



WAI



Wykłady 3 i 4. Sieci neuronowe. Uczenie i zastosowania. Wstęp do logiki rozmytej.

Literatura:

S. Osowski, *Sieci neuronowe w ujęciu algorytmicznym*. WNT, Warszawa 1997.

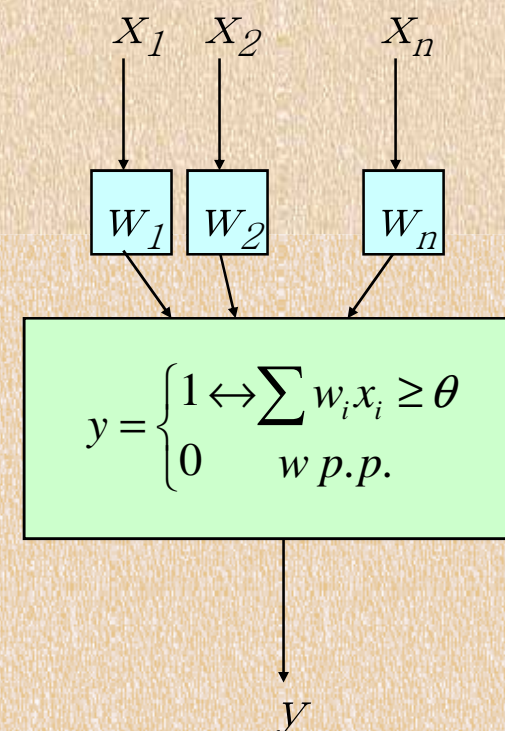
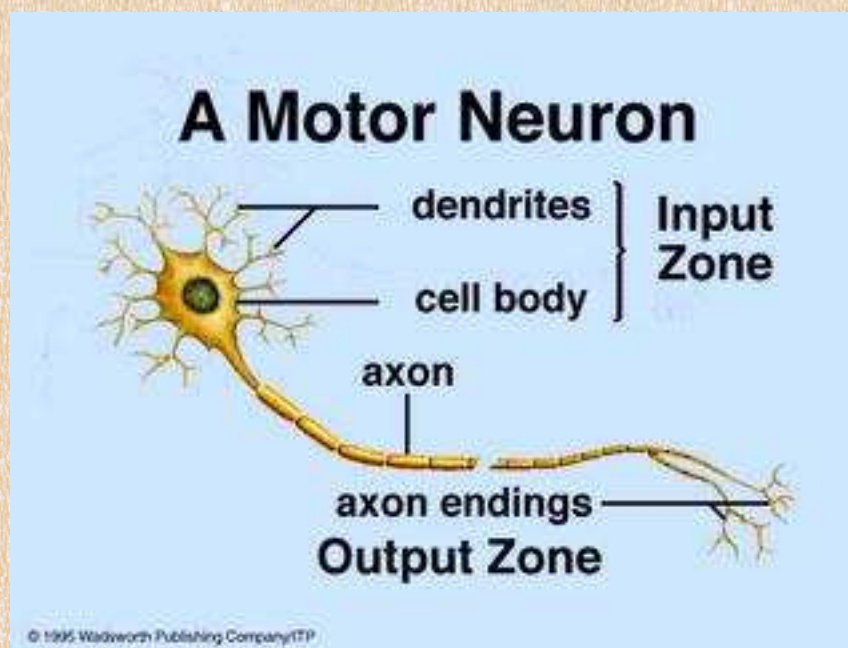
D. Rutkowska, M. Piliński i L. Rutkowski, *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*, PWN, Warszawa 1997

R. Tadeusiewicz, *Sieci neuronowe*. Akademicka Oficyna Wydawnicza RM, Warszawa, 1993, 1999

Żurada Jacek, Barski Mariusz, Jędruch Wojciech, *Sztuczne sieci neuronowe*, Wydawnictwo

Naukowe PWN, Warszawa, 1996.

Perceptron - przypomnienie





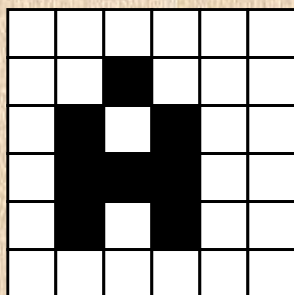
Przypomnienie. Jak opisać perceptron? Co charakteryzuje perceptron?

- Perceptron jest opisywany jednoznacznie przez zbiór wag $w_1, \dots, w_n \in \mathbb{R}$ oraz wartość progowa $\theta \in \mathbb{R}$
- Wartości $x_1, \dots, x_n \in \mathbb{R}$ to zmienne pojawiające się na wejściu do perceptronu
- Funkcja aktywacji:
$$y = \begin{cases} 1 & \Leftrightarrow \sum w_i x_i \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

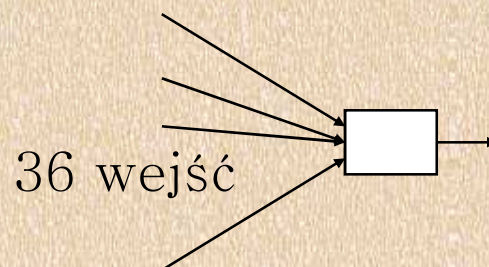
Uczenie perceptronu



Przykład: rozpoznawanie znaków

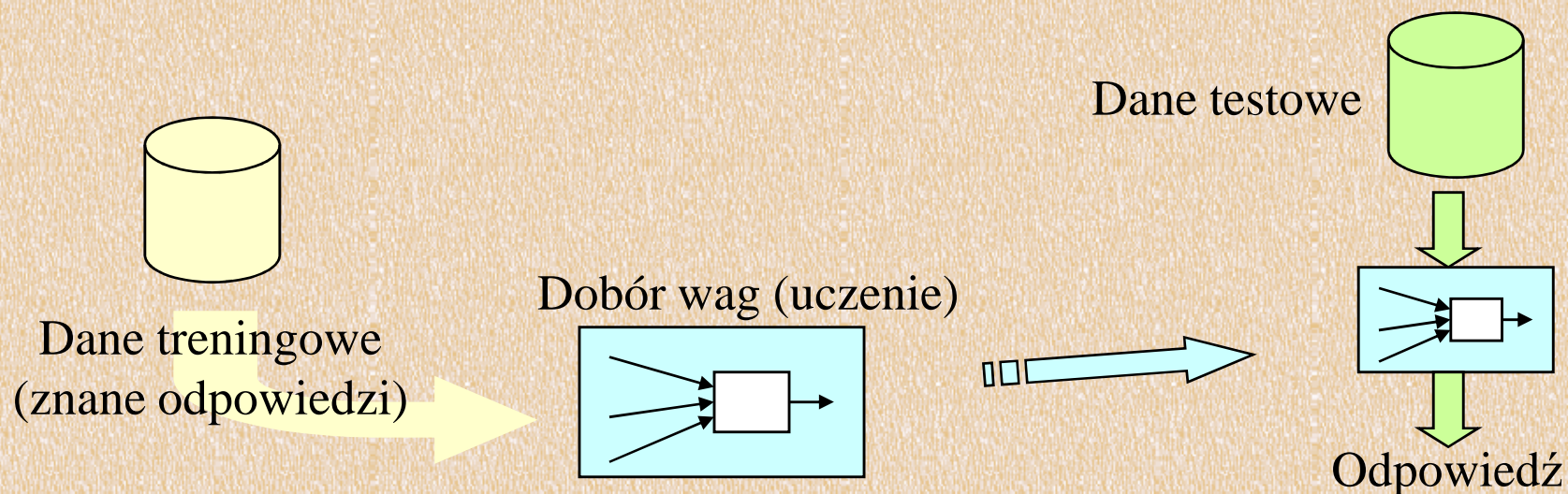


Siatka 6×6



Wyjście: 1, jeśli na wejściu pojawia się litera "A", zaś 0 w p.p.

Zadanie: dobrać wagi wejść i wartość progową tak, by uzyskać zaplanowany efekt





Uczenie perceptronu, $n=2$



- **Wejście:**
 - Ciąg przykładów uczących ze znanymi odpowiedziami
- **Proces uczenia:**
 - Inicjujemy wagi losowo
 - Dla każdego przykładu, jeśli odpowiedź jest nieprawidłowa, to

$$w_1 + = \alpha x_1$$

$$w_2 + = \alpha x_2$$

$$\theta - = \alpha$$

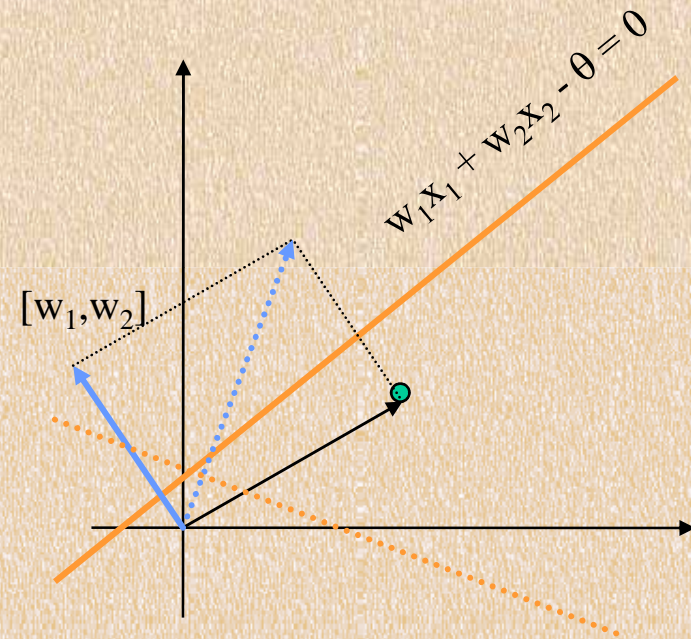
$$w_1(k+1) = w_1(k) + w_1 + ,$$

podobnie dla w_2 ,

$$\theta(k+1) = \theta(k) - \theta - ,$$

k -krok iteracji, epoka

gdzie α jest równe różnicy odpowiedzi sieci i prawidłowej odpowiedzi.

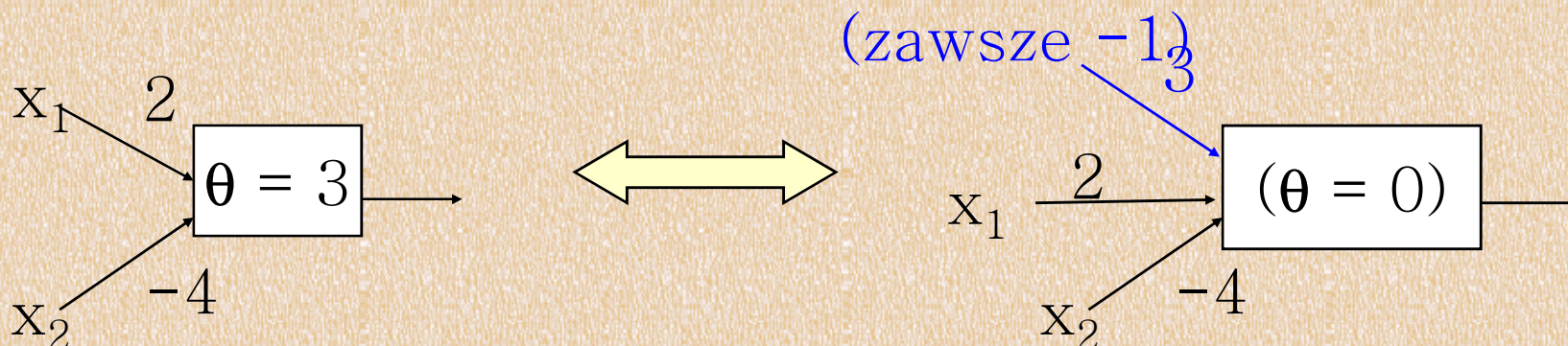




Uczenie perceptronu



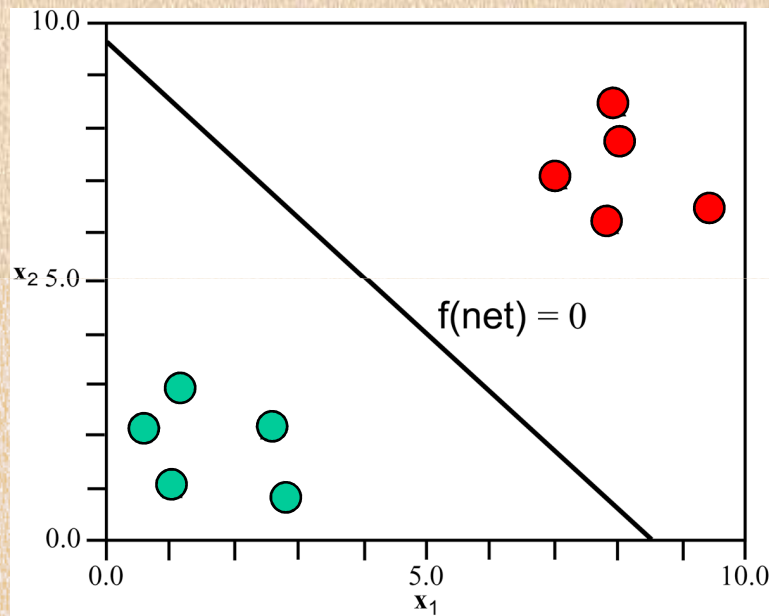
- Często α mnoży się dodatkowo przez niewielki współczynnik uczenia
- Po wyczerpaniu przykładów, zaczynamy proces uczenia od początku, dopóki następują jakiegokolwiek zmiany wag połączeń
- Próg θ można traktować jako wagę dodatkowego wejścia o wartości -1:



Przykład: Uczenie neuronu



- Zbiór punktów na wykresie jest liniowo separowalny.



| x_1 | x_2 | Output |
|-------|-------|--------|
| 1.0 | 1.0 | 1 |
| 9.4 | 6.4 | -1 |
| 2.5 | 2.1 | 1 |
| 8.0 | 7.7 | -1 |
| 0.5 | 2.2 | 1 |
| 7.9 | 8.4 | -1 |
| 7.0 | 7.0 | -1 |
| 2.8 | 0.8 | 1 |
| 1.2 | 3.0 | 1 |
| 7.8 | 6.1 | -1 |

Funkcja aktywacji:

$$y = \begin{cases} 1 & \Leftrightarrow \sum w_i x_i \geq \theta \\ -1 & \text{otherwise} \end{cases}$$

- Niech $w_1=1$, $w_2=1$, $\theta = 1$, wsp. uczenia $\eta=1$



• Pierwszy przykład jest dobrze, ale drugi nie, modyfikujemy zatem wagi:

$$w_1 + = (-1 - 1) 9.4$$

$$w_2 + = (-1 - 1) 6.4$$

$$\theta - = (-1 - 1)$$

- Otrzymamy

$$w_1 = - 18.8$$

$$w_2 = - 12.2$$

$$\theta = 3$$

• Drugi przykład jest dobry, ale trzeci nie...

| x_1 | x_2 | Output |
|-------|-------|--------|
| 1.0 | 1.0 | 1 |
| 9.4 | 6.4 | -1 |
| 2.5 | 2.1 | 1 |
| 8.0 | 7.7 | -1 |
| 0.5 | 2.2 | 1 |
| 7.9 | 8.4 | -1 |
| 7.0 | 7.0 | -1 |
| 2.8 | 0.8 | 1 |
| 1.2 | 3.0 | 1 |
| 7.8 | 6.1 | -1 |



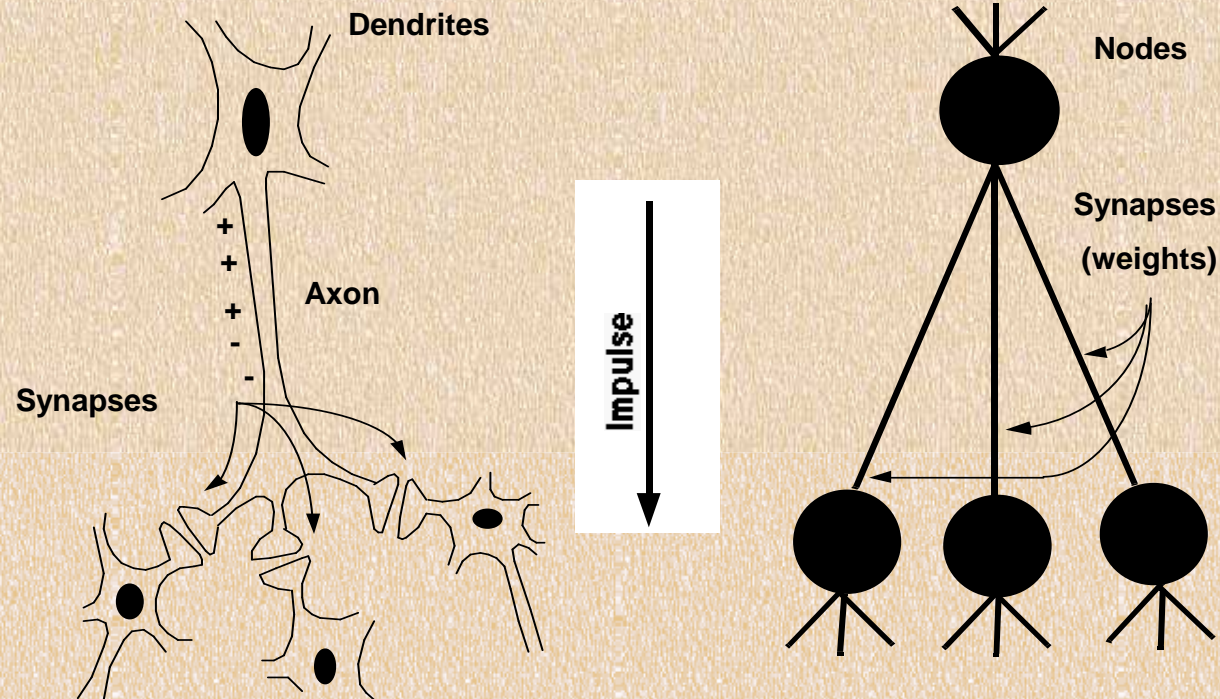
Uczenie perceptronu



- Opisany schemat jest w miarę przejrzysty tylko dla pojedynczych perceptronów, lub niewielkich sieci
- Ciężko jest stosować reguły tego typu dla skomplikowanych modeli
 - Tymczasem np. do rozpoznawania wszystkich liter potrzeba by sieci złożonej z 26 takich perceptronów



Sieci perceptronów



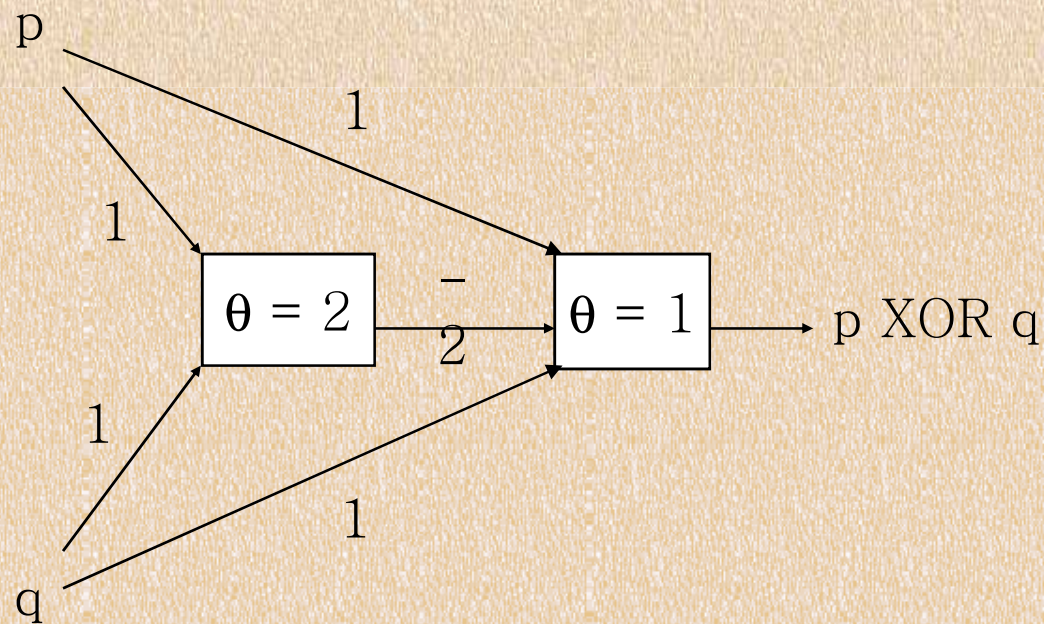
Ograniczenia pojedynczych perceptronów spowodowały w latach 80-tych wzrost zainteresowania sieciami wielowarstwowymi i opracowanie algorytmu ich uczenia (propagacja wsteczna)



SIECI PERCEPTRONÓW

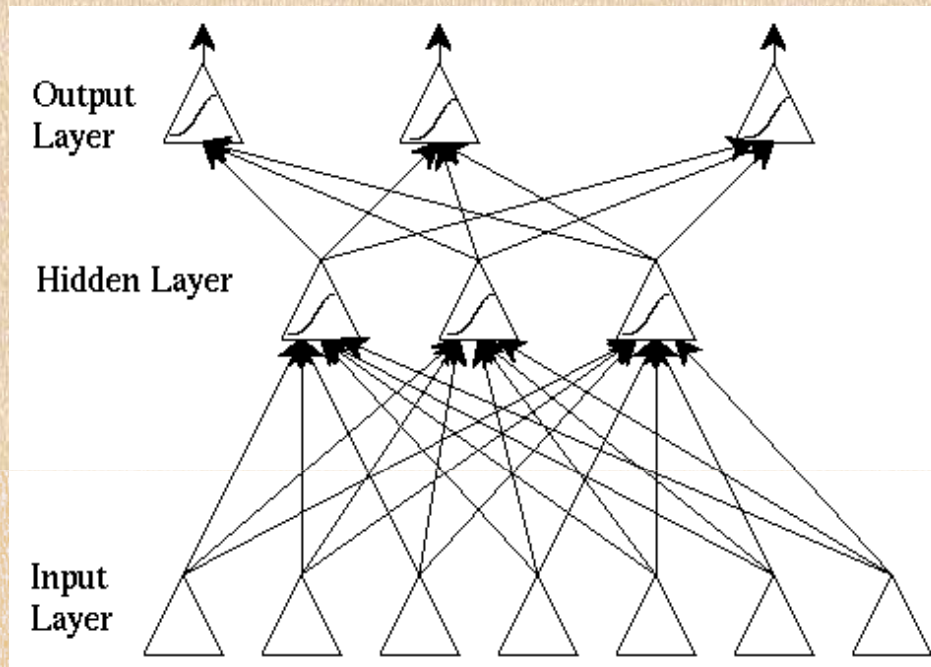


Potrafia reprezentować dowolną funkcję boolowską (opartą na rachunku zdań)





SIECI WIELOWARSTWOWE



- Wyjścia neuronów należących do warstwy niższej połączone są z wejściami neuronów należących 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

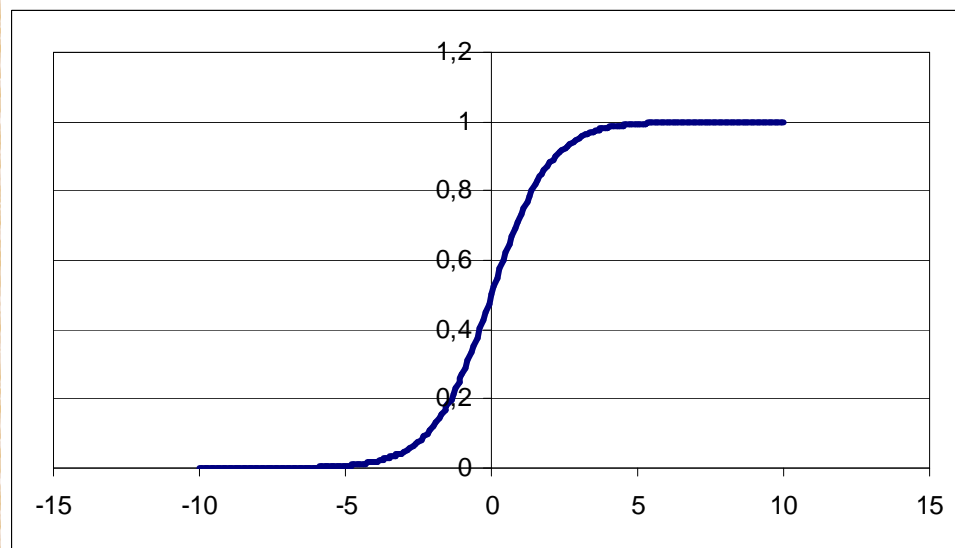
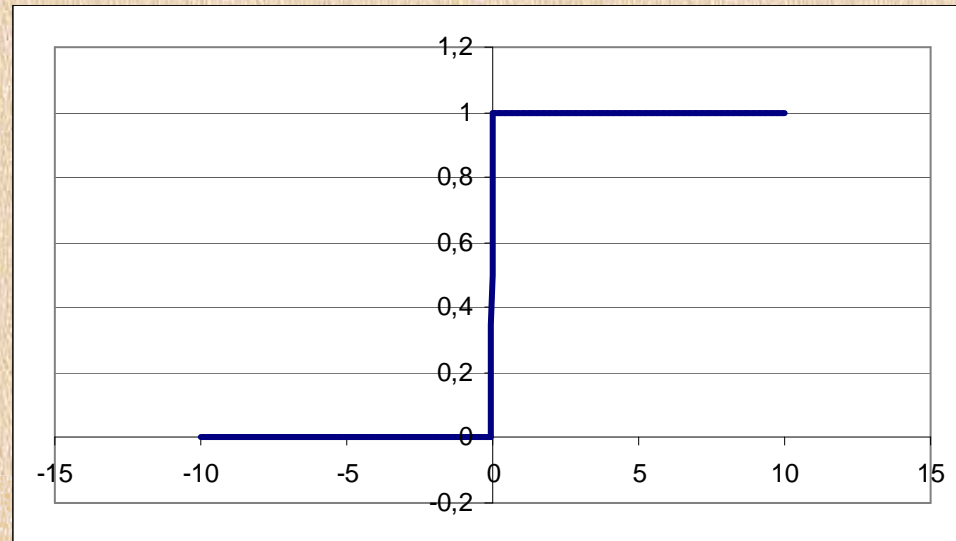


Funkcje aktywacji



- Progowe

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



- Sigmoidalne

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

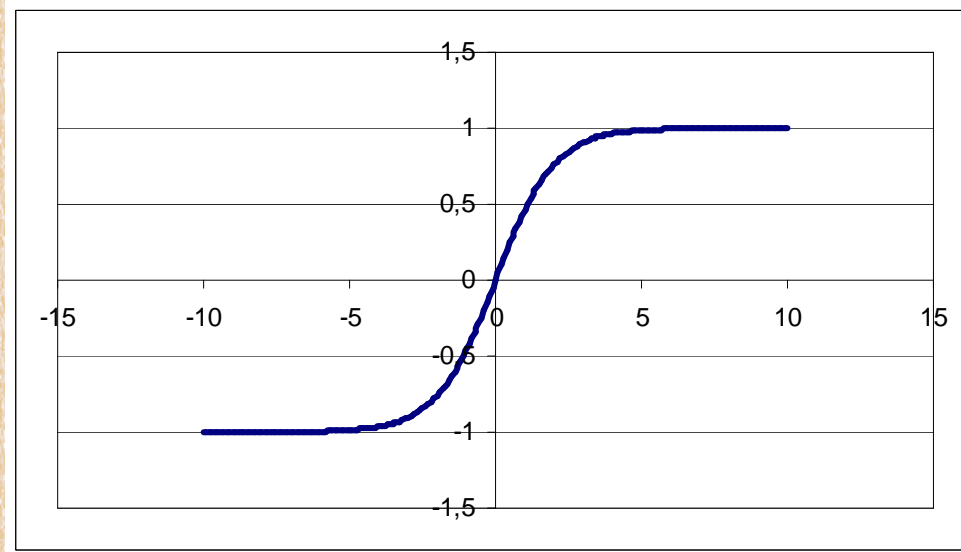
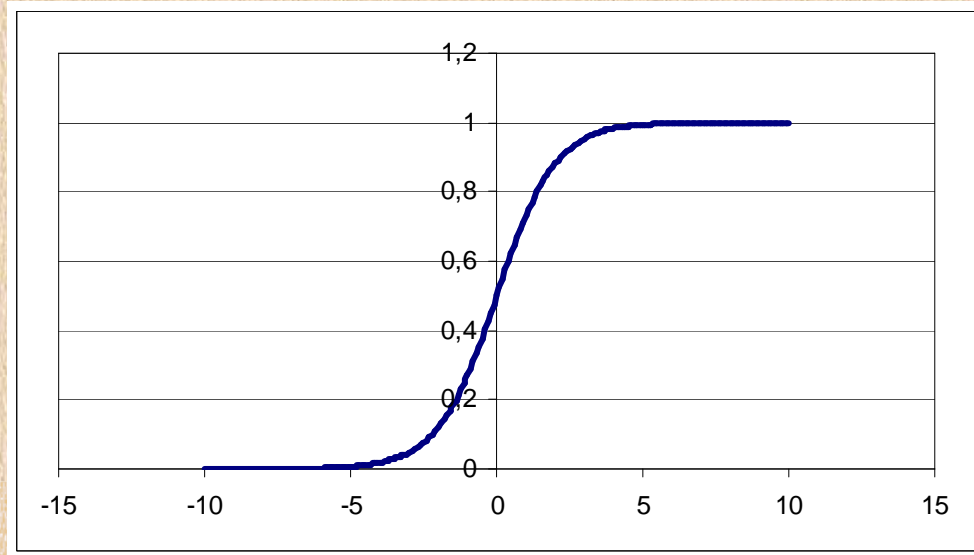


FUNKCJE AKTYWACJI (2)



- Unipolarne

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

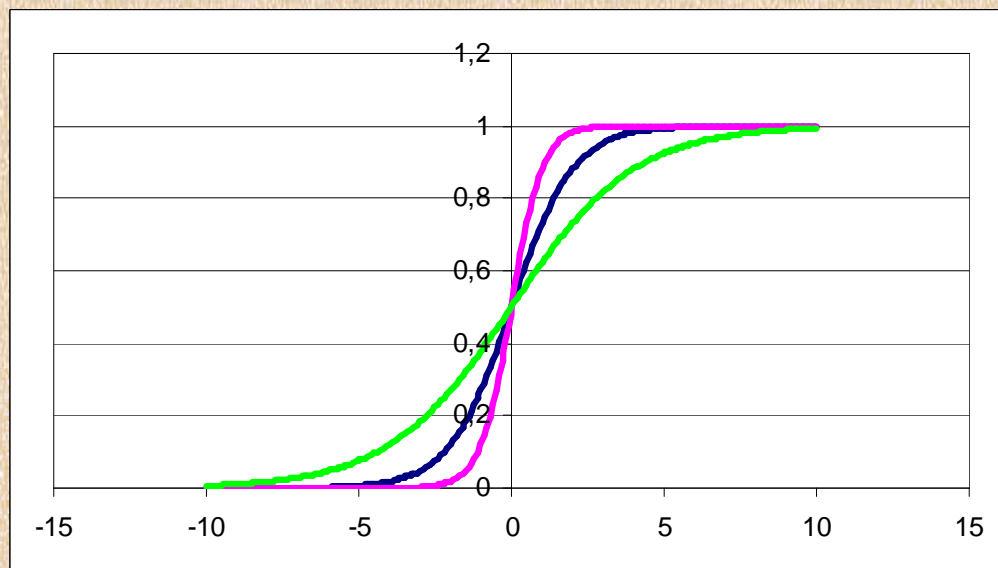


- Bipolarne

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



FUNKCJE AKTYWACJI (3)



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

 $\alpha = 2.0$

 $\alpha = 1.0$

 $\alpha = 0.5$

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

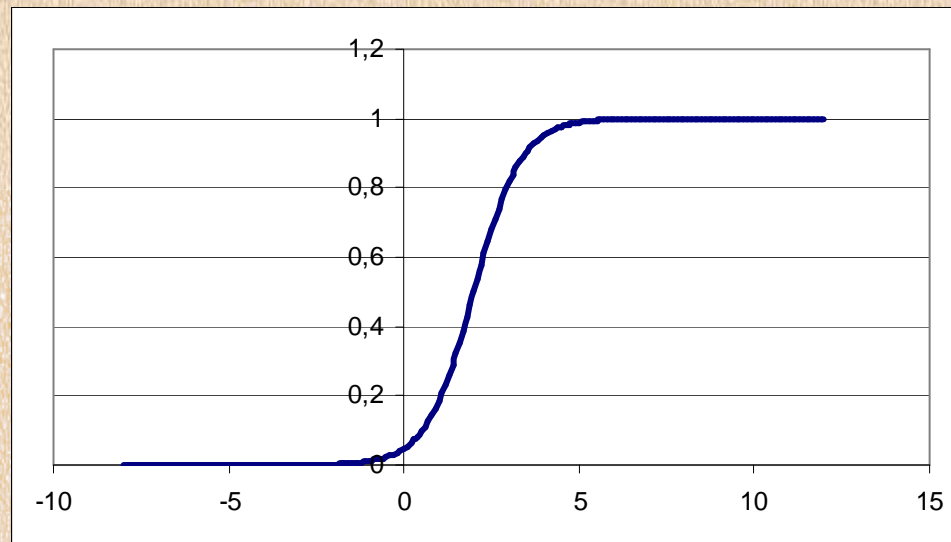
FUNKCJE AKTYWACJI (4)



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

$$\theta = 2$$

$$\alpha = 1.5$$



FUNKCJE AKTYWACJI (5)



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



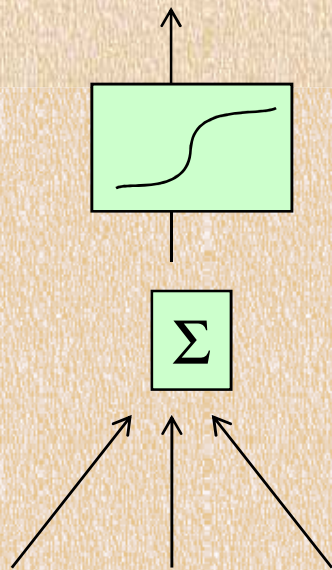
UNIWERSYTET
KAZIMIERZA WIELKIEGO
W BYDGOSZCZY

SIECI NEURONOWE



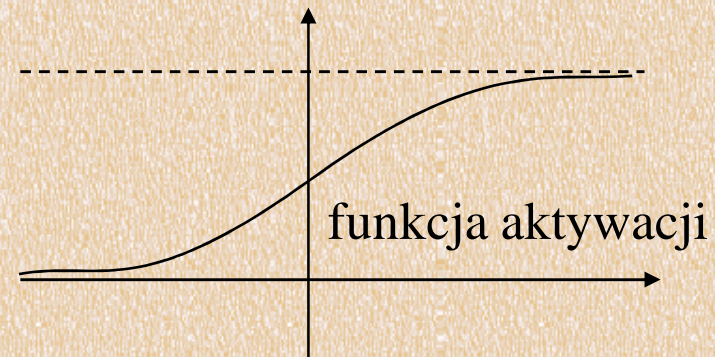
Potrafią modelować (dowolnie dokładnie przybliżać) funkcje rzeczywiste

(z tw. Kołmogorowa)



$$y = f\left(w_0 + \sum_{i=1}^n w_i x_i\right)$$

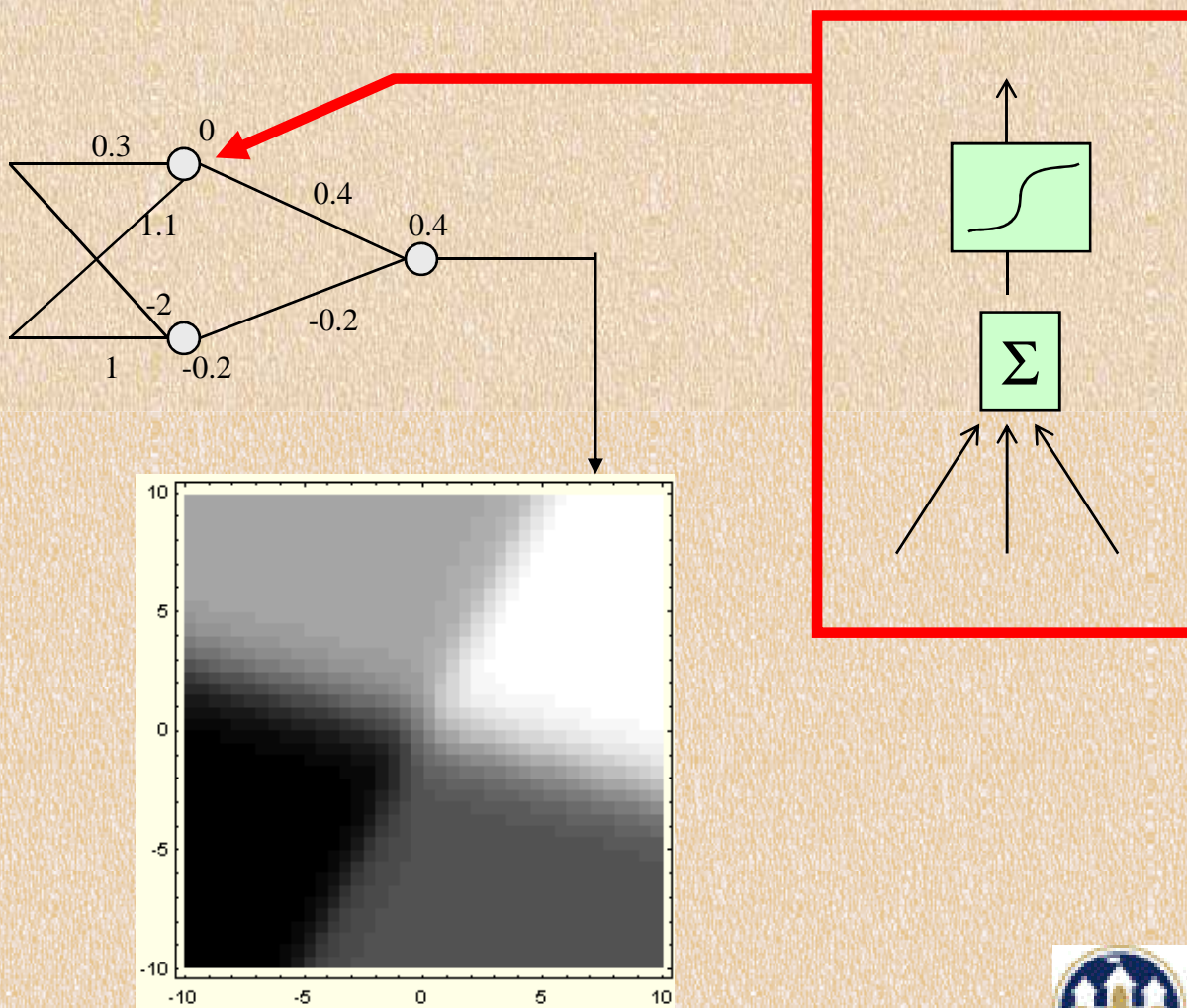
$$f(s) = \frac{1}{1 + e^{-s}}$$





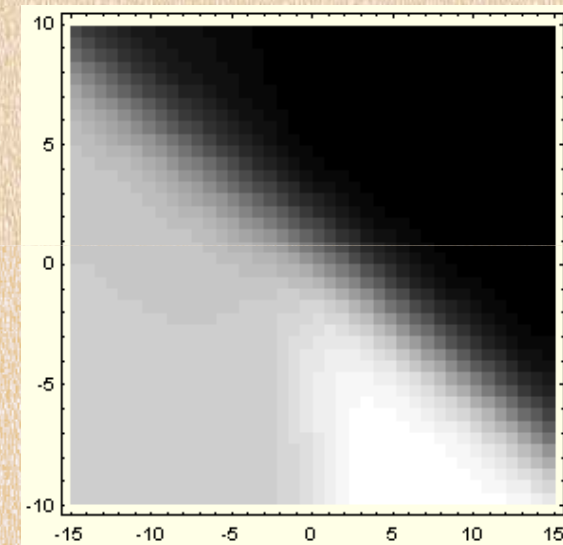
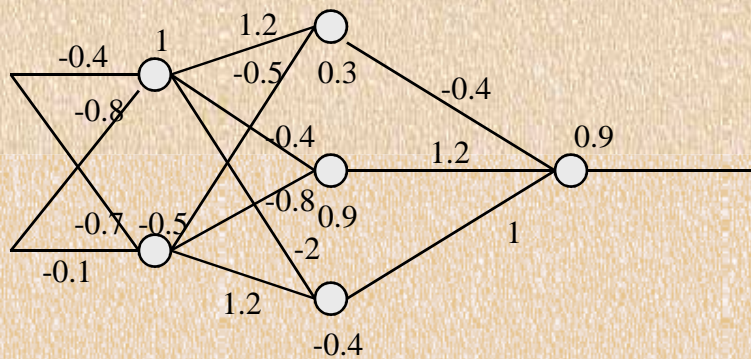
SIECI NEURONOWE

Sieć tworzy teksturę



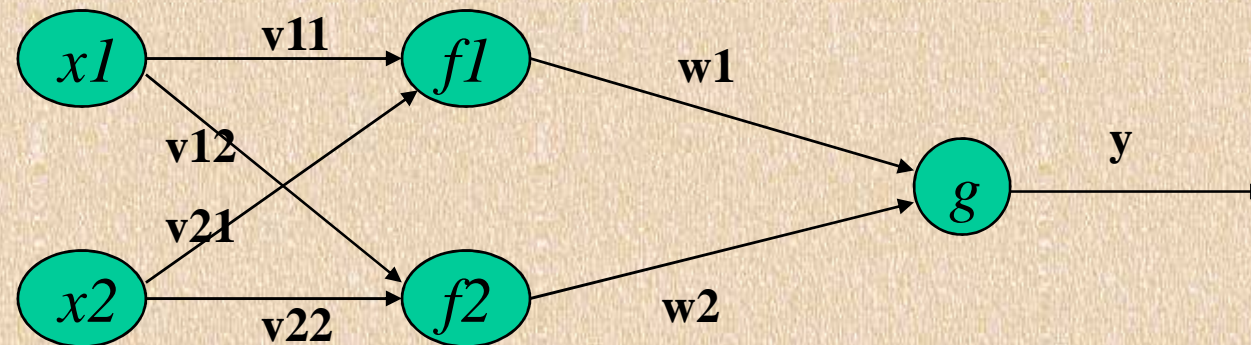


SIECI NEURONOWE





SIECI JAKO FUNKCJE ZŁOŻONE (1)



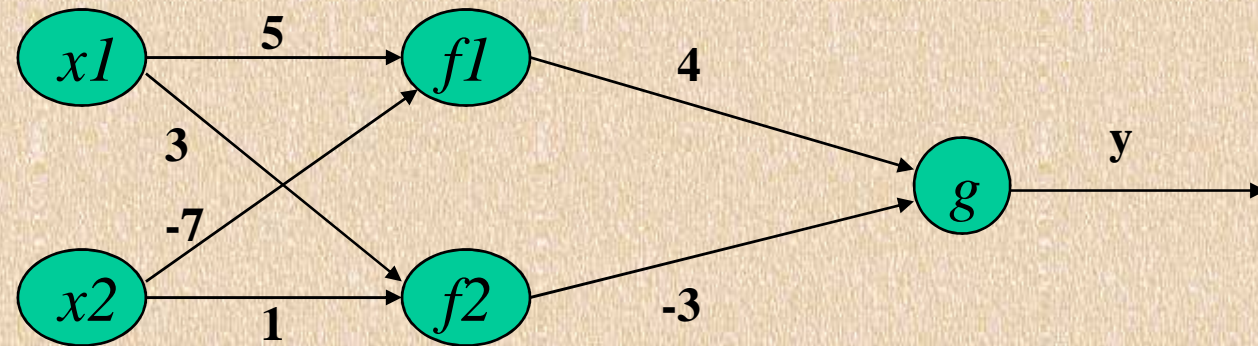
$$y = g(w_1 f_1(v_{11}x_1 + v_{21}x_2) + w_2 f_2(v_{12}x_1 + v_{22}x_2))$$

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





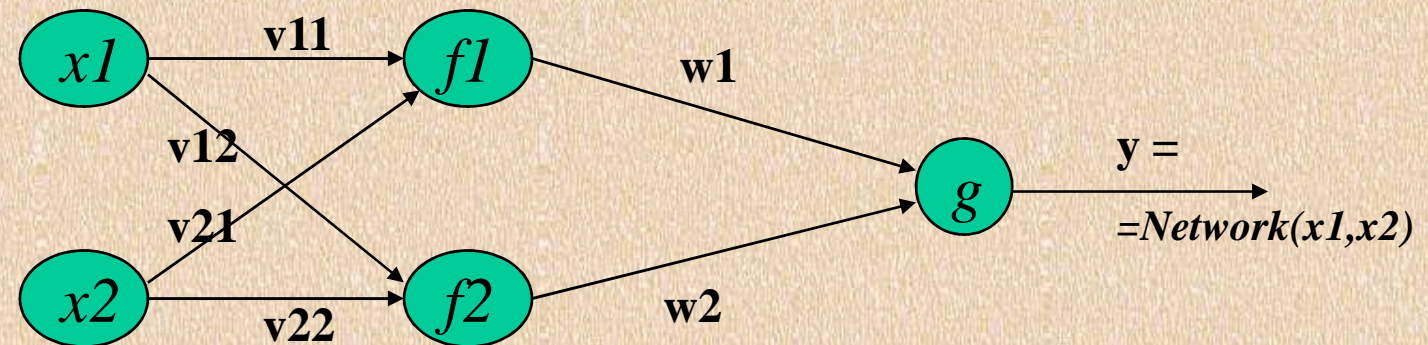
SIECI JAKO FUNKCJE ZŁOŻONE (2)



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



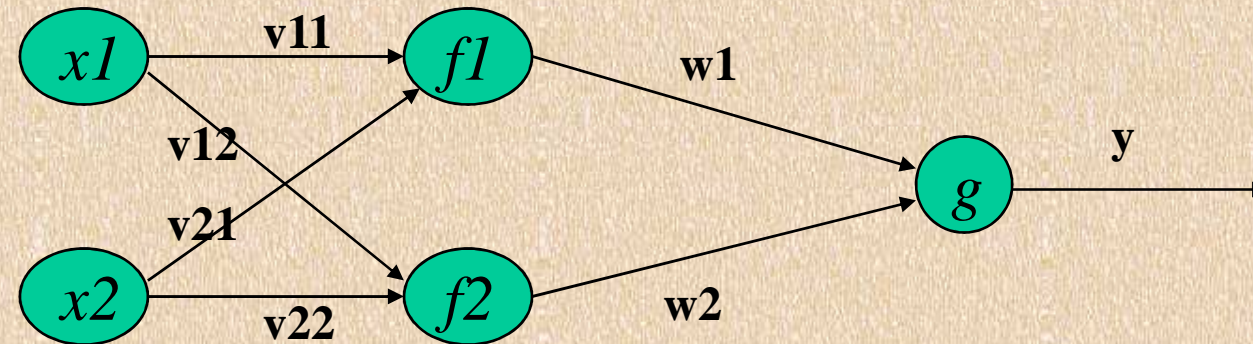
SIECI JAKO FUNKCJE ZŁOŻONE (3)



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



SIECI JAKO FUNKCJE ZŁOŻONE – przypadek liniowy



- Niech

$$f_i(x1, x2) = a_i * (x1 * v_{i1} + x2 * v_{i2}) + b_i$$

$$g(z1, z2) = a * (z1 * w1 + z2 * w2) + b$$

- Wtedy

$$Network(x1, x2) = A1 * x1 + A2 * x2 + B$$

- Np.:

$$A1 = a * (a1 * v1 * w1 + a2 * v2 * w2)$$





PROPAGACJA WSTECZNA BŁĘDU (1)

$E - f. błędów$

- Chcemy “wytrenować” wagi połączeń między kolejnymi warstwami neuronów. Jest to tzw. **proces adaptacji** wag. Jego algorytm odpowiada zadaniu minimalizacji funkcji błędu. Jest to **uczenie pod nadzorem**, zwane z nauczycielem, gdyż mamy zbiór danych trenujących.
- Inicjujemy wagi losowo (na małe wartości)
- Dla danego wektora uczącego obliczamy odpowiedź sieci (warstwa po warstwie)
- Każdy neuron wyjściowy oblicza swój błąd, odnoszący się do różnicy pomiędzy obliczoną odpowiedzią y oraz poprawną odpowiedzią t .
- Następnie ten błąd jest rozkładany na poszczególne połączenia, zaczynając od połączenia wyjściowego.

$$\eta(k) \\ \eta > 0$$

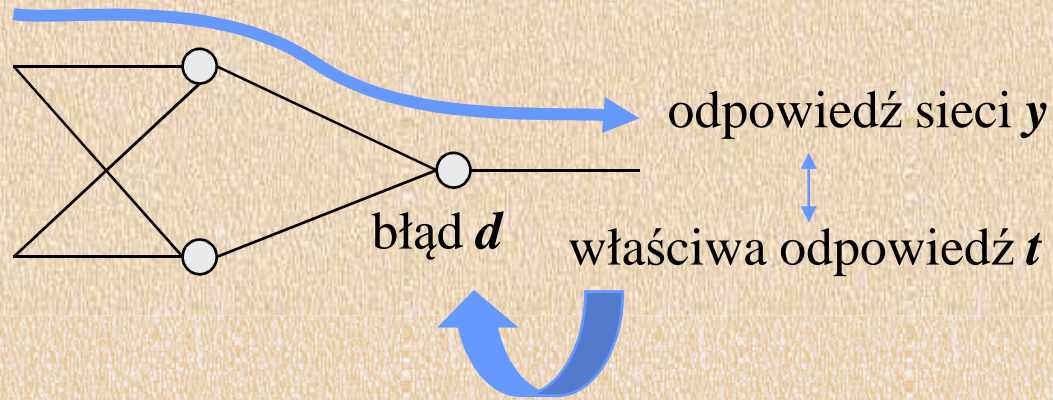
$$\underline{w}(k+1) = \underline{w}(k) - \eta \frac{\partial E(k)}{\partial \underline{w}} \\ \underline{w}(0) = \underline{w}_0$$



PROPAGACJA WSTECZNA BŁĘDU (2)



dane uczące



Błąd sieci definiowany jest zazwyczaj jako

$$d = \frac{1}{2} (y - t)^2$$



PROPAGACJA WSTECZNA BŁĘDU (3)



- Oznaczmy przez:
 - $f: \mathbf{R} \rightarrow \mathbf{R}$ – funkcję aktywacji w neuronie
 - w_1, \dots, w_K – wagi połączeń wchodzących
 - z_1, \dots, z_K – sygnały napływające do neuronu z poprzedniej warstwy
- Błąd neuronu traktujemy jako funkcję wag połączeń do niego prowadzących:

$$d(w_1, \dots, w_K) = \frac{1}{2} \left(f(w_1 z_1 + \dots + w_K z_K) - t \right)^2$$



PRZYKŁAD (1)

- Rozpatrzmy model, w którym:
 - Funkcja aktywacji przyjmuje postać

$$f(s) = \frac{1}{1 + e^{-3(s+2)}}$$

- Wektor wag połączeń = [1;-3;2]
- Załóżmy, że dla danego przykładu:
 - Odpowiedź powinna wynosić $t = 0.5$
 - Z poprzedniej warstwy dochodzą sygnały [0;1;0.3]



PRZYKŁAD (2)

- Liczymy wejściową sumę ważoną:

$$s = w_1x_1 + w_2x_2 + w_3x_3 = 1 \cdot 0 + (-3) \cdot 1 + 2 \cdot 0.3 = -2.4$$

- Liczymy odpowiedź neuronu:

$$y = f(s) = \frac{1}{1 + e^{-3(-2.4+2)}} = \frac{1}{1 + e^{1.2}} \approx 0.23$$

- Błąd wynosi:

$$d = \frac{1}{2} (0.23 - 0.5)^2 \approx 0.036$$



IDEA ROZKŁADU BŁĘDU



- Musimy „rozłożyć” otrzymany błąd na połączenia wprowadzające sygnały do danego neuronu
- Składową błędu dla każdego j -tego połączenia określamy jako pochodną cząstkową funkcji błędu $d(x, y, t)$ względem j -tej wagi
- Składowych tych będziemy mogli użyć do zmodyfikowania ustawień poszczególnych wag połączeń

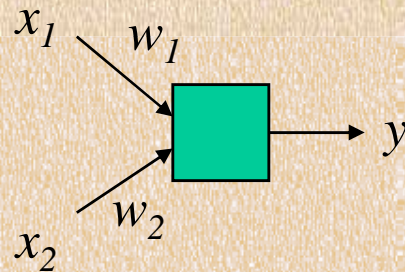


IDEA ROZKŁADU BŁĘDU (2)



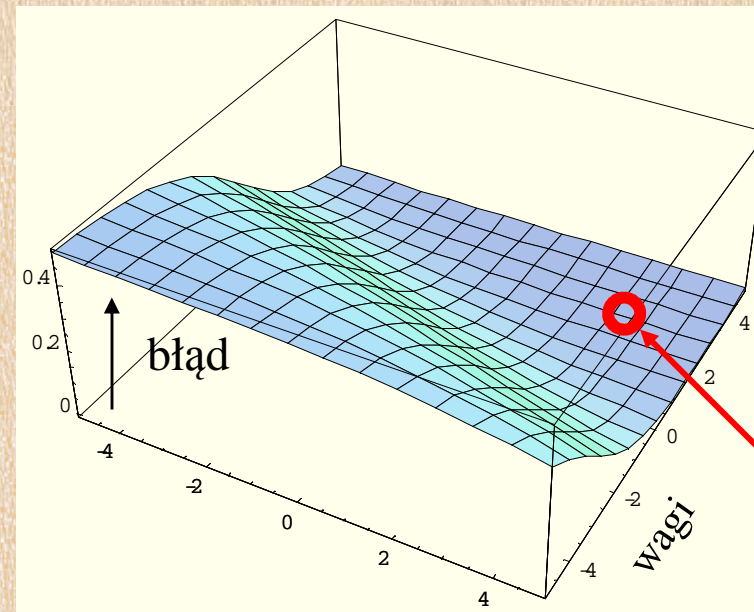
Założmy, że mamy neuron z wagami $w_0=0$, $w_1=2$, $w_2=3$. Mamy dane wektor wejściowy: $[0.3, 0.7]$, przy czym oczekiwana odpowiedź to $t=1$. Jak należy zmienić wagi, aby błąd był jak najmniejszy?

Mozemy błąd przedstawić jako funkcję w_1, w_2 :



$$y = f\left(w_0 + \sum_{i=1}^n w_i x_i\right)$$

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



wartość błędu
dla wag $[2, 3]$

Wagi powinniśmy zmienić się w kierunku spadku wartości błędu.

KIERUNEK ZMIANY WAG



Jeśli rozważymy większą liczbę przykładów, funkcja średniego błędu będzie miała bardziej skomplikowany kształt.

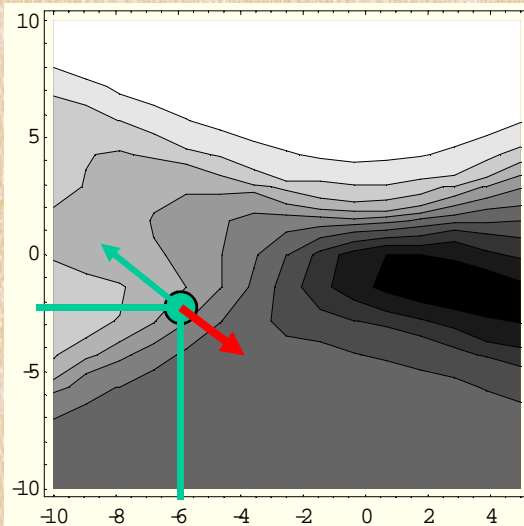
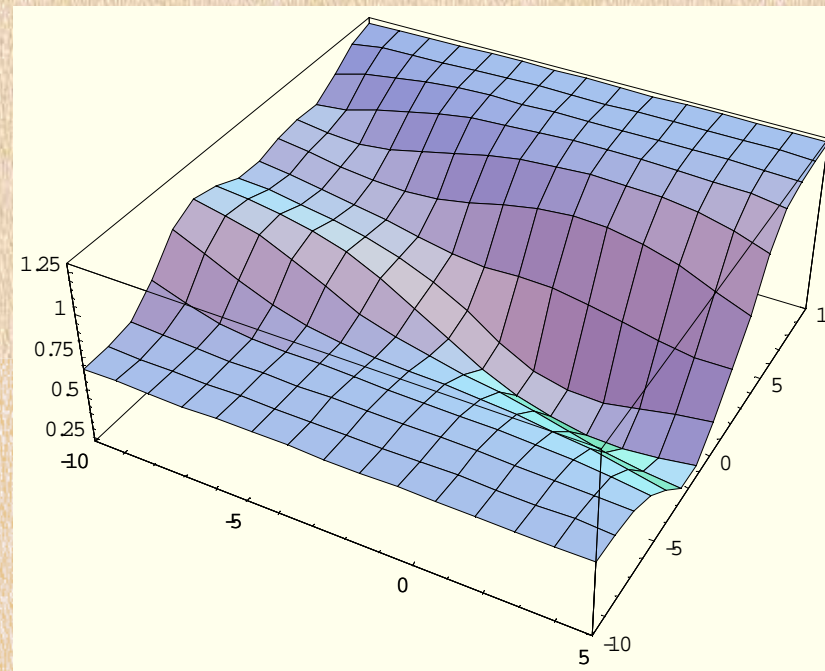
$[0.3, 0.7]$, $t=1$

$[0.2, 0.9]$, $t=0.1$

$[-0.6, 1]$, $t=0$

$[0, -0.8]$, $t=0.5$

$[0.6, 1]$, $t=0.3$



Nachylenie wykresu w danym punkcie (odpowiadającym aktualnym wartościom wag) dane jest przez **gradient**, czyli wektor pochodnych cząstkowych.

Zmiana wag powinna nastąpić w kierunku przeciwnym.



OBLICZANIE POCHODNEJ



$$\frac{\partial d(w_1, \dots, w_K)}{\partial w_j} = (y - t) \cdot f'(s) \cdot z_j$$

$$= \frac{\partial \frac{1}{2} (f(w_1 z_1 + \dots + w_K z_K) - t)^2}{\partial w_j}$$

$$= \frac{\partial \frac{1}{2} (y - t)^2}{\partial y} \cdot \frac{\partial f(s)}{\partial s} \cdot \frac{\partial (w_1 z_1 + \dots + w_K z_K)}{\partial w_j}$$



- Idea:
 - Wektor wag połączeń powinniśmy przesunąć w kierunku **przeciwnym** do wektora gradientu błędu (z pewnym współczynnikiem uczenia η)
 - Możemy to zrobić po każdym przykładzie uczącym, albo sumując zmiany po kilku przykładach.
- Realizacja:
$$\Delta w_j = \eta \cdot (t - y) \cdot f'(s) \cdot z_j$$

Prosty przykład: wagi $w_1=1$, $w_2=1$, dane wejściowe: $[0.5, 0.5]$, $t = 1$.

Funkcja sigmoidalna: $f(s) = \frac{1}{1 + e^{-s}}$ więc: $f'(s) = \frac{e^{-s}}{(1 + e^{-s})^2}$

Stąd: $s = 0.5 + 0.5 = 1$, $y = 0.731$, zmiana $w = (1 - 0.731) \cdot 0.19 \cdot 0.5 = 0.026$.

A więc nowe wagi to 1.026. Ten sam przykład da tym razem odpowiedź $y=0.736$.

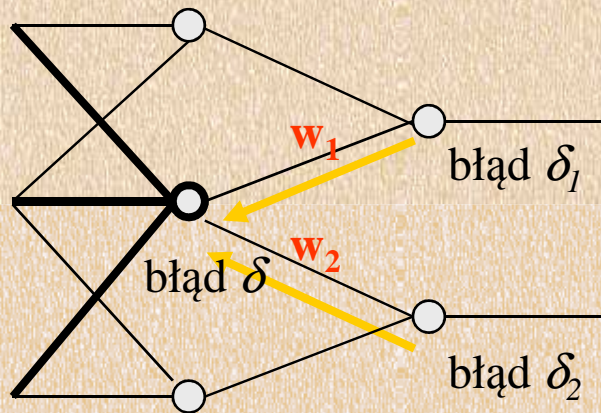


Błędy są następnie propagowane w kierunku poprzednich warstw. Wprowadźmy pomocniczo współczynnik błędu δ zdefiniowany dla ostatniej warstwy jako:

$$\delta = f'(s) \cdot (t - y)$$

a dla pozostałych warstw:

$$\delta = f'(s) \cdot \sum_{i=1}^n w_i \delta_i$$



czyli neuron w warstwie ukrytej “zbiera” błąd z neuronów, z którymi jest połączony.

Zmiana wag połączeń następuje po fazie propagacji błędu i odbywa się według wzoru:

$$\Delta w = \eta \cdot \delta \cdot z$$

Oznaczenia: w - waga wejścia neuronu, z - sygnał wchodzący do neuronu danym wejściem, δ - współczynnik błędu obliczony dla danego neuronu, s - wartość wzbudzenia (suma wartości wejściowych pomnożonych przez wagi) dla danego neuronu.



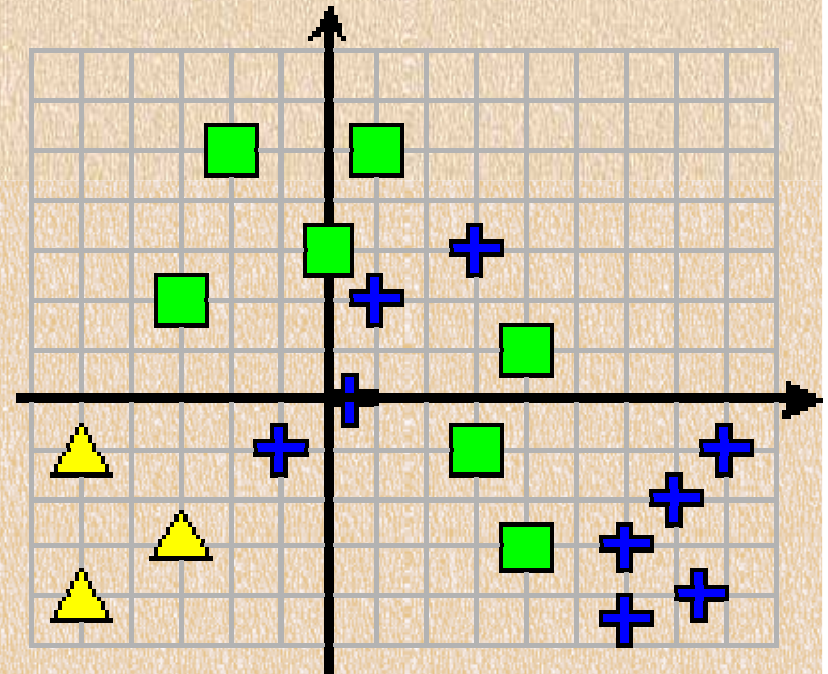
Zadania sprawdzające:

1. Co charakteryzuje prosty perceptron?
2. Podać inną funkcję logiczną niż XOR, której nie potrafi obliczyć sieć neuronowa.
3. Jaką własność posiada każda funkcja aktywacji?
4. Co to jest równanie perceptronowe? Jakie jest jego znaczenie?
5. Co potrafi zrobić pojedynczy neuron?



Co potrafi układ perceptronów?

- Klasyfikować punkty na płaszczyźnie należące do kilku różnych obszarów
- Jeśli funkcje decyzyjne neuronów w warstwie wewnętrznej są afiniczne, to różne obszary są rozdzielane prostymi (ogólnie: hiperpłaszczyznami w przestrzeni n -wymiarowej).
- Układ perceptronów, który jest już siecią neuronową perceptronową realizuje klasyfikator.





ROZPOZNAWANIE WZORCÓW

- Wzorce: obrazy, nagrania, dane personalne, sposoby prowadzenia pojazdu, etc.
- Reprezentacja wzorca:
 - Wektor cech (wejść do sieci neuronowej)
- Klasyfikacja wzorców: Klasyfikacja do jednej z istniejących klas
- Formowanie klas wzorców , tutaj sieć samoorganizująca się, np. ART, Kohonena, uczenie bez nauczyciela
- Asocjacyjne odtwarzanie wzorców, tutaj sieć Hopfielda: każdy neuron połączony z każdym
 - Odtwarzanie wzorców podobnych
 - Uzupełnianie wzorców
 - Odzyskiwanie (czyszczenie) wzorców



Przykład zagadnienia praktycznego

- Znaleźć, odczytać i zapamiętać numer rejestracyjny samochodu na podstawie zdjęcia:



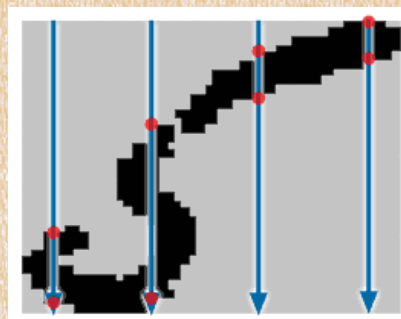
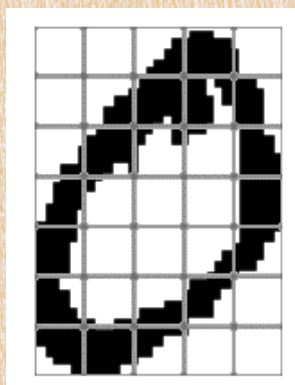
Odczytywanie tablic rejestracyjnych (2)



Wyselekcjonowany obszar



Lokalizacja znaków



Rozpoznawanie znaków:

- znajdowanie istotnych cech liczbowych
- klasyfikacja na podstawie cech (systemy uczące się)



Wykorzystywane technik sztucznej inteligencji i ich narzędzi

- Sieci neuronowe
- Wnioskowanie, indukcja reguł
- Algorytmy ewolucyjne
- Systemy wieloagentowe (współpraca)
- Automaty komórkowe
- Metody przeszukiwania możliwych rozwiązań i ich optymalizacji...



PRZYKŁADOWE POLE DO POPISU

- Analiza dźwięku, obrazu, bądź danych multimedialnych, nie może opierać się ani wyłącznie na sieciach neuronowych, ani na, np., drzewach decyzyjnych czy AG .
- Konieczne jest połączenie metod numerycznych, naśladujących działanie ludzkich zmysłów, z metodami symbolicznymi, naśladującymi ludzkie rozumowanie .



Zbiory rozmyte

Sposób formalnego opisu nieprecyzyjności

Literatura

1. **Piegat A.** *Modelowanie i sterowanie rozmyte*. Akademicka Oficyna Wydawnicza EXIT Warszawa 1999
2. **Rutkowska D., Piliński M, Rutkowski L.** *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*. Wyd. Naukowe PWN Warszawa 1997



Zbiory rozmyte

naśladowanie ludzkiej nieprecyzyjnej oceny otoczenia

- Ludzie patrzą na świat nieprecyzyjnie: „bardzo zimno”, „szybko”, „niedaleko”
- Ludzie potrafią radzić sobie mimo nieprecyzyjnej oceny nawet w ekstremalnych sytuacjach : przechodzenie przez jezdnię, sterowanie samolotem



Przynależność do zbioru

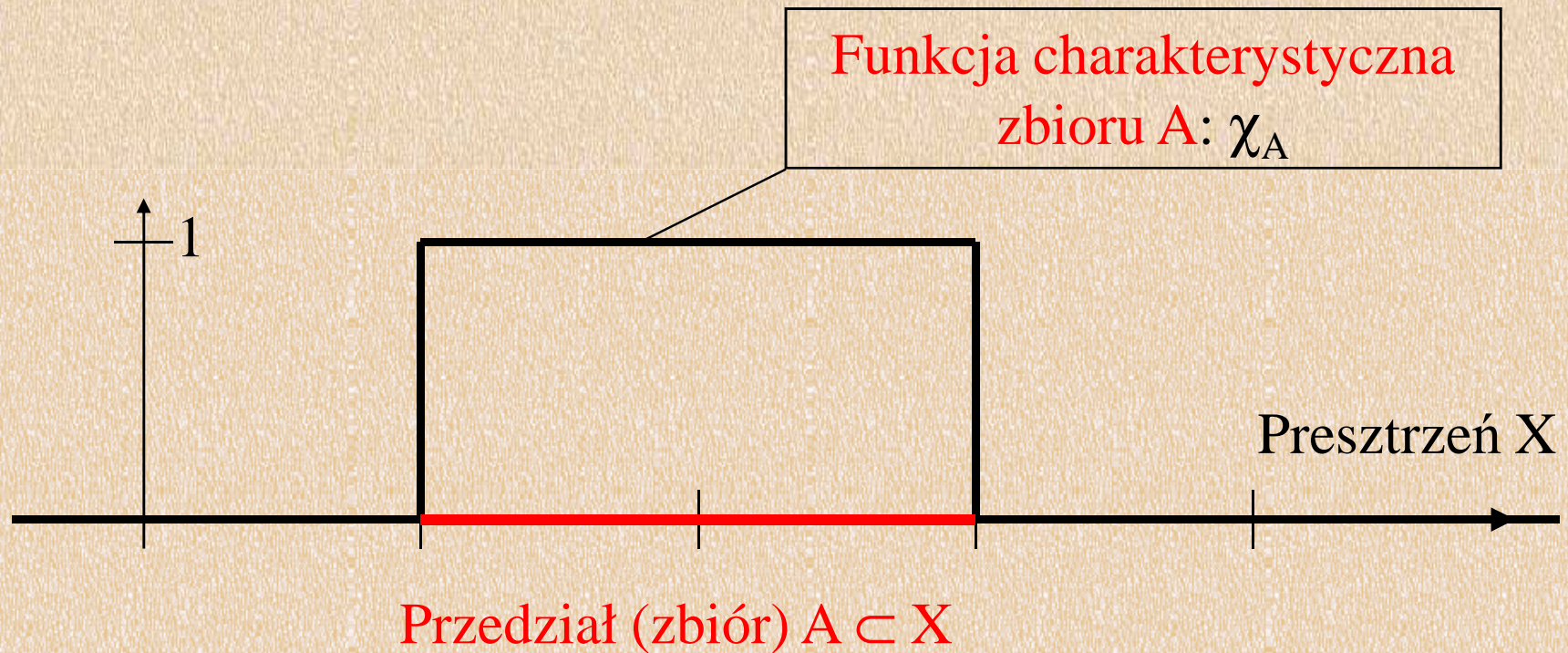
- **Zbiory klasyczne** – przynależność całkowita
- Czy duży stos kamieni przestanie być dużym stosem kamieni, gdy zabierzemy jeden? A jak dwa, a jak 22?
- Czy po zabraniu części kamienia myślimy o dużym stosie jako o nieco mniejszym?
- Czy cena za produkt 3,99 jest w codziennym życiu równoważna cenie 4,00?
- **Zbiory rozmyte** – przynależność częściowa
- Przestrzeń zbiorów klasycznych jest podzbiorem przestrzeni zbiorów rozmytych, poprzez funkcję charakterystyczną tego zbioru, jako szczególnym przypadkiem funkcji przynależności zbioru rozmytego



Zbiór klasyczny

jak jednoznacznie opisać?

Funkcja charakterystyczna - odpowiednik zbioru klasycznego





Definicje



DEFINICJA

Zbiorem rozmytym A na pewnej przestrzeni X , nazywamy zbiór par:

$$A = \{(x, \mu_A(x))\} \quad \forall x \in X$$

gdzie: μ_A jest funkcją, która przypisuje każdemu elementowi $x \in X$ (przyjętej przestrzeni rozważań X) **jego stopień przynależności** do zbioru A , przy czym:

$$\mu_A: X \rightarrow [0, 1],$$

zatem $\mu_A(x) \in [0, 1]$. Można to odebrać jako zdanie w logice wielowartościowej, gdzie 0 – fałsz, 1 – prawda.



Funkcja μ_A nazywana jest **funkcją przynależności**, zaś jej wartość dla danego argumentu nazywana jest **stopniem przynależności x do zbioru rozmytego A** . Stopień przynależności określa, w jakim stopniu rozpatrywany argument należy do zbioru rozmytego A . Można zauważyć, że funkcja μ_A wraz z dziedziną jednoznacznie wyznaczają zbiór A .

Zbiór rozmyty, którego funkcja przynależności osiąga wartość 1 dla co najmniej jednego elementu nazywany jest zbiorem rozmytym normalnym.



Dla każdego zbioru rozmytego wyznacza się często jego integralny parametr pomocny przy określaniu i analizie różnych własności - nośnik (*ang. support*).

DEFINICJA

Nośnikiem zbioru rozmytego A w X jest zbiór nierozmyty oznaczany jako $\text{supp}(A)$ i określony następująco:

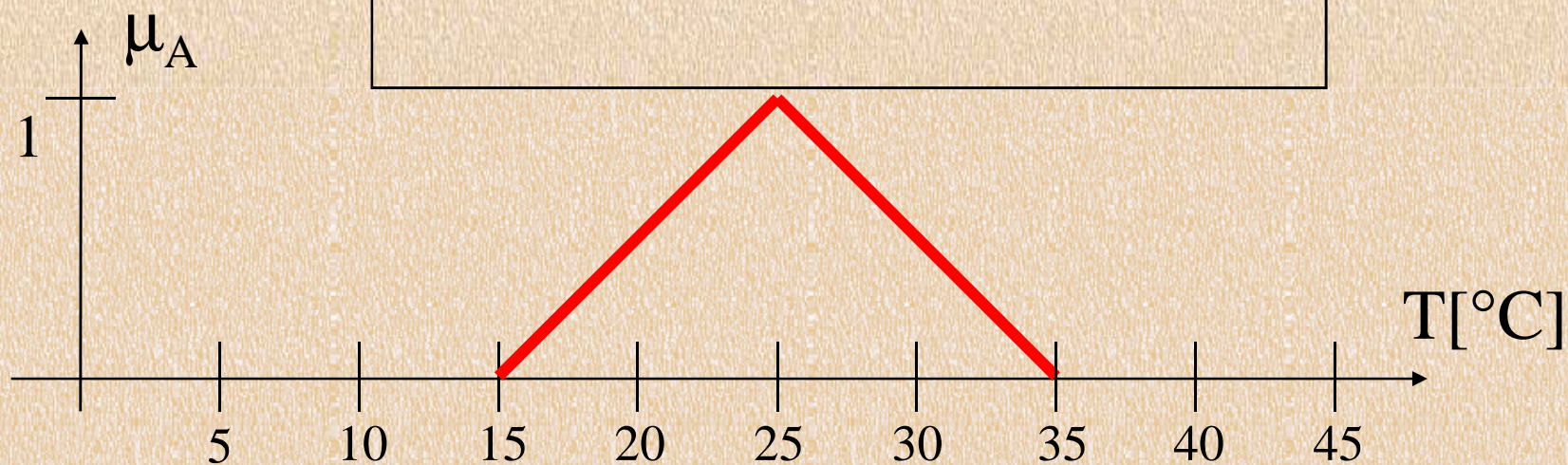
$$\text{supp}(A) = \{x: \mu_A(x) > 0\}.$$

Inaczej mówiąc, nośnikiem nazywamy taki podzbiór dziedziny funkcji przynależności, dla którego elementów, wartości funkcji są większe od zera.



Przykład zbioru rozmytego (1)

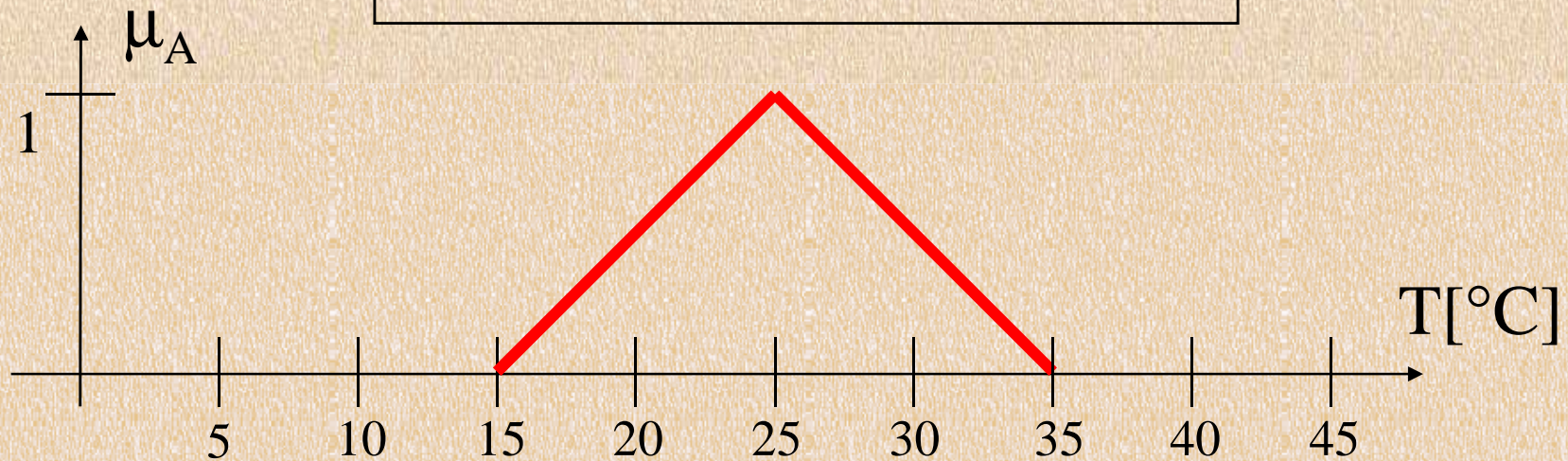
Zbiór rozmyty reprezentujący
określenie „ciepła pogoda”.





Przykład zbioru rozmytego (2)

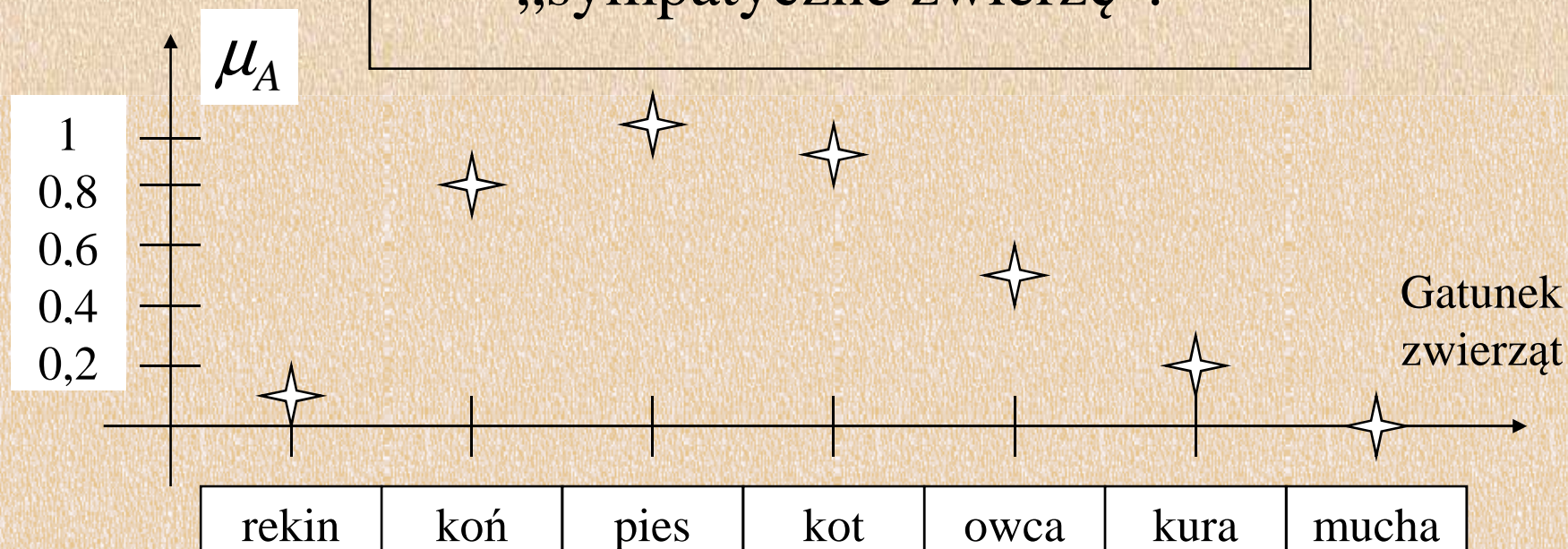
Zbiór rozmyty
reprezentujący określenie
„ciepła pogoda”.





Przykład zbioru rozmytego (3)

Zbiór rozmyty (dyskretny)
reprezentujący określenie
„sympatyczne zwierzę”.



Działania na zbiorach rozmytych

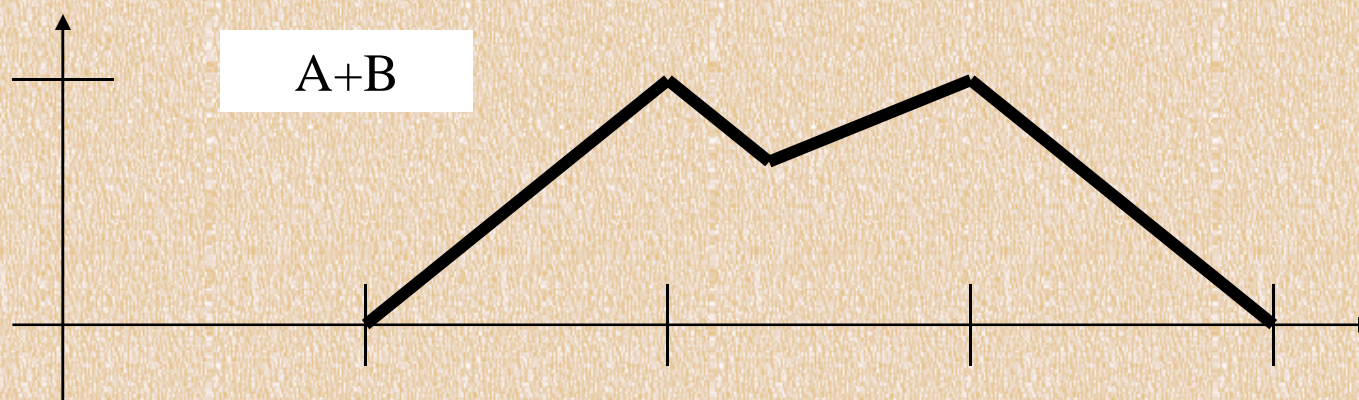
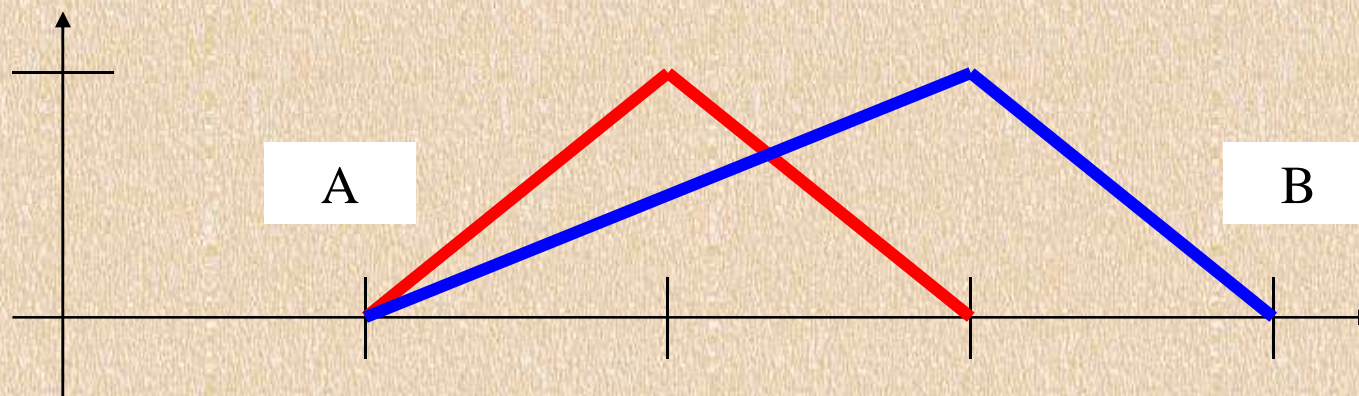


Istnieją różne sposoby definiowania działań na zbiorach rozmytych. Tutaj zostaną omówione te zaproponowane przez Zadeha w 1965r. zwane działaniami mnogościowymi.

Sumą zbiorów rozmytych A i B z funkcjami przynależności (odpowiednio μ_A i μ_B) określonymi na tym samym zbiorze X nazywamy zbiór C wyznaczony przez funkcję przynależności μ_C

$$\mu_C(x) = \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

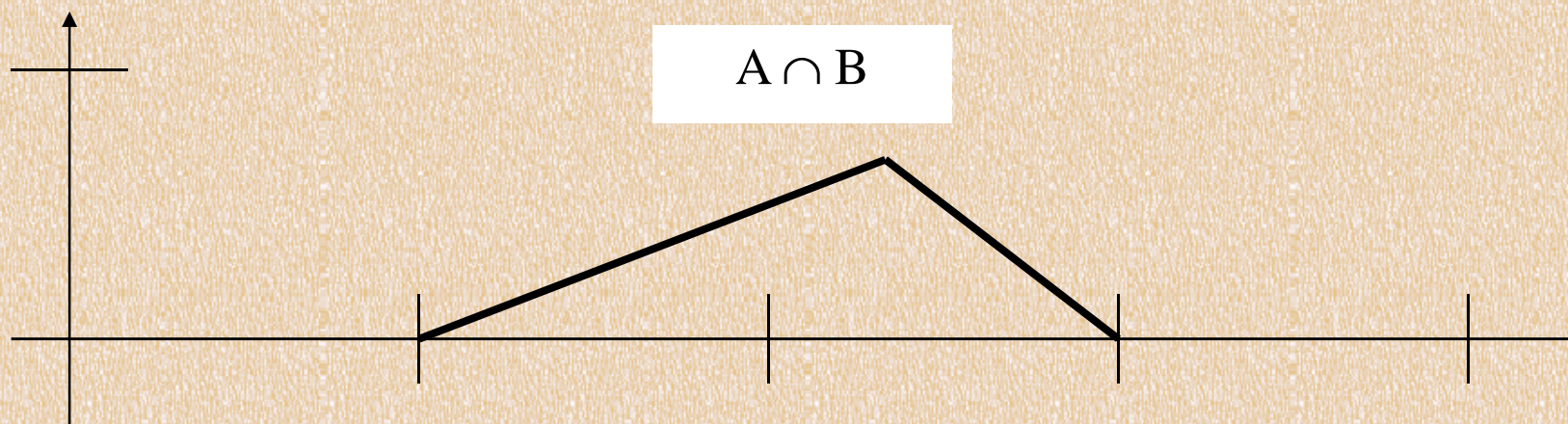
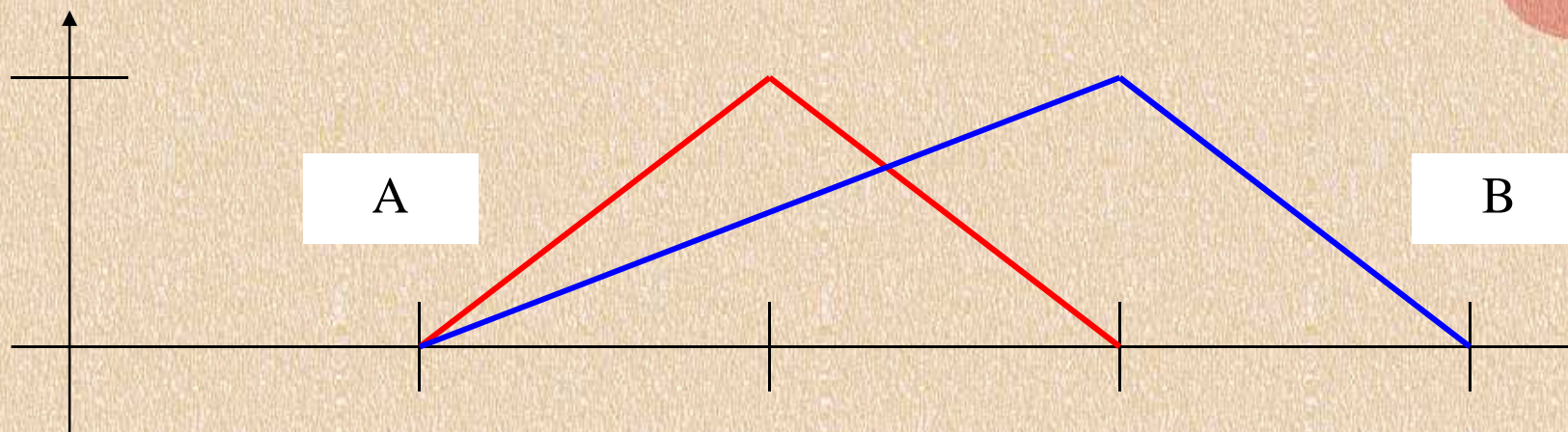
gdzie $x \in X$.





Iloczynem (przecięciem) zbiorów rozmytych A i B z funkcjami przynależności (odpowiednio μ_A i μ_B) określonymi na tym samym zbiorze X nazywamy zbiór C wyznaczony przez funkcję przynależności μ_C

$$\mu_C(x) = \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad \text{gdzie } x \in X.$$

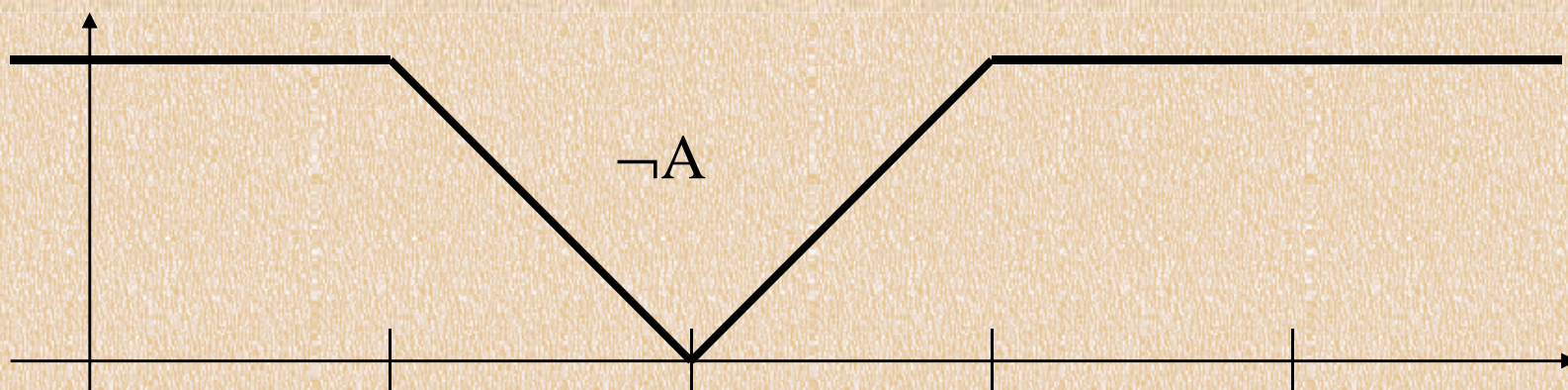
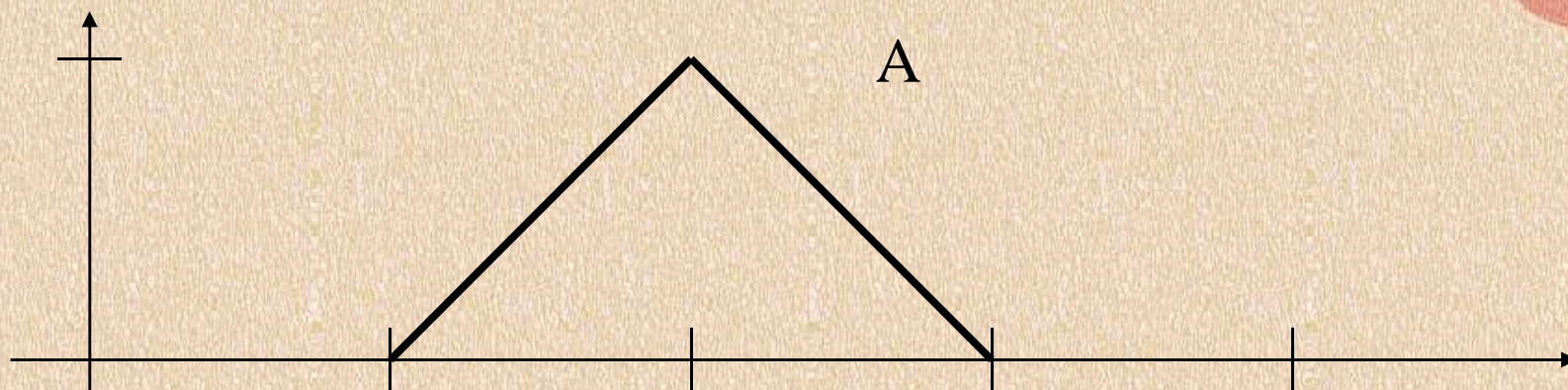




Dopełnieniem zbioru A określonego na przestrzeni X jest zbiór rozmyty $\neg A$ wyznaczony przez funkcję przynależności $\mu_{\neg A}$

$$\mu_{\neg A}(x) = 1 - \mu_A(x)$$

gdzie $x \in X$.





Własności działań w klasycznej teorii zbiorów

| | |
|------------------------------|--|
| Inwolucja (podwójna negacja) | $A = \neg(\neg A)$ |
| Przemienność | $A \cup B = B \cup A$ $A \cap B = B \cap A$ |
| Łączność | $(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap C)$ |
| Rozdzielność | $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ |
| Idempotencja | $A = A \cap A, A = A \cup A$ |
| Pochłanianie (absorpcja) | $A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$ |



| | |
|--------------------------------------|--|
| Pochłanianie dopełnienia | $A \cup (\neg A \cap B) = A \cup B$ $A \cap (\neg A \cup B) = A \cap B$ |
| Pochłanianie przez \emptyset i U | $A \cup U = U$ $A \cap \emptyset = \emptyset$ |
| Identyczność | $A \cup \emptyset = A$ $A \cap U = A$ |
| Prawo zaprzeczenia | $A \cap \neg A = \emptyset$ |
| Prawo wyłączonego środka | $A \cup \neg A = U$ |
| Prawa de Morgana | $\neg(A \cap B) = \neg A \cup \neg B$ $\neg(A \cup B) = \neg A \cap \neg B$ |

U – uniwersum do którego należą rozważane zbiory A , B i C

\emptyset - zbiór pusty, jego funkcja charakterystyczna jest stała i równa zero



Własności spełniane przez działania mnogościowe na zbiorach rozmytych

| | |
|--------------------------------------|-----|
| Inwolucja | tak |
| Przemienność | tak |
| Łączność | tak |
| Rozdzielność | tak |
| Idempotencja | tak |
| Pochłanianie | tak |
| Pochłanianie dopełnienia | nie |
| Pochłanianie przez \emptyset i U | tak |
| Identyczność | tak |
| Prawo zaprzeczenia | nie |
| Prawo wyłączonego środka | nie |
| Prawa de Morgana | tak |



Operatory t -normy i s -normy – normy trójkątne

Istnieją różne rodzaje działań, które można nazywać sumą lub iloczynem zbiorów. Warunki, które muszą być spełnione, by dane działanie było sumą nazywane są s -normą, iloczynem – t -normą. Ogólnie nazywa się je normami trójkątnymi.

s -normą nazywa się funkcję $S: [0, 1] \times [0, 1] \rightarrow [0, 1]$ taką, że dla każdego $a, b, c \in [0, 1]$ spełnione są warunki

- łączność – $S(S(a, b), c) = S(a, S(b, c))$
- przemienność – $S(a, b) = S(b, a)$
- monotoniczność – dla $b \leq c$ zachodzi $S(a, b) \leq S(a, c)$
- warunek brzegowy (element neutralny) – $S(a, 0) = a$



t – normą nazywa się funkcję $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$ niemalejącą (monotoniczną) oraz spełniającą warunki łączności, przemienności (jak w przypadku \underline{s} -normy), a także warunek brzegowy:

$$T(a, 1) = a$$

Dla każdej konkretnej normy trójkątnej istnieje norma do niej dualna inaczej nazywana jej ko-normą. Warunkiem tego, by s -norma była dualna do danej t -normy (i na odwrót) jest spełnianie poniższych zależności:

$$S(a, b) = 1 - T(1 - a, 1 - b)$$

$$T(a, b) = 1 - S(1 - a, 1 - b),$$

które można rozpatrywać jak uogólnienie praw de Morgana.



Przykładowe często wykorzystywane normy trójkątne:

Norma maksyminowa

t – norma – minimum: $T(a, b) = a \wedge b = \min(a, b)$

s – norma – maksimum: $S(a, b) = a \vee b = \max(a, b)$

Norma Larsena

t – norma - iloczyn algebraiczny: $T(a, b) = a \cdot b$

s – norma - iloczyn probablistyczny: $S(s, b) = a + b - (a \cdot b)$

Mimo, iż normy trójkątne podają ogólne warunki, jakie musi spełniać dane działanie, by można je było nazwać dodawaniem lub mnożeniem, to są wygodnym narzędziem służącym do definiowania działań także na zbiorach rozmytych (zatem także liczbach rozmytych).



Przykładowe częściej wykorzystywane normy trójkątne:

Norma **maksyminowa**

t – norma – minimum: $T(a, b) = a \wedge b = \min(a, b)$

s – norma – maksimum: $S(a, b) = a \vee b = \max(a, b)$

Norma **Larsena**

t – norma - **iloczyn algebraiczny**: $T(a, b) = a \cdot b$

s – norma - **iloczyn probabilistyczny**: $S(s, b) = a + b - (a \cdot b)$

Mimo, iż normy trójkątne podają ogólne warunki, jakie musi spełniać dane działanie, by można je było nazwać dodawaniem lub mnożeniem, to są wygodnym narzędziem służącym do definiowania działań na zbiorach rozmytych (zatem także liczbach rozmytych, które są szczególnym przypadkiem gdy $X=R$).