Spis treści

1 Model McCullocha-Pittsa

Jest to model matematyczny mający naśladować działanie fizjologicznych neuronów. Składa się on z n wejść u_i o wagach w_i i jednego wyjścia y. Neuron aktywuje się, gdy suma iloczynów wejść i wag jest większa od pewnej wartości progowej θ.

$$n_i, y \in \{0.0, 1.0\} \subset \mathbb{R}$$

$$w_i, \theta \in \mathbb{R}$$

$$f(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

$$y(\vec{u}, \vec{w}) = f((\sum_{i=1}^{n} w_i u_i) - \theta)$$

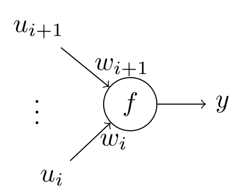

Diagram 1: Wizualizacja modelu McCullocha-Pittsa

1.1 Przykład

u_1	u_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 1: Tabela prawdy dla funkcji logicznej AND

- $u_1 = u_2 = 0 \to y = 0 = f(-\theta) \leftrightarrow \theta \geq 0$
- $u_1 = 0, u_2 = 1 \to y = 0 = f(w_2 - \theta) \leftrightarrow w_2 < \theta$
- $u_1 = 1, u_2 = 0 \to y = 0 = f(w_1 - \theta) \leftrightarrow w_1 < \theta$

- $u_1 = u_2 = 1 \rightarrow y = 1 = f(w_1 + w_2 - \theta) \leftrightarrow w_1 + w_2 \geq \theta$

$$\theta = 3, w_1 = 2, w_2 = 2$$

1.2 Reprezentacja wektorowa

$$\vec{u} = (u_1, u_2, \dots, u_n)$$
$$\vec{w} = (w_1, w_2, \dots, w_n)$$
$$y(\vec{u}, \vec{w}) = f(\vec{w} \cdot \vec{u} - \theta)$$

2 Liniowa separowalność

$$U_- = \{\vec{u_1}, \dots, \vec{u_n}\} \subset \mathbb{R}^n$$
$$U_+ = \{\vec{u_{n+1}}, \dots, \vec{u_{n+m}}\} \subset \mathbb{R}^n$$
$$U_- \cap U_+ = \emptyset$$

Mówimy, że zbiory wektorów (wejść) U_- i U_+ są liniowo separowalne, jeśli istnieje jakikolwiek $\vec{w}$ taki, że: $\vec{w} \cdot \vec{u} < 0 : \vec{u} \in U_-$ oraz $\vec{w} \cdot \vec{u} > 0 : \vec{u} \in U_+$. Innymi słowy jeśli istnieje hiperpłaszczyzna, która dzieli zbiory U_- i U_+.

2.1 Przykład

Dla bramki AND mamy:

$$U_- = \{(0,0), (0,1), (1,0)\}, U_+ = \{(1,1)\}$$

$$\vec{w} = (2,2), \theta = 3$$

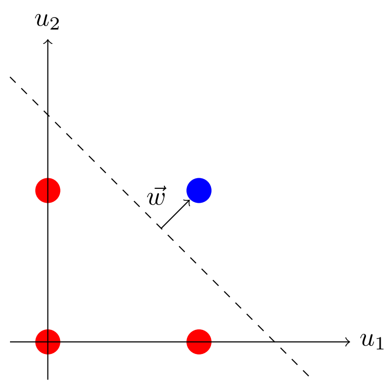

Diagram 2: Liniowa separowalność dla bramki AND

3 Associatron

Jest to model pamięci asocjacyjnej, skojarzeniowej, który pozwala na kojarzenie danych. W tym modelu dane reprezentujemy wektorami binarnymi. Jako, że de facto tworzymy macierz, to dowolne dane można asocjować.

3.1 Liniowy model pamięci

$$U = \{\vec{u_1}, \vec{u_2}, \dots, \vec{u_n}\} \in \mathbb{R}^n$$

$$\vec{u_t} = (u_{t1}, u_{t2}, \dots, u_{tn}), u_{ti} \in \{0, 1\}$$

3.2 Cel

Celem modelu jest stworzenie funkcji, która dla danego wektora $\vec{u_t}$ zwróci wektor $\vec{u_s}$, który jest najbardziej podobny do $\vec{u_t}$.

$$\varphi : U \to Y$$
$$U, Y \in \mathbb{R}^n$$

3.3 Macierz wag

$$\vec{y_t} = W \cdot \vec{u_t}$$
$$W = [w_{ij}] = \frac{1}{n} \sum_{t=1}^{N} \vec{y_t} \cdot \vec{u_t}^T$$

Macierz wag jest stała dla danego zbioru wektorów U.

3.4 Liniowa funkcja asocjacyjna

Zakładając, że wszystke wektory w U są ortogonalne ($\vec{u_t} \perp \vec{u_s} \leftrightarrow \vec{u_t} \cdot \vec{u_s} = 0$) to wówczas:

$$\varphi(\vec{u_i}) = W \cdot \vec{u_i} = \frac{1}{n} \sum_{\vec{u_t} \in U} \vec{y_t}(\vec{u_t} \cdot \vec{u_i}) = \frac{1}{n} \cdot \vec{y_i} \cdot \vec{u_i} \cdot \vec{u_i} = \frac{1}{n} \cdot \vec{y_i} \cdot n = \vec{y_i}$$

Czyli dla każdego wektora $\vec{u_i}$ zwracamy wektor $\vec{y_i}$. W istocie w powyższym równaniu szukamy takiego wektora w U, który po pomnożeniu przez wejście, nie będzie ortogonalny, czyli zwróci n.

$$\vec{u_t} \cdot \vec{u_t} = u_t^T u_t = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} = n$$

3.5 Nieliniowa funkcja asocjacyjna

Ograniczenie, że wektory w U są ortogonalne jest bardzo silne. W praktyce nie jesteśmy w stanie tego założyć. Zatem możemy wprowadzić funkcję φ', która pozwala na pewnego rodzaju błąd.

$$\varphi'(\vec{u_t}) = \begin{bmatrix} sgn(x_1) \\ sgn(x_2) \\ \vdots \\ sgn(x_n) \end{bmatrix}, \vec{x} = W \cdot \vec{u_t}$$

$$sgn(x) = \begin{cases} -1 & x < 0 \\ 1 & x \geq 0 \end{cases}$$