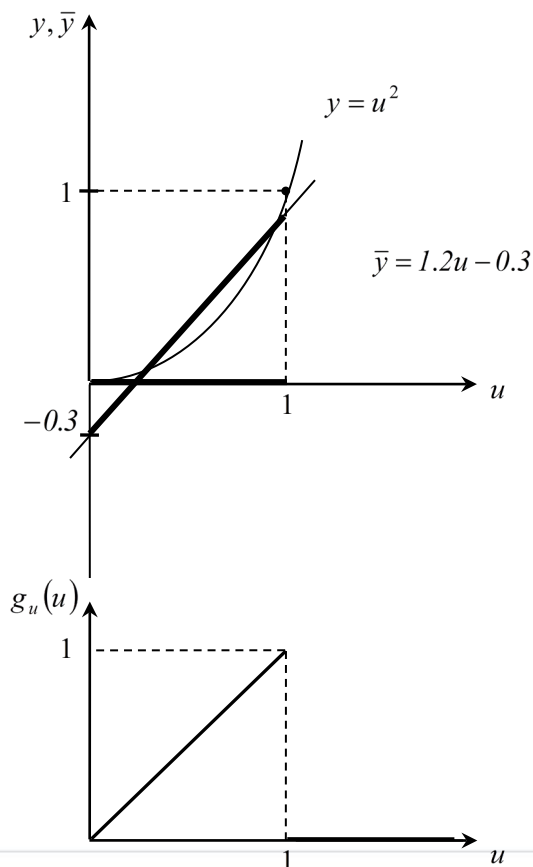
$$\theta^* = \begin{bmatrix} \theta^{(1)*} \\ \theta^{(2)*} \end{bmatrix} = \begin{bmatrix} \frac{\int_0^1 u^3 du \times \int_0^1 u^3 du - \int_0^1 u^4 du \times \int_0^1 u^2 du}{\int_0^1 u du \times \int_0^1 u^3 du - \left(\int_0^1 u^2 du\right)^2} \\ \frac{\int_0^1 u^4 du \times \int_0^1 u du - \int_0^1 u^3 du \times \int_0^1 u^2 du}{\int_0^1 u du \times \int_0^1 u^3 du - \left(\int_0^1 u^2 du\right)^2} \end{bmatrix} = \begin{bmatrix} \frac{\frac{1}{4}u^4 \Big|_0^1 \times \frac{1}{4}u^4 \Big|_0^1 - \frac{1}{5}u^5 \Big|_0^1 \times \frac{1}{3}u^3 \Big|_0^1}{\frac{1}{2}u^2 \Big|_0^1 \times \frac{1}{4}u^4 \Big|_0^1 - \left(\frac{1}{3}u^3 \Big|_0^1\right)^2} \\ \frac{\frac{1}{5}u^5 \Big|_0^1 \times \frac{1}{2}u^2 \Big|_0^1 - \frac{1}{4}u^4 \Big|_0^1 \times \frac{1}{3}u^3 \Big|_0^1}{\frac{1}{2}u^2 \Big|_0^1 \times \frac{1}{4}u^4 \Big|_0^1 - \left(\frac{1}{3}u^3 \Big|_0^1\right)^2} \end{bmatrix} = \begin{bmatrix} \frac{\frac{1}{4} \times \frac{1}{4} - \frac{1}{5} \times \frac{1}{3}}{\frac{1}{2} \times \frac{1}{4} - \left(\frac{1}{3}\right)^2} \\ \frac{\frac{1}{5} \times \frac{1}{2} - \frac{1}{4} \times \frac{1}{3}}{\frac{1}{2} \times \frac{1}{4} - \left(\frac{1}{3}\right)^2} \end{bmatrix} = \begin{bmatrix} -0.3 \\ 1.2 \end{bmatrix}$$



Linear case - example

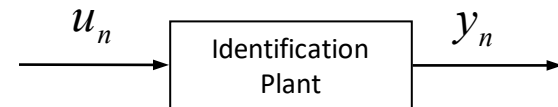
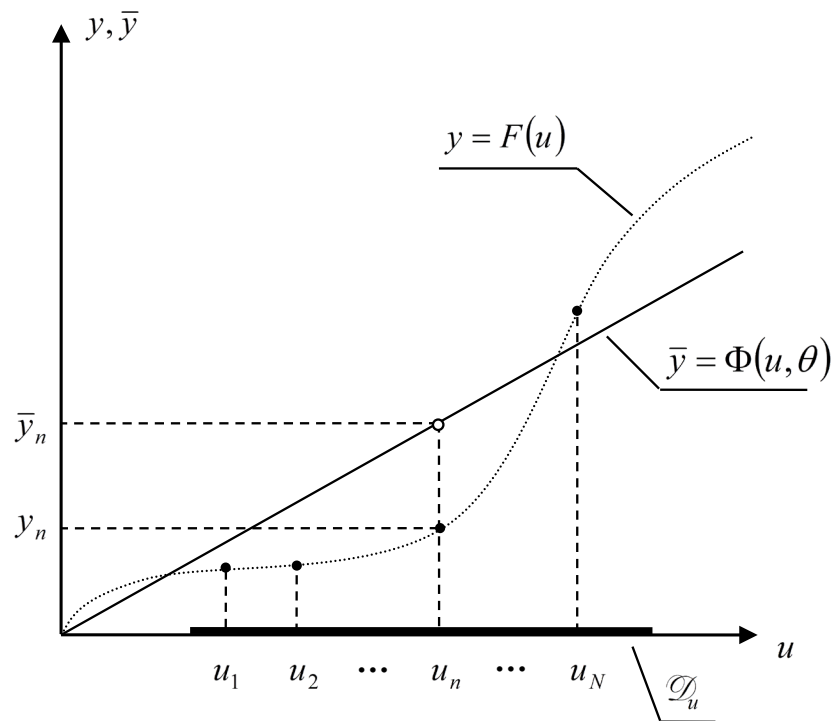
Optimal model: $\bar{y} = 1.2u - 0.3$

Graphical interpretation:





Choice of the best model based on the noise free measurements





Choice of the best model based on the noise free measurements

- Problem formulation

Experiment: $U_N = [u_1 \ u_2 \ \cdots \ u_N]$, $Y_N = [y_1 \ y_2 \ \cdots \ y_N]$

Approximation function (model): $\bar{y} = \Phi(u, \theta)$

Measure of the difference: $\forall n = 1, 2, \dots, N \quad q(y_n, \bar{y}_n) = q(y_n, \Phi(u_n, \theta))$



Choice of the best model based on the noise free measurements

- Problem formulation

Performance index: $Q_N(\theta) = \|Y_N - \bar{Y}_N(\theta)\|_{U_N}$

where: $\bar{Y}_N(\theta) \stackrel{df}{=} [\Phi(u_1, \theta) \quad \Phi(u_2, \theta) \quad \dots \quad \Phi(u_N, \theta)]$

$$Q_N(\theta) = \sum_{n=1}^N \alpha_n q(y_n, \bar{y}_n) = \sum_{n=1}^N \alpha_n q(y_n, \Phi(u_n, \theta)) \quad \text{e. g. :} \quad Q_N(\theta) = \sum_{n=1}^N |y_n - \bar{y}_n| = \sum_{n=1}^N |y_n - \Phi(u_n, \theta)|$$

$$Q_N(\theta) = \max_{1 \leq n \leq N} \{q(y_n, \bar{y}_n)\} = \max_{1 \leq n \leq N} \{q(y_n, \Phi(u_n, \theta))\} \quad \text{e. g. :} \quad Q_N(\theta) = \max_{1 \leq n \leq N} \{|y_n - \bar{y}_n|\} = \max_{1 \leq n \leq N} \{|y_n - \Phi(u_n, \theta)|\}$$

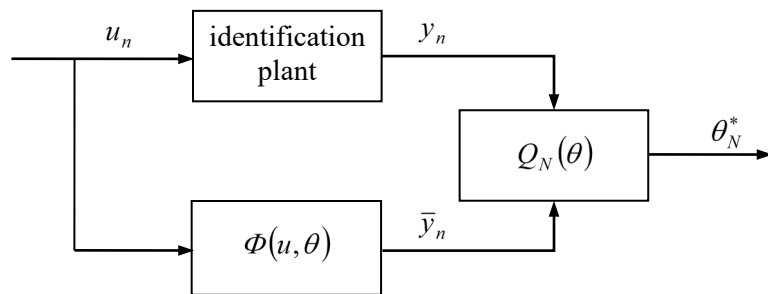


Choice of the best model based on the noise free measurements

- Problem formulation

Optimal model: $\bar{y} = \Phi(u, \theta_N^*)$

$$\theta_N^* \rightarrow Q_N(\theta_N^*) = \min_{\theta} Q_N(\theta)$$



The model is optimal for:

- given measurement sequence
- proposed model
- performance index



Approximation task

$$y = F(u) \quad \bar{y} = \Phi(u, \theta^*)$$

where: θ^* – optimal vector parameter:

$$\theta^* \rightarrow Q(\theta^*) = \min_{\theta \in \Theta} Q(\theta)$$

$$Q(\theta) = \int_{\mathcal{Q}_u} q(F(u), \Phi(u, \theta)) g_u(u) du$$

Identification task

$$U_N = [u_1 \quad u_2 \quad \cdots \quad u_N], \quad Y_N = [y_1 \quad y_2 \quad \cdots \quad y_N] \quad \bar{y} = \Phi(u, \theta_N^*)$$

where: θ_N^* – optimal vector parameter:

$$\theta_N^* \rightarrow Q_N(\theta_N^*) = \min_{\theta} Q_N(\theta)$$

$$Q_N(\theta) = \sum_{n=1}^N \alpha_n q(y_n, \bar{y}_n) = \sum_{n=1}^N \alpha_n q(y_n, \Phi(u_n, \theta))$$

$$\theta_N^* \sim \theta^*$$

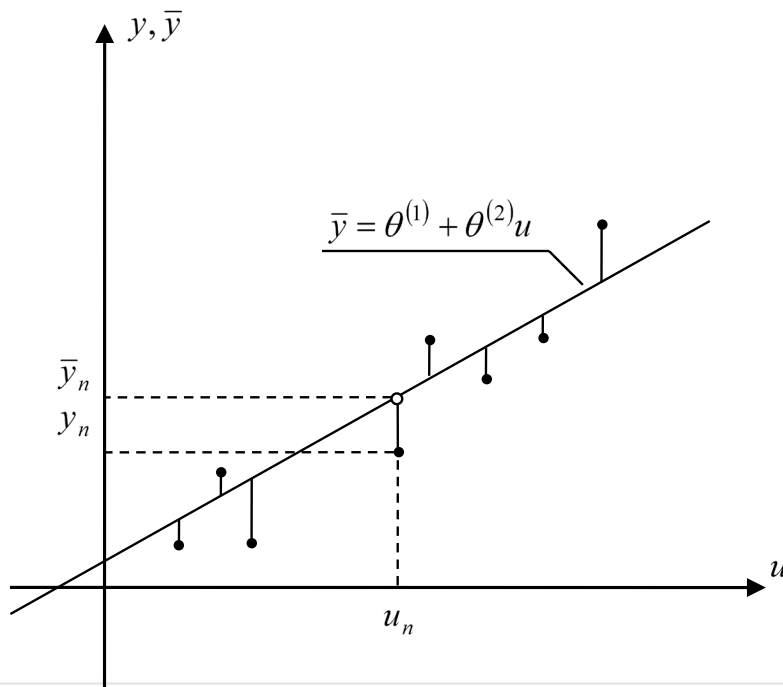
$$\theta_N^* \xrightarrow{U_N, \alpha_n \sim g_u(u)} \theta^*$$





Linear case – example

Experiment: $U_N = [u_1 \quad u_2 \quad \cdots \quad u_N]$, $Y_N = [y_1 \quad y_2 \quad \cdots \quad y_N]$



Model: $\bar{y} = \Phi(u, \theta) = \theta^T \varphi(u) = \theta^{(1)} + \theta^{(2)}u$

where $\varphi(u) = \begin{bmatrix} 1 \\ u \end{bmatrix}$, $\theta = \begin{bmatrix} \theta^{(1)} \\ \theta^{(2)} \end{bmatrix}$

Performance index:

$$Q_N(\theta) = \sum_{n=1}^N (y_n - \bar{y}_n)^2 = \sum_{n=1}^N (y_n - \theta^T \varphi(u_n))^2$$



Linear case – example

Optimality condition has the form: $\text{grad}_{\theta} Q_N(\theta) = -2 \sum_{n=1}^N (y_n - \theta_N^{*T} \varphi(u_n)) \varphi(u_n) = 0_2$

Solution has the form: $\theta_N^* = \left[\sum_{n=1}^N \varphi(u_n) \varphi^T(u_n) \right]^{-1} \times \sum_{n=1}^N y_n \varphi(u_n)$

$$\theta_N^* = \begin{bmatrix} \theta_N^{*(1)} \\ \theta_N^{*(2)} \end{bmatrix} = \begin{bmatrix} \frac{\left(\sum_{n=1}^N u_n^2 \right) \times \left(\sum_{n=1}^N y_n \right) - \left(\sum_{n=1}^N u_n \right) \times \left(\sum_{n=1}^N y_n u_n \right)}{N \sum_{n=1}^N u_n^2 - \left(\sum_{n=1}^N u_n \right) \times \left(\sum_{n=1}^N y_n u_n \right)} \\ \frac{N \sum_{n=1}^N y_n u_n - \left(\sum_{n=1}^N u_n \right) \times \left(\sum_{n=1}^N y_n \right)}{N \sum_{n=1}^N u_n^2 - \left(\sum_{n=1}^N u_n \right) \times \left(\sum_{n=1}^N y_n u_n \right)} \end{bmatrix}$$





Linear case - recursive algorithm

$$Q_N(\theta) = \sum_{n=1}^N (y_n - \bar{y}_n)^2 = \sum_{n=1}^N (y_n - \theta^T \varphi(u_n))^2$$

For N measurements $U_N = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \end{bmatrix}$, $Y_N = \begin{bmatrix} y_1 & y_2 & \cdots & y_N \end{bmatrix}$

$$\theta_N^* = \left[\sum_{n=1}^N \varphi(u_n) \varphi^T(u_n) \right]^{-1} \times \sum_{n=1}^N y_n \varphi(u_n)$$

New $N+1$ measurement point u_{N+1}, y_{N+1}

How to adopt vector of parameters using new measurement?

$$\theta_{N+1}^* = A(\theta_N^*, u_{N+1}, y_{N+1})$$

$$Q_{N+1}(\theta) = \sum_{n=1}^{N+1} (y_n - \bar{y}_n)^2 = \sum_{n=1}^{N+1} (y_n - \theta^T \varphi(u_n))^2$$

$$\theta_{N+1}^* = \left[\sum_{n=1}^{N+1} \varphi(u_n) \varphi^T(u_n) \right]^{-1} \times \sum_{n=1}^{N+1} y_n \varphi(u_n) =$$

$$\left[\sum_{n=1}^N \varphi(u_n) \varphi^T(u_n) + \varphi(u_{N+1}) \varphi^T(u_{N+1}) \right]^{-1} \times \left[\sum_{n=1}^N y_n \varphi(u_n) + y_{N+1} \varphi(u_{N+1}) \right]$$



Linear case - recursive algorithm

$$(\mathbf{A} + \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} + \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1}$$

$$\mathbf{B} = \varphi \quad - \text{wektor kolumnowy}$$

$$\mathbf{D}^{-1} = 1$$

$$\mathbf{C} = \varphi^T$$

$$(\mathbf{A} + \varphi\varphi^T)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\varphi(1 + \varphi^T\mathbf{A}^{-1}\varphi)^{-1}\varphi^T\mathbf{A}^{-1}$$

$$\mathbf{D}^{-1} = -1$$

$$(\mathbf{A} - \varphi\varphi^T)^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\varphi(1 - \varphi^T\mathbf{A}^{-1}\varphi)^{-1}\varphi^T\mathbf{A}^{-1}$$





Linear case - recursive algorithm

Let: $\varphi_n = \varphi(u_n), n = 1, 2, \dots, N + 1$

$$\begin{aligned} \left(\sum_{n=1}^{N+1} \varphi_n \varphi_n^T \right)^{-1} &= \left(\sum_{n=1}^N \varphi_n \varphi_n^T + \varphi_{N+1} \varphi_{N+1}^T \right)^{-1} = \\ &= \left(\sum_{n=1}^N \varphi_n \varphi_n^T \right)^{-1} - \left(\sum_{n=1}^N \varphi_n \varphi_n^T \right)^{-1} \frac{\varphi_{N+1} \varphi_{N+1}^T}{1 + \varphi_{N+1}^T \left(\sum_{n=1}^N \varphi_n \varphi_n^T \right)^{-1} \varphi_{N+1}} \left(\sum_{n=1}^N \varphi_n \varphi_n^T \right)^{-1} \end{aligned}$$

Let: $P_N = \left(\sum_{n=1}^N \varphi_n \varphi_n^T \right)^{-1}$

$$P_{N+1} = P_N - P_N \frac{\varphi_{N+1} \varphi_{N+1}^T}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} P_N$$





Linear case - recursive algorithm

$$\begin{aligned}
 \theta_{N+1}^* &= \left(P_N - P_N \frac{\varphi_{N+1} \varphi_{N+1}^T}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} P_N \right) \left(\sum_{n=1}^N \varphi_n y_n + \varphi_{N+1} y_{N+1} \right) = \\
 &= P_N \sum_{n=1}^N \varphi_n y_n + P_N \varphi_{N+1} y_{N+1} - \frac{P_N \varphi_{N+1} \varphi_{N+1}^T P_N}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} \sum_{n=1}^N \varphi_n y_n - \frac{P_N \varphi_{N+1} \varphi_{N+1}^T P_N \varphi_{N+1} y_{N+1}}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} = \\
 &= P_N \sum_{n=1}^N \varphi_n y_n + \frac{P_N \varphi_{N+1} y_{N+1} + P_N \varphi_{N+1} y_{N+1} \varphi_{N+1}^T P_N \varphi_{N+1} - P_N \varphi_{N+1} y_{N+1} \varphi_{N+1}^T P_N \varphi_{N+1} - P_N \varphi_{N+1} \varphi_{N+1}^T P_N \sum_{n=1}^N \varphi_n y_n}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} = \\
 &= P_N \sum_{n=1}^N \varphi_n y_n + \frac{P_N \varphi_{N+1} y_{N+1} - P_N \varphi_{N+1} \varphi_{N+1}^T P_N \sum_{n=1}^N \varphi_n y_n}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} y = \\
 &= P_N \sum_{n=1}^N \varphi_n y_n + \frac{P_N \varphi_{N+1} y_{N+1}}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} \left(y_{N+1} + \varphi_{N+1}^T P_N \sum_{n=1}^N \varphi_n y_n \right) \\
 &= \theta_N^* + K_{N+1} (y_{N+1} - \varphi_{N+1}^T \theta_N)
 \end{aligned}$$

$$K_{N+1} = \frac{P_N \varphi_{N+1}}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}}$$

where:





Linear case - recursive algorithm

Finally:

$$\theta_N^* = \left(\sum_{n=1}^N \varphi_n \varphi_n^T \right)^{-1} \sum_{n=1}^N \varphi_n y_n = P_N \sum_{n=1}^N \varphi_n y_n, \text{ gdzie: } P_N = \left(\sum_{n=1}^N \varphi_n \varphi_n^T \right)^{-1}$$

$$\theta_{N+1}^* = A(\theta_N^*, u_{N+1}, y_{N+1}) = \theta_N^* + K_{N+1} [y_{N+1} - \varphi(u_{N+1})^T \theta_N^*]$$

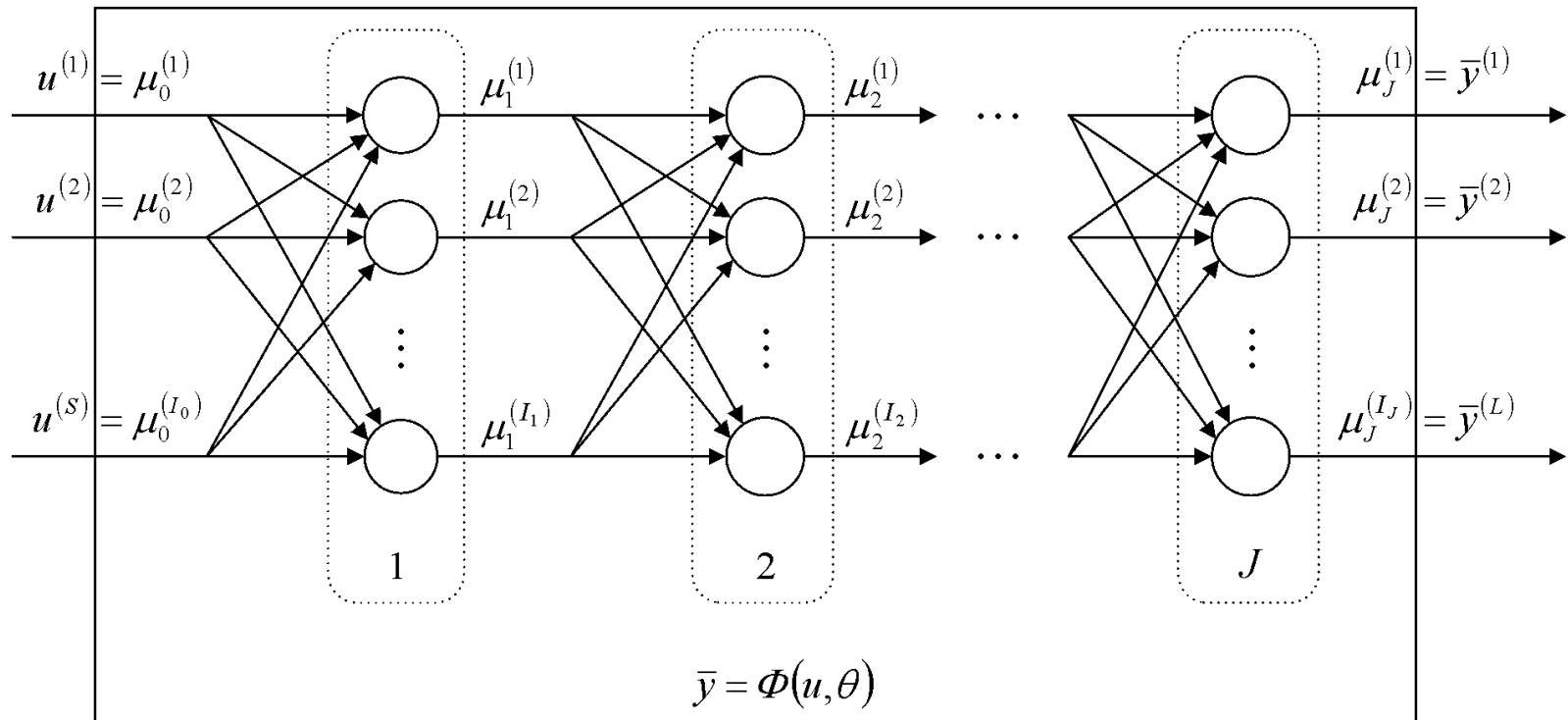
$$K_{N+1} = \frac{P_N \varphi(u_{N+1})}{1 + \varphi(u_{N+1})^T P_N \varphi(u_{N+1})}$$

$$P_{N+1} = P_N - \frac{P_N \varphi(u_{N+1}) \varphi(u_{N+1})^T P_N}{1 + \varphi(u_{N+1})^T P_N \varphi(u_{N+1})}$$



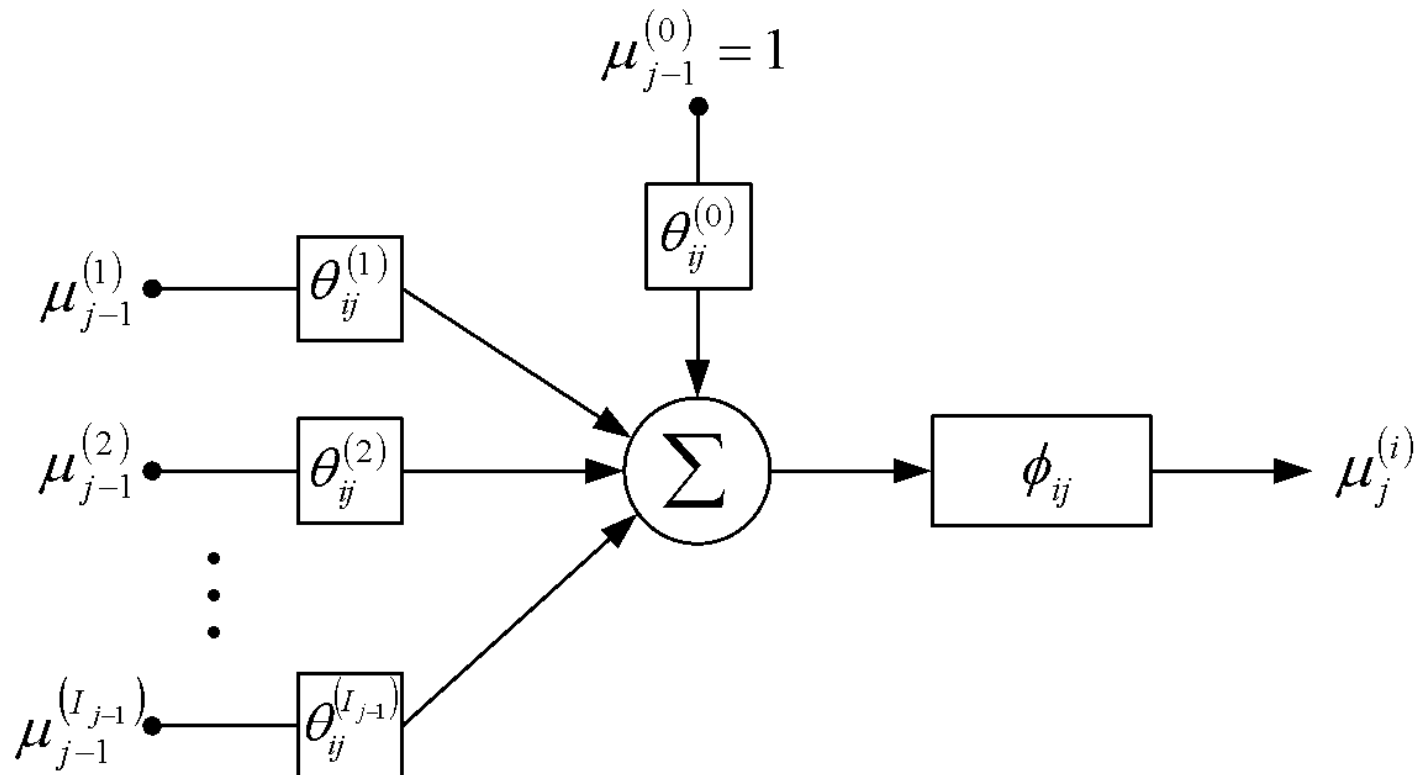


Neural networks



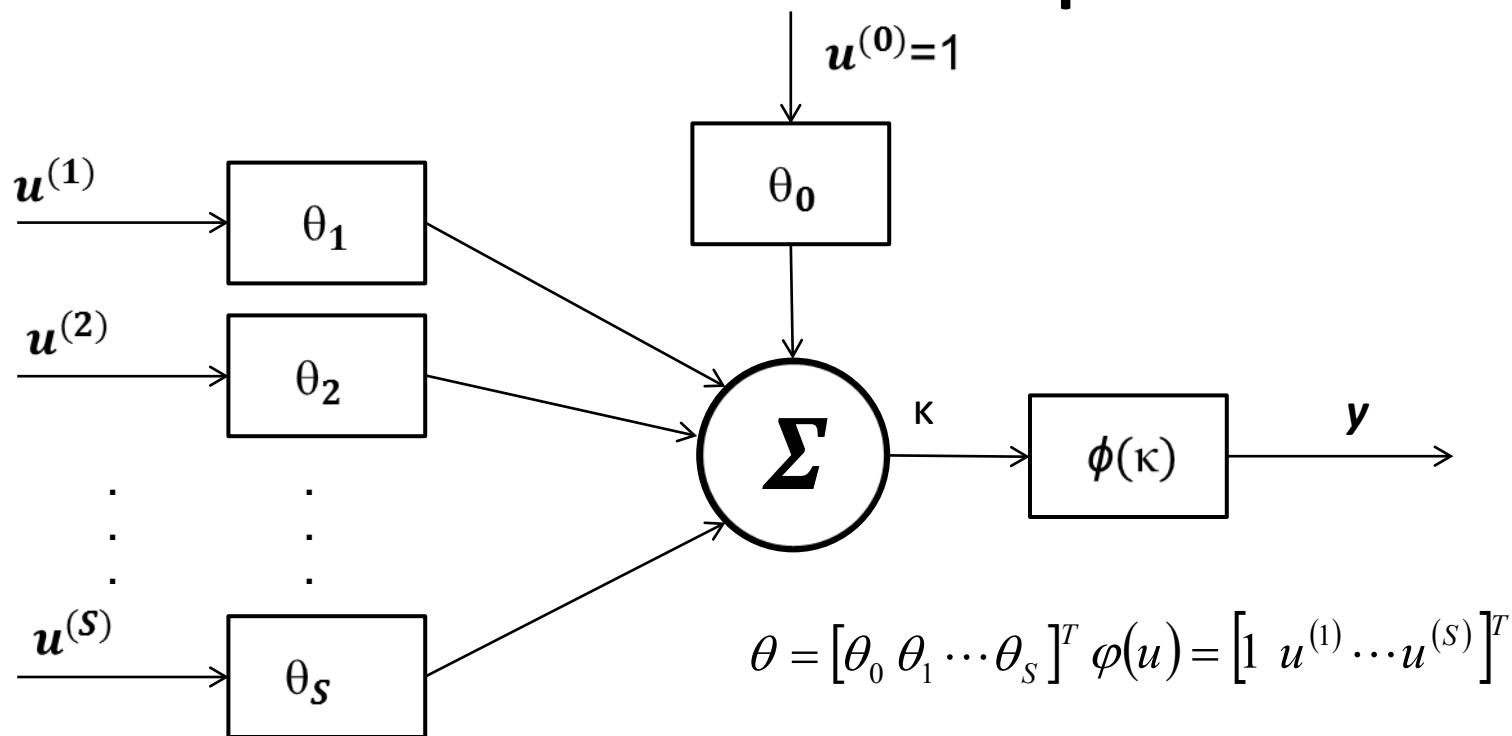


Neuron model





Neuron model simplification



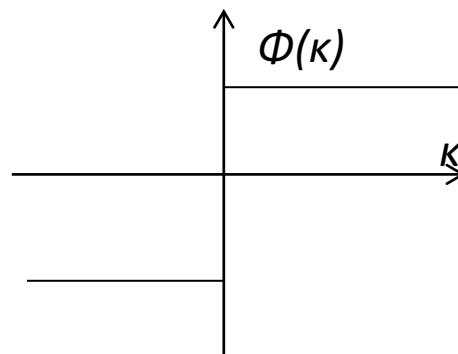
$$y = \phi\left(\sum_{s=1}^S \theta_s u^{(s)} + \theta_0\right) = \phi(\theta^T \varphi(u)) \quad \Phi - \text{activation function}$$



Activation function

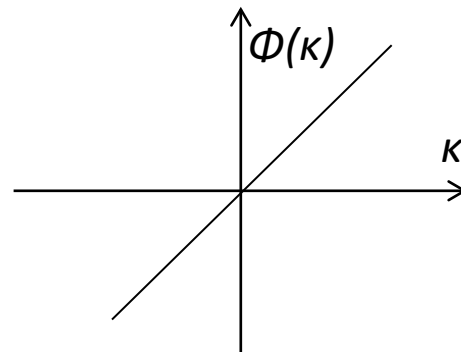
$$\phi(\kappa) = \begin{cases} 1 & \text{dla } \kappa > 0 \\ -1 & \text{dla } \kappa \leq 0 \end{cases}$$

Perceptron



$$\phi(\kappa) = \kappa$$

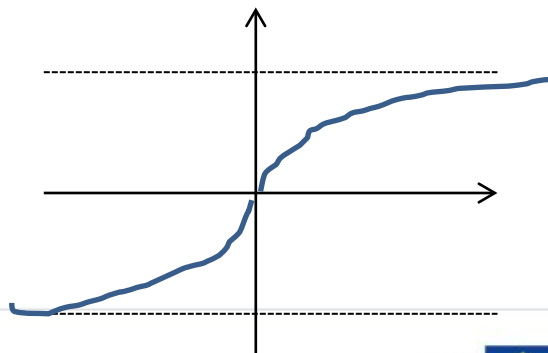
Adaline
Adaptive Linear Neuron



$$\phi(\kappa) = \frac{1}{1 + e^{-\beta\kappa}}$$

$$\phi(\kappa) = \tanh(\beta\kappa) = \frac{1 - e^{-\beta\kappa}}{1 + e^{-\beta\kappa}}$$

Sigmoidal Neuron

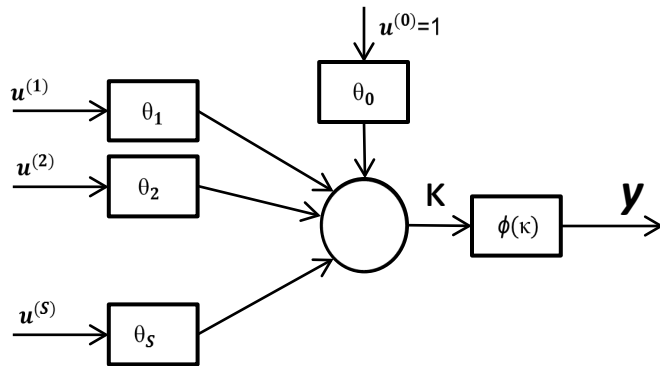




Neuron learning

Adaline
Adaptive Linear Neuron

$$\phi(\kappa) = \kappa$$



$$y = \phi\left(\sum_{s=1}^S \theta_s u^{(s)} + \theta_0\right) = \phi(\theta^T \varphi(u)) = \theta^T \varphi(u)$$

$$\theta = [\theta_0 \ \theta_1 \ \cdots \ \theta_S]^T \quad \varphi(u) = [1 \ u^{(1)} \ \cdots \ u^{(S)}]^T$$

$$[u_1 \ u_2 \ \cdots \ u_N] = U_N \quad [y_1 \ y_2 \ \cdots \ y_N] = Y_N$$

$$Q_N(\theta) = \sum_{n=1}^N (y_n - \bar{y}_n)^2 = \sum_{n=1}^N (y_n - \theta^T \varphi(u_n))^2$$

$$\theta_N^* = \left[\sum_{n=1}^N \varphi(u_n) \varphi^T(u_n) \right]^{-1} \times \sum_{n=1}^N y_n \varphi(u_n)$$

Or recursive algorithm



Recursive algorithm

$$\theta_{N+1}^* = A(\theta_N^*, u_{N+1}, y_{N+1}) = \theta_N^* + K_{N+1} [y_{N+1} - \varphi(u_{N+1})^T \theta_N^*]$$

$$K_{N+1} = \frac{P_N \varphi(u_{N+1})}{1 + \varphi(u_{N+1})^T P_N \varphi(u_{N+1})}$$

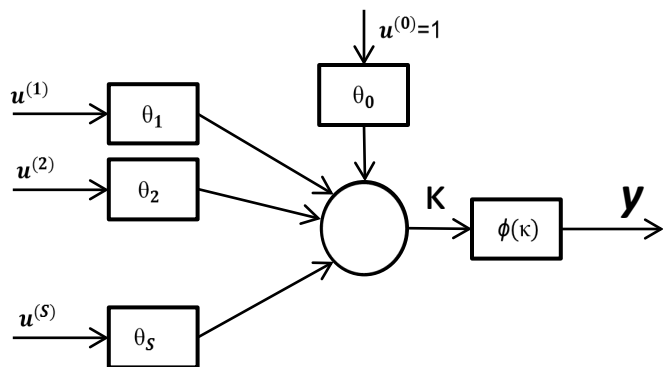
$$P_{N+1} = P_N - \frac{P_N \varphi(u_{N+1}) \varphi(u_{N+1})^T P_N}{1 + \varphi(u_{N+1})^T P_N \varphi(u_{N+1})}$$



Neuron learning

Like neuron Adaline (Adaptive Linear Neuron)

$\phi(\kappa)$ - One to one mapping



$$y = \phi\left(\sum_{s=1}^S \theta_s u^{(s)} + \theta_0\right) = \phi(\theta^T \varphi(u))$$

$$\theta = [\theta_0 \ \theta_1 \ \dots \ \theta_S]^T \quad \varphi(u) = [1 \ u^{(1)} \ \dots \ u^{(S)}]^T$$

Uczenie: $[u_1 \ u_2 \ \dots \ u_N] = U_N \quad [y_1 \ y_2 \ \dots \ y_N] = Y_N$

$$Q_N(\theta) = \sum_{n=1}^N (\kappa_n - \bar{\kappa}_n)^2 = \sum_{n=1}^N (\phi^{-1}(y_n) - \theta^T \varphi(u_n))^2 \quad \text{Like before}$$

$$\theta_N^* = \left[\sum_{n=1}^N \varphi(u_n) \varphi^T(u_n) \right]^{-1} \times \sum_{n=1}^N \phi^{-1}(y_n) \varphi(u_n) \quad \text{Or recursive algorithm}$$



Recursive algorithm

$$\theta_{N+1}^* = A(\theta_N^*, u_{N+1}, y_{N+1}) = \theta_N^* + K_{N+1} [\phi^{-1}(y_{N+1}) - \phi(u_{N+1})^T \theta_N^*]$$

$$K_{N+1} = \frac{P_N \phi(u_{N+1})}{1 + \phi(u_{N+1})^T P_N \phi(u_{N+1})}$$

$$P_{N+1} = P_N - \frac{P_N \phi(u_{N+1}) \phi(u_{N+1})^T P_N}{1 + \phi(u_{N+1})^T P_N \phi(u_{N+1})}$$





Neuron learning – delta rule

$$Q(\theta) = \frac{1}{2} (y - \theta^T \varphi(u))^2, \quad \theta_0$$

Numerical optimization method (it will be)

$$\theta^* \rightarrow Q(\theta^*) = \min_{\theta} Q(\theta)$$

$$\theta_{n+1}^* = \theta_n^* - \eta \nabla_{\theta} Q(\theta_n^*), \quad \theta_0^* \quad \eta - \text{learning factor}$$

$$\nabla Q(\theta) = -(y - \theta^T \varphi(u)) \varphi(u), \quad \theta_0$$

$$\theta_{n+1}^* = \theta_n^* + \eta (y_{n+1} - \theta_n^{*T} \varphi(u_{n+1})) \varphi(u_{n+1}), \quad \theta_0^*$$

$$\Delta_n = (y_{n+1} - \theta_n^{*T} \varphi(u_{n+1}))$$

$$\theta_{n+1}^* = \theta_n^* + \eta \Delta_n \varphi(u_{n+1})$$



Neuron learning – delta roll

$$Q(\theta) = \frac{1}{2} (y - \phi(\theta^T \varphi(u)))^2, \quad \theta_0 \quad \Phi - \text{activation function, differentiable}$$

Numerical optimization method (it will be)

$$\theta^* \rightarrow Q(\theta^*) = \min_{\theta} Q(\theta)$$

$$\theta_{n+1}^* = \theta_n^* - \eta \nabla_{\theta} Q(\theta_n^*), \quad \theta_0^* \quad \eta - \text{learning factor}$$

$$\nabla Q(\theta) = - (y - \phi(\theta^T \varphi(u))) \frac{\partial \phi(\kappa)}{\partial (\kappa)} \varphi(u), \quad \theta_0$$

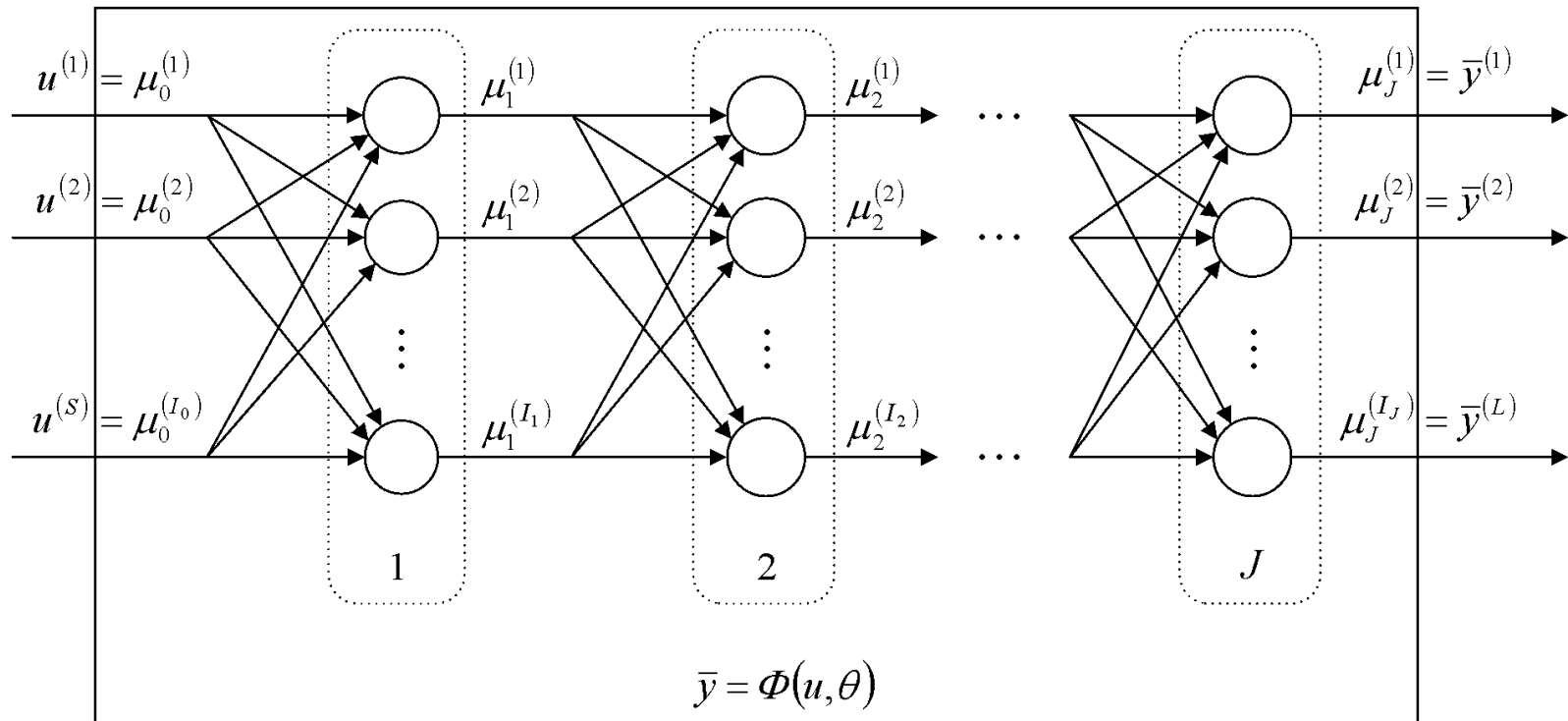
$$\theta_{n+1}^* = \theta_n^* + \eta (y_{n+1} - \phi(\theta_n^{*T} \varphi(u_{n+1}))) \frac{\partial \phi(\kappa)}{\partial \kappa} \varphi(u_{n+1}), \quad \theta_0^*$$

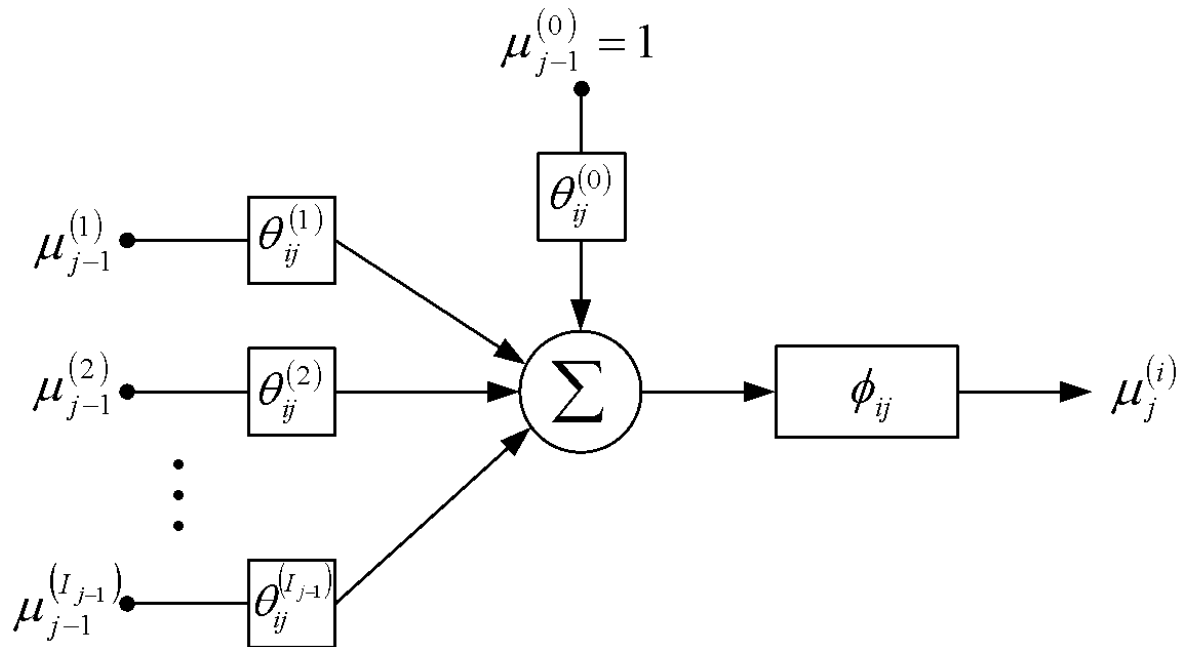
$$\Delta_n = (y_{n+1} - \phi(\theta_n^{*T} \varphi(u_{n+1}))) \frac{\partial \phi(\kappa)}{\partial \kappa}$$

$$\theta_{n+1}^* = \theta_n^* + \eta \Delta_n \varphi(u_{n+1})$$



Multilayer network



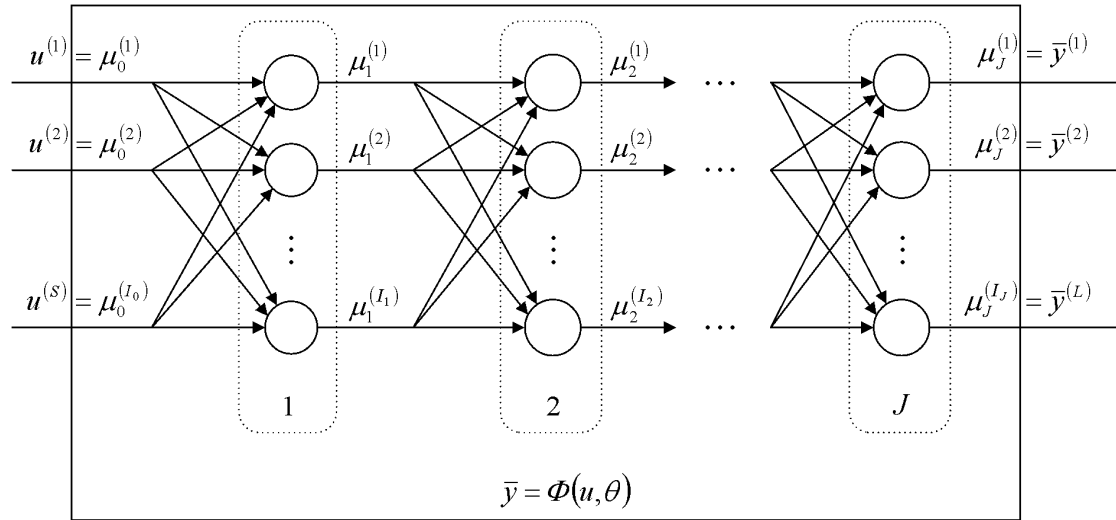


$$\mu_j^{(i)} = \phi_{ij} \left(\bar{\mu}_{j-1}^T \theta_{ij} \right).$$

$$\mu_j^{(i)} = \phi_{ij} \left(\sum_{s=1}^{I_{j-1}} \mu_{j-1}^{(s)} \theta_{ij}^{(s)} + \theta_{ij}^{(0)} \right), \quad \mu_{j-1} = \begin{bmatrix} \mu_{j-1}^{(1)} \\ \mu_{j-1}^{(2)} \\ \vdots \\ \mu_{j-1}^{(I_{j-1})} \end{bmatrix}, \quad \bar{\mu}_{j-1} \stackrel{\text{df}}{=} \begin{bmatrix} \mu_{j-1}^{(0)} \\ \mu_{j-1} \end{bmatrix} = \begin{bmatrix} \mu_{j-1}^{(0)} \\ \mu_{j-1}^{(1)} \\ \mu_{j-1}^{(2)} \\ \vdots \\ \mu_{j-1}^{(I_{j-1})} \end{bmatrix}, \quad \theta_{ij} \stackrel{\text{df}}{=} \begin{bmatrix} \theta_{ij}^{(0)} \\ \theta_{ij}^{(1)} \\ \vdots \\ \theta_{ij}^{(I_{j-1})} \end{bmatrix},$$



Multilayer network



$$\mu_j = \begin{bmatrix} \mu_j^{(1)} \\ \mu_j^{(2)} \\ \vdots \\ \mu_j^{(I_j)} \end{bmatrix} = \begin{bmatrix} \phi_{1j}(\bar{\mu}_{j-1}^T \theta_{1j}) \\ \phi_{2j}(\bar{\mu}_{j-1}^T \theta_{2j}) \\ \vdots \\ \phi_{I_{jj}}(\bar{\mu}_{j-1}^T \theta_{I_{jj}}) \end{bmatrix} \stackrel{\text{df}}{=} \phi_j(\bar{\mu}_{j-1}, \theta_j),$$

$$\theta_j \stackrel{\text{df}}{=} \begin{bmatrix} \theta_{1j} & \theta_{2j} & \dots & \theta_{I_{jj}} \end{bmatrix} = \begin{bmatrix} \theta_{1j}^{(0)} & \theta_{2j}^{(0)} & \dots & \theta_{I_{jj}}^{(0)} \\ \theta_{1j}^{(1)} & \theta_{2j}^{(1)} & \dots & \theta_{I_{jj}}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{1j}^{(I_{j-1})} & \theta_{2j}^{(I_{j-1})} & \dots & \theta_{I_{jj}}^{(I_{j-1})} \end{bmatrix}.$$

$$\mu_j = \phi_j(\bar{\mu}_{j-1}, \theta_j), j = 1, 2, \dots, J.$$

$$\mu_1 = \phi_1(\bar{\mu}_0, \theta_1) = \phi_1(u, \theta_1).$$

$$\mu_j = \phi_j(\bar{\mu}_{j-1}, \theta_j), j = 2, 3, \dots, J-1.$$

$$\bar{y} = \mu_J = \phi_J(\bar{\mu}_{J-1}, \theta_J).$$

$$\bar{y} = \begin{bmatrix} \bar{y}^{(1)} \\ \bar{y}^{(2)} \\ \vdots \\ \bar{y}^{(L)} \end{bmatrix} = \begin{bmatrix} \mu_J^{(1)} \\ \mu_J^{(2)} \\ \vdots \\ \mu_J^{(I_J)} \end{bmatrix} = \phi_J(\phi_{J-1}(\dots \phi_1(u, \theta_1), \dots, \theta_{J-1}), \theta_J) \stackrel{\text{df}}{=} \begin{bmatrix} \Phi^{(1)}(u, \theta) \\ \Phi^{(2)}(u, \theta) \\ \vdots \\ \Phi^{(L)}(u, \theta) \end{bmatrix} \stackrel{\text{df}}{=} \Phi(u, \theta),$$

$$\theta \stackrel{\text{df}}{=} \{\theta_1, \theta_2, \dots, \theta_J\}.$$



Multilayer network learning

$$Q_n(\theta) = [y_n - \Phi(u_n, \theta)]^T [y_n - \Phi(u_n, \theta)] = \sum_{l=1}^L (y_n^{(l)} - \Phi^{(l)}(u_n, \theta))^2 = \sum_{l=1}^L \varepsilon_n^{(l)2} \quad \varepsilon_n^{(l)} \stackrel{\text{df}}{=} y_n^{(l)} - \bar{y}_n^{(l)} = y_n^{(l)} - \Phi^{(l)}(u_n, \theta), \quad l = 1, 2, \dots, L$$

$$\left. \frac{\partial Q_n(\theta)}{\partial \theta_{ij}^{(s)}} \right|_{\theta_{ij}^{(s)} = \tilde{\theta}_{ij,n}^{(s)}} = -2\delta_{jn}^{(i)} \mu_{j-1,n}^{(s)}, \quad \tilde{\theta}_{ij,n+1}^{(s)} = \tilde{\theta}_{ij,n}^{(s)} - \eta_n \left. \frac{\partial Q_n(\theta)}{\partial \theta_{ij}^{(s)}} \right|_{\theta_{ij}^{(s)} = \tilde{\theta}_{ij,n}^{(s)}}, \quad s = 0, 1, \dots, I_{j-1}, \quad i = 1, 2, \dots, I_J, \quad j = 1, 2, \dots, J,$$

$$\varepsilon_{jn}^{(i)} \stackrel{\text{df}}{=} \begin{cases} \varepsilon_n^{(i)} & \text{dla } j = J \\ \sum_{p=1}^{I_{j+1}} \delta_{j+1,n}^{(p)} \tilde{\theta}_{pj+1,n}^{(i)} & \text{dla } j = J-1, J-2, \dots, 1 \end{cases}$$

$$\delta_{jn}^{(i)} \stackrel{\text{df}}{=} \varepsilon_{jn}^{(i)} \left. \frac{d\phi_{ij}(\kappa_j^{(i)})}{d\kappa_j^{(i)}} \right|_{\theta_{ij}^{(s)} = \tilde{\theta}_{ij,n}^{(s)}}$$

$$\kappa_j^{(i)} \stackrel{\text{df}}{=} \sum_{s=1}^{I_{j-1}} \mu_{j-1,n}^{(s)} \theta_{ij}^{(s)} + \theta_{ij}^{(0)}.$$

$$\tilde{\theta}_{ij,n+1}^{(s)} = \tilde{\theta}_{ij,n}^{(s)} + 2\eta_n \delta_{jn}^{(i)} \mu_{j-1,n}^{(s)}, \quad s = 0, 1, \dots, I_{j-1}, \quad i = 1, 2, \dots, I_J, \quad j = 1, 2, \dots, J.$$





Thank you for attention

