Perceptron

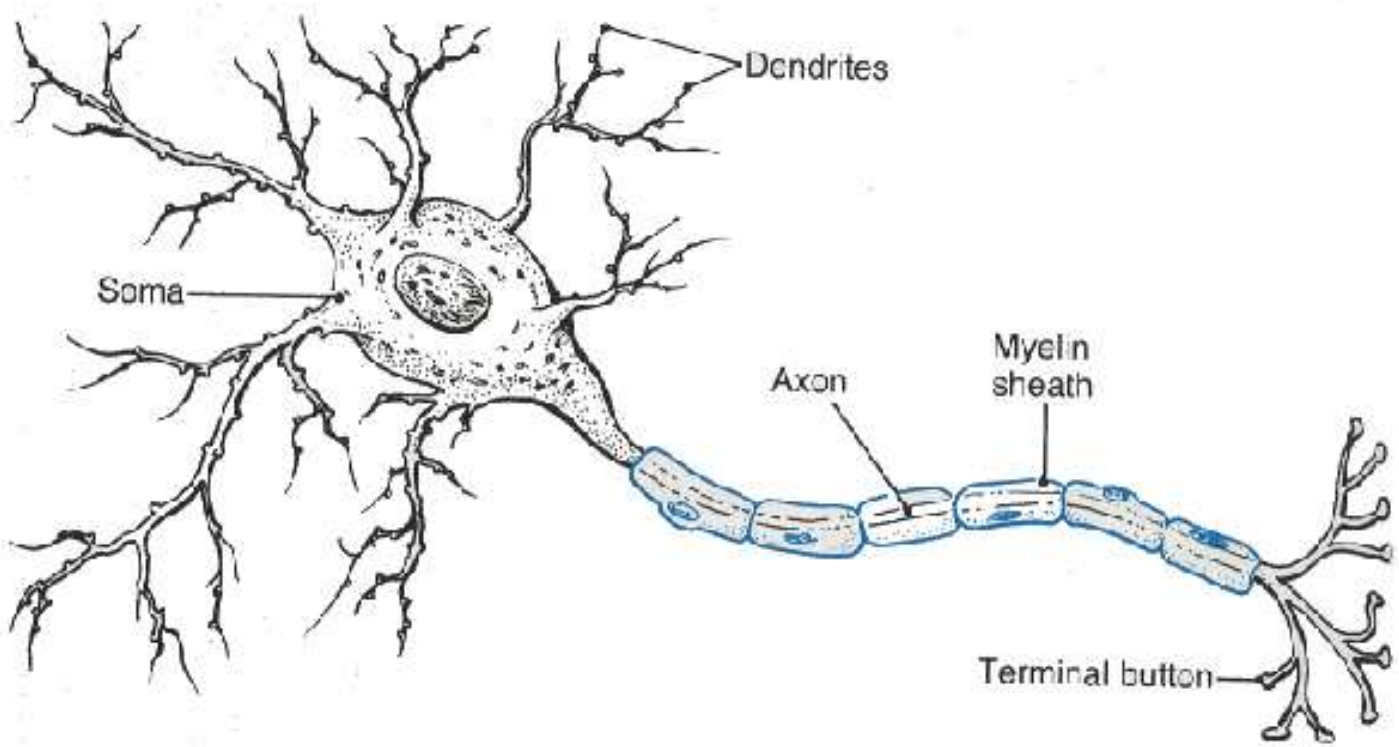
(c) Marcin Sydow

Zagadnienia:

- Neuron i jego własności
- Matematyczny model neuronu: Perceptron
- Perceptron jako klasyfikator
- Uczenie Perceptronu (Reguła Delta)

Neuron

System nerwowy stanowił naturalne źródło inspiracji dla sztucznej inteligencji. Stąd zainteresowanie *neuronem* – podstawową jednostką systemu nerwowego.



Własności Neuronu

- Transmituje sygnał będący funkcją sygnałów z "wejść", po jego przetworzeniu wewnątrz komórki, na "wyjście" podłączone do wejść innych neuronów
- Nieliniowe przetwarzanie sygnału: wyjście nie jest po prostu sumą sygnałów z wejść
- Dynamicznie modyfikuje czułość (wagi) połączeń z innymi neuronami (przez synapsy), dzięki czemu może wzmacniać lub osłabiać połączenia z innymi neuronami, w zależności od wykonywanego zadania

Perceptron: sztuczny neuron

Perceptron jest prostym matematycznym modelem neuronu.

Historycznie, celem prac nad sztucznymi sieciami neuronowymi było osiągnięcie umiejętności uczenia się i uogólniania typowych dla ludzkiego mózgu

Obecnie sztuczne sieci neuronowe skutecznie rozwiązują po prostu pewne konkretne zadania obliczeniowe

Pojedyczny perceptron może służyć jako klasyfikator lub regresor

Perceptron jest też punktem wyjściowym do budowania bardziej złożonych modeli sztucznych sieci neuronowych zdolnych do rozwiązywania trudnych problemów:

- uczenie z nadzorem lub bez
- sterowanie złożonymi urządzeniami (np. w robotyce)

Perceptron - prosty model naturalnego neuronu

Perceptron składa się z:

- n wejść $x_1, \ldots, x_n$ (nawiązują do dendrytów)
- n wag $w_1, \ldots, w_n$ (nawiązują do synaps)
  Każda waga w_i jest związana z $i - tym$ wejściem x_i
- progu Θ
- wyjścia y

(Wszystkie powyższe zmienne są typu rzeczywistego)

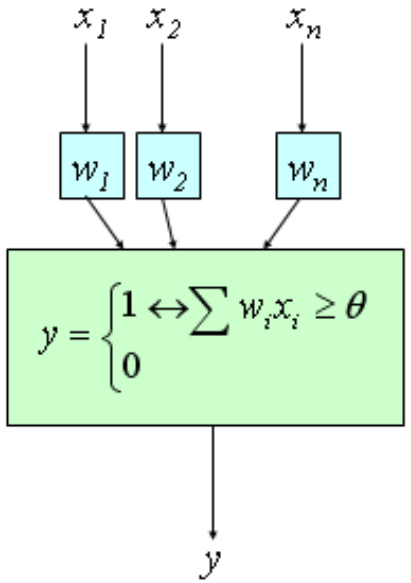
Wyjście y jest obliczane następująco:

$$y = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i \cdot x_i = W^T X \geq \Theta \quad \text{(perceptron "aktywny")} \\ 0 & \text{else} \quad \text{("nieaktywny")} \end{cases}$$

$W, X \in R^n$ oznaczają wektory wag i wejść

Obliczanie wyjścia perceptronu

Perceptron jest "aktywowany" (y=1) tylko jeśli iloczyn skalarny $W^T X$ (wartość ta jest często nazywana "net") przekracza wartość progu Θ



Nazywamy perceptron **dyskretnym** jeśli $y \in \{0, 1\}$ (lub $\{-1, 1\}$)
ciągłym jeśli $y \in [0, 1]$ (or $[-1, 1]$)

Perceptron: interpretatacja geometryczna

Obliczanie wyjścia perceptronu ma prostą interpretację geometryczną

Rozważmy n-wymiarową przestrzeń wektorów wejściowych (każdy jej element jest potencjalnym wektorem wejściowym $X \in R^n$).

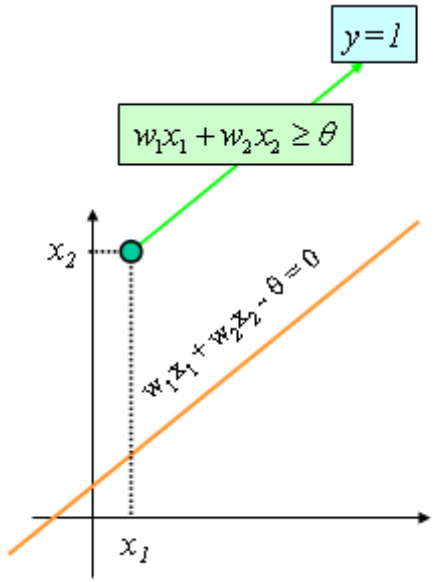
Wektor wag $W \in R^n$ jest **wektorem normalnym hiperpłaszczyźnie decyzyjnej**, która rozgranicza wektory na "aktywujące" i "nieaktywujące"

Perceptron aktywuje się (y=1) tylko jeśli wektor wejść X jest po tej samej stronie hiperpłaszczyzny decyzyjnej co wektor wag W

Osiągana jest tym większa wartość "net" ($W^T X$) im bliżej wektor X jest wektora W, jest zerowa gdy są prostopadłe i ujemna gdy są po przeciwnych stronach.

Perceptron: interpretatacja geometryczna

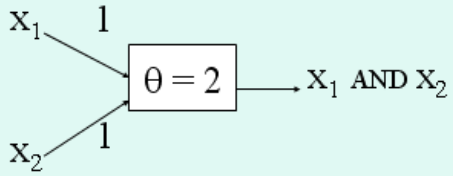
Wymagane działanie perceptronu jest osiągane przez odpowiednie ustawienie wag i progu



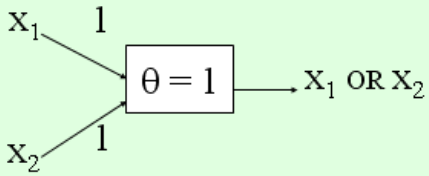
Wektor wag W wyznacza "kierunek" hiperpłaszczyzny decyzyjnej. Próg Θ wyznacza jak daleko ta hiperpłaszczyzna będzie odsunięta od początku układu współrzędnych

Przykład: Perceptron może modelować bramki logiczne

x_1	x_2	$x_1 \wedge x_2$
0	0	0
0	1	0
1	0	0
1	1	1



x_1	x_2	$x_1 \vee x_2$
0	0	0
0	1	1
1	0	1
1	1	1



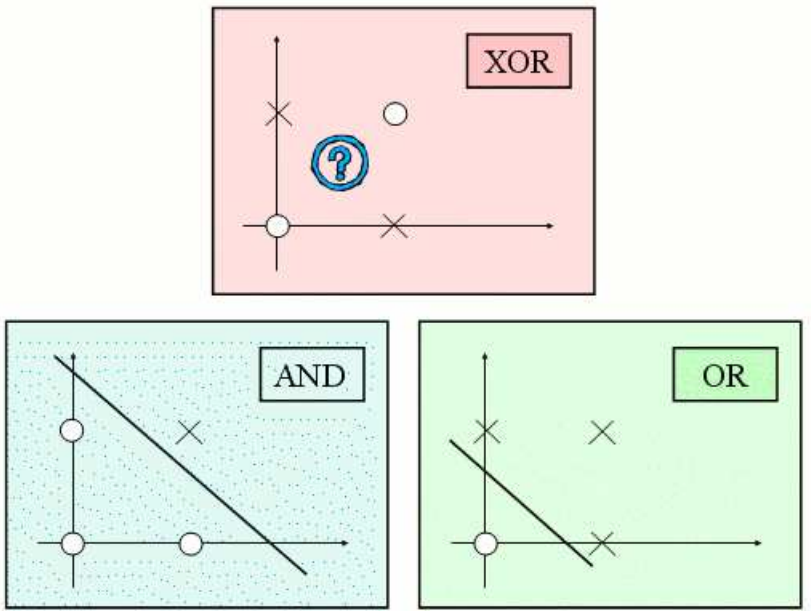
Ograniczenia pojedynczego perceptronu

Perceptron może służyć jako klasyfikator gdy mamy dokładnie 2 klasy.
Może jednak "rozróżniać" jedynie obszary, które są **liniowo separowalne**.

W szczególności, nie jest zdolny do modelowania np. funkcji logicznej XOR

Ograniczenia pojedynczego perceptronu

Funkcje logiczne AND, OR są liniowo separowalne. Funkcja XOR, nie jest.



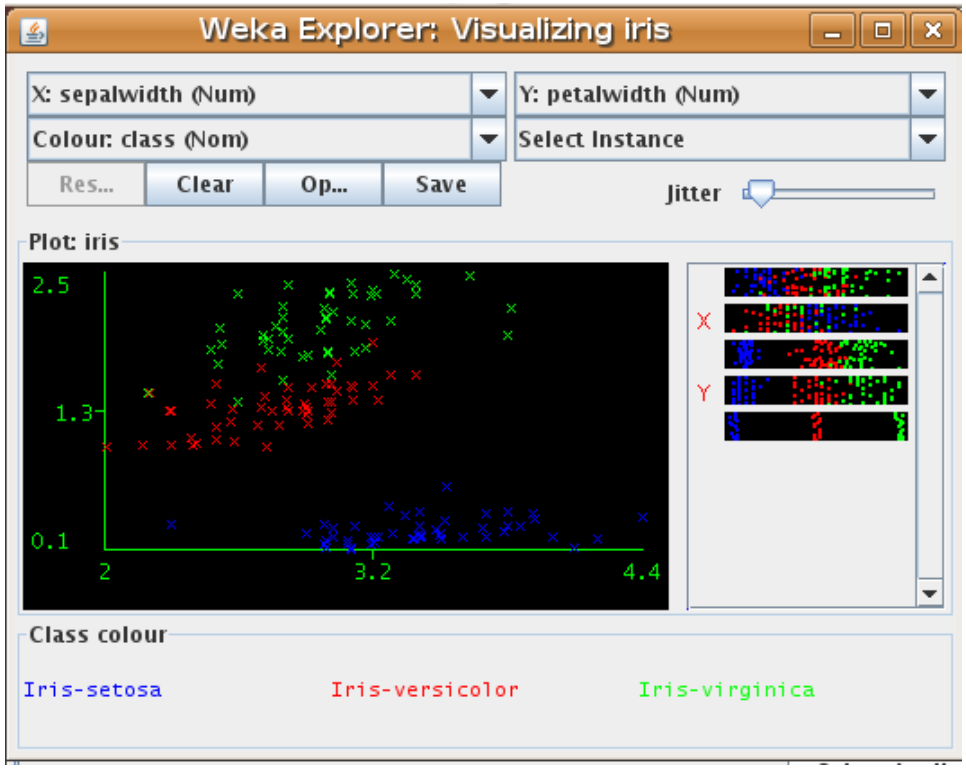
Sieci składające się z wielu neuronów

Neurony można łączyć (wyjścia do wejść). Można w ten sposób rozszerzyć możliwości obliczeniowe takiego modelu.

Na przykład, XOR może być reprezentowany już przez 2 połączone perceptrony

Przykład: IRIS

Z obrazka widać, że pojedynczy perceptron może np. odróżnić Iris-setosa od obu pozostałych podgatunków.



Może mieć jednak problemy z rozróżnieniem pozostałych 2 podgatunków.

Perceptron jako "uczący się" klasyfikator

Perceptron może być więc użyty jako klasyfikator do 2 klas (dla jednej klasy zwraca 1, dla drugiej 0)

Aby to osiągnąć perceptronowi podaje się wektory wejściowe ze zbioru treningowego wraz z prawidłowymi decyzjami (klasa 1 lub 0)

- wektor wejściowy
- prawidłowa odpowiedź (0 lub 1)

Co najważniejsze, istnieje prosty algorytm (Reguła Delta), która pozwala perceptronowi automatycznie "nauczyć się" odpowiednich wag i progu po obejrzeniu wystarczających przykładów trenujących

Reguła Delta uczenia Perceptronu

Podajemy perceptronowi przykłady uczące ze zbioru treningowego po kolei. Po każdym przykładzie, jeśli odpowiedź (wyjście perceptronu) jest niewłaściwa, modyfikujemy wagi w następujący sposób:

$$W' = W + (d - y)\alpha X$$

d - decyzja (wyjście) prawidłowa
y - faktyczna decyzja
$0 < \alpha < 1$ - parametr (stała uczenia) wpływający na intensywność uczenia

Interpretacja geometryczna reguły delta uczenia perceptronu

Docelowo, wektor wag powinien być możliwie "blisko" przykładów pozytywnych (y=1).

Stąd, wzór:

$$W' = W + (d - y)\alpha X$$

- "przyciąga" wektor W do "pozytywnych" przykładów X jeśli odpowiedź jest 0 zamiast 1 ("za słaba aktywacja")
- "odpycha"wektor W od "negatywnych" przykładów X jeśli odpowiedź jest 1 zamiast 0 ("zbyt mocna aktywacja")

Zwykle, "nauczenie" perceptronu wymaga wielokrotnego przejścia całego zbioru treningowego, zanim się nauczy odpowiednich wag i progu.

Rola progu

Jeśli próg Θ wynosi 0, perceptron może rozróżnić tylko wektory rozdzielone hiperpłaszczyzną przechodzącą przez początek układu współrzędnych.

Aby przesunąć hiperpłaszczyznę, próg musi mieć wartość niezerową z odpowiednim znakiem.

Włączenie progu do procesu uczenia

Rozważmy następującą sztuczkę notacyjną:
$W^T X \geq \Theta$

$W^T X - \Theta \geq 0$

Wektor, X może być rozszerzony o dodatkową "sztuczną" współrzędną -1 a wektor W może być rozszerzony o wartość progu Θ. Gdy oznaczymy przez W' oraz X' rozszerzone wektory (w przestrzeni n+1 wymiarowej), otrzymujemy:

$W'^T X' \geq 0$ - taka sama forma jak bez progu

W ten sposób reguła Delta może być zastosowana również do automatycznego uczenia progu a nie tylko wag.

Przykładowe Pytania

- neuron i jego ważne własności
- matematyczny model Perceptronu
- obliczanie wyjścia Perceptronu
- interpretacja geometryczna działania perceptronu
- perceptron jako klasfyikator
- matematyczne ograniczenia perceptronu
- reguła Delta uczenia pereptronu
- interpretacja geometryczna reguły Delta

Dziękuję za uwagę