

# Konvolucijske neuronske mreže za klasifikaciju objekata

---

Mehinović, Lejla

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:811211>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-15**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli  
Fakultet informatike u Puli

**LEJLA MEHINOVIĆ**

**KONVOLUCIJSKE NEURONSKE MREŽE  
ZA KLASIFIKACIJU OBJEKATA**

Diplomski rad

Pula, lipanj, 2020. godine

Sveučilište Jurja Dobrile u Puli  
Fakultet informatike u Puli

**LEJLA MEHINOVIĆ**

**KONVOLUCIJSKE NEURONSKE MREŽE  
ZA KLASIFIKACIJU OBJEKATA**

Diplomski rad

**JMBAG: 0303086562, redoviti student**

**Studijski smjer: Diplomski studij informatike**

**Predmet: Umjetna inteligencija**

**Znanstveno područje: Društvene znanosti**

**Znanstveno polje: Informatičke i komunikacijske znanosti**

**Znanstvena grana: Informatički sustavi i informatologija**

**Mentor: doc. dr. sc. Darko Etinger**

Pula, lipanj, 2020. godine



## IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Lejla Mehinović, kandidat za magistra informatike ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

Pula, lipanj, 2020. godine



## IZJAVA

o korištenju autorskog djela

Ja, Lejla Mehinović dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „Konvolucijske neuronske mreže za klasifikaciju objekata“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, 22. lipnja 2020. godine

Potpis

## SADRŽAJ

1. UVOD .....	1
2. UMJETNE NEURONSKE MREŽE.....	3
2.1 POVIJEST NEURONSKIH MREŽA .....	3
2.2 ARHITEKTURA NEURONSKIH MREŽA .....	4
2.2.1 BIOLOŠKI NEURON .....	4
2.2.2 UMJETNI NEURON .....	5
2.3 VRSTE AKTIVACIJSKIH FUNKCIJA .....	7
2.4 PROCES UČENJA NEURONSKIH MREŽA .....	13
2.4.1 UČENJE S UČITELJEM .....	14
2.4.2 UČENJE BEZ UČITELJA.....	15
2.5 FUNKCIJA TROŠKA.....	17
2.5.1 GRADIJENTI SPUST .....	19
2.6 ALGORITAM PROPAGACIJE POGREŠKE UNATRAG .....	20
3. KONVOLUCIJSKE NEURONSKE MREŽE .....	23
3.1 OPERACIJA KONVOLUCIJE .....	24
3.1.1 KONVOLUCIJA JEDNODIMENZIONALNIH SIGNALA .....	25
3.1.2 2D KONVOLUCIJA.....	28
3.2 FILTERI ZA PROCESUIRANJE SLIKA.....	32
3.3 MOTIVACIJA .....	35
3.4 ARHITEKTURA KONVOLUCIJSKIH NEURONSKIH MREŽA .....	37
3.4.1 KONVOLUCIJSKI SLOJ .....	38
3.4.2 SLOJ SAŽIMANJA.....	42
3.4.3 POTPUNO POVEZANI SLOJ.....	44
3.5 ALGORITAM PROPAGACIJE POGREŠKE UNATRAG .....	44
3.5.1 PROPAGACIJA UNATRAG U SLOJU KONVOLUCIJE .....	44
3.5.2 PROPAGACIJA UNATRAG U SLOJU SAŽIMANJA.....	48
3.6 PRIMJERI KONVOLUCIJSKIH NEURONSKIH MREŽA .....	49
4. PROBLEM KLASIFIKACIJE .....	55
4.1 HAM 10000 (HUMAN AGAINST MACHINE) SKUP PODATAKA .....	57
4.2 PRETPROCESIRANJE PODATAKA .....	62
4.2.1 ANALIZA PODATAKA .....	63
4.2.2 PRIPREMA PODATAKA ZA KLASIFIKACIJU .....	66

4.3 MODEL ZA KLASIFIKACIJU DERMATOSKOPSKIH SLIKA .....	68
4.3.1 MOBILE NET .....	68
4.3.2 FIPU MODEL ZA KLASIFIKACIJU DERMATOSKOPSKIH SLIKA .....	70
5. TESTIRANJE MODELA I PRIKAZ REZULTATA .....	73
5.1 BUDUĆA POBOLJŠANJA FIPU MODELA .....	75
6. ZAKLJUČAK .....	77
LITERATURA .....	78
POPIS SLIKA .....	81
POPIS TABLICA .....	84
SAŽETAK .....	85
ABSTRACT .....	86

## 1. UVOD

U današnjem vremenu sva automatizirana obrada podataka vrši se pomoću digitalnih računala. Čini se kao da računala obrađuju veliki broj podataka, ali taj broj je zapravo malen s obzirom na one podatke koje čovjek svakodnevno procesuiru u svom mozgu. Postoji mnogo problema čije rješenje nije lako predstaviti jednostavnim algoritmom. Takvi problemi ovise o više čimbenika, kao npr. prilikom kupovine nekretnina s obzirom na sve čimbenike ljudski mozak je u mogućnosti približno izračunati cijenu. Bez algoritma računalo ne može uraditi isto. Glavni razlog tome je što ljudski mozak ima sposobnost učenja koristeći se različitim strategijama i sposobnost donošenja ispravnih odluka na temelju logičkih pravila za razliku od računala. S druge strane, računalo ima mogućnost da rješava kompleksne numeričke kalkulacije u kratkom vremenu. Kada bi računalo imalo dio sposobnosti učenja kakvog ima čovjek, predstavljalo bi snažan alat koji može doprinijeti samom čovjeku u vremenu kada je ljudski mozak preopterećen informacijama. Dio umjetne inteligencije koji se naziva umjetne neuronske mreže radi upravo na sposobnosti učenja računala kako bi bilo u mogućnosti rješavati svakodnevne probleme. Problem klasifikacije (eng. classification problem) jedan je od mnogih problema s kojima se čovjek svakodnevno susreće. Primjer takvog problema jeste kupovina namirnica, s obzirom da zahtijeva klasifikaciju istih u nekoliko kategorija. Kako se čovjek svakodnevno suočava sa problemom klasifikacije namirnica, ovakav tip klasifikacije mu ne predstavlja težak izazov. Može se reći kako je ovakav tip klasifikacije nešto što je čovjek sposoban izvršavati bez prevelikog razmišljanja i napora. Međutim, nisu sve vrste problema klasifikacije jednostavne čovjeku. Postoje problemi čije rješavanje iziskuje previše vremena. U takvu klasu problema spada medicinska dijagnostika. Oblast medicine koja se naglo širi i zahtijeva pomoć umjetne inteligencije jeste dermatoskopija (eng. dermatoscopy). Kao posljedica sve većeg izlaganja suncu i štetnih UV zraka javlja se veliki porast oboljelih od kožnih oboljenja. Da bi se ustanovio tip i stupanj oboljenja pacijenti često prolaze kroz dugotrajan proces što u konačnici može rezultirati bržim napretkom same bolesti. Pacijent koji primijeti određenu nepravilnost na koži prvobitno mora posjetiti svog liječnika koji ga svojevrijem prosljeđuje na daljnje pretrage. Pretrage najčešće uključuju dermatoskopske slike kožnog oštećenja. Nakon što pacijent obavi neophodne pretrage, liječnik na osnovu dermatoskopskih slika klasificira oštećenje kože



u neku od kategorija kožnih oboljenja. Uloga liječnika u ovom slučaju je ključna, a liječnik je kao i svaki čovjek sklon greškama. Pogrešno klasificirano oboljenje utiče negativno na samog pacijenta, nerijetko dovodi i do kobnog ishoda. Kada bi računalo imalo mogućnost klasificiranja kožnih oboljenja doprinijelo bi velikoj pomoći liječnicima i pacijentima. Uloga liječnika i dalje bi ostala ključna, ali bi se skratilo vrijeme za dijagnostiku oboljenja. Dio neuronskih mreža koji se bavi klasifikacijom slika naziva se konvolucijske neuronske mreže.

U ovom radu obrađen je pojam konvolucijskih neuronskih mreža i njihova primjena na konkretan problem klasifikacije dermatoskopskih slika. U prvom dijelu diplomskog rada obrađen je pojam umjetnih neuronskih mreža, motivacija, arhitektura, proces učenja i najznačajniji algoritmi vezani za neuronske mreže. U drugom dijelu diplomskog rada detaljno su obrađene konvolucijske neuronske mreže, njihova motivacija, arhitektura, značajni algoritmi i neki od postojećih primjera konvolucijskih neuronskih mreža. U trećem dijelu diplomskog rada definiran je problem klasifikacije i predstavljena je praktična primjena konvolucijskih neuronskih mreža. Odabran je skup podataka koji se sastoji od dermatoskopskih slika koje je potrebno klasificirati u nekoliko različitih kategorija oboljenja. Podaci su detaljno analizirani i izvršeno je pretprocesiranje istih kako bi se kreirao model za klasifikaciju. Izgradnja modela prikazana je korak po korak, kao i rezultati klasifikacije. Na kraju trećeg dijela analizirani su rezultati modela za klasifikaciju i definirane ideje za buduća poboljšanja.

## 2. UMJETNE NEURONSKE MREŽE

Istraživanje umjetnih neuronskih mreža motivirano je njihovom sličnošću sa biološkim sustavom koji se sastoji od velikog broja nervnih ćelija. Paralelno procesuiranje podataka i sposobnost učenja najvažnija je odlika nervnih ćelija. U ovom poglavlju obrađene su sve važne karakteristike umjetnih neuronskih mreža koje služe za razumijevanje konvolucijskih neuronskih mreža i njihovu praktičnu primjenu.

### 2.1 POVIJEST NEURONSKIH MREŽA

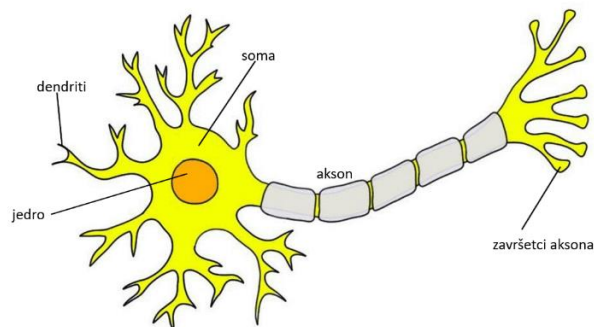
Povijest neuronskih mreža započinje u ranim 40-tim godinama prošlog stoljeća. Warren McCulloch i Walter Pitts (Massachusetts Institute of Technology) istražujući neurofiziološke karakteristike čovjeka predstavili su matematički model neuronske mreže u okviru teorije automata. Računala u tom periodu nisu imala sposobnost implementacije neuronskih mreža, zbog čega je nastao period stagnacije. Razvojem hardvera, 50-tih godina prošlog stoljeća započinje i razvoj umjetnih neuronskih mreža. Ovaj period nazvan je zlatno doba za koji se vežu sljedeći naučnici: Marvin Minsky koji je razvio neuro-računalo Snark, Frank Rosenblatt i Charles Wightman koji su skupa s drugim naučnicima razvili neuro-računalo Mark I perceptron, Bernard Widrom i Marcian E. Hoff koji su predstavili sustav ADALINE (ADaptive LInear Neuron) i drugi. Nakon perioda koji je nazvan zlatno doba, neuronske mreže pale su u zaborav i nastao je period tzv. šutnje koja je trajala približno 20 godina. Umjetne neuronske mreže doživjele su renesansu 90-tih godina prošlog vijeka. Od ovog perioda pa sve do danas, umjetne neuronske mreže doživljavaju eksplozivan rast.

## 2.2 ARHITEKTURA NEURONSKIH MREŽA

Arhitektura umjetnih neuronskih mreža motivirana je arhitekturom biološkog nervnog sustava koji se sastoji od velikog broja nervnih ćelija tzv. **neurona**, ali s određenim modifikacijama.

### 2.2.1 BIOLOŠKI NEURON

Postoji više od 100 vrsta bioloških neurona, raspoređenih po točno definiranom rasporedu. Ljudski mozak sastoji se od velikog broja neurona od kojih je svaki od njih povezan sinapsama sa tisućama drugih neurona što čini izuzetno kompleksnu arhitekturu neuronske mreže. Građa biološkog neurona prikazana je na sljedećoj slici.



Slika 1. Građa biološkog neurona

Svaki biološki neuron od susjednih neurona pomoću dendrita prima ulazne signale. Izlazne signale šalje drugim neuronima preko aksona. Završeci aksona povezani su sa dendritima drugih neurona. Prijenos informacija odvija se na sljedeći način: U tijelu jedne stanice sumiraju se izlazni potencijali tisuća susjednih neurona. Kada ukupni napon prijeđe određeni prag, neuron se pali i generira tzv. akcijski potencijal. Informacija se pomoću akcijskog potencijala prenosi preko završetaka aksona koji proizvode i otpuštaju kemikalije, tzv. **neurotransmitere**. Prijenos informacije je jednosmjernan.

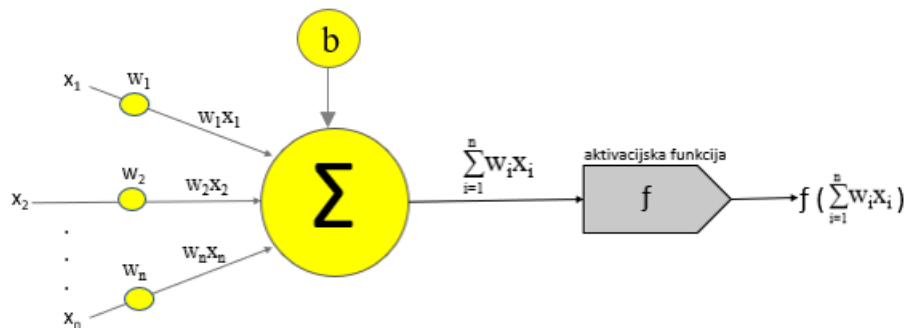
## 2.2.2 UMJETNI NEURON

Građa umjetnog neurona motivirana je građom biološkog neurona, zbog čega se prijenos informacija između neurona odvija na sličan način.

Umjetni neuron prima potencijale od susjednih neurona, s tim što umjesto funkcije praga ima neku drugu funkciju. Takvu funkciju nazivamo **aktivacijska funkcija**.

Označimo ulazne signale sa  $x_1, x_2, \dots, x_n$ , težine sinapse sa  $w_1, w_2, \dots, w_n$  i pomak (eng. bias) sa  $b$ . Kako su završeci aksona povezani sa dendritima drugih neurona, međusobna interakcija se predstavlja tako što se množe ulazni signali i težine sinapse. Svaki neuron na taj način računa težinsku sumu. Na dobivenu sumu primjenjuje se aktivacijska funkcija što u konačnici predstavlja izlazni signal. Neuroni se aktiviraju s obzirom na vrijednost izlaznog signala.

Na sljedećoj slici prikazana je građa umjetnog neurona i proces prijenosa signala.

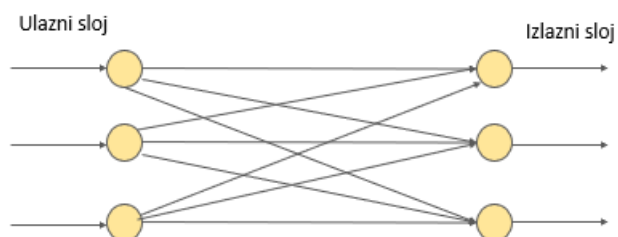


Slika 2. Građa umjetnog neurona

**Arhitektura umjetnih neuronskih mreža** sastoji se od umjetnih neurona koji su organizirani u slojeve. Na osnovu broja slojeva i smjera veze između neurona, definirana su tri osnovna tipa neuronskih mreža:

1. Jednoslojne mreže bez povratnih veza (eng. Single-Layer Feedforward Networks)
2. Višeslojne mreže bez povratnih veza (eng. Multilayer Feedforward Networks)
3. Mreže s povratnim vezama (eng. Recurrent Networks)

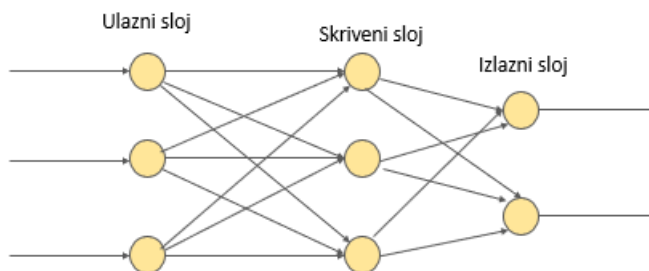
Najjednostavniji tip neuronske mreže je jednoslojna mreža bez povratnih veza. Kod ovakvog tipa mreže ulazni sloj sastoji se od čvorova koji propagiraju ulazne podatke na neurone izlaznog sloja. S obzirom da se na ulaznom sloju ne vrše nikakva računanja, ovaj sloj se ne računa u slojeve mreže. Računanja i transformacije ulaznih podataka vrše se na izlaznom sloju neuronske mreže. U ovakvom tipu neuronske mreže ne postoje povratne veze koje bi izlazni signal pretvorile u ulazni, zbog čega je ova mreža deklarirana kao mreža bez povratnih veza.



Slika 3. Jednoslojna mreža bez povratnih veza

Za razliku od jednoslojne mreže, višeslojna mreža bez povratnih veza osim ulaznog i izlaznog sloja sadrži i više skrivenih slojeva neurona.<sup>1</sup>

Na slici 4. prikazana je struktura višeslojne mreže bez povratnih veza koja se sastoji od 3 ulazna signala, 3 neurona u skrivenom sloju i 2 neurona u izlaznom sloju.



Slika 4. Višeslojna mreža bez povratnih veza

Mreža sa povratnim vezama (eng. Recurrent Network) nastaje kada se u jednosmjernu neuronsku mrežu uključi i povratna veza na način da se izlaz jednog neurona veže s ulazom drugog neurona ili više njih. Mreže s povratnim vezama

---

<sup>1</sup> (Nielsen, 2020) 10-11.str

mogu imati veze između neurona istog sloja, uključujući i povratnu vezu neurona u samog sebe (eng. self-feedback). Mreže s povratnim vezama daju određenu kvalitetu mreži, ali i znatno otežavaju proces učenja. Jedna od prednosti jeste mogućnost izračunavanja kompleksnijih funkcija zbog svog dinamičnog djelovanja. Mreže s povratnim vezama imaju mogućnost preslikavanja funkcija vremena, što znači da imaju mogućnost učenja niza uzoraka koji su vremenski ovisni. U konkretnim slučajevima primjene u kojima mreže s povratnim vezama daju dobre rezultate, odgovarajuća jednosmjerna mreža bi zahtijevala veći broj skrivenih neurona kako bi ostvarila jednako dobre rezultate kao i mreža s povratnim vezama.

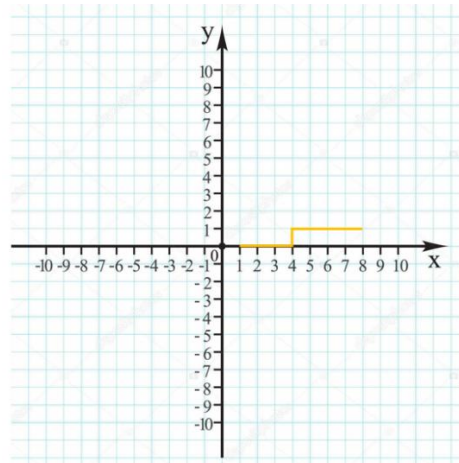
### 2.3 VRSTE AKTIVACIJSKIH FUNKCIJA

Postoji nekoliko vrsta aktivacijskih funkcija po kojima se mogu razlikovati umjetni neuroni. Neka je  $x = \sum x_i w_i$ , težinska suma svih ulaznih signala.

Prvi i najjednostavniji oblik aktivacijske funkcije je:  $f(x)=x$

U ovom slučaju izlazni signal jednak je težinskoj sumi ulaznih signala. Ovakav oblik aktivacijske funkcije koristio se u modelu umjetnog neurona ADALINE (ADaptive LInear Element).

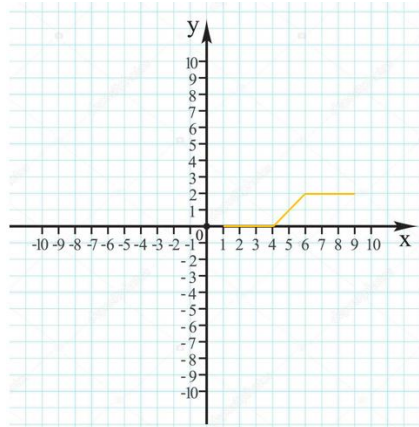
Drugi oblik aktivacijske funkcije je **funkcija skoka** (eng. treshold function, step function). Izlazni signal neurona koji koristi ovu aktivacijsku funkciju može biti 0 ili 1. Ukoliko je vrijednost izlaznog signala 1, neuron se aktivira, u protivnom ostaje neaktiviran. Graf funkcije skoka prikazan je na slici 5.



$$f(x) = \begin{cases} 0 & \text{za } x < 0 \\ 1 & \text{inače} \end{cases}$$

Slika 5. Funkcija skoka

Sljedeći oblik aktivacijske funkcije je **linearna funkcija**, pravolinijska funkcija kod koje je ulazni signal jednak težinskoj sumi ulaznih signala. Aktivacijska funkcija može biti definirana linearno po dijelovima. Graf linearne funkcije prikazan je na sljedećoj slici.



$$f(x) = \begin{cases} 0 & \text{za } x \leq a \\ x & \text{za } a < x < b \\ 1 & \text{za } x \geq b \end{cases}$$

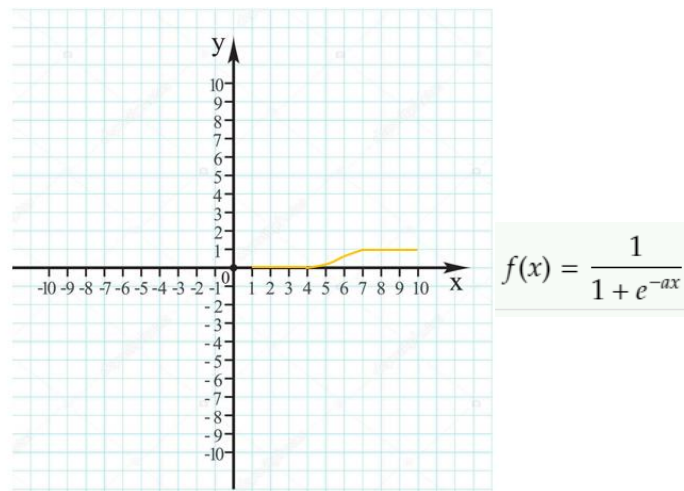
Slika 6. Linearna funkcija

U ovom slučaju izlazni signal neurona može biti 0, 1 ili  $y$ , što znači da postoji rang vrijednosti izlaznih signala. Moguće je spojiti više neurona i ukoliko ih je aktivirano više od jednog uzeti maksimalnu vrijednost i na taj način odlučivati. Problem linearne funkcije jeste što je derivacija jednaka konstanti (derivacija funkcije  $y=ax$  je  $a$ ). Gradijent ove funkcije je konstantan jer nema vezu sa  $x$ . Ukoliko dođe do greške u predikciji, promjene koje su nastale metodom propagacije unatrag (koja je objašnjena kasnije u ovom radu) su konstanta i ne ovise o ulaznom signalu  $x$ . Navedeni problem nije jedini nedostatak linearne aktivacijske funkcije.

Kada postoji više slojeva neuronske mreže i kada svi koriste linearnu aktivacijsku funkciju, posljednja aktivacijska funkcija je zapravo linearna funkcija ulaznih signala prethodnih slojeva.<sup>2</sup>

To znači da n slojeva neuronske mreže može biti zamijenjeno s jednim slojem zbog toga što je kombinacija linearnih funkcija i dalje linearna funkcija.

Najčešći oblik aktivacijske funkcije jeste **sigmoidna** funkcija. Prva prednost ove funkcije jeste što je nelinearna i što je kombinacija takvih funkcija također nelinearna funkcija. Graf sigmoidne funkcije prikazan je na slici 7.



Slika 7. Sigmoidna funkcija

Parametar  $a$  u ovom slučaju određuje nagib funkcije. Sigmoidna funkcija ponekad se naziva **logističkom funkcijom**. Rješavanje binarne klasifikacije definirano sigmoidnom funkcijom naziva se još i logistička regresija. Umjetni neuroni mogu se razvrstati prema vrsti ulaznih signala (realni brojevi ili Booleove vrijednosti) i prema obliku integrirajuće funkcije (ograničeno na razmatranje težinske sume). Ukoliko ulazni ili izlazni podaci nisu u uobičajenom intervalu  $[-1, 1]$ , potrebno je provesti pred-procesiranje odnosno post-procesiranje podataka što predstavlja linearno preslikavanje podataka u željeni interval.<sup>3</sup>

Sljedeća aktivacijska funkcija jeste **tahn** funkcija koja zapravo predstavlja skaliranu sigmoidnu funkciju. Karakteristike tahn funkcije su slične karakteristikama sigmoidne

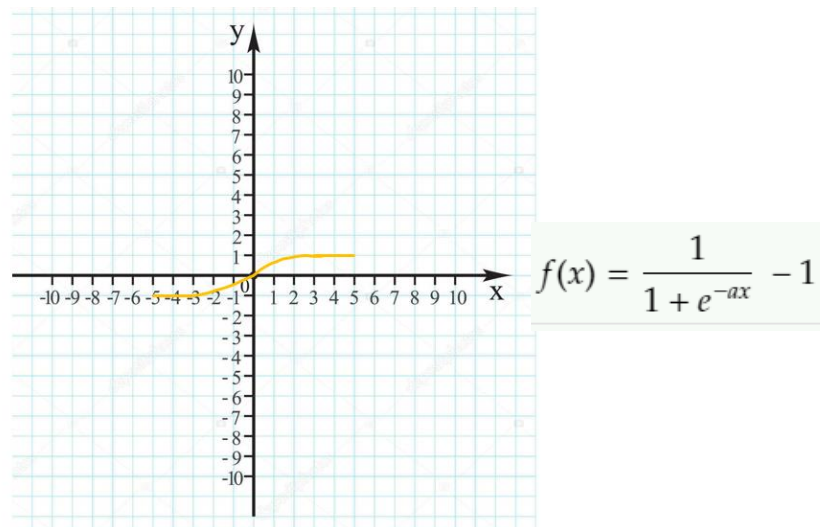
<sup>2</sup> (Sharma V, 2020)

<sup>3</sup> (Dalbello Bašić, Čupić i Šnajder, 2008) 10.str



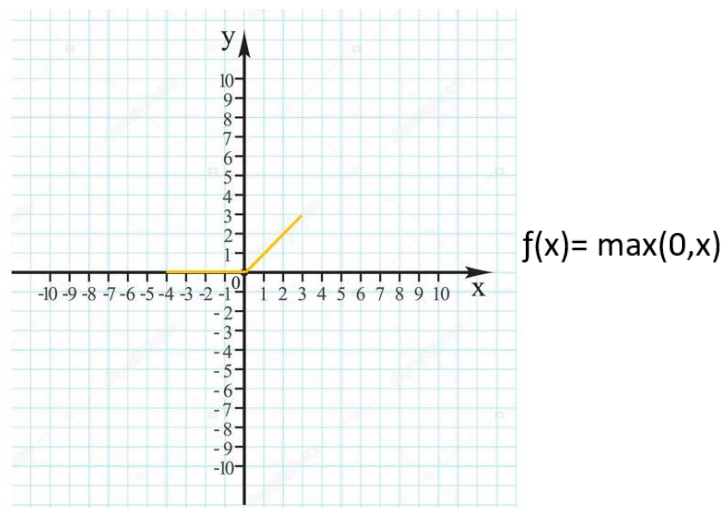
funkcije. Nelinearna je funkcija u kojoj se izlazni podatci nalaze u intervalu  $[-1,1]$ . Razlika između sigmoidne i tahn funkcije jeste u jačini gradijenta. Tahn funkcija ima jači gradijent od sigmoidne. Ova funkcija, kao i sigmoidna, veoma je popularna i u širokoj upotrebi.

Graf tahn funkcije prikazan je na slici 8.



Slika 8. Tahn funkcija

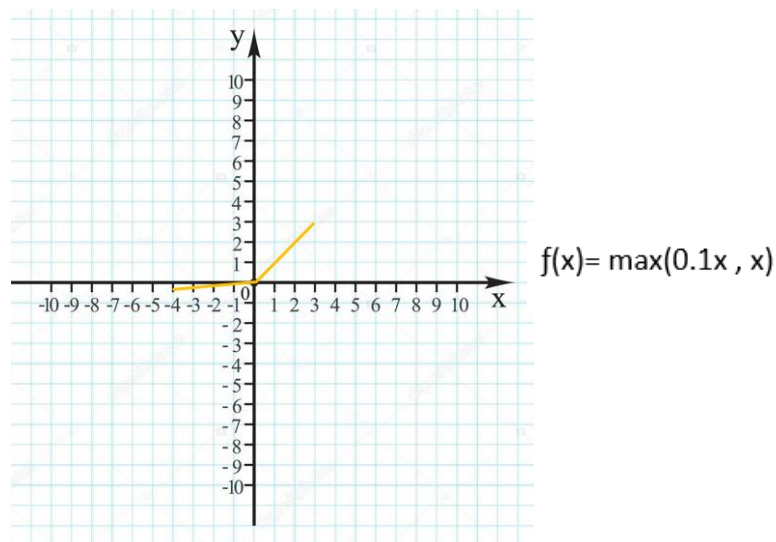
Još jedna važna aktivacijska funkcija jeste **ReLU** (Rectified Linear Unit) funkcija. Neka je  $x$  ulaz funkcije ReLU. Ako je  $x$  pozitivan, izlaz će biti  $x$ . Ako je  $x$  negativan, izlaz će biti 0. Graf i formula funkcije ReLU prikazani su na slici 9.



Slika 9. ReLU funkcija

Na prvi pogled izgleda kao da će funkcija ReLu imati iste probleme kao i linearna funkcija. Međutim, ReLu funkcija nije linearna i kombinacija više ReLu funkcija samim tim nije linearna i neće imati problem da se slaganje više slojeva može predstaviti kao jedan sloj. Derivacija ReLu funkcije nije konstanta, što implicira da je moguće primijeniti algoritam propagacije unatrag. Jedna od prednosti ReLu funkcije jeste što izbjegava da se većina neurona aktivira analogno, što čini aktivaciju gustom i skupom. ReLu funkcija rješava ovaj problem time što sve negativne vrijednosti  $x$  postaju 0. Na taj način manji broj neurona se aktivira i mreža postaje manje opterećena. ReLu funkcija ima i određene nedostatke. Jedan od njih jeste „Dying ReLu problem“, koji nastaje kada je ulaz u funkciju 0 ili negativan. Tada i gradijent funkcije postane 0 i nemoguće je primijeniti algoritam propagacije unatrag što doprinosi nemogućnost učenja mreže.

Kao rješenje Dying ReLu problema pojavila se funkcija **Leaky ReLu**. Graf i formula ove funkcije prikazani su na slici 10.



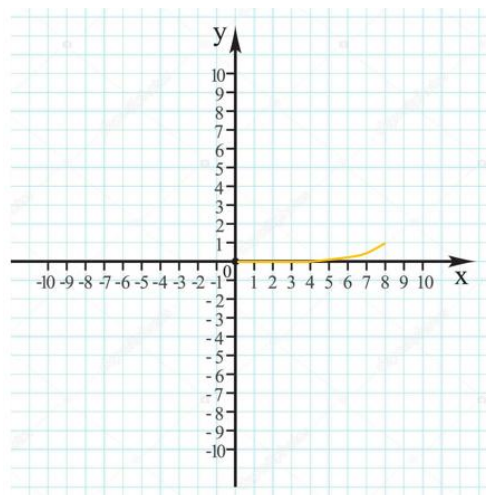
Slika 10. Leaky ReLu funkcija

Kao što je prikazano na prethodnoj slici, Leaky ReLu funkcija ima mali pozitivni nagib u negativnom dijelu y osi, zbog čega gradijent funkcije u tom slučaju nije 0 te je moguće primijeniti algoritam propagacije unatrag. Međutim, nedostatak ove funkcije

jeste to što rezultati dobiveni za negativne ulaze ne garantiraju konzistentne predikcije.<sup>4</sup>

Naredna aktivacijska funkcija je **Softmax** funkcija. Koristi se u modelu logističke regresije kod klasifikacije objekata u više klasa, za razliku od sigmoidne funkcije. Softmax funkcija kao rezultat daje vjerovatnosti koje su u rangu  $[0,1]$  i čija suma je jednaka 1. Izlaz softmax funkcije zapravo predstavlja vjerojatnost distribucije, odnosno vjerojatnost kojom jedan objekt pripada svakoj od ponuđenih klasa.

Na taj način moguće je odrediti kojoj klasi zapravo objekt pripada. Iako se softmax funkcija često koristi u posljednjem sloju neuronske mreže, može biti korištena i u drugim slojevima.<sup>5</sup>



Slika 11. Softmax funkcija <sup>6</sup>

Osnovne razlike između softmax i sigmoidne funkcije prikazane su na sljedećoj tablici.

---

<sup>4</sup> (Taguri, Erlichmen and Lussato, 2020)

<sup>5</sup> (How does the Softmax activation function work? – MachineCurve, 2020)

<sup>6</sup> (Depositphotos, 2020)

	SOFTMAX funkcija	SIGMOIDNA funkcija
1.	U modelu logističke regresije koristi se za klasifikaciju u <b>više kategorija</b>	U modelu logističke regresije koristi se za <b>binarnu</b> klasifikaciju
2.	Suma svih vjerojatnosti <b>mora</b> biti jednaka 1	Suma svih vjerojatnosti <b>ne mora</b> biti jednaka 1
3.	Koristi se u više slojeva neuronske mreže	Koristi se kao aktivacijska funkcija prilikom izrade neuronske mreže

Tablica 1. Razlike između softmax i sigmoidne funkcije

Ovo su bile neke od najvažnijih aktivacijskih funkcija sa svojim karakteristikama. Pitanje koje se često postavlja jeste: „Koju aktivacijsku funkciju koristiti?“. Ne postoji dobra ili loša aktivacijska funkcija. Postoji samo ona aktivacijska funkcija koja bolje ili lošije rješava određeni problem. Zbog toga prije odabira odgovarajuće aktivacijske funkcije potrebno je znati osnovne karakteristike svake od njih kao i sve karakteristike problema kojeg pokušavamo riješiti.<sup>7</sup>

## 2.4 PROCES UČENJA NEURONSKIH MREŽA

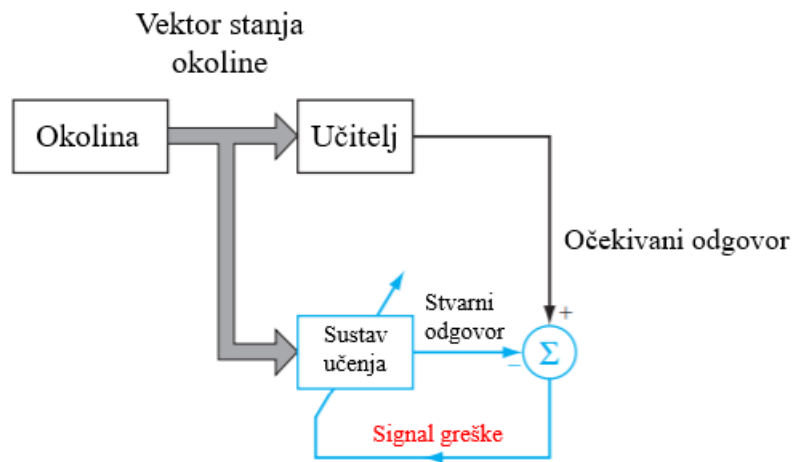
Proces učenja neuronskih mreža inspiriran je procesom učenja čovjeka. Čovjek svakodnevno na različite načine uči iz vlastite okoline koja ga okružuje. Proces učenja neuronskih mreža podrazumijeva iterativan postupak u kojem se predočavaju ulazni podaci sa ili bez odgovarajućih izlaznih podataka. Obzirom na to da li nam je u postupku učenja uz ulazne podatke poznat i izlaz, možemo kategorizirati proces učenja neuronskih mreža u dvije grupe: učenje s učiteljem i učenje bez učitelja.

---

<sup>7</sup> (Gharat, 2020)

## 2.4.1 UČENJE S UČITELJEM

Učenje s učiteljem naziva se još i nadzirano učenje (eng. supervised learning). Pod pojmom nadzirano učenje misli se na kontroliranje procesa učenja na osnovu prethodnog iskustva. Prethodna iskustva podrazumijevaju znanje o okolini iz koje neuronska mreža uči u obliku ulaznih i izlaznih podataka. Blok dijagram učenja s učiteljem prikazan je na slici 12.



Slika 12. Učenje s učiteljem

Okolina i vektor stanja okoline djeluju na sustav učenja i učitelja, odnosno učitelj i sustav učenja imaju informacije o vektoru stanja iz iste okoline. Učitelj na osnovu znanja iz okoline može definirati očekivane odgovore za željeni vektor. Očekivani odgovori predstavljaju optimalno rješenje koje neuronska mreža može postići. Sustav učenja s obzirom na utjecaj iz okoline definira skup sinaptičkih težina na odgovarajuće vrijednosti. Vektor ulaznih podataka i skup sinaptičkih težina kreiraju izlaznu vrijednost u obliku stvarnog odgovora.

Signal greške predstavlja razliku između očekivanog odgovora i stvarnog odgovora. Ovaj proces ponavlja se iterativno kako bi formirao neuronsku mrežu koja emulira učitelja, s ciljem da neuronska mreža omogući optimalne rezultate. Na ovaj način se znanje iz okoline koje je dostupno učitelju prenosi na neuronsku mrežu treniranjem podataka i

njihovim arhiviranjem u formi fiksnih sinaptičkih težina koje predstavljaju stalnu memoriju. Kada se ovakvo stanje dostigne, postoji mogućnost da se neuronska mreža sama (bez učitelja) nosi sa okolinom. Nadzirano učenje predstavlja osnovu učenja korekcijom pogreške.<sup>8</sup>

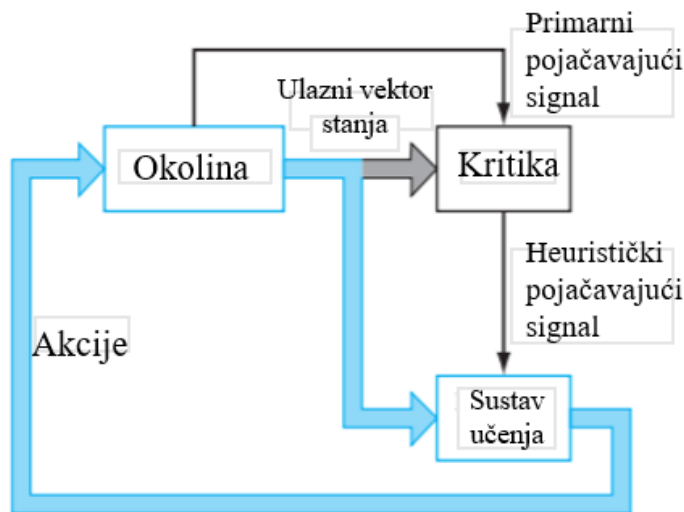
## 2.4.2 UČENJE BEZ UČITELJA

U nadgledanom učenju proces učenja odvija se pod nadzorom učitelja. Paradigma učenja bez učitelja, kao što samo ime govori, implicira na proces učenja koji se odvija bez nadzora učitelja. Ovo znači da ne postoji vektor stanja okoline s točno određenim ulaznim i izlaznim veličinama, odnosno ne postoje primjeri iz okoline koje neuronska mreža treba naučiti.

Proces učenja bez učitelja razlikuje dvije kategorije: pojačano i nenadzirano učenje.

### 1. Sustav pojačanog učenja

Blok dijagram pojačanog učenja prikazan je na slici 13.



Slika 13. Pojačano učenje

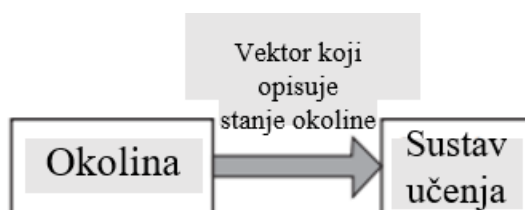
<sup>8</sup> (Haykin, 2009) 36.str

Na slici 13. vidimo da se proces pojačanog učenja odvija u zatvorenoj povratnoj petlji. Osnovna razlika između blok dijagrama učenja sa učiteljem i pojačanog učenja jeste okolina koja je u ovom slučaju dio petlje. Ovakav sustav učenja bez učitelja izgrađen je oko kritičara (eng. critic) koji pretvara primarni pojačani signal primljen iz okoline u kvalitetniji signal pojačanja koji se naziva heurističkim. Sustav pojačanog učenja generira izlazni signal kao kombinaciju ulaznog vektora stanja okoline i heurističkog pojačavajućeg signala, koji se ponovo vraća u petlju procesa. Cilj ovakvog učenja je minimiziranje cost-to-go funkcije koja je definirana kao očekivanje kumulativnog troška akcija obrađenih slučajeva, kroz sekvencu koraka učenja, umjesto jednostavnog pojedinačnog troška slučaja. Proces pojačanog učenja ima nedostatke iz slijedećih razloga: Prvi razlog je nepostojanje učitelja koji doprinosi tome da je željeni odgovor dostupan u svakom koraku. Drugi razlog je to što sustav mora dodijeliti zasluge i okriviti pojedinačno svaku akciju u vremenskom slijedu koraka koji su doveli do konačnog ishoda, dok primarno pojačanje može samo procijeniti ishod.<sup>9</sup>

Bez obzira na navedene nedostatke, ovakav sustav učenja je veoma zanimljiv zbog mogućnosti interakcije sa okolinom.

## 2. Nenadzirano učenje

Blok dijagram nenadziranog učenja prikazan je na slici 14.



Slika 14. Nenadzirano učenje

Na slici 14. vidimo da se proces nenadziranog učenja odvija bez učitelja i kritičara koji nadziru proces učenja. Za izvođenje ovakvog procesa učenja koristi se natjecateljsko pravilo. Na primjer, moguće je koristiti neuronsku mrežu koja se sastoji od dva sloja:

---

<sup>9</sup> (Haykin, 2009) 37-38.str

ulaznog i natjecateljskog. Ulazni sloj prima dostupne podatke. Natjecateljski sloj sastoji se od neurona koji se natječu jedni s drugima u skladu s pravilima učenja za odgovor na značajke sadržane u ulaznim podacima. U najjednostavnijem obliku mreža funkcionira u skladu sa strategijom „pobjednik-dobiva-sve“. U takvoj strategiji, neuron s najvećim ukupnim brojem ulaza pobjeđuje konkurenciju i uključuje se, dok se svi ostali neuroni u mreži tada isključuju.<sup>10</sup>

## 2.5 FUNKCIJA TROŠKA

Važnu ulogu kod dizajna umjetne neuronske mreže ima funkcija troška (eng. cost function) koja se još naziva i funkcija gubitka (eng. loss function). Funkciju troška koristimo kako bismo mogli procijeniti nezadovoljstvo izlaznim rezultatom klasifikacije. Prethodno je opisano da su težine  $W$  i pomaci  $b$  parametri neuronske mreže na koje algoritam može utjecati dok su ulazni podaci oni na koje algoritam nema utjecaj. Ono što želimo jeste algoritam koji će utjecati na parametre  $W$  i  $b$ , odnosno na težine i pomake tako da minimiziramo funkciju troška. Neka je izlazni rezultat predstavljen kao  $f(x_i, W)$ , stvarne vrijednosti kao  $y_i$  i ulazni podaci kao  $x_i$ . Funkcija troška definira se kao odstupanje izlaznog rezultata od stvarne vrijednosti. Trošak, odnosno gubitak bit će veći kada pogrešno klasificiramo objekt. Postoji više vrsta funkcije troška. U nastavku će biti definirane neke od najčešće korištenih.

Prva funkcija troška koju ćemo definirati jeste **kvadratna funkcija troška**, poznata još kao srednje-kvadratna greška (eng. mean squared error):

$$C(w, b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2$$

U prethodnom izrazu  $w$  predstavlja skup svih težina u mreži,  $b$  pomake,  $n$  ukupni broj ulaza,  $a$  je vektor izlaza iz mreže kada je  $x$  ulaz. Možemo zaključiti da je  $C(w, b)$  ne-negativno s obzirom da je svaki uvjet u sumi ne-negativan. Također,  $C(w, b)$  predstavlja

---

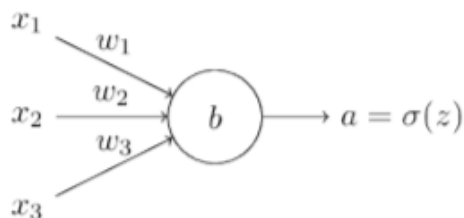
<sup>10</sup> (Haykin, 2009) 37-38.str



malu vrijednost, približno jednaku nuli kada je  $y(x)$  približno jednako vrijednosti  $a$ , za sve vrijednosti  $x$ . Reći ćemo da je algoritam za treniranje uspješan ukoliko može pronaći vrijednosti  $w$  i  $b$  tako da je  $C(w, b) \approx 0$ . Obratno, algoritam za treniranje nije uspješan ukoliko je vrijednost  $C(w, b)$  velika.<sup>11</sup>

Cilj je pronaći skup težina i pomaka takvih da je trošak što manji. Pitanje koje se postavlja jeste zašto minimizirati funkciju troška umjesto direktnog maksimiziranja izlaza (npr. broj klasificiranih slika)? Problem kod maksimiziranja broja točno klasificiranih slika jeste što taj broj ne predstavlja funkciju težina i pomaka u neuronskoj mreži. U većini slučajeva male promjene na težini i pomaku ne uzrokuju promjene u broju točno klasificiranih slika. U tom slučaju znatno nam je teže promijeniti težine i pomak kako bi dobili bolje rezultate. Ako umjesto toga koristimo funkciju troška postaje nam jednostavnije napraviti male promjene na težinama i pomaku kako bismo smanjili trošak. Zbog toga je primarni fokus na minimiziranju troška.

**Unakrsna entropija** (eng. cross-entropy) je sljedeća važna funkcija troška. Pretpostavit ćemo da želimo trenirati neuron sa nekoliko ulaznih signala  $x_1, x_2, \dots, x_n$ , odgovarajućim težinama  $w_1, w_2, \dots, w_n$  i pomakom  $b$ .



Slika 15. Umjetni neuron

Na slici 15. vidimo da je  $a = \sigma(z)$  izlazni signal neurona gdje  $z = \sum_i w_i x_i + b$  predstavlja težinsku sumu ulaznih signala. Definiramo funkciju unakrsne entropije za ovaj neuron na sljedeći način:

$$C = -\frac{1}{n} \sum [y \ln a + (1 - y) \ln(1 - a)] ,$$

<sup>11</sup> (Nielsen, 2020) 42.str

gdje  $n$  predstavlja ukupni broj podataka za treniranje a  $y$  željeni izlaz za odgovarajući ulaz  $x$ . Vrijednost  $C$  je ne-negativan s obzirom da su svi čimbenici u sumi negativni (oba logaritma su brojevi od 0 do 1) a ispred sume stoji znak minus, što vrijednost  $C$  čini pozitivnim. Ukoliko je izlaz neurona sličan željenom izlazu za sve trenirane ulaze  $x$ , tada će funkcija unakrsne entropije biti blizu nule. Željeni izlaz u ovom slučaju je 0 ili 1, što unakrsnu entropiju čini pogodnom za problem binarne klasifikacije. Dakle, unakrsna entropija zadovoljava osnovne uvjete za funkciju troška: pozitivna je i teži ka nuli s poboljšanjem izlaza  $y$ , za sve trenirane ulaze  $x$ . Kvadratna funkcija troška koja je prethodno opisana također zadovoljava ova dva osnovna uvjeta za funkciju troška. Razlika između kvadratne funkcije troška i unakrsne entropije jeste to što unakrsna entropija nema problem s usporavanjem učenja kao kvadratna funkcija.

### 2.5.1 GRADIJENTI SPUST

Ideja učenja neuronske mreže jeste pronaći skup težina i pomaka tako da trošak bude minimalan. Takav rezultat moguće je postići s algoritmom **gradijentnog spusta**. Gradijentni spust kao cilj ima optimizaciju odnosno minimiziranje funkcije troška. Prethodno je opisano da parametri mreže na koje algoritam može utjecati su težine  $W$  i pomaci  $b$ , dok ulazni podaci ( $x$  i  $y$ ) ostaju nepromijenjeni. Gradijent možemo definirati kao promjenu težine u odnosu na promjenu pogreške. Gradijentom možemo opisati nagib funkcije. Što je gradijent veći, nagib funkcije je strmiji i model može učiti brže. U matematičkom smislu gradijent predstavlja parcijalnu derivaciju s obzirom na ulazne podatke. Algoritam gradijentnog spusta funkcionira na način da se u svakoj iteraciji parametri modela ažuriraju s ciljem pada funkcije troška. Metodu ažuriranja gradijentnog spusta moguće je opisati na slijedeći način:

$$w(k+1) = w(k) - \Delta w(k) ,$$

gdje  $w(k)$  predstavlja vrijednost  $w$  u iteraciji  $k$  tokom gradijentnog spusta.

$$\Delta w = \mu \frac{\partial C}{\partial w}$$

U prethodnoj jednakosti  $\mu$  opisuje koliko veliki koraci se izvode tokom gradijentnog spusta, a  $\partial C/\partial w$  predstavlja gradijent funkcije troška  $C$  u odnosu na težinu  $w$ . Za svaki uzorak  $i$  gradijent možemo raspisati kao:

$$\frac{\partial C_i}{\partial w} = \frac{\partial C_i}{\partial y_i} \frac{\partial y_i}{\partial w}$$

gdje je  $C$  kvadratna funkcija troška, tako da dio  $\partial C_i/\partial y_i$  može biti raspisan kao:

$$\frac{\partial C_i}{\partial y_i} = \frac{\partial (t_i - y_i)^2}{\partial y_i} = -2(t_i - y_i) = 2(y_i - t_i)$$

Sada,  $\Delta w$  možemo raspisati kao:

$$\Delta w = \mu \frac{\partial C}{\partial w} = \mu 2x_i(y_i - t_i)$$

U konačnici, dodajemo gradijente za svaki uzorak na slijedeći način:

$$\Delta w = \mu * 2 * \frac{1}{N} \sum_{i=1}^N x_i(y_i - t_i)$$

## 2.6 ALGORITAM PROPAGACIJE POGREŠKE UNATRAG

Prethodno je opisano kako pomoću gradijenta optimizirati odnosno minimizirati funkciju troška. Pitanje koje se postavlja jeste kako izračunati gradijent funkcije troška? Za računanje gradijenta funkcije troška koristi se algoritam propagacije greške unatrag. Algoritam su 1970-tih godina predstavili Widrow i Hoff ali popularnost je doživio 1986. godine zahvaljujući Rumelhartu, Hintonu i Williamsu. U svom radu ovi naučnici opisali su neuronske mreže u kojima se algoritam propagacije pogreške unatrag izvršava znatno brže nego u dotadašnjim istraživanjima. Na taj način postalo je moguće riješiti probleme koji su do tada bili nerješivi. Danas, algoritam propagacije pogreške unatrag je neizostavan dio kod učenja neuronskih mreža. Korake u algoritmu propagacije pogreške unatrag možemo opisati na sljedeći način:

Za početak potrebno je označiti sve ulazne čvorove koji sadrže ulazne varijable u odgovarajućem redoslijedu i propagirati ih unaprijed kroz slojeve mreže prema izlaznom

sloju. Zatim, potrebno je zapisivati vrijednosti izlaznih čvorova i izračunati grešku svakog izlaznog čvora. Na kraju se vrši propagacija greške unatrag koja polazi od izlaznog sloja i vrši promjenu sinaptičkih težina s ciljem minimiziranja greške.

Nekoliko matematičkih izraza bit će prikazano i objašnjeno prije definiranja algoritma propagacije unatrag s ciljem lakšeg razumijevanja:

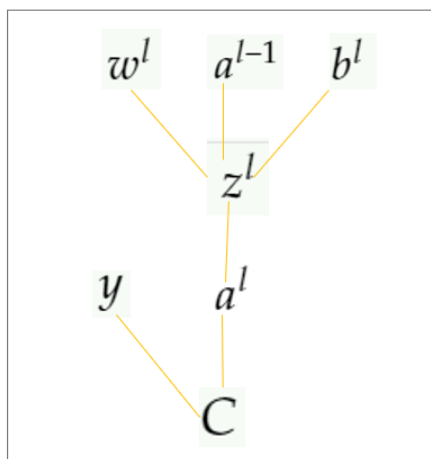
- $\frac{\partial C}{\partial w_{jk}^l}$  i  $\frac{\partial C}{\partial b_j^l}$  predstavljaju parcijalne derivacije kojima matematički izražavamo promjenu težine i pomaka u neuronskoj mreži i način na koji promjena utječe na funkciju troška
- $\Delta z_j^l$  prikazuje promjenu u j-tom neuronu u l-tom sloju neuronske mreže
- $\delta_j^l \equiv \frac{\partial C}{\partial z_j^l}$  prikazuje grešku u j-tom neuronu u l-tom sloju neuronske mreže
- $\delta_j^l = \frac{\partial C}{\partial a_j^l} \sigma'(z_j^l)$  prikazuje jednakost kojom se računa greška u izlaznom sloju. Prvi dio jednakosti prikazuje koliko brzo se trošak mijenja kao funkcija aktivacije j-tog neurona. Drugi dio jednakosti mjeri koliko brzo se mijenja aktivacijska funkcija. Jednakost se može zapisati u matricnoj formi na slijedeći način:  $\delta^L = \nabla_a C \odot \sigma'(z^L)$
- $\delta^l = \left( (w^{l+1})^T \delta^{l+1} \right) \odot \sigma'(z^l)$  prikazuje jednakost pomoću koje se računa greška l-tog sloja pod uvjetima greške u l+1-om sloju
- $\delta_j^l = \frac{\partial C}{\partial b_j^l}$  prikazuje promjenu troška u odnosu na bilo koji pomak u mreži
- $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$  promjenu troška u odnosu na bilo koju težinu u mreži

#### Algoritam 1: Propagacija greške unatrag

1. Ulaz x: Postaviti odgovarajuće aktivacije za ulazni sloj  $a^l$
2. Propagacija unaprijed: Za svaki  $l=2,3,\dots,L$  izračunati  $z^l = w^l a^{l-1} + b^l$  i  $a^l = \sigma(z^l)$
3. Greška  $\delta^L$ : Izračunati vektor  $\delta^L = \nabla_a C \odot \sigma'(z^L)$
4. Propagacija greške unatrag: Za svaki  $l=L-1,L-2,\dots,2$  izračunati  $\delta^l = \left( (w^{l+1})^T \delta^{l+1} \right) \odot \sigma'(z^l)$
5. Izlaz: Gradijent funkcije troška  $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$  i  $\frac{\partial C}{\partial b_j^l} = \delta_j^l$

Funkcija troška predstavlja funkciju izlaznih signala neuronske mreže, zbog čega se greška propagira kroz mrežu unatrag. Da bismo razumjeli kako se trošak mijenja sa

prethodnim težinama i pomacima potrebno je primjenjivati pravilo lanca, ponavljajući ga od krajnjeg sloja prema početnom.<sup>12</sup>



Slika 16. Pravilo lanca

---

<sup>12</sup> (Nielsen, 2020) str. 43-50

### 3. KONVOLUCIJSKE NEURONSKE MREŽE

Konvolucijske neuronske mreže (eng. Convolutional neural networks ili CNN) predstavljaju specijaliziran tip neuronskih mreža za procesiranje podataka koji se mogu svrstati u kategoriju mrežaste topologije (eng. grid-like topology). Izuzetno su uspješne u praktičnim primjenama kod nestrukturiranih podataka, naročito slika, teksta, zvuka i govora. Naziv „konvolucijske neuronske mreže“ potiče od matematičke operacije zvane konvolucija. Konvolucija predstavlja specijalizirani tip linearne operacije, koja će biti posebno opisana u ovom poglavlju. Konvolucijske neuronske mreže moguće je definirati kao klasične neuronske mreže koje primjenjuju operaciju konvolucije na barem jednom sloju.

Povijest konvolucijskih neuronskih mreža datira još iz srednjih godina 20. stoljeća. Značajnim otkrićima doprinijeli su Hubel i Wiesel radom o vizualnom procesiranju podataka za koji im je dodijeljena Nobelova nagrada iz područja medicine 1981. godine. Hubel i Wiesel pratili su aktivnost pojedinih neurona u mozgu mačke, neurona vizualnog korteksa. Preko mrežnice neurona vizualnog korteksa mačke vršili su kretanje svjetlom linijom prilikom čega su zabilježili slijedeće: neuroni su bili aktivirani samo kada je linija bila u određenom mjestu mrežnice, aktivacija neurona ovisila je o orijentaciji linije i ponekad su neuroni bili aktivirani samo onda kada se linija kretala u određenom smjeru. Eksperiment Hubela i Wiesela predstavlja osnove razumijevanja o tome kako neuroni izvlače složene informacije iz uzorka svjetlosti koji se nalazi na mrežnici kako bi konstruirali sliku. Svojim radom dokazali su da su neuroni u vizualnom korteksu organizirani u preciznoj arhitekturi tako što su stanice sa sličnim funkcijama organizirane u stupce, koji informaciju prosljeđuju na više nivoa mozga gdje se vizualna slika formira.<sup>13</sup> Rad Hubela i Wiesela predstavljao je inspiraciju za daljnje razvijanje konvolucijskih neuronskih mreža. U ovom poglavlju bit će opisana operacija konvolucije kao i motivacija za primjenu operacije konvolucije u neuronskim mrežama. Definirat će se nekoliko varijanti operacije konvolucije koji su najčešće korišteni u praktičnom smislu neuronskih mreža. Nakon operacije konvolucije, bit će opisana operacija poliranja (eng. pooling) koju

---

<sup>13</sup> (Fehlhaber, 2020)

skoro sve konvolucijske mreže primjenjuju. Također, bit će opisana i arhitektura mreže sa opisom svakog sloja. Nakon arhitekture bit će definiran proces učenja konvolucijske neuronske mreže, kao i tehnike optimizacije i regularizacije mreže.

### 3.1 OPERACIJA KONVOLUCIJE

Konvolucija predstavlja operaciju na dvije funkcije s realnim domenama. Prije definiranja konvolucije bit će definirani primjeri funkcije na koje se konvolucija može primijeniti. Primjer funkcije na koji možemo primijeniti operaciju konvolucije jeste signal. Konvolucija bilo kojeg signala s drugim signalom producira treći signal koji je u mogućnosti da prikazuje više informacija o prvom signalu nego sami prvi signal. Modificirani signal može predstavljati bolju opciju za određeni zadatak nego inicijalni signal. Recimo, ako primijenimo operaciju konvolucije na 2D signal koji predstavlja crno-bijelu sliku sa drugim signalom, zvanim filter ili jezgra, izlazni signal može sadržavati rubove originalne slike koji mogu biti korisni u određenim aplikacijama. Za razumijevanje operacije konvolucije potrebno je znanje o linearnim vremenski invarijantnim sustavima (eng. Linear time invariant system ili LTI) i linearnim sustavima invarijantnima na pomak (eng. Linear shift invariant system ili LSI).

Neka je ulazni signal označen sa  $x(t)$ , a izlazni signal sa  $y(t)$ . Izlazni signal  $y(t)$  izražavamo kao:

$$y(t) = f(x(t))$$

Da bi sustav zadovoljavao uvjet linearnosti, potrebno je da ispunjava sljedeća svojstva:

Skaliranje:  $f(\alpha x(t)) = \alpha f(x(t))$

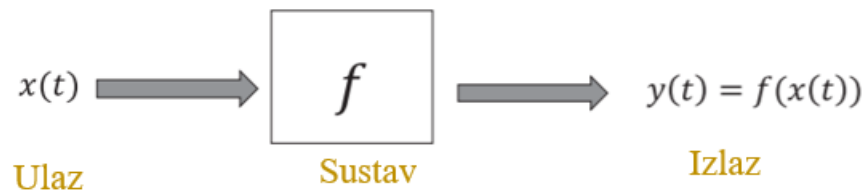
Superpozicija:  $f(\alpha x_1(t) + \beta x_2(t)) = \alpha f(x_1(t)) + \beta f(x_2(t))$

Slično, da bi sustav zadovoljavao uvjet vremenske invarijantnosti, potrebno je da ispunjava svojstvo:  $f(x(t - \tau)) = y(t - \tau)$  <sup>14</sup>

Ulazni-izlazni sustav kao skica prikazan je na slici 17.

---

<sup>14</sup> (Pattanayak, 2018) 153-154.str



Slika 17. Ulazni-izlazni sustav

### 3.1.1 KONVOLUCIJA JEDNODIMENZIONALNIH SIGNALA

Operaciju konvolucije moguće je predstaviti kao stupanj preklapanja između jedne funkcije i zrcalno okrenute i translahirane druge funkcije. Operator konvolucije označava se sa  $*$ .

Konvoluciju vremenski diskretnih funkcija  $x$  i  $h$  definiramo na sljedeći način:

$$y(t) = \sum_{\tau=-\infty}^{+\infty} x(\tau) h(t - \tau)$$

Ukratko, konvolucija funkcija  $x$  i  $h$  označava se kao:  $y(t) = x(t) * h(t)$

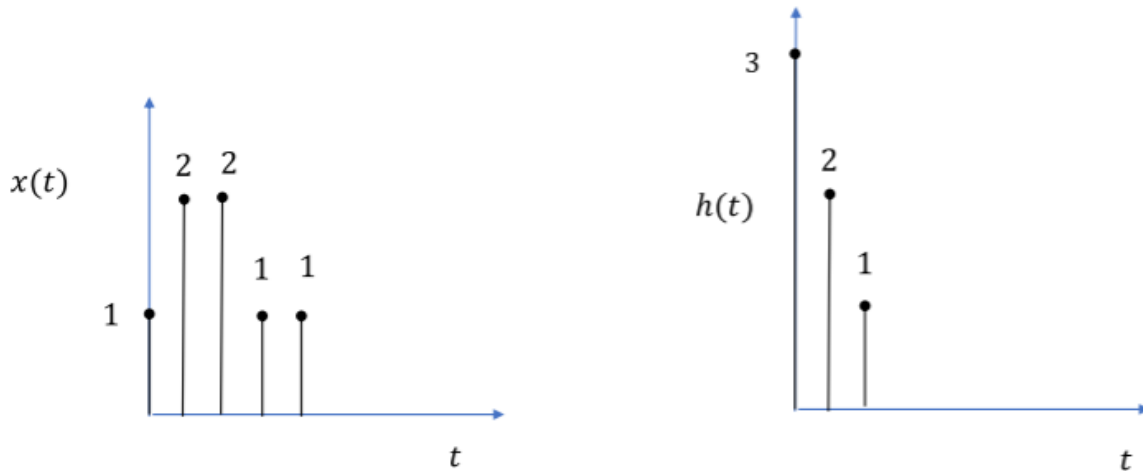
Slično, konvoluciju vremenski neprekidnih funkcija  $x$  i  $h$  možemo izraziti kao:

$$y(t) = x(t) * h(t) = \int_{\tau=-\infty}^{+\infty} x(\tau) h(t - \tau) d\tau$$

Za sustav možemo reći da je vremenski neprekidan ako na njemu djeluju funkcije koje su definirale na skupu realnih brojeva, ili na nekom drugom neprekidnom podskupu realnih brojeva. Na sljedećem primjeru bit će primijenjena operacija konvolucije na dva diskretna signala  $x$  i  $h$  koja ovise od vremena  $t$ .<sup>15</sup>

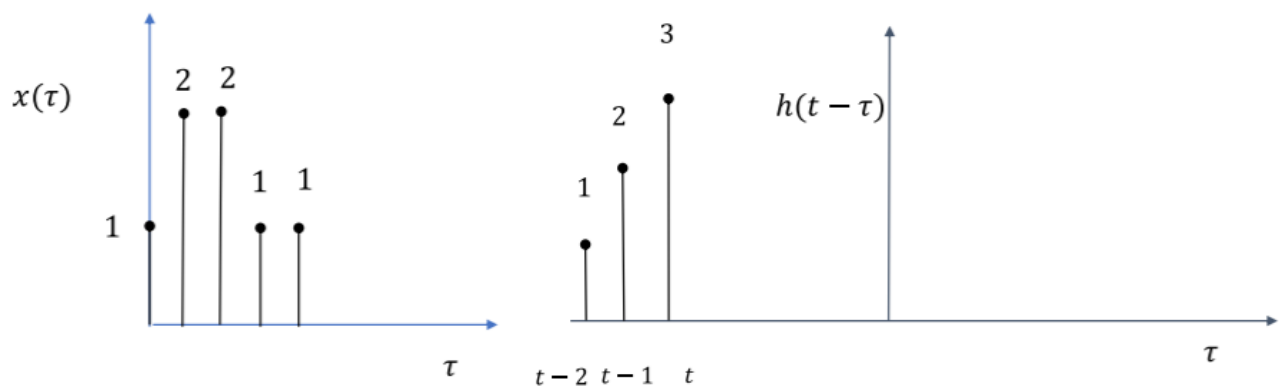
<sup>15</sup> (Pattanayak, 2018) 155-156.str





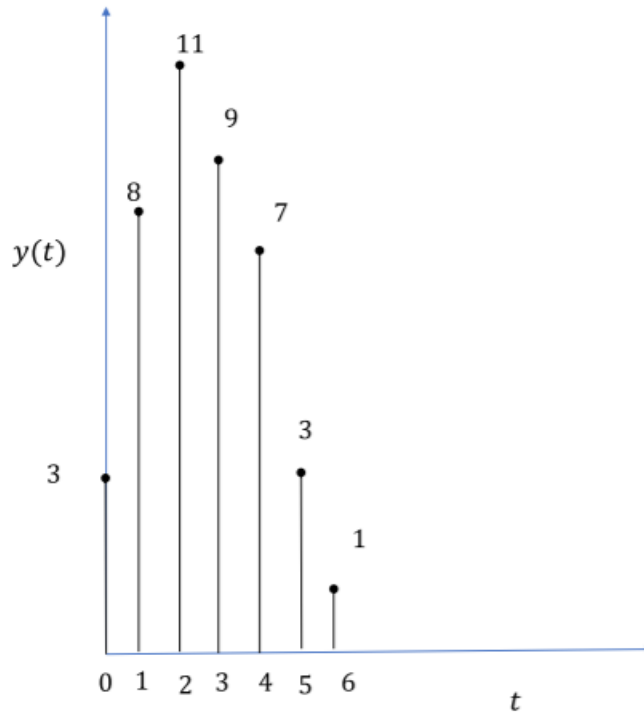
Slika 18. Ulazni signali

Na slici 19. prikazane su funkcije za računanje operacije konvolucije.



Slika 19. Funkcije za računanje operacije konvolucije

Rezultat operacije konvolucije prikazan je na slici 20.



Slika 20. Izlazni signal  $y(t)$

Na slici 20. funkciju  $h(t-\tau)$  računamo za različite vrijednosti  $t$  kojeg pomjeramo po horizontalnoj osi. Za svaku vrijednost  $t$  računa se konvolucijska suma  $\sum_{\tau=-\infty}^{+\infty} x(\tau) h(t-\tau)$ .

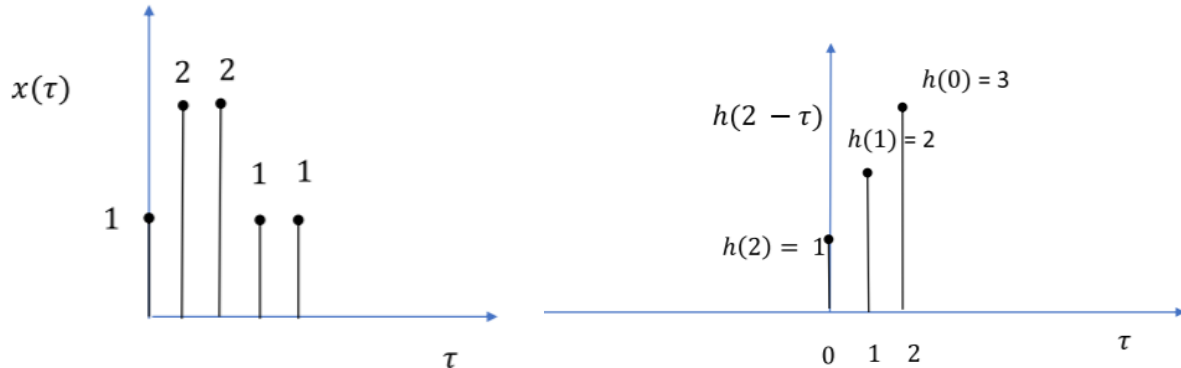
Sumu možemo smatrati težinskim prosjekom od  $x(\tau)$  sa težinama od  $h(t-\tau)$  :

- Kada je  $t=-1$  težina je dana sa  $h(1-\tau)$ , ali nema preklapanja sa  $x(\tau)$  pa je suma 0.
- Kada je  $t=0$  težine su dane sa  $h(-\tau)$  i jedini element  $x(\tau)$  koji se preklapa s težinama je  $x(\tau=0)$ , s preklapajućom težinom  $h(0)$ . Konvolucijska suma u tom slučaju je:  $x(\tau=0) * h(0)=1*3=3$ . Dakle,  $y(0)=3$ .
- Kada je  $t=1$  težine su dane sa  $h(1-\tau)$ . Elementi  $x(0)$  i  $x(1)$  su u preklapanju sa težinama  $h(1)$  i  $h(0)$  respektivno. Konvolucijska suma u tom slučaju je:  $X(0)*h(1) + x(1)*h(0)=1*2 + 2*3=8$ .

- Kada je  $t=2$  težine su dane sa  $h(2-\tau)$ . Elementi  $x(0)$ ,  $x(1)$ , i  $x(2)$  su u preklapanju s težinama  $h(2)$ ,  $h(1)$  i  $h(0)$  respektivno. Konvolucijska suma u tom slučaju je:  

$$X(0)*h(2) + x(1)*h(1) + x(2)*h(0)=1*1+2*2+2*3=11.$$

Na slici 21. prikazano je preklapanje funkcija u konvoluciji za vrijednost  $t=2$ .



Slika 21. Preklapanje funkcija u konvoluciji za  $t=2$

Generalno, sve što prikazuje varijacije u vremenu i/ili prostoru reprezentira signal. Stoga, signal je funkcija vremena i/ili prostora. Signali mogu biti analogne i digitalne prirode. Međutim, računalo ne može obrađivati analogni signal, stoga se svaki takav mora pretvoriti u digitalni. Primjerice, govor je akustični signal. Kada se govor prenosi preko mikrofona, kontinuirani akustični signal pretvara se u električni kontinuirani signal. Također, ako želimo procesuirati analogni električni signal u računalo, potrebno je da pretvorimo analogni kontinuirani signal u diskretni signal. Ovo se postiže metodom uzorkovanja i kvantizacijom analognog signala. Uzorkovanje se odnosi na uzimanje amplitude signala u fiksnom, određenom vremenskom intervalu.<sup>16</sup>

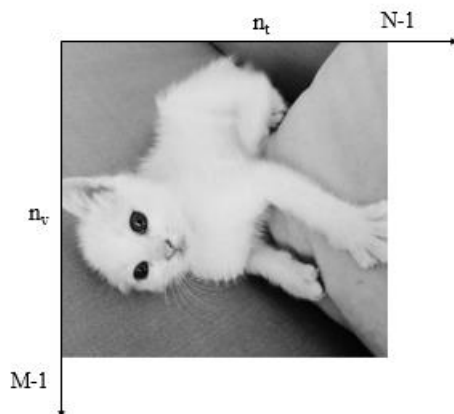
### 3.1.2 2D KONVOLUCIJA

Svaka crno-bijela slika dimenzija  $N \times M$  može se izraziti kao skalar 2D signala njezinih prostornih koordinata. Signal je moguće predstaviti kao:

$$x(n_v, n_t), \quad 0 < n_v < M-1, \quad 0 < n_t < N-1,$$

<sup>16</sup> (Pattanayak, 2018) 155-158.str

gdje su  $n_v$  i  $n_t$  diskretne prostorne koordinate na horizontalnoj i vertikalnoj osi respektivno, dok  $x(n_v, n_t)$  predstavlja intenzitet piksela na prostornim koordinatama. Intenzitet piksela ima vrijednosti od 0 do 255. Na slici 22. predstavljena je crno-bijela slika u obliku 2D signala.



Slika 22. 2D signal

Obojena RGB (eng. Red-Green-Blue) slika također predstavlja vektor 2D signala. Signal RGB slike dimenzija  $N \times M \times 3$  moguće je predstaviti kao:

$$x(n_v, n_t) = [x_R(n_v, n_t), x_G(n_v, n_t), x_B(n_v, n_t)], \quad 0 < n_v < M - 1, \quad 0 < n_t < N - 1,$$

gdje su  $x_R$ ,  $x_G$  i  $x_B$  intenziteti piksela crvenog, zelenog i plavog kanala.

Nakon što je crno-bijela slika prikazana kao 2D signal, signale želimo procesuirati kroz 2D konvoluciju. Slika može biti konvoluirana s različitim 2D filterima za procesuiranje slika. Da bi se uklonili šumovi sa slike koriste se filteri za redukciju šumova. Jedan od njih je *Gaussov filter* koji će biti opisan kasnije u ovom radu. Također, za detekciju rubova koriste se filteri koji izvlače visoko-frekventne komponente iz slike. Najčešće korišteni filteri bit će opisani kasnije. Filtere za procesuiranje slika možemo shvatiti kao sustave za procesuiranje slika koji su linearni i invarijantni na pomak (LSI sustavi za procesuiranje slika). Prije definiranja 2D konvolucije, bit će definirana dvodimenzionalna step funkcija pomoću koje se prikazuje dvodimenzionalni signal.

Dvodimenzionalna step funkcija  $\delta(n_1, n_2)$  gdje su  $n_1$  i  $n_2$  horizontalne i vertikalne koordinate definirana je sa:

$$\delta(n_1, n_2) = \begin{cases} 1 & \text{za } n_1 = 0 \text{ i } n_2 = 0 \\ 0 & \text{inače} \end{cases}$$

Slično, step funkcija s pomakom  $\delta(n_1-k_1, n_2-k_2)$  definirana je sa:

$$\delta(n_1 - k_1, n_2 - k_2) = \begin{cases} 1 & \text{za } n_1 = k_1 \text{ i } n_2 = k_2 \\ 0 & \text{inače} \end{cases}$$

Svaki diskretni 2D signal moguće je izraziti kao težinsku sumu funkcije  $\delta$  po različitim koordinatama, na sljedeći način:

$$x(n_1, n_2) = \sum_{k_2=-\infty}^{+\infty} \sum_{k_1=-\infty}^{+\infty} x(k_1, k_2) \delta(n_1 - k_1, n_2 - k_2)$$

Svaka operacija konvolucije kao prvi argument sadrži ulazni podatak. Drugi argument konvolucije je filter ili jezgra. Filter predstavlja LSI sustav za procesuiranje slika. Primjerice, kada pokušavamo snimiti fotografiju i zbog pokreta ruke fotografija postane zamagljena, zamagljenost možemo tretirati kao LSI sustav sa specifičnom step funkcijom  $f$  definiranom ispod. Pomoću step funkcije konvoluirana se realna fotografija i producira zamagljenu fotografiju kao izlaz. Dakle, kamera može biti definirana kao LSI sustav sa specifičnom step funkcijom. Svaki diskretni 2D signal, kao prethodno navedeni, prolazi kroz transformaciju  $f$ . Zbog linearnosti LSI sustava vrijedi sljedeće:

Funkciju  $f$  možemo izraziti kao:  $f(x(n_1, n_2)) = \sum_{k_2=-\infty}^{+\infty} \sum_{k_1=-\infty}^{+\infty} x(k_1, k_2) f(\delta(n_1 - k_1, n_2 - k_2))$

$$f(x(n_1, n_2)) = \sum_{k_2=-\infty}^{+\infty} \sum_{k_1=-\infty}^{+\infty} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2) ,$$

gdje je zbog invarijantnosti na pomak:  $f(\delta(n_1 - k_1, n_2 - k_2)) = h(n_1 - k_1, n_2 - k_2) .$

Svaka digitalna slika predstavlja 2D signal.

**Konvolucija slike**  $x(n_1, n_2)$  dimenzija  $N \times M$  sa 2D filterom za procesuiranje slika  $h(n_1, n_2)$  dana je sa:

$$y(n_1, n_2) = \sum_{k_2=-\infty}^{N-1} \sum_{k_1=-\infty}^{M-1} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2)$$

gdje vrijedi  $0 \leq n_1 \leq N-1, 0 \leq n_2 \leq M-1$ .

Procesuiranje slike pomoću filtera djeluje na crno-bijelim (2D) slikama prilikom čega se producira izlazna 2D slika. U slučaju procesuiranja slika koje nisu 2D također je moguće koristiti 2D filtere. U takvim situacijama potrebno je procesuirati svaki kanal slike kao 2D signal ili pretvoriti sliku u 2D dimenziju.<sup>17</sup>

S obzirom da je konvolucija komutativne prirode, prethodnu formulu koja predstavlja konvoluciju slike  $x(n_1, n_2)$  dimenzija NxM sa 2D filterom za procesuiranje slika  $h(n_1, n_2)$  možemo pisati kao:

$$y(n_1, n_2) = \sum_{k_2=-\infty}^{N-1} \sum_{k_1=-\infty}^{M-1} x(n_1 - k_1, n_2 - k_2) h(k_1, k_2)$$

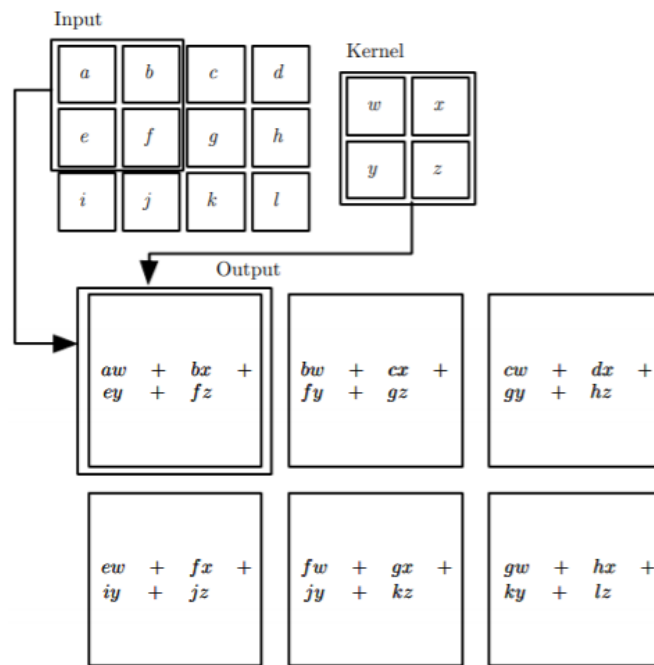
Svojstvo komutacije u konvoluciji nastaje zbog okretanja filtera (eng. kernel) nad ulaznom slikom. Jedini razlog za okretanje filtera jeste da se dobije svojstvo komutacije, koje je važno kod pisanja matematičkih dokaza ali ne toliko važno kod implementacije neuronske mreže. Zapravo, mnoge biblioteke neuronskih mreža implementiraju funkciju zvanu unakrsna korelacija, koja je ista kao konvolucija ali bez okretanja filtera. Biblioteke strojnog učenja često implementiraju unakrsnu korelaciju koju nazivaju konvolucijom. U nastavku ovog rada koristit će se pojam konvolucija za oba slučaja, ali će biti naknadno specificirano ako se radi o okretanju filtera. U slučaju dviju dimenzija, dvostruka cirkularna matrica odgovara konvoluciji. S obzirom na uvjet da nekoliko elemenata bude jednako, konvolucija često odgovara matrici čiji su ulazni podaci nule. Ovo se dešava zbog toga što je filter inače mnogo manji od originalne slike.<sup>18</sup>

Na slici 23. prikazan je primjer 2D konvolucije bez okretanja filtera. Ulazna slika prikazana je u matricnom obliku. U takvom slučaju izlaz je ograničen samo na pozicije u kojima filter djeluje u potpunosti, što se u nekim slučajevima naziva validna konvolucija.

---

<sup>17</sup> (Pattanayak, 2018) 161-166.str

<sup>18</sup> (Goodfellow, Bengio and Courville, 2016) 331-334.str



Slika 23. Primjer 2D konvolucije bez okretanja filtera

### 3.2 FILTERI ZA PROCESUIRANJE SLIKA

U nastavku bit će definirani najčešće korišteni filteri za procesuiranje slika u 2D dimenziji. Bez obzira da li je prikazan filter za procesuiranje slike ili slika u koordinatnom sustavu,  $n_1$  i  $n_2$  su diskretne koordinate za x i y smjer.

**Filter srednje vrijednosti** (eng. mean filter) ili filter prosječne vrijednosti računa lokalni prosjek jačine piksela u svakoj specifičnoj točki. Filter u matricnoj formi možemo prikazati na sljedeći način:

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Filter srednje vrijednosti najčešće se koristi za reduciranje šumova na slici, tako što računa prosjek jačine piksela u njegovom susjedstvu. Ukoliko je na slici prisutan Gaussov šum, koristimo filter srednje vrijednosti kako bi ga reducirali. Na slici 24. prikazano je djelovanje filtera srednje vrijednosti na sliku koja sadrži Gaussov šum.



Slika koja sadži Gaussov šum



Slika nakon konvolucije sa filterom srednje vrijednosti

Slika 24. Filter srednje vrijednosti na slici Mona Lise

**Medijan filter** zamjenjuje svaki piksel u susjedstvu s vrijednošću medijana jačine piksela u tom susjedstvu. Ovaj filter koristan je za uklanjanje tzv. „sol-papar šumova“ koji nastaju na formi crno-bijelih piksela. Uzročnik takvih šumova je slučajna smetnja prilikom nastajanja fotografije.



Slika koja sadži sol-papar šum

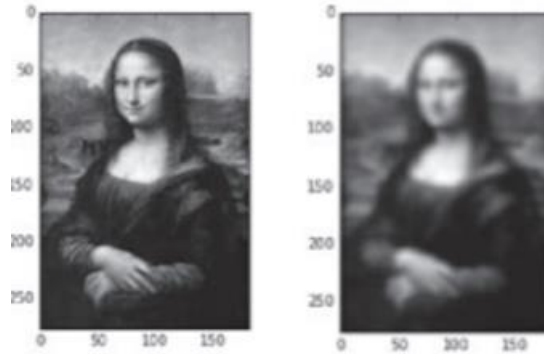


Slika nakon primjene medijan filtera

Slika 25. Primjena medijan filtera na sliku Mona Lise

**Gaussov filter** predstavlja modificiranu verziju filtera srednje vrijednosti, gdje su vrijednosti težina filtera distribuirane normalno oko središta. Težine su najveće u središtu filtera i opadaju prema krajevima. Shodno tome i Gaussov filter kada je primijenjen na sliku, ima najveći intenzitet u težištu koji opada prema krajevima. Ovaj filter koristi se za redukciju šumova tako što suzbija piksele visokih frekvencija. Kao rezultat primjene ovog filtera nastaje zamagljena slika, što se naziva Gaussova zamagljenost (eng. Gaussian blur).



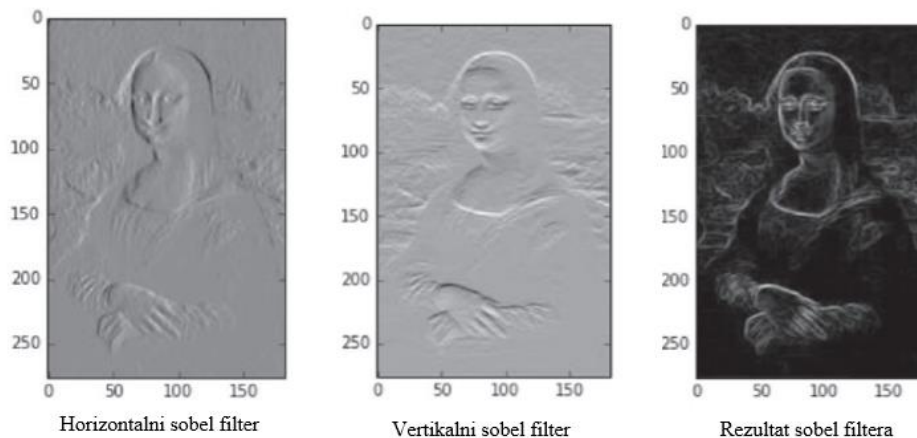


Slika 26. Primjena Gaussovog filtera na sliku Mona Lise

**Sobel filter (za detekciju rubova)** pronalazi horizontalne i vertikalne rubove na slikama. Rezultat ovog filtera na horizontalnoj i vertikalnoj osi može se predstaviti pomoću matrica  $H_x$  i  $H_y$  respektivno:

$$H_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad H_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Sobel filter sastoji se od horizontalnog sobel filtera koji detektira rubove u horizontalnoj osi i vertikalnog sobel filtera koji detektira rubove u vertikalnoj osi. Djelovanje ovih filtera respektivno kao i krajnji rezultat primjene sobel filtera prikazan je na slici 27. <sup>19</sup>



Slika 27. Primjena sobel filtera za detekciju rubova

Najpoznatiji filteri za procesuiranje slika kao i njihova upotreba kratko su opisani u tablici 2.

<sup>19</sup> (Pattanayak, 2018) 169-178.str

FILTER	PRIMJENA
Filter srednje vrijednosti	Reducira Gaussov šum
Medijan filter	Reducira sol-papar šum
Unsharp filter	Filter za poboljšanje ruba
Crimmins Speckle Removal filter	Filter za uklanjanje točkica sa slike
Sobel filter	Detektira rubove na slici
Gaussov filter	Filter za redukciju šumova
Frequency filter	Detektira frekvencije piksela na slici
Canny filter	Detektira rubove na slici
Conservative Smoothing filter	Filter za redukciju šumova
Weiner filter	Reducira šumove i zamagljenje

Tablica 2. Filteri i njihova primjena

### 3.3 MOTIVACIJA

Konvolucija sadrži tri važna svojstva koja mogu poboljšati sustav strojnog učenja: **raspršenu povezanost, dijeljenje parametara i ekvivarijantnu reprezentaciju.**

Klasične neuronske mreže koriste matrično množenje parametara sa odvojenim parametrom koji opisuje interakciju između svake ulazne i izlazne jedinice. Ovo znači da svaka ulazna jedinica vrši interakciju sa svakom izlaznom jedinicom, što implicira da su slojevi međusobno ovisni. Za razliku od klasičnih neuronskih mreža, konvolucijske neuronske mreže imaju raspršenu povezanost, koja je postignuta time što je filter (kernel) manjih dimenzija od ulazne slike. Primjerice, tokom procesuiranja slike koja ima milione piksela, pomoću filtera moguće je detektirati male značajke kao što su rubovi slike koje imaju znatno manje piksela. Ovakav način procesuiranja slike donosi znatne uštede u memoriji i doprinosi efikasnosti modela time što za računanje izlaznih jedinica koristi znatno manje operacija. Uštedi memorije doprinosi i odgovarajući izbor filtera za procesuiranje slika opisanih u prethodnoj tablici.

Grafički prikaz raspršene povezanosti nalazi se na slici 28.

Dijeljenje parametara podrazumijeva korištenje istog parametra u više funkcija modela. U klasičnim neuronskim mrežama, svaki element matrice težina je korišten jedanput u računanju izlaznog sloja. Za razliku od klasičnih neuronskih mreža, u konvolucijskim neuronskim mrežama svaka težina filtera korištena je na svakoj ulaznoj jedinici. Ovo implicira da umjesto učenja različitog skupa parametara na svakoj lokaciji, učimo samo parametre filtera.

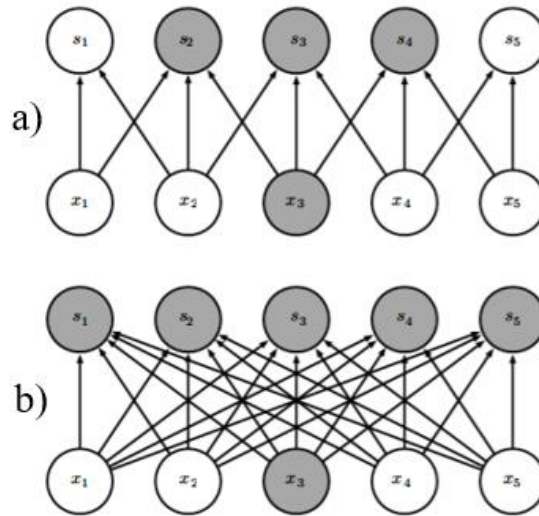
U slučaju konvolucije, dijeljenje parametara uzrokuje svojstvo koje se naziva translatorna ekvivarijentnost. Za funkciju kažemo da je ekvivarijantna ukoliko promjena ulaza uzrokuje promjenu izlaza na isti način. Svojstvo ekvivarijentnosti možemo definirati na sljedeći način:

Funkcija  $f(x)$  je translatorno ekvivarijantna funkciji  $g(x)$  ako vrijedi:  $f(g(x))=g(f(x))$ .

U slučaju konvolucije, ako je funkcija  $g$  ona koja translacija ulaznu sliku, tada je funkcija konvolucije ekvivarijantna  $g$ . Operacija konvolucije kreira 2D mapu gdje se određene značajke ulaza pojavljuju u cijeloj slici. Ovo je korisno kada znamo da određena funkcija malog broja susjednih piksela može biti korisna kada se primjeni na više lokacija ulaza. Primjerice, kod procesuiranja slike korisno je detektirati rubove na prvom sloju konvolucijske mreže, s obzirom na to da će se rubovi pojavljivati na cijeloj slici. Konvolucija nije prirodno ekvivarijantna na druge transformacije kao što su skaliranje slike ili njezina rotacija.<sup>20</sup>

---

<sup>20</sup> (Goodfellow, Bengio and Courville, 2016) 335-339.str



Slika 28. Raspršena povezanost-ulazna jedinica  $x_3$  i izlazne jedinice  $s$  na koje  $x$  ima utjecaj

a)  $s$  je formiran konvolucijom filtera širine 3,  $x$  ima utjecaj na samo tri neurona.

b)  $s$  je formiran množenjem matrica i povezanost nije raspršena što implicira da  $x$  ima utjecaj na sve neurone.

### 3.4 ARHITEKTURA KONVOLUCIJSKIH NEURONSKIH MREŽA

Konvolucijske neuronske mreže fokusiraju se na procesuiranje nestrukturiranih podataka kao što su slike, tekst, zvuk i govor. Kada je riječ o konvolucijskim neuronskim mrežama koje procesuiraju slike, na slikama vrši se detekcija značajki koje su bazirane na filteru koji se uči u procesu treniranja. U procesu konvolucije moguće je primijeniti neki od poznatih filtera prethodno definiranih u tablici 2., međutim, moguće je primijeniti filter koji konvolucijska neuronska mreža uči kroz proces treniranja. U nadziranom učenju, filteri se uče na način da reduciraju funkciju troška što je više moguće. Generalno, prvi sloj konvolucije detektira rubove, dok drugi sloj može detektirati neke više kompleksne oblike na slici kao što su krugovi ili pravokutnici. Treći sloj i slojevi nakon fokusiraju se na učenje mnogo više kompliciranih značajki bazirajući se na značajkama iz prethodnih slojeva.

Dakle, konvolucijska neuronska mreža sastoji se od niza slojeva pri čemu svaki sloj vrši transformaciju prethodnog pomoću određene funkcije. Prethodno su definirane prednosti konvolucijskih neuronskih mreža. Raspršena povezanost je jedna od tih prednosti, koja omogućava reduciranje broja parametara za učenje. Isti filter može učiti da detektira rubove na bilo kojem dijelu slike zbog svojstva ekvivarijantnosti.

U nastavku će biti definirane osnovne komponente konvolucijskih neuronskih mreža:

**Ulazni sloj** čuva vrijednosti piksela ulazne slike. Generalno, ulaz u konvolucijsku neuronsku mrežu može biti monokromatska slika ili slika u boji. Primjerice, ulazna slika dužine 64, visine 64 i dubine 3 (RGB slika) imat će ulazne dimenzije 64x64x3.

**Konvolucijski sloj** uzima sliku iz prethodnog sloja i primjenjuje određeni broj filtera s ciljem da kreira izlaznu sliku zvanu mapa značajki (eng. feature maps). Broj izlaznih slika jednak je broju filtera.

**Aktivacijska funkcija** za konvolucijske neuronske mreže je, najčešće, ReLU funkcija definirana u prethodnom poglavlju. Dimenzija izlaza je ista kao i dimenzija ulaza nakon primjene aktivacijske funkcije. ReLU sloj dodaje svojstvo nelinearnosti u mrežu i u isto vrijeme doprinosi nezasićenosti gradijenta za pozitivne ulaze.

**Sloj sažimanja** (eng. **Pooling layer**) smanjuje 2D aktivacijske mape tako da nakon ovog sloja imaju manju visinu i širinu. Dubina i broj aktivacijskih mapi ostaje isti.

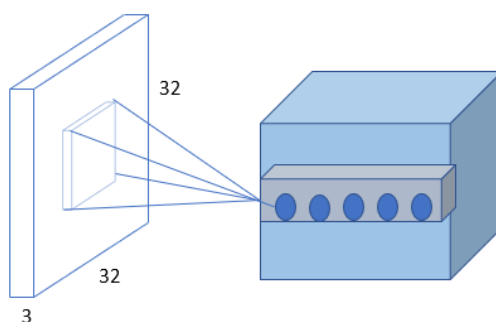
**Potpuno povezani slojevi** sadrže neurone koji primaju različit skup težina iz prethodnog sloja. Svaki neuron iz ovog sloja povezan je sa svim neuronima prethodnog.

### 3.4.1 KONVOLUCIJSKI SLOJ

U konvolucijskom sloju izvodi se operacija konvolucije, stoga se ovaj sloj naziva srcem konvolucijske neuronske mreže. Što je konvolucijski sloj dublji, više značajki na slici može detektirati. Primjerice, inicijalni konvolucijski sloj može detektirati rubove na slici dok slijedeći konvolucijski sloj može spajati rubove i detektirati neke složenije oblike tipa kruga ili pravokutnika. Dublji sloj od ovog može detektirati neke kompliciranije značajke, tipa

dijelove tijela i slično. U konvolucijskim neuronskim mrežama specificirana je jedino veličina filtera, dok su ostale težine postavljene na proizvoljne vrijednosti prije početka treninga. Težine filtera uče se tokom procesa treniranja konvolucijskih neuronskih mreža. Primjerice, filter dimenzija  $3 \times 3$  imat će 9 težina. U procesu konvolucije, filteri se inicijaliziraju i klizno pomiču preko ulazne slike. Za vrijeme pomicanja unaprijed, pomiče se filter po širini i visini ulaznog volumena i računa proizvod u točkama između ulaznih vrijednosti i vrijednosti filtera u bilo kojoj poziciji. Ovakvo pomicanje filtera unaprijed definira proces konvoluiranja. Pomicanjem filtera po širini i visini producira se dvodimenzionalna aktivacijska mapa koja daje odgovor tog filtera na svakoj poziciji. Mreža na taj način provodi proces učenja filtera tako što se filteri aktiviraju kada prepoznaju određenu značajku (npr. rub). Nakon pomicanja filtera po cijelom ulaznom volumenu dobivamo cijeli set filtera u svakom konvolucijskom sloju, pri čemu svaki od njih producira različitu dvodimenzionalnu aktivacijsku mapu. Aktivacijske mape se spremaju duž dimenzije dubine i produciraju izlazni volumen.

Pretpostavimo da ulazni sloj ima dimenziju  $[32 \times 32 \times 3]$  (dimenzija 3 predstavlja dubinu odnosno RGB sliku). Ukoliko je dimenzija filtera  $5 \times 5$  tada će svaki neuron u konvolucijskom sloju imati dimenziju  $[5 \times 5 \times 3]$  i ukupno 75 težina ( $5 \times 5 \times 3$ ) +1 parametar za pomak (eng. bias). Na slici 29. nalazi se primjer volumena ulaznog sloja i volumena neurona u prvom konvolucijskom sloju.



Slika 29. Primjer volumena ulaznog sloja i volumena neurona u prvom konvolucijskom sloju

Prethodno je definirana povezanost neurona između ulaznog i konvolucijskog sloja, ali ne i broj neurona u izlaznom sloju kao i njihov raspored. Postoje tri važna parametra koja utiču na veličinu izlaznog volumena: veličina filtera, korak i dopunjavanje.

**Veličina filtera** (eng. **filter size**) predstavlja visinu i širinu filterove mreže (kernela). Filter veličine 5x5 imat će 25 težina. Kao što je spomenuto prije, filter se inicijalizira i klizno pomiče (konvoluira) preko ulazne slike bez okretanja filtera. Ukoliko se konvolucija izvodi bez okretanja filtera u teoriji se naziva unakrsna korelacija. Dubina izlaznog volumena predstavlja važan parametar konvolucijske mreže. Dubina izlaznog volumena odgovara broju filtera koji su korišteni u konvolucijskom sloju, pri čemu svaki od tih filtera traži određene značajke u ulaznom volumenu.

**Korak** (eng. **stride**) označava broj piksela za koji se pomjeramo u svim prostornim smjerovima tokom izvršavanja konvolucije. U slučaju da ne preskačemo nijedan piksel, računamo konvolucijsku sumu za svaki od piksela i tada će korak biti jednak 1. U slučaju da preskačemo piksele i odabiremo primjerice svaki drugi, korak će biti jednak 2. Što se više piksela preskače, izlazni volumen bit će manji.

**Nadopunjavanje** (eng. **padding**) predstavlja pristup koji nadopunjuje nule na granicama slike kako bi se kontrolirala veličina izlaznog volumena prilikom konvolucije. Nadopunjavanje nulama omogućava kontrolu izlaznog volumena na način da najčešće izjednačava širinu i visinu ulaznog i izlaznog volumena.

Dužina konvoluirane izlazne slike duž specifične prostorne dimenzije dana je sa:

$$I = \frac{U - F + 2D}{K} + 1, \text{ gdje je}$$

U – dužina ulazne slike u specifičnim dimenzijama

F – dužina filtera u specifičnim dimenzijama

D – broj nadopunjenih nula na granicama

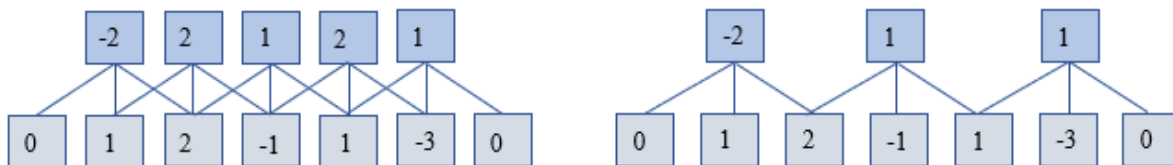
K – broj koraka u konvoluciji

Generalno, kada je korak jednak jedan tj. kada nema preskakanja piksela, veličina slike duž svake dimenzije je reducirana sa  $(F-1)/2$  na svakom kraju, gdje je F dužina kernela. Stoga, da bi izlazni volumen bio isti kao i ulazni, potrebna je dopuna dužine  $(F-1)/2$ .<sup>21</sup>

---

<sup>21</sup> (Pattanayak, 2018) 180-181.str

Primjerice, za ulazni volumen dimenzije 7x7 i 3x3 filterom sa korakom jedan i dopunom nula dobit će se izlazni volumen dimenzije 5x5. S korakom 2 izlazni volumen imao bi dimenziju 3x3.



Slika 30. Ilustracija prostornog rasporeda neurona na x osi

Lijevo: Ulazna veličina  $U=5$ , veličina filtera  $F=3$ , nadopuna  $D=1$  i korak  $K=1$  što daje izlaznu veličinu  $I=(5-3+2)/1+1=5$ .

Desno: Ulazna veličina  $U=5$ , veličina filtera  $F=3$ , nadopuna  $D=1$  i korak  $K=2$  što daje izlaznu veličinu  $I=(5-3+2)/2+1=3$ .

Primijetimo da na primjeru iznad nije moguće koristiti korak  $K=3$ , s obzirom da u tom slučaju izlazna veličina ne bi bila cijeli broj. Dakle, da li je određenu veličinu koraka moguće koristiti možemo provjeriti na opisani način. Također možemo primijetiti da je na slici iznad lijevo ulazna veličina jednaka 5 i izlazna veličina jednaka 5. Izlazna veličina jednaka je ulaznoj zbog nadopunjavanja nulama, inače bi bila jednaka 3 jer je to veličina filtera koji je korišten. Kao što je spomenuto iznad, ukoliko je korak prilikom konvolucije jednak 1, potrebno je postaviti dopunjavanje na  $D=(F-1)/2$  kako bi ulazna i izlazna veličina bila jednaka. Dopunjavanje može biti validno i nepromjenjivo. **Validno dopunjavanje** (eng. valid padding) podrazumijeva da ne dopunjujemo mapu značajki tako da izlazna veličina ostane ista.

**Nepromjenjivo** (eng. same padding) podrazumijeva oblik dopune predstavljen na primjeru iznad. U tom slučaju dopunjujemo mapu značajki oko rubova kako bi ulazna i izlazna veličina ostale iste.<sup>22</sup>

Prethodno su opisani svi važni parametri konvolucijskog sloja. Kao što je spomenuto na početku, konvolucijski sloj predstavlja srce konvolucijske mreže zbog toga što se u njemu

<sup>22</sup> (Convolutional Neural Networks for Visual Recognition, 2020)

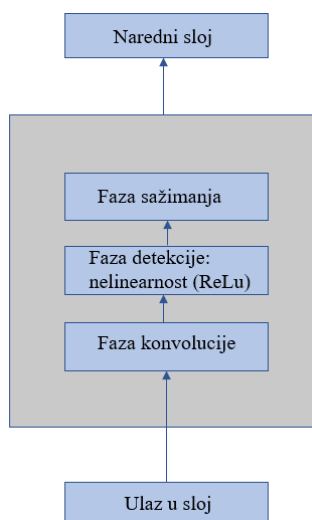


izvodi najvažnija operacija-operacija konvolucije. U literaturi se često označava kao **CONV** sloj.

U nastavku rada bit će opisani drugi važni slojevi konvolucijske mreže, kao što su sloj sažimanja i potpuno povezani slojevi.

### 3.4.2 SLOJ SAŽIMANJA

Uobičajeni sloj konvolucijske neuronske mreže sastoji se od tri faze. U prvoj fazi sloj primjenjuje nekoliko operacija konvolucije paralelno kako bi producirao skup linearnih aktivacija. U drugoj fazi na svaku linearnu aktivaciju primjenjuje se nelinearna aktivacijska funkcija, najčešće ReLu aktivacijska funkcija. Ova faza konvolucijske neuronske mreže naziva se i faza detekcije (eng. detector stage). U trećoj fazi primjenjuje se funkcija sažimanja kako bi modificirala rezultat prethodnog sloja. Na slici 31. prikazane su uobičajene komponente sloja konvolucijske neuronske mreže.

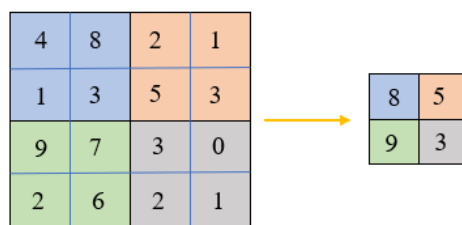


Slika 31. Komponente sloja konvolucijske neuronske mreže

Faza sažimanja kao ulaz prima podatke o aktivacijskim mapama iz prethodnog sloja, a kao rezultat daje aktivacijske mape s umanjenim dimenzijama. Funkcija sažimanja (eng. pooling function) mijenja izlazni rezultat mreže na određenoj lokaciji sa sumiranim statističkim podacima u lokalnoj okolini.

Postoje dvije vrste operacije sažimanja koje se najčešće koriste: **sažimanje maksimalnih vrijednosti** (eng. **max pooling**) i **prosječno sažimanje** (eng. **average pooling**). Sažimanje maksimalnih vrijednosti pronalazi maksimalnu vrijednost u okolini pravokutnog oblika. Primjer sažimanja maksimalnih vrijednosti nalazi se na slici 2.16. Prosječno sažimanje u pravokutnoj okolini pronalazi prosječnu vrijednost. U oba slučaja, proces sažimanja pomaže kod invarijantnosti malih translacija ulaznih podataka. Invarijantnost translacije znači ako ulaznu vrijednost transliramo za mali iznos, izlazna vrijednost nakon primjene operacije sažimanja neće se previše promijeniti. Invarijantnost lokalne translacije može biti veoma korisno svojstvo onda kada nam je važno znati da li je određena značajka prisutna na slici, ali nam nije previše važna njena lokacija.

Primjerice, kada želimo detektirati da li slika sadrži lice potrebno je da znamo da se lijevo oko nalazi na lijevoj strani lica, a desno oko na desnoj strani lica. U ovom slučaju nije potrebno znati točnu lokaciju piksela na kojem se nalaze oči. Korištenjem operacije sažimanja funkcije koje sloj uči trebaju biti invarijantne na male translacije. Kada se ovaj uvjet postigne, statistička efikasnost mreže se povećava. U zadatku klasifikacije, sažimanje je neophodno za rad sa podacima nejednakih veličina. Ulazni podaci u procesu klasifikacije trebaju biti jednakih veličina, što se može postići promjenom veličine podataka tokom procesa sažimanja. Na taj način sloj klasifikacije prima uvijek isti broj sumiranih statistika bez obzira na veličinu ulaznih podataka. Primjerice, posljednji sloj sažimanja može biti definiran tako da daje 4 skupa sumiranih statistika, jedan za svaki kvadrant slike, bez obzira na veličinu slike.<sup>23</sup>



Slika 32. Sažimanje maksimalnih vrijednosti sa 2x2 filterom i korakom 2

<sup>23</sup> (Goodfellow, Bengio and Courville, 2016) 339-344.str

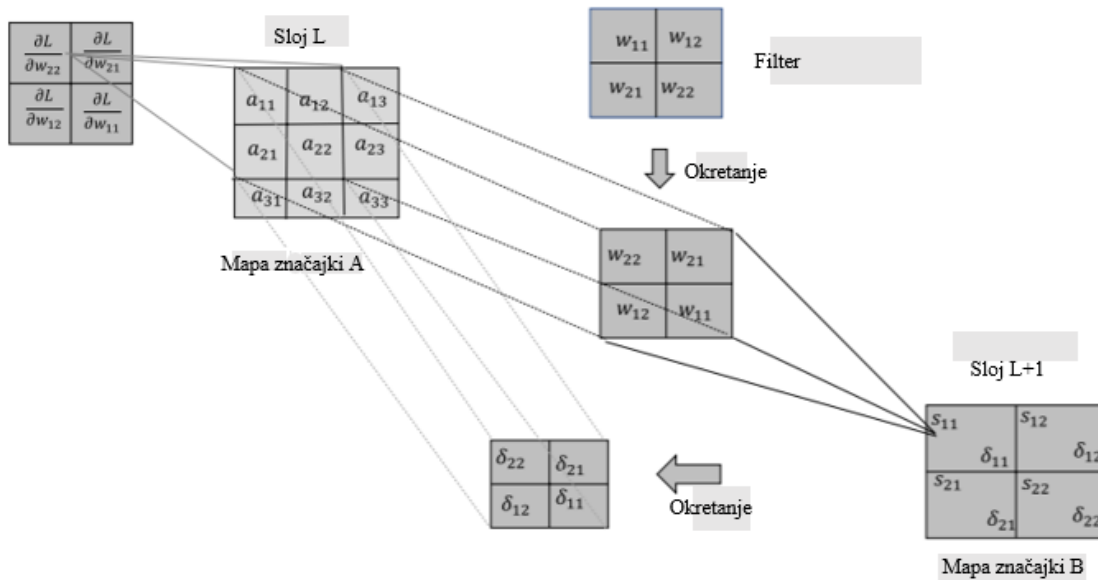
### 3.4.3 POTPUNO POVEZANI SLOJ

Neuroni u potpuno povezanom sloju imaju potpunu povezanost sa svim aktivacijama iz prethodnog sloja. Prisjetimo se, ovakav princip povezanosti neurona spomenut je u prethodnom poglavlju o klasičnim neuronskim mrežama. Aktivacija ovih neurona može se računati matričnim množenjem s dodatkom pomaka. Potpuno povezani sloj skraćeno se označava kao FC sloj (Fully connected layer). Razlika između konvolucijskog i potpuno povezanog sloja jeste to da su neuroni u konvolucijskom sloju povezani samo u lokalnoj regiji ulaznih podataka kao što je prethodno opisano. Neuroni oba sloja računaju proizvod u točkama, tako da je njihova funkcionalnost ista. Zbog iste funkcionalnosti, moguće je izvršiti konverziju iz potpuno povezanog sloja u konvolucijski sloj.

### 3.5 ALGORITAM PROPAGACIJE POGREŠKE UNATRAG

U nastavku će biti definiran algoritam propagacije unatrag kroz konvolucijski sloj i sloj sažimanja.

#### 3.5.1 PROPAGACIJA UNATRAG U SLOJU KONVOLUCIJE



Slika 33. Propagacija unatrag kroz sloj konvolucije

Propagacija unatrag kroz konvolucijski sloj slična je propagaciji unatrag kroz duboku neuronsku mrežu koja je opisana u prethodnom poglavlju. Jedina razlika je u povezanosti neurona, s obzirom da su težine u konvolucijskom sloju raspršene što implicira da iste težine čine različite ulazne podatke kako bi kreirale izlaznu mapu značajki. Svaka izlazna mapa značajki rezultat je konvolucije slike ili mape značajki iz prethodnog sloja s filterom čije težine se uče tokom propagacije unatrag. Na slici 2.17 prikazana je propagacija unatrag kroz sloj konvolucije. Mapa značajki A u sloju L konvoluirana s filterom pri čemu nastaje mapa značajki B u L+1. sloju. Rezultati izlazne mape značajki su zapravo rezultati konvolucije i mogu se izraziti kao:

$$\begin{aligned} s_{11} &= w_{22} * a_{11} + w_{21} * a_{12} + w_{12} * a_{21} + w_{11} * a_{22} \\ s_{12} &= w_{22} * a_{12} + w_{21} * a_{13} + w_{12} * a_{22} + w_{11} * a_{23} \\ s_{21} &= w_{22} * a_{21} + w_{21} * a_{22} + w_{12} * a_{31} + w_{11} * a_{32} \\ s_{22} &= w_{22} * a_{22} + w_{23} * a_{22} + w_{12} * a_{32} + w_{11} * a_{33} \end{aligned}$$

Prethodne jednakosti možemo napisati kao:

$$s_{ij} = \sum_{n=1}^2 \sum_{m=1}^2 w_{(3-m)(3-n)} * a_{(i-1+m)(j-1+n)} \quad (1)$$

Definirajmo sada gradijent funkcije troška L u odnosu na ulazne podatke  $S_{ij}$  na slijedeći način:

$$\frac{\partial L}{\partial s_{ij}} = \delta_{ij} \quad (2)$$

Izračunajmo sada gradijent funkcije troška u odnosu na težinu  $W_{22}$ . Težina  $W_{22}$  povezana je sa svim ulaznim podacima  $S_{ij}$ , stoga će imati sve komponente prethodno definiranog gradijenta.

$$\frac{\partial L}{\partial w_{22}} = \sum_{j=1}^2 \sum_{i=1}^2 \frac{\partial L}{\partial s_{ij}} \frac{\partial s_{ij}}{\partial w_{22}} = \sum_{j=1}^2 \sum_{i=1}^2 \delta_{ij} \frac{\partial s_{ij}}{\partial w_{22}} \quad (3)$$

Iz prethodne jednakosti za različite vrijednosti  $S_{ij}$  vrijedi slijedeće:

$$\frac{\partial s_{11}}{\partial w_{22}} = a_{11}, \quad \frac{\partial s_{12}}{\partial w_{22}} = a_{12}, \quad \frac{\partial s_{13}}{\partial w_{22}} = a_{21}, \quad \frac{\partial s_{14}}{\partial w_{22}} = a_{22} \quad (4)$$

Jednakost (3) sada možemo pisati kao:

$$\frac{\partial L}{\partial w_{22}} = \delta_{11} * a_{11} + \delta_{12} * a_{12} + \delta_{21} * a_{21} + \delta_{22} * a_{22} \quad (5)$$

Slično, za težinu  $W_{21}$  vrijedi slijedeće:

$$\frac{\partial L}{\partial w_{21}} = \sum_{j=1}^2 \sum_{i=1}^2 \frac{\partial L}{\partial s_{ij}} \frac{\partial s_{ij}}{\partial w_{21}} = \sum_{j=1}^2 \sum_{i=1}^2 \delta_{ij} \frac{\partial s_{ij}}{\partial w_{21}} \quad (6)$$

$$\frac{\partial s_{11}}{\partial w_{21}} = a_{12}, \quad \frac{\partial s_{12}}{\partial w_{21}} = a_{13}, \quad \frac{\partial s_{21}}{\partial w_{21}} = a_{22}, \quad \frac{\partial s_{22}}{\partial w_{21}} = a_{23} \quad (7)$$

$$\frac{\partial L}{\partial w_{21}} = \delta_{11} * a_{12} + \delta_{12} * a_{13} + \delta_{21} * a_{22} + \delta_{22} * a_{23} \quad (8)$$

Primijenimo isti princip na dvije preostale težine:  $W_{11}$  i  $W_{12}$  respektivno.

$$\frac{\partial L}{\partial w_{11}} = \sum_{j=1}^2 \sum_{i=1}^2 \frac{\partial L}{\partial s_{ij}} \frac{\partial s_{ij}}{\partial w_{11}} = \sum_{j=1}^2 \sum_{i=1}^2 \delta_{ij} \frac{\partial s_{ij}}{\partial w_{11}} \quad (9)$$

$$\frac{\partial L}{\partial w_{11}} = \delta_{11} * a_{22} + \delta_{12} * a_{23} + \delta_{21} * a_{32} + \delta_{22} * a_{33} \quad (10)$$

$$\frac{\partial s_{11}}{\partial w_{11}} = a_{22}, \quad \frac{\partial s_{12}}{\partial w_{11}} = a_{23}, \quad \frac{\partial s_{21}}{\partial w_{11}} = a_{32}, \quad \frac{\partial s_{22}}{\partial w_{11}} = a_{33} \quad (11)$$

$$\frac{\partial L}{\partial w_{11}} = \delta_{11} * a_{22} + \delta_{12} * a_{23} + \delta_{21} * a_{32} + \delta_{22} * a_{33} \quad (12)$$

$$\frac{\partial L}{\partial w_{12}} = \sum_{j=1}^2 \sum_{i=1}^2 \frac{\partial L}{\partial s_{ij}} \frac{\partial s_{ij}}{\partial w_{12}} = \sum_{j=1}^2 \sum_{i=1}^2 \delta_{ij} \frac{\partial s_{ij}}{\partial w_{12}} \quad (13)$$

$$\frac{\partial s_{11}}{\partial w_{12}} = a_{21}, \quad \frac{\partial s_{12}}{\partial w_{12}} = a_{22}, \quad \frac{\partial s_{21}}{\partial w_{12}} = a_{31}, \quad \frac{\partial s_{22}}{\partial w_{12}} = a_{32} \quad (14)$$

$$\frac{\partial L}{\partial w_{12}} = \delta_{11} * a_{21} + \delta_{12} * a_{22} + \delta_{21} * a_{31} + \delta_{22} * a_{32} \quad (15)$$

S obzirom na prethodno definirane gradijente funkcije troška L i definiranih težina respektivno, možemo pisati:

$$\frac{\partial L}{\partial w_{ij}} = \sum_{n=1}^2 \sum_{m=1}^2 \delta_{mn} * a_{(i-1+m)(j-1+n)} \quad (16)$$

Prethodno definirane jednakosti možemo pisati i u matričnom obliku. U jednakosti (17) znak (x) označava unakrsnu korelaciju.

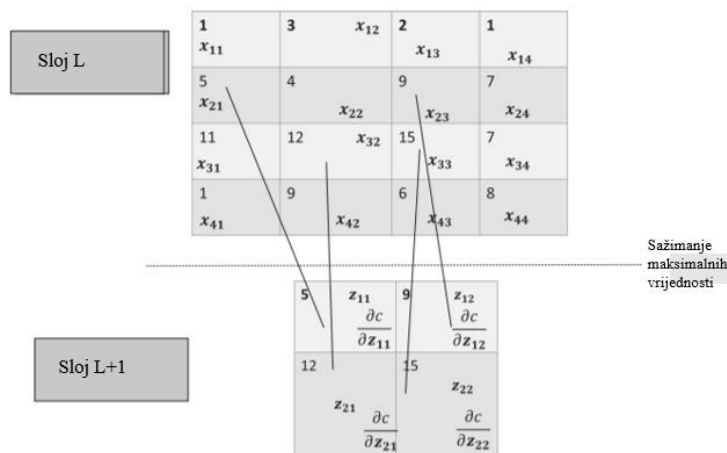
$$\begin{bmatrix} \frac{\partial L}{\partial w_{22}} & \frac{\partial L}{\partial w_{21}} \\ \frac{\partial L}{\partial w_{12}} & \frac{\partial L}{\partial w_{11}} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{21} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} (x) \begin{bmatrix} \delta_{11} & \delta_{12} \\ \delta_{21} & \delta_{22} \end{bmatrix} \quad (17)$$

Unakrsna korelacija prve matrice nakon znaka jednakosti iz jednakosti (17) s matricom može biti konvolucija te iste matrice s okrenutim filterom.

Filter možemo okrenuti na slijedeći način:  $\begin{bmatrix} \delta_{22} & \delta_{21} \\ \delta_{12} & \delta_{11} \end{bmatrix}$ . Sada jednakost (17) u slučaju konvolucije možemo pisati kao:

$$\begin{bmatrix} \frac{\partial L}{\partial w_{22}} & \frac{\partial L}{\partial w_{21}} \\ \frac{\partial L}{\partial w_{12}} & \frac{\partial L}{\partial w_{11}} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{21} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} (*) \begin{bmatrix} \delta_{22} & \delta_{21} \\ \delta_{12} & \delta_{11} \end{bmatrix} \quad (18)$$

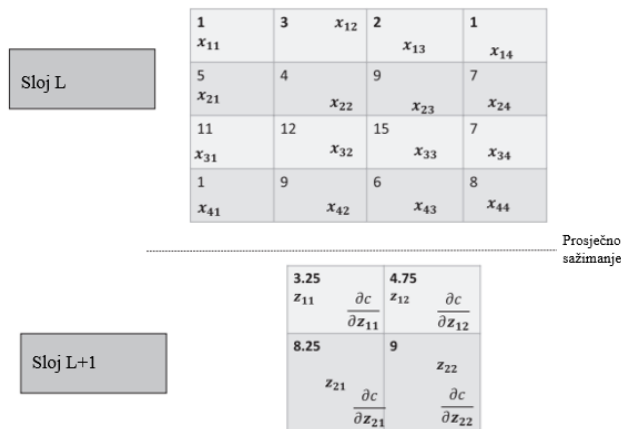
### 3.5.2 PROPAGACIJA UNATRAG U SLOJU SAŽIMANJA



Slika 34. Propagacija unatrag kroz sloj sažimanja maksimalnih vrijednosti

Na slici 34. prikazana je operacija sažimanja maksimalnih vrijednosti. Mapa značajki u sloju L, nakon primjene operacije konvolucije i ReLU aktivacijske funkcije, prolazi kroz operaciju sažimanja maksimalnih vrijednosti u sloju L+1 kako bi producirala izlaznu mapu značajki. Veličina filtera u sloju sažimanja maksimalnih vrijednosti je 2x2, a korak je veličine 2. Izlazni sloj nakon sažimanja je veličine  $\frac{1}{4}$  od ulazne mape značajki, a izlazne vrijednosti su predstavljene sa:  $z_{ij} \forall i, j \in \{1, 2\}$ . Na slici iznad vidimo da  $z_{11}$  prima vrijednost 5 s obzirom da je maksimalna vrijednost u tom 2x2 bloku jednaka 5. Na slici 35. postupak je identičan, osim što se koristi operacija prosječnog sažimanja.

U tom slučaju  $z_{11}$  prima vrijednost 3.25 s obzirom da je prosječna vrijednost u tom 2x2 bloku jednaka 3.25.



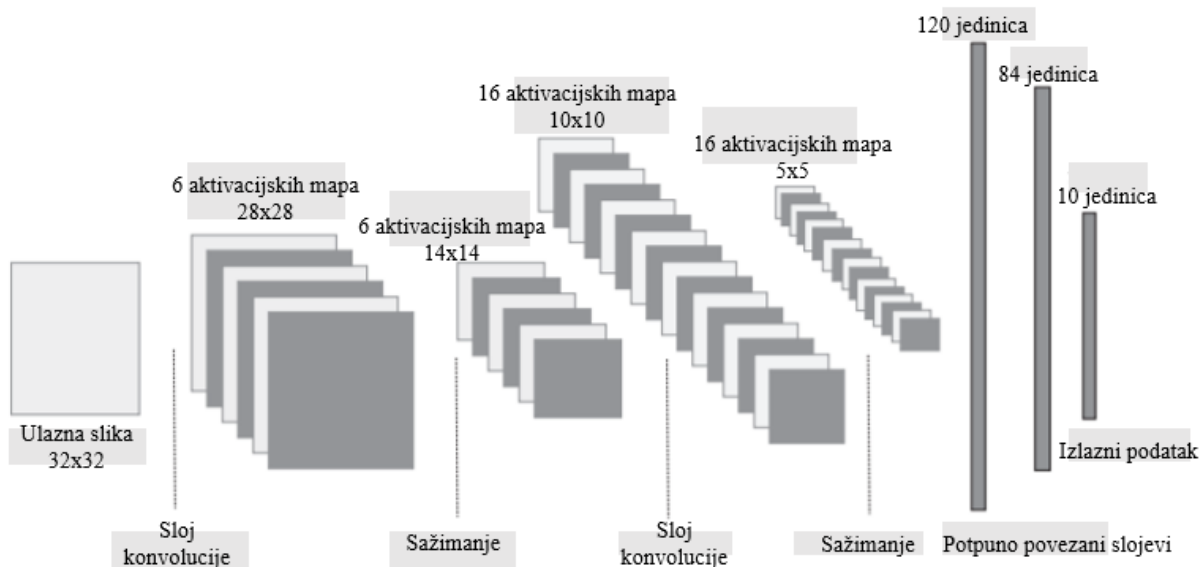
Slika 35. Propagacija unatrag kroz sloj prosječnog sažimanja

### 3.6 PRIMJERI KONVOLUCIJSKIH NEURONSKIH MREŽA

U ovom dijelu bit će definirane neke od često korištenih arhitektura konvolucijskih neuronskih mreža. Ove neuronske mreže nisu korištene isključivo u problemu klasifikacije, već i u problemima kao što su segmentacija, lokalizacija i detekcija. Izuzev LeNet arhitekture, svaka od ovih arhitektura osvojila je *ImageNet* takmičenje klasifikacije tisuća klasa.

**LeNet arhitektura** prva je uspješna konvolucijska neuronska mreža koju je kreirao Yann LeCun u 1990. godini s ciljem klasifikacije ručno pisanih brojeva. LeNet5 je posljednja verzija ove arhitekture. Ulazni podaci u ovu mrežu su slike dimenzije 32x32 koje prolaze kroz sloj konvolucije i produciraju 6 aktivacijskih mapa dimenzije 28x28. Ovih 6 aktivacijskih mapa tada prolaze kroz sloj sažimanja kako bi producirali 6 slika dimenzije 14x14. Slijedeći konvolucijski sloj sadrži 16 aktivacijskih mapa dimenzije 10x10, dok sloj sažimanja u ovom slučaju reducira dimenzije aktivacijskih mapa na 5x5. Ovaj proces prate dva potpuno povezana sloja. Dijagram LeNet5 arhitekture prikazan je na slici 36.



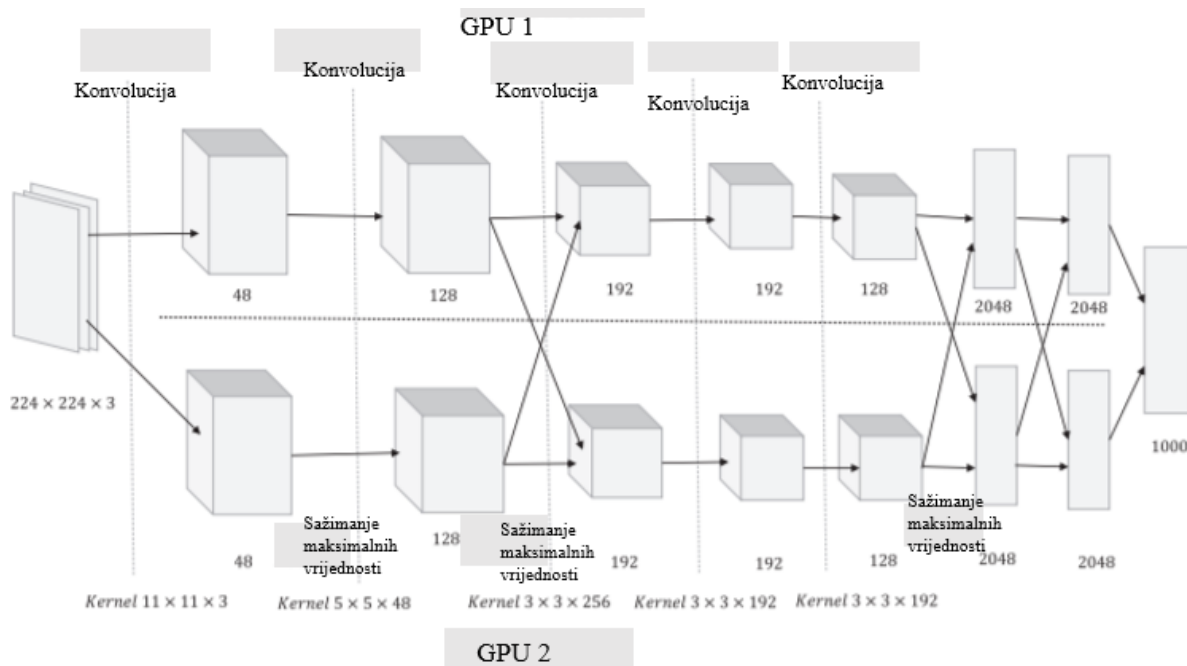


Slika 36. LeNet5 arhitektura

Neke od bitnih značajki LeNet5 arhitekture:

- Izlazni sloj ove arhitekture imao je 10 klasa koje odgovaraju ciframa od 0-9.
- Sloj sažimanja uzima 2x2 susjedne piksele i zbraja intenzitete 4 piksela. Na sumu se zatim primjenjuje operacija skaliranja s treniranim težinama i pomakom, a zatim prolazi kroz sigmoidnu aktivacijsku funkciju. Ovakav proces se razlikuje od prethodno objašnjenog sažimanja maksimalnih vrijednosti i prosječnog sažimanja.
- Filterova mreža je dimenzije 5x5. Aktivacijska funkcija je RBF funkcija (eng. radial basis function) umjesto SoftMax funkcije koja se najčešće koristi. 84 jedinice u potpuno povezanom sloju imaju 84 veze s drugim klasama što implicira da postoje 84 odgovarajuće težine. 84 težine prikazuju karakteristike svake od klasa. Razlika između SoftMax i RBF funkcije jeste što kod SoftMax funkcije računamo produkt u točki za svaki vektor težina, dok kod RBF funkcije računamo Euklidovu distancu između ulaznih i izlaznih vektora težina.
- Reprezentativne težine svake klase su fiksirane unaprijed, u prijevodu, nisu težine koje se uče.

**AlexNet** arhitektura nastala je 2012. godine, a razvili su je Alex Krizhevsky, Ilya Sutskever i Geoffrey Hinton. Konvolucijska neuronska mreža osvojila je takmičenje pod nazivom *ImageNet Large-Scale Visual Recognition Challenge* pri čemu je postigla stopu greške od 15.4% dok je drugoplasirana mreža imala stopu greške od 26.2%. AlexNet konvolucijska mreža sastoji se od 5 konvolucijskih slojeva, slojeva sažimanja maksimalnih vrijednosti, slojeva isključivanja i tri potpuno povezana sloja. Izlazni sloj prepoznavao je tisuću klasa. Ulazni podaci mreže su slike dimenzije 224x224x3. Prvi konvolucijski sloj producira 96 aktivacijskih mapa koje odgovaraju 96 filterovih mreža dimenzije 11x11x3 s korakom od 4 piksela. Drugi konvolucijski sloj producira 256 aktivacijskih mapa koje odgovaraju filterovoj mreži veličine 5x5x48. Prva dva konvolucijska sloja popraćena su slojem maksimalnog sažimanja, dok preostala tri konvolucijska sloja su smještena jedan poslije drugog, bez slojeva sažimanja. Peti konvolucijski sloj popraćen je slojem maksimalnog sažimanja, dva potpuno povezana sloja s 4096 jedinica i izlaznim SoftMax slojem koji sadrži tisuću klasa. Treći konvolucijski sloj sadrži 384 filterove mreže veličine 3x3x256, dok četvrti i peti konvolucijski sloj imaju 384 i 256 filterovih mreža gdje je svaki veličine 3x3x192. Isključivanje od 0.5 korišteno je u zadnja dva potpuno povezana sloja. AlexNet konvolucijska mreža u periodu u kojem je nastala bila je tehnički skupa za izvedbu, stoga je treniranje moralo biti podijeljeno u dvije različite GPU jedinice. Dijagram AlexNet arhitekture prikazan je na slici 37. Primijetimo da je kod trećeg konvolucijskog sloja korištena unakrsna povezanost pa je mreža filtera dimenzije 3x3x256, a ne 3x3x128. Isti princip unakrsne povezanosti primjenjuje se na potpuno povezanim slojevima koji se prikazuju kao regularno potpuno povezani slojevi sa 4096 jedinica.



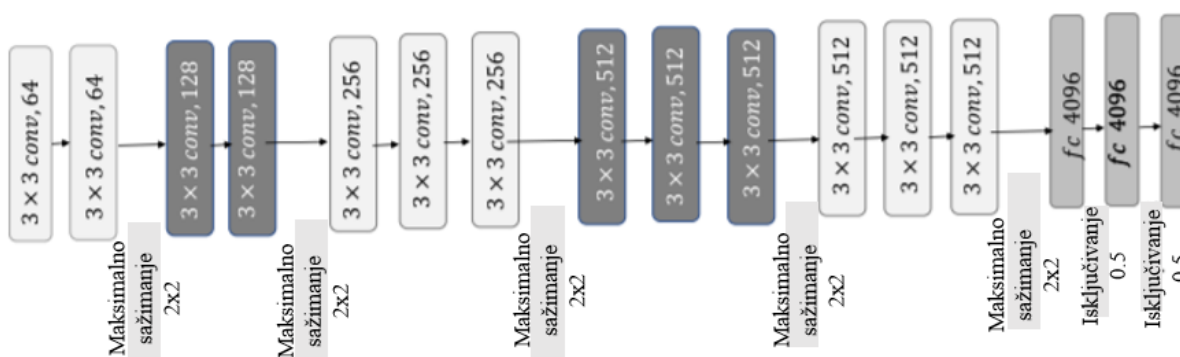
Slika 37. AlexNet arhitektura

Neke od važnih značajki AlexNet arhitekture su:

- Za svojstvo nelinearnosti korištena je ReLu aktivacijska funkcija, koja je doprinijela lakšem računanju u odnosu na prethodno korištene sigmoidne ili tahn aktivacijske funkcije.
- Isključivanje (eng. dropout) korišteno je da reducira preveliko učenje modela.
- Korišteno je preklapajuće sažimanje.
- Model je treniran na dvije GPU jedinice i za taj proces bilo mu je potrebno 5 dana.
- Podaci su bili uvećani raznim tehnikama kao što je translacija, horizontalna refleksija i slično.

**VGG16** arhitektura predstavljena je 2014.godine na ILSVRC-2014 takmičenju gdje je osvojila prvu nagradu. Ova 16-slojna arhitektura predstavlja duboku, ali sasvim jednostavnu mrežu koja je nakon predstavljanja doživjela veliku popularnost. Autori ove arhitekture su Karen Simonyan i Andrew Zisserman. Umjesto korištenja velike filterove mreže za konvoluciju, VGG16 arhitektura koristi 3x3 filter i ReLu aktivacijsku funkciju. Razlog korištenja manjeg filtera bio je činjenica da je koristiti dva 3x3 konvolucijska sloja

ekvivalentno korištenju jednog konvolucijskog sloja dimenzije 5x5 zadržavajući prednosti manjeg filtera. Neke od prednosti manjeg filtera su redukcija broja parametara i ostvarenje boljeg svojstva nelinearnosti zbog postojanja dva konvolucijska sloja. Važno svojstvo ove mreže predstavlja upravo redukcija broja parametara, što se dešava zbog operacije konvolucije i sažimanja maksimalnih vrijednosti. Ulazeći dublje u mrežu broj parametara se smanjuje, dok se broj aktivacijskih mapa povećava skupa s brojem filtera. Na slici 38. prikazan je dijagram VGG16 arhitekture.



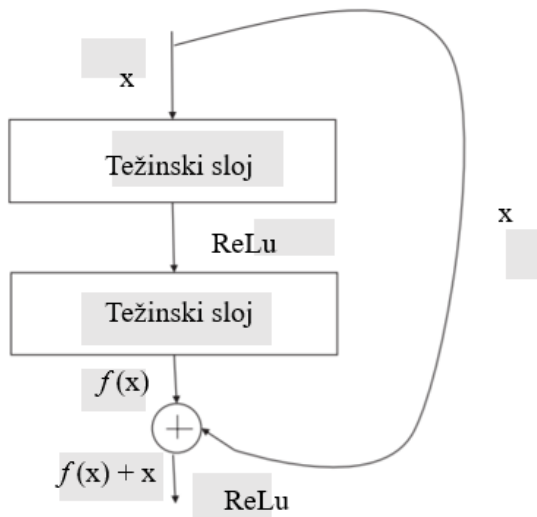
Slika 38. RGGF arhitektura

Neke od značajki RGG16 arhitekture su:

- Ulaz u mrežu su slike dimenzije 224x224x3. Prva dva konvolucijska sloja produciraju 64 aktivacijske mape, na koje se primjenjuje sloj maksimalnog sažimanja.
- Filter u operaciji konvolucije ima dimenziju 3x3 s korakom 1. Operacija maksimalnog sažimanja ima dimenziju 2x2 s korakom 2.
- Treći i četvrti konvolucijski sloj produciraju 128 aktivacijskih mapa.
- Na kraju mreže dolaze tri potpuno povezana sloja s 4096 jedinica, svaki popraćen s izlaznim SoftMax slojem koji sadrži tisuće klasa.
- Sve jedinice u mreži koriste ReLu aktivacijsku funkciju.

**ResNet** arhitektura predstavljena je 2015.godine od strane Microsoft-a kada je osvojila *ILSVRC 2015* takmičenje s greškom od samo 3.6% što se smatra boljim od stope ljudske pogreške od 5-10%. ResNet mreža implementira jedinstvenu ideju tzv. rezidualnih

blokova. Nakon svake serije *konvolucija-ReLu-konvolucija* ulaz u operaciju vraća se na izlaz operacije. Ideja mreže jeste da dodaje već naučene težine iz prethodnih slojeva. Primjer rezidualnog bloka prikazan je na slici 39. <sup>24</sup>



Slika 39. Rezidualni blok

---

<sup>24</sup> (Pattanayak, 2018) 206-211.str

## 4. PROBLEM KLASIFIKACIJE

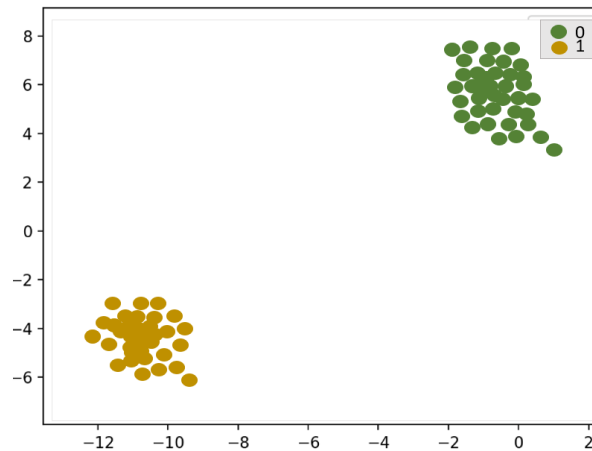
Nastanak strojnog učenja posljedica je potrebe čovjeka da određene probleme rješava uz pomoć računala. Problem klasifikacije predstavlja centralni problem strojnog učenja i može se definirati kao učenje stroja kako da klasificira podatke po određenom kriteriju. Zadatak klasifikacije jeste da nepoznati objekt identificira kao člana neke od poznatih kategorija. Ukoliko su unaprijed definirane i opisane kategorije za klasifikaciju, riječ je o nadziranom učenju. Postoji i verzija klasifikacije nastala od nenadziranog učenja koja se naziva klasteriranje. U slučaju klasteriranja kategorije za klasifikaciju nisu unaprijed definirane, već stroj koji vrši klasifikaciju pronalazi slične karakteristike na osnovu kojih grupira podatke. Jedan od primjera problema klasifikacije jeste detekcija neželjene pošte (eng. e-mail spam detection). Kao što ljudski mozak može prepoznati neželjenu e-poštu, računalo je procesom učenja postiglo zavidne rezultate. Proces učenja sastoji se od toga da je treniran veliki broj poruka e-pošte koje su raspoređene u dvije kategorije: e-pošta i neželjena e-pošta. Na taj način računalo je naučilo sve potrebne karakteristike za prepoznavanje neželjene e-pošte što je rezultiralo filtriranjem prilikom primanja takve vrste poruka. Osim klasifikacije neželjene pošte postoji još mnogo primjena problema klasifikacije kao što su: klasifikacija slika, prepoznavanje govora, identifikacija muzike i slično. Postoje dvije vrste klasifikacije:

1. Binarna klasifikacija (eng. binary classification)
2. Klasifikacija u više kategorija (eng. multi-class classification)

**Binarna klasifikacija** podrazumijeva klasificiranje podataka u dvije kategorije koje obično označavaju normalno i abnormalno stanje. Primjer ovakvog tipa klasifikacije jeste upravo filtriranje neželjene pošte gdje je svaka poruka označena kao „neželjena“ ili „nije-neželjena“. Binarna klasifikacija može se poistovjetiti sa Bernulijevom distribucijom (eng. Bernouli distribution) u kojoj se događaju dodjeljuje vrijednost 0 ili 1. U slučaju klasifikacije svakom objektu dodjeljuje se vjerovatnost pripadanja klasi 0 i klasi 1. Poznati algoritmi koji se koriste za binarnu klasifikaciju su:

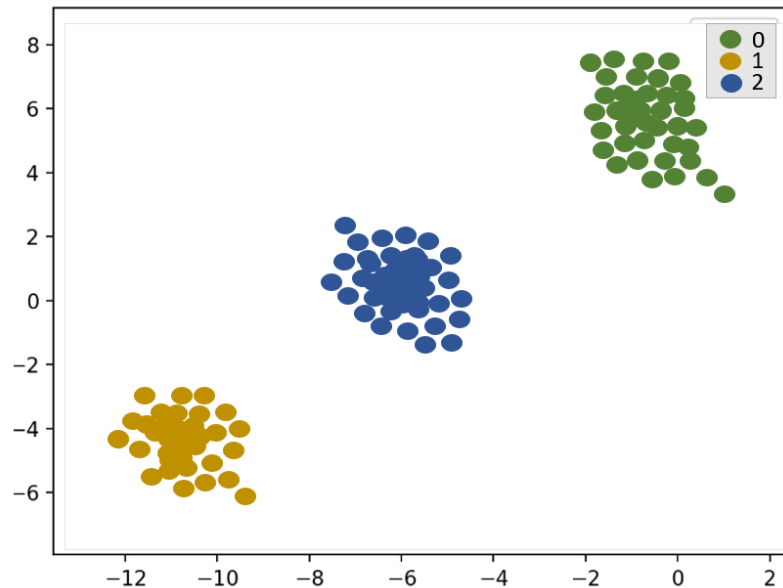
- Logistička regresija
- K-najbližih susjeda

- Stablo odluke
- Stroj potpornih vektora
- Naive Bayes algoritam i drugi.



Slika 40. Prikaz binarne klasifikacije

**Klasifikacija u više kategorija** podrazumijeva klasificiranje objekata u više od dvije kategorije. Za razliku od binarne klasifikacije koja razlikuje normalno i abnormalno stanje, ovakav tip klasifikacije razlikuje nekoliko predefiniраниh stanja u koje svrstava objekte na osnovu određenih kriterija. Primjer ovakve klasifikacije jeste klasifikacija dermatoskopskih slika nepravilnosti na koži u nekoliko kategorija koje predstavljaju bolesti kože (koja će biti predstavljena u ovom radu). Slika 40. predstavlja prikaz binarne klasifikacije, dok se na slici 41. nalazi prikaz klasifikacije u više kategorija.



Slika 41. Prikaz klasifikacije u više kategorija

U nastavku rada pod pojmom klasifikacije podrazumijevat će se klasifikacija u više kategorija, te će se kao klasifikator koristiti konvolucijske neuronske mreže.

#### 4.1 HAM 10000 (HUMAN AGAINST MACHINE) SKUP PODATAKA

HAM 10000 predstavlja skup podataka koji sadrži aproksimativno deset tisuća dermatoskopskih slika za treniranje. Slike su prikupljene širom svijeta od ljudi različite populacije, godina i s različitih dijelova tijela. Skup podataka dostupan je javno preko ISIC arhive (<https://isic-archive.com/>) u svrhu akademskog obrazovanja iz područja strojnog učenja. Slike su prikupljane u periodu od 20 godina s dva različita instituta u Austriji i Australiji. Više od 50% bolesti predstavljenih u ovom skupu podataka potvrđene su od strane stručnjaka iz oblasti patologije, dok je kod ostalih slučajeva riječ o stručnom mišljenju dermatologa i mikroskopskih nalaza. Dermatoskopija (eng. dermatoscopy) je tehnika dijagnostike benignih i malignih oboljenja na koži. Dermatoskopske slike pogodne su za treniranje neuronskih mreža u cilju klasifikacije istih. Na taj način moguće je usporediti točno dijagnosticirana oboljenja od strane modela neuronskih mreža i stručnjaka iz oblasti dermatologije. Treniranje neuronskih mreža u oblasti dermatoskopije započelo je davne 1994. godine kada su



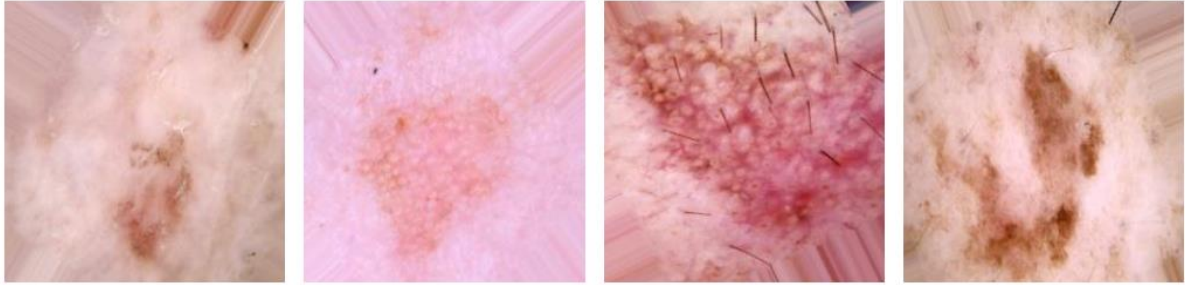
dermatoskopske slike korištene u svrhu razlikovanja malignih od benignih oboljenja.<sup>25</sup> Iako su rezultati bili obećavajući, problem je predstavljao nedovoljan broj dermatoskopskih slika kao i neravnomjerna distribucija podataka o malignim i benignim oboljenjima. Treniranje neuronskih mreža u ovoj oblasti zahtijeva veliki broj visokokvalitetnih dermatoskopskih slika, što u ovom slučaju nije bilo moguće. 200 dermatoskopskih slika od čega je 160 benignih i 40 malignih oboljenja bilo je dostupno 2013. godine, koje su služile kao skup podataka za klasificiranje kožnih oboljenja sve do danas.<sup>26</sup> ISIC arhiva predstavlja kolekciju baza podataka koje trenutno sadrže 13786 dermatoskopskih slika. Istraživanja u oblasti dermatoskopije kao i treniranje neuronskih mreža u ovoj oblasti često koriste dermatoskopske slike iz ISIC arhive. Nedostatak i ovog skupa podataka jeste činjenica da 12893 slika od ukupno 13786 pripada klasi nevusa ili melanome. Klasifikacija dermatoskopskih slika u više kategorija oboljenja predstavlja mnogo veći izazov od klasifikacije benignih i malignih oboljenja. Pouzdane predikcije u oblasti dermatoskopije do 2018. godine nisu postojale, zbog čega je pokrenut ISIC 2018 izazov u klasifikaciji dermatoskopskih slika. Skup za treniranje bio je raspoređen u sedam različitih kategorija koje predstavljaju različita oboljenja na koži. U praksi je najvažnija klasifikacija između malignih i benignih oboljenja, ali točna dijagnoza oboljenja je također veoma bitna s obzirom na činjenicu da se dva slična oboljenja mogu liječiti na potpuno različite načine. U nastavku rada bit će detaljnije definirana sva oboljenja koja će biti korištena kao kategorije za klasifikaciju.

**Aktinična keratoza** (eng. actinic keratosis), poznata kao i solarna keratoza, predstavlja jedno od čestih oboljenja kože. Najčešće nastaje na dijelovima kože koji su često izloženi suncu i može se razviti u karcinom pločastih stanica. Veću šansu za aktiničnu keratozu imaju osobe sa svijetlom ili crvenom kožom, kao i osobe koje puno vremena provode na otvorenom.

---

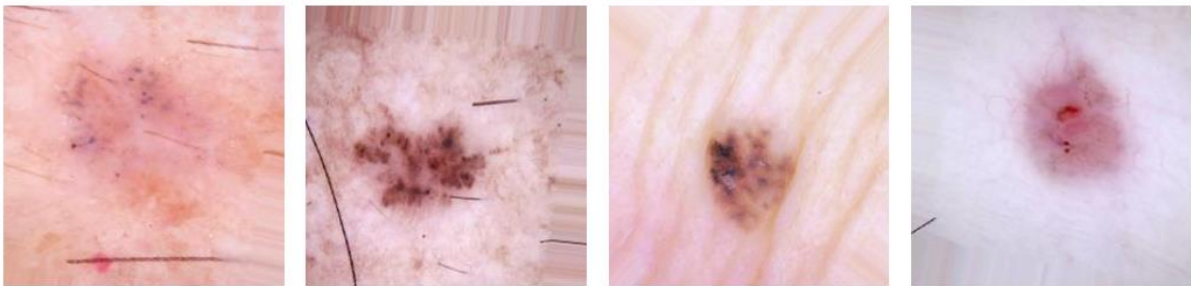
<sup>25</sup> (Binder et al., 1994)

<sup>26</sup> (Mendonca et al., 2013)



Slika 42. Aktinična keratoza

**Bazocelularni karcinom** (eng. basal cell carcinoma) ili bazaliom je najčešći oblik raka bazalnih stanica epidermisa. Rijetko metastazira ali raste destruktivno ukoliko se ne liječi na vrijeme. Kao i aktinična keratoza najčešće zahvata osobe svijetle puti. Ova vrsta raka kože smatra se najmanje opasnim.



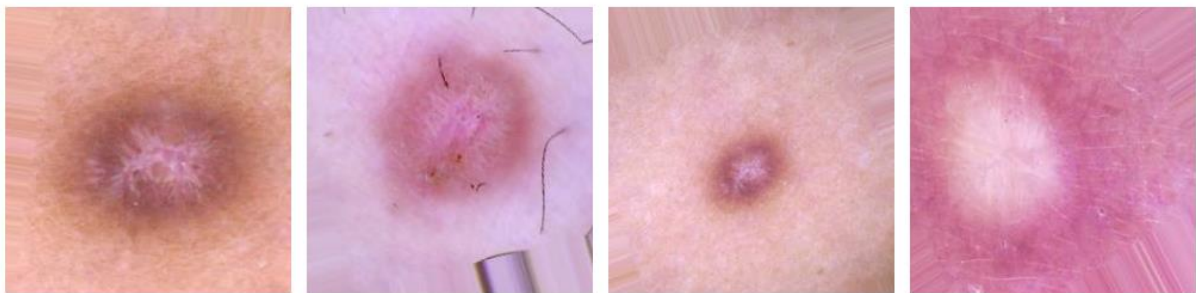
Slika 43. Bazocelularni karcinom

**Benigna keratoza** (eng. benign keratosis) uključuje bolesti kao što su seboreična keratoza i solarni lentigo. Grupirane su zajedno u istu kategoriju zbog svojih bioloških sličnosti. Iako ova vrsta nepravilnosti na koži izgleda kao rak kože, ovakve izrasline zapravo nisu kancerogene. Ovakve bolesti se najčešće javljaju kod starije populacije. Kao i prethodno opisana oboljenja najčešće se pojavljuje na dijelovima tijela koja se najviše izlažu suncu.



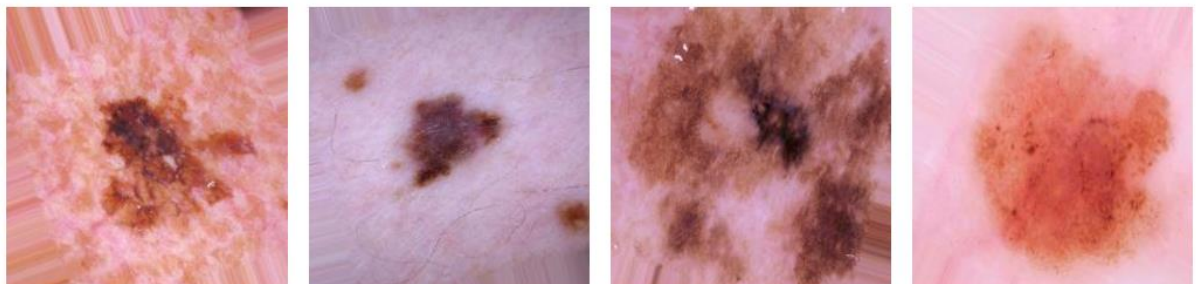
Slika 44. Benigna keratoza

**Dermatofibrom** (eng. dermatofibroma) je benigno oboljenje kože koje se ubraja u dobroćudne tumore. Od dermatofibroma češće obolijevaju žene nego muškarci, te se više javlja kod osoba mlađe dobi. Poprilično lako se uklanjaju zbog svoje sličnosti sa nekim drugim nepravilnostima na koži kao što su madeži.



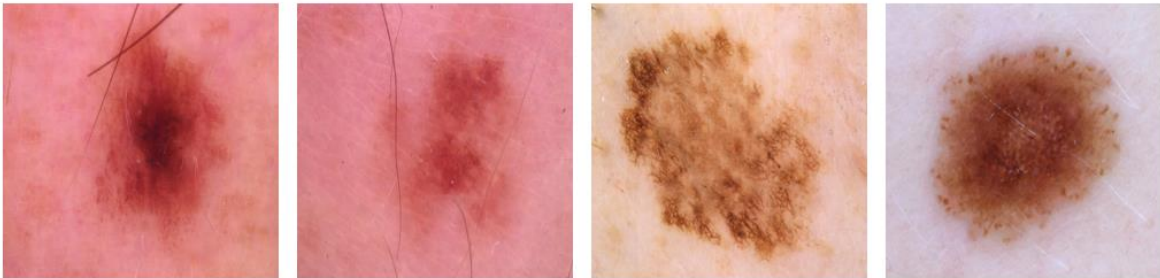
Slika 45. Dermatofibrom

**Melanocitni nevus** (eng. melanocytic nevi) predstavlja benigno oštećenje melanocita koji se može pojaviti već na rođenju ili tijekom životnog vijeka. Melanocitni nevusi mogu biti svijetlosmeđi ili tamnosmeđi. Iako ne predstavljaju oboljenje opasno po život, ukoliko se ne prate mogu postati ozbiljna maligna oboljenja.



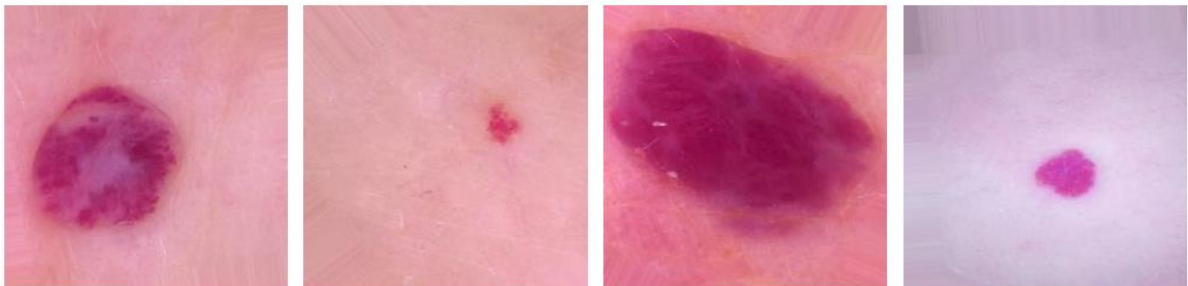
Slika 46. Melanocitni nevus

**Melanom** (eng. melanoma) smatra se najagresivnijim malignim oboljenjem koje se pojavljuje u nekoliko varijanti. Ukoliko se otkrije u ranom stadiju može biti otklonjen kirurškim putem. Postoje četiri najčešća oblika melanoma: površinsko šireći, lentigo maligni, nodularni i akralni lentigonozni. Sva četiri oblika su objedinjena u jednu kategoriju melanom. Ozbiljniji oblici melanoma liječe se ciljanom terapijom, imunoterapijom i kemoterapijom. Ovakav oblik oboljenja kože često dovodi i do smrti.



Slika 47. Melanom

**Vaskularna kožna oboljenja** (eng. vascular skin lesions) uključuju vaskularne tumore i vaskularne malformacije. Predstavljaju dobroćudne tumore koji nisu smrtonosni. Najčešće su crvenkasto-ružičaste boje i javljaju se kod osoba mlađe dobi.



Slika 48. Vaskularna kožna oboljenja

Prethodno definirana oboljenja predstavljaju 7 kategorija za klasifikaciju.<sup>27</sup>

---

<sup>27</sup> (Tschandl, Rosendahl and Kittler, 2018)

U nastavku rada bit će detaljnije opisan HAM 10000 skup podataka kao i njihovo pretprocesiranje u procesu klasifikacije.

## 4.2 PRETPROCESIRANJE PODATAKA

Podaci HAM 10000 preuzeti su s HARVARD Dataverse i dostupni su javnosti na slijedećoj poveznici: <https://dataverse.harvard.edu/dataset.xhtml?>. Podaci se sastoje od dermatoskopskih slika koje su raspoređene u dva foldera i csv dokumenta koji sadrži sve potrebne podatke o slikama. Prvi korak u pretprocesiranju podataka jeste detaljno ih analizirati. Za analizu HAM 10000 podataka korišten je csv dokument i jupyter notebook. Za početak potrebno je učitati csv dokument i prikazati prvih 5 redaka.

```
data.head()
```

	lesion_id	image_id	dx	dx_type	age	sex	localization
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear

Slika 49. HAM10000\_metadata.csv

Na prethodnoj slici prikazano je prvih 5 redaka csv dokumenta iz kojih je moguće zaključiti koje sve parametre podaci sadrže. Dokument se sastoji od 10015 redaka koji predstavljaju slike i 7 parametara koji predstavljaju obilježje oboljenja. Detaljni opis svih parametara nalazi se u tablici 3.

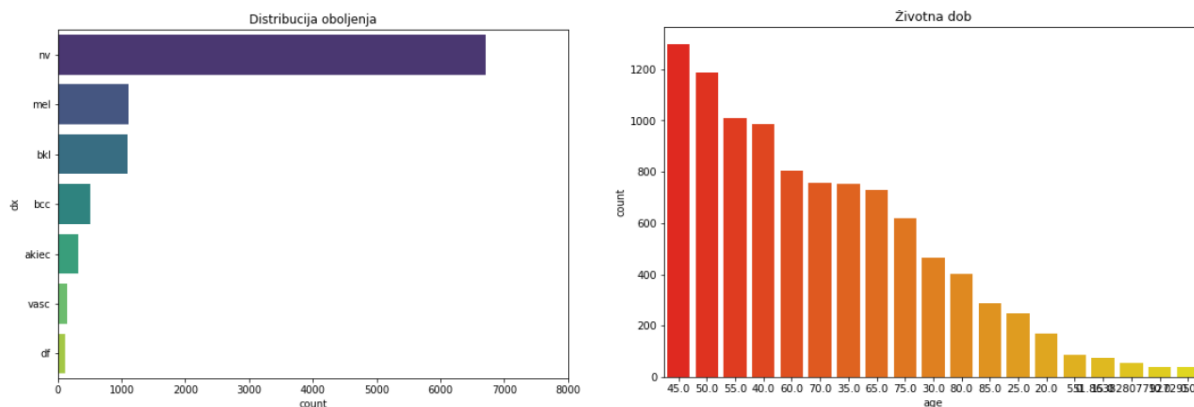
Parametar	Opis
lesion_id	Određeno oboljenje sadrži više slika što se može pratiti preko ovog parametra
image_id	Jedinstveni broj svake slike
dx	Kategorija oboljenja
dx_type	Način na koje je oboljenje ustanovljeno
age	Životna dob
sex	Spol
localization	Lokacija oboljenja na tijelu

Tablica 3. Opis parametara iz HAM10000\_metadata.csv

Nakon učitanih parametara potrebno je provjeriti da li neki od parametara sadrži NULL vrijednosti. Ukoliko takvi parametri postoje onda mjesta u dokumentu koja sadrže NULL vrijednost mijenjaju se sa srednjom vrijednosti preostalih vrijednosti iz dokumenta koji odgovaraju tom parametru. Sada, ako u dokumentu ne postoje polja s NULL vrijednostima moguće je nastaviti s analiziranjem podataka.

#### 4.2.1 ANALIZA PODATAKA

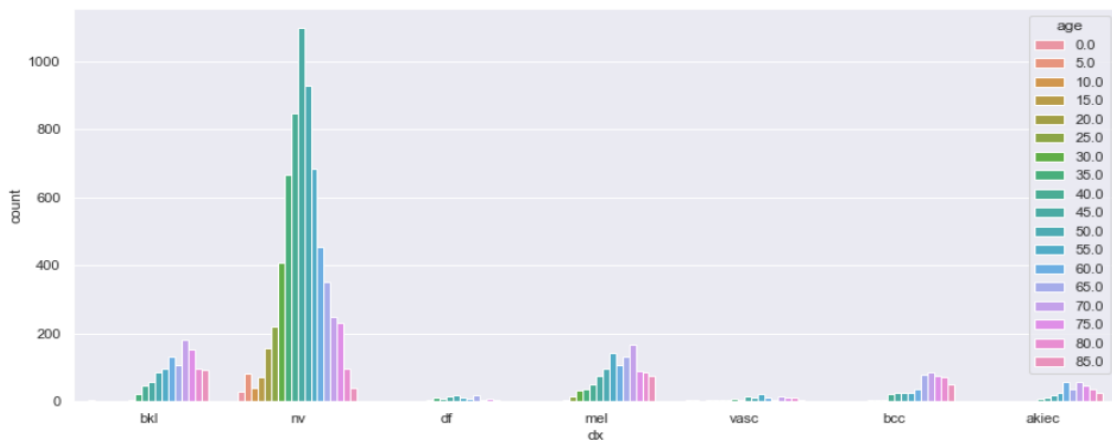
Najvažniji parametar u analizi podataka jeste kategorija oboljenja. Važno je analizirati koliko je koje oboljenje zastupljeno u podacima. Nakon kategorije oboljenja za analizu mogu biti važni životna dob, lokacija oboljenja i spol. Analiza ova četiri parametra izvršena je grafički i prikazana je na narednim slikama.



Slika 50. Analiza oboljenja i životne dobi

Prvi problem podataka HAM 10000 jeste neravnomjerna distribucija oboljenja. Oboljenje melanocitni nevus sadrži skoro 70% ukupnih podataka. Najmanje zastupljeno oboljenje jeste dermatofibrom sa svega 2% oboljenja, a zatim vaskularna oboljenja s 3%. Neravnomjerna distribucija podataka predstavlja veliki problem za klasifikaciju podataka, s obzirom da bi skup za treniranje trebao sadržavati približno isti broj podataka za sve kategorije. Priprema podataka za klasifikaciju bit će opisana nakon analize podataka.

Od kožnih oboljenja najčešće obolijevaju osobe u dobi od 35-60 godina.

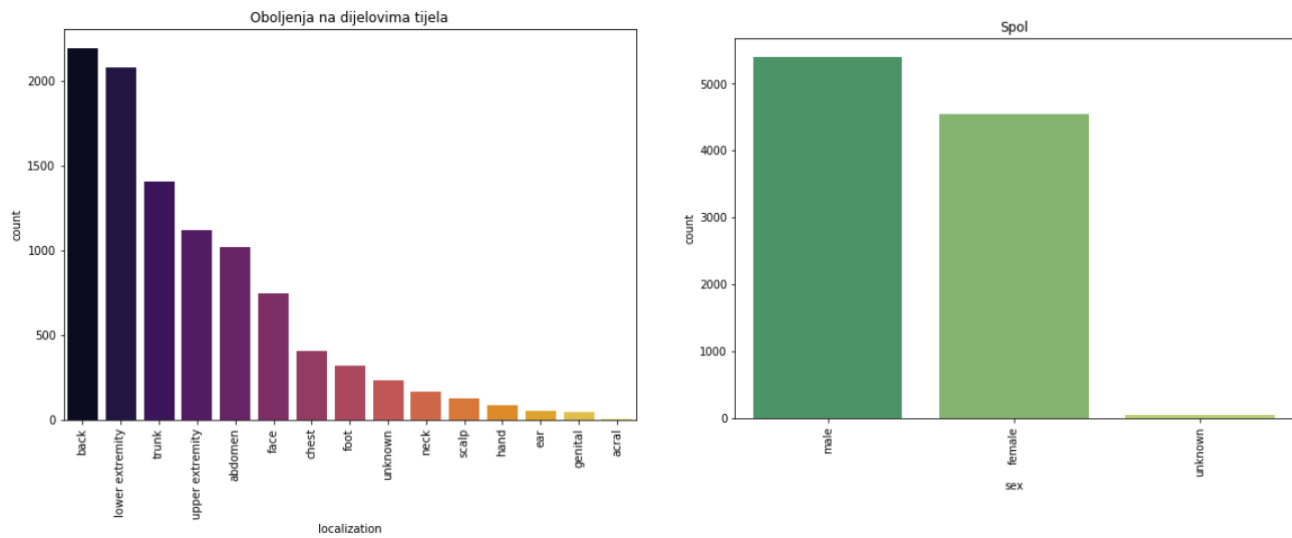


Slika 51. Oboljenja u odnosu na životnu dob

Od melanocitnog nevusa i dermatofibroma najčešće obolijevaju osobe srednje životne dobi, dok od melanoma i bazocelularnog karcinoma najčešće obolijevaju osobe starije životne dobi.

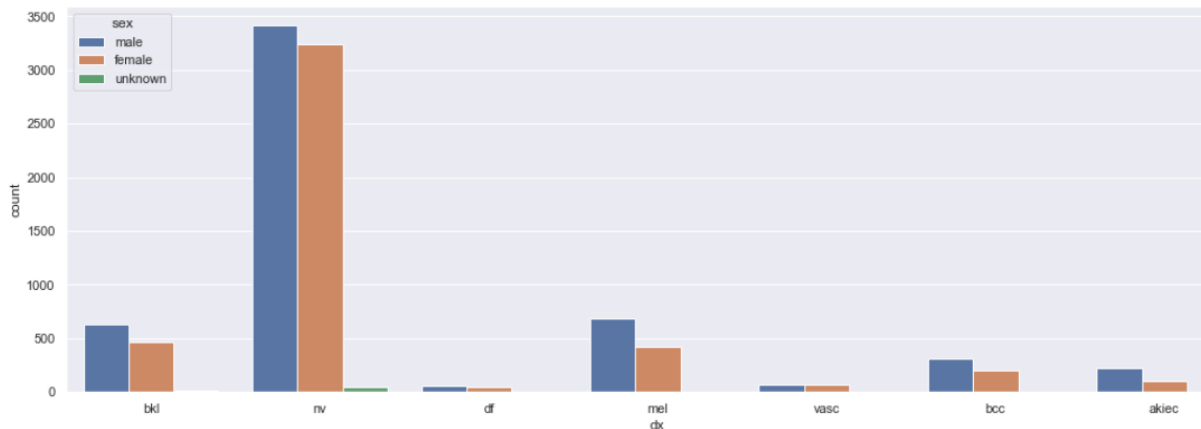
Melanocitni nevus bolest je koja zahvata i mlađu populaciju.





Slika 52. Analiza lokacije oboljenja na tijelu i spola

Najčešća lokacija kožnih oboljenja jesu leđa i niži ekstremiteti, zatim trbuh i viši ekstremiteti. Na genitalijama, ušima i šakama kožna oboljenja su najmanje zastupljena. Od kožnih oboljenja podjednako obolijevaju i muškarci i žene.



Slika 53. Oboljenja u odnosu na spol

Kožna oboljenja su procentno više zastupljena kod muškaraca. Od bazocelularnog karcinoma, melanocitnog nevusa, melanoma, aktinične keratoze i benigne keratoze više obolijevaju muškarci, dok su bolesti dermatofibrom i vaskularna oboljenja zastupljeni podjednako između muškaraca i žena.<sup>28</sup>

<sup>28</sup> (Marcelino, 2019)



## 4.2.2 PRIPREMA PODATAKA ZA KLASIFIKACIJU

Nakon analize i prije izgradnje modela potrebno je izvršiti pripremu podataka kako bi model postigao što bolje rezultate. Priprema podataka podrazumijeva to da svi podaci imaju iste fizičke specifikacije kao što su dimenzija i boja. Podatke za treniranje potrebno je rasporediti u kategorije za klasifikaciju.

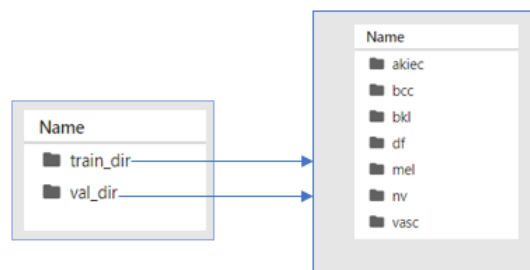
Prvi korak u pripremi HAM 10000 podataka jeste da se podaci rasporede u dva direktorija: direktorij za treniranje (`train_dir`) i direktorij za validaciju (`val_dir`), tako da u kategoriji za treniranje bude 90% podataka, a u kategoriji za validaciju preostalih 10%.

```
[29]: # Prikazati koliko podataka sadrži skup za treniranje i testiranje
print(df_train.shape)
print(df_val.shape)

(9013, 7)
(1002, 7)
```

Slika 54. Broj slika u kategorijama za treniranje i validaciju

U svakom od tih direktorija kreirano je sedam direktorija sa skraćenim nazivima koje označavaju oboljenja prethodno opisana u prijašnjem poglavlju.



Slika 55. Prikaz direktorija za treniranje i validaciju

Veoma je važan korak provjera koliko slika sadrži svaka kategorija. Kako bi model postigao što bolje rezultate poželjno je da svaka kategorija sadrži približno isti broj slika.

```
# Prikazati koliko podataka svaka klasa u skupu za treniranje sadrži
df_train['dx'].value_counts()
```

```
nv      6034
mel     1002
bkl     989
bcc     463
akiec   294
vasc    128
df      103
```

Slika 56. Broj slika u kategorijama skupa za treniranje

Na prethodnoj slici vidimo da broj slika u svim kategorijama nije približno isti. Oboljenje melanocitni nevus sadrži više slika nego sve preostale kategorije. Ukoliko treniramo model na ovakvom skupu podataka desit će se da većinu podataka za validaciju model klasificira kao melanocitni nevus. Takav model nema zadovoljavajuće rezultate naročito u slučaju kada se oboljenje melanoma klasificira kao melanocitni nevus. S obzirom na osjetljivost podataka korištenje takvog modela u praksi bilo bi izuzetno opasno.

Kako bi se postiglo da sve kategorije imaju približno isti broj slika koristi se augmentacija podataka. Augmentacija podataka u ovom slučaju izvodi se tako da se nad slikama vrši rotacija i promjena dimenzije. Nakon augmentacije podataka potrebno je izvršiti provjeru broja slika u svim kategorijama skupa za treniranje.

```
# Koliko slika za treniranje ima u svakoj kategoriji
print(len(os.listdir('base_dir/train_dir/nv')))
print(len(os.listdir('base_dir/train_dir/mel')))
print(len(os.listdir('base_dir/train_dir/bkl')))
print(len(os.listdir('base_dir/train_dir/bcc')))
print(len(os.listdir('base_dir/train_dir/akiec')))
print(len(os.listdir('base_dir/train_dir/vasc')))
print(len(os.listdir('base_dir/train_dir/df')))
```

```
6034
5810
5984
5606
5930
5170
4170
```

Slika 57. Broj slika nakon augmentacije

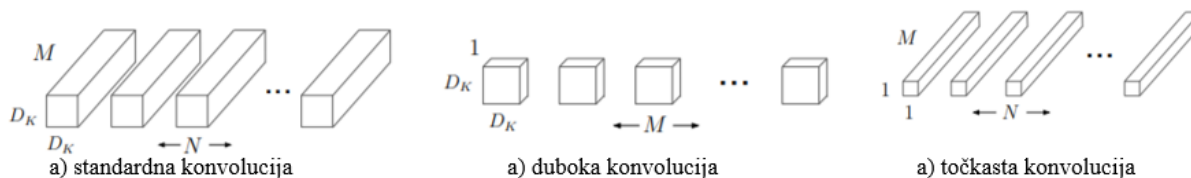
Kada su podaci ravnomjerno raspoređeni u svim kategorijama i kada imaju istu dimenziju moguće je započeti proces izgradnje modela.

### 4.3 MODEL ZA KLASIFIKACIJU DERMATOSKOPSKIH SLIKA

Nakon detaljne analize i pretprocesiranja podataka dolazi korak izgradnje modela za klasifikaciju. Osim HAM 10000 podataka, korišteno je transferirano učenje i MobileNet model konvolucijskih neuronskih mreža. Konvolucijske neuronske mreže često se koriste za klasifikaciju slika. Kod izgradnje modela za klasifikaciju konvolucijskih neuronskih mreža moguće je trenirati isključivo one podatke koji se koriste za klasifikaciju, a moguće je koristiti i transferirano učenje. Ovakav tip učenja podrazumijeva korištenje unaprijed istreniranih mreža koji se uz određene preinake mogu ponovno trenirati i prilagoditi konkretnom problemu. Jedna od takvih unaprijed istreniranih mreža jeste upravo MobileNet. U nastavku rada definirani su MobileNet konvolucijska mreža i proces izgradnje modela za klasifikaciju.

#### 4.3.1 MOBILE NET

MobileNet model baziran je na dubokoj (eng. depthwise) razdvojenoj konvoluciji koja predstavlja jedan oblik faktorizirane konvolucije. Ovakav oblik konvolucije faktorizira standardnu konvoluciju u duboku konvoluciju i  $1 \times 1$  konvoluciju koja se još naziva i točkasta konvolucija (eng. pointwise). Točkasta konvolucija primjenjuje  $1 \times 1$  konvoluciju kako bi kombinirala izlaz iz duboke konvolucije. Standardna konvolucija kombinira ulazne podatke u izlazne podatke pomoću filtera. Duboka konvolucija razdvaja isti proces u dva sloja, jedan za filtriranje i jedan za kombiniranje. Ovakva faktorizacija reducira vrijeme potrebno za treniranje i samu veličinu modela. Na slici 58. nalazi se grafički prikaz standardne i faktorizirane konvolucije.



Slika 58. Standardna, duboka i točkasta konvolucija

Struktura MobileNet mreže sastoji se od prethodno opisane duboke konvolucije osim prvog sloja koji sadrži standardnu konvoluciju. Arhitektura MobileNet mreže sadrži 28 slojeva ukoliko se slojevi duboke konvolucije i točkaste konvolucije broje posebno.

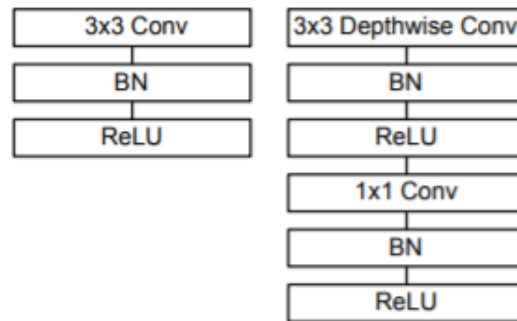
Arhitektura koristi i ReLU funkciju za nelinearnost osim posljednjeg sloja koji ne sadrži funkciju za nelinearnost i dodaje softmax sloj za klasifikaciju. Prije potpuno povezanog sloja funkcijom average pooling reducira se rezolucija u 1x1.<sup>29</sup> Arhitektura MobileNet mreže prikazana je na tablici 3.2.

Tip	Filter	Ulazna veličina
Conv/s2	3x3x3x32	224x224x3
Conv dw /s1	3x3x32dw	112x112x32
Conv /s1	1x1x32x64	112x112x32
Conv dw/s2	3x3x64dw	112x112x64
Conv/s1	1x1x64x128	56x56x64
Conv dw/s1	3x3x128dw	56x56x128
Conv/s1	1x1x128x256	56x56x128
Conv dw/s1	3x3x256dw	56x56x128
Conv/s1	1x1x256x256	56x56x128
Conv dw/s1	3x3x256dw	56x56x256
Conv/s1	1x1x256x512	56x56x256
Conv dw/s2	3x3x512dw	56x56x256
Conv/s1	1x1x256x512	14x14x256
5x Conv dw/s1	3x3x512dw	14x14x512
Conv/s1	1x1x512x1024	14x14x512
Conv dw/s2	3x3x512dw	14x14x512
Conv/s1	1x1x512x1024	7x7x512
Conv dw/s2	3x3x1024dw	7x7x1024
Conv/s1	1x1x1024x1024	7x7x1024
Avg Pool /s1	Pool 7x7	7x7x1024
FC / s1	1024x1000	7x7x1024
Softmax/s1	Klasifikator	1x1x10000

Tablica 4. Arhitektura MobileNet mreže

<sup>29</sup> (G. Howard et al., 2017)

Razlika između standardne i duboke razdvajajuće konvolucije s dubokim i točkastim slojem popraćena normalizacijom i ReLU funkcijom prikazana je na slici 59.



Slika 59. Razlika u standardnoj i duboko razdvajajućoj konvoluciji

#### 4.3.2 FIPU MODEL ZA KLASIFIKACIJU DERMATOSKOPSKIH SLIKA

U izgradnji modela prvi korak jeste definiranje skupa za treniranje i testiranje. Prethodno je u skup za treniranje smješteno približno 90% podataka, odnosno 9013 slika raspoređenih u 7 različitih kategorija. Skup za validaciju sadrži približno 10% podataka, odnosno 1002 slike.

Prije definiranja modela potrebno je definirati generatore podataka za treniranje i validaciju. Generatori predstavljaju važne karakteristike podataka koje je potrebno definirati prije treniranja modela, kao što su: lokacija podataka, dimenzija podataka i broj koraka u svakoj iteraciji tokom treniranja. Generator podataka na skupu za treniranje prikazan je na slici 60.

```

# Postaviti generatore za model
# U modelu FIPU koristit ćemo MobileNet mrežu
train_batches = ImageDataGenerator(
    preprocessing_function= \
        keras.applications.mobilenet.preprocess_input).flow_from_directory(
    train_path,
    target_size=(224, 224),
    batch_size=10)

```

Slika 60. ImageDataGenerator

(train\_path predstavlja putanju podataka za treniranje, target size predstavlja željenu dimenziju slika i batch\_size predstavlja broj koraka u svakoj iteraciji)

Nakon definiranih generatora potrebno je učitati model koji će biti korišten za klasifikaciju podataka. S obzirom da model FIPU koristi transferirano učenje učitat će MobileNet model koji je prethodno opisan. U slučaju da se model izrađuje u potpunosti samostalno ispočetka, učitao bi se tzv. sekvencijalni model (eng. sequential model).

```
#Učitati MobileNet model
mobile = keras.applications.mobilenet.MobileNet()
```

Slika 61. Učitavanje MobileNet modela

Nakon što je MobileNet model učitán potrebno ga je modificirati i trenirati na skupu za treniranje.<sup>30</sup>

Na modelu MobileNet isključeno je zadnjih 5 slojeva i dodan je dropout i dense sloj. Nakon modifikacije MobileNet modela potrebno je definirati koliko slojeva modela će biti trenirano s novim podacima za treniranje. U slučaju modela FIPU treniranje modela vršit će se na zadnja 23 sloja.

Prije treniranja modela potrebno je odabrati optimizator i kompajlirati model. U slučaju FIPU modela korišten je ADAM optimizator. Adam optimizator primjenjuje algoritam stohastičkog spuštavanja gradijenta i bazira se na adaptivnoj procjeni prvog i drugog trenutka. Jedan od argumenata funkcije za optimizaciju jeste learning rate, koji predstavlja stopu učenja u vremenu.

```
# Kompajliramo model i dodajemo Adam optimizator
model.compile(Adam(lr=0.01), loss='categorical_crossentropy', metrics=[categorical_accuracy, top_2_accuracy, top_3_accuracy])
```

Slika 62. Kompajliranje i optimizacija

Važan parametar za treniranje modela jesu težine svake klase. Na način da se svakoj klasi dodijeli različita težina moguće je postići da određene klase imaju prednost. U modelu FIPU težine svakoj klasi dodijeljene su po kriteriju opasnosti oboljenja i broja slika za treniranje. Tako je oboljenju melanom dodijeljena najveća težina. Prikaz težina svake kategorije u modelu FIPU nalazi se na slici 63.

---

<sup>30</sup> (Kyrkou, 2019)

```

#Težine svake kategorije
class_weights={
    0: 5.263406408094435, # akiec
    1: 4.4520156974669995, # bcc
    2: 5.233957219251337, # bkl
    3: 8.640767386091127, # df
    4: 9.992426850258177, # mel
    5: 3.207159429897249, # nv
    6: 3.7431334622823984, # vasc
}

```

Slika 63. Težine svake kategorije

Nakon što su definirane težine možemo nastaviti sa treniranjem modela. Parametri koje je potrebno definirati kod treniranja modela su: putanja podataka za treniranje i validaciju, broj koraka koji se izvode u svakoj epohi, broj epoha i funkciju za reduciranje stope učenja. U slučaju modela FIPU model je treniran na 5 epoha zbog ograničenosti resursa za treniranje. Funkcija za treniranje prikazana je na slici 64. Broj slika koje se treniraju u jednoj epohi jednak je broju ukupnih slika podijeljen sa parametrom `batch_size` što u ovom slučaju iznosi 902. Vrijeme potrebno za treniranje jedne epohe jeste približno 21min. Vrijeme za treniranje može se reducirati na način da se smanje dimenzije slika, ali u tom slučaju moguć je gubitak određenih značajki na slikama.<sup>31</sup>

```

# Treniramo model
history = model.fit_generator(train_batches,
                             steps_per_epoch=train_steps,
                             class_weight=class_weights,
                             validation_data=valid_batches,
                             validation_steps=val_steps,
                             epochs=5,
                             verbose=1,
                             callbacks=callbacks_list)

```

Slika 64. Funkcija za treniranje

<sup>31</sup> (Chaturvedi, Gupta and Prasad, 2019)

## 5. TESTIRANJE MODELA I PRIKAZ REZULTATA

Nakon treniranja modela potrebno je izvršiti evaluaciju kako bi analizirali rezultate. Parametri koji su korišteni za evaluaciju modela su:

- **validation categorical accuracy** koji možemo definirati kao procent točnosti modela, odnosno koliko često model točno klasificira slike
- **validation loss** koji možemo definirati kao parametar gubitka podataka
- **validation accuracy** predstavlja sposobnost modela da točno klasificira slike

Rezultati evaluacije modela FIPU prikazani su na slici 65.

```
val_loss: 0.030062701553106308
val_cat_acc: 0.5808383226394653
val_top_2_acc: 0.8273453116416931
val_top_3_acc: 0.9151696562767029
```

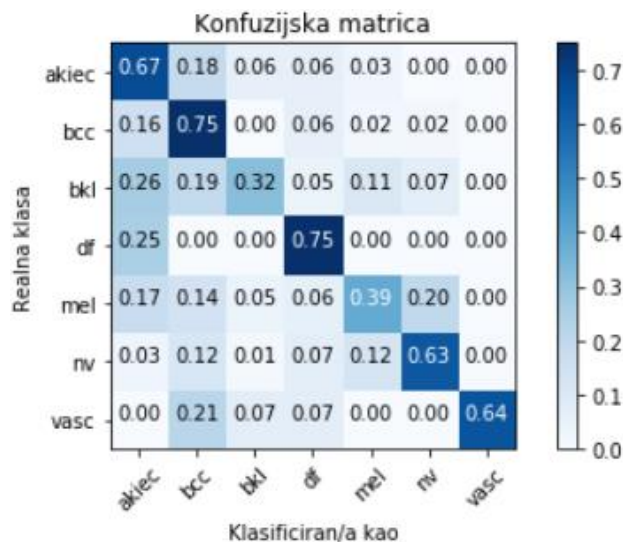
Slika 65. Rezultati evaluacije modela

U odnosu na prethodnu sliku, možemo zaključiti da model FIPU ima zadovoljavajuće rezultate. Validation top 2 accuracy iznosi 82%, dok je validation top 3 accuracy 91%. Categorical accuracy iznosi 58%, a gubitak podataka 0,03.

Nakon evaluacije rezultata modela možemo prijeći na korak predikcije podataka. Predikcija podrazumijeva testiranje modela na podacima koji nisu označeni u kategorije. Cilj predikcije podataka jeste postići što veći broj točno klasificiranih slika. Svaka slika procentualno se raspoređuje u 7 različitih kategorija. Sliku klasificiramo u onu kategoriju za koju model dodijeli najveći procent. Broj slika korištenih za predikciju jeste 101.

Nakon završetka predikcije rezultati su prikazani grafički pomoću konfuzijske matrice. Konfuzijska matrica prikazuje broj točno klasificiranih slika. Za svaku klasu prikazuje procent točno i netočno klasificiranih slika. Konfuzijska matrica modela FIPU prikazana je na slici 66.





Slika 66. Konfuzijska matrica modela FIPU

U klasi **aktinična keratoza** točno je klasificirano 67% slika. 18% slika koji pripadaju klasi aktinična keratoza klasificirano je kao bazocelularni karcinom, 6% kao benigna keratoza, 6% kao dermatofibrom i 3% kao melanom.

U klasi **bazocelularni karcinom** točno je klasificirano 75% podataka. 16% slika koje pripadaju klasi bazocelularni karcinom klasificirano je kao aktinična keratoza, 6% kao dermatofibrom, 2% kao melanom i 2% kao melanocitni nevus.

U klasi **benigna keratoza** točno je klasificirano 32% podataka. 26% slika koje pripadaju klasi benigna keratoza klasificirano je kao aktinična keratoza, 19% kao bazocelularni karcinom, 5% kao dermatofibrom, 11% kao melanom i 7% kao melanocitni nevus.

U klasi **dermatofibrom** točno je klasificirano 75% podataka. 25% slika koje pripadaju klasi dermatofibrom klasificirano je kao aktinična keratoza.

U klasi **melanom** točno je klasificirano 39% slika. 17% slika koje pripadaju klasi melanom klasificirani su kao aktinična keratoza, 14% kao bazocelularni karcinom, 5% kao benigna keratoza, 6% kao dermatofibrom i 20% kao melanocitni nevus.

U klasi **melanocitni nevus** točno je klasificirano 63% podataka. 3% podataka koji pripadaju klasi melanocitni nevus klasificirano je kao aktinična keratoza, 12% kao

bazocelularni karcinom, 1% kao benigna keratoza, 7% kao dermatofibrom i 12% kao melanom.

U klasi **vaskularna kožna oboljenja** točno je klasificirano 64% podataka. 21% podataka koji pripadaju klasi vaskularna kožna oboljenja klasificirano je kao bazocelularni karcinom, 7% kao benigna keratoza i 7% kao dermatofibrom.

## 5.1 BUDUĆA POBOLJŠANJA FIPU MODELA

U prethodno opisanim rezultatima možemo zaključiti da većinski broj slika svake kategorije pripada točnoj kategoriji. U istraživanju iz 2018. godine navedeno je da dermatolozi imaju mogućnost od 65-80% točno klasificirati melanom u odnosu na druga kožna oboljenja.<sup>32</sup> Kada se radi o klasifikaciji u više kategorija, i sami dermatolozi primjenjuju druge metode kako bi utvrdili o kojem se kožnom oboljenju radi. Najveći nedostatak modela FIPU jeste poprilično mali postotak točno klasificiranih slika kod kategorije melanom. 39% točno klasificiranih slika u slučaju melanoma nije zadovoljavajući rezultat. Kako bi model postigao bolje rezultate potrebno je prikupiti veći broj dermatoskopskih slika. Bolja predikcija kategorija postigla bi se većim skupom za treniranje. Ukoliko bi se prikupilo više dermatoskopskih slika za treniranje određenih kategorija kao što je melanom, model bi imao šansu da postane prvi korak u analizi kožnih oboljenja. Na taj način mogao bi doprinijeti znatnim uštedama novca u dermatološkim ordinacijama. Nakon što model klasificira kožno oboljenje, pacijent bi se uputio na daljnja istraživanja u određenoj oblasti. Tako bi prvu posjetu dermatologu zamijenio model za klasifikaciju kožnih oboljenja. U periodu od 2008-2018.godine godišnji broj oboljelih od melanoma povećao se za 53% zbog prevelikog izlaganja UV zrakama. Zbog znatno većeg broja kožnih oboljenja nastale su velike gužve u dermatološkim ordinacijama. Tako je u Istarskoj županiji pregled kod dermatologa moguće čekati i do nekoliko mjeseci, koji

---

<sup>32</sup> (Brinker et al., 2018 str.2)

kod određenog oboljenja mogu biti kobni po pacijenta. Model za klasifikaciju kožnih oboljenja ukoliko pokaže zadovoljavajuće rezultate, može doprinijeti znatnim uštedama vremena i novca.

Kao inspiracija za temu istraživanja korišten je vodič za duboko učenje u oblasti medicine.<sup>33</sup>

Za izgradnju modela korišten je jupyter notebook i paket KERAS.<sup>34</sup>

Jupyter notebook koji sadrži izvorni kod modela FIPU dostupan je na slijedećoj poveznici:<https://github.com/LejlaMehinovic/FIPU-model-za-klasifikaciju-dermatoskopskih-slika.git> .

Kao nedostatak HAM 10000 skupa podataka navodim činjenicu da su sva oboljenja prikazana na koži svijetle puti. Kako bi model mogao pokazati zadovoljavajuće rezultate potrebno je prikupiti dermatoskopske slike kožnih oboljenja na koži tamnije puti.

---

<sup>33</sup> (Misra, 2020)

<sup>34</sup> (Team, 2020)

## 6. ZAKLJUČAK

Čovjek se svakodnevno susreće s velikim brojem zadataka i obaveza. Nerijetko se dešava da pokušavamo učiniti više od onoga što je u našim mogućnostima u određenom vremenu. Samim tim u svakom poslu dešavaju se greške. Pitanje koje se postavlja jeste koliko greški možemo sebi dopustiti? Odgovor na ovo pitanje ovisi od vrste i oblika zadatka. Ono što je sigurno jeste to da čovjek u mnogim oblastima treba pomoć koju mu računalo može pružiti. Klasifikacija objekata jedan je tip zadatka s kojim se svakodnevno čovjek susreće. Ukoliko svaki dan vršimo klasifikaciju objekata na isti način, zašto ne bismo naučili računalo da taj zadatak obavlja umjesto nas? U tome nam pomažu umjetna inteligencija i konvolucijske neuronske mreže koje su detaljno obrađene u ovom radu.

Oblast koja zasigurno zahtijeva pomoć umjetne inteligencije jeste medicinska dijagnostika. S obzirom na prečesto izlaganje suncu i štetne UV zrake kožna oboljenja su zastupljena sve više. Kako bi se utvrdila vrsta kožnog oboljenja pacijenti moraju obaviti pretrage koje nerijetko uključuju dermatoskopske slike. Liječnici na osnovu dermatoskopskih slika klasificiraju oboljenja. Ono što predstavlja problem ovakvog pristupa jeste vrijeme potrebno za dijagnozu oboljenja. Kako su kožna oboljenja sve više zastupljena, liječnici su poprilično zauzeti i ne stignu pregledati sve pacijente u vremenu koje bi bilo optimalno za određenu vrstu oboljenja. U ovom slučaju ispaštaju i pacijent i liječnik. Model za klasifikaciju kožnih oboljenja u 7 najčešćih kategorija doprinosi vremenu potrebnom za dijagnostiku. Modeli za klasifikaciju dermatoskopskih slika u današnjici uspjeli su ostvariti točnost u više od 90% slučajeva što je više nego zadovoljavajući rezultat. Ono što modelima predstavlja najveći problem jeste nedovoljan broj dermatoskopskih slika za učenje. S obzirom na sve veći broj kožnih oboljenja, postoji mogućnost da se prikupi veliki broj slika koje mogu služiti za učenje modela. Ovakvi modeli nemaju cilj da zamijene liječnike u oblasti dermatoskopije, nego da im pruže neophodnu pomoć koja bi najviše doprinijela samim pacijentima. Model FIPU koji je predstavljen i opisan u ovom radu uz odgovarajuća poboljšanja ima šansu da postane jedan korak u oblasti dermatoskopije.

## LITERATURA

1. Nielsen, M., 2020. **Neural Networks And Deep Learning**.
2. Sharma V, A., 2020. **Understanding Activation Functions In Neural Networks**. [online] Medium. Dostupno na: <https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0> [23 ožujak 2020].
3. Dalbello Bašić, p., Čupić, m. i Šnajder, m., 2008. **Umjetne Neuronske Mreže**. 1. izdanje [elektronska knjiga] Zagreb, pp.10 11. Dostupno na: <https://www.fer.hr/download/repository/UmjetneNeuronskeMreze.pdf> [23 ožujak 2020].
4. Taguri, Y., Erlichmen, S. and Lussato, R., 2020. **7 Types Of Activation Functions In Neural Networks: How To Choose?**. [online] MissingLink.ai. Dostupno na: <https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/> [24 ožujak 2020].
5. MachineCurve. 2020. **How Does The Softmax Activation Function Work? – Machinecurve**. [online] Dostupno na: <https://www.machinecurve.com/index.php/2020/01/08/how-does-the-softmax-activation-function-work/> [25 ožujak 2020].
6. Depositphotos, I., 2020. **Stock Photos, Royalty Free Images, Vectors, Footage | Depositphotos**. [online] Depositphotos. Dostupno na: [https://depositphotos.com/?gclid=aw.ds&&utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=DP\\_EU\\_EN\\_Brand\\_Search&utm\\_term=deposit%20photos&qclid=Cj0KCQjw0Mb3BRCaARIsAPSNGpXjIK-gkDQenFPynRSrj-](https://depositphotos.com/?gclid=aw.ds&&utm_source=google&utm_medium=cpc&utm_campaign=DP_EU_EN_Brand_Search&utm_term=deposit%20photos&qclid=Cj0KCQjw0Mb3BRCaARIsAPSNGpXjIK-gkDQenFPynRSrj-) [25 ožujak 2020].
7. Gharat, S., 2020. **What, Why And Which?? Activation Functions**. [online] Medium. Dostupno na: <https://medium.com/@snaily16/what-why-and-which-activation-functions-b2bf748c0441> [25 ožujak 2020].
8. Haykin, S. (2009). **Neural Networks and Learning Machines**. Third Edition. PEARSON, Prentice Hall

9. Peterroelants.github.io. 2020. ***How To Implement A Neural Network - Gradient Descent.*** [online] Dostupno na: <https://peterroelants.github.io/posts/neural-network-implementation-part01/> [8 travanj 2020].
10. Fehlh Haber, K., 2020. *Hubel And Wiesel & The Neural Basis Of Visual Perception - Knowing Neurons.* [online] Knowing Neurons. Dostupno na: <https://knowingneurons.com/2014/10/29/hubel-and-wiesel-the-neural-basis-of-visual-perception/> [20 Travanj 2020].
11. Pattanayak, S., 2018. *Pro Deep Learning With Tensorflow.* Berkeley, CA: Apress L.P.
12. Goodfellow, I., Bengio, Y. and Courville, A., 2014. *Deep Learning.* MIT Press.
13. Cs231n.github.io. 2020. *Convolutional Neural Networks For Visual Recognition.* [online] Dostupno na: <https://cs231n.github.io/convolutional-networks/> [5 svibanj 2020].
14. Binder, M., Steiner, A., Schwarz, M., Knollmayer, S., Wolff, K. and Pehamberger, H., 1994. *Application Of An Artificial Neural Network In Epiluminescence Microscopy Pattern Analysis Of Pigmented Skin Lesions: A Pilot Study.*
15. Mendonca, T., Ferreira, P., Marques, J., Marcal, A. and Rozeira, J., 2013. *PH<sup>2</sup> - A Dermoscopic Image Database For Research And Benchmarking.*
16. Tschandl, P., Rosendahl, C. and Kittler, H., 2018. *The HAM10000 Dataset, A Large Collection Of Multi-Source Dermatoscopic Images Of Common Pigmented Skin Lesions.*
17. Marcelino, P., 2019. *Comprehensive Data Exploration With Python.* [online] Kaggle.com. Dostupno na : <https://www.kaggle.com/pmarcelino/comprehensive-data-exploration-with-python> [16 lipanj 2020].

18. G. Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.
19. Kyrkou, C., 2019. *Changing Input Size Of Pre-Trained Models In Keras*. [online] Medium. Dostupno na: <https://medium.com/@ckyrkou/changing-input-size-of-pre-trained-models-in-keras-3dfbe3ca3091> [18 lipanj 2020].
20. Chaturvedi, S., Gupta, K. and Prasad, P., 2019. Skin Lesion Analyser: An Efficient Seven-Way MultiClass Skin Cancer Classification Using MobileNet.
21. Brinker, T., Hekler, A., Utikal, J., Grabe, N., Schadendorf, D., Klode, J., Berking, C., Steeb, T., Enk, A. and von Kalle, C., 2018. Skin Cancer Classification Using Convolutional Neural Networks: Systematic Review. *Journal of Medical Internet Research*, 20(10), p.e11936.
22. Misra, A., 2020. *Classifying Skin Lesions With Convolutional Neural Networks*. Medium. Dostupno na: <https://towardsdatascience.com/classifying-skin-lesions-with-convolutional-neural-networks-fc1302c60d54> [18 lipanj 2020].
23. Team, K., 2020. *Keras Documentation: About Keras*. [online] Keras.io. Dostupno na: <https://keras.io/about/> [19 lipanj 2020].

## POPIS SLIKA

1. Slika 1. Građa biološkog neurona. Preuzeto sa: <https://www.neuroscientificallychallenged.com/glossary/neuron> (17.03.2020)
2. Slika 2. Građa umjetnog neurona. Izradio autor.
3. Slika 3. Jednoslojna mreža bez povratnih veza. Izradio autor.
4. Slika 4. Višeslojna mreža bez povratnih veza. Izradio autor.
5. Slika 5. Funkcija skoka. Prilagođeno prema: (Depositphotos, 2020)
6. Slika 6. Linearna funkcija. Prilagođeno prema: (Depositphotos, 2020)
7. Slika 7. Sigmoidna funkcija. Prilagođeno prema: (Depositphotos, 2020)
8. Slika 8. Tahn funkcija. Prilagođeno prema: (Depositphotos, 2020)
9. Slika 9. ReLu funkcija. Prilagođeno prema: (Depositphotos, 2020)
10. Slika 10. Leaky ReLu funkcija. Prilagođeno prema: (Depositphotos, 2020)
11. Slika 11. Softmax funkcija. Prilagođeno prema: (Depositphotos, 2020)
12. Slika 12. Učenje s učiteljem Preuzeto od (Haykin, 2009 – 35 str.)
13. Slika 13. Pojačano učenje Preuzeto od (Haykin, 2009 – 36 str.)
14. Slika 14. Nenadzirano učenje Preuzeto od (Haykin, 2009 – 37 str.)
15. Slika 15. Umjetni neuron Preuzeto od (Nielsen, 2020 – 62 str.)
16. Slika 16. Pravilo lanca. Izradio autor.
17. Slika 17. Ulazni-izlazni sustav. Preuzeto od (Pattanayak, 2018 – 154 str.)
18. Slika 18. Ulazni signali. Preuzeto od (Pattanayak, 2018 – 156 str.)
19. Slika 19. Funkcije za računanje operacije konvolucije. Preuzeto od (Pattanayak, 2018 – 156 str.)
20. Slika 20. Izlazni signali  $y(t)$  Preuzeto od (Pattanayak, 2018 – 157 str.)
21. Slika 21. Preklapanje funkcija u konvoluciji za  $t=2$  Preuzeto od (Pattanayak, 2018 – 158 str.)
22. Slika 22. 2D signal. Izradio autor.
23. Slika 23. Primjer 2D konvolucije bez okretanja filtera. Preuzeto od (Goodfellow, Bengio and Courville, 2016-334.str )



24. Slika 24. Filter srednje vrijednosti na slici Mona Lise. Preuzeto od (Pattanayak, 2018 – 171 str.)
25. Slika 25. Primjena medijan filtera na sliku Mona Lise Preuzeto od (Pattanayak, 2018 – 172 str.)
26. Slika 26. Primjena Gaussovog filtera na sliku Mona Lise. Preuzeto od (Pattanayak, 2018 – 173 str.)
27. Slika 27. Primjena sobel filtera za detekciju rubova .Preuzeto od (Pattanayak, 2018 – 177 str.)
28. Slika 28. Raspršena povezanost-ulazna jedinica  $x_3$  i izlazne jedinice  $s$  na koje  $x$  ima utjecaj
  - a)  $S$  je formiran konvolucijom filtera širine 3,  $x$  ima utjecaj na samo tri neurona
  - b)  $S$  je formiran množenjem matrica i povezanost nije raspršena što implicira da  $x$  ima utjecaj na sve neurone. Preuzeto od Goodfellow, Bengio and Curville, 2016 – 336 str
29. Slika 29. Primjer volumena ulaznog sloja i volumena neurona u prvom konvolucijskom sloju. Izradio autor.
30. Slika 30. Ilustracija prostornog rasporeda neurona na  $x$  osi. Izradio autor.
31. Slika 31. Komponente sloja konvolucijske neuronske mreže. Izradio autor.
32. Slika 32. Sažimanje maksimalnih vrijednosti sa  $2 \times 2$  filterom i korakom 2. Izradio autor.
33. Slika 33. Propagacija unatrag kroz sloj konvolucije. Preuzeto od (Pattanayak, 2018 – 182 str.)
34. Slika 34. Propagacija unatrag kroz sloj sažimanja maksimalnih vrijednosti. Preuzeto od (Pattanayak, 2018 – 186 str.)
35. Slika 35. Propagacija unatrag kroz sloj prosječnog sažimanja. Preuzeto od (Pattanayak, 2018 – 187 str.)
36. Slika 36. LeNet5 arhitektura. Preuzeto od (Pattanayak, 2018 – 206 str.)
37. Slika 37. AlexNet arhitektura. Preuzeto od (Pattanayak, 2018 – 209 str.)
38. Slika 38. RGGF arhitektura. Preuzeto od (Pattanayak, 2018 – 210 str.)
39. Slika 39. Rezidualni blok. Preuzeto od (Pattanayak, 2018 – 211 str.)
40. Slika 40. Prikaz binarne klasifikacije. Izradio autor.

41. Slika 41. Prikaz klasifikacije u više kategorija. Izradio autor.
42. Slika 42. Aktinična keratoza. Preuzeto iz HAM 10000 skupa podataka.
43. Slika 43. Bazocelularni karcinom. Preuzeto iz HAM 10000 skupa podataka.
44. Slika 44. Benigna keratoza. Preuzeto iz HAM 10000 skupa podataka.
45. Slika 45. Dermatofibrom. Preuzeto iz HAM 10000 skupa podataka.
46. Slika 46. Melanocitni nevus. Preuzeto iz HAM 10000 skupa podataka.
47. Slika 47. Melanom. Preuzeto iz HAM 10000 skupa podataka.
48. Slika 48. Vaskularna kožna oboljenja. Preuzeto iz HAM 10000 skupa podataka.
49. Slika 49. HAM10000\_metadata.csv
50. Slika 50. Analiza oboljenja i životne dobi. Izradio autor.
51. Slika 51. oboljenja u odnosu na životnu dob. Izradio autor.
52. Slika 52. Analiza lokacije oboljenja na tijelu i spola. Izradio autor.
53. Slika 53. Oboljenja u odnosu na spol. Izradio autor.
54. Slika 54. Broj slika u kategorijama za treniranje i validaciju. Izradio autor.
55. Slika 55. Prikaz direktorija za treniranje i validaciju. Izradio autor.
56. Slika 56. Broj slika u kategorijama skupa za treniranje. Izradio autor.
57. Slika 57. Broj slika nakon augmentacije. Izradio autor.
58. Slika 58. Standardna, duboka i točkasta konvolucija. Preuzeto od (g. Hoiward et al. 2017 str 3.)
59. Slika 59. Razlika u standardnoj i duboko razdvajajućoj konvoluciji. Preuzeto od (g. Hoiward et al. 2017 str 4.)
60. Slika 60. Image Dana Generator. Izradio autor.
61. Slika 61. Učitavanje MobileNet modela. Izradio autor.
62. Slika 62. Kompajliranje i optimizacija. Izradio autor.
63. Slika 63. Težine svake kategorije. Izradio autor.
64. Slika 64. Funkcija za treniranje. Izradio autor.
65. Slika 65. Rezultati evaluacije modela. Izradio autor.
66. Slika 66. Konfuzijska matrica modela FIPU. Izradio autor.

## POPIS TABLICA

1. Tablica 1. Razlike između softmax i sigmoidne funkcije. Izradio autor.
2. Tablica 2. Filteri i njihova primjena. Izradio autor.
3. Tablica 3. Opis parametara iz HAM10000\_metadata.csv Izradio autor.
4. Tablica 4. Arhitektura MobileNet mreže. Prilagođeno prema: (G. Howard et al., 2017 str 4.)

## SAŽETAK

Ovaj diplomski rad bavi se konvolucijskim neuronskim mrežama za klasifikaciju objekata. U prvom dijelu diplomskog rada obrađen je pojam umjetnih neuronskih mreža, povijest neuronskih mreža, arhitektura neuronskih mreža, vrste aktivacijskih funkcija, proces učenja neuronskih mreža i algoritam propagacije pogreške unatrag. U drugom dijelu diplomskog rada obrađen je pojam konvolucijskih neuronskih mreža, funkcija konvolucije, filteri za procesuiranje slika, arhitektura konvolucijskih neuronskih mreža, algoritam propagacije pogreške unatrag kod konvolucijskih neuronskih mreža kao i nekoliko primjera konvolucijskih neuronskih mreža. U trećem dijelu diplomskog rada prikazana je praktična primjena konvolucijskih neuronskih mreža za klasifikaciju objekata koja se izvodi u nekoliko koraka: analiza podataka HAM 10000, pretprocesiranje podataka za izgradnju modela, izgradnja modela, testiranje modela i prikaz rezultata kao i buduća poboljšanja modela. Cilj rada je detaljno objasniti pojam konvolucijskih neuronskih mreža i primijeniti iste na vrlo važan problem klasifikacije kožnih oboljenja.

Ključne riječi: umjetne neuronske mreže, konvolucijske neuronske mreže, klasifikacija, HAM 10000 skup podataka

## ABSTRACT

This graduate thesis deals with convolutional neural networks for object classification problem. Part one of this graduate thesis deals with artificial neural networks, their history, their architecture, activation functions, process of learning and backpropagation algorithm. Part two of this graduate thesis deals with convolutional neural networks, convolution, picture processing filters, architecture of convolutional neural networks, backpropagation algorithm and a few examples of convolutional neural networks. Part three of this graduate thesis deals with convolutional neural networks for object classification in a few steps: analysing HAM 10000 dataset, preprocessing of the data, building model for object classification and analysing results with future plans. The aim of this graduate thesis was to explain everything about convolutional neural networks and use them on very important problem of skin lesions classification.

Keywords: artificial neural networks, convolutional neural networks, classification, HAM 10000 dataset