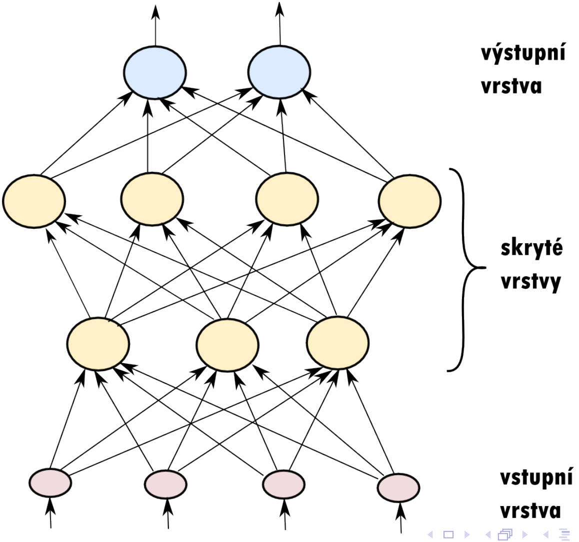
Neuronová síť

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
- $I \subseteq N$ je neprázdná množina vstupních neuronů
- $O \subseteq N$ je neprázdná množina výstupních neuronů
- $w : C \rightarrow R$ je váhová funkce
- $t : N \rightarrow R$ je prahová funkce

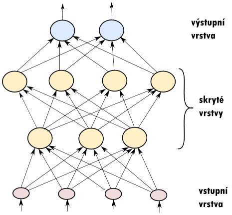
Vrstevnatá neuronová síť



Vrstevnatá neuronová síť

Definice: Vrstevnatá neuronová síť (BP-síť) je neuronová síť, která splňuje následující vlastnosti:

- Její architektura je acyklická (dopředná).
- Množina vrcholů N je tvořena posloupností (l+2) vzájemně disjunktních podmnožin zvaných vrstvy.
- Množina hran C obsahuje pouze hrany z i-té vrstvy do (i+1)-ní vrstvy.
- C obsahuje hranu pro každou dvojici neuronů z i-té vrstvy a (i+1)-ní vrstvy



Vrstevnatá neuronová síť

Definice - pokračování:

- **Vstupní vrstva** - první vrstva, je tvořena **n** vstupními neurony vstupní neuron - nevede do něho hrana z žádného jiného neuronu, jeho vstupní hodnota x je rovna jeho výstupní hodnotě.
- **Výstupní vrstva** - poslední vrstva , je tvořena **m** výstupními neurony
  výstupní neuron - nevede z něho hrana do žádného jiného neuronu.
- **Skryté vrstvy** - zbylých l vrstev neuoronů.

BP-síť implementuje funkci $\varphi : \Re^n \rightarrow A^m$.

Perceptronová síť

- vrstevnatá neuronová síť tvořená perceptrony (neurony se skokovou přenosovou funkcí).
- Role každé skryté vrstvy:
  - každý neuron dělí booleovský prostor na dva poloprostory.
  - celkem: konvexní útvary

Pomocí perceptronu lze realizovat základní logické funkce

- AND ... průnik konvexních útvarů
- OR ... sjednocení konvexních útvarů
- NOT, ID

$\rightarrow$ **logický prahový obvod**

Perceptronová síť jako logický prahový obvod

Věta

Každá booleovská formule lze vyjádřit v disjunktně konjunktním tvaru, kde atomy tvoří literály nebo jejich negace.

- $F = K_1 \vee K_2 \vee \ldots \vee K_n$
- $K_i = A_{i1} \wedge A_{i2} \wedge \ldots \wedge A_{in_i}$
- $A_{ij} = L$ nebo $A_{ij} = notL$

$\rightarrow$

Věta

Každou boolovskou funkci mohu vyjádřit pomocí perceptronové sítě.

Cvičení 1

Navrhněte schéma takové perceptronové sítě. Kolik vrstev stačí?

Perceptronová síť jako logický prahový obvod

Cvičení 2

Navrhněte váhy a prahy neuronu pro realizaci binárních funkcí (pro binární a biparitní model):

- AND ... průnik konvexních útvarů
- OR ... sjednocení konvexních útvarů
- NOT, ID

Cvičení 3

- Navrhněte (co nejmenší) architekturu, váhy a prahy perceptronové sítě pro realizaci funkce XOR (pro binární a biparitní model)

Perceptronová síť jako logický prahový obvod

Problém

Učící algoritmus pro vícevrstvý perceptron?

Řešení

Použít spojitý model (neuronu, sítě)

- Lineární či sigmoidální přenosová funkce

Neuron se sigmoidální přenosovou funkcí

- binární model: $f(\xi) = \frac{1}{1+e^{-\lambda\xi}}$ ... logsig
- bipolární model: $f(\xi) = \frac{1-e^{-\lambda\xi}}{1+e^{-\lambda\xi}}$ ... tansig
- parametr *lambda* ... strmost
  určuje míru nepřesnosti výsledku (klasifikace na hranicích)
  - *lambda* $\rightarrow \infty$ ... diskrétní model
  - čím je *lambda* menší ... tím je širší hranice mezi odlišnými případy
  - *lambda* $\rightarrow 0$ ... neuron nerozlišuje (výstup vždy 0.5 resp. 0)
  - Obvyklá volba $\lambda = 1$ nebo $\lambda = 1/4$ pro logsig, $\lambda = 2$ pro tansig

Algoritmus zpětného šíření (Back-propagation)

(Werbos, Rumelhart, 1974-1986)

Máme k dispozici

- Trénovací množina T s N trénovacími vzory (x^p, d^p).
  - $x^p = (x_1^p, \dots, x_n^p)$ ... vstupní vzor
  - $d^p = (d_1^p, \dots, d_m^p)$ ... požadovaný výstup
- Vrstevnatá neuronová síť s danou architekturou s n vstupními a m výstupními neurony. Neurony musí mít **spojitou, diferencovatelnou** přenosovou funkci.

Problém

- Nastavit váhy a práhy všech neuronů v síti tak, aby byl skutečný výstup sítě stejný jako požadovaný.

Algoritmus zpětného šíření (Back-propagation)

Cílová (chybová) funkce

- MSE ... vyjadřuje odchylku mezi skutečnou a požadovanou odezvou sítě:
  pro jeden trénovací vzor: $E^p = \frac{1}{2}\sum_i (d_i^p - y_i^p)^2$
  pro celou trénovací množinu:
  $E = \frac{1}{N}\sum_p E^p = \frac{1}{2N}\sum_p \sum_i (d_i^p - \xi_i^p)^2$
  i je index přes výstupní neurony
  p je index přes trénovací vzory

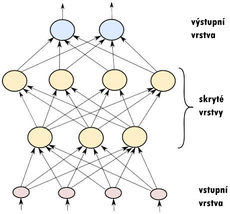
Cíl algoritmu zpětného šíření

- minimalizace chybové funkce E na dané trénovací množině T

Algoritmus zpětného šíření (Back-propagation)

Základní princip

- Spočteme skutečnou odezvu sítě pro daný trénovací vzor.
- Porovnáme skutečnou a požadovanoy odezvu sítě.
- Adaptujeme váhy a prahy:
  - proti směru gradientu chybové funkce
  - od výstupní vrstvy směrem ke vstupní



Domácí úkol

Napište skript (nebo funkci) v Matlabu, který nastaví architekturu, váhy a prahy a prahy perceptronové sítě tak, aby implementovala funkci XOR.

1. Pro binární model
2. Pro biparitní model

Program otestuje, že sítě dávají pro všechny vstupy správný výstup. Program mi pošlete na email.