Dopredné neurónové siete

Back – propagation algoritmus

Viacvrstvová NS

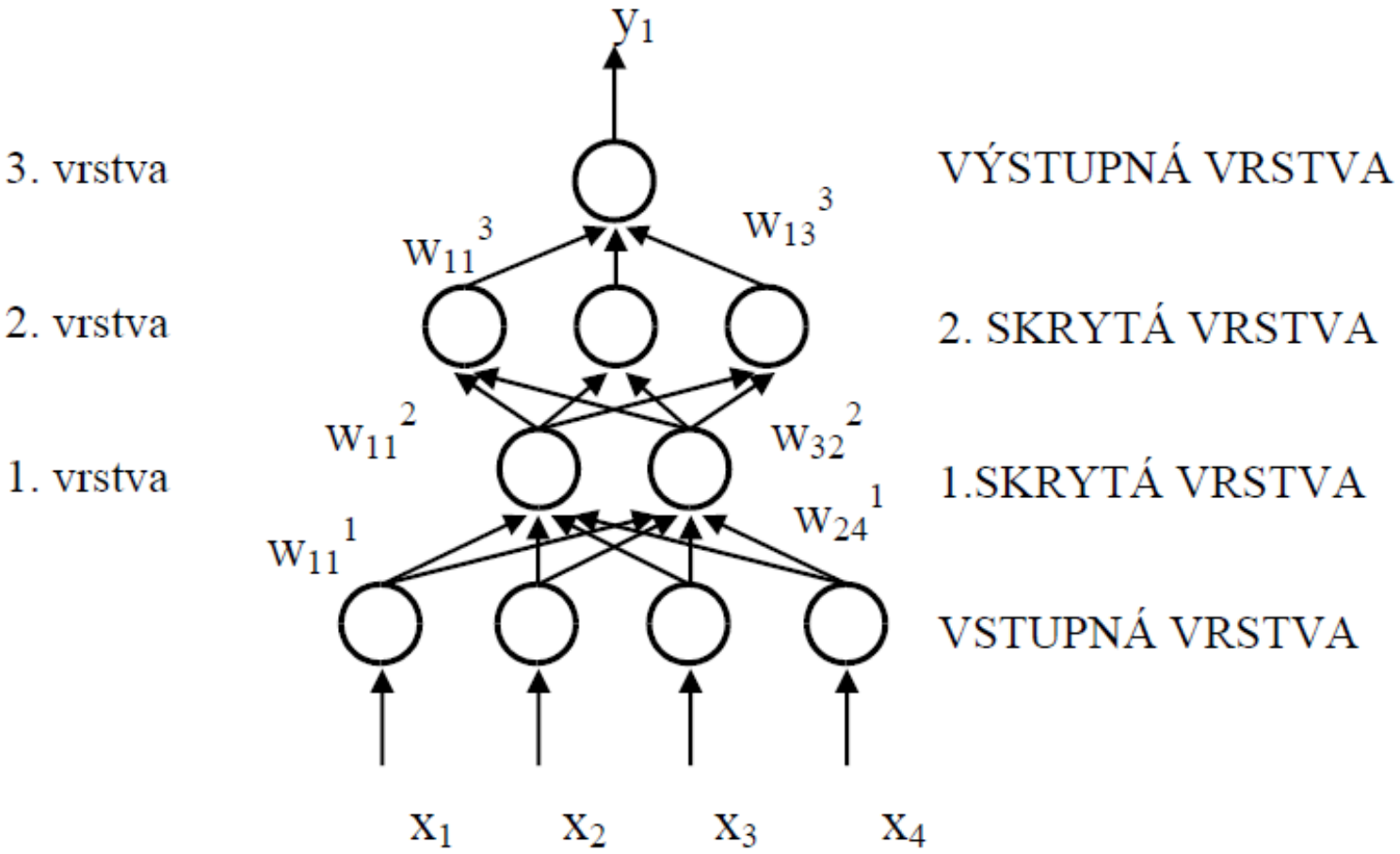
- Vo **viacvrstvových dopredných sieťach** (angl. feed-forward multilayer networks) sú neuróny v sieti usporiadané do niekoľkých vrstiev (minimálne dvoch), pričom rozoznávame tri typy vrstiev :

- ***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 **$n - k_1 - k_2 - \ldots - k_r - s$.**

- 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



Obr.1.3. : *Viacvrstvová sieť typu 4-2-3-1*

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 **y** , **y=*f*(*x*)**, pričom je daný vstup siete **x** a *f* je dané zobrazenie, a to podľa nasledujúceho postupu:

- Na vstup siete predložíme n- rozmerný vektor **x** , 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.
- Každý neurón skrytej vrstvy vypočíta svoj výstup podľa vzťahu a pošle ho po svojich spojeniach na vstup každého neurónu nasledujúcej vrstvy.
- Túto činnosť sieť vykonáva až kým nevypočíta výstupy y_i (i = 1, 2, ..., s) výstupnej vrstvy. Takto získaný vektor **y**=(y_1, y_2, ..., y_s) je hľadanou približnou hodnotou ***f*(*x*)**.

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 [$\mathbf{x}$, $f(\mathbf{x})$] - 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.

Hľadanie neurónovej siete

Nájsť sieť znamená

- určiť počet skrytých vrstiev a počet neurónov v nich
- stanoviť váhy všetkých spojení a prahy všetkých neurónov siete
- ak to určitá metóda hľadania siete vyžaduje, tak aj niektoré ďalšie parametre

Pri adaptácii siete sa jej parametre môžu meniť dvoma spôsobmi, a na základe toho rozoznávame dva typy adaptácii sietí:

- **prírastková adaptácia**
- parametre siete sa menia po každej prezentácii tréningového vzoru na vstup. Takže výsledné nastavenie parametrov je výsledkom postupných zmien a prispôsobovania sa počas celého procesu učenia.
- **dávkovaná adaptácia**
- parametre siete sa nastavia až po ukončení tréningového cyklu, a to na základe údajov, ktoré sa počas učenia siete počítajú a uchovávajú.

Gradientná metóda

Základom týchto metód je nájsť také hodnoty parametrov siete, ktoré by minimalizovali danú **chybovú funkciu E**. Z daných počiatočných hodnôt parametrov sa postupuje v smere klesajúceho gradientu príslušnej nadplochy k menším hodnotám.

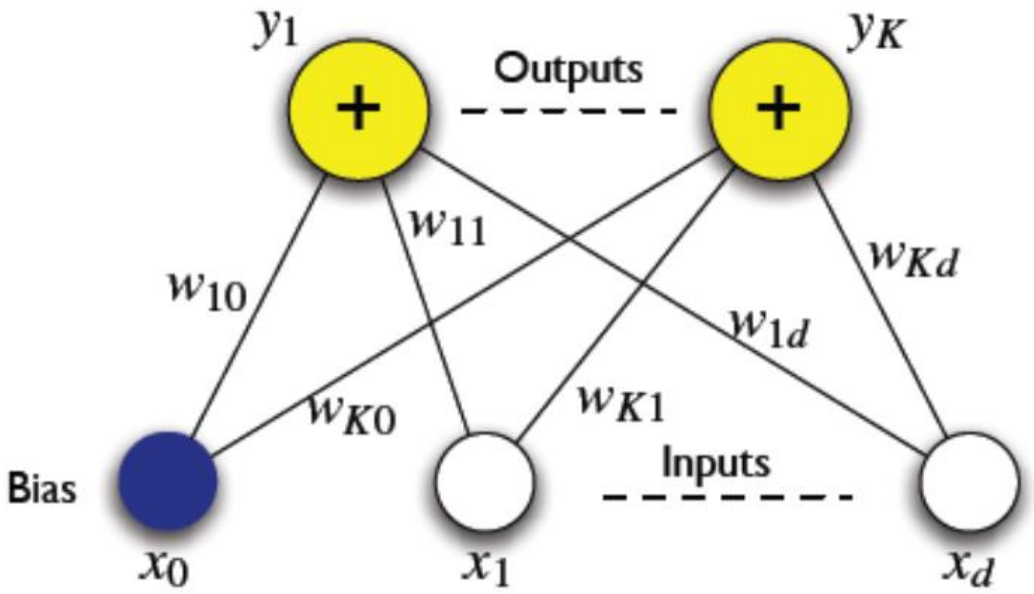
- Ako prvé vznikli metódy minimalizujúce globálnu chybu GE, ktorá je na sieti s pevnou topológiou definovaná takto :

$$\varepsilon \equiv GE = \frac{1}{2}\sum_{T}\sum_{i=1}^{n}(y_i - d_i)^2$$

kde

- $\mathbf{d_i}$ je očakávaná odozva siete na daný vstupný vektor v i-tom výstupnom neuróne a
- $\mathbf{y_i}$ je vypočítaná odozva siete v i-tom výstupnom neuróne, ktorú vypočíta sieť v aktívnom móde pri danom nastavení váh a prahov.
- **T** je počet tréningových vzorov, ktoré boli sieti predkladané v určitom časovom intervale,
- **n** je počet výstupných neurónov.

Jednovrstvová sieť - analýza



Tréningová množina

- Body v rovine klasifikovať do troch tried:
- ((3, 7), (100))
- ((4, 5), (100)) …
- ((8, 11), (010))
- ((11,11),(010)) …
- ((15,7), (001))
- ((19,7), (001)) …

- Body môžu byť usporiadané aj inak

Algoritmus backpropagation

Odvodený algoritmus dostal názov **backpropagation** (BP) - metóda spätného šírenia chyby, pretože zatiaľ čo sa aktivita siete šíri smerom zdola nahor (od vstupnej vrstvy k výstupnej), chyby a jej dôsledky sa šíria naopak - zhora nadol (od výstupnej vrstvy k vrstve vstupnej). Obvykle totiž dochádza pri úprave váh na spojeniach medzi nižšími vrstvami k prenosu časti chyby z vyšších vrstiev na nižšie zložitým rekurzívnym výpočtom.

- Uvažujme doprednú neurónovú sieť s M vrstvami, pričom použijeme nasledujúce označenie:
- $\mathbf{V_i^m}$ (m=1,...,M) je výstup i-teho neurónu m-tej vrstvy,
- $\mathbf{V_i^0}$ je výstup i-teho neurónu vstupnej vrstvy,
- $\mathbf{w_{ij}^m}$ je hodnota váhy medzi j-tym neurónom (m-1)-vej vrstvy a i-tym neurónom m-tej vrstvy.

Algoritmus backpropagation

KROK 1: Inicializujeme váhy na malé náhodné hodnoty.

KROK 2: Vyberieme vzorku $\mathbf{x_i}$ z tréningovej množiny a predložíme ju na vstup siete, t.j.

$$V_i^0 = \mathbf{x_i}, \quad \text{pre každé i}$$

KROK 3: Vstupný signál šírime cez celú sieť, pričom výstupy jednotlivých neurónov siete vypočítame nasledovne :

$$V_i^m = g(h_i^m) = g(\sum_j w_{ij}^m V_j^{m-1}), h_i^m = \sum_j w_{ij}^m V_j^{m-1}$$

pre každé i, m, kým nevypočítame výstupy V_i^M.

Algoritmus backpropagation

KROK 4: Vypočítame delty δ_i^M pre výstupnú vrstvu :

$$\delta_i^M = g'(h_i^M)(y_i - V_i^M),$$

pre každé i, pričom porovnávame očakávaný výstup y_i s vypočítaným V_i pre predložený vzor **x**.

KROK 5: Vypočítame delty δ_i^m pre predchádzajúce vrstvy, propagovaním chyby od výstupnej ku vstupnej vrstve :

$$\delta_i^{m-1} = g'(h_i^{m-1}) \sum_j w_{ji}^m \delta_j^m$$

pre m= M, M-1, ...,2, kým nevypočítame delty pre všetky neuróny.

Algoritmus backpropagation

KROK 6: Zmenu váh všetkých spojení v sieti vypočítame nasledovne :

$$\Delta w_{ij}^{m} = \eta \delta_i^{m} V_j^{m-1}$$

Aktualizujeme váhy :

$$w_{ij}^{nové} = w_{ij}^{staré} + \Delta w_{ij}$$

KROK 7: Opakujeme prechodom na krok 2 pre ďalšiu vstupnú vzorku.

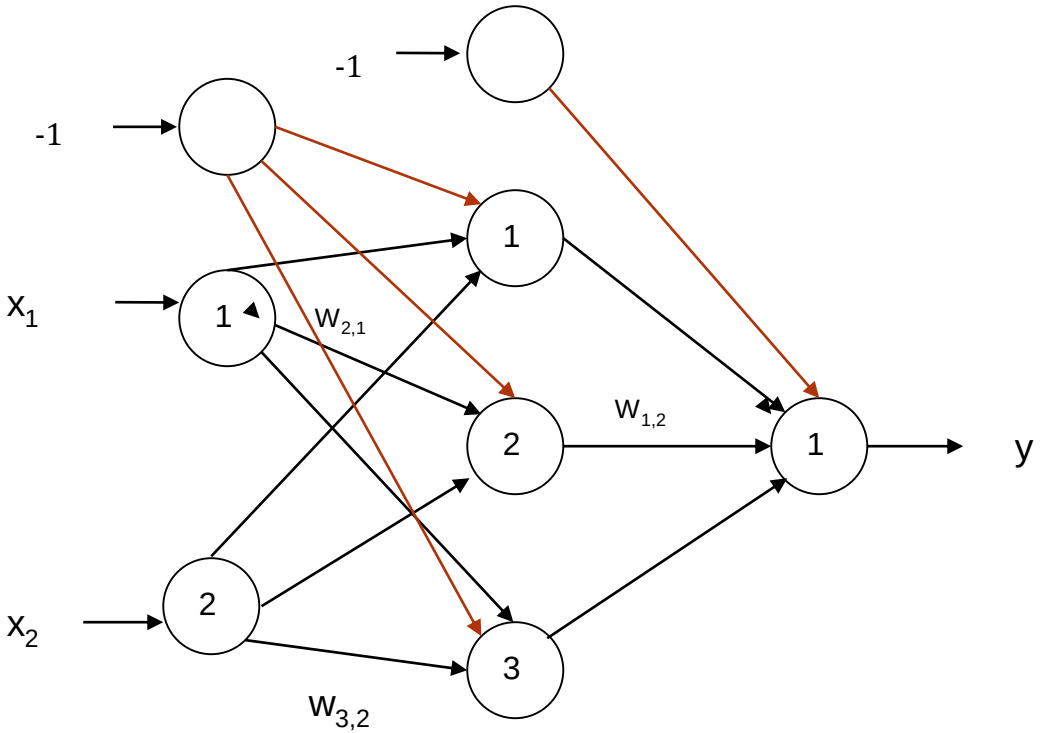
Poznámky

- Jedným z faktorov, ktoré majú vplyv na rýchlosť konvergencie, je **učiaci pomer η**. Pre malé η, bude zostup do minima trvať dlho, ale pri veľkom η môžeme hľadané minimum preskočiť.

- Na **dobu výpočtu** má vplyv aj počet neurónov. Počet neurónov vo vstupnej a výstupnej vrstve je daný samotnou úlohou, ale počty skrytých vrstiev a neurónov v týchto vrstvách zvyčajne určuje sám riešiteľ.

- Príliš **veľký počet skrytých neurónov** nielenže predlžuje dobu výpočtu, ale môže mať za následok zbytočne veľa variantov riešení (dochádza k zlej generalizácii siete), navyše sa sieť zle vyrovnáva s nepresnými dátami (šumom) a má tendenciu vystihovať aj tento nepodstatný šum. Avšak príliš málo skrytých neurónov sťažuje alebo úplne znemožňuje konvergenciu.

Príklad



Vstupy a očakávaný výstup:
((2, 1), 0.8)
((-1,1), 0.4)
((3, 2,), 0.7)
Aktivačná funkcia neurónov: $g(z)=1/(1+\exp(-z))$,

$$g'(z)=(-1+\exp(-z))) /((1+\exp(-z))*(1+\exp(-z)))$$

Váhy neurónov si zvolíme.
Aplikujme metódu backpropagation na tomto príklade.