Sztuczne sieci neuronowe - przypomnienie

Dodatkowe slajdy

do wykładu UM dla ISWD 2018

Jerzy Stefanowski

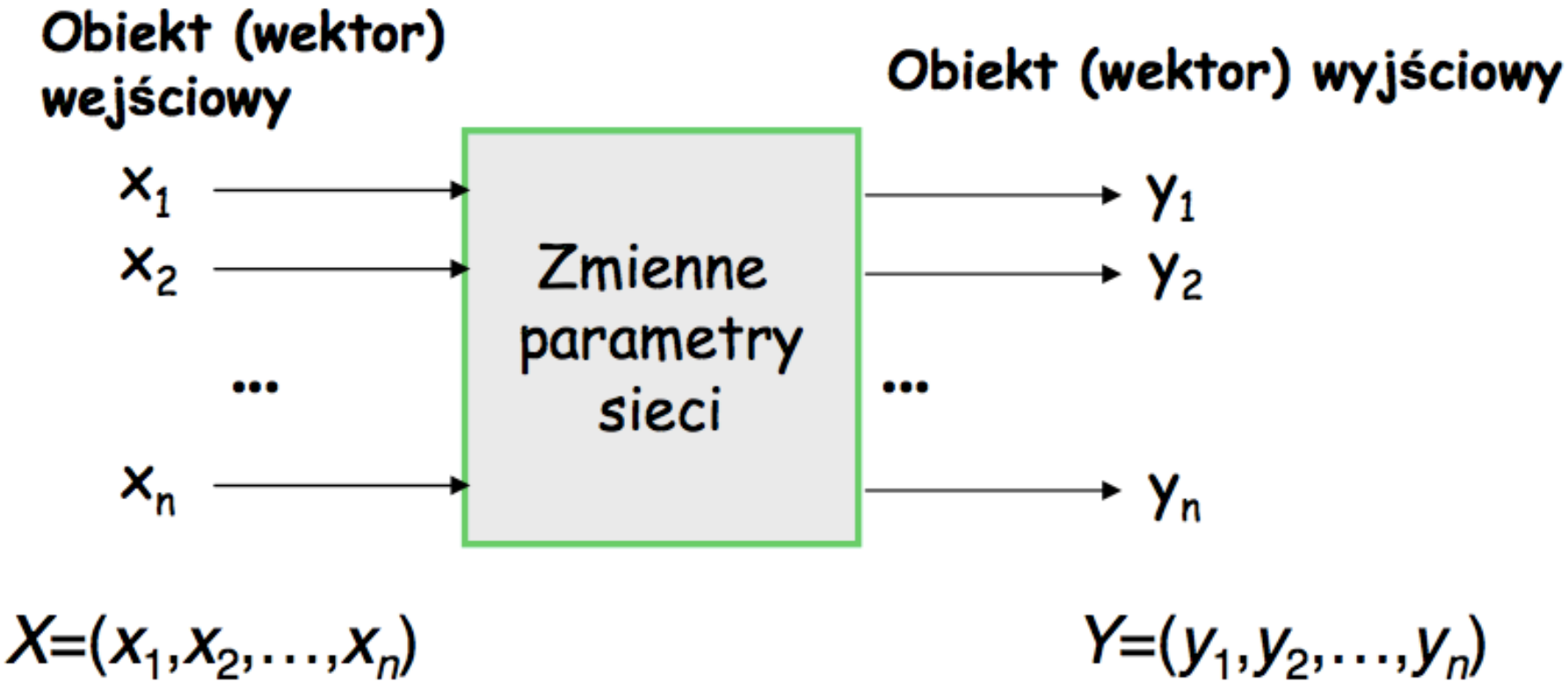
Podziękowania

W trakcie przygotowania wykładu wykorzystano także slajdy oraz inspiracje z materiałów i wykładów prof. Ryszarda Tadeusiewicza

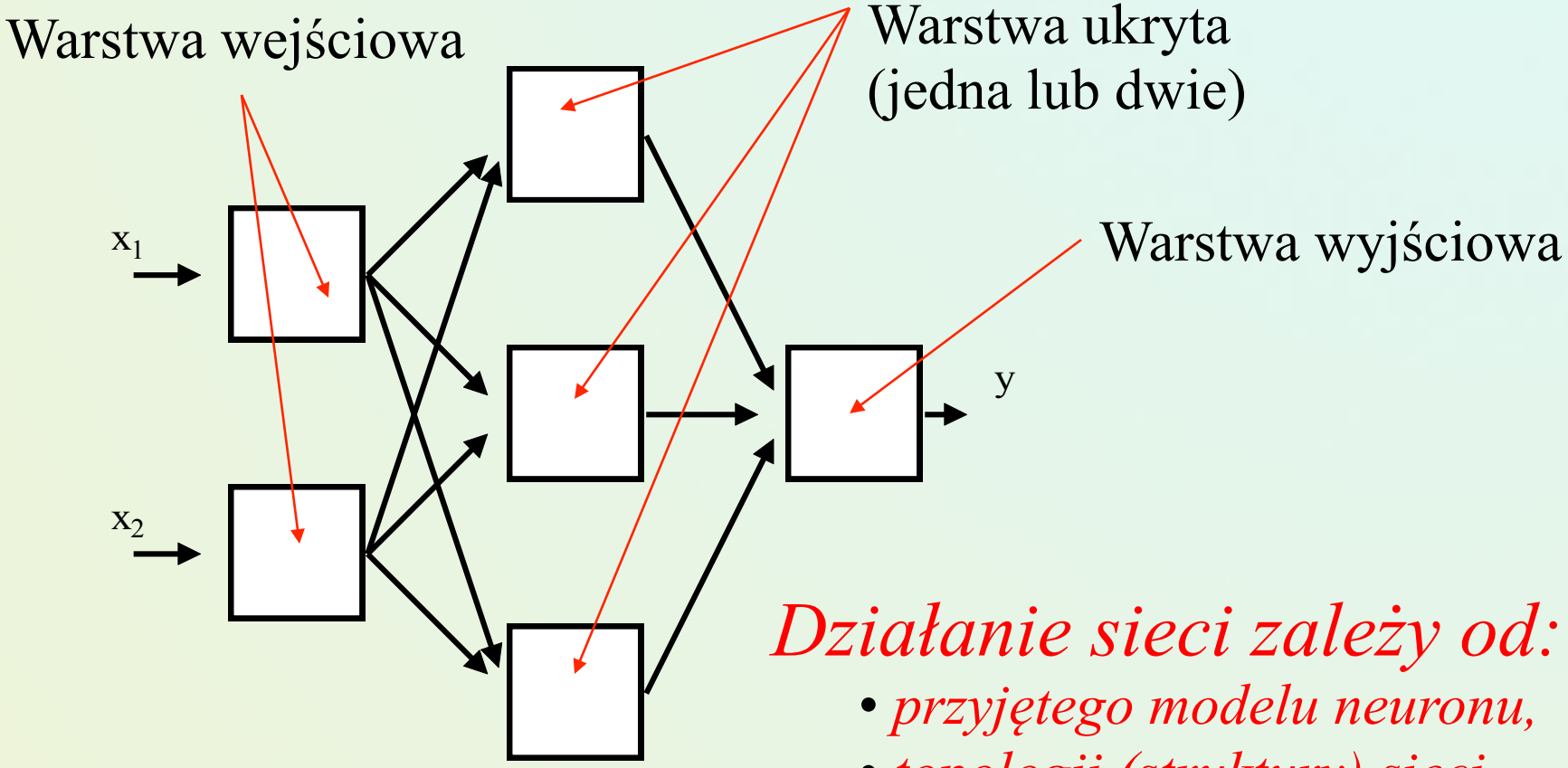
www.uci.agh.edu.pl/uczelnia/tad

Wprowadzenie

ANN „umieją” odwzorować różne złożone zależności pomiędzy sygnałami wejściowymi i wyjściowymi.



Schemat sztucznej sieci neuronowej (uproszczonej)



Działanie sieci zależy od:

- *przyjętego modelu neuronu,*
- *topologii (struktury) sieci,*
- *wartości parametrów neuronu, ustalanych w wyniku uczenia*

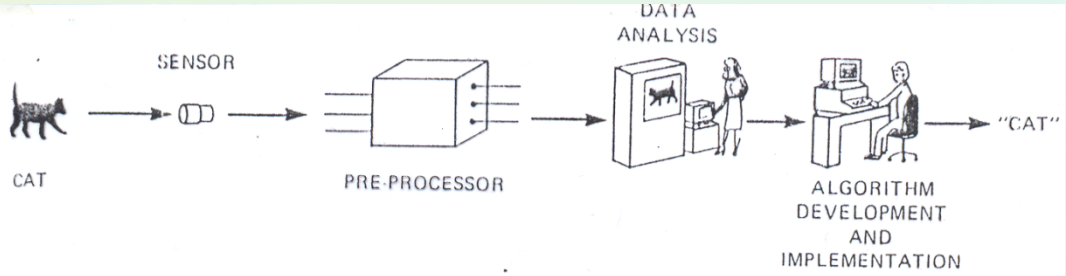
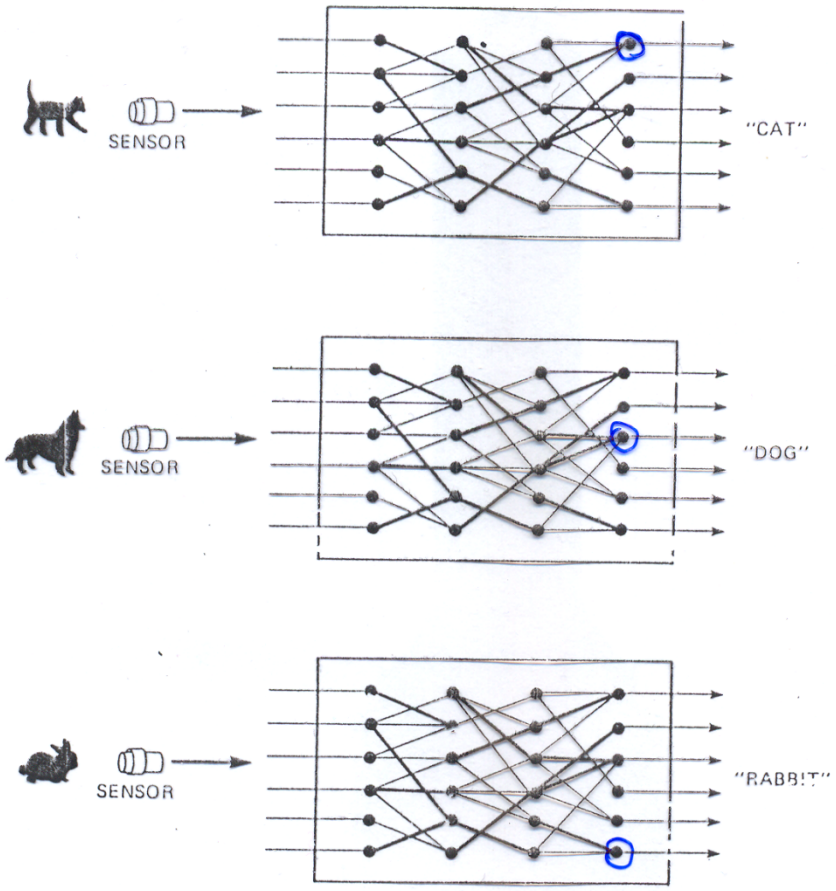


Figure 1-7a. Traditional approach to a pattern classification problem (adapted from DARPA Neural Network Study, 1988).



Porównanie idei obliczeń tradycyjnych i neuronowych

Modele neuronowe są zawsze trochę magiczne… lecz mogą być skuteczne

$$\oint \mathbf{E} \cdot d\mathbf{A} = \frac{q_{enc}}{\varepsilon_0}$$

$$\oint \mathbf{B} \cdot d\mathbf{A} = 0$$

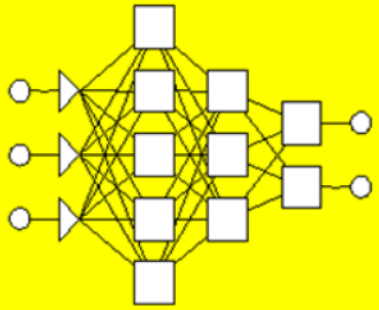
$$\oint \mathbf{E} \cdot d\mathbf{s} = -\frac{d\Phi_B}{dt}$$

$$\oint \mathbf{B} \cdot d\mathbf{s} = \mu_0 \varepsilon_0 \frac{d\Phi_E}{dt} + \mu_0 i_{enc}$$

Pozwala zrozumieć działanie systemu

Jest trudny w użyciu

Model behawio-ralny



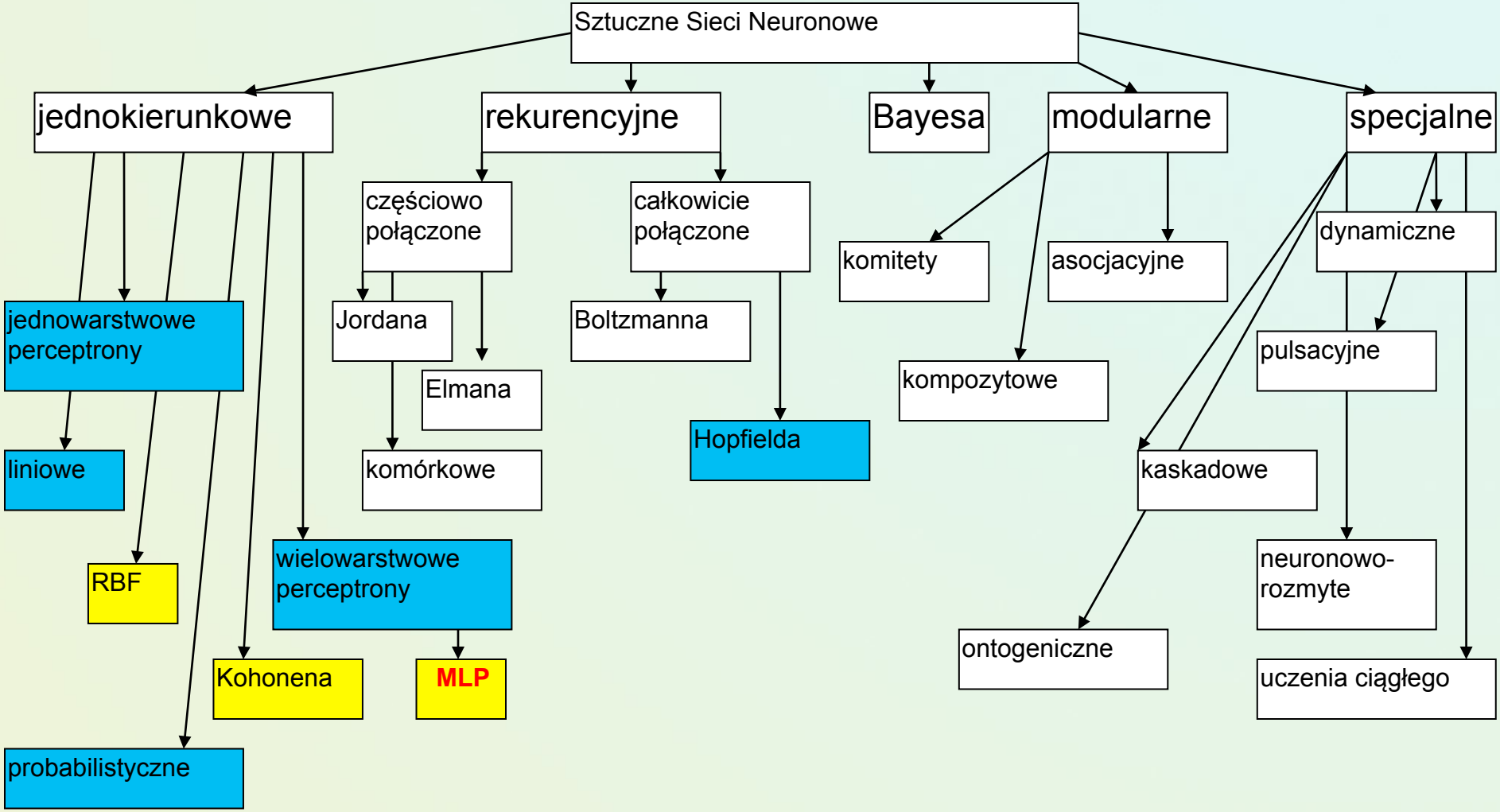
Pozwala skutecznie kontrolować system

Jest łatwy w użyciu

Nie rozwija wiedzy

Brak wyjaśnialności sieci neuronowych

Typy sztucznych sieci neuronowych



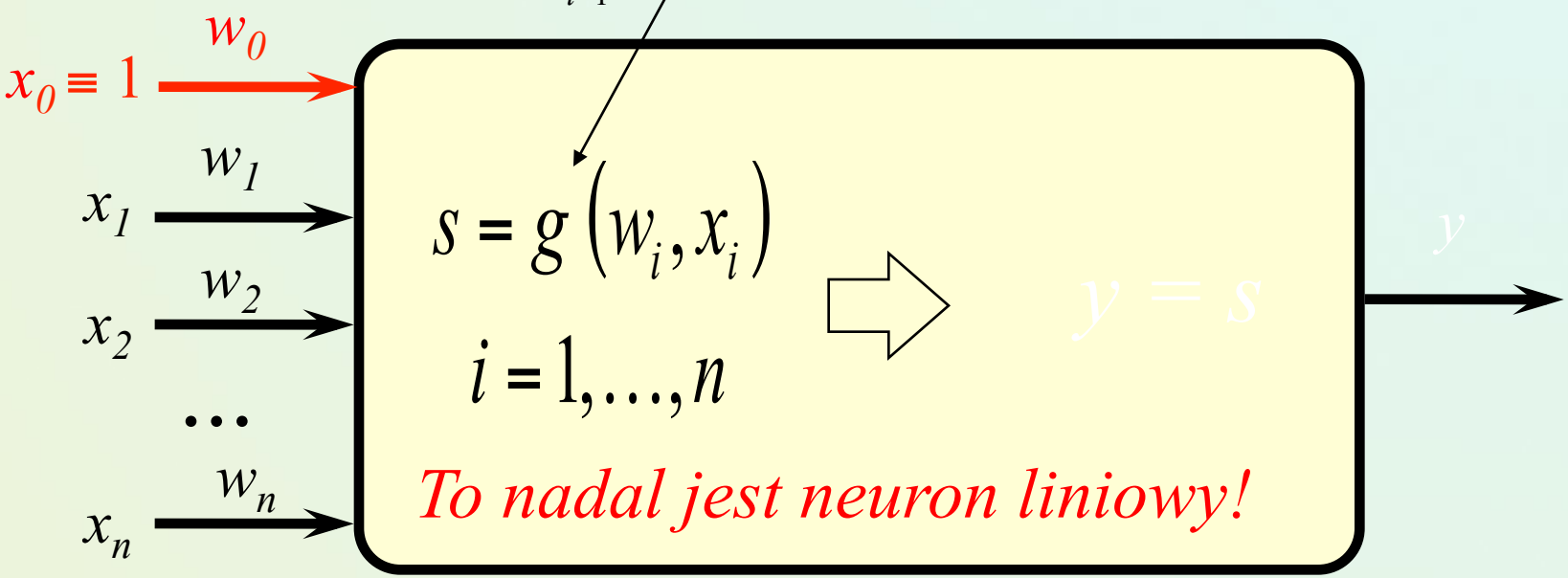
+ Głebokie uczenie

Neuron liniowy

Czysta agregacja liniowa:

$$s = \sum_{i=1}^{n} w_i x_i$$

ma wadę, polegającą na tym, że charakterystyka neuronu musi tu przechodzić przez początek układu



Żeby zachować liniową postać wzoru opisującego neuron dodaje się dodatkowe pseudo-wejście nazywane BIAS, które zawsze dostarcza sygnał ≡ 1

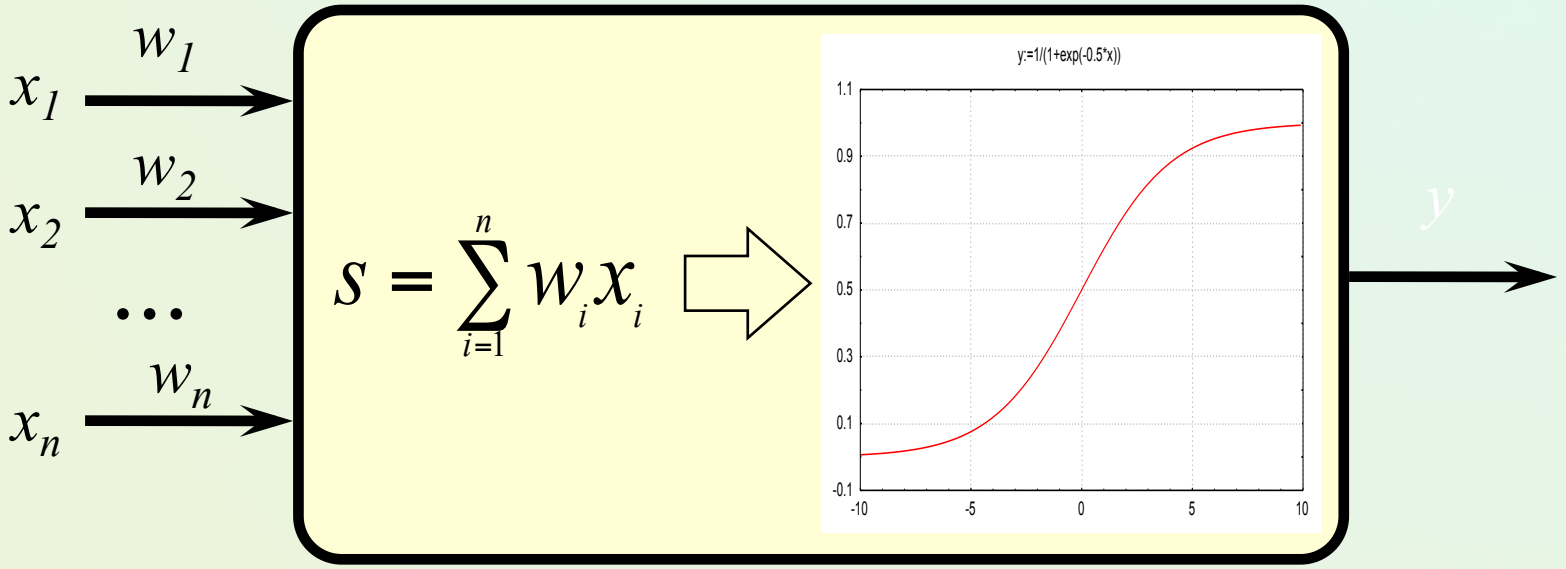
Bogatsze możliwości daje agregacja afiniczna (z wyrazem wolnym w formule):

$$s = \sum_{i=1}^{n} w_i x_i + w_0$$

Wtedy agregacja *jest nadal* liniowa:

$$s = \sum_{i=0}^{n} w_i x_i$$

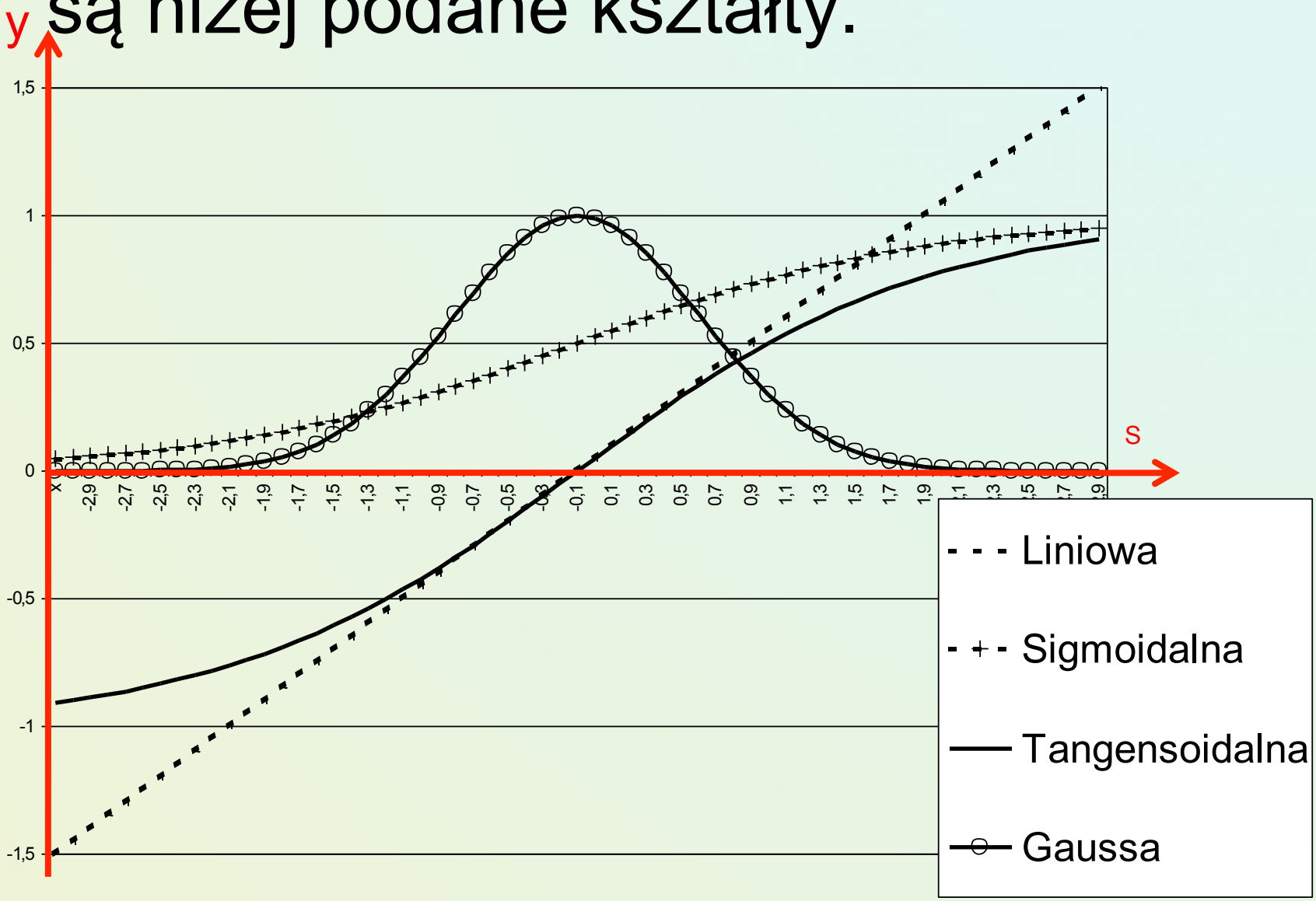
Najbardziej popularny neuron nieliniowy sigmoidalny, nadający się do budowy sieci **MLP**



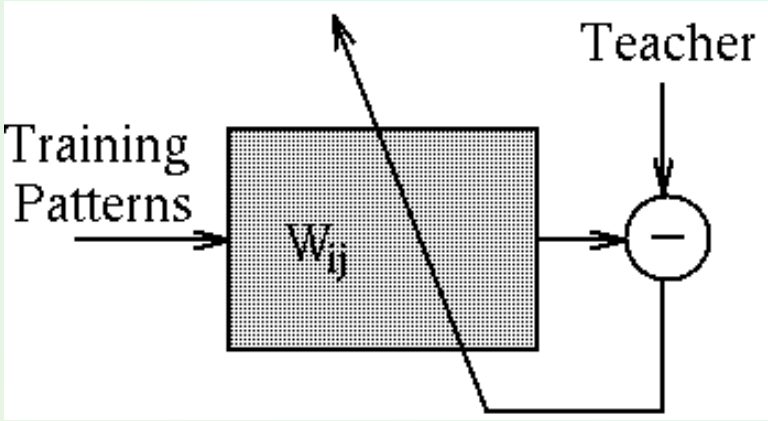
Różne przykładowe formuły matematyczne, wykorzystywane jako funkcje przejścia

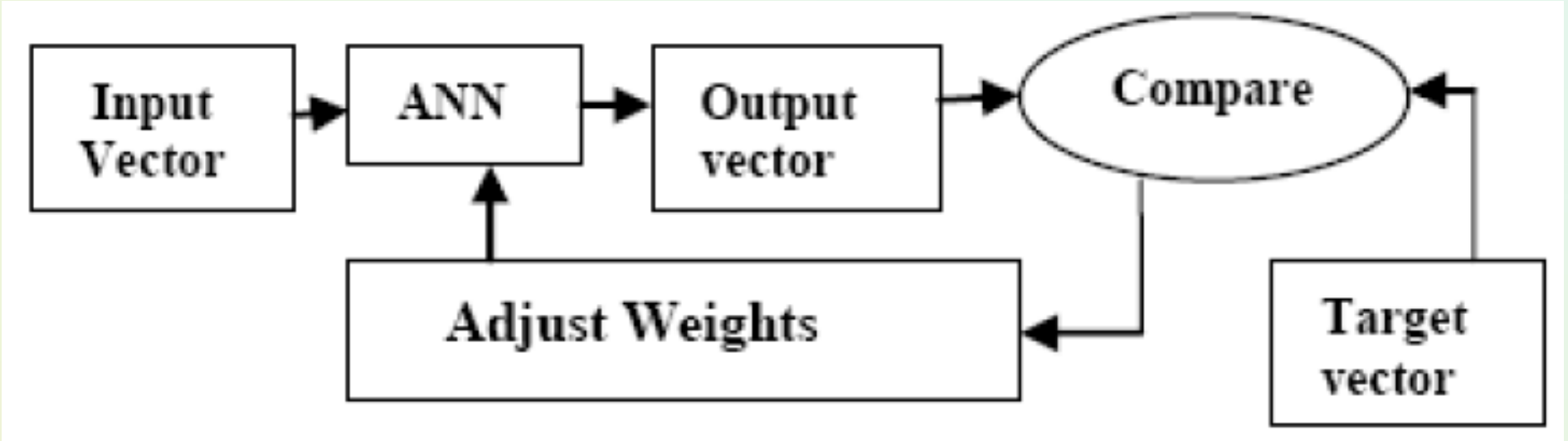
funkcja aktywacji	pochodna funkcji aktywacji	funkcja aktywacji	pochodna funkcji aktywacji
$F(x)=\frac{1}{1+e^{-x}}$	$\frac{dF(x)}{dx}=F(x)(1-F(x))$	$F(x)=1-e^{-x^2}$	$\frac{dF(x)}{dx}=2x(1-F(x))$
$F(x)=\frac{2}{1+e^{-2x}}-1$	$\frac{dF(x)}{dx}=1-F^2(x)$	$F(x)=\begin{cases}1-e^{-x}, & \text{dla } x\geq 0\\ -1+e^{x}, & \text{dla } x<0\end{cases}$	$\frac{dF(x)}{dx}=1-\lvert F(x)\rvert$
$F(x)=\frac{2}{\pi}\arctan(x)$	$\frac{dF(x)}{dx}=\frac{2}{\pi}\frac{1}{1+x^2}$	$F(x)=\begin{cases}\sin(x) & -\pi/2<x<\pi/2\\ \operatorname{sgn}(x) & -\pi/2\geq x\geq \pi/2\end{cases}$	$\frac{dF(x)}{dx}=\sqrt{1-\left[F(x)\right]^2}$
$F(x)=\begin{cases}\ln(1+x), & x\geq 0\\ -\ln(1-x), & x<0\end{cases}$	$\frac{dF(x)}{dx}=\begin{cases}1/(1+x), & x\geq 0\\ 1/(1-x), & x<0\end{cases}$	$F(x)=\int_0^x\left[F(t)(1-F(t))\right]^p dt$	$\frac{dF(x)}{dx}=F^p(x)(1-F(x))^p,$ $F(x=0)=1/2$
$F(x)=e^{-x^2}$	$\frac{dF(x)}{dx}=-2xF(x)$	$F(x)=\int_0^x\left(1-\lvert F(t)\rvert^s\right)dt$	$\frac{dF(x)}{dx}=1-\lvert F(x)\rvert^s,$ $F(x=0)=0$

Funkcje aktywacji neuronu może być dowolna, ale najczęściej stosowane są niżej podane kształty.



Ogólna idea procesu uczenia

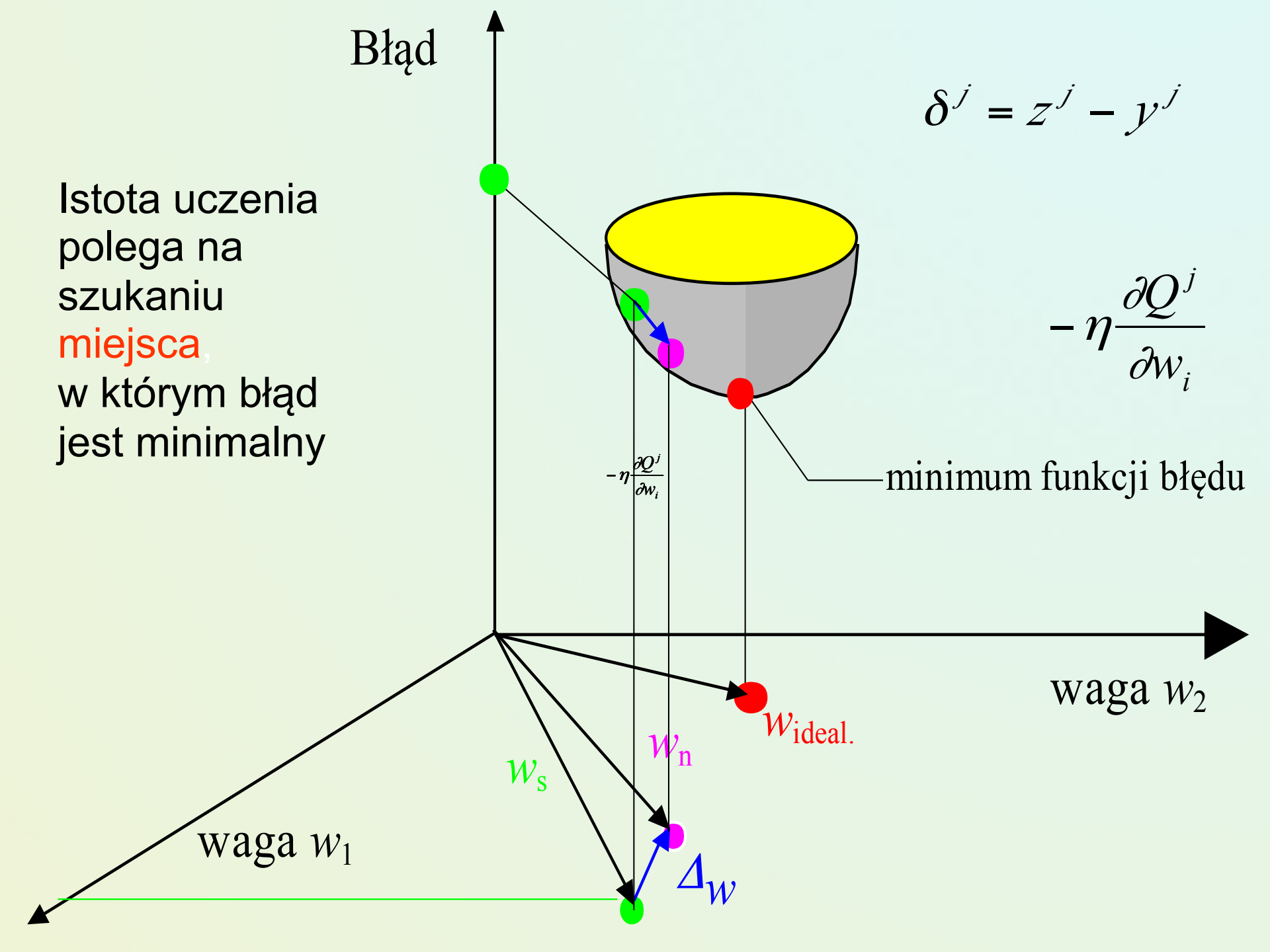




Wówczas najczęściej stosowane miary błędu są następujące:

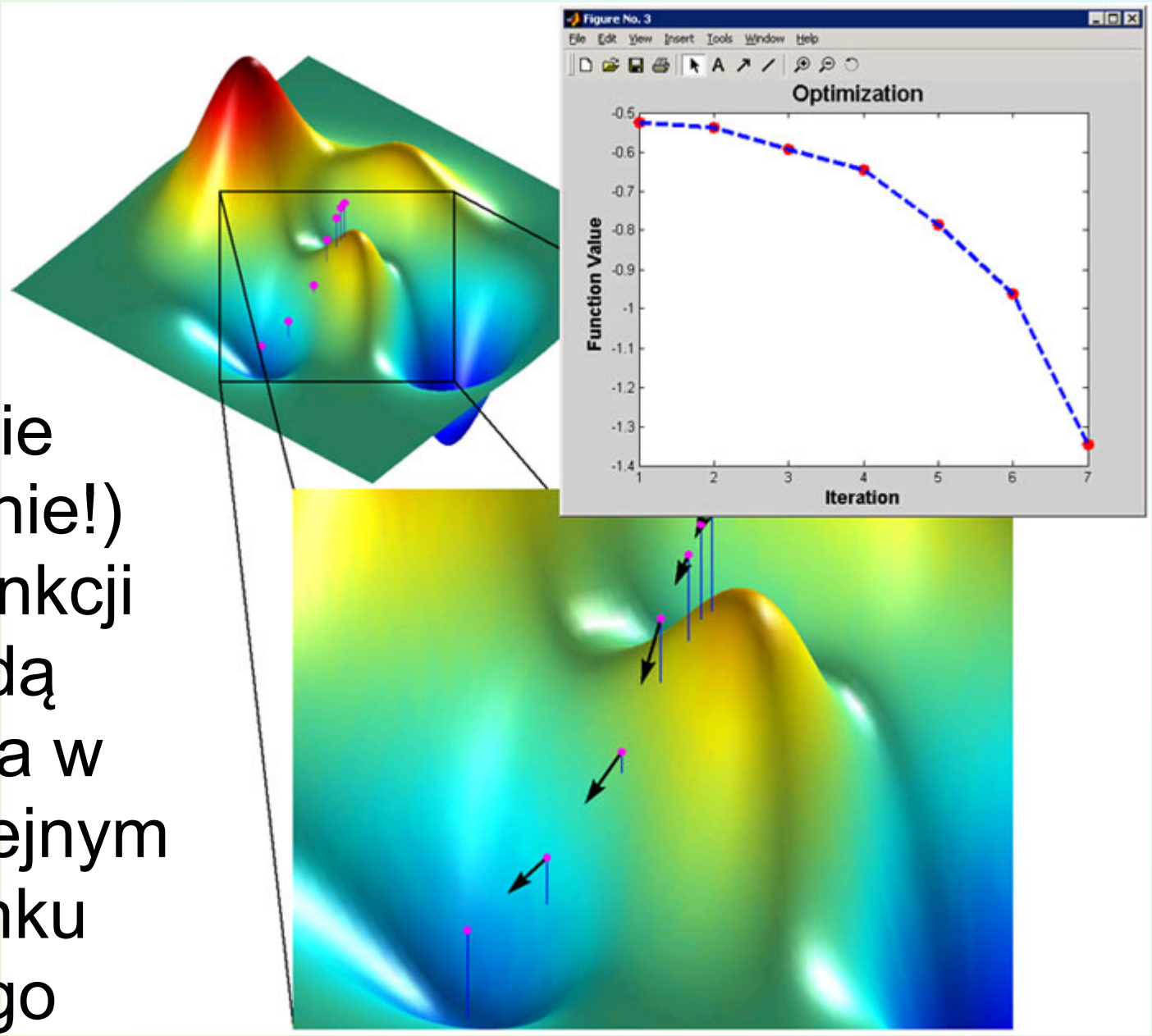
błąd średniokwadratowy *Mean Squared Error* (MSE), spierwiastkowany błąd średniokwadratowy *Root Mean Squared Error* (RMSE) oraz spierwiastkowany znormalizowany błąd średniokwadratowy *Root Mean Square Normalized Error* (RMSNE)

	Error metric	Formula
1	MSE	$\frac{\sum_{i=1}^{n}(y_i - h_i)^2}{n}$
2	RMSE	$\sqrt{\frac{\sum_{i=1}^{n}(y_i - h_i)^2}{n}}$
3	RMSNE	$\sqrt{\frac{\sum_{i=1}^{n}\left(\frac{y_i - h_i}{y_i}\right)^2}{n}}$


Błąd
$\delta^j = z^j - y^j$
Istota uczenia polega na szukaniu miejsca, w którym błąd jest minimalny
$-\eta\frac{\partial Q^j}{\partial w_i}$
$-\eta\frac{\partial Q^j}{\partial w_i}$
minimum funkcji błędu
waga w_2
$w_{ideal.}$
w_n
w_s
waga w_1
Δ_w

$$-\eta \frac{\partial Q^j}{\partial w_i}$$

Poszukiwanie (i znajdowanie!) minimum funkcji błędu metodą wyznaczania w każdym kolejnym kroku kierunku najszybszego spadku



Reguły uczenia pojedynczego neuronu

Tzw. reguła delta w oparciu o błąd

$$\delta^j = z^j - y^j = z^j - \mathbf{w}^T \mathbf{x}^j$$

Neuron liniowy

$$\Delta w_i = \eta \cdot \delta^j \cdot x_i^j$$

Neuron nieliniowy (sigmoidalna funkcja)

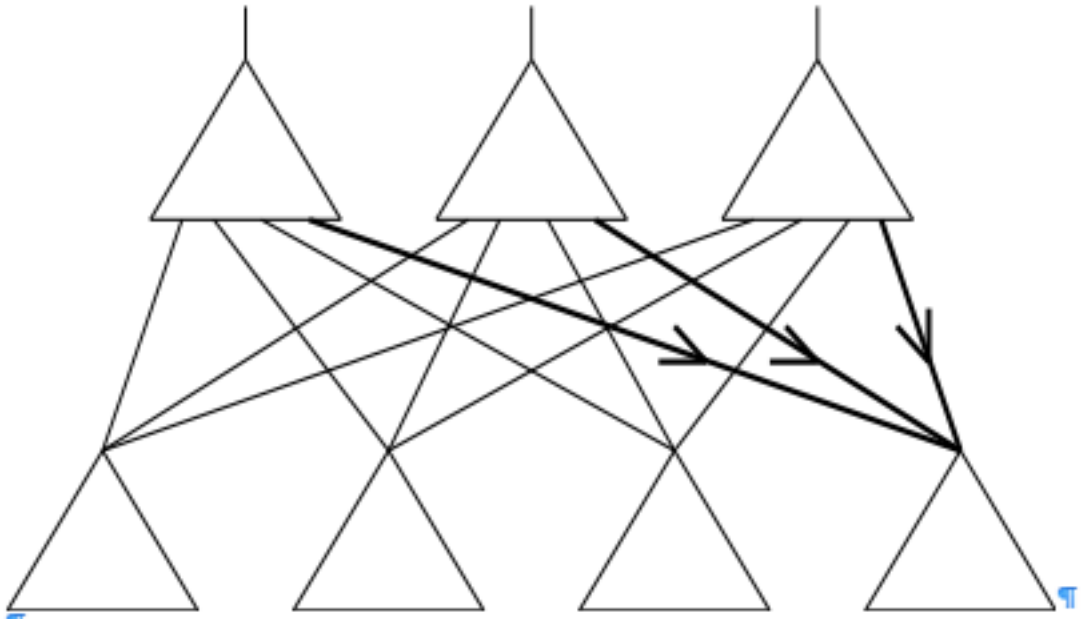
$$\Delta w_i = \eta \cdot \delta^j \cdot (1 - y^j) \cdot y^j \cdot x_i^j$$

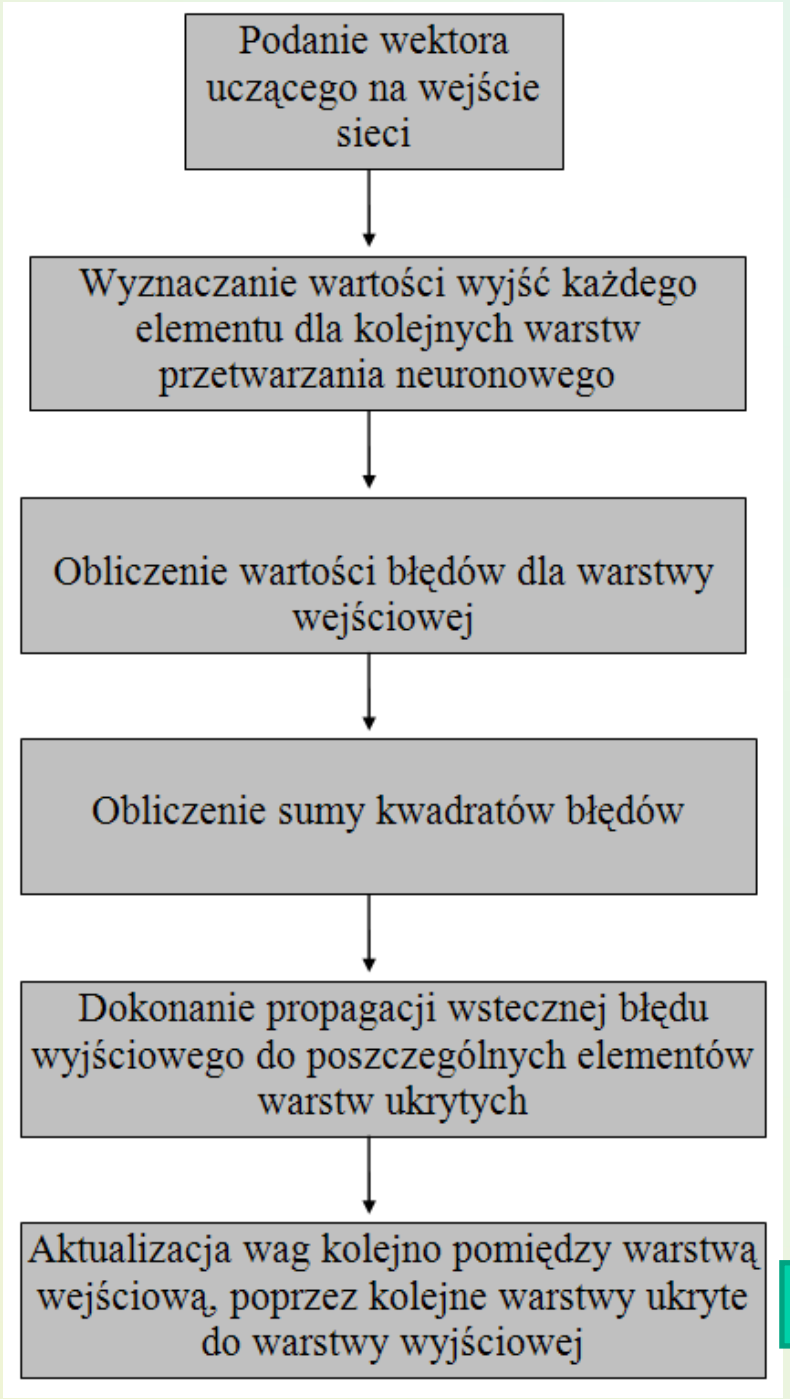
Detale i wyprowadzanie = patrz główny wykład
oraz laboratorium

Algorytm wstecznej propagacji błędów

Błąd k-tego neuronu w l-tej warstwie jest równy sumie błędów popełnionych przez neurony (p) z warstwy l+1-szej ważonych po wagach $w_{k(p,l+1)}$ łączących ten neuron z neuronami tej warstwy:

$$\delta^{j}_{(k,l)} = \sum_{p=1}^{N_{l+1}} w^{j}_{k(p,l+1)} \delta^{j}_{(p,l+1)}$$

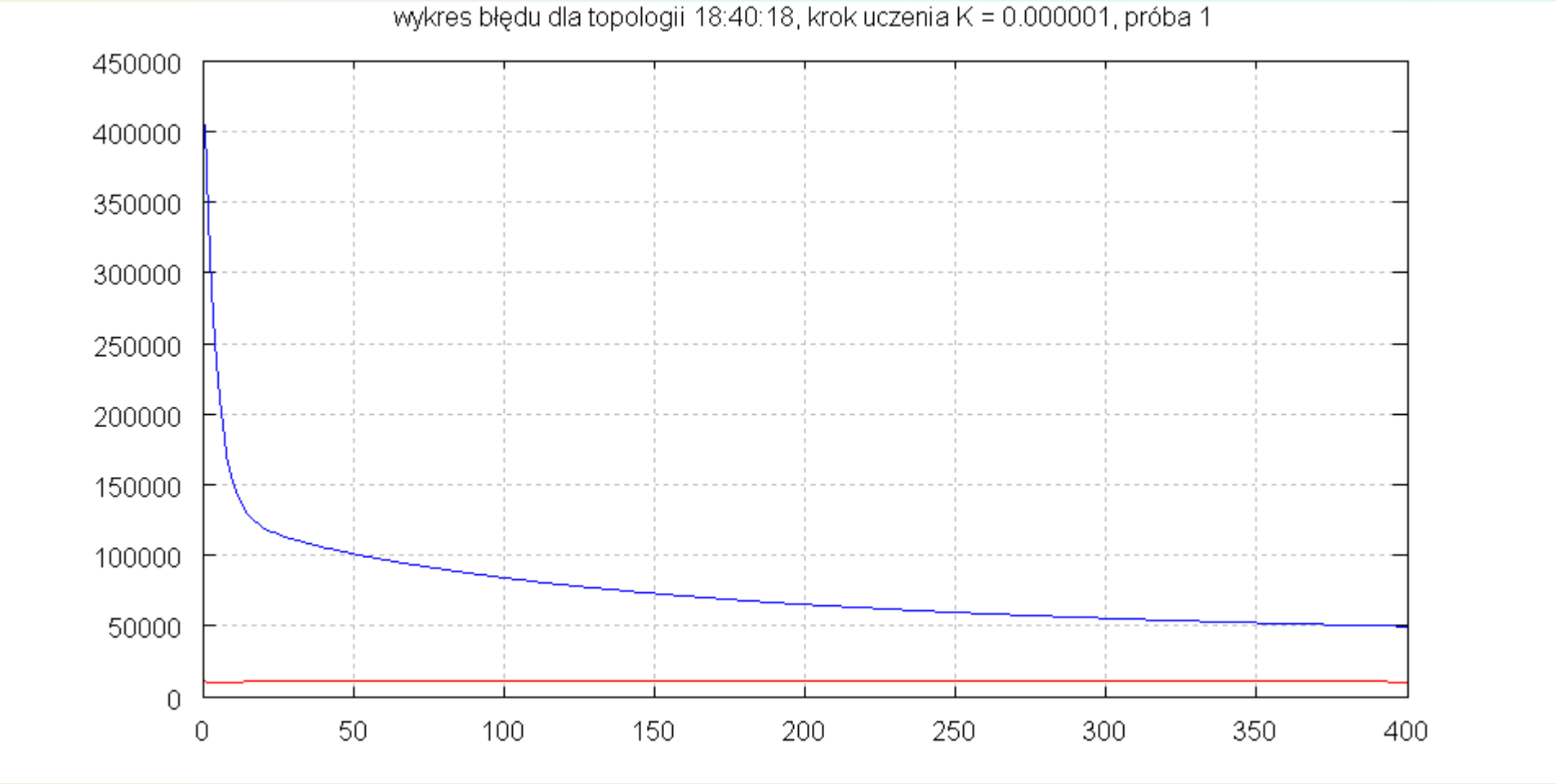




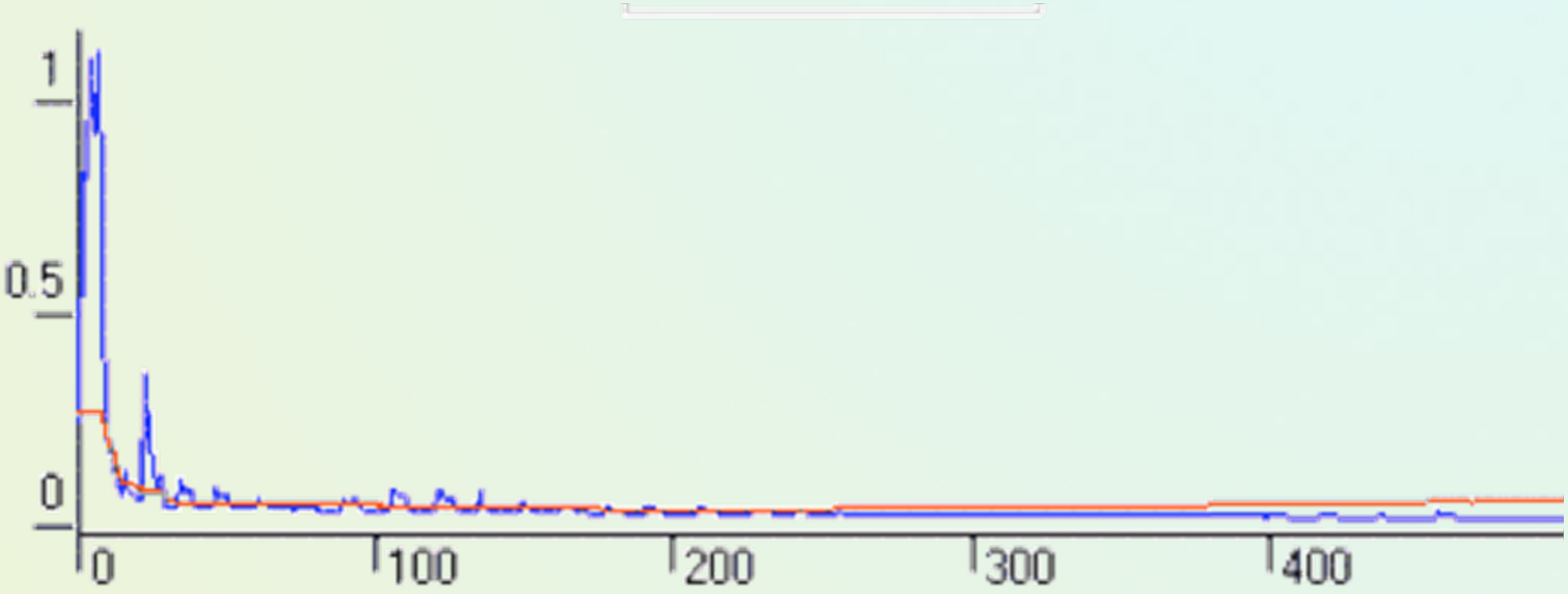
Istota algorytmu uczenia ze wsteczną propagacją błędu

Sieć uzyskuje nowe wagi i ma mniejszy błąd

Typowy przebieg błędu w trakcie procesu uczenia



W czasie uczenia sieci obserwuje się często chwilowe pogorszenia wartości błędu



Wynika to z faktu, że w zbiorze uczącym bywają przypadki szczególnie trudne lub nietypowe, których użycie zawsze na chwilę psuje wyniki uczenia. Zjawisko to jednak zwykle szybko zanika.

Powierzchnie błędów bywają dosyć skomplikowane!