

# SZTUCZNA INTELIGENCJA

## WYKŁAD 8. SZTUCZNE SIECI NEURONOWE – INNE ARCHITEKTURY

Dr hab. inż. Grzegorz Dudek  
Wydział Elektryczny  
Politechnika Częstochowska

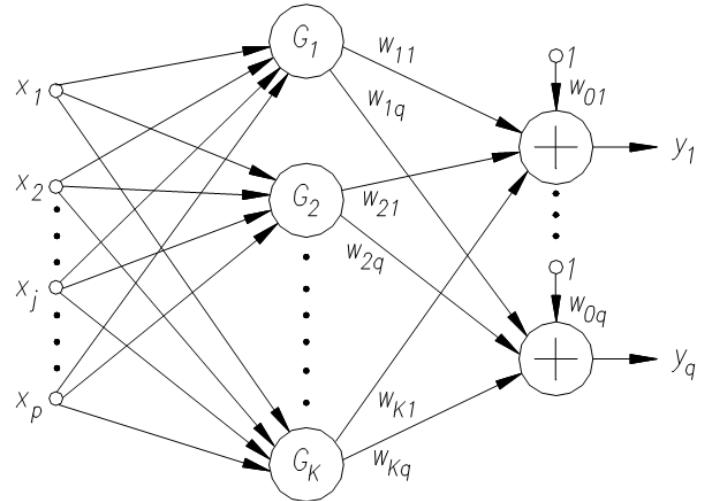
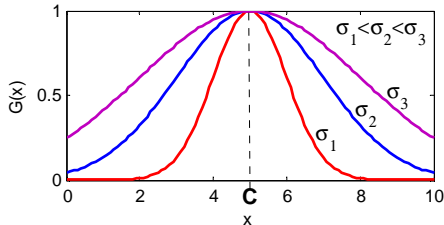
Częstochowa 2014

# SIEĆ O RADIALNYCH FUNKCJACH BAZOWYCH (RBF)

Funkcja aktywacji:

$$G_i(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{C}_i\|^2}{\sigma_i^2}\right)$$

gdzie:  $\|\cdot\|$  -- norma euklidesowa,  $\mathbf{C}$  – centrum,  $\sigma$  – parametr decydujący o szerokości funkcji radialnej.



Wyjście: 
$$F(\mathbf{x}) = \mathbf{w}_0 + \sum_{i=1}^K \mathbf{w}_i G_i(\mathbf{x})$$

## Sieć RBF uczy się trójetapowo:

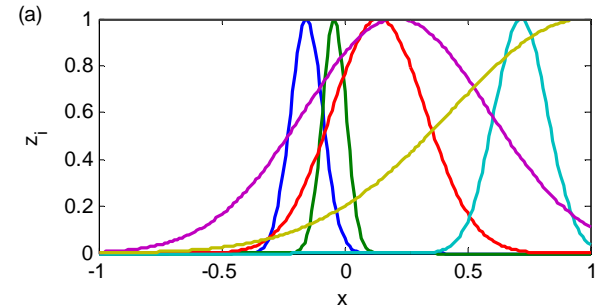
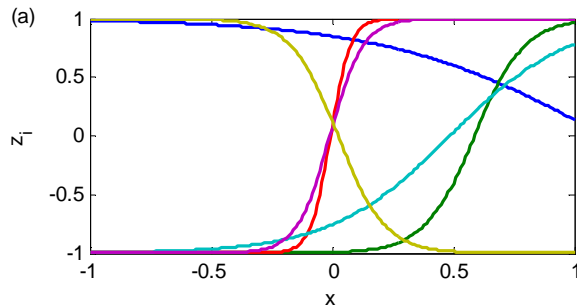
1. ustala się liczbę neuronów w warstwie ukrytej oraz dobiera się centra funkcji radialnych (np. metodami grupowania danych),
2. wyznacza się szerokości pól recepcyjnych ( $\sigma$ ) każdego neuronu ukrytego (np. jako funkcję odległości między centrami),
3. oblicza się wagi powiązań między warstwą ukrytą i wyjściową

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

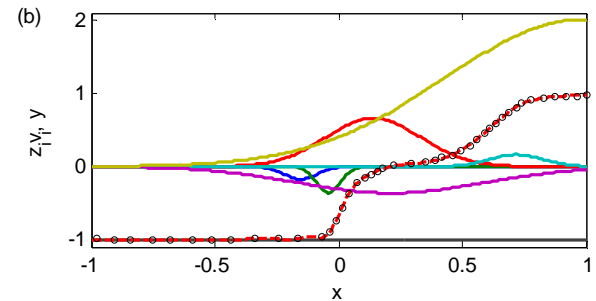
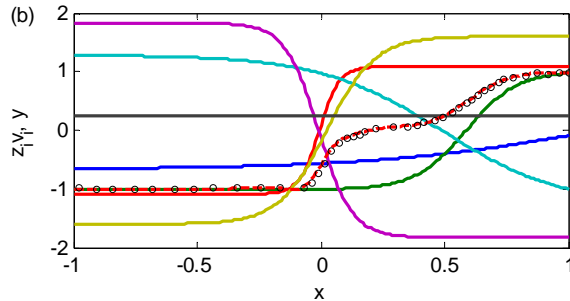
gdzie  $\mathbf{G}'$  to macierz odpowiedzi warstwy ukrytej na wszystkie wektory uczące, rozszerzona o wiersz jedynek,  $\mathbf{Y}$  – macierz pożądaných odpowiedzi dla wszystkich wektorów uczących.

Aproksymacja funkcji wielowarstwowym perceptronem (po lewej) i siecią RBF (po prawej)

Wyjścia neuronów warstwy ukrytej



Wejścia do neuronu warstwy wyjściowej i odpowiedź sieci

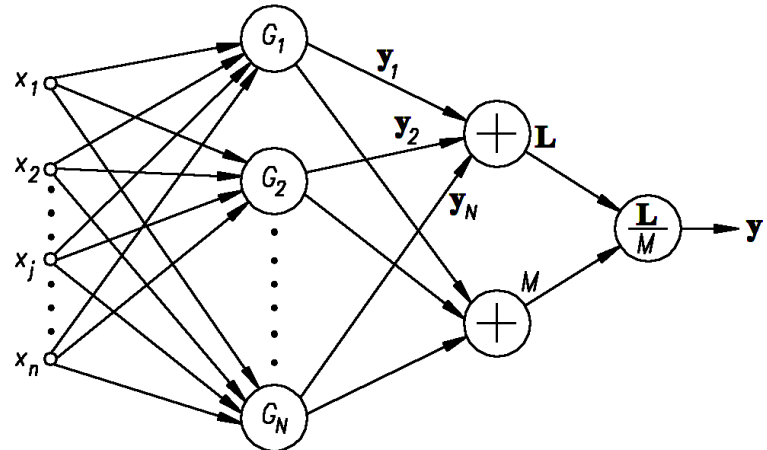


GRNN (*Generalized Regression Neural Network*) – sieć neuronowa realizująca regresję uogólnioną

Liczba neuronów radialnych = liczbie punktów uczących.

Odpowiedzi warstwy z neuronami radialnymi przemnażane są przez wartości pożądaných odpowiedzi ( $G^T Y$ ), a wynik  $L$  dzielony jest przez sumę odpowiedzi neuronów radialnych  $M$ .

W trakcie treningu ustala się jedynie szerokości funkcji radialnych  $\sigma$ .



Wyjście: 
$$F(\mathbf{x}) = \frac{\sum_{j=1}^N G_j y_j}{\sum_{j=1}^N G_j}$$

GMDH (*Group Method of Data Handling*) – sieć wielomianowa; adaptacyjna rozbudowa sieci.

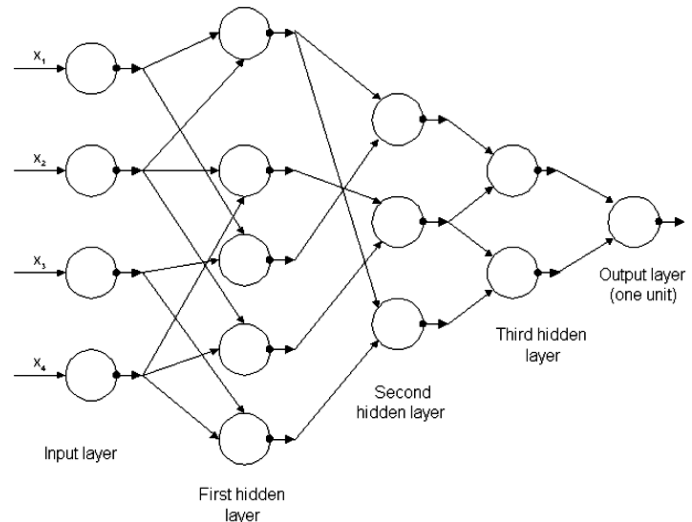
Sieć GMDH realizuje tzw. wielomian Iwachnienki.

Wielomian ten, zaimplementowany w strukturze sieci, ewoluuje od postaci elementarnej do optymalnej poprzez selekcję różnych kombinacji prostych modeli cząstkowych.

Modele cząstkowe (realizowane przez neurony) to funkcje dwóch zmiennych  $x_i$  i  $x_j$ :

$$f(x_i, x_j) = a_0 + a_1x_i + a_2x_j + a_{11}x_i^2 + a_{22}x_j^2 + a_{12}x_ix_j$$

gdzie:  $x_i$  i  $x_j$  – sygnały wejściowe neuronów,  
 $a$  – współczynniki.



## Sposób syntezy sieci GMDH:

- Tworzy się warstwę wejściową neuronów; każdy z nich realizuje funkcję  $f(x_i, x_j)$  dla innej kombinacji sygnałów wejściowych  $x_i, x_j$ .
- Dla każdego neuronu określa się współczynniki wielomianu  $f(x_i, x_j)$  minimalizujące błąd aproksymacji (np. metodą najmniejszych kwadratów).
- Selekcja neuronów – usuwane są te neurony, dla których obserwuje się duży błąd.
- Analogicznie buduje się kolejne warstwy, przy czym sygnałami wejściowymi do neuronów są sygnały wyjściowe z neuronów zoptymalizowanej warstwy poprzedniej.
- Proces syntezy sieci GMDH należy przerwać w momencie wzrostu wartości błędu w kolejnych warstwach lub gdy błąd ten osiąga zadany próg.
- Warstwa, w której została osiągnięta satysfakcjonująca wartość błędu pełni rolę warstwy wyjściowej z jednym neuronem „zwycięskim”. Neurony z warstwy wyjściowej oraz warstw wewnętrznych, które nie prowadzą do neuronu „zwycięskiego” są usuwane.

## SIEĆ Z ROZSZERZENIEM FUNKCYJNYM

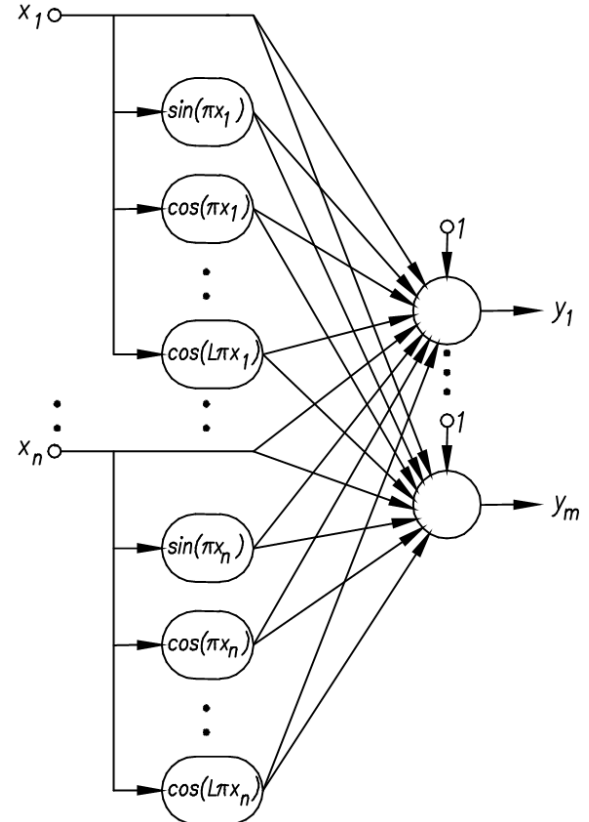
Wektor wejściowy jest poszerzony o funkcje składowych wektora oryginalnego.

Najczęściej są to funkcje ortogonalne:  $\sin \pi x$ ,  $\cos \pi x$ ,  $\sin 2\pi x$ ,  $\cos 2\pi x$  itd. lub iloczyny poszczególnych składowych wektora oryginalnego:  $x_i x_j$ ,  $x_i x_k$ ,  $x_j x_k$ ,  $x_i x_j x_k$  itd.

Rzutowanie danych wejściowych na przestrzeń o większej liczbie wymiarów pozwala uprościć problem.

Zysk:

- zamiast skomplikowanej wielowarstwowej sieci można zastosować jedną warstwę neuronów
- krótszy czas uczenia



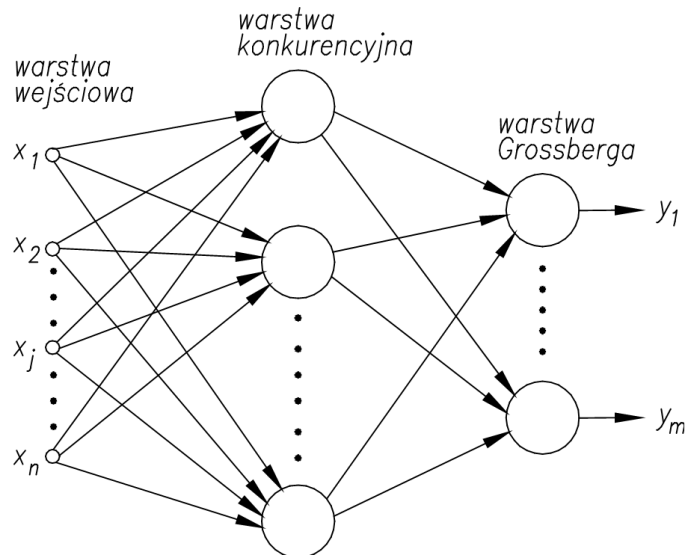


## SIEĆ TYPU COUNTERPROPAGATION

Sieć CP zawiera warstwę z konkurencyjną regułą uczenia bez nadzoru (warstwa Kohonena) oraz warstwę Grossberga uczoną z nauczycielem.

W warstwie Grossberga uczeniu podlegają tylko wagi powiązań „zwycięzcy” z neuronami wyjściowymi według reguły *outstar* Grossberga:

$$\Delta w_k = \eta (y_k - w_k)$$



Warstwa konkurencyjna dzieli dane wejściowe na skupiska reprezentowane przez wagi wejściowe neuronu konkurencyjnego. Wagi wyjściowe neuronu konkurencyjnego stanowią wyjścia sieci.

Sieć CP może działać w dwóch kierunkach:  $X \rightarrow Y$  lub  $Y \rightarrow X$ .

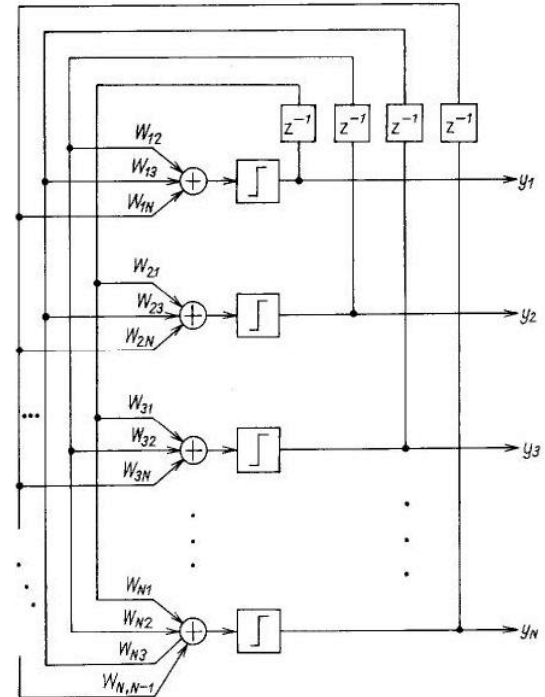
# SIEĆ REKURENCYJNA HOPFIELDA

Sieć Hopfielda – pamięć asocjacyjna: zadaniem sieci jest zapamiętanie zbioru wzorców uczących w taki sposób, aby w trakcie odtwarzania przy prezentacji nowego wzorca układ mógł wygenerować odpowiedź, która będzie odpowiadać jednemu z wzorców uczących, położonemu najbliższej próbki testowej.

Występują tu zależności dynamiczne, a zmiana stanu jednego neuronu przenosi się na całą sieć.

Zakłada się symetryczność połączeń  $w_{ij} = w_{ji}$  i brak połączenia wyjścia  $i$ -tego neuronu z jego wejściem.

Dobór wag:  $\mathbf{W} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$



## Tryb odtworzeniowy

Wyznacza się odpowiedzi sieci w kolejnych iteracjach  $k$

$$y_i(k+1) = \text{sgn}\left(\sum_{j=1}^N w_{ij}y_j(k)\right)$$

przy czym  $y_i(0) = x_i$ , gdzie  $x$  to wzorec testowy podany na wejście.

Proces kończy się gdy sieć osiadzie w atraktorze (stan stabilny):

$$y_i(k+1) = y_i(k) \quad \text{dla } i = 1, \dots, N$$

Sieć potrafi odtworzyć nauczone wzorce na podstawie uszkodzonego lub zaszumionego wzorca wejściowego:

