Závěrečný projekt ze Strojového učení v Počítačovém vidění

David Beinhauer

1 Popis řešení

Úkol je řešen za pomocí knihovny obsahující algoritmy pro strojové učení scikit-learn. Pro oba úkoly jsme použili převážně stejné transformace data i klasifikátory. Většina odlišností je dána především rozdílností dat a manipulací s nimi (více tříd apod.), jinak je binární klasifikace i klasifikace do více tříd téměř totožná. Pro spuštění programu s požadovanými paramatry lze využít příkazy zapsány v souboru *commands.txt*

Řešení problému jsme začali úpravou vstupních dat. Velká část úprav je provedena pomocí třídy *Dataset*. Soubor s daty bylo potřeba uložit do adresáře s programem, který je nejprve rozbalil, následně načetl a zformátoval do podoby 4-dimenzionálního tensoru ($pocet\_dat \times vyska \times sirka \times kanaly$). Tento formát je potřeba pro další úpravy dat. Ty jsou provedeny s využitím knihovny scikit-learn.
Nejprve jsme převedli data z obrázků v RGB formátu do šedého spektra (s vestavěnou heuristikou upravující výraznost každé složky na základě vnímání složek barev lidským okem). Úprava je provedena především z důvodu snížení dimenzionality parametrů (snížíme velikost obrázků na třetinu). Následně provedeme úpravu pro zvýraznění kontrastu obrázku (použita varianta z knihovny skimage). Myšlenka vedoucí k zmíněné úpravě je motivována skutečností, že při zvýraznění kontrastu jsou objekty často lépe odlišitelné od pozadí, a tedy pro klasifikátor bude jednodušší učit se objekty (může to ale také vézt k ztrátě části dat). Poslední úpravou dat je normalizace dat, aby měla každá složka stejnou střední hodnotu a rozptyl. Především abychom vyrovnali důležitost jednotlivých složek dat (důležité například pro MLP).
Dělení na trénovací, validační a testovací sadu je částečně vyřešen formátem staženého datasetu (je již rozdělen na trénovací a testovací sadu). Rozdělení na validační sadu je provedeno pomocí knihovny scikit-learn (GridSearchCV). Takto lze také natrénovat modely s různými parametry. Následně je zde uložen model s nejlepšími parametry a natrénován na trénovacích datech. Zmíněná funkce provádí stratifikovanou cross-validaci, neboli snaží se při každé generaci validační sady co nejrovnoměrněji rozdělit data z různých tříd (pro dosažení nejlepší generalizace dat, tedy i nejlepší natrénovaní pro neznámá data).
Pro trénování jsme použili 3 různé metody strojového učení s učitelem. Tyto metody jsou Support Vector Machines (ozn. SVM), Multilayer Perceptron (MLP) a Gradient Boosted Trees (BOOST). Metody jsme zvolili zejména z důvodu, že se výrazně liší a můžeme tak porovnat různé modely. Metodu MLP jsme zvolili, abychom měli mezi modely zastoupení neuronových sítí, které jsou v dnešní době snad nejlepší metodou pro klasifikaci obrázku. Vhodnější by bylo použití konvolučních sítí, to je ale výpočetně velmi náročné a nebylo by použitelné vzhledem k výpočetní síle mého počítače. Metodu SVM jsme zvolili především jako méně výpočetně náročnou náhradu neuronových sítí. Nakonec BOOST jsme použili jako pravděpodobně nejlepší metodu používající rozhodovací stromy. Volba metod byla také výrazně ovlivněna znalostmi z jiných již dokončených předmětů zabývajících se strojovým učením, kde nám bylo doporučeno při výběru modelu otestovat zmíněné 3 metody.
Samotné učení, predikce i výpočty metrik jsou provedeny s pomocí funkcí z knihovny scikit-learn. Pro zakreslení výsledku jsme použili ROC křivku pro každou třídu i macro a micro averaged variantu. ROC jsme zvolili, protože je z ní poměrně pěkně vidět kvalita klasifikátoru (čím větší plocha pod křivkou, tím lépe funguje).

2 Parametry a výsledky

Pořadí parametrů modelů určené cross-validací podle kvality parametrů jsou zapsány v souborech v adresáři *training_outputs* (soubory pojmenovány podle typu klasifikace a modelu). Výsledky predikcí jsou zapsány v adresáři *prediction_outputs* a grafy ROC křivek jsou uloženy v adresáři *plots*. Všechny jména souborů jsou dány použitým klasifikátorem a úlohou (fine - klasifikace do

podtříd jedné supertřídy, super - klasifikace pro odlišení 2 supertříd).

Pro MLP jsme testovali různé velikosti skryté vrstvy (pouze malé, protože jinak by nedotrénovalo v rozumném čase), dále různé hodnoty L2 regularizačního parametru a počáteční learning rate. Cross-validací jsme určili, že nejlepší je mít ze zvolených parametrů větší skrytou vrsvu (tedy 50 neuronů), dále je pro obě klasifikace lepší spíše nižší L2 regularizační parametr (0.001). Parametry pro binární klasifikaci se liší od multinomiální pouze v počátečním learning rate, který je pro binání klasifikaci vyšší. To si odůvodňuji především ze skutečnosti, že pro více tříd by bylo potřeba pomalejšího učení o více epochách (protože s vysokým learning rate pravděpodobně klasifikátor nedokáže vhodně konvergovat).
Z confusion matrix, precission, recall i ROC křivky lze vyčíst, že model se částečně natrénoval, bylo by ale pravděpodobně vhodnější trénovat síť více epoch a pravděpodobně také rozšířit skrytou vrstvu. Zmíněné řešení by ale výrazně zpomalilo trénování a nebylo by tak možné dotrénovat model v rozumném čase (na mém stroji). Macro-average precission se pro multiclass klasifikaci pohybovala kolem hodnoty 0.38 a pro binary kolem 0.75, recall pro multiclass asi 0.376 a pro binary 0.75. Obecně byly výsledky pro binární klasifikaci pro MLP nejlepší ze všech modelů jak v precision, tak i recall. Pro multiclass dopadlo MLP mírně hůř než pro BOOST. Z ROC křivek, lze ale vyčíst, že lepších hodnot dosahoval spíše BOOST klasifikátor (plocha pod křivkou je větší než u MLP).

Pro BOOST jsme testovali různé varianty learnig rate a volby počtu charakteristických rysů použitých pro hledání nejlepšího rozdělení následujícího vrcholu (buď všechny a nebo pouze odmocninu celkového počtu charakteristických rysů). Dospěli jsme nakonec k závěru, že nejlepší varianta je spíše větší learning rate (0.2) a na volbě počtu charakteristických rysů spíše nezáleží. Tyto výsledky jsou pravděpodobně způsobeny spíše malým množstvím stromů, kde vyšší počet by pravděpodobně přinesl lepší výsledky, ale celkově bylo nejnáročnější tento klasifikátor natrénovat a vyšší počet by již bylo téměř nemožné natrénovat.
Z confusion matrix, precission, recall i ROC křivky lze vyčíst, že tento model dosahoval obecně nejlepších modelů jak pro multiclass, tak i pro binary klasifikaci. S precission pohybující se kolem hodnot 0.39 pro multiclass a 0.74 pro binary a recall 0.398 pro multiclass a 0.74 pro binary jsou výsledky pro multiclass lepší než v případě MLP a pro binary jsou velmi vyrovnané (MLP je mírně lepší). Z ROC křivek, lze ale pěkně vyčíst, že BOOST varianta je pravděpodobně nejlepší z použitých.

Klasifikátor SVM dosahoval obecně výrazně horších výsledků než 2 výše zmíněné. Pro testovali jsme různé varianty parametru tolerance (pro zastavení trénování) a regularizačních parametrů. Nakonec se ukázalo, že je lepší zvolit spíše menší regularizační parametr a na toleranci moc nezáleží. Celkově vypadaly modely pro obě úlohy jako nedotrénované a pravděpodobně by bylo vhodné zvýšit počet iterací.
Ze stejných metrik jako ve výše uvedených modelech lze vyčíst, že SVM modely výrazně zaostávají za zbylými 2 modely. Pro multiclass klasifikaci se pohybují hodnoty precision i recall kolem hodnot 0.24 (tedy téměř jako nenatrénovaný model) a pro binární okolo 0.663. Všechny výsledky byly výrazně horší něž v případě MLP i BOOST. Zároveň z ROC křivky lze vyčíst, že například model pro multiclass klasifikaci se téměř vůbec nenatrénoval.

3 Závěr

Většina klasifikátorů pracovala docela podle mých představ. Překvapily mě ale výrazně špatné výsledky modelu využívajícího Support Vector Machines, který se téměř nenatrénoval. Další významnější problémy mi způsobovala velikost trénovacích dat, která značně komplikovala trénování modelů (speciálně pro různé parametry). Důsledkem byla nemožnost na mém počítači dostatečně natrénovat modely, jelikož nebylo možné modely dotrénovat v rozumném čase. Mírně mě také překvapily dobré výsledky klasifikátoru využívajícího Gradient Boosted Trees, který byl lepší i než Multilayer Perceptron. Pokud bych ale měl k dispozici větší výpočetní sílu, pak by ale pravděpodobně konvoluční sítě dosahovaly ještě lepších výsledků.