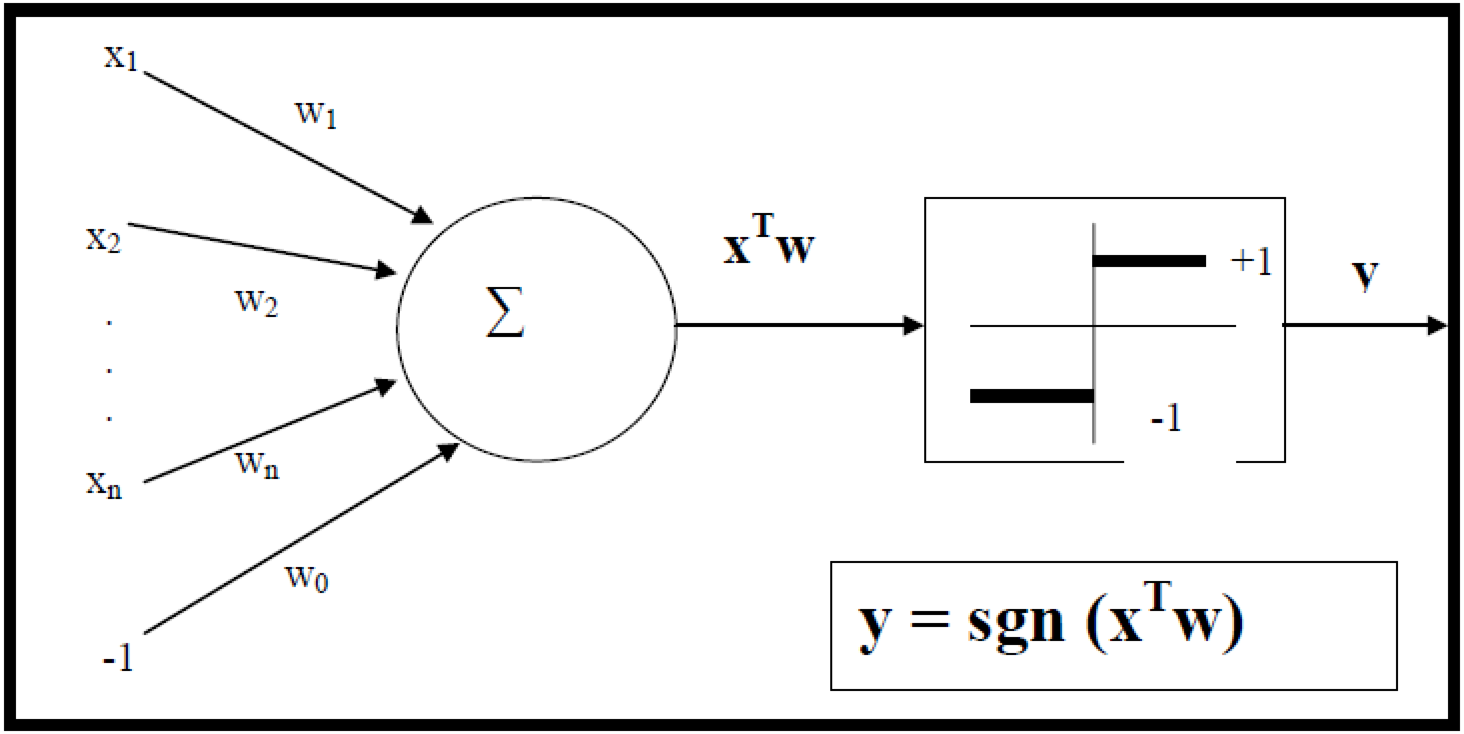
Perceptróny, model neurónovej siete

Perceptróny, učiaci algoritmus, konvergencia k riešeniu, klasifikácia do dvoch tried



Trénujúca množina

Učiace pravidlá pre perceptróny

Predpokladajme, že trénujeme perceptrón pomocou **tréningovej množiny** – dvojice

$\{ (\mathbf{x}^1, d^1), (\mathbf{x}^2, d^2), \ldots, (\mathbf{x}^m, d^m) \}$,

kde $\mathbf{x}^k \in R^{n+1}$ je k-tý vstupný vektor (príklad) a $d^k \in \{-1,+1\}$, $k=1, 2,\ldots, m$,

je predpokladaná trieda (cieľ) pre k - tý vstupný vektor (poradie dvojíc - náhodné)

Rosenblattovo pravidlo

Správna klasifikácia

Cieľ - navrhnúť perceptrón, ktorý pre každý vstupný vektor $\mathbf{x}^k$ tréningovej množiny dá výstup y^k, ktorý sa zhoduje s cieľom d^k.

Teda $y^k = \mathrm{sgn}(\mathbf{w}^T\mathbf{x}^k) = d^k$, $k=1, 2,\ldots, m$.

Riešenie $\mathbf{w}^*$ - **učiace pravidlo (Rosenblatt, 1962)**:

$$\begin{cases} \mathbf{w}^1 = \text{náhodne vybraté}, \\ \mathbf{w}^{k+1} = \mathbf{w}^k + ro\,(d^k - y^k)\mathbf{x}^k \quad k=1,2,\ldots \end{cases}$$

kde ro je kladná konštanta - **učiaci pomer (0<ro<1)**.

Príklad

- Máme dané trénujúce príklady

m	x^m	d^m
1	(2,6)	1
2	(-1,-4)	-1
3	(2,2)	1
4	(1,-3)	-1
5	(-3,-1)	1

Je potrebné nájsť perceptrón, ktorý bude správne klasifikovať uvedené body v rovine.

Príklad

Krok\Prem.	m	k	ρ	x^m	d^m	y^m	$d^m=x^m$	nezhoda	w	m< M	nezhoda = false
1	1	1	0,5					FALSE	(10, -1, -2)		
2				$x^1=(2,6)$	$d^1=1$	$y^1=sgn(10.(-1)+(-1).2+(-2).6)=-1$					
3							-	TRUE	(10, -1, -2)+1/2(1-(-1)).(-1, 2, 6) = (9, 1, 4)		
4	2									+	
5				$x^2=(-1,-4)$	$d^2=-1$	$y^2=sgn(9.(-1)+1.(-1)+4.(-4))=-1$					
6							+		(9, 1, 4)		
7	3									+	
8				$x^3=(2,2)$	$d^3=1$	$y^3=sgn(9.(-1)+1.2+4.2)=1$					
9							+		(9, 1, 4)		
10	4									+	
11				$x^4=(1,-3)$	$d^4=-1$	$y^4=sgn(9.(-1)+1.1+4.(-3))=-1$					
12							+		(9, 1, 4)		
13	5									+	
14				$x^5=(-3,-1)$	$d^5=1$	$y^5=sgn(9.(-1)+1.(-3)+4.(-1))=-1$					
15							-	TRUE	(9, 1, 4)+1/2(1-(-1)).(-1, -3, -1) = (8, -2, 3)		
16	6									+	
17				$x^6=(-1,1)$	$d^6=1$	$y^6=sgn(8.(-1)+(-2).(-1)+3.1)=-1$					
18							-	TRUE	(8, -2, 4)+1/2(1-(-1)).(-1, -1, 1) = (7, -3, 4)		
19	7									+	
20				$x^7=(6,1)$	$d^7=-1$	$y^7=sgn(7.(-1)+(-3).6+4.1)=-1$					
21							+		(7, -3, 4)		
22										-	
23	1	2									-

m	x^m	d^m
1	(2,6)	1
2	(-1,-4)	-1
3	(2,2)	1
4	(1,-3)	-1
5	(-3,-1)	1

- tréningová množina obsahuje príklady zapísané v Tab.
- počiatočné nastavenie váh je = (-1, -2) ; hodnota prahu =10;
- učiaci pomer = 0,5

Poznámky

- ak riešenie existuje, potom je dosiahnuté po konečnom počte iterácií;
- vo všeobecnosti táto metóda poskytuje nekonečný počet riešení;
- učiace pravidlo pre perceptróny konverguje k jednému z riešení;
- riešenie je citlivé na hodnotu učiaceho pomeru ro, ktorá koriguje prezentáciu dvojice;
- počet korekcií k_0 závisí od výberu počiatočných hodnôt vektora váh $\mathbf{w}^1$;
- ak riešenie neexistuje, perceptrón to neodhalí;
- vektor riešenia existuje, vtedy a len vtedy daná tréningová množina je lineárne separovateľná

Poznámky

- **Veta:** [Novikoff, 1962; Ridgway, 1962; Nilsson,1965]}
- Ak tréningová množina je lineárne separovateľná, potom učiace pravidlo pre perceptróny konverguje k riešeniu v konečnom počte iterácií.

Iné pravidlá

2. Zovšeobecnenie učiaceho pravidla perceptrónu

a)

$$\begin{cases} \mathbf{W}^1 & \text{- ľubovoľné} \\ \mathbf{W}^{k+1} = \mathbf{w}^k + \rho^k \mathbf{z}^k, & \text{ak} \quad (\mathbf{z}^k)^T \mathbf{w}^k \leq b \\ \mathbf{W}^{k+1} = \mathbf{w}^k & \text{,inak} \end{cases}$$

Veta : Ak trénujúca vzorka je separovateľná a platí

1. $\rho^k \geq 0$
2. $\lim_{m\to\infty} \sum_{k=1}^{m} \rho^k = \infty$
3. $\lim_{m\to\infty} \dfrac{\sum_{k=1}^{m} (\rho^k)^2}{\left(\sum_{k=1}^{m} \rho^k\right)^2} = 0$

potom $\mathbf{w}^k$ konverguje k riešeniu $\mathbf{w}^*$, ktoré spĺňa podmienku

$$(\mathbf{z}^i)^T \mathbf{w}^* > b \quad \text{, pre i = 1,2,...,m.}$$

Ak ρ je konštanta v ρ^k, t.j. $\rho^k = \rho/k$, $\rho^k = k\rho$, potom učiace pravidlo konverguje v konečnom čase.

Čo je to neurónová sieť

- **Neurónovú sieť** (angl. neural network - NN) - systém vzájomne poprepájaných základných stavebných prvkov, ktoré nazývame **neuróny**.
- Neurónovú sieť môžeme považovať za orientovaný graf, ktorého vrcholy reprezentujú neuróny a orientované hrany spojenia medzi neurónmi. Graf prepojenia sa nazýva **topológia siete**.
- Základné stavebné prvky dostali názov neuróny práve preto, lebo svojou činnosťou a približne aj štruktúrou pripomínajú skutočné neuróny centrálnej nervovej sústavy (CNS) živých organizmov.
- Úlohou neurónov v CNS je prijímať stimuly z okolia a podnety od ostatných neurónov, s ktorými sú spojené, spracovať ich a šíriť ich ďalej k ďalším neurónom. A práve túto vlastnosť sa snažia simulovať aj neuróny NN.

Neurón ako základný stavebný prvok

- **Neurón** ako prvok neurónovej siete môže z okolia, alebo od ďalších neurónov prijímať ľubovoľný (konečný) počet vstupov, ktoré do neho vstupujú po **neurónových spojeniach** (tiež nazývaných **neurónové spoje**).

- Predpokladajme, že neurón má **n** vstupov prichádzajúcich od iných neurónov alebo od okolia.
- Nech na i-tom vstupe prichádza informácia x_i (vo všeobecnosti $x_i \in R$, pre $i = 1, 2, \ldots, n$).

- Vstup do neurónu je potom reprezentovaný n - rozmerným vektorom $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.

Ako pracuje neurón

- Avšak nie každá stimulujúca informácia má rovnakú hodnotu - váhu, preto je každé spojenie ohodnotené tzv. **váhou spojenia**, ktorú označujeme w_i (w_i je reálne číslo, $i = 1, 2, ..., n$),
- $\theta \in R$ označuje **prahovú hodnotu neurónu** alebo jednoducho **prah**. Je to vstup do neurónu z okolitého sveta.
- Vnútorný potenciál neurónu

$$\xi = \sum_{i=1}^{n} w_i x_i - \theta$$

Ako pracuje neurón

Aktivačná funkcia

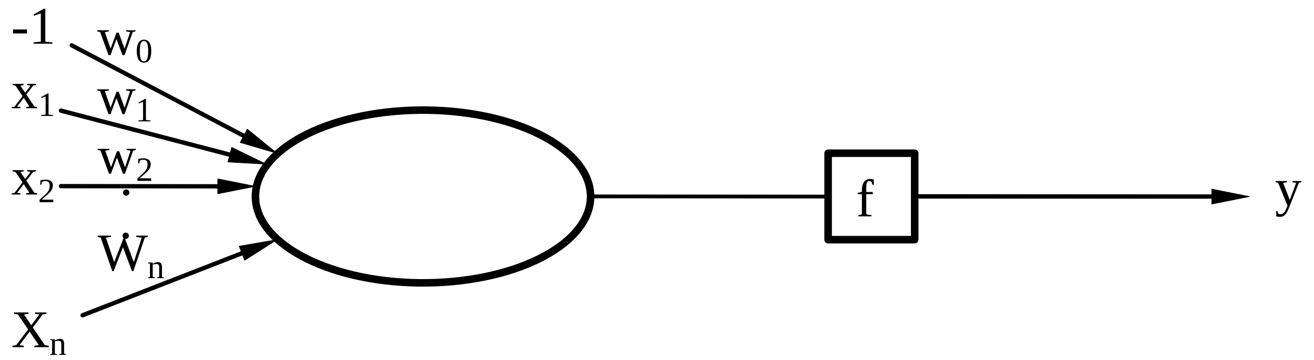
Pomocou vstupu a **aktivačnej funkcie *f*** vypočítame **stav neurónu V**. Stav neurónu V je funkciou vstupu ξ :

$$V = f(\xi)$$

t.j.:

$$V = f\left(\sum_{i=1}^{n} w_i x_i - \theta\right)$$

Neurón bez prahu



Ako aktivačná funkcia sa najčastejšie používa nelineárna sigmoidálna funkcia monotónne rastúca medzi dvoma asymptotickými hodnotami 0 a 1 :

$$f(x) = \frac{1}{1 + \exp(-x)}$$

Ďalšie aktivačné funkcie môžu byť :

- *funkcia s pevným ohraničením*, napr. $f(x) = \mathrm{sgn}(x)$
- *funkcia lineárna*, napr. f(x)=x
- *funkcia nelineárna*, napr. $f(x) = \tanh(x)$

Ako pracuje neurón

- **Výstup neurónu y** vypočítame pomocou výstupnej funkcie *f* ako funkciu stavu neurónu V.
- Funkciou *f* je veľmi často identita (ďalej budeme stále uvažovať výstupnú funkciu *f* = id). Z toho vyplýva, že y = V, a teda výstup neurónu vypočítame :

$$y = f\left(\sum_{i=1}^{n} w_i x_i - \theta\right)$$

Topológia neurónovej siete

- **Topológiou neurónovej siete** rozumieme graf prepojenia danej NN. Ide o orientovaný graf, ktorého vrcholy reprezentujú neuróny a orientované hrany spojenia medzi neurónmi.

- Vlastnosti takýchto všeobecných sietí sa ťažko analyzujú, a preto sa študujú najprv siete s nejakými pravidelnými štruktúrami.

- Jednou z nich (doposiaľ asi najviac preskúmanou) je **viacvrstvová štruktúra**. Vo viacvrstvových sieťach vieme neuróny usporiadať do vrstiev, pričom rozlišujeme vrstvu vstupnú, skrytú a výstupnú

Topológia neurónovej siete

- Vo všeobecnosti rozdeľujeme topológiu NN do dvoch základných skupín :

- **dopredné NN (angl. feed-forward NN)** - signál sa šíri po orientovaných hranách len jedným smerom - dopredu
- **rekurentné NN (ang. recurrent NN)** - je ťažké rozdeliť neuróny do vrstiev, pretože obsahujú okrem dopredných spojení aj prepojenia, ktoré dovoľujú šíriť signál aj späť - spätné prepojenia, niekedy aj prepojenia samých so sebou.

Viacvrstvová NS

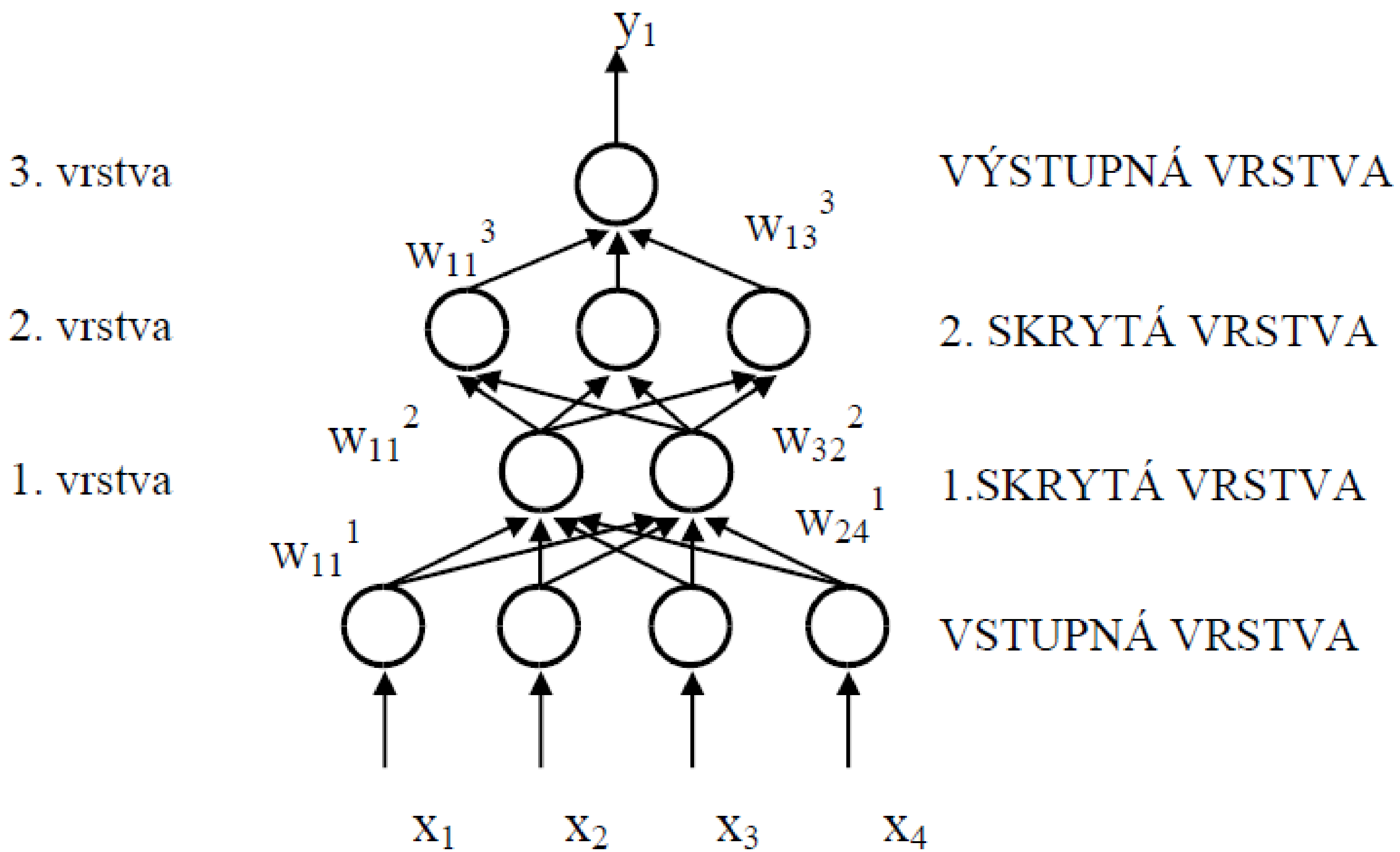
- Vo **viacvrstvových dopredných sieťach** (angl. feed-forward multilayer networks) sú neuróny v sieti usporiadané do niekoľkých vrstiev (minimálne dvoch), pričom rozoznávame tri typy vrstiev :

- ***vstupná vrstva***
- ***skryté vrstvy***
- ***výstupná vrstva***

- Každá sieť má práve jednu vstupnú, práve jednu výstupnú a ľubovoľný konečný (aj nulový) počet ***r*** skrytých vrstiev. Vstupnú vrstvu označujeme ako 0-tú vrstvu, ďalej sú vrstvy číslované tak, ako za sebou nasledujú. Nech vstupná vrstva je tvorená ***n*** neurónmi, výstupná ***s*** neurónmi a s-tá skrytá vrstva pozostáva z k_s neurónov. Hovoríme o **(r+1)-vrstvovej sieti** typu $\mathbf{n - k_1 - k_2 - \ldots - k_r - s}$.

- Každý neurón **vstupnej vrstvy** má práve jeden vstup, ktorý je z vonkajšieho sveta (prah teraz nebudeme započítavať medzi vstupy) a výstup, ktorý pokračuje k ďalším neurónom.

Štruktúra neurónovej siete



Činnosť neurónovej siete

- Rozlišujeme dve činnosti neurónovej siete, a to

- ***aktívnu prácu ,*** hovoríme tiež o **aktívnom (pracovnom) móde** a
- o **adaptačnom (učiacom) móde**. Pri *učení* predkladáme sieti vstupy a upravujeme jej parametre za účelom získať na výstupe siete očakávanú odozvu, pri *aktívnej práci* sieť pri danom nastavení parametrov a danom vstupe určí výstupy siete podľa daného algoritmu.

Aktívny (pracovný) mód

Sieť pracuje v **aktívnom móde**, ak pri danom nastavení parametrov siete, t.j. keď poznáme počet skrytých vrstiev, počet neurónov vo vrstvách a váhy všetkých spojení, hľadá približnú hodnotu celkového výstupu siete , *f*( ) , pričom je daný vstup siete a *f* je dané zobrazenie, a to podľa nasledujúceho postupu :

1. Na vstup siete predložíme n- rozmerný vektor , t.j. vstupom i-teho neurónu (i = 1, 2, ..., n) vstupnej vrstvy je $x_{i.}$ Vstupná vrstva nič nepočíta a svoj vstup pošle nezmenený na výstup neurónu a šíri ho po spojeniach do všetkých neurónov nasledujúcej vrstvy.
2. Každý neurón skrytej vrstvy vypočíta svoj výstup a pošle ho po svojich spojeniach na vstup každého neurónu nasledujúcej vrstvy.
3. Túto činnosť sieť vykonáva až kým nevypočíta výstupy y_i (i = 1, 2, ..., s) výstupnej vrstvy. Takto získaný vektor =(y_1, y_2, ..., y_s) je hľadanou približnou hodnotou výsledku.

Adaptačný (učiaci) mód

Adaptačným (učiacim) módom nazývame takú činnosť siete, pri ktorej sieť nastavuje svoje parametre tak, aby v aktívnom móde pracovala správne, teda aby vypočítaný výstup skutočne odpovedal približnej hodnote zobrazenia f pre daný vstup.

Neurónové siete vznikli ako zjednodušená abstrakcia CNS živých organizmov, ktoré sa učia na základe príkladov - na určitý druh stimulu formujú určitú odozvu. Množina **príkladov -** dvojíc typu [stimul, žiadaná odozva], je tá množina, na základe ktorej sa učia nielen živé organizmy, ale aj NN.

V terminológii neurónových sietí hovoríme o množine dvojíc

[vstup, žiadaný výstup] ako o **tréningovej množine T**.

Keď chceme "naučiť" NN, aby počítala hodnoty zobrazenia f s ľubovoľnou presnosťou, musíme *určiť zobrazenie f*, a to množinou usporiadaných dvojíc - tréningovou množinou príkladov T. Na základe príkladov z T hľadáme sieť tak, aby realizovala zobrazenie f čo najpresnejšie, a aby čo najviac extrahovala poznatky z množiny T.

- Spracované na základe knihy prof. Kvasničku a prof. Pospíchala.