Záznam z přednášky č.9

Václav Kandrnal, Ondřej Špinka

20.5.2002

1 Mnohovrstvý perceptron

1.1 Úvod

Problém omezené výpočetní schopnosti perceptronu se vyřešil až v roce 1986, kdy Rumelhart, Hinton a Williams představili metodu zpětného šíření (backpropagation) pro trénování dopředné neuronové sítě se skrytými neurony.

Tzv. vícevrstvé dopředné umělé neuronové sítě (multilayer feed forward ANN), které se trénují tímto pravidlem, jsou schopny řešit i nelineární problémy.

1.2 Dopředné neuronové sítě

Dopředné neuronové sítě se vyznačují tím, že v nich existují pouze dopředná spojení mezi neurony. Každý neuron jedné vrstvy vysílá signály na každý neuron následující vrstvy.

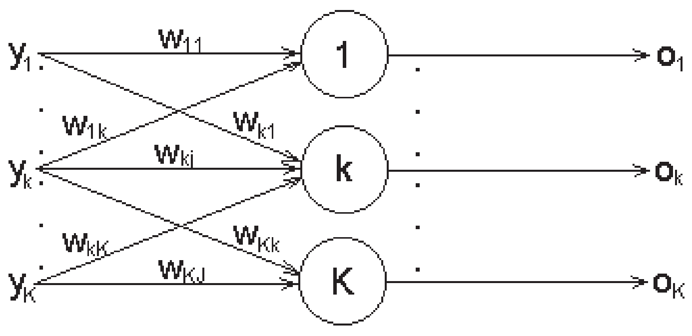
Nejprve zobecněme pravidlo δ pro jednovrstvou neuronovou síť složenou ze spojitých perceptronů. Výstup spojitého perceptronu je dán vztahem:

$$o = f(\varphi) = f(\vec{w}.\vec{x}) = f(\sum_{j=1}^{n+1} w_j x_j) = f(\sum_{j=1}^{n} w_j x_j - \emptyset) \tag{1}$$

Aktivační funkce spojitého perceptronu $f(\varphi)$ může být libovolná diferencovatelná funkce. Položme

$$f(\varphi) = \frac{1}{1 + e^{-\lambda\varphi}} \qquad \text{(Unipolární sigmoida)} \tag{2}$$

Konstanta $\lambda > 0$ se nazývá strmost sigmoidy. Obvykle se používá $\lambda = 1$. V limitě pro $\lambda \to \infty$ přejde unipolární sigmoida ve skokovou funkci. Mějme jednovrstvou neuronovou síť (viz. Obr.1)



Obrázek 1: jednovrstvá neuronová síť (w_{AB} je váha hrany z B do A)

1.2.1 Označení

Vstupní vektor $\vec{y} = (y_1, \ldots, y_j, \ldots, y_J)$
Výstupní vector $\vec{o} = (o_1, \ldots, o_k, \ldots, o_K)$ $\quad o_k = f(\varphi_k)$

$$\varphi_k = \sum_{j=1}^{J} w_{kj}.y_j \tag{3}$$

Nechť vždy $y_J = -1$ a $w_{kJ} = o_{kj}$. Požadovaný výstup sítě je $\vec{d} = (d_1, \ldots, d_k, \ldots, d_K)$ (d-desired).

1.2.2 Učení

Chceme, aby se po naučení skutečný výstup sítě rovnal požadovanému výstupu, resp. aby se mu přiblížil co nejvíce, a to pro všechny vzory $p = 1, 2, \ldots, P$ z trénovací množiny A. Definujeme účelovou funkci - chybová funkce (error function) tvaru:

$$E_p = \frac{1}{2}\sum_{k=1}^{K}(d_{pk} - o_{pk})^2 \tag{4}$$

kde p je index vzoru a E_p je sumou čtverců chyb na všech výstupních neuronech. Učení sítě spočívá v modifikování vah tak, aby se minimalizovala E_p. Na hledání minima E_p aplikujeme metodu záporného gradientu. Změna váhy je nepřímo úměrná parciální derivaci E_p podle w_{kj}:

$$\triangle W_{kj} = -\alpha\frac{\partial E_p}{\partial w_{kj}} = -\alpha\frac{\partial E_p}{\partial \varphi_k}.\frac{\partial \varphi_k}{\partial w_{kj}} = \alpha\delta_{ok}y_j \tag{5}$$

kde α je rychlost učení. Záporná parciální derivace $\frac{\partial E_p}{\partial \varphi_k} = \delta_{ok}$ je zobecněný učící signál produkovaný $k-$tým výstupním neuronem.

Parciální derivace

$$\frac{\partial \varphi_k}{\partial w_{kj}} = y_j \tag{6}$$

(viz.(3)).

Čemu se rovná δ_{ok}:

$$\delta_{ok} = -\frac{\partial E_p}{\partial \varphi_k} = -\frac{\partial E_p}{\partial o_k}.\frac{\partial o_k}{\partial \varphi_k} = (d_{pk} - o_{pk}).f'_k \tag{7}$$

kde f'_k je derivace aktivační funkce neuronu (sigmoidy) podle φ_k. Pro náš případ je to

$$f'_k = o_k(1 - o_k) \tag{8}$$

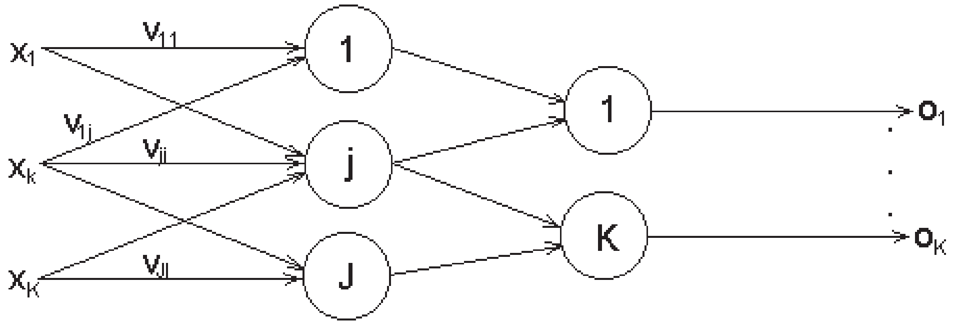
pravidlo na změny $j-$té váhy $k-$tého výstupního neuronu je potom

$$\triangle w_{kj} = \alpha\underbrace{(d_{pk} - o_{pk}).f'_k}_{\delta_{ok}}y_j \tag{9}$$

kde δ_{ok} je zobecněný chybový signál šířící se na všechny váhy přicházející na $k-$tý výstupní neuron.

Pozn. Položíme-li $f'_k = 1$, dostaneme učící pravidlo pro perceptron.

Přidejme nyní do sítě další vrstvu neuronů, kterou nazveme skrytá vrstva (viz. následující obrázek):



Obrázek 2: skrytá vrstva

Vstup je totožný se vstupem pro skrytou vrstvu: $\vec{x} = (x_1, \ldots, x_i, \ldots, x_I)$. Výstupní neurony zpracovávají výstup skryté vrstvy $\vec{y} = (y_1, \ldots, y_j, \ldots, y_J)$, $y_i = f(\varphi_j)$.

$$\varphi_j = \sum_{i=1}^{I} v_{ji} x_i \tag{10}$$

V dalším odvodíme vztahy pro modifikaci vah mezi vstupem a skrytou vrstvou. Také tyto váhy je třeba měnit tak, aby se minimalizovala chyba E_p, vyjádřená v (4) pomocí gradientní metody nejmenšího spádu:

$$\triangle v_{ji} = -\alpha \frac{\partial E_p}{\partial v_{ji}} = -\alpha \frac{\partial E_p}{\partial \varphi_j} . \frac{\partial \varphi_j}{\partial v_{ji}} = \alpha \delta y_j x_i \tag{11}$$

Záporná parciální derivace $-\frac{\partial E_p}{\partial \varphi_j} = \delta_{yj}$, což je zobecněný učící signál produkovaný $j-$tým neuronem. δ_{yj} je zobecněná chyba šířená na vstupní váhy.

$$\frac{\partial \varphi_j}{\partial v_{ji}} = x_i \qquad (\text{viz.}(10))$$

Dále platí:

$$\delta_{yj} = -\frac{\partial E_p}{\partial \varphi_j} = -\frac{\partial E_p}{\partial y_j} . \frac{\partial y_j}{\partial \varphi_j} = -\frac{\partial E_p}{\partial y_j} . f'_j \tag{12}$$

f'_j je derivace výstupní funkce skrytého neuronu podle φ_j.

$$\frac{\partial E_p}{\partial y_j} = -\sum_{k=1}^{K} (d_{pk} - o_{pk}) \frac{\partial f(\varphi_k)}{\partial y_j} = -\sum_{k=1}^{K} (d_{pk} - o_{pk}) \frac{\partial f(\varphi_k)}{\partial \varphi_k} . \frac{\partial \varphi_k}{\partial y_j} \tag{13}$$

Je-li f'_k derivace sigmoidy výstupního neuronu podle φ_k a $\frac{\partial \varphi_k}{\partial y_j} = w_{kj}$ potom

$$\frac{\partial E_p}{\partial y_j} = -\sum_{k=1}^{K} (d_{pk} - o_{pk}) f'_k w_{kj} = -\sum_{k=1}^{K} \delta_{ok} w_{kj} \tag{14}$$

Po dosazení do (12) dostáváme pro δ_{yj}

$$\delta_{yj} = \left(\sum_{k=1}^{K} \delta_{ok} w_{kj} \right) f'_j \tag{15}$$

A nakonec vztah pro určení vah mezi vstupem a skrytými neurony:

$$\triangle v_{ji} = \alpha \left(\sum_{k=1}^{K} \delta_{ok} w_{kj} \right) f_j' x_i \tag{16}$$

Zobecněním pro n skrytých vrstev platí pro $n-$tou vrstvu:

$$\triangle v_{ji}^{n} = \alpha \delta_{yj}^{n} . x_i^{n-1} \tag{17}$$

kde

$$\delta_{yj}^{n} = \left(\sum_{k=1}^{K} \delta_{ok}^{n+1} w_{kj}^{n+1} \right) (f_j^n)' \tag{18}$$

kde $(f_j^n)'$ je derivace výstupní funkce $j-$tého skrytého neuronu ve $w-$té skryté vrstvě podle φ_j^n.

2 RBF (radial basis function)

2.1 Úvod

- Patří mezi nejmladší typy neuronových sítí.
- Motivace k jejich studiu lze nalézt v numerické matematice, konkrétně ve studiu interpolací a aproximací dat. Řešení aproximačních úloh hledáme jako lineární kombinaci bazických funkcí v nějakém konkrétním tvaru.
- Radiální funkce je určena svým středem a její hodnota závisí na vzdálenosti argumentu od tohoto středu.

  Síť typu RBF realizuje lineární kombinaci radiálních bazických funkcí.
- Základním stavebním kamenem RBF jsou tzv. RBF jednotky. Tato realizuje radiální funkci. Má n reálných vstupů $(x_1, \ldots, x_n)$ a jeden reálný výstup y. Každá jednotka je určena $n-$rozměrným vektorem $\vec{c}$, který nazýváme střed a může mít šířku b.

Vnitřní potenciál se počítá jako euklidovská vzdálenost vstupního vektoru $\vec{x}$ od $\vec{c}$ dělenou šířkou b:

$$\varphi = \frac{\|\vec{x} - \vec{c}\|}{b} \tag{19}$$

Výstup RBF y je roven hodnotě příslušné radiální funkce g v bodě φ. Nejpoužívanější radiální funkcí je Gaussova:

$$g(z) = e^{-z^2} \tag{20}$$

=> RBF realizuje funkci

$$y(\vec{x}) = e^{-\left(\frac{\|\vec{x}-\vec{c}\|}{b}\right)^2} \tag{21}$$

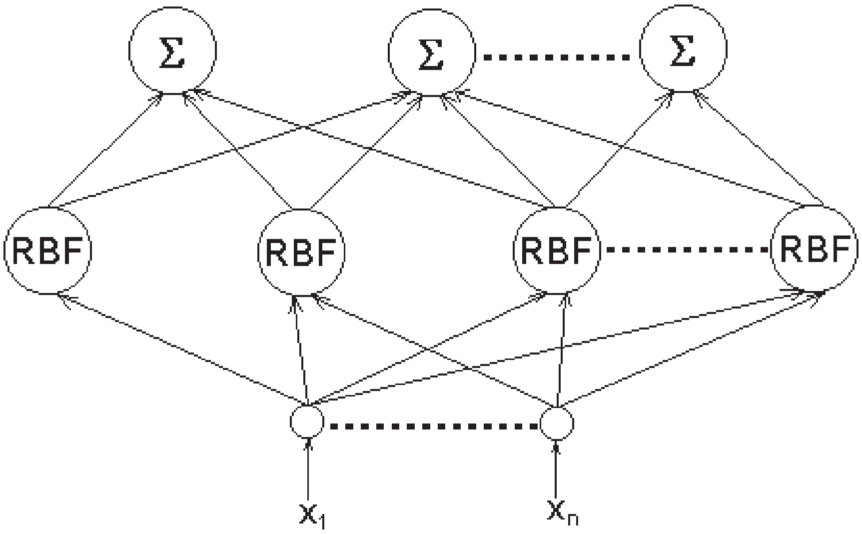
Pro srovnání klasický perceptron:

$$y(\vec{x}) = \sigma\left(\sum_{i=0}^{n} w_i x_i\right), \text{ kde } \sigma(x) = \begin{cases} 0 & z < 0 \\ 1 & z \geq 0 \end{cases} \tag{22}$$

RBF jednotka i perceptron realizují nelineární funkci z $\mathbb{R}^n$ do $\mathbb{R}$. Podíváme-li se však na význam těchto funkcí, vidíme, že Gaussova RBF jednotka má významný výstup pouze v okolí svého středu (lokální jednotka), zatímco perceptron rozděluje vstupní prostor globálně na dva podprostory.

2.2 Model RBF sítě

RBF je třívrstvá dopředná neuronová síť (viz. následující obrázek):



Obrázek 3: model RBF sítě

První vrstva obsahuje n vstupních jednotek, které slouží pouze k předání vstupních hodnot. Druhá - skrytá vrstva obsahuje h RBF jednotek. Poslední výstupní vrstva je lineární a počet jejích jednotek je označen m.

RBF síť realizuje reálnou funkci z $\mathbb{R}^n \to \mathbb{R}^m$, a to tak, že nejprve vrstva RBF jednotek transformuje vstupní vektor $(x_1, \ldots, x_n)$ na vector $(y_1, \ldots, y_n)$ a pak výstupní vrstva transformuje vektor $(y_1, \ldots, y_n)$ na $(f_1, \ldots, f_m)$

$$f_s(\vec{x}) = \sum_{j=1}^{h} w_{sj} g\left(\frac{\|\vec{x} - c_j\|}{b}\right) \tag{23}$$

kde w_{sj} je váha spoje od $j-$té skryté jednotky do $s-$té výstupní jednotky.

Funkce sítě je tedy tvořena ze dvou funkcí, první nelineární funkce z $\mathbb{R}^n$ do $\mathbb{R}^h$ a druhé lineární funkce z $\mathbb{R}^h$ do $\mathbb{R}^m$.

Budeme-li se dívat na síť jako na klasifikátor, můžeme říci, že úkolem první vrstvy je transformovat vstupní vektor tak, aby byl ohodnocen lineární kombinací jeho nových souřadnic.

2.3 Vážená norma

Potenciál RBF jednotky počítáme jako $\frac{\|\vec{x}-\vec{c}\|}{b}$, kde $\|\vec{x} - \vec{c}\|$ je vzdálenost $\vec{x}$ od $\vec{c}$ počítaná dle euklidovské metriky.

Další možností je využití vážené normy, která je definovaná takto:

$$\|\vec{x}\|_c^2 = (\underline{c}\vec{x})^T(\underline{c}\vec{x}) = \vec{x}^T \underline{c}^T \underline{c}\vec{x} \tag{24}$$

kde $\underline{c}$ je matice $n \times n$ a n je dimenze vstupního vektoru. Pokud je $\underline{c}$ rovna jednotkové matici, dostaneme standardní euklidovskou normu.

$$\|\vec{x}\|_c^2 = \|\vec{x}\|^2 \tag{25}$$

Použití diagonální matice odpovídá přiřazení náhodných koeficientů jednotlivým vstupům.

$$\|\vec{x}\|_c^2 = \sum_{k=1}^{n} c_k^2 x_b^2 \tag{26}$$

obecná matice představuje jednoduchou afinní transformaci.

$$\|\vec{x}\|_c^2 = \sum_{k=1}^{n} \sum_{l=1}^{n} a_{kl} x_k x_l \tag{27}$$

kde a_{kl} je $kl-$tý prvek matice $\underline{c^T c}$.

Nahradíme-li euklidovskou metriku právě popsanou váženou normou, RBF síť bude počítat funkci:

$$f_s(\vec{x}) = \sum_{j=1}^{h} w_{sj} g\left(\frac{\|\vec{x} - \vec{c}_j\|c}{b_j}\right) \tag{28}$$

V tomto případě všechny RBF jednotky používají při výpočtu vzdálenosti stejnou matici c, tedy u všech je prováděna stejná transformace.

V obecném případě každé jednotce přidat další parametr c a funkce sítě má pak tvar:

$$f_s(\vec{x}) = \sum_{j=1}^{h} w_{sj} g\left(\frac{\|\vec{x} - \vec{c}_j\|c_j}{b_j}\right) \tag{29}$$

Použití vážené normy je vhodné v případě, kdy jednotlivé složky vstupního vektoru jsou různého charakteru.

2.4 Učení RBF sítí

viz. Backpropagation.

2.5 Regularizace

Často klademe na zobrazení realizované sítí další požadavky. Naučená síť by měla dávat rozumné výsledky i pro vstupy, které nejsou obsaženy v trénovací množině. Také situace, kdy funkce sítě prochází všemi body z trénovací množiny není vždy ideální, neboť v praxi často obsahuje tréningová množina šum.

Lze tedy přidat k chybové funkci další člen E_k:

$$E = E_1 + \lambda E_k \tag{30}$$

E_k je zvolen tak, aby měl velké hodnoty pro funkce s velkou oscilací. Parametr λ určuje míru vlivu tohoto regulačního členu na chybovou funkci. Lze zvolit E_k tak, aby penalizoval funkce s velkou druhou derivací:

$$E_k = \frac{1}{2} \sum_{t=1}^{k} \sum_{s=1}^{m} \sum_{i=1}^{n} \left(\frac{\partial^2 f_s}{\partial x_i^2}\right)^2 \tag{31}$$

3 Seznam použité literatury

1. Ben Krüse,Patrick van der Smagt: *Neural network*

2. Petra Kudova: *Neuronové sítě typu RBF pro analýzu dat*

3. KVASNIČKA, V., BEŇUŠKOVÁ, Ľ., POSPÍCHAL, J., FARKAŠ, I., TIŇO, P., KRÁĽ, A.: *Úvod do teórie neurónových sietí. Bratislava: Iris, 1997*