UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

Denis Aličić

REŠAVANJE PROBLEMA KLASTEROVANJA KORIŠĆENJEM AUTOENKODERA I METODA GLOBALNE OPTIMIZACIJE

master rad

Beograd, 2021.

Mentor:

dr Miroslav Marić, redovni profesor
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

dr Mladen Nikolić, docent
Univerzitet u Beogradu, Matematički

dr Nina Radojičić Matić, docent
Univerzitet u Beogradu, Matematički fakultet

Datum odbrane: 30.09.2021.

Naslov master rada: Rešavanje problema klasterovanja korišćenjem autoenkodera i metoda globalne optimizacije

Rezime: Klasterovanje predstavlja proces grupisanja podataka u skladu sa njihovom sličnošću. Potreba za klasterovanjem se može javiti u različitim situacijama kao što su: analiza socijalnih mreža, ustanovljavanje zajedničkog porekla, analiza dokumenata i sl. Većina poznatih algoritama klasterovanja (K-sredina, DBSCAN itd.) ne koristi tehnike dubokog učenja koje su doživele veliku popularnost u poslednjih nekoliko godina. Pokazalo se da performanse ovih algoritama značajno opadaju sa povećanjem dimenzionalnosti podataka. Jedan od načina za smanjenje dimenzionalnosti podataka je korišćenje autoenkodera. U ovom radu predstavljen je razvijeni sistem za klasterovanje visokodimenzionalnih podataka i analiza uticaja korišćenja autoenkodera nad takvim podacima. Priroda problema klasterovanja je takva da se često ne može sa sigurnošću znati koliko je klasterovanje dobro. Postoje definisane funkcije koje približno evaluiraju kvalitet klasterovanja u zavisnosti od domena podataka nad kojima se ono vrši. Sistem za klasterovanje podataka, koji je razvijen za potrebe ovog master rada, koristi pogodan autoenkoder za smanjenje dimenzionalnosti podataka, a zatim algoritam klasterovanja zasnovan na inteligenciji rojeva (PSO) koji direktno optimizuje funkciju evaluacije klasterovanja.

Ključne reči: klasterovanje, računarstvo, nenadgledano učenje, računarska inteligencija, optimizacija rojem čestica, autoenkoder, neuronske mreže

Sadržaj

Glava 1

Uvod

Potreba za automatskim uviđanjem zakonitosti u velikim količinama podataka je sve izraženija. Od učenja reprezentacije podataka, klasterovanja do generisanja novih podataka na osnovu postojećih. Sve ovo spada u domen nenadgledanog učenja. Jedan od najistraživanijih problema nenadgledanog učenja je problem klasterovanja.

Klasterovanje se može opisati kao proces grupisanja sličnih podataka. Ne postoji jasna definicija šta su slični podaci. Podaci mogu biti vektori, signali, grafovi itd. U zavisnosti od problema koji se rešava, npr. detekcije anomalija u podacima, automatskog generisanja albuma fotografija, ustanovljavanja zajedničkog porekla i slično, sličnost instanci kao i struktura klastera se definiše na različite načine. Bitni pojmovi vezani za klasterovanje, kao i opis najpoznatijeg algoritma klasterovanja K-sredina, su dati u poglavlju 4.

Za uspešno rešavanje i primenu sistema za klasterovanje podataka najčešće presudnu ulogu ima domensko znanje. Jedan od ciljeva ovog master rada je razvoj sistema za klasterovanje za različite vrste podataka koji će imati mogućnost jednostavnog uključivanja domenskog znanja. To je postignuto tako što algoritam klasterovanja direktno zavisi od funkcije kvaliteta klasterovanja. U delu 4.4 prikazano je nekoliko funkcija kvaliteta koje su korišćene u eksperimentalnim rezultatima prikazanim u delu 7.

Visokodimenzionalni podaci, kakvi najčešće dolaze iz realnog sveta, donose sa sobom niz problema. Neki od njih su računska zahtevnost nad podacima velike dimenzije, prokletstvo dimenzionalnosti koje je detaljnije opisano u 6.1, nemogućnost vizuelizacije itd. Primećeno je da podaci najčešće leže u prostoru znatno manje dimenzije. Tradicionalno, najpoznatija metoda za smanjenje dimenzionalnosti je algoritam analize glavnih komponenti zasnovan na linearnoj algebri. Problem sa

korišćenjem ovog algoritma je u njegovoj nemogućnosti nalaženja nelinearnih potprostora.

U sistemu razvijenom za potrebe ovog master rada, korišćeni su autoenkoderi za smanjivanje dimenzionalnosti. Opis sistema i način korišćenja autoenkodera je prikazan u poglavlju 6. S obzirom da su autoenkoderi neuronske mreže koji imaju sposobnost učenja nelinearnih veza oni imaju sposobnost nalaženja nelinearnih prostora. Takođe pored smanjivanja dimenzionalnosti, velika prednost autoenkodera je što se u zavisnosti od domena podataka može izabrati odgovarajuća arhitektura autoenkodera. Detaljniji opis autoenkodera i izbora arhitekture je dat u poglavlju 2.4. Opis i definicije bitnih pojmova vezanih za neuronske mreže su date u poglavlju 2. Kao osnova za definisanje relevatnih pojmova je korišćena knjiga [18].

U poglavlju 3 opisan je metaheuristički algoritam optimizacije rojem čestica. Na tom algoritmu je zasnovan razvijeni algoritam klasterovanja. U delu 8 dat je zaključak na osnovu rezultata, mogućnosti daljeg razvoja i pravci istraživanja.

Glava 2

Neuronske mreže

Neuronske mreže predstavljaju jedan od najpopularnijih metoda mašinskog učenja. Doživele su veliku popularnost u poslednjoj deceniji. Ipak, ideja o njihovom konceptu nastala je kada i moderni računari, tokom i neposredno nakon Drugog svetskog rata. Razlog zašto su tek skoro stekle veliku popularnost i primenjivost leži u tome što je njihovo obučavanje zahtevan računski proces. Razvoj grafičkih kartica sa velikom memorijom i brojem jedinica je omogućio njihovu upotrebu u realnom vremenu.

Neronska mreža se može predstaviti kao graf, čiji čvorovi su neuroni. Svaki neuron poseduje svoje parametre i funkciju aktivacije 2.3. Specifičnim povezivanjem ovakvih čvorova u graf, dobija se neuronska mreža koja omogućava aproksimaciju funkcija. Obučavanje mreže predstavlja izbor parametara koji minimizuju funkciju greške. Funkcija greške govori koliko je aproksimacija loša. Minimizacijom funkcije greške dobija se sve manje loša, tj. sve bolja aproksimacija. Poželjno je da funkcija greške bude diferencijabilna, da bi bilo moguće optimizovati je nekom varijantom gradijentnog spusta [12][18].

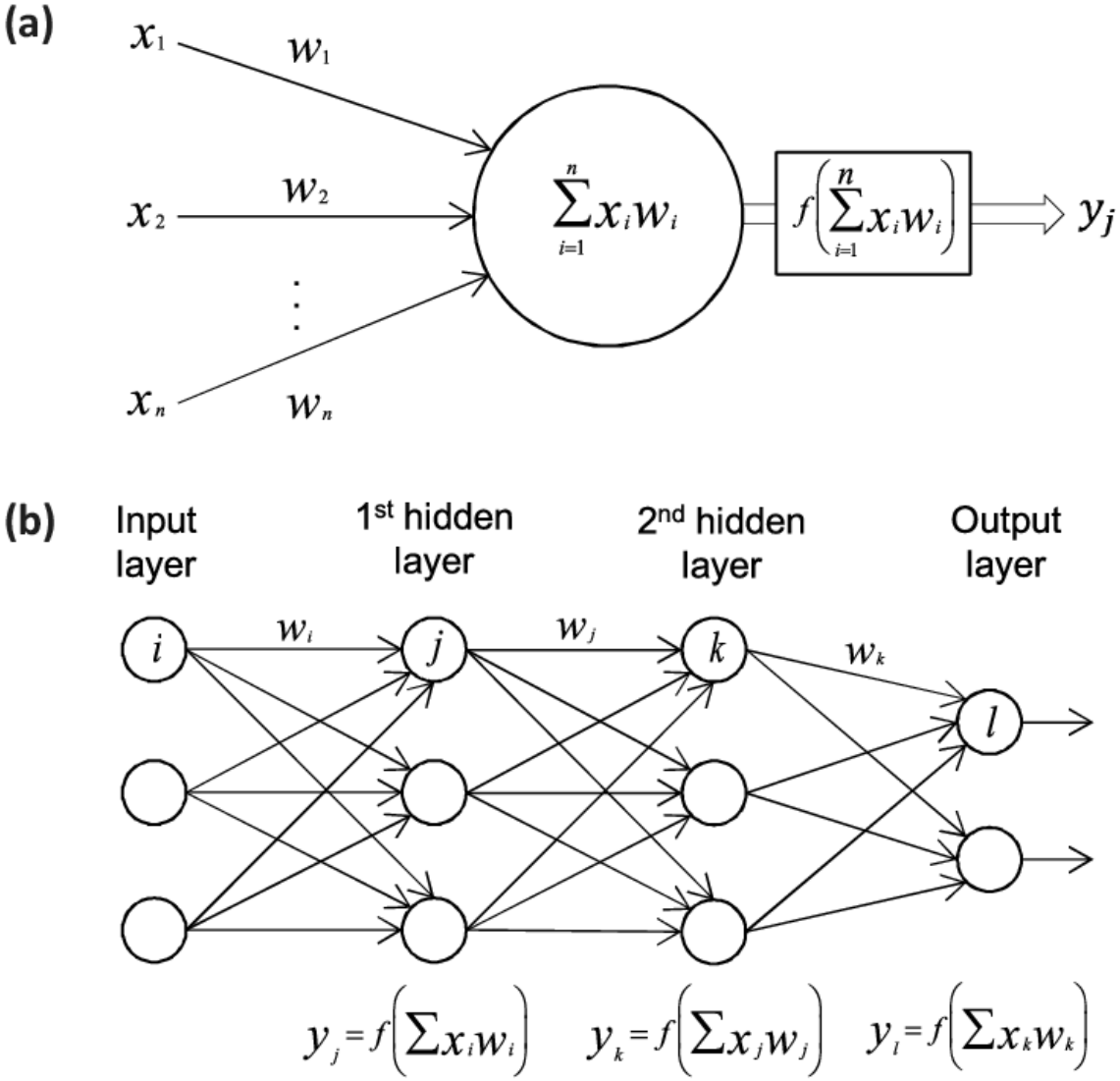
Neke od najpoznatijih arhitektura neuronskih mreža su:

- potpuno povezane (eng. *fully connected*),
- konvolutivne (eng. *convolutional*),
- rekurentne (eng. *recurrent*).

Za potrebe ovog rada bitne su potpuno povezane i konvolutivne, tako da će o njima biti više reči.

2.1 Potpuno povezane neuronske mreže

Potpuno povezane neuronske mreže se grade od više slojeva (eng. *layer*) neurona. Neuroni jednog sloja dobijaju podatke od prethodnog, primenjuju nelinearnu transformaciju nad njima i šalju ih narednom. Prvi sloj mreže se naziva *ulazni*, poslednji se naziva *izlazni*, dok se jedan ili više slojeva između nazivaju *skriveni slojevi*. Vizuelni prikaz jednog neurona, kao i potpuno povezane mreže je dat na slici 2.1[1].

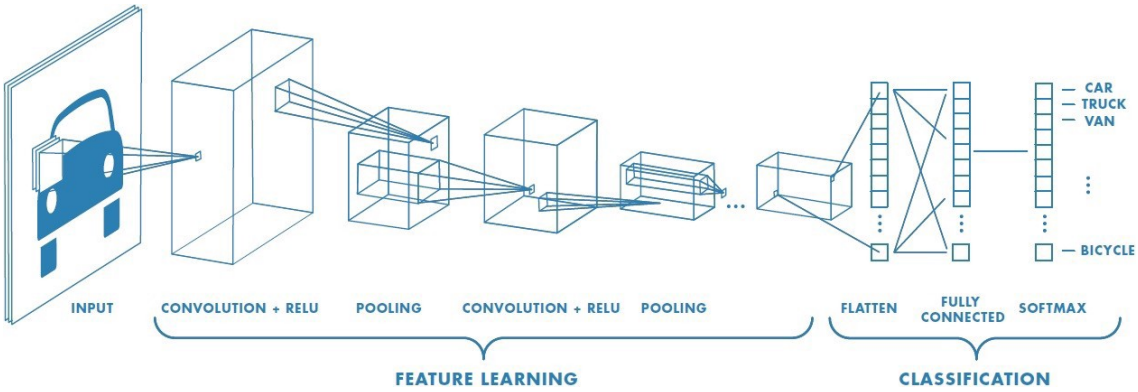


Slika 2.1: Prikaz jednog neurona (a) i potpuno povezane mreže neurona (b).

[1]Slika je preuzeta sa [9].

2.2 Konvolutivne neuronske mreže

Konvolutivne neuronske mreže predstavljaju neuronske mreže čiji neuroni primenjuju operaciju konvolucije nad podacima iz prethodnog sloja, umesto skalarnog proizvoda vektora i parametara u slučaju potpuno povezane neuronske mreže. S obzirom da se operacija konvolucije najčešće definiše nad matričnim podacima, konvolutivne neuronske mreže su pogodne za obradu signala kao što su slika, zvuk, vremenske serije i slično. Pored funkcije konvolucije, konvolutivne mreže sadrže i funkciju agregacije. Vizuelni prikaz arhitekture konvolutivne neuronske mreže je dat na slici 2.2[2].



Slika 2.2: Prikaz arhitekture konvolutivne neuronske mreže.

Konvolucija

Konvolucija je operacija šira od matrične interpetacije, ali pošto se u konvolutivnim mrežama najčešće koristi i interpretira kao operacija nad matricama, u nastavku će biti data njena definicija u kontekstu matrica.

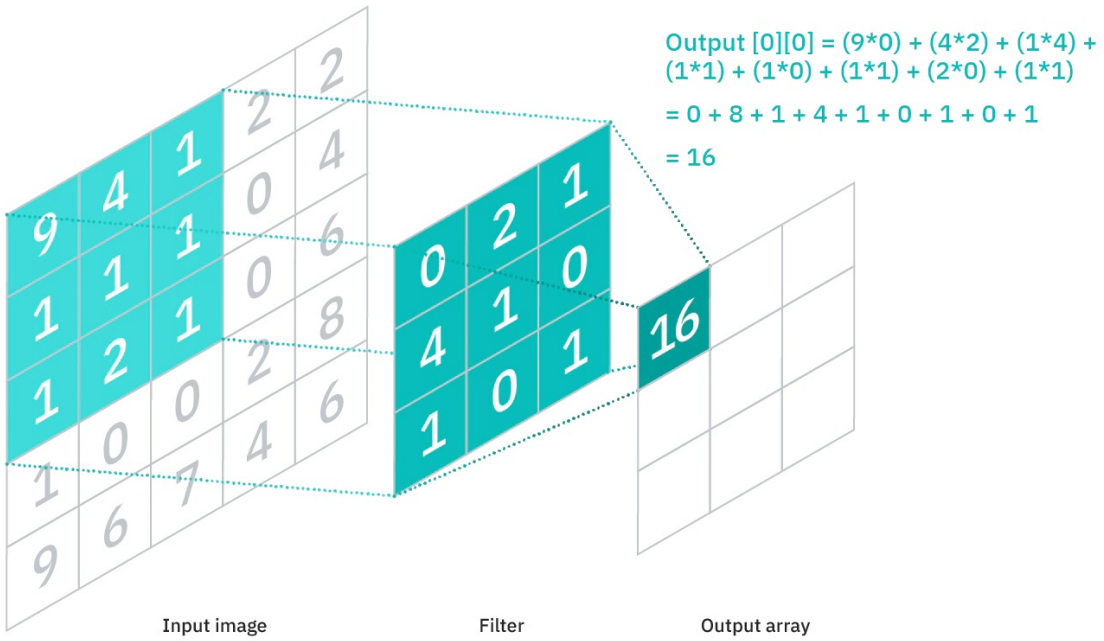
Neka je A matrica dimenzija $m \times n$ i B matrica dimenzija $p \times q$. Operacija konvolucije u oznaci $*$ nad matricama A i B matrice se definiše kao:

$$(A * B)_{ij} = \sum_{k=0}^{p-1} \sum_{l=0}^{q-1} A_{i-k,i-l} B_{k,l}. \tag{2.1}$$

U kontekstu konvolutivnih neuronskih mreža, matrica A predstavlja ulaz u neuron, dok matrica B predstavlja parametrizovani filter ili kernel koji se za svaki

[2]Slika je preuzeta sa [3].

neuron određuje tokom procesa obučavanja neuronske mreže. Na slici 2.3[3] prikazan je efekat konvolucije nad dve matrice.



Slika 2.3: Prikaz operacije konvolucije nad dve matrice.

Agregacija

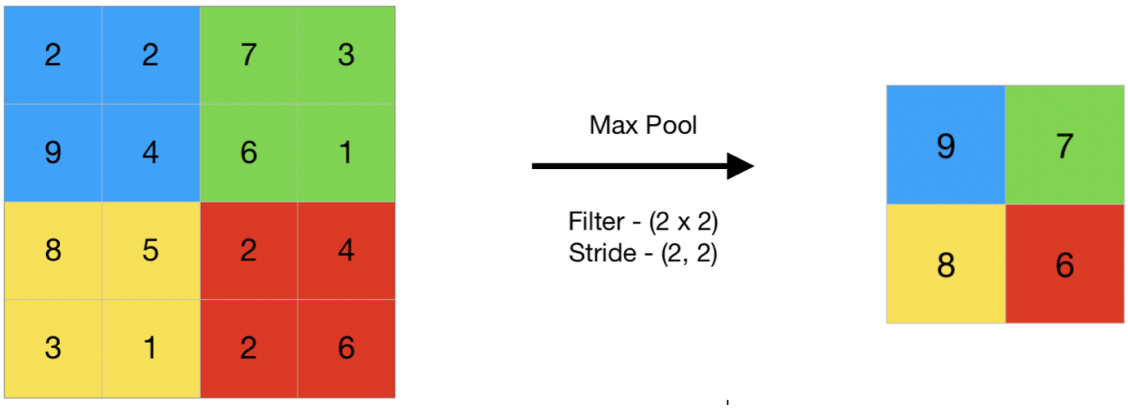
Agregacija (eng. *pooling*) je operacija nad matricom koja se koristi da smanji veličinu matrica bez značajnog gubitka informacija. Podmatrica najčešće veličine 2×2 ili 3×3 se menja jednim brojem. To je najčešće maksimalni element podmatrice ili prosečni element. Na slici 2.4[4] dat je vizuelni prikaz operacije agregacije.

2.3 Funkcije aktivacije

Funkcije aktivacije služe da neuronskoj mreži daju sposobnost nelinearne aproksimacije. Primenjuju se na izlaz svakog nerona. Najkorišćenije funkcije aktivacije su:

- sigmoidna funkcija σ,

[3]Slika je preuzeta sa [4].
[4]Slika je preuzeta sa [2].

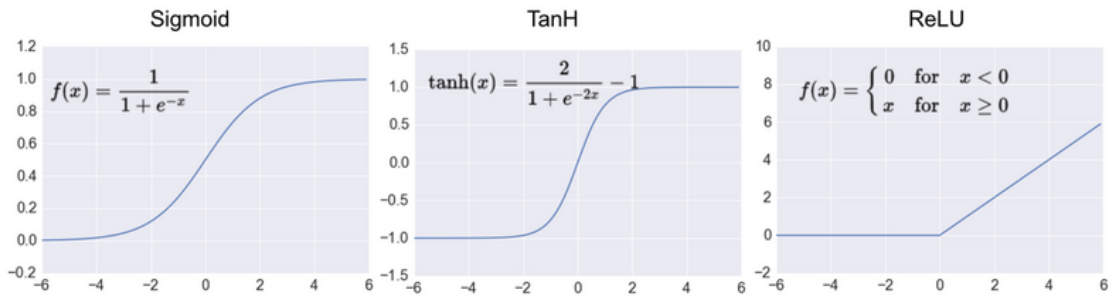


Slika 2.4: Operacija agregacije maksimalnim elementom.

- tangens hiperbolički *tanh*,
- ispravljena linearna jedinica (eng. *Rectified linear unit - Relu*).

Date su jednačinama 2.2 i graficima 2.5[5].

$$\begin{aligned} \sigma(x) &= \frac{1}{1+e^{-x}} \\ tanh(x) &= \frac{e^{2x}-1}{e^{2x}+1} \\ relu(x) &= max(0,x) \end{aligned} \tag{2.2}$$



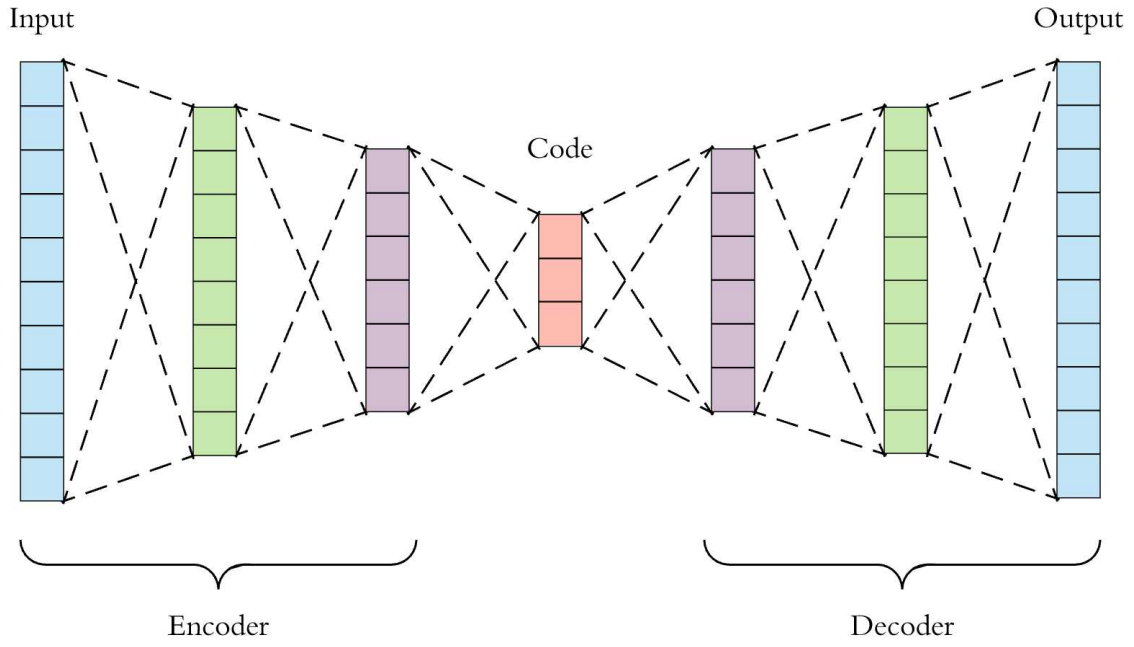
Slika 2.5: Grafici funkcija aktivacije.

[5]Slika je preuzeta sa [1].

2.4 Autoenkoderi

Autoenkoder je neuronska mreža čiji cilj je da nauči funkciju $f(x) = x$. Njegova osnovna arhitektura je prikazana na slici 2.4[6]. Da bi autoenkoder naučio funkciju $f(x) = x$, funkcija greške koja se minimizuje je:

$$E(w, D) = \sum_{i=1}^{N} \|(x - f(x))\|_2^2 \tag{2.3}$$



Slika 2.6: Osnovna arhitektura autoenkodera.

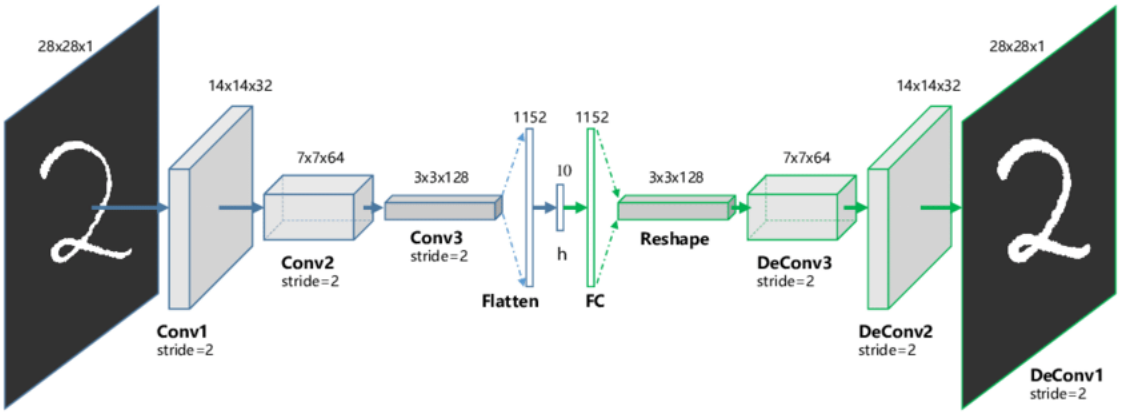
Ova arhitektura neuronske mreže je prvi put predstavljena 1980-ih od strane Hintona i PDP grupe [14]. Iz definicije funkcije greške je jasno da mreža pokušava da rekonstruiše ulaze na svojim izlazima sa što manjim gubitkom informacije. Konceptualno dosta jednostavno i na prvi pogled ne deluje previše korisno. Suština autoenkodera se ogleda u njegovom srednjem sloju, koji je najčešće mnogo manje dimenzije od ulaza. Mreža pored toga što uči da rekonstruiše ulaze, uči i kompaktnu reprezentaciju ulaza, koju predstavlja taj srednji sloj. Reprezentacija u prostoru manje dimenzije se najčešće naziva latentni prostor (eng. *latent space*). Iako autoenkoder predstavlja jednu neuronsku mrežu, on se može posmatrati iz dva dela.

[6]Slika je preuzeta sa [25].

Prvi deo se naziva enkoder (eng. *encoder*). To je deo koji preslikava ulaz u latentni prostor. Dok se drugi najčešće zove dekoder (eng. *decoder*), tj. deo koji preslikava latentni prostor u polazni.

U praksi se pokazalo da se bliski podaci ulaznog prostora često slikaju u bliske podatke u latentnom prostoru. Ova osobina autoenkodera omogućava efikasno smanjenje dimenzionalnosti podataka, bilo u svrhu klasterovanja, klasifikovanja ili nečega sličnog gde je potrebno zadržati međusobnu bliskost podataka.

S obzirom da neuronska mreža ima sposobnost učenja nelinearnih veza, autoenkoder se pokazao kao dosta moćniji od nekih tradicionalnih algoritama smanjenja dimenzionalnosti (PCA, SVD itd.). Autoenkoder ima još jednu značajnu prednost u odnosu na pomenute algoritme. U zavisnosti od tipa i domena podataka, njegova arhitektura može da varira. Za podatke koji su prirodno predstavljeni vektorima, kao slojevi neuronske mreže se koriste potpuno povezani slojevi. Dok za podatke koji su prirodno predstavljeni matricama (npr. slike) moguće je koristiti arhitekturu konvolutivne neuronske mreže. Arhitektura jednog konvolutivnog autoenkodera prikazana je na slici 2.4. Mogućnosti konvolutivne mreže u obradi slika su neuporedivo veće u odnosu na druge metode.



Slika 2.7: Konvolutivni autoenkoder.

Glava 3

Optimizacija rojem čestica

Optimizacija rojem čestica (eng. *Particle Swarm Optimization - PSO*) [22] je metaheuristiki algoritam zasnovan na ponašanju pojedinačnih jedinki unutar određene grupe (na primer, jata ptica ili roja insekata). Ukoliko se, vođeno instiktom, jato ptica uputi u određenom smeru u potrazi za hranom, očekivanje je da će čitavo jato slediti upravo onu pticu koja je pronašla izvor hrane. Međutim, i svaka ptica ponaosob može biti vođena sopstvenim instiktom i time na trenutak u potrazi za hranom napustiti jato. Tada se verovatno može desiti da, ukoliko pronađe bolji izvor hrane, čitavo jato upravo krene da sledi tu pticu.

PSO pripada skupu algoritama koji se zasnivaju na inteligenciji roja (eng. *swarm intelligence*). Algoritam radi nad skupom jedinki, koji se naziva rojem. Elementi ovog skupa se nazivaju česticama. Čestice se na unapred definisan način kreću po prostoru pretraživanja koje odgovara skupu potencijalnih rešenja. Njihovo kretanje se usmerava imajući u vidu njihovu trenutnu poziciju, njihovu do sada najbolju poziciju, kao i do sada najbolju poziciju čitavog roja. Pod najboljom pozicijom čitavog roja se podrazumeva do sada najbolja pozicija, uzimajući u obzir sva njegova rešenja. Proces se ponavlja dok ne bude zadovoljen kriterijum zaustavljanja, a u svakoj iteraciji se ažurira najbolja vrednost rešenja za svaku česticu, kao i za roj u celini.

Pseudokod osnovnog PSO algoritma je prikazan na 1. Neka je sa X označen roj i neka su svakoj čestici iz skupa X dodeljeni vektori $x_i \in R^n$ i $v_i \in R^n$, $i \in X$, koji predstavljaju njene vektore pozicije i brzine. Dodatno, n-dimenzionim vektorom p_i označena je trenutna najbolja pozicija čestice $i \in X$, a n-dimenzionim vektorom g trenutna globalna najbolja pozicija. Pri inicijalizaciji čestice i, vrednosti koordinata vektora x_i se biraju uniformno iz skupa (l, u), a vrednosti koordinata vektora v_i

Algoritam 1: PSO algoritam

Ulaz: f - funkcija cilja, X - skup rešenja
Izlaz: vektor rešenja, vrednost rešenja
za svaki *rešenja (čestice) i iz skupa rešenja (roja) X;*
čini
  Izabrati koordinate vektora x_i uniformno iz intervala (l, u);
  $p_i = x_i$;
  ako $f(x_i) < f(g)$ **onda**
    $g = x_i$;
  kraj
  Označiti koordinate vektora v_i da su jednaki nuli ili uniformno iz intervala $(-(u-l), (u-l))$;
kraj
dok *nije postignut kriterijum zaustavlja* **čini**
  za svaki *čestici i iz roja X* **čini**
    Izabrati brojeve r_g i r_p uniformno iz intervala $(0, 1)$;
    $v_i = c_v v_v + c_p r_p(p_i - x_i) + c_g r_g(g - x_i)$;
    $x_i = x_i + v_i$;
    ako $f(x_i) < f(p_i)$ **onda**
      $p_i = x_i$;
    kraj
    ako $f(x_i) < f(g)$ **onda**
      $g = x_i$;
    kraj
  kraj
kraj
Rešenje je vektor g, a vrednost rešenja broj $f(g)$

postavljaju na nule ili biraju uniformno iz skupa $(-(u-l), (u-l))$. Ovde je sa l i u označeno donje i gornje ograničenje pretraživačkog prostora. Pri inicijalizaciji algoritma se dodeljuju početne vrednosti vektorima p_i i g.
U svakoj iteraciji se, za svaku česticu i, vrši ažuriranje vrednosti vektora brzine sa:

$$v_i = c_v v_i + c_p r_p(p_i - x_i) + c_g r_g(g - x_i), \tag{3.1}$$

a zatim i vektora trenutne pozicije sa $x_i = x_i + v_i$. Parametri r_g i r_p se u svakoj iteraciji biraju uniformno iz intervala $(0, 1)$, dok su c_v, c_p i c_g unapred definisani parametri koji se mogu eventualno i menjati tokom algoritma. U svakoj iteraciji se ažurira i vrednost vektora p_i i g. Konačno, odgovarajuće rešenje je sadržano u vektoru g. Opisani proces se ponavlja sve dok nije ispunjen kriterijum zaustavljanja.

Važan deo svakog algoritma optimizacije je njegova sposobnost pretrage širokog prostora rešenja u odnosu na postizanje lokalnog optimuma u nekoj manjoj oblasti. Kod PSO algoritma ovaj problem je rešen korišćenjem dva parametra koji se odnose na kognitivnu i sociološku komponentu (c_p i c_q). Kognitivna komponenta c_p čestice daje značajnost njenoj najboljoj poziciji, dok se sociološka komponenta c_g odnosi na najbolju poziciju celog roja. Menjanjem vrednosti ove dve komponente balansira se između nalaženja globalnog i lokalnog optimuma.

Iako je klasičan PSO namenjen pre svega problemima kontinualne prirode, postoji više načina da se PSO pristup prilagodi diskretnim problemima. Prvi je preslikavanje diskretnog pretraživačkog prostora u kontinualni, primene algoritma na opisan način, a zatim preslikavanje nazad u diskretan prostor. Drugi način predstavlja drugačije predstavljanje vektora brzine i pozicije, kao i modifikovanje operatora. U osnovnoj verziji algoritma vektor pozicije je n-torka realnih brojeva, a korišćeni operatori su standardni aritmetički. Međutim, koordinate vektora pozicije mogu biti i binarni brojevi, a operatori mogu biti definisani kao npr. operatori nad skupovima, itd.

Glava 4

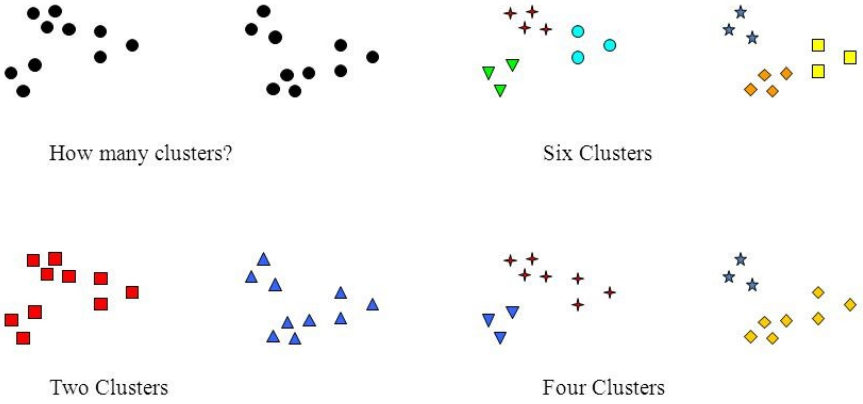
Klasterovanje

Klasterovanje predstavlja grupisanje objekata na osnovu njihove sličnosti. Cilj je da objekti u jednom klasteru budu slični i različiti od objekata u drugom klasteru. Što su objekti u jednom klasteru sličniji i što su objekti iz različitih klastera različitiji, to je klasterovanje bolje. U mnogim realnim primenama nije moguće imati savršeno razdvojene klastere. Nekada je dosta teško odlučiti šta su to dobro razdvojeni klasteri i šta čini jedan klaster. Ovaj problem je ilustrovan na slici 4. Slika pokazuje dvadeset tačaka i tri različita načina da se ove tačke podele u klastere. Ukoliko bi dozvolili ugnježdene klastere, onda bi najverovatnija podela ovih tačaka bila da su to dva klastera, koja u sebi sadrže po tri podklastera. Razumno bi bilo podeliti i u 4 klastera. Ovaj primer dokazuje da nije lako odrediti šta je dobro klasterovanje, i da to zavisi od vrste podataka i željene primene.

4.1 Vrste klastera

U zavisnosti od vrste podataka i željenih rezultata klasterovanja, izdvojilo se nekoliko definicija klastera. Potrebno je spomenuti da pored pojma sličnosti i razdvojenosti klastera, u definicijama navedenim ispod se za neke od njih smatra da rezultat klasterovanja može biti klaster koji u sebi sadrži podklastere. Ti podklasteri su najčešće bez preklapanja sa drugim klasterima.

- Dobro razdvojeni klasteri: Klaster je takav skup tačaka u kom je svaka tačka klastera bliža svakoj tački unutar tog klastera nego bilo kojoj tački koja nije unutar tog klastera.



Slika 4.1: Različit broj klastera za iste tačke.

- Klasteri zasnovani na centroidama: Klaster je skup tačaka u kom je svaka tačka klastera bliža centru klastera kom pripada nego centru bilo kog drugog klastera.
- Klasteri zasnovani na najbližem susedu: Klaster je skup tačaka u kom je svaka tačka klastera bliža jednoj ili više tačaka unutar tog klastera nego bilo kojoj tački koja nije unutar tog klastera.
- Klasteri zasnovani na gustini: Klaster je gust region tačaka koji je razdvojen regionom manje gustine od drugih klastera.

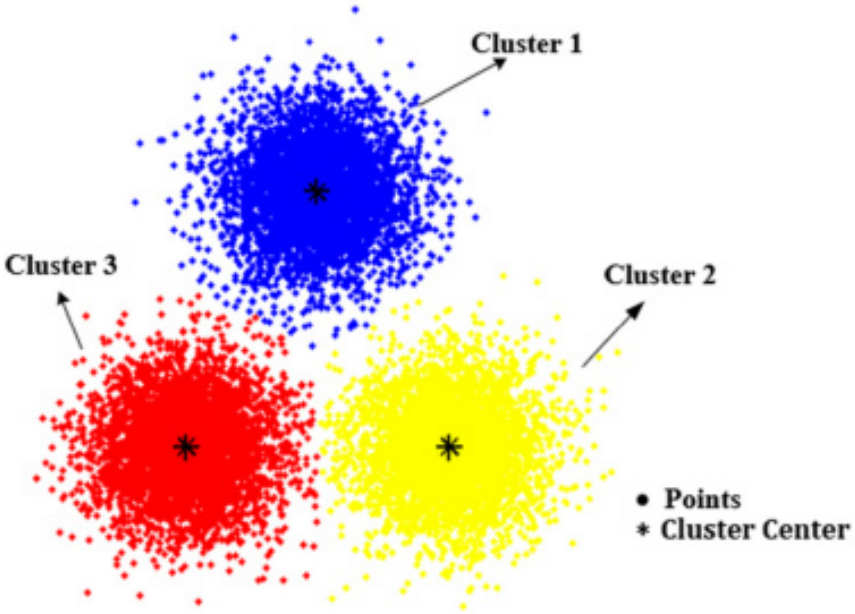
4.2 K-sredina

Algoritam K-sredina (eng. *k-means*) pronalazi k klastera tako što pronalazi k centroida, zatim svakoj tački dodeljuje klaster najbliže centroide. Centroida klastera se najčešće izračunava kao sredina ili medijana tačaka jednog klastera. Za funkciju blizine se najčešće uzima euklidska distanca, mada mogu i neke druge (kosinusna, Menhetn itd.). Početnih k centroida se bira nasumično. Zatim se u svakoj iteraciji ponavljaju sledeći koraci:

1. Dodeliti klaster svakoj tački iz skupa podataka na osnovu funkcije blizine.

2. Izračunati nove centroide kao prosek ili medijanu na osnovu novodobijenog klasterovanja.

Kada u dve uzastopne iteracije se centroide ne promene proces se zaustavlja. Primer klastera dobijenih algoritmom K-sredina je prikazan na slici 4.2[1]. Ovaj algoritam je osetljiv na prisustvo odudarajućih podataka jer se centroid računa kao prosek svih tačaka unutar tog klastera.



Slika 4.2: Klasteri dobijeni algoritmom K-sredina.

4.3 Klasterovanje kao optimizacioni problem

Kod problema klasifikacije, koji je vid nadgledanog učenja, postoje dobro definisane funkcije koje nam mogu reći kakav je kvalitet dobijenog modela. To su pre svega tačnost (eng. *accuracy*) i preciznost (eng. *precision*), mada postoje još neke poput F mere i odziva (eng. *recall*) [21].

Klasterovanje je problem nenadgledanog učenja, tako da za konkretno grupisanje ne postoje jednoznačne funkcije koje nam sa sigurnošću mogu reći koliko je ono

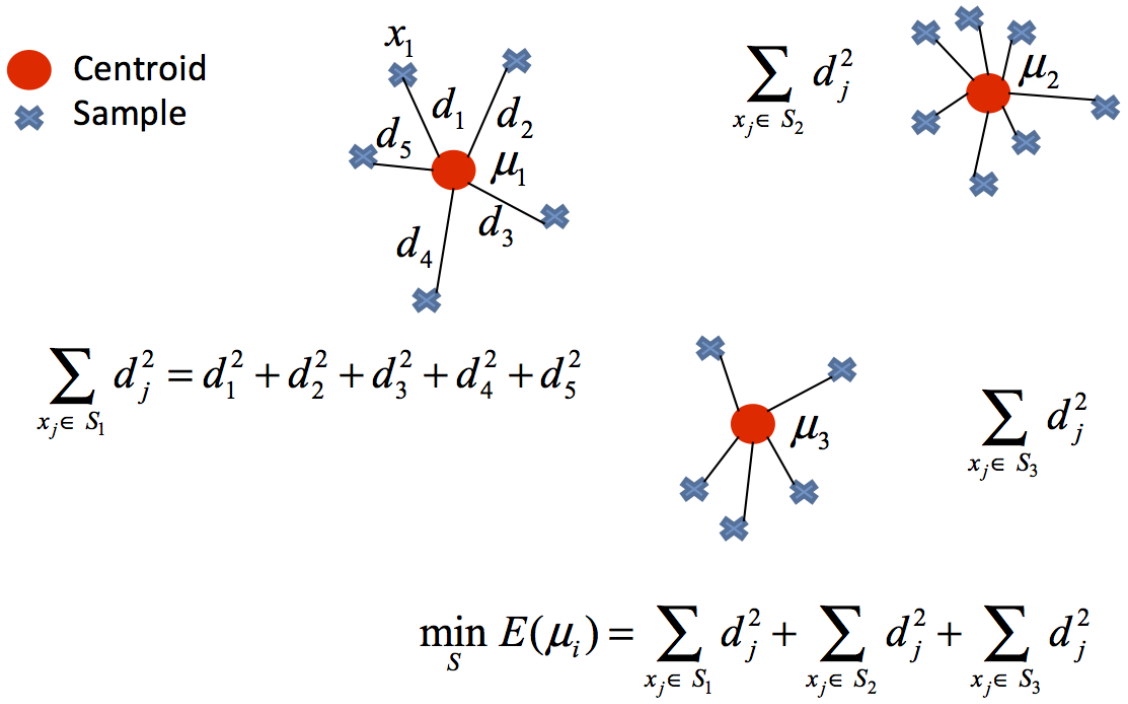
[1]Slika je preuzeta sa [27].

dobro. Ipak, postoje neke funkcije, definisane tokom vremena od raznih istraživača, koje nam mogu dati ocenu kvaliteta klasterovanja [13][17][23].

Zanimljivo je primetiti, da iako na prvi pogled ne deluje tako, algoritam K-sredina se takođe može posmatrati kao optimizacioni algoritam. Funkcija koju taj algoritam optimizuje je:

$$\sum_{i=1}^{k} \sum_{x \in C_i} d(x, c_i)^2, \tag{4.1}$$

gde je d rastojanje, najčešće euklidsko, ali može biti i neko drugo, a c_i centroida i-tog klastera. Vizuelna reprezentacija se može videti na slici 4.3[2].



Slika 4.3: Vizuelizacija funkcije koju optimizuje algoritam K-sredina.

4.4 Funkcije evaluacije klasterovanja

Slično metrikama koje se koriste za evaluaciju klasifikacionih modela, postoje funkcije koje koriste informaciju o stvarnim klasama da bi ocenile kvalitet klasterovanja. Neke od njih su:

[2]Slika je preuzeta sa [5].

- Rand indeks,
- Homogenost,
- V-mera.

U realnim primenama stvarne klase nisu dostupne, tako da ove funkcije nisu korišćene niti prilikom implementacije algoritma, niti prilikom evaluacije, te ni u radu neće biti dalje razmatrane.

Funkcije koje su korišćene prilikom implementacija i eksperimenata su Davies-Bouldin indeks i Calinski-Harabasz indeks. Za izračunavanje ovih funkcija potrebna je informacija o centroidima klastera i dodeli klastera svakoj instanci iz skupa podataka.

Davies-Bouldin indeks

Davies-Bouldin predložen u [13] definisan je kao:

$$DB = \frac{1}{c}\sum_{i=1}^{c}\max_{i\neq j}\left(\frac{\sigma_i + \sigma_j}{d(C_i, C_j)}\right), \tag{4.2}$$

gde je c broj klastera, σ prosečno rastojanje svih instanci unutar jednog klastera od njegovog centroida, d rastojanje (najčešće euklidsko) i C_i centroid i-tog klastera.

Minimizacijom ove funkcije po C_i dobijamo grupe koje imaju malo rastojanje unutar istog klastera, a veliko rastojanje između različitih klastera [13]. Generalno, to je ideja većine ovih funkcija zasnovanih na centroidama, s tim što na različite načine kvantifikuju rastojanja unutar klastera i između različitih klastera.

Minimalna vrednost ove funkcije je 0. Treba biti obazriv sa minimizacijom ove funkcije, jer metode sa mogućnošću povećavanja broja klastera, prilikom optimizacije ove mere teže tome da svaka tačka bude pojedinačni klaster.
U situaciji da postoji beskonačno instanci, samim tim i beskonačno klastera jer je svaka instanca poseban klaster, deo funkcije $\frac{1}{c}$ teži ka nuli, a onda i cela funkcija teži minimalnoj vrednosti.

Calinski-Harabasz indeks

Calinski-Harabasz indeks [17] je nešto složeniji za izračunavanje i interpretaciju. Za dati skup podataka E, veličine n_E, koji se klasteruje u k klastera, vrednost

indeksa je definisana formulom:

$$s = \frac{tr(B_k)}{tr(W_k)} \times \frac{n_E - k}{k - 1}, \tag{4.3}$$

gde je $tr(B_k)$ trag matrice disperzije centroida klastera i $tr(W_k)$ trag matrice disperzije unutar pojedinačnog klastera.

Matrice se izračunavaju po formulama:

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T, \tag{4.4}$$

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T, \tag{4.5}$$

gde je C_q skup instanci u klasteru q, c_q centroid klastera q, c_E centroida celog skupa instanci E i n_q broj instanci u klasteru q.

Što je ovaj indeks veći, to je klasterovanje bolje, centroidi su udaljeniji, a rastojanja unutar klastera manja.

Zbog načina računanja disperzije klastera, koje se računa kao kvadriran zbir udaljenosti instance od centroida, ovaj indeks je generalno veći za konveksne klastere, što je za algoritam koji je predstavljen u ovom radu prednost jer očekivani izlaz algoritma jesu konveksni klasteri.

Glava 5

Algoritam klasterovanja zasnovan na PSO

Algoritam implementiran za potrebe ovog rada je zasnovan na optimizaciji rojem čestica. Pseudokod algoritma je prikazan na slici 2. Osnovu PSO algoritma čini jedna čestica roja. Čestica predstavlja jedno rešenje optimizacionog problema. Jednu česticu u algoritmu razvijenom za potrebe ovog rada predstavlja niz centroida klastera. Centroidi klastera zajedno sa funkcijom blizine predstavljaju jedinstveno određeno rešenje problema klasterovanja. Za svaku instancu skupa nad kojim se vrši klasterovanje se određuje pripadajući klaster kao najbliži centroid koristeći funkciju udaljenosti koja je prosleđena algoritmu kao parametar. Od prosleđene funkcije udaljenosti (euklidska distanca, kosinusna ili neka druga), zavisi oblik klastera. Algoritam je generički u smislu izbora funkcije udaljenosti i funkcije evaluacije koja se optimizuje. Bitna karakteristika algoritma za rešavanje problema klasterovanja je njegova otpornost na prisustvo odudarajućih podataka (eng. *outliers*). Otpornost predloženog algoritma najvećim delom zavisi od izbora funkcije evaluacije, tj. uticaja takvih podataka na njenu vrednost.

Algoritam 2: PSO algoritam klasterovanja

Rezultat: Broj klastera k, k centroida i dodeljivanje klastera svakoj instanci
Ulaz: Skup podataka, maksimalan broj klastera k, funkcija evaluacije, funkcija udaljenosti
Izlaz: Skup podataka sa pripadajucim klasterima, maksimum k centroida
inicijalizuj početni roj;
odredi najbolju česticu;
dok *nije postignut kriterijum zaustavlja* **čini**
  za svaki *česticu u roju* **čini**
    Ažuriraj brzinu čestice na osnovu pozicije najbolje čestice i najbolje pozicije trenutne čestice;
    Promeni poziciju čestice na osnovu izračunate brzine;
    Na osnovu funkcije udaljenosti odredi pripadnost svakoj instanci skupa odgovarajućem klasteru;
    Izračunaj funkciju evaluacije na osnovu dodeljenog klasterovanja;
    ako *nova pozicija bolja od prethodne najbolje pozicije čestice;*
     onda
      ažuriraj najbolju vrednost trenutne čestice;
    kraj
    ako *nova pozicija bolja od pozicije najbolje čestice;*
     onda
      ažuriraj poziciju najbolje čestice;
    kraj
  kraj
kraj
vrati *klasterovanje najbolje čestice, broj klastera i njene centroide*

Glava 6

Sistem za klasterovanje visokodimenzionalnih podataka

6.1 Prokletstvo dimenzionalnosti

Termin je prvi upotrebio Ričard Belman u knjizi o teoriji upravljanja [11]. Na 97. strani Belman kaže: „S obzirom na sve ono što smo rekli u prethodnim odeljcima, mnoge prepreke izgleda da smo savladali, ali šta je presudno za našu proslavu pobede? To je prokletstvo dimenzionalnosti, prokletstvo koje muči naučnike od najranijih dana." Pitanje o kom Belman diskutuje je nemogućnost optimizovanja funkcija sa mnogo promenljivih algoritmom grube sile.

U slučaju klasterovanja, najrelevantniji aspekt „proklestva dimezionalnosti" je efekat porasta raspršenosti tačaka povećavanjem dimenzionalnosti. Da bi se shvatilo kako povećanje dimenzionalnosti utiče na raspršenost tačaka, posmatraće se 100 tačaka dobijenih slučajnim odabirom iz uniformne raspodele na intervalu $(0, 1)$. Ukoliko se se ceo interval podeli na 10 jednakih podeoka, verovatno će se u svakom podeoku naći nekoliko tačaka. Ukoliko se zadrži broj tačaka, ali se tačke raspoređuju u jedinični kvadrat, što odgovara povećanju dimenzionalnosti na dve dimenzije, i zadrži se jedinica diskretizacije prostora na 0.1, onda postoji 100 dvodimenzionalnih podeka i verovatno će neki podeok ostati bez tačaka u njemu. Za 100 tačaka u tri dimenzije, verovatno će većina od 1000 podeoka ostati prazna, jer je broj ćelija daleko veći od broja tačaka. Konceptualno, podaci su „izgubljeni u prostoru" kako se povećava dimenzija prostora.

Ova pojava predstavlja problem većini algoritama jer njihov rezultat najčešće zavisi od udaljenosti ili gustine tačaka. Jedan od načina borbe sa „prokletstvom di-

menzionalnosti" je projektovanje tačaka u prostor manje dimenzije. Ideja iza ovog poduhvata je da podaci leže u prostoru manje dimenzije i da se smanjenjem dimenzionalnosti gubi relativno mali deo informacija. U nekim slučajevima ova tehnika pomaže i u uklanjanju odudarajućih podataka. Najčešći način smanjenja dimenzionalnosti je primenjivanje algoritma analize glavnih komponenti (eng. *PCA - Principal component analysis*) [20] ili singularne dekompozicije (eng. *SVD - Singular value decomposition*) [26].

6.2 Opis sistema za klasterovanje visokodimenzionalnih podataka

Razvijeni sistem za klasterovanje visokodimenzionalnih podataka uključuje korišćenje pogodnog autoenkodera za smanjivanje dimenzionalnosti podataka, na način opisan u 2.4. Korišćenjem enkodera podaci se preslikavaju u prostor manje dimenzije i izvršava algoritam opisan u 5. Zatim se nad tako izvršenim klasterovanjem korišćenjem dekodera podaci vraćaju u polazni prostor dok se oznaka klastera zadržava. Sistem je namenjen da se koristi za širok domen podataka. U zavisnosti od domena podataka bira se odgovarajući autoenkoder (potpuno povezan, konvolutivni, rekurentni, graf itd.) i funkcija evaluacije koje finije modeluje željeni oblik i strukturu klastera.

Glava 7

Eksperimentalni rezultati

7.1 Rezultati algoritma klasterovanja zasnovanog na PSO

Razvijeni algoritam je testiran na poznatim skupovima podataka: IRIS i WINE. U eksperimentima su upoređeni rezultati algoritma K-sredina sa PSO algoritmom. Eksperimenti su vršeni na računaru sa Intelovim i7 procesorom devete generacije, taktom procesora 2.60GHz i radnom memorijom od 32GB. Vrednosti prikazane u tabelama 7.1 i 7.2 predstavljaju vrednosti funkcija evaluacije klasterovanja, koje su detaljno opisane u 2.1.1. i 2.1.2.

Za optimizaciju obe funkcije evaluacije korišćenje su vrednosti kognitivne i sociološke komponente, 1 i 2 redom. Taj izbor govori da na jednu česticu više utiče ceo roj, tj. najbolja čestica roja, nego najbolja pozicija trenutne čestice. Veličina roja je 20 čestica i broj iteracija je 500, što je bio i kriterijum zaustavljanja.

Skup podataka Iris

Skup podataka Iris [10] je jedan od najpoznatijih skupova podataka. Sastoji se od 4 numerička atributa:

- dužina krunice,
- širina krunice,
- dužina čašice,
- širina čašice.

Primarno je namenjen za testiranje algoritama koji rešavaju problem klasifikacije jer sadrži i peti, kategorički atribut, koji predstavlja vrstu cveta iris. Skup podataka sadrži 3 klase od po 50 instanci. Za potrebe ovog rada iskorišćena su gore navedena četiri atributa bez informacije o pripadajućoj klasi.

Zanimljivo je primetiti da je za Davies-Bouldin index broj klastera 2. U sekciji 5 je naznačeno da se razvijenom algoritmu prosleđuje maksimalan broj klastera i da se taj broj može smanjiti tokom izvršavanja algoritma. Prilikom testiranja algoritma nad ovim skupom podataka za različite parametre kognitivne i sociološke komponente, broja čestica, broja iteracija itd. primećeno je da algoritam kao izlaz da 2 klastera. Analizom skupa podataka je utvrđeno da je jedna klasa linearno razdvojiva od druge dve, što je i navedeno u opisu skupa podataka [7]. Tako da se može opravdano pretpostaviti da je ovo jedno od validnih klasterovanja.

Očekivano je PSO algoritam nadmašio algoritam K-sredina, jer je direktno optimizovao funkcije koje su prikazane u tabeli 1. Izvršavanje algoritma je trajalo 21.53 sekundi za optimizaciju Davies-Bouldin indeksa i 25.08 sekundi za Calinski-Harabasz indeks.

Skup podataka Wine

Skup podataka Wine [8] je takođe jedan od poznatih skupova. Sastoji se od 13 numeričkih atributa i ciljne klase. Atributi predstavljaju vrednosti različitih hemijskih supstanci do kojih se došlo hemijskom analizom 3 vrste vina koja potiču iz Italije. Skup se sastoji od 178 instanci. Kao i u prethodnom skupu podataka, ni u ovom nije korišćena informacija o klasi.

Algoritam je kao parametar prosleđen broj 5 kao maksimalan broj klastera. Prilikom optimizacije obe funkcije, skoro svaki put je algoritam smanjio broj klastera na 3, što itekako ima smisla s obzirom da u skupu podataka zaista postoje 3 klase i za taj broj klastera su funkcije bila minimalne. To takođe znači da su izabrane funkcije pogodne za rešavanje problema klasterovanja nad ovim skupom.

Skup podataka Wine se smatra za jedan od lakših primera problema klasifikacije, ali treba imati u vidu da je ovde reč o klasterovanju i da algoritam prilikom izvršavanju ni na koji način nije imao informaciju od broju klasa.

U tabeli 2 su prikazani rezultati izvršavanja PSO algoritma i algoritma K-sredina. Očekivano, PSO je i u ovom slučaju nadmašio K-sredina. Vreme potrebno za iz-

Algoritam	IRIS			
	DB		CZ	
	c	index	c	index
K-Sredina	2	0.40	3	561.62
PSO	2	**0.28**	3	**601.05**

Tabela 7.1: Vrednosti funkcija evaluacije nad skupom podataka IRIS.

vršavanje algortima je 63.44 i 84.64 sekunde za optimizaciju Calinski-Harabasz i Davies-Bouldin indeksa redom.

Algoritam	WINE			
	DB		CZ	
	c	index	c	index
K-Sredina	3	0.53	3	561.81
PSO	3	**0.43**	3	**585.38**

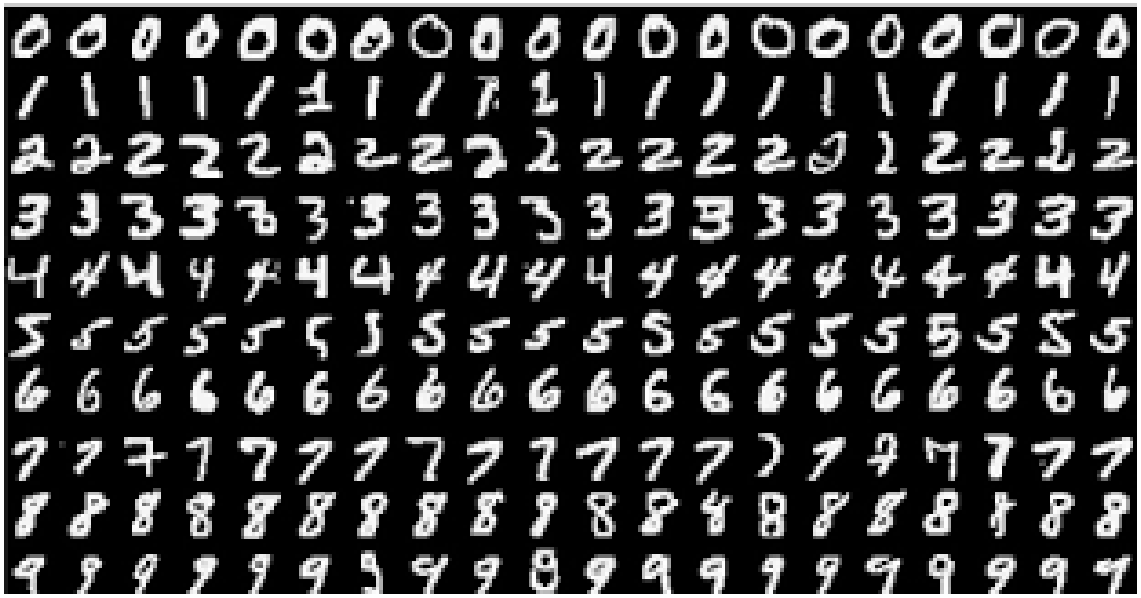
Tabela 7.2: Vrednosti funkcija evaluacije nad skupom podataka WINE.

7.2 Rezultati razvijenog sistema za klasterovanje visokodimenzionalnih podataka

Sistem je testiran na skupu podataka MNIST [15]. Ovaj skup podataka se sastoji od slika ručno pisanih cifara od 0 do 9. Sve slike su crno bele, centrirane i veličine su 28×28 piksela. Primarno je namenjen za razvoj sistema za prepoznavanje i klasifikaciju slika. Skup je originalno podeljen na skup za treniranje, koji se sastoji od 60 000 slika i skup za validaciju koji se sastoji od 10 000 slika.

Često se koristi kao polazni skup za treniranje konvolutivnih mreža namenjenih za klasifikaciju, jer kada se u tu svrhu koristi, ne predstavlja previše težak zadatak. Ipak, ovo je skup podataka koji dolazi iz realnog sveta, pa je samim tim zanimljiv mnogim istraživačima.

Iz tog razloga i zbog toga što svaka instanca ovog skupa predstavlja ulazni podatak velike dimenzije izabran je za skup nad kojim će se testirati razvijeni sistem za potrebe ovog master rada. Kao što je već spomenuto, slike su dimenzija 28×28, tako da se jedna instanca ovog skupa se može posmatrati kao ulazni vektor dimenzije 784. Primer slika iz ovog skupa je prikazan na slici 7.2. U svrhu smanjivanja dimen-



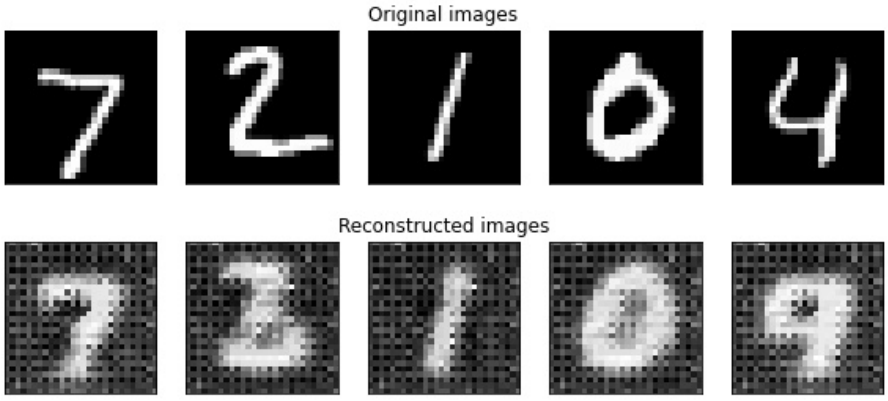
Slika 7.1: Primer instanci skupa MNIST.

zionalnosti, korišćen je konvolutivni autoenkoder s obzirom da je ulazni podatak slika.

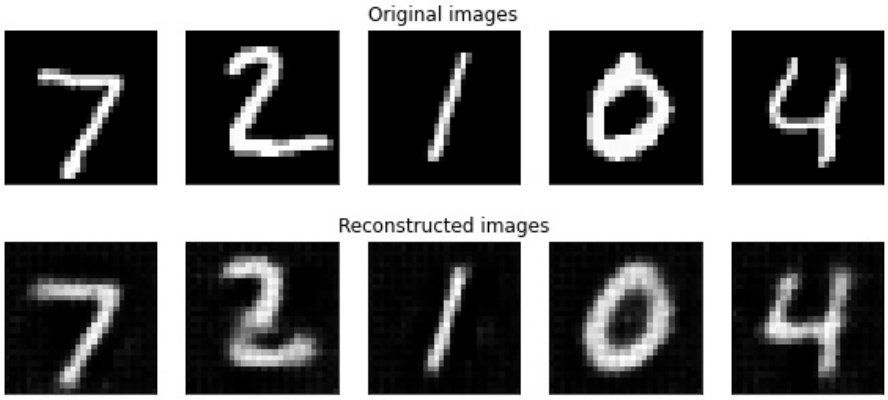
Prvi deo autoenkodera, koji se još naziva i enkoder, se sastoji od 3 konvolutivna sloja koji redom imaju 8,16 i 32 neurona. Svaki od slojeva koristi Relu aktivacionu funkciju. Nakon drugog i trećeg sloja korišćena je unutrašnja standardizacija (eng. *batch normalization*) [19]. Nakon sloja koji matrični oblik transformiše u vektor dužine 288, slede dva sloja sa brojem neurona 128 i 4 koji predstavlja dimenzionalnost latentnog prostora. Drugi deo mreže, često nazivan dekoder, se sastoji od simetričnih slojeva i po broju neurona u svakom i po vrsti sloja. Tako da umesto konvolutivnog sloja, koristi se sloj sa transponovanom konvolucijom (eng. *transposed convolution*) [16].

Kao metod optimizacije korišćen je Adam [24] sa koeficijentom učenja 0.001. Za funkciju greške je iskorišćena srednje-kvadratna greška što je najčešći izbor prilikom treniranja ovakve vrste autoenkodera. Mreža je trenirana u 30 epoha. Na slikama od 7.2 do 7.2 može se videti poboljšanje rekonstrukcije slika tokom obučavanja mreže. Prikazane su slike nakon svakih 5 epoha.

Promena funkcije greške na skupovima za treniranje i validaciju tokom treniranja mreže se može videti na slici 7.2. Greška na skupu za testiranje, koji ni na koji način nije korišćen prilikom obučanja modela iznosi 0.026267, dok je na skupu za trening 0.025210. Ovo je indikator da se mreža nije preprilagodila (eng. *overfit*), dok vizuelni



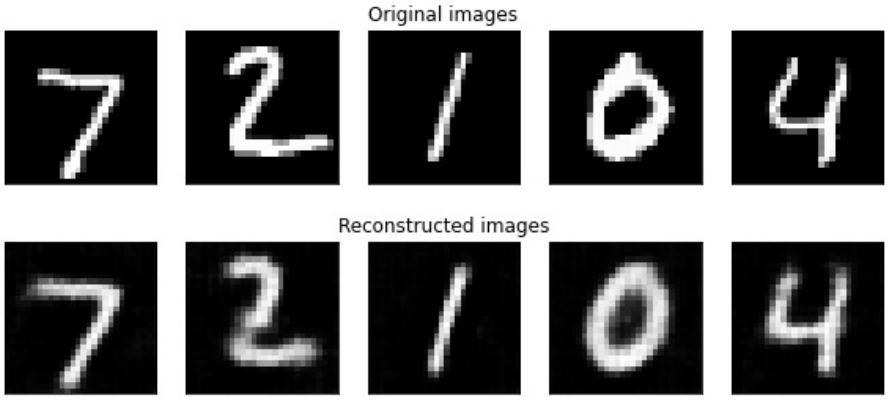
Slika 7.2: Rezultat rekonstrukcije autoenkodera nakon prve epohe.



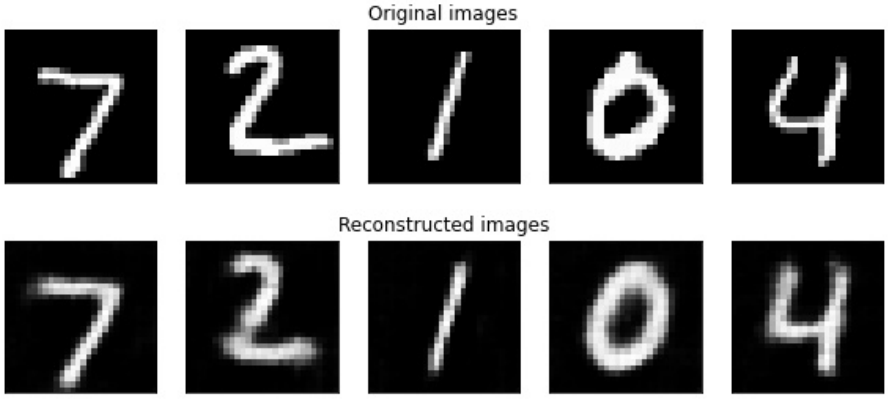
Slika 7.3: Rezultat rekonstrukcije autoenkodera nakon pete epohe.

rezultati govore o tome i da se nije potprilagodila (eng. *underfit*). S obzirom da je latentni prostor dimenzionalnosti 4 u odnosu na ulazni koji je dimenzionalnosti 784, model se izuzetno dobro pokazao i potvrdio tezu da podaci često leže u prostoru značajno manje dimenzije.

Nad vektorima dužine 4 dobijenim kao izlaz enkoder dela autoenkodera, korišćenjem razvijenog algoritma opisanog u 5 izvršeno je klasterovanje. Potrebno je naglasiti da se za dobijanje skupa nad kojim je izvršeno klasterovanje koristio test skup, nad kojim mreža nije obučavana. Test skup je veličine 10 000. Promena vrednosti Davies-Bouldin indeksa tokom procesa optimizacije se može videti na slici 7.10.



Slika 7.4: Rezultat rekonstrukcije autoenkodera nakon desete epohe.

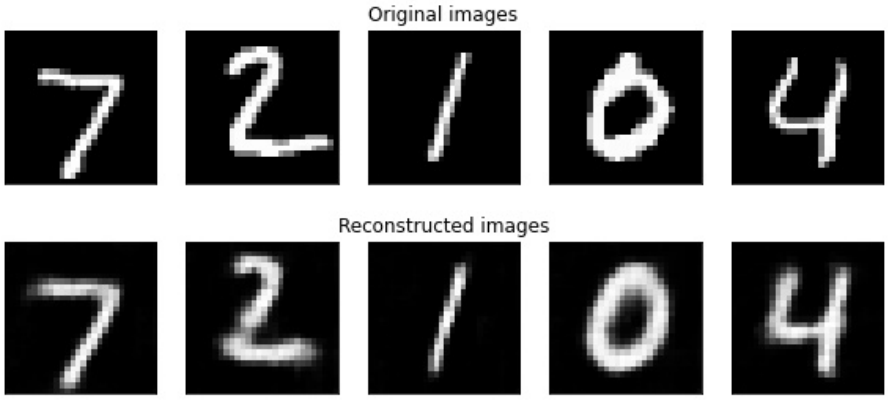


Slika 7.5: Rezultat rekonstrukcije autoenkodera nakon petnaeste epohe.

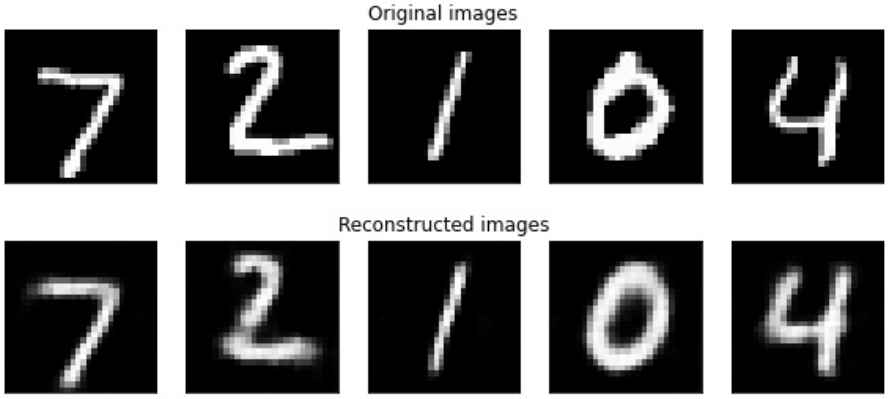
Minimalna vrednost dostignuta tokom procesa optimizacije indeksa je 1.09, dok je minimalna vrednost Davies-Bouldin indeksa postignuta algoritmom K-sredina 1.29.

Jedan on načina provere algoritma klasterovanja, kada su dostupne ciljne klase, je proveravanje raspodele ciljnih klasa unutar jednog klastera i njihova detaljnija analiza. Raspodele ciljnih klasa unutar svakog pojedinačnog klastera se mogu videti na slikama od 7.11 do 7.20. Potrebno je spomenuti da oznake klastera, od 0 do 9, ne ukazuju na stvarne klase instanci, jer se radi od nenadgledanom učenju, već ih treba posmatrati u formi neoznačenih grupa ili klastera.

Na slici 7.11 dominiraju instance, tj. slike broja 8, dok se javlja značajan broj



Slika 7.6: Rezultat rekonstrukcije autoenkodera nakon dvadesete epohe.



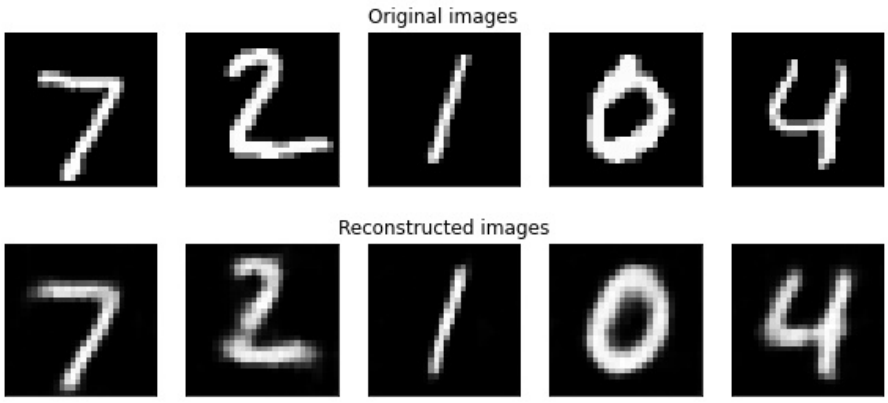
Slika 7.7: Rezultat rekonstrukcije autoenkodera nakon dvadeset pete epohe.

slika broja 5 i broja 3. To ima smisla jer su ta 3 broja vizuelno slična. Takođe treba imati u vidu da se skupo podataka MNIST sastoji od ručno pisanih cifara, tako da i ljudsko oko nekada nije u stanju da u potpunosti razazna koja je cifra na slici.

Analiza klastera prikazanog na slici 7.12 pokazuje da je algoritam većinom grupisao slike broja 0 i 6. Isto objašnjenje kao u prethodnom pasusu važi i za ovaj slučaj.

U klasterima 2, 3, 5, 6, 7 i 8 primetna je dominacija jedne ciljne klase.

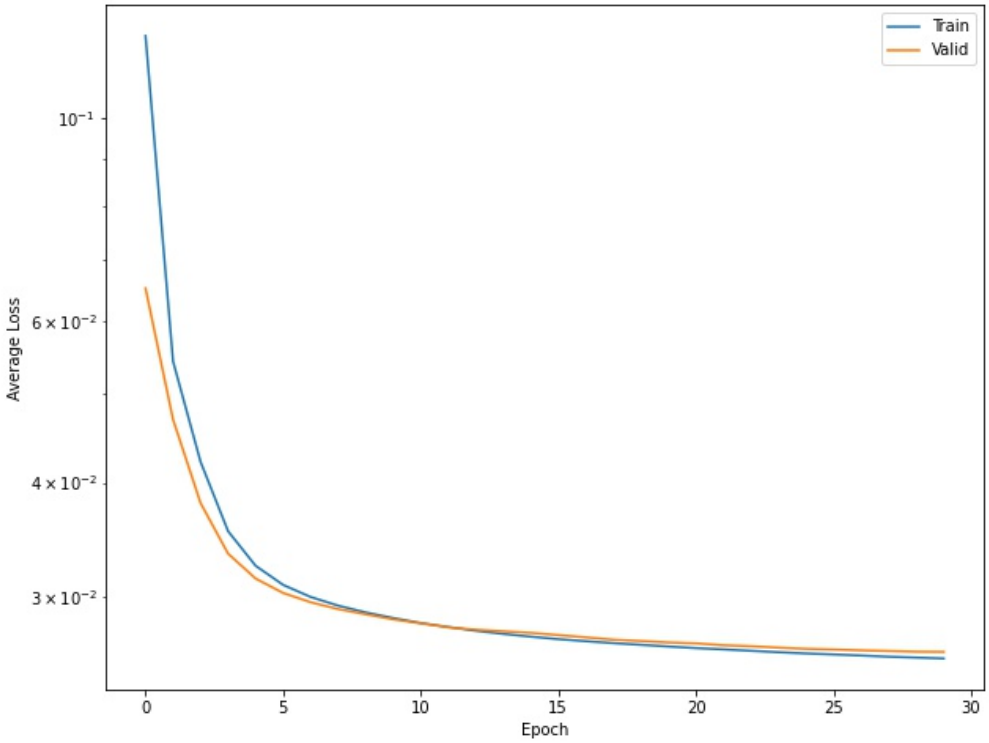
Analiza klastera 9, prikazana na slici 7.20 prikazuje da su zajedno grupisane cifre



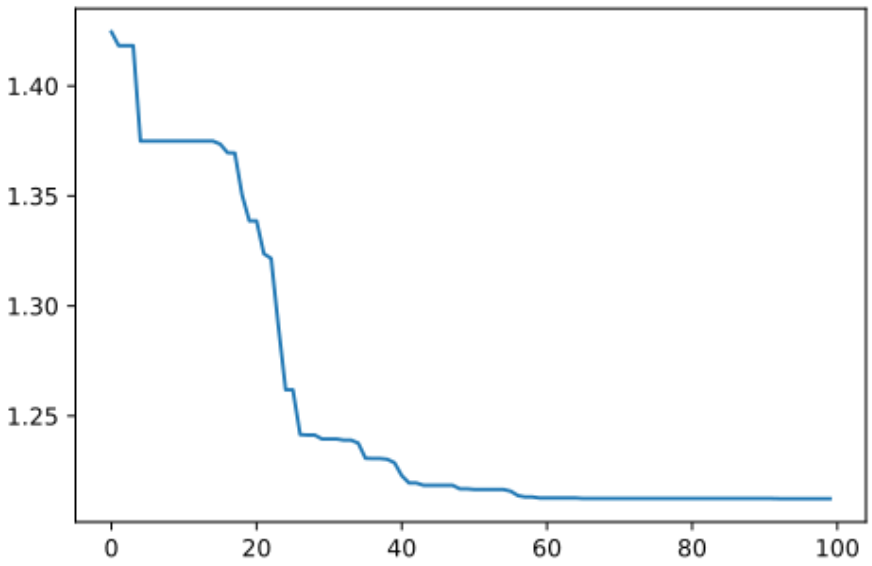
Slika 7.8: Rezultat rekonstrukcije autoenkodera nakon tridesete epohe.

4, 7 i 9. Na slici 7.21[1] dato je nekoliko primera ovih cifara iz skupa podataka koji govore u prilog dobijenom grupisanju.

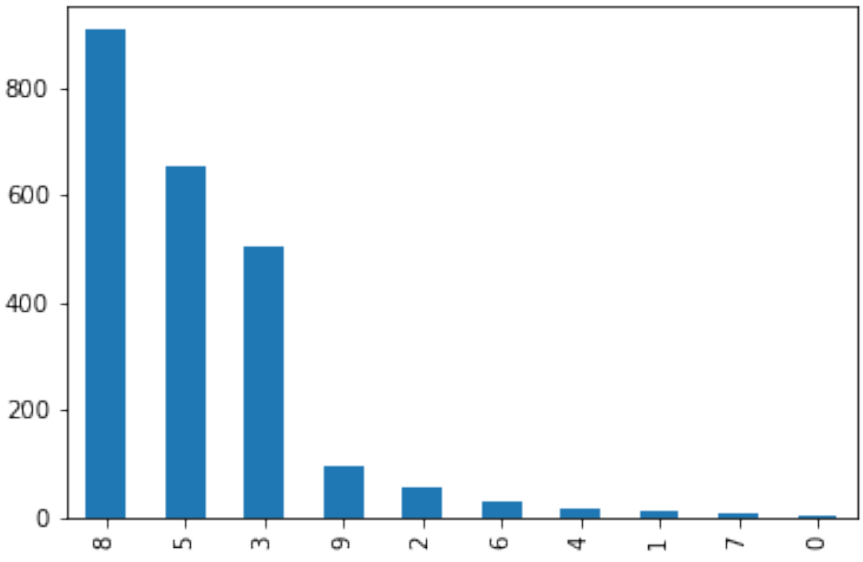
[1]Slika je preuzeta sa [6].



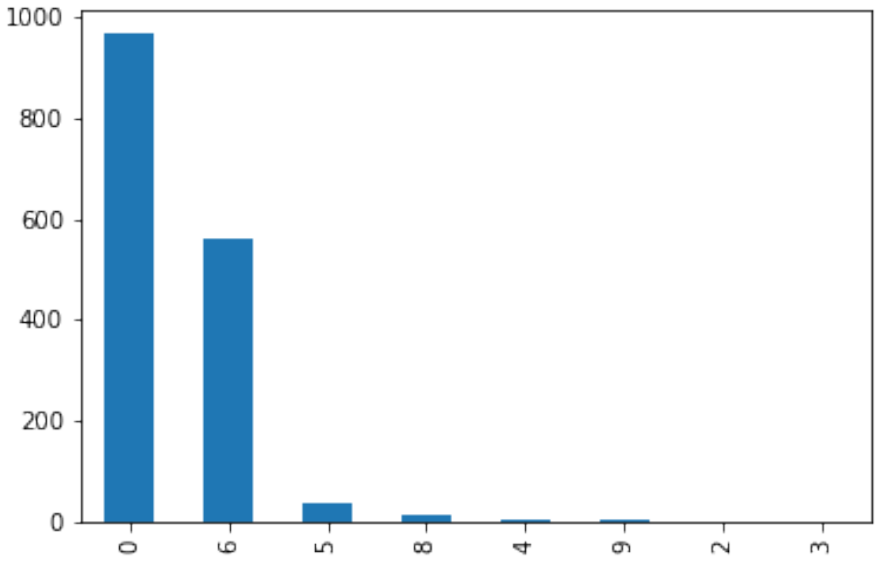
Slika 7.9: Opadanje srednje-kvadratne greške na skupovima za treniranje i validaciju tokom treniranja autoenkodera.



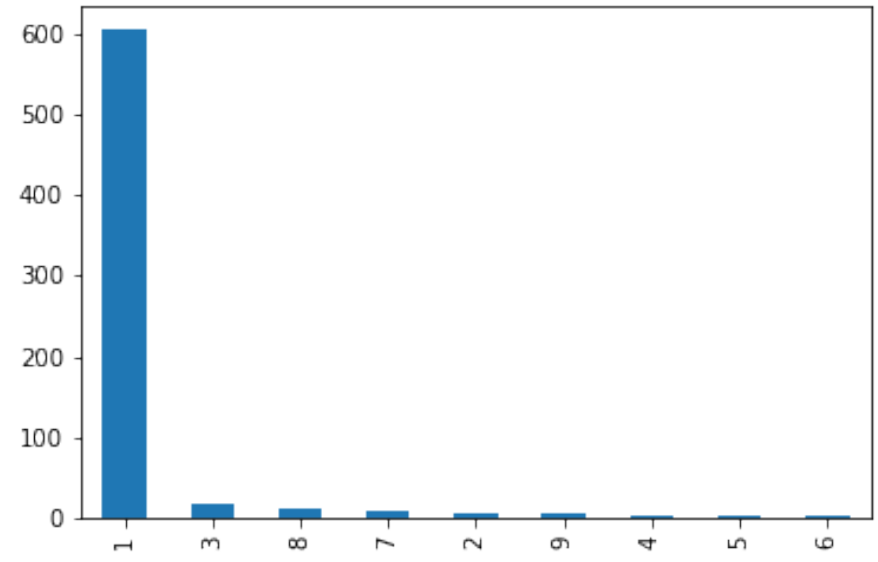
Slika 7.10: Promena Davies-Bouldin indeksa tokom optimizacija PSO algoritmom.



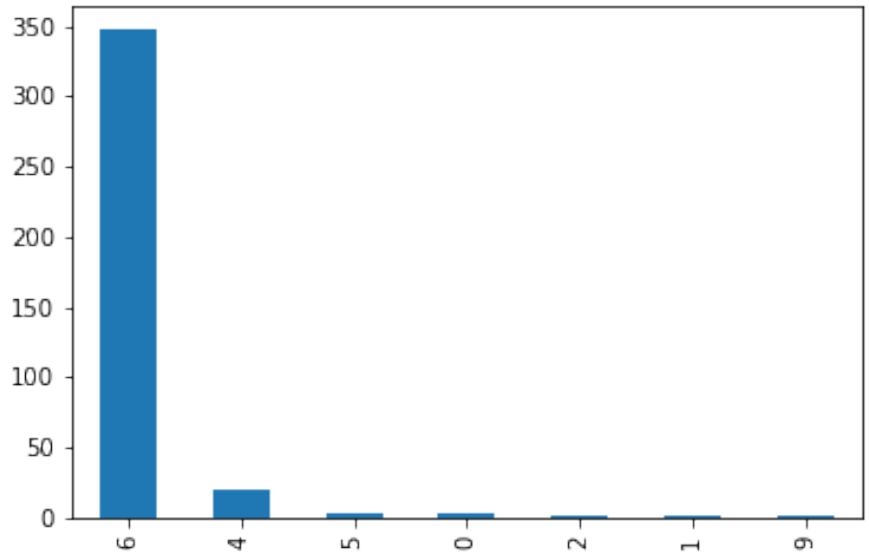
Slika 7.11: Distribucija cifara unutar klastera 0.



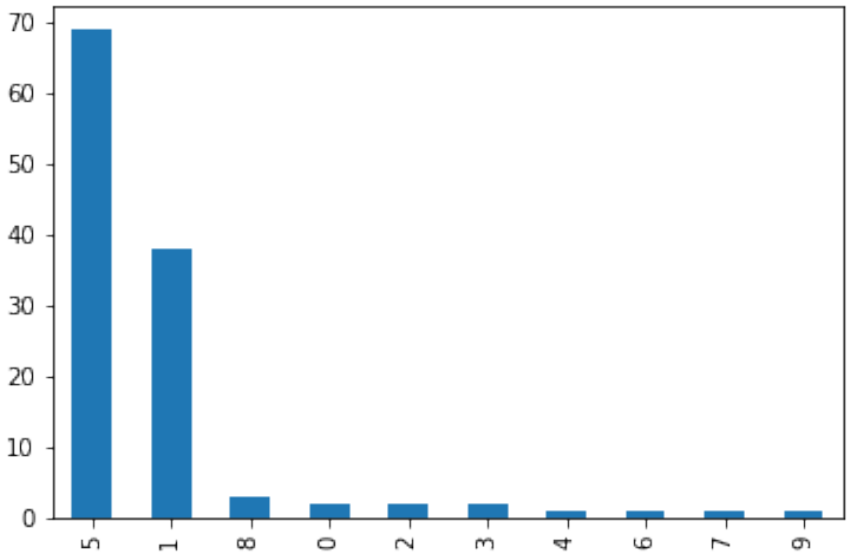
Slika 7.12: Distribucija cifara unutar klastera 1.



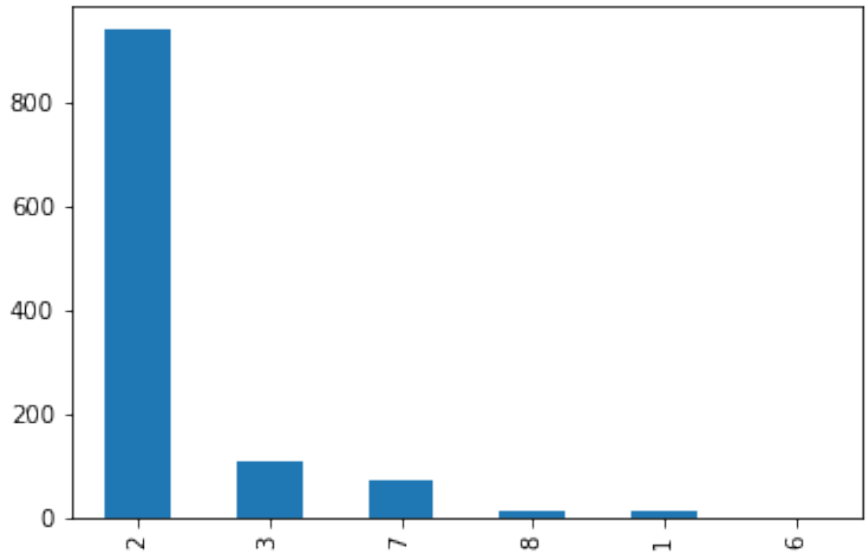
Slika 7.13: Distribucija cifara unutar klastera 2.



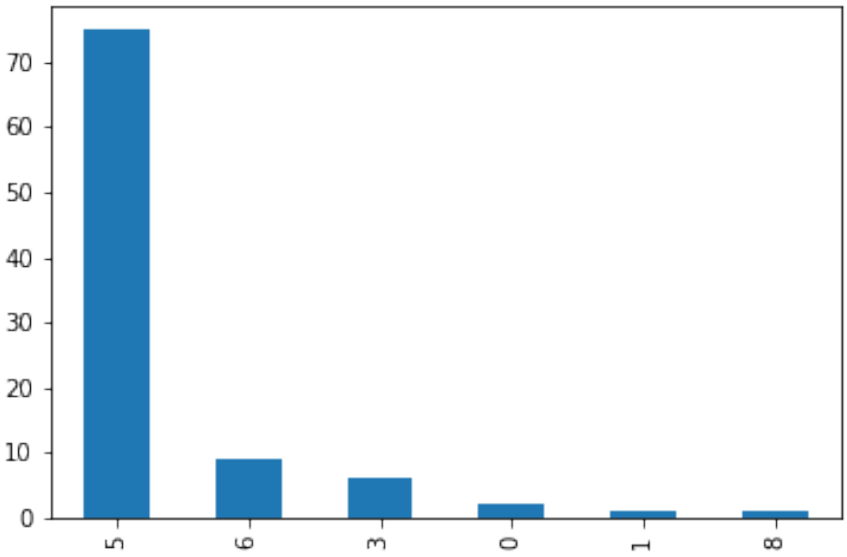
Slika 7.14: Distribucija cifara unutar klastera 3.



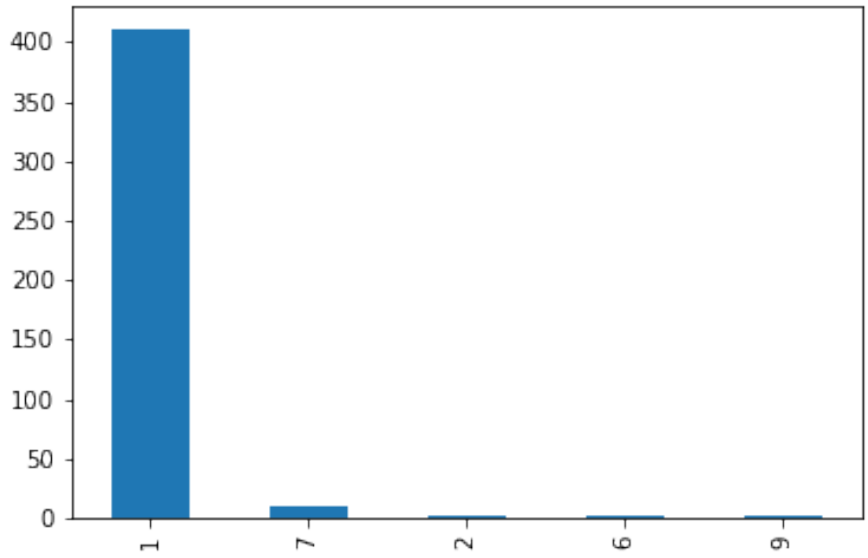
Slika 7.15: Distribucija cifara unutar klastera 4.



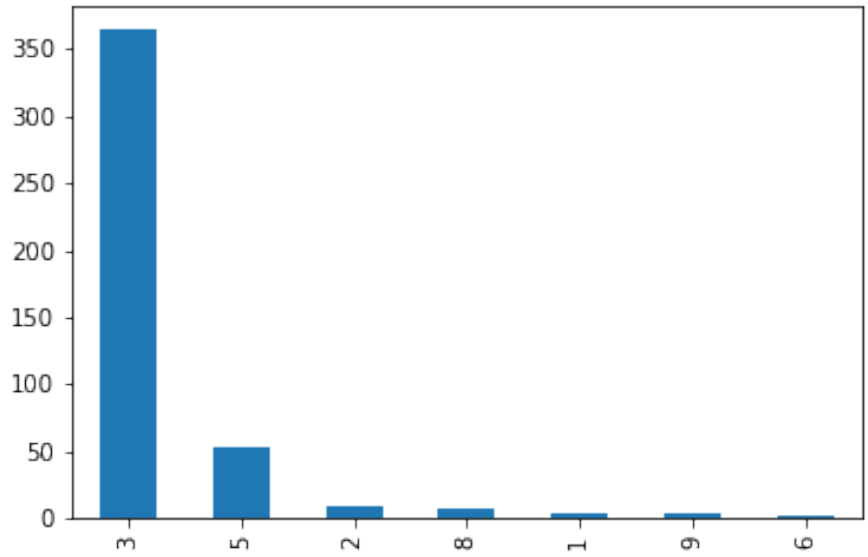
Slika 7.16: Distribucija cifara unutar klastera 5.



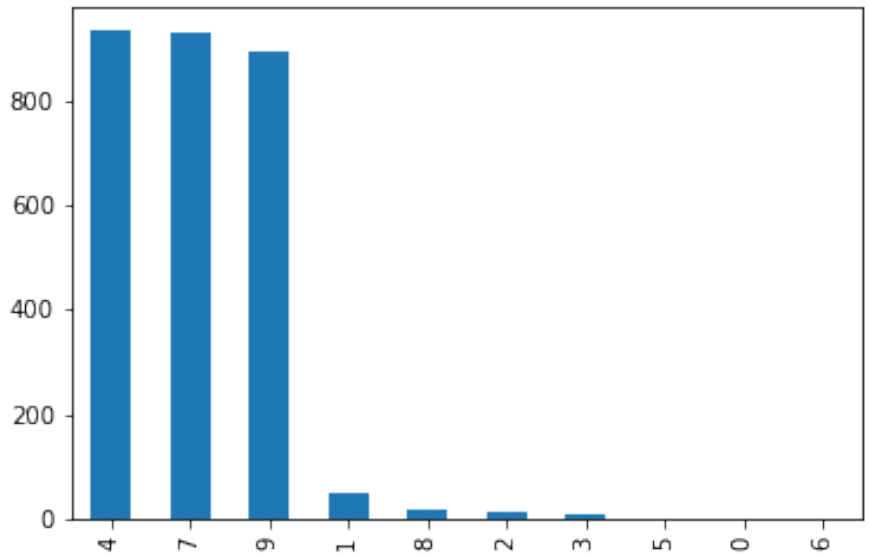
Slika 7.17: Distribucija cifara unutar klastera 6.



Slika 7.18: Distribucija cifara unutar klastera 7.



Slika 7.19: Distribucija cifara unutar klastera 8.



Slika 7.20: Distribucija cifara unutar klastera 9.

Slika 7.21: Primer cifara 4, 7 i 9 iz skupa podataka MNIST.

Glava 8

Zaključak

Nenadgledano učenje nad različitim domenima i vrstama podataka predstavlja jednu od najaktuelnijih tema istraživanja. Razlog tome su mogućnosti razvoja novih algoritama i metodologija, što je privlačno naučnim institucijama i istraživačima, kao i za potrebe industrije, jer su primene nendagledanog učenja velike.

U tezi je predstavljen novi algoritam klasterovanja zasnovan na PSO optimizaciji, kao i razvoj celokupnog sistema za klasterovanje visokodimenzionalnih podataka. Predstavljeni su osnovni koncepti relevantnih oblasti i ukratko spomenuti već postojeći algoritmi.

Algoritam klasterovanja zasnovan na centroidama se pokazao kao uspešan u rešavanju problema klasterovanja. Testiran je direktno nad skupovima podataka Wine i Iris i pokazao se uspešnim.

Celokupni sistem za klasterovanje visokodimenzionalnih podataka obučava i koristi odgovarajući autoenkoder u cilju smanjenja dimenzionalnosti, a zatim razvijeni algoritam da bi bio rešen problem klasterovanja. Razvijeni sistem je fleksibilan u smislu izbora arhitekture i vrste autoenkodera, kao i funkcije koja opisuje kvalitet klasterovanja. To omogućuje da znanje domenskog eksperta može dosta lako biti uključeno u ceo sistem, definisanjem odgovarajuće funkcije kvaliteta.

Na skupu podataka MNIST koji sadrži 60 000 ručno pisanih cifara, sistem se pokazao vrlo uspešnim. Većina dobijenih klastera sadrži ili većinom slike jedne cifre ili nekoliko cifara koje zaista jesu slične.

Može se zaključiti da predloženi sistem daje zadovoljavajuće rezultate, ali i otvara prostor za dalja istraživanja i poboljšanja. Jedan način poboljšanja celokupnog sistema bi bilo korišćenje informacije o kvalitetu klasterovanja tokom obučavanja autoenkodera.

Bibliografija

[1] Attention mechanism + relu activation function: adaptive parameterized relu activation function. https://developpaper.com/attention-mechanism-relu-activation-function-adaptive-parameterized-relu-activation-function/. Accessed: 2021-09-25.

[2] Cnn-introduction-to-pooling-layer. https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/. Accessed: 2021-09-25.

[3] A comprehensive guide to convolutional neural networks. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. Accessed: 2021-09-25.

[4] Convolutional neural networks. https://www.ibm.com/cloud/learn/convolutional-neural-networks. Accessed: 2021-09-25.

[5] K-means algorithm applied to image classification and processing. https://www.unioviedo.es/compnum/labs/PYTHON/kmeans.html. Accessed: 2021-09-25.

[6] Stackoverflow. https://stackoverflow.com/questions/32565438/why-does-my-neural-network-trained-on-mnist-data-set-not-predict-7-and-9-correct. Accessed: 2021-09-25.

[7] UCI Machine Learning repository: Iris data. https://archive.ics.uci.edu/ml/datasets/iris. Accessed: 2021-06-14.

[8] UCI Machine Learning repository: Wine data. https://archive.ics.uci.edu/ml/datasets/wine. Accessed: 2021-06-14.

[9] Using deep learning to investigate the neuroimaging-correlates of psychiatric and neurological disorders - methods and applications.

https://www.researchgate.net/Using-deep-learning-to-investigate-the-neuroimaging-correlates-of-psychiatric-and-neurological-disorders-Methods-and-applications. Accessed: 2021-09-25.

[10] Edgar Anderson. The species problem in iris. *Annals of the Missouri Botanical Garden*, 23(3):457–509, 1936.

[11] Richard E Bellman. *Adaptive control processes*. Princeton university press, 1961.

[12] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[13] David L. Davies and D. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:224–227, 1979.

[14] G.E. Hinton D.E. Rumelhart and R.J. Williams. Learning internal representations by error propagation. in parallel distributed processing. vol 1: Foundations. *MIT Press*, 1986.

[15] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[16] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning, 2018.

[17] Calinski T Harabasz and M Karoński. A dendrite method for cluster analysis. In *Communications in Statistics*, volume 3, pages 1–27. 1974.

[18] Nikolić M. i Zečević A. *Mašinsko učenje*. 2019.

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[20] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., USA, 1988.

[21] K Sparck Jones and Cornelis Joost Van Rijsbergen. Information retrieval test collections. *Journal of documentation*, 1976.

[22] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

[23] J. Kennedy and R. Eberhart. A novel approach of data clustering using an improved particle swarm optimization based k–means clustering algorithm. *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 1-6.*, 2020.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[25] Derong Liu, Shengli Xie, Yuanqing Li, Dongbin Zhao, and El-Sayed El-Alfy. *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part II*. 01 2017.

[26] Gilbert Strang. *Linear Algebra and its Applications*. 1986.

[27] Junfeng Zhang, Wei Chen, Mingyi Gao, and Gangxiang Shen. K-means-clustering-based fiber nonlinearity equalization techniques for 64-qam coherent optical communication system. *Opt. Express*, 25(22):27570–27580, Oct 2017.

Biografija autora

Denis Aličić (*Loznica, 15. septembar 1996. — *) završio je OŠ „Borivoje Ž. Milojević" u Krupnju kao nosilac Vukove diplome i đak generacije. Gimanziju u Krupnju završava takođe sa odličnim uspehom i dobitnik je Vukove diplome. Matematički fakultet Univerziteta u Beogradu - smer Informatika upisuje 2015. godina i završava 2019. sa prosekom 8.67. Upisuje master studije 2019. godine i od tada radi na Matematičkom fakultetu kao saradnik u nastavi sa procentom angažovanja od 50%. Tokom dosadašnjeg angažovanja držao je vežbe na predmetima: Programiranje 1, Objektno-orijentisano programiranje, Veštačka inteligencija i Računarska inteligencija. Tokom osnovnih studija završava praksu u MDCS (eng. *Microsoft Development Center Serbia*) gde je radio na razvoju i implementaciji modela mašinskog učenja. Od 2019. do 2021. radi u kompaniji Endava kao C++ programer. Od juna 2021. je zaposlen u kompaniji Nordeus gde radi na poziciji inženjera veštačke inteligencije. Bio je član organizacionog odbora konferencije „Symopis2021" na kojoj je predstavio dva rada zajedno sa drugim kolegama.