**Projekt 2 – Viaсvrstvový perceptrón**

**Úvod**

Našou úlohou bolo naprogramovať viacvrstvový perceptrón schopný rozdeliť dvojrozmerné dáta do troch tried. Ako predloha nám poslúžili slidy k prednáškam a cvičenia, kde sa riešila podobná úloha na štvorrozmerných dátach.

**Algoritmus**

Majme neurónovú sieť obsahujúcu jednu a viac skrytých vrstiev neurónov. **Back-propagation** algoritmus má za cieľ preniesť chybu z výstupnej vrstvy a postupne ju dostať cez skryté vrstvy od tej najviac vonkajšej po tú najvnútornejšiu, aby sa pomocou nej neuróny na jednotlivých vrstvách vytrénovali.

Prvý fáza je **forward pass** – počas nej rátame zaradom výstupy 1., 2., ... až poslednej, výstupnej vrstvy. Výstup vonkajšej vrstvy porovnáme s očakávaným výstupom a aplikujeme na trénovanie pre jednotlivé neuróny vo vrstve podobné pravidlo ako pre jednoduchý perceptrón, a túto chybu postupne ďalej propagujeme do vnútorných vrstiev, ktoré sa ďalej tiež trénujú.

Aby sa sieť rýchlejšie a kvalitnejšie natrénovala, tak sme vstupné dáta **normalizovali**, t.j. pre každý rozmer dát sme odčítali jeho priemernú hodnotu a predelili sme ho štandardnou odchýlkou**.** Priemer, resp. štandardnú odchýlku sme počítali pred samotným trénovaním nad všetkými trénovacími dátami (množina 800 vstupov).

Ďalej sme pri trénovaní aplikovali **momentum**, t.j. na aktuálnu zmenu váh mala s určitým podielom vplyv aj zmena váh v predošlej epoche.

Takisto sme pri trénovaní využili **k-fold cross validation**, ktorý spočíva v tom, že si vstupnú množinu dát rozdelíme v určitom pomere (v našom prípade 7:1) na estimačnú a validačnú podmnožinu v našom prípade ôsmimi rôznymi spôsobmi, a pre každý spôsob rozdelenia jednotlivé modely trénujeme 8-krát nezávisle na každej estimačnej podmnožine a testujeme na validačnej podmnožine. Po odskúšaní všetkých možných modelov vyberieme ten, ktorý dosiahol v priemere najmenšiu chybu na validačných podmnožinách, natrénujeme ho na všetkých vstupných dátach a otestujeme na testovacích. Trénovali sme fixný počet iterácii a vždy sme vybrali tie váhy, s ktorými sme dosiahli najmenšiu chybu na estimačnej množine.

Ako **aktivačnú funkciu** sme použili klasický sigmoid.

**Hľadanie optimálneho modelu**

Naším cieľom bolo optimalizovať voľné parametre neurónovej siete, t.j. počet neurónov v skrytej vrstve, rýchlosť učenia (alfa) a momentum. Každý model sme trénovali na 1000 iterácií s použitím k-fold cross validation na podmnožinách 2d.trn.dat. Alfy sme skúšali v rozsahu 0.05-0.3, momentum v rozsahu 0.0-0.5 a počet neurónov v skrytej vrstve bol 16 alebo 24. Vybrali sme model vykazujúci najmenšiu priemernú chybu na validačných dátach.

**Tabuľka chýb skúšaných modelov**

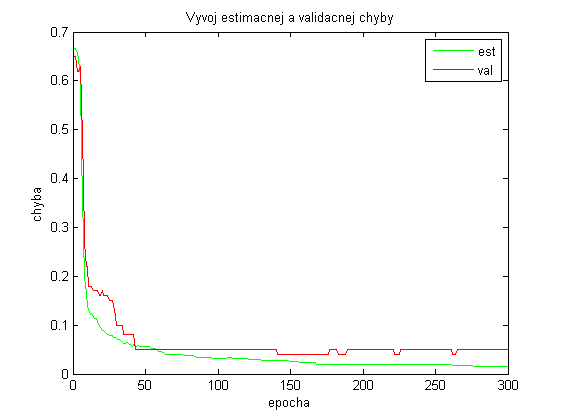
Pre každý model sme uviedli estimačnú chybu a validačnú chybu v percentách ako priemer z chýb na validačných resp. estimačných dátach pri k-fold cross validation (využijúc dáta z 2d.trn.dat). Za najlepší sme pokladali model s najmenšou priemernou validačnou chybou a tento sme následne aj zvolili na otestovanie na testovacích dátach (2d.tst.dat). Označený je červenou farbou.

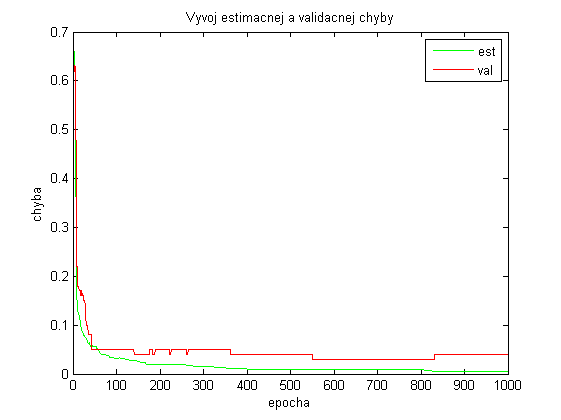
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **alpha** | **momentum** | **veľ. skr. vrstvy** | **est. chyba [%]** | **val. chyba [%]** |
| 0,05 | 0 | 16 | 2,00000 | 1,91071 |
| 0,05 | 0 | 24 | 2,12500 | 2,76786 |
| 0,05 | 0,1 | 16 | 2,50000 | 2,80357 |
| 0,05 | 0,1 | 24 | 2,75000 | 2,42857 |
| 0,05 | 0,2 | 16 | 2,25000 | 2,51786 |
| 0,05 | 0,2 | 24 | 2,37500 | 2,25000 |
| 0,05 | 0,3 | 16 | 3,00000 | 4,76786 |
| 0,05 | 0,3 | 24 | 2,75000 | 2,76786 |
| 0,05 | 0,4 | 16 | 1,87500 | 1,75000 |
| 0,05 | 0,4 | 24 | 1,87500 | 3,30357 |
| 0,05 | 0,5 | 16 | 2,37500 | 2,26786 |
| 0,05 | 0,5 | 24 | 2,50000 | 2,67857 |
| 0,1 | 0 | 16 | 2,37500 | 2,94643 |
| 0,1 | 0 | 24 | 2,12500 | 2,33929 |
| 0,1 | 0,1 | 16 | 2,25000 | 2,21429 |
| 0,1 | 0,1 | 24 | 2,25000 | 2,12500 |
| 0,1 | 0,2 | 16 | 2,00000 | 2,91071 |
| 0,1 | 0,2 | 24 | 2,62500 | 1,98214 |
| 0,1 | 0,3 | 16 | 2,37500 | 2,71429 |
| 0,1 | 0,3 | 24 | 2,62500 | 2,66071 |
| 0,1 | 0,4 | 16 | 2,75000 | 2,57143 |
| 0,1 | 0,4 | 24 | 2,62500 | 2,98214 |
| 0,1 | 0,5 | 16 | 2,62500 | 2,60714 |
| 0,1 | 0,5 | 24 | 1,75000 | 2,64286 |
| 0,15 | 0 | 16 | 2,50000 | 2,89286 |
| 0,15 | 0 | 24 | 2,37500 | 3,75000 |
| 0,15 | 0,1 | 16 | 2,50000 | 3,66071 |
| 0,15 | 0,1 | 24 | 2,37500 | 3,05357 |
| 0,15 | 0,2 | 16 | 2,12500 | 2,41071 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,15 | 0,2 | 24 | 2,75000 | 3,21429 |
| 0,15 | 0,3 | 16 | 1,50000 | 1,94643 |
| 0,15 | 0,3 | 24 | 2,25000 | 2,19643 |
| 0,15 | 0,4 | 16 | 1,87500 | 2,50000 |
| 0,15 | 0,4 | 24 | 1,62500 | 1,46429 |
| 0,15 | 0,5 | 16 | 2,00000 | 2,19643 |
| 0,15 | 0,5 | 24 | 1,75000 | 2,87500 |
| 0,2 | 0 | 16 | 2,00000 | 2,69643 |
| 0,2 | 0 | 24 | 1,87500 | 2,39286 |
| 0,2 | 0,1 | 16 | 2,50000 | 1,75000 |
| 0,2 | 0,1 | 24 | 2,00000 | 2,55357 |
| 0,2 | 0,2 | 16 | 2,25000 | 2,58929 |
| 0,2 | 0,2 | 24 | 2,50000 | 2,60714 |
| 0,2 | 0,3 | 16 | 2,00000 | 1,83929 |
| 0,2 | 0,3 | 24 | 1,75000 | 2,03571 |
| 0,2 | 0,4 | 16 | 2,75000 | 2,62500 |
| 0,2 | 0,4 | 24 | 2,00000 | 2,25000 |
| 0,2 | 0,5 | 16 | 1,75000 | 2,28571 |
| 0,2 | 0,5 | 24 | 2,25000 | 2,57143 |
| 0,25 | 0 | 16 | 2,25000 | 2,08929 |
| 0,25 | 0 | 24 | 1,75000 | 2,08929 |
| 0,25 | 0,1 | 16 | 2,50000 | 2,67857 |
| 0,25 | 0,1 | 24 | 2,12500 | 3,17857 |
| 0,25 | 0,2 | 16 | 2,25000 | 2,00000 |
| 0,25 | 0,2 | 24 | 2,50000 | 2,17857 |
| 0,25 | 0,3 | 16 | 2,12500 | 2,00000 |
| 0,25 | 0,3 | 24 | 1,87500 | 2,53571 |
| 0,25 | 0,4 | 16 | 2,00000 | 2,41071 |
| 0,25 | 0,4 | 24 | 2,62500 | 2,78571 |
| 0,25 | 0,5 | 16 | 2,00000 | 1,85714 |
| 0,25 | 0,5 | 24 | 2,25000 | 2,39286 |
| 0,3 | 0 | 16 | 2,25000 | 2,14286 |
| 0,3 | 0 | 24 | 2,12500 | 2,10714 |
| 0,3 | 0,1 | 16 | 2,12500 | 2,39286 |
| 0,3 | 0,1 | 24 | 2,37500 | 1,55357 |
| 0,3 | 0,2 | 16 | 2,25000 | 2,66071 |
| 0,3 | 0,2 | 24 | 2,00000 | 2,82143 |
| 0,3 | 0,3 | 16 | 2,37500 | 2,62500 |
| 0,3 | 0,3 | 24 | 2,62500 | 2,89286 |
| 0,3 | 0,4 | 16 | 2,25000 | 2,44643 |
| 0,3 | 0,4 | 24 | 2,37500 | 2,71429 |
| 0,3 | 0,5 | 16 | 2,37500 | 2,91071 |
| 0,3 | 0,5 | 24 | 2,50000 | 2,16071 |

Vyhral model s alfou 0.15, momentom 0.4 a 24 neurónmi v skrytej vrstve. Na validačných množinách mal v priemere najnižšiu chybu, a to len 1.55%. Pozrime sa, ako sa správal na testovacej množine dát (2d.tst.dat).

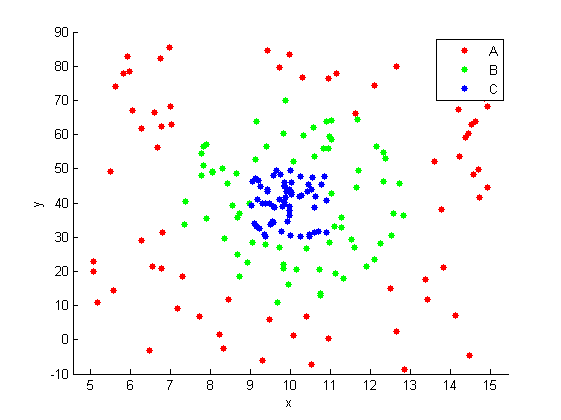
**Vývoj validačnej a estimačnej chyby na najlepšom modeli**

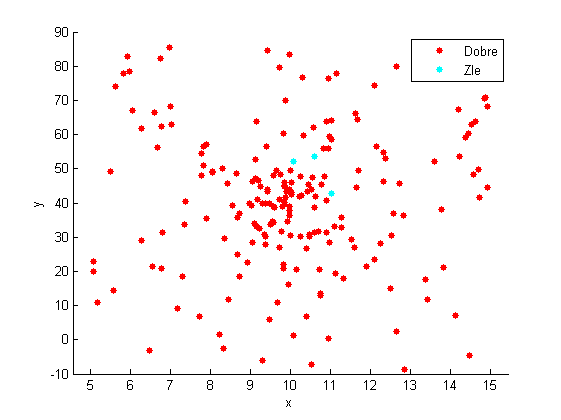
Vybrali sme jeden náhodný trénovací a testovací set spomedzi ôsmich, na ktoré sme rozdelili 2d.trn.dat. Na prvom obrázku vidíme vývoj estimačnej a validačnej chyby počas prvých 300 epôch, na druhom vidíme výsledok na tom istom modeli po 1000 epochách.



Vidno, že obe chyby najprudšie klesali okolo prvých 50 iterácii a potom klesali už len pozvoľna, pričom okolo iterácie 800 môžeme začať badať pravdepodobne prejavy overfittingu na estimačnej množine, čo má za následok postupné zhoršovanie výsledkov na validačnej množine. Na otestovanie na „ostrých“ dátach (2d.tst.dat) sme zvolili potom váhy, ktoré počas trénovania vykázali najmenšiu validačnú chybu.

**Výstupy vybraného modelu znázornené v 2D priestore**

Na prvom obrázku vidno distribúciu dát na testovacej množine (2d.tst.dat), na druhom vidíme chyby klasifikácie na najlepšom modeli po natrénovani pomocou náhodnej estimačnej a validačnej množiny:****

****

**Confussion matrix na testovacích dátach**

Riadky reprezentujú požadované výsledky, stĺpce výsledky vypočítané v najlepšom modeli. Vrchná matica reprezentuje nominálne hodnoty, spodná percentá.

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | C |
| A | 67 | 0 | 0 |
| B | 0 | 68 | 3 |
| C | 0 | 0 | 62 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | C |
| A | 100 | 0 | 0 |
| B | 0 | 95,7746 | 4,2253 |
| C | 0 | 0 | 100 |

**Záver**

Vyskúšali sme rôzne modely viacvrtstvového perceptrónu na klasifikáciu 2D dát. Ako najdôležitejšia úprava na získanie kvalitných výsledkov sa ukázala byť normalizácia dát a hľadanie optimálneho modelu siete pomocou k-cross validácie. Optimálnou konfiguráciou sa ukázala byť alfa=0.15, momentum=0.4 a počet neurónov v skrytej vrstve 24. Po natrénovaní na 2d.trn.dat sa nám podarilo dosiahnuť celkovú chybu klasifikácie na testovacích dátach na úrovni 1.5%.