Wewnętrzne reprezentacje w warstwach ukrytych w skierowanych sieciach neuronowych

Jarosław Piersa
Wydział Matematyki i Informatyki, UMK

2013-04-11 — 13

Czym jest sieć neuronowa?

- macierze wag $W_1..W_l$, gdzie $W_1 \in \mathbb{R}^{n \times d_1}$, $W_2 \in \mathbb{R}^{d_1 \times d_2}$, ... , $W_l \in \mathbb{R}^{d_{l-1} \times d_l = 1}$
- wejścia: $\bar{x} = (x_1..x_n) \in \mathbb{R}^n$
- wynik działania na $\bar{x}$:

$$O^{(1)}_{W_1..W_l}(x) = \sigma(W_1^{tr} \cdot x) \in R^{d_1}$$

$$O^{(2)}_{W_1..W_l}(x) = \sigma(W_2^{tr} \cdot O^{(1)}(x)) = \sigma(W_2^{tr} \cdot \sigma(W_1^{tr} \cdot x))$$

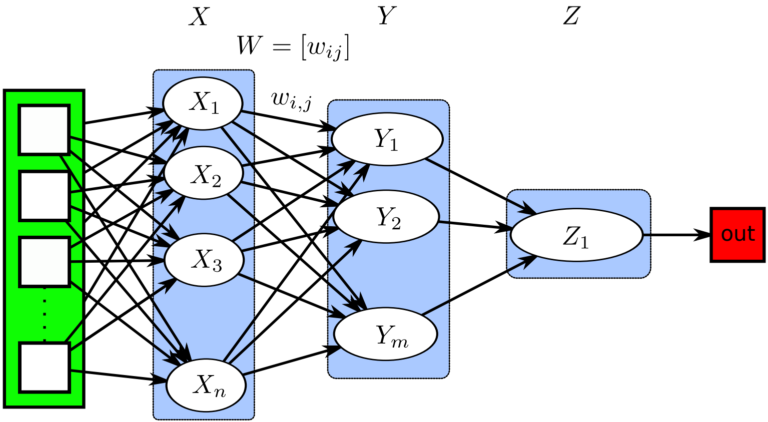
$$\ldots$$

$$O^{(l)}_{W_1..W_l}(x) = \sigma(W_l^{tr} \cdot O^{(l-1)}(x)) = \sigma(W_l^{tr} \cdot \sigma(...\sigma(W_1^{tr} \cdot x)...))$$

- gdzie σ jest operacją braną po współrzędnych:

$$\sigma(x) = \frac{1}{1 + exp(-x)}$$

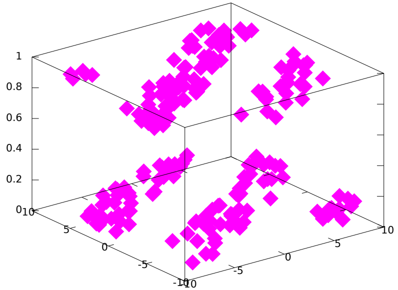
Czym jest sieć neuronowa?

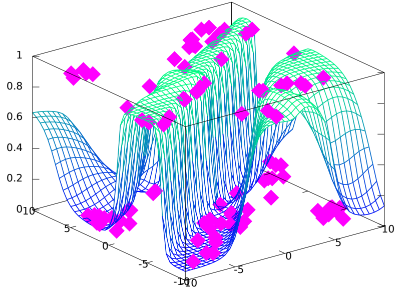


Uczenie sieci

- dane uczące: $(E^i, C^i), i = 1..p$, gdzie $E^{(i)} = (e^(i)_1 ... e^(i)_n$, $C^i \in (0, 1)$
- znaleźć $W_1..W_l$ (zwykle l=3), takie że $O^l_{W_1..W_l}(E^i) \simeq C^i$ dla $i = 1..p$

Uczenie Sieci





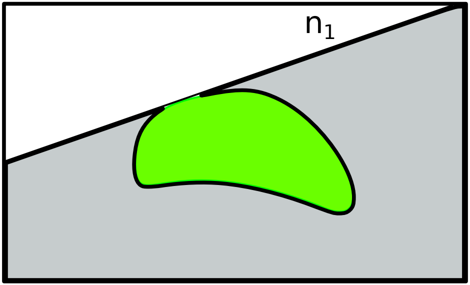
Uczenie sieci

Przyjmujemy uproszczenie:

- kategoryzacja binarna: $C^i \in \{0,1\}$
- $O'(\bar{x}) = \begin{cases} 1 & \text{jeżeli } O_{W_1..W_l}(\bar{x}) \geq \frac{1}{2} \\ 0 & \text{wpw} \end{cases}$
- Cel: znaleźć $W_1, .., W_l$

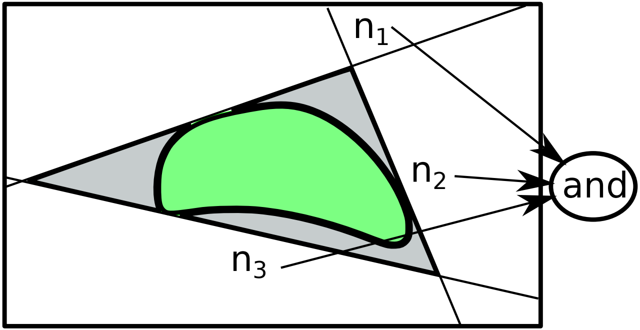
Co zrobiłby człowiek?

- $O_{n_1}(\bar{x}) = \frac{1}{1+\exp(-\sum_j w_{j,n_1} \cdot x_j)}$



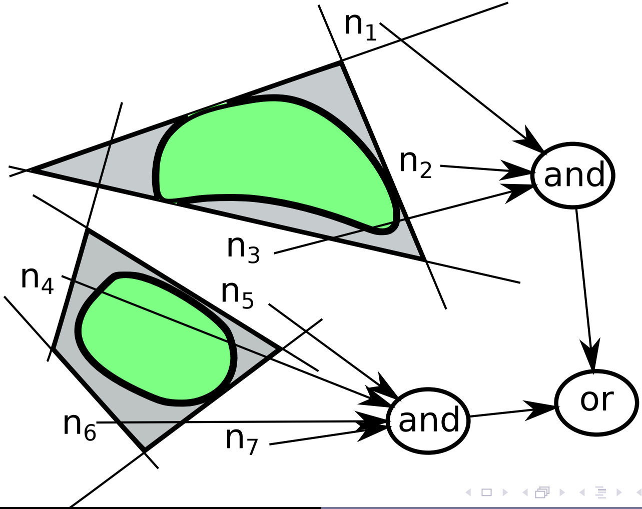
Co zrobiłby człowiek?

- $O_{AND}(\bar{x}) = \frac{1}{1+\exp(-\sum_j w_{n_j,and} \cdot O_{n_j}(\bar{x}))}$



Co zrobiłby człowiek?

- $O_{OR}(\bar{x}) = \frac{1}{1+\exp(-\sum_j w_{and_j,or} \cdot O_{and_j}(\bar{x}))}$



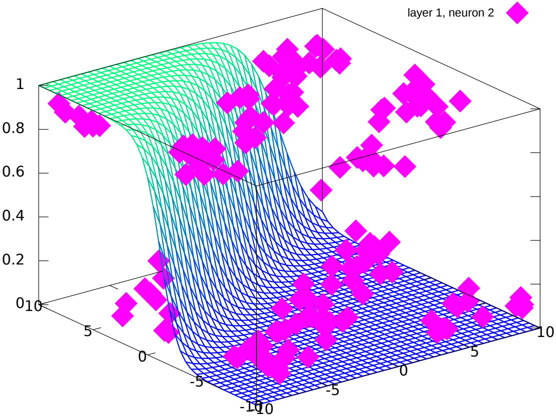
Jak uczy komputer?

Werbos (1985)

1. wybierz przykład E
2. Zapamiętaj wyniki działania na przykładzie E również w warstwach pośrednich o_j, sumy ważone in_j
3. Uaktualnij wagi w warstwie zewnętrznej (dla neuronu n_i):
   - Oblicz: $err_i = C_i - o_i$ oraz $\Delta_i = err_i \cdot \sigma'(in_i)$
   - Skoryguj wagi:
   $$w_{j,i} = w_{j,i} + \eta \cdot I_{j,i} \cdot \Delta_i$$
4. Uakktualnij wagi w pozostałych warstwach (dla neuronu n_j)
   - Oblicz błąd $err_j = \sum_l w_{j,l} \cdot \Delta_l$ oraz $\Delta_j = \sigma'(in_j) \cdot err_j$
   - Uaktualnij wagi:
   $$w_{k,j} = w_{k,j} + \eta \cdot I_{k,j} \cdot \Delta_j$$

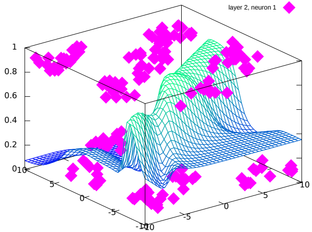
Warstwa 1

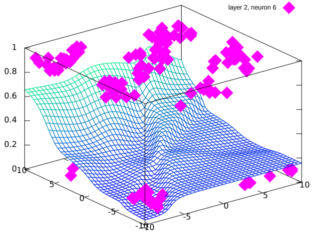
- $O(x, y) = \sigma(w_1 x_1 + w_2 x_2 + w_0) = \frac{1}{1+exp(\sum_i w_i x_i)}$

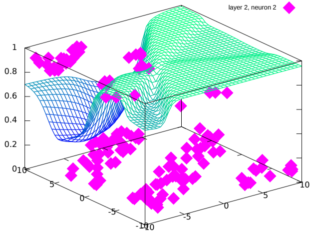


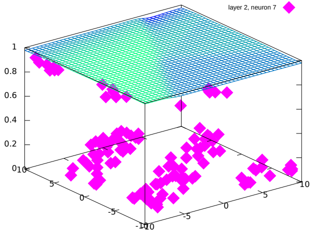
Warstwa 2

- $O_j(x, y) = \sigma(\sum_i w_{ij} \cdot O_{W_1}(\bar{x}))$



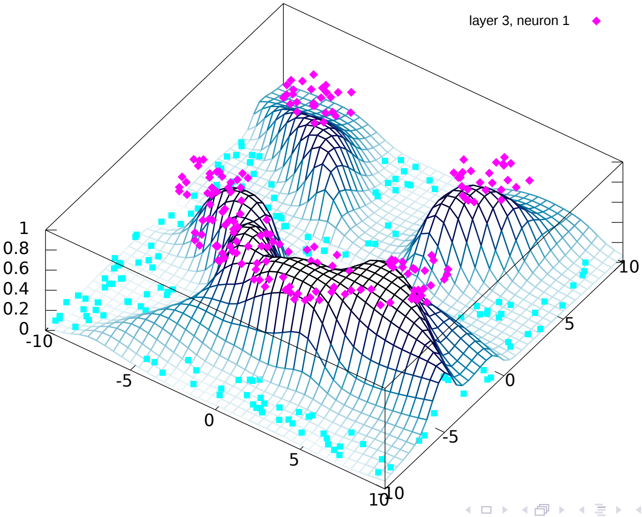




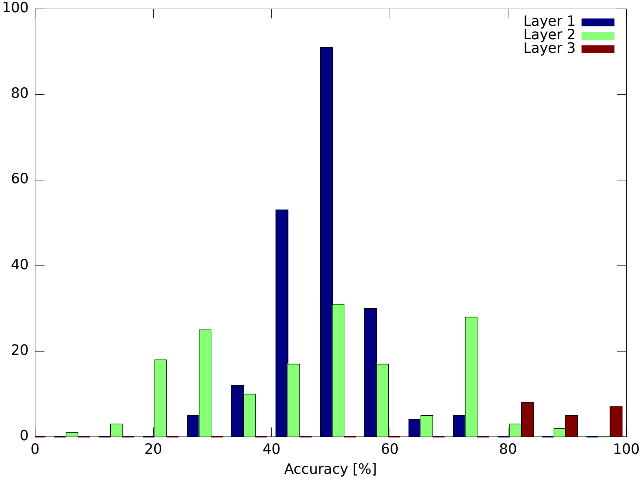


Reprezentacje

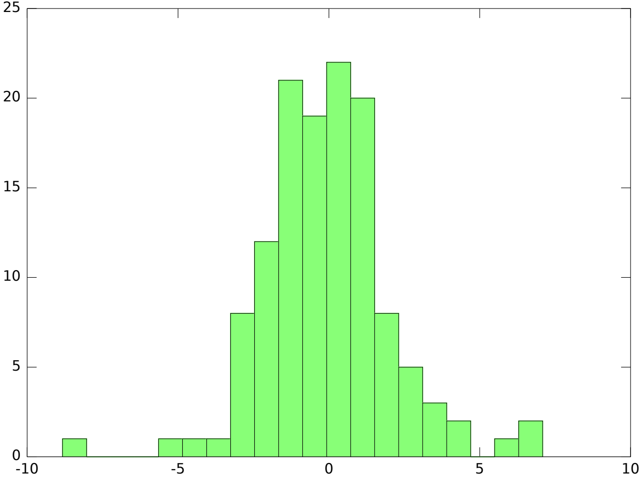
YouTube



Poprawność klasyfikacji w warstwach ukrytych



Wagi wychodzące z warstwy

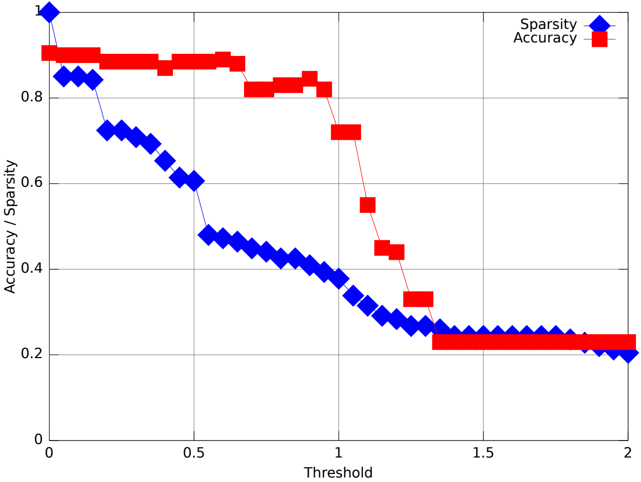


Czy da się zmniejszyć ilość wag?

Wycinanie nieistotnych wag

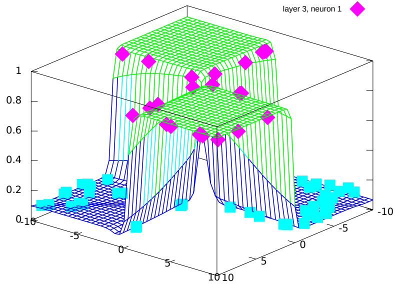
- wybierz parametr $\theta > 0$
- dla $i = 1..l$ wykonaj
  - jeżeli $|W_i[n, m]| < \theta$ to przypisz $W_i[n, m] := 0$
- dodatkowo dla warstwy wyjściowej l
- jeżeli $|W_l[m]| < \theta$ to przypisz $W_{l-1}[1..d_{l-1}, m] := 0$

Wynik

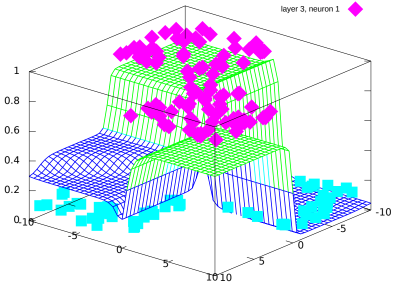

Sparsity
Accuracy
Accuracy / Sparsity
Threshold
0
0.2
0.4
0.6
0.8
1
0
0.5
1
1.5
2

Przypadek ekstremalny

- poprawność 99.5%
- pełne tablice wag (ok 150)



- poprawność 96%
- niezerowych wag 39%



Przypadek ekstremalny

- sieci szerokie, a płytkie
- analiza redundancji wag