POLITECHNIKA GDAŃSKA

Modelowanie i identyfikacja

Materiał wykładowy: 11 – Sieci neuronowe w modelowaniu i identyfikacji

Kierunek: Automatyka i robotyka - studia stacjonarne 2 stopnia

Przedmiot: kierunkowy

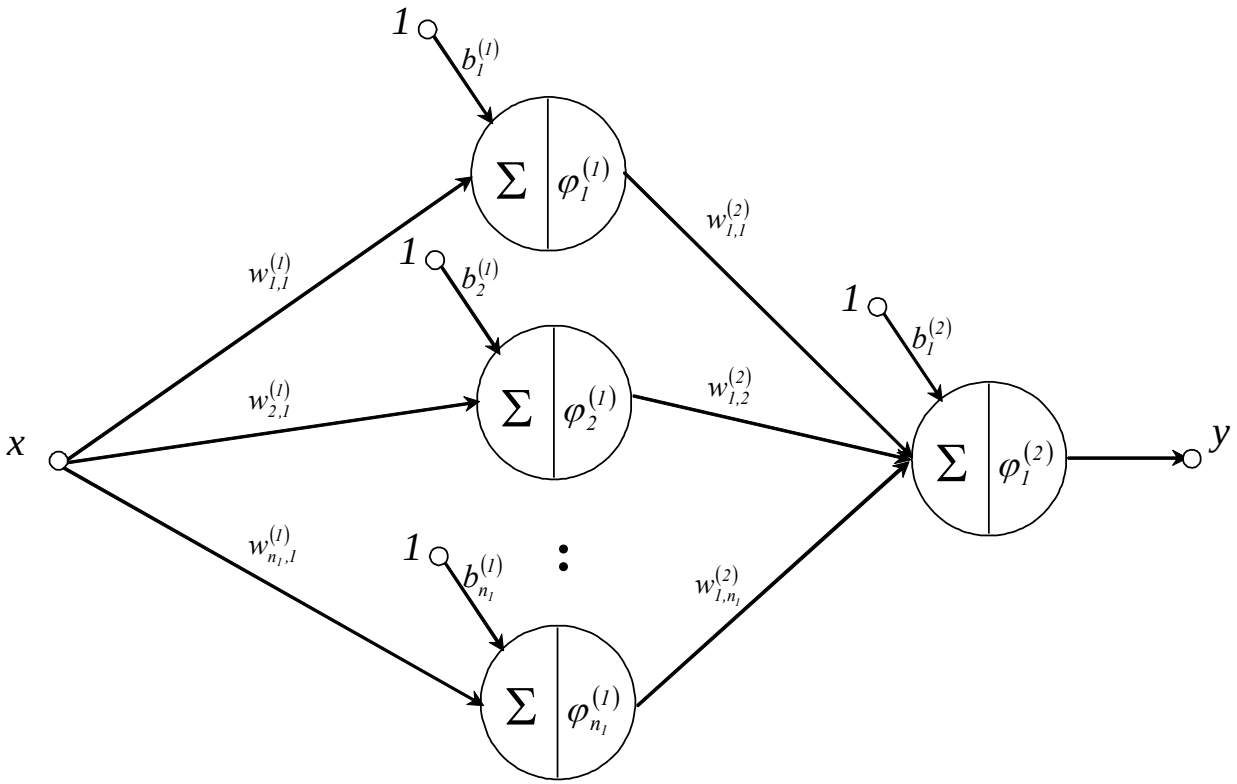
Kazimierz Duzinkiewicz, dr hab. inż., prof. nadzw. PG

Data rozpoczęcia prezentacji materiału: 2019.05.30

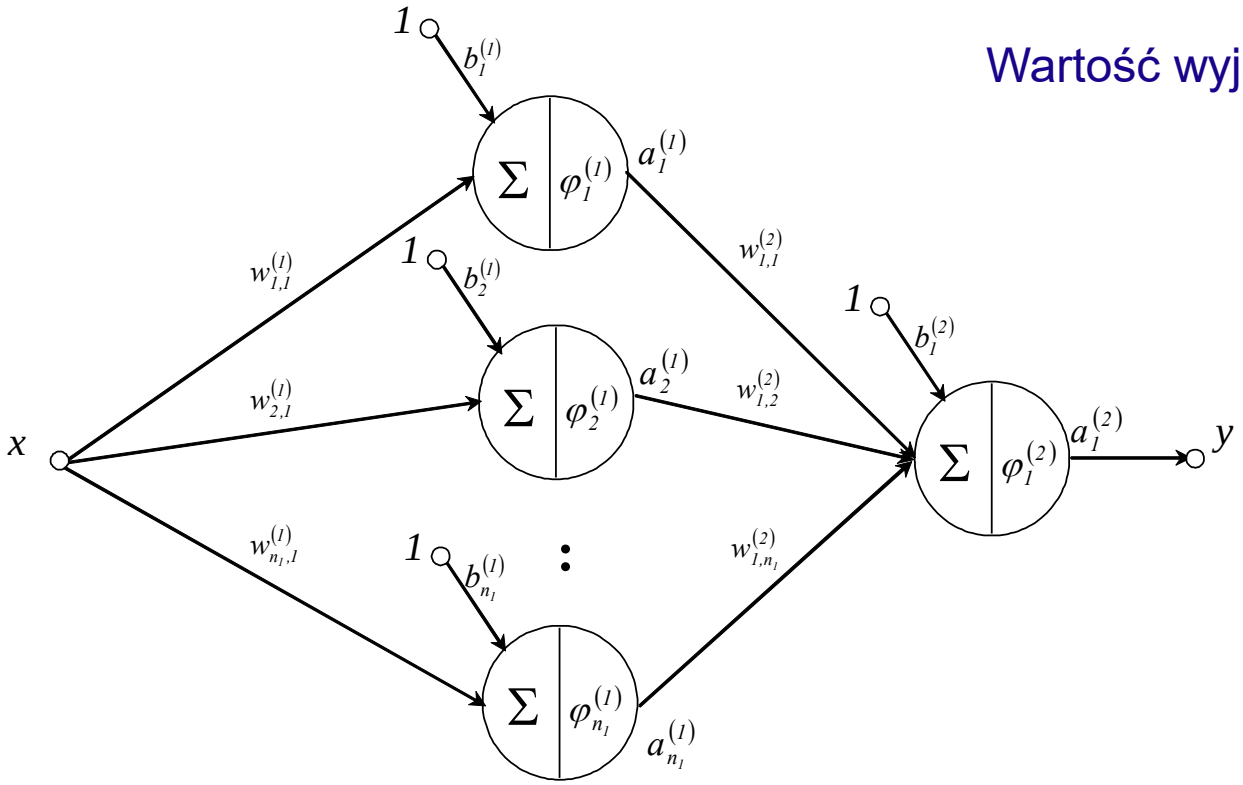
W modelowaniu i identyfikacji wykorzystywane są zdolności sztucznych sieci neuronowych do aproksymacji zależności pomiędzy zmiennymi

Rozpocznijmy od prostego przykładu

Dla uproszczenia, skupmy się dalej na sieciach o jednym wyjściu i jednym wejściu (SISO)



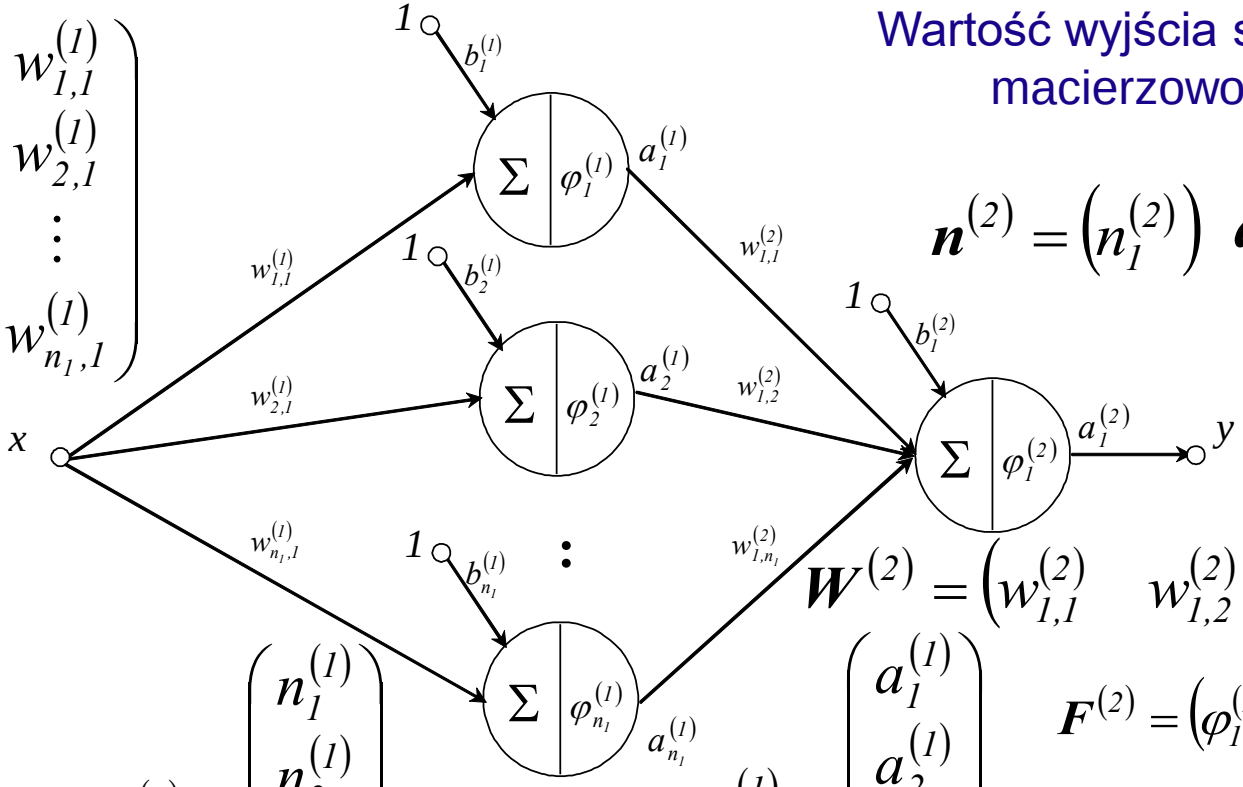
Wartość wyjścia sieci



$$y = a_1^{(2)} = \varphi_1^{(2)}\left(n_1^{(2)}\right) = \varphi_1^{(2)}\left(\left(\sum_{j_1=1}^{n_1} w_{1,j_1}^{(2)} \cdot a_{j_1}^{(1)}\right) + b_1^{(2)}\right) = \varphi_1^{(2)}\left(\left(\sum_{j_1=1}^{n_1} w_{1,j_1}^{(2)} \cdot \varphi_{j_1}^{(1)}\left(n_{j_1}^{(1)}\right)\right) + b_1^{(2)}\right)$$

$$= \varphi_1^{(2)}\left(\left(\sum_{j_1=1}^{n_1} w_{1,j_1}^{(2)} \cdot \varphi_{j_1}^{(1)}\left(w_{j_1,1}^{(1)} \cdot x + b_{j_1}^{(1)}\right)\right) + b_1^{(2)}\right)$$

Wartość wyjścia sieci - macierzowo



$$\boldsymbol{W}^{(1)} = \begin{pmatrix} w_{1,1}^{(1)} \\ w_{2,1}^{(1)} \\ \vdots \\ w_{n_1,1}^{(1)} \end{pmatrix}$$

$$\boldsymbol{n}^{(2)} = \left(n_1^{(2)}\right) \quad \boldsymbol{a}^{(2)} = \left(a_1^{(2)}\right)$$

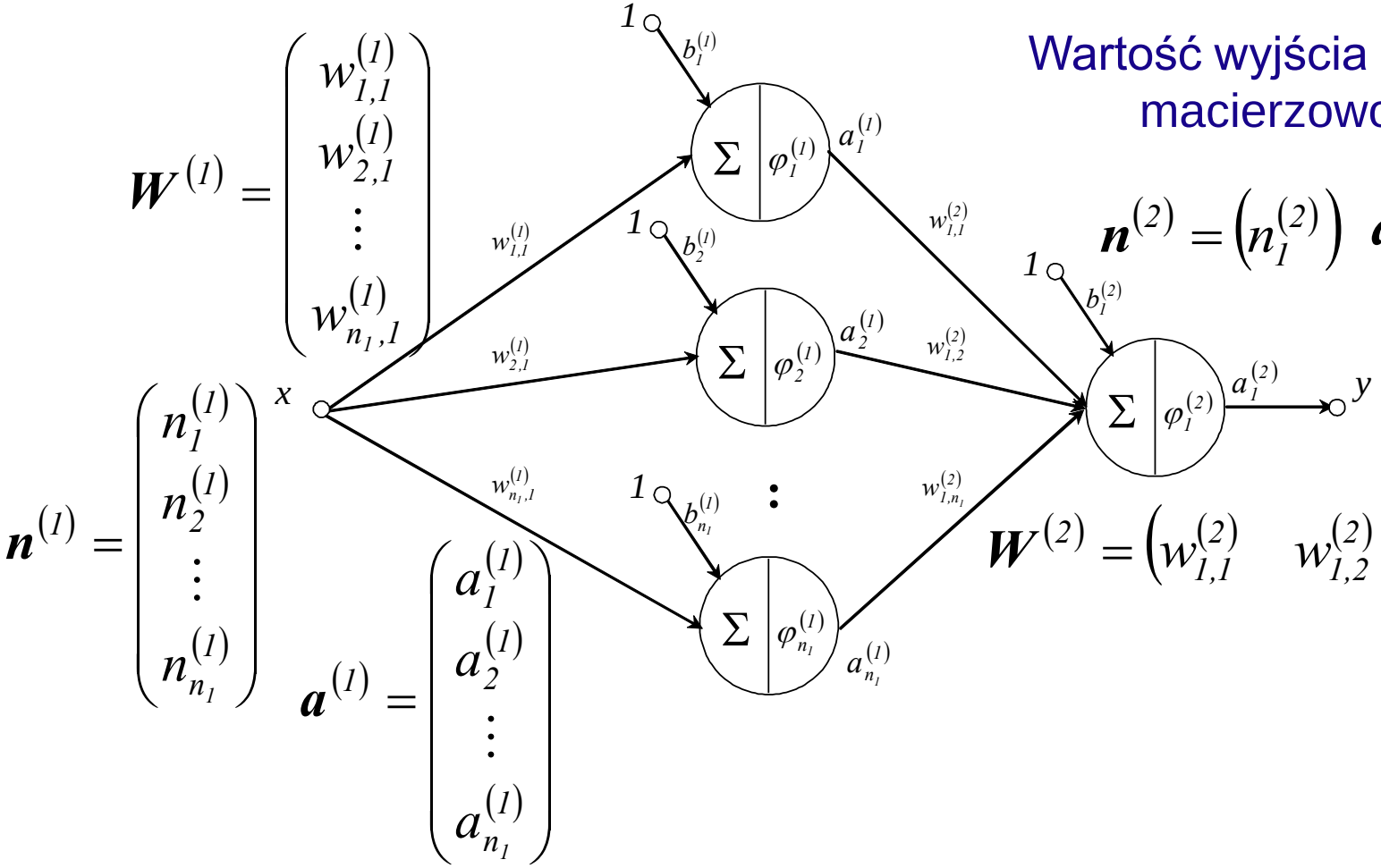
$$\boldsymbol{W}^{(2)} = \begin{pmatrix} w_{1,1}^{(2)} & w_{1,2}^{(2)} & \cdots & w_{1,n_1}^{(2)} \end{pmatrix}$$

$$\boldsymbol{F}^{(2)} = \left(\varphi_1^{(2)}\right)$$

$$\boldsymbol{n}^{(1)} = \begin{pmatrix} n_1^{(1)} \\ n_2^{(1)} \\ \vdots \\ n_{n_1}^{(1)} \end{pmatrix} \qquad \boldsymbol{a}^{(1)} = \begin{pmatrix} a_1^{(1)} \\ a_2^{(1)} \\ \vdots \\ a_{n_1}^{(1)} \end{pmatrix}$$

$$\boldsymbol{F}^{(1)} = \begin{pmatrix} \varphi_1^{(1)} & & & \\ & \varphi_2^{(1)} & & \\ & & \ddots & \\ & & & \varphi_{n_1}^{(1)} \end{pmatrix}$$

Wartość wyjścia sieci - macierzowo



$$\boldsymbol{W}^{(1)} = \begin{pmatrix} w_{1,1}^{(1)} \\ w_{2,1}^{(1)} \\ \vdots \\ w_{n_1,1}^{(1)} \end{pmatrix}$$

$$\boldsymbol{n}^{(1)} = \begin{pmatrix} n_1^{(1)} \\ n_2^{(1)} \\ \vdots \\ n_{n_1}^{(1)} \end{pmatrix} \qquad \boldsymbol{a}^{(1)} = \begin{pmatrix} a_1^{(1)} \\ a_2^{(1)} \\ \vdots \\ a_{n_1}^{(1)} \end{pmatrix}$$

$$\boldsymbol{n}^{(2)} = \left(n_1^{(2)}\right) \quad \boldsymbol{a}^{(2)} = \left(a_1^{(2)}\right)$$

$$\boldsymbol{W}^{(2)} = \begin{pmatrix} w_{1,1}^{(2)} & w_{1,2}^{(2)} & \cdots & w_{1,n_1}^{(2)} \end{pmatrix}$$

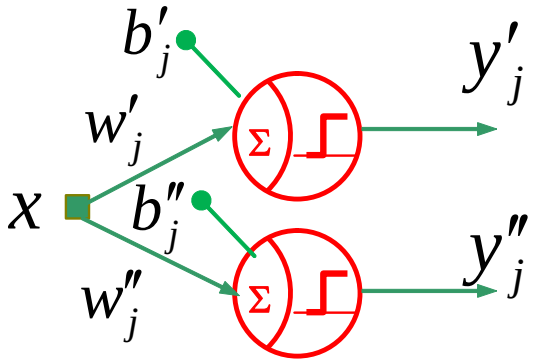
$$\boldsymbol{y} = \boldsymbol{a}^{(2)} = \boldsymbol{F}^{(2)}\left(\boldsymbol{n}^{(2)}\right) = \boldsymbol{F}^{(2)}\left(\boldsymbol{W}^{(2)}\boldsymbol{a}^{(1)} + \boldsymbol{b}^{(2)}\right) = \boldsymbol{F}^{(2)}\left(\boldsymbol{W}^{(2)}\boldsymbol{F}^{(1)}\left(\boldsymbol{n}^{(1)}\right) + \boldsymbol{b}^{(2)}\right)$$
$$= \boldsymbol{F}^{(2)}\left(\boldsymbol{W}^{(2)}\boldsymbol{F}^{(1)}\left(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}\right) + \boldsymbol{b}^{(2)}\right)$$

Zbudujmy przykładową sieć perceptronową, tzn. sieć jednokierunkową z funkcją pobudzenia SUMA i funkcją aktywacji PRZEKAŹNIK NIESYMETRYCZNY

Założenie:

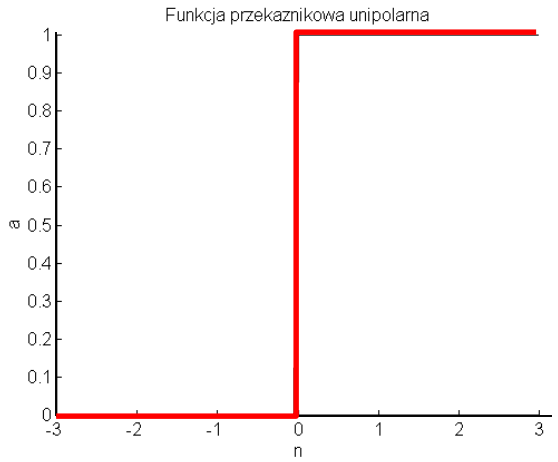
- wiemy jak przetwarza docierające do niego sygnały neuron o podanej charakterystyce , w szczególności jak aktualne wartości wag i progów dla poszczególnych neuronów wpływają na ich działanie

Utworzenie komórki elementarnej jednowarstwowej



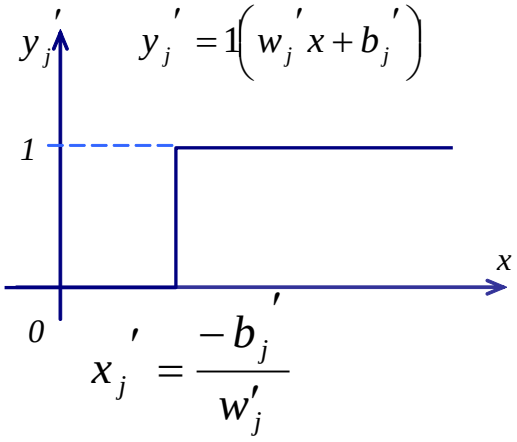
Elementarna komórka jednowarstwowa o numerze j (dwuwarstwowa)

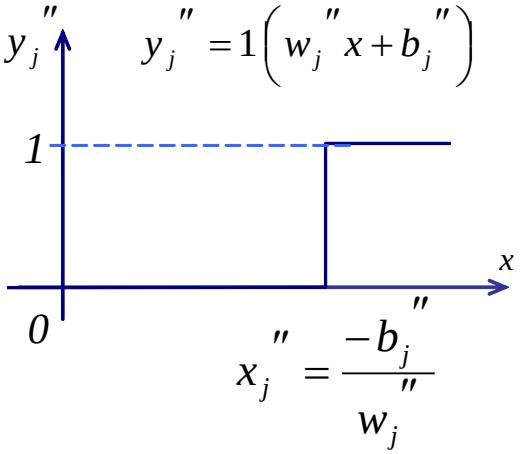
- Funkcja pobudzenia: funkcja sumy
- Funkcja aktywacji: funkcja przekaźnikowa niesymetryczna



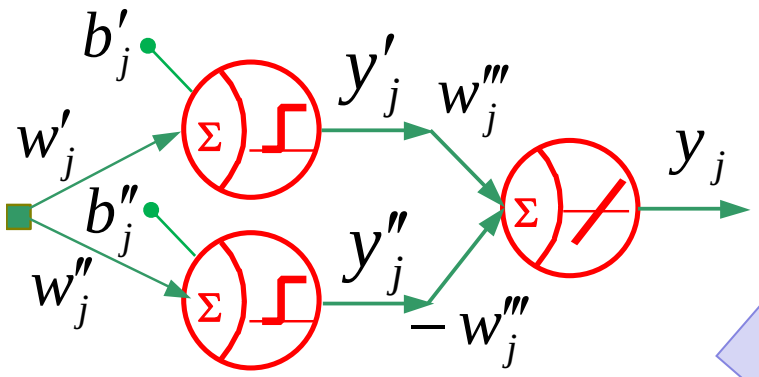
$$y = \begin{cases} 0 & gdy \quad n < 0 \\ 1 & gdy \quad n \geq 0 \end{cases}$$

$$n = wx + b \Rightarrow y = \begin{cases} 0 & gdy \quad x < \dfrac{-b}{w} \\ 1 & gdy \quad x \geq \dfrac{-b}{w} \end{cases}$$

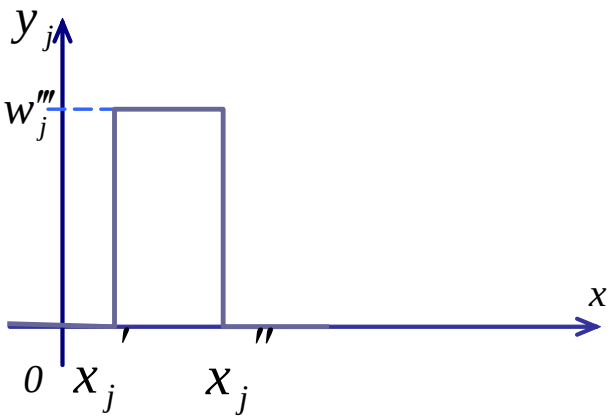




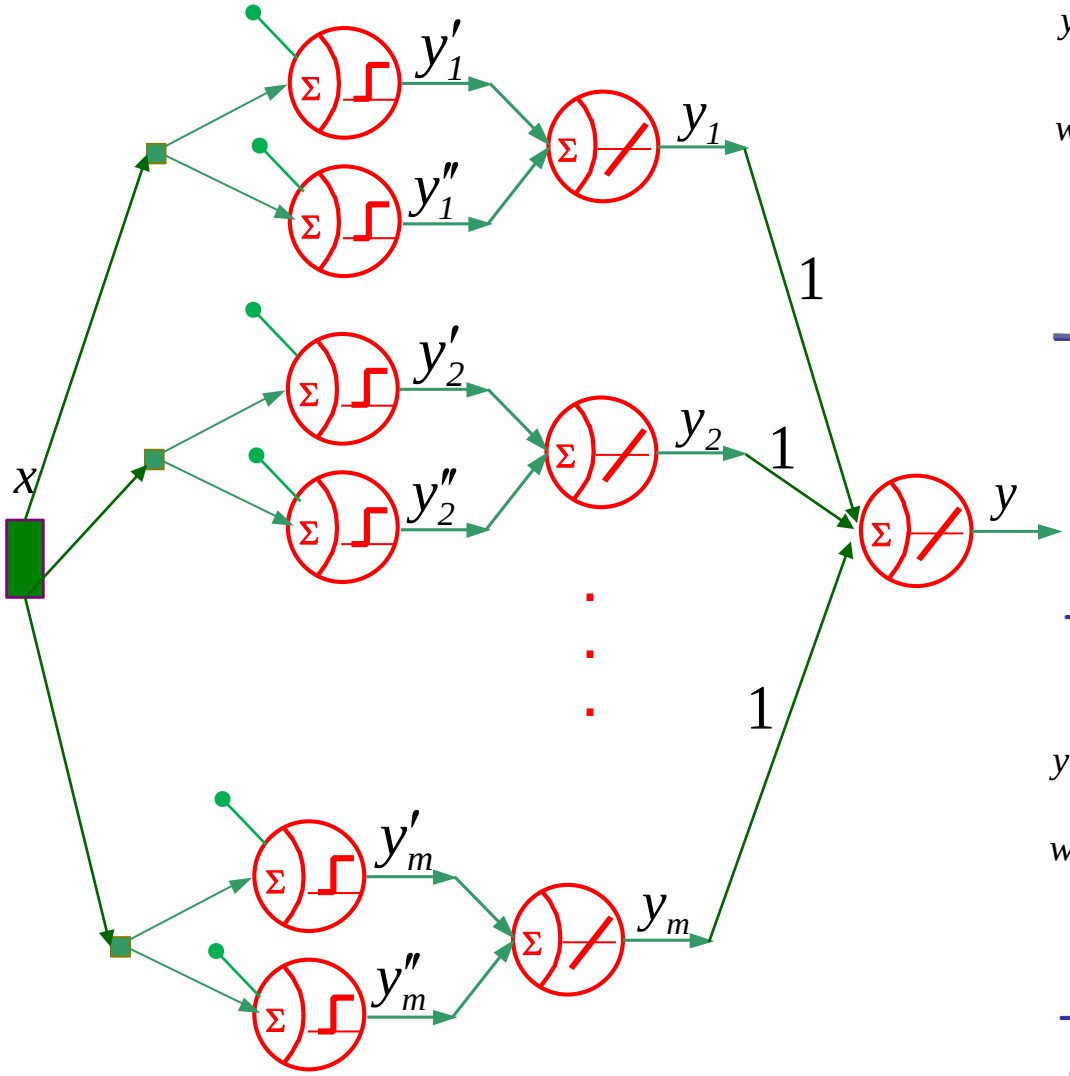
Utworzenie komórki elementarnej dwuwarstwowej

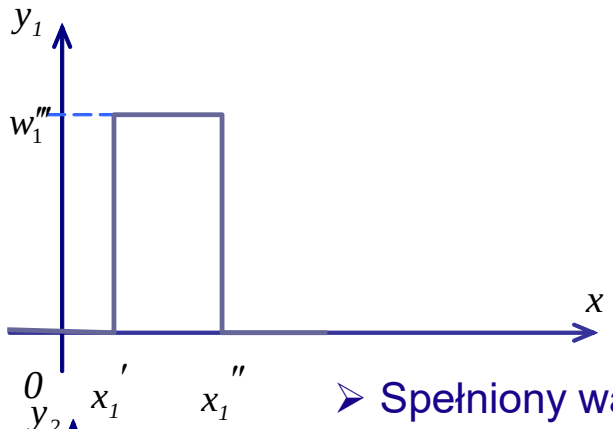


$$w'''_j = y_j$$



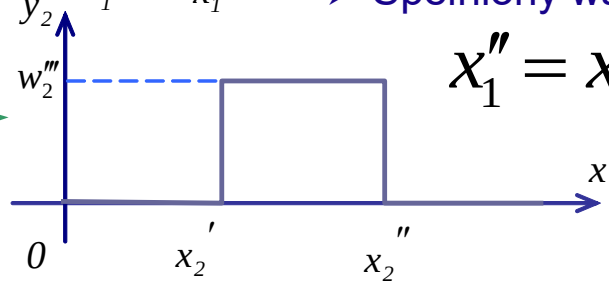
Połączenie m komórek elementarnych dwuwarstwowych





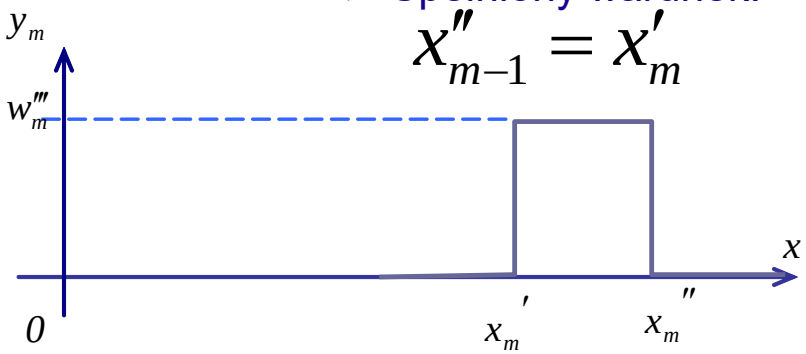
➢ Spełniony warunek:

$$x''_1 = x'_2$$

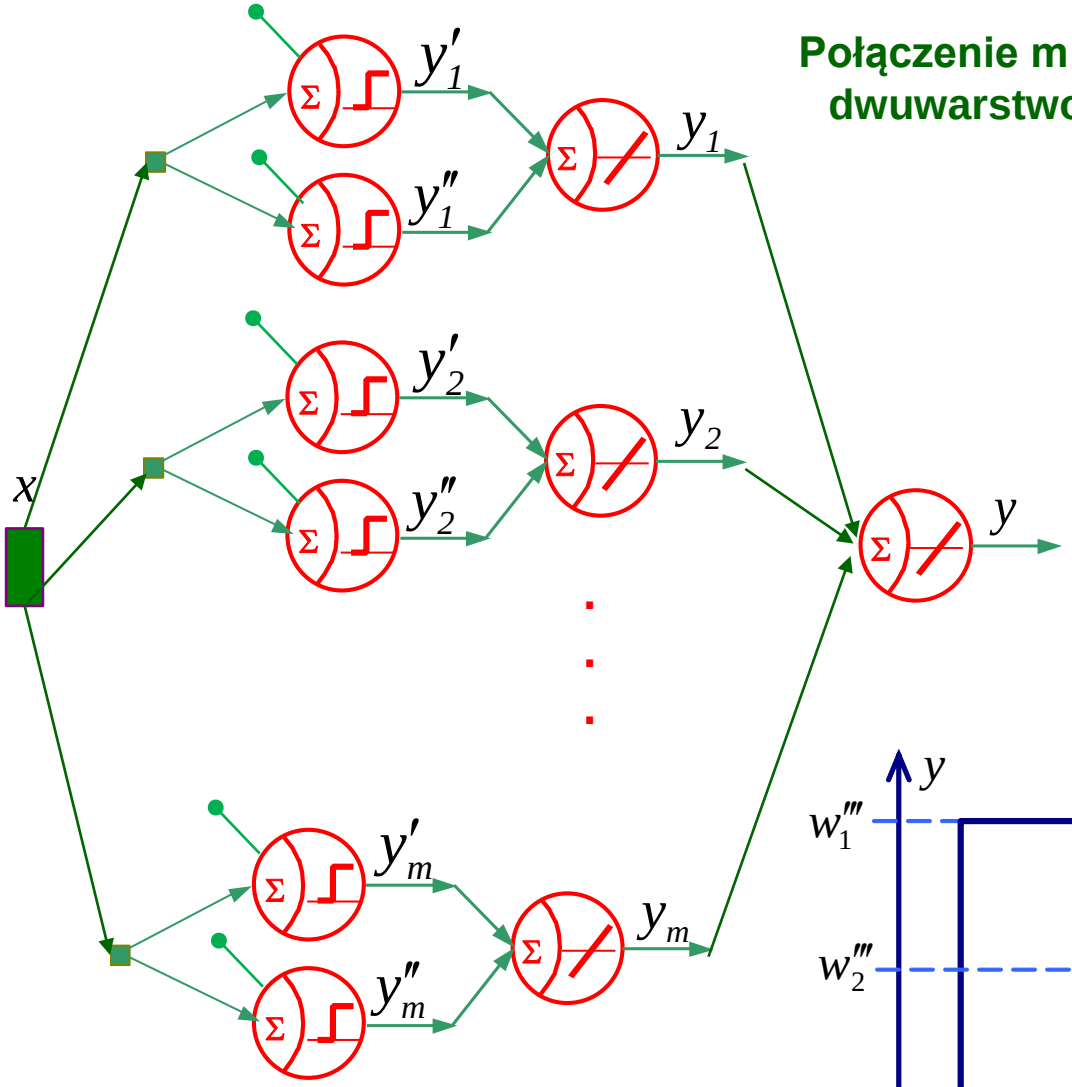


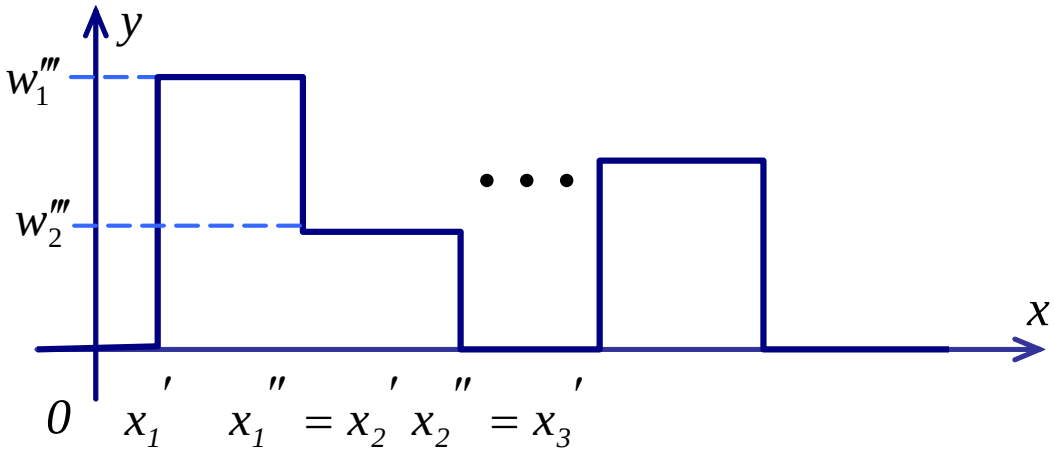
➢ Spełniony warunek:

$$x''_{m-1} = x'_m$$

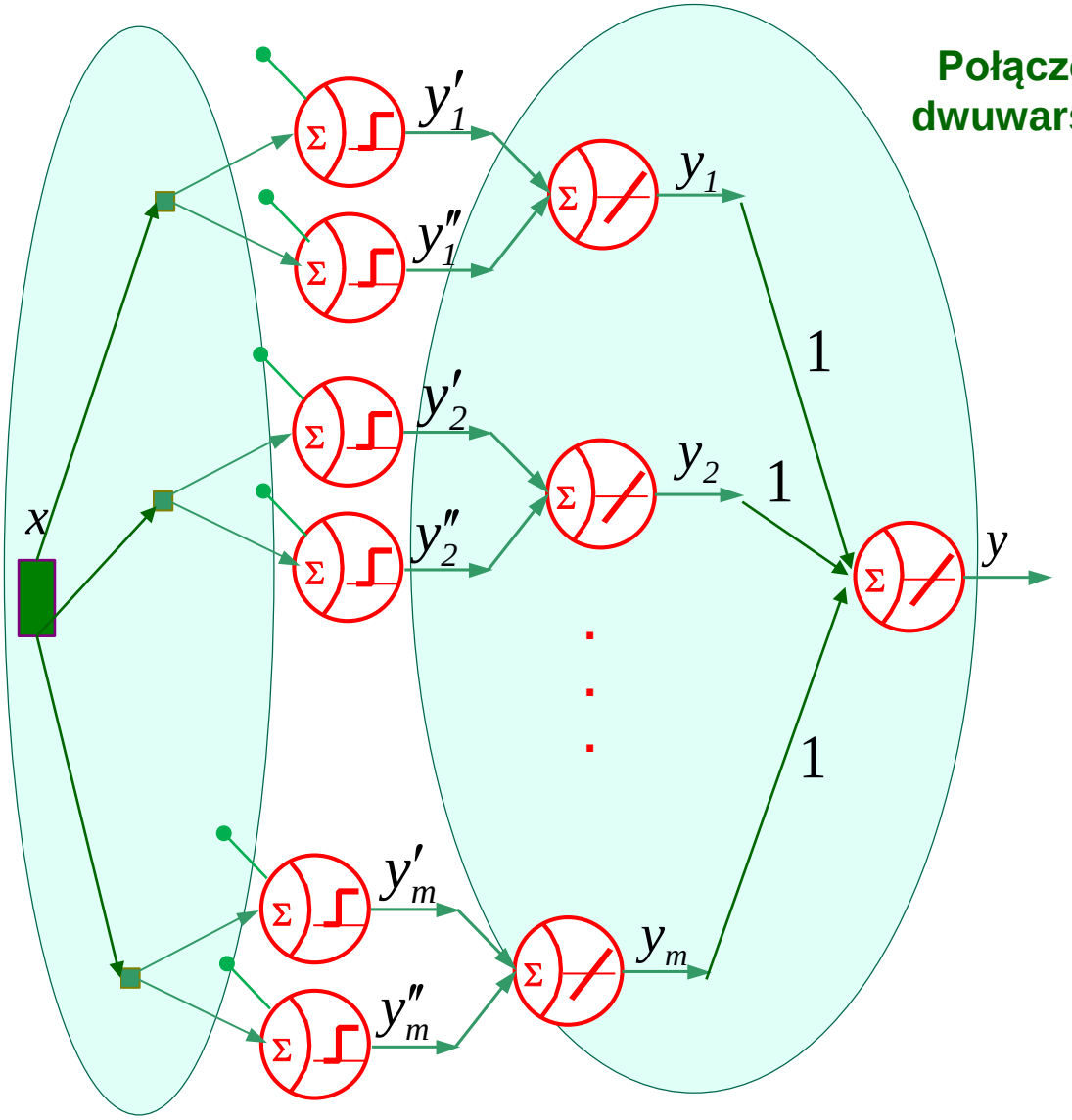


Połączenie m komórek elementarnych dwuwarstwowych – wynik końcowy

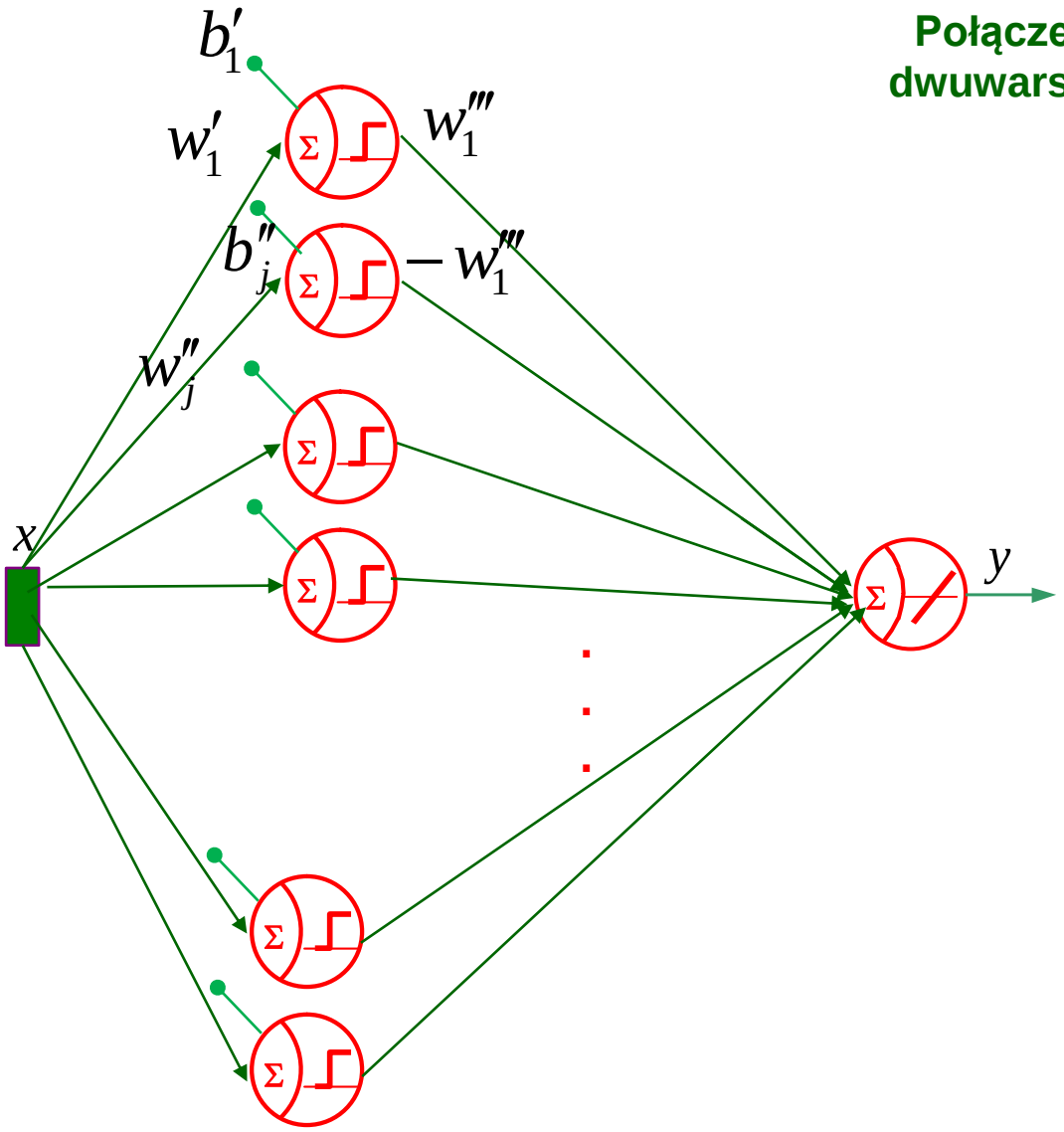




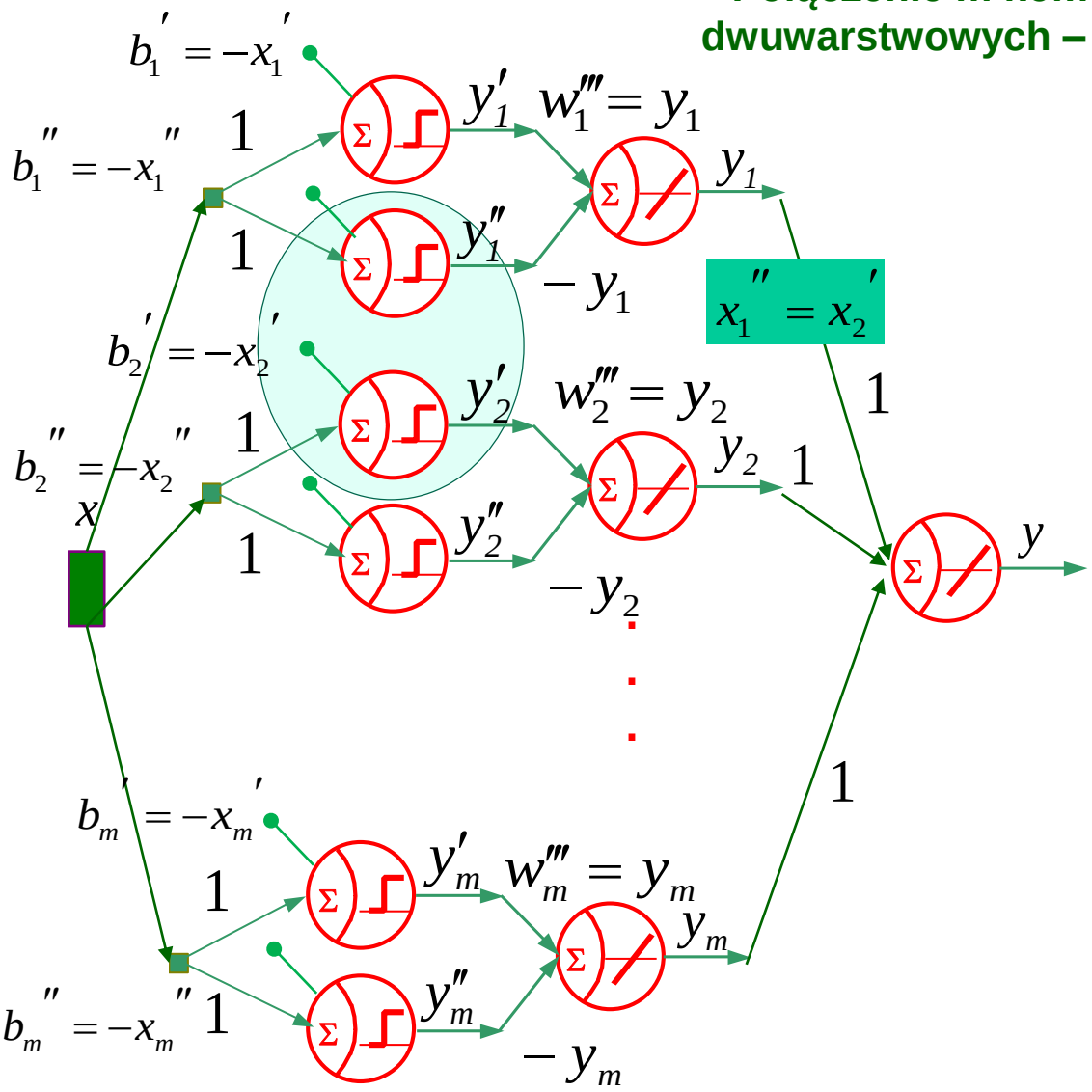
Połączenie m komórek elementarnych dwuwarstwowych – próba uproszczenia I



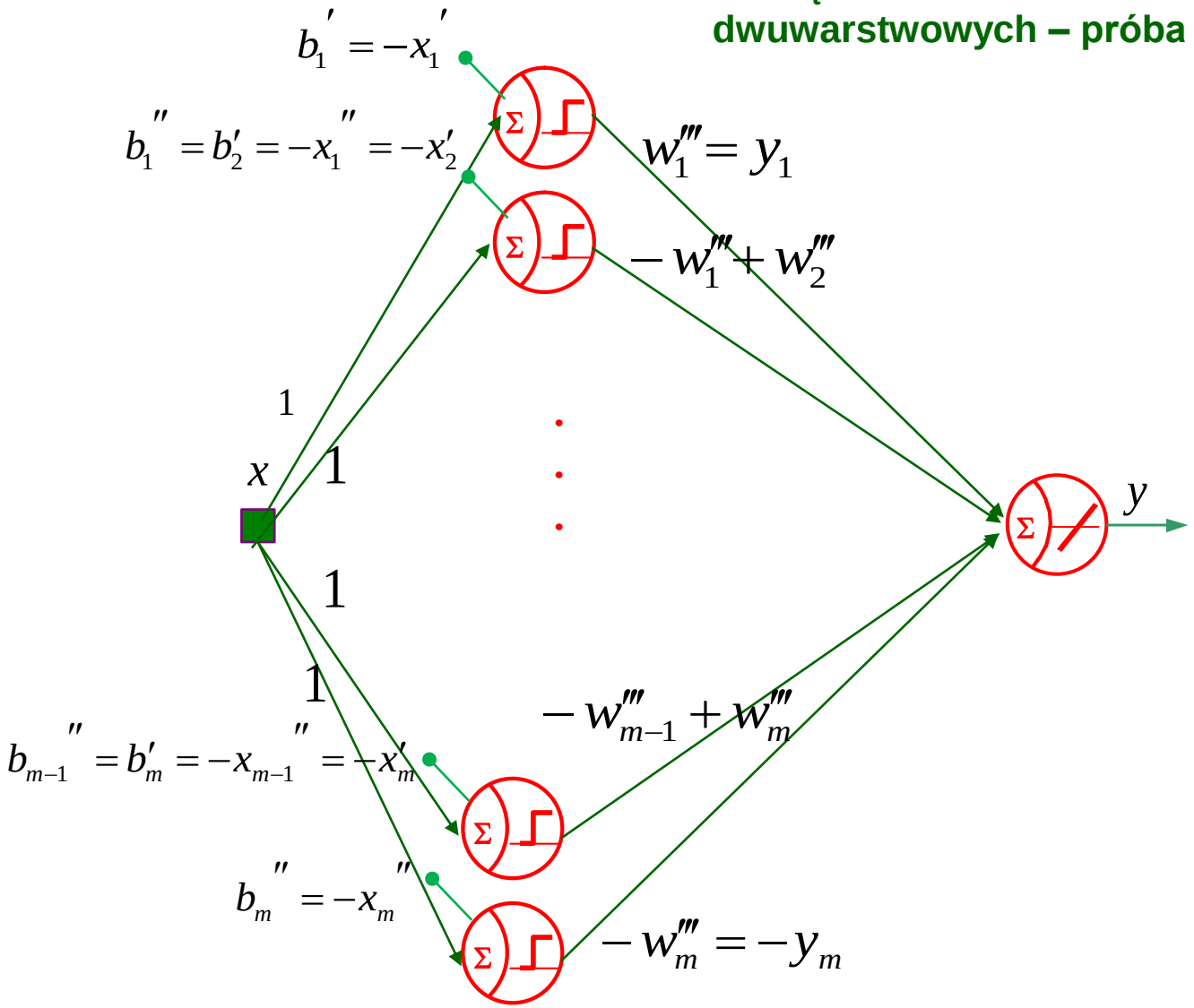
Połączenie m komórek elementarnych dwuwarstwowych – próba uproszczenia I



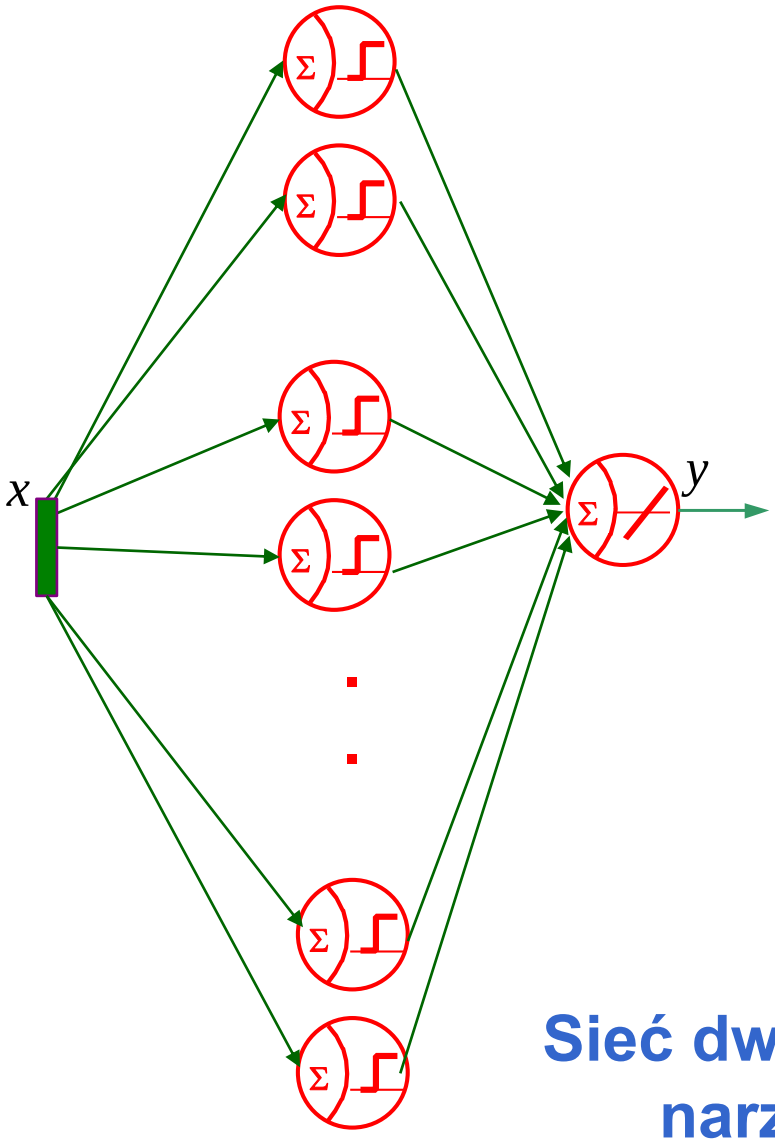
Połączenie m komórek elementarnych dwuwarstwowych – próba uproszczenia II

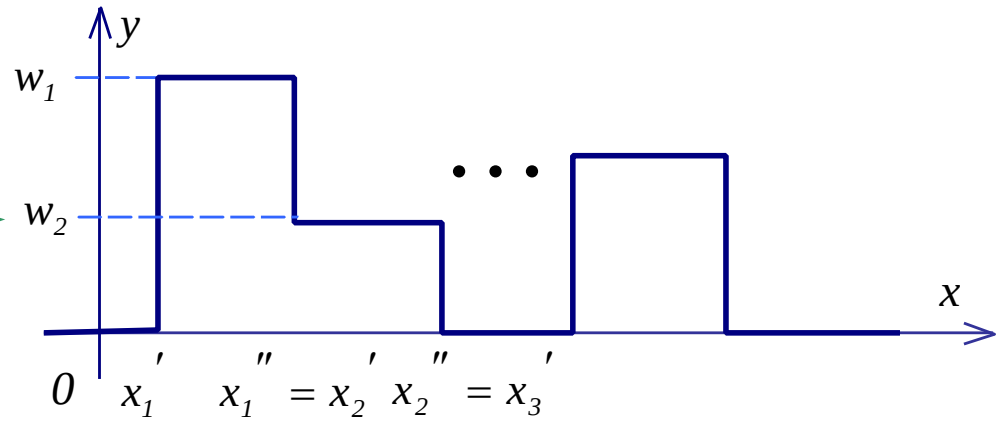


Połączenie m komórek elementarnych dwuwarstwowych – próba uproszczenia II



Uproszczenia i wynik końcowy





Sieć dwuwarstwowa (trójwarstwowa)

Sieć dwuwarstwowa (trójwarstwowa) – narzędzie aproksymacji funkcji

Przykład

Weźmy dwuwarstwową (trójwarstwową) sieć o strukturze 1-2-1

Liczba wejść

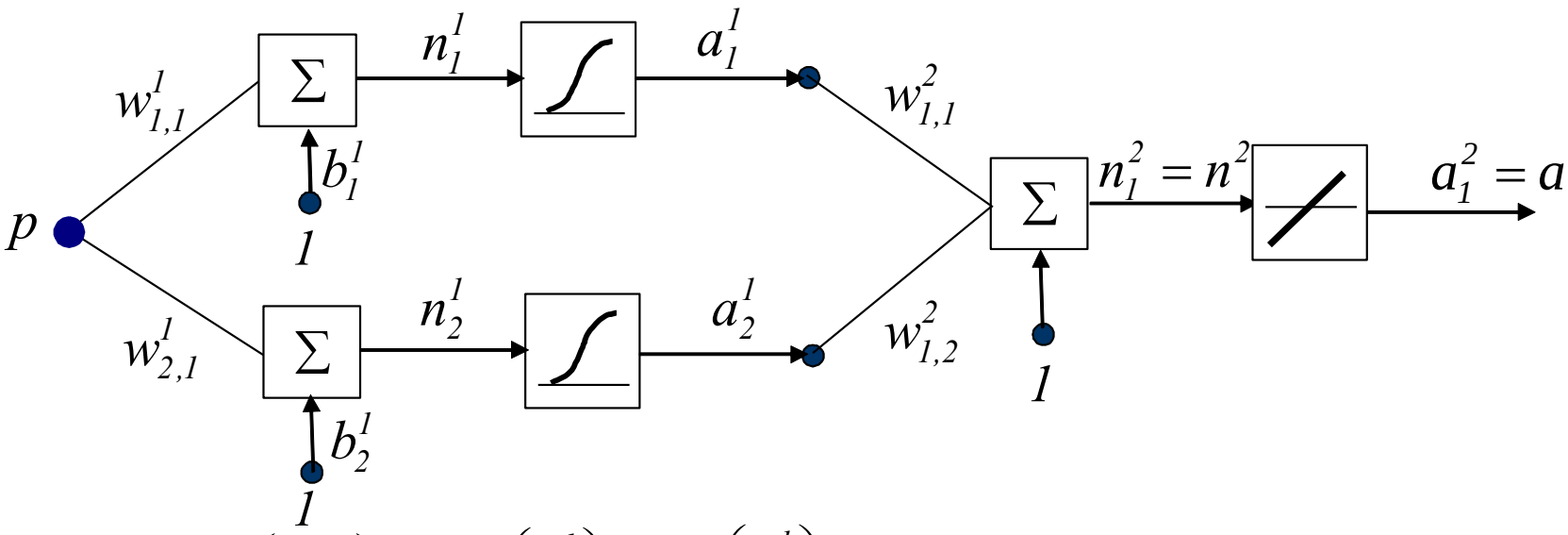
Liczba neuronów warstwy 1

Liczba neuronów warstwy 2 – liczba wyjść

Funkcje aktywacji kolejnych warstw są następujące

$$f^1\left(n^1(:)\right)=\frac{1}{1+e^{-n^1(:)}};\quad f^2\left(n^2(:)\right)=n^2(:)$$

Struktura sieci



$$\boldsymbol{p} = (p) \quad \boldsymbol{W}^1 = \begin{pmatrix} w^1_{1,1} \\ w^1_{2,1} \end{pmatrix} \quad \boldsymbol{n}^1 = \begin{pmatrix} n^1_1 \\ n^1_2 \end{pmatrix} \quad \boldsymbol{a}^1 = \begin{pmatrix} a^1_1 \\ a^1_2 \end{pmatrix} \quad \boldsymbol{W}^2 = \begin{pmatrix} w^2_{1,1} & w^2_{1,2} \end{pmatrix} \quad \boldsymbol{n}^2 = \begin{pmatrix} n^2_1 \end{pmatrix} \quad \boldsymbol{a}^2 = \begin{pmatrix} a^2_1 \end{pmatrix}$$

$$\boldsymbol{b}^1 = \begin{pmatrix} b^1_1 \\ b^1_2 \end{pmatrix} \qquad \boldsymbol{b}^2 = \begin{pmatrix} b^2_1 \end{pmatrix}$$

$$\boldsymbol{a}^1 = \boldsymbol{logsig}\left(\boldsymbol{W}^1 p + \boldsymbol{b}^1\right) \qquad \boldsymbol{a}^2 = purelin\left(\boldsymbol{W}^2 \boldsymbol{a}^1 + b^2\right)$$

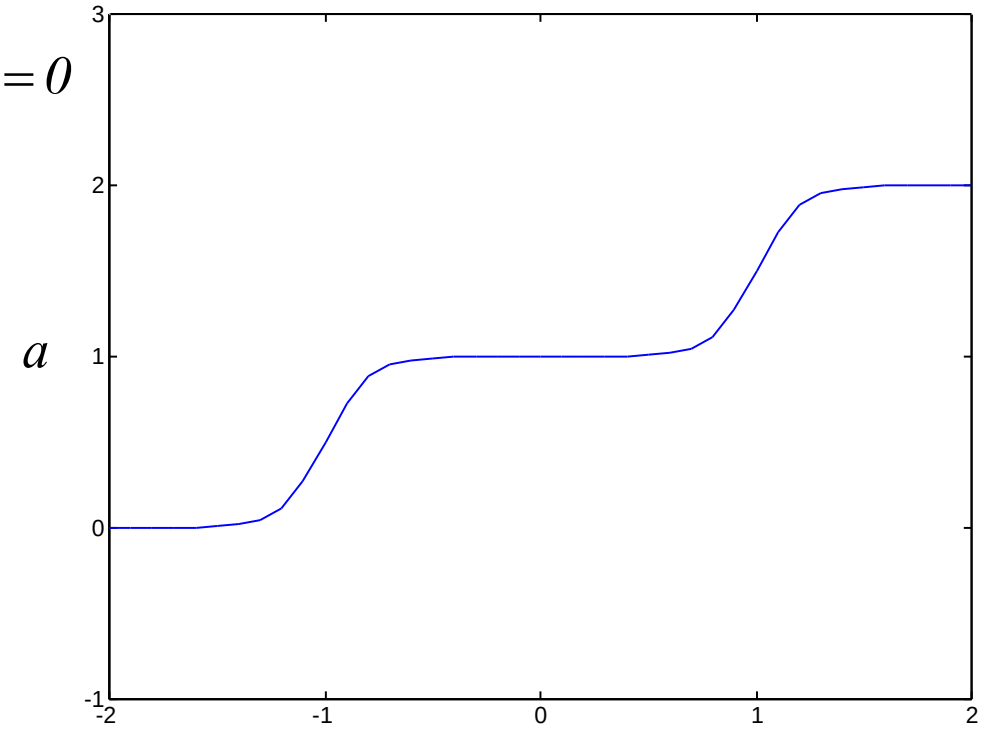
Bieżące wartości wag i progów

$$\boldsymbol{W}^1 = \begin{pmatrix} w_{1,1}^1 \\ w_{2,1}^1 \end{pmatrix} = \begin{pmatrix} 10 \\ 10 \end{pmatrix} \qquad \boldsymbol{W}^2 = \begin{pmatrix} w_{1,1}^2 & w_{1,2}^2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}$$

$$\boldsymbol{b}^1 = \begin{pmatrix} b_1^1 \\ b_2^1 \end{pmatrix} = \begin{pmatrix} -10 \\ 10 \end{pmatrix} \qquad \boldsymbol{b}^2 = \begin{pmatrix} b_1^2 \end{pmatrix} = 0$$

Zakres zmian wejścia sieci

$$\boldsymbol{p} = \begin{pmatrix} p \end{pmatrix} \quad p \in \begin{bmatrix} -2 & 2 \end{bmatrix}$$



$$p \in \begin{bmatrix} -2 & 2 \end{bmatrix}$$

Składowe odpowiedzi sieci

$$\boldsymbol{n}^1 = \begin{pmatrix} n_1^1 \\ n_2^1 \end{pmatrix} = \begin{pmatrix} w_{1,1}^1 p + b_1^1 \\ w_{2,1}^1 p + b_2^1 \end{pmatrix} = \begin{pmatrix} 10p - 10 \\ 10p + 10 \end{pmatrix}$$

$$\boldsymbol{a}^1 = \begin{pmatrix} a_1^1 \\ a_2^1 \end{pmatrix} = \begin{pmatrix} log\,sig\left(n_1^1\right) \\ log\,sig\left(n_2^1\right) \end{pmatrix} = \begin{pmatrix} log\,sig\left(w_{1,1}^1 p + b_1^1\right) \\ log\,sig\left(w_{2,1}^1 p + b_2^1\right) \end{pmatrix} = \begin{pmatrix} log\,sig\left(10p - 10\right) \\ log\,sig\left(10p + 10\right) \end{pmatrix}$$

$$\begin{aligned} \boldsymbol{n}^2 = \left(n_1^2\right) &= \left(w_{1,1}^2 a_1^1 + w_{1,2}^2 a_2^1 + b_1^2\right) \\ &= \left(w_{1,1}^2\, log\,sig\left(w_{1,1}^1 p + b_1^1\right) + w_{1,2}^2\, log\,sig\left(w_{2,1}^1 p + b_2^1\right) + b_1^2\right) \\ &= \left(w_{1,1}^2\, log\,sig\left(10p - 10\right) + w_{1,2}^2\, log\,sig\left(10p + 10\right) + b_1^2\right) \\ &= \left(1 \cdot log\,sig\left(10p - 10\right) + 1 \cdot log\,sig\left(10p + 10\right)\right) \end{aligned}$$

$$\begin{aligned} \boldsymbol{a}^2 = \left(a_1^2\right) = a = n_1^2 &= w_{1,1}^2\, log\,sig\left(w_{1,1}^1 p + b_1^1\right) + w_{1,2}^2\, log\,sig\left(w_{2,1}^1 p + b_2^1\right) + b_1^2 \\ &= 1 \cdot log\,sig\left(10p - 10\right) + 1 \cdot log\,sig\left(10p + 10\right) \end{aligned}$$

$$a = w^2_{1,1}\, log\, sig\left(w^1_{1,1}p + b^1_1\right) + w^2_{1,2}\, log\, sig\left(w^1_{2,1}p + b^1_2\right) + b^2_1$$

$$= 1 \cdot log\, sig\left(10p - 10\right) + 1 \cdot log\, sig\left(10p + 10\right)$$

Punkt środkowy funkcji sigmoidalnej pojedynczych neuronów

$$n = wp + b = 0$$

Neuron 1, warstwa 1

$$n^1_1 = w^1_{1,1}p + b^1_1 = 0$$

$$n^1_1 = 10p - 10 = 0 \Rightarrow p = 1$$

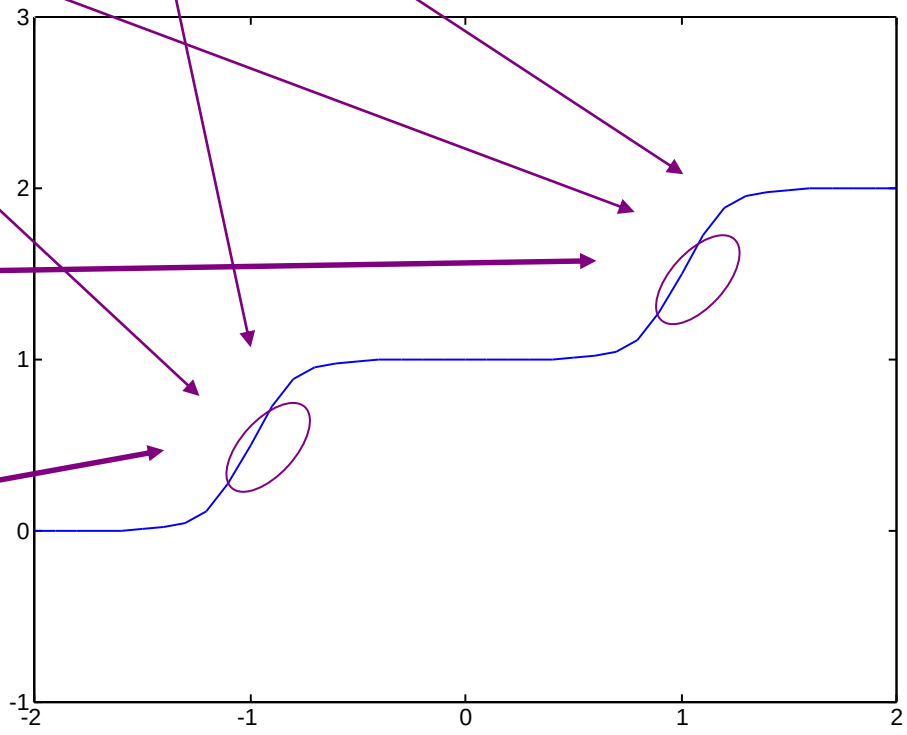
Neuron 2, warstwa 1

$$n^1_2 = w^1_{2,1}p + b^1_2 = 0$$

$$n^1_2 = 10p + 10 = 0 \Rightarrow p = -1$$

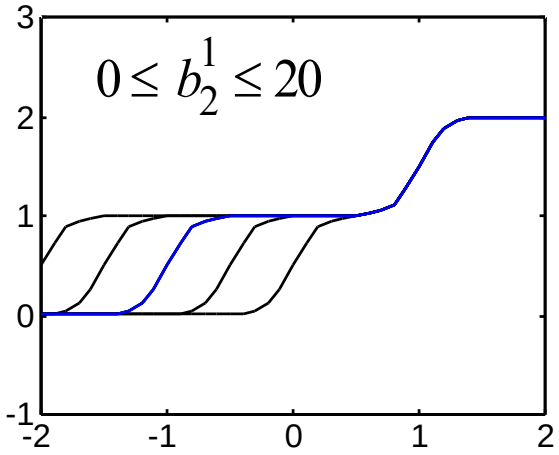
?

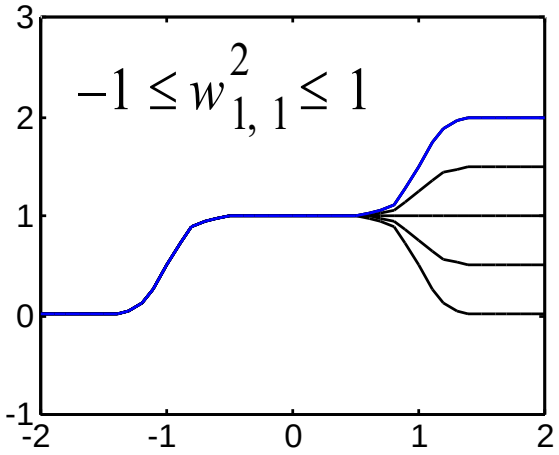
?

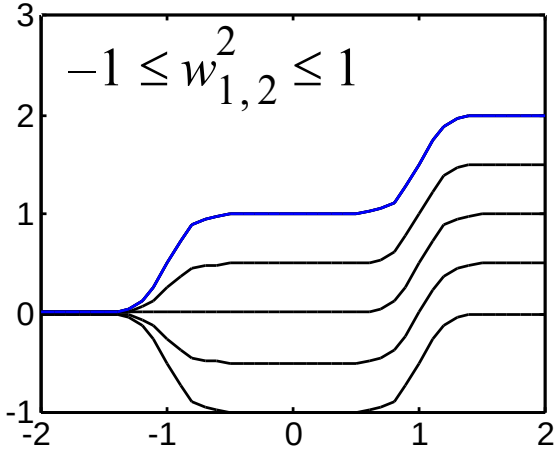


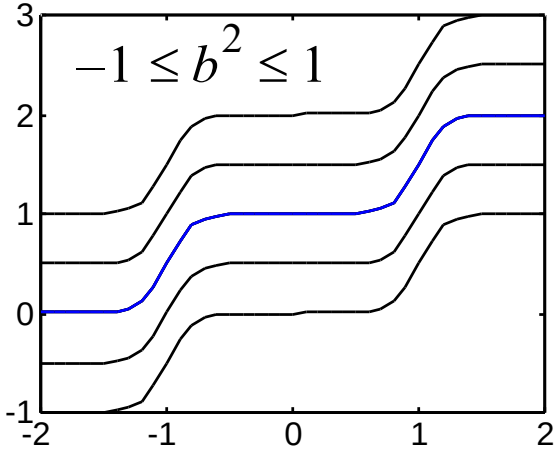
Zmiany wartości wag i progów i odpowiedź sieci

$$a = w^2_{1,1}\, log\, sig\left(w^1_{1,1}p + b^1_1\right) + w^2_{1,2}\, log\, sig\left(w^1_{2,1}p + b^1_2\right) + b^2_1 = log\, sig\left(10p - 10\right) + log\, sig\left(10p + 10\right)$$

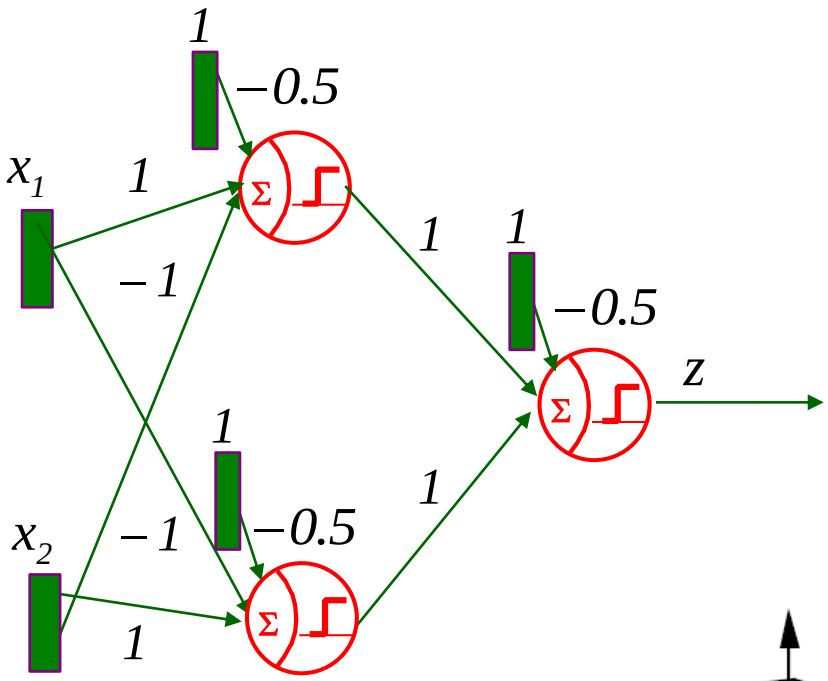


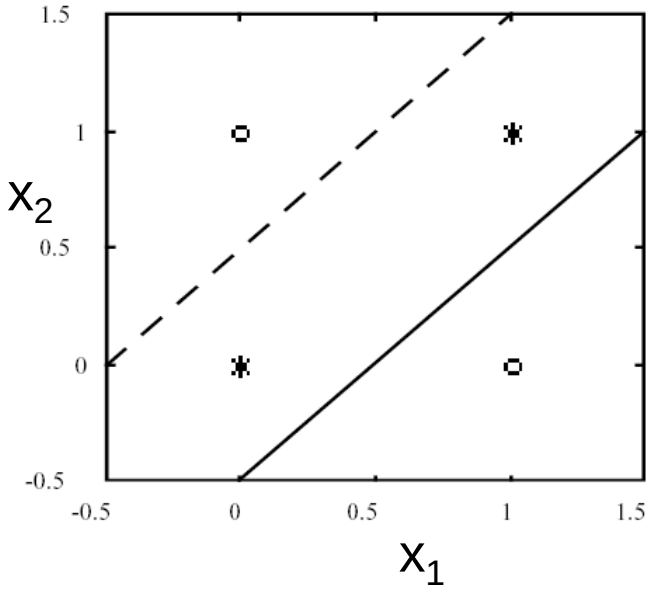


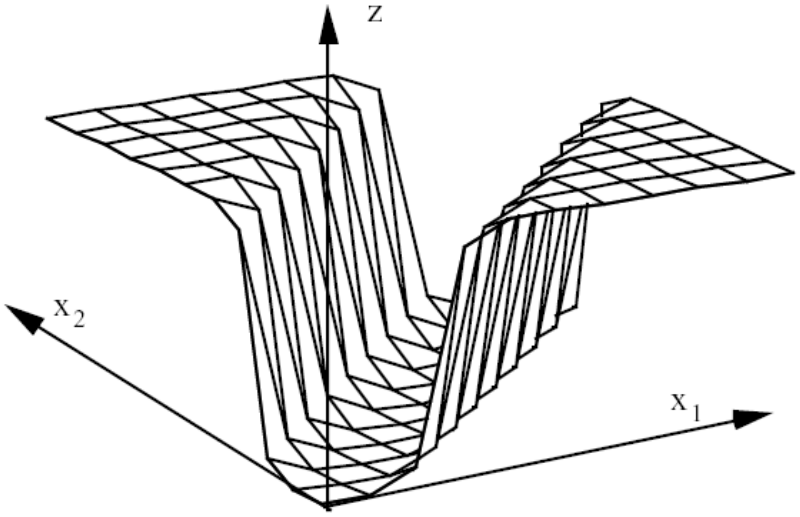


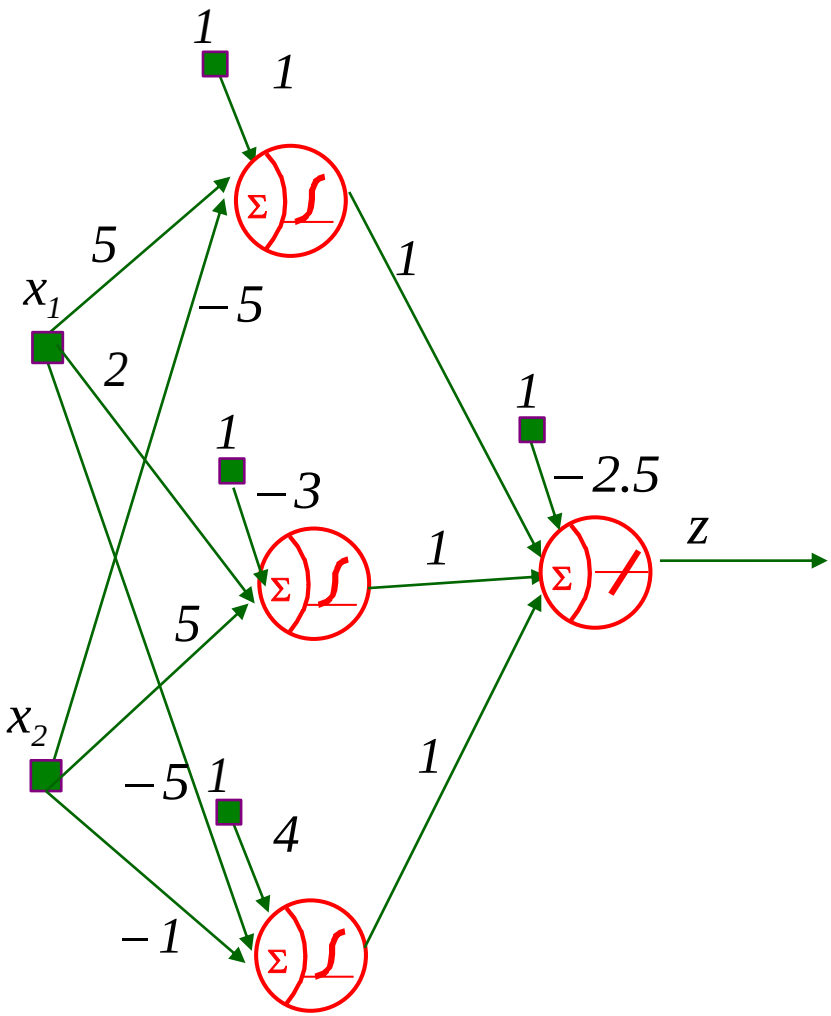


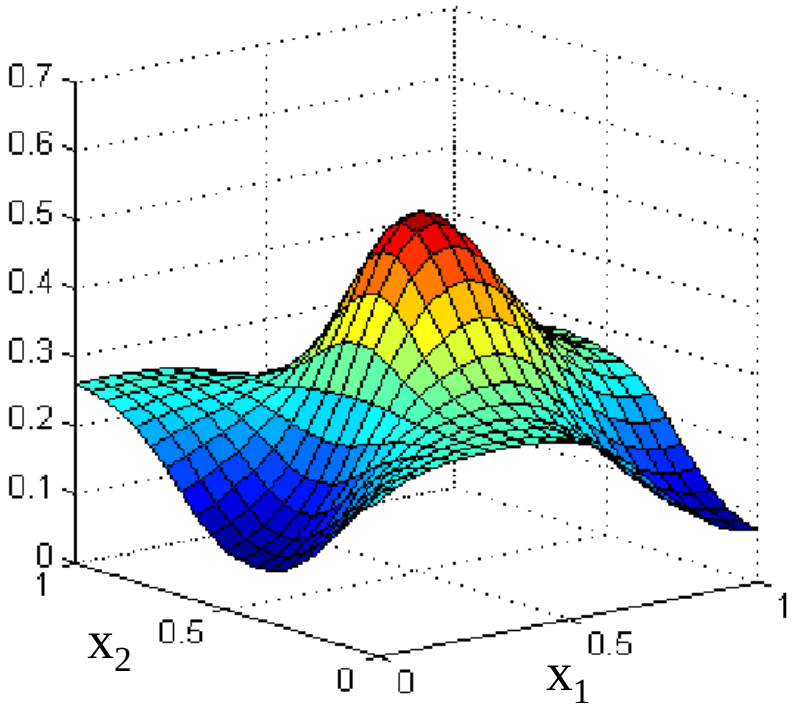
Przykłady – funkcje jednej zmiennej; a funkcje większej liczby zmiennych?








1
1
5
x_1
2
−5
1
1
−3
1
−2.5
z
5
x_2
−5
1
4
1
1
−1


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
x_2
0.5
0
0
0.5
1
x_1

Właściwości aproksymacyjne perceptronów wielowarstwowych

Odpowiednio skonstruowane sieci wielowarstwowe są uniwersalnymi aproksymatorami !

Podam twierdzenie, które zapewnia, że standardowa sieć wielowarstwowa z pojedynczą warstwą ukrytą składającą się z skończonej liczby neuronów jest uniwersalnym aproksymatorem

W 1989r. Funahashi udowodnił następujące twierdzenie

Niech $\phi(x)$ będzie niestałą, ograniczoną i monotonicznie rosnącą funkcją. Niech ponadto $K \subset R^n$ będzie zbiorem zwartym i

$$f : K \to R$$

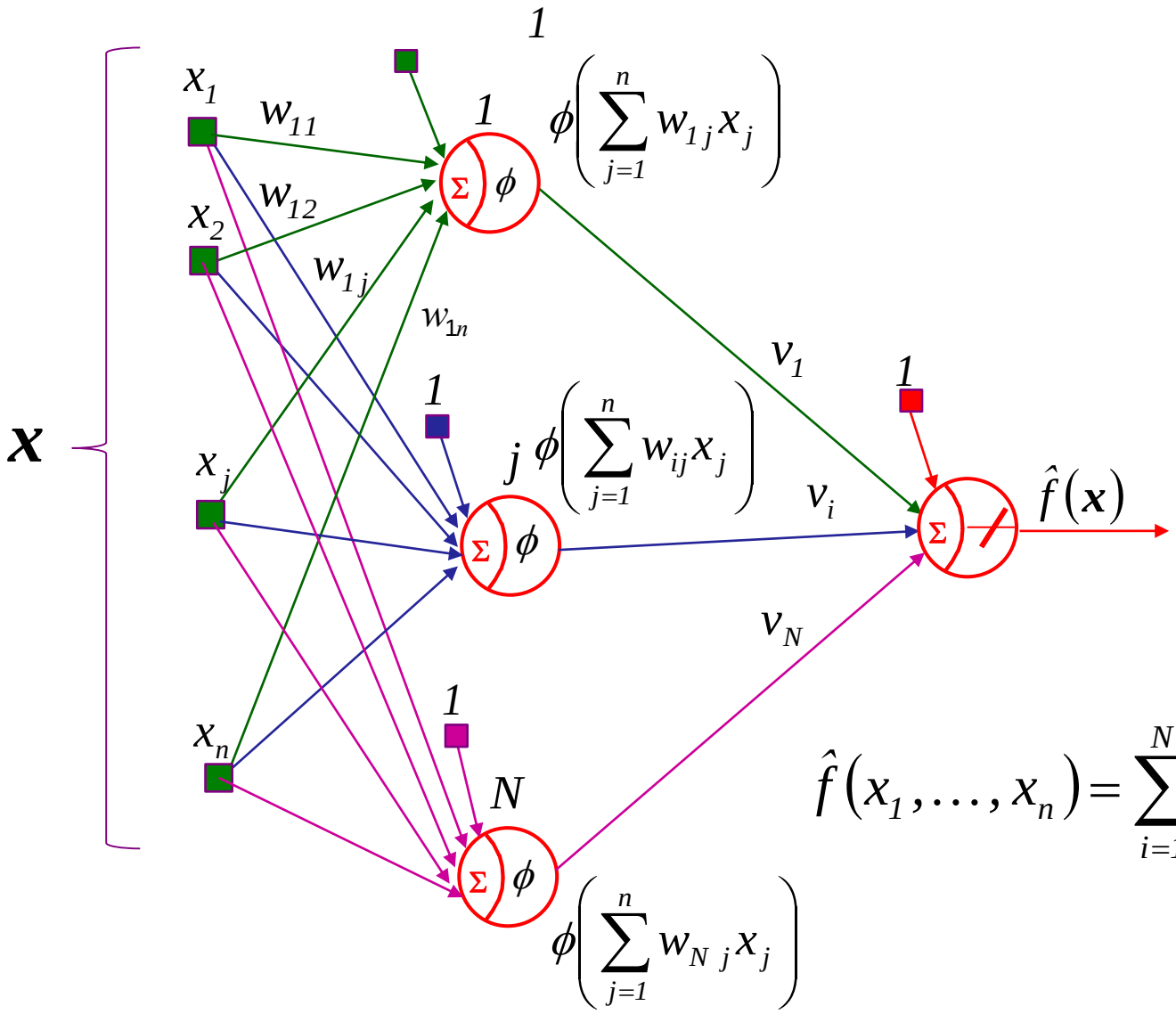
będzie rzeczywistowartościową funkcją na K. Wówczas dla dowolnej wartości $\varepsilon > 0$ istnieją stała N oraz stałe rzeczywiste v_i, w_{ij} takie, że

$$\hat{f}(x_1, \ldots, x_n) = \sum_{i=1}^{N} v_i \phi\left(\sum_{j=1}^{n} w_{ij} x_j \right)$$

spełnia

$$\left\| f - \hat{f} \right\|_\infty = \sup_{x \in K} \left| f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}) \right| \leq \varepsilon$$

Sieć neuronowa Funahashi

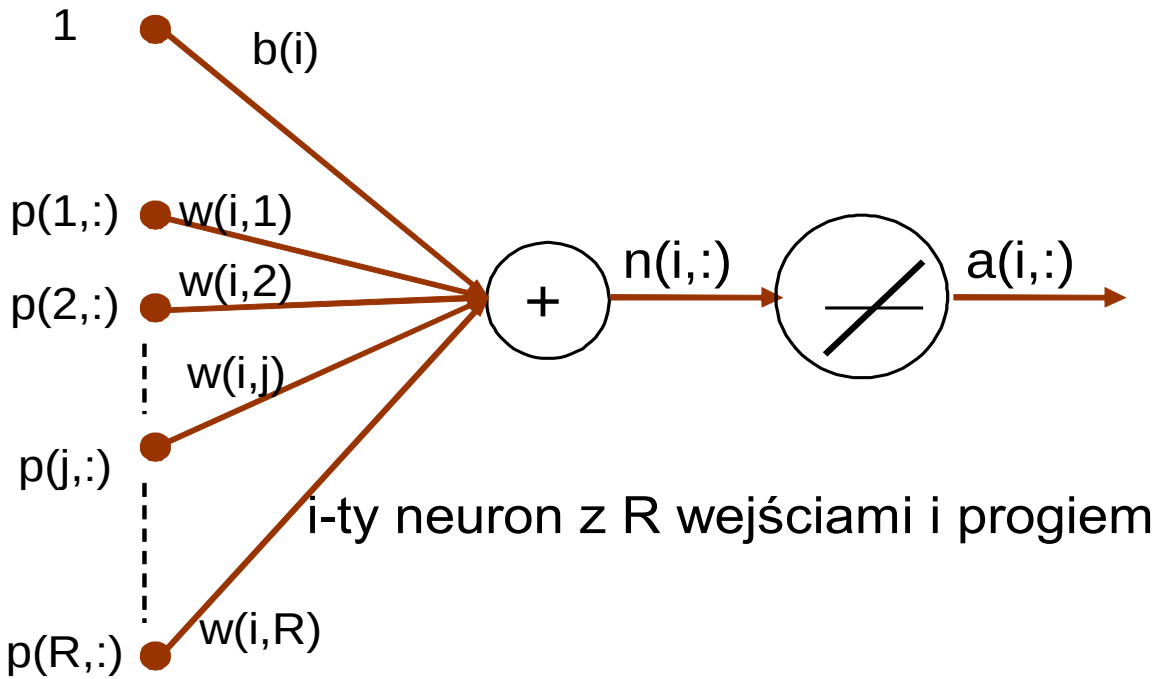


$$\hat{f}(x_1,\ldots,x_n)=\sum_{i=1}^{N} v_i \phi\left(\sum_{j=1}^{n} w_{ij}x_j\right)$$

Perceptrony proste liniowe - Adaline

Dwa sposoby przedstawiania Adaline'ów

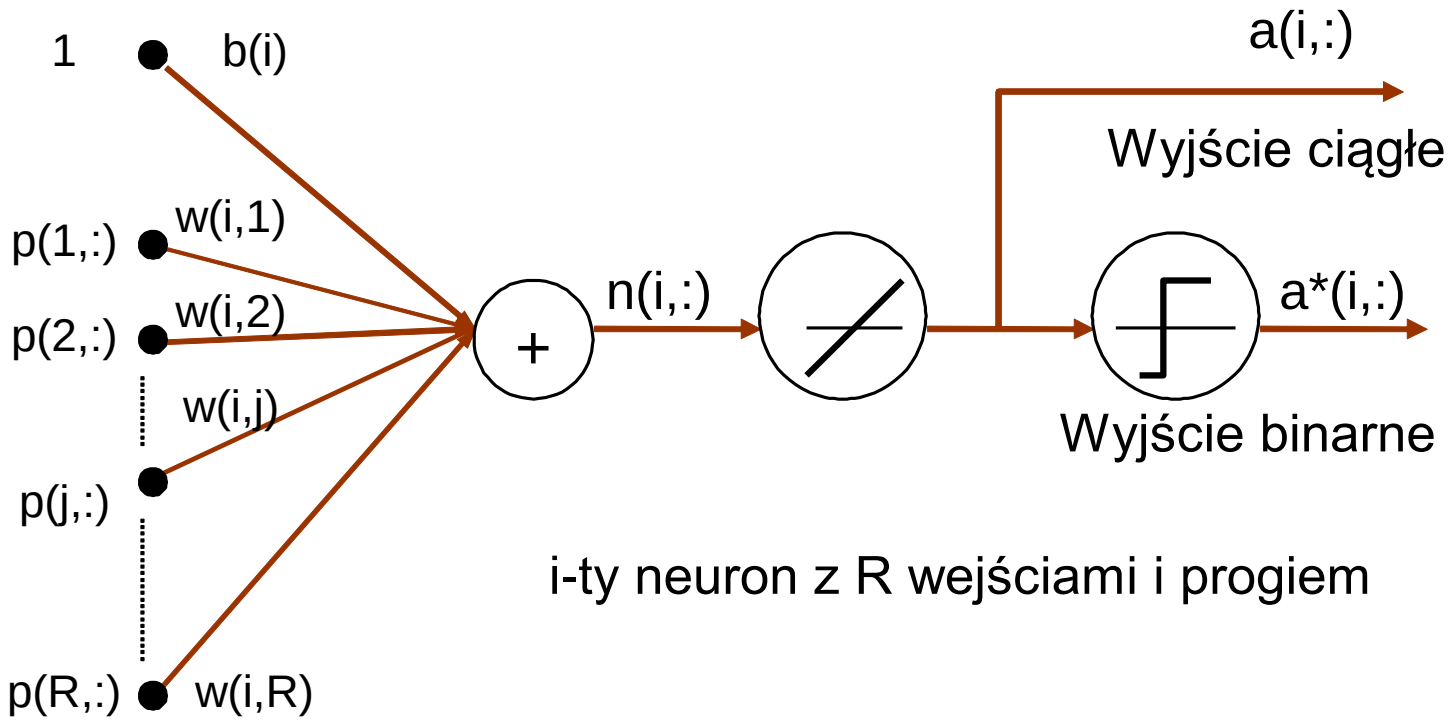
a) Adaline tylko z wyjściem analogowym:



Perceptrony proste liniowe - Adaline

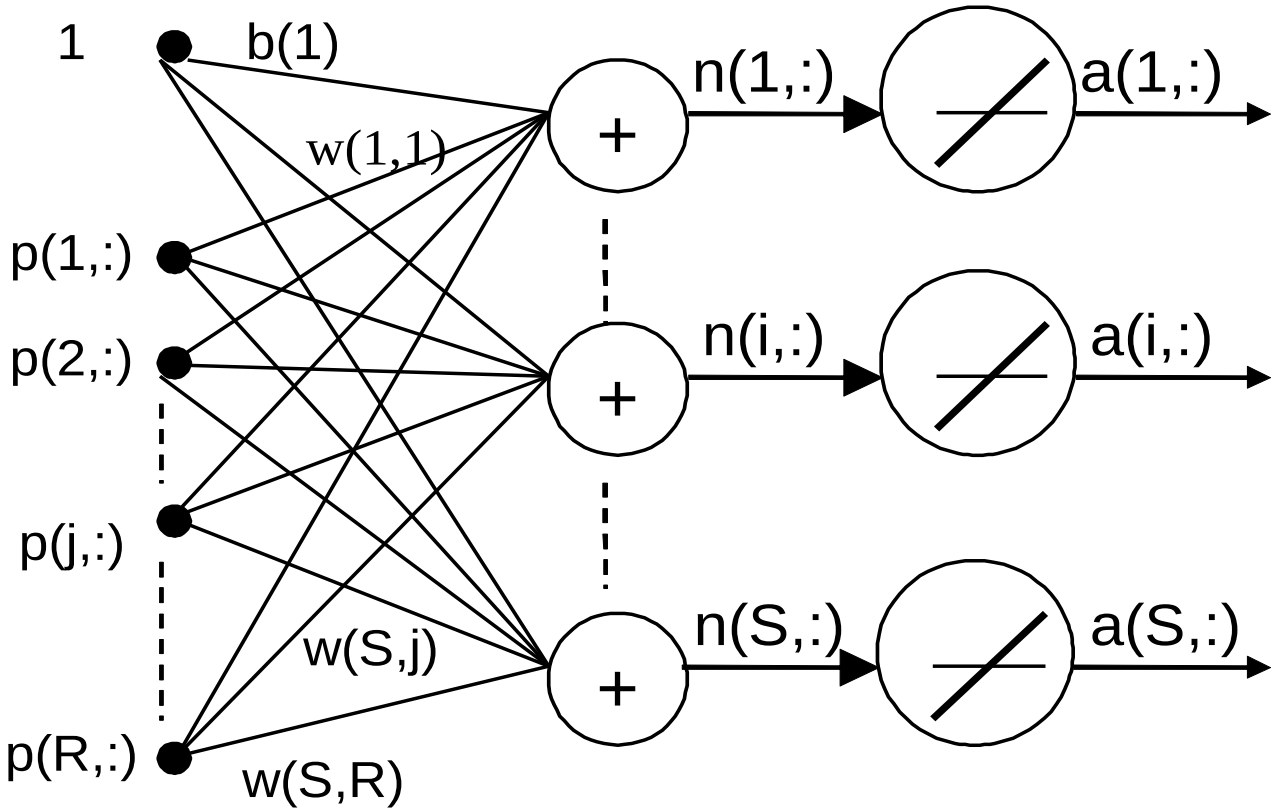
Dwa sposoby przedstawiania Adaline'ów

b) Adaline z wyjściem analogowym i binarnym:



Perceptrony proste liniowe - Adaline

Perceptrony proste liniowe - warstwa



Perceptrony proste liniowe - Adaline

Realizowane przetwarzanie

♦Wyjście analogowe

$$y_i = n_i \quad i = \overline{1,S}$$

♦Wyjście binarne

➢ Funkcja przekaźnikowa symetryczna (zwykle – jeżeli jest stosowane

$$y_i = \varphi(n_i) = sign(n_i) = \begin{cases} 1, & n_i \geq 0 \\ -1, & n_i < 0 \end{cases} \quad i = \overline{1,S}$$

gdzie:

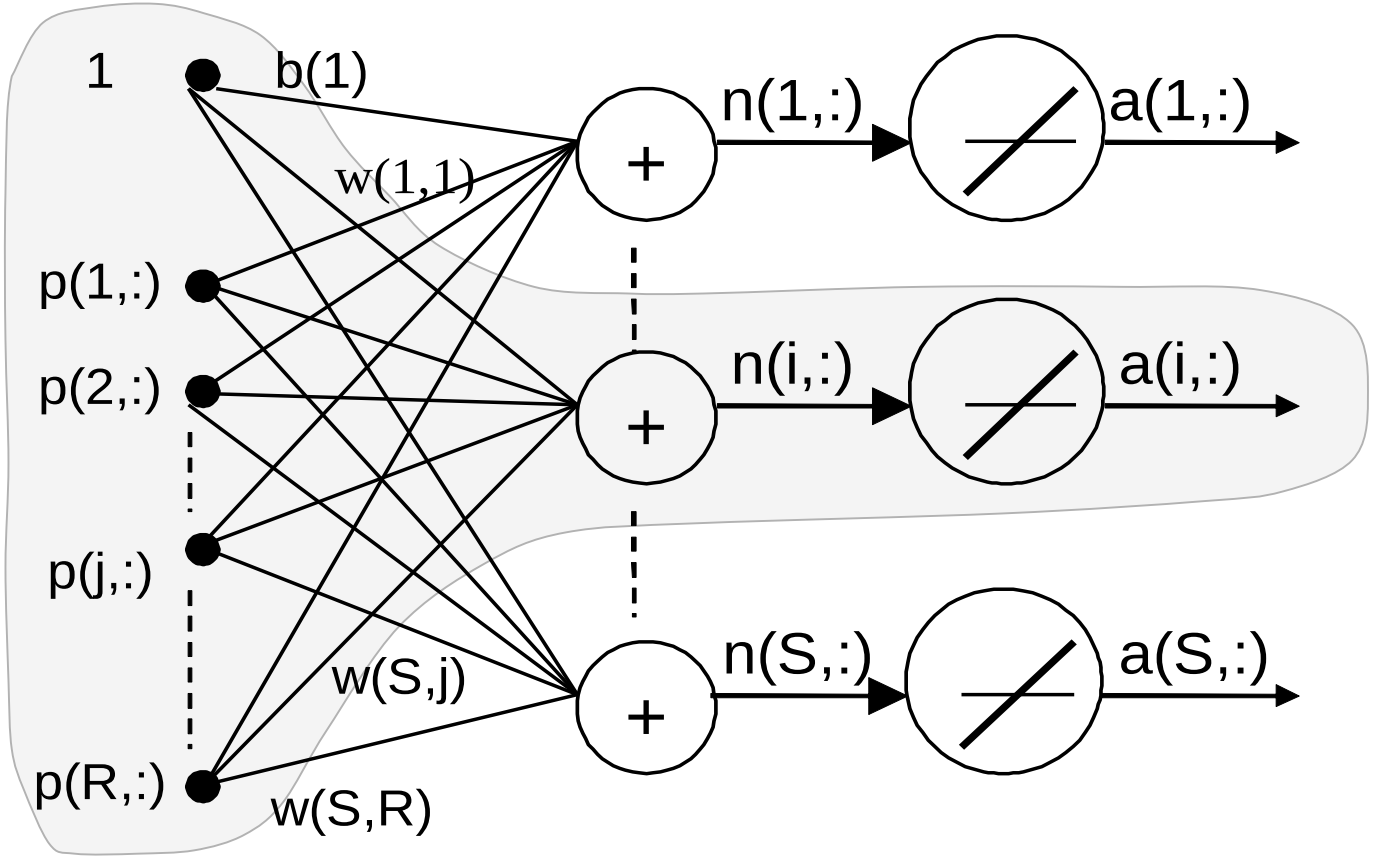
$$n_i = \sum_{j=0}^{m} w_{ij} x_j = \sum_{j=1}^{m} w_{ij} x_j + b_i; \quad i = \overline{1,S}$$

Perceptrony proste liniowe - Adaline

Reguła uczenia Widrow'a – Hoff'a

Wyprowadzenie bazujące na błędzie średnim kwadratowym

Rozważamy pojedynczy neuron w warstwie (i-ty)



Perceptrony proste liniowe - Adaline

Przyjęte oznaczenia:

- Wektor wag dla i-tego neuronu (i-ty wiersz macierzy wag i wektora progów transformowany)

$$\boldsymbol{x} = \boldsymbol{x}(i) = \begin{bmatrix} \boldsymbol{w}^T(i,:) \\ b(i) \end{bmatrix}$$

- Wektor wzorców wejściowych (k - dowolne)

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{p}(:,k) \\ 1 \end{bmatrix}$$

Perceptrony proste liniowe - Adaline

- Skalar wzorca wyjściowego rzeczywistego i-tego neuronu (k - dowolne)

$$a = a(i,k)$$

- Skalar wzorca wyjściowego docelowego i-tego neuronu (k - dowolne)

$$t = t(i,k)$$

- Skalar błędu i-tego neuronu (k - dowolne)

$$e = e(i,k) = t(i,k) - a(i,k)$$

Perceptrony proste liniowe - Adaline

$$a = \boldsymbol{x}^T \boldsymbol{z}\,; \quad dla\ dowolnego\ k$$

Wektor uczący **z** traktujemy jako zmienną losową; zmiennymi losowymi będą zatem również skalary a, t oraz e. Wybrane, w liczbie Q, wzorce uczące możemy traktować jako realizacje tych zmiennych losowych (próbki z populacji)

Średni błąd kwadratowy uczenia sieci

$$B(\boldsymbol{x}) = E\left[e^2\right] = E\left[(t-a)^2\right] = E\left[\left(t-\boldsymbol{x}^T\boldsymbol{z}\right)^2\right]$$

gdzie: E[ ] oznacza wartość średnią (oczekiwaną) zmiennej losowej. Wartość oczekiwana liczona jest po wszystkich zbiorach par uczących populacji. Zakładamy przy tym, że wybory kolejnych par uczących są niezależne od siebie

Perceptrony proste liniowe - Adaline

Po przekształceniach funkcjonałowi jakości działania sieci można nadać postać

$$B(\boldsymbol{x}) = E\left[t^2 - 2t\boldsymbol{x}^T\boldsymbol{z} + \boldsymbol{x}^T\boldsymbol{z}\boldsymbol{z}^T\boldsymbol{x}\right] = E\left[t^2\right] - 2\boldsymbol{x}^T E[t\boldsymbol{z}] + \boldsymbol{x}^T E\left[\boldsymbol{z}\boldsymbol{z}^T\right]\boldsymbol{x}$$

lub bardziej dogodną

$$\mathrm{B}(\mathbf{x}) = \mathrm{c} - 2\mathbf{x}^{\mathrm{T}}\mathbf{h} + \mathbf{x}^{\mathrm{T}}\mathbf{R}\mathbf{x}$$

gdzie:

$\mathrm{c} = \mathrm{E}[\mathrm{t}^2]$ - wartość średnia wzorców wyjściowych docelowych t

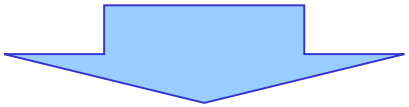
$\mathbf{h} = \mathrm{E}[\mathrm{t}\mathbf{z}]$ - wektor korelacji wzajemnej wzorców wejściowych **z** i wyjściowych docelowych t

$\mathbf{R} = \mathrm{E}[\mathbf{z}\mathbf{z}^{\mathrm{T}}]$ - macierz korelacji własnej wzorców wejściowych **z**

Jak można oszacować te wielkości? – zadanie-pytanie

Perceptrony proste liniowe - Adaline

Problem uczenia sieci: znaleźć wartości wag i progów minimalizujące wartość funkcjonału jakości działania sieci B(x)



Zadanie minimalizacji bez ograniczeń funkcjonału B(x)

Funkcjonał jakości działania sieci B(x) ma postać formy kwadratowej

$$F(\boldsymbol{x}) = c + \boldsymbol{d}^T \boldsymbol{x} + \frac{1}{2} \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$$

w której:

$$\boldsymbol{d} = -2\boldsymbol{h}, \boldsymbol{A} = 2\boldsymbol{R}$$

Perceptrony proste liniowe - Adaline

Dodatek 1:

Miary jakości działania sieci – więcej o formach kwadratowych

Dodatek 2:

Optymalizacja miar jakości sieci – więcej o metodach iteracyjnych poszukiwania ekstremów

Perceptrony proste liniowe - Adaline

Przypomnienia:

- Macierze korelacji własnej, jako symetryczne, są albo dodatnio określone, albo dodatnio półokreślone

- Gradient (jakobian) i hessian formy kwadratowej są odpowiednio równe

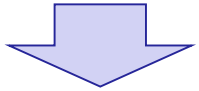
$$\nabla B(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{d} = 2\boldsymbol{R}\boldsymbol{x} - 2\boldsymbol{h}$$

$$\nabla^2 B(\boldsymbol{x}) = \boldsymbol{A} = 2\boldsymbol{R}$$

Perceptrony proste liniowe - Adaline

♣ Warunek konieczny pierwszego rzędu ekstremum: Gradient funkcjonału jest równy zeru

$$\nabla B(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{d} = 2\boldsymbol{R}\boldsymbol{x} - 2\boldsymbol{h}$$



$$\boldsymbol{R}\boldsymbol{x} - \boldsymbol{h} = 0$$

♣ Warunek konieczny drugiego rzędu ekstremum: Hessian funkcjonału jest dodatnio półokreślony

$$\nabla^2 B(\boldsymbol{x}) = \boldsymbol{A} = 2\boldsymbol{R}$$

$\boldsymbol{R}$ - jest dodatnio półokreślony

Perceptrony proste liniowe - Adaline

- Dodatnia określoność hessianu – warunek wystarczający ekstremum lokalnego silnego

Stąd:

- Jeżeli macierz korelacji własnej wzorców wejściowych jest dodatnio określona, to istnieje jednoznacznie określony punkt stacjonarny wartości wag x

$$\boldsymbol{x}^* = \boldsymbol{R}^{-1}\boldsymbol{h}$$

który jest silnym minimum funcjonału B(**x**)

Perceptrony proste liniowe - Adaline

Możemy stwierdzić:

Istnienie jednoznacznego rozwiązania problemu uczenia perceptronu prostego liniowego zależy tylko od macierzy korelacji własnej **R** wektorów wejściowych uczących

czyli

To jakie są wektory wzorców wejściowych określa jednoznacznie istnienie lub nie, jednoznacznego rozwiązania problemu uczenia sieci liniowej

Nie musimy przeprowadzać uczenia sieci!

Jeżeli możemy i chcemy obliczyć wielkości statystyczne **R** oraz **h** lub potrafimy oszacować ich wartości możemy obliczyć najlepsze wartości wag bezpośrednio

Perceptrony proste liniowe - Adaline

Poszukiwanie iteracyjne najlepszych wartości wag sieci Adaline

Zastępujemy (estymujemy)

- wartość oczekiwaną kwadratu błędu $B(\boldsymbol{x})$
- kwadratem błędu w k-tej iteracji (po pokazaniu sieci k-tej pary uczącej) dla i – tego neuronu $\hat{B}(\boldsymbol{x})$

$$\hat{B}(\boldsymbol{x}) = e^2(i,k) = (t(i,k) - a(i,k))^2 = \left(t(i,k) - \boldsymbol{x}^T \boldsymbol{z}\right)^2$$
$$= (t(i,k) - [w(i,:), b(i)][p(:,k), 1])^2 =$$
$$= \left(t(i,k) - \sum_{j=1}^{R} w(i,j)p(j,k) - b(i) \right)^2$$

Perceptrony proste liniowe - Adaline

Będziemy poszukiwać minimum $\hat{B}(\boldsymbol{x})$ metodą iteracyjną gradientu prostego; musimy zatem określić

1. kierunek gradientu (kierunek zmian **x**)

2. wielkość zmiany **x** w kierunku gradientu (wielkość kroku w kierunku gradientu)

Perceptrony proste liniowe - Adaline

Estymata gradientu wynosi

$$\hat{\nabla} B(\boldsymbol{x}) = \nabla e^2(i,k) = \begin{bmatrix} \frac{\partial e^2(i,k)}{\partial w(i,1)} \\ \vdots \\ \frac{\partial e^2(i,k)}{\partial w(i,j)} \\ \vdots \\ \frac{\partial e^2(i,k)}{\partial w(i,R)} \\ \frac{\partial e^2(i,k)}{\partial b(i)} \end{bmatrix} = \begin{bmatrix} 2e(i,k)\frac{\partial e(i,k)}{\partial w(i,1)} \\ \vdots \\ 2e(i,k)\frac{\partial e(i,k)}{\partial w(i,j)} \\ \vdots \\ 2e(i,k)\frac{\partial e(i,k)}{\partial w(i,R)} \\ 2e(i,k)\frac{\partial e(i,k)}{\partial b(i)} \end{bmatrix}$$

Perceptrony proste liniowe - Adaline

Wartości pochodnych $\frac{\partial e(i,k)}{\partial w(i,j)};\ j=\overline{1,R}$ oraz $\frac{\partial e(i,k)}{\partial b(i)}$

wynoszą

$$\frac{\partial e(i,k)}{\partial w(i,j)}=\frac{\partial(t(i,k)-a(i,k))}{\partial w(i,j)}=\frac{\partial}{\partial w(i,j)}\left(t(i,k)-\left(\boldsymbol{w}^T(i,:)\boldsymbol{p}(:,k)+b(i)\right)\right)=$$

$$=\frac{\partial}{\partial w(i,j)}\left(t(i,k)-\left(\sum_{j=1}^{R}w(i,j)p(j,k)+b(i)\right)\right)=-p(j,k)$$

$$\frac{\partial e(i,k)}{\partial b(i)}=\frac{\partial(t(i,k)-a(i,k))}{\partial b(i)}=\frac{\partial}{\partial b(i)}\left(t(i,k)-\left(\boldsymbol{w}^T(i,:)\boldsymbol{p}(:,k)+b(i)\right)\right)=$$

$$=\frac{\partial}{\partial b(i)}\left(t(i,k)-\left(\sum_{j=1}^{R}w(i,j)p(j,k)+b(i)\right)\right)=-1$$

Perceptrony proste liniowe - Adaline

Otrzymaliśmy

$$\frac{\partial e(i,k)}{\partial w(i,j)} = -p(j,k)$$

$$\frac{\partial e(i,k)}{\partial b(i)} = -1$$

Możemy napisać ostatecznie

$$\nabla \hat{B}(\boldsymbol{x}) = \nabla e^2(i,k) =$$

$$= -2e(i,k)\boldsymbol{z}(k) = -2\left(t(i,k) - a(i,k)\right)\begin{bmatrix} \boldsymbol{p}(:,k) \\ 1 \end{bmatrix}$$

Perceptrony proste liniowe - Adaline

Wyrażenie

$$\delta(i,k) = e(i,k) = \left(t(i,k) - a(i,k)\right)$$

nazywane jest deltą (błędem) i-tego neuronu przy pokazaniu k-tego wzorca sieci perceptronów prostych liniowych

Metoda gradientu prostego daje nam regułę zmiany wartości wag i progów po pokazaniu sieci k-tej pary wzorców uczących

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k} - \alpha \nabla \hat{B}(\boldsymbol{x}^{k})$$

Perceptrony proste liniowe - Adaline

Podstawiając uzyskane wyniki otrzymamy

$$\mathbf{x}^{k+1}(i) = \mathbf{x}^{k}(i) + 2\alpha\delta(i,k)\mathbf{z}(:,k)$$
$$= \mathbf{x}^{k}(i) + 2\alpha e(i,k)\mathbf{z}(:,k)$$

lub

$$\mathbf{w}^{k+1}(i) = \mathbf{w}^{k}(i) + 2\alpha e(i,k)\mathbf{p}(:,k)$$
$$= \mathbf{w}^{k}(i) + 2\alpha\left(t(i,k) - a(i,k)\right)\mathbf{p}(:,k)$$
$$b^{k+1}(i) = b^{k}(i) + 2\alpha e(i,k)$$
$$= b^{k}(i) + 2\alpha\left(t(i,k) - a(i,k)\right)$$

Ostatnie zależności noszą nazwę reguły uczenia Widrow'a-Hoff'a (reguły delty, reguły LMS)

Perceptrony proste liniowe - Adaline

Uogólnienie reguły Widrow'a-Hoff'a dla warstwy

$$\boldsymbol{W}^{k+1} = \boldsymbol{W}^{k} + 2\alpha \boldsymbol{e}(:,k)\boldsymbol{p}^{T}(:,k)$$

$$\boldsymbol{b}^{k+1} = \boldsymbol{b}^{k} + 2\alpha \boldsymbol{e}(:,k)$$

Perceptrony proste liniowe - Adaline
Wybór długości kroku (wartości α) poszukiwania najlepszych wartości wag

Modyfikacje wartości wag i progów prowadzące do minimalizacji błędu są procesem dynamicznym. Istotnym problemem jest określenie warunków zbieżności (stabilności) tego procesu

Znamy warunki zbieżności procesu poszukiwania optimum za pomocą metody gradientu prostego dla formy kwadratowej

$$F(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} + \boldsymbol{d}^T \boldsymbol{x} + c$$

$$0 < \alpha < \frac{2}{\lambda_{\max}}$$

gdzie λ_{max} jest największa wartością własną macierzy A - hessianu formy kwadratowej

Perceptrony proste liniowe - Adaline

Korzystając z tych wyników, pokażemy, że proces uczenia korzystający z reguły Widrow'a – Hoff'a jest zbieżny do

$$\boldsymbol{x}^* = \boldsymbol{R}^{-1}\boldsymbol{h}$$

czyli do minimalnego błędu średniokwadratowego $\min\left\{E\left[e^2(i,k)\right]\right\}$

Przywołajmy regułę Widrow'a – Hoff'a

$$\boldsymbol{x}^{k+1}(i) = \boldsymbol{x}^{k}(i) + 2\alpha e(i,k)\boldsymbol{z}(:,k)$$

i weźmy wartości oczekiwane obydwu stron

$$E\left[\boldsymbol{x}^{k+1}(i)\right] = E\left[\boldsymbol{x}^{k}(i)\right] + 2\alpha\, E\left[e(i,k)\boldsymbol{z}(:,k)\right]$$

Perceptrony proste liniowe - Adaline

Podstawiając

$$e(i,k) = t(i,k) - \boldsymbol{x}^k(i)^T \boldsymbol{z}(:,k)$$

otrzymujemy

$$E\left[\boldsymbol{x}^{k+1}(i)\right] = E\left[\boldsymbol{x}^k(i)\right] + 2\alpha\, E\left[t(i,k)\boldsymbol{z}(:,k) - \left(\boldsymbol{x}^k(i)^T \boldsymbol{z}(:,k)\right)\boldsymbol{z}(:,k)\right]$$

Ponieważ

$$\boldsymbol{x}^k(i)^T \boldsymbol{z}(:,k) = \boldsymbol{z}(:,k)^T \boldsymbol{x}^k(i)$$

to

$$E\left[\boldsymbol{x}^{k+1}(i)\right] = E\left[\boldsymbol{x}^k(i)\right] + 2\alpha \left[E\left[t(i,k)\boldsymbol{z}(:,k)\right] - E\left[\left(\boldsymbol{z}(:,k)\boldsymbol{z}(:,k)^T\right)\boldsymbol{x}^k(i)\right]\right]$$

Perceptrony proste liniowe - Adaline

Wielkość $\boldsymbol{x}^k(i)$ jest zależna od $\boldsymbol{z}(:,1),\boldsymbol{z}(:,2),\ldots,\boldsymbol{z}(:,k-1)$

Jeżeli założymy, że kolejne pokazywane sieci wzorce $\boldsymbol{z}(:,k)$ są niezależne stochastycznie to również $\boldsymbol{x}^k(i)$ jest niezależne od $\boldsymbol{z}(:,k)$

Wówczas

$$E\left[\boldsymbol{x}^{k+1}(i)\right]=E\left[\boldsymbol{x}^k(i)\right]+2\alpha\left(\boldsymbol{h}-\boldsymbol{R}E\left[\boldsymbol{x}^k(i)\right]\right)$$

lub inaczej

$$E\left[\boldsymbol{x}^{k+1}(i)\right]=\left[\boldsymbol{I}-2\alpha\boldsymbol{R}\right]E\left[\boldsymbol{x}^k(i)\right]+2\alpha\boldsymbol{h}$$

Tak opisany system dynamiczny, będzie stabilny, jeżeli wartości własne macierzy

$$\left[\boldsymbol{I}-2\alpha\boldsymbol{R}\right]$$

będą leżały wewnątrz okręgu jednostkowego

Perceptrony proste liniowe - Adaline

Korzystając z wyników uzyskanych dla metody gradientu i formy kwadratowej możemy stwierdzić, że warunek stabilności (zbieżności) ma postać

$$0 < \alpha < \frac{1}{\lambda_{max}}$$

gdzie λ_{max} jest największą wartością własną macierzy **R**

Do jakiej wartości będzie zbieżny proces iteracyjnego uczenia?

Rozwiązanie stacjonarne spełnia równanie

$$E[\boldsymbol{x}_s(i)] = [\boldsymbol{I} - 2\alpha\boldsymbol{R}]E[\boldsymbol{x}_s(i)] + 2\alpha\boldsymbol{h}$$

Stad:

$$E[\boldsymbol{x}_s(i)] = \boldsymbol{R}^{-1}\boldsymbol{h} = \boldsymbol{x}^*$$

Perceptrony proste liniowe - Adaline

Perceptrony proste liniowe - algorytm uczenia (jeden z możliwych)

Krok 0. Określ dopuszczalny sumaryczny błąd uczenia sieci ε (sumowanie po neuronach warstwy)

Określ maksymalna dopuszczalną liczbę epok uczenia $Epoch_{max}$

Krok 1. Nadaj wagom i progom wartości początkowe; mogą to być wartości losowe lub nadane w sposób celowy; należy jednak przestrzegać zasady, aby nie były to wartości jednakowe oraz by były niezbyt duże

Krok 2. Określ wartość współczynnika szybkości uczenia α (patrz poprzednie slajdy)

Perceptrony proste liniowe - Adaline

Krok 3. Rozpocznij kolejną epokę uczenia perceptronu liniowego

Krok 4. Pokaż sieci kolejny k-ty wzorzec wejściowy

Krok 5. Oblicz wzorzec wyjściowy rzeczywisty a(:,k)

Krok 6. Oblicz **wektor błędów odpowiedzi** e(:,k) = t(:,k) - a(:,k).

Krok 7. Dla kolejnych neuronów $i = \overline{1,S}$, wykonaj:

√ Jeżeli e(i,k) = 0; weź kolejne i (kolejny neuron)

√ Jeżeli e(i,k) ≠ 0; przeprowadź modyfikacje wartości wag i progów według reguły W-H; weź kolejne i (kolejny neuron)

Perceptrony proste liniowe - Adaline

Krok 8. √ Jeżeli był to ostatni wzorzec uczący przejdź do **kroku 9**

√ Jeżeli nie był to ostatni wzorzec uczący; przejdź do **kroku 4**

Krok 9. √ Jeżeli spełniony był warunek:

$$\sum_k \sum_i e(i,k) \le \varepsilon$$

zakończ uczenie

√ Jeżeli nie spełniony był warunek:

$$\sum_k \sum_i e(i,k) \le \varepsilon$$

przejdź do **kroku 3** (lub kroku 2).

Perceptrony proste liniowe - Adaline

Przykład 1: zadanie liniowej aproksymacji zależności pomiędzy dwoma zmiennymi (aproksymacja, regresja liniowa)

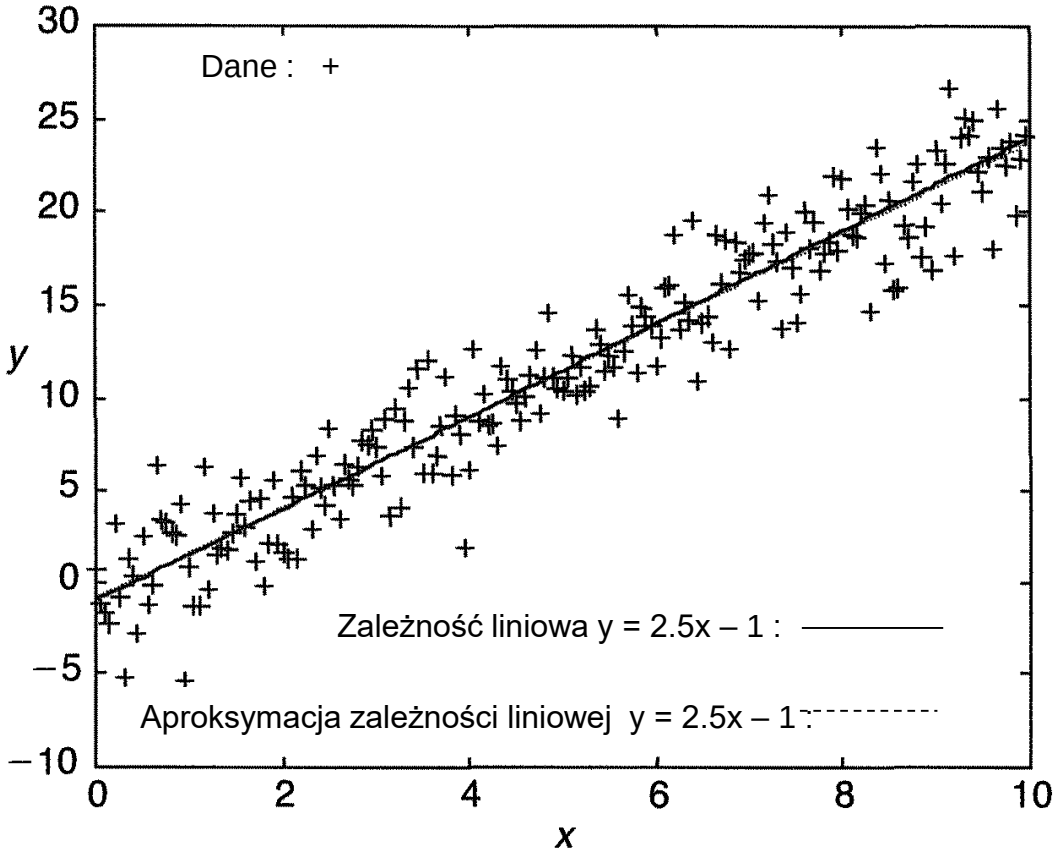
Poszukiwana jest zależność pomiędzy dwoma zmiennymi x oraz y. Przygotowane zostały dwa zbiory uczące: I – zawiera 200 pomierzonych par wartości i II – zawiera 10 pomierzonych par wartości zmiennych. Sposób wygenerowania zbiorów uczących: przyjęto, że proces opisany jest zależnością liniową

$$y = 2.5x - 1 + n$$

gdzie, $x \in [0,10]$ n jest gaussowską zmienną losową o wartości średniej równej zero i wariancją taką, że zaburza ona poprawną wartość y 20% szumem.

Perceptrony proste liniowe - Adaline

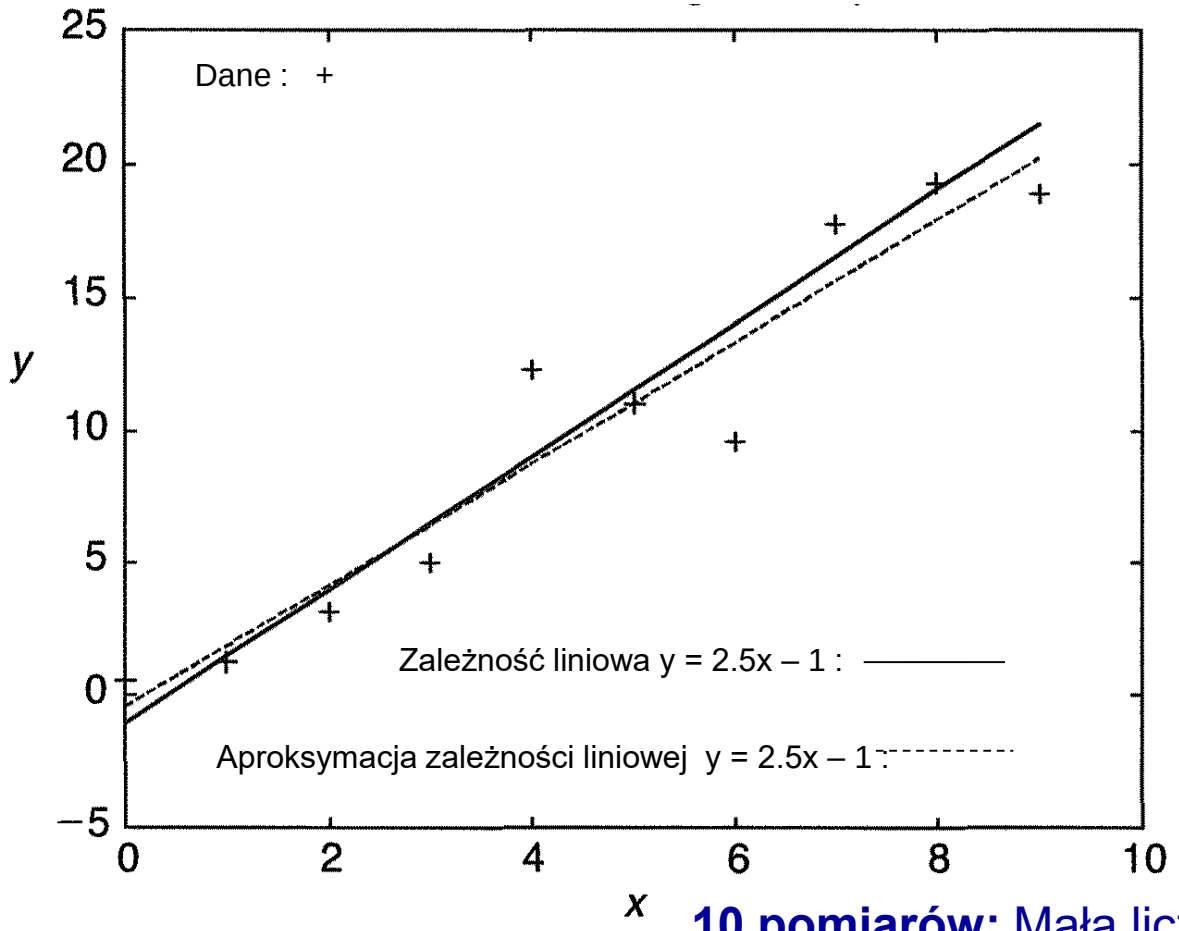
Rozwiązanie zadania aproksymacji liniowej przez adalin:



200 pomiarów: Duża liczba pomiarów – wartości wag i progu adalinu dobrze estymują wartości nieznanych parametrów

Perceptrony proste liniowe - Adaline

Rozwiązanie zadania aproksymacji liniowej przez adalin:



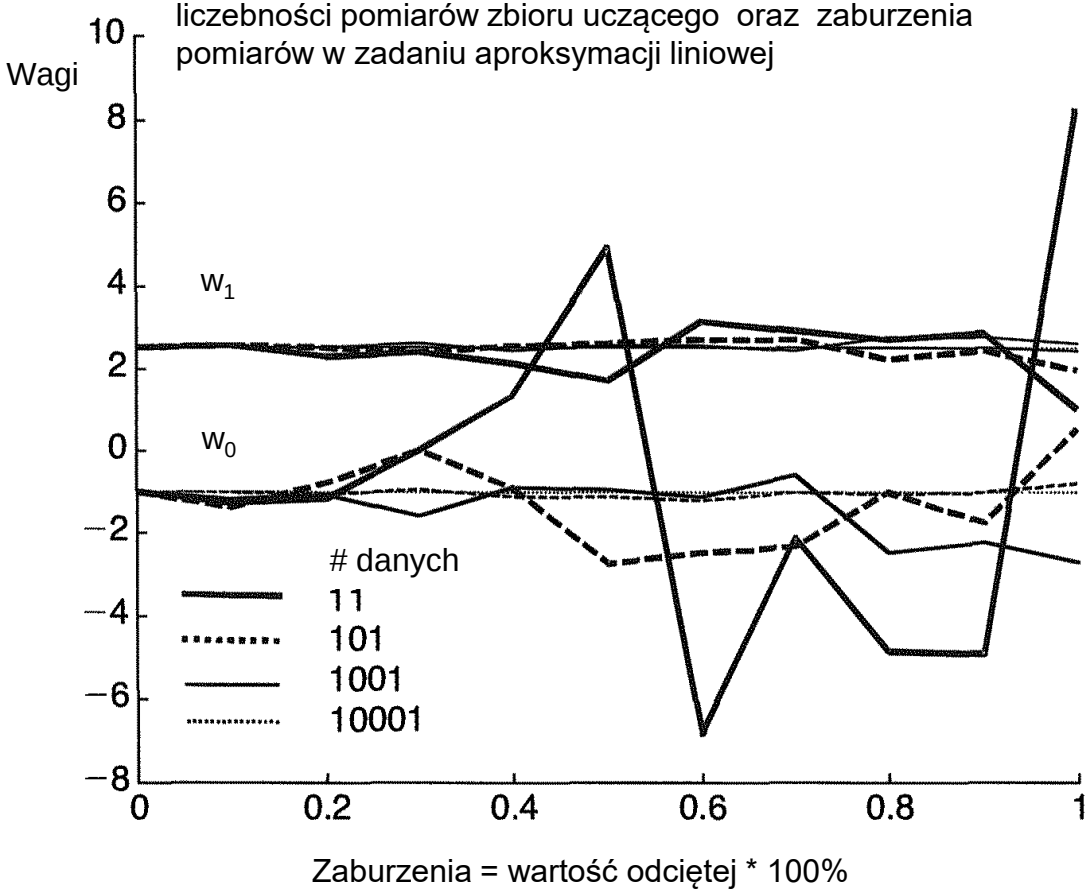
Wniosek: większa liczba pomiarów sprzyja jakości aproksymacji

10 pomiarów: Mała liczba pomiarów – wartości wag i progu adalinu gorzej estymują wartości nieznanych parametrów

Perceptrony proste liniowe - Adaline

Rozwiązanie zadania aproksymacji liniowej przez adalin:

Wartości estymowane w_1 oraz w_0 (b) w zależności od liczebności pomiarów zbioru uczącego oraz zaburzenia pomiarów w zadaniu aproksymacji liniowej



Spostrzeżenia:

- Rozwiązanie zadania jest możliwe nawet przy wysokim poziomie zaburzeń
- Im wyższy poziom zaburzeń, tym większa liczebność zbioru uczącego jest niezbędna dla uzyskania dobrych estymat parametrów
- Estymacja wartości progu jest bardziej wrażliwa zarówno na poziom zaburzeń, jak i liczebność zbioru uczącego

Perceptrony proste liniowe - Adaline

Przykład 2: zadanie liniowej aproksymacji zależności pomiędzy trzema zmiennymi (aproksymacja, regresja liniowa)

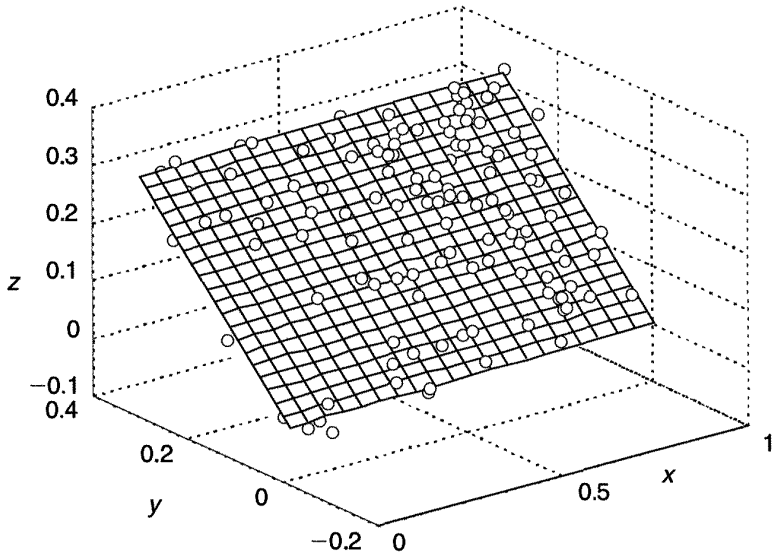
Poszukiwana jest zależność pomiędzy trzema zmiennymi x, y oraz z. Przygotowane został zbiór uczący: 200 pomierzonych par wartości zmiennych powiązanych zależnością liniową

$$z = 0.004988x + 0.995y + n$$

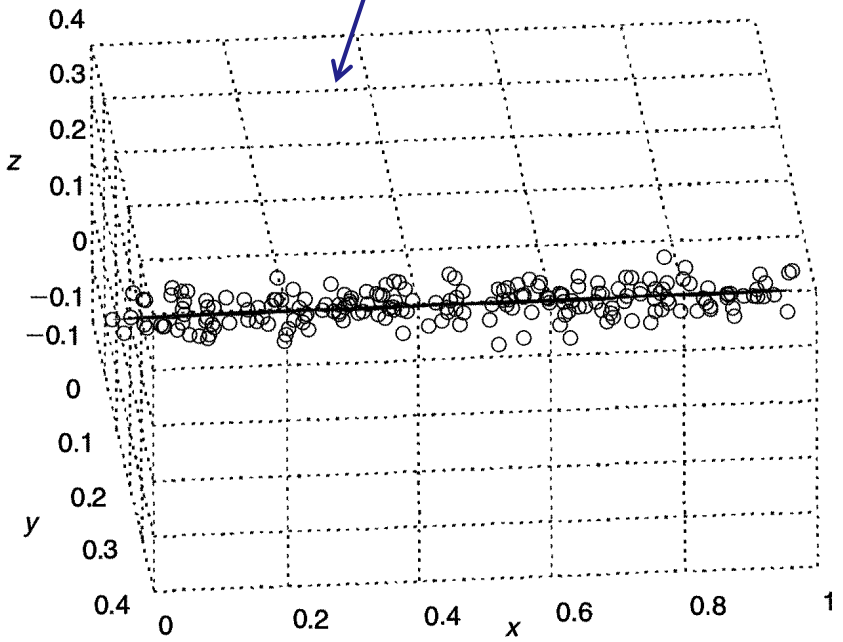
gdzie, $x \in [0,1]$ $y \in [-0.25,+0.25]$, n jest gaussowską zmienną losową o wartości średniej równej zero i wariancją taką, że zaburza ona poprawną wartość z 20% szumem.

Perceptrony proste liniowe - Adaline

Rozwiązanie zadania aproksymacji liniowej przez adalin:



Spojrzenie w płaszczyźnie regresji



Perceptrony proste liniowe - Adaline

Przykład 3: zadanie identyfikacji dynamicznego obiektu liniowego opisywanego ogólnym równaniem

$$y_k = a_1 y_{k-1} + a_2 y_{k-2} + \cdots + a_n y_{k-n} + b_1 u_{k-1} + b_2 u_{k-2} + \cdots + b_n u_{k-n} + n_k,$$

gdzie, y_{k-i} oraz u_{k-i} są przeszłymi wyjściami i wejściami, a n_k jest addytywnym białym szumem.

Problem identyfikacji może być rozważany jako problem aproksymacji liniowej w przestrzeni R^{n+n+1} (ogólniej R^{m+n+1})

Rozważamy problem identyfikacji obiektu drugiego rzędu:

$$Y(s) = \frac{3}{s^2 + s + 3} U(s)$$

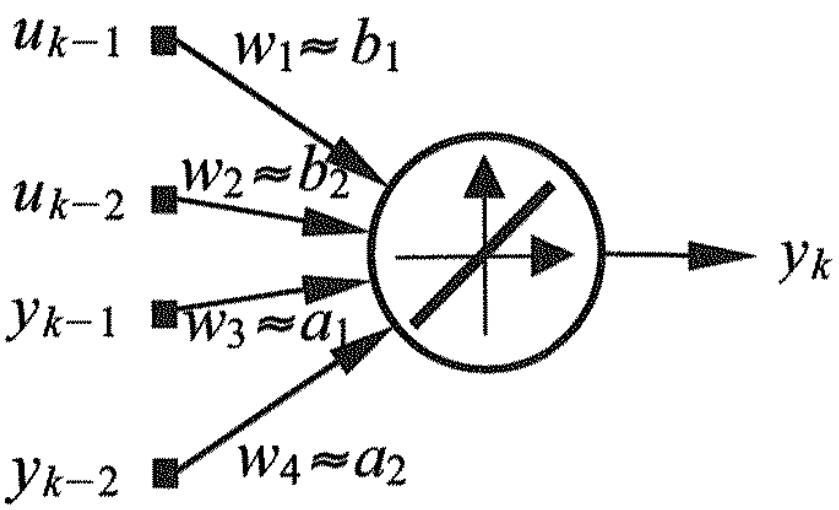
W dziedzinie czasu jednostkami dla tego obiektu są sekundy. Przy kroku próbkowania $\Delta T = 0.25s$ otrzymamy równanie dyskretne:

$$y_k = 1.615 y_{k-1} - 0.7788 y_{k-2} + 0.08508 u_{k-1} + 0.07824 u_{k-2}$$

Perceptrony proste liniowe - Adaline

Do identyfikacji wykorzystamy zbiór uczący 50 par pomiarów wejście – wyjście. Wejście uczące – pseudolosowy sygnał binarny (PLSB), a wyjście systemu zaburzane jest sygnałem białego szumu o wartości średniej zero i wariancji dającej zniekształcenie wyjścia na poziomie 5%.

Struktura sieci:



Wzorce wejściowe:

$$p_k = \begin{pmatrix} u_{k-1} \\ u_{k-2} \\ y_{k-1} \\ y_{k-2} \end{pmatrix}; \quad k = \overline{3, K}$$

Wzorce wyjściowe docelowe:

$$t_k = y_k; \quad k = \overline{3, K}$$

Perceptrony proste liniowe - Adaline

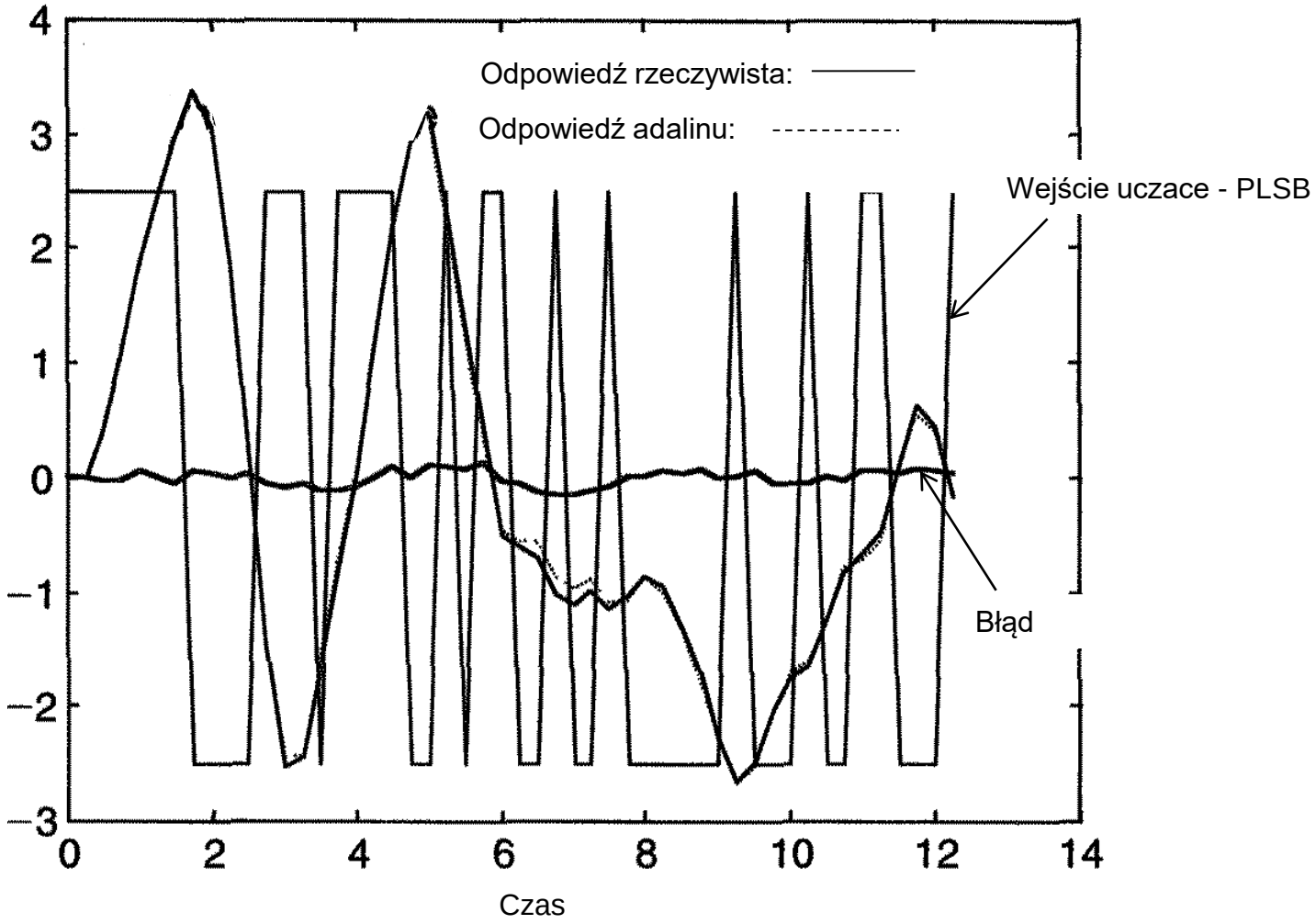
Wynik procesu uczenia – wartości wag w_i będące estymatami współczynników a oraz b:

- współczynniki określające hiperpłaszczyznę w przestrzeni R^5 na której powinny leżeć wszystkie trajektorie wyjścia y dla odpowiadających wejść u, identyfikowanego obiektu przy braku zaburzeń (szumów).

- współczynniki równania różnicowego drugiego rzędu

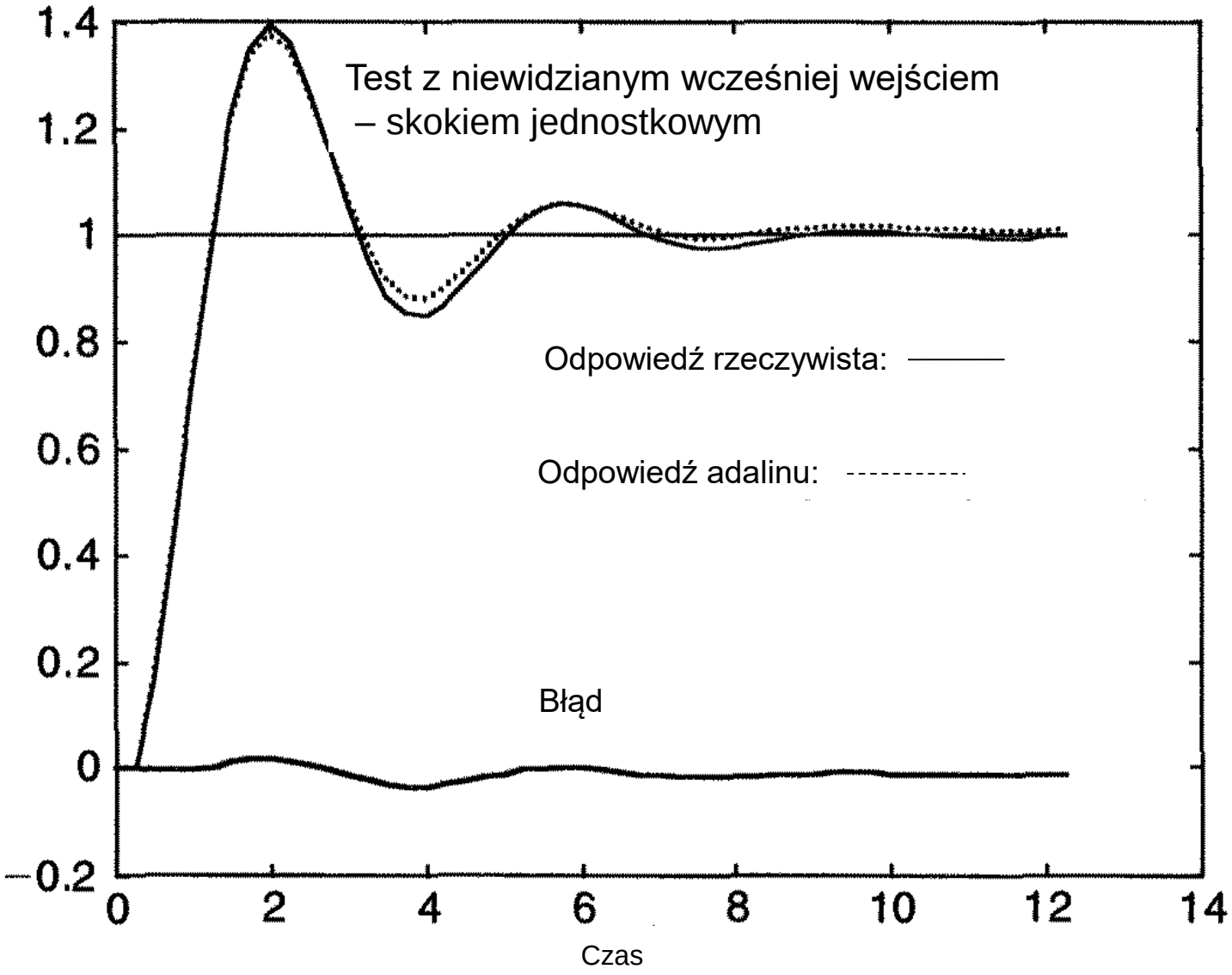
Perceptrony proste liniowe - Adaline

Rozwiązanie zadania identyfikacji liniowej przez adalin – wyniki uczenia:



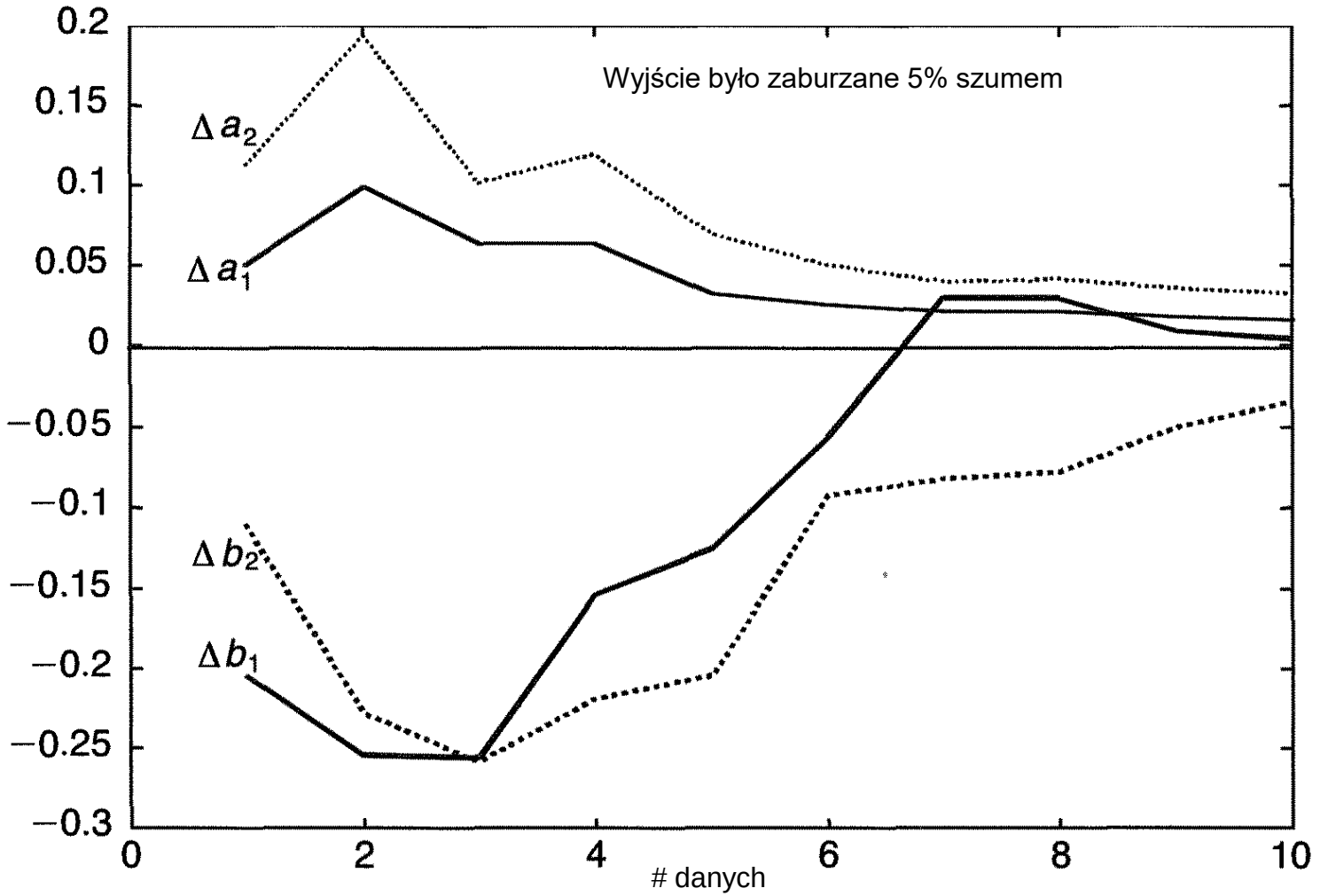
Perceptrony proste liniowe - Adaline

Rozwiązanie zadania identyfikacji liniowej przez adalin – wyniki testu:



Perceptrony proste liniowe - Adaline

Rozwiązanie zadania identyfikacji liniowej przez adalin – błąd względny estymowanych parametrów; zależność od liczebności zbioru uczącego



1 – 10, 2 – 12, 3 – 17, 4 – 30, 5 – 62, 6 – 154, 7 – 460 , 8 – 1648, 9 – 7080, 10 – 36573

Dziękuję
– koniec materiału prezentowanego podczas wykładu