Klasifikace znaků dle statistických parametrů jejich obrazů

Martin Tůma

***Abstrakt*— Semestrální práce se zabývá možnostmi neuronových sítí v oblasti rozpoznávání obrazů písem dle atributů získaných statistickým zpracováním obrazových bodů jednotlivých obrazů.**

I. Zadání

Úkolem je vytvořit neuronovou síť(sítě), která bude na základě atributů jednotlivých písmen určovat, o jaké písmeno se jedná. Jedná se tedy o klasickou klasifikační úlohu. Vstupem je 16 parametrů u každého písmene, výstupem by pak mělo být jednoznačně určené písmeno.

II. Úvod

Vstupní atributy znaků byly získány statistickým převodem (pomocí statistických momentů a počtů hran) z černobílých obrazů písem složených ze čtvercových pixelů. Z každého z celkem 20 použitých fontů bylo do datového souboru náhodně vybráno takové množství písmen, aby v celkovém počtu (20 000) datových vzorků bylo rozložení jednotlivých písmen přibližně stejné. Množina vzorků je tedy pro daný problém poměrně reprezentativní.

A. Atributy písem

–	capital letter	(A-Z)
1.	horizontal position of box	(0-15)
2.	vertical position of box	(0-15)
3.	width of box	(0-15)
4.	height of box	(0-15)
5.	total # on pixels	(0-15)
6.	mean x of on pixels in box	(0-15)
7.	mean y of on pixels in box	(0-15)
8.	mean x variance	(0-15)
9.	mean y variance	(0-15)
10.	mean x y correlation	(0-15)
11.	mean of x * x * y	(0-15)
12.	mean of x * y * y	(0-15)
13.	mean edge count left to right	(0-15)
14.	correlation of x-ege with y	(0-15)
15.	mean edge count bottom to top	(0-15)
16.	correlation of y-ege with x	(0-15)

III. Cíl práce

Cílem práce je dosáhnout použitelné úrovně rozpoznávání jednotlivých písmen. Při výzkumu[1], při kterém data „vznikla“ bylo dosaženo 80% úspěšnosti klasifikace. Úspěchem by tedy bylo se této hodnotě přiblížit, v ideálním případě ji překročit.

[1]P. W. Frey and D. J. Slate (Machine Learning Vol 6 #2 March 91): „Letter Recognition Using Holland-style Adaptive Classifiers“

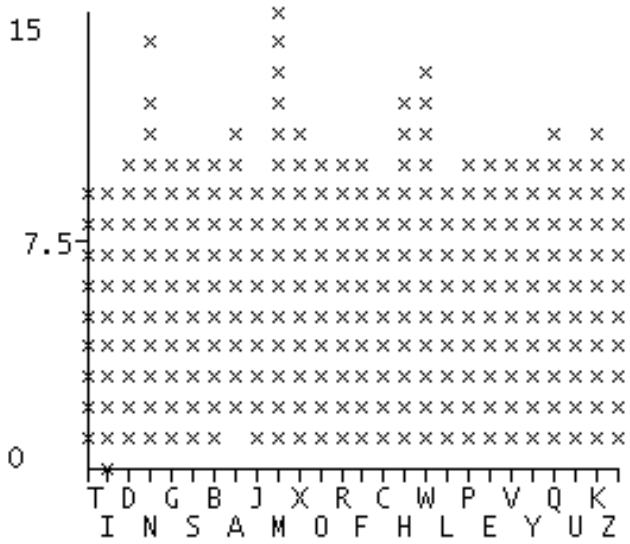

Fig. 1. Rozložení hodnot atributu č. 3 - šířka boxu.

IV. Experimenty

A. Rozbor dat

Rozbor dat provedený pomocí programu WEKA[2] byl zaměřen především na zjištění „významnosti“ jednotlivých atributů. Z provedených vizualizací vyplývá, že prakticky všechny vstupní atributy jsou si co do významnosti rovnocené. Z grafů je patrné, že jednotlivé množiny mají jen slabou rozlišovací schopnost, tzn. z hodnot jednoho atributu lze jen těžko usuzovat, o jaké písmeno by se mohlo jednat – jeden atribut dobře identifikuje pouze pár znaků. Jednotlivé množiny jsou nicméně v tomto ohledu navzájem odlišné – znaky, které jednotlivé atributy (alespoň „vizuálně“) identifikují, jsou pro různé atributy různé. Ukázkové rozložení hodnot atributu pro atribut č3, šířka boxu, je zobrazeno na obrázku 1.

Na základě provedených „měření“ se tedy lze domnívat, že klasifikace podle daných atributů bude sice velmi obtížná, nicméně ne nemožná.

B. Jednoduché neuronové sítě

Jako první metoda, jak klasifikovat jednotlivá písmena byla navržena jednoduchá neuronová síť typu *backpropagation*, se 16 vstupy a jedním výstupem. Tato síť měla 2 skryté vrstvy a celková „topologie“ sítě byla $\{16, 7, 3, 1\}$. Myšlenkou bylo jednotlivá písmena rovnoměrně rozdelit v určitém intervalu (vzhledem k použitému software a přenosovým funkcím v něm použitých neuronů byl zvolen interval 0-1) a podle toho, do

[2]http://www.cs.waikato.ac.nz/ml/weka/

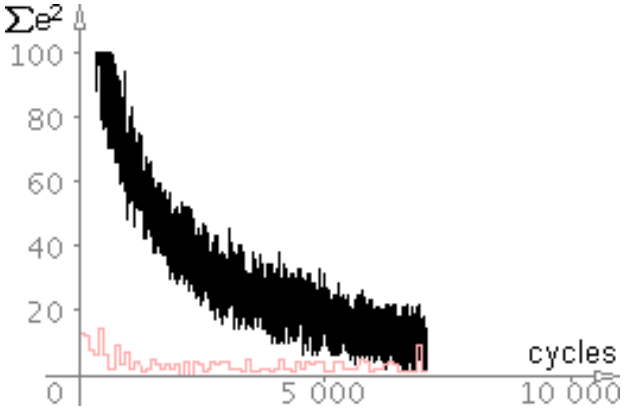

Fig. 2. Průběh učení neuronové sítě – znak H

jakého intervalu spadá výsledná hodnota vrácená neuronovou sítí určit o jaké písmeno se jedná. Již při učení sítě se však ukázalo, že závislost mezi vstupnímy daty a výstupním symbolem je poměrně složitá – chyba učení dosahovala extrémně vysokých hodnot. Výsledky na testovací množině pak byly zcela nepoužitelné (úspěšnost klasifikace v jednotkách procent).

Druhou „triviální“ testovanou metodou bylo zkonstruování sítě s 26 výstupy, kde každému výstupu odpovídal jeden znak abecedy. Podle hodnoty výstupu (0 nebo 1) se pak určovalo, jestli se jedná o dané písmeno. Ani v tomto případě však nebyly výsledky prakticky použitelné – úspěšnost klasifikace se prakticky nelišila od první metody.

C. Hlavní experiment

Jelikož se ukázalo, že jako celek je pro standartní backpropagation síť klasifikace do takto velkého počtu tříd velice složitá, bylo nutné problém rozdělit na několik menších „podproblémů“ a jejich výsledky následně skombinovat.

Pro každé jednotlivé písmeno byla vytvořena vlastní neuronová síť, jejímž úkolem je rozhodovat pouze zda-li se jedná o dané písmeno, či nikoliv. Jednotlivé sítě jsou opět standartní, dopředné, plně propojené backpropagation sítě se strukturou $\{16, 7, 3, 1\}$. U výstupní vrstvy byla navíc použita výstupní funkce *Out_Treshold05* pro 50% práh. Hodnoty < 0.5 jsou tedy klasifikovány jako 0 a hodnoty ≥ 0.5 jako 1. Jako přenosové funkce neuronů vnitřních vrstev byly zvoleny standartní sigmoidy (*Act_Logistic*).

Učící a validační množiny byly zkonstruovány tak, aby obsahovaly přibližně stejný počet hledaných písmen a písmen jiných. Tyto množiny byly získány z prvních 16000 vzorků, zbylých 4000 vzorků pak, stejně jako v původním výzkumu, tvořilo testovací množinu.

Učení sítí probíhalo klasickou backpropagation metodou a pro naučení sítě bylo v průměru zapotřebí okolo 5000 učících cyklů. Obrázek 2 ukazuje typický průběh učení sítě.

D. Získaná data

Výsledky jednotlivých neuronových sítí na testovací množině (4000 prvků) jsou uvedeny v následující tabulce:

	počet chyb	**úspěšnost**	**chyba v %**
A	188	0,953	4,7
B	209	0,948	5,2
C	163	0,959	4,1
D	131	0,967	3,3
E	193	0,952	4,8
F	210	0,947	5,3
G	169	0,958	4,2
H	164	0,959	4,1
I	165	0,959	4,1
J	133	0,967	3,3
K	207	0,948	5,2
L	196	0,951	4,1
M	77	0,981	1,9
N	148	0,963	3,7
O	276	0,931	6,9
P	136	0,966	3,4
Q	158	0,960	4,0
R	248	0,938	6,2
S	168	0,958	4,2
T	111	0,972	2,8
U	122	0,969	3,1
V	138	0,965	3,5
W	93	0,977	2,3
X	155	0,961	3,9
Y	135	0,966	3,4
Z	127	0,968	3,2
ϕ	–	0,959	4,1

Celková úspěšnost byla určena jako poměr jednoznačně klasifikovaných znaků ku celkovému počtu znaků. Dále byl určen počet obou možných druhů chyb – *false positive* (nepravdivé označení znaku za určité písmeno) a *false negative* (neoznačení znaku jako určitého písmena).

	počet chyb	**úspěšnost**	**chyba v %**
chyb celkem	2277	0,431	56,9
false positives	2267	–	–
false negatives	118	–	–

Poznámka: Součet false positives a false negatives nedávající v součtu celkový počet chyb není chybou, obě množiny jen *nejsou* disjunktní.

V. Diskuse

Z naměřených výsledků by se mohlo zdát, že celková úspěšnost metody je „pouze“ 43%, nicméně není tomu tak. Pouhých 5% chyb typu *false negative* dává možnosti jak úspěšnost dále zvýšit. Znamená to totiž, že prakticky všechny chyby jsou způsobeny tím, že je „za své“ prohlásí více sítí zároveň. Toto však znamená, že pouhou náhodnou volbou jedné z možností je možné celkovou úspěšnost metody výrazně zvýšit! U množiny *false positive* výsledků byl proto orientačně vyhodnocen průměrný počet možností výběru na záznam a výsledná hodnota je **2,809**. Za předpokladu rovnoměrného rozložení písmen v množině tedy náhodným výběrem získáme dalších $T\%$ úspěšnosti.

$$T = \frac{E_t - E_{fn}}{q} = \frac{0,569 - 0,029}{2,809} = 19,2\%$$

Kde E_t je celková chyba, E_{fn} *false negatives* chyba (vzhledem k počtu testovacích prvků) a q průměrný počet možností výběru ve *false positives* množině. Celková úspěšnost metody tak bude přibližně **62%** a to ještě vůbec není brána v potaz nějaká sofistikovanější metoda na výběr správného znaku (Statistické rozložení písmen v jazyce, další neuronová síť…).

Dosažený výsledek klasifikace je o 18% horší, než byl výsledek původního výzkumu nad danými daty, je tedy zpracování pomocí backpropagation neuronových sítí pro danou úlohu nevhodné? Není, neboť v „implementaci“ jsou ještě podstatné rezervy. Například velký počet *false positive* chyb je s největší pravděpodobností způsoben výběrem trénovacích množiny. Ty byly z časových důvodů[3] volené výběrem všech daných znaků a odpovídajícího počtu znaků jiných z celkové trénovací množiny. Průměrně tak byly jednotlivé sítě učeny necelou desetinou možných vzorků. V případě, že by ke tvorbě trénovacích množin byla zvolena metoda „seznamující“ síť s více „jinými“ znaky, byl by počet *false positives* pravděpodobně výrazně nižší.

Pro dosažení kýžené 80% úspěšnosti klasifikace by dle naměřených hodnot a teorie pravděpodobnosti bylo potřeba u jednotlivých sítí dosahovat chyby menší než 98% procent, což je o pouhá 3% více, než bylo dosaženo!

VI. Závěr

Provedené experimenty ukázaly, že backpropagation neuronová síť je přinejmenším použitelným nástrojem pro danou klasifikaci. Podařilo se dosáhnout **62% úspěšnosti klasifikace**, přičemž byly nastíněny postupy, kterými by měla jít tato úspěšnost dále zvyšovat. Jedná se konkrétně o:

- Zlepšení výběru trénovacích množin s důrazem na větší počet „nehledaných“ znaků.
- Nasazení sofistikovanější metody pro vyhodnocení celkového výstupu klasifikátoru z výstupů jednotlivých neuronových sítí.

VII. Další materiály

Veškerá získaná data, stejně tak jako kompletní vstupní množinu dat, transformační skripty i jednotlivé neuronové sítě (vše ve formátu pro JavaNNS[4]) použité při zpracování úlohy naleznete na *http://tumic.wz.cz/fel/#36NAN*.

Literatura

[1] Šnorek M., *Neuronové sítě a neuropočítaře*, skriptum ČVUT, 2002.
[2] Wikipedia, the free encyclopedia, *http://wikipedia.org*

[3] Naučení jednotlivých sítí je, poměrně časově náročná činnost. Při 26 sítích se již jedná o nezanedbatelný časový interval. (Na PC s CPU Duron 900MHz trvalo kompletní naučení všech sítí okolo 10 hodin)

[4] http://www-ra.informatik.uni-tuebingen.de/software/JavaNNS