

Semestrální práce INUI2

Hrozny

Bc. Zdeněk Novotný

Zadání

Zadání úkolu A

V přiložených archivech se nacházejí vizuální data obsahující tři druhy informací. Jednak se jedná o celé bobule hroznů révy vinné (positive), dále pak obrázky obsahující necelé bobule (Grape negative) a obrázky neobsahující bobule (Environment). Provedte aplikaci Kohonenovy mapy pro shlukovou analýzu těchto dat a diskutujte výsledek (v optimálním případě by mapa měla data roztrždit do tří shluků podle typu). K dispozici je také druhá (testovací) skupina dat, na které provedte shlukovou analýzu pomocí Kohonenovy mapy vytvořené v prvním kroku a diskutujte výsledek. Pro práci je třeba provést image preprocessing obrazových dat tak, aby mohla být Kohonenova mapa aplikovatelná. Volba způsobu preprocessingu není určena.

Postup práce pečlivě zdokumentujte tak, aby byl reprodukovatelný. Ke shlukové analýze použijte Kohonenovu mapu. Není povoleno využívat hotová řešení třetích stran (knihovny, tutoriály, zveřejněné kódy, ...)

Zadání úkolu B

V přiložených archivech se nacházejí vizuální data obsahující tři druhy informací. Jednak se jedná o celé bobule hroznů révy vinné (positive), dále pak obrázky obsahující necelé bobule (Grape negative) a obrázky neobsahující bobule (Environment). Provedte aplikaci dopředné vícevrstvé umělé neuronové sítě k rozdělení dat do daných skupin. K trénování a validaci sítě použijte trénovací množinu dat, k finálnímu testování pak testovací množinu. Pro práci je třeba provést image preprocessing obrazových dat tak, aby mohla být dopředná vícevrstvá síť aplikovatelná. Volba způsobu preprocessingu není určena.

Postup práce pečlivě zdokumentujte tak, aby byl reprodukovatelný. K návrhu je možno použít nástroje TensorFlow a Keras.

Zadání úkolu C

V přiložených archivech se nacházejí vizuální data obsahující tři druhy informací. Jednak se jedná o celé bobule hroznů révy vinné (positive), dále pak obrázky obsahující necelé bobule (Grape negative) a obrázky neobsahující bobule (Environment). Provedte aplikaci konvoluční umělé neuronové sítě k rozdělení dat do daných skupin. K trénování a validaci sítě použijte trénovací množinu dat, k finálnímu testování pak testovací množinu.

Postup práce pečlivě zdokumentujte tak, aby byl reprodukovatelný. K návrhu je možno použít nástroje TensorFlow a Keras.

Zhodnocení

V závěrečném zhodnocení výsledků práce provedte porovnání všech tří přístupů ke klasifikaci hroznů a diskutujte z hlediska kvality výsledků, návrhové a výpočetní náročnosti.

Projekt

Přiložen je projekt vývojového prostředí IntelliJ IDEA, včetně trénovací a testovací množiny. Celý projekt je implementován v jazyce Python. Pro jednotlivé úkoly byly vytvořeny třídy, představující implementace odpovídajících umělých neuronových sítí a dále třídy a skripty, sloužící k jejich testování.

Soubory jazyka Python lze samozřejmě otevřít i v programu IDLE, pravděpodobně však bude nutné upravit importy (odmazat jméno balíčku „sem“).

Dále lze v projektu najít zmíněné trénovací a testovací soubory, ve složce img. Ve složce output jsou pak k nahlédnutí některé výstupy testování.

Úkol A

V rámci úkolu A byla implementována Kohonenova mapa – třída Kohonen v souboru `net_Kohonen_map.py`. Třída je pečlivě zdokumentována. V rámci implementace byl kladen důraz na zachování variability parametrů sítě pro účely následného testování.

Tato třída byla obalena testovací třídou (třída `Almighty` v souboru `almighty_test_class.py`) a testována v souborech `main.py` a `main_Kohonen_map.py`. Využívá též třídu `ImagePreProcessor` (soubor `image_preprocessor.py`) pro přípravu obrázků.

Samotné testování je však bohužel velice časově náročné a bylo provedeno jen v omezené míře. V souborech `output/1` a `output/2` jsou zdokumentovány dva testy, lišící se velikostí bloku `preprocessor`. Varianta, používající jemnější rozbor hran, se bohužel nestihla vypočítat.

Z pokusů je zřejmé, že samotná velikost bloku žádný zásadní rozdíl nezajistí. Překvapivá je pak celkově nízká kvalita rozdělení. Kohonenova mapa zde rozdělí správně jen zhruba dvě třetiny obrázků a to navíc i z původní trénovací množiny. Vypsáním výsledků se ukázalo, že tato síť téměř spolehlivě rozpozná obrázky bez bobulí, nedokáže však správně rozdělit obrázky s bobulemi a jejich částmi. Její učení navíc trvá velice dlouho.

Úkol B

Ve druhém úkolu byla naimplementována vrstvená dopředná umělá neuronová síť (třída FFNN v souboru `net_feedforward_Keras.py`). I tato třída je pečlivě zdokumentována ve zdrojovém kódu, je dbáno na zachování variability parametrů, využívá třídu `ImagePreProcessor` a je obalena třídou `Almighty` k testování (soubory `main.py`, `main_feedforward_Keras.py`). Využívá nástroje `tensorflow` a `keras`.

Testování bylo opět velmi strohé z časových důvodů. Výstupy z něj jsou v souborech `output/3`, `output/4` a `output/5`. Zde je překvapivě nízká schopnost rozpoznání testovací množiny, zatímco trénovací množinu se síť naučí zcela bezchybně. Nabízí se možnost, že zvolený počet iterací byl příliš vysoký, ale při menším počtu iterací se přesnost nezlepšila.

Nejlépších výsledků dosahuje síť s více skrytými vrstvami, avšak rozdíl není markantní.

Úkol C

V posledním úkolu byla realizována implementace konvoluční umělé neuronové sítě (třída CNN v souboru `net_convolution.py`) a to dle stejného standardu, jako v předchozích bodech. Konvoluční neuronová síť zpracovává přímo obrázky, takže nevyužívá třídu `ImagePreProcessor`. Stejně jako ostatní je obalena testovací třídou `Almighty` a testována v souborech `main.py` a `main.py` a `main_convolution.py`. Využívá nástroje `tensorflow` a `keras`.

Ani zde, bohužel, nezbylo na testování mnoho času, nicméně z výstupů v souboru `output/3` je jasné, že tato síť dosahuje zdaleka nejlepších výsledků. Tak jako dopředná vícvrstvá síť i konvoluční zcela bezchybně zařadila všechny trénovací obrázky, navíc dosahuje osmdesátiprocentní přesnosti na testovací množině. I zde se promítlo určité zlepšení při zvýšení počtu skrytých vrstev, ale taktéž není výrazné.

Zhodnocení

Dalece nejpomalejší přístup k výsledkům poskytuje Kohonenova mapa. Je zřejmě výpočetně náročná a vzhledem k explicitní implementaci nevyužívá multiprocesorové zpracování, oproti tensorflow. Výsledky jsou navíc opravdu špatné.

Naproti tomu vrstvená dopředná umělá neuronová síť poskytuje výsledky velmi rychle, avšak její přesnost na testovací množině se ukázala být přímo ubohá.

Nad konvoluční neuronovou sítí jsem strávil asi nejvíce času, nicméně poskytuje nejkvalitnější výsledky a v rozumném čase.

Lze však s jistotou říci, že ani jedna síť není dostatečně spolehlivá. Bohužel jsou však výpočty časově náročné a na optimalizaci nezbyl čas.