Optymalizacja wieloetapowa

Optymalizacja wieloetapowa

$$x_1^*, x_2^*, \ldots, x_S^* \to F(x_1^*, x_2^*, \ldots, x_S^*) = \min_{x_1, x_2, \ldots, x_S \in \mathscr{D}_x} F(x_1, x_2, \ldots, x_S)$$

$$\mathscr{D}_x = (x_1, x_2, \cdots, x_S)$$

$$\left\{ [x_1 \; x_2 \cdots x_S]^T \in \mathscr{R}^S : \varphi_l(x_1, x_2, \cdots, x_S) = 0, l = 1, 2, \ldots, L, \right.$$
$$\left. \psi_m(x_1, x_2, \cdots, x_S) \le 0, m = 1, 2, \ldots, M \right\}$$

Powyższe zadanie można rozwiązać krok po kroku optymalizując po jednej zmiennej i uzależniając od pozostałych.

Przyjmijmy oznaczenia:

$$F \equiv F_S, \quad \varphi_l \equiv \varphi_{lS}, l = 1, 2, \ldots, L, \quad \psi_m = \psi_{mS}, m = 1, 2, \ldots, M \quad \mathscr{D}_x \equiv \mathscr{D}_{x_S}$$

Optymalizacja wieloetapowa

Krok 1.

$$x_S^* = G_S(x_1, \ldots, x_{S-1}) \rightarrow F_S(x_1, x_2, \ldots, x_S^*) = \min_{x_S \in \mathscr{D}_{xS}} F_S(x_1, x_2, \ldots, x_S)$$

Wartość funkcji celu w punkcie optymalnym:

$$F_{S-1}(x_1, x_2, \ldots, x_{S-1}) \stackrel{\Delta}{=} F_S(x_1, x_2, \ldots, x_S^*) = F_S(x_1, x_2, \ldots, G_S(x_1, \ldots, x_{S-1}))$$

Ograniczenia w punkcie optymalnym:

$$\mathscr{D}_{xS-1}(x_1, \ldots, x_{S-1}) \stackrel{\Delta}{=} \mathscr{D}_{xS}(x_1, \ldots, x_{S-1}, x_S^* = G_S(x_1, \ldots, x_{S-1})) =$$

$$\left\{ \begin{array}{l} [x_1 \; x_2 \cdots x_{S-1}]^T \in \mathscr{R}^{S-1} : \\ \varphi_{lS}(x_1, x_2, \cdots, G_S(x_1, \ldots, x_{S-1})) = \varphi_{lS-1}(x_1, x_2, \cdots, x_{S-1}) = 0, \, l = 1, 2, \ldots, L, \\ \psi_{mS}(x_1, x_2, \cdots, G_S(x_1, \ldots, x_{S-1})) = \psi_{mS-1}(x_1, x_2, \cdots, x_{S-1}) \leq 0, \, m = 1, 2, \ldots, M \end{array} \right\}$$

Optymalizacja wieloetapowa

Krok 2.

$$x_{S-1}^{*} = G_{S-1}(x_1, \ldots, x_{S-2}) \rightarrow F_{S-1}(x_1, x_2, \ldots, x_{S-1}^{*}) = \min_{x_{S-1} \in \mathcal{D}_{xS-1}} F_{S-1}(x_1, x_2, \ldots, x_{S-1})$$

Wartość funkcji celu w punkcie optymalnym:

$$F_{S-2}(x_1, x_2, \ldots, x_{S-2}) \overset{\Delta}{=} F_{S-1}(x_1, x_2, \ldots, x_{S-1}^{*}) = F_{S-1}(x_1, x_2, \ldots, G_{S-1}(x_1, \ldots, x_{S-2}))$$

Ograniczenia w punkcie optymalnym:

$$\mathcal{D}_{xS-2}(x_1, \ldots, x_{S-2}) \overset{\Delta}{=} \mathcal{D}_{xS-1}(x_1, \ldots, x_{S-2}, x_{S-1}^{*} = G_{S-1}(x_1, \ldots, x_{S-2})) =$$

$$\left\{ \begin{array}{l} [x_1 \; x_2 \cdots x_{S-2}]^T \in \mathcal{R}^{S-2} : \\ \varphi_{lS-1}(x_1, x_2, \cdots, G_{S-1}(x_1, \ldots, x_{S-2})) = \varphi_{lS-2}(x_1, x_2, \cdots, x_{S-2}) = 0, \; l = 1, 2, \ldots, L, \\ \psi_{mS-1}(x_1, x_2, \cdots, G_{S-1}(x_1, \ldots, x_{S-2})) = \psi_{mS-2}(x_1, x_2, \cdots, x_{S-2}) \leq 0, \; m = 1, 2, \ldots, M \end{array} \right\}$$

$$\vdots$$

Optymalizacja wieloetapowa

$\vdots$

Krok S-1.

$$x_2^* = G_2(x_1) \rightarrow F_2(x_1, x_2^*) = \min_{x_2 \in \mathscr{D}_{x2}} F_2(x_1, x_2)$$

Wartość funkcji celu w punkcie optymalnym:

$$F_1(x_1) \stackrel{\Delta}{=} F_2(x_1\, x_2^*) = F_2(x_1, G_2(x_1))$$

Ograniczenia w punkcie optymalnym:

$$\mathscr{D}_{x1}(x_1) \stackrel{\Delta}{=} \mathscr{D}_{x2}(x_1, x_2^* = G_2(x_1)) = \left\{ \begin{array}{l} x_1 \in \mathscr{R}: \\ \varphi_{l2}(x_1, G_2(x_1)) = \varphi_{l1}(x_1) = 0,\ l = 1, 2, \ldots, L, \\ \psi_{m2}(x_1, G_2(x_1)) = \psi_{m1}(x_1) \leq 0,\ m = 1, 2, \ldots, M \end{array} \right\}$$

Optymalizacja wieloetapowa

Krok S.

$$x_1^* \rightarrow F_1\left(x_1^*\right) = \min_{x_1 \in \mathcal{D}_{x1}} F_1\left(x_1\right)$$

Teraz możemy powrócić do zależności „*G*" wyznaczonych w poprzednich krokach

$$x_1^*$$

$$x_2^* = G_2\left(x_1^*\right)$$

$$\vdots$$

$$x_{S-1}^* = G_{S-1}\left(x_1^*, x_2^* \ldots, x_{S-2}^*\right)$$

$$x_S^* = G_S\left(x_1^*, x^*, \ldots, x_{S-1}^*\right)$$

Optymalizacja wieloetapowa

Przykład:

$$F(x_1, x_2) = x_1^2 + x_1 x_2 + x_2^2 = F_2(x_1, x_2)$$

Krok 1.

$$x_2^* = G_2(x_1) \rightarrow F_2(x_1, x_2^*) = \min_{x_2} \left\{ x_1^2 + x_1 x_2 + x_2^2 \right\}$$

$$\frac{\partial}{\partial x_2} F_2(x_1, x_2^*) = x_1 + 2x_2^* = 0 \Rightarrow x_2^* = -\frac{1}{2} x_1 = G_2(x_1)$$

$$F_1(x_1) \stackrel{\Delta}{=} F(x_1, x_2^*) = F(x_1, G_2(x_1)) = (x_1)^2 + x_1\left(-\frac{1}{2} x_1\right) + \left(-\frac{1}{2} x_1\right)^2$$

$$F_1(x_1) = \frac{3}{4} x_1^2$$

Optymalizacja wieloetapowa

Krok 2.

$$x_1^* =\rightarrow F_1(x_1)=\min_{x_1}\left\{\frac{3}{4}x_1^2\right\}$$

$$\frac{d}{dx_1}F_1(x_1)=2\frac{3}{4}x_1=0\Rightarrow x_1^*=0$$

Teraz wracamy:

$$x_1^*=0$$

$$x_2^*=G_2(x_1^*)=-\frac{1}{2}x_1^*=0$$

Programowanie dynamiczne

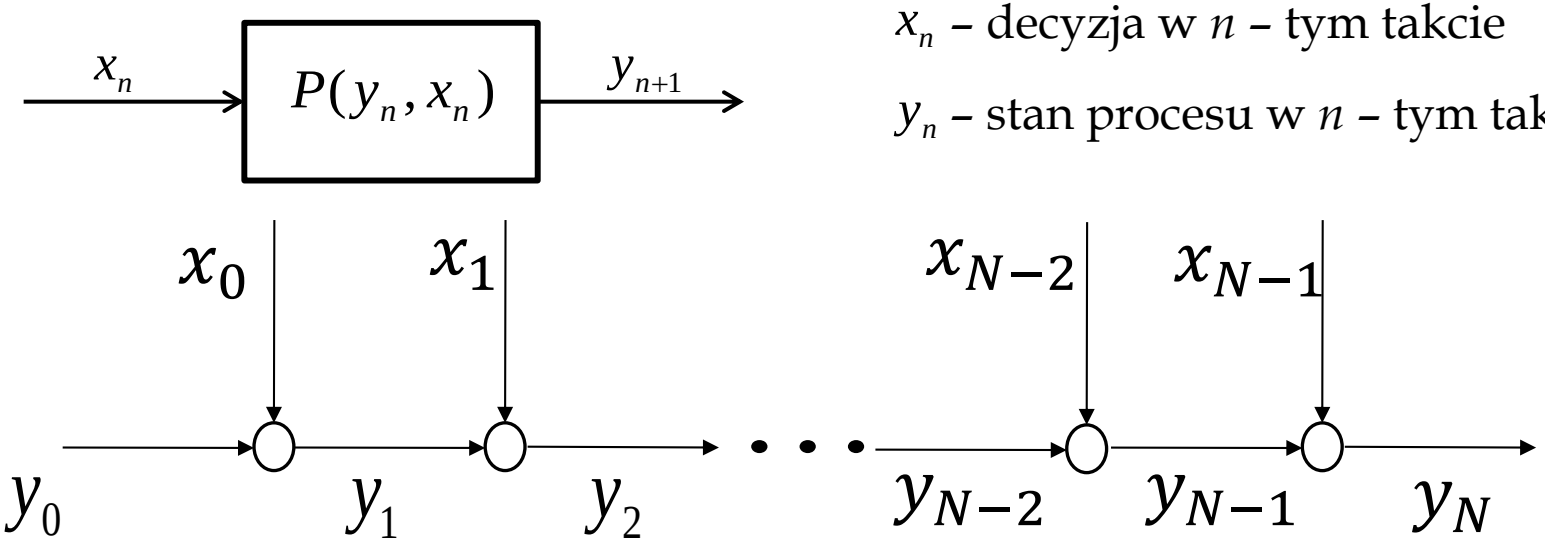
Programowanie dynamiczne

Proces dynamiczny: $y_{n+1} = P(y_n, x_n),\ y_o$

n – takt $n = 1, 2, \ldots, N$

x_n – decyzja w n – tym takcie

y_n – stan procesu w n – tym takcie



Zadanie decyzyjne: znaleźć ciąg decyzji: $x_0^*, x_1^*, \ldots, x_{N-1}^*,$

Po co? Po N krokach zamierzamy coś osiągnąć np.: $y_N = y^*$ - zadany stan,

Przy czym wskaźnik $Q(x_0, \ldots, x_{N-1}, y_1, \ldots, y_N)$ przyjmuje wartość minimalną.

Programowanie dynamiczne

Określmy przykładowe oceny ciągu decyzji oraz efektów decyzji

$$Q(x_0, x_1, \ldots, x_{N-1}, y_1, y_2, \ldots, y_N)$$

Np.:

$$Q(x_0, x_1, \ldots, x_{N-1}, y_1, y_2, \ldots, y_N) = \sum_{n=0}^{N-1} x_n^2, \quad y_N = y^*$$

$$Q(x_0, x_1, \ldots, x_{N-1}, y_1, y_2, \ldots, y_N) = \sum_{n=0}^{N-1} x_n^2, \quad \underline{y} \le y_N \le \bar{y}$$

$$Q(x_0, x_1, \ldots, x_{N-1}, y_1, y_2, \ldots, y_N) = \sum_{n=1}^{N-1} (y_n - y_n^*), \quad 0 \le x_n \le \bar{x}$$

$$Q(x_0, x_1, \ldots, x_{N-1}, y_1, y_2, \ldots, y_N) = \sum_{n=0}^{N-1} A_{n+1}(x_n, y_{n+1}), \quad \textit{przy ograniczeniach } x_n \textit{ i } y_n$$

Programowanie dynamiczne

Można postąpić w następujący sposób:

$$y_0$$

$$y_1 = P(y_0, x_0)$$

$$y_2 = P(y_1, x_1) = P(P(y_0, x_0), x_1) \stackrel{\Delta}{=} P_1(y_0, x_0, x_1)$$

$$\vdots$$

$$y_N = P(y_{N-1}, x_{N-1}) = P(P(y_{N-2}, x_{N-2}), x_{N-1}) = \stackrel{y_{n+1}=P(y_n, x_n)}{\cdots} = P_{N-1}(y_0, x_0, x_1, \ldots, x_N)$$

Po podstawieniu do *Q(.)* otrzymujemy:

$$Q(x_0, x_1, \ldots, x_{N-1}, y_1, y_2, \ldots, y_N) = \sum_{n=0}^{N-1} A_{n+1}(x_n, y_{n+1}) \stackrel{\Delta}{=} F(y_0, x_0, x_1, \ldots, x_{N-1})$$

Programowanie dynamiczne

Ponieważ:

$$Q(x_0, x_1, \ldots, x_{N-1}, y_1, y_2, \ldots, y_N) = \sum_{n=0}^{N-1} A_{n+1}(x_n, y_{n+1}) \stackrel{\Delta}{=} F(y_0, x_0, x_1, \ldots, x_{N-1})$$

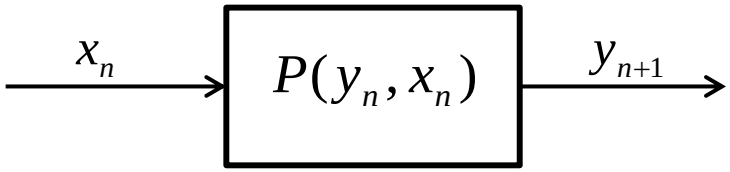
Zadanie optymalizacji:

$$x_0^*, x_1^*, \ldots, x_{N-1}^* \rightarrow Q(x_0^*, x_1^*, \ldots, x_{N-1}^*, y_1, y_2, \ldots, y_N) = \min_{x_0, x_1, \ldots, x_{N-1}} Q(x_0, x_1, \ldots, x_{N-1}, y_1, y_2, \ldots, y_N)$$

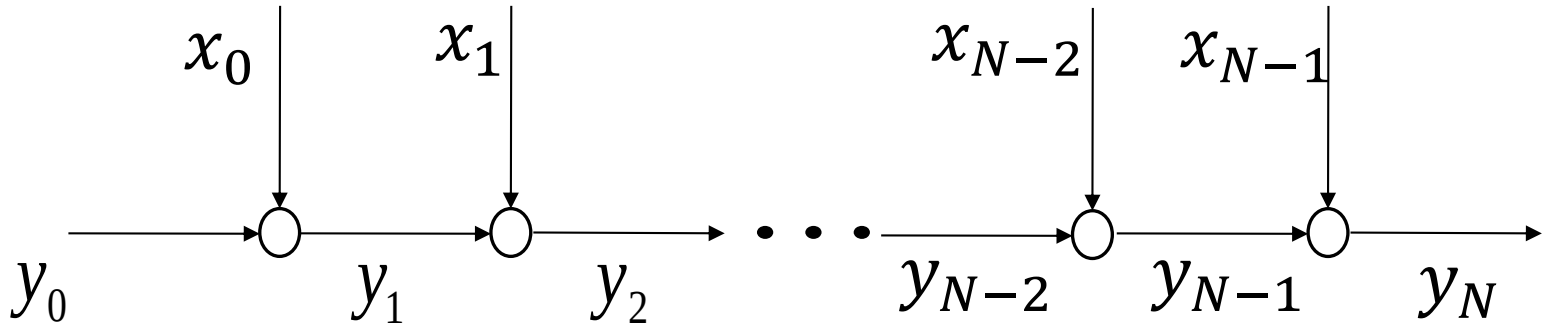
Jest równoważne zadaniu:

$$x_0^*, x_1^*, \ldots, x_{N-1}^* \rightarrow F(y_0, x_0^*, x_1^*, \ldots, x_{N-1}^*) = \min_{x_0, x_1, \ldots, x_{N-1}} F(y_0, x_0, x_1, \ldots, x_{N-1})$$

Do rozwiązania tego zadania można zastosować podejście wieloetapowe.



$$y_{n+1} = P(y_n, x_n),\ y_o$$



$$Q(x_0, x_1, \ldots, x_{N-1}, y_1, y_2, \ldots, y_N) = \sum_{n=0}^{N-1} A_{n+1}(x_n, y_{n+1}) \stackrel{\Delta}{=} F(y_0, x_0, x_1, \ldots, x_{N-1})$$

Programowanie dynamiczne

$$Q(x_0, x_1, \ldots, x_{N-1}, y_1, y_2, \ldots, y_N) = \sum_{n=0}^{N-1} A_{n+1}(x_n, y_{n+1}) \stackrel{\Delta}{=} F(y_0, x_0, x_1, \ldots, x_{N-1})$$

Uwaga:

Ostatni stan y_N zależy od decyzji x_{N-1} oraz stanu y_{N-1}, który zależy od poprzednich decyzji x_0, x_1, …, x_{N-2}.

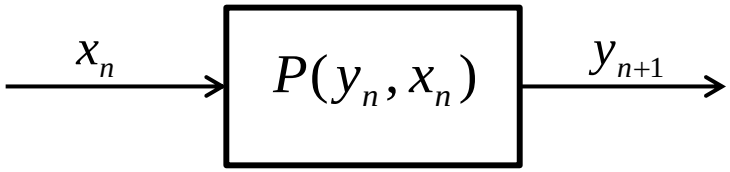
Poprzedni stan y_{N-1} zależy od decyzji x_{N-2} oraz stanu y_{N-2}, który zależy od poprzednich decyzji x_0, x_1, …, x_{N-3}.

⋮

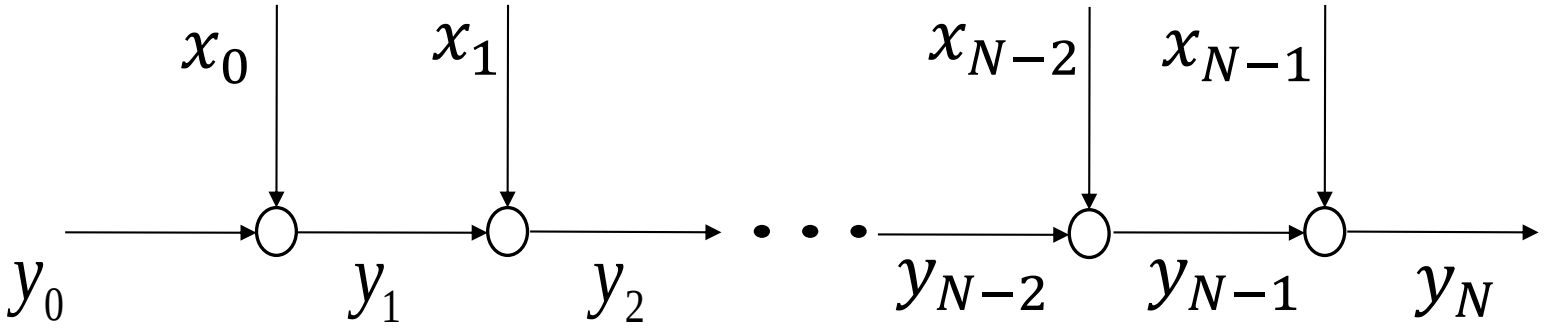
Stan y_2 zależy od decyzji x_1 oraz stanu y_1, który zależy od poprzednich decyzji x_0, x_1.

Stan y_1 zależy od decyzji x_0 oraz stanu y_0, który to wartości są znane.

Do rozwiązania zadania optymalizacji powyższej funkcji można zastosować podejście wieloetapowe. Ze względu na specyficzną postać kryterium (suma funkcji decyzji x_n oraz stanu y_{n+1} w kolejnych taktach. Optymalizując kolejno od ostatniego składnika uzależniamy rozwiązanie od poprzedniego stanu i poprzednich decyzji.



$$y_{n+1} = P(y_n, x_n),\ y_o$$



$$Q(x_0, x_1, \ldots, x_{N-1}, y_1, y_2, \ldots, y_N) = \sum_{n=0}^{N-1} A_{n+1}(x_n, y_{n+1}) \stackrel{\Delta}{=} F(y_0, x_0, x_1, \ldots, x_{N-1})$$

Programowanie dynamiczne

$$Q(x_0, x_1, \ldots, x_{N-1}, y_1, y_2, \ldots, y_N) = \sum_{n=0}^{N-1} A_{n+1}(x_n, y_{n+1})$$

Krok 1. $$x_{N-1}^* \to \min_{x_{N-1}} A_N(x_{N-1}, y_N)$$

Wiemy że: $$y_N = P(y_{N-1}, x_{N-1})$$

$$x_{N-1}^* = G_{N-1}(y_{N-1}) \to \min_{x_{N-1}} A_N(x_{N-1}, P(y_{N-1}, x_{N-1}))$$

$$V_{N-1}(y_{N-1}) \stackrel{\Delta}{=} \min_{x_{N-1}} A_N(x_{N-1}, P(y_{N-1}, x_{N-1})) =$$

$$= A_N(x_{N-1}^*, P(y_{N-1}, x_{N-1}^*)) = A_N(G_{N-1}(y_{N-1}), P(y_{N-1}, G_{N-1}(y_{N-1})))$$

Programowanie dynamiczne

Krok 2. $$x^*_{N-2} \rightarrow \min_{x_{N-2}} \left\{ A_{N-1}\left(x_{N-2}, y_{N-1}\right) + V_{N-1}\left(y_{N-1}\right) \right\}$$

Wiemy że: $$y_{N-1} = P\left(y_{N-2}, x_{N-2}\right)$$

$$x^*_{N-2} = G_{N-2}\left(y_{N-2}\right) \rightarrow \min_{x_{N-2}} \left\{ A_{N-1}\left(x_{N-2}, P\left(y_{N-2}, x_{N-2}\right)\right) + V_{N-1}\left(P\left(y_{N-2}, x_{N-2}\right)\right) \right\}$$

$$V_{N-2}\left(y_{N-2}\right) \stackrel{\Delta}{=} \min_{x_{N-2}} \left\{ A_{N-1}\left(x_{N-2}, P\left(y_{N-2}, x_{N-2}\right)\right) + V_{N-1}\left(P\left(y_{N-2}, x_{N-2}\right)\right) \right\} =$$

$$= \left\{ A_{N-1}\left(x^*_{N-2}, P\left(y_{N-2}, x^*_{N-2}\right)\right) + V_{N-1}\left(P\left(y_{N-2}, x^*_{N-2}\right)\right) \right\} =$$

$$= A_{N-1}\left(G_{N-2}\left(y_{N-2}\right), P\left(y_{N-2}, G_{N-2}\left(y_{N-2}\right)\right)\right) + V_{N-1}\left(P\left(y_{N-2}, G_{N-2}\left(y_{N-2}\right)\right)\right)$$

$$\vdots$$

Programowanie dynamiczne

Krok N-1.

$$x_1^* \rightarrow \min_{x_1}\left\{A_2(x_1, y_2)+V_2(y_2)\right\}$$

Wiemy że:

$$y_2 = P(y_1, x_1)$$

$$x_1^* = G_1(y_1) \rightarrow \min_{x_1}\left\{A_2(x_1, P(y_1, x_1))+V_2(P(y_1, x_1))\right\}$$

$$V_1(y_1) \overset{\Delta}{=} \min_{x_1}\left\{A_2(x_1, P(y_1, x_1))+V_2(P(y_1, x_1))\right\} =$$

$$= A_2(x_1^*, P(y_1, x_1^*))+V_2(P(y_1, x_1^*))$$

$$= A_2(G_1(y_1), P(y_1, G_1(y_1)))+V_2(P(y_1, G_1(y_1)))$$

Programowanie dynamiczne

Krok N.

$$x_0^* \rightarrow \min_{x_0}\left\{A_1(x_0, y_1) + V_1(y_1)\right\}$$

Wiemy że: $y_1 = P(y_0, x_0)$

$$x_0^* = G_0(y_0) \rightarrow \min_{x_0}\left\{A_1(x_0, P(y_0, x_0)) + V_1(P(y_0, x_0))\right\}$$

y_0 jest znaną wartością i teraz możemy wyliczyć kolejne wartości decyzji $x_0^*, x_1^*, \ldots, x_{N-1}^*$,

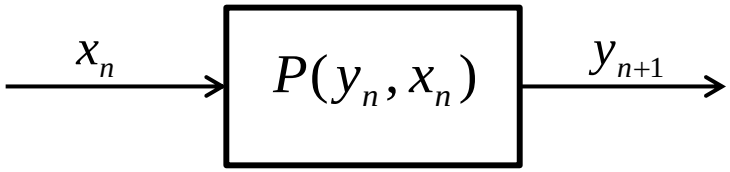
$$x_0^* = G_0(y_0) \rightarrow y_1 = P(y_0, x_0^*)$$

$$x_1^* = G_1(y_1) \rightarrow y_2 = P(y_2, x_2^*)$$

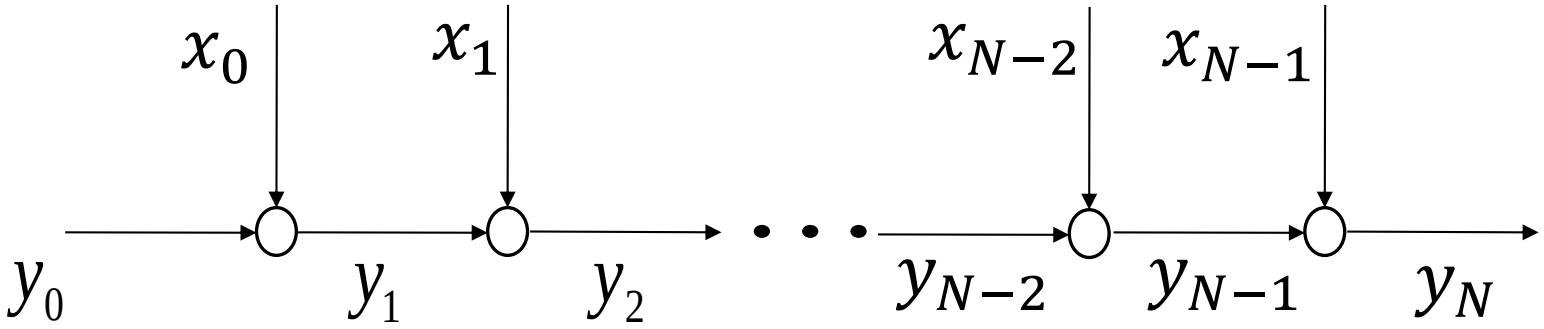
$$\vdots$$

$$x_{N-2}^* = G_{N-2}(y_{N-2}) \rightarrow y_{N-1} = P(y_{N-2}, x_{N-2}^*)$$

$$x_{N-1}^* = G_{N-1}(y_{N-1}) \rightarrow y_N = P(y_{N-1}, x_{N-1}^*)$$



$$y_{n+1} = P(y_n, x_n),\ y_o$$



$$Q(x_0, x_1, \ldots, x_{N-1}, y_1, y_2, \ldots, y_N) = \sum_{n=0}^{N-1} A_{n+1}(x_n, y_{n+1}) \stackrel{\Delta}{=} F(y_0, x_0, x_1, \ldots, x_{N-1})$$

Programowanie dynamiczne

Przykład: $y_{n+1} = P(y_n, x_n) = 2y_n + x_n, \quad y_o = 0$

$$Q(x_0, x_1, y_1, y_2) = \sum_{n=0}^{1} A_{n+1}(x_n, y_{n+1}) = \sum_{n=0}^{1} \left(x_n^2 + (y_{n+1} - 5)^2\right) =$$
$$= \left(x_0^2 + (y_1 - 5)^2\right) + \left(x_1^2 + (y_2 - 5)^2\right)$$

Krok 1.

$$x_1^* \to \min_{x_1}\left(x_1^2 + (y_2 - 5)^2\right)$$

$$y_2 = 2y_1 + x_1$$

$$x_1^* = G_1(y_1) \to \min_{x_1}\left(x_1^2 + (2y_1 + x_1 - 5)^2\right)$$

$$2x_1^* + 2\left(2y_1 + x_1^* - 5\right) = 0 \Rightarrow x_1^* = G_1(y_1) = \frac{5}{2} - y_1$$

$$V_1(y_1) = \left(\frac{5}{2} - y_1\right)^2 + \left(2y_1 + \frac{5}{2} - y_1 - 5\right)^2 = 2\left(y_1 - \frac{5}{2}\right)^2$$

Programowanie dynamiczne

Krok 2. $x_0^* \to \min_{x_0}\left\{\left(x_0^2+(y_1-5)^2\right)+2\left(y_1-\frac{5}{2}\right)^2\right\}$

$$y_1 = 2y_0 + x_0$$

$$x_0^* = G_0(y_0) \to \min_{x_0}\left\{\left(x_0^2+(2y_0+x_0-5)^2\right)+2\left(2y_0+x_0-\frac{5}{2}\right)^2\right\}$$

$$2x_0^* + 2\left(2y_0 + x_0^* - 5\right) + 4\left(2y_0 + x_0 - \frac{5}{2}\right) = 0 \Rightarrow x_0^* = G_0(y_0) = \frac{15}{8} - \frac{3}{2}y_0 = \frac{15}{8}$$

Teraz wracamy:

$$x_0^* = \frac{15}{8} \to y_1 = 2\times 0 + \frac{15}{8} = \frac{15}{8}$$

$$x_1^* = \frac{5}{2} - \frac{15}{8} = \frac{5}{8} \to y_1 = 2\times 0 + \frac{5}{8} = \frac{5}{8}$$

Przypadek liniowo – kwadratowy Algorytm rekurencyjny

$$Q_N(\theta)=\sum_{n=1}^{N}(y_n-\bar{y}_n)^2=\sum_{n=1}^{N}\left(y_n-\theta^T\varphi(u_n)\right)^2 \quad U_N=[u_1 \quad u_2 \quad \cdots \quad u_N], \quad Y_N=[y_1 \quad y_2 \quad \cdots \quad y_N]$$

Wektor parametrów θ_N^* wyznaczamy na podstawie N pomiarów

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

Pojawia się nowy $N+1$ pomiar u_{N+1}, y_{N+1}

Jak wyznaczyć (poprawić) parametr korzystając z nowego pomiaru?

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

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

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

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

Przypadek liniowo – kwadratowy Algorytm rekurencyjny

Przydatne przekształcenia algebraiczne:

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

$\mathbf{B}=\varphi$ – wektor kolumnowy

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

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

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

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

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

Przypadek liniowo – kwadratowy Algorytm rekurencyjny

Oznaczmy przez: $\varphi_n = \varphi(u_n), n = 1, 2, \ldots, N+1$

$$\left(\sum_{n=1}^{N+1} \varphi_n \varphi_n^T \right)^{-1} = \left(\sum_{n=1}^{N} \varphi_n \varphi_n^T + \varphi_{N+1} \varphi_{N+1}^T \right)^{-1} =$$

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

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

Powyższe przyjmuje postać:

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

Przypadek liniowo – kwadratowy
Algorytm rekurencyjny

$$\theta^*_{N+1} = \left(P_N - P_N \frac{\varphi_{N+1}\varphi^T_{N+1}}{1+\varphi^T_{N+1}P_N\varphi_{N+1}} P_N \right)\left(\sum_{n=1}^{N} \varphi_n y_n + \varphi_{N+1} y_{N+1} \right) =$$

$$= P_N \sum_{n=1}^{N} \varphi_n y_n + P_N \varphi_{N+1} y_{N+1} - \frac{P_N \varphi_{N+1}\varphi^T_{N+1} P_N}{1+\varphi^T_{N+1}P_N\varphi_{N+1}} \sum_{n=1}^{N} \varphi_n y_n - \frac{P_N \varphi_{N+1}\varphi^T_{N+1} P_N \varphi_{N+1} y_{N+1}}{1+\varphi^T_{N+1}P_N\varphi_{N+1}} =$$

$$= P_N \sum_{n=1}^{N} \varphi_n y_n + \frac{P_N \varphi_{N+1} y_{N+1} + P_N \varphi_{N+1} y_{N+1} \varphi^T_{N+1} P_N \varphi_{N+1} - P_N \varphi_{N+1} y_{N+1} \varphi^T_{N+1} P_N \varphi_{N+1} - P_N \varphi_{N+1}\varphi^T_{N+1} P_N \sum_{n=1}^{N} \varphi_n y_n}{1+\varphi^T_{N+1}P_N\kappa_{N+1}} =$$

$$= P_N \sum_{n=1}^{N} \varphi_n y_n + \frac{P_N \varphi_{N+1} y_{N+1} - P_N \varphi_{N+1}\varphi^T_{N+1} P_N \sum_{n=1}^{N} \varphi_n y_n}{1+\varphi^T_{N+1}P_N\varphi_{N+1}} y =$$

$$= P_N \sum_{n=1}^{N} \varphi_n y_n + \frac{P_N \varphi_{N+1} y_{N+1}}{1+\varphi^T_{N+1}P_N\varphi_{N+1}} \left(y_{N+1} + \varphi^T_{N+1} P_N \sum_{n=1}^{N} \varphi_n y_n \right)$$

$$= \theta^*_N + K_{N+1}\left(y_{N+1} - \varphi^T_{N+1}\theta_N \right)$$

gdzie: $K_{N+1} = \dfrac{P_N \varphi_{N+1}}{1+\varphi^T_{N+1}P_N\varphi_{N+1}}$

Przypadek liniowo – kwadratowy Algorytm rekurencyjny

Podsumowując, otrzymujemy następujący algorytm

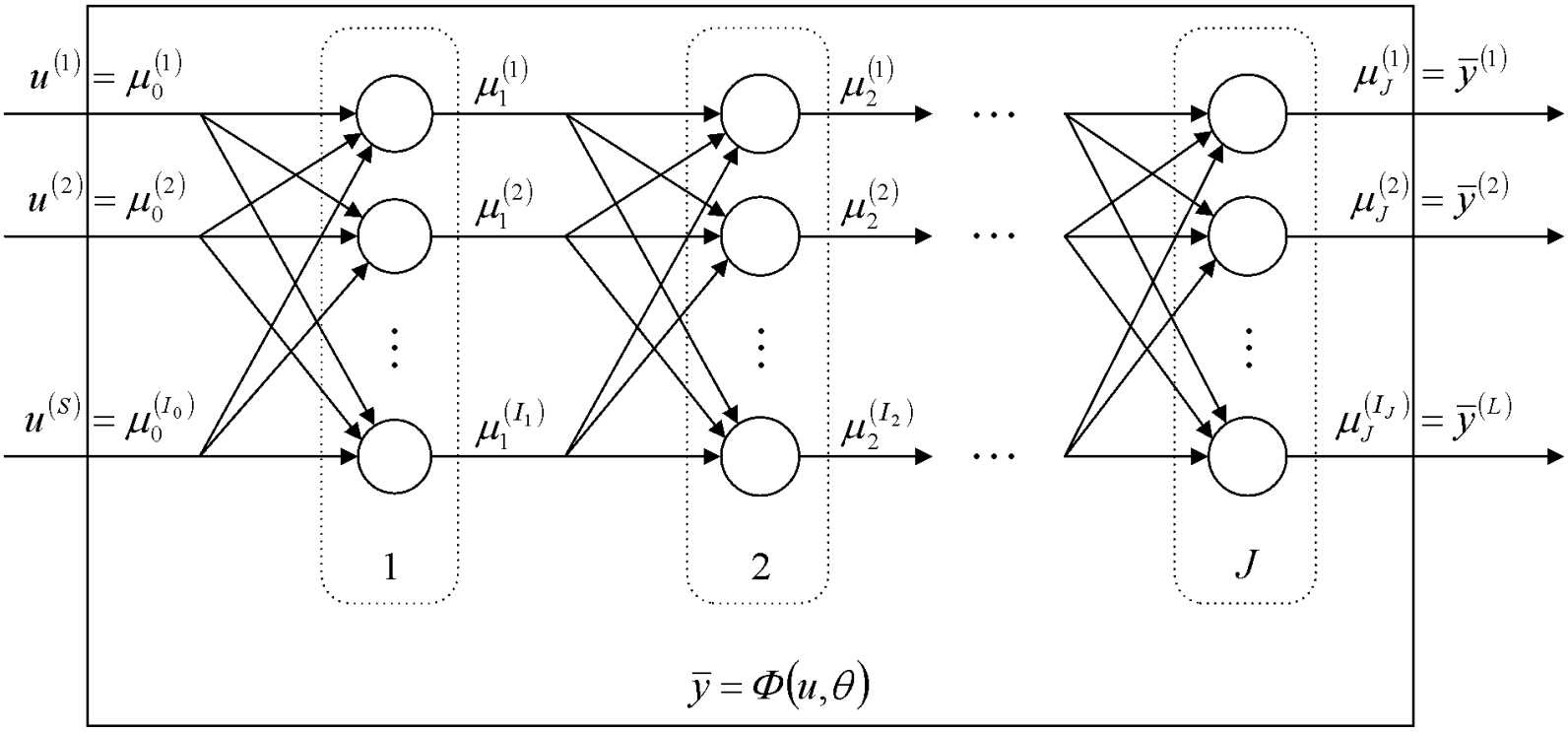
$$\theta_N^* = \left(\sum_{n=1}^{N} \varphi_n \varphi_n^T \right)^{-1} \sum_{n=1}^{N} \varphi_n y_n = P_N \sum_{n=1}^{N} \varphi_n y_n \,, \text{ gdzie: } P_N = \left(\sum_{n=1}^{N} \varphi_n \varphi_n^T \right)^{-1}$$

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

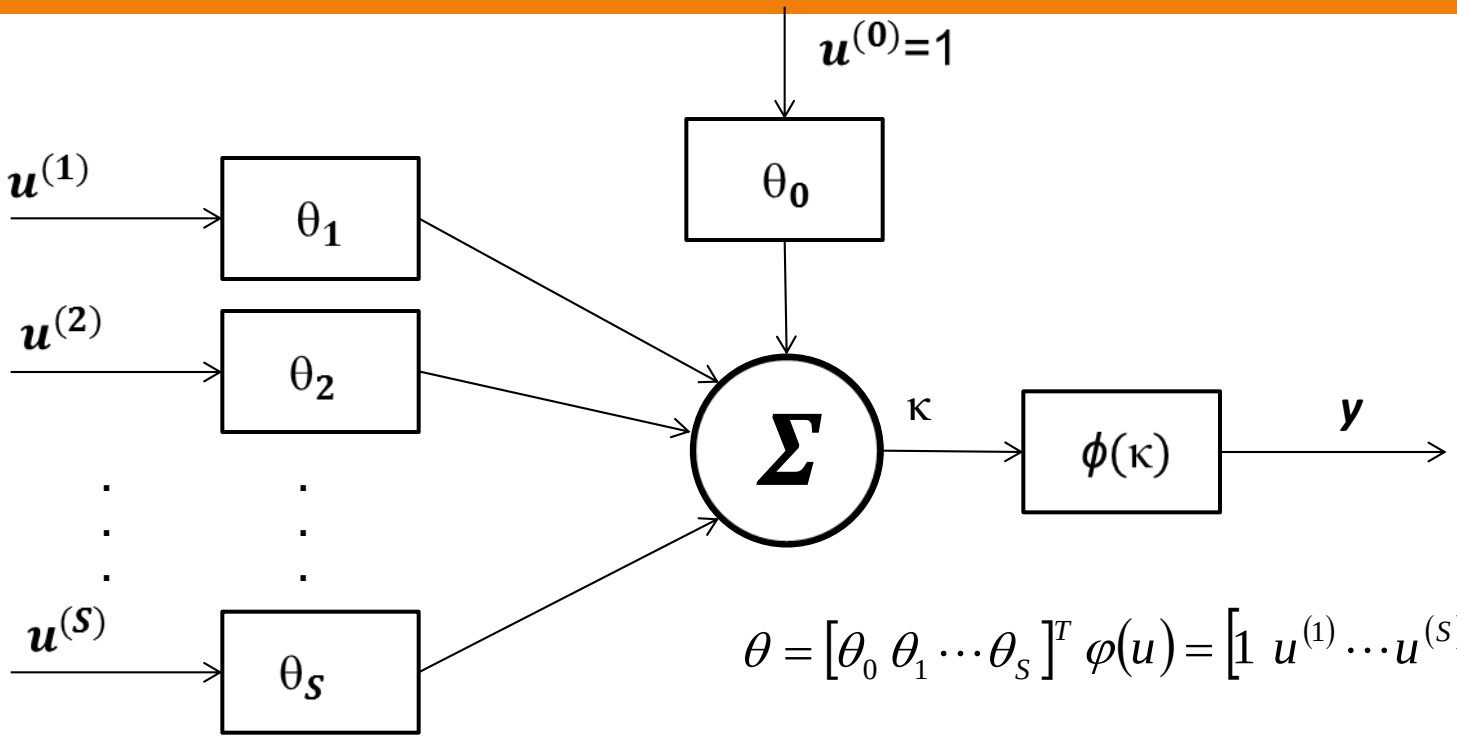
$$K_{N+1} = \frac{P_N \varphi\left(u_{N+1}\right)}{1 + \varphi\left(u_{N+1}\right)^T P_N \varphi\left(u_{N+1}\right)}$$

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

Sieć wielowarstwowa



Model neuronu (uproszczenie)



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

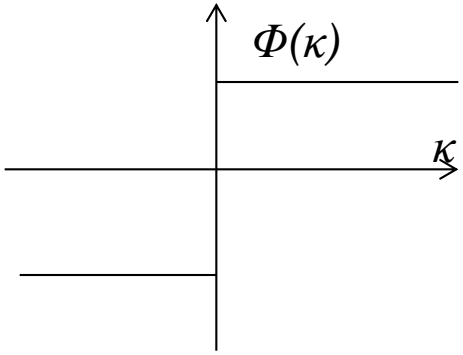
$$y = \phi\left(\sum_{s=1}^{S} \theta_s u^{(s)} + \theta_0\right) = \phi\left(\theta^T \varphi(u)\right)$$

Φ – funkcja aktywacji

Funkcja aktywacji

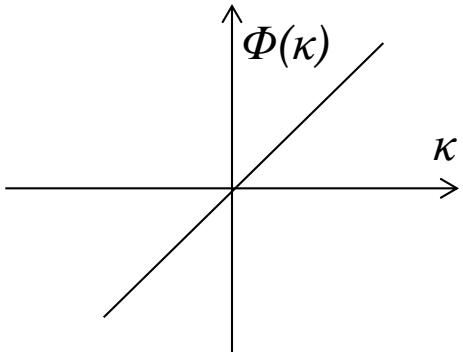
$$\phi(\kappa) = \begin{cases} 1 \; dla \; \kappa > 0 \\ -1 \; dla \; \kappa \le 0 \end{cases}$$

Perceptron



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

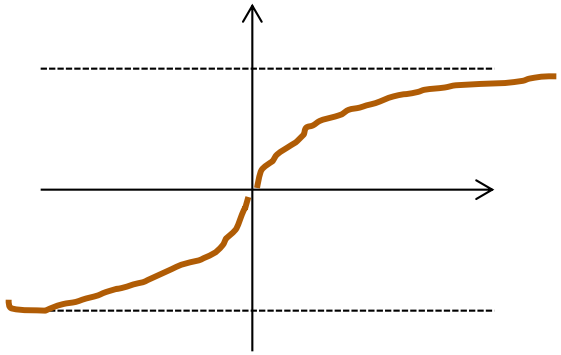
Adaline
Adaptive Linear Neuron



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

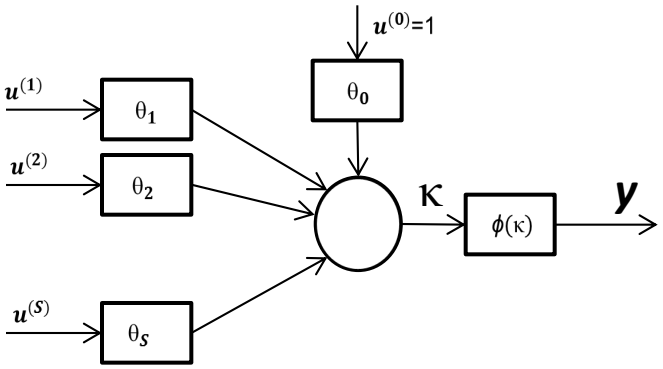
$$\phi(\kappa) = \tanh(\beta\kappa) = \frac{1 - e^{-\beta\kappa}}{1 + e^{-\beta\kappa}}$$

Neuron sigmoidalny



Uczenie neuronu

Neuron Adaline (Adaptive Linear Neuron) $\phi(\kappa) = \kappa$



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

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

Uczenie: $\left[u_1\ u_2 \cdots u_N\right] = U_N \qquad \left[y_1\ y_2 \cdots y_N\right] = Y_N$

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

To już było

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

lub rekurencyjnie

Algortm rekurencyjny

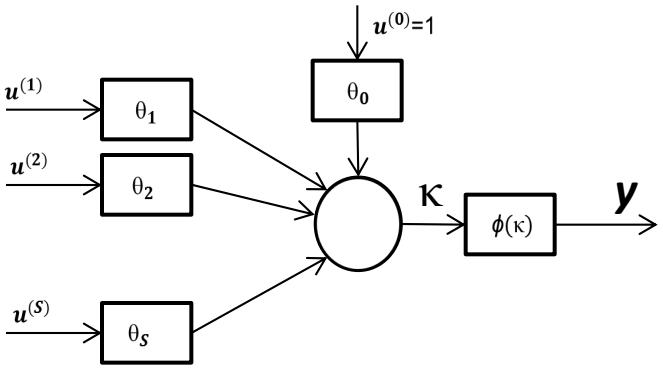
$$\theta^*_{N+1} = A\left(\theta^*_N, u_{N+1}, y_{N+1}\right) = \theta^*_N + K_{N+1}\left[y_{N+1} - \varphi\left(u_{N+1}\right)^T \theta^*_N\right]$$

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

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

Uczenie neuronu

Podobnie jak neuron Adaline (Adaptive Linear Neuron) $\phi(\kappa)$ - wzajemnie jednoznaczna



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

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

Uczenie: $\left[u_1\ u_2 \cdots u_N\right] = U_N \qquad \left[y_1\ y_2 \cdots y_N\right] = Y_N$

$$Q_N(\theta) = \sum_{n=1}^{N} \left(\kappa_n - \bar{\kappa}_n\right)^2 = \sum_{n=1}^{N} \left(\phi^{-1}(y_n) - \theta^T \varphi(u_n)\right)^2$$ To już było

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

lub rekurencyjnie

Algorytm rekurencyjny

$$\theta^*_{N+1} = A\left(\theta^*_N, u_{N+1}, y_{N+1}\right) = \theta^*_N + K_{N+1}\left[\phi^{-1}\left(y_{N+1}\right) - \varphi\left(u_{N+1}\right)^T \theta^*_N\right]$$

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

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

Uczenie neuronu
reguła delta

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

Numeryczna metoda optymalizacji (to będzie)

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

$$\theta^*_{n+1} = \theta^*_n - \eta \nabla_\theta Q(\theta^*_n), \quad \theta^*_0$$ η – współczynnik uczenia

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

$$\theta^*_{n+1} = \theta^*_n + \eta\left(y_{n+1} - \theta^{*T}_n \varphi(u_{n+1})\right)\varphi(u_{n+1}), \quad \theta^*_0$$

$$\Delta_n = \left(y_{n+1} - \theta^{*T}_n \varphi(u_{n+1})\right)$$

$$\theta^*_{n+1} = \theta^*_n + \eta \Delta_n \varphi(u_{n+1})$$

Uczenie neuronu
reguła delta

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

ϕ – funkcja aktywacji, różniczkowalna

Numeryczna metoda optymalizacji (to będzie)

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

$$\theta^*_{n+1} = \theta^*_n - \eta \nabla_\theta Q\left(\theta^*_n\right), \quad \theta^*_0$$

η – współczynnik uczenia

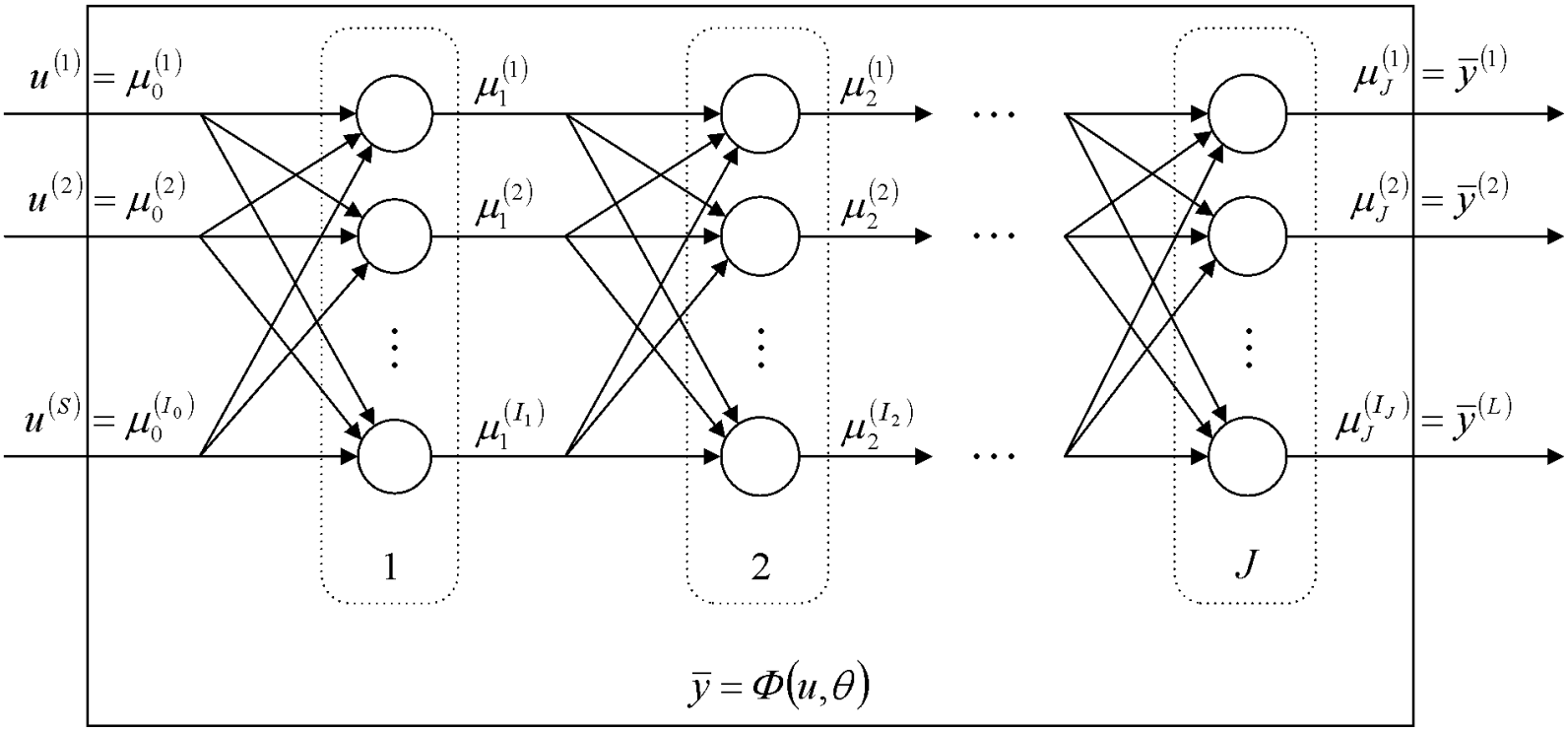
$$\nabla Q(\theta) = -\left(y - \phi\left(\theta^T \varphi(u)\right)\right)\frac{\partial \phi(\kappa)}{\partial(\kappa)}\varphi(u), \quad \theta_0$$

$$\theta^*_{n+1} = \theta^*_n + \eta\left(y_{n+1} - \phi\left(\theta^{*T}_n \varphi\left(u_{n+1}\right)\right)\right)\frac{\partial \phi(\kappa)}{\partial \kappa}\varphi\left(u_{n+1}\right), \quad \theta^*_0$$

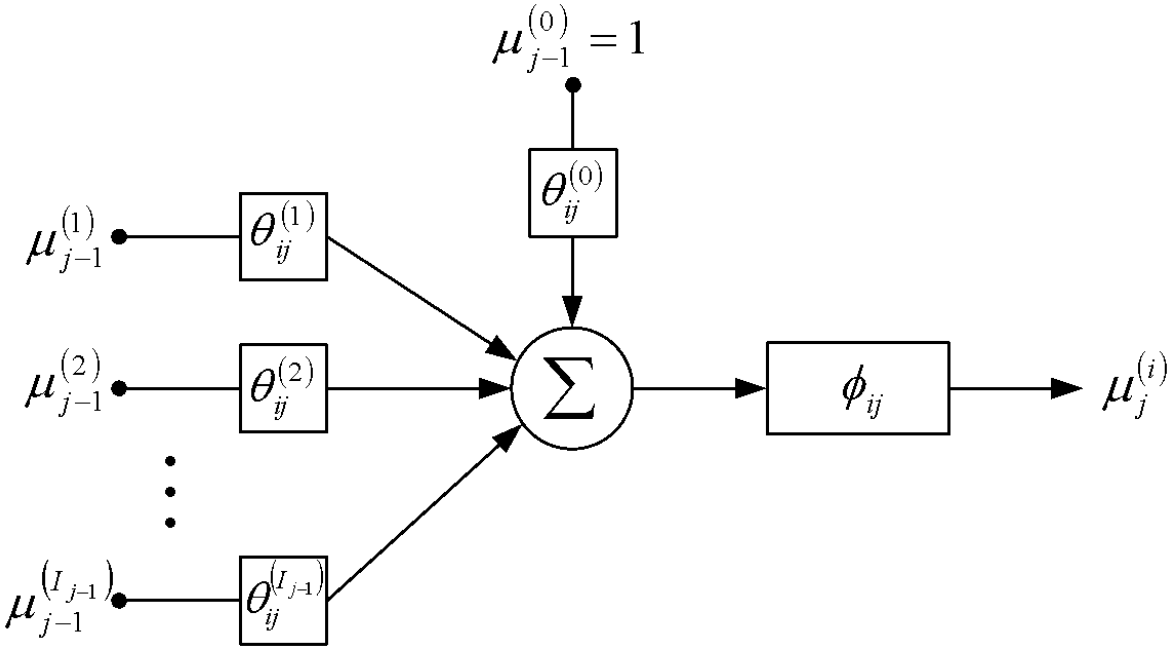
$$\Delta_n = \left(y_{n+1} - \phi\left(\theta^{*T}_n \varphi\left(u_{n+1}\right)\right)\right)\frac{\partial \phi(\kappa)}{\partial \kappa}$$

$$\theta^*_{n+1} = \theta^*_n + \eta \Delta_n \varphi\left(u_{n+1}\right)$$

Sieć wielowarstwowa



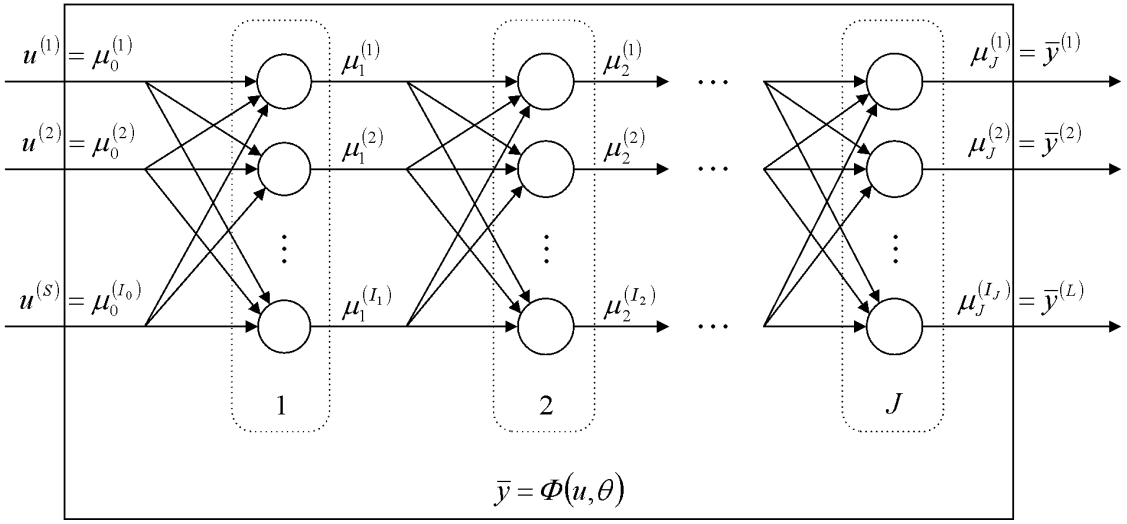
Model neuronu (wyrwany z sieci)



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

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

Sieć wielowarstwowa



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

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

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

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

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

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

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

$$\theta \overset{\text{df}}{=} \{\theta_1, \theta_2, \ldots, \theta_J\}.$$

$$Q(\theta) = \sum_{n=1}^{N} Q_n(\theta) = \sum_{n=1}^{N} [y_n - \Phi(u_n, \theta)]^T [y_n - \Phi(u_n, \theta)] = \sum_{n=1}^{N} \sum_{l=1}^{L} \left(y_n^{(l)} - \Phi^{(l)}(u_n, \theta)\right)^2$$

Uczenie sieci wielowarstwowej

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

$$\frac{\partial Q_n(\theta)}{\partial \theta_{ij}^{(s)}}\bigg|_{\theta_{ij}^{(s)} = \tilde{\tilde{\theta}}_{ij,n}^{(s)}} = -2\delta_{jn}^{(i)} \mu_{j-1,n}^{(s)},$$

$$\tilde{\tilde{\theta}}_{ij,n+1}^{(s)} = \tilde{\tilde{\theta}}_{ij,n}^{(s)} - \eta_n \frac{\partial Q_n(\theta)}{\partial \theta_{ij}^{(s)}}\bigg|_{\theta_{ij}^{(s)} = \tilde{\tilde{\theta}}_{ij,n}^{(s)}}, \quad s = 0, 1, \ldots, I_{J-1},\ i = 1, 2, \ldots, I_J,\ j = 1, 2, \ldots, J,$$

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

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

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

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

Optymalizacja funkcjonału

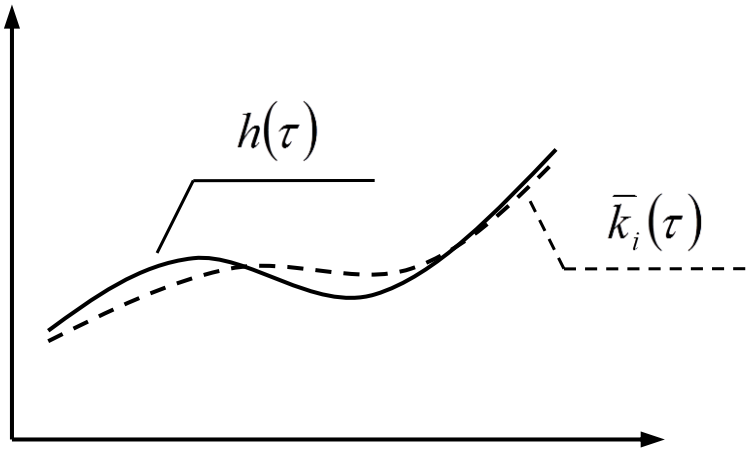
$$x^*(t) \rightarrow \mathcal{F}\left(x^*(t)\right) = \min_{x(t)} \mathcal{F}\left(x(t)\right)$$

Wariacja funkcjonału:

dla $x(t) + \delta h(t)$ $\mathcal{F}\left(x(t) + \delta h(t)\right)$

Warunek konieczny

$$\left.\frac{\partial \mathcal{F}\left(x(t) + \delta h(t)\right)}{\partial \delta}\right|_{\delta = 0} = 0, \ \forall h(t)$$



Optymalizacja funkcjonału przykład

$$\mathscr{F}\left(x(t)\right)=\int_{t_a}^{t_b}\left(x^2(t)-2x(t)\right)dt$$

$$\mathscr{F}\left(x(t)+\delta h(t)\right)=\int_{t_a}^{t_b}\left(\left(x(t)+\delta h(t)\right)^2-2\left(x(t)+\delta h(t)\right)\right)dt$$

$$\left.\frac{\partial\mathscr{F}\left(x(t)+\delta h(t)\right)}{\partial\delta}\right|_{\delta=0}=0,\ \forall h(\tau)$$

$$\left.\frac{\partial\mathscr{F}\left(x(t)+\delta h(t)\right)}{\partial\delta}\right|_{\delta=0}=2\int_{t_a}^{t_b}\left(x(t)+\delta h(t)-1\right)h(t)dt\Bigg|_{\delta=0}=0\quad\forall h(t)$$

$$\left(x(t)-1\right)=0\quad\Rightarrow\quad x(t)=1,\quad t\in\left[t_a,t_b\right]$$

Optymalizacja funkcjonału z ograniczeniami - przykład

$$x^*(t) \to \mathscr{F}\left(x^*(t)\right) = \min_{x(t) \in \mathscr{D}\left(x(t)\right)} \mathscr{F}\left(x(t)\right)$$

Przy ograniczeniach

$$\mathscr{D}\left(x(t)\right) = \left\{x(t); \mathscr{G}_m\left(x(t)\right) = 0,\ m = 1, 2, \ldots, M\right\}$$

Funkcjonał Lagrange'a

$$\mathscr{L}\left(x(t), \lambda\right) = \mathscr{F}\left(x(t)\right) + \sum_{l=1}^{L} \lambda_l \mathscr{G}_l\left(x(t)\right)$$

Wariacja Funkcjonału Lagrange'a

$$\mathscr{L}\left(x(t) + \delta h(t), \lambda\right) = \mathscr{F}\left(x(t) + \delta h(t)\right) + \sum_{l=1}^{L} \lambda_l \mathscr{G}_l\left(x(t) + \delta h(t)\right)$$

Warunek konieczny

$$\left.\frac{\partial \mathscr{L}\left(x(t) + \delta h(t)\right)}{\partial \delta}\right|_{\delta = 0} = 0,\ \forall h(t)$$

Optymalizacja funkcjonału z ograniczeniami - przykład

$$\mathscr{F}\left(x(t)\right) = y(t_b) = \int_{t_a}^{t_b} k_i(t_a - t)x(t)dt$$

$$\mathscr{D}\left(x(t)\right) = \left\{ x(t), t \in [t_a, t_b]; \mathscr{G}\left(x(t)\right) = \int_{t_a}^{t_b} x^2(t)dt - E = 0, \right\}$$

$$\mathscr{L}\left(x(t), \lambda\right) = \int_{t_a}^{t_b} k_i(t_a - t)x(t)dt + \int_{t_a}^{t_b} x^2(t)dt - E$$

$$\mathscr{L}\left(x(t) + \delta h(t), \lambda\right) = \int_{t_a}^{t_b} k_i(t_a - t)\left(x(t) + \delta h(t)\right)dt$$

$$+ \lambda \int_{t_a}^{t_b} \left(x(t) + \delta h(t)\right)^2 dt - E$$

Optymalizacja funkcjonału z ograniczeniami - przykład

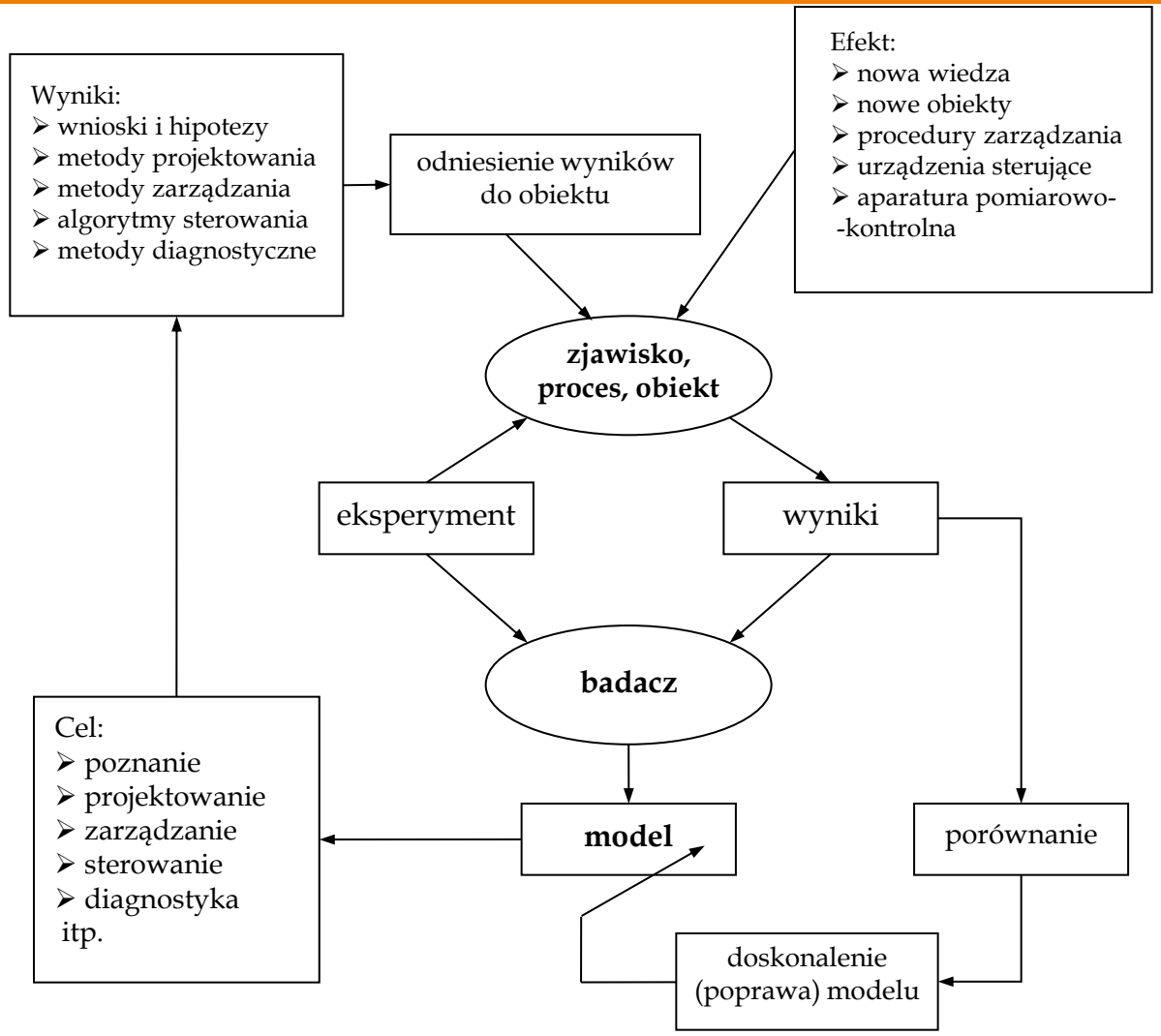
$$\left.\frac{\partial \mathscr{L}\left(x(t)+\delta h(t)\right)}{\partial \delta}\right|_{\delta=0} = 2\int_{t_a}^{t_b}\left(k_i\left(t_a-t\right)+2\lambda\left(x(t)+\delta h(t)\right)h(t)\right)dt\Bigg|_{\delta=0} = 0 \quad \forall h(t)$$

$$k_i\left(t_a-t\right)+2\lambda x(t)=0 \quad \Rightarrow \quad x(t)=-\frac{1}{2\lambda}k_i\left(t_a-t\right), \quad t\in\left[t_a,t_b\right]$$

$$\int_{t_a}^{t_b}x^2(t)dt-E=0-E=0 \Rightarrow \int_{t_a}^{t_b}\left(-\frac{1}{2\lambda}k_i\left(t_a-t\right)\right)^2 dt=E$$

$$\lambda^*=\pm\frac{1}{2\sqrt{E}}\int_{t_a}^{t_b}k_i{}^2\left(t_a-t\right)dt, \qquad x(t)=\mp\frac{\sqrt{E}}{\int_{t_a}^{t_b}k_i{}^2\left(t_a-t\right)}k_i\left(t_a-t\right), \quad t\in\left[t_a,t_b\right]$$

Model w badaniach systemowych



Dziękuję za uwagę