



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Алекса Миленовић

Поређење више приступа за класификацију слика одевних предмета

ДИПЛОМСКИ РАД

-Основне академске студије-

Нови Сад, 2022.



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Поређење више приступа за класификацију слика одевних предмета

ДИПЛОМСКИ РАД

-Основне академске студије-

кандидат

Алекса Миленовић, ИН30/2017

ментор

проф. др Татјана Лончар-Турукало

Нови Сад, 2022.



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА


Редни број, РБР:	
Идентификациони број, ИБР:	
Тип документације, ТД:	МОНОГРАФСКА ПУБЛИКАЦИЈА
Тип записа, ТЗ:	ТЕКСТУАЛНИ ШТАМПАНИ МАТЕРИЈАЛ
Врста рада, ВР:	ДИПЛОМСКИ РАД
Аутор, АУ:	АЛЕКСА МИЛЕНОВИЋ
Ментор, МН:	Др ТАТЈАНА ЛОНЧАР-ТУРУКАЛО
Наслов рада, НР:	ПОРЕЂЕЊЕ ВИШЕ ПРИСТУПА ЗА КЛАСИФИКАЦИЈУ СЛИКА ОДЕВНИХ ПРЕДМЕТА
Језик публикације, ЈП:	СРПСКИ
Језик извода, ЈИ:	СРПСКИ
Земља публикавања, ЗП:	СРБИЈА
Уже географско подручје, УГП:	ВОЈВОДИНА
Година, ГО:	2022.
Издавач, ИЗ:	АУТОРСКИ РЕПРИНТ
Место и адреса, МА:	НОВИ САД, ТРГ ДОСИТЕЈА ОБРАДОВИЋА 6а
Физички опис рада, ФО: (поглавља/страна/цитата/табела/слика/графика/прилога)	4 поглавља / 33 стране / 18 цитата / 10 табела / 14 слика / 1 графика / 0 прилога
Научна област, НО:	ЕЛЕКТРОТЕХНИКА / ИНФОРМАЦИОНИ ИНЖЕЊЕРИНГ
Научна дисциплина, НД:	ПРЕПОЗНАВАЊЕ ОБЛИКА
Предметна одредница/Кључне речи, ПО:	Класификација, азлагање на главне компоненте, К најближих суседа, Машине на бази вектора носача, Вишеслојни перцептрон, Конволуциона неуронска мрежа, FASHION-MNIST база
УДК	
Чува се, ЧУ:	БИБЛИОТЕКА ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА, НОВИ САД, ТРГ ДОСИТЕЈА ОБРАДОВИЋА 6а
Важна напомена, ВН:	НЕМА
Извод, ИЗ:	У овом раду описују се и евалуирају више класификационих алгоритама на проблему категоризације слика одевних предмета (FASHION MNIST база). Уз претрагу најбољих параметара и три скупа за обуку (30% базе, цео тренинг део и уз редукцију димензионалности), показало се да се најбоље перформансе од 92,11% просечне тачности постижу конволуционим неуронским мрежама.
Датум прихватања теме, ДП:	09.09.2022.
Датум одбране, ДО:	14.09.2022.
Чланови комисије, КО:	Председник: др. Милан Сечујски, ред. проф.
	Члан: др. Никша Јаковљевић, ванр. проф.
	Члан, ментор: др. Татјана Лончар-Турукало, ред. проф.
	Потпис ментора



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Accession number, ANO :			
Identification number, INO :			
Document type, DT :	MONOGRAPHIC PUBLICATION		
Type of record, TR :	TEXTUAL MATERIALS, PRINTED		
Contents code, CC :	BACHELOR THESIS		
Author, AU :	ALEKSA MILENOVIĆ		
Mentor, MN :	TATJANA LONČAR-TURUKALO, PhD		
Title, TI :	COMPARISON OF MULTIPLE APPROACHES FOR CLASSIFICATION OF GARMENT IMAGES		
Language of text, LT :	SERBIAN		
Language of abstract, LA :	SERBIAN		
Country of publication, CP :	SERBIA		
Locality of publication, LP :	VOJVODINA		
Publication year, PY :	2022.		
Publisher, PB :	AUTHOR'S REPRINT		
Publication place, PP :	FTN, NOVI SAD, TRG DOSITEJA OBRADOVIĆA 6a		
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	4 chapters / 33 pages / 18 ref. / 10 tables / 14 pictures / 1 graphs / 0 appendixes		
Scientific field, SF :	ELECTRICAL ENGINEERING / INFORMATION ENGINEERING		
Scientific discipline, SD :	PATTERN RECOGNITION		
Subject/Key words, S/KW :	Classification, Principal component analysis, K nearest neighbors, Support vector machines, Multilayer perceptron, Convolutional neural network, FASHION-MNIST		
UC			
Holding data, HD :	LIBRARY OF THE FACULTY OF TECHNICAL SCIENCES, NOVI SAD, TRG DOSITEJA OBRADOVIĆA 6a		
Note, N :	NONE		
Abstract, AB :	In this thesis several classification algorithms have been evaluated for categorization of garment images (FASHION-MNIST). With hyperparameter tuning and three different input options (30% of the training set and whole training set with and without dimensionality reduction), the best average accuracy of 92,11% has been achieved with convolutional neural networks.		
Accepted by the Scientific Board on, ASB :	09.09.2022.		
Defended on, DE :	14.09.2022.		
Defended Board, DB :	President:	Milan Sečujski, PhD, full professor	Mentor's sign
	Member:	Nikša Jakovljević, PhD, associate professor	
	Member, mentor:	Tatjana Lončar-Turukalo, PhD, full professor.	

	УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА	Датум:
	21000 НОВИ САД, Трг Доситеја Обрадовића 6	
	ЗАДАТАК ЗА ИЗРАДУ ДИПЛОМСКОГ (BACHELOR) РАДА	Лист/Листова: 1/39

(Податке уноси предметни наставник - ментор)

Врста студија:	<input type="checkbox"/> Основне академске студије
Студијски програм:	Informacioni inženjering
Руководилац студијског програма:	dr Slavica Koridić

Студент:	Aleksa Milovanović	Број индекса:	IN 30/2017
Област:	Prepoznavanje oblika		
Ментор:	dr Tatjana Lončar-Turukalo		
НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА ДИПЛОМСКИ (Bachelor) РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА: - проблем – тема рада; - начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна; - литература			

НАСЛОВ ДИПЛОМСКОГ (BACHELOR) РАДА:

Поређење више приступа за класификацију слика одевних предмета
--

ТЕКСТ ЗАДАТКА:

<ul style="list-style-type: none"> • Proučiti i opisati FASHION-MNIST bazu i izvršiti odgovarajuću predobradu slika i odgovarajuću podelu na trening i test skup. • Odabrati nekoliko klasifikacionih pristupa koji će biti primenjeni na opisanoj bazi • Opisati izabrane klasifikacione pristupe • Izabrati i opisati jedan pristup za redukciju dimenzionalnosti podataka • Izvršiti obuku više odabranih klasifikatora na osnovu 30% podataka, celokupnog trening skupa i celokupnog trening skupa uz redukciju dimenzionalnosti • Proceniti performanse dobijenih klasifikatora • Sumirati rezultate i izvesti odgovarajuće zaključke

Руководилац студијског програма:	Ментор рада:

Примерак за: <input type="checkbox"/> - Студента; <input type="checkbox"/> - Ментора
--

SADRŽAJ

1. Uvod	1
2. Metodologija	3
2.1 Baza podataka	3
2.2 Metoda za smanjenje obeležja.....	5
2.3 Algoritmi mašinskog učenja.....	8
2.3.1 K najbližih suseda.....	8
2.3.2 Mašine na bazi vektora nosača.....	10
2.3.3 Višeslojni perceptron.....	14
2.3.4 Konvoluciona neuronska mreža	18
2.4 Način evaluacije rezultata.....	21
3. Rezultati modela.....	23
3.1 Rezultati modela na bazi K najbližih suseda.....	23
3.2 Rezultati modela na bazi vektora nosača	25
3.3 Rezultati modela na bazi višeslojnog perceptrona	27
3.4 Rezultati modela na bazi konvolucione neuronske mreže.....	29
4. Zaključak	31
Literatura	32

1. Uvod

21. vek obeležen je značajnim razvojem polja veštačke inteligencije (eng. *artificial intelligence* – AI). Zavisnost čoveka od računara je na najvišem stepenu u istoriji čovečanstva. U prošlosti, pojam *veštačke inteligencije* je korišćen kako bi opisao mašine koje oponašaju i prikazuju ljudske kognitivne veštine poput učenja i rešavanja problema. Ova definicija je danas odbačena, a istraživači AI opisuju u smislu racionalnosti i racionalnog delovanja, čime se ne postavlja ograničenje načina na koji se inteligencija može artikulirati. Kao jedna od značajnih oblasti veštačke inteligencije ističe se mašinsko učenje (eng. *machine learning*) koje omogućava softverskim aplikacijama veću preciznost pri predviđanju željenih ishoda, bez eksplicitne pomoći od strane programera, na osnovu velikog broja (obeležanih) uzoraka određenog problema.

Desetine miliona slika se zabeleži svakog dana za različite primene u skoro svim domenima ljudskog postojanja. Ljudi su počeli da se u sve većoj meri oslanjaju na mašinsko učenje, koje podrazumeva i duboko učenje, u analiziranju i izvlačenju korisnih informacija iz snimljenih slika a kroz prepoznavanje obrazaca koji će korisnicima biti od nekog značaja. Poslednjih godina, ove metode, posebno one koje koriste duboko učenje, imaju visok nivo performansi koje prevazilaze ljudska delovanja u određenim aspektima [1].

Osnovni zadaci računarske vizije obuhvataju metode za pribavljanje, obradu, analizu i razumevanje digitalnih slika, uključujući i videa, kao i izdvajanje visokodimenzionih podataka iz stvarnog sveta kako bi se proizvele korisne numeričke informacije u obliku donošenja određenih odluka. Klasifikacija slika ima veliku ulogu u automobilske industriji, bezbednosnoj i medicinske industriji.

Tokom poslednjih godina povećalo se interesovanje za Fashion-MNIST bazu podataka, koja predstavlja dobru osnovu za razumevanje računarske vizije, a prisutna je i težnja da se dobije što bolja tačnost pri klasifikaciji njenih uzoraka. Ova baza je sačinjena od 70000 slika različitih odevnih predmeta, pri čemu svaki pripada jednoj od 9 klasa. U ovom diplomskom radu korišćeni su algoritmi mašinskog učenja sa ciljem prepoznavanja klasa kojima pripadaju odevni predmeti sa slika u Fashion MNIST. U njih spadaju K najbližih suseda (eng. *k nearest neighbors*), mašina na bazi vektora nosača (eng. *support vector machine*), višeslojni perceptron (eng. *multilayer perceptron*) i konvoluciona neuronska mreža (eng. *convolutional neural network*).

Razvojem dubokih neuronskih mreža ostvareni su značajni uspesi u različitim poljima i ujedno se pokazalo da tip konvolucionih neuronskih mreža ostvaruje odlične rezultate u radu sa Fashion MNIST bazom. Primera radi, u [2] predlaže se klasifikacija slika uz pretragu i podešavanje arhitekture (DARTS- eng. *differential architecture search*) korišćenjem fiksnih operacija. DARTS je računarski jednostavna, široko primenjena stohastička metoda koja pravi veliki broj aproksimacija kako bi se ubrzala izračunavanja. U njih spadaju korišćenje konačne razlike aproksimacija (eng. *finite difference approximation*) i korišćenje težinskih koeficijenata narednog koraka umesto optimalnih težinskih koeficijenata, čime se postižu slabije performanse. Uvođenje fiksnih operacija inspirisano je dobrom metodom podešavanja neuronskih mreža kako bi se postigli najbolji rezultati. One koriste module pažnje (eng. *attention modules*) koji se koriste kako bi usmerili fokus konvolucione neuronske mreže na važne informacije umesto da uče nekorisne pozadinske informacije. Njihova metoda primenjuje

automatsku pretragu arhitekture (eng. *automatic architecture search*) uz ručno kreirane operacije koje poboljšavaju klasifikaciju. DARTS predlaže metodu kojom se arhitektura ažurira minimizacijom funkcije cene nad skupom za validaciju korišćenjem delta pravila. Arhitektura je sledeća: u fazi sažimanja, svaki modul pažnje koristi globalni prosečan operator sažimanja. Nakon ove faze slede dva mala, potpuno povezana sloja, posle čega dolazi operacija predimenzionisanja. Odnos redukcije r je hiperparametar koji omogućava promenu kapaciteta i računске težine modula pažnje. Vrednosti parametara korišćenih u radu su početna brzina učenja $\alpha = 0.025$, veličina uzoračkog podskupa 32 i broj epoha 50. U ovom radu postignuta je tačnost (eng. *accuracy*) pri klasifikaciji na Fashion-MNIST bazi od 96.91% [2].

U radu [3] vrši se klasifikacija Fashion-MNIST baze podataka korišćenjem LeNet-5 arhitekture. Ovde je po prvi put primenjena arhitektura neuronske mreže nad ovom bazom. LeNet-5 je izgrađen na principima lokalnih receptivnih polja, deljenih težina i posebnog načina sažimanja. Njegova arhitektura je sačinjena od 8 slojeva: ulaznog sloja, dva sloja sažimanja pomoću srednje vrednosti, tri konvoluciona sloja, jednog potpuno povezanog sloja i potpuno povezanog *softmax* izlaznog sloja. Prvi konvolucionni sloj prima za ulaz slike dimenzija 32×32 sa 6 filtra dimenzija 5×5 i jednim korakom za pomeranje filtra. Drugi sloj prima ulaz iz konvolucionog sloja sa filtrom veličine 2×2 i dva koraka nakon prolaska kroz sloj sažimanja. Ovi slojevi sažimanja su dodati nakon prvog i drugog konvolucionog sloja. Vrednosti parametara korišćenih u ovom radu su brzina učenja $\alpha = 0.005$, veličina uzoračkog podskupa 32 i broj epoha 10. Njihov model je ostvario prosečnu tačnost od 98,9% unakrsnom validacijom podelom trening skupa na 10 podskupova [3].

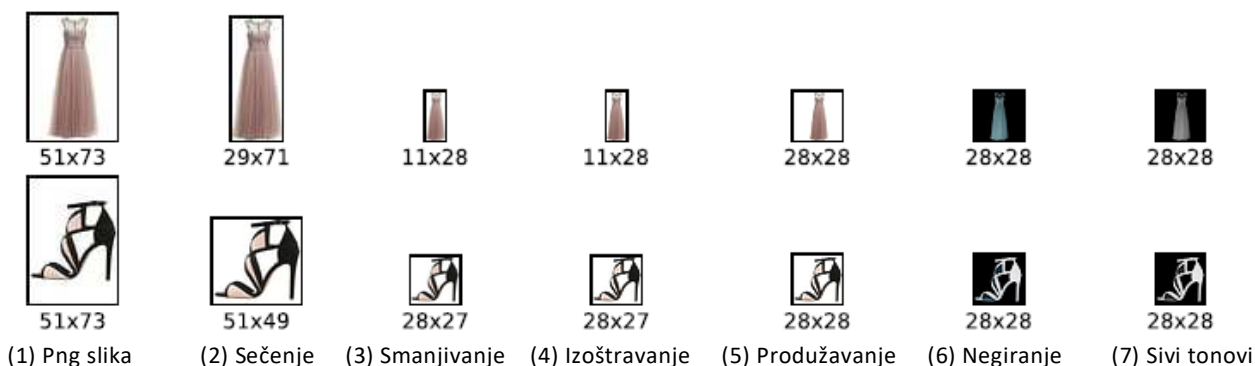
Ova teza sastoji se sastoji iz četiri poglavlja, u kojima je pored uvodnog dela opisana korišćena metodologija, izloženi rezultati svih obučavanih modela i izvedeni odgovarajući zaključci. U poglavlju, koje se odnosi na metodologiju, se nalazi detaljan opis baze podataka nad kojom su modeli obučeni, kao i metoda za smanjenje broja obeležja. Ovde je predstavljena i teorijska osnova četiri algoritma mašinskog učenja koji su korišćeni kao i načini na koje je izvršena evaluacija rezultata. U poglavlju sa rezultatima prikazani su rezultati dobijeni unakrsnom validacijom algoritama pre i nakon korišćenja metode za smanjenje obeležja. U poslednjem poglavlju – zaključku, sumirani su rezultati i dat je predlog na koji način bi se rad mogao dalje poboljšati i razviti.

2. Metodologija

2.1 Baza podataka

Fashion-MNIST baza podataka je napravljena na osnovu asortimana odeće i obuće Zalando modne internet prodavnice. Svaki od artikala je uslikan na svetlo sivoj pozadini iz više različitih uglova kako bi njegovi različiti aspekti bili prikazani i ove slike su dostupne u više rezolucija. Slike koje se nalaze u ovoj bazi obuhvataju slike odevnih predmeta sa rezolucijama ikonica (eng. *thumbnail*) dimenzija 51×73 piksela JPEG formata, a koje su radi dalje obrade prebačene u PNG format (Slika 1.1). Nakon ovoga je izvršeno izdvajanje regiona od interesa (ROI) svake slike, odn. uklanjanje pozadine (Slika 1.2). Kao merilo se pritom koristi maksimalno odstupanje RGB boje piksela do 5%, gde su kao referentne tačke uzete boje piksela na samim ćoškovima slika.

Dalje, modifikacija ulaznih slika izvršena je poduzorkovanjem piksela (Slika 1.3), pri čemu su pojedine kolone i vrste izbačene tako da je nova dužina duže ivice slike 28 piksela. Izoštavanje slika (Slika 1.4) je urađeno korišćenjem Gausovog operatora radijusa i standardne devijacije 1.0, čiji je efekat izraženiji na samim obodima slika. Izjednačavanje dužina stranica je postignuto produžavanjem kraće stranice na dužinu od 28 piksela, nakon čega su odevni predmeti postavljeni u centar slika (Slika 1.5). Odrađena je negacija boja slika (Slika 1.6), a zatim je izvršena konverzija u sivu skalu (Slika 1.7) pri čemu pikseli imaju 8-bitnu vrednost sivih tonova (eng. *grayscale*). Ceo postupak prikazan je na Slici 1 [4].



Slika 1. Proces transformacije slika korišćenih u Fashion-MNIST bazi [4]

U bazi podataka se nalaze trening i test skupovi, pri čemu se trening skup sastoji iz 60000 uzoraka, a test skup od njih 10000. Uzorci su predstavljeni nizovima od 785 obeležja, gde prvo obeležje označava klasu kojoj odevni predmet pripada, a ostalih 784 obeležja su pikseli slike. Vrednost pojedinačnog piksela je predstavljena u obliku broja koji uzima vrednost sivih tonova između 0 i 255, gde 0 odgovara skroz crnom pikselu, a 255 skroz belom pikselu. Postoji 10 različitih klasa koje su označene brojevima od 0 do 9 (Tabela 1).

Sva obeležja su kategorička i ne postoje nedostajuće vrednosti u bazi podataka. Klase su podjednako zastupljene u trening i test skupovima. U trening skupu se nalazi po 6000 uzoraka od svakog odevnog predmeta, a u test skupu po njih 1000. Postoji značajna pravilnost koja se uočava kod svakog uzorka u vidu koncentracije crnih piksela po obodima slika koji su nastali kao posledica transformacija originalnih slika odevnih predmeta u bazu podataka. Ova pojava se preslikava na podatke u vidu

postojanja velikog broja obeležja čija je vrednost ista za sve uzorke. Uzimajući u obzir da obeležja predstavljaju piksele slika, posmatranje statističkih parametara poput srednje vrednosti ili odstupajućih vrednosti nema značajnog uticaja.


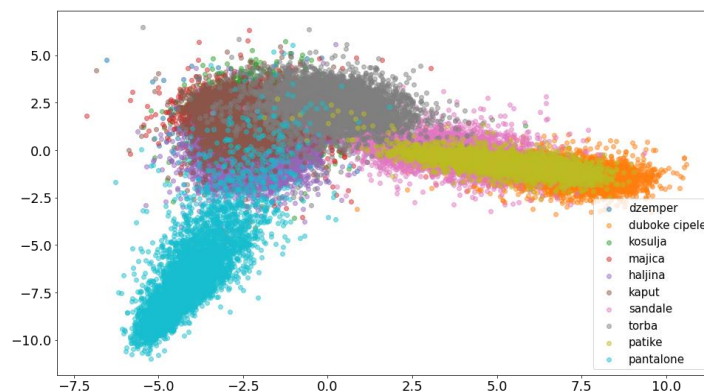
Brojčana vrednost klase	Naziv klase na		Primeri slika klase
	srpskom jeziku	engleskom jeziku	
0	Majica	T-shirt/Top	
1	Pantalone	Trousers	
2	Džemper	Pullover	
3	Haljina	Dress	
4	Kaput	Coat	
5	Sandale	Sandals	
6	Košulja	Shirt	
7	Patike	Sneaker	
8	Torba	Bag	
9	Duboke cipele	Ankle boots	

Tabela 1. Imena klase i primeri njihovih slika

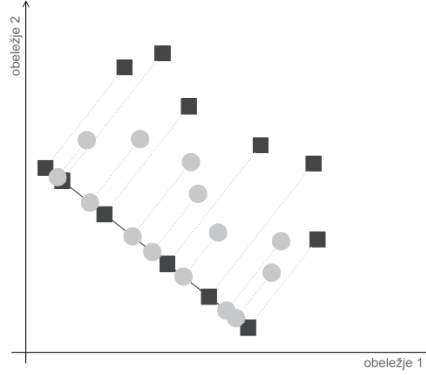
Grafičko prikazivanja podataka (Grafik 1) izvršeno je smanjenjem dimenzionalnosti piksela sa 784 na 2 dimenzije, korišćenjem LDA redukcije, pri čemu se primećuje da ne postoji značajnija razdvojenost između određenih klasa. Ovo je direktna posledica fizičkog izgleda i karakteristika samih odevnih predmeta. Uočava se da postoji velika sličnost između klasa *sandale*, *patike* i *duboke cipele*, kao i da se klasa *pantalone* jasno razdvaja od ostalih klasa.



Grafik 1. Podaci prikazani u 2d prostoru nakon LDA redukcije

2.2 Metoda za smanjenje obeležja

Razlaganje na glavne komponente (eng. *principal component analysis - PCA*) predstavlja metodu za redukciju dimenzionalnosti čija je svrha transformacija skupa obeležja iz višedimenzionalnog prostora u prostor sa manjim brojem dimenzija uz očuvanje varijanse između podataka (Slika 2). Ona spada u grupu tehnika reprezentacije jer obezbeđuje kvalitetan prikaz informacije o rasipanju podataka. Primenom linearnih transformacija PCA transformiše korelisane promenljive u nove nekorelisane promenljive koje se nazivaju glavnim komponentama.



Slika 2. Reprezentacija uzoraka iz 2d prostora u 1d prostoru

Neka je svaki uzorak u skupu podataka dat u obliku vektora \mathbf{x} koji ima d obeležja i koji se nalazi u d dimenzionom prostoru čije su glavne ose određene njegovim obeležjima. Zadatak PCA je da predstavi ove uzorke preko novih ortonormalnih baznih vektora \mathbf{u}_i , $i = 1, \dots, d$:

$$\mathbf{x} = \sum_{i=1}^d x_{u_i} \mathbf{u}_i, \quad (1)$$

gde x_{u_i} predstavlja projekciju vektora \mathbf{x} na određene \mathbf{u}_i pravce tj. $x_{u_i} = \mathbf{x}^T \mathbf{u}_i$. Kako bi se izvršila redukcija potrebno je odabrati bazu \mathbf{u}_i takvu da se uzorak \mathbf{x} može aproksimirati sa M baznih vektora ($M \ll d$), pri čemu će on i dalje biti verno predstavljen. PCA nastoji da pronađe bazu razlaganja i vrednost M koja će praviti najmanju grešku. Prvih M baznih vektora se zadržavaju u originalnim koordinatama, dok se preostale $M + 1, \dots, d$ koordinate zamenjuju sa konstantama b_i :

$$\hat{\mathbf{x}}(M) = \sum_{i=1}^M x_{u_i} \mathbf{u}_i + \sum_{i=M+1}^d b_i \mathbf{u}_i, \quad (2)$$

gde je $\hat{\mathbf{x}}(M)$ aproksimacija uzorka sa M vektora. Greška se nalazi na onim koordinatama koje su aproksimirane konstantama b_i i ona se lako može izraziti preko vektora koji se odbacuje. Dakle, b_i su konstante kojima se menjaju stvarne projekcije vektora \mathbf{x} na svaki od pravaca i greška se onda zapisuje u sledećem obliku:

$$\Delta \mathbf{x}(M) = \mathbf{x} - \hat{\mathbf{x}}(M) = \sum_{i=1}^d x_{u_i} \mathbf{u}_i - \left(\sum_{i=1}^M x_{u_i} \mathbf{u}_i + \sum_{i=M+1}^d b_i \mathbf{u}_i \right) = \sum_{i=M+1}^d (x_{u_i} - b_i) \mathbf{u}_i \quad (3)$$

Vektor \mathbf{u}_i se bira tako da je greška najmanja, a ona se meri preko norme vektora razlike iz izraza (3). Za računanje greške koristi se srednjkvadratna greška:

$$\varepsilon(M) = E(\|\Delta \mathbf{x}(M)\|^2) = \sum_{i=M+1}^d E((x_{u_i} - b_i)^2) \quad (4)$$

Koeficijenti b_i za koje je ova razlika najmanja, dobijaju se kada se uradi prvi izvod greške po svakom od b_i i dobijene jednačine izjednače sa 0. Dobija se da je b_i jednaka očekivanju vrednosti projekcija vektora \mathbf{x} u novom prostoru određenom baznim vektorima \mathbf{u}_i :

$$\frac{\partial}{\partial b_i} E((x_{u_i} - b_i)^2) = -2(E(x_{u_i}) - b_i) = 0 \rightarrow b_i = E(x_{u_i}) \quad (5)$$

Kada se prethodno dobijena vrednost b_i ubaci u jednačinu za računanje srednjkvadratne greške (izraz 4) i kada se vrednosti x_{u_i} zamene skalarnim proizvodom vektora \mathbf{x} sa odgovarajućim vektorima baze \mathbf{u}_i dobija se sledeći izrazi:

$$\begin{aligned} \varepsilon(M) &= \sum_{i=M+1}^d E((x_{u_i} - b_i)^2) = \sum_{i=M+1}^d E((x_{u_i} - E(x_{u_i}))^2) = \sum_{i=M+1}^d E((\mathbf{x}^T \mathbf{u}_i - E(\mathbf{x}^T \mathbf{u}_i))^2) \\ &= \sum_{i=M+1}^d E((\mathbf{x}^T \mathbf{u}_i - E(\mathbf{x}^T \mathbf{u}_i))^T (\mathbf{x}^T \mathbf{u}_i - E(\mathbf{x}^T \mathbf{u}_i))) = \sum_{i=M+1}^d \mathbf{u}_i^T E((\mathbf{x} - E(\mathbf{x}))(\mathbf{x} - E(\mathbf{x}))^T) \mathbf{u}_i = \sum_{i=M+1}^d \mathbf{u}_i^T \Sigma_x \mathbf{u}_i, \end{aligned} \quad (6)$$

gde Σ_x predstavlja kovarijansnu matricu ulaznog skupa podataka, i gde se uočava da je ta greška u vezi sa kovarijansnom matricom. Ovde je primenjena činjenica da su \mathbf{u}_i vektori baza na koje se operator očekivanja ne odnosi pa se mogu izdvojiti ispred zagrada.

Kako bi se izračunali vektori baza radi se minimizacija izraza za grešku (izraz 6) pod ograničenjem da vektori baze moraju da budu ortonormalni, a ovo ograničenje se uvodi korišćenjem Lagranžovih multiplikatora λ_i [5]:

$$\varepsilon(M) = \sum_{i=M+1}^d \mathbf{u}_i^T \Sigma_x \mathbf{u}_i + \sum_{i=M+1}^d \lambda_i (1 - \mathbf{u}_i^T \mathbf{u}_i), \quad (7)$$

gde se rade parcijalni izvodi po svakom od baznih vektora nakon čega se dobijene jednačine izjednačavaju sa 0. Ovde se utvrđuje da bazni vektori zapravo predstavljaju karakteristične λ_i vektore kovarijanske matrice Σ_x :

$$\frac{\partial}{\partial \mathbf{u}_i} \varepsilon(M) = \frac{\partial}{\partial \mathbf{u}_i} \left[\sum_{i=M+1}^d \mathbf{u}_i^T \Sigma_x \mathbf{u}_i + \sum_{i=M+1}^d \lambda_i (1 - \mathbf{u}_i^T \mathbf{u}_i) \right] = 2(\Sigma_x \mathbf{u}_i - \lambda_i \mathbf{u}_i) = 0 \quad (8)$$

Dalje se pokazuje da je greška jednaka sumi karakterističnih vrednosti onih vektora koji se izbacuju prilikom aproksimacije [5]:

$$\varepsilon(M) = \sum_{i=M+1}^d \mathbf{u}_i^T \Sigma_x \mathbf{u}_i = \sum_{i=M+1}^d \mathbf{u}_i^T \boldsymbol{\lambda}_i \mathbf{u}_i = \sum_{i=M+1}^d \lambda_i \quad (9)$$

Transformacijom u prostor \mathbf{u}_i baznih vektora dobija se novi način da se predstave uzorci. Ovim tehnikama originalna matrica se rekonstruiše u potpunosti ukoliko je $M = d$, tj. dobija se vektor \mathbf{x} a ne njegova aproksimacija, ali kako je cilj PCA da smanji dimenzionalnost bira se $M < d$ koje će dovoljno dobro da aproksimira matricu. Parametar M se pronalazi kroz uslov:

$$\frac{\frac{1}{N} \sum_{i=1}^N \|\mathbf{x}^{(i)} - \hat{\mathbf{x}}_{(M)}^{(i)}\|^2}{\frac{1}{N} \sum_{i=1}^N \|\mathbf{x}^{(i)}\|^2} \leq l, \quad (10)$$

gde je $\hat{\mathbf{x}}_{(M)}^{(i)}$ aproksimacija $\mathbf{x}^{(i)}$ na osnovu PCA i M novih obeležja, a l predstavlja maksimalno odstupanje. Kriterijum izbora parametra l se može zapisati i na sledeći način ako se greška predstavi kao suma karakterističnih vredosti onih karakterističnih vektora koji će se odbaciti:

$$\frac{\sum_{i=1}^m \lambda_i}{\sum_{i=1}^d \lambda_i} \geq 1 - l \quad (11)$$

pri čemu $1 - l$ predstavlja procenat varijanse koji je potrebno zadržati, koji je u ovom radu postavljen na 90%.

2.3 Algoritmi mašinskog učenja

2.3.1 K najbližih suseda

K najbližih suseda (eng. *k-nearest neighbors* - *KNN*) je neparametarska metoda nadgledanog učenja čiji je cilj da na osnovu trening skupa podataka i labela klasa K najbližih uzorka odredi labelu klase novog uzorka. Ona pretpostavlja da su uzorci predstavljeni u prostoru u vidu tačaka i da se oni uzorci koji pripadaju istim klasama nalaze na međusobno manjim rastojanjima. Ova metoda zavisi od trening skupa, metode za određivanje udaljenosti između uzoraka i vrednosti parametra k .

Susedstvo uzoraka određuje se na osnovu izabrane metričke funkcije. Neka je uzorački prostor označen sa χ , a sa $\rho, \rho: \chi \times \chi \rightarrow \mathbb{R}$ metrička funkcija čija je povratna vrednost rastojanje između bilo koja dva uzorka. Neka je $S = \{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ skup podataka za obuku. Za svako $x \in \chi$ neka je $\pi_1(x), \dots, \pi_N(x)$ preuređen niz $\{1, \dots, N\}$ uzoraka u zavisnosti od njihovih udaljenosti do x , $\rho(x, x_i)$, tako da za svako $i < m$ važi [6]:

$$\rho(x, x^{\pi_i(x)}) \leq \rho(x, x^{\pi_{i+1}(x)}) \quad (12)$$

K najbližih suseda se može primeniti i na regresionim i klasifikacionim problemima i zbog toga izbor metrike zavisi od tipa podataka koji se nalaze u uzoračkom skupu. Euklidska metrika predstavlja jednu od metrika koja se koristi kod regresionih problema kada je $\chi = \mathbb{R}^d$:

$$\rho(x, x') = \|x - x'\| = \sqrt{\sum_{i=1}^d (x_i - x'_i)^2} \quad (13)$$

Izborom male vrednosti za parametar k metoda je sklona šumu, dok se za velike vrednosti gubi lokalni karakter procene, zato što postoji mogućnost da u najbliže uzorke upadnu oni koji su veoma udaljeni. Odabirom velikog k povećava se već velika računaska složenost. Kod klasifikacionih problema za određivanje izlazne vrednosti uzorka posmatra se klasa sa najvećom učestalošću između njemu k najbližih, a kod regresionih problema ona se određuje na osnovu aritmetičke sredine k najbližih suseda:

$$h_S(x) = \frac{1}{k} \sum_{i=1}^k y_{\pi_i}(x) \quad (14)$$

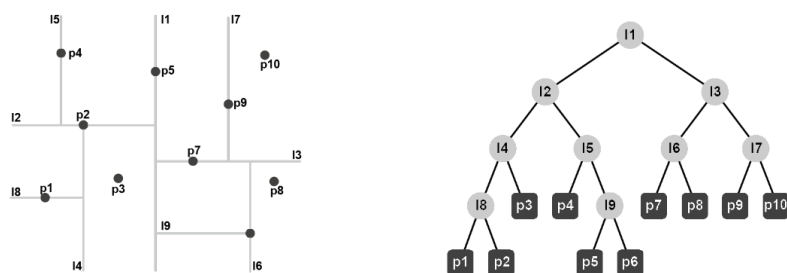
Jedno unapređenje algoritma dobija se ponderisanjem doprinosa svakog od K suseda u zavisnosti od njihove udaljenosti od posmatranog uzorka. Dodeljuju se težinski koeficijenti, pri čemu njihova vrednost raste što je sused bliži uzorku, a opada povećanjem njihovog međusobnog rastojanja. Vrednost težinskog koeficijenta proporcionalna je obrnutoj udaljenosti od posmatranog uzorka. Parametar algoritma *weights* koji ovo reguliše ima vrednost *distance*, a ukoliko nije izvršeno ponderisanje težinskih koeficijenti vrednost *uniform*, pri čemu koeficijenti imaju uniformnu vrednost.

Ova metoda pripada grupi metoda kasnog učenja (eng. *lazy learning*). Ovo znači da se model ne obučava unapred, već se za svaki novi uzorak za koji je potrebno izvršiti klasifikaciju, obavljaju izračunavanja udaljenosti do svakog od uzorka iz trening skupa. Za svaki novi uzorak potrebno je mnogo računati i potreban je veliki memorijski prostor za čuvanje celog trening skupa.

Kvalitet procene zavisi od broja dimenzija problema. Povećanjem dimenzionalnosti prostora eksponencijalno raste potreban broj uzoraka kako prostor ne bi bio razređen. Što je ovo povećanje veće to će se teže kvalitetno proceniti gustina raspodele verovatnoće. Ovaj problem naziva se prokletstvom dimenzionalnosti (eng. *curse of dimensionality*) [7].

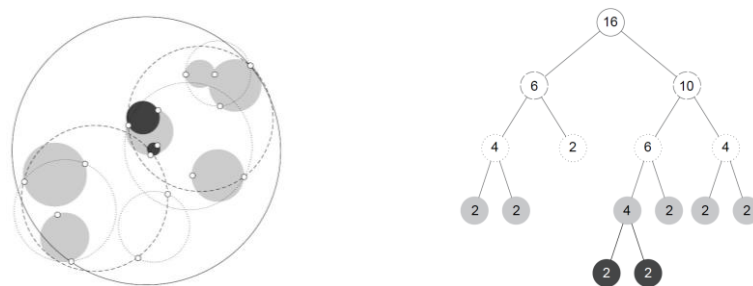
Za obuku algoritma K najbližih suseda potrebno je $O(n)$ vremena da se uskladišti n uzoraka iz trening skupa. Iako je ovo vreme brzo, mana algoritma je činjenica da klasifikacija jednog uzorka takođe zahteva $O(n)$ vremena, zato što je potrebno ponovo uporediti ovaj uzorak sa svim drugim uzorcima u trening skupu kako bi se utvrdili njemu najbliži. Vreme izvršavanja klasifikacije moguće je smanjiti čuvanjem trening skupa u složenim strukturama. Postoje dve strukture koje se primenjuju kod ovog algoritma: KD stabla i metričko stablo [7].

Prvi metod skladištenja trening skupa je KD stablo (eng. *kd tree*). To je binarno stablo koje uzorke razdvaja pomoću hiperravni (ravan u višedimenzionalnom prostoru) i rekursivno deli prostor na manje podoblasti. Naziva se KD stablo zato što skladišti skup tačaka u k -dimenzionom prostoru, gde k predstavlja broj obeležja. Algoritam započinje identifikovanjem jedne izabrane dimenzije svih uzoraka, nakon čega se izračunava medijana svih uzoraka po toj dimenziji i zatim se u toj tački povlači hiperravan koja prostor deli na dva dela (Slika 3). Postupak se dalje ponavlja po narednim dimenzijama. Prag za završetak algoritma je kada broj tačaka u nekoj od hiperravni padne ispod unapred zadatog broja.



Slika 3. Algoritam KD stabla

Drugi metod je metričko stablo (eng. *ball tree*) koji predstavlja unapređenje prethodnog algoritma. On predlaže da se umesto hiperravni za podelu prostora koriste hipersfere, kako se u pretrazi najbližih suseda ne bi zalazilo u više kvadranta ukoliko hipersfera preseca njihove ćoškovke. Algoritam započinje odabiranjem proizvoljne tačke, nakog čega se traži njoj najudaljenija tačka. Ove dve tačke se spoje i oko njih se crta hipersfera. Zatim se pronalazi centar i traže se dve najudaljenije tačke od njega iz čega proilaze dve nove hipersfere. Ovaj postupak se ponavlja rekursivno (Slika 4) [8].



Slika 4. Algoritam metričkog stabla [8]

U cilju obuke klasifikatora evaluirane su vrednosti od 1 do 10 za broj suseda k , dok su za računanje rastojanja razmotrene vrednosti *euclidean*, *manhattan*, *chebyshev* i *minkowski*, pri čemu je za *minkowski* metriku uzet treći stepen. Za težinske koeficijente razmatrane su vrednosti *uniform* i *distance*. U cilju ubrzanja izvršavanja algoritma odabrana je vrednost 20 za parametar n_jobs , koja predstavlja broj paralelnih poslova koji će se istovremeno obavljati prilikom pretrage najbližih suseda. Kao dodatno ubrzanje, za algoritme izračunavanja najbližih suseda razmotreni su *ball_tree* i *kd_tree*. Klasifikacija je izvršena sa i bez korišćenja PCA metode.

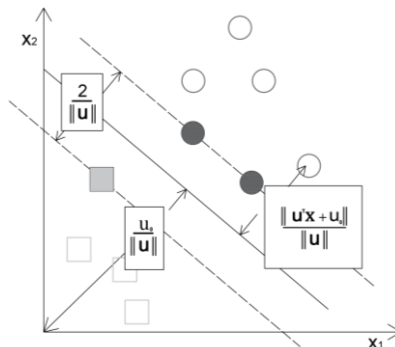
2.3.2 Mašine na bazi vektora nosača

Mašina na bazi vektora nosača (eng. *support vector machine* - *SVM*) predstavlja algoritam nadgledanog učenja koji probleme klasifikacije rešava identifikacijom hiperravan koja razdvaja prostor obeležja tako da se u odnosu na nju uzorci različitih klasa nalaze na najvećoj udaljenosti. Ovakva hiperravan sa maksimalno mogućim rastojanjem od uzoraka iz svake od klasa, trebalo bi da obezbedi dobru generalizaciju. Pored linearno razdvojivih problema koji se rešavaju klasifikatorom maksimalne margine, SVM predstavlja njegovo unapređenje koje uspešno rešava nelinearne probleme preslikavajući uzorke u višedimenzionalni prostor korišćenjem pomoćnih funkcija.

Neka je $S = \{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ trening skup podataka, gde $y^{(i)} \in \{-1, 1\}$ u zavisnosti od klase kojoj i -ti uzorak pripada pri binarnoj klasifikaciji. Hiperravan koja deli hiperprostor na dva dela se definiše jednačinom:

$$u_0 + \mathbf{u}^T \mathbf{x} = 0, \quad (15)$$

gde je u_0 slobodan član koji, ako ima vrednost 0, označava da hiperravan prolazi kroz koordinantni početak. Vektor \mathbf{u} predstavlja vektor normalan na hiperravan razdvajanja, a koji je definiše zajedno sa slobodnim članom u_0 , a sa \mathbf{x} se označava proizvoljna tačka u prostoru. Ako je za neki uzorak \mathbf{x} zadovoljena nejednakost $u_0 + \mathbf{u}^T \mathbf{x} < 0$ on pripada jednoj klasi, a ako važi da je $u_0 + \mathbf{u}^T \mathbf{x} > 0$ on pripada drugoj klasi.



Slika 5. Karakteristična rastojanja hiperravan kod klasifikatora maksimalne margine

Optimalna hiperravan u hiperprostoru je ona za koju su uzorci različitih klasa na najvećem međusobnom rastojanju, tj. ona koja formira najveću marginu. Klasifikator koji vrši klasifikaciju na

osnovu ove margine se zove klasifikator maksimalne margine (eng. *maximum margin classifier*). Postoji beskonačno mnogo ovakih hiperravan, a od njih se bira kanonička hiperravan koja zadovoljava uslov da za uzorke najbliže njoj važi $|u_0 + \mathbf{u}^T \mathbf{x}| = 1$. Rastojanje između tačke i hiperravan (Slika 5) se računa preko formule:

$$r = \frac{|u_0 + \mathbf{u}^T \mathbf{x}|}{\|\mathbf{u}\|}, \quad (16)$$

gde je $\|\mathbf{u}\| = \sqrt{\mathbf{u}^T \mathbf{u}}$, pri čemu je širina margine korišćenjem kanoničke hiperravan $r = \frac{2}{\|\mathbf{u}\|}$. Najveća razdvojitost između dve klase se obezbeđuje maksimizacijom izraza za r , pri čemu je potrebno minimizovati delilac $\|\mathbf{u}\|$. Funkcija cene je koja se minimizuje je $J(\mathbf{u}) = \frac{1}{2} \|\mathbf{u}\|^2$, uz uslov da je [9]:

$$y^{(i)}(u_0 + \mathbf{u}^T \mathbf{x}^{(i)}) \geq 1, \quad i = 1, \dots, N. \quad (17)$$

Prethodni izraz predstavlja kanoničku reprezentaciju hiperravan koja razdvaja hiperprostor obeležja na regione odlučivanja. Za one uzorke za koje je zadovoljena jednakost, tj. za one koji se nalaze na samoj margini, kaže se da su ograničenja aktivna (eng. *active*), dok su za ostale ograničenja neaktivna (eng. *inactive*) i uzorci se nalaze van margine. Uvek postoji barem jedna tačka za koju je ograničenje aktivno, zato što po definiciji uvek mora postojati jedna najbliža tačka hiperravan, dok nakon maksimizacije širine margine postoje barem dve tačke sa aktivnim ograničenjima [10].

Optimizacioni problem funkcije cene se rešava korišćenjem Lagranžove optimizacione funkcije pri čemu se funkcija cene zapisuje u obliku:

$$L_p(\mathbf{u}, u_0, \lambda) = \frac{1}{2} \|\mathbf{u}\|^2 - \sum_{i=1}^N \lambda_i (y^{(i)}(\mathbf{u}^T \mathbf{x}^{(i)} + u_0) - 1), \quad (18)$$

gde su $\lambda_i \geq 0$ Lagranžovi multiplikatori. Diferenciranjem prethodne jednačine po vektoru \mathbf{u} i slobodnom članu u_0 , nakon čega se dobijeni izrazi izjednačavaju sa 0, dolazi se do uslova da je $\mathbf{u} = \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i$ i $\sum_{i=1}^N \lambda_i y_i = 0$. Kada se ove dve jednačine uvrste u jednačinu za L_p (izraz 18) formira se Lagranžov dualni problem:

$$L_p(\mathbf{u}, u_0, \lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \quad \lambda_i \geq 0 \quad i \quad \sum_{i=1}^N \lambda_i y_i = 0, \quad (19)$$

Ovim izrazom se trenutni problem svodi na problem koji zavisi samo od multiplikatora λ_i , a ne više od \mathbf{u} i u_0 i ujedno više ne zavisi od pojedinačnih uzoraka \mathbf{x} već od skalarnog proizvoda svih uzoraka. Ovo predstavlja poboljšanje zato što problem više ne raste povećanjem dimenzionalnosti prostora i jednačine od \mathbf{u} . Ograničenje Lagranžove jednakosti obezbeđuje da je za svaki uzorak zadovoljena jednakost:

$$\lambda_i (y^{(i)}(\mathbf{u}^T \mathbf{x}^{(i)} + u_0) - 1) = 0, \quad \forall i = 1, \dots, N, \quad (20)$$

Gde, kada je λ_i različito od 0 ograničenje je aktivno i uzorak se zove vektor nosač (eng. *support vector*), a kada je jednako 0 ograničenje nije aktivno.

Ograničenje izraza (20) onemogućava da uzorak uđe u prostor margine, ili pređe hiperravan razdvajanja, čime dolazi do velike osetljivosti položaja hiperravan na položaj pojedinačnih uzoraka iz trening skupa, kao i smanjenja generalizacije. Ovo se posebno ogleda kod slučajeva gde uzorci nisu linearno razdvojivi. Zbog ovoga je potrebno dozvoliti da neki uzorci budu pogrešno klasifikovani ili da uđu u prostor margine. Ovakav klasifikator naziva se klasifikator na bazi vektora nosača (eng. *soft Margin Classifier*).

Svakom uzorku iz trening skupa se dodeljuje pomoćna promenljiva tolerancije ε_i (eng. *slack variable*) koja uzorcima daje slobodu da se nađu unutar margine ili sa suprotne strane hiperravan. Ako je prag tolerancije prevelik, veliki broj uzoraka će biti pogrešno klasifikovano ili će se naći unutar margine, a ako je premali, ovo će biti dozvoljeno malom broju uzoraka. Za uzorke koji su ušli u marginu ali su i dalje sa dobre strane hiperravan odlučivanja tolerancija ima vrednosti $0 \leq \varepsilon_i \leq 1$, za one koji su se našli sa pogrešne strane hiperravan odlučivanja $\varepsilon_i > 1$, a za one koji nisu prešli marginu ima vrednost 0. Kako bi se obezbedio njihov ulazak u marginu potrebno je da je zadovoljeno $|u_0 + \mathbf{u}^T \mathbf{x}| \geq 1 - \varepsilon_i$, gde, ako je ε_i veće od 1 dolazi do promene znaka koji postaje negativan i time označava prelazak margine. Cilj je da se margina maksimizuje dok se tačke koje se nalaze sa pogrešne strane margine u maloj meri penalizuju, tako da je nova funkcija cene koja treba da se minimizuje [10]:

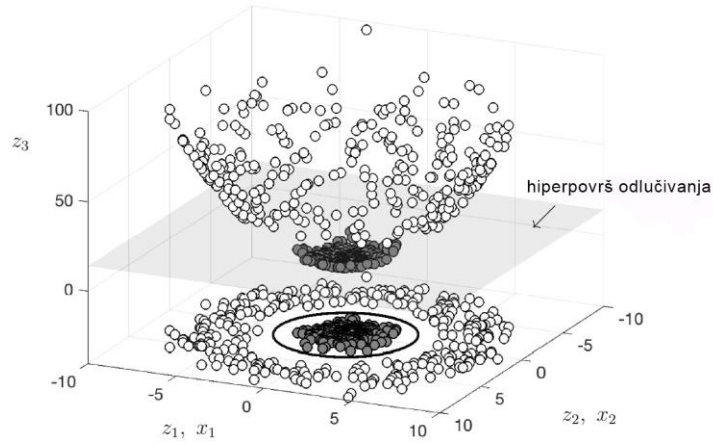
$$J(\mathbf{u}) = \frac{1}{2} \|\mathbf{u}\|^2 + C \sum_{i=1}^N \varepsilon_i, \quad (21)$$

gde je C faktor regularizacije koji predstavlja težinu koja se daje drugom sabirku. Ako je C veliko, drugi sabirak će preovladati, što znači da će se smanjiti broj uzoraka koji mogu da budu pogrešno klasifikovani i time će margina postati uska. Ako se C smanji, povećaćće se broj ovih uzoraka i margina će biti šira.

Ukoliko trening skup nije linearno razdvojiv nije moguće pronaći hiperravan koja uspešno razdvaja prostor. Ovaj problem se rešava korišćenjem novog skupa obeležja. Nerazdvojivi uzorci se posmatraju u novom prostoru gde se traži transformacija koja, kada se izvrši, obezbeđuje da uzorci u višedimenzionalnom prostoru postanu linearno razdvojivi. Jednačina Lagranžovog dualnog problema ukazuje na to da nisu svi uzorci od podjednakog značaja za pronalaženje rešenja, već samo oni koji se nalaze na margini ili unutar meke margine, tj. oni koji imaju aktivno ograničenje ($\lambda_i > 0$). Formula za određivanje klase uzorka \mathbf{x} je:

$$f(\mathbf{x}) = u_0 + \mathbf{u}^T \mathbf{x} = \sum_{i \in S} \lambda_i y_i \mathbf{x}_i^T \mathbf{x} + u_0, \quad \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i = \mathbf{u}. \quad (22)$$

Uočava se da u proizvodu $\mathbf{u}^T \mathbf{x}$ učestvuju skalarni proizvod uzorka koji se trenutno klasifikuje i svih drugih uzoraka koji su vektori nosači i koji pripadaju skupu S . Kako bi se skalarni proizvod izračunao koristi se trik (eng. *kernel trick*) koji podrazumeva računanje ovog proizvoda korišćenjem kernel funkcije $K(\mathbf{x}, \mathbf{x}_i)$, bez potrebe za eksplicitnim prelaskom u novi višedimenzionalni prostor u kojem su uzorci potencijalno linearno razdvojivi (Slika 6). Prema tome, nije potrebno razmatrati uvođenje dodatnih transformacija originalnog prostora, već se originalni vektori prosleđuju u ove funkcije, a kao rezultat se vraća skalarni proizvod u novom višedimenzionalnom prostoru.



Slika 6. Preslikavanje uzoraka u višedimenzionalni linearno razdvojni prostor [11]

Nova funkcija za određivanje klase uzorka je data u sledećem obliku:

$$f(\mathbf{x}) = u_0 + \sum_{i \in S} \lambda_i y_i K(\mathbf{x}, \mathbf{x}_i), \quad K(\mathbf{x}, \mathbf{x}_i) = (h(\mathbf{x}), h(\mathbf{x}_i)) \quad (23)$$

Postoje različite kernel funkcije pri čemu je, ukoliko je u pitanju polinomijalni kernel (eng. *polynomial kernel*) stepena $m > 2$, formula kernel funkcije:

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \left(1 + \sum_{k=1}^d x_k^{(i)} x_k^{(j)} \right)^m, \quad (24)$$

a ako se posmatra radijani kernel (eng. *radial basis function* - RBF):

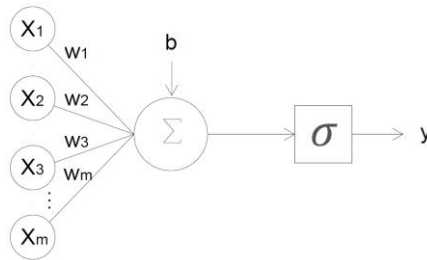
$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = e^{-\frac{\sum_{k=1}^d (x_k^{(i)} - x_k^{(j)})^2}{2\sigma^2}}. \quad (25)$$

Ukoliko je problem koji je potrebno rešiti višeklasan, tj. postoje više od dve klase ($K > 2$), za njihovo rešavanje moguća su dva različita pristupa. Prvi pristup je korišćenje jedna protiv svih (eng. *one-versus-rest, ovr*) metode kojom se obučava K SVM klasifikatora f_i , pri čemu se trening uzorci dele na one uzorke koji pripadaju klasi K_i , za koje je labela klase $y_i = +1$, i sve one uzorke koji ne pripadaju klasi K_i za koje je $y_i = -1$. U fazi odlučivanja, uzorku se dodeljuje klasa za koju je izraz $\mathbf{u}_i^T \mathbf{x}$ imao najveću vrednost, pri čemu je ukupno vreme klasifikacije ovom metodom $O(Kn^2)$. Druga metoda je svaki protiv svakog (eng. *one-versus-one, ovo*) kojom se konstruiše $\binom{K}{2}$ klasifikatora, pod uslovom da K nije veliki broj. Svaki od ovih klasifikatora radi klasifikaciju u određen par klasi K_i i K_j , gde uzorci koji pripadaju klasi i imaju labelu klase $+1$, a oni koji pripadaju klasi j -1 . Konačna klasa uzorka je ona klasa kojoj je uzorak bio dodeljen najviše puta, a vreme izvršavanja klasifikacije ovom metodom je $O(n^2)$ [12].

U cilju obuke klasifikatora evaluirane su vrednosti od 0.1 do 20 za regularizacioni parametar C , dok su kao kernel funkcije razmotrene *linear*, *poly*, *rbf* i *sigmoid*, pri čemu su uzete vrednosti *scale* i *auto* za *gamma*, koje predstavljaju koeficijent korišćen kod kernel funkcija. Za *scale* prosleđuje se vrednost $\frac{1}{n\sigma^2}$, a za *auto* vrednost $\frac{1}{n}$, gde je n broj klasi. Za rešavanje klasifikacije sa više od dve klase posmatrani su *ovr* i *ovo*. Klasifikacija je izvršena sa i bez korišćenja PCA metode.

2.3.3 Višeslojni perceptron

Višeslojni perceptron (eng. *multilayer perceptron* - *MLP*) predstavlja klasu potpuno povezanih slojeva veštačkih neurona (eng. *fully connected layers*) veštačke neuronske mreže. Slojevi su povezani u vidu usmerenog grafa (eng. *feedforward network*) gde se signal kreće jednosmerno kroz neurone od ulaznog do izlaznog sloja. Treniranje predstavlja učenje vrednosti koeficijenata grana mreže korišćenjem algoritma propagacije unazad (eng. *back propagation*). Neuronska mreža funkcioniše po analogiji sa neuronskim mrežama ljudskog mozga.



Slika 7. Struktura neurona

Veštački neuron predstavlja osnovu neuronske mreže. U najranijim istraživanjima, veštački neuron perceptron ima ulaze u obliku vektora x , a njegov izlaz y ima jednu binarnu vrednost. Svaki ulaz x_i se množi sa odgovarajućim realnim težinskim koeficijentom w_i , koji predstavlja značajnost datog ulaznog obeležja na izlaz. Vrednost izlaza neurona, koja može biti 0 ili 1, određena je poređenjem $\sum x_i w_i$ i odabranog realnog praga p (eng. *threshold value*). Ova suma može se zapisati i u obliku vektorskog proizvoda $\mathbf{w} \cdot \mathbf{x}$, a uvodi se i pristrasnost $b = -p$ (eng. *bias term*). Primenom ovih zamena klasifikacija se vrši na osnovu sledećeg izraza:

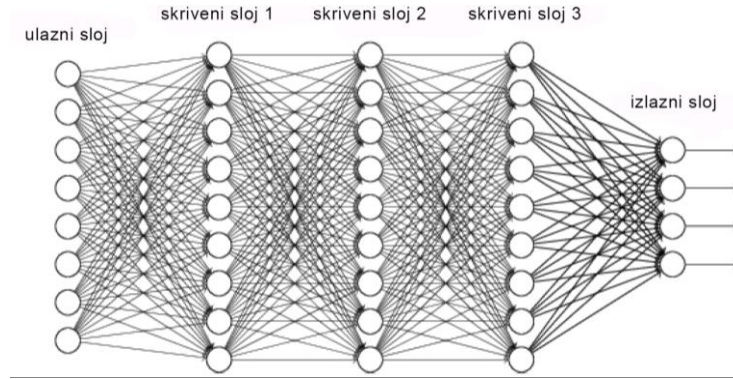
$$y = \begin{cases} 0, & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ 1, & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases} \quad (26)$$

Problem sa perceptronom nastaje ukoliko u neuronskoj mreži dođe do male promene težinskih koeficijenata ili pristrasnosti b samo jednog perceptrona, pri čemu se izlaz mreže može u potpunosti promeniti. Ovo se rešava novom nadograđenom strukturom perceptrona koja se naziva sigmoidni neuron. Za razliku od perceptrona čije izlazne vrednosti mogu biti samo 0 ili 1, sigmoidni neuron za izlaz može uzimati sve realne vrednosti od 0 do 1. Korišćenjem ove vrste neurona izlaz mreže je ujedno i manje osetljiv na malu promenu težinskih koeficijenata. Na prvobitni izlaz neurona dodaje se aktivaciona funkcija σ , koja je u ovom slučaju sigmoidna, pri čemu je izlaz je onda definisan izrazom $\sigma(\mathbf{w} \cdot \mathbf{x} + b)$ gde je [13]:

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \quad (27)$$

a promenjiva $z = \mathbf{w} \cdot \mathbf{x} + b$. Pored ove aktivacione funkcije mogu se koristiti i ispravljačka linearna jedinica (eng. *rectified linear unit* – *ReLU*) i tangens hiperbolička funkcija (*TanH*). Glavnu razliku između sigmoidnog i tangens hiperboličkog neurona predstavlja opseg izlazne vrednosti, koja je kod sigmoidnog između 0 i 1, a kod tangens hiperboličkog neurona između -1 i 1, što znači da se normalizacija izlaza mora raditi na drugačiji način [13].

Višeslojni perceptron predstavlja više spojenih slojeva veštačkih neurona. On se sastoji iz ulaznog sloja koji sadrži ulazne neurone koji imaju vrednosti obeležja barem jednog skrivenog sloja i izlaznog sloja (Slika 8), a gde se skriveni slojevi tako nazivaju zato što izlazi njihovih neurona nisu vidljivi. Na samom izlazu neuronske mreže se primenjuje *softmax* funkcija koja će dodeliti verovatnoće ishoda svakom izlazu. Slojevi su potpuno povezani ukoliko ulaz u jedan neuron obuhvata sve izlaze neurona iz prethodnog sloja.



Slika 8. Arhitektura neuronske mreže [13]

Postoje različite arhitekture neuronskih mreža, gde pored nerekurzivnih mreža postoje i rekurzivne neuronske mreže (eng. *recurrent neural networks* - *RNN*) i konvolucione neuronske mreže (eng. *convolutional neural networks* - *CNN*). Kod prve vrste mreža na aktivaciju neurona, pored aktivacija prethodnog sloja, mogu uticati i aktivacije u prethodnim vremenskim trenucima, dok se kod druge vrste primenjuju dodatni slojevi koji će biti opisani u narednom poglavlju [13].

Neka je neuronska mreža sačinjena od L slojeva i neka se u r -tom sloju nalazi k_r neurona. Skup za obuku je sačinjen od N parova vektora $(\mathbf{x}^i, \mathbf{y}^i)$, gde $\mathbf{x}^i = [x_1^i, x_2^i, \dots, x_{k_0}^i]^T$ predstavlja vrednosti obeležja i -tog uzorka postavljenog na ulaz, a $\mathbf{y}^i = [y_1^i, y_2^i, \dots, y_{k_L}^i]^T$ predstavlja stvarnu klasu tog uzorka napisanu u obliku gde je za klasu kojoj pripada vrednost $y_j^i = 1$, a za sve ostale vrednost 0. Prilikom obuke predviđen izlaz $\hat{\mathbf{y}}^i$ treba da u što manjoj meri odstupa od stvarne labele uzorka, što se postiže minimizacijom funkcije cene odstupanja. Kao najčešće funkcije cene koriste se međuentropijska funkcija cene za klasifikacione probleme i srednje kvadratna greška (eng. *mean square error*) za regresione probleme:

$$J = \frac{1}{2} \sum_{i=1}^N \sum_{m=1}^{k_L} (\hat{y}_m^i - y_m^i)^2, \quad (28)$$

gde unutrašnja suma prolazi kroz sve neurone na izlazu jednog uzorka, a spoljašnja kroz sve uzorke trening skupa. Na osnovu vrednosti ove funkcije cene vrši se određivanje težinskih koeficijenata. Prilikom obuke potrebno je za svaki sloj naučiti vrednosti $\mathbf{w}_j^r = [w_{j0}^r, w_{j1}^r, \dots, w_{jk_{r-1}}^r]^T$ koje predstavljaju vektore težinskih koeficijenata j -tog neurona u r -tom sloju. Ove vrednosti se inicijalno postavljaju kao male proizvoljne vrednosti u opsegu od 0 do 1, a zatim se iterativno menjaju po postupku:

$$\mathbf{w}_{j \text{ (novo)}}^r = \mathbf{w}_{j \text{ (staro)}}^r - \alpha \frac{\partial J}{\partial \mathbf{w}_j^r}, \quad (29)$$

gde se ovo rešava metodom opadanja gradijenta, kojom se smanjuje funkcija cene kako bi greška bila minimalna. Funkcija cene se računa za svaki od uzoraka iz trening skupa. Izvod funkcije cene računa se po formuli:

$$\frac{\partial J(w)(i)}{\partial \mathbf{w}_j^r(i)} = \frac{\partial J(w)(i)}{\partial z_j^r(i)} \cdot \frac{\partial J(w)(i)}{\partial \mathbf{w}_j^r(i)} = \frac{\partial J(w)(i)}{\partial z_j^r(i)} \cdot \mathbf{a}^{r-1}(i) = \delta_j^r(i) \cdot \mathbf{a}^{r-1}(i), \quad (30)$$

gde z predstavlja linearnu kombinaciju ulaza i odgovarajućih težinskih koeficijenata koji ulaze u j -ti neuron r -tog sloja, \mathbf{a} je vektor aktivacije prethodnog sloja, a δ je senzitivnost neurona koja ukazuje na promenu greške pri aktivaciji sloja.

Pri obuci neuronske mreže koriste se dva osnovna algoritma. Prvi je propagacija unapred (eng. *forward propagation*) koja predstavlja izračunavanje svih vrednosti od ulaza do izlaza mreže, tj. računanje vrednosti svakog z_j^r , \mathbf{a}_j^r za svaki od uzoraka $x(i)$ kao i vrednosti funkcije cene za trenutne težinske koeficijente. Drugi algoritam je propagacija unazad (eng. *back propagation*) kojim se izračunavaju parcijalni izvodi funkcije cene po svim težinskim koeficijentima i pristrasnostima od poslednjeg izlaznog sloja unazad ka ulaznom sloju, pri čemu postoji razlika u izračunavanju između poslednjeg skrivenog sloja i izlaznog sloja, i svih ostalih slojeva.

U slučaju gde se posmatraju težinski koeficijenti između poslednjeg skrivenog sloja neurona i izlaznog sloja (eng. *hidden-to-output*) izvod funkcije cene se računa po formuli:

$$\frac{\partial J(w)(i)}{\partial w_{kj}^L(i)} = \frac{\partial J(w)(i)}{\partial a_k^L(i)} \cdot \frac{\partial a_k^L(i)}{\partial z_k^L(i)} \cdot \frac{\partial z_k^L(i)}{\partial w_{kj}^L(i)}, \quad (31)$$

pri čemu je w_{kj}^L težinski koeficijent k -tog neurona, gde j predstavlja izlaze neurona prethodnog sloja koji ulaze u k -ti neuron, a L je poslednji skriveni sloj. Aktivaciona funkcija koju neuroni u mreži imaju je sigmoidna. Vrednosti činioca iz izraza (31) se mogu izvesti na sledeći način:

$$\frac{\partial J(w)(i)}{\partial a_k^L(i)} = \frac{\partial}{\partial a_k^L(i)} \left(\sum_{n=1}^{k_L} (a_n^L - y_n)^2 \right) = (a_k^L(i) - y_k(i)), \quad (32)$$

$$\frac{\partial a_k^L(i)}{\partial z_k^L(i)} = \frac{\partial}{\partial z_k^L(i)} \left(\frac{1}{1 + e^{-z_k^L}} \right) = (1 - a_k^L(i)) a_k^L(i), \quad (33)$$

$$\frac{\partial z_k^L(i)}{\partial w_{kj}^L(i)} = \frac{\partial}{\partial w_{kj}^L(i)} \left(\sum_{n=1}^{k_L} w_{kn}^L(i) a_n^{L-1}(i) \right) = a_j^{L-1}(i), \quad (34)$$

nakon čega se uvrštavanjem ova tri izraza u izraz (31) dobija sledeća jednačina:

$$\frac{\partial J(w)(i)}{\partial w_{kj}^L(i)} = (a_k^L(i) - y_k(i)) \cdot (1 - a_k^L(i)) a_k^L(i) \cdot a_j^{L-1}(i) = \delta_k^L(i) \cdot a_j^{L-1}(i) \quad (35)$$

Problem nastaje prilikom izračunavanja ovog izraza između ulaza i ostalih skrivenih slojeva neurona:

$$\frac{\partial J(w)(i)}{\partial w_{kj}^1(i)} = \frac{\partial J(w)(i)}{\partial a_k^1(i)} \cdot \frac{\partial a_k^1(i)}{\partial z_k^1(i)} \cdot \frac{\partial z_k^1(i)}{\partial w_{kj}^1(i)}. \quad (36)$$

U ovoj jednačini prvi činilac nije moguće izračunati zato što izlazi neurona u skrivenim slojevima nisu poznati. Konstatuje se da skriveni neuroni ne prave greške, već samo doprinose konačnim greškama na izlaznim slojevima i zbog toga, iako ih nije moguće izračunati, moguće je odrediti u kojoj meri doprinose narednim slojevima. Izrazi za drugi i treći činilac su analogno jednaki izrazima (33) i (34) u prethodnom slučaju pri čemu se u izrazu (34) $a_j^{l-1}(i)$ menja sa x_i . Izvod greške neurona jednak je sumi doprinosa izlaza tog neurona greškama ka svim k_L neuronima:

$$\frac{\partial J(w)(i)}{\partial w_{kj}^1(i)} = \left[\sum_{n=1}^{k_L} \frac{\partial J(w)(i)}{\partial a_k^2(i)} \cdot \frac{\partial a_k^2(i)}{\partial z_k^2(i)} \cdot \frac{\partial z_k^2(i)}{\partial w_{kj}^1(i)} \right] \cdot (1 - a_k^1(i)) a_k^1(i) x_i = \left[\sum_{n=1}^{k_L} \delta_n^2(i) \cdot w_{nj}^2(i) \right] \cdot (1 - a_k^1(i)) a_k^1(i) x_i. \quad (37)$$

U cilju minimizacije funkcije metodom opadanja gradijenta (eng. *gradient descent*), na ovaj način određuje se izvod funkcije cene po svakom parametru \mathbf{w} nakon čega će se ažurirati svi parametri. Na pronalaženje minimuma utiče brzina učenja koja može osigurati da se ne preskoči minimum.

Epoha predstavlja jedan od parametara čije se podešavanje može raditi prilikom obuke neuronske mreže i ona označava jedan prolazak svih uzoraka trening skupa kroz mrežu. Kao jedan kriterijum zaustavljanja obuke može se uzimati unapred zadat broj epoha koji treba da se izvrši, ali se takođe može posmatrati i minimalna promena funkcije cene koja je manja od nekog određenog praga tokom isto određenog broja uzastopnih epoha.

Kako bi mreža uspešno učila neophodno je da uči nove vrednosti težinskih koeficijenata kao i da ažurira njihove stare vrednosti. Postoje dva načina na koje se oni mogu ažurirati, od čega je jedan da se nakon prolaska svakog pojedinačnog uzorka kroz mrežu (uzorački režim rada) izvrši ažuriranje izraza (29), što će kao posledicu imati male promene konvergencija. Drugi način je ažuriranje parametara nakon prolaska određenog podskupa uzoraka kroz mrežu (grupni režim rada) koji je unapred zadate veličine (eng. *batch*).

Jedna od posledica potpuno povezanog karaktera neuronske mreže i velikog broja parametara koji se uče, jeste činjenica da je sklona natprilagođenju. Kako bi se ovo izbeglo postoje određene tehnike regularizacije koje se mogu primenjivati. Jedna od tehnika je metoda izostavljanja (eng. *dropout*) koja tokom obuke zanemaruje pojedine različite neurone tokom različitih epoha, primoravajući mrežu da uči nove konekcije između neurona, čime se sprečava stvaranje velikih zavisnosti između neurona različitih slojeva. Druga metoda je propadanje težina (eng. *weight decay*). Ukoliko se primeti da dolazi do natprilagođenja, što se ogleda porastom vrednosti funkcije cene na validacionom skupu uz njeno propratno smanjenje na trening skupu, može se primeniti metoda ranog zaustavljanja (eng. *early stopping*) koja zaustavlja obuku [14].

U cilju obuke klasifikatora evaluirane su vrednosti *logistic*, *tanh* i *relu* za aktivacione funkcije, a za optimizacione metode *lbfgs*, *sgd* i *adam*. Razmatrane su neuronske mreže sa 2 i 3 skrivena sloja pri čemu je uzeto da slojevi imaju kombinaciju od 128, 96, 64, 32 i 16 neurona po sloju, gde naredni sloj ima manje ili jednako neurona kao prethodni. Klasifikacija je izvršena sa I bez korišćenja PCA metode.

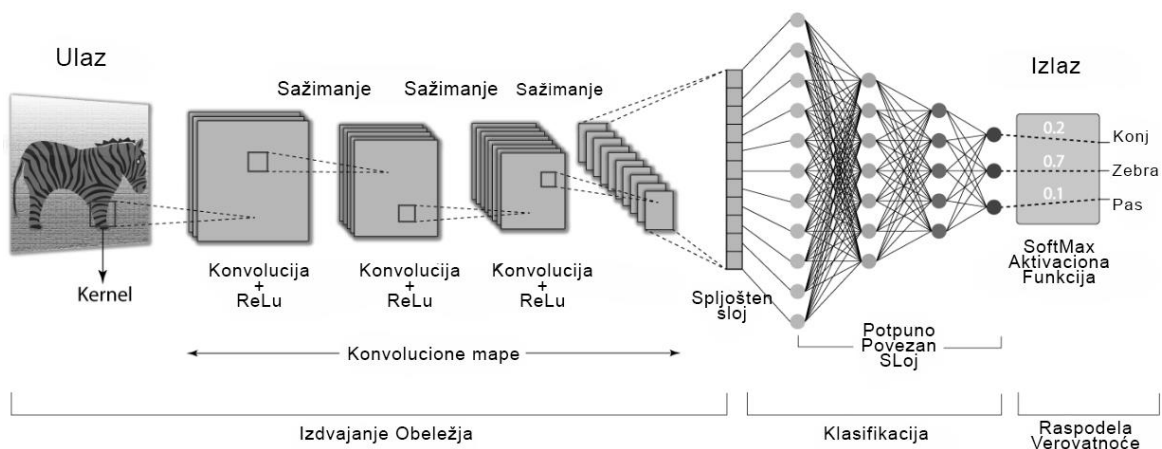
2.3.4 Konvoluciona neuronska mreža

Konvoluciona neuronska mreža (eng. *convolutional neural network* - *CNN*) predstavlja klasu nerekurzivne veštačke neuronske mreže koja se najčešće koristi za potrebe analize slika, koje se u reprezentaciji preko piksela dovode na ulaz mreže koja potom koristi operaciju konvolucije nad ulaznim podacima, i koja se svrstava u duboke neuronske mreže (eng. *deep neural networks*), odnosno ima veći broj slojeva neurona. Arhitektura ove mreže inspirisana je organizacijom vizuelnog korteksa. Pojedinačni neuroni u korteksu reaguju na stimulse u ograničenim regionima vidnog polja. Regije različitih neurona se delimično poklapaju tako da se pokrije celo vidno polje.

CNN najveću primenu pronalazi kod klasifikacije slika i prepoznavanja predmeta sa njih. Za razliku od potpuno povezanog višeslojnog perceptrona, čija arhitektura ne uzima u obzir strukturu slika, već piksele gleda zasebno u nepovezanom kontekstu, konvoluciona neuronska mreža pridodaje veliki značaj njihovom prostornom rasporedu. Pikseli koji su prostorno na većoj međusobnoj udaljenosti nemaju isti značaj kao pikseli koji su na manjem rastojanju i kod kojih postoji korelacija. Konvoluciona neuronska mreža pravi kompleksnije paterne u podacima, poput pronalaženja ivica predmeta sa slika korišćenjem manjih i jednostavnijih paterna utisnutih u njihovim filterima [15].

Pronalaženje paterna funkcioniše po principu iskorišćavanja lokalnih karakteristika (eng. *local features*) koje se mogu uočiti na manjim podregionima slika. Informacije koju nose ove pojedinačne karakteristike se u kasnijim fazama mogu spojiti, rezultat čega je moguća detekcija kompleksnijih karakteristika. Kod konvolucione mreže za ovo su zadužena tri mehanizma: lokalna receptivna polja (eng. *local receptive fields*), deljenje težina (eng. *weight sharing*) i poduzorkovanje (eng. *subsampling*) [16].

Umesto da svaki veštački neuron bude povezan sa svim neuronima u susednom polju oni se raspoređuju u 3D strukturu (širina \times visina \times dubina), pri čemu se svaki neuron povezuje sa određenim regionom. Povezivanje svih neurona sa svim mogućim regionima ulazne 3D strukture nije praktično jer može dovesti do prevelikog broja težinskih koeficijenata za treniranje. Pojam receptivnog polja se odnosi na deo slike koji je filteru vidljiv u datom trenutku.



Slika 9. Arhitektura konvolucione neuronske mreže [18]

Osnovnu arhitekturu konvolucione neuronske mreže čine ulazni sloj, skriveni slojevi i izlazni sloj. Kod ovog tipa mreže u skrivenom sloju se nalaze slojevi konvolucije (eng. *convolutional layers*), slojevi sažimanja (eng. *pooling layers*), slojevi aktivacionih funkcija (eng. *activation function*) kao i potpuno povezan sloj (eng. *fully connected layer*) (Slika 9).

Ulazni sloj dat je u vidu matrice dimenzije $N \times N$, čiji elementi predstavljaju numeričke vrednosti svakog piksela slike. Ukoliko je u pitanju slika u boji, ulaz u mrežu će obuhvatati tri matrice vrednosti, gde svakoj od njih, respektivno, odgovaraju vrednosti crvene, zelene i plave boje (eng. *red green blue - RGB*) koje se nalaze u opsegu 0-255. RGB slika se predstavlja 3D strukturom, koja se zove ulazna zapremina (eng. *input volume*). Ulazni sloj, zajedno sa konvolucionim slojevima i slojevima sažimanja, spada u deo za izdvajanje karakteristika (eng. *feature extraction*) [17].

Konvolucionni sloj je zadužen za detektovanje skrivenih paterna na slikama, otkrivanje značajnih karakteristika, ali i redukciju rezolucije slike koristeći operaciju konvolucije, po kojoj je i sam dobio ime. U ovoj operaciji učestvuju kao prvi operand matrica ulaza, a kao drugi matrica filtera (eng. *kernel*) koja je dimenzije $d \times d$, pri čemu je $d < N$. Rezultat ove operacije je predstavljen u vidu izlaza na mapu karakteristika (eng. *feature map*). Tokom obuke ovog algoritma mašinskog učenja mreža uči da optimizuje filtere koji predstavljaju matrice težinskih koeficijenata [17].

Kao rezultat operacije konvolucije nad ulaznom matricom može se dobiti mapa karakteristika čije su dimenzije manje od dimenzije ulazne matrice, ili mapa čije su dimenzije jednake dimenzijama ulazne. Ovo se postiže korišćenjem validne (eng. *valid padding*) ili podjednake dopune (eng. *same padding*), čija je glavna razlika način puštanja filtera da klizi preko ulazne matrice tokom konvolucije. Kod validne klizanje je ograničeno ivicama ulazne matrice, dok se kod podjednake dopune konvolucija započinje već po prvom preseku između matrica filtera i ulaza. Formula za izračunavanje vrednosti izlazne matrice primenom kod validne dopune je:

$$y_{m,n} = x_{m,n} * k_{m,n} = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x_{i,j} \cdot k_{m-i,n-j}, \quad (38)$$

gde m predstavlja redni broj vrste, n redni broj kolone, a promenljive x, k, y , respektivno, predstavljaju elemente ulazne matrice, filter matrice i matrice mape karakteristika. Oznaka $*$ je znak koji označava operaciju konvolucije. Iza odabranih konvolucionih slojeva u neuronskoj mreži se primenjuje sloj aktivacione funkcije, koji će primeniti neku od aktivacionih funkcija na izlaznu matricu prethodnog sloja.

Obuka mreže započinje algoritmom propagacije unapred. Neka ulazni sloj sadrži jednu ili više matrica ulaza X_i , u zavisnosti od n broja kanala boja koje slika sadži. Nakon ovog sloja u mreži se nalazi proizvoljan broj slojeva d u kojima se nalaze filteri K_{ij} koji imaju istu dubinu kao ulazni sloj. Ovi filteri predstavljaju težinske koeficijente. Svakom sloju filtera dodeljuje se po jedna matrica pristrasnosti B_i (eng. *bias matrix*) koja je istih dimenzija kao izlazna mapa. Izlazna matrica mape Y_i se izračunava po formuli:

$$Y_i = B_i + \sum_{j=1}^n X_i * K_{ij}, \quad i = 1, \dots, d \quad (39)$$

Sloj sažimanja ima ulogu smanjenja dimenzionalosti prethodno izračunatih mapa karakteristika pri čemu se najvažniji delovi čuvaju, a oni od manjeg značaja odbacuju. Ovaj sloj se primenjuje kako bi se u daljim slojevima ubrzala potrebna izračunavanja, koja su zahtevnija usled velike prostorne dimenzije slike. Postoji više tipova sažimanja od kojih je jedan od najčešćih maksimalno sažimanje (eng. *max pooling*) kod kog se novi filter pušta da klizi po originalnoj mapi karakteristika, dok se u regionima kroz koje prolazi pamti najveća vrednost koja se zatim čuva je u novoj izlaznoj mapi karakteristika. Slojevi konvolucije i sažimanja se primenjuju više puta kako bi omogućili otkrivanje značajnijih apstraktnih karakteristika visokog nivoa, koje su predstavljene u matricama što manjih dimenzija, a koje imaju mogućnost da dobro predstave ulaznu sliku pojedinačnog uzorka.

Drugi deo konvolucione neuronske mreže je zadužen za klasifikaciju. On se sastoji iz potpuno povezanih slojeva veštačkih neurona veoma sličnih višeslojnom perceptronu, pri čemu ulaz u ovaj deo nisu uzorci, već matrice apstraktnih karakteristika za svaki od uzoraka. Matrica karakteristika se zatim pretvara u niz koji će činiti ulaz koji se pušta kroz deo za klasifikaciju.

U cilju obuke klasifikatora evaluirane su različite arhitekture konvolucione neuronske mreže kombinovanjem i dodavanjem različitih slojeva. Za veličine kernela testirane su dimenzije 3x3 i 5x5, aktivacione funkcije *logistic*, *tanh* i *relu* i veličine filtera 64, 32 i 16 u konvolucionim slojevima. Kod potpuno povezanih slojeva proverene su vrednosti 128, 96 i 64 za broj neurona. Klasifikacija je izvršena sa i bez korišćenja PCA metode.

2.4 Način evaluacije rezultata

Početni skup podataka (baza podataka) je unapred podeljen na trening i test skup, tako da se u njima nalaze 60000 i 10000 uzoraka, respektivno. Usled obimnosti podataka kao i dužeg vremena izvršavanja algoritama mašinskog učenja, originalni trening skup podataka je dodatno smanjen na 30% njegove početne veličine pri čemu je sačuvana podjednaka zastupljenosti klasa sa po 1800 uzoraka svake klase. Na ovom smanjenom skupu određeni su optimalni parametri svakog od algoritama korišćenjem metode unakrsne validacije kroz 5 iteracija, gde 80% posto uzoraka čini podatke za treniranje, a 20% validacioni skup, čime se obezbeđuje preciznija procena. Nakon dobijenih optimalnih parametara, modeli su obučeni na celom originalnom trening skupu i testirani na originalnom test skupu. Ovaj postupak je zatim ponovljen, ali sa prethodnom primenom PCA redukcije dimenzionalnosti baze podataka.

Pre obuke modela, obeležja su standardizovana kako bi se svela na isti opseg, čime bi se sprečila mogućnost odlučivanja na osnovu manjeg broja obeležja koje variraju u većem opsegu. Ona je takođe neophodna pre izvršenja PCA algoritma kako bi se smanjila standardna devijacija obeležja. Obeležja se normalizuju tako da imaju srednju vrednost 0 i standardnu devijaciju 1 korišćenjem formule:

$$x^i = \frac{x_i - \mu_i}{\sigma_i}, \quad (40)$$

gde je x_i vektor vrednosti i -tog obeležja za sve uzorke iz trening skupa, μ_i srednja vrednost i -tog obeležja, a σ_i standardna devijacija i -tog obeležja.

Kao mere performansi klasifikatora korišćeni su matrica konfuzije, udeo ispravno pogođenih i vreme izvršavanje klasifikacije.

Svrha matrice konfuzije je da numerički predstavi uporedni prikaz usklađenosti stvarnih i predviđenih labela prilikom klasifikacije. Data je u vidu kvadratne matrice čije dimenzije odgovaraju broju klasa, gde pojedine vrste odgovaraju stvarnim vrednostima labela klasa, a pojedine kolone predviđenim vrednostima labela klasa, čime se obezbeđuje jasan uvid u meru preciznosti kao i pogrešne klasifikacije. Na glavnoj dijagonali nalaze se ispravno klasifikovani uzorci. Na osnovu matrice konfuzije računaju se tačnost, preciznost, osetljivost, specifičnost i F-mera. Svi oni predstavljaju mere uspešnosti klasifikatora.

Kako bi se ove mere izračunale neophodno je odrediti matricu konfuzije, odn. broj stvarno pozitivnih (eng. *true positive* - *TP*), stvarno negativnih (eng. *true negative* - *TN*), lažnih pozitivna (eng. *false positive* - *FP*) i lažnih negativa (eng. *false negative* - *FN*) među uzorcima nakon predikcije (Slika 10) Stvarni pozitiv predstavlja uzorak koji potiče iz pozitivne klase i klasifikator je ispravno predvideo da pripada pozitivnoj klasi, njegova stvarna i predviđena labela imaju vrednost 1. Stvarni negativ predstavlja uzorak koji potiče iz negative klase i klasifikator je ispravno predvideo da pripada negativnoj klasi, njegova stvarna i predviđena labela imaju vrednost 0. Lažni pozitiv predstavlja uzorak koji potiče iz negative klase, a klasifikator je predvideo da pripada pozitivnoj klasi. Njegova stvarna labela ima vrednost 0, a predviđena vrednost 1. Lažni negativ predstavlja uzorak koji potiče iz pozitivne klase, a

klasifikator je predvideo da pripada negativnoj klasi. Njegova stvarna labela ima vrednost 1, a predviđena vrednost 0. U tabeli 2. predstavljene su načini računanja mera uspešnosti klasifikatora za dve klase.

Mera	Formula
Tačnost	$\frac{TP + TN}{TP + TN + FP + FN}$
Preciznost	$\frac{TP}{TP + FP}$
Osetljivost	$\frac{TP}{TN + FP}$
Specifičnost	$\frac{TN}{TN + FP}$
F-mera	$2 \cdot \frac{\text{preciznost} \cdot \text{osetljivost}}{\text{preciznost} + \text{osetljivost}}$

Tabela 2. Mere uspešnosti za dve klase

		predviđene		
		A	B	C
prave	A	TP	FN	FN
	B	FP	TN	TN
	C	FP	TN	TN

		predviđene		
		A	B	C
prave	A	TN	FP	TN
	B	FN	TP	FN
	C	TN	FP	TN

		predviđene		
		A	B	C
prave	A	TN	TN	FP
	B	TN	TN	FP
	C	FN	FN	TP

Slika 10. Matrica konfuzije za tri klase na kojima su naznačeni TP, TN, FP i FN

Kod klasifikatora koji vrše klasifikaciju u više različitih klasa postoje dva načina za računanje mera performansi: mikroprosečne (eng. *micro-average*) i makroprosečne mere (eng. *macro-average*). Mikroprosečne mere obavljaju izračunavanja korišćenjem svih TP, FP i FN po klasama datog modela. U ovom radu, broj uzoraka svake klase je podjednako zastupljen tako da je korišćena makroprosečna mera koja određuje globalnu vrednost mere performansi na osnovu proseka vrednosti te mere za svaku od klasa. Kao glavno merilo performansi modela posmatrana je preciznost koja predstavlja prosečan udeo ispravno klasifikovanih uzoraka:

$$\frac{1}{K} \sum_{i=1}^K \frac{tp_i}{tp_i + fp_i}, \quad (41)$$

gde K predstavlja ukupan broj klasa. Kvalitet ispravno klasifikovanih uzoraka kao i odabir najboljeg modela za svaki od algoritama izvršen je na osnovu najvećih vrednosti prosečnog udela ispravno klasifikovanih uzoraka, uz posmatranje vremena izvršavanja i osetljivosti po klasi.

Algoritmi mašinskog učenja, standardizacija i mere performansi korišćeni su primenom *scikit-learn* biblioteke u programskom jeziku *python 3.9*, pri čemu je za obuku konvolucione neuronske mreže korišćena *tensorflow* biblioteka.

3. Rezultati modela

3.1 Rezultati modela na bazi K najbližih suseda

Odabrani model na bazi K najbližih suseda koji je obučavan na originalnom skupu obeležja, 30% trening skupa, koristi *manhattan* metriku, pri čemu uzima u obzir 4 najbliža suseda uzorku koji se klasifikuje, i koristi ponderisanje K najbližih suseda na osnovu rastojanja. Vreme izvršavanja dodatno ubrzavaju *ball_tree* algoritam za izračunavanje suseda i odabir od 20 paralelnih poslova koji se istovremeno obavljaju prilikom pretrage suseda. Na isti način izvršena je obuka i nad celim trening skupom.

Odabrani model na bazi K najbližih suseda koji je obučavan na skupu obeležja sa PCA smanjenjem dimenzionalnosti, koristi *manhattan* metriku, pri čemu uzima u obzir 6 najbliža suseda uzorku koji se klasifikuje, i koristi ponderisanje K najbližih suseda na osnovu rastojanja. Vreme izvršavanja dodatno ubrzavaju *ball_tree* algoritam za izračunavanje suseda i odabir od 20 paralelnih poslova koji se istovremeno obavljaju prilikom pretrage suseda.

U tabeli 3. predstavljeni su dobijeni rezultati, tj. prosečni udeli ispravno pogođenih klasa uzoraka na bazi K najbližih suseda nad originalnim skupom obeležja 30% uzoraka, originalnim skupom obeležja svih uzoraka i nad skupom obeležja primenom PCA redukcije. Vreme obuke nad 30% originalnog trening skupa je 1.51061 sekundi, nad celim trening skupom 5.74123 sekundi a nad skupom nad kojim je primenjeno PCA smanjenje dimenzionalnosti 1.09000 sekundi. Vreme klasifikacije jednog uzorka nakon obuke nad celim trening skupom sa PCA redukcijom je 0.02180 sekundi. Matrica konfuzije dobijena na test skupu korišćenjem modela obučenog na skupu obeležja sa PCA redukcijom je predstavljena u tabeli 4.

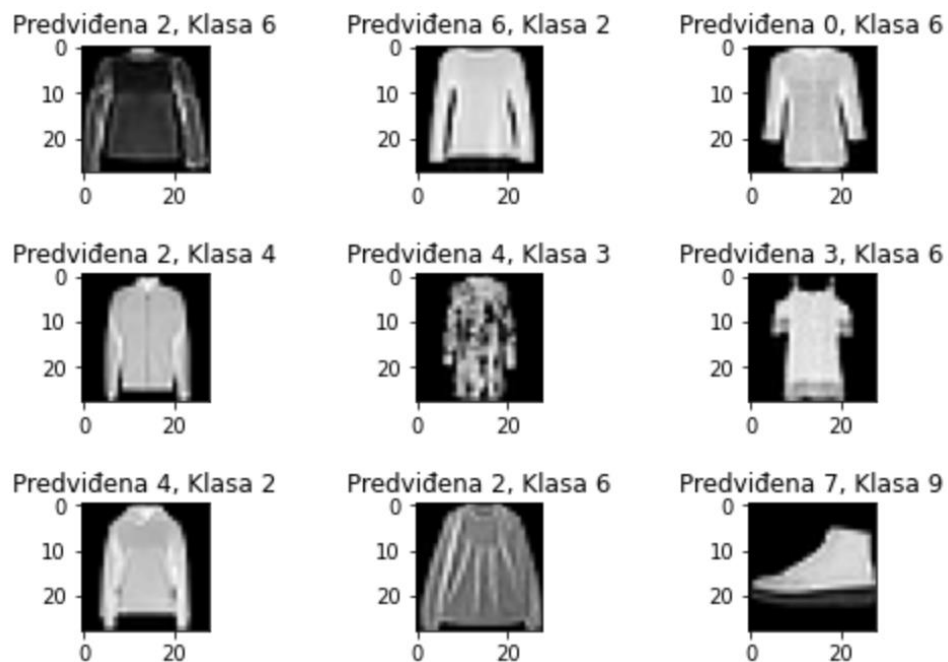
	Originalni skup obeležja 30% trening skupa	Originalni skup obeležja celog trening skupa	Skup obeležja primenom PCA metode
Validacioni skup	84.02%	86.47%	86.91%
Test skup	84.75%	87.06%	87.32%

Tabela 3. Prikaz prosečnog udela ispravno pogođenih klasa uzoraka na bazi K najbližih suseda

852	1	13	16	5	0	107	1	4	1
4	974	0	15	2	0	5	0	0	0
16	2	764	13	112	1	91	0	1	0
28	5	7	901	35	0	24	0	0	0
1	0	57	20	834	0	87	0	1	0
1	0	0	0	0	892	4	61	3	39
164	1	72	24	72	1	658	0	8	0
0	0	0	0	0	11	0	937	0	52
4	1	6	1	7	1	13	4	960	3
0	0	0	0	0	4	1	35	0	960

Tabela 4. Matrica konfuzije na test skupu

Na slici 11. prikazano je 9 primera pogrešno klasifikovanih odevnih predmeta iz test skupa, korišćenjem klasifikatora na bazi k najbližih suseda koji je obučavan nad skupom obeležja primenom PCA redukcije. Iznad svake slike levi broj predstavlja predviđenu klasnu labelu, a desni stvarnu klasu predmeta.



Slika 11. Primeri pogrešno klasifikovanih uzoraka

3.2 Rezultati modela na bazi vektora nosača

Odabrani model na bazi vektora nosača koji je obučavan na originalnom skupu obeležja, 30% trening skupa, koristi vrednost 7 regularizacionog parametra C , radijalnu kernel funkciju, i pristup višeklasnoj klasifikaciji jedan naspram svih. Parametar γ ima vrednost *auto* i on iznosi 0.00128. Isti model obučavan je i na celom trening skupu. Na isti način izvršena je obuka i nad celim trening skupom.

Odabrani model na bazi vektora nosača koji je obučavan na skupu obeležja sa PCA smanjenjem dimenzionalnosti, koristi vrednost 13 regularizacionog parametra C , radijalnu kernel funkciju, i pristup višeklasnoj klasifikaciji jedan naspram jednog. Parametar γ ima vrednost *scale* i on iznosi 0.00142.

U tabeli 5. predstavljeni su dobijeni rezultati, tj. prosečni udeli ispravno pogođenih klasa uzoraka na bazi vektora nosača nad originalnim skupom obeležja 30% uzoraka, originalnim skupom obeležja svih uzoraka i nad skupom obeležja primenom PCA redukcije. Vreme obuke nad 30% originalnog trening skupa je 38.64264 sekundi, nad celim trening skupom 312.21240 sekundi a nad skupom nad kojim je primenjeno PCA smanjenje dimenzionalnosti 89.02761 sekundi. Vreme klasifikacije jednog uzorka nakon obuke nad celim trening skupom sa PCA redukcijom je 0.00401 sekundi. Matrica konfuzije dobijena na test skupu korišćenjem modela obučenog na skupu obeležja sa PCA redukcijom je predstavljena u tabeli 6.

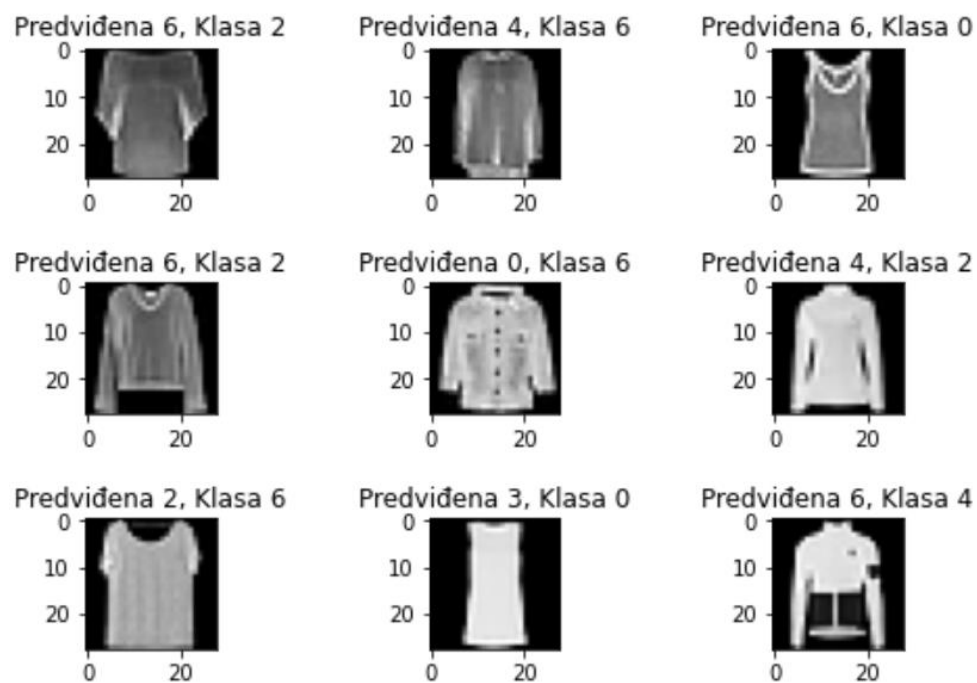
	Originalni skup obeležja 30% trening skupa	Originalni skup obeležja celog trening skupa	Skup obeležja primenom PCA metode
Validacioni skup	88.23%	90.14%	90.06%
Test skup	88.80%	90.76%	90.69%

Tabela 5. Prikaz prosečnog udela ispravno pogođenih klasa uzoraka na bazi vektora nosača

874	0	14	19	0	0	85	0	8	0
3	983	2	11	0	0	0	0	1	0
14	0	843	11	70	0	56	0	6	0
16	7	7	930	25	1	13	0	1	0
1	0	57	23	865	0	53	0	1	0
0	0	0	0	0	953	1	28	7	11
122	0	68	25	50	0	723	0	12	0
0	0	0	0	0	13	0	962	0	25
7	0	4	2	1	1	3	1	980	1
1	0	0	0	0	7	0	34	2	956

Tabela 6. Matrica konfuzije na test skupu

Na slici 12. prikazano je 9 primera pogrešno klasifikovanih odevnih predmeta iz test skupa, korišćenjem klasifikatora na bazi vektora nosača koji je obučavan nad skupom obeležja primenom PCA redukcije. Iznad svake slike levi broj predstavlja predviđenu klasnu labelu, a desni stvarnu klasu predmeta.



Slika 12. Primeri pogrešno klasifikovanih uzoraka

3.3 Rezultati modela na bazi višeslojnog perceptrona

Odabrani model na bazi višeslojnog perceptrona koji je obučavan na originalnom skupu obeležja, 30% trening skupa, koristi algoritam za pronalaženje minimuma *adam*, fiksnu brzinu učenja, aktivacionu funkciju *rectified linear unit*, i ima 3 skrivena sloja sa po 128, 64 i 16 neurona. Omogućeno je rano zaustavljanje algoritma, sa maksimalnim brojem iteracija 100 i veličinom uzoračkog podskupa 50. Na isti način izvršena je obuka i nad celim trening skupom.

Odabrani model na bazi višeslojnog perceptrona koji je obučavan na skupu obeležja sa PCA smanjenjem dimenzionalnosti, koristi algoritam za pronalaženje minimuma *adam*, fiksnu brzinu učenja, aktivacionu funkciju *rectified linear unit*, i ima 2 skrivena sloja sa po 128 i 128 neurona. Omogućeno je rano zaustavljanje algoritma, sa maksimalnim brojem iteracija 100 i veličinom uzoračkog podskupa 50.

U tabeli 7. predstavljeni su dobijeni rezultati, tj. prosečni udeli ispravno pogođenih klasa uzoraka na bazi višeslojnog perceptrona nad originalnim skupom obeležja 30% uzoraka, originalnim skupom obeležja svih uzoraka i nad skupom obeležja primenom PCA redukcije. Vreme obuke nad 30% originalnog trening skupa je 13.59952 sekundi, nad celim trening skupom 38.37395 sekundi a nad skupom nad kojim je primenjeno PCA smanjenje dimenzionalnosti 18.81113 sekundi. Vreme klasifikacije jednog uzorka nakon obuke nad celim trening skupom sa PCA redukcijom je 0.00010 sekundi. Matrica konfuzije dobijena na test skupu korišćenjem modela obučenog na skupu obeležja sa PCA redukcijom je predstavljena u tabeli 8.

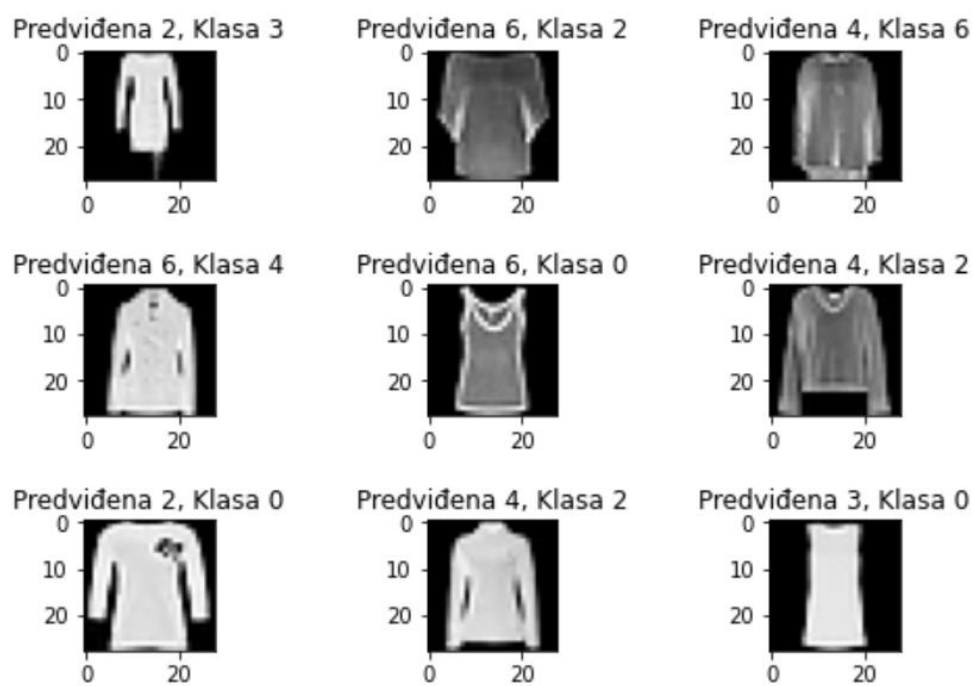
	Originalni skup obeležja 30% trening skupa	Originalni skup obeležja celog trening skupa	Skup obeležja primenom PCA metode
Validacioni skup	86.53%	88.57%	88.87%
Test skup	87.38%	88.85%	89.21%

Tabela 7. Prikaz prosečnog udela ispravno pogođenih klasa uzoraka na bazi vektora nosača

865	0	11	10	3	0	102	0	8	1
5	973	2	14	0	0	6	0	0	0
20	0	764	11	124	1	77	0	3	0
31	8	4	899	28	1	27	0	2	0
0	0	41	22	866	0	68	0	2	0
0	0	0	0	0	941	1	35	7	16
136	1	49	13	54	0	736	0	11	0
0	0	0	0	0	20	0	944	0	36
1	0	6	3	2	6	12	2	967	1
0	0	0	0	0	9	0	44	0	947

Tabela 8. Matrica konfuzije na test skupu

Na slici 13. prikazano je 9 primera pogrešno klasifikovanih odevnih predmeta iz test skupa, korišćenjem klasifikatora na bazi višeslojnog perceptrona koji je obučavan nad skupom obeležja primenom PCA redukcije. Iznad svake slike levi broj predstavlja predviđenu klasnu labelu, a desni stvarnu klasu predmeta.



Slika 13. Primeri pogrešno klasifikovanih uzoraka

3.4 Rezultati modela na bazi konvolucione neuronske mreže

Odabrani model na bazi konvolucione neuronske mreže koji je obučavan na originalnom skupu obeležja, 30% trening skupa, sastoji se iz konvolucionog sloja sa 32 kernela dimenzija 5×5 , sloja za maksimalno sažimanje sa filtera dimenzija 2×2 , *dropout* sloja, sloja za pretvaranje matrice u niz, jednog aktivacionog sloja koji primenjuje *relu* funkciju, i aktivacionog sloja koji primenjuje *softmax* funkciju. Koristi *adam* kao algoritam za pronalaženje minimuma, veličinu uzoračkog podskupa 256 i 15 epoha. Na isti način izvršena je obuka i nad celim trening skupom.

Odabrani model na bazi konvolucione neuronske mreže koji je obučavan na skupu obeležja sa PCA smanjenjem dimenzionalnosti sastoji se iz konvolucionog sloja sa 32 kernela dimenzija 5×5 , sloja za maksimalno sažimanje sa filtera dimenzija 2×2 , *dropout* sloja, sloja za pretvaranje matrice u niz, jednog aktivacionog sloja koji primenjuje *relu* funkciju, i aktivacionog sloja koji primenjuje *softmax* funkciju. Koristi *adam* kao algoritam za pronalaženje minimuma, veličinu uzoračkog podskupa 256 i 15 epoha.

U tabeli 9. predstavljeni su dobijeni rezultati, tj. prosečni udeli ispravno pogođenih klasa uzoraka na bazi konvolucione neuronske mreže nad originalnim skupom obeležja 30% uzoraka, originalnim skupom obeležja svih uzoraka i nad skupom obeležja primenom PCA redukcije. Vreme obuke nad 30% originalnog trening skupa je 22.83930 sekundi, nad celim trening skupom 82.08993 sekundi a nad skupom nad kojim je primenjeno PCA smanjenje dimenzionalnosti 9.38217 sekundi. Vreme klasifikacije jednog uzorka nakon obuke nad celim trening skupom 0.03502 sekundi. Matrica konfuzije dobijena na test skupu korišćenjem modela obučenog na skupu obeležja sa PCA redukcijom je predstavljena u tabeli 10.

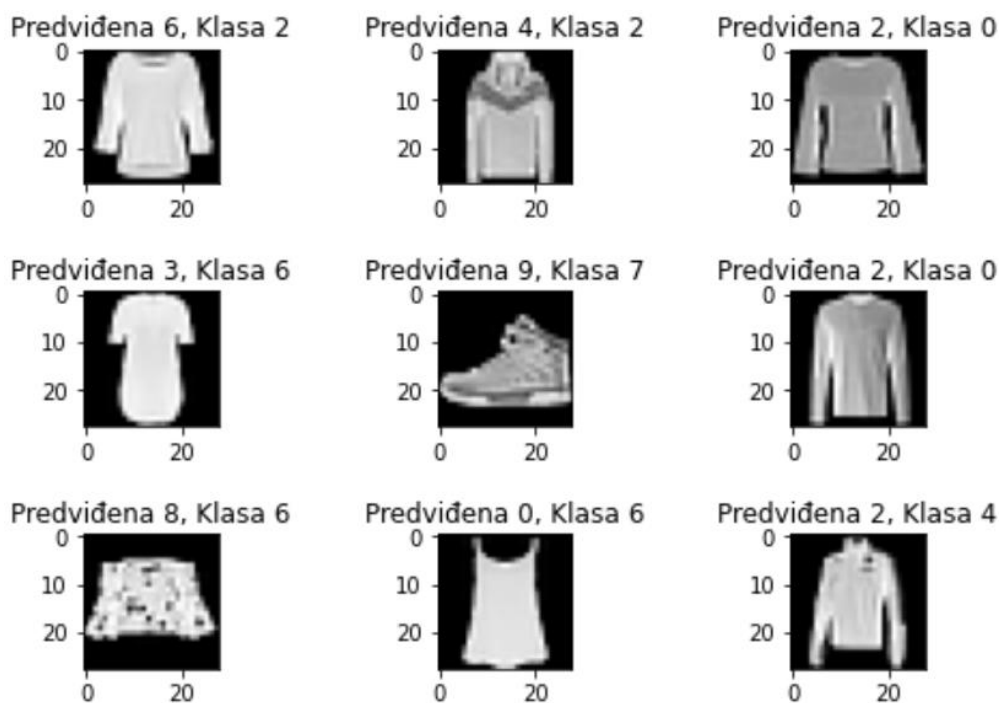
	Originalni skup obeležja 30% trening skupa	Originalni skup obeležja celog trening skupa	Skup obeležja primenom PCA metode
Validacioni skup	89.73%	91.45%	86.78%
Test skup	90.34%	92.11%	87.49%

Tabela 9. Prikaz prosečnog udela ispravno pogođenih klasa uzoraka na bazi vektora nosača

861	0	3	13	0	1	98	0	15	0
1	990	1	6	1	0	1	0	0	0
22	1	913	12	67	0	57	0	0	0
19	8	3	939	31	0	15	0	0	0
0	0	31	16	861	0	58	0	0	0
1	1	0	0	0	980	0	15	1	3
90	3	41	17	35	0	759	0	1	0
0	0	0	0	0	6	0	972	0	22
6	1	3	3	2	2	8	3	980	1
0	0	0	0	0	5	0	20	0	975

Tabela 10. Matrica konfuzije na test skupu

Na slici 14. prikazano je 9 primera pogrešno klasifikovanih odevnih predmeta iz test skupa, korišćenjem klasifikatora na bazi konvolucione neuronske mreže koji je obučavan nad skupom obeležja primenom PCA redukcije. Iznad svake slike levi broj predstavlja predviđenu klasnu labelu, a desni stvarnu klasu predmeta.



Slika 14. Primeri pogrešno klasifikovanih uzoraka

4. Zaključak

U ovom radu su prikazana četiri različita algoritma mašinskog učenja koji su korišćeni za obučavanje modela sa ciljem predviđanja klasnih labela odevnih predmeta sa slike. Od modela koji su obučavani korišćenjem originalnih obeležja celog skupa podataka najbolje rezultate za vrednost udela ispravno pogođenih klasa ima algoritam konvolucione neuronske mreže, dok je mašina na bazi vektora nosača dala veoma bliske rezultate. Na ovom problemu, sa izabranim evaluiranim parametrima, algoritmi k najbližih suseda, i višeslojnog perceptrona su dali lošije rezultate. Prilikom obučavanja na skupu obeležja smanjenih dimenzija PCA redukcijom algoritmi su održali isti poredak u performansama, međutim pokazalo se da su lošiji rezultati nego kada su modeli obučavani na celom skupu obeležja.

Posmatranjem druge mere za kvalitet performansi modela, tj. vremena izvršavanja obuke i predviđanja modela, kao najbolji se pokazao algoritam višeslojnog perceptrona, dok je konvolucionoj mreži i mašini na bazi vektora nosača potrebno više vremena. Iako je vreme izvršavanja algoritma konvolucione neuronske mreže, gde je model obučen nad skupom sa PCA redukcijom, brzo, algoritam ne daje toliko dobre rezultate kao i kada je model obučen nad celim originalnim trening skupom. Iako PCA metoda nije pomogla poboljšanju udela ispravno pogođenih klasa, značajno je uspela da smanji vremena izvršavanja ostalih algoritama. Svi algoritmi pokazali su izuzetno brza vremena klasifikacije pojedinačnog uzorka, pri čemu je ubedivo najbolje vreme ostvario algoritam višeslojnog perceptrona.

Prethodne konstatacije ukazuju da odabir optimalnog algoritma za obuku modela na ovom problemu zavisi od preference korisnika koji će ga primenjivati. Ukoliko vreme izvršavanja obuke nije od presudnog značaja najbolje je primeniti model koji je obučen na algoritmu konvolucione neuronske mreže, dok ukoliko je bitno ostvarivanje najefikasnijeg vremena, najbolje je primeniti model obučen algoritmom višeslojnog perceptrona.

Razvoj projekta moguće je nastaviti daljim unapređenjem arhitekture konvolucione neuronske mreže koja se pokazala kao najbolja za rešavanje ovog tipa problem. Ovo se može postići dodavanjem dodatnih konvolucionih slojeva, kao i povećanja broja epoha za obuku, čime će se vreme izvršavanja povećati, ali će kao rezultat doprineti boljem udelu ispravno klasifikovanih uzoraka. Takođe je moguće primeniti različite vrste selekcije ili redukcije obeležja. kNN algoritam je moguće još više ubrzati paralelizacijom, tj. izvršavanjem algoritma na računaru sa većim brojem procesora ili većem broju umreženih računara, kako bi raspodela posla bila brža i efikasnija.

Literatura

- [1] Kumar, M., Sharma, R. K., & Sethi, I. (2021). *Machine Learning in Image Analysis and Pattern Recognition*, MDPI AG
- [2] Tanveer, M. S., Khan, M. U. K., & Kyung, C. M. (2021, January). *Fine-tuning darts for image classification*. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 4789-4796)
- [3] Kayed, M., Anter, A., & Mohamed, H. (2020, February). *Classification of garments from fashion MNIST dataset using CNN LeNet-5 architecture*. In *2020 international conference on innovative trends in communication and computer engineering (ITCE)* (pp. 238-243)
- [4] Xiao, H., Rasul, K., & Vollgraf, R. (2017). *A novel image dataset for benchmarking machine learning algorithms*. *arXiv preprint arXiv:1708.07747*.
- [5] Jolliffe, I.T. (2010), *Principal Component Analysis, Second Edition*, Springer Berlin (pp. 1-10)
- [6] Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press (pp. 258-267)
- [7] Flach, P. (2012). *Machine learning: the art and science of algorithms that make sense of data*. Cambridge university press (pp. 242-253)
- [8] Witten, I.H. & Frank, E. (2005) *Data mining: Practical machine learning tools and techniques*. 2nd Edition, Morgan Kaufmann Publisher, Burlington (pp. 161-169)
- [9] Back, A. D. (2003). Classification using support vector machines. *Riken Brain Science Institute, Wako-shi, Japan*. Available on the Web at: <http://andrewback.com/webpapers/svm/index.php>.
- [10] Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*. New York: springer (pp. 291-302)
- [11] Peters, T. (2019). *Data-driven science and engineering: machine learning, dynamical systems, and control: by SL Brunton and JN Kutz, 2019, Cambridge, Cambridge University Press* (pp. 180-185)
- [12] Herbrich, R. (2001). *Learning kernel classifiers: theory and algorithms*. MIT press (pp. 49-59)
- [13] Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 25). San Francisco, CA, USA: Determination press (pp. 2-53)
- [14] Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*. New York: springer (pp. 225-262)
- [15] Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 25). San Francisco, CA, USA: Determination (pp. 167-210)

- [16] Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*. New York: springer (pp. 267-272)
- [17] Dabović, M. M., & Tartalja, I. I. (2017). Duboke konvolucijske neuronske mreže–koncepti i aktuelna istraživanja. *Zbornik 61. Konferencije za elektroniku, telekomunikacije, računarstvo, automatiku i nuklearnu tehniku, ETRAN 2017, Kladovo*
- [18] Kothiya, A. (2021, June 23). *Understanding “convolution” operations in CNN*. <https://medium.com/analytics-vidhya/convolution-operations-in-cnn-deep-learning-computer-vision-128906ece7d3>