Podział danych na treningowe / testowe (/walidacyjne)

MNIST
50k/10k/10k

Irys: 150 : 120/30

- trenujemy sieć na danych treningowych
- sprawdzamy dokładność jej przewidywań na danych testowych (bez uczenia)

zwykle dokładność na danych testowych ≤ dokładności na danych treningowych

Jeśli różnica jest znaczna, to można, że doszło do przeuczenia sieci (overfitting)

Zwykle świadczy to o tym, że model jest zbyt skomplikowany w stosunku do ilości danych uczących.

Można też wprowadzić regularyzację, tzn. do funkcji kosztu dodać np. kwadrat normy L^2 wag (bez wyrazów wolnych)

$$\frac{\partial L}{\partial w} = (\text{jak bez regularyzacji}) + 2\alpha \cdot w$$

Uaktualnianie wag: $w* = (1-2\alpha)$; $w - = c \cdot \frac{\partial L}{\partial w}$

'40 pierwsze modele matematyczne neuronu (Aksamit?)

'50 pierwsza implementacja + sieć spiętrowa

'80 większa popularność, sieci splotowe

2000 –

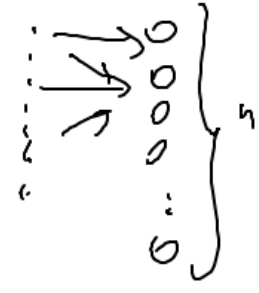
1989

Tw. (Cybenko)

σ – funkcja ciągła, sigmoidalna (tzn. $\sigma(-\infty)=0$, $\sigma(\infty)=1$, $\sigma \nearrow$).

Wtedy funkcja postaci

$$G(x)=\sum_{j=1}^{n} \alpha_j \sigma\left(y_j^T x + \theta_j\right) \qquad , x \in [0,1]^m$$



należy w $\left(C([0,1]^m), \|\cdot\|_\infty\right)$.

Regresja logistyczna

Klasyfikacja binarna $X \in \mathbb{R}^n \rightsquigarrow Y \in \{0,1\}$

klasa

model: $p(X) = P(Y=1|X)$

$\varphi : \mathbb{R} \to [0,1]$, $\varphi \nearrow$

$$p(X) = \varphi(\beta_0 + \beta_1 X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}} = \frac{e^{\beta_0+\beta_1 X}}{e^{\beta_0+\beta_1 X}+1}, \quad \beta_0 \in \mathbb{R},\ \beta_1 \in \mathbb{R}^n$$

φ ↗

$$\varphi(x) = \frac{1}{1+e^{-x}}$$

Dopasowanie współczynników: metoda maksymalizacji największej wiarygodności:

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(X_i) \cdot \prod_{i:y_i=0} (1-p(X_i))$$

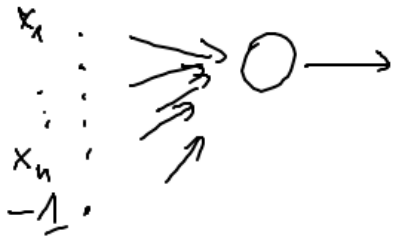
$$e^{\beta_0+\beta_1 X} = p(X)\left(e^{\beta_0+\beta_1 X}+1\right)$$

$$e^{\beta_0+\beta_1 X}\left(1-p(X)\right) = p(X)$$

$$e^{\beta_0+\beta_1 X} = \frac{p(X)}{1-p(X)}$$

$$\beta_0+\beta_1 X = \log \frac{p(X)}{1-p(X)}$$

To odpowiada sieci z jednym neuronem, z funkcją aktywacji $\varphi(x) = \frac{1}{1+e^{-x}}$



z funkcją kosztu $-\ell(\beta_0, \beta_1)$

Regresja logistyczna dla więcej niż dwóch klas:

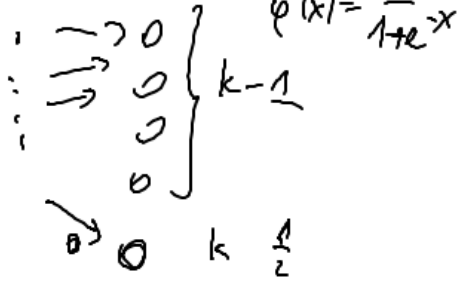
$$X \in \mathbb{R}^n \rightsquigarrow Y \in \{0, 1, \dots, k-1\} \text{ - klasy}$$

model: $\log \frac{P(Y=j|X)}{P(Y=k-1|X)} = \beta_0^{(j)} + \beta_1^{(j)} \cdot X \quad , j = 0, \dots, k-2$

$$\rightarrow P(Y=j|X) = \frac{e^{\beta_0^{(j)} + \beta_1^{(j)} \cdot X}}{1 + \sum_{i=0}^{k-2} e^{\beta_0^{(i)} + \beta_1^{(i)} X}} \quad , j = 0, \dots, k-2$$

$$P(Y=k-1|X) = \frac{1}{1 + \sum_{i=0}^{k-2} e^{\beta_0^{(i)} + \beta_1^{(i)} X}}$$

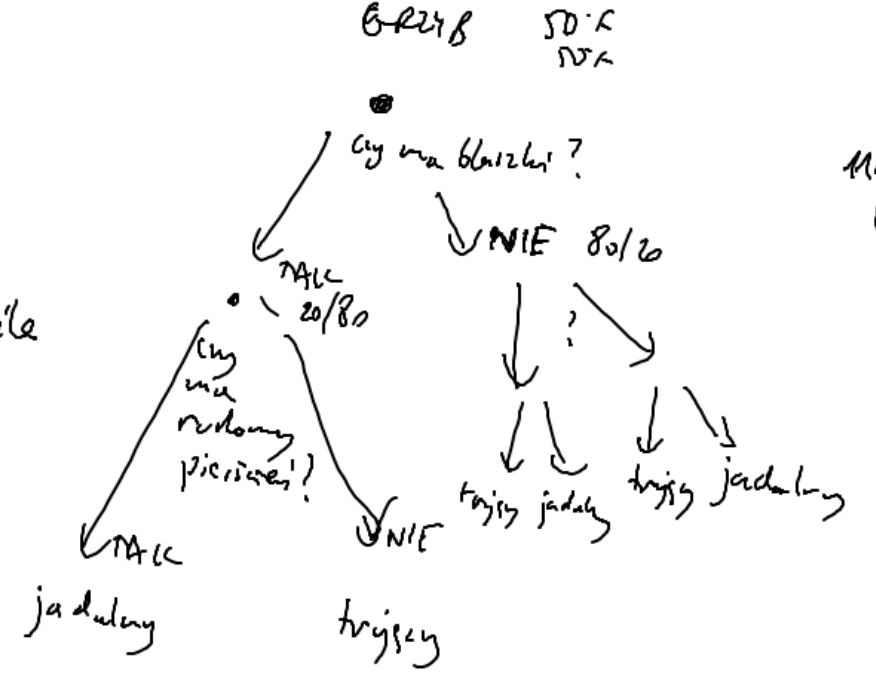
softmax:

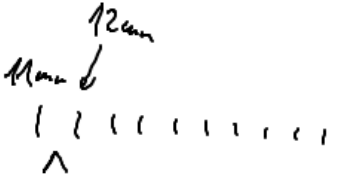


Drzewa decyzyjne

Drzewa decyzyjne zwykle budują się za pomocą algorytmu zachłannego:

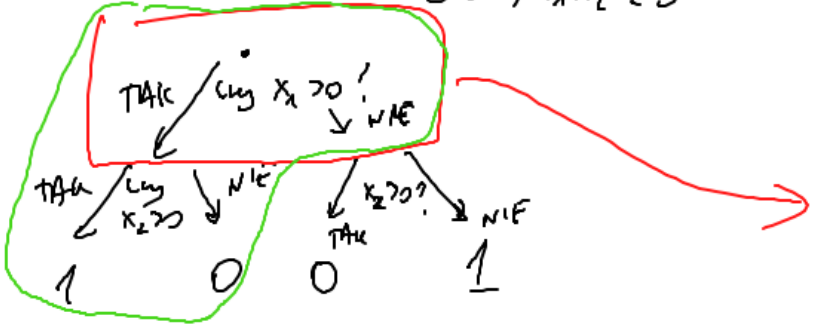
- Spośród możliwych pytań w danym węźle drzewa wybiera takie, które minimalizuje pewną funkcję kosztu.

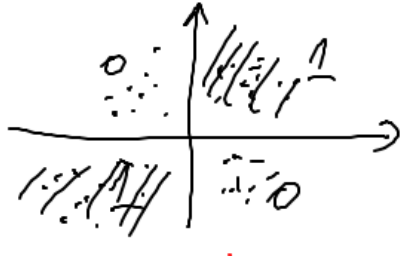


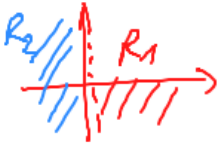


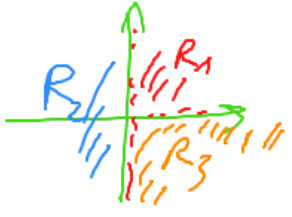
---

np. $X \in \mathbb{R}^2$, $y = \begin{cases} 1, & x_1 x_2 > 0 \\ 0, & x_1 x_2 \leq 0 \end{cases}$









$X = (x_1, \dots, x_p) \rightsquigarrow Y$ klasa lub $\rightsquigarrow Y \in \mathbb{R}$ dane: $X^{(1)}, X^{(2)}, \dots, X^{(m)}$, $Y^{(1)}, Y^{(2)}, \dots, Y^{(m)}$

algorytm:

1) Powiedzmy, że X został podzielony na pewną liczbę obszarów $R_1, \dots, R_J$ (prostokątów); początkowo $J = 1$, $R_1 = X$.

2) Rozważamy wszystkie R_j, $j = 1, \dots, J$. Niech $R = R_j$.

Rozważamy „wszystkie pytania, które można zadać", tzn. dla każdego $k = 1, \dots, p$

a) jeśli X_k ma wartości w $\mathbb{R}$, to rozważamy wszystkie $s \in R$ i dzielimy postaci

$$R_1(j,s) = R \cap \{x_j < s\}, \quad R_2(j,s) = R \cap \{x_j \geq s\}$$

b) jeśli X_k ma wartości w pewnym zbiorze skończonym, to rozważamy wszystkie jego podzbiory S i $R_1(j,s) = R \cap \{x_j \in S\}$, $R_2(j,s) = R \cap \{x_j \notin S\}$

3) wybieramy podzbiór $R : (j, s)$ tak, aby zminimalizować pewną funkcję kosztu.

Np. dla regresji:

$$\sum_{i : x^{(i)} \in R_1(j,s)} \left(y_i - \hat{y}_{R_1(j,s)}\right)^2 + \sum_{i : x^{(i)} \in R_2(j,s)} \left(y_i - \hat{y}_{R_2(j,s)}\right)^2 + \sum_{R \in \{R_1, \dots, R_J\} | R} \sum_{i : x^{(i)} \in R} (y_i - \hat{y}_R)^2$$