Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

UMELÁ INTELIGENCIA

Akademický rok 2018/19

Rozpoznávanie vzorov pomocou strojového učenia

2019

Bratislava

riešiteľ

Daniel Minárik

rok štúdia: **druhý**

Obsah

Zadanie

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 **Random Forest**

Úlohy:

- Vyhodnoťte kvalitu každého modelu pomocou confusion matrix a celkovej error rate.
- Do dát doplníte 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?

1 Modely strojového učenia

1.1 Neurónová sieť

Neurónová sieť tohto projektu bola vytvorená pomocou frameworku weka. Tento framework v sebe zahrňuje vytvorenie neurónovej siete, konkrétne MultiLayerPerceptron (viacvrstvový perceptrón). Tento model obsahoval 3 vrstvy neurónov, a to prvá vstupná vrstva obsahujúca 784 neurónov, vonkajšia vrstva obsahujúca 10 neurónov a vnútorná (skrytá) vrstva obsahujúca 150 neurónov. Na tomto modely bol použitý trénovací algoritmus učenia metódou spätného šírenia chýb, známeho ako „backpropagation“. Pre dosahovanie čo najnižšieho error rate, boli pre tento model nastavané aj ďalšie parametre a to najmä počet epoch, resp. iterácií pri ktorých bude prebiehať trénovanie neurónovej siete. Taktiež bola nastavená rýchlosť učenia a momentum na hodnoty 0,2 a 0,1. Pri konečnom riešení som dospel k číslu tri, z dôvodu, aby nedošlo k pretrénovaniu neurónovej siete, a taktiež s časových dôvodov, aby jej trénovanie netrvalo príliš dlho. V súčasnosti toto vybudovanie modelu trvá približne 700 sekúnd.

1.2 Rozhodovací strom

Rozhodovací strom bol vytvorený taktiež pomocou frameworku weka. Táto knižnica podporovala vytvorenie rozhodovacieho stromu interne zvaného J48, ktorý je vylepšením známejšieho algoritmu C4.5. Tento model bol taktiež nastavený tak, aby sa čo najviac znížila celková chybovosť tohto modelu. Konkrétne bol nastavaný ako „Unpruned“ a taktiež bola bol na stavený minimálny počet objektov na jednej úrovni nastavený na hodnotu 3. Táto zmena pomáha dosahovať to, že tento strom dosahuje menších hĺbok, ako keď mať štandardne nastavenú hodnotu 2. Trénovanie tohto modelu trvá približne 300 sekúnd.

1.3 Random Forest

Model Random Forest bol vytvorený pomocou frameworku weka. Táto knižnica umožňuje vytvorenie modulu Rundom Forest s možnými rôznymi nastaveniami. Hlavným nastavením, ktoré som použil je počet stromov, ktoré bude tento „les“ obsahovať. Na základe pozorovanie výsledkov testovania som dospel k číslu 50, kedy sa

dosahovalo relatívne nízkej chybovosti. Ďalším dôležitým nastavením tohto modelu je maximálna hĺba jednotlivých stromov. V tomto prípade je nastavená tak, že hĺbka nie je obmedzovaná. K takémuto rozhodnutiu som dospel z dôvodu dostatočne, že tento Random forest obsahuje dostatočné veľké množstvo stromov, preto sa nevytvárajú stromy, ktoré sú priveľmi hlboké. Trénovanie tohto modelu trvá približne 300 sekúnd.

1.4 Práca s datasetom

Všetky tieto modely, ktoré sú vytvorené pomocou nástrojov, ktoré sú poskytované frameworkom weka požadujú, aby pri čítaní údajov z datasetu bol tento súbor typu ARFF. Súbory tohto typu majú presnú štruktúru , kde sú pomocou anotácií definované atribúty, t.j. ich názov a hodnoty a taktiež sú definované dáta, ktoré zodpovedajú týmto atribútom. Tento súbor sa vytvára v triede Mnist (createDataSet();) pomocou nástrojov, ktoré poskytuje weka. Jediným problémov je, že tento nástroj nedokáže zadefinovať atribút Label, konkrétne nedokáže zadefinovať, že tento atribút nie je typu numeric, ale je typu, že môže nadobúdať hodnoty 0 až 9. Tento nedostatok tohto projektu, je riešený len ručným zmenením jedného riadku vo výslednom arff súbore.

Po vytvorení týchto súborov pre trénovací a testovací dataset, môžu jednotlivé modely vytvoriť inštancie týchto datasetov. Následne je na týchto dátach použitý normalizačný filter. Tento filter zabezpečuje, aby všetky tieto dáta boli len z rozsahu 0 až 1.

1.5 Ukladanie a načítanie modelov

Modely vytvorené už veľakrát spomínaným nástrojom weka podporujú ukladanie a opätovné načítanie natrénovaných modelov. V tomto projekte ja táto možnosť využívané hlavne z dôvodu, aby pri každom spustení programu nebolo potrebné čakať priveľkú dobu, pokým by sa všetky tri modely vybudovali.

2 Porovnanie modelov a testovanie

2.1 Neurónová sieť

Dosahované výsledky:

Correctly Classified Instances	9534	95.34 %
Incorrectly Classified Instances	466	4.66 %
Total Number of Instances	10000	

Confusion matrix:

	0	1	2	3	4	5	6	7	8	9
0	964	1	2	1	2	2	2	1	5	0
1	0	1 119	3	4	0	2	3	0	4	0
2	6	1	988	6	7	0	0	9	15	0
3	0	0	14	972	0	9	0	5	10	0
4	1	0	3	1	964	0	6	0	2	5
5	5	1	2	8	6	854	7	1	8	0
6	6	4	3	2	5	11	922	0	5	0
7	1	14	15	13	8	0	0	968	6	3
8	1	0	5	6	7	2	6	4	943	0
9	5	7	1	14	81	8	0	29	24	840

Neurónová sieť dosahovala relatívne nízke hodnoty error rate a to konkrétne 4,66%. Z výsledkov, hlavne z matice vyplýva, že najväčšia chybovosť je pri čísle 9. Toto číslo je najviac mýlene s číslom 4 kedy, kde 81 krát nastala situácia, kedy bol 9 predikovaná hodnota 4. Číslo 9 je taktiež ešte mýlené s číslami 7 a 8. Pre ostatné čísla sa v matici nenachádzajú extrémne hodnoty, čo značí, že tieto čísla s relatívne vysokou presnosťou.

Nevýhodou tohto modelu je fakt, že jeho trénovanie v porovnaní s odstavnými dvoma modelmi trvá príliš dlho.

Možnosťou na zlepšenie tohto modelu je trénovanie na viacerých iteráciách alebo pridanie upravenie počtu vnútorných uzlov na väčšie číslo. Avšak tieto úpravy spôsobia ešte dlhšie vybudovanie tohto modelu.

2.2 Rozhodovací strom

Dosahované výsledky:

Correctly Classified Instances	8881	88.81 %
Incorrectly Classified Instances	1119	11.19 %
Total Number of Instances	10000	

Confusion matrix:

	0	1	2	3	4	5	6	7	8	9
0	942	0	5	5	1	4	10	5	3	5
1	4	1 085	11	8	2	4	4	1	16	0
2	12	10	918	25	5	10	6	23	21	2
3	3	14	43	847	5	42	1	15	24	16
4	7	5	13	9	867	11	5	4	26	35
5	16	10	9	50	11	725	14	10	23	24
6	8	6	16	4	12	20	883	0	7	2
7	1	8	27	18	14	7	0	920	10	23
8	17	9	18	31	23	24	18	9	807	18
9	8	5	3	24	31	16	1	13	21	887

Rozhodovací strom dosahoval spomedzi všetkých modelov najväčšiu mieru chybovosti, konkrétne 11,19 %. Z matice môžeme vyčítať, že táto chybovosť je sa nachádza takmer pri každom predikovanom čísle, okrem nuly a jednotky, kde sa oproti ostatným číslam vysoká miera presnosti. Najväčšie nepresnosť je pre čísla 8 a 9.

Možnou nevýhodou tohto modelu na tento účel je fakt, že dataset obsahuje veľké množstvo atribútov, konkrétne 784. Vzhľadom na tento fakt tento rozhodovací strom potrebuje mať rozhodnutie pre všetky tieto atribúty zabezpečené, preto sa nám vytvára príliš hlboký rozhodovací strom.

Možnosťou na zlepšenie tohto modelu je upravenie datasetu, resp. pridanie váh pre jednotlivé atribúty, vďaka čomu bolo možné jednoduchšie rozhodovanie. Ďalším možným vylepšením by bolo nájsť optimálnu maximálnu hĺbku stromu.

2.3 Random Forest

Dosahované výsledky:

Correctly Classified Instances	9669	96.69 %
Incorrectly Classified Instances	331	3.31 %
Total Number of Instances	10000	

Confusion matrix:

	0	1	2	3	4	5	6	7	8	9
0	970	1	1	0	0	1	3	1	2	1
1	0	1 122	1	3	0	2	3	1	3	0
2	8	1	994	5	2	0	4	10	7	1
3	0	0	10	968	0	10	0	9	10	3
4	1	0	1	0	956	0	5	0	2	17
5	4	0	0	12	1	862	4	1	5	3
6	6	3	1	0	4	6	933	0	5	0
7	1	4	19	2	0	0	0	987	3	12
8	7	0	6	8	4	3	8	4	922	12
9	7	3	2	13	16	2	1	5	5	955

Model random forest dosahoval zo všetkých modelov najlepšie výsledky. Tento model dosahoval úspešnosť 96,69 % z čoho vyplýva, že error rate tohto modelu je 3,31%. Z matice môžeme vyčítať, že najväčšia chybovosť bola pri čísle 9 a 7, avšak určite nie je porovnateľná s chybovosťou modelu rozhodovacieho stromu.

Tento model sa oproti modelu samostatného rozhodovacieho stromu javí ako lepšie riešenie. Dôležité je aj to že tento model potreboval približne rovnaký čas na vybudovanie ako rozhodovací strom (aj ako neurónová sieť), pričom dosahoval značne lepšie výsledky. Z týchto výsledkov vyplýva, že je lepšie použiť viac rozhodovacích stromov menšej hĺbky ako jeden veľký rozhodovací strom.

Možnosť zlepšenia tohto modelu vidím v možnom nastavení počtu stromov. Taktiež možné zlepšenie by sa prejavilo aj v prípade upravenia vstupného datasetu, kde by mohli byť pridané nové atribúty.

2.4 Skombinovanie modelov

	0	1	2	3	4	5	6	7	8	9
0	969	0	7	0	1	5	6	1	5	5
1	1	1 122	1	0	0	0	3	5	0	4
2	1	1	998	13	2	0	1	19	5	2
3	0	4	6	969	0	8	0	2	4	15
4	0	0	2	0	959	1	3	1	3	24
5	1	2	0	10	0	863	5	0	2	5
6	3	3	3	0	5	6	935	0	8	0
7	1	1	9	9	0	1	0	986	4	5
8	3	2	6	7	2	5	5	3	933	9
9	1	0	0	2	13	3	0	11	10	940

Skombinovanie modelov sa v projekte realizovalo pomocou cyklu, kde sa osobitne pre každý jeden prvok testovacieho datasetu vyhodnocovala jeho predikovaná hodnota pomocou všetkých troch modelov. Pri tomto vyhodnocovaní mohli nastať tri situácie. Prvá situácie, ktorá je najpozitívnejšia je, keď sa všetky tri modely zhodnú na predikovanom čísle. Táto situácia nastala presne 8 707 krát z 10 000 testovaní. Druhou situáciou je, keď sa zhodnú len dva modely. Táto situácia nastala 1 174 krát. Vysokú mieru nezhodný v tomto prípade mal model rozhodovacieho stromu. Poslednou kategóriou je, keď sa všetky tri modely. Táto situácia nastala 119 krát. Celková úspešnosť tohto skombinovaného modelu je 96,67 % z čoho vyplýva že celkový error rate je na úrovni 3,26 %. Z týchto čísel nám vyplýva že tento prístup k vyhodnocovaniu dosahoval najlepšie výsledky zo všetkých modelov. Avšak môžeme si všimnúť, že tento výsledok nie je nejako veľmi odlišný od úspešnosti modelu random forest. Konkrétne tento prístup dosahuje len o 0,05 % lepšiu úspešnosť ako spomínaný model.

Možným vylepšením rozhodovanie pri tomto prístupe by bolo zmenou riešenia stavu, kde sa všetky modely nezhodnú. V súčasnom stave projektu sa táto situácie rieši tak, že výsledkom je predikovaná hodnota od modelu random forest. K tomuto rozhodnutiu som dospel z dôvodu, že tento model dosahoval najlepšie výsledky. Možným zlepšením by bolo rozhodnúť sa na základe percentuálnej hodnote s ktorou jednotlivé modely tento prvok predikujú. Avšak toto riešenie je v nástroji weka ťažko realizovateľné, pretože sa mi nepodarilo nájsť nástroj, pomocou ktorého by som sa k tomuto číslu dokázal dostať.

3 Atribúty

3.1 Pridávanie atribútov

V tomto projekte sa realizovalo pridávanie nových odvodených atribútov do trénovacieho ako aj do testovacieho datasetu. Pre pridávanie možných odvodených dát som vyskúšal množstvo riešení, avšak u niektorých to spôsobilo opačný efekt, aký sa očakával. Pri niektorých testoch tieto nové atribúty spôsobili, že celkový error rate sa zvýšil aj o jeden % bod.

Čiastočne úspešným riešením bolo pridanie piatich nových atribútov. A to konkrétne:

1. Počet stĺpcov, kde je hodnota väčšia ako 0
2. Súčet čísel v pomyselnom strede matice
3. Maximálna súvislá postupnosť, kde sú všetky čísla väčšie ako 0
4. X súradnica centra objektu, na základe rozloženia čísel
5. Y súradnica centra objektu, na základe rozloženia čísel

Napriek týmto vylepšeniam datasetu sa mi nepodarilo zvýšiť úspešnosť modelov o 1 % bod. Avšak na druhú stranu sa mi podarilo zlepšenie o približne 0,5 %. Tento fakt môže byť zapríčinením aj toho, že tieto modely dosahujú relatívne vysoké percentá úspešnosti. Zvýšenie úspešnosti pri takýchto veľkých číslach je oveľa zložitejšie, ako keď modely pracujú s nízkou úspešnosťou. Toto tvrdenie môžem potvrdiť svojím pozorovaním, kedy keď model neurónovej siete dosahoval úspešnosť len 80 %, tak tieto atribúty zmenili jeho úspešnosť približne 1,5 %, avšak po jeho prerobení a zvýšení jeho úspešnosti na terajšiu hodnotu sa mi takéto zvýšenie úspešnosti nepodarilo zopakovať.

Vytváranie týchto atribútov sa realizuje pri čítaní z pôvodného csv súboru, kde sa jednotlivé riadky rozdelia pomocou delimitra na jednotlivé čísla, a následne sa z týchto čísel odvodzujú spomínane nové atribúty. Následne sa z pôvodných + nových dát vytvorí súbor ARFF súbor, s ktorým sa pracuje rovnako, ako bolo spomínané vyššie v kapitole 1.4.

3.2 Atribúty s najvyšším vplyvom

Nástroje weka podporujú analyzovanie datasete, resp. vypočítanie dôležitosti jednotlivých atribútov. V tomto projekte som pomocou týchto nástrojov realizoval ohodnotenie atribútov trénovacieho datasetu a následne jeho zoradenie podľa dôležitosti. Takéto testovanie som vykonal na pôvodnom ako aj na mojím rozšíreným datasetom (rozširujúce atribúty). Z výsledkov zobrazím prvých a posledných 5 atribútov z tohto usporiadaného zoznamu.

Na prvých piatich miestach sa umiestnili:

14x15 15x15 16x15 17x14 13x15

Z týchto čísel môžeme vidieť to že všetky tieto čísla sa nachádzajú v strede matice, resp. v strede polohy na ktorom sa nachádza číslo. Tento fakt môže byť daným tým, že čísla, ktoré sú oválne, resp. nemajú vyplnený stred napríklad 0, sú týmito atribútmi takmer isto určiteľné. Po použití tohto istého programu na rozšírený dataset v prvej päťke umiestnili hneď tri odvodené atribúty, a to : tretí, prvý a druhý (podľa čísel v predchádzajúcej kapitole). Tieto jeden z týchto odpodených atribútov je práve ten ktorý obsahuje číslo súčtu stredov, resp. atribútov, ktoré sa umiestnili na prvých priečkach. Týmto testom som zistil, že odvodené atribúty sú v datasete relevantné a napomáhajú pri predikovaní čísel.

Na posledných piatich miestach sa umiestnili:

24x2 24x1 23x28 23x27 1x1

Tieto atribúty sa takmer vôbec nepodieľajú na predikovani čísle. Sú to najmä atribúty, ktoré sa nachádzajú na okrajoch plochy, resp. najme na okrajoch spodnej časti, kde pre takmer žiadna číslo nenachádza žiadna hodnota. Keďže pre takmer každú inštanciu čísla z datasetu sú tieto hodnoty nulové, tak sa nemôžu výrazne podieľať na vyhodnocovaní čísel. Pri použití na rozšírený dataset sa na týchto posledných pozíciách nevyskytoval žiaden odvodený atribút, čo je dobrá správa, že všetky odvodené atribúty sa v určitej nenulovej miere podieľajú na predikovani hodnôt.

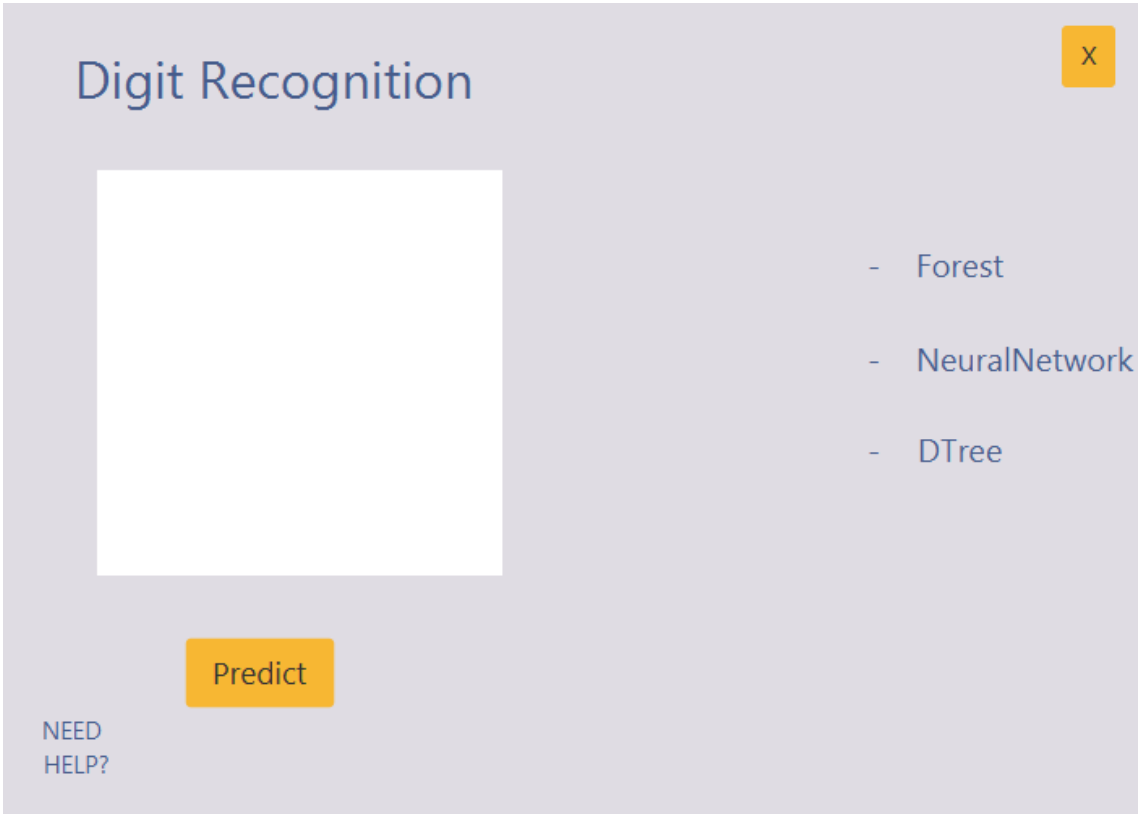
4 Používatelská príručka

Hlavný program trénovania s testovania je spustiteľný z triedy Luncher. V tejto triede sa nachádzajú tri nastaviteľné atribúty, a to TRAIN čo označuje či sa má vykonať vybudovanie modelu trénovaním na trénovacom datasete, alebo v opačnom prípade, či má byť načítaný zo súboru. EXTEND značí, či sa má použiť rozšírený dataset, v opačnom prípade sa použije pôvodný. MODEL označuje model, ktorý sa má spustiť, a to:

1. Neurónová sieť
2. Rozhodovací strom
3. Random Forest
4. Kombinovaný model

Po všetkých týchto modeloch nasleduje vyhodnotenie pomocou confusion matrix.

V projekte je taktiež vytvorené používateľské okno v javafx. Toto okno je spustiteľné z triedy Main. Po spustení sa otvorí dané okno:



V tomto okne biela plocha reprezentuje priestor, na ktorom je možné nakreslenie čísla. Pre jeho vymazanie sa použije pravé tlačidlo na myši. Po stlačení tlačidla Predict sa na pravej strane zobrazí obrázok tohto čísla ako je chápaný programov, t.j. v odtieňoch sivej farby, taktiež sa pre jednotlivé modely vypíše číslo, ktoré je predikované týmto modelom. Odkaz na ukážku s prácou v tomto okne : *https://i.imgur.com/XDKuE6c.mp4*

Tento projekt je vytvorený za pomoci Maven dependecies, ktoré sa nachádzajú v súbore pom.xml. Pre správne fungovanie tohto projektu, je potrebné aby všetky tieto dependencies boli neimportované. Taktie6 je potrebné, aby sa v adresári projektu nachádzali datasety v požadovanom názve a to vo formáte arff alebo cvs, z ktorého je potrebné pomocou programu vytvoriť požadovaný arff súbor.

5 Zhodnotenie

Všetky tri spôsoby strojového učenia dosahujú relatívne vysoké miery úspešnosti. Avšak na druhú stranu majú veľmi vysoké časové náročnosti a tak isto aj náročnosť na hardware. Pre tieto výpočty je najlepšie použiť nástroj, ktorý podporuje využívanie GPU, vďaka čomu by sa skrátila doba budovania model o značnú mieru. Avšak v mojom prípade som nenašiel takýto nástroj, ktorý by takto priamo pracoval v jave.

Nakoniec musím poďakovať nášmu cvičiacemu za zaujímavé zadanie, pretože doteraz som nemal príležitosť pracovať s týmto strojovým učením, ktoré bolo zaujímavé.