Rozpoznávání tištěných znaků pomocí LVQ sítí

Neuronové sítě 2006/2007

Jan Hroník, Pavel Krč

Zadání problému

- Rozpoznávání tištěných znaků pomocí jednoduchých statistik získaných z rastrové podoby znaku (např. po skenování papírové předlohy)
- Neřešíme předzpracování, zajímá nás pouze proces identifikace znaku
  - Vstup: hodnoty statistik
  - Výstup: přiřazení k jedné z 26 tříd (velká písmena latinské abecedy)

Návaznost

- Stejný problém řešili v roce 2005 Rudolf Kadlec a Jiří Šejnoha
- Vstupní data pocházejí z databáze *Delve* z University of Toronto (http://www.cs.toronto.edu/~delve/)
  - Sbírka vstupních dat pro různé metody učení
  - My řešíme problém *letter*

Vstupní data

- Celkem 20 000 vzorků písmen
- Přibližně rovnoměrné rozdělení znaků A – Z
- Písmena jsou tištěna 20 různými fonty, přidán šum
- Každý vzorek se skládá z 16 statistických charakteristik a určení správné třídy

Charakteristiky znaku

1. x-box - horizontal position of box (integer)
2. y-box - vertical position of box (integer)
3. width - width of box (integer)
4. high - height of box (integer)
5. onpix - total # on pixels (integer)
6. x-bar - mean x of on pixels in box (integer)
7. y-bar - mean y of on pixels in box (integer)
8. x2bar - mean x variance (integer)
9. y2bar - mean y variance (integer)
10. xybar - mean x y correlation (integer)
11. x2ybr - mean of x * x * y (integer)
12. xy2br - mean of x * y * y (integer)
13. x-ege - mean edge count left to right (integer)
14. xegvy - correlation of x-ege with y (integer)
15. y-ege - mean edge count bottom to top (integer)
16. yegvx - correlation of y-ege with x (integer)

Přístup k řešení

- Kadlec & Šejnoha řešili problém pomocí *dopředných vrstevnatých sítí* učením algoritmem *backpropagation*
- Kvůli časovým a paměťovým nárokům nakonec rozpoznávali jen polovinu (13) tříd
- Dosáhli nejlepší účinnosti **90,5%** při konfiguraci 16–28–13

Přístup k řešení (2)

- My jsme zkusili řešit problém pomocí *Kohonenových sítí* učených s učitelem pomocí algoritmu *LVQ*
- Rozpoznávali jsme všech 26 tříd na různě velkých vzorcích dat i na celé vstupní množině

Postup trénování

- Síť jsme ladili v prostředí *MATLAB* s využitím *Neural network toolboxu* pomocí naprogramovaného skriptu
- Vstupní data jsme rozdělili v poměru 65:35 na *trénovací* a *ověřovací*
- Bylo třeba ladit několik parametrů v široké škále hodnot:
  - Typ algoritmu (LVQ1 vs. LVQ2)
  - Počet neuronů (26 – cca. 800)
  - Koeficient učení (*learning rate*)
  - Počet epoch učení
- Test výhodnosti aplikace PCA analýzy

Postup trénování (2)

- Z důvodu časové náročnosti jsme trénovali na více strojích zároveň
- Trénování probíhalo jak dávkově (pevný počet epoch), tak interaktivně (sledováním průběhu, čekáním na stabilizaci)
- Na celých vstupních datech trvalo jedno trénování (s pevnými parametry) několik hodin
- Používali jsme i menší vzorky vstupních dat
  - Značně rychlejší učení
  - Většina parametrů se téměř shodovala
  - Větší tendence k přeučení

Ladění parametrů
Typ algoritmu

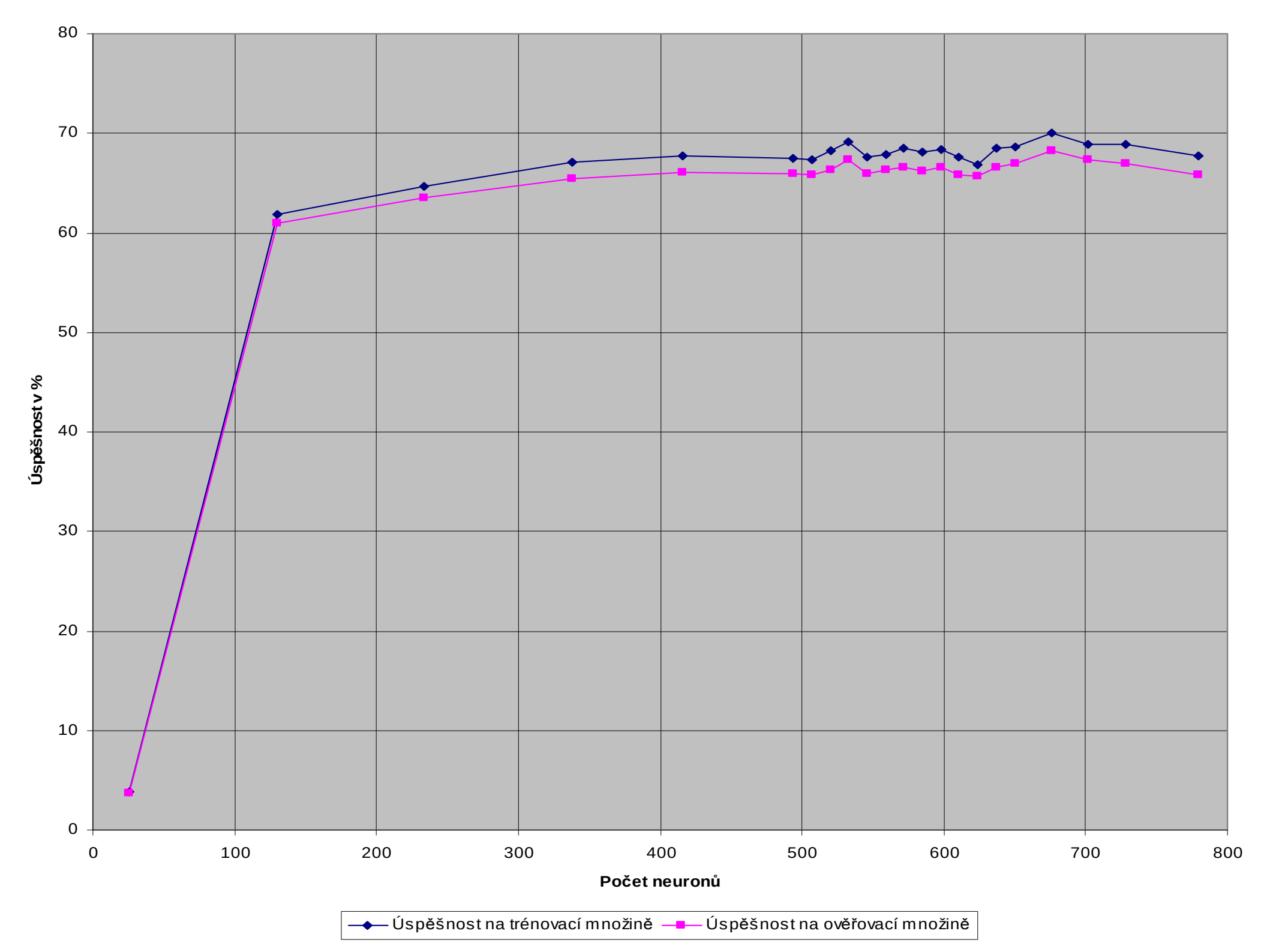
- V několika různých konfiguracích jsme zkoušeli algoritmus *LVQ2*
- Většinou byl znatelně horší než *LVQ1*, někdy se vůbec nezačal učit
- Proto jsme dále používali pouze *LVQ1*

Ladění parametrů
PCA analýza

- Vzhledem k povaze vstupních dat se nabízelo použití PCA analýzy
- Použili jsme PCA analýzu z toolboxu
- Po vyladění parametrů analýzy se díky ní zrychlilo učení o jednotky až desítky procent, ale pouze na větších množstvích dat

Ladění parametrů
Počet neuronů

- Nabízelo se několik rozumných odhadů pro počet neuronů
  - 26 výstupních tříd (předpokládá se podobnost fontů)
  - 26 tříd * 20 fontů = 520 shluků
  - Případně i více kvůli snadnějšímu učení
- Zkoušeli jsme i hodnoty mezi odhady


80
70
60
50
40
30
20
10
0
Úspěšnost v %
0
100
200
300
400
500
600
700
800
Počet neuronů
Úspěšnost na trénovací množině
Úspěšnost na ověřovací množině

Ladění parametrů
Koeficient učení

- Koeficient učení má přímou souvislost s ideálním počtem epoch a s kvalitou výsledku
- Na plných datech jsme nalezli ideální hodnotu **0,01**
  - Vystačí s rozumným počtem epoch (40)
  - Další snižování nepřináší znatelně lepší výsledky
  - Vyšší hodnoty dávají znatelně horší výsledky

Ladění parametrů

Závislost *learning rate* vs. počet epoch (méně dat)

Koeficient učení	**0,2**	**0,1**	**0,01**	**0,001**
Počet epoch pro stabilizaci	100	120	250	1700
Úspěšnost na trénovací množině	69%	75%	78%	78%
Úspěšnost na ověřovací množině	14%	34%	37%	31%

Celkový výsledek

- Nejlepší dosažená úspěšnost: **70,33%** na ověřovacích datech
  - Kompletní vstupní množina
  - LVQ1, PCA analýza
  - **676** neuronů
  - Koeficient učení **0,01**
  - **297** epoch, podobné výsledky již od 40 epoch, stablilizace na cca. 120 epochách
  - 14 hodin učení

Závěr

- Celkový čistý výpočetní čas *cca. 50 hodin*
- Nedosáhli jsme sice tak dobrých výsledků jako Kadlec & Šejnoha, zato však na všech třídách
- Ověřili jsme použitelnost LVQ sítí pro daný typ úlohy