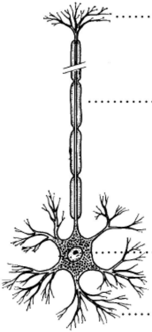
Co jsme probírali minule

- Umělé neuronové sítě - stručná historie
- Biologický neuron
- Strojové učení - základní pojmy



Strojové učení

Princip

- vychází z biologických principů, model se "vytvoří sám"
- výpočetní model se učí na základě dat (trénovací množina) nebo předchozích zkušeností

Metody učení

- **Učení s učitelem (supervised learning)**
  - trénovací množina ve tvaru [*vstup, požadovaný výstup*]
- **Učení bez učitele (unsupervised learning, samoorganizace)**
  - trénovací množina ve tvaru [*vstup*]
- **Zpětnovazebné učení (reinforcement learning)**
  - program se učí optimální strategii na základě předchozích zkušeností

Strojové učení

Úloha strojového učení s učitelem

- je dána množina trénovacích dat: vstupní a výstupní vzory (požadované výstupy)
- **aproximace** – chceme, aby model co nejlépe aproximoval neznámou funkci $\rightarrow$ aby pro každý předložený vstupní správně predikoval hodnotu výstupu
- **generalizace = zobecňování** – model by měl dát správný výstup i pro data, která nejsou v trénovací množině

Typy úloh strojového učení s učitelem

- **regrese** – predikujeme numerickou hodnotu / hodnoty
- **klasifikace** – predikujeme diskrétní hodnotu / hodnoty
- **učení strukturovaných dat**

Strojové učení

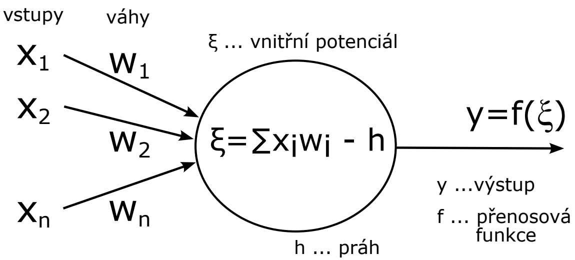
Typický průběh řešení úlohy strojového učení

- Předzpracování dat
  - převedení dat do formátu, se kterým se bude modelu strojového učení nejlépe pracovat
  - např. výběr příznaků
- Učení modelu
  - jaký typ modelu? (záleží na problému)
  - jaký model daného typu? (volba vhodných parametrů)
- Vyhodnocení modelu – nejlépe na nových datech

Dnešní hodina

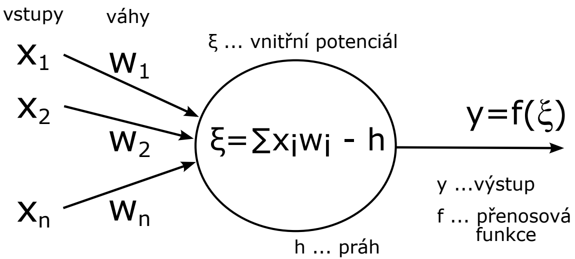
1. Matematický model neuronu a jeho geometrická interpretace
2. Nejstarší modely umělých neuronů: Perceptrons (Rosenblatt, 1955) , Culloch and Pitts neurons (1943)
3. Perceptron a reprezentace logických funkcí
4. Perceptron a lineární separabilita
5. asi až příště: Perceptron a algoritmy učení

Matematický model neuronu



- parametry neuronu:
  - vektor vah $\vec{w} = (w_1, ..., w_n) \in R^n$,
  - práh h
  - přenosová funkce $f : R \to R$
- neuron pro vstup $\vec{x} \in R^n$ spočte výstup $y \in R$ jako hodnotu přenosové funkce $f_{\vec{w},h}(\vec{x})$

Matematický model neuronu



Výpočet výstupu neuronu:

- vnitřní potenciál: $\xi = \sum_{i=1}^{n} w_i.x_i - h = \vec{w}\vec{x}^T - h$
- výstup: $y = f(\xi)$

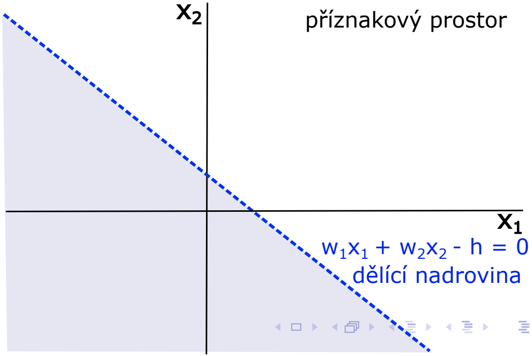
Matematický model neuronu

Geometrická interpretace

- vstupy neuronu si představme jako body v n-rozměrném Euklidovském prostoru (vstupní, příznakový prostor)
- položme vnitřní potenciál neuronu $\xi = 0$ a získáme rovnici dělící nadroviny

$$\xi = w_1 x_1 + w_2 x_2 - h = 0$$

$$x_2 = -\frac{w_1}{w_2} x_1 + \frac{h}{w_2}$$

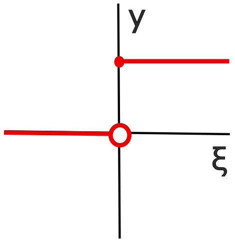


Matematický model neuronu

- používaná terminologie vychází z nejstarších modelů neuronu: Perceptron (Rosenblatt, 1955), Culloch and Pitts neurons (1943)
- oba modely používali skokovou přenosovou funkci

Skoková přenosová funkce:

- $f(\xi) = 1$ pro $\sum_{i=1}^{n} w_i.x_i \geq h$, tj.
  $\xi = \sum_{i=1}^{n} w_i.x_i - h \geq 0$
  ... neuron je **aktivní**
- $f(\xi) = 0$ pro $\sum_{i=1}^{n} w_i.x_i < h$, tj.
  $\xi = \sum_{i=1}^{n} w_i.x_i - h < 0$
  ... neuron je **pasivní**

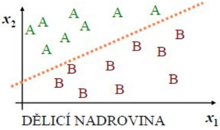


Perceptron (Rosenblatt, 1955) , Culloch and Pitts neurons (1943)

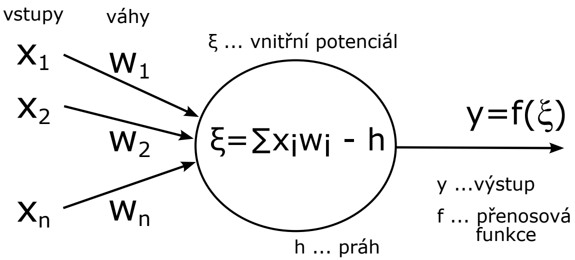
$\rightarrow$ perceptron může sloužit jako **lineární klasifikátor:** klasifikuje vzory do dvou tříd (zde A, B)

Skoková přenosová funkce:

- $f(\xi) = 1$ pro $\sum_{i=1}^{n} w_i.x_i \geq h$, tj. $\xi \geq 0$
  ... neuron je **aktivní** (třída A)
- $f(\xi) = 0$ pro $\sum_{i=1}^{n} w_i.x_i < h$, tj. $\xi < 0$
  ... neuron je **pasivní** (třída B)



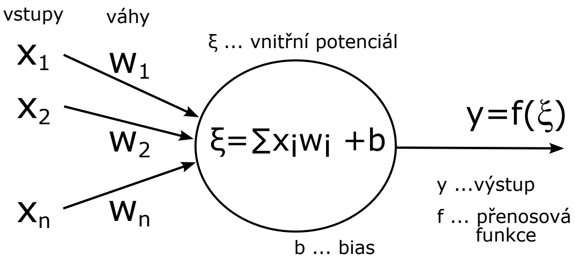
Matematický model neuronu



Klasická definice: práh h

- vnitřní potenciál: $\xi = \sum_{i=1}^{n} w_i . x_i - h = \vec{w}\vec{x}^T - h$
- výstup: $y = f(\xi)$

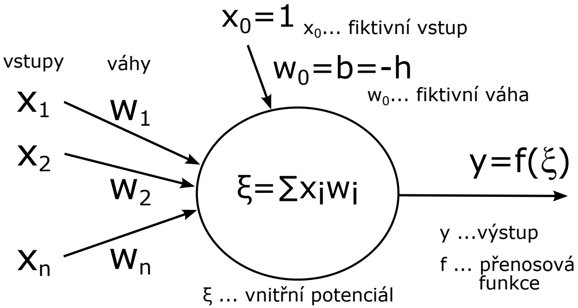
Matematický model neuronu



Alternativní definice: práh → bias

- vnitřní potenciál: $\xi = \sum_{i=1}^{n} w_i.x_i + b = \vec{w}\vec{x}^T + b$
- výstup: $y = f(\xi)$

Matematický model neuronu



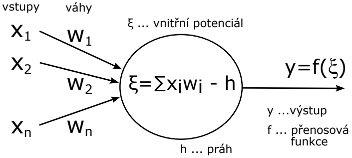
Alternativní definice: zavedení fiktivního vstupu

- rozšířený příznakový prostor ... $\vec{x} = (x_0 = 1, x_1, ..., x_n)$
- rozšířený prostor vah ... $\vec{w} = (w_0 = b = -h, w_1, ..., w_n)$
- vnitřní potenciál: $\xi = \sum_{i=0}^{n} w_i.x_i = \vec{w}\vec{x}^T$
- výstup: $y = f(\xi)$

Culloch and Pitts neurons (1943)

Binární varianta:

- binární vstupy: $x_i \in \{0, 1\}$
- binární výstupy: $y \in \{0, 1\}$
- váhy: $w_i \in \{-1, 1\}$
- skoková přenosová funkce



Bipolární varianta:

- bipolární vstupy: $x_i \in \{-1, 1\}$
- bipolární výstupy: $y \in \{-1, 1\}$
- váhy: $w_i \in \{-1, 1\}$
- skoková přenosová funkce

$\rightarrow$ využití: reprezentace logických funkcí (AND, OR, NOT,...) ... ukážeme si za chvilku

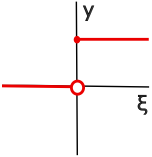
- **nevýhoda**: pro model neexistoval učící algoritmus (až později ... hebbovské učení)

Perceptrony (Rosenblatt, 1955)

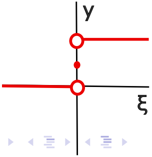
- reálné vstupy ... $x_i \in R$
- reálné váhy a prahy ... $w_i \in R$
- výstupy:
  - **binární** ... $y \in \{0, 1\}$
  - **bipolární** ... $y \in \{-1, 1\}$

Varianty skokové přenosové funkce pro binární perceptron:

$$f(\xi) = \begin{cases} 1 & \text{pro } \xi \geq 0 \quad \text{... neuron je aktivní} \\ 0 & \text{pro } \xi < 0 \quad \text{... neuron je pasivní} \end{cases}$$



$$f(\xi) = \begin{cases} 1 & \text{pro } \xi > 0 \quad \text{.. neuron je aktivní} \\ 0.5 & \text{pro } \xi = 0 \quad \text{... neuron je tichý} \\ 0 & \text{pro } \xi < 0 \quad \text{... neuron je pasivní} \end{cases}$$



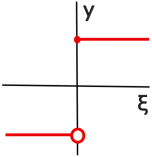
$\rightarrow$ funkce **signum** (bsign)

Perceptrony (Rosenblatt, 1955)

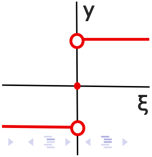
- reálné vstupy ... $x_i \in R$
- reálné váhy ... $w_i \in R$ a prahy
- výstupy:
  - **binární** ... $y \in \{0, 1\}$
  - **bipolární** ... $y \in \{-1, 1\}$

Varianty skokové přenosové funkce pro bipolární perceptron:

$$f(\xi) = \begin{cases} 1 & \text{pro } \xi \geq 0 \quad \text{... neuron je aktivní} \\ -1 & \text{pro } \xi < 0 \quad \text{... neuron je pasivní} \end{cases}$$



$$f(\xi) = \begin{cases} 1 & \text{pro } \xi > 0 \quad \text{... neuron je aktivní} \\ 0 & \text{pro } \xi = 0 \quad \text{... neuron je tichý} \\ -1 & \text{pro } \xi < 0 \quad \text{... neuron je pasivní} \end{cases}$$



$\rightarrow$ funkce **signum** (sign)

Logický prahový obvod

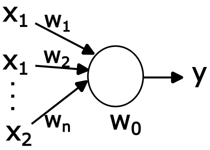
Pomocí perceptronu lze realizovat základní logické funkce

- NOT (negace)
- ID (identita)
- AND (konjunkce)
- OR (disjunkce)

$\rightarrow$ z perceptronů můžeme složit **logický prahový obvod**
... reprezentace libovolných logických (booleovských) funkcí

Logický prahový obvod

Budeme uvažovat následující model perceptronu:



$$\xi = \vec{w} \cdot \vec{x}^T = \sum_{i=0}^{n} w_i x_i$$

$$= w_0 + \sum_{i=1}^{n} w_i x_i$$

Binární perceptron

- vstupy: $x_i \in \{0, 1\}$
- výstupy: $y \in \{0, 0.5, 1\}$
- $y = f(\xi) = bsign(\xi)$

$$bsign(\xi) = \begin{cases} 1 & \text{pro } \xi > 0 \\ 0.5 & \text{pro } \xi = 0 \\ 0 & \text{pro } \xi < 0 \end{cases}$$

Bipolární perceptron

- vstupy: $x_i \in \{-1, +1\}$
- výstupy: $y \in \{-1, 0, +1\}$
- $y = f(\xi) = sign(\xi)$

$$sign(\xi) = \begin{cases} 1 & \text{pro } \xi > 0 \\ 0 & \text{pro } \xi = 0 \\ -1 & \text{pro } \xi < 0 \end{cases}$$

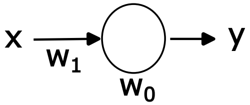
Logický prahový obvod - NOT (negace)

bipolární model

x	$y = \neg x$
-1	1
1	-1

binární model

x	$y = \neg x$
0	1
1	0



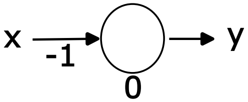
$$y = sign(w_0 + w_1 x)$$

- jak zvolíme w_0 a w_1 v případě bipolárního modelu?
- a jak v případě binárního modelu?

Logický prahový obvod - NOT (negace)

bipolární model

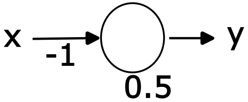
x	$y = \neg x$
-1	1
1	-1



$$y = sign(w_0 + w_1 x) = sign(-x)$$

binární model

x	$y = \neg x$
0	1
1	0

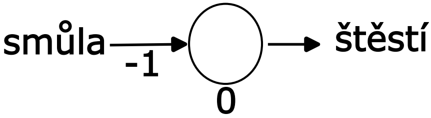


$$y = bsign(w_0 + w_1 x) = bsign(0.5 - x)$$

Otázka: Napadly by vás i další řešení?

Logický prahový obvod - NOT (negace)

Příklad: štěstí $= \neg$ smůla



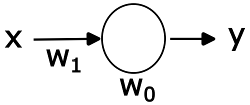
Logický prahový obvod - ID (identita)

bipolární model

x	y = x
-1	-1
1	1

binární model

x	y = x
0	0
1	1



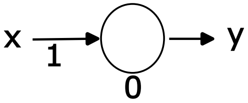
$$y = sign(w_0 + w_1 x)$$

- jak zvolíme w_0 a w_1 v případě bipolárního modelu?
- a jak v případě binárního modelu?

Logický prahový obvod - ID (identita)

bipolární model

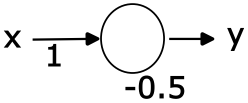
x	y = x
-1	-1
1	1



$$y = sign(w_0 + w_1 x) = sign(x)$$

binární model

x	y = x
0	0
1	1



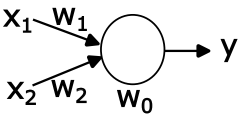
$$y = bsign(w_0 + w_1 x) = bsign(-0.5 + x)$$

→ není co řešit

Logický prahový obvod - AND (konjunkce)

bipolární model

x_1	x_2	$y = x_1 \wedge x_2$
-1	-1	-1
-1	+1	-1
+1	-1	-1
+1	+1	+1



binární model

x_1	x_2	$y = x_1 \wedge x_2$
0	0	0
0	1	0
1	0	0
1	1	1

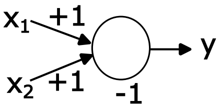
$$y = sign(w_0 + w_1 x_1 + w_2 x_2)$$

- jak zvolíme w_0, w_1 a w_2 v případě bipolárního modelu?
- a jak v případě binárního modelu?

Logický prahový obvod - AND (konjunkce)

bipolární model

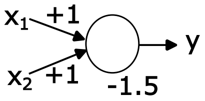
x_1	x_2	$y = x_1 \wedge x_2$
-1	-1	-1
-1	$+1$	-1
$+1$	-1	-1
$+1$	$+1$	$+1$



$$y = sign(w_0 + w_1 x_1 + w_2 x_2)$$
$$= sign(-1 + x_1 + x_2)$$

binární model

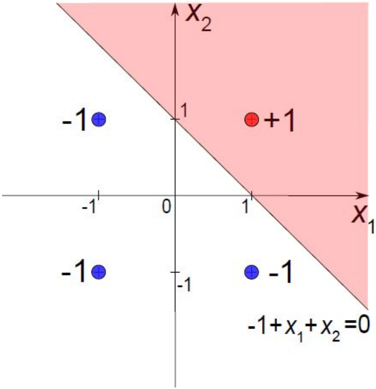
x_1	x_2	$y = x_1 \wedge x_2$
0	0	0
0	1	0
1	0	0
1	1	1



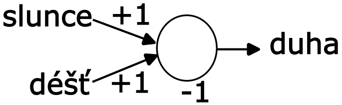
$$y = bsign(w_0 + w_1 x_1 + w_2 x_2)$$
$$= bsign(-1.5 + x_1 + x_2)$$

Otázka: Mohla by být i další řešení?

Logický prahový obvod - AND (konjunkce)



Příklad: duha $=$ slunce $\wedge$ déšť

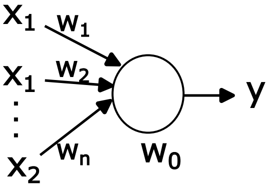


(bipolární model)

Logický prahový obvod - AND (konjunkce)

jak nastavíme váhy v obecném případě? ...

$y = x_1 \wedge x_2 \wedge \cdots \wedge x_n$

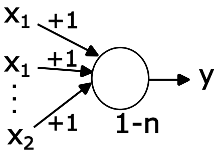


$$y = sign(w_0 + \sum_{i=1}^{n} w_i x_i)$$

Logický prahový obvod - AND (konjunkce)

$$y = x_1 \wedge x_2 \wedge \cdots \wedge x_n$$

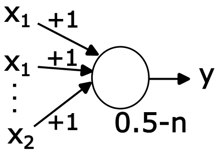
bipolární model



$$y = sign(w_0 + \sum_{i=1}^{n} w_i x_i)$$

$$= sign(1 - n + \sum_{i=1}^{n} x_i)$$

binární model



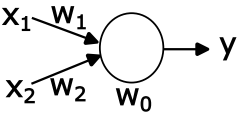
$$y = bsign(w_0 + \sum_{i=1}^{n} w_i x_i)$$

$$= bsign(0.5 - n + \sum_{i=1}^{n} x_i)$$

Logický prahový obvod - OR (disjunkce)

bipolární model

x_1	x_2	$y = x_1 \vee x_2$
-1	-1	-1
-1	+1	+1
+1	-1	+1
+1	+1	+1



binární model

x_1	x_2	$y = x_1 \vee x_2$
0	0	0
0	1	1
1	0	1
1	1	1

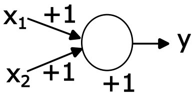
$$y = sign(w_0 + w_1 x_1 + w_2 x_2)$$

- jak zvolíme w_0, w_1 a w_2 v případě bipolárního modelu?
- a jak v případě binárního modelu?

Logický prahový obvod - OR (disjunkce)

bipolární model

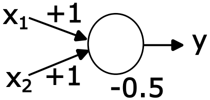
x_1	x_2	$y = x_1 \vee x_2$
-1	-1	-1
-1	+1	+1
+1	-1	+1
+1	+1	+1



$$y = sign(w_0 + w_1 x_1 + w_2 x_2) = sign(1 + x_1 + x_2)$$

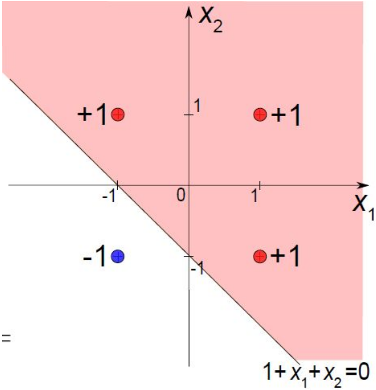
binární model

x_1	x_2	$y = x_1 \vee x_2$
0	0	0
0	1	1
1	0	1
1	1	1

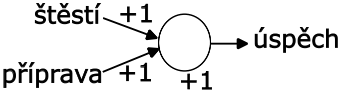


$$y = bsign(w_0 + w_1 x_1 + w_2 x_2) = bsign(-0.5 + x_1 + x_2)$$

Logický prahový obvod - OR (disjunkce)



Příklad: úspěch $=$ štěstí $\vee$ příprava

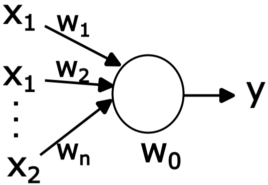


(bipolární model)

Logický prahový obvod - OR (disjunkce)

jak nastavíme váhy v obecném případě? ...

$y = x_1 \vee x_2 \vee \cdots \vee x_n$

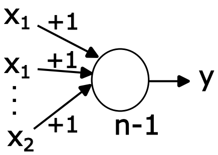


$$y = sign(w_0 + \sum_{i=1}^{n} w_i x_i)$$

Logický prahový obvod - OR (disjunkce)

$y = x_1 \vee x_2 \vee \cdots \vee x_n$

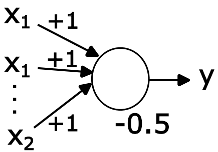
bipolární model



$$y = sign(w_0 + \sum_{i=1}^{n} w_i x_i)$$

$$= sign(n - 1 + \sum_{i=1}^{n} x_i)$$

binární model

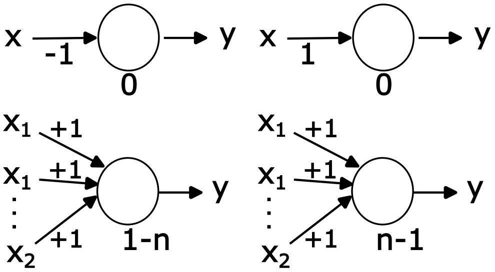


$$y = bsign(w_0 + \sum_{i=1}^{n} w_i x_i)$$

$$= bsign(-0.5 + \sum_{i=1}^{n} x_i)$$

Logický prahový obvod

- pomocí perceptronů pro NOT, ID, AND a OR můžeme sestavit složitější logické funkce



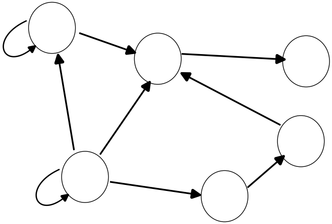
Odbočka: nejprve si definujeme umělou neuronovou síť

Neuronová síť

- Skládá se z neuronů, které jsou navzájem pospojovány tzv. hranami
- Výstup jednoho neuronu může být vstupem jednoho nebo více dalších neuronů

Architektura (topologie) neuronové sítě

- Orientovaný graf, neurony představují uzly, synaptické vazby představují hrany



Neuronová síť

Výstupní neurony

- jejich výstupy tvoří dohromady výstup neuronové sítě
- **typicky:** nevedou z nich žádné hrany do jiných neuronů

Vstupní neurony

- mají na vstupu vstupní vzory
- **typicky:** nevedou do nich žádné hrany z jiných neuronů

Výstup (odezva) neuronové sítě

- Výstupy (aktivity) výstupních neuronů

Neuronová síť

Definice: Neuronová síť je šestice (N,C,I,O,w,t):

- N je konečná neprázdná množina neuronů,
- $C \subseteq N \times N$ je neprázdná množina orientovaných spojů mezi neurony (hran)
- $I \subseteq N$ je neprázdná množina vstupních neuronů
- $O \subseteq N$ je neprázdná množina výstupních neuronů
- $w : C \to R$ je váhová funkce
- $t : N \to R$ je prahová funkce

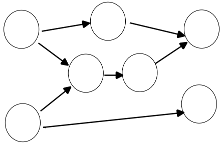
Konfigurace neuronové sítě

- Váhy všech hran a prahy všech neuronů

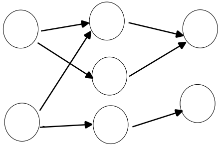
Neuronová síť

Architektura neuronové sítě

- cyklická, rekurentní
- acyklická, dopředná - ,,všechny hrany jdou stejným směrem" (tj. graf lze topologicky uspořádat)

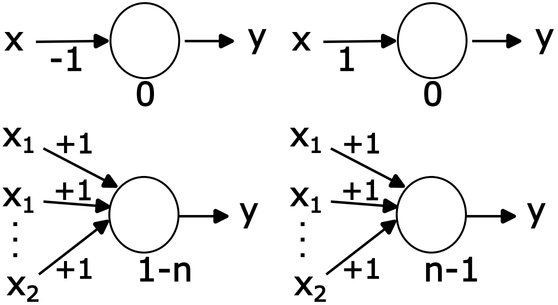


- hierarchická (vrstevnatá) - dělí se na vrstvy, propojeny jsou jen neurony ze dvou po sobě jdoucích vrstev



Logický prahový obvod

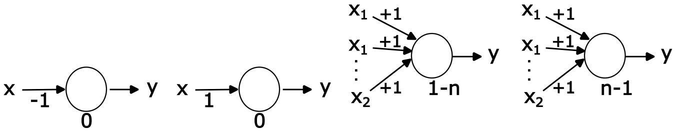
- z perceptronů pro NOT, ID, AND, OR a ID můžeme sestavit neuronovou sť reprezentující složitější logické funkce



- AND realizuje průnik konvexních útvarů a OR jejich sjednocení

Logický prahový obvod

- z perceptronů pro NOT, ID, AND, OR a ID můžeme sestavit neuronovou sť reprezentující složitější logické funkce



Záludné otázky: Jak se změní logická funkce realizovaná perceptronem, když:

- vynásobím všechny váhy (včetně prahu) kladným číslem?
- vynásobím všechny váhy (včetně prahu) záporným číslem (např. -1)?
- vynásobím některé váhy -1?

Logický prahový obvod - příklady

Příklad 1: Předpověď úrody

úroda = ((teplo ∧ déšť) ∨ (teplo ∧ zavlažování)) ∧ hnojiva ∧ ¬škůdci

1. Navrhněte perceptronovou síť pro tuto logickou funkci s využitím základních logických operací.
   - Kolik má vstupů a výstupů, kolik neuronů a kolik vrstev?
2. Navrhněte perceptronovou síť, která bude mít hierarchickou (vrstevnatou) architekturu.
3. Minimalizujte tuto logickou funkci a navrhněte neuronovou síť pro minimalizovanou verzi.
4. Lze tuto logickou funkci reprezentovat jedním perceptronem?

Logický prahový obvod - příklady

Příklad 2: Majoritní obvod

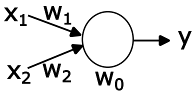
$y = (x_1 \wedge x_2) \vee (x_1 \wedge x_3) \vee (x_2 \wedge x_3)$

1. Navrhněte neuronovou síť pro majoritní obvod s využitím základních logických operací.
2. Lze ho reprezentovat jedním perceptronem?
3. Jaké bude řešení v obecném případě (pro obecné n)?

Logický prahový obvod - příklady

Příklad 3: Exkluzivní OR (XOR)
bipolární model

x_1	x_2	$y = x_1 \otimes x_2$
-1	-1	-1
-1	+1	+1
+1	-1	+1
+1	+1	-1



binární model

x_1	x_2	$y = x_1 \otimes x_2$
0	0	0
0	1	1
1	0	1
1	1	0

$$y = sign(w_0 + w_1 x_1 + w_2 x_2)$$

- jak zvolíme w_0, w_1 a w_2 v případě bipolárního modelu?
- a jak v případě binárního modelu?

Logický prahový obvod - příklady

Příklad 3: Exkluzivní OR (XOR)
bipolární model

x_1	x_2	$y = x_1 \otimes x_2$
-1	-1	-1
-1	$+1$	$+1$
$+1$	-1	$+1$
$+1$	$+1$	-1

$$sign(w_0 - w_1 - w_2) = -1$$
$$sign(w_0 - w_1 + w_2) = +1$$
$$sign(w_0 + w_1 - w_2) = +1$$
$$sign(w_0 + w_1 + w_2) = -1$$

$$1 \ldots w_0 - w_1 - w_2 < 0$$
$$2 \ldots w_0 - w_1 + w_2 > 0$$
$$3 \ldots w_0 + w_1 - w_2 > 0$$
$$4 \ldots w_0 + w_1 + w_2 < 0$$

sečteme 1. a 4., 2. a 3.:

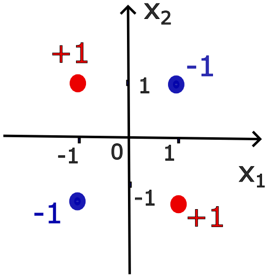
$$2w_0 < 0$$
$$2w_0 > 0$$

$\rightarrow$ spor

Logický prahový obvod - příklady

Příklad 3: Exkluzivní OR (XOR)

- XOR nelze realizovat jedním perceptronem



- XOR lze realizovat pomocí perceptronové sítě

Logický prahový obvod - příklady

Příklad 3: Exkluzivní OR (XOR) - dobrovolný domácí úkol do příští přednášky

1. XOR můžeme pomocí základních logických operací (AND, OR, NOT) reprezentovat různě, zvládnete navrhnout několik různých způsobů?
2. Navrhněte co nejmenší neuronovou síť, která bude reprezentovat XOR. Kolik bude obsahovat neuronů?

x_1	x_2	$y = x_1 \otimes x_2$
-1	-1	-1
-1	+1	+1
+1	-1	+1
+1	+1	-1

Logický prahový obvod - příklady

Příklad 3: Exkluzivní OR (XOR) - dobrovolný domácí úkol do příští přednášky

1. XOR můžeme pomocí základních logických operací (AND, OR, NOT) reprezentovat různě, zvládnete navrhnout několik různých způsobů?
2. Navrhněte co nejmenší neuronovou síť, která bude reprezentovat XOR. Kolik bude obsahovat neuronů?