Umelá inteligencia

Autor: Štefan Lapšanský

Zadanie

Rozpoznávanie vzorov pomocou strojového učenia

MNIST dataset obsahuje obrázky ručne písaných číslic a prislúchajúci label. Vytvorte model s pomocou algoritmov strojového učenia, ktorý dokáže na základe obrázku ručne písaného čísla klasifikovať aké číslo sa nachádza na obrázku.

Postup

1. Vytvorte model pomocou neurónovej siete trénovanej algoritmom backpropagation.
2. Vytvorte model pomocou vybraného algoritmu pre tvorbu rozhodovacích stromov.
3. Vytvorte model pomocou algoritmu RandomForest

Úlohy:

- Vyhodnoťte kvalitu každého modelu pomocou confusionmatrix a celkovej error rate.
- Do dát doplňte aspoň 3 odvodené atribúty tak, aby ste znížili error rate celkovo o minimálne 1 percentuálny bod
- Ak skombinujete všetky 3 modely do jedného modelu, o koľko sa zvýši úspešnosť klasifikácie?
- Ktorý z atribútov má najvyšší vplyv na kvalitu modelu a prečo?

Opis riešenia

Toto zadanie som riešil v jazyku python. Program má jeden .py súbor. V tomto súbore sa okrem mainu nachádzajú ďalšie štyri funkcie.

Funkcie:

dec_tree() – v tejto funkcii vytváram model pomocou algoritmu Decision Tree

backprop() – vo funkcii vytváram model pomocou algoritmu Backpropagation

rforest() – v tejto funkcii vytváram model pomocou algoritmu Random Forest

combination() – funkcia v ktorej sú spojené všetky tri modely do jedného

Funkcie dec_tree(), backprop() a rforest() sú podobné, to isté sa v nich vykonáva len za pomoci rôzneho algoritmu. V každej z týchto funkcií si na začiatku načítam trénovací dataset. Ako si postupne načítavam zo súboru potrebné dáta, label a potom samotné dáta zodpovedajúce labelu, tak ich načítavam do dvoch rôznych premenných a to kvôli tomu, že do jednej premennej hneď pridávam atribúty. Takže každá z týchto funkcií bude obsahovať dva modely, jeden bude obsahovať dáta s atribútmi a druhý bez atribútov. Po načítaní dát na ktorých bude trénovať sa pomocou funkcie fit model vytrénuje. Po vytrénovaní sa načíta dataset, na ktorom sa to bude testovať, načíta sa najprv label a potom dáta, opäť podobne ako tréningový dataset kde sú dve premenné, z toho jedna bude obsahovať pridané atribúty.

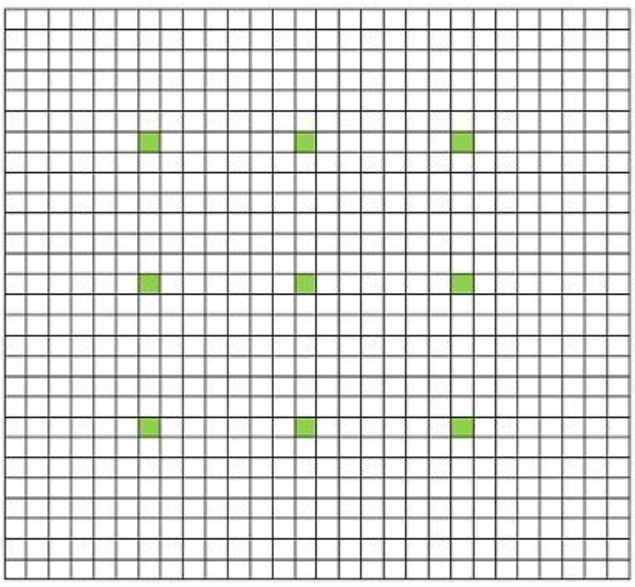
Po načítaní si model pomocou funkcie predict otestujem. Pridávam tie sité atribúty pre trénovanie a potom aj pre testovanie. Po otestovaní si pomocou funkcie confusion_matrix, accuracy_score a zero_one_loss vyhodnotím model, ktorý obsahoval aj doplnené atribúty a hneď za ním vyhodnotím model bez atribútov.

Funkcia combination() opäť načítavam dataset na tréning a následne vytrénujem modely pre Decision tree, Backpropagation a Random Forest. Nasleduje načítanie testovacie datasetu. Po načítaní modely prejdú testom a vypíše sa ich úspešnosť. Aby som spojil modely do jedného, tak to mi pomáha funkcia BlendEnsemble(). Následne vytrénujem tento model, ktorý obsahuje všetky tri modely. Po vytrénovaní spustím funkciu na testovanie. Na výpis error rate používam funkcie rmse, úspešnosť si vypočítam 100 – error rate.

Atribúty, ktoré spomínam, sú rôzne. Jedným z atribútov sú rôzne políčka z obrázku, ktoré pridám ako atribúty. Vo funkcii dec_tree() mám sedem atribútov. Šesť atribútov sú konkrétne políčka z dát práve načítaného obrázka a siedmy atribút počet políčok, na ktorých je číslo. Skúšal som pridať viac políčok z načítaného obrázka alebo zadať menej týchto políčok, ale najlepšie výsledky som mal pri týchto šiestich políčkach a počtu políčok, na ktorých je číslo. Skúšal som pridať atribút ako počet políčok kde sa nenachádza číslo, ale presnosť, teda error rate, sa veľmi nezlepšila. Funkcia backprop() má taktiež ako atribúty políčka, ale konkrétne tri políčka a další atribút, ktorý dostanem výpočtom, počet políčok kde sa nenachádza číslo deleno počet políčok na ktorých sa nachádza číslo. Skúšal som pridať aj viacej políčok a aj pridať počet políčok, kde sa číslo nachádza, ale s týmito atribútmi, ktoré sú tam teraz, mám najlepšie výsledky. Vo funkcii rforest() mám tri konkrétne políčka, počet políčok na ktorých sa nachádza číslo a číslo ktoré dostanem výpočtom, počet políčok na ktorých nie je čislo deleno počet políčok na ktorých je číslo. Skúšal som pridať atribúty ako napríklad ďalšie políčko alebo počet políčok na ktorých sa nenachádza číslo, ale výsledky neboli priaznivé.

Dôležitosť atribútov

Atribúty ktoré pridávam sú konkrétny pixel(políčko), počet políčok na ktorých sa nachádza číslo a počet políčok na ktorých sa nenachádza číslo deleno počet políčok, na ktorých sa nachádza. Atribút konkrétneho políčka mi nepríde ako ten najdôležitejší atribút, hoci nepridávam náhodné políčko, ale sám som si určil, ktoré políčka budú najdôležitejšie (vyfarbené) a tie som pridával, samozrejme nie všetky, ale len niektoré.



Dôležitejší atribút je počet políčok na ktorých číslo je, pretože na rôzne čísla potrebuje rôzny počet políčok. Napríklad na napísanie jednotky stačí v podstate palička, ale na napísanie trojky budú políčka viac zapísané. Atribút ktorý dostanem výpočtom ako som uviedol vyššie, tak ten by som taktiež radil medzi ten dôležitý, lebo akonáhle som ho pridal, tak som videl, že úspešnosť sa zvýšila. Tieto dva uvedené atribúty mi prídu dôležité, lebo po ich pridaní sa mi úspešnosť zvýšila.

Spôsob testovania

Všetky tri modely som testoval na tréningovom datasete v ktorom bolo 10 000 riadkov a testovacom datasete, v ktorom bolo tak isto 10 000 riadkov. Nasledujú výsledky pre jednotlivé modely:

Decision Tree:

```
Result with attributes...
Accuracy  42.98 %
Error rate  57.02 %
```

```
Result without attributes...
Accuracy  40.09 %
Error rate  59.91 %
```

Backpropagation:

```
Result with attributes...
Accuracy  68.42
Error rate  31.58
```

```
Result without attributes...
Accuracy  66.08 %
Error rate  33.92 %
```

Random Forest:

```
Result with attributes...
Accuracy  68.33
Error rate  31.67
```

```
Result without attributes...
Accuracy  66.92 %
Error rate  33.08 %
```

Počas testovania týchto troch modelov som mal rôzne výsledky – dáta s atribútmi nemali niekedy menší error rate, niekedy zas rozdiel error rate bol minimálny, ale ako je znázornené na obrázku, boli aj testy, kedy dáta s atribútmi mali nižší error rate.

Ďalej som to testoval aj na väčšom datasete pre tréning, konkrétne 20 000 riadkov, dataset pre test ostal ten istý, 10 000 riadkov. Výsledky pre jednotlivé modely :

DecisionTree:

```
Result with attributes...
Accuracy  84.83 %
Error rate  15.17 %
```

```
Result without attributes...
Accuracy  84.67 %
Error rate  15.33 %
```

Backpropagation :

```
Result with attributes...
Accuracy  96.36
Error rate  3.64
```

```
Result without attributes...
Accuracy  96.08 %
Error rate  3.92 %
```

Random Forest :

```
Result with attributes...
Accuracy  95.77
Error rate  4.23
```

```
Result without attributes...
Accuracy  95.74 %
Error rate  4.26 %
```

Na týchto výsledkoch je vidieť, že ak som tréningový dataset zväčšil, tak aj úspešnosť je lepšia. Počas viacerých testov ktoré som spúšťal boli výsledky podobné ako sú výsledky na obrázkoch. Ak sa do dát pridajú atribúty, tak sa error rate veľmi nezníži, poprípade sa zníži, ale len veľmi málo.

Zhodnotenie riešenia

Pri trénovaní na väčšom datasete je úspešnosť väčšia, ale ak sa do dát pridajú atribúty, tak error rate je približne taký istý ako pri dátach bez atribútov a pohybuje sa veľmi málo. Na rozdiel od menšieho datasetu, kde úspešnosť je menšia ale error rate sa už pohybuje oveľa viac, taže testy na 10 000 datasete pre tréning a aj pre test skutočne hýbu s error rate a vedia pomôcť, aby sa error rate znížil.

Po skombinovaní všetkých troch modelov do jedného úspešnosť vzrástla. Trénovací a aj testovací dataset mali 10 000 riadkov keď som to testoval a nasledujúci obrázok znázorňuje úspešnosť skombinovaných modelov. Dáta boli bez atribútov.

```
All in one result...
Accuracy  97.82 %
Error rate  2.18 %
```

Pre možné zlepšenie programu a zníženie error rate by sa pridal atribút, ktorý by počítal počet miest, na ktorých by boli prázdne políčka, napríklad nula by mala dve takéto miesta a napríklad osmička tri takéto miesta.