Vizuelizacija ogromnih količina podataka uz primenu tehnika optimizacije ili učenja

Seminarski rad u okviru kursa
Računarska inteligencija
Matematički fakultet

Vukajlović Nikoleta, Trtica Mateja
mi16144@matf.bg.ac.rs, mi16166@matf.bg.ac.rs

17. jun 2020.

Sažetak

Ovaj rad se bavi vizuelizacijom velikog skupa podataka i to tehnikama učenja i topološkom analizom podataka. Biće prikazani i upoređeni rezultati dobijeni autoenkoder neuronskim mrežama, Mapper algoritmom i Kohonenovim samoorganizujućim mapama.

Sadržaj

1 Uvod

U svetu velikih podataka, vizuelizacija je neophodna za analizu ogromnih količina informacija i donošenje odluka zasnovanih na podacima. Poželjno je pronaći razne načine za prikaz podataka koji omogućavaju kvalitativno razumevanje istih putem direktne vizuelizacije. Ona predstavlja grafički prikaz informacija i podataka i pruža pristupačan način da se vide i razumeju trendovi, veza između podataka, podaci izvan granica i obrasci u podacima. U ovom radu bavićemo se metodama koje preslikavaju podatke u raštrkani skup tačaka, kao i metodama koje imaju mogućnost predstavljanja podataka u vidu grafova.

Neuronske mreže imaju širok spektar primena, a ovde će biti objašnjeno na koji način se one mogu iskoristiti za vizuelizaciju i kakvi rezultati su postignuti primenom na skup podataka koji sadrži informacije o karcinomu dojke. Konkretno, korišćene su autoenkoderi i samoorganizujuće mape. Takođe, dobijeni rezultati biće upoređeni sa rezultatima dobijenim topološkom analizom podataka.

Ovde se postavlja pitanje zlatnog standarda, odnosno šta je tačno. Sa tim u vezi, predstavićemo redukciju trodimenzionalne mačke na dvodimenzionalnu, jer u tom slučaju znamo šta je očekivani rezultat.

2 Algoritmi

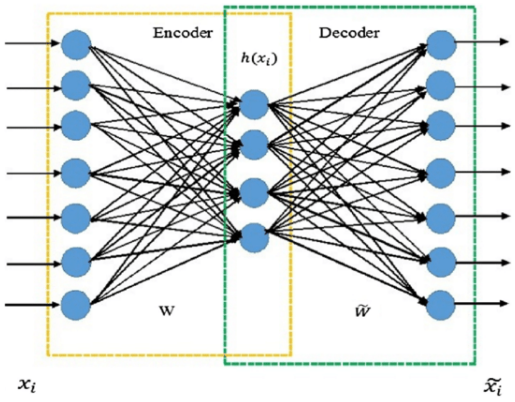
U nastavku će zasebno biti opisane karakteristike algoritama koji su korišćeni u istraživanju. Akcenat je bačen kako na sam opis algoritma, tako i na način implementacije. Svi algoritmi su pisani u programskom jeziku *python*.

2.1 Autoenkoderi

Autoenkoder je neuronska mreža koja je trenirana da približno kopira svoj ulaz na izlazni sloj. Autoenkoder ima skriveni sloj čiji se izlaz koristi za dimenzionalnu redukciju ulaza. Izlaz tog skrivenog sloja se naziva latentna reprezentacija. Mreža se može podeliti na dva dela: enkoder i dekoder. Enkoder predstavlja funkciju $h = f(x)$ koja slika ulaz u latentnu reprezentaciju h, a dekoder funkciju $g(h)$ koja pokušava da vrati latentnu reprezentaciju h u ulaz x [2]. Arhitektuta autoenkodera je prikazana na slici 1.

Kada bi jedina svrha autoenkodera bila kopiranje ulaza na izlaz, on bi bio beskoristan. Umesto toga, teži se da se autoenkoder obučava tako da latentno predstavljen h poprimi korisna svojstva. Jedan način da se dobije autoenkoder korisnih karakteristika jeste ograničavanjem da h ima manje dimenzije nego x i takav autoenkoder se naziva *nepotpuni* [3]. Baš on je u ovom radu korišćen za smanjenje dimenzionalnosti radi vizuelizacije. Sa odgovarajućim ograničenjima autoenkoderi mogu naučiti projekcije podataka koje su zanimljivije od PCA i drugih popularnih tehnika.

U ovom radu autoenkoderi su implementirani pomoću *Keras* biblioteke. To je biblioteka za duboko učenje visokog nivoa koja za izračunavanje koristi biblioteke nižeg nivoa kao što je *TensorFlow*. Visok nivo podrazumeva da Keras pruža funkcionalnosti koje podržavaju stvaranje modela sa fokusom na ideji, a ne na konkretnoj implementaciji. Modelovanje arhitekture modela u Kerasu se zasniva na sekvenci slojeva modela. Keras pruža funkcionalnosti modela mašinskog učenja preko metoda klase *Model*.



Slika 1: Arhitektura autoenkodera

Keras posmatra bilo koju transformaciju podataka kao sloj u arhitekturi mreže. Naprimer, kompletno povezani sloj je Keras sloj u mreži, sa svojom funkcijom aktivacije, regularizacijom i ograničenjima, ali su i slojevi aktivacije i regularizacije takode slojevi sami po sebi. Tako se može napraviti jedan kompletno povezani sloj pomoću nekoliko slojeva u kontinuitetu, počev od kompletno povezanog sloja sa linearnom funkcijom aktivacije, praćcen slojem aktivacije, praćen slojem regularizacije. Slojevi koji su dostupni u Kerasu, a korišćeni su prilikom implementacije autoenkodera su:

- Input - sloj ulaza u neuronsku mreu. Ovaj sloj transformiše ulazne podatke u Keras tenzor, koji predstavlja tenzor biblioteke koju koristi keras. onstruktor ima opcioni parametar *shape* koji sadrži oblik ulaznog tenzora.
- Dense - regularni kompletno povezani sloj neuronske mreže. Obavezni parametar konstruktora ovog tipa sloja je broj neurona u sloju. U konstruktoru se takođe prosleđuje i tip funkcije aktivacije. Inicijalizacija težina se može podešavati pomoću parametra *kernel initializer*. Dostupne su i mogućnosti regularizacije [1].

2.2 Samoorganizujuće mape

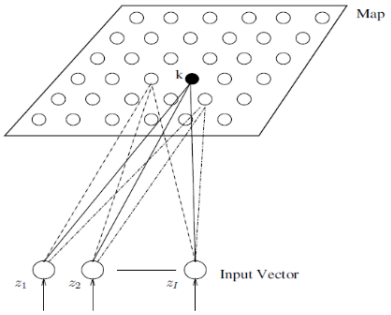
Samoorganizujuća mapa(SOM) je tip veštačke neuronske mreže čija obuka se vrši nenadgledanim učenjem kako bi se dobila niskodimenzionalna (najčešće dvodimenzionalna), diskretna reprezentacija ulaznih uzoraka. Ovakva diskretna reprezentacija podataka zove se mapa [5]. Ideja je zadržavanje topološke strukture ulaznog prostora, odnosno ukoliko su dva podatka bliska u ulaznom prostoru, biće bliska i u izlaznom. Prvi ih je opisao finski naučnik Teuvo Kohonen, pa se zato i nazivaju Kohonenove mape.

Samoorganizujuće mape rade u dve faze: učenje i mapiranje. Učenje izgrađuje mapu pomoću ulaznih podataka. Ovaj proces je kompetitivan, odnosno neuroni se takmiče za dozvolu da se aktiviraju. Ovaj proces se naziva i kvantizacija vektora. Mapiranje vrši klasifikaciju ulaznog vektora. Šema SOM-a je prikazana na slici 2.

SOM se sastoji iz komponenti koje se nazivaju čvorovi ili neuroni. Svaki neuron na poziciji (k, j) ima vektor težina nasumično podešen na:

$W_(k, j) = (w_(kj1), w_(kj2), .., w_(KJI))$, istih dimenzija kao i vektor ulaznih podataka, kao i poziciju u prostoru mape. Procedura smeštanja vektora iz prostora podataka u mapu sastoji se iz:

1. pronalaženja neurona čiji vektor težina ima vrednosti najbliže vektoru iz prostora podataka
2. nad izabranim neuronom vrši se korekcija težina na osnovu ulaznog podatka, kao i korekcija težina susednih neurona proporcionalno njihovoj udaljenosti od izabranog neurona
3. dodeljivanja koordinata mape ovog neurona vektoru.



Slika 2: Šema samoorganizujuće mape

Opisane osobine mape čine je korisnom za vizuelizaciju visoko-dimenzionog prostora na nisko-dimenzioni. U tu svrhu, korišćena je *python* biblioteka MiniSom.

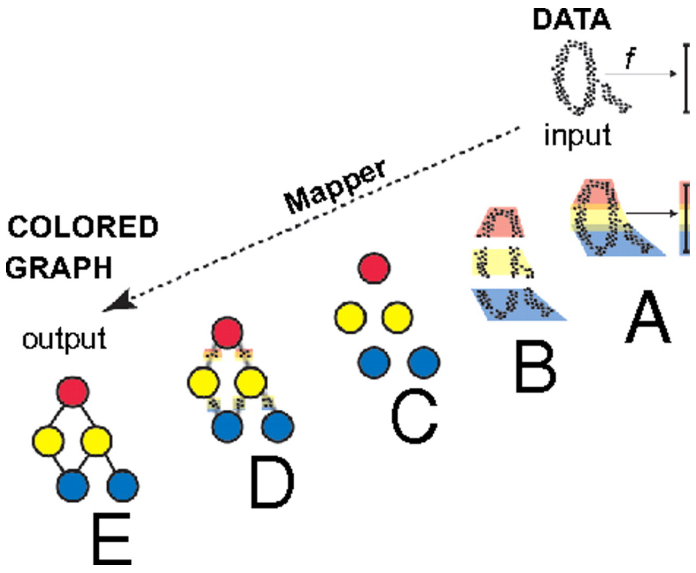
2.3 Mapper algoritam

Mapper algoritam predstavlja sredstvo topološke analize podataka, nedavno razvijenog oblika analize podataka koji ima veći stepen robusnosti u odnosu na metode kao što su PCA ili višedimenziono skaliranje. Konkretno, Mapper ima sleceća svojstva:

1. njegov izlaz je kombinatorni grafikon, a ne linearni prostor ili raštrkani skup tačaka u malom dimenzionalnom euklidskom prostoru
2. podaci se mogu gledati na različitim skalama rezolucije, što je korisno za razlikovanje stvarnih karakteristika i artifakata
3. metoda ima mogućnost prepoznavanja detalja čak i u velikim skupovima podataka
4. metoda se može primeniti u bilo kojoj situaciji u kojoj postoji pojam sličnosti ili blizine, ne samo u Euklidskim podacima [4].

Ovaj algoritam identifikuje oblik skupa podataka pomoću prethodno dodeljene filter funkcije. U svom najjednostavnijem obliku, metoda u osnovi deluje na sledeći način: započinje se sa funkcijom f koja je definisana na podacima i fragmentiranjem raspona f u preklapajuće komade. Zatim odvojeno klasterujemo delove podataka koji se preslikavaju na svaki deo. Svaki lokalni klaster može biti posmatran kao posude tačaka podataka. Nakon što su sve tačke dodeljene po posudama podataka, dodaju se ivice koje povezuju posude: dve posude koje imaju zajedničke tačke podataka

se povezuju ivicom, čime se stvara graf čiji oblik obuhvata važne spekte oblika podataka. Posude se boje prosečnom vrednošću filter funkcije definisane na tačkama podataka unutar posude. Te numeričke vrednosti se pretvaraju u boje [4]. Slika 3 ilustruje kako Mapper pretvara skup tačaka kružnog oblika u kraf koji prepoznaje ovaj oblik.



Slika 3: Mapper započinje sa skupom tačaka i filter funkcijom f i proizvodi obojeni graf koji prepoznaje oblik podataka. (A) Slika funkcije f je podaljena u preklapajuće regione. (B) Svaki deo je klasterovan odvojeno. (C) Svaki klaster je predstavljen kao obojeni disk - posuda podataka. (D) Identifikuju se parovi posuda koji imaju zajedničke tačke. (E) Povežu se parovi posuda koji imaju zajedničke tačke.

Za implementaciju korišćena je biblioteka *kmapper*. Informacije koje se prosleđuju su:

- funkcije projekcije podataka - može se koristiti bilo koja funkcija iz matematike, statistike, ekonomije ili mašinskog učenja
- algoritam koji se koristi za prekrivanje preklapajućih intervala - u teoriji se može koristiti bilo koja veličina intervala, ali KepplerMapper podržava n-dimenzione intervale
- algoritam koji se koristi za klasterovanje unutar intevala - može se koristiti bilo koji algoritam i bilo koja metrika [6].

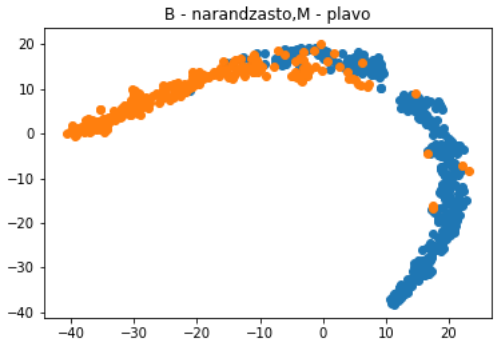
3 Primena algoritama

U ovom poglavlju će biti prikazani rezultati eksperimenata, kao i sam opis eksperimenata, korišćenih modela i podataka. Izabrani prostorni podaci imaju interesantna svojstva, koja omogućavaju da se uporede performanse ovih modela. Biće prikazane vizuelizacije u 2 dimenzije, a u kodu se mogu videti i rezultati dobijeni u 3 dimenzije.

3.1 Primena na skup podataka kancer dojke

Skup podataka *kancer dojke* sadrži 33 atributa, od kojih jedan predstavlja dijagnozu koja može biti M(maligna) i B(benigna). Maligne dijagnoze zauzimaju 62% , dok benigne zauzimaju 32% podataka. Sa tim u vezi, metode bi trebalo da vizuelizuju otprilike 2 klastera, pa su po tom osnovu upoređivane. U preprocesiranju podataka izbačena je samo jedna kolona, čije su sve vrednosti bile NaN.

U ovu svrhu je korišćen i T-SNE algoritam, koji ne spada u tehnike učenja ili optimizacije, ali je poznat kao algoritam koji vrši dimenzionu redukciju tako što čuva početne klastere. Zbog ovog svojstva, pokazao se dobro kao uporediva metoda. T-SNE je najsporiji od svih korišćenih metoda, međutim veoma lepo prikazuje postojanje 2 klastera. To se može videti na slici 4. Postoje podaci koji su upali u pogrešne klastere, ali se na primer primenom SVM-a mogu na veoma jasan način razdvojiti podaci.



Slika 4: Prikaz rezultata dobijenog T-SNE-om

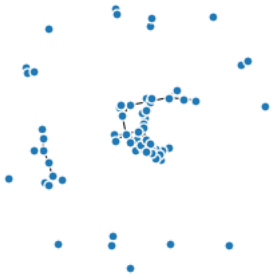
Rezultati Mapper algoritma prikazani su na slici 5. Ovde se može videti da je dobijen jedan veliki klaster, jedam mali klaster i šum. Podaci su između ostalog projektovani i algoritmom IsolationForest, koji vrši detekciju anomalija, tako da je on dostao uticao na rezultat rada. Broj intervala je postavljen na 10, sa preklapanjem od 0.2, a za klasterovanje unutar modela korišćen je algoritam K-sredina, sa postavljena 2 klastera.

Samoorganizujuće mape su u ovom slučaju dale najbeskorisniji rezultat. On je prikazan na slici 6. Ovde se vidi postojanje tačno jednog velikog klastera, što nije u skladu sa očekivanim rezultatima.

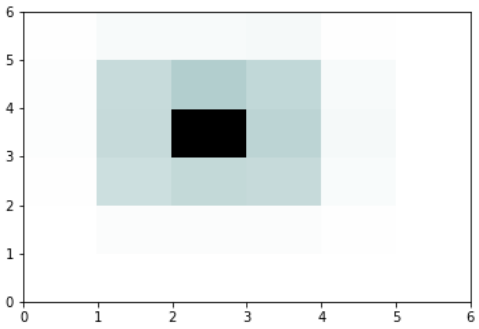
Kod autoenkodera napravljeni su modeli koji koriste 3 različite funkcije aktivacije, a to su:

1. *sigmoid* - sigmoidna (logička)
2. *relu* - ispravljena linearna
3. *tanh* - hiperbolički tangens.

Ove funkcije se koriste i u enkoderu i u dekoderu. Enkoder se sastoji od 2 skrivena sloja, koji koriste 200 i 100 neurona. Razlike u dobijenim modelima se mogu videti na slici 7. Može se videti da se model koji koristi hiperbolički tangens najlošije pokazao. Tačke su raštrkane po prostoru bez ikakvog uređenja, te se iz ovoga ne može ništa zaključiti. Dalje, iz druga dva modela se primenom nekih odgovarajućih metoda klasterovanja i može



Slika 5: Prikaz rezultata dobijenog Mapper algoritmom na kancer skup podataka



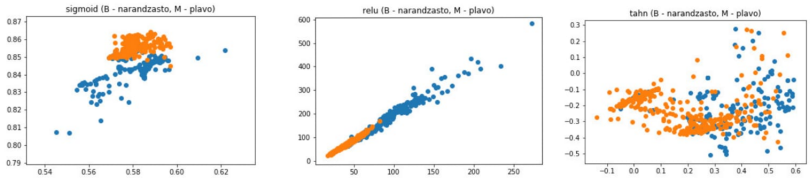
Slika 6: SOM primenjen na kancer skup podataka

zaključiti postojanje 2 klastera i određene količine podataka koji predstavljaju šum. Podaci koji pripadaju istom klasteru su uglavnom preslikani u blizke 2D koordinate, međutim razmak između postojećih klastera nije jasno vidljiv. Modeli su veoma nestabilni, odnosno prilikom svakog novog pokretanja algoritama dobijaju se rezultati koji nisu toliko povezani sa prethodnim.

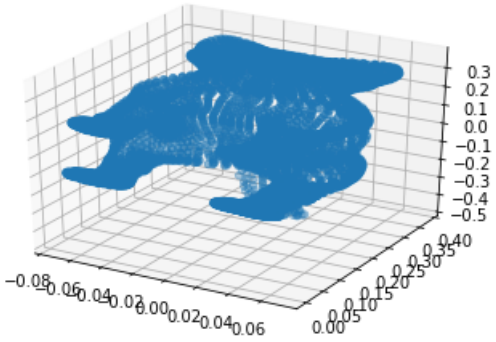
3.2 Primena na skup podataka mačka

Iz prethodnog poglavlja se da zaključiti da je pouzdanost prikazanih modela diskutabilna. Primenom algoritama na skup podataka *mačka*, ideja je bila da se proveri korektnost svih modela. Odnosno, uzet je skup podataka koji sadrži tačke na granicama 3D mačke i propušten kroz algoritme koji slikaju na 2D. Prikaz tačaka iz datog skupa podataka se može videti na slici 8.

Napravljena su 3 modela autoenkodera i to svaki sa po 2 skrivena sloja, kod kojih prvi ima 8 neurona u sloju, a drugi 6. Modeli se razlikuju u funkcijama aktivacije koje su korišćene, kao i kod prethodnog skupa podataka, a one su: sigmoid, relu i tahn. U svakom modelu date funkcije

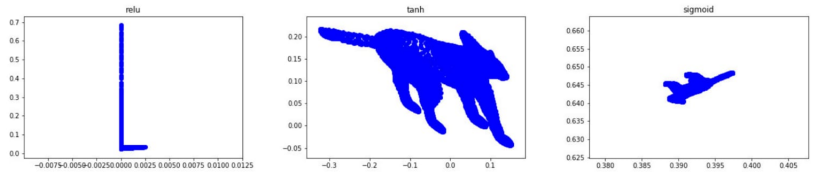


Slika 7: Dobijeni rezultati primenom (1) sifmoid, (2) tahn i (3) relu aktivacione funkcija



Slika 8: Prikaz 3D tačaka koje ilustruju mačku iz polaznog skupa

se koriste i enkoderu i u dekoderu. Rezultati dobijeni primenom ova tri modela predstavljeni su na slici 9. Ovaj rezultat dobijen je 10x10

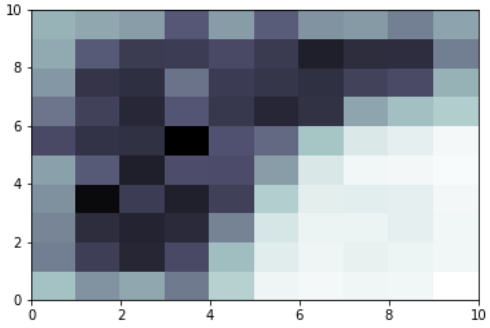


Slika 9: Rezultati dobijeni primenom aktivacionih funkcija: (1) relu, (2) tanh, (3) sigmoid

Iz priloženog se da zaključiti da se model dobijen hiperboličkim tangensom najbolje pokazao. Prilikom svakog pokretanja davao je vizueliza-

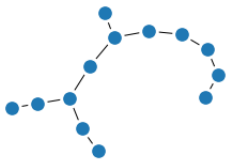
ciju iz koje se moglo zaključiti da je životinja u pitanju. Model koji koristi sigmoidnu funkciju aktivacije može da da dobro rešenje, međutim nije stabilan. Rezultat dosta zavisi od trenutka u kom se pokreće bez obzira što se koriste isti parametri, tako da su pri pokretanju dobijane i beskorisne vizuelizacije. Relu model se pokazao kao najlošiji.

Najlošije od sva tri algoritma i kod ovog skupa podataka pokazale su se samoorganizujuće mape. Vizuelizacija dobijena njihovom primenom se može videti na slici 10. U ovom slučaju je korišćena mapa 10x10, ali se bolji rezultati nisu dobijali ni postavljanjem drugih parametara. Na slici se vidi postojanje jednog velikog klastera, ali iz toga nije moguće zaključiti o čemu je reč.



Slika 10: Primena samoorganizujuće mape na skup podataka mačka

Kao i kod prethodnog skupa, Mapper algoritam je dao veoma dobre rezultate. Vizuelizacija dobijena njime prikazana je na slici 11. On je veoma dobro izvukao suštinu oblika polaznih podataka. Uzeto je 10 intervala, sa preklapanjem 0.2, a za klasterovanje unutar algoritma je korišćen DBSCAN.



Slika 11: Rezultat rada Mapper algoritma

4 Zaključak

Cilj ovog rada bio je da se uporede efikasnosti modela koji vrše vizuelizaciju ogromnih količina podataka. Svi rezultati ovih eksperimenata

su pokazali da postoji nada da se količina prostornih podataka koja se koristi za ulaz u algoritme mašinskog učenja može znatno smanjiti. Ovo smanjenje može da utiče na povećanje brzine i performansi drugih algoritama mašinskog učenja i pretrage nad ovim podacima. Iz priloženog se može se zaključiti da je model koji koristi Mapper algoritam najpouzdaniji i u slučaju provere očuvanja klastera i u slučaju provere korektnosti. Jedina prednost modela koji koristi arhitekturu autoenkodera jeste što se redukcija može izvršiti u n dimenzija. Međutim, ova osobina nije toliko značajna za vizuelizaciju, jer je uglavnom cilj vizuelizacije prebaciti polazni skup podataka u 2D ili 3D prostor. Najlošije rezultate dale su samoorganizujuće mape.

Literatura

[1] Francois Chollet. Building autoencoders in keras, 2016.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[3] Nathan Hubens. Deep inside: Autoencoders, 2018.

[4] Monica Nicolau, Arnold J. Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270, 2011.

[5] Abhinav Ralhan. Self organizing maps, 2018.

[6] Hendrik Jacob van Veen and Nathaniel Saul. Keplermapper. http://doi.org/10.5281/zenodo.1054444, Jan 2019.