

# WYKŁAD 4

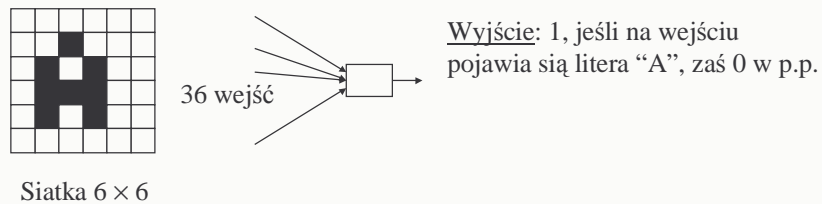
Sieci neuronowe:  
Algorytmy uczenia &  
Dalsze zastosowania

## PLAN WYKŁADU

- Metody uczenia sieci:
  - Uczenie perceptronu
  - Propagacja wsteczna
- Zastosowania
  - Sterowanie (powtórzenie)
  - Kompresja obrazu
  - Rozpoznawanie wzorców

## REGUŁA UCZENIA PERCEPTRONU (1)

**Przykład:** rozpoznawanie znaków



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

## REGUŁA UCZENIA PERCEPTRONU (2)

### ■ Wejście:

- Ciąg przykładów uczących ze znanymi odpowiedziami

### ■ Proces uczenia:

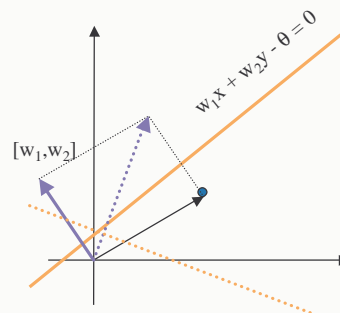
- Inicjujemy wagi losowo
- Dla każdego przykładu, jeśli odpowiedź jest nieprawidłowa, to

$$w_1 + = \alpha x_1$$

$$w_2 + = \alpha x_2$$

$$\theta - = \alpha$$

gdzie  $\alpha$  jest równe odpowiedzi prawidłowej



### REGUŁA UCZENIA PERCEPTRONU (3)

- Często  $\alpha$  mnoży się dodatkowo przez niewielki współczynnik uczenia
- Po wyczerpaniu przykładów, zaczynamy proces uczenia od początku, dopóki następują jakiegokolwiek zmiany wag połączeń

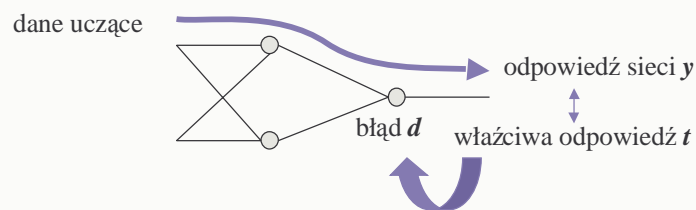
### REGUŁA UCZENIA PERCEPTRONU (4)

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

## PROPAGACJA WSTECZNA (1)

- Chcemy “wytrenować” wagi połączeń między kolejnymi warstwami neuronów
- 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  $t$

## PROPAGACJA WSTECZNA (2)



Błąd sieci definiowany jest zazwyczaj jako

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

## PROPAGACJA WSTECZNA (3)

- Oznaczmy przez:

- $g: \mathbf{R} \rightarrow \mathbf{R}$  – 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:

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

## PRZYKŁAD (1)

- Rozpatrzmy model, w którym:

- Funkcja aktywacji przyjmuje postać

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

- 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 ważeń:

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

- Liczymy odpowiedź neuronu:

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

- 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 błędów dla każdego  $j$ -tego połączenia określamy jako pochodną cząstkową błędów względem  $j$ -tej wagi
- Składowych tych błędów mogli użyć do zmodyfikowania ustawień poszczególnych wag połączeń

## OBLICZANIE POCHODNEJ

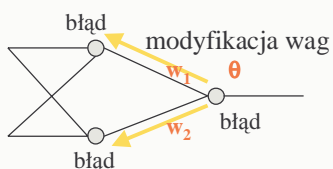
$$\frac{\partial d(w_1, \dots, w_K)}{\partial w_j} = (y - t) \cdot g'(s) \cdot z_j$$
$$= \frac{\partial \frac{1}{2} (g(w_1 z_1 + \dots + w_K z_K) - t)^2}{\partial w_j}$$
$$= \frac{\partial \frac{1}{2} (y - t)^2}{\partial y} \cdot \frac{\partial g(s)}{\partial s} \cdot \frac{\partial (w_1 z_1 + \dots + w_K z_K)}{\partial w_j}$$

## PROPAGACJA WSTECZNA (4)

### ■ Idea:

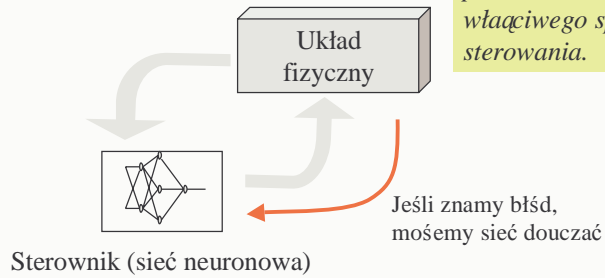
- Wektor wag połączeń powinniśmy przesunąć w kierunku przeciwnym do wektora rozkładu błęd

### ■ Realizacja:



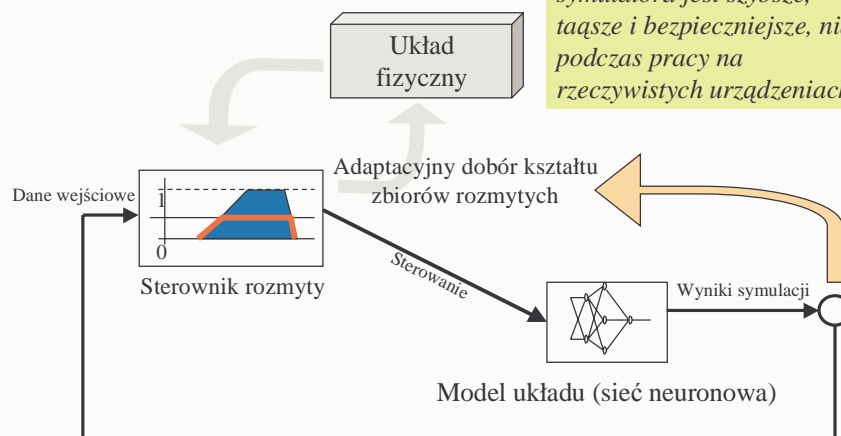
## STEROWANIE (1)

*Sieć neuronowa jest zwykle trenowana na modelu (symulatorze) układu, albo przez „obserwację” właściwego sposobu sterowania.*



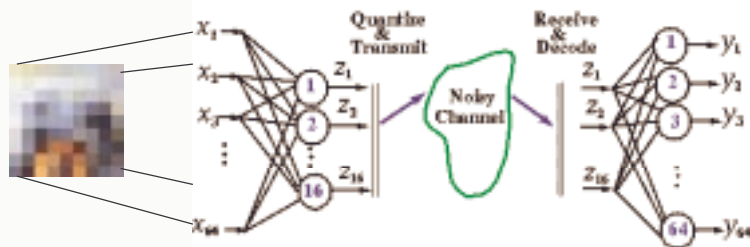
## STEROWANIE (2)

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





## KOMPRESJA OBRAZU



- Obraz dzielony jest na kwadraty (np. 8x8)
- Sieć neuronowa o 16 wyjąciach zamienia dane wejściowe na 16 liczb
- Liczby są kwantowane i przekazywane drugiej sieci, odtwarzającej dane oryginalne
- Kryterium nauki: minimalizacja błędów rekonstrukcji

## ROZPOZNAWANIE WZORCÓW

- Wzorce: obrazy, nagrania, dane personalne, sposoby prowadzenia pojazdu, etc.
- Reprezentacja:
  - Wektor cech (wejść do sieci neuronowej)
- Klasyfikacja wzorców:
  - Klasyfikacja do jednej z istniejących klas
  - Formowanie klas wzorców
- Asocjacyjne odtwarzanie wzorców
  - Odtwarzanie wzorców podobnych
  - Uzupełnianie wzorców
  - Odzyskiwanie (czyszczenie) wzorców

## RÓŻNE STRUKTURY SIECI

- Wielowarstwowe skierowane
- Wielowarstwowe rekurencyjne
- Samoorganizujące się

## SIECI REKURENCYJNE

