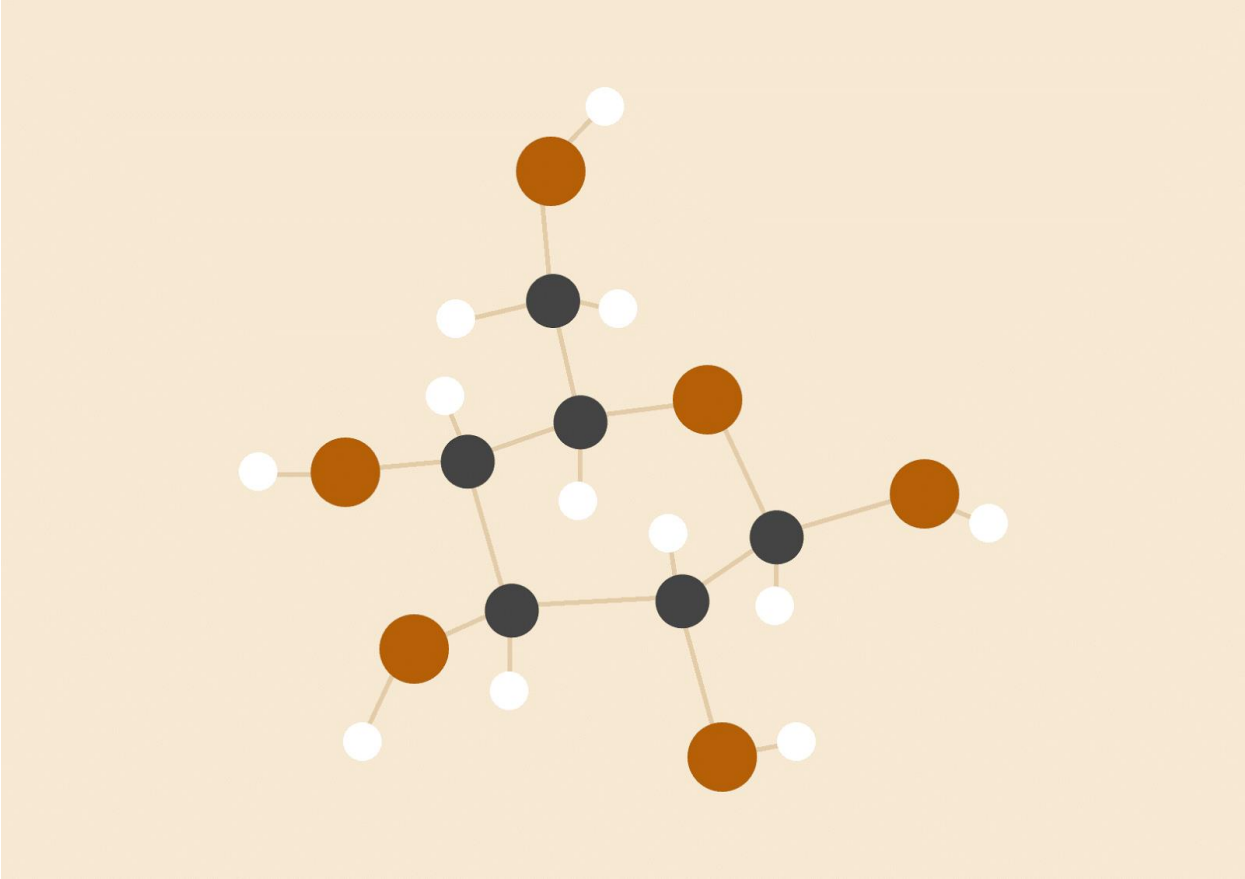
MOZGALO TEHNIČKA DOKUMENTACIJA

Nenadzirana kategorizacija slika



Sadržaj

1. UVOD 1

2. KORIŠTENE TEHNOLOGIJE 3

3. OPIS RJEŠENJA 8

3.1. Koraci 8

4. REZULTATI 9

5. DODACI OSNOVNOM ZADATKU 11

6. ZAKLJUČAK 12

7. REFERENCE 14

1. UVOD

Mozgalo je edukativno studentsko natjecanje u dubinskoj analizi velikih količina podataka i izradi Big Data rješenja, odnosno procesu pronalaženja korisnih informacija iz velikih količina strukturiranih i nestrukturiranih podataka. Pri tome se koriste različite statističke metode, tehnike strojnog učenja i umjetne inteligencije. Prvo je i jedino studentsko natjecanje takvoga tipa u Hrvatskoj te se održava na Sveučilištu u Zagrebu te pridruženim fakultetima ostalih hrvatskih sveučilišta u okviru kojeg studentski timovi izrađuju vlastita analitička rješenja.

Zadatak na ovogodišnjem natjecanju jest nenadzirana kategorizacija slika (engl. *unsupervised image classification*) tj. grupiranje (engl. i dalje *clustering*) slika prema njihovom sadržaju. Svaki tim dobio je neoznačeni skup podataka sa slikama koje treba svrstati u određeni broj kategorija. Zadatak je odrediti broj kategorija te svaku sliku svrstati u jednu od njih.

Skup podataka sastoji se od gotovo 7000 slika u boji, a njihova raspakirana veličina iznosi 95MB. Slike su različitih dimenzija, uz ograničenje da je najveća dimenzija pojedine slike (visina ili širina) 200 piksela, a pohranjene su u različitim formatima (JPEG, PNG, GIF, TIFF).

Kao što je već prije spomenuto, u rješavanju ovakvih problema koriste se metode strojnog učenja. Strojno učenje je programiranje računala tako da optimiziraju neki kriterij uspješnosti temeljem podatkovnih primjera ili prethodnog iskustva. Mi kao programeri posjedujemo model koji je definiran do na neke parametre, a učenje se svodi na optimizaciju parametara modela temeljem podataka. Razloga za njegovu uporabu ima mnogo, a od bitnijih valja spomenuti složene probleme (problemi za koje ne postoji ljudsko znanje o procesu ili ljudi ne mogu dati objašnjenje o procesu - npr. raspoznavanje govora), ogromne količine podataka (otkrivanje znanja u skupovima podataka - engl. *Data mining*) te sustave koji se dinamički mijenjaju za koje je potrebna prilagodba (npr. prilagodba korisničkih sučelja).

Pri radu s velikim količinama podataka potrebno je prionuti u potragu za određenim metodama ubrzanja. Jedna od metoda je paralelizacija. Paralelizirati se može na različitim skalama. Najčešće se paralelizira korištenjem multiprocesorskih okruženja koji su prisutni u više manje svakom današnjem računalu. S druge strane postoje distribuirani sustavi koji se bave paralelizacijom na razini više umreženih računala. Uz navedene, postoji i paraleliziranje na grafičkoj kartici koje je korišteno u ovom radu. Takav način paraleliziranja ima jako velik potencijal budući da se na grafičkoj kartici nalazi mnogo procesora s mogućnošću paralelnog procesiranja.

2. KORIŠTENE TEHNOLOGIJE

FACEBOOK RESNET [1]

Facebookov ResNet (engl. *Deep Residual Network*) je radni okvir (engl. *framework*) koji implementira treniranje neuronske mreže na ImageNet setu (baza podataka slika organizirana prema WordNetovoj hijerarhiji). Unaprijed trenirani modeli su otvorenog koda te su dostupni na službenom Facebookovom repozitoriju na Githubu.

GIT [2]

Git je sustav za kontrolu verzija (engl. *version control system, VCS*) koji služi za praćenje promjena u računalnim datotekama i koordinaciju posla između više ljudi. Primarno se koristi u razvijanju softvera, ali se može koristiti u praćenju promjena u bilo kojim datotekama. Kao sustav za distribuiranu kontrolu verzije Git cilja prema brzini, integritetu podataka, te podršci za distribuirane nelinearne poslove.

JUPYTER NOTEBOOK [3]

Jupyter bilježnica je biblioteka otvorenog koda web aplikacija koja dopušta da se stvaraju i dijele dokumenti koji sadrže interaktivni kod, jednadžbe, vizualizacije i tekstove objašnjenja. Neki od načina korištenja su: čišćenje podataka i transformacije, numeričke simulacije, statističko modeliranje, te strojno učenje. Jupyter bilježnica podržava preko 40 programskih jezika uključujući jezike popularne u znanosti o podacima kao Python, R, Julia te Scala.

LUA [4]

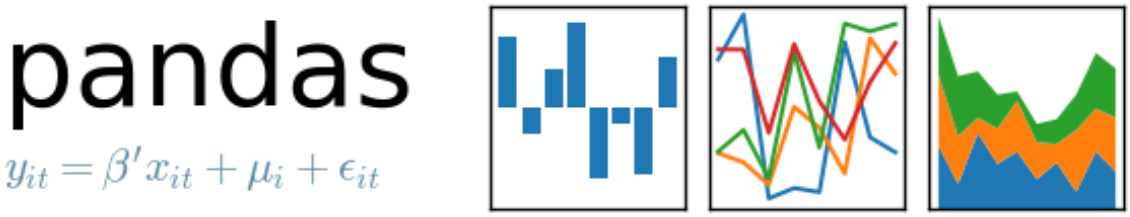
Lua je skriptni programski jezik koji je vrlo lagano ugraditi u široki spektar aplikacija, od igara do web aplikacija i procesiranja slika, za što smo ga u konačnici i koristili.

NLTK [5]

NLTK je vodeća platforma za izgradnju python programa i aplikacija za rad sa podacima o ljudskom jeziku. Nudi mnogo sučjelja za rad sa leksičkim reusrsima (od najbitnijih vrijedi spomenuti WordNet koji je korišten u radu), kako i mnoge biblioteke za klasifikaciju, tokenizaciju, označavanje, parsiranje te mnoge druge.

PANDAS [6]

Pandas je BSD-licencirana biblioteka otvorenog koda koja pruža lagane za korištenje podatkovne strukture i alate za analizu podataka visokih performansi za Python. Dizajniran je za olakšavanje rada s relacijskim ili označenim podacima, te je zapravo jedan od najbitnijih alata za analizu podataka u Pythonu.



PYTHON [7]

Python programski jezik je široko korišteni programski jezik visoke razine koji se koristi za opće programiranje. Python programski jezik se ne kompajlira nego se izvodi pomoću interpretera, podržava nekoliko programskih paradigmi uključujući objektno orijentiranu, imperativnu, funkcijsko programiranje te proceduralnu programsku paradigmu. Python ima veliku i opsežnu standardnu biblioteku. Python interpreteri su dostupni za mnogo operativnih sustava, time dopuštajući da se Python kod vrti na raznim sustavima.

SCIKIT-LEARN [8]

Scikit-learn (bivši scikits.learn) je biblioteka za strojno učenje u programskom jeziku Python. Sadrži implementacije različitih algoritama koji su tipični za strojno učenje, poput klasifikacije, regresije i grupiranja uključujući stroj potpornih vektora (engl. *Support Vector Machine*), slučajne šume (engl. *Random Forests*), gradijentni boosting, k-means, DBSCAN te mnoge druge. Također je dizajniran uzimajući u obzir Pythonove numeričke i znanstvene biblioteke NUmPy [9] i SciPy [10].

TORCH [11]

Torch je znanstveni komputacijski radni okvir (engl. *framework*) koji pruža široku potporu za algoritme strojnog učenja koji stavlja grafičke kartice (engl. *GPU - Graphical Processing Unit*) na prvo mjesto. Lagan je za korištenje i iznimno efikasan zahvaljujući brzom skriptnom jeziku Lui u kojem je pisan te se bazira na C/CUDA implementaciji.

WORDNET [12]

WordNet je velika leksikografska baza podataka engleskog jezika. Imenice, glagoli, pridjevi, prilozi su grupirani u skupove kognitivnih sinonima (engl. i dalje *synsets*), gdje svaka predstavlja izrazit koncept. Synsets su povezani konceptualno-semantičkim i leksičkim vezama. WordNetova struktura ga čini korisnim alatom u računalnoj lingvistici te procesiranju prirodnog jezika.WordNet je besplatan i javno slobodan za preuzimanje.

3. OPIS RJEŠENJA

Rješavanje problema odvijalo se u nekoliko koraka koji će biti objašnjeni u sljedećem odlomku.

3.1. Koraci

Izvlačenje značajki (eng. feature extraction)

Da bismo došli do značajki koje će nam kvalitetno reprezentirati naše slike okrenuli smo se području dubokog učenja koje se pokazalo efektivnim za izvlačenje značajki iz slike. Konkretna biblioteka koju smo koristili u tu svrhu je Torch, a model s kojim smo radili je unaprijed istrenirana Facebookova inačica ResNeta.

Odabir značajki (feature selection)

Kod odabira značajki koristili smo klasični postupak odabira pod nazivom PCA [13] (engl. *Principal Component Analysis*). PCA je metoda redukcije dimenzija koja početne varijable transformira u prostor manjih dimenzija tako što varijable koje su linearno korelirane zamijeni linearno nekoreliranim. Za odluku o broju značajki odlučeno je pomoću pretraživanja rešetke broja značajki i ustanovljeno da je najmanji broj koji odgovara od promatranih parametara jednak 1000. Način na koji je odabran optimalan broj je naša subjektivna procjena koja je ovisila o tome kako je naš model odredio koji su clusteri.

Opis izlaza modela

Za problem clusteriranja korišten je model Gaussove mješavine. Razlog korištenju tog modela je veća moć razdvajanja od recimo modela K-means [14]. Uz određivanje klase dodani su i opisnici pojedine kategorije koji se sastoje od nekoliko riječi koje opisuju što se nalazi u tom clusteru. Da bismo odredili taj opisnik korišten je WordNet, unaprijed istrenirani klasifikacijski model, koji nam je s određenom vjerojatnošću dao top pet predikcija za danu sliku i korištenjem dobivenih imena i njihovih hiperonima određivali smo što se nalazi u tom clusteru.

4. REZULTATI

Budući da se kod problema grupiranja ne može eksplicitno izračunati broj clustera iz samih podataka morali smo subjektivnim vizualnim metodama poput metoda lakta [15] (engl. *elbow method*) i rezultata silueta (engl. *silhouette score*) [16] određivati broj clustera.

Za dani skup podataka navest ćemo samo ukratko par primjera koji će prikazati što se događa promjenom broja clustera.

Za 4 model daje sljedeće clustere: (avione, kuće, zgrade, ovce, motore, aute), cvijeće, ljude i kolače.

Za 10 model daje sljedeće clustere: avione, automobile, cvjetove, geparde, kolače, kuće u predgrađu, motocikle, ljude, ovce i naposljetku stariju gradnju.

Za 14 model će dati sljedeće clustere: avione, aute, cvijeće u krupnom planu, geparde kojima je u pozadini stepa, kolače, kuće u predgrađu, motore, ljude, ovce, stariju gradnju, motore, cvijeće koje ima više cvjetova, geparde u krupnom planu, cvijeće okrugla oblika.

Za 18 model daje sljedeće clustere: horizontalno orijentirane avione, automobile, cvijeće, geparde sa stepskom pozadinom, kolače, kuće u predgrađu, motore, ljude, ovce, starije građevine, motore, cvijeće koje ima više cvjetova, geparde u krupnom planu, cvijeće okrugla oblika u krupnom planu, dvije skupine kolača, cvijeće, kolače.

Za 25 model daje sljedeće clustere: cvijeće, kuće u predgrađu, motore koji nemaju bijelu pozadinu, avione u zraku, ljude, stare zgrade gotičke arhitekture, kolače, automobile, geparde, ovce, cvijeće okrugla oblika, prepoznaje jedan auto, avione na pisti, motore s velikim dijelom bijele pozadine, kolače, motore sa skoro čisto bijelom pozadinom, avione, geparde, starije građevine okrugla oblika, starije motore, geparde u krupnom planu, razne aute, cvijeće raznog oblika, cvijeće većinom žute boje te kolače.

Kao što se vidi iz navedenih rezultata, povećanjem broja clustera model će naći suptilnije razlike među clusterima, primjerice, geparde sa stepskom pozadinom i

geparde u krupnom planu ili avione na pisti i avione u zraku. U jednom od pokusa dobili smo da model razdvoji muškarce i žene u posebne grupe.

5. DODACI OSNOVNOM ZADATKU

Uz rješavanje osnovnog zadatka grupiranja slika koje su nam bile dodijeljene, odlučili smo se i na istraživanje kako bi naše rješenje radilo na nekom drugom podatkovnom skupu. Nakon promatranja raznih dostupnih opcija, odlučili smo se za malo teži skup slika (Caltechov skup [17]) koji se sastoji od 30607 slika, a sadrži objekte iz 256 kategorija.

Naše rješenje je uspješno grupiralo slike u kategorije, a također smo odlučili napraviti i hijerarhijsko clusteriranje te smo dobili rješenje koje je dostupno na sljedećoj poveznici (slika je prevelika za stavljanje izravno u dokumentaciju stoga se nalazi na sljedećem linku):
https://www.dropbox.com/s/v83cjnq4781n0nv/hierarchical.png?dl=0.

Druga stvar koju smo isprobali odnosila se na početni dataset. Kada smo dobili potrebne kategorije, proučavali smo načine kako programskim putem odrediti što se točno nalazi u pojedinoj skupini slika. Kao polazišnu točku uzeli smo Facebookovu unaprijed istreniranu mrežu koja kao rezultat vraća oznake i vjerojatnosti pojedinih oznaka. Probali smo par načina kako dobiti optimalni rezultat: za svaku klasu top 5 primjera najbližih centroidima, top 10, top 15 te naposljetku 10 nasumično odabranih primjera iz clustera, što se pokazalo kao najbolje rješenje. Za svaki primjer smo tako tražili njihove opisnike, kao i hiperonime tih opisnika. Na kraju smo prošli kroz svaki reprezent za klasu i kao opis klase uzeli onaj opisnik koji se najviše puta pojavio. Rezultat je bio raznolik, za aute, motore i geparde smo uglavnom dobiveni dobri opisi, dok je za neke druge clustere bio lošiji, npr. ljudska lica nisu bila precizno opisana zato što se osim njih na slici nalaze još i odjeća koja je također s velikom vjerojatnošću pronađena.

6. ZAKLJUČAK

Gaussov miješani model je kao izlaz dao različita rješenja za ponovna pokretanja, ali to je zato što je bio omogućen nasumičan odabir te su se svaki put drugačije inicijalizirala početna središta. Ako se modelu dao dovoljan broj clustera on bi dobro odvajao i klasificirao primjere do te mjere da je odvojio ljude po spolovima u primjerima gdje su lica.

Inženjerstvo značajki [18] (engl. i dalje *feature engineering*) je proces u kojem iz raznih izvora i ulaza podataka izlučujemo reprezentativne značajke (informacije koje nam predstavljaju znanje i po kojima su neki podaci različiti od drugih podataka). Feature engineering je temelj u aplikaciji strojnog učenja, te je težak i računalno skup. Potrebu za ručnim izdvajanjem značajki možemo zamijeniti automatiziranim procesom koji se zove učenje značajki.

Duboko učenje [19] (engl. *Deep learning*) je jedno od područja strojnog učenja koje se bavi algoritmima inspiriranima strukturama i funkcijama mozga poznatijima pod nazivom umjetne neuronske mreže (engl. *Artificial Neural Networks*).

Proces grupiranja slika je zanimljivo područje u kojem zapravo dolazi do sukoba između prethodna dva koncepta. Duboko učenje poboljšava mogućnosti algoritama da pronađu optimalne značajke i pronađu indirektne veze između značajki i ciljeva. Uzevši to u obzir, zašto onda još uvijek ručno modeliramo značajke ako novi algoritmi mogu naći optimalne značajke automatski?

Zato što duboko učenje još uvijek nije “magična crna kutija” koja će nam otkriti sve moguće strukture u bilo kakvim tipovima podataka. Ono zbog svog automatskog izvlačenja značajki dopušta programerima (znanstvenicima) da smisle apstraktnije značajke i apstraktnije strategije za otkrivanje istih. Problem se tako onda pomiče s uglavnom dobro razumljivog i detaljno obrađenog područja učenja značajki na slabije poznato područje percepcije i inženjerstva akcijske meta-strukture (engl. *action meta-structure engineering*). Tako učenje značajki se polako pretvara u smišljanje strategijskih struktura značajki (engl. *feature strategy architecture*).

Mozgalo projekt bio je jedno zanimljivo i nadasve zabavno iskustvo, te se nadamo da će se nastaviti održavati u ovakvom formatu jer nama studentima nudi jedno novo iskustvo rada na projektu kakvo rijetko možemo steći za vrijeme studija, kao i mogućnost povezivanja s raznim firmama sponzorima te širenje znanja.

KLJUČNE RIJEČI: strojno učenje, grupiranje, nenadzirano učenje, duboko učenje, značajke, izlučivanje značajki, neuronska mreža, word embedding, GMM, python

7. REFERENCE

1. https://github.com/facebook/fb.resnet.torch, Facebook ResNet, zadnje pogledano 19.5.2017.
2. https://gitlab.com, Gitlab, zadnje gledano 19.5.2017.
3. http://jupyter.org, Jupyter Notebook, zadnje gledano 19.5.2017.
4. https://www.lua.org, Lua, zadnje gledano 19.5.2017.
5. http://www.nltk.org, Natural Language Toolkit, zadnje gledano 21.5.2017.
6. http://pandas.pydata.org, Pandas, zadnje gledano 19.5.2017.
7. https://www.python.org, Python, zadnje gledano 19.5.2017.
8. http://scikit-learn.org, Scikit-learn, zadnje gledano 19.5.2017.
9. http://www.numpy.org, Numpy, zadnje gledano 19.5.2017.
10. https://www.scipy.org, Scipy, zadnje gledano 19.5.2017.
11. http://torch.ch, Torch, zadnje gledano 19.5.2017.
12. https://wordnet.princeton.edu, WordNet, zadnje gledano 19.5.2017.
13. https://en.wikipedia.org/wiki/Principal_component_analysis, PCA, zadnje gledano 19.5.2017.
14. https://en.wikipedia.org/wiki/K-means_clustering, k-means, zadnje gledano 19.5.2017.
15. https://en.wikipedia.org/wiki/Elbow_method_(clustering), metoda lakta, zadnje gledano 19.5.2017.
16. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html, scikit-learn, zadnje gledano 19.5.2017.
17. http://authors.library.caltech.edu/7694/, Caltech 256 skup podataka, zadnje gledano 19.5.2017
18. https://en.wikipedia.org/wiki/Feature_engineering, inženjerstvo značajki, zadnje gledano 19.5.2017.
19. https://en.wikipedia.org/wiki/Deep_learning, duboko učenje, zadnje gledano 19.5.2017.