

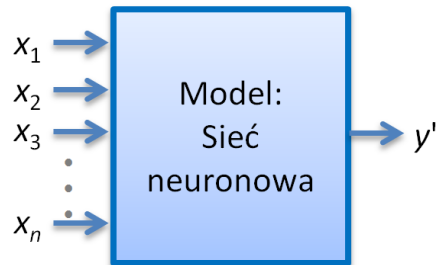
SZTUCZNE SIECI NEURONOWE JAKO APROKSYMATORY

GRZEGORZ DUDEK

*Instytut Informatyki
Wydział Elektryczny
Politechnika Częstochowska
gdudek.el.pcz.pl*

1. Wstęp
2. Problem aproksymacji funkcji
3. Model neuronu
4. Aproksymacja funkcji złożonych
5. Sztuczna sieć neuronowa
6. Uczenie sieci neuronowych
7. *Extreme learning machine*
8. Sieć o radialnych funkcjach bazowych
9. Sieć regresji uogólnionej GRNN
10. Przykłady zastosowań sieci neuronowych

Modelowanie



Cechy sztucznych sieci neuronowych

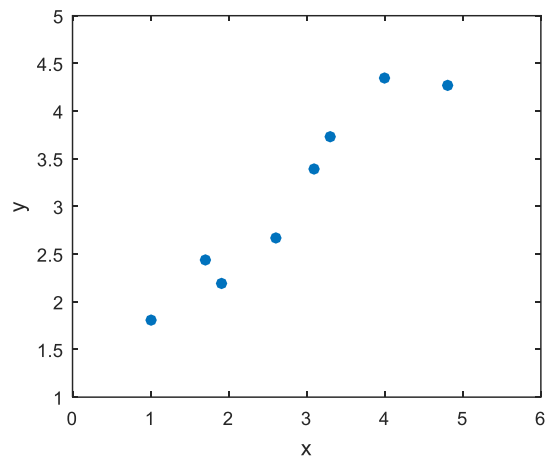
- własność uniwersalnego aproksymatora
- uczenie się, pozyskiwanie wiedzy z danych
- odporność na zakłócenia danych
- równoległa architektura
- adaptacyjność

Zastosowania sztucznych sieci neuronowych

- sterowanie procesów przemysłowych
- identyfikacja systemów
- rozpoznawanie i klasyfikacja wzorców
- predykcja szeregów czasowych
- analiza danych statystycznych
- rozpoznawanie sekwencji
- odsumianie i kompresja obrazu i dźwięku
- diagnostyka techniczna i medyczna
- prognozowanie
- ...

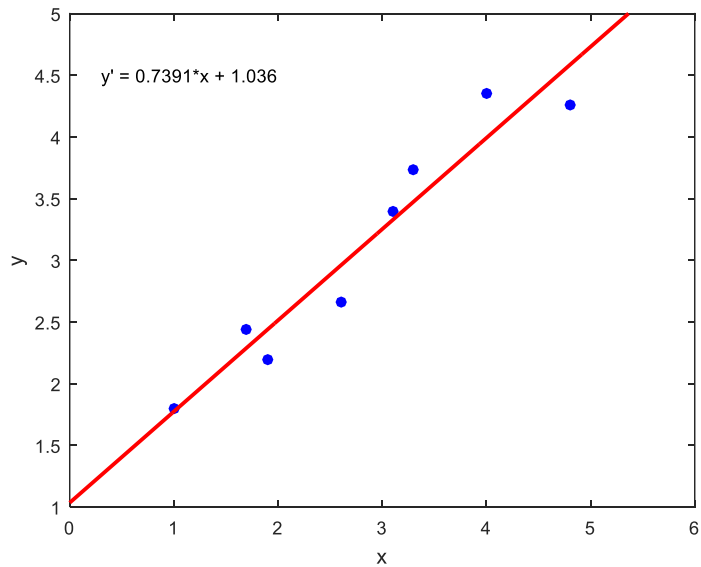
Aproksymacja funkcji polega na przybliżeniu pewnej funkcji $f(x)$ danej w postaci zbioru punktów (np. pomiarowych) funkcją $h(x)$ zapisaną wzorem.

x	1,00	1,70	1,90	2,60	3,10	3,30	4,00	4,80
y	1,80	2,44	2,19	2,66	3,39	3,73	4,34	4,26



Liniowa funkcja aproksymująca:

$$h(x) = ax + b$$

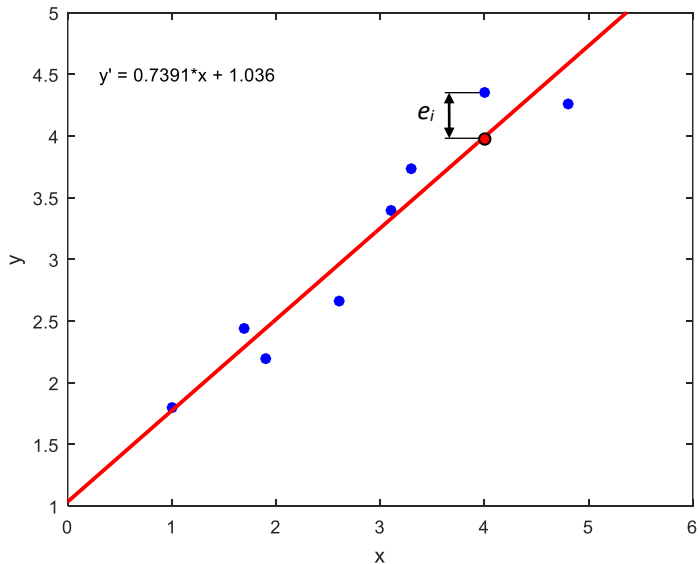


Błąd aproksymacji

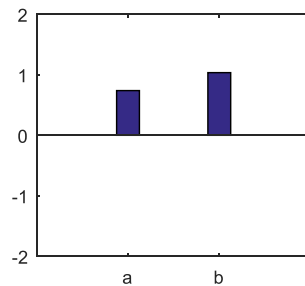
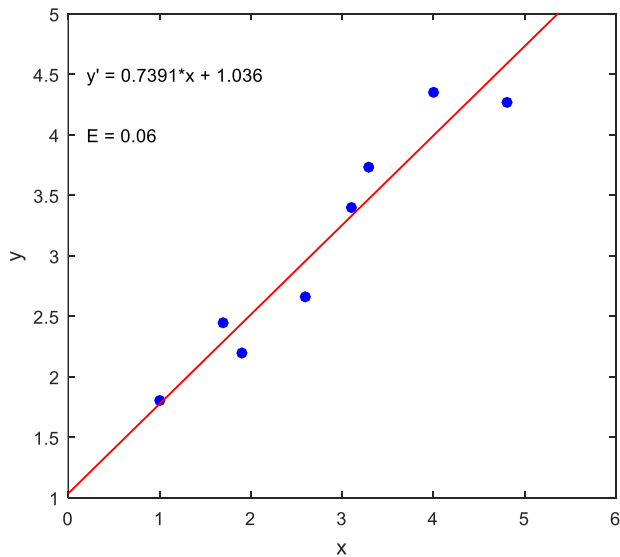
Miarą przybliżenia funkcji $f(x)$ przez funkcję $h(x)$ jest najczęściej średni błąd kwadratowy (*mean squared error, MSE*) :

$$E = \frac{1}{M} \sum_{i=1}^M [y_i - h(x_i)]^2 = \frac{1}{M} \sum_{i=1}^M e_i^2$$

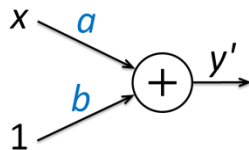
$$E \rightarrow \min$$



APROKSYMACJA FUNKCJI



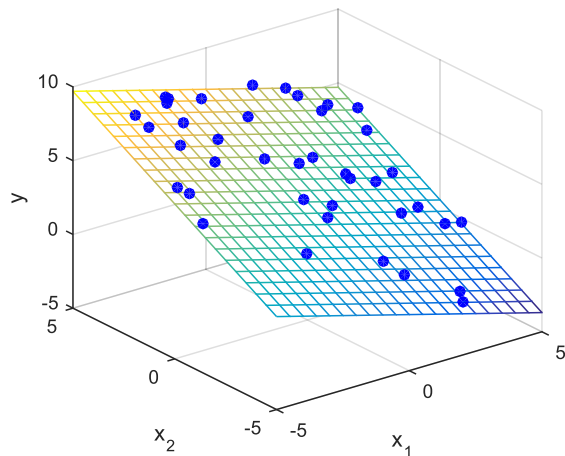
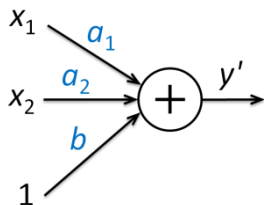
$$h(x) = ax + b$$



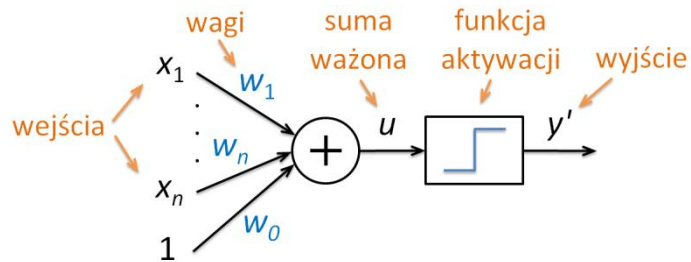
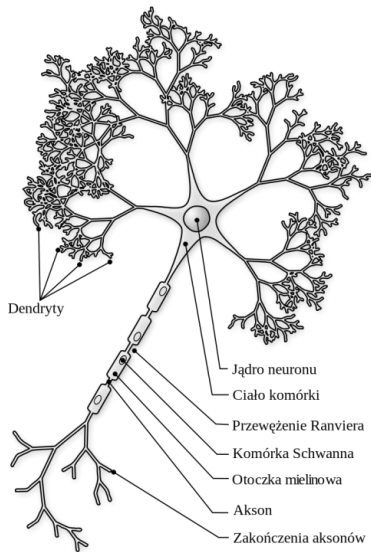
APROKSYMACJA FUNKCJI

x_1	-1,96	-4,54	-3,05	2,20	2,22	3,78	...	-4,29
x_2	4,44	0,49	2,28	0,77	-4,74	-0,53	...	0,21
y	7,71	5,68	6,44	2,54	-2,09	0,66	...	5,32

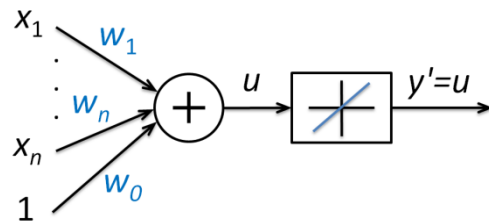
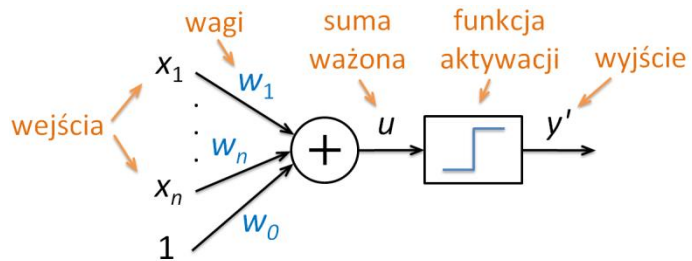
$$h(x) = a_1x_1 + a_2x_2 + b$$



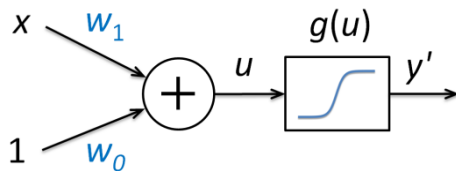
Neuron McCullocha-Pittsa



MODEL NEURONU



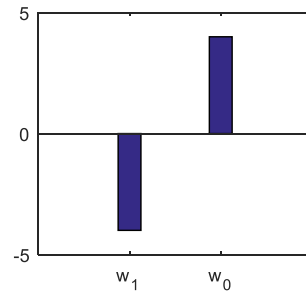
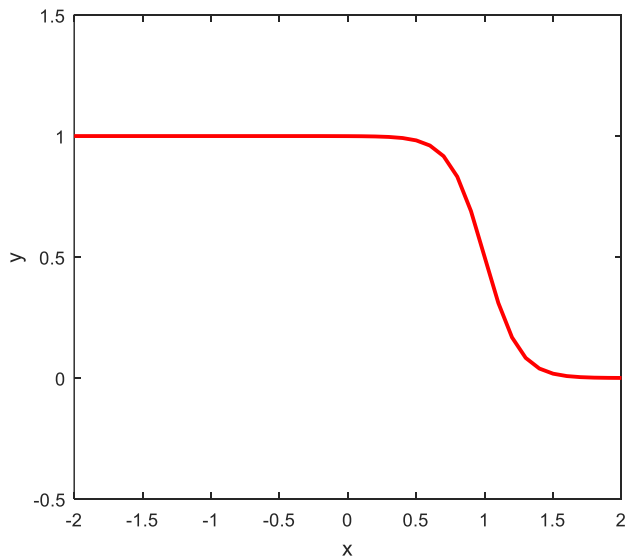
$$h(x) = w_1x_1 + w_2x_2 + \dots + w_nx_n + w_0 = \sum_{i=1}^n w_ix_i + w_0$$



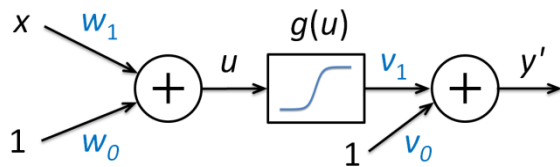
$$g(u) = \frac{1}{1 + \exp(-\beta u)} = \frac{1}{1 + \exp[-\beta(w_1 x + w_0)]}$$

$$\beta = \text{const} \neq 0$$

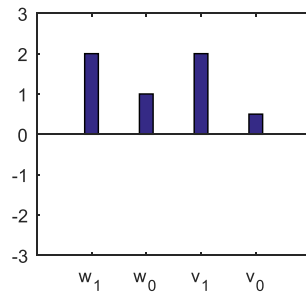
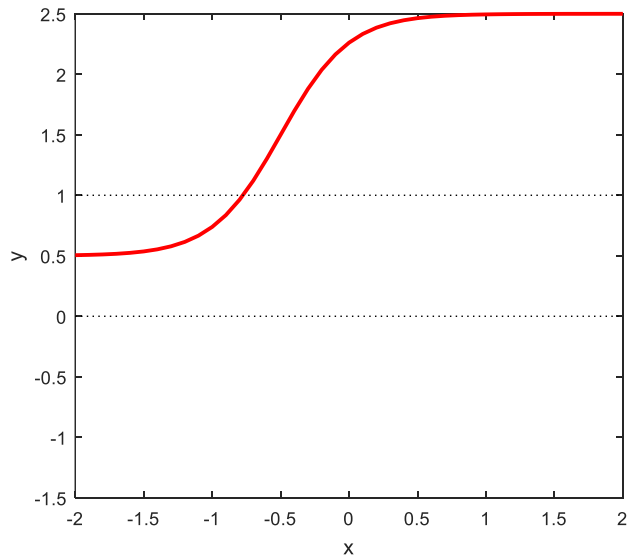
$$g(u) = \frac{1}{1 + \exp(-\beta u)} = \frac{1}{1 + \exp[-\beta(w_1 x + w_0)]}$$



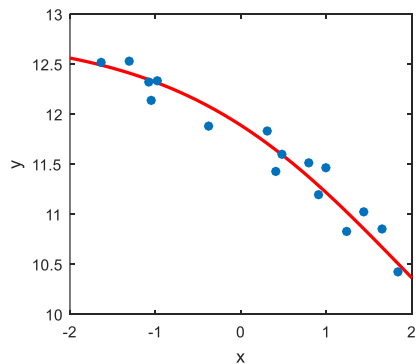
$$h(x) = g(u)v_1 + v_0 = \frac{1}{1 + \exp[-\beta(w_1x + w_0)]}v_1 + v_0$$



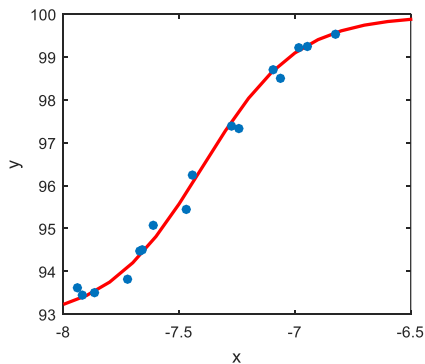
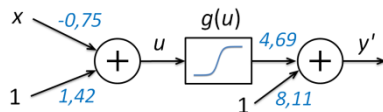
MODEL NEURONU



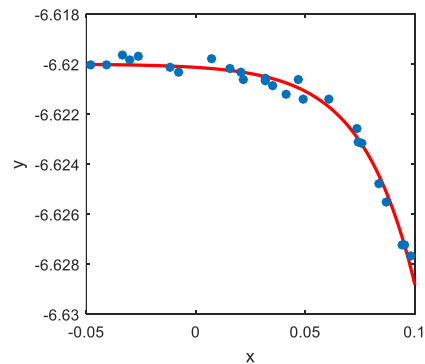
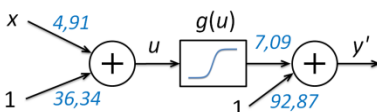
MODEL NEURONU



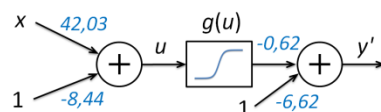
$$h(x) = \frac{1}{1 + \exp[-\beta(-0,75x + 1,42)]} \cdot 4,69 + 8,11$$



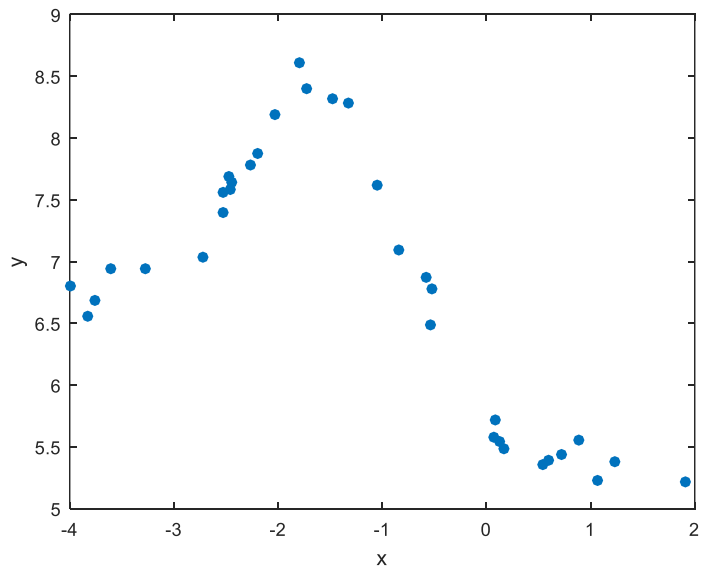
$$h(x) = \frac{1}{1 + \exp[-\beta(4,91x + 36,34)]} \cdot 7,09 + 92,87$$



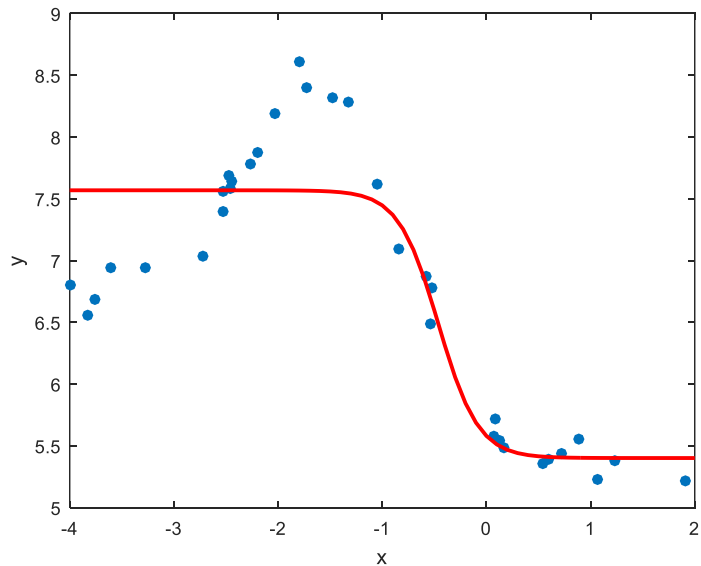
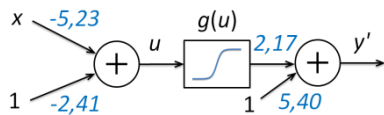
$$h(x) = \frac{1}{1 + \exp[-\beta(42,03x - 8,44)]} \cdot (-0,62) - 6,62$$



MODEL NEURONU

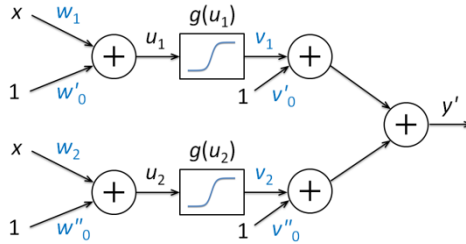


APROKSYMACJA FUNKCJI ZŁOŻONYCH



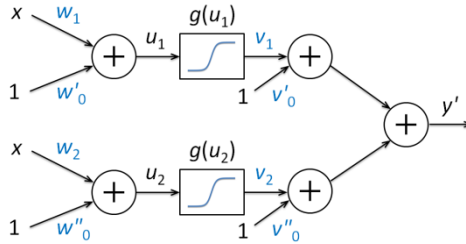
APROKSYMACJA FUNKCJI ZŁOŻONYCH

$$h(x) = [g(u_1)v_1 + v'_0] + [g(u_2)v_2 + v''_0]$$

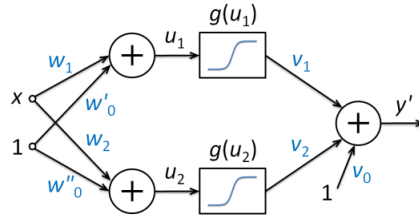


APROKSYMACJA FUNKCJI ZŁOŻONYCH

$$h(x) = [g(u_1)v_1 + v'_0] + [g(u_2)v_2 + v''_0]$$

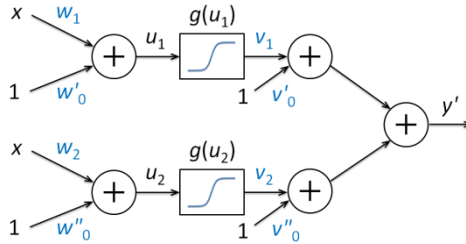


$$h(x) = g(u_1)v_1 + g(u_2)v_2 + v_0$$

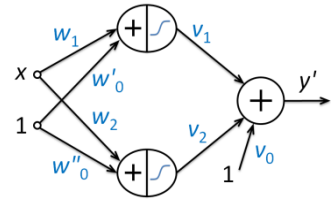
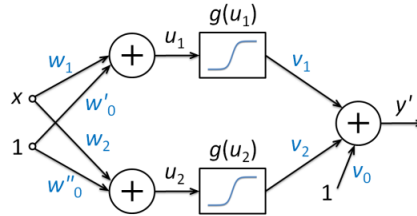


APROKSYMACJA FUNKCJI ZŁOŻONYCH

$$h(x) = [g(u_1)v_1 + v'_0] + [g(u_2)v_2 + v''_0]$$

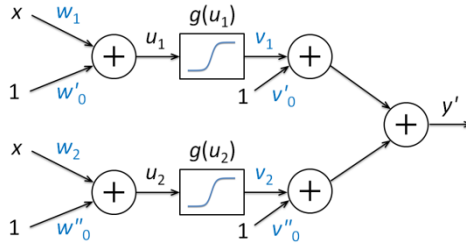


$$h(x) = g(u_1)v_1 + g(u_2)v_2 + v_0$$

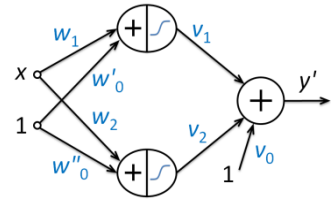
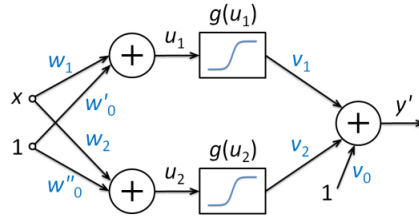


APROKSYMACJA FUNKCJI ZŁOŻONYCH

$$h(x) = [g(u_1)v_1 + v'_0] + [g(u_2)v_2 + v''_0]$$

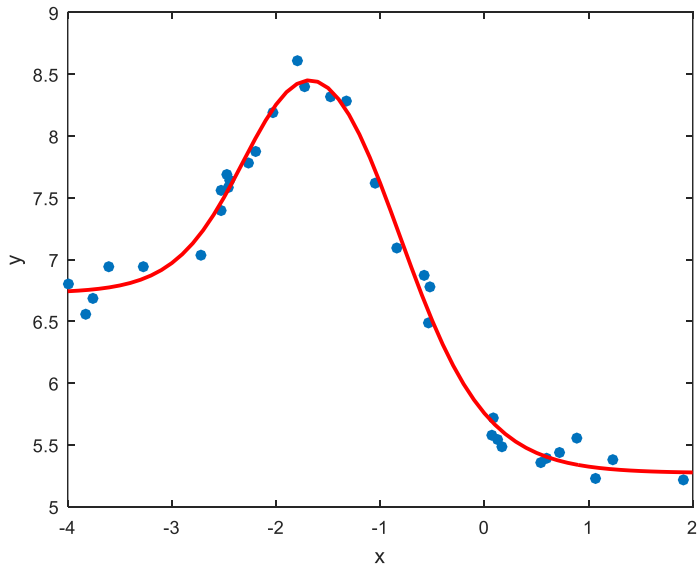
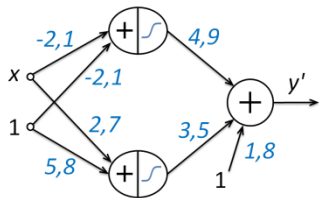


$$h(x) = g(u_1)v_1 + g(u_2)v_2 + v_0$$

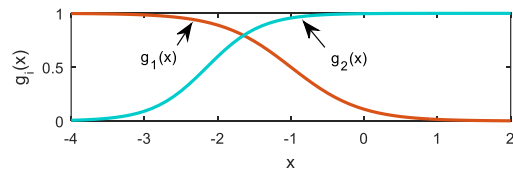
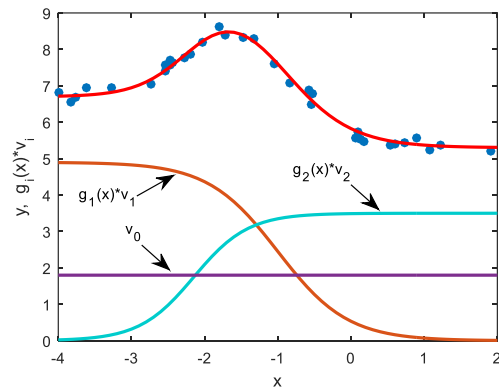
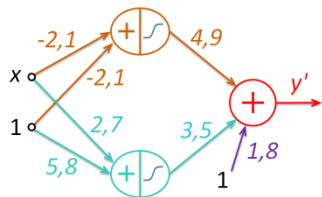


$$h(x) = g_1(x)v_1 + g_2(x)v_2 + v_0 = \frac{1}{1 + \exp[-\beta(w_1x + w'_0)]}v_1 + \frac{1}{1 + \exp[-\beta(w_2x + w''_0)]}v_2 + v_0$$

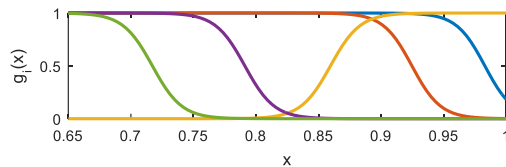
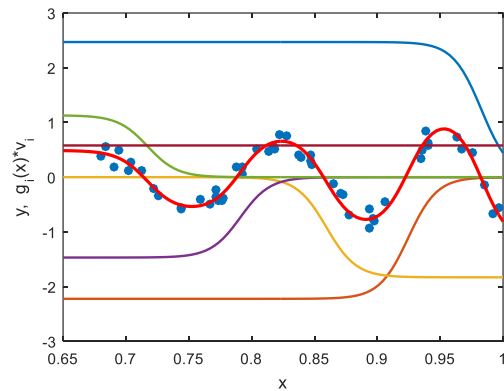
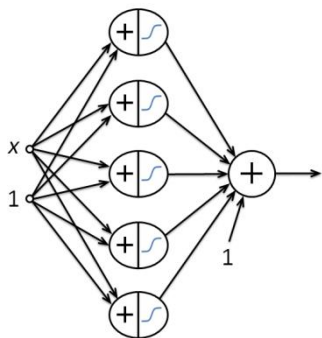
APROKSYMACJA FUNKCJI ZŁOŻONYCH



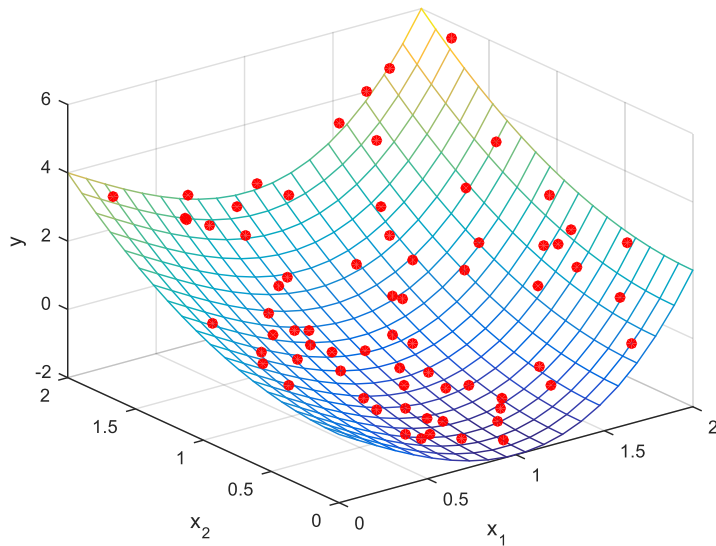
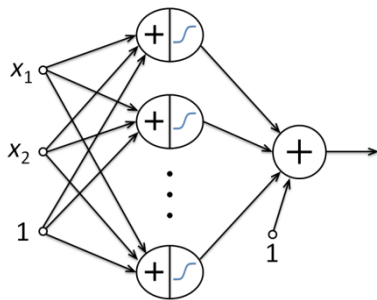
APROKSYMACJA FUNKCJI ZŁOŻONYCH



APROKSYMACJA FUNKCJI ZŁOŻONYCH

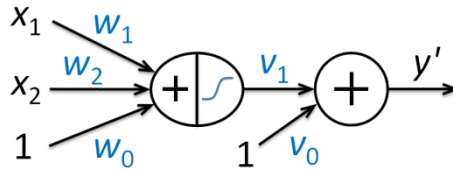


APROKSYMACJA FUNKCJI ZŁOŻONYCH

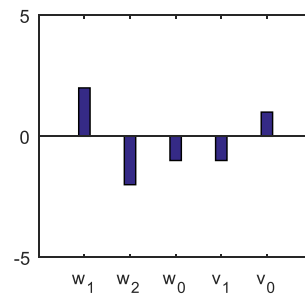
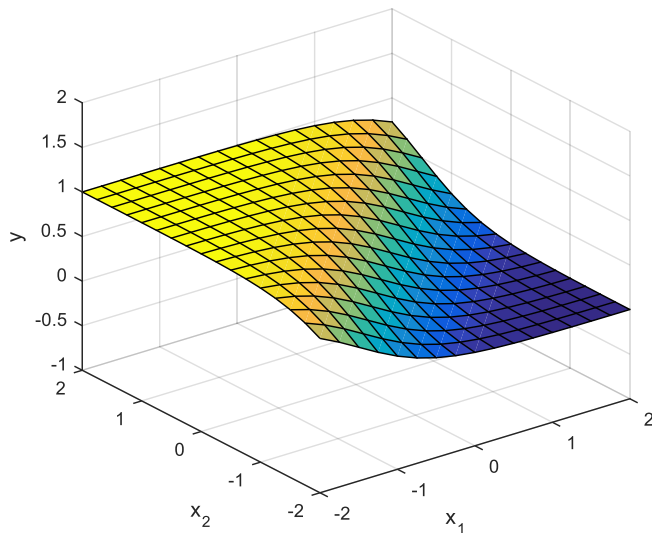


APROKSYMACJA FUNKCJI ZŁOŻONYCH

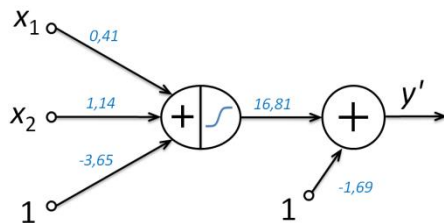
$$h(x) = \frac{1}{1 + \exp[-\beta(w_1x_1 + w_2x_2 + w_0)]} v_1 + v_0$$



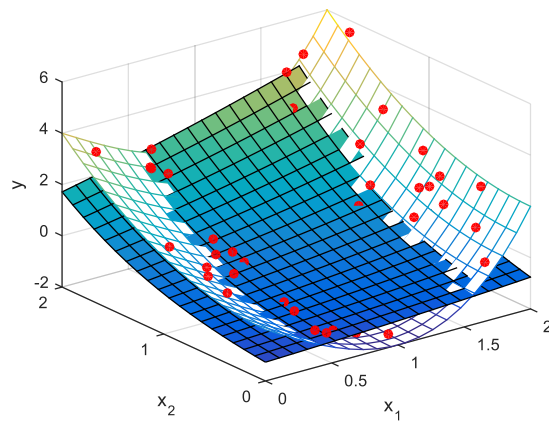
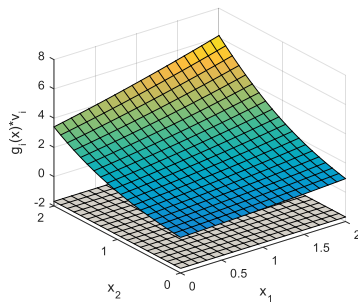
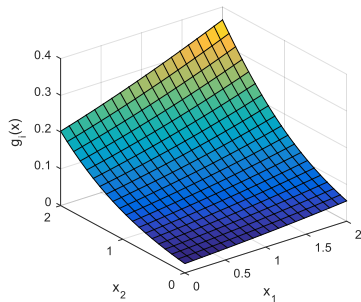
APROKSYMACJA FUNKCJI ZŁOŻONYCH



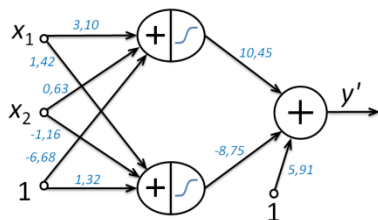
APROKSYMACJA FUNKCJI ZŁOŻONYCH



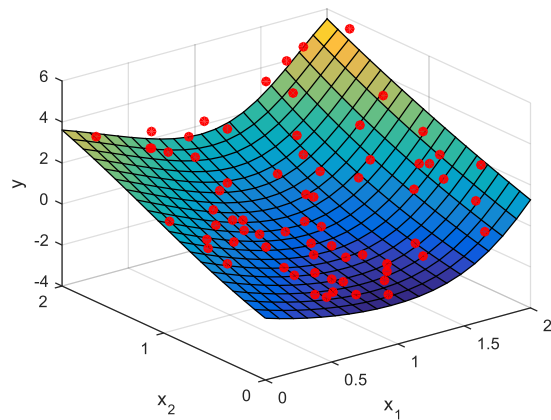
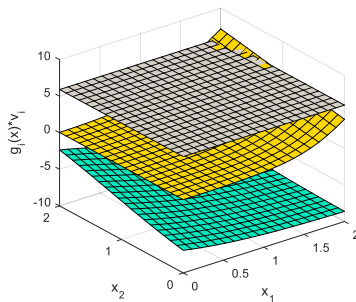
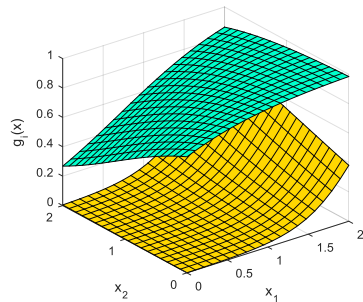
$$E = 0,9881$$



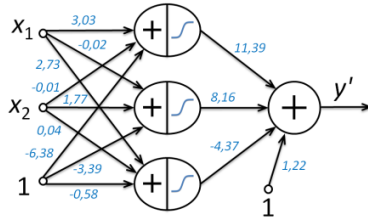
APROKSYMACJA FUNKCJI ZŁOŻONYCH



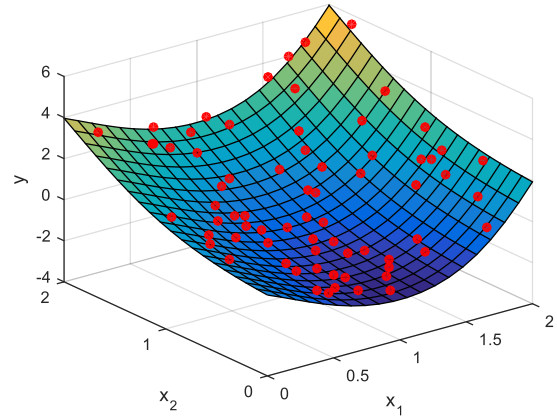
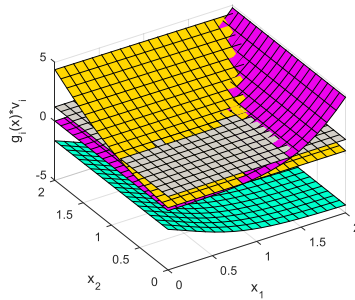
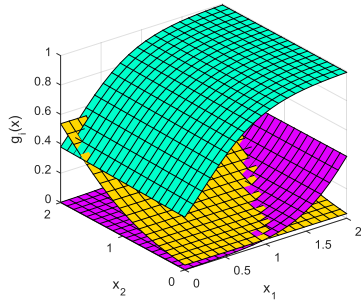
$$E = 0,0524$$



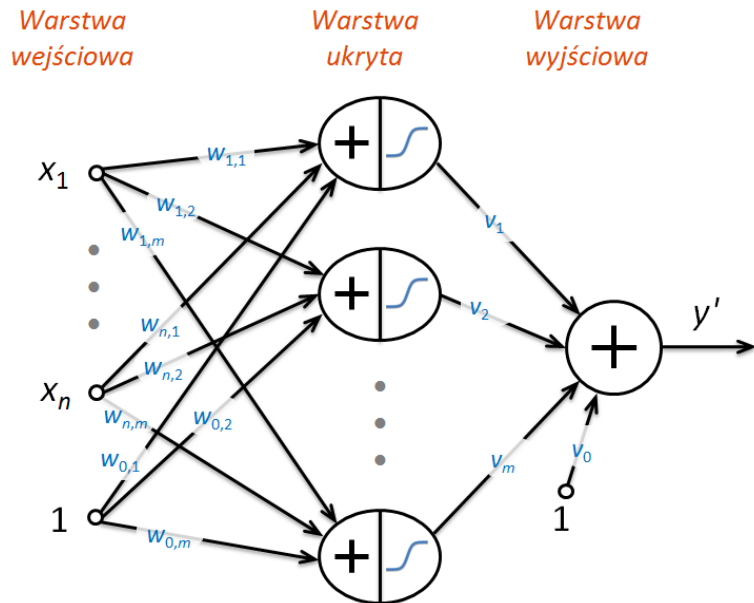
APROKSYMACJA FUNKCJI ZŁOŻONYCH



$E = 0,0006$

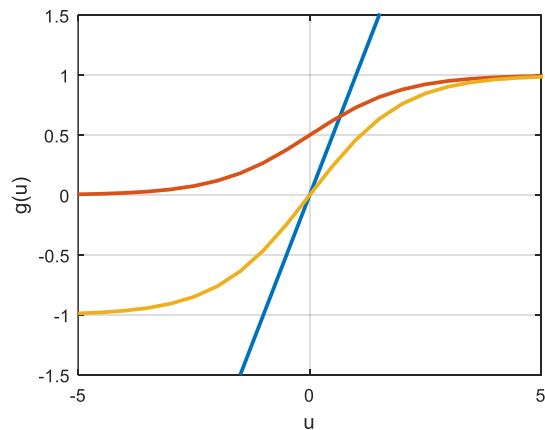


SZTUCZNA SIĘĆ NEURONOWA



$$h(\mathbf{x}) = \sum_{i=1}^m g_i(\mathbf{x})v_i + v_0$$

Funkcje aktywacji neuronów



— – sigmoidalna unipolarna

$$g(u) = \frac{1}{1 + \exp(-\beta u)}$$

— – sigmoidalna bipolarna

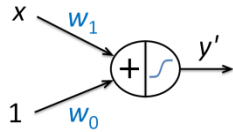
$$g(u) = \frac{2}{1 + \exp(-\beta u)} - 1$$

— – liniowa

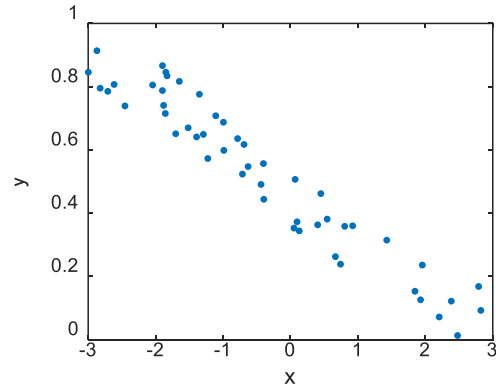
$$g(u) = u$$

$$u = \sum_{i=1}^n w_i x_i + w_0 \quad \beta = \text{const} \neq 0$$

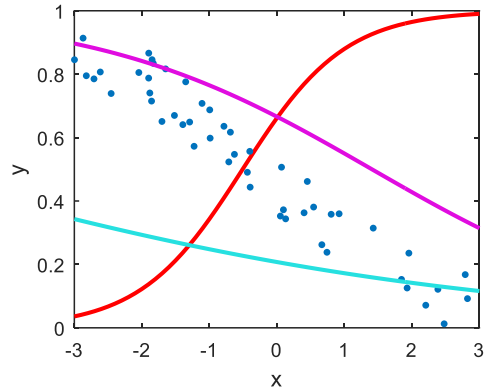
UCZENIE SIECI NEURONOWYCH



$$h(x) = \frac{1}{1 + \exp[-\beta(w_1 x_1 + w_0)]}$$



UCZENIE SIECI NEURONOWYCH



$$w_1 = 1.32$$

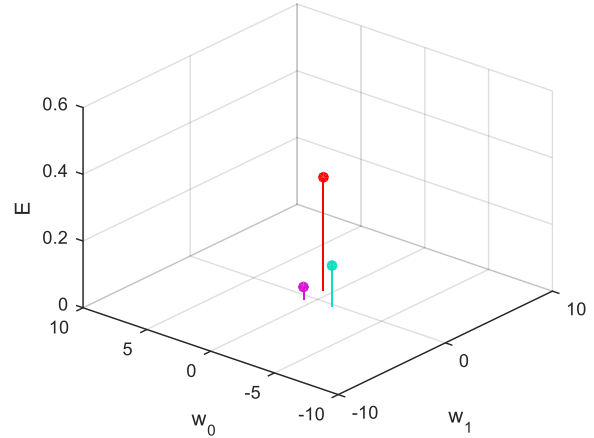
$$w_0 = 0.67$$

$$w_1 = -0.49$$

$$w_0 = 0.69$$

$$w_1 = -0.23$$

$$w_0 = -1.34$$

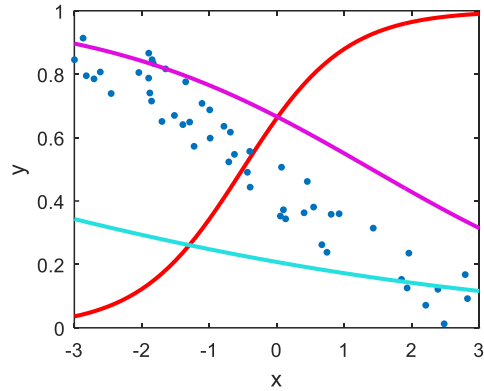


$$E = 0.3352$$

$$E = 0.0378$$

$$E = 0.1236$$

UCZENIE SIECI NEURONOWYCH



$$w_1 = 1.32$$

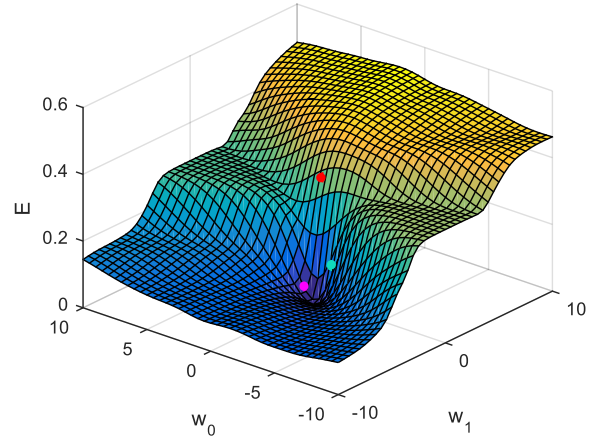
$$w_0 = 0.67$$

$$w_1 = -0.49$$

$$w_0 = 0.69$$

$$w_1 = -0.23$$

$$w_0 = -1.34$$



$$E = 0.3352$$

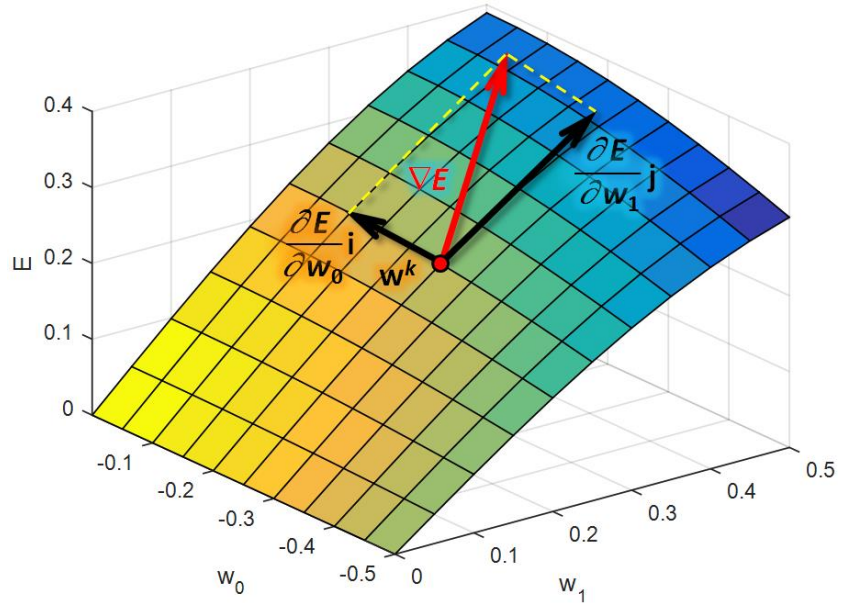
$$E = 0.0378$$

$$E = 0.1236$$

UCZENIE SIECI NEURONOWYCH

$$\nabla E(\mathbf{w}) = \frac{\partial E(\mathbf{w})}{\partial w_0} \mathbf{i} + \frac{\partial E(\mathbf{w})}{\partial w_1} \mathbf{j}$$

$$\nabla E(\mathbf{w}) = \begin{bmatrix} \frac{\partial E(\mathbf{w})}{\partial w_0} & \frac{\partial E(\mathbf{w})}{\partial w_1} \end{bmatrix}$$



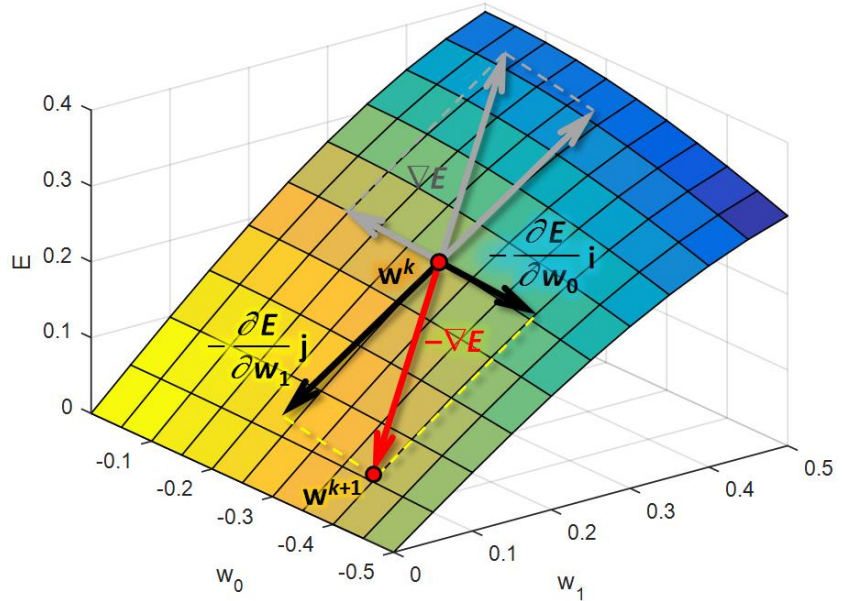
UCZENIE SIECI NEURONOWYCH

$$\nabla E(\mathbf{w}) = \frac{\partial E(\mathbf{w})}{\partial w_0} \mathbf{i} + \frac{\partial E(\mathbf{w})}{\partial w_1} \mathbf{j}$$

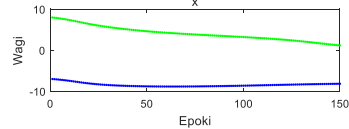
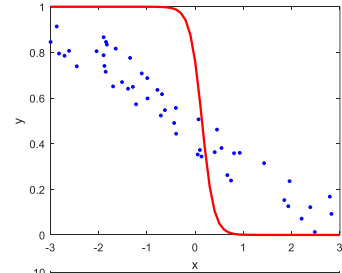
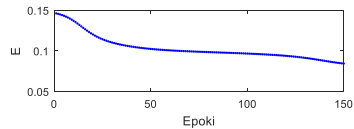
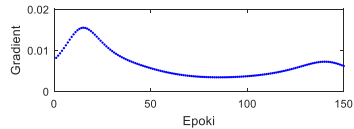
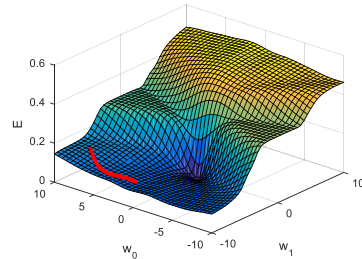
$$\nabla E(\mathbf{w}) = \begin{bmatrix} \frac{\partial E(\mathbf{w})}{\partial w_0} & \frac{\partial E(\mathbf{w})}{\partial w_1} \end{bmatrix}^T$$

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \eta \nabla E(\mathbf{w}^k)$$

$$\mathbf{w}^{k+1} = \begin{bmatrix} w_0^k - \eta \frac{\partial E(\mathbf{w}^k)}{\partial w_0} & w_1^k - \eta \frac{\partial E(\mathbf{w}^k)}{\partial w_1} \end{bmatrix}^T$$



UCZENIE SIECI NEURONOWYCH



Wagi startowe: $w_1 = -7.00$, $w_0 = 8.00$

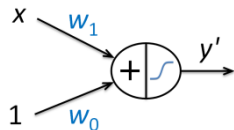
$\eta = 0.05$

Epoka = 150

Gradient = 6.13×10^{-3}

$E = 0.0843$

$w_1 = -8.19$, $w_0 = 1.14$



$$g(x) = \frac{1}{1 + \exp[-\beta(w_1x + w_0)]}$$

$$E = \frac{1}{2} (y^* - y)^2 = \frac{1}{2} [g(u) - y]^2 = \frac{1}{2} \left[\frac{1}{1 + \exp[-\beta(w_1x + w_0)]} - y \right]^2$$

$$\frac{\partial E(\mathbf{w})}{\partial w_0} = [g(u) - y] \frac{\partial g(u)}{\partial u} \frac{\partial u}{\partial w_0} = [g(u) - y] \frac{\partial g(u)}{\partial u}$$

$$\frac{\partial E(\mathbf{w})}{\partial w_1} = [g(u) - y] \frac{\partial g(u)}{\partial u} \frac{\partial u}{\partial w_1} = [g(u) - y] \frac{\partial g(u)}{\partial u} x$$

$$\frac{\partial g(u)}{\partial u} = \beta[1 - g(u)]g(u)$$

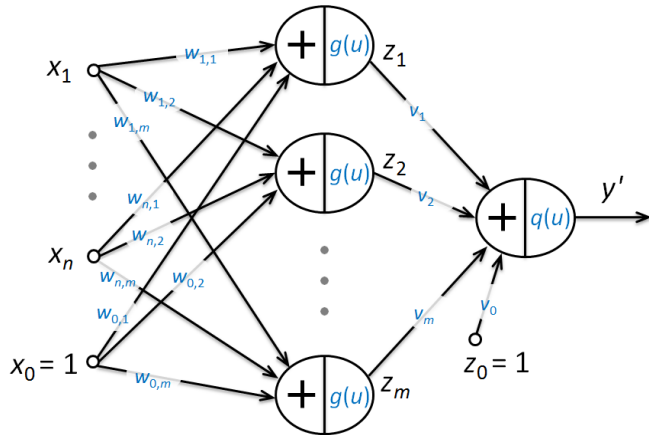
$$\frac{\partial E(\mathbf{w})}{\partial w_0} = [g(u) - y]\beta[1 - g(u)]g(u)$$

$$\frac{\partial E(\mathbf{w})}{\partial w_1} = [g(u) - y]\beta[1 - g(u)]g(u)x$$

$$w_0^{k+1} = w_0^k - \eta \frac{\partial E(\mathbf{w}^k)}{\partial w_0} = w_0^k - \eta \beta(y' - y)(1 - y')y'$$

$$w_1^{k+1} = w_1^k - \eta \frac{\partial E(\mathbf{w}^k)}{\partial w_1} = w_1^k - \eta \beta(y' - y)(1 - y')y'x$$

Sieci wielowarstwowe



$$v_i^{k+1} = v_i^k - \eta \delta'' z_i$$

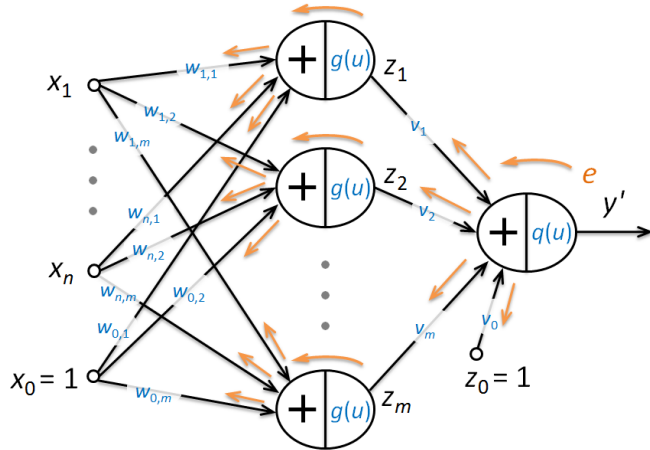
$$\delta'' = (y' - y) \frac{\partial q(u)}{\partial u}$$

$$z_i = g \left(\sum_{j=1}^n w_{j,i} x_j + w_{0,i} \right)$$

$$w_{j,i}^{k+1} = w_{j,i}^k - \eta \delta'_i x_j$$

$$\delta'_i = \frac{\partial g(u_i)}{\partial u_i} v_i \delta''$$

Sieci wielowarstwowe



$$v_i^{k+1} = v_i^k - \eta \delta'' z_i$$

$$\delta'' = (y' - y) \frac{\partial q(u)}{\partial u}$$

$$z_i = g \left(\sum_{j=1}^n w_{j,i} x_j + w_{0,i} \right)$$

$$w_{j,i}^{k+1} = w_{j,i}^k - \eta \delta'_i x_j$$

$$\delta'_i = \frac{\partial g(u_i)}{\partial u_i} v_i \delta''$$

Algorytm uczenia - przyrostowa aktualizacja wag

1. Inicjalizacja: topologii sieci (liczba neuronów ukrytych), wag, η , kryterium stopu
2. Wykonuj cyklicznie do momentu spełnienia kryterium stopu (epoki $k = 1, 2, \dots$)
 - 2.1. Wykonuj M -krotnie
 - 2.1.1. Wybierz punkt uczący $(\mathbf{x}_m, y_m)$
 - 2.1.2. Wyznacz odpowiedź sieci na $(\mathbf{x}_m, y_m) - y'$
 - 2.1.3. Oblicz błąd $e = (y' - y_m)$
 - 2.1.4. Zmodyfikuj wagi: $v_i^{k+1} = v_i^k - \eta \delta'' z_i$, $w_{j,i}^{k+1} = w_{j,i}^k - \eta \delta_i' x_j$

Algorytm uczenia - skumulowana aktualizacja wag

1. Inicjalizacja: topologii sieci (liczba neuronów ukrytych), wag, η , kryterium stopu

2. Wykonuj cyklicznie do momentu spełnienia kryterium stopu (epoki $k = 1, 2, \dots$)

2.1. Wykonuj M -krotnie

2.1.1. Wybierz punkt uczący ($\mathbf{x}_m, y_m$)

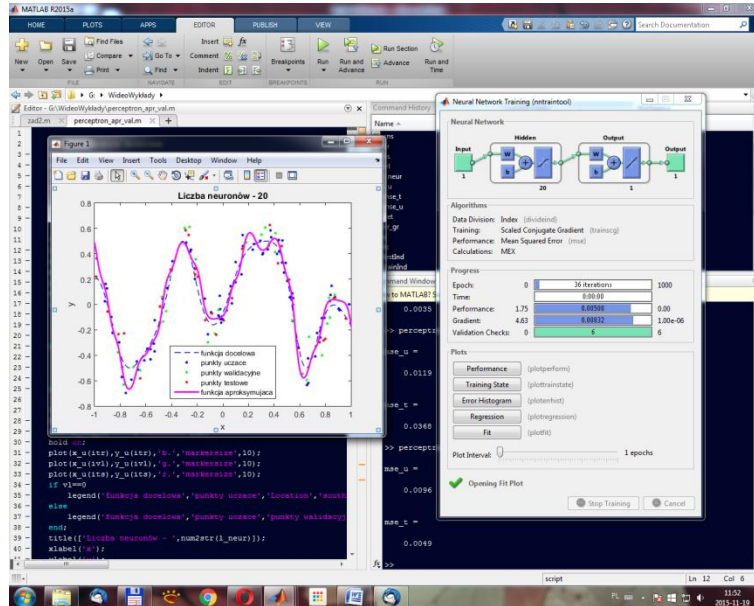
2.1.2. Wyznacz odpowiedź sieci na ($\mathbf{x}_m, y_m$) - y'

2.1.3. Oblicz błąd $e = (y' - y_m)$

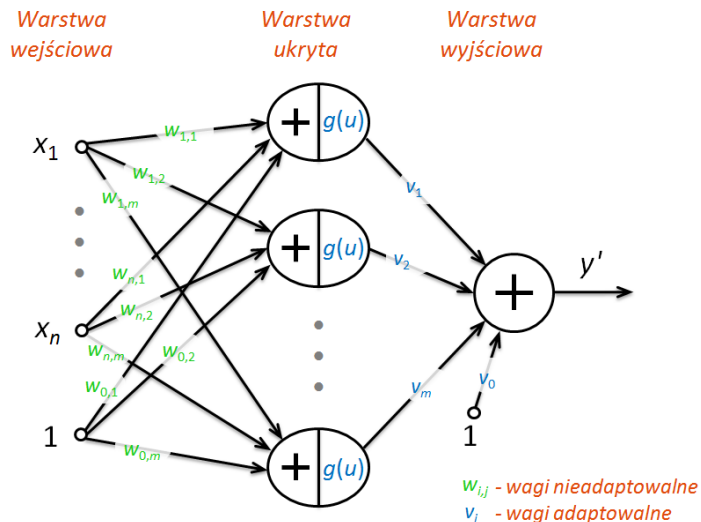
2.1.4. Wyznacz poprawki wag: $\Delta v_i^k = \Delta v_i^k + \eta \delta'' z_i$, $\Delta w_{j,i}^k = \Delta w_{j,i}^k + \eta \delta_i' x_j$

2.2. Zmodyfikuj wagi: $v_i^{k+1} = v_i^k - \Delta v_i^k$, $w_{j,i}^{k+1} = w_{j,i}^k - \Delta w_{j,i}^k$

UCZENIE SIECI NEURONOWYCH

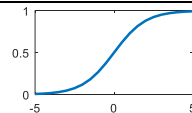


EXTREME LEARNING MACHINE

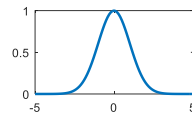


$g(u)$

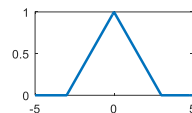
$$\frac{1}{1 + \exp(-u)}$$



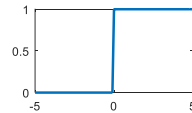
$$\exp(-u^2)$$



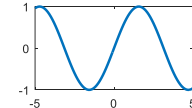
$$\begin{cases} 1 - |u|, & \text{jeśli } |u| \leq 1 \\ 0, & \text{jeśli } |u| > 1 \end{cases}$$



$$\begin{cases} 1, & \text{jeśli } u \leq 0 \\ 0, & \text{jeśli } u > 0 \end{cases}$$

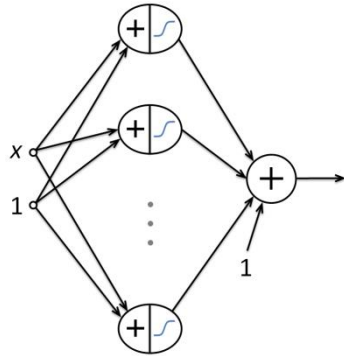


$$\sin(u)$$

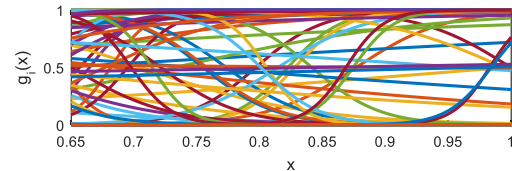
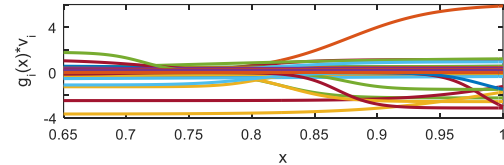
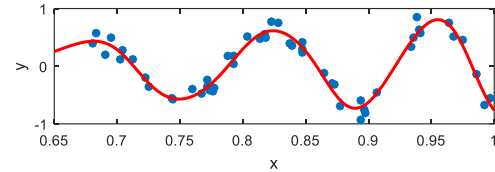
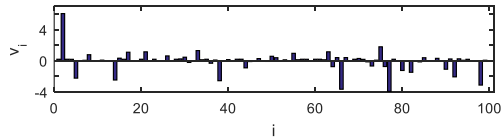


EXTREME LEARNING MACHINE

Konstrukcja aproksymanty



$$h(x) = \sum_{i=1}^m g_i(x)v_i + v_0$$



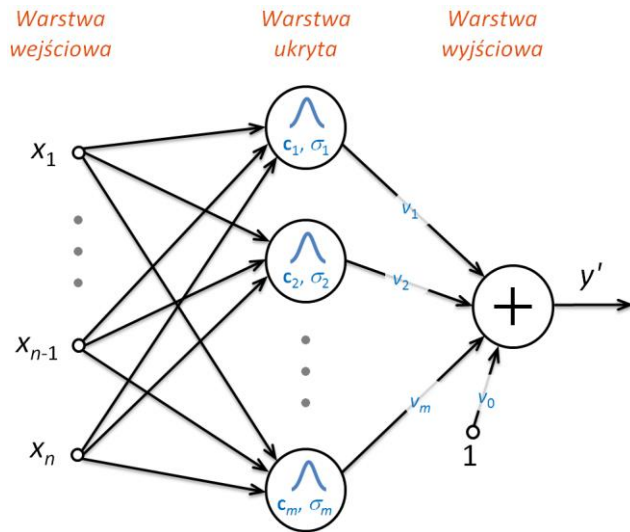
Uczenie sieci ELM

1. Ustal liczbę neuronów w warstwie ukrytej i wylosuj wagi $\mathbf{w}_i$
2. Wyznacz odpowiedzi neuronów ukrytych na wszystkie punkty uczące
3. Oblicz wagi powiązań między warstwą ukrytą i wyjściową

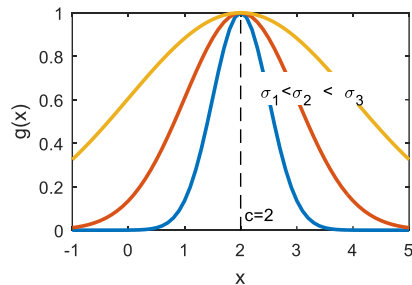
$$\mathbf{v} = (\mathbf{G}'^T \mathbf{G}')^{-1} \mathbf{G}'^T \mathbf{Y}$$

gdzie $\mathbf{G}'$ to macierz odpowiedzi warstwy ukrytej na wszystkie wektory uczące, rozszerzona o wiersz jedynek, $\mathbf{Y}$ – macierz pożądanych odpowiedzi dla wszystkich wektorów uczących.

SIEĆ O RADIALNYCH FUNKCJACH BAZOWYCH



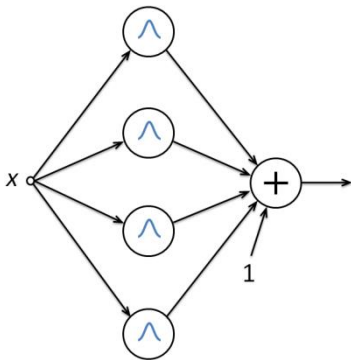
$$g(\mathbf{x}) = \exp\left(-\frac{d^2(\mathbf{x}, \mathbf{c})}{2\sigma^2}\right)$$



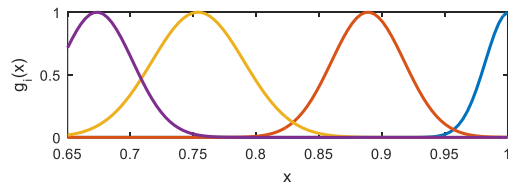
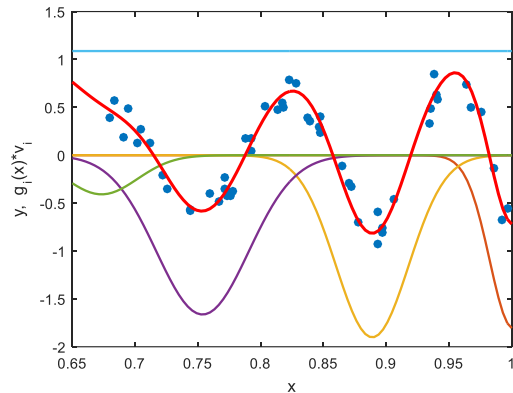
$$h(\mathbf{x}) = \sum_{i=1}^m g_i(\mathbf{x})v_i + v_0$$

SIEĆ O RADIALNYCH FUNKCJACH BAZOWYCH

Konstrukcja aproksymanty



$$h(x) = \sum_{i=1}^m g_i(x)v_i + v_0$$



SIEĆ O RADIALNYCH FUNKCJACH BAZOWYCH

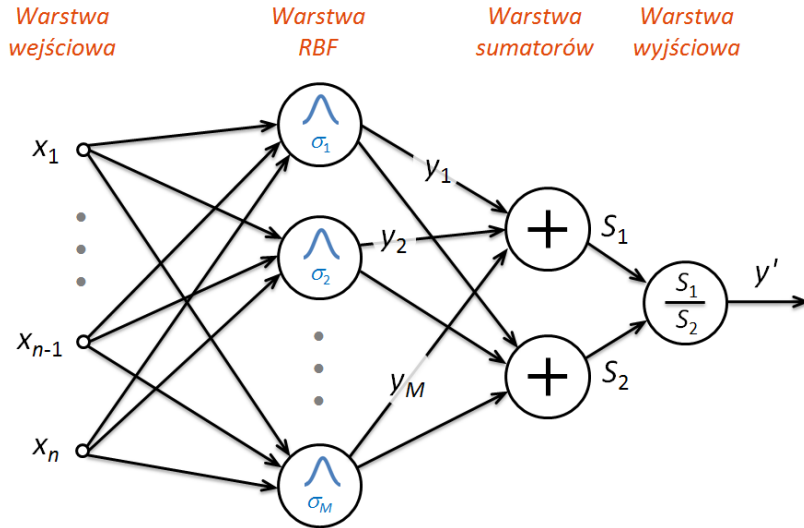
Uczenie sieci RBF

1. Ustal liczbę neuronów w warstwie ukrytej i dobierz centra $\mathbf{c}$ funkcji radialnych (np. metodami grupowania danych)
2. Wyznacz szerokości funkcji radialnych σ każdego neuronu ukrytego (np. jako funkcję odległości między centrami)
3. Oblicz wagi powiązań między warstwą ukrytą i wyjściową

$$\mathbf{v} = (\mathbf{G}'^T \mathbf{G}')^{-1} \mathbf{G}'^T \mathbf{Y}$$

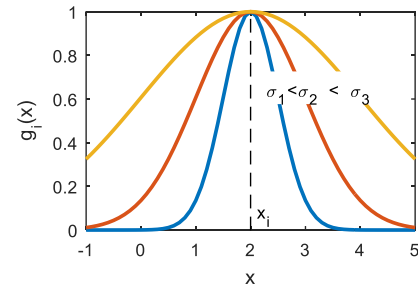
gdzie $\mathbf{G}'$ to macierz odpowiedzi warstwy ukrytej na wszystkie wektory uczące, rozszerzona o wiersz jedynek, $\mathbf{Y}$ – macierz pożądanych odpowiedzi dla wszystkich wektorów uczących.

SIEĆ REGRESJI UOGÓLNIONEJ GRNN



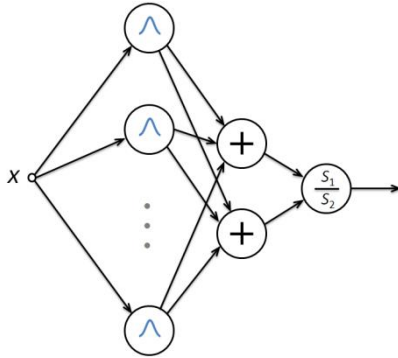
$$g_i(\mathbf{x}) = \exp\left(-\frac{d^2(\mathbf{x}, \mathbf{x}_i)}{2\sigma_i^2}\right), \quad i = 1, 2, \dots, M$$

$$h(\mathbf{x}) = \frac{\sum_{i=1}^M g_i(\mathbf{x}) y_i}{\sum_{i=1}^M g_i(\mathbf{x})}$$

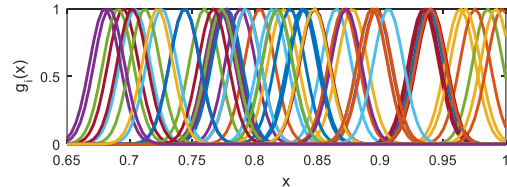
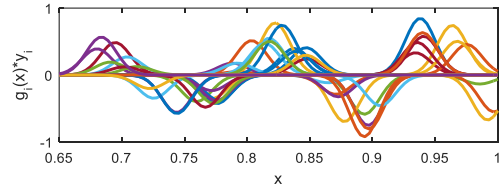
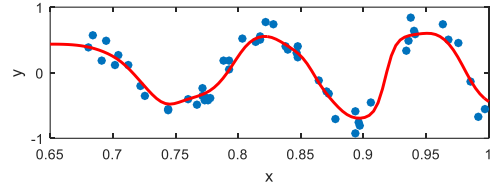


SIEĆ REGRESJI UOGÓLNIONEJ GRNN

Konstrukcja aproksymanty

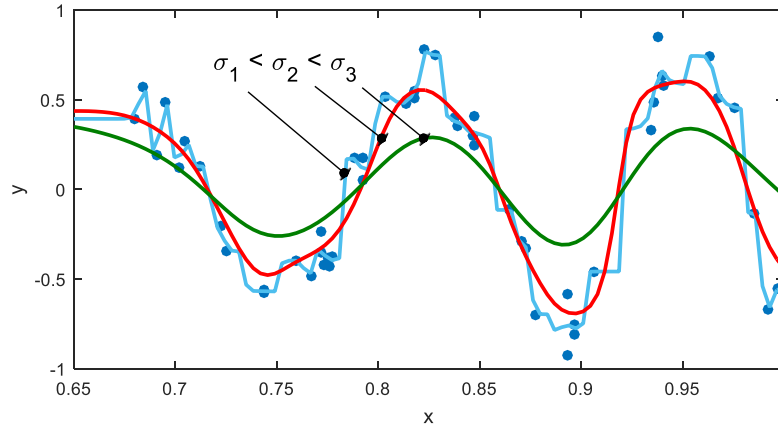


$$h(x) = \frac{\sum_{i=1}^M g_i(x) y_i}{\sum_{i=1}^M g_i(x)}$$



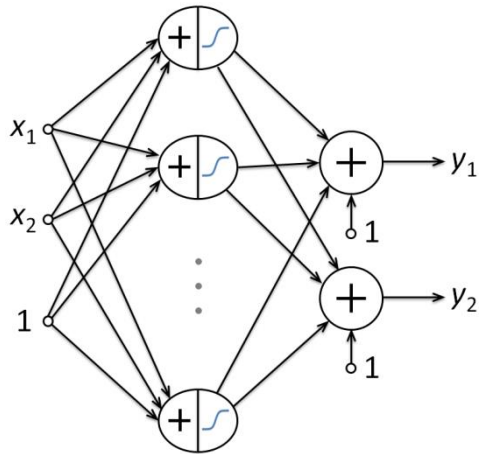
Uczenie sieci GRNN

Dobieramy szerokość funkcji radialnych σ



PRZYKŁADY ZASTOSOWAŃ SIECI NEURONOWYCH

Model silnika spalinowego



x_1 – zużycie paliwa

x_2 – prędkość obrotowa

y_1 – moment obrotowy

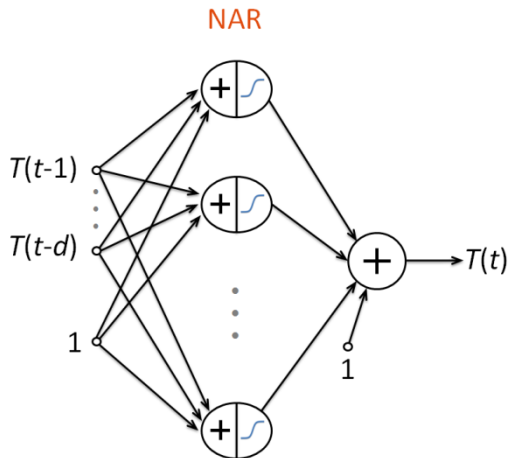
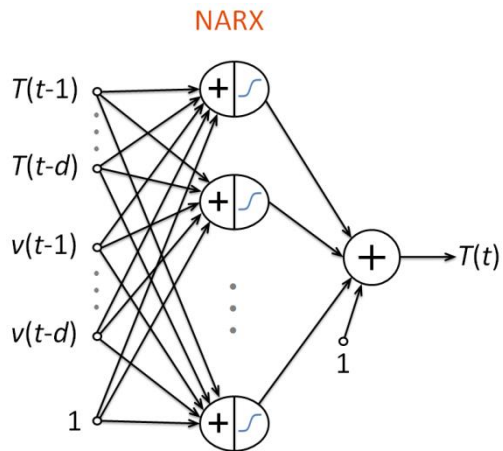
y_2 – poziom emisji tlenków azotu

PRZYKŁADY ZASTOSOWAŃ SIECI NEURONOWYCH

Model silnika spalinowego

PRZYKŁADY ZASTOSOWAŃ SIECI NEURONOWYCH

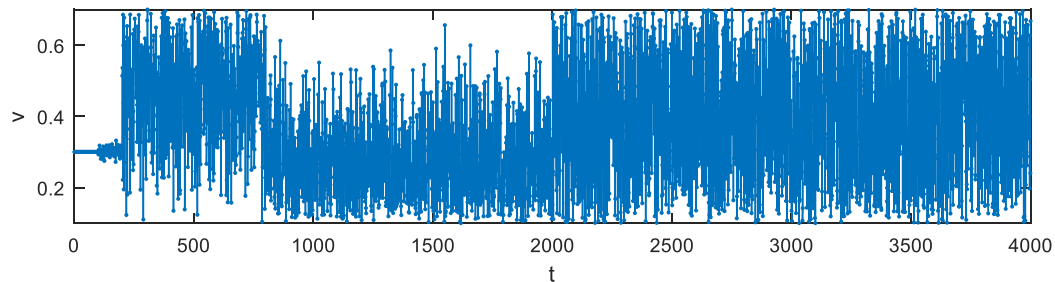
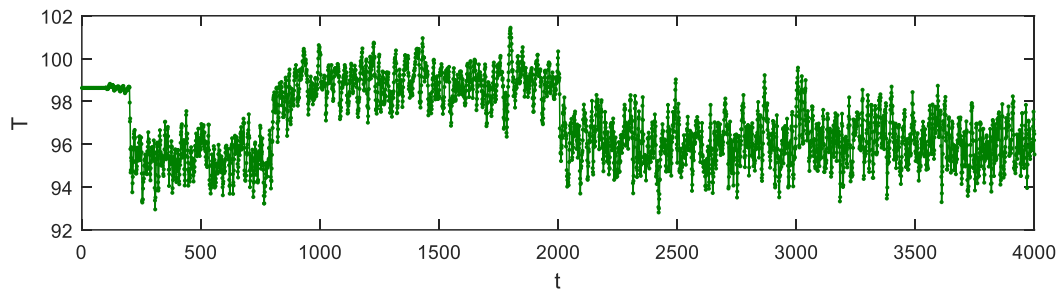
Modele dynamiczne wymiennika ciepłego



$T(t)$ – temperatura
wyściowa cieczy w
chwili t

$v(t)$ – prędkość
przepływu cieczy przez
wymiennik w chwili t

PRZYKŁADY ZASTOSOWAŃ SIECI NEURONOWYCH

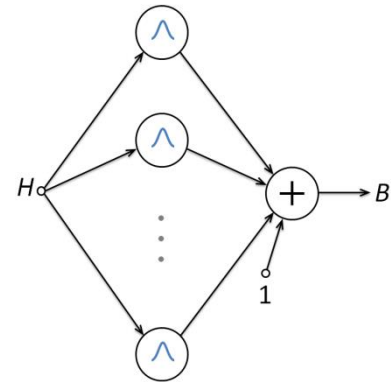
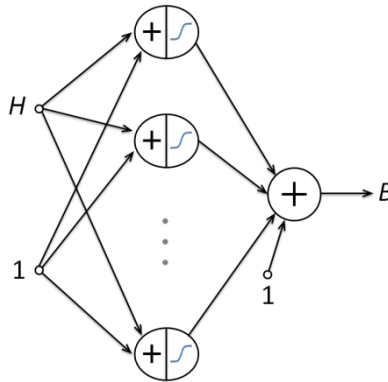
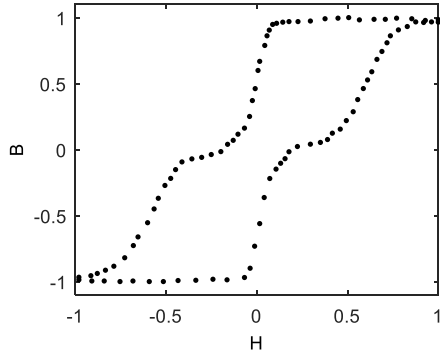


PRZYKŁADY ZASTOSOWAŃ SIECI NEURONOWYCH

Modele dynamiczne wymiennika ciepłego

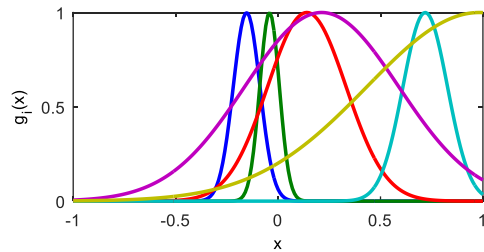
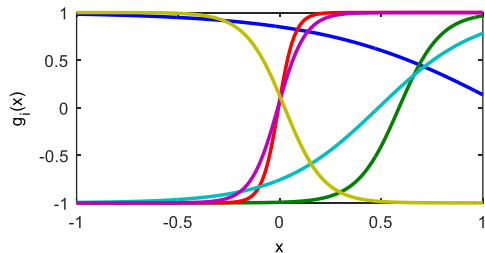
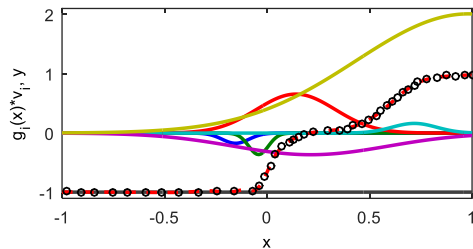
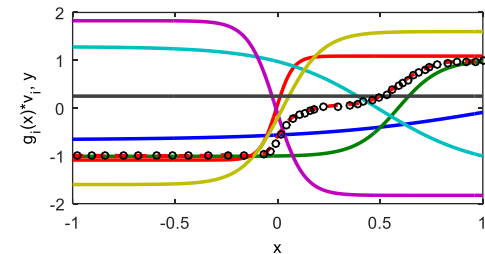
PRZYKŁADY ZASTOSOWAŃ SIECI NEURONOWYCH

Aproksymacja pętli histerezy magnetycznej



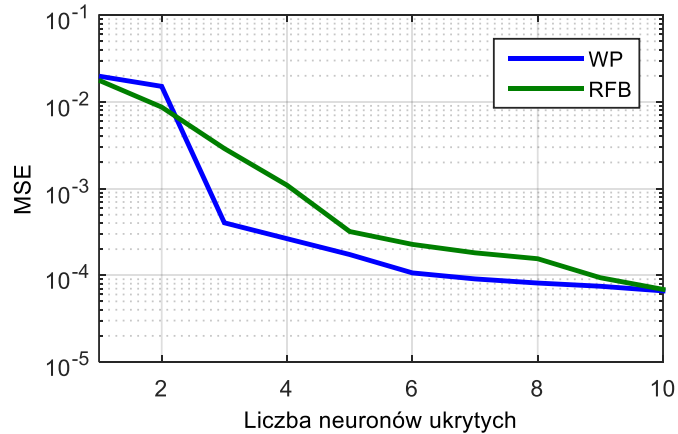
PRZYKŁADY ZASTOSOWAŃ SIECI NEURONOWYCH

Aproksymacja pętli histerezy magnetycznej



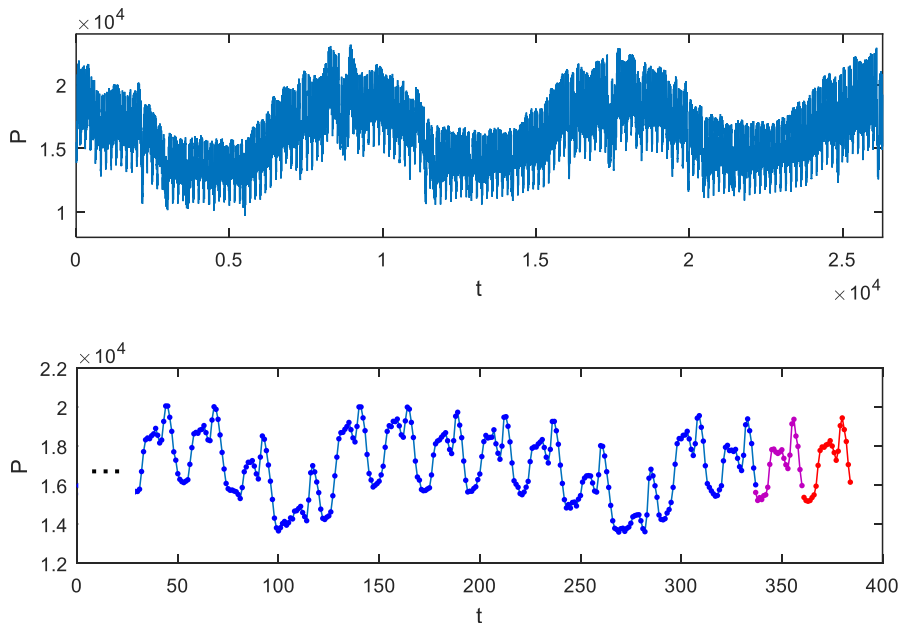
PRZYKŁADY ZASTOSOWAŃ SIECI NEURONOWYCH

Aproksymacja pętli histerezy magnetycznej



PRZYKŁADY ZASTOSOWAŃ SIECI NEURONOWYCH

Predykcja szeregów czasowych



PRZYKŁADY ZASTOSOWAŃ SIECI NEURONOWYCH

Predykcja szeregów czasowych

Wektor obciążeń doby i $\mathbf{P}_i = [P_{i,1} \ P_{i,2} \ \dots \ P_{i,24}]^T$ reprezentowany jest przez obraz $\mathbf{x}_i = [x_{i,1} \ x_{i,2} \ \dots \ x_{i,24}]^T$

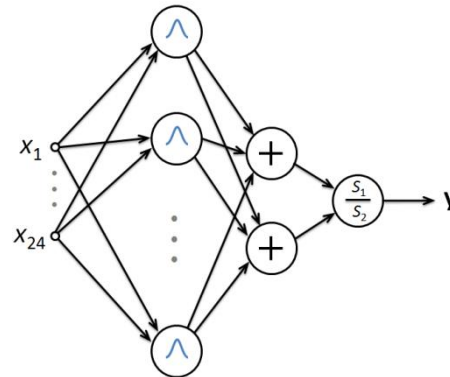
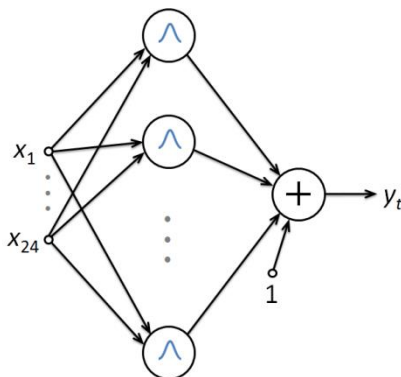
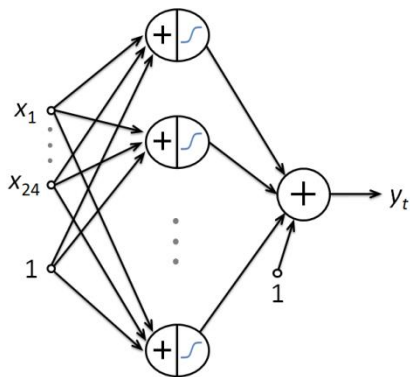
$$x_{i,t} = \frac{P_{i,t} - \bar{P}_i}{\sqrt{\sum_{l=1}^{24} (P_{i,l} - \bar{P}_i)^2}}$$

Wektor obciążeń doby prognozowanej $i+\tau$ $\mathbf{P}_{i+\tau} = [P_{i+\tau,1} \ P_{i+\tau,2} \ \dots \ P_{i+\tau,24}]^T$ reprezentowany jest przez obraz $\mathbf{y}_i = [y_{i,1} \ y_{i,2} \ \dots \ y_{i,24}]^T$

$$y_{i,t} = \frac{P_{i+\tau,t} - \bar{P}_i}{\sqrt{\sum_{l=1}^{24} (P_{i,l} - \bar{P}_i)^2}}$$

PRZYKŁADY ZASTOSOWAŃ SIECI NEURONOWYCH

Predykcja szeregów czasowych

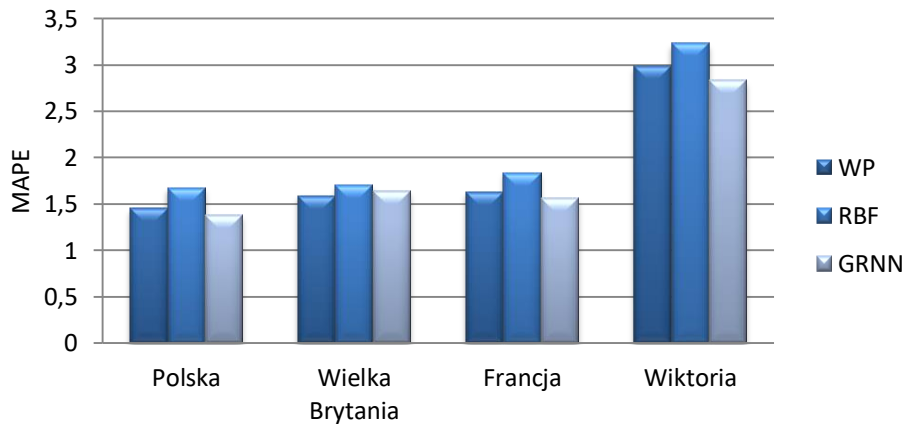


⇒ Dudek G.: **Neural Networks for Pattern-based Short-Term Load Forecasting: A Comparative Study**. Neurocomputing, 2016

⇒ Dudek G.: **Forecasting Time Series with Multiple Seasonal Cycles using Neural Networks with Local Learning**. Springer LNCS 7894, pp. 52-63, ICAISC 2013

PRZYKŁADY ZASTOSOWAŃ SIECI NEURONOWYCH

Predykcja szeregów czasowych



DZIĘKUJĘ ZA UWAGĘ

GRZEGORZ DUDEK

*Instytut Informatyki
Wydział Elektryczny
Politechnika Częstochowska
gdudek.el.pcz.pl*