

Dokumentace k projektu do předmětu SFC

Rozpoznávání znaků pomocí neuronové sítě

Back Propagation

Autor: Bc. Petr Vácha – xvacha03

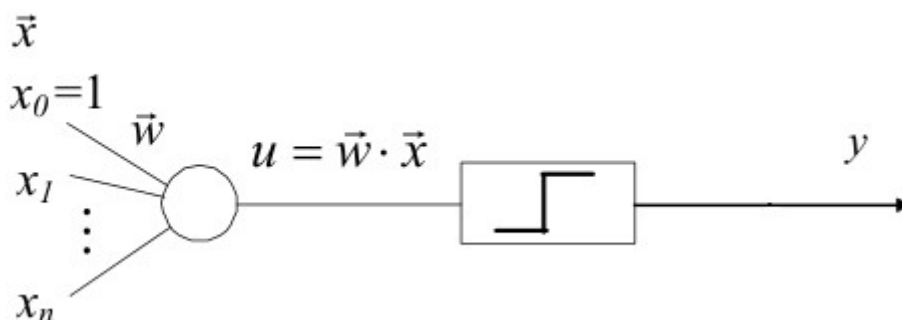
Datum: 17.12.2010

Zadání

Zadání projektu bylo rozpoznávání ručně psaných znaků pomocí neuronové sítě typu Backpropagation. Za ručně psané znaky budou brány znaky, které jsou kresleny pomocí počítačové myši v grafickém editoru, viz dále.

Neuronová síť

Neuronová síť se skládá z umělých neuronů. Každý neuron se skládá z několika vstupů a právě jednoho výstupu. Každému vstupu x_i se přiřazuje váha w_i . Neuron typu perceptron má svou bazovou funkci u , která je dána sumou všech součinu jeho vstupů a příslušných vah. Pomocí aktivační funkce g se spočítá výstupní hodnota y , která je vstupem do další vrstvy.



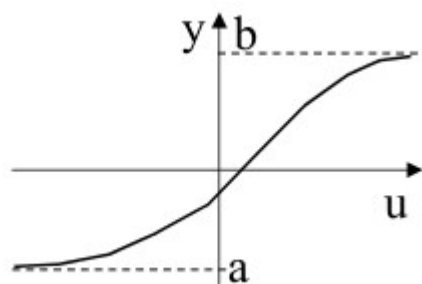
$$y = g(u) = g(f(\vec{x}))$$

Obrázek 1: Neuron

f... bazová funkce (lineární)

g... aktivační funkce (sigmoida)

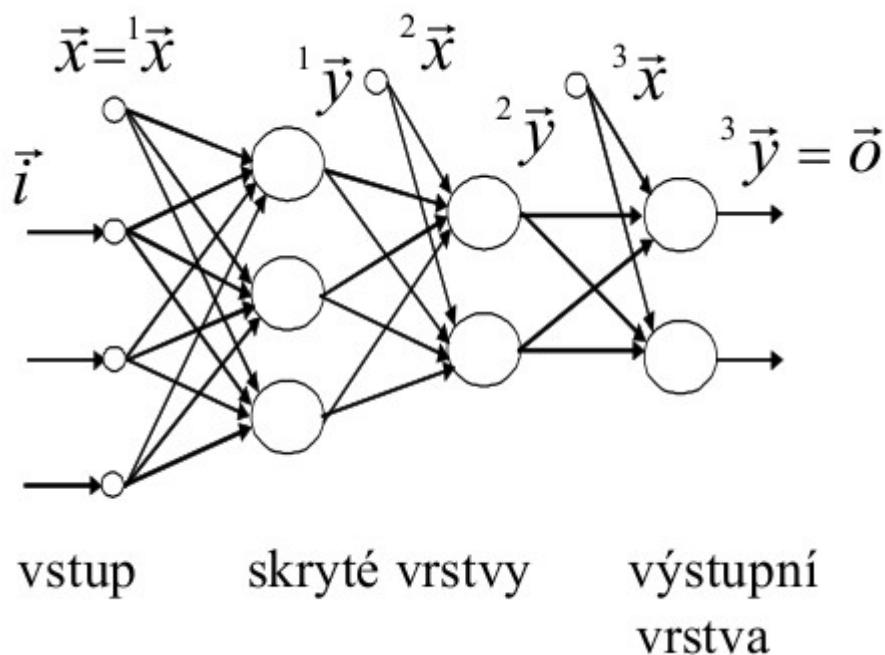
$$y = a + \frac{b-a}{1+e^{(-\lambda u+c)}}$$



Obrázek 2: sigmoidální funkce a příslušná sigmoida

Back Propagation

Back Propagation je jedna z nejznámějších klasických neuronových dopředných sítí. Počet vrstev sítě tohoto typu je větší jak 1. Vrstvy neuronové sítě můžeme rozdělit celkem do tří typů: vstupní, skryté a výstupní. Váhy jsou nastavovány na základě zpětného šíření chyby. Každá vrstva má svůj tzv. prahový vstup x_0 o konstantní hodnotě 1 s přednastavenými vahami na hodnotu 0. Ostatní váhy jsou na začátku inicializovány náhodným číslem od -0.5 do 0.5.



Obrázek 3: Neuronová síť Back Propagation

Implementace

Projekt neuronové sítě je implementován v jazyce Python verze 2.6 s knihovnou Python Imaging Library (PIL – <http://www.pythonware.com/products/pil/>), která slouží pro zpracování obrázků. Projekt je dekomponován do tří souborů. Soubor `picture.py` obsahuje třídu na zpracování vstupních obrázků. Třída obrázků převede na černobílý a metodou `getVector` vrátí jeho reprezentaci ve formě vektoru nul a jedniček. Soubor `perceptron.py` představuje stejnojmennou třídu reprezentující jeden perceptronový neuron. Dále obsahuje metody pro nastavení vstupních hodnot neuronu, nastavení vstupních vah neuronu, výpočet aktivační funkce, výpočet výstupní hodnoty a jiné. Posledním souborem je soubor `NN.py`, který zajišťuje řízení celého programu. Obsahuje implementaci algoritmu Back Propagation.

Vstupem do neuronové sítě je vektor, jehož délka je dána počtem pixelů daného vstupního obrázku. Celá neuronová síť se skládá ze 3 vrstev, čili má pouze jednu vrstvu skrytou.

Vstupní hodnoty neuronové sítě jsou na počítači ručně kreslené čísla od 0 do 9. Z tohoto důvodu má výstupní vrstva 10 neuronů. Každý neuron tak představuje jedno číslo. V praxi tak naučená síť má na výstupu pro číslo dva u třetího výstupu (počítáme od nuly) číslo blíží se 1 a u ostatních výstupu jsou čísla blíží se 0.

Spuštění a interpretace výsledků

Příklad spuštění:

```
python NN.py hand 0.8
```

Příklad výsledku:

```
//spuštění na ručně psané sadě s učící konst. 0.8
```

```
python NN.py hand 0.8
```

```
// první číslo určuje počet iterací
```

```
0 Globální chyba: 0.0
10 Globální chyba: 40.6263766707
25 Globální chyba: 40.5633529187
50 Globální chyba: 40.1246394677
100 Globální chyba: 22.6884738834
```

Výsledek naučené neuronové sítě:

```
// první číslo určuje který soubor byl otestován, první cifra říká o jaké číslo se jednalo. Poslední číslo je výstupní hodnota pro jeho přiřazený neuron. Tento výsledek říká, jestli se síť dokázala naučit na všechny vstupní trénovací množiny. Pokud je u všech řádků OK, tak se dokázala naučit.
```

```
24 OK 0.871467746276
25 OK 0.862301329361
26 OK 0.881960472898
27 OK 0.885401324336
20 OK 0.892361685582
21 OK 0.87817347285
22 OK 0.874873087594
```

```
...
```

```
44 OK 0.872759588937
45 OK 0.872127922354
42 OK 0.882837422376
43 OK 0.870555659905
40 OK 0.865698589227
41 OK 0.866381535096
77 OK 0.893590787154
76 OK 0.87556153774
75 OK 0.904744482603
74 OK 0.904923551928
73 OK 0.907519795537
72 OK 0.908629051623
71 OK 0.904942604713
70 OK 0.905392989137
78 OK 0.899256260123
```

Test neuronové sítě:

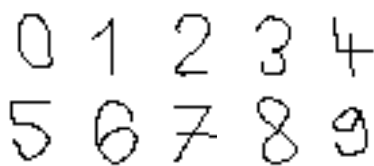
```
// test na jiné sadě čísel, se kterou se neuronová síť v trénovací množině nesetkala
```

```
39 FAIL 0.0240071334312
59 OK 0.674552250959
49 OK 0.883777501747
19 OK 0.89406354611
09 OK 0.83963538185
29 OK 0.390503127879
99 FAIL 0.045018464278
89 OK 0.649410556455
69 FAIL 0.0287259134768
79 OK 0.854294430385
```

Výsledek interpretujeme tak, že se neuronová síť naučila na 70 %. ČÍSLA 3, 9 a 6 nedokázala rozpoznat.

Testování a vyhodnocení

Testování probíhalo na sadě 90 trénovacích obrázků, které autor ručně kreslil v grafickém editoru. Obrázky mají stejné rozměry 20x28 pixelů.



Obrázek 4: Sada testovacích čísel

V případě testování a hledání ideálního nastavení byla největším problémem časová náročnost hledání vah. Pro experimentální hledání ideálního počtu neuronů pro vstupní a skrytou vrstvu byl tento fakt velmi nepohodlný. Kromě hledání již zmíněného ideálního počtu neuronů, se muselo hledat i nastavení učící konstanty μ a postačující ukončovací podmínky ϵ .

Experimentálně nalezené optimální nastavení bylo 6 x 11 x 10 neuronů s koeficientem učení 0.8. Čas vyhodnocení se pohyboval kolem 4 minut.

Výsledky experimentů	
Úspěšnost	Špatně vyhodnocené znaky
70,00%	3,5,6
70,00%	3,9,6
80,00%	3,6
90,00%	3
80,00%	2,6
60,00%	3,9,8,6
80,00%	2,6
70,00%	3,9,6
60,00%	3,0,6,7
60,00%	2,9,8,6
70,00%	3,9,6
40,00%	3,5,0,8,6,7
80,00%	5

Celková úspěšnost na základě tohoto testu je 70 %.

Výsledky úspěšnosti jednotlivých znaků	
0	84,62%
1	100,00%
2	76,92%
3	30,77%
4	100,00%
5	76,92%
6	15,38%
7	84,62%
8	76,92%
9	61,53%

Jednotlivé výsledky nejsou příliš pozitivní. Jako řešení autor navrhuje rychlejší implementaci například pomocí jazyka C, které by přispělo k rychlejšímu hledání lepšího nastavení sítě. Dále větší množství trénovacích dat a určitá segmentace vstupních hodnot, kde by nemusel každý pixel představovat unikátní vstup do neuronové sítě.

Zdroje

- [1] František Vítězslav Zbořil: Studijní materiál k 2. přednášce předmětu SFC, [online], Naposledy změněno 2010
URL https://www.fit.vutbr.cz/study/courses/SFC/private/10sfc_2.pdf
- [2] Principles of training multi-layer neural network using backpropagation [online], Naposledy změněno 1992
URL http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html [online]
- [3] ZDENĚK DOUPOVEC: Bakalářská práce - ROZPOZNÁVÁNÍ ČÍSLIC POMOCÍ NEURONOVÉ SÍTĚ [online], Naposledy změněno 2009
URL <http://www.fit.vutbr.cz/study/DP/rpfile.php?id=8672>