

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Soft Computing – projekt

Neurónová sieť Backpropagation

1 Backpropagation Networks

Táto sieť pozostáva z neurónov a spojení medzi nimi, čo imituje funkcionality ľudského mozgu. Jednotlivé neuróny sú ukladané do vrstiev a interagujú medzi sebou prostredníctvom spojení. Každé spojenie má svoju váhu a spája výstupy neurónov v jednej vrstve na vstup neurónov v nasledujúcej vrstve. Táto sieť je dopredná, teda nedovoľuje spätné spojenia neurónov.

Sieť sa učí nastavovaním váh podľa daného príkladu - ide teda o sieť ktorá sa učí pod dohľadom. Sieť je predstavený príklad z tréningovej množiny a ona upraví svoje váhy tak aby sa jej výstup priblížil výstupu z príkladu. Výstup siete je výsledkom propagácie výstupov jednotlivých neurónov dopredu - feedforward algorithm. Úprava váh zase prebieha v spätnom prechode ktorý propaguje chybu - backpropagation algorithm.

1.1 Forward Pass

Táto časť algoritmu má za úlohu na základe aktuálneho vstupu a váh vyprodukovať výsledok podobný tomu z tréningovej množiny počas učenia, prípadne výstup počas testovania. Forward Pass prebieha nastavovaním dvoch parametrov a to najskôr celkový vstup (Net Input) a výstup alebo aktivácia.

Celkový vstup

Pre každý neurón skrytej a výstupnej vrstvy je daný jeho celkový vstup na základe vzťahu:

$$net_j = \sum_{i=0}^{n_L} (w_{ji-1} * out_{i-1})$$

kde n_L značí počet neurónov vo vrstve

Aktivácia

Výstup pre neuróny vstupnej vrstvy je daný na základe vstupného vektoru a výstup pre neuróny ostatných vrstiev na základe vzťahu:

$$out_j = \varphi(net_j)$$

1.2 Backward Pass

Úlohou tohto algoritmu je nastaviť vnútorné váhy siete tak aby reagovala na aktuálny vstup a výstup počas učenia. Pozostáva z spätného priechodu siete a výpočtom Deltý pre všetky uzly a následným výpočtom zmeny váh. Zmeny váh za účelom minimalizácie výslednej chyby sú aplikované na základe negatívneho gradientu. Pre tento výpočet je potrebné mať derivovateľnú aktivačnú funkciu.

Delta výpočet

Pre rozlíšenie vplyvu jednotlivých váh na celkovú výslednú chybu je potrebné zderivovať výslednú chybu podľa danej váhy. Po aplikácii parciálnych derivácií je možné dostať tieto algoritmické vzťahy: Výstupná vrstva:

$$\delta_{O_i}(target_i - out_i)out_i(1 - out_i)$$

Skryté vrstvy:

$$\delta_j^L = \sum_{k=0}^{n_L} (\delta_k^{L+1} w_{kj}^{L+1}) out_j^L (1 - out_j^L)$$

kde $out_i(1 - out_i)$ predstavuje deriváciu aktivačnej funkcie pri sigmoidálnej aktivačnej funkcii.

Zmena váh

Na základe Delta rule možno vzťah pre výpočet nových váh vyjadriť nasledovne:

$$\Delta w_{ji}^L = \alpha \delta_j^L out_i^L$$

kde α značí koeficient učenia, j index neurónu, i index vstupu a L index vrstvy.

2 Manuál k programu

Program možno preložiť a spustiť pomocou pravidiel v súbore Makefile, manuálnym definovaním argumentov alebo pomocou konfiguračného súboru. Program bol testovaný na systémoch Ubuntu 16.10 a merlin.fit.vutbr.cz.

Makefile

Pre zjednodušenie manipulácie s programom je tento program možno jednoducho preložiť použitím príkazu:

```
make
```

Pre spustenie programu s prednastavenými hodnotami je možné použiť príkaz:

```
make run
```

Argumenty programu

Hidden Layer Number - celkový počet skrytých vrstiev nerátajúc vstupnú a výstupnú vrstvu

Nodes Per Layer - celkový počet neurónov v jednej vrstve

Learning rate - konštanta učenia používaná pri zmene váh

Epochs to train - počet cyklov behu siete nad datasetom

Verbosity - granularita výstupu

Number of Inputs - počet vstupov na základe datasetu

Datafile Path - cesta ku datasetu

Testfile Path - cesta ku súboru pre testovanie funkcionality

Konfiguračný súbor

Konfiguračný súbor *configuration.cfg*, prípadne iný manuálne vytvorený špecifikuje všetky argumenty programu v tvare argument=hodnota. Je potrebné nevkladať pred hodnotu medzery a dodržať poradie argumentov. Spustenie programu s konfiguračným súborom je možné pomocou príkazu:

```
./bpnetwork configuration.cfg
```

3 Implementačné detaily

Program je implementovaný v jazyku C++ s využitím štandardu C++11. Používa .csv súbor pre konfiguráciu a ako vstupný dataset.

Abstrakcia a modely

Program pozostáva z niekoľkých tried ktoré majú za cieľ abstrahovať reálne prvky siete a ďalších pomocných tried a metod.

Hlavná trieda *Network* obsahuje všetky ostatné objekty a riadi beh programu. Trieda *Layer* reprezentuje abstrakciu vrstvy a trieda *Neuron* predstavujúca jeden článok v triede *Layer*. Tento objekt práve obsahuje základné údaje pre výpočet algoritmu ako *netInput*, *nodeOutput* a *deltaValue*. Trieda *Link* reprezentuje spojenie neurónov dvoch vrstiev a obsahuje danú váhu. Pomocná trieda *Logger* slúži len pre spracovanie výstupu programu a pomocná trieda *Arguments* slúži len na spracovanie vstupných argumentov.

Symmetry breaking

Pri inicializácii váh siete dochádza k javu kedy sa sieť nemôže učiť pretože zmena pre všetky váhy je rovnaká. Tento jav ma za výsledok konštantný výstup, často okolo strednej hodnoty. Riešením je práve tkzv. symmetry breaking teda inicializácia náhodnými váhami.

Aktivačná funkcia

Jednou z bežne používaných aktivačných funkcií, ktorú som zvolil aj pre tento program je sigmoida. Má tvar $\varphi(x) = \frac{1}{(1+e^{-x})}$. Je vhodná kvôli jej derivovateľnosti a faktu že pre akýkoľvek vstup vracia hodnotu v rozmedzí 0 - 1.

4 Dataset

Dataset je dostupný na UCI Machine Learning Repository pod názvom Wine Quality Data Set ¹. Dataset je založený na štúdií ², ktorá skúmala možnosť predikcie preferencií ľudskej chuti na základe analytických meraní chemických vlastností vína.

Atributy Datasetu: Vstupné premenné založené na fyziochemických testoch:

1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates
11. alcohol

Výstupná premenná:

12 - quality (score between 0 and 10)

Normalizácia

Kvôli použitiu sigmoidy ako aktivačnej funkcií je potrebné zvolený dataset normalizovať. Normalizácia bola vykonaná prevedením výstupnej predikovanej hodnoty značiacej kvalitu vína z rozsahu 0 - 10 na rozsah 0.0 - 10.0. Ďalej bol tento dataset zredukovaný o prvý riadok - hlavičku značiacu legendu jednotlivých polí.

¹<https://archive.ics.uci.edu/ml/datasets/Wine+Quality>

²<http://www.sciencedirect.com/science/article/pii/S0167923609001377>

Holdout validation

Dataset bol rozdelený na dva menšie datasety - trénovací a testovací. Trénovací dataset slúži k naučeniu siete, zatiaľ čo testovací je použitý k overeniu správnosti naučenej siete. Pôvodný dataset bol rozdelený na trénovací a testovací v pomere 67% a 33% respektíve.

Predikcia

Predikcia výslednej kvality prebieha učením siete na trénovacom datasete a následne hodnotením dát v testovacom datasete.

Rozloženie

Špecifickosťou datasetu je rozloženie kvality vína okolo hodnôt 5-6.

	3	4	5	6	7	8	9
red	10	53	681	636	199	18	0
white	20	163	1457	2198	880	175	5

Table 1: Rozloženie kvality vína v datasetoch

5 Výsledky

Následovné výsledky boli dosiahnuté pri spustení programu s prednastavenými parametrami (pravidlo make run) a ich spriemerovaním.

Testovanie

Testovaním siete na rôzne argumenty som došiel k záveru že ako počet epoch je vhodné zvoliť parameter v rádoch tisícok epoch. V prípade nízkeho parametru nedôjde k naučeniu siete a generuje hodnoty blízke 6. Pre výstup 6 má teda vysokú precíznosť ale nulovú pre ostatné výstupy.

Štatistické výsledky

Pre jednotlivé 3 behy sú namerané a zpriemerované nasledovné štatistické udaje:

Range min, max - min, max $|target - output|$.

Mean Absolute Deviation - daný vzťahom: $MAD = \frac{\sum |target - output|}{N}$

Total Error - daný vzťahom: $ERR = \sum \frac{(target - output)^2}{2}$

Namerané hodnoty

Range - minimum: 0.000082

Range - maximum: 0.231495

Mean Absolute Deviation: 0.057499

Total Error: 1.536651

Class 4 Correct: 9.8039

Class 5: Correct: 70.5179

Class 6: Correct: 48.29932