

Detekcija prisutnosti ljudi u Wi-Fi polju pomoću strojnog učenja

Tonka Hrboka*

mentori prof. dr. sc. Hrvoje Buljan i izv. prof. dr. sc. Dario Jukić
Fizički odsjek, Prirodoslovno-matematički fakultet, Bijenička 32, Zagreb

22. siječnja 2023.

Sažetak

Razmatra se metoda detekcije ljudi u Wi-Fi polju korištenjem činjenice da je koeficijent permitivnosti ljudskog tijela velik na Wi-Fi frekvenciji. Cilj je strojnim učenjem odrediti broj osoba u prostoriji iz podataka o iznosu i fazi električnog polja. Iznesene su teorijske osnove strojnog učenja te je provedena analiza podataka simuliranih u programu Comsol. U programskom jeziku Python izrađeni su algoritmi logistička regresija i neuralna mreža te je postignuta maksimalna točnost klasifikacije 62.5% za podjelu u pet klasa te 100% za podjelu u tri klase. Očekuju se daljnji napretci korištenjem algoritama na eksperimentalnim podatcima.

1. Uvod

Današnja tehnologija dosegla je velike uspjehe u detekciji predmeta i ljudi, već se rutinski koristi u autonomnim vozilima, videonadzoru pa čak i u sportu. Metode se uglavnom temelje na kamerama i računalnom vidu te su pretežito namijenjene za uporabu na otvorenom prostoru, u zatvorenim prostorima vidno polje kamera je ograničenije jer je u pravilu gustoća predmeta veća. Korištenje kamera za videonadzor nije toliko popularno i zbog narušenja privatnosti, iako bi mogućnost detekcije uljeza u školama, zdravstvenim ustanovama, a i privatnim domovima bila poželjna. Ovaj seminar bavit će se alternativnom metodom detekcije ljudi u zatvorenom prostoru koristeći Wi-Fi polje. Wi-Fi odašiljači široko su rasprostranjeni, polje je dovoljno velikog dosega, prolazi nepromijenjeno kroz predmete, ali se značajno mijenja prolaskom kroz ljudsko tijelo što ga čini savršenim za ovakvu uporabu.

2. Teorijska pozadina

Wi-Fi polje elektromagnetski je val frekvencije 2.4 GHz. Električno i magnetsko polje u homogenom izotropnom mediju zadovoljava valnu jednadžbu:

$$\begin{aligned}\epsilon_0\epsilon_r\mu_0\mu_r\partial_t^2\vec{E} &= \Delta\vec{E} \\ \epsilon_0\epsilon_r\mu_0\mu_r\partial_t^2\vec{B} &= \Delta\vec{B}\end{aligned}\tag{1}$$

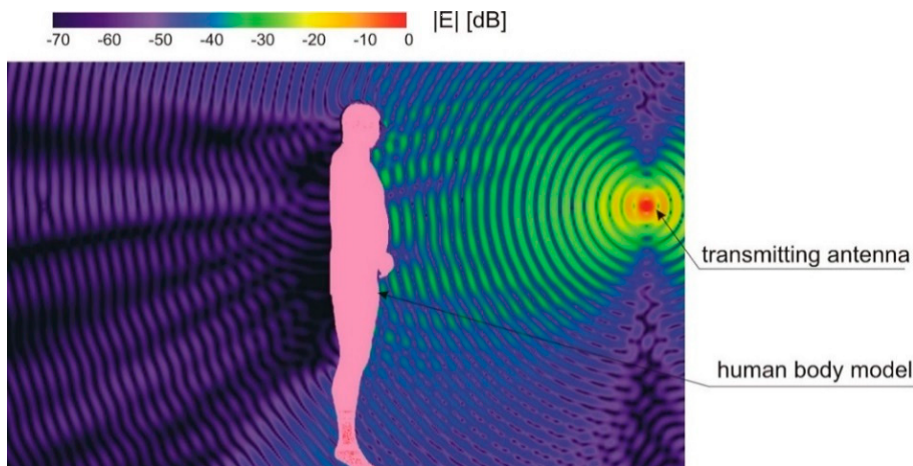
koja se lako izvede koristeći Maxwellove jednadžbe. Koeficijenti relativne permitivnosti ϵ_r i relativne permeabilnosti μ_r ovise o materijalu pa se na granici dva medija mijenja smjer širenja

* tonka.hrboka@student.pmf.hr, tonka.hrboka@gmail.com

vala po Snellovom zakonu:

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{n_2}{n_1} \quad (2)$$

pri čemu su θ upadni kutevi, a $n = \sqrt{\epsilon_r \mu_r}$ indeksi loma.



Slika 1. Iznos električnog polja u blizini ljudskog tijela, preuzeto iz reference [1]

Zanimljivo je da je koeficijent permitivnosti ljudskog tijela na Wi-Fi frekvenciji velik, stoga ljudsko tijelo značajno mijenja električno polje u kojem se nalazi, polje je atenuirano te se mijenja smjer širenja valova, prikazano na Slici 1. Ideja je mjerenjem iznosa i faze električnog polja duž prostorije odrediti broj ljudi koji se nalaze u prostoriji. Uvođenjem niza pretpostavki mogle bi se izračunati očekivane vrijednosti polja za dani raspored ljudi, ali za obrnut postupak rekreacije broja ljudi na temelju vrijednosti polja potrebno je koristiti strojno učenje.

3. Strojno učenje

Termin strojno učenje odnosi se na automatsku detekciju značajnih uzoraka u podacima te je relativno nedavno postalo glavni alat u zadacima koji iziskuju dobivanje informacija iz velikog broja podataka: koristi se u filtriranju mailova, sortiranju rezultata u pretraživanju, personaliziranju reklama, detekciji prevara u kartičnim transakcijama, prepoznavanju lica i govora na pametnim mobitelima, ali i u znanosti. Svim primjerima je zajedničko da su uzorci koje je potrebno prepoznati iznimno složeni te programer ne može eksplicitno napisati naredbe koje bi zadatak izvršile korak po korak, već po uzoru na ljude program "uči" svoje ponašanje na temelju iskustva, odnosno danih primjera. [2]

Ovisno o zadatku koji je potrebno riješiti i dostupnim podacima, pojam strojno učenje može se podijeliti na nadzirano i nenadzirano učenje. Nenadziranim učenjem traže se pravilnosti i uzorci u samim podacima, a u nadziranom učenju, kojim će se prvenstveno i baviti ovaj seminar, dostupan je niz primjera sa već označenim vrijednostima te je cilj programa na temelju tih primjera odrediti oznake za nove neviđene primjere. Dalje u nadziranom učenju postoje regresijski i klasifikacijski problemi, što se odnosi na to jesu li moguće oznake brojčane vrijednosti ili svrstavanje primjera u grupe koje se nazivaju klase. Na primjer, ako je problem predvidjeti cijenu nekretnine, radi se o regresijskom problemu, a filtriranje elektroničke pošte u kategorije je klasifikacijski problem.

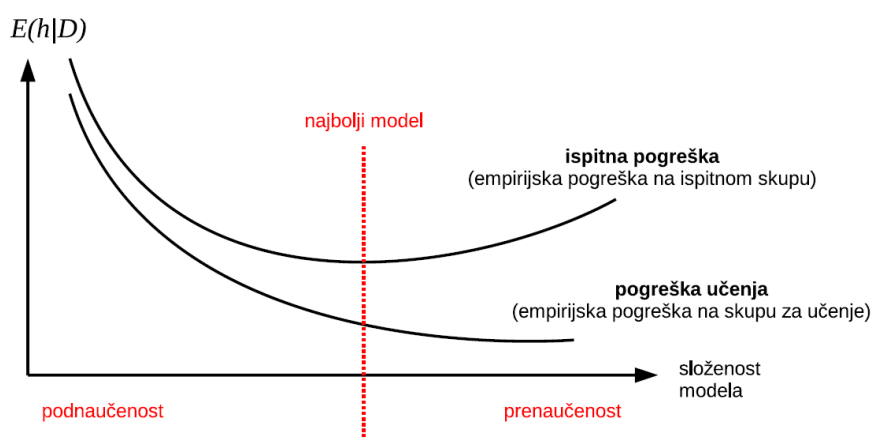
Već spomenuti "primjer", naziv je za ulaz algoritma strojnog učenja, konkretno to može biti jedna nekretnina. Primjer se prikazuje kao vektor značajki, vrijednosti relevantnih za taj primjer, za nekretninu bi to bile površina, starost, pozicija itd. Svrha nadziranog strojnog učenja je naučiti funkciju koja primjerima dodjeljuje oznake, ta funkcija naziva se hipoteza. Naravno unaprijed

egzaktna funkcija nije poznata, u suprotnom ne bi postojala svrha učenja, ali potrebno je odlučiti se za oblik funkcije, uzimajući u obzir dostupno znanje o problemu. Odabrana funkcija definirana je do na parametre koje određuje sam program. [3]

Svaki algoritam strojnog učenja sastoji se od tri glavne komponente:

- Model
- Funkcija pogreške
- Optimizacijski postupak

Model je naziv za skup svih mogućih hipoteza koje se razlikuju do na iznos parametara. Odabir optimalnog modela je zadatak programera, a program optimizacijskim postupkom pretražuje skup hipoteza te odabire najbolju unutar zadanog modela. Funkcija gubitka numerički govori o tome koliko se razlikuje predviđanje modela od označenog primjera.



Slika 2. Teorijska ovisnost pogreške o složenosti modela na skupu za učenje i ispitnom skupu, preuzeto iz reference [3]

3.1. Unakrsna provjera

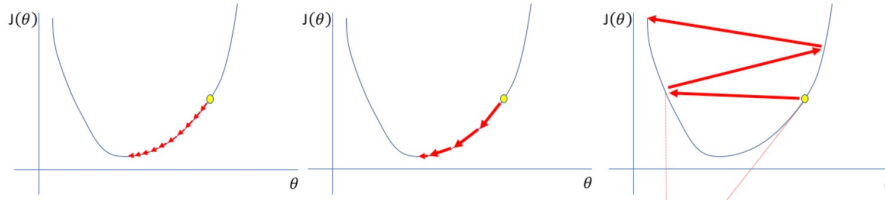
S obzirom da je konačan cilj dobiti algoritam koji uspješno označava neviđene primjere, za odabir najboljeg modela potrebno je testirati algoritme na takvim primjerima. U tu svrhu koristi se unakrsna provjera; skup dostupnih primjera dijeli se na skup za učenje i ispitni skup, optimizacija modela provodi se samo na primjerima iz skupa za učenje te dobiven model predviđa oznake za primjere iz skupa za testiranje. Greške se računaju na oba skupa, predviđeno ponašanje u ovisnosti o složenosti modela jest prikazano na Slici 2. Ako je model jako složen, greška na skupu za učenje će biti minimalna, ali to nije poželjno. Podatci nikad nisu savršeni, skup za učenje sadrži šum koji sprječava da primjeri savršeno odgovaraju matematičkom modelu. Cilj je naći model koji unatoč šumu opisuje stvarni trend odnosno dobro generalizira, stoga je optimalan model onaj koji minimizira ispitnu grešku.

3.2. Gradijentni spust

Gradijentni spust optimizacijski je postupak koji se koristi za modele koji će biti opisani u sljedećim poglavljima. Odabire se početni vektor težina \vec{w} te se on ažurira pomicanjem u smjeru suprotnom od gradijenta u danoj točki niz površinu funkcije pogreške za stopu η :

$$\vec{w} \leftarrow \vec{w} - \eta \vec{\nabla} E \quad (3)$$

Koraci se ponavljaju dok se ne pronađe optimalni vektor težina \vec{w}^* koji odgovara globalnom minimumu funkcije pogreške. Poželjno je definirati konveksnu funkciju pogreške, kako se optimizacija ne bi zaustavila na lokalnom minimumu. Također je bitno dobro odabrati stopu učenja, prevelika stopa može rezultirati divergencijom, a premala uzrokuje presporu konvergenciju, prikazano na Slici 3. Postoje razne metode za određivanje stope, poput linijskog pretraživanja u kojem se traži točka na pravcu u smjeru negativnog gradijenta koja odgovara minimumu funkcije ili optimizacije drugog reda pri kojima se računa i promjena gradijenta, poput Newtonovog postupka.



Slika 3. Ovisnost greške o parametru modela, prikazuje odabir stope pri gradijentnom spustu, s lijeva na desno: premala stopa, optimalna stopa i prevelika stopa, preuzeto iz reference [4]

3.3. Logistička regresija

Logistička regresija je klasifikacijski algoritam čiji je model za binarnu klasifikaciju linearna funkcija omotana aktivacijskom funkcijom sigmoidom:

$$h(\vec{x}; \vec{w}) = \sigma(\vec{x} \cdot \vec{w}) = \frac{1}{1 + e^{-\vec{x} \cdot \vec{w}}} \quad (4)$$

gdje je \vec{x} vektor značajki, a \vec{w} vektor parametara, odnosno težina. Funkcija gubitka izvodi se probabilistički kao negativan logaritam izglednosti te se konačan izraz naziva gubitak unakrsne entropije:

$$L(y, h(\vec{x})) = -y \ln(h(\vec{x})) - (1 - y) \ln(1 - h(\vec{x})) \quad (5)$$

pri čemu su y oznake označenih primjera te mogu poprimiti vrijednosti 0 ili 1. Poopćenje binarne logističke regresije na više klasa naziva se multinomijalna logistička regresija, u modelu se koristi zaseban vektor težina \vec{w}_k za svaku klasu te se skalarni umnožak $\vec{w}_k \cdot \vec{x}$ stavlja u funkciju softmax:

$$h_k(\vec{x}; \vec{w}_1, \dots, \vec{w}_K) = \frac{\exp(\vec{w}_k \cdot \vec{x})}{\sum_{j=1}^K \exp(\vec{w}_j \cdot \vec{x})} = P(y = k; \vec{w}_1, \dots, \vec{w}_K) \quad (6)$$

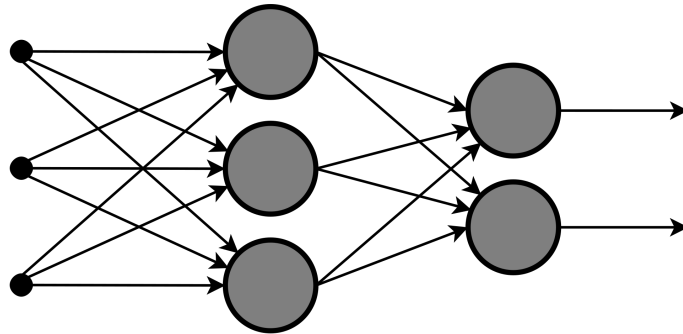
koja normalizira vrijednosti radi mogućnosti probabilističke interpretacije te povećava velike vrijednosti, a smanjuje male. Funkcija gubitka definira se analogno kao za binarni problem, pri čemu je oznaka y_k jednaka 1 ako primjer pripada klasi k , a inače je jednaka nuli:

$$L(y, h_k(\vec{x})) = - \sum_{k=1}^K y_k \ln(h_k(\vec{x})) \quad (7)$$

Logističkom regresijom može se modelirati i nelinearna granica među klasama koristeći preslikavanje ulaznog prostora u prostor značajki više dimenzije.

3.4. Neuralna mreža

Neuralna mreža algoritam je modeliran po uzoru na ljudski mozak, sastoji se od jednostavnih operativnih jedinica neurona koji su međusobno povezani. Neuron se postavlja u slojeve tako da izlazi jednog sloja predstavlja ulaze sljedećem sloju, kao što je prikazano na Slici 4. Neuron računa linearnu kombinaciju ulaznih vrijednosti s pripadnim vrijednostima težina koje se određuju procesom učenja te primjenjuje aktivacijsku funkciju. Odabir aktivacijske funkcije ovisi o podacima i zadatku koji se izvršava, neki primjeri su već spomenute sigmoida i softmax, funkcija zglobnica ili tangens hiperbolni. Postoje i neke izvedbe koje ne koriste linearnu kombinaciju poput rezidualne mreže i transformer mreže.



Slika 4. Shema rada neuronske mreže, dvoslojna mreža sa 3 i 2 neurona, preuzeto iz reference [6]

3.5. Regularizacija

Jedna od metoda kojima se sprječava prenaučenosť jest regularizacija, ideja je funkciji pogreške dodati regularizacijski izraz koji time onemogućuje dodjeljivanje velikih težina nekim značajkama. Regularizacijski izraz može se definirati kao p-norma vektora težina:

$$\|\vec{w}\| = \sum_{j=1}^m |w_j^p|^{1/p} \quad (8)$$

S obzirom da program nastoji minimizirati pogrešku, sam algoritam potiskuje vrijednosti težina čime se smanjuje složenost modela. Ovako definiran regularizacijski izraz za $p = 1, 2$ vodi do često korištenih regularizacija: L1-Lasso i L2-Ridge. L1-regularizacija više kažnjava male vrijednosti od L2 (jer se iznos ne kvadrira), stoga ona rezultira "rijetkim" modelima u kojima su neke težine pritegnute na nulu i korisna je ako je cilj identificirati bitne značajke za model. L2-regularizacija korisna je zbog toga što je regularizacijski izraz kontinuiran pa se lakše mogu provoditi matematičke manipulacije, na primjer za L2-regularizirani linearni model regresije optimizacija ima rješenje u zatvorenoj formi što značajno smanjuje računalnu složenost.

Za neuralne mreže koristi se i dropout tehnika regularizacije pri kojoj se nasumično, sa predefiniranom vjerojatnošću, isključuju neuroni tijekom učenja što osigurava da se model ne oslanja previše na jedan neuron ili pojedinu grupu neurona. Preostali neuroni preuzimaju potrebnu reprezentaciju za točno predviđanje što u konačnici čini mrežu osjetljivijom i proces učenja je brži. [5]

4. Podatci

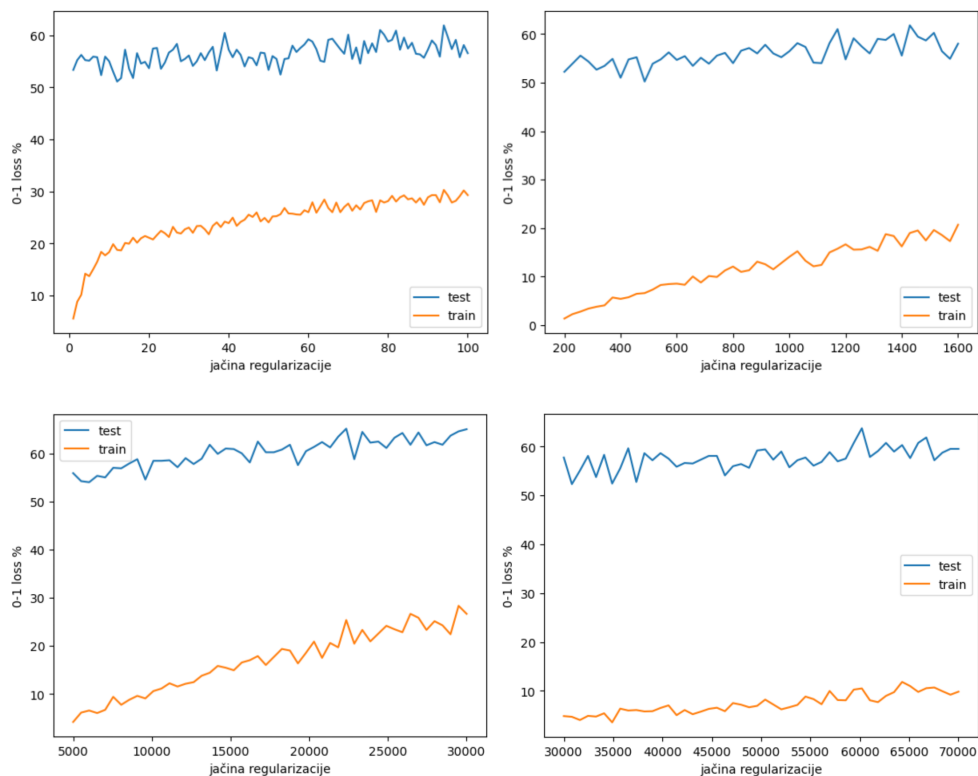
Skup podataka koji će se analizirati dobiven je simulacijom u programu Comsol. Ljudi su aproksimirani vodenim cilindrima smještenim u prostoriju u kojoj se nalazi 15 detektora koji bilježe kompleksnu vrijednost električnog polja. Napravljeno je 15 simulacija, po 3 seta podataka za

	% x_cyl1 (m)	y_cyl1 (m)	freq (GHz)	Electric field, z component (V/m), Point Probe 1	Electric field, z component (V/m), Point Probe 2
0	2.531433	1.876546	2.4	129.71268991231247+89.88570310539056i	312.53412424485856-259.25192202306556i
1	0.842063	3.389266	2.4	207.43611006017025-28.31932299402317i	577.4981325573726+56.30056324345263i
2	1.549929	2.140417	2.4	-32.24836906129431+323.6849120619208i	255.03846513476083+200.8633338170793i
3	3.704058	2.063407	2.4	311.8766364728494+4.1532356831568755i	423.33511705994096-348.572860545794i
4	0.616881	1.194783	2.4	526.5747988019667+553.7652696016759i	480.34718757566003+173.2734721374108i
5	4.215417	1.966693	2.4	215.78702270414217+314.7807370708401i	486.3321605178132-193.25024265914593i
6	3.421323	2.372180	2.4	76.13026080300682+13.978200041080445i	539.5994017635433-27.736729403505684i
7	2.454436	2.537407	2.4	252.20872762723383-14.73899911119828i	717.7104917722033-252.5746971676793i
8	2.814100	1.686546	2.4	-25.39063724550902+371.54477648874314i	340.25626413356486-89.44927473082052i
9	1.449134	1.602310	2.4	-210.8360943507636+96.46443183310589i	342.96566224691264-243.92345005242785i

Slika 5. Primjer podataka dobivenih simulacijom

1, 3, 5, 7 i 9 ljudi u prostoriji na 10 različitim pozicija u prostoru, što ukupno daje 150 primjera za učenje, od kojih svaki ima 30 značajki. Značajke su vrijednosti električnog polja dobivene na detektorima, ima ih dvostruko više od broja detektora jer se odvojeno tretiraju realni i imaginarni dijelovi. Podatci su zabilježeni u .csv datotekama, na Slici 5. prikazan je primjer dijela seta podataka, početni stupci označavaju poziciju osobe u prostoriji, a posljednja dva stupca prikazuju realne i imaginarne vrijednosti električnog polja na detektorima (sa slike je izostavljeno ostalih 13 stupaca radi preglednosti).

5. Rezultati i diskusija



Slika 6. Ovisnost gubitka logističke regresije o jačini regularizacije bez preslikavanja (gore lijevo) te s preslikavanjem u polinom drugog stupnja (gore desno), polinom trećeg stupnja (dolje lijevo) i polinom četvrtog stupnja (dolje desno)

Svi kodovi i rezultirajući grafovi izrađeni su u programskom jeziku Python. Podatci su podijeljeni

ljeni na skupove za učenje i provjeru funkcijom `train_test_split` iz biblioteke `sklearn.model_selection`.

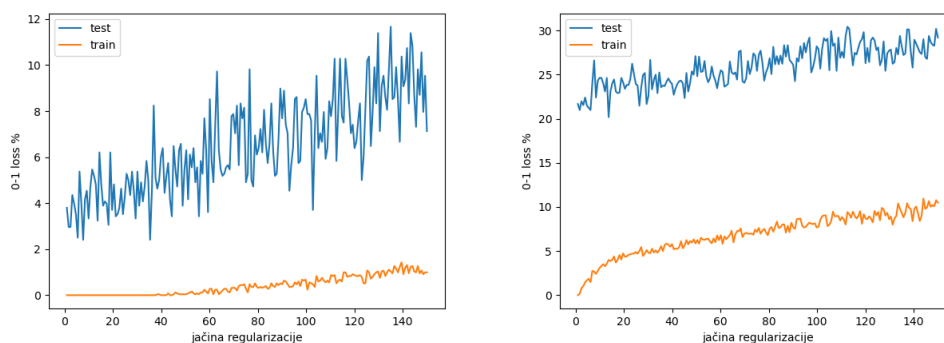
5.1. Logistička regresija

Ideja je na početku testirati točnost logističke regresije jer se jednostavniji modeli preferiraju ako mogu dati dobre rezultate. Jednostavniji model manje je računalno složen, a za razliku od neuronske mreže koja djelomično funkcionira kao crna kutija, rezultati bi trebali biti transparentniji i lakši za interpretaciju. Za implementaciju algoritma logističke regresije korištena je funkcija `LogisticRegression` iz biblioteke `sklearn.linear_model`. U argumentu funkcije odabrana je L2 regularizacija, multinomijalan pristup problemu više klasa te `'lbfgs'` optimizacijski algoritam što je u suštini gradijentni spust u kojem se stopa bira aproksimacijom druge derivacije funkcije pogreške. Podatci su skalirani funkcijom `StandardScaler` iz biblioteke `sklearn.preprocessing`. Za modeliranje nelinearne granice, provedeno je i učenje sa preslikavanjima u prostor polinoma drugog, trećeg i četvrog stupnja za što je korištena funkcija `PolynomialFeatures` iz biblioteke `sklearn.preprocessing`.

Dobiveni rezultati prikazani su na Slici 6., prikazana je jednostavna 0-1 greška koja ne kažnjava točno klasificirane primjere, a netočno klasificirane primjere kažnjava s 1 te se ukupan zbroj dijeli s brojem primjera. Maksimalna točnost dobivena za svaki model prikazana je u Tablici 1. Rezultati nisu zadovoljavajući, točnost na testnom skupu u svim je modelima manja od 50% te se ne uočava trend povećanja točnosti pri preslikavanju primjera u prostor viših dimenzija.

stupanj preslikavanja	najveća točnost [%]
1	48.89
2	49.78
3	46.00
4	47.67

Tablica 1. Najveća postignuta točnost modela logističke regresije za različite stupnjeve preslikavanja u prostor viših dimenzija



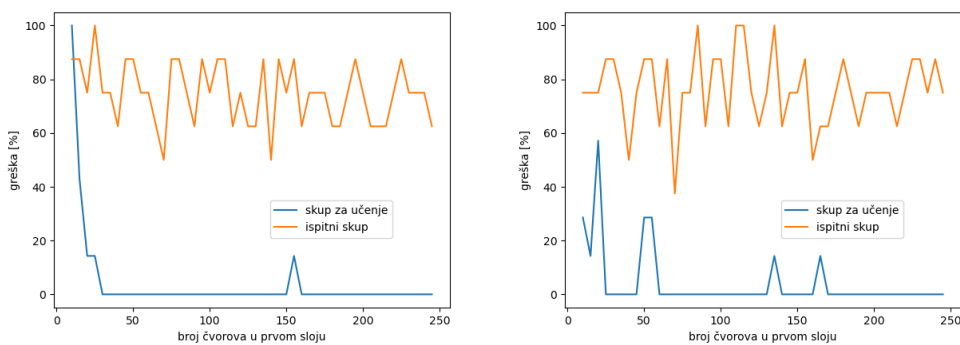
Slika 7. Ovisnost greške logističke regresije o jačini regularizacije na podskupovima podataka klase 1 i 9 (lijevo) te klase 1, 5 i 9 (desno)

Kako bi se dodatno testirao model logističke regresije, početni problem je pojednostavljen te je provedeno učenje na podskupu primjera dobivenih simulacijom. Odabrane su samo određene klase koje je potrebno raspoznati te su rezultati prikazani na Slici 7. Dobivena je značajno veća točnost: 97.59% za raspoznavanje klasa 1 i 9 ljudi u prostoru te 79.81% za raspoznavanje klasa 1, 5, i 9. Postoji mogućnost da je ovakav problem linearno odvojiv za razliku od potpunog problema pa zbog toga logistička regresija ima značajno bolje rezultate, ali možda je problem i samo u premalom broju podataka. Smanjenjem broja klasa ostvareno je da su primjeri različitih klasa više udaljeni u faznom prostoru, stoga je moguće točnije pronaći njihovu granicu. Za relativno malen broj primjera i velik broj klasa primjeri su međusobno prebliski te ih ukupno nema dovoljno kako bi se

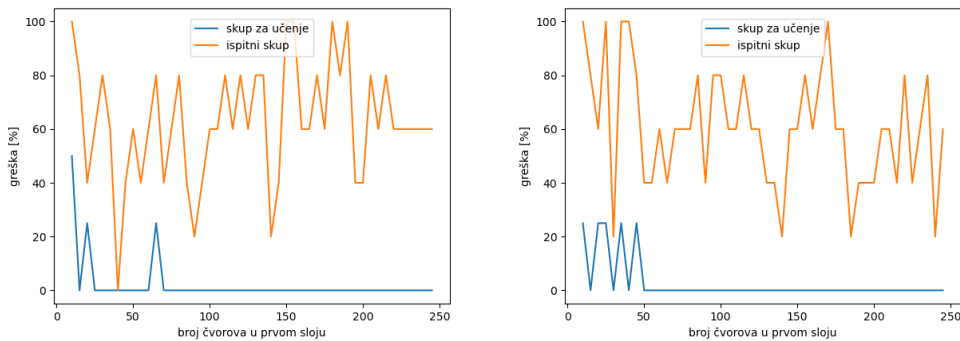
uspješno mogla naći granica. Na nijednom od dosadašnjih grafova ne vidi se očekivani trend na Slike 2. (regularizacija je direktno povezana sa složenosti modela) što je problem također pripisan premalom broju primjera.

5.2. Neuralna mreža

Za implementaciju algoritma neuralne mreže korištena je biblioteka *tensorflow*, konkretno modul *keras* i funkcija *Sequential* te je odabrana aktivacijska funkcija zglobnice *'relu'*. Za učenje je odabrana funkcija pogreške prikladna za rad s više klasa čije su oznake cijeli brojevi *SparseCategoricalCrossentropy* te optimizator *Adam* koji koristi stohastički gradijentni spust. Napravljena je mreža sa tri i četiri sloja, variran je broj čvorova u prvom sloju mreže, a za čvorove u preostalim slojevima odabrane su konstantne vrijednosti 60 i 10, odnosno 60, 60 i 10.



Slika 8. Ovisnost greške o broju čvorova u prvom sloju neuralne mreže, za mrežu s ukupno tri sloja (lijevo) te mrežu s ukupno četiri sloja (desno) na cijelom skupu podataka



Slika 9. Ovisnost greške o broju čvorova u prvom sloju neuralne mreže, za mrežu s ukupno tri sloja (lijevo) te mrežu s ukupno četiri sloja (desno) na podskupu podataka - klase 1, 5 i 9 ljudi u prostoriji

Rezultati su prikazani na Slici 8., korištena je jednostavna 0-1 greška kao i u prethodnom poglavlju. Maksimalna dobivena točnost za mrežu sa tri sloja jest 50%, a za mrežu s četiri sloja 62.5%. Ponovljeno je učenje i na podskupu podataka, rezultati prikazani na Slici 9. te je najveća dobivena točnost 100% za mrežu s tri sloja i 80% za mrežu sa četiri sloja. Neuralna mreža u konačnici pokazuje bolje rezultate od logističke regresije.

6. Zaključak

Modeli logistička regresija i neuralna mreža trenirani su na simuliranom skupu podataka te je uspješnost modela testirana unakrsnom provjerom. Najveća točnost dobivena na ispitnom skupu iznosi 62.5% i postignuta je četveroslojnom neuralnom mrežom. Ograničavanjem na manji broj klasa algoritmi rade značajno bolje. Glavni uzrok nezadovoljavajuće točnosti je premali broj primjera za učenje. U vremenu otkad je počela izrada seminara, proveden je i realan eksperiment te je dobivena veća baza podataka, stoga se očekuju značajno bolji rezultati primjenom izrađenih algoritama na većem skupu za učenje.

Literatura

- [1] Januszkiewicz, L. (2018) "Analysis of human body shadowing effect on wireless sensor networks operating in the 2.4 ghz band," *Sensors*, 18(10), p. 3412. Dostupno na: <https://doi.org/10.3390/s18103412>.
- [2] Shalev-Shwartz, S. and Ben-David, S. (2014) *Understanding machine learning: From theory to algorithms*. Cambridge: Cambridge University Press.
- [3] Jan Šnajder, Bilješke s predavanja - Strojno učenje. Dostupno na: <https://www.fer.unizg.hr/predmet/struce1/materijali>
- [4] https://duchesnay.github.io/pystatsml/optimization/optim_gradient_descent.html
- [5] Jabir, B. and Falih, N. (2021) "Dropout, a basic and effective regularization method for a deep learning model: A case study," *Indonesian Journal of Electrical Engineering and Computer Science*, 24(2), p. 1009. Dostupno na: <https://doi.org/10.11591/ijeecs.v24.i2.pp1009-1016>.
- [6] https://hr.wikipedia.org/wiki/Neuronska_mreža