



Soft computing - projekt

Praktická úloha predikcie za použitia backpropagation

27. listopadu 2022

Šimon Galba (xgalba03)

1 Dataset a riešená úloha

Úlohou mnou vytvorenej neurónovej siete je, ako už z názvu vyplýva, predpovedať nejakú udalosť na základe vstupných dát. Rozhodol som sa pracovať s datasetom obsahujúcim rôzne informácie o zdravotnom stave pacienta a informáciu o tom, či prekonal alebo neprekonan infarkt. Popis datasetu je vložený nižšie. Na základe tohto datasetu sa sieť naučí predpovedať riziko infarktu pacienta na základe ôsmich atribútov jeho zdravia.

1.1 Backpropagation

Na tréning neurónovej siete bol využitý algoritmus backpropagation. V skratke ide o aktualizáciu váh neurónov, kde spätná propagácia znamená derivovanie aktívnej funkcie.

1.2 Dataset

Pre potreby učenia neurónovej siete bolo potrebné zčísliť vstupný dataset. Nasledujúce dve tabuľky obsahujú ten istý riadok datasetu, prvý pred úpravou a druhý po nej.

Položky ID a Prekonan infarkt sú z datasetu odstránené. Položky zo stĺpca Prekonan infarkt sú však uložené na tréning a testovanie siete v samostatnom poli.

1. ID
2. Pohlavie - Male\Female = 1\0
3. Vek - >50\<50 = 1\0
4. Zvýšený krvný tlak
5. Srdcové choroby
6. V manželstve - Yes\No = 1\0
7. Priemerný krvný cukor - >160\<160 = 1\0
8. Bmi - >30\<30 = 1\0
9. Fajčiar - Yes\No = 1\0
10. Prekonan infarkt

1	2	3	4	5	6	7	8	9	10
9046	Male	67	0	1	Yes	228.69	36.6	formerly smoked	1
x	1	1	0	1	1	1	1	1	x

2 Architektúra siete

Sieť sa skladá z nasledujúcich vrstiev:

1. Jednej vstupnej vrstvy, obsahujúcej počet neurónov rovný počtu atribútov z datasetu.
2. Jednej skrytej vrstvy, kde počet neurónov je rovný priemeru počtu neurónov vstupnej a výstupnej vrstvy. Tento počet je zvolený na základe neoficiálneho pravidla od autora Jeffa Heatona [1]
3. Výstupnej vrstvy, ktorá má jeden neurón.

3 Učenie siete a predikcia

3.1 Dataset

Učenie siete prebieha na podmnožine datasetu, kde sú umelo vybrané vzorky, a to kvôli nerovnomernému rozdeleniu tried v datasete. Dataset obsahuje len 250 pozitívnych prípadov infarktu, no až 4500 negatívnych. Kvôli takejto nerovnováhe dát a jednoduchosti siete bolo nutné zmenšiť množinu negatívnych prípadov infarktu. Množina infarktov sa taktiež zmenšila na polovicu, kvôli korektnosti učenia, aby sa oddelili testovacie a učiace dáta.

3.2 Parametre učenia

Váhy neurónovej siete sú na začiatku inicializované náhodne. Učiacia konštanta *learning_rate* je nastavená na hodnotu 0.01. Táto konštanta vynásobená gradientom sa potom odčítava od váhy pre výpočet novej váhy v ďalšom kroku, pričom gradient je počítaný ako násobok δ *(momentálny vstup učenia).

3.3 Učenie

Po inicializácii prvotných váh neurónov, je z vstupnej trénovacej množiny náhodne vybraný vstupný riadok. Pre tento vstup je zavolaná funkcia predikt, ktorá vracia predikciu našej siete, a zároveň aj správny výsledok tejto predikcie z datasetu. Následne sa pre každú vrstvu siete (počínajúc výstupnou vrstvou smerujúc

ku začiatku siete) vypočíta hodnota nových váh neurónov na základe správneho výsledku a konštanty učenia. Následne sa tieto váhy prenesú na samotné vrstvy. Poradie výpočtu a aktualizácie váh je dôležité, kvôli využitiu hodnoty δ skrytými vrstvami.

3.4 Predikcia

Výpočet predikcia siete je v tomto prípade pomerne jednoduchý. V iterácií sa pre každý atribút vstupu vypočíta jeho aktivácia pomocou jeho násobku s danou váhou atribútu. Tieto aktivácie sa potom sčítajú pre všetky atribúty a táto hodnota sa vloží do funkcie *Sigmoid*, ktorá nám určí číselnú hodnotu výstupu.

4 Spustenie

4.1 Závislosti

Projekt je spustiteľný na každom stroji s prostredím Python3 a nainštalovanými týmito knižnicami:

- random
- math
- pickle
- csv
- argparse
- sys

4.2 Módy spustenia

Projekt je písaný v jazyku Python3, nie je teda potrebné vykonávať preklad zdrojových súborov. Hlavný kód *sfc_main* je spustiteľný v troch módoch.

- **python3 sfc_main.py train** - trénovanie modelu, vytvorí model.pkl v adresári kde sa nachádza zdrojový kód
- **python3 sfc_main.py test -model model** - testovanie modelu, využíva už predtrénovaný model siete, vypíše výsledky siete
- **python3 sfc_main.py xor_example** - krátka ukážka činnosti siete na úlohe XOR, s architektúrou 3-2-1

5 Príklady výstupu

```
Iteration 1499000 with error = 0.19, total: 57.0  
[SUCCESSFUL RUN]  
  
Process finished with exit code 0
```

Obrázek 1: Koniec tréovania modelu

```
Accuracy: 0.7735470941883767  
Number of detected strokes: 194  
Detected no strokes 192  
Detected in percentage: 77.60000000000001  
False positives: 59  
Undetected 54  
  
Process finished with exit code 0
```

Obrázek 2: Spustenie testovania modelu

```
Iteration 97000 with error = 0.0, total: 0.0  
Iteration 98000 with error = 0.0, total: 0.0  
Iteration 99000 with error = 0.0, total: 0.0  
Expected 1 , got: 1.0  
Expected 1 , got: 1.0  
Expected 0 , got: 0.0  
Expected 0 , got: 0.0  
  
Process finished with exit code 0
```

Obrázek 3: Ukážka učenia funkcie XOR

Literatura

- [1] Heaton, J.: *Introduction to Neural Networks with Java*. Heaton Research, 2008, ISBN 9781604390087. Dostupné z: <https://books.google.sk/books?id=Swlcw7M4uD8C>