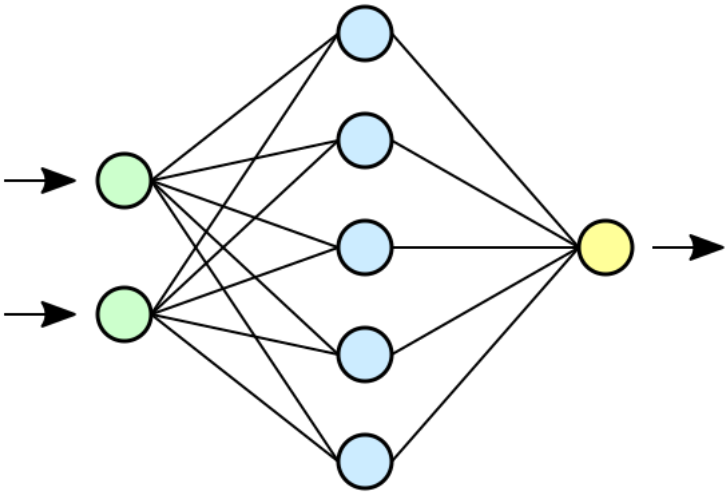
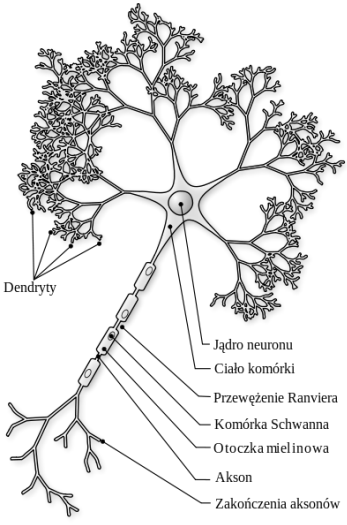
Sieci neuronowe

Michał Błauciak

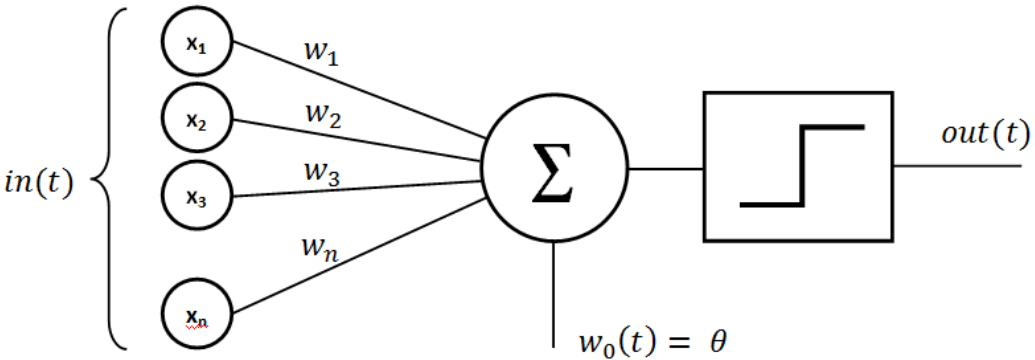
Ogólny schemat



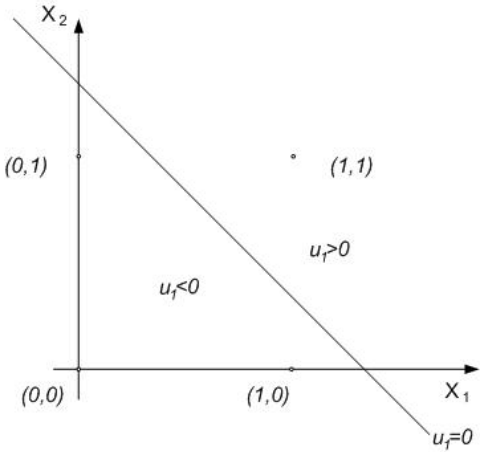
Neuron


Dendryty
Jądro neuronu
Ciało komórki
Przewężenie Ranviera
Komórka Schwanna
Otoczka mielinowa
Akson
Zakończenia aksonów

Perceptron



Problem XOR



Problem XOR

Wejścia: x_1 i x_2

Neuron 1: $y_1 = 2x_1 - 2x_2$

Funkcja nieliniowa po neuronie 1: $z_1 = 1$ jeśli $y_1 > 1$, w przeciwnym wypadku $z_1 = 0$

Neuron 2: $y_2 = 2x_2 - 2x_1$

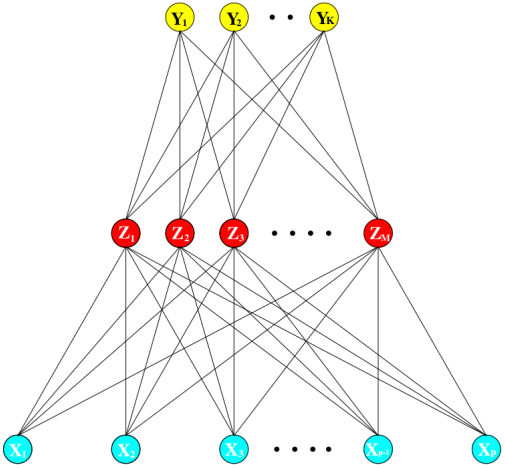
Funkcja nieliniowa po neuronie 2: $z_2 = 1$ jeśli $y_2 > 1$, w przeciwnym wypadku $z_2 = 0$

Neuron 3: $w = 2z_1 + 2z_2$

Funkcja nieliniowa po neuronie 3: $v = 1$ jeśli $w > 1$, w przeciwnym wypadku $v = 0$

Wejścia		**Wyjścia warstwy 1**		**Po funkcji nieliniowej**		**Wyjścia warstwy 2**	**Po funkcji nieliniowej (ostateczny wynik)**
x_1	x_2	y_1	y_2	z_1	z_2	w	v
0	0	0	0	0	0	0	0
0	1	−2	2	0	1	2	1
1	0	2	−2	1	0	2	1
1	1	0	0	0	0	0	0

Model z jedną warstwą ukrytą

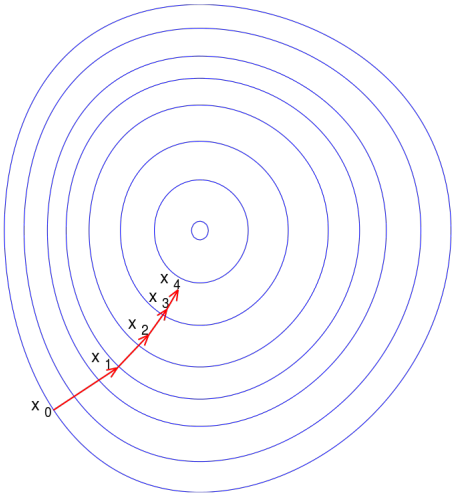


$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X),\ m = 1, \ldots, M,$$
$$T_k = \beta_{0k} + \beta_k^T Z,\ k = 1, \ldots, K,$$
$$f_k(X) = g_k(T),\ k = 1, \ldots, K,$$

$$Z = (Z_1, Z_2, \ldots, Z_M),$$
$$T = (T_1, T_2, \ldots, T_K)$$

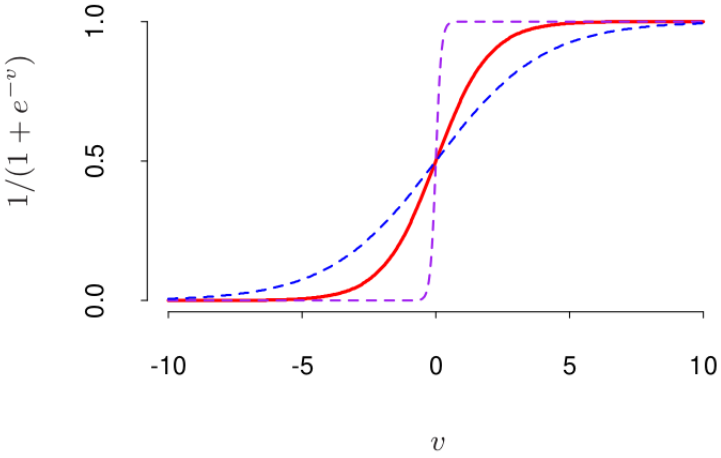
$$g_k(T) = \frac{e^{T_k}}{\sum_{\ell=1}^{K} e^{T_\ell}}$$

Metoda gradientu



1. Wybierz punkt startowy $\mathbf{x_0}$.
2. $\mathbf{x_{k+1}} = \mathbf{x_k} - \alpha_k \nabla f(\mathbf{x_k})$.
3. Sprawdź kryterium stopu, jeśli jest spełniony to STOP.
4. Jeżeli $f(\mathbf{x_{k+1}}) \geqslant f(\mathbf{x_k})$ to zmniejsz wartość α_k i powtórz punkt 2 dla kroku k-tego.
5. Powtórz punkt 2 dla następnego kroku $(k+1)$.

Funkcja aktywacji



Funkcja straty

$$\{\alpha_{0m}, \alpha_m;\ m = 1, 2, \ldots, M\}\ \ M(p+1) \text{ weights},$$
$$\{\beta_{0k}, \beta_k;\ k = 1, 2, \ldots, K\}\ \ K(M+1) \text{ weights}.$$

$$R(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{N} (y_{ik} - f_k(x_i))^2$$

$$G(x) = \text{argmax}_k f_k(x)$$

Wyliczanie wag

$$z_{mi} = \sigma(\alpha_{0m} + \alpha_m^T x_i)$$

$$z_i = (z_{1i}, z_{2i}, \ldots, z_{Mi})$$

$$R(\theta) \equiv \sum_{i=1}^{N} R_i$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} (y_{ik} - f_k(x_i))^2$$

$$\frac{\partial R_i}{\partial \beta_{km}} = -2(y_{ik} - f_k(x_i)) g_k'(\beta_k^T z_i) z_{mi},$$

$$\frac{\partial R_i}{\partial \alpha_{m\ell}} = -\sum_{k=1}^{K} 2(y_{ik} - f_k(x_i)) g_k'(\beta_k^T z_i) \beta_{km} \sigma'(\alpha_m^T x_i) x_{i\ell}$$

$$\beta_{km}^{(r+1)} = \beta_{km}^{(r)} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \beta_{km}^{(r)}}$$

$$\alpha_{m\ell}^{(r+1)} = \alpha_{m\ell}^{(r)} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \alpha_{m\ell}^{(r)}}$$

$$\frac{\partial R_i}{\partial \beta_{km}} = \delta_{ki} z_{mi}$$

$$\frac{\partial R_i}{\partial \alpha_{m\ell}} = s_{mi} x_{i\ell}$$

$$s_{mi} = \sigma'(\alpha_m^T x_i) \sum_{k=1}^{K} \beta_{km} \delta_{ki}$$

Problemy

- *learning rate*
- wartości początkowe parametrów
- przeuczenie – *weight decay*: $R(\theta) + \lambda \left(\sum \beta_{km}^2 + \sum \alpha_{ml}^2 \right)$
- liczba warstw i neuronów

Problemy

- *learning rate*
- wartości początkowe parametrów
- przeuczenie – *weight decay*: $R(\theta) + \lambda \left(\sum \beta_{km}^2 + \sum \alpha_{ml}^2 \right)$
- liczba warstw i neuronów

Problemy

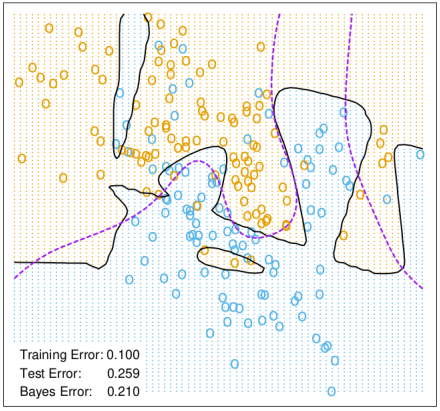
- *learning rate*
- wartości początkowe parametrów
- przeuczenie – *weight decay*: $R(\theta) + \lambda \left(\sum \beta_{km}^2 + \sum \alpha_{ml}^2 \right)$
- liczba warstw i neuronów

Problemy

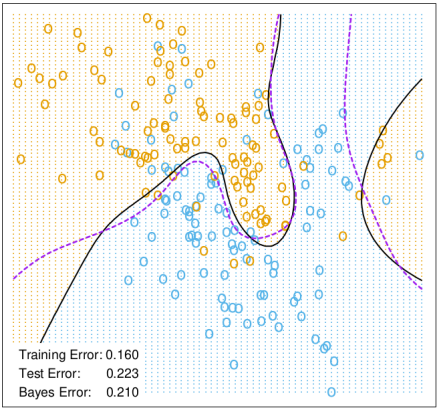
- *learning rate*
- wartości początkowe parametrów
- przeuczenie – *weight decay*: $R(\theta) + \lambda \left(\sum \beta_{km}^2 + \sum \alpha_{ml}^2 \right)$
- liczba warstw i neuronów

Regularyzacja

Neural Network - 10 Units, No Weight Decay



Neural Network - 10 Units, Weight Decay=0.02



Rozpoznawanie cyfr

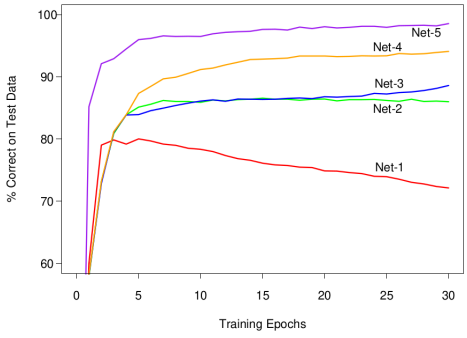
Net-1: No hidden layer, equivalent to multinomial logistic regression.

Net-2: One hidden layer, 12 hidden units fully connected.

Net-3: Two hidden layers locally connected.

Net-4: Two hidden layers, locally connected with weight sharing.

Net-5: Two hidden layers, locally connected, two levels of weight sharing.



Bibliografia

- T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*
- Wikipedia, *Sieć neuronowa*, *Perceptron*, *Neuron*, *Metoda gradientu prostego*
- Materiały agh.edu.pl, *XOR problem theory*