

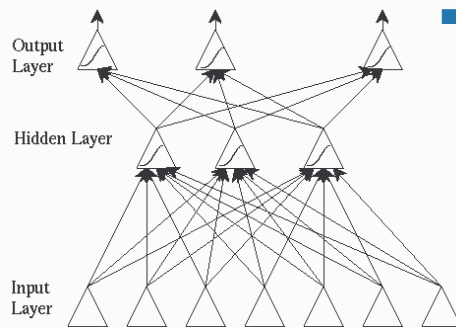
# WYKŁAD 3

Sieci neuronowe:  
Struktury &  
Zastosowania

## PLAN WYKŁADU

- Utrwalenie materiału:
  - Sieci wielowarstwowe
  - Funkcje aktywacji
  - Sieci jako funkcje złożone
- Zastosowania w sterowaniu
  - Kilka słów o zbiorach rozmytych
  - Model układu sterującego

# SIECI WIELOWARSTWOWE



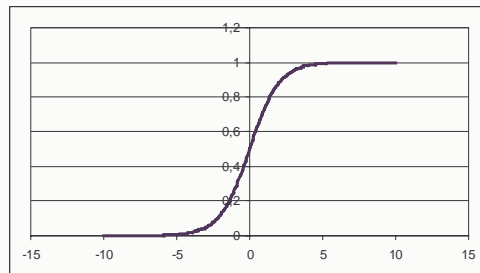
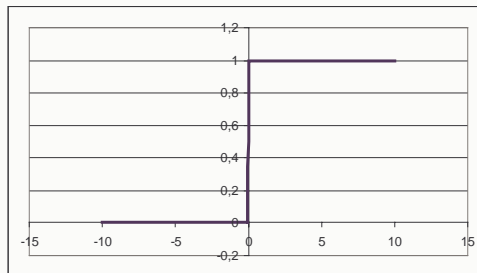
- Wyjścia neuronów należycych do warstwy niższej połączone są z wejściami neuronów należycych do warstwy wyższej
  - np. metodą „każdy z każdym”

- Działanie sieci polega na liczeniu odpowiedzi neuronów w kolejnych warstwach
- Nie jest znana ogólna metoda projektowania optymalnej architektury sieci neuronowej

# FUNKCJE AKTYWACJI (1)

- Progowe

$$f(z) = \begin{cases} 1 & \Leftrightarrow z \geq 0 \\ 0 & \Leftrightarrow z < 0 \end{cases}$$



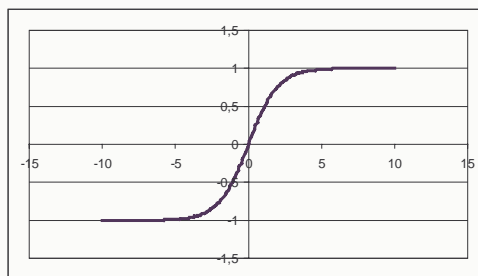
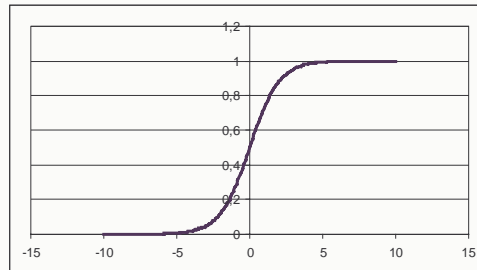
- Sigmoidalne

$$f(z) = \frac{1}{1 + e^{-z}}$$

## FUNKCJE AKTYWACJI (2)

### ■ Unipolarne

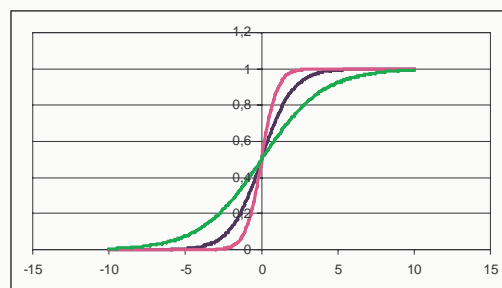
$$f(z) = \frac{1}{1 + e^{-z}}$$



### ■ Bipolarne

$$f(z) = \frac{2}{1 + e^{-z}} - 1$$

## FUNKCJE AKTYWACJI (3)



$$f_{\alpha}(z) = \frac{1}{1 + e^{-\alpha z}}$$

■  $\alpha = 2.0$

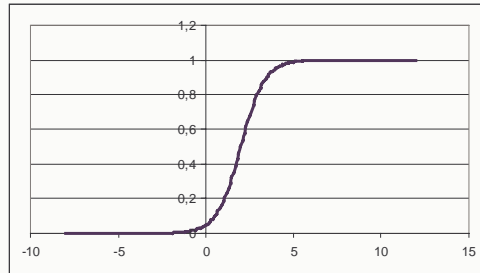
■  $\alpha = 1.0$

■  $\alpha = 0.5$

$$\lim_{\alpha \rightarrow 0} f_{\alpha}(z) = 0.5 \quad \lim_{\alpha \rightarrow +\infty} f_{\alpha}(z) = \begin{cases} 1 & \Leftrightarrow z > 0 \\ 0.5 & \Leftrightarrow z = 0 \\ 0 & \Leftrightarrow z < 0 \end{cases}$$

## FUNKCJE AKTYWACJI (4)

$$f_{\theta,\alpha}(z) = \frac{1}{1 + e^{-\alpha(z-\theta)}}$$



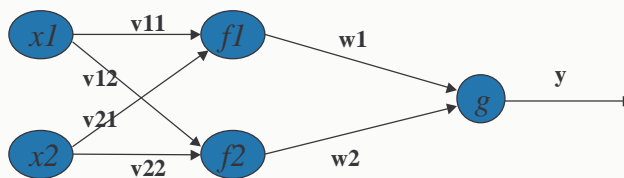
$$\theta = 2$$

$$\alpha = 1.5$$

## FUNKCJE AKTYWACJI (5)

- Zasady ogólne:
  - Ciągłość (zachowanie stabilności sieci jako modelu rzeczywistego)
  - Różniczkowalność (zastosowanie propagacji wstecznej)
  - Monotoniczność (intuicje związane z aktywacją komórek neuronowych)
  - Nieliniowość (możliwość ekspresji)

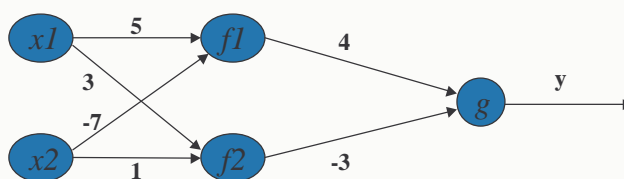
## SIECI JAKO FUNKCJE ZŁOŻONE (1)



$$y = g(w_1 f_1(v_{11}x_1 + v_{21}x_2) + w_2 f_2(v_{12}x_1 + v_{22}x_2))$$

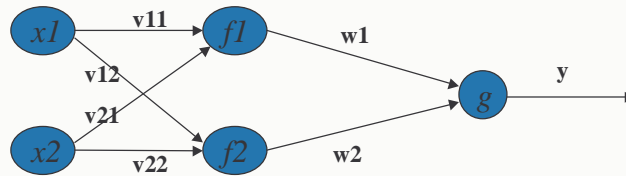
$$y = \text{Network}(x_1, x_2)$$

## SIECI JAKO FUNKCJE ZŁOŻONE (2)



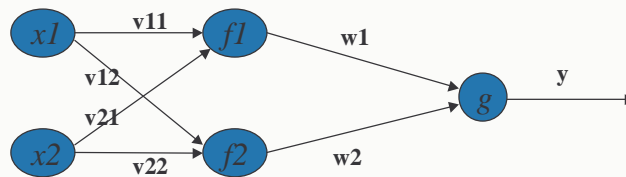
$$y = \begin{cases} 1 \Leftrightarrow \frac{4}{1+e^{-3(5x_1-7x_2)}} - 3 \left( \frac{2}{1+e^{-2(3x_1+x_2)}} - 1 \right) \geq 8 \\ 0 \Leftrightarrow \frac{4}{1+e^{-3(5x_1-7x_2)}} - 3 \left( \frac{2}{1+e^{-2(3x_1+x_2)}} - 1 \right) < 8 \end{cases}$$

## SIECI JAKO FUNKCJE ZŁOŻONE (3)



- Jeśli wszystkie poszczególne funkcje aktywacji są liniowe, to funkcja *Network* jest również liniowa
- Architektura wielowarstwowa daje zatem nowe możliwości tylko w przypadku stosowania funkcji nieliniowych

## SIECI JAKO FUNKCJE ZŁOŻONE (4)



- Niech
 
$$f_i(x_1, x_2) = a_i \cdot (x_1 \cdot v_{1i} + x_2 \cdot v_{2i}) + b_i$$

$$g(z_1, z_2) = a \cdot (z_1 \cdot w_1 + z_2 \cdot w_2) + b$$
- Wtedy
 
$$\text{Network}(x_1, x_2) = A_1 \cdot x_1 + A_2 \cdot x_2 + B$$
- Np.:
 
$$A_1 = a \cdot (a_1 \cdot v_{11} \cdot w_1 + a_2 \cdot v_{21} \cdot w_2)$$

## ZBIORY ROZMYTE (1)

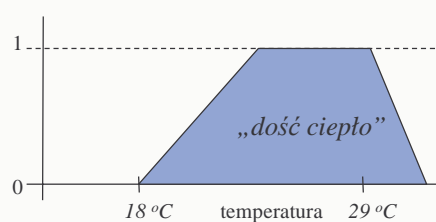
Metoda reprezentacji wiedzy wyrażonej w języku naturalnym:

*Temperatura wynosi 29 °C* → *Jest dość ciepło*

informacja liczbowo – naturalna dla systemów komputerowych

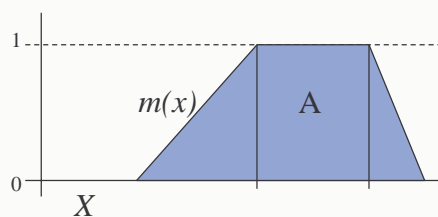
informacja opisowa – naturalna dla człowieka

Zamiast dwóch wartości logicznych (prawda i fałsz), dopuszcza się istnienie nieskończenie wielu wartości (odpowiadających liczbom rzeczywistym od 0 do 1)



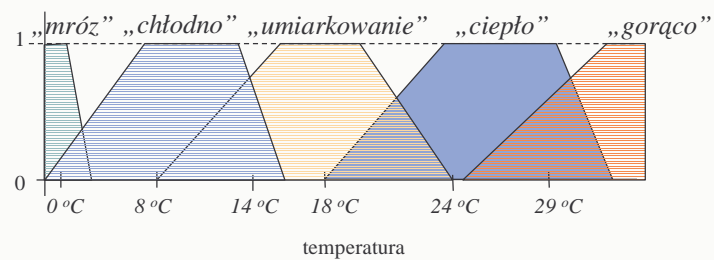
## ZBIORY ROZMYTE (2)

$m : X \rightarrow [0,1]$   
funkcja przynależności  
(funkcja charakterystyczna)  
zbioru rozmytego A



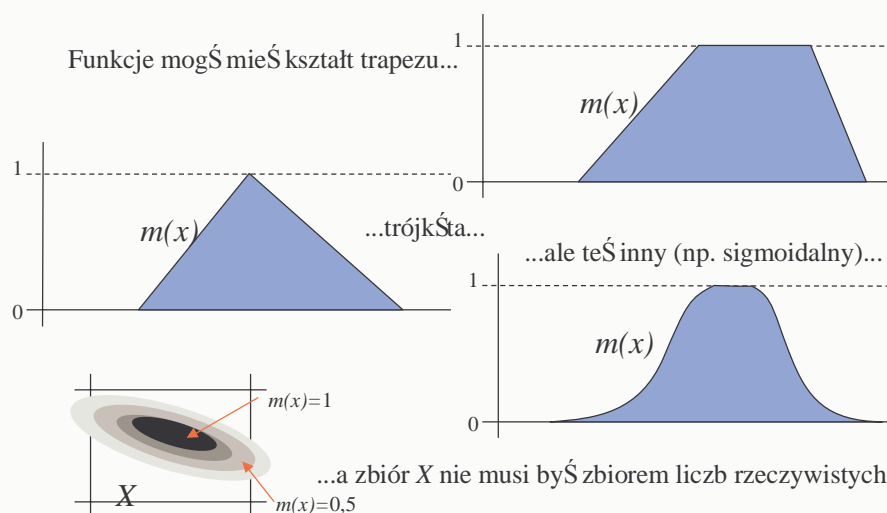
*Funkcja przynależności mówi nam, w jakim stopniu bylibyśmy skłonni uznać daną wartość za należącą do zbioru, np. w jakim stopniu powietrze o temperaturze 20 °C może być uznane za „dość ciepłe”*

## ZBIORY ROZMYTE (3)



Pojęcia „ciepło” czy „gorąco” s<sup>ę</sup> okre<sup>ś</sup>lone w sposób nieostry: trudno jednoznacznie okre<sup>ś</sup>li<sup>ć</sup> ich granice, ich zakresy mog<sup>ę</sup> si<sup>ę</sup> cz<sup>ę</sup>ściowo pokrywa<sup>ć</sup>

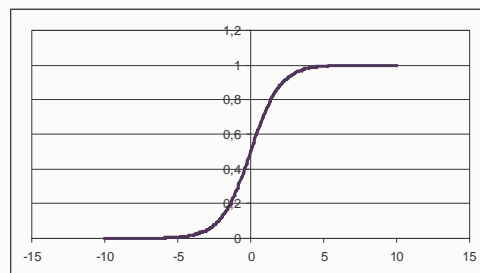
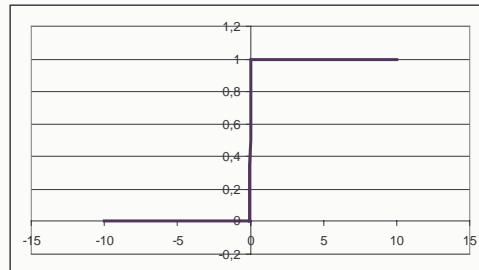
## FUNKCJE PRZYNALEŚNOŚCI





## PRZYNALEŻNOŚĆ A AKTYWACJA

Funkcja charakterystyczna odpowiadająca pojęciu liczby dodatniej



Funkcja przynależności dla zbioru rozmytego odpowiadającego pojęciu liczby dodatniej

## REGUŁY ROZMYTE (1)

- Reguły, których przesłanki lub wnioski wyrażone są w języku zbiorów rozmytych
  - Jeżeli x jest **małe** i y jest **średnie**, to uruchom alarm
  - Jeżeli x jest **małe** i y jest **małe**, to ustaw z na **duże**
  - Jeżeli x jest **duże**, to ustaw z na **małe**
- Reguły pochodzące od ekspertów zwykle wyrażone są w języku nieprecyzyjnym
- Zbiory rozmyte pozwalają nam przełożyć ten język na konkretne wartości liczbowe

## REGUŁY ROZMYTE (2)

	Sun (%)	Temp. (°C)	Humid. (%)	Wind (km/h)	Run (km/h)
1	100	31	90	10	6
2	90	22	85	50	8
3	50	25	95	20	12
4	0	15	80	0	13
5	10	4	70	10	15
6	30	7	55	40	7
7	40	8	65	60	15
8	70	14	90	20	10
9	80	1	70	30	14
10	20	13	60	0	14
11	80	11	60	70	14
12	60	17	80	50	13
13	50	26	55	30	16
14	20	12	95	60	9

Zbiory rozmyte pozwalają konstruować reguły typu

jeśli temperatura jest wysoka i wilgotność jest niska, to sąsiad biega

w języku naturalnym, przekładalne jednak na zależności numeryczne

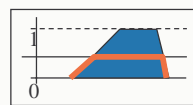
## REGUŁY ROZMYTE (3)

- Metoda powiązania cech modeli symbolicznych, takich jak np. drzewa decyzyjne, oraz modeli numerycznych, takich jak np. sieci neuronowe
- W zastosowaniach, wymagany jest proces uczenia kształtów zbiorów rozmytych dla poszczególnych zmiennych występujących w regułach

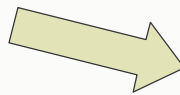
## STEROWANIE (1)

sieć neuronowa zamiast reguł rozmytych

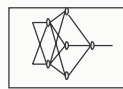
*Reguły sterowania wyrażone w języku zbliżonym do naturalnego (tzn. w języku zbiorów przybliżonych) można „przetłumaczyć” na strukturę sieci neuronowej. Uczenie wag takiej sieci odpowiada uczeniu parametrów (kształtów) zbiorów rozmytych.*



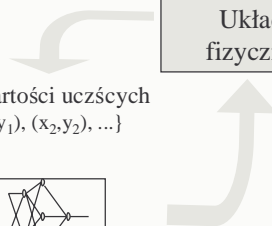
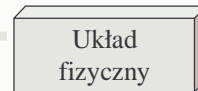
Reguły rozmyte



Ciąg wartości uczących  
 $\{(x_1, y_1), (x_2, y_2), \dots\}$



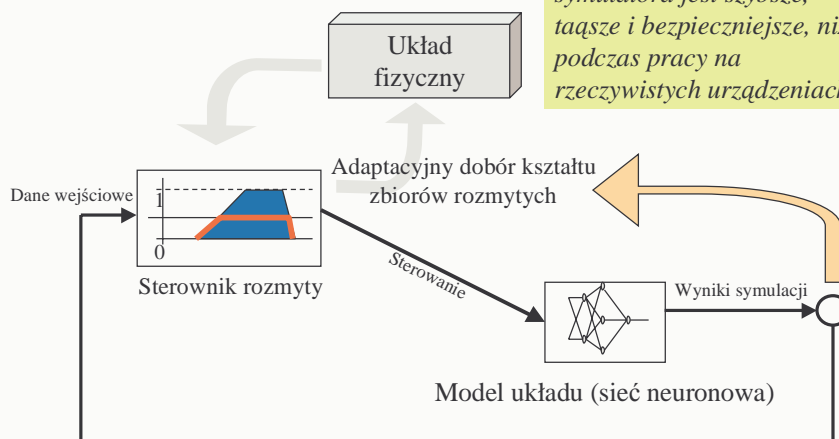
Model zbioru reguł (sieć neuronowa),  
uczona na danych treningowych



## STEROWANIE (2)

sieć neuronowa jako symulator  
procesu fizycznego

*Dobieranie parametrów sterownika rozmytego za pomocą neuronowego symulatora jest szybsze, tańsze i bezpieczniejsze, niż podczas pracy na rzeczywistych urządzeniach*



## FUZZY-NEURO RICE COOKER



- Fuzzy Logic Controls The Cooking Process
- Self Adjusting For Rice & Water Conditions
- Cooks Brown Rice In Addition To White, Sweet (Glutinous, Mochigome) Mixed Variety, Porridge
- Porridge Setting Can Also Be Used As A Slow Cooker
- Automatically Cooks and Switches to Keep Warm
- Will Finish Cooking When You Want It To With Its 24 Hour Preset Timer
- Fast Cook of White Rice 13 Minutes Faster Than Regular Cycle Additional Reheat Function For Piping Hot Rice
- Sale Price: **\$119.00**