

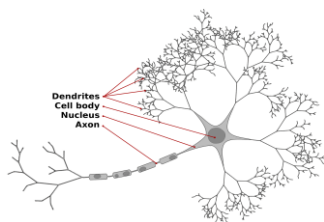
Inżynieria Mechatroniczna

Podstawy Sztucznej Inteligencji i Uczenia Głębokiego:

4: Sieci Neuronowe – na trzy sposoby

Ziemowit Dworakowski
AGH w Krakowie

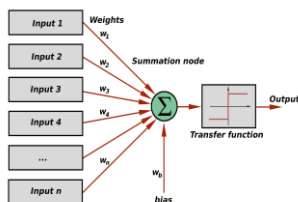
1



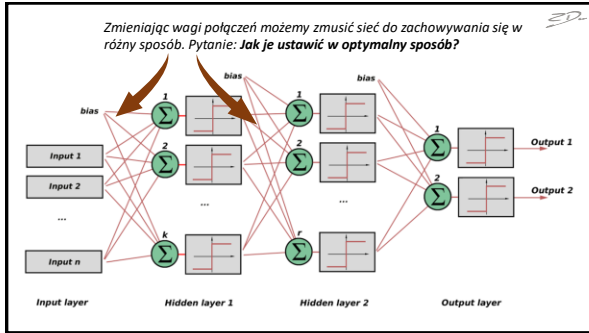
2

Zbudujmy model neuronu:

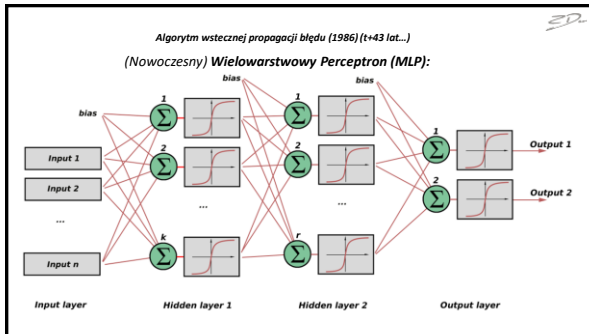
McCulloch – Pitts neuron (1943)



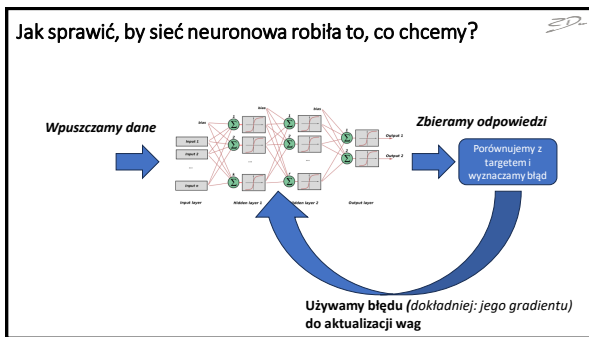
3



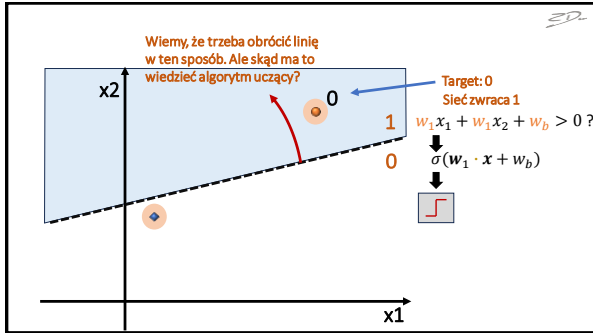
4



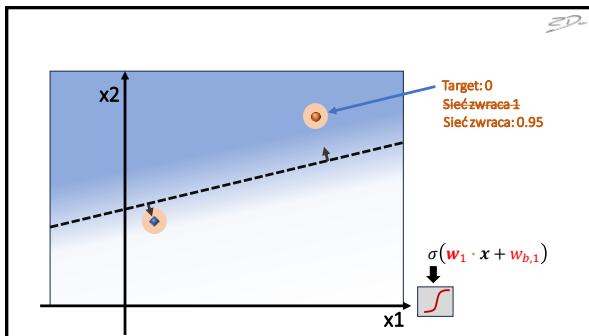
5



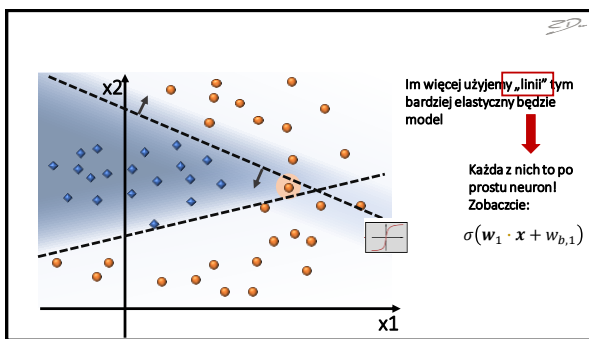
6



7



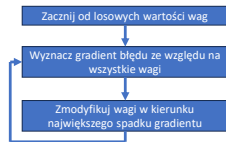
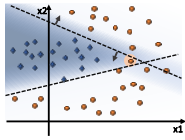
8



9

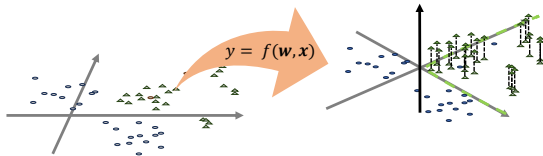
Algorytm propagacji wstecznej z zastosowaniem największego spadku gradientu

(podstawowy, bez dodatków)



10

Funkcja mapująca dane w inną przestrzeń



Będziemy potrzebować wielu stopni swobody, nieliniowości...
I możliwości modyfikowania zgodnie z zapotrzebowaniem...

11

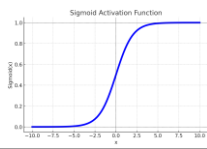
$$y = f(\mathbf{w}, \mathbf{x})$$

Pewnie warto zbudować taką funkcję z elementów składowych:

$$y = f_1(\mathbf{x}) + f_2(\mathbf{x}) + \dots + f_n(\mathbf{x})$$

Dobrze, by dało się zarządzać nimi za pomocą parametrów:

$$y = \sigma(\mathbf{w}_1, \mathbf{x}) + \sigma(\mathbf{w}_2, \mathbf{x}) + \dots + \sigma(\mathbf{w}_n, \mathbf{x})$$



$$\sigma(\mathbf{w}, \mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}\mathbf{x} + w_0)}}$$

12

SD

$$y = f(\mathbf{w}, \mathbf{x})$$

Pewnie warto zbudować taką funkcję z elementów składowych:

$$y = f_1(\mathbf{x}) + f_2(\mathbf{x}) + \dots + f_n(\mathbf{x})$$

Dobrze, by dało się zarządzać nimi za pomocą parametrów:

$$\begin{aligned} x'_1 &= \sigma(\mathbf{w}_{1,1}, \mathbf{x}) + \sigma(\mathbf{w}_{2,1}, \mathbf{x}) + \dots + \sigma(\mathbf{w}_{n,1}, \mathbf{x}) \\ x'_2 &= \sigma(\mathbf{w}_{1,2}, \mathbf{x}) + \sigma(\mathbf{w}_{2,2}, \mathbf{x}) + \dots + \sigma(\mathbf{w}_{n,2}, \mathbf{x}) \\ &\vdots \end{aligned}$$

W tym wektorze są nowe współrzędne!

I teraz „finalna decyzja” – kolejna nieliniowa transformacja:

$$y = \sigma(\mathbf{w}_{n+1,1}, \mathbf{x}')$$

Jeśli chcemy, możemy to zagnieździć ponownie (y staje się $\mathbf{x}''$, itd...)

13

SD

Sieci neuronowe na trzy sposoby:

Sieć neuronowa to:

- Złożona funkcja** w nieliniowy sposób mapująca dane z przestrzeni wejściowej w przestrzeń wyjściową, upraszczając etapowo ich strukturę
- Zestaw hiperpłaszczyzn** w przestrzeni cech tworzący złożone i dopasowywalne do danych powierzchnie decyzyjne
- Inspirowana biologicznie sieć węzłów** które potrafią przysyłać między sobą informacje, mająca emergentną umiejętność interpretacji danych

14

Rozważmy prosty wielowarstwowy perceptron (MLP):

Mamy wartość docelową t przypisaną do każdego wektora $\mathbf{x}$. Stąd, możemy wyznaczyć błąd d :

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

Teraz możemy obliczyć gradient d ze względu na wagi $\mathbf{w}$:

$$\nabla d = \left[\frac{\partial d}{\partial w_1}, \frac{\partial d}{\partial w_2}, \frac{\partial d}{\partial w_3}, \dots, \frac{\partial d}{\partial w_9} \right]$$

Gradient mówi nam, w którym kierunku zmieniać wagi

Możemy je połączyć by dostać postać ekwiwalent funkcji sieci $f(\mathbf{w}, \mathbf{x})$

$$y = \sigma(w_8 \cdot x'_1 + w_9 \cdot x'_2 + w_7)$$

$$x'_1 = \sigma(w_3 \cdot x_1 + w_5 \cdot x_2 + w_1)$$

$$x'_2 = \sigma(w_2 \cdot x_1 + w_6 \cdot x_2 + w_2)$$

15

Rozważmy prosty wielowarstwowy perceptron (MLP):

Mamy wartość docelową t przypisaną do każdego wektora x . Stąd, możemy wyznaczyć błąd d :

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

Teraz możemy obliczyć gradient d ze względu na wagi w :

$$\nabla d = \left[\frac{\partial d}{\partial w_1}, \frac{\partial d}{\partial w_2}, \frac{\partial d}{\partial w_3}, \dots, \frac{\partial d}{\partial w_9} \right]$$

Gradient mówi nam, w którym kierunku zmieniać wagi

$y = \sigma(w_8 \cdot x'_1 + w_9 \cdot x'_2 + w_7)$

$$\frac{\partial d}{\partial w_9} = \frac{\partial d}{\partial y} \frac{\partial y}{\partial p} = \frac{\partial d}{\partial y} \frac{\partial y}{\partial p} \frac{\partial p}{\partial w_9} = (y - t) \dots$$

$\frac{\partial d}{\partial y} = y - t$

16

$y = \sigma(p) = \frac{1}{1 + e^{-p}}$

$$\frac{\partial \sigma}{\partial p} = \frac{\partial}{\partial p} (1 + e^{-p})^{-1} = \cancel{-(1 + e^{-p})^{-2}} \cdot e^{-p} \cdot \cancel{1}$$

$$\frac{\partial \sigma}{\partial p} = \frac{1}{1 + e^{-p}} \cdot \frac{1}{1 + e^{-p}} \cdot e^{-p}$$

$$\frac{\partial \sigma}{\partial p} = \frac{e^{-p}}{1 + e^{-p}} \cdot \frac{1}{1 + e^{-p}}$$

$$\frac{\partial \sigma}{\partial p} = \frac{1 + e^{-p} - 1}{1 + e^{-p}} \cdot \frac{1}{1 + e^{-p}}$$

$$\frac{\partial \sigma}{\partial p} = \left(1 - \frac{1}{1 + e^{-p}} \right) \cdot \frac{1}{1 + e^{-p}} = (1 - y) \cdot y$$

17

Rozważmy prosty wielowarstwowy perceptron (MLP):

Mamy wartość docelową t przypisaną do każdego wektora x . Stąd, możemy wyznaczyć błąd d :

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

Teraz możemy obliczyć gradient d ze względu na wagi w :

$$\nabla d = \left[\frac{\partial d}{\partial w_1}, \frac{\partial d}{\partial w_2}, \frac{\partial d}{\partial w_3}, \dots, \frac{\partial d}{\partial w_9} \right]$$

Gradient mówi nam, w którym kierunku zmieniać wagi

$y = \sigma(w_8 \cdot x'_1 + w_9 \cdot x'_2 + w_7)$

$$\frac{\partial d}{\partial w_9} = \frac{\partial d}{\partial y} \frac{\partial y}{\partial p} = \frac{\partial d}{\partial y} \frac{\partial y}{\partial p} \frac{\partial p}{\partial w_9} = (y - t) y (1 - y) \dots$$

$y(1 - y) \leftarrow y = \sigma(p) = \frac{1}{1 + e^{-p}}$

18

Rozważmy prosty wielowarstwowy perceptron (MLP):

Mamy wartość docelową t przypisaną do każdego wektora x . Stąd, możemy wyznaczyć błąd d :

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

Teraz możemy obliczyć gradient d ze względu na wagi w :

$$\nabla d = \left[\frac{\partial d}{\partial w_1}, \frac{\partial d}{\partial w_2}, \frac{\partial d}{\partial w_3}, \dots, \frac{\partial d}{\partial w_9} \right]$$

Gradient mówi nam, w którym kierunku zmieniać wagi

$y = \sigma(w_8 \cdot x'_1 + w_9 \cdot x'_2 + w_7)$

$$\frac{\partial d}{\partial w_9} = \frac{\partial d}{\partial y} \frac{\partial y}{\partial p} = \frac{\partial d}{\partial y} \frac{\partial y}{\partial p} \frac{\partial p}{\partial w_9} = (y - t)y(1 - y) \cdot x'_2$$

$\frac{\partial p}{\partial w_9} = x'_2$

19

Rozważmy prosty wielowarstwowy perceptron (MLP):

Mamy wartość docelową t przypisaną do każdego wektora x . Stąd, możemy wyznaczyć błąd d :

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

Teraz możemy obliczyć gradient d ze względu na wagi w :

$$\nabla d = \left[\frac{\partial d}{\partial w_1}, \frac{\partial d}{\partial w_2}, \frac{\partial d}{\partial w_3}, \dots, \frac{\partial d}{\partial w_9} \right]$$

Gradient mówi nam, w którym kierunku zmieniać wagi

$y = \sigma(w_8 \cdot x'_1 + w_9 \cdot x'_2 + w_7)$

$x'_2 = \sigma(w_4 \cdot x_1 + w_6 \cdot x_2 + w_1)$

$$\frac{\partial d}{\partial w_9} = \frac{\partial d}{\partial y} \frac{\partial y}{\partial p} = \frac{\partial d}{\partial y} \frac{\partial y}{\partial p} \frac{\partial p}{\partial w_9} = (y - t)y(1 - y) \cdot x'_2$$

$$\frac{\partial d}{\partial w_6} = \frac{\partial d}{\partial y} \frac{\partial y}{\partial p} \frac{\partial p}{\partial x'_2} \frac{\partial x'_2}{\partial w_6} = (y - t)y(1 - y) \cdot x'_2 \cdot w_1$$

Poruszając się od wyjścia do wejścia, propagujemy obliczanie gradientu błędu – używając algorytmu wstecznej propagacji błędów

20

Zatrzymujemy trening jeśli:

- Błąd walidacji nie poprawia się przez dłuższy czas
- Skończy nam się czas
- Gradient spadnie (prawie) do zera

Mały pakiet (batch) danych

Odpowiadające etykiety

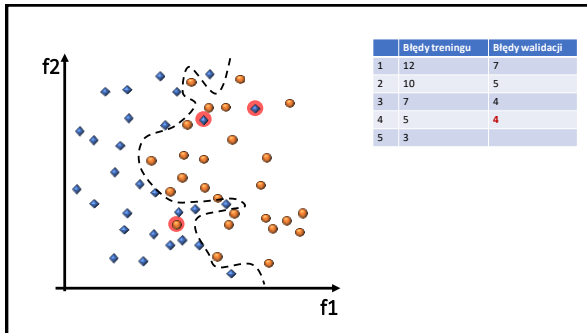
Błąd

Iteracje epoki

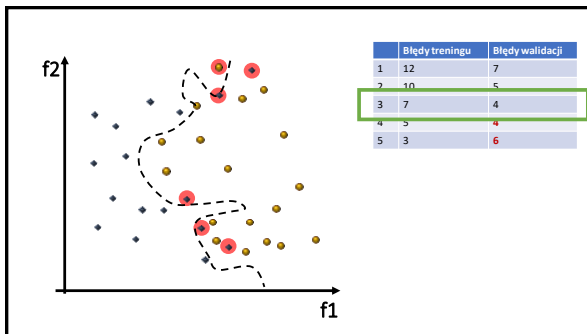
Powinniśmy zatrzymać trening tutaj

Epoka oznacza jedno przejście pełnego zbioru danych przez sieć

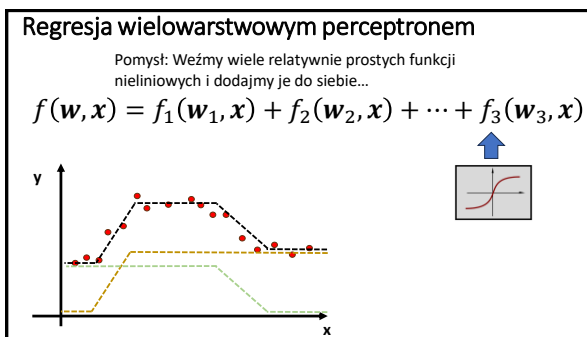
21



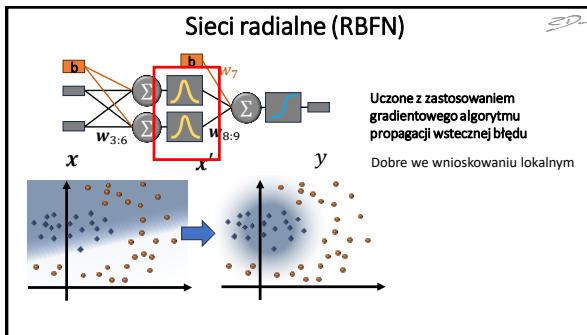
22



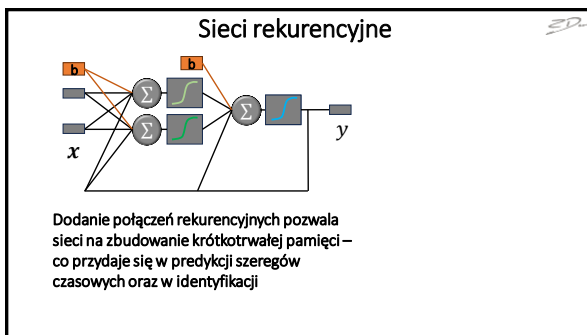
23



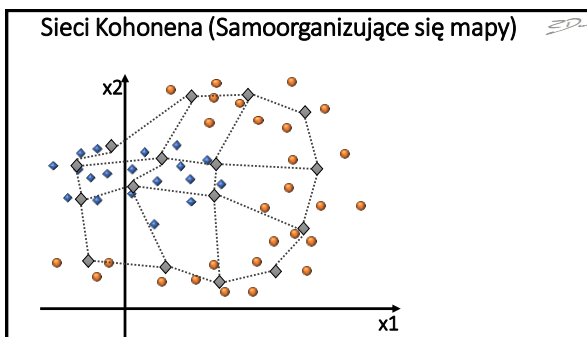
24



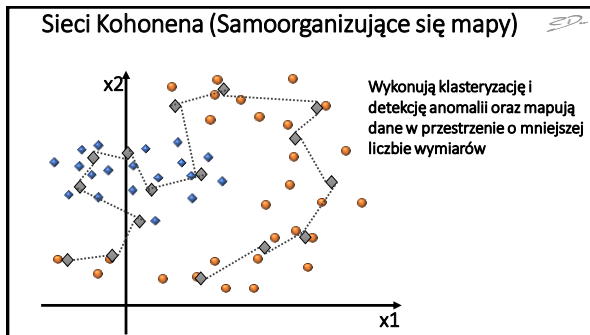
25



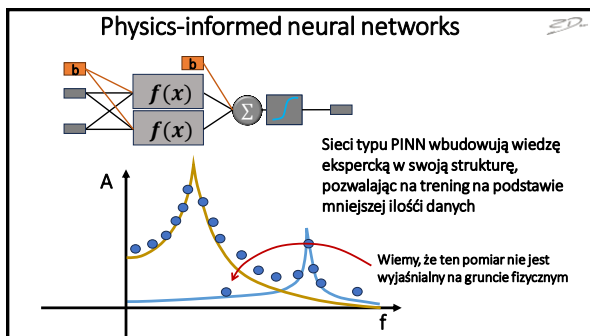
26



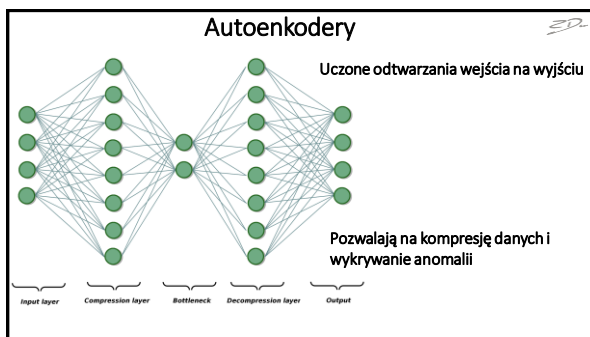
27



28



29



30

Materiał do powtórki:

SD

1. Wyjaśnij działanie sieci MLP na jeden z trzech sposobów
2. Narysuj schemat neuronu McCullocha-Pittsa i opisz jego elementy
3. Wyjaśnij dlaczego neuron McCullocha-Pittsa jest reprezentowany przez hiperpłaszczyznę w przestrzeni cech
4. Narysuj schemat sieci MLP i opisz jej elementy
5. Wyjaśnij zasadę działania algorytmu propagacji wstecznej błędów
(nie ma konieczności przedstawiania przykładowych obliczeń ani wzorów)
6. Napisz równanie sigmoidalnej funkcji aktywacji
7. Wyjaśnij podstawowy sposób ochrony przed przeuczeniem (jak w tym celu wykorzystać zbiór treningowy i walidacyjny?)
8. Wyjaśnij zasadę działania i cel stosowania sieci Kohonena, Radialnych, PINN, Autoenkoderów i Rekurencyjnych
