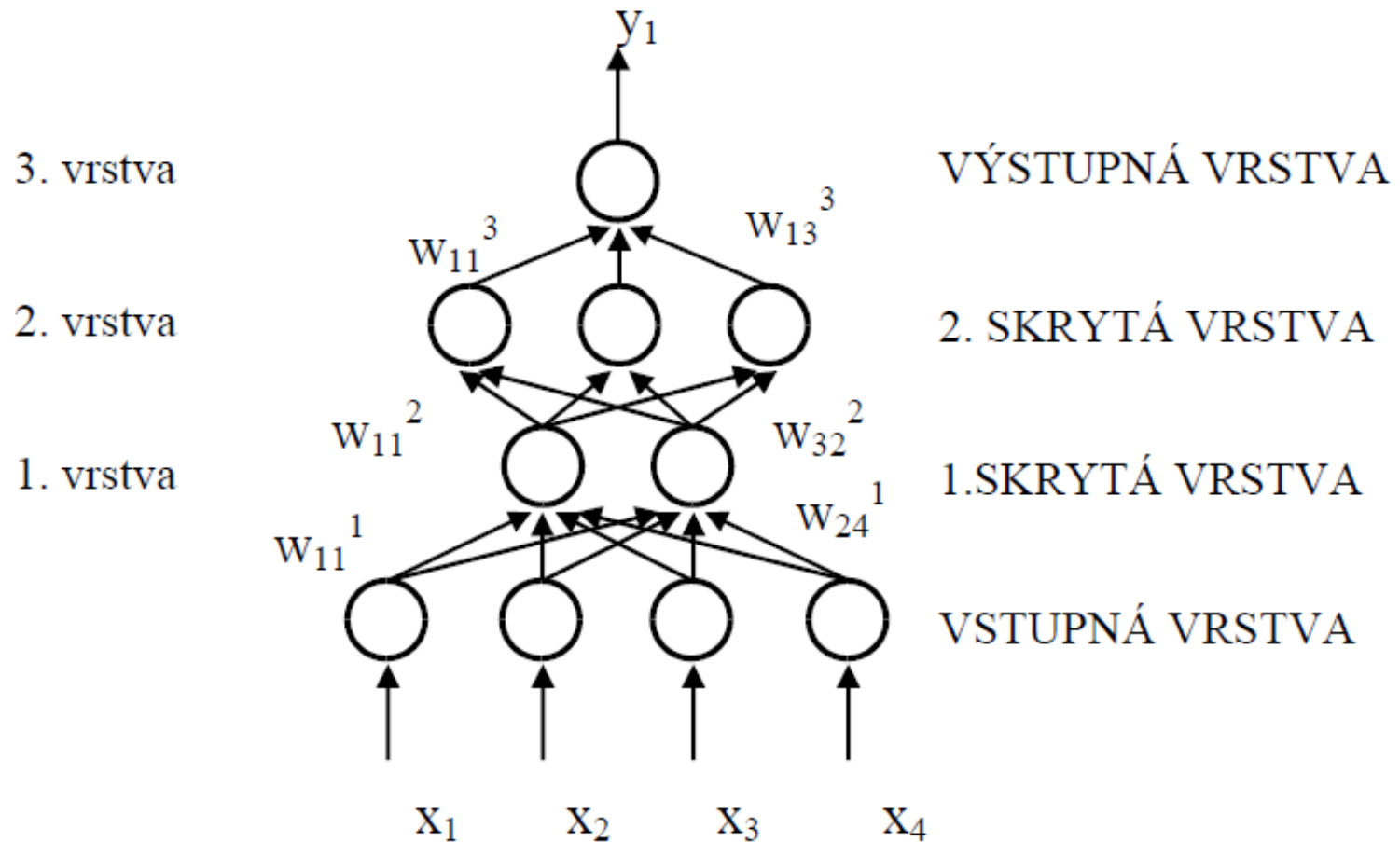


Dopredné neurónové siete

Back-propagation algoritmus

Štruktúra neurónovej siete



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

Algoritmus backpropagation

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:

- V_i^m ($m=1,\dots,M$) je výstup i -teho neurónu m -tej vrstvy,
- V_i^0 je výstup i -teho neurónu vstupnej vrstvy, teda vstupné hodnoty,
- 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 :

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

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 porovnáваме očakávaný výstup y_i s vypočítaným V_i pre predložený vzor \mathbf{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 $m = M, M-1, \dots, 2$, 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 :

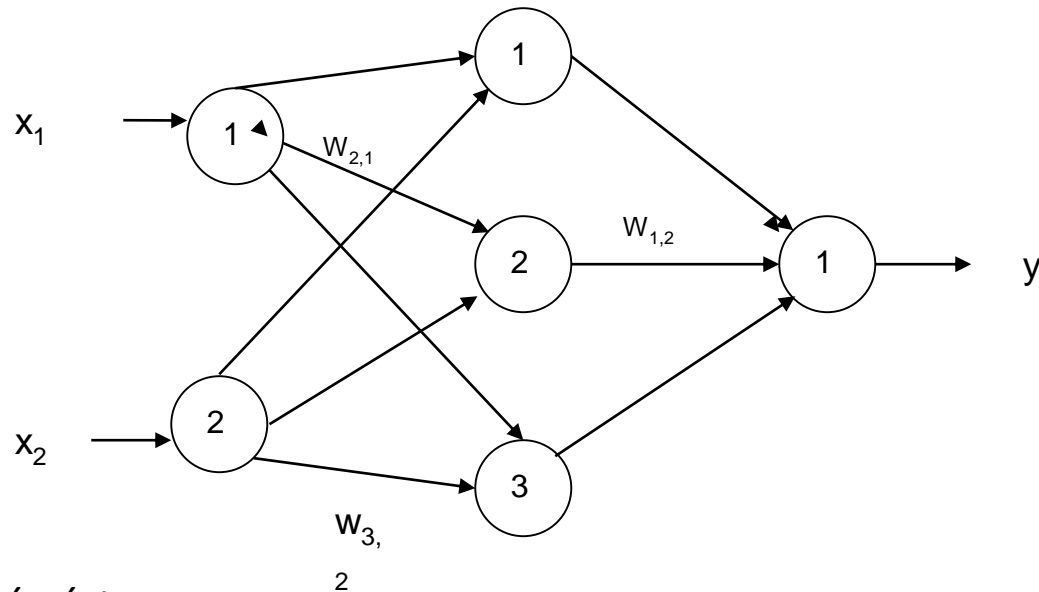
$$w_{ij}^{\text{nové}} = w_{ij}^{\text{staré}} + \Delta w_{ij}$$

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

Poznámky

- Jedným z faktorov, ktoré majú vplyv na rýchlosť konvergenencie, 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 konvergenziu.

Príklad



Vstupy a očakávaný výstup:

$((2, 1), 0.2)$

$((-1, 1), 0.3)$

$((3, 2), 0.3)$

Aktivačná funkcia neurónov: $g(z) = 1/(1 + \exp(-z))$

Váhy neurónov si zvolíme.

Aplikujme metódu back-propagation na tomto príklade.