Univerzitet u Beogradu
Matematički fakultet

Milica Radivojević

Novel dimensionality reduction technique

- Računarska Inteligencija -

Beograd, februar 2024.

Uvodni deo

U ovom radu opisana je kombinacija tehnika za redukciju dimenzionalnosti kroz primer u Python kodu.
Polazi se od pretpostavke problema u višedimenzionom prostoru, kao i pretpostavke da u ovom prostoru postoje krucijalniji podaci preko kojih se drugi mogu predstaviti.
Tehnike redukcije dimenzionalnosti smanjuju dimenziju prostora tako što svode prostor na najvažnije podatke, odnosno one preko kojih se ostali (nelinearni) podaci mogu predstaviti.

Kroz različita merenja, često će se dešavati da su promenljive međusobno zavisne (na primer kroz linearnu kombinaciju) pa bi zato bilo zgodno svesti skup na nezavisne promenljive, čime će suvišni podaci biti uklonjeni, a koji se mogu dobiti logičnim sledom.

Zamislimo da imamo rezultate nekog merenja vezanog za različite ljude. Rezultati sadrže informacije o visini, težini, broju godina, broju koraka koji dati čovek pređe u jednom danu, količinu hrane koju čovek pojede u jednom danu itd. Primenom neke tehnike za redukciju dimenzionalnosti, moguće je smanjiti dimenziju problema na samo, recimo, 3 dimenzije.
Na primer, svi parametri se mogu shvatiti kao logički sled iz informacija o težini, visini i broju godina, te smo uspeli da optimizujemo dati problem po broju dimenzija.

Najpoznatije tehnike su: UMAP, LDA, PCA, autoenkoderi, t-SNE,...

Domen upotrebe može biti vrlo različit: bioinformatika, mašinsko učenje, medicinska istraživanja...

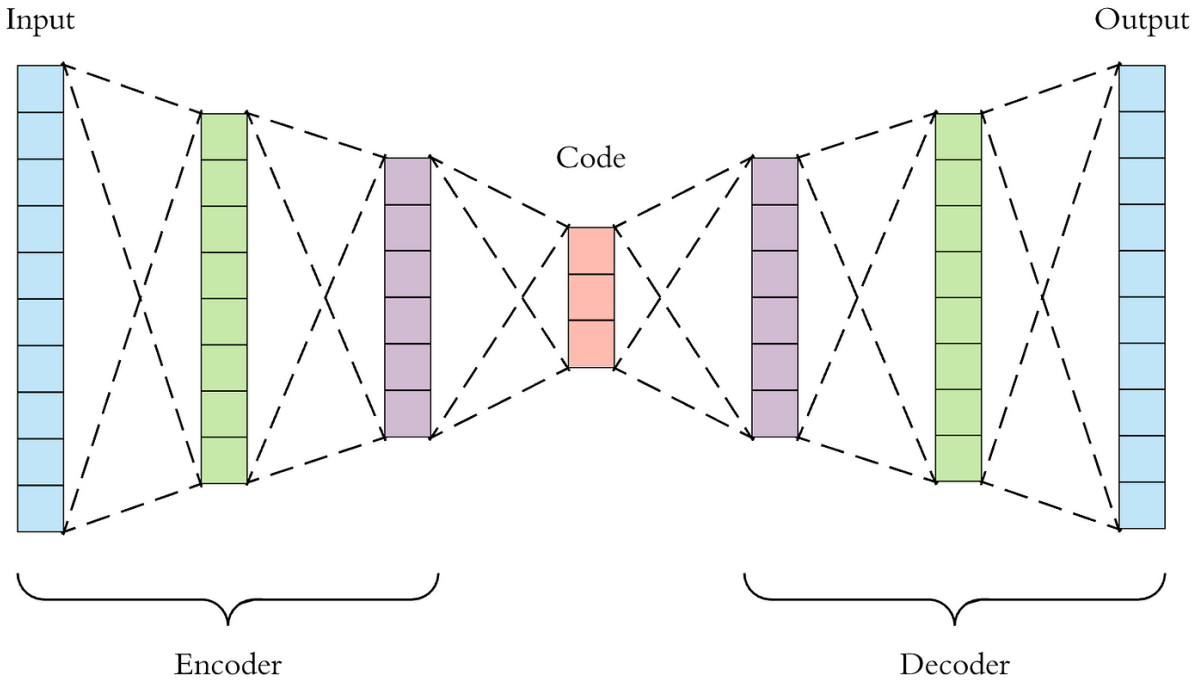
Opis vašeg rešenja zadatog problema

Primer koji sam navela u kodu, kombinacija je postojećih metoda za redukciju dimenzionalnosti.
Problem je bio sledeći: Kako da dobijenu sliku dimenzije 28x28 piksela prvo drastično smanjimo (do veličine 4x4 piksela), a da potom od te smanjene slike rekonstruišemo polaznu.
Koristila sam PCA tehniku i autoenkodere za rešenje ovog problema.

Autoenkoder je veštačka neuronska mreža koja ima za cilj prevođenje podataka sa ulaza u kompresovane podatke manjeg formata nego sa ulaza, a potom i rekonstruisanje ovih kompresovanih podataka u podatak koji je što bliži originalnom podatku sa ulaza.

Njegove 3 najbitnije komponente su:

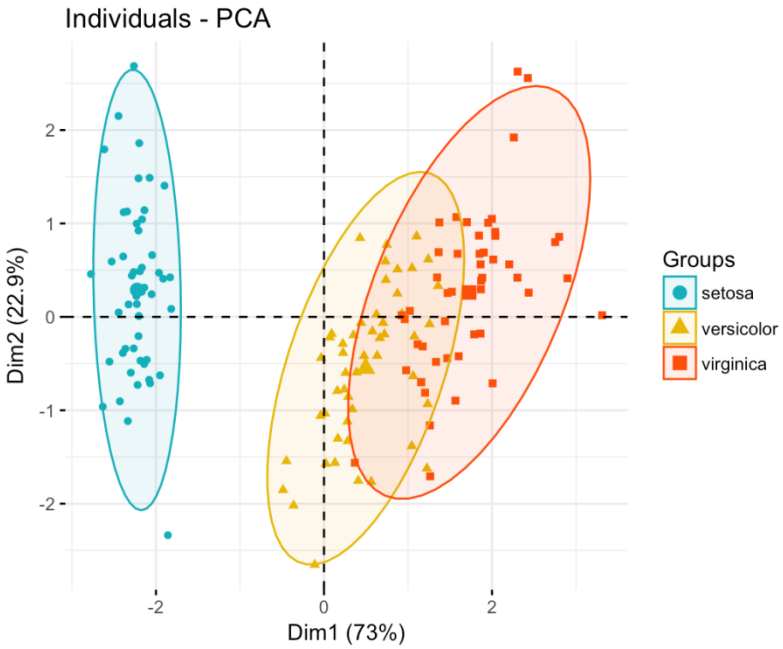
1) Enkoder – Ovde model uči kako da podatak sa ulaza (obično slika, u mom radu dimenzije 28x28 piksela) kompresuje u isti taj podatak, ali manjeg formata(u mom radu, kompresuje se u 4x4 piksela).
2) Usko grlo – ovaj sloj sadrži kompresovan podatak manjeg formata. Predstavlja najniži sloj u mreži.
3) Dekoder – Ovde model uči kako da kompresovani podatak sa ulaza (koji je izlaz iz prethodnog sloja) transformiše u podatak koji je što sličniji podatku sa ulaza u autoencoder.



Slika 1. Autoenkoder

Druga tehnika koju sam koristila jeste PCA. Evo koraka na kojima se ona zasniva.

1. Standardizacija: Na početku je potrebno da ulazni podatak oduzmemo do stanja u kom mu je srednja vrednost 0 i podelimo standardnom devijacijom za svaku vrednost svake promenljive.
2. Izračunavanje kovarijacione matrice: To je simetrična matrica koja ima kovarijanse između svih parova promenljivih.
3. Računanje sopstvenih vrednosti i sopstvenih vektora: Za kovarijacionu matricu iz prethodnog koraka računamo sve sopstvene vrednosti i sopstvene vektore, potom sortiramo sopstvene vrednosti i tako odlučujemo koliko ćemo komponenata zadržati.



Slika 2. Primena PCA tehnike

Iskombinovala sam ove dve tehnike na sledeći način.

Najpre sam učitala skup podataka (mnist/fashion mnist) koji se sastoji od slika dimenzija 28x28 piksela i na taj način dobila trening i test skupove, pomoću kojih sam trenirala i testirala modele.

Potom sam napravila PCA model.
Ovaj model ne prihvata slike kao ulazni parameter, te je zato najpre bilo neophodno transformisati ulazne podatke (slike) u vektore odgovarajućih dimenzija (funckijom reshape), bez gubitka podataka, tako što se sve kolone ulazne slike svrstaju u jedan veliki vektor.

PCA model potom trenira na trening skupu i tom prilikom uči kako da smanji suvišne informacije i svede problem na problem manje dimenzije.
Kao ulazni parameter ima promenljivu koja određuje koliko ima značajnih komponenti, a model dalje uči kako da ih odredi, odnosno koje podatke da odbaci.

Ovaj deo u mom radu je imao ulogu enkodera u autoenkoderu i upravo to ga razlikuje od klasičnog autoenkodera. Dimenziju slike sam smanjila PCA tehnikom do ulaza u decoder.

Dakle, nakon ovog koraka, trening i test skup čine vektori dimenzije 16, koji mogu da se reshape-uju u sliku veličine svega 4x4 piksela.
Upravo oni predstavljaju sada ulaz u decoder, koji od ove smanjene slike treba da rekonstruiše sliku sa ulaza u autoenkoder.

Dekoder prima vektor veličine 16, a potom kroz razne slojeve provlači ovaj vector i na svom izlazu daje vektor koji predstavlja što verniju kopiju odgovarajuće početne slike iz trening/test skupa sa ulaza u autoenkoder.

Poglavlje sa eksperimentalnim rezultatima

Svoj rad sam demonstrirala na dve biblioteke – fashion_mnist i mnist.
Rezultate testiranja sam prikazala vizuelno, tako što sam iz rezultata testiranja izvukla prvih 5 kao primer.

U svakom rezultatu sam prikazala 4 slike.

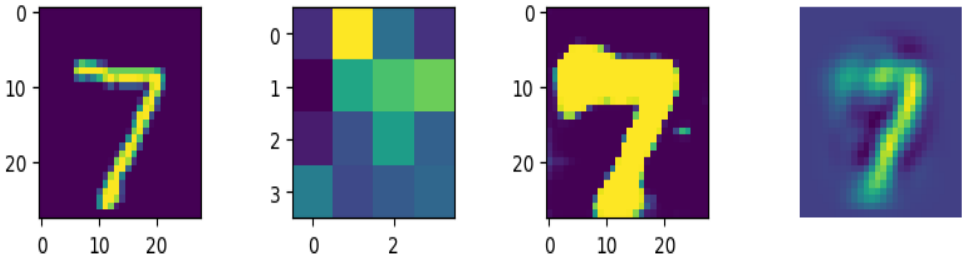
Prva od njih predstavlja originalnu sliku iz test skupa.

Druga predstavlja sliku koja se dobija nakon primene PCA tehnike koja smanjuje sliku veličine 28x28 piksela na sliku veličine svega 4x4 piksela.

Treća slika predstavlja izlaz iz dekodera, koji je na ulazu dobio prethodno predstavljenu sliku od 4x4 piksela i koji je potom pokušao da je rekonstruiše i da predstavi sliku koja se našla na ulazu u autoenkoder a koja je iz originalnog trening/test skupa.

Četvrta slika nije vezana za autoenkoder, već samo daje predstavu o tome kako bi slika izgledala ako pokuša da se ponovo dobije iz izlaza PCA modela, primenom inverzne funkcije na sliku 4x4 piksela.

Model se trenira kroz 10 epoha i to zahteva malo duže vreme prevođenja i pokretanja programa.
Program sam testirala u razvojnom okruženju Visual Studio Code, upotrebom Python3 kompajlera i na operativnom sistemu Linux.
Projekat zahteva instalirane biblioteke sklearn i tensorflow.



Slika 3. Rezultati koje sam dobila

Zaključak

Tehnike redukcije dimenzionalnosti često mogu biti od velike pomoći, kako i u vremenskom, tako i u memorijskom smislu.
Konkretno, jedna od upotreba autoenkodera može biti otklanjanje šuma sa slike. Postupak bi bio vrlo sličan ovom koji sam ja opisala i prezentovao bi moć ovih korisnih tehnika.
Sa razvojem tehnologije, optimizacija i smanjenje vremenske i memorijske složenosti postaju sve važnije i značajnije, što ovoj oblasti daje još veću bitnost.

Literatura

[1] chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://www.pca.narod.ru/DimensionReductionBrifReview.pdf
[2] https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726
[3] https://builtin.com/data-science/step-step-explanation-principal-component-analysis