

Detekcija metastatskog tkiva na patohistološkim slikama presjeka limfnih čvorova uporabom dubokog učenja

Miličević, Andrija

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, School of Medicine / Sveučilište u Zagrebu, Medicinski fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:105:790669>

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

Download date / Datum preuzimanja: **2025-03-31**



Repository / Repozitorij:

[Dr Med - University of Zagreb School of Medicine Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
MEDICINSKI FAKULTET**

Andrija Miličević

**Detekcija metastatskog tkiva na
patohistološkim slikama presjeka limfnih
čvorova uporabom dubokog učenja**

DIPLOMSKI RAD



Zagreb, 2019.

Ovaj diplomski rad izrađen je u Zavodu za patologiju Medicinskog fakulteta Sveučilišta u Zagrebu pod vodstvom doc. dr. sc. Gorane Aralice i predan je na ocjenu u akademskoj godini 2018./2019.

SADRŽAJ

| | |
|--|-----------|
| 1. Uvod | 1 |
| 1.1. Strojno učenje | 1 |
| 1.2. Primjena strojnog učenja u patologiji | 3 |
| 2. Definicija problema | 5 |
| 2.1. Generalizacija algoritma strojnog učenja | 5 |
| 2.2. Skup podataka | 6 |
| 2.3. Ulaz, izlaz, funkcija gubitka i procjena generalizacije | 7 |
| 3. Hardverska i programska potpora | 9 |
| 3.1. Korišteni alati | 9 |
| 4. Duboko učenje | 11 |
| 4.1. Potpuno povezani sloj | 13 |
| 4.2. Konvolucijski sloj | 14 |
| 4.3. Sloj sažimanja | 16 |
| 5. Predloženi model i rezultati | 17 |
| 6. Zaključak | 20 |
| Zahvale | 21 |
| Literatura | 22 |
| Životopis | 25 |

Detekcija metastatskog tkiva na patohistološkim slikama presjeka limfnih čvorova uporabom dubokog učenja

Sažetak

Algoritmi umjetne inteligencije omogućuju računalima obavljanje inteligentnih zadataka. Veliki doprinos umjetne inteligencije očekuje se u medicini gdje bi ovakvi algoritmi mogli postati standard u svakodnevnom radu liječnika. Patologija je grana medicine koja se bavi proučavanjem bolesti, a velik dio posla odnosi se na mikroskopsku dijagnostiku bolesti, što znači da patolozi rade sa slikama. Duboke neuronske mreže inspirirane su načinom kako mozak obrađuje informacije, a posebno su prikladne za rad sa slikama. U ovom radu predložena je metoda za detekciju metastatskog tkiva na patohistološkim slikama presjeka limfnih čvorova, a temelji se na dubokim neuronskim mrežama. Predložena metoda postiže točnost od 86,57% na PatchCamelyon skupu podataka, a uspoređena je s drugim objavljenim metodama. PatchCamelyon je skup podataka od 327 680 slika u boji, veličine 96 x 96. Za svaku sliku, označeno je postojanje metastatskog tkiva u središnjem dijelu slike. Ovaj skup podataka nastao je nasumičnim izrezivanjem segmenata 400 preparata iz Camelyon16 skupa podataka. Augmentacija PatchCamelyon skupa podataka značajno utječe na točnost predložene metode. Korištenje većeg skupa podataka za učenje moglo bi dodatno povećati točnost predložene metode. Ovom metodom moguće je obraditi 250 slika u sekundi na prosječnom računalu. Podizanjem praga klasifikacije moguće je povećati preciznost predložene metode i koristiti je za brzu detekciju metastaza nad cijelim slikama patohistoloških presjeka limfnih čvorova. Ova metoda mogla bi ubrzati rad patologa označavanjem regija preparata koji nisu dio normalnog tkiva limfnog čvora.

Ključne riječi: detekcija metastatskog tkiva, duboke neuronske mreže, patologija

Metastatic Tissue Detection in Histopathological Images of Lymph Node Sections Using Deep Learning

Summary

Artificial intelligence algorithms allow computers to perform intelligent tasks. Big impact is expected in medicine where such algorithms could become a standard in the day-to-day work of doctors. Pathology is a branch of medicine that studies disease. A big part of it is related to microscopic disease diagnostics, which means pathologist work with images. Deep neural networks are inspired by the way the brain processes information and are particularly suitable for working with images. In this thesis, a method for detecting metastatic tissue in histopathological images of lymph node sections using deep neural networks is proposed. The proposed method achieves 86,57% accuracy on the PatchCamelyon data set and is compared with other published methods. PatchCamelyon is a collection of 327 680 color images with dimensions 96 x 96. For each image, there are annotations of metastatic tissue presence in the central image region. This data set was generated by extracting segments of 400 whole-slide images from the Camelyon16 data set. Augmentation of the PatchCamelyon data set significantly influences the accuracy of the proposed method. Using a larger training data set could increase the accuracy of this method. This method can be used on an average computer in real-time, as it can process 250 images per second. By increasing the classification threshold, it is possible to increase the accuracy of the proposed method so it can be used for fast detection of metastases in whole-slide images of lymph nodes. This method could accelerate the work of pathologists by marking regions of a whole-slide image which are not part of the normal lymph node tissue.

Keywords: metastatic tissue detection, deep neural networks, pathology

1. Uvod

Umjetna inteligencija je znanost o tome kako postići da strojevi izvode zadatke koji bi, kada bi ih radio čovjek, iziskivali inteligenciju (1).

Umjetna inteligencija je grana računarstva koja omogućuje računalima inteligentno ponašanje. Postoji generalna podjela na jaku i slabu umjetnu inteligenciju. Jaka umjetna inteligencija u svojem opisu sadrži ideju da je računalo zapravo um koji posjeduje svjesnost. Takvo računalo bilo bi funkcionalno razvijeno barem kao čovjek. No, kada govorimo o umjetnoj inteligenciji u užem smislu, najčešće mislimo na slabu umjetnu inteligenciju. Slaba umjetna inteligencija odnosi se na učenje računala obavljanju jasno specificiranih zadataka za koje se smatra kako iziskuju inteligenciju (2). Umjetna inteligencija se uspješno koristi za zadatke poput otkrivanja neželjene elektroničke pošte (3), detekcije kartičnih prijevara (4), predviđanje kretanja cijena dionica (5) i prepoznavanje lica (6). Sam koncept umjetne inteligencije nije toliko nov. John McCarthy je još 1955. osmislio naziv, a u ljeto 1956. organizirao je poznatu Dartmouth konferenciju, gdje je umjetna inteligencija po prvi puta definirana kao znanstvena disciplina (7). Razlog zašto je toliko vremena bilo potrebno do procvata umjetne inteligencije, leži u eksponencijalnom rastu računalne moći i generirane količine podataka. Najpopularniji pristupi umjetne inteligencije najčešće zahtijevaju veliku količinu računalne snage i veliku količinu podataka za rješavanje korisnih zadataka.

1.1. Strojno učenje

Strojno učenje je dio umjetne inteligencije čiji je cilj programiranje računala kako bi imali sposobnost učenja iz podatkovnih primjera ili prethodnog iskustva te mogli riješiti neki problem (8). Strojno učenje se najčešće koristi za rješavanje vrlo usko strukturiranih zadataka. Problem prepoznavanja objekata na slici trenutno se najuspješnije rješava strojnim učenjem. CIFAR-10 je primjer skupa podataka namijenjenog za strojno učenje, a koristi se kao mjerilo, tj. za usporedbu različitih algoritama strojnog učenja za prepoznavanje objekata. Skup podataka se sastoji od 60 000 slika s pripadajućim oznakama, tj. za svaku sliku je poznato što se na njoj nalazi. Postoji 10 vrsta oznaka: avion, automobil, ptica, mačka, jelen, pas, žaba, konj, broj i kamion (9). Trenutno najbolje rješenje metodama strojnog učenja postiže točnost prepoznavanja objekta, na prethodno navedenom skupu podataka, od 99%. Postoji i

proširena verzija skupa podataka sa 100 vrsta oznaka, a najbolje rješenje postiže točnost od 91,3% (10). Rješenje je osmislio Google-ov istraživački tim.

Strojno učenje omogućuje računalu da samo nauči što je značajno u podacima te na koji način su povezanosti određenih varijabli značajne. Upravo se u medicini očekuje značajan doprinos i transformacija prelaskom iz ekspertnog sustava u principe strojnog učenja. Strojno učenje će omogućiti individualizirani pristup bolesniku, a sve to na temelju analize ogromne količine podataka (11).

Postoje tri glavne vrste strojnog učenja: nadzirano, nenadzirano i podržano učenje. Cilj nadziranog učenja je naučiti funkciju(f) koja preslikava ulazne vrijednosti u izlazne vrijednosti:

$$y = f(x),$$

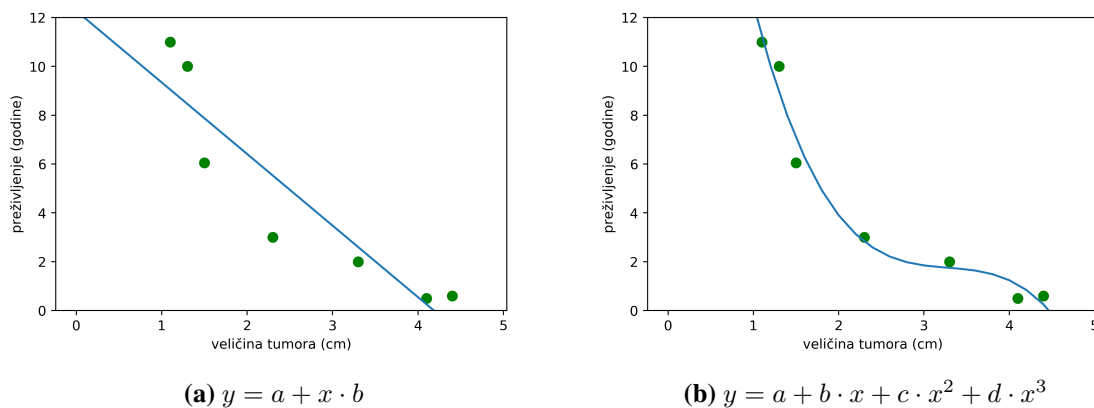
pri čemu je x vektor ulaznih vrijednosti, a y vektor izlaznih vrijednosti. Za probleme kojima je y vektor realnih brojeva kažemo da su regresijski problemi. Klasifikacijski problemi su oni kojima je y diskretna vrijednost koja predstavlja kategoriju ulaznog primjera. Kod nenadziranog učenja, podaci za učenje sastoje se samo od ulaznih vektora x , a cilj je pronaći pravilnosti u podacima. Jedan od primjera je automatizirano grupiranje podataka. Podržano učenje sastoji se od pronalaska optimalnih akcija u određenim situacijama, a temelji se na metodi pokušaja i pogrešaka (12). Igranje igara upravo je dobar primjer za podržano učenje. Prije par godina Google-ov tim je uspio razviti sustav koji igra poznatu igru *Go* bolje nego najbolji čovjek, a temelji se na podržanom učenju (13). *Go* je igra koja je za računalo puno kompleksija nego šah.

Nadziranim učenjem moguće je naučiti računalo predviđanje preživljenja bolesnika na temelju veličine tumora, zahvaćenosti limfnih čvorova i postojanju metastaza. Računalo na temelju prethodno viđenih primjera, za koje je poznato preživljenje, uči funkciju(f) koja preslikava parametre tumora u realnu vrijednost koja predstavlja predviđeno preživljenje. Kvaliteta naučene funkcije f jako ovisi o količini podataka, ali i varijablama koje se nalaze u podacima. Pri nadziranom učenju potrebno je unaprijed odabrati model. Model je zapravo skup funkcija f , među kojima je učenjem potrebno pronaći onu koja najbolje opisuje podatke. Pojednostavljena verzija prethodno navedenog primjera je predviđanje preživljenja samo na temelju veličine tumora. x je u ovom slučaju samo jedna vrijednost, tj. realan broj koji opisuje veličinu tumora, a y realan broj opisuje preživljenje. Vrlo jednostavan i intuitivan model za ovaj problem opisan je izrazom u nastavku:

$$y = a + b \cdot x,$$

pri čemu je x veličina tumora, a a i b parametri modela koje je potrebno pronaći procesom učenja. Svaka kombinacija određuje točno jednu funkciju f , a odabirom ovakvog modela uvedena je restrikcija pri čemu funkcija f mora biti u obliku pravca. Slika 1.1 prikazuje primjer podataka i najbolje funkcije za model koji je pravac, tj. polinom prvog stupnja te model

koji je krivulja, tj. polinom trećeg stupnja. Iz ovoga je vidljivo kako je nadzirano učenje zapravo optimizacijski problem pronalaska najboljih parametra te kako je uz odabir modela potrebno, također unaprijed, odrediti kako mjeriti kvalitetu funkcije f , tj. odrediti funkciju gubitka i način na koji će se parametri istraživati, tj. algoritam učenja. Cilj strojnog učenja je pronaći funkciju f koja najbolje generalizira, a u prikazanom slučaju je premalo podataka za usporedbu različitih modela. Moguće je da je drugi model prenaučio neke rubne primjere te zapravo radi puno lošije od jednostavnijeg modela. Također, sigurno je moguće preciznije odrediti preživljenje ako se kao ulazi dodatno koriste i drugi podaci, poput zahvaćenosti limfnih čvorova i postojanje metastaza.



Slika 1.1: Primjer odabira dva različita modela za predviđanje preživljenja na temelju veličine tumora. Napomena: podaci su izmišljeni, a primjer pojednostavljen.

1.2. Primjena strojnog učenja u patologiji

Postavljanje kliničke dijagnoze vrlo je zahtjevan problem. Potrebno je dugotrajno obrazovanje i veliko iskustvo kako bi liječnik bio dobar kliničar i vješt u postavljanju kliničke dijagnoze. Ovo je vrlo težak problem za računalo. Za strojno učenje bitno je jasno definirati što su ulazne, a što izlazne vrijednosti, tj. varijable. Pri postavljanju kliničke dijagnoze vrlo je teško odrediti što bi točno bile ulazne varijable. Kliničar može ispitati bolesnika na različite načine, napraviti klinički pregled te uputiti na daljnju dijagnostiku. Strojno učenje je izvrsno za rješavanje vrlo specifičnih, tj. jasno definiranih zadataka u kojima možemo jasno definirati ulazne varijable. Postavljanje kliničke dijagnoze zasada to još uvijek nije.

Patologija je znanost koja se bavi proučavanjem bolesti. U modernoj medicini patologija se kao medicinska struka bavi laboratorijskom, pretežno mikroskopskom dijagnostikom bolesti (14). Može se reći da je jedna od ključnih uloga patologa rad sa slikama. Poznata izreka, koja govori kako je svaka slika kao tisuću riječi, vrlo dobro opisuje zašto je analiza patoloških slika vrlo pogodna za strojno učenje. U računalnom smislu, slike su zapravo veliki niz

brojeva koji je moguće vizualizirati na čovjeku razumljiv način. Slika može sadržavati puno podataka.

Postavljanje patohistološke dijagnoze iz mikroskopske slike tkiva puno je prikladniji problem za algoritme strojnog učenja. Ulaz je vrlo jasno definiran kao jedna slika, tj. kao tri matrice realnih brojeva. Svaki broj predstavlja intenzitet crvene, zelene ili plave boje određenog piksela slike. Iako je u teoriji moguće naučiti računalo da na temelju takve slike može postaviti bilo koju poznatu dijagnozu, to u praksi zasada nije izvedivo. U praksi se najčešće razvija algoritam strojnog učenja koji će za zadanu sliku predvidjeti vjerojatnost jedne, unaprijed određene, patohistološke dijagnoze. Primjer za prepoznavanje objekata sa slika, spomenut na početku ovog poglavlja, jasno prikazuje kako točnost algoritma strojnog učenja jako pada s povećanjem opsega zadatka.

Prednost računala je mogućnost učenja nad milijun slika s istom dijagnozom. Upravo zato računalo može naučiti obavljati tako usko definirane zadatke s visokom točnošću, koja može biti bolja čak i od ljudskog eksperta. No, i ovakav pristup ima svoje mane. Predoči li se upravo ovakvom algoritmu slika koja uopće nije tkivo, nego npr. kuća, algoritam će se ponašati nepredvidljivo. Čovjeku će odmah biti jasno da nešto nije u redu sa slikom. Umjetna inteligencija je trenutno samo alat koji je najkorisniji kada ga nadgleda i njime upravlja čovjek, tj. ekspert. Umjetna inteligencija ima veliki potencijal ubrzati rad liječnika, ali i podići njegovu kvalitetu. Razvoj algoritama strojnog učenja, koji rade sa slikama, je trenutno u vrlo intenzivnoj fazi, a upravo je zato patologija grana medicine koja ima veliki potencijal za primjenu strojnog učenja.

Ovaj rad se bavi problemom detekcije metastatskog tkiva na patohistološkim slikama presjeka limfnih čvorova nadziranim učenjem. Cilj rada je pokazati kako računalo može naučiti prepoznati sadrži li presjek limfnog čvora metastatsko tkivo. Ovakav pristup mogao bi pomoći u svakodnevnom radu patologa, ali ne i u potpunosti ga zamijeniti.

2. Definicija problema

Problem detekcije metastatskog tkiva raka dojke na patohistološkim slikama presjeka limfnih čvorova potrebno je svesti na oblik koji je razumljiv računalu. U obliku teksta, problem je moguće opisati na sljedeći način:

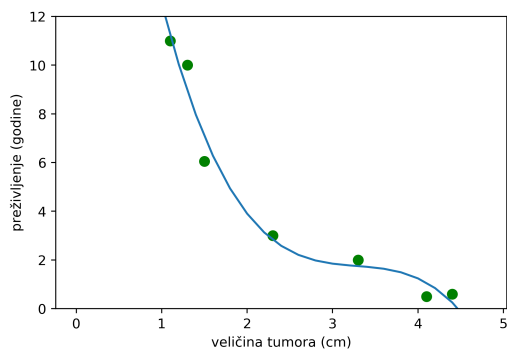
Za dani skup patohistoloških slika presjeka limfnih čvorova s pripadajućim oznakama postojanja metastatskog tkiva potrebno je pronaći i naučiti model strojnog učenja koji najbolje generalizira, tj. najbolje prepoznaje prisutnost metastatskog tkiva na neviđenim patohistološkim slikama presjeka limfnih čvorova.

2.1. Generalizacija algoritma strojnog učenja

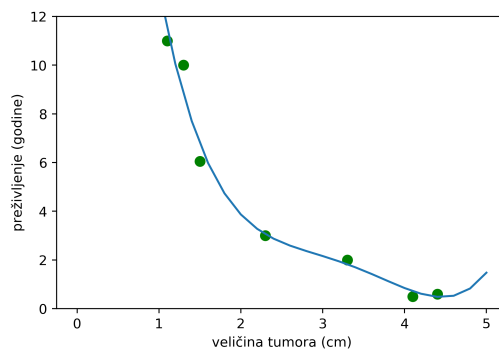
Smisao strojnog učenja nije prenaučiti poznate primjere, već naučiti generalizirati. U realnosti najčešće nije moguće egzaktno ispitati koliko dobro algoritam strojnog učenja generalizira. Za to bi bilo potrebno ispitati rad algoritma nad svim mogućim slikama limfnih čvorova, a teoretski postoji beskonačan broj takvih slika. Slika je zapravo samo niz realnih brojeva, a računala koriste fiksni broj bitova za prikaz realnih brojeva te je u praksi broj takvih slika konačan. No, on je još uvijek jako velik. Ispitivanje rada algoritma na svim presjecima limfnih čvorova koji su ikad snimljeni predstavlja neizvediv zadatak, a i isti preparat snimljen različitim skenerom ili osvjetljenjem također predstavlja novu sliku računalu.

Za pristupe strojnog učenja postoje dokazi kako mogu naučiti bilo koju funkciju f koja preslikava x u y (15). To znači kako je doslovno moguće naučiti skup podataka te ih bez pogreške preslikavati, a uz lošu generalizaciju. To se jasno vidi na Slici 2.1 gdje je za primjer objašnjen u uvodu korišten model s još većim kapacitetom. Kao model korišten je polinom četvrtog stupnja, koji puno bolje opisuje skup podataka, ali očito lošije generalizira. Nije za očekivati da će bolesnici s tumorom veličine 5 cm imati veće preživljenje od onih s 4 cm. No, potrebno je više podataka kako bi se to moglo sa sigurnošću utvrditi.

Regularizacija je metoda sprječavanja prenaučivosti. Korištenje jednostavnijeg modela s manjim kapacitetom i rano zaustavljanje učenja primjer su regularizacije (12). Najbolji način sprječavanja prenaučivosti modela s velikim kapacitetom je korištenje velikog skupa podataka za učenje.



(a) Polinom trećeg stupnja.



(b) Polinom četvrtkog stupnja.

Slika 2.1: Prenaučenost algoritma strojnog učenja. Napomena: podaci su izmišljeni, a primjer pojednostavljen.

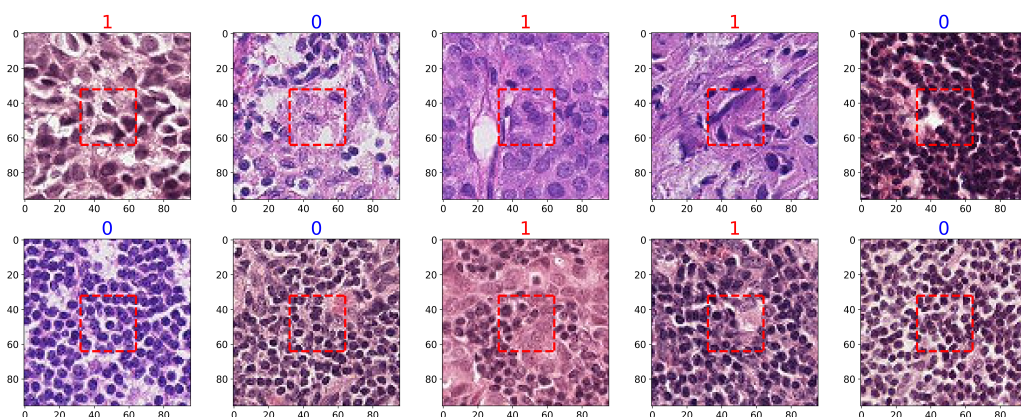
Mnogi modeli strojnog učenja imaju hiperparametre. Ime potječe od činjenice kako su to zapravo parametri već parametriziranog modela. Najčešće se parametrima smatraju vrijednosti modela koje se pronalaze procesom učenja, a hiperparametri se određuju prije početka učenja. No, to nije u potpunosti točno. Postoje metode kojima je moguće naučiti i hiperparametre (16). Bolji opis bio bi da su parametri vrijednosti koje su dio funkcije koja opisuje preslikavanje ulaza u izlaze, dok su hiperparametri vrijednosti koje nisu dio te iste funkcije, ali imaju utjecaj na sam proces učenja. Optimizacijski algoritam učenja, brzina učenja te snaga regularizacije samo su neki od primjera hiperparametara. Pristrani odabir hiperparametara nad skupom podataka za učenje također može uzrokovati prenaučnost.

Kako bi riješili problem generalizacije, u ovom radu korišten je velik skup podataka. Dodatno, podaci su podijeljeni u tri skupa podataka: skup za učenje, skup za validaciju te skup za ispitivanje. Računalo uvijek uči nad skupom za učenje. Kako bi izbjegli prenaučnost primjera, ali i hiperparametara te omogućili dobru generalizaciju, kvalitetu rješenja pratimo nad skupom za validaciju. Bitno je naglasiti kako pri učenju modela računalo ne vidi skup za validaciju, ali ga koristi za odabir ili čak učenje hiperparametara te na neki način model ima uvid u skup podataka za validaciju. Konačno rješenje, tj. konačni algoritam strojnog učenja ispitan je na neviđenim primjerima koji se nalaze u skupu za testiranje te se to koristi kao ocjena generalizacije.

2.2. Skup podataka

U ovom radu je korišten skup podataka PatchCamelyon (17). Skup podataka se sastoji od 327 680 slika veličine 96 x 96 piksela. Svaka slika predstavlja mali segment presjeka limfnog čvora čuvara obojenog hematoksilin-eozin bojanjem u jedne od bolesnica s dijagnozom karcinoma dojke. Za svaku sliku poznato je da li središnji dio slike veličine 32 x 32 piksela sadrži metastatsko tkivo, tj. da li barem jedan piksel predstavlja metastatsko tkivo. Slika 2.2

prikazuje primjer slika iz PatchCamelyon skupa podataka.



Slika 2.2: Primjer slika iz PatchCamelyon skupa podataka (17). Uz slike su prikazane i oznake postojanja metastatskog tkiva u središnjem, crveno označenom, dijelu.

Računalo bi moglo raditi i samo sa središnjim djelom slike, no na ovaj način dobiva širi kontekst, tj. može dobiti uvid u okolinu razmatranog dijela. Također, neki pristupi strojnog učenja koji rade sa slikama postupno smanjuju prostornu količinu informacija s rubova, tj. transformacijama ulazne slike postupno gube podatke na rubovima. PatchCamelyon je nastao nasumičnim izrezivanjem malih dijelova preparata iz Camelyon16 skupa podataka. Camelyon16 skup podataka nastao je suradnjom dva medicinska centra iz Nizozemske, a sadrži 400 cijelih preparata limfnih čvorova uz označeno metastatsko tkivo. (18).

PatchCamelyon predstavlja javno dostupan skup podataka koji se koristi za usporedbu algoritma strojnog učenja te je unaprijed podijeljen u skupove za učenje, validaciju i ispitivanje. Skup za učenje se sastoji od 262 144 slike, a skupovi za validaciju i testiranje od 32 768 slika. Bitno je naglasiti da se dijelovi istog preparata iz Camelyon16 skupa podataka mogu naći samo u jednom od prethodno navedenih skupova. Nema preklapanja između skupova, što je vrlo bitno za pravilnu procjenu generalizacije. Svaki od prethodno navedenih skupova sadrži 50% slika s metastatskim tkivom.

Skup podataka je dostupan u HDF5 formatu. To je format korišten za rad s velikim skupovima podataka, a prikladan je kada nije moguće učitati cijeli skup podataka u memoriju (19).

2.3. Ulaz, izlaz, funkcija gubitka i procjena generalizacije

Ulazni vektor je niz brojeva duljine 27 648, a kako je riječ o slikama, organiziran je u obliku tri matrice s dimenzijama 96 x 96. Tri matrice predstavljaju tri RGB kanala, odnosno intenzitet crvene, zelene i plave boje za svaki piksel ulazne slike. Svaka vrijednost je prikazana cijelim brojem iz intervala $[0,255]$, a svaka zauzima 1 bajt. Kao što je već prije rečeno, slika predstavlja dio presjeka limfnog čvora čuvara obojenog hematoksilin-eozin bojanjem.

Izlazni vektor zapravo je samo jedan broj koji opisuje da li barem jedan od piksela u središnjem (32 x 32) dijelu slike označava metastatsko tkivo. Svaka slika u skupu podataka označena je binarnim brojem. Jedan (1) označava prisutnost, a nula (0) odsutnost metastatskog tkiva. No, kako bi pri učenju računalo imalo bolji uvid u to koliko griješi te lakše učilo, kao izlaz bolje je postaviti jedan realan broj iz intervala [0,1]. Korištenjem sigmoidne funkcije na izlazu moguće je dobiti realan broj upravo iz prethodno navedenog intervala.

Funkcija gubitka označava koliko dobro algoritam strojnog učenja preslikava ulazni vektor (x) iz promatranog skupa podataka u očekivani izlazni vektor (y) (20). Negativna log-izglednost (NLL) korištena je kao funkcija gubitka te je prikazana u nastavku (21):

$$\text{NLL} = -\frac{1}{N} \sum_{i=0}^{N-1} (\hat{y}_i \cdot \ln(y_i) + (1 - \hat{y}_i) \cdot \ln(1 - y_i)),$$

pri čemu N označava broj slika iz promatranog skupa podataka, y_i označava izlaz algoritma strojnog učenja za i -tu sliku, a označava \hat{y}_i očekivani izlaz za i -tu sliku.

Tijekom procesa učenja računalo pokušava pronaći funkciju preslikavanja (f) tako da postupno smanjuje funkciju gubitka nad skupom za učenje. Kvaliteta odabira hiperparametara procjenjuje se usporedbom funkcija gubitaka nad skupom za validaciju. Konačan cilj strojnog učenja je naučiti dobro generalizirati, a to procjenjujemo izračunom funkcije gubitka nad skupom podataka za ispitivanje.

Dimenzija ulaza i izlaza te funkcija gubitka prikazani su na Tablici 2.1.

Tablica 2.1: Dimenzije ulaza, dimenzije izlaza i funkcija gubitka.

| ulaz | izlaz | funkcija gubitka |
|---------|-------|------------------|
| 96x96x3 | 1 | NLL |

3. Hardverska i programska potpora

U ovom radu korišteno je prijenosno računalo Lenovo Thinkpad T450s s Windows 10 operativnim sustavom. Računalo se sastoji od 2-jezgrenog procesora Intel Core i5-5200U, 8 GB RAM memorije te NVIDIA GeForce 940M grafičke kartice s 1 GB memorije. Ovo računalo korišteno je za analizu podataka te pokretanje jednostavnijih algoritama.

Većina eksperimenata pokrenuta je na računalu opremljenom za učenje dubokih neuronskih mreža. Računalo se sastoji od dva 6-jezgrema procesora Intel Xeon E5-2630, 128 GB RAM memorije te dvije NVIDIA TITAN Xp grafičke kartice s 12 GB memorije i 3840 jezgre, a koristi Ubuntu 18.04 operativni sustav.

3.1. Korišteni alati

Sav kod implementiran je koristeći Jupyter Notebook aplikaciju u obliku elektroničkih bilježnica. Ovo omogućuje brzo eksperimentiranje, ali i jednostavniju objavu, što omogućuje lakšu reprodukciju dobivenih rezultata (22).

Python (3.6.7) korišten je za implementaciju algoritama strojnog učenja i izradu svih vizualizacija prikazanih u ovom radu. Python je jezik visoke razine, a odlika mu je jednostavnost i sažetost koda, što omogućuje lako i brzo eksperimentiranje. No, posljedično tome, izvođenje Python koda često je značajno sporije od jezika niže razine. Python je postao standard za znanstveno programiranje zbog jako razvijenih znanstvenih *open-source* biblioteka. Većina tih biblioteka implementirana je u vrlo brzim jezicima niže razine poput jezika C i C++, što omogućuje puno brže izvođenje nego kada bi se izvodio isključivo Python kod. Biblioteke omogućuju efikasan i brz rad s algoritmima strojnog učenja te značajno skraćuju razvoj (23). U sklopu ovog rada korištene su biblioteke: Tensorflow, Keras i Matplotlib. U trenutku pisanja ovog rada Tensorflow nije podržavao rad s najnovijim Python (3.7.x) verzijama na Windows operativnim sustavima.

Tensorflow je biblioteka za strojno učenje, ali s fokusom na duboko učenje. Google Brain je tim unutar Google-a koji je zadužen za razvoj i održavanje. Biblioteka sadrži implementacije svih značajnih sastavnica algoritama dubokog učenja te značajno skraćuje vrijeme istraživanja i razvoja algoritama dubokog učenja. Duboko učenje često zahtjeva puno računalne snage u fazi učenja. No, dosta dijelova samog procesa učenja moguće je paralelizirati

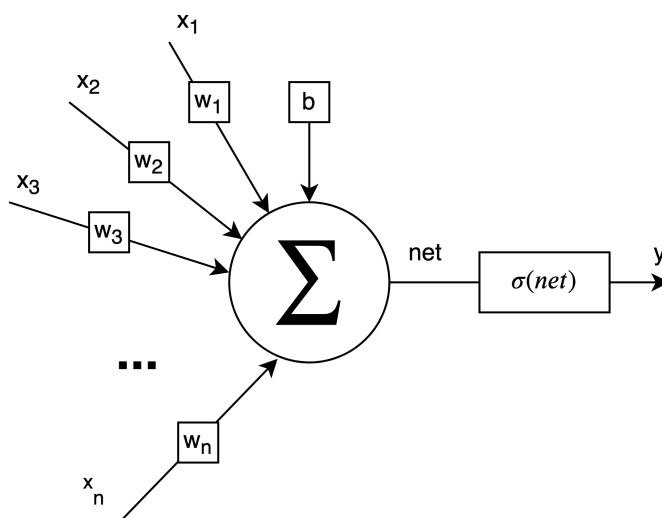
te izvoditi na grafičkim karticama. Ovo značajno ubrzava proces učenja. Biblioteka Tensorflow podržava rad s grafičkim karticama (24). Korištena je najnovija stabilna verzija (1.13).

Keras je biblioteka za duboko učenje. Predstavlja biblioteku koja je razina iznad Tensorflow-a, a isti koristi za pokretanje algoritama dubokog učenja. Kod napisan u Keras-u znatno je kraći u odnosu na ekvivalentan kod napisan u Tensorflow-u. Ovo predstavlja još dodatno pojednostavljenje rada s algoritmima dubokog učenja (25). Korištena je najnovija stabilna verzija (2.2.4).

Matplotlib je biblioteka za izradu različitih vizualizacija. Vrlo je jednostavna za korištenje te je često korištena uz rad prethodno navedenih biblioteka (26). Korištena je najnovija stabilna verzija (3.1.0).

4. Duboko učenje

Umjetne neuronske mreže predstavljaju vrlo popularnu skupinu modela strojnog učenja, a inspirirane su načinom kako mozak obrađuje informacije. Osamdeset i šest milijardi međusobno povezanih neurona čini ljudski mozak (27). Svaki neuron predstavlja vrlo jednostavnu jedinicu koja prima podražaje od drugih neurona, sumira ih, a ako je ukupan podražaj dovoljno jak, podražaj šalje dalje drugim neuronima. Umjetni neuron radi na istom principu, a prikazan je na Slici 4.1.



Slika 4.1: Generalni model umjetnog neurona.

Ulazni vektor (\mathbf{x}) najčešće je predstavljen kao niz realnih brojeva duljine n . Vektor težina (\mathbf{w}) predstavljen je kao niz realnih brojeva duljine n . Svaki ulaz (x_i) ima pripadajuću težinu (w_i) koja određuje koliki je utjecaj pojedinog ulaza u aktivaciji neurona. Parametar b se naziva pomak. Izlaz neurona je realan broj koji označava koliko se signala prenosi dalje drugim neuronima, a dobiva se koristeći aktivacijsku funkciju f . Izraz u nastavku opisuje matematički zapis izračuna izlazne vrijednosti iz ulaznih vrijednosti (21):

$$net = \left(\sum_{i=1}^n x_i \cdot w_i \right) + b = \mathbf{w}^T \mathbf{x} + b.$$

Najčešće se koristi sigmoidna funkcija (σ) te je upravo ona prikazana u nastavku (12):

$$\sigma(net) = \frac{1}{1 + e^{-net}}.$$

Sigmoidna funkcija nelinearno preslikava ukupni podražaj u realan broj iz intervala $(0,1)$. Aktivacijska funkcija nije povezana s funkcijom preslikavanja opisanom u Poglavlju 1, no obje su označene s oznakom f jer je to uobičajena matematička notacija.

Jedan neuron predstavlja vrlo jednostavnu jedinicu, a značajnu korist postiže tek povezivanjem u velike, paralelno povezane, neuronske mreže koje imaju veliku moć u obradi informacija. Upravo tako funkcionira i ljudski mozak. Umjetne neuronske mreže predstavljaju moćne modele strojnog učenja, a osnovni oblik nastao je povezivanjem prethodno opisanih umjetnih neurona u smislene cjeline. U ovom radu korištene su unaprijedno povezane neuronske mreže gdje su umjetni neuroni organizirani u slojeve, a svaki sloj povezan je samo sa slojem prije i sa slojem poslije. Prvi sloj naziva se ulazni sloj, a posljednji izlazni sloj. Svi ostali slojevi čine skrivene slojeve (21).

Duboko učenje, tj. duboke neuronske mreže zapravo su umjetne neuronske mreže s puno slojeva. Umjetna neuronska mreža sa samo jednim skrivenim slojem može aproksimirati bilo koju funkciju preslikavanja (15). Dokazano je kako je potrebno eksponencijalno manje neurona za aproksimaciju iste funkcije dubokom neuronskom mrežom u odnosu na neuronsku mrežu s jednim skrivenim slojem (28). No, duboke neuronske mreže još uvijek zahtijevaju značajnu računalnu moć i vrijeme za proces učenja, a to se posebno odnosi na rad s velikim skupom podataka u obliku slika. Proces izvođenja najčešće nije toliko zahtjevan. U ovom radu korišten je skup podataka koji se sastoji od preko 300 000 slika. Dovoljno je objasniti kako računalo uči nad cijelim skupom podataka, a pri konačnoj uporabi najčešće se izvodi nad pojedinačnim slikama.

Duboke unaprijedne neuronske mreže grade se nizanjem slojeva. Većina najznačajnijih vrsta slojeva implementirana je u bibliotekama Keras i Tensorflow. Implementacija sadrži dio za izvođenje te dio za učenje. Učenje se izvodi propagacijom unatrag, a izvođenje propagacijom unaprijed. Propagacija unatrag označava ažuriranje parametra modela, tj. duboke neuronske mreže. Za svaki parametar određuje se gradijent u odnosu na funkciju gubitka i promatrani skup podataka te se ažurira malim pomakom u smjeru negativnog gradijenta.

Svaki sloj sastoji se od neurona. Izlazi jednog sloja služe kao ulazi u sljedeći sloj. Ono što slojeve razlikuje je raspored i broj neurona, perceptivno polje pojedinog neurona te operacija koju neuroni izvode. Raspored neurona predstavlja oblik tenzora koji se dobiva prolaskom kroz taj sloj, a služi kao ulaz sljedećem sloju. Perceptivno polje označava koji dio izlaza prethodnog sloja utječe na rad pojedinog neurona promatranog sloja. Moguće je da neuron trenutnog sloja bude pod utjecajem cijelog ili samo dijela izlaza prethodnog sloja. Operacija koju neuroni izvode označava matematičku operaciju koja se izvodi nad izlaznim vrijednostima prethodnog sloja. Svaki sloj zapravo je jedno preslikavanje ulaza u izlaze, a nizanjem puno takvih slojeva dobiva se veća mogućnost ekspresije uz manji broj parametara. Bitno je da slojevi vrše nelinearno preslikavanje.

U nastavku su opisani korišteni slojevi. Uz svaki sloj opisan je postupak propagacije unaprijed. Njega je bitno poznavati radi razumijevanja rada korištene duboke neuronske mreže.

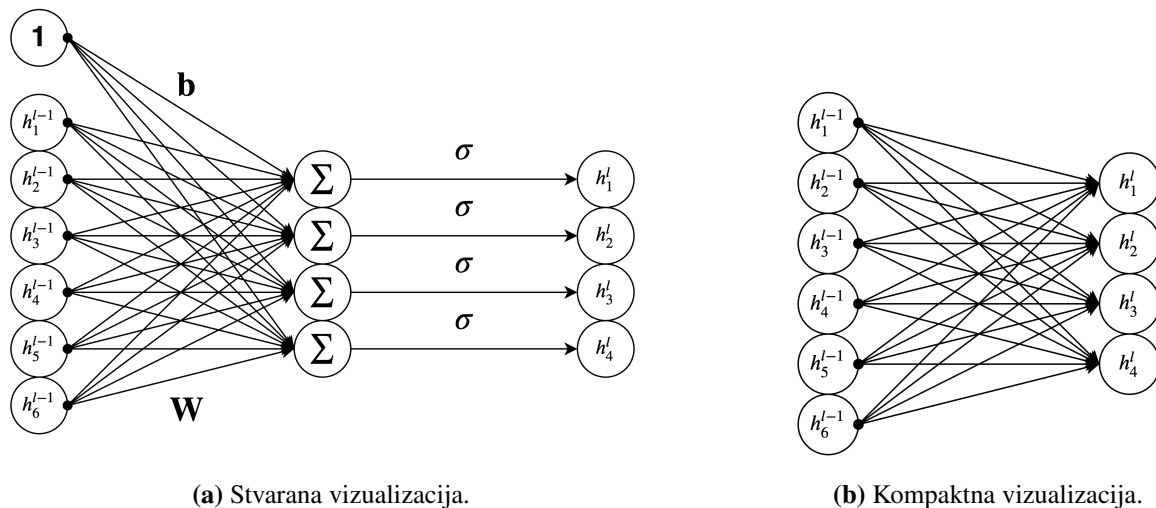
Za postupak propagacije unatrag najčešće je dovoljno znati princip na kojem radi, a biblioteke Keras i Tensorflow omogućuju proces učenja bez dubokog poznavanje matematičke podloge.

4.1. Potpuno povezani sloj

Potpuno povezani sloj zahtijeva da mu ulaz bude u obliku tenzora reda jedan, tj. vektora realnih brojeva, a sam sloj predstavljen je kao niz neurona prikazanih na Slici 4.1. Svaki neuron ovog sloja povezan je sa svim neuronima prethodnog sloja. Ulazi i izlazi skrivenih slojeva označavaju se sa simbolom \mathbf{h} kako bi se razlikovali od ulaza (\mathbf{x}) i izlaza (\mathbf{y}) cijele mreže. \mathbf{h}^l označava izlaz l -tog sloja, a predstavlja ulaz u sloj $l+1$. Matematički zapis potpuno povezanog sloja dan je u nastavku (20):

$$\mathbf{h}^l = \sigma(\mathbf{W}^l \cdot \mathbf{h}^{l-1} + \mathbf{b}^l),$$

pri čemu je \mathbf{h}^l izlaz sloja i , \mathbf{h}^{l-1} izlaz sloja $l-1$, tj. ulaz sloja l , \mathbf{W}^l matrica težina sloja l , \mathbf{b}^l vektor pomaka sloja l , a σ sigmoidalna funkcija. Moguće je koristiti i druge aktivacijske funkcije umjesto sigmoidalne, a sve se primjenjuju zasebno na svaki element tako da je izlaz iz ovog sloja vektor. Slika 4.2 prikazuje jedan potpuno povezani sloj sa šest ulaza i četiri izlaza.

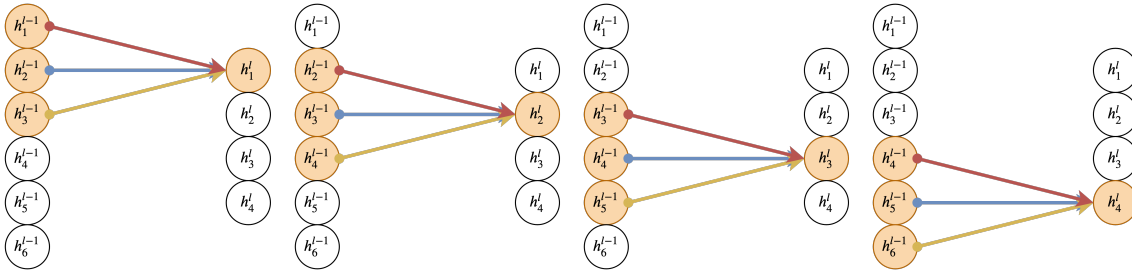


Slika 4.2: Potpuno povezani sloj.

Slika 4.2a prikazuje vizualizaciju prethodno navedenog matematičkog izraza koji opisuje potpuno povezani sloj. Slika 4.2b prikazuje sažet oblik koji je pregledniji ako mreža ima puno slojeva. Podrazumijeva se postojanje vektora pomaka (\mathbf{b}) i aktivacijske funkcije (σ). Bez aktivacijske funkcije, neuronske mreže bi mogle aproksimirati samo linearne funkcije.

4.2. Konvolucijski sloj

Konvolucijski sloj radi s podacima rešetkaste strukture te je ulaz u konvolucijski sloj tenzor. Keras podržava rad s 1D, 2D i 3D tenzorima. U odnosu na potpuno povezani sloj, konvolucijski sloj ima lokalno receptivno polje. Neuron konvolucijskog sloja vidi samo manji dio prethodnog sloja. 1D konvolucijski sloj sa 6 ulaznih neurona te filtrom veličine 3 prikazan je na Slici 4.3. Podrazumijeva se korištenje vrijednosti pomaka b^l . Veličina filtra označava broj neurona koji čine lokalno perceptivno polje svakog izlaznog neurona. U ovom primjeru nije korištena nadopuna rubova te se zbog toga broj neurona izlaznog sloja smanjio sa 6 na 4 neurona.



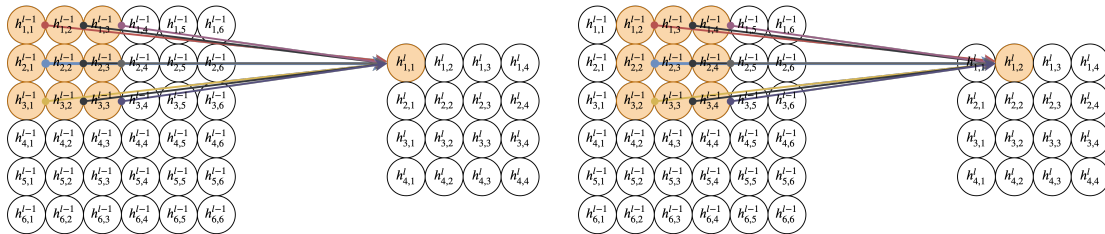
Slika 4.3: 1D operator konvolucije.

Generalni matematički zapis 1D konvolucijskog operatora prikazan je u nastavku (20):

$$h_i^l = \sum_{n=0}^{N-1} h_{i+n}^{l-1} \cdot K_n^l + b^l,$$

pri čemu h_i^l označava i -ti izlaz l -tog sloja, b^l vrijednost pomaka, a N dimenziju 1D filtra K^l .

Slike su u računalu reprezentirane kao matrice te se za rad sa slikama koriste 2D konvolucijski slojevi. Primjer jednog takvog sloja s ulaznom matricom dimenzije 6x6 i filtrom dimenzije 3x3 prikazan je na Slici 4.4. Bez korištenja nadopune izlaz ovakvog sloja ima dimenzije 4x4. Podrazumijeva se korištenje vrijednosti pomaka b^l .



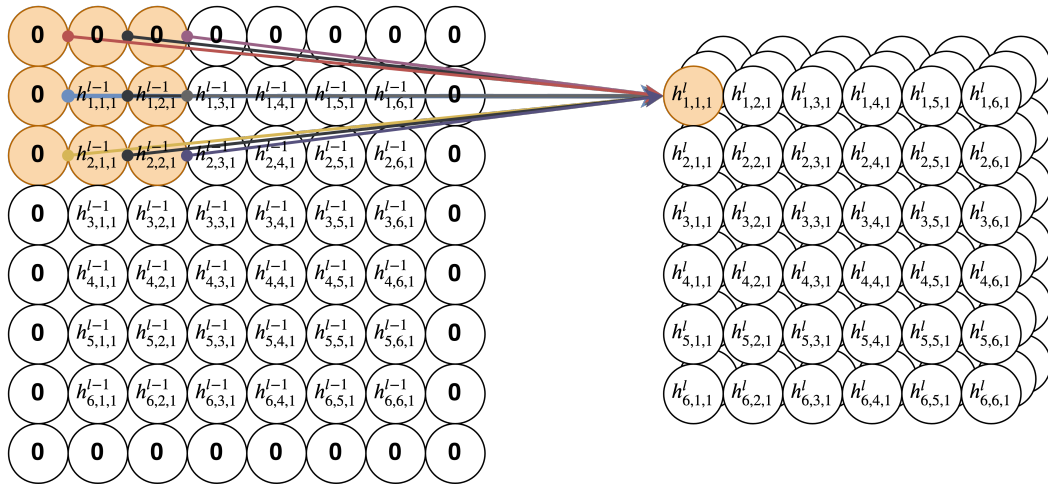
Slika 4.4: 2D operator konvolucije.

Generalni matematički zapis 2D konvolucijskih operatora prikazan je u nastavku (20):

$$h_{i,j}^l = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} h_{i+n,j+m}^{l-1} \cdot K_{n,m}^l + b^l,$$

pri čemu $h_{i,j}^l$ označava izlaz na poziciji i,j sloja l , b^l vrijednost pomaka, a N i M dimenzije 2D filtra K^l .

Prethodni primjeri koriste konvolucijski sloj bez nadopune te samo jedan filtar. Kako bi se očuvala ista dimenzija ulaza i izlaza, često se koristi nadopuna nulama na rubnim dijelovima. Korištenje više filtara u istom sloju povećava kapacitet mreže te je ona sposobnija naučiti kompleksnije preslikavanje. Slika 4.5 prikazuje konvolucijski sloj s ulaznim dimenzijama 6×6 i 3 filtra dimenzije 3×3 . Podrazumijeva se korištenje vrijednosti pomaka $b^{l,k}$ uz svaki filtar. Sloj koristi nadopunu nulama te izlaz zadržava dimenzije 6×6 , ali uz 3 kanala te se izlaz zapravo sastoji od 3 matrice. Ulaz također može imati više kanala. Ulazne slike korištene u ovom radu imaju dimenzije 96×96 i 3 kanala za boje.



Slika 4.5: 2D konvolucijski sloj koji koristi nadopunu nulama te 3 filtra.

Matricu, tj. tenzor reda 2 s više kanala moguće je promatrati kao tenzor reda 3. No, smisao 2D filtra konvolucijskog sloja je da svaki neuron toga sloja prima informacije samo od lokalnog dijela ulaza, ali da prima informacije od svih kanala tog istog dijela ulaza. Kanali predstavljaju različite značajke iste prostorne lokacije, a lokacija je opisana tenzorom reda 2. Ako se konvolucijski sloj postavi kao prvi sloj duboke neuronske mreže, logično je da svaki neuron vidi samo lokalni djelić ulazne slike, ali sva 3 kanala koji predstavljaju intenzitete boje pojedinog piksela.

Matematički zapis 2D konvolucijskog sloja prikazan je u nastavku (21):

$$h_{i,j,k}^l = \text{ReLU}\left(\sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \sum_{o=0}^{O-1} h_{i+n,j+m,o}^{l-1} \cdot K_{n,m,o}^{l,k} + b^{l,k}\right),$$

pri čemu N i M predstavljaju dimenzije filtra, O broj kanala ulaza, $h_{i,j,k}^l$ vrijednost izlaza na lokaciji i,j i kanalu k , $h_{i+n,j+m,o}^{l-1}$ vrijednost ulaza na lokaciji $i+n,j+m$ i kanalu o , $K_{n,m,o}^{l,k}$ vrijednost k -tog filtra sloja l na poziciji n,m,o , a $b^{l,k}$ vrijednost pomaka k -tog filtra sloja l . ReLU označava zglobnicu, a ona je najčešće korištena aktivacijska funkcija. Zglobnica je

definirana izrazom u nastavku:

$$ReLU(net) = \max(0, net).$$

Konvolucijski sloj koristi niz filtara koji zajedno predstavljaju jedan tenzor reda 3. Računalo ne računa svaku vrijednost iz prethodne formule pojedinačno, već koristi operacije nad tenzorima. Takve operacije moguće je paralelizirati te znatno brže izvesti na grafičkim karticama.

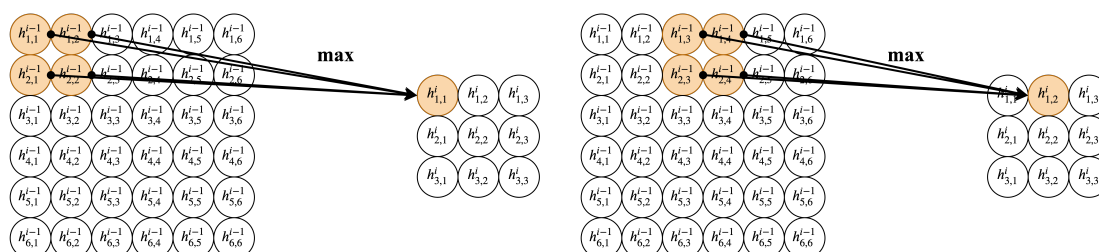
Kod potpuno povezanog sloja, svi izlazni neuroni povezani su sa svim ulaznim neuronima, a svaka ta veza predstavlja jedan parametar. Kod konvolucijskog sloja, jedan kanal predstavlja jednu skupinu parametara koji se dijele među neuronima. Upravo to dijeljenje parametara te korištenje filtara manjih dimenzija omogućuje konvolucijskom sloju da može imati znatno manje parametara od potpuno povezanog sloja.

Konvolucijski slojevi često se kombiniraju sa slojevima sažimanja koji su opisani u odjeljku koji slijedi. Upravo ta kombinacija omogućuje dubokim neuronskim mrežama invarijantnost na pomake, skaliranje i rotacije.

4.3. Sloj sažimanja

Sloj sažimanja predstavlja sloj koji također radi s lokalnim perceptivnim područjem. Ovaj sloj nema parametara koje je potrebno naučiti. Svi parametri ovog sloja definiraju se pri odabiru modela duboke neuronske mreže.

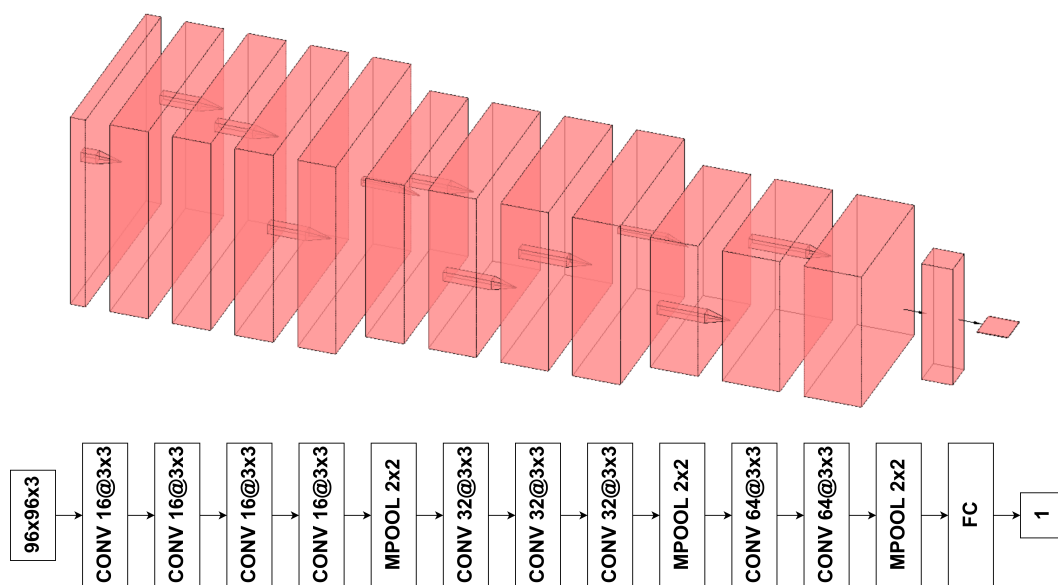
Izlazni neuron sloja sažimanja opisuje neku statističku značajku lokalnog receptivnog polja ulaznih neurona. Najčešće se ovaj sloj koristi tako da nema preklapanja lokalnih receptivnih polja, a to za posljedicu ima da su dimenzije izlaza manje od dimenzije ulaza. U ovom radu korišten je sloj sažimanja maksimalnom vrijednošću dimenzija 2x2, a time dimenzije izlaza postaju duplo manje od ulaznih. Filtar sažimanja primjenjuje se zasebno na svaki kanal te tako broj kanala ostaje nepromjenjiv. (21) Slika 4.6 prikazuje princip rada sloja sažimanja maksimalnom vrijednošću s dimenzijama 2x2.



Slika 4.6: 2D sloj sažimanja maksimalnom vrijednošću.

5. Predloženi model i rezultati

Slika 5.1 prikazuje korišteni model duboke neuronske mreže. Mreža se sastoji se od 9 konvolucijskih slojeva isprepletenih s 4 sloja sažimanja maksimalnom vrijednošću. Konvolucijski slojevi koriste zglobnicu kao aktivacijsku funkciju. Na kraju mreže je 1 potpuno povezani sloj sa sigmoidnom aktivacijskom funkcijom. Kako bi duboka neuronska mreža brže učila, nakon ulaznog sloja te prije svih aktivacijskih funkcija konvolucijskih slojeva korištena je normalizacija po grupi (29). Prikazani model sastoji se od 96 349 parametara, a od kojih je 95 767 moguće mijenjati pri procesu učenja.



Slika 5.1: Korištena duboka neuronska mreža. CONV označava konvolucijski sloj, MPOOL sloj sažimanja maksimalnom vrijednošću, a FC potpuno povezani sloj.

Korišteni skupovi podataka sadrže podjednaku raspodjelu slika s i bez metastaza te je točnost prikladna mjera za usporedbu različitih metoda. Tablica 5.1 prikazuje točnost predloženog modela na sva tri skupa podataka te usporedbu točnosti pri korištenju različitih funkcija gubitka te augmentacije skupa podataka za učenje. Izlaz (y) iz predložene duboke neuronske mreže realan je broj iz intervala $(0,1)$, a za izračun točnosti potrebna je binarna klasifikacija. Odabirom praga klasifikacije (θ) moguće je realan broj pretvoriti u binaran, a funkcija pretvorbe prikazana je izrazom u nastavku:

$$g(y) = \begin{cases} 1 & y > \theta \\ 0 & \text{inače} \end{cases}$$

Za klasifikaciju korištena je vrijednost praga 0,5.

Tablica 5.1: Rezultati predloženog modela.

| funkcija gubitka | augmentacija | UČENJE (%) | VALIDACIJA (%) | ISPITIVANJE (%) |
|------------------|--------------|------------|----------------|-----------------|
| MSE | ne | 94,41 | 86,02 | 79,97 |
| NLL | ne | 95,91 | 86,64 | 81,22 |
| MSE | da | 94,51 | 89,14 | 85,76 |
| NLL | da | 92,67 | 88,18 | 86,57 |

Postupak ispitivanja točnosti predloženih pristupa sastojao se od učenja nad augmentiranim skupom za učenje kroz 21 epohu. Kao optimizator korišten je RMSprop (21) uz početnu stopu učenja 10^{-4} . Svakih 7 epoha, stopa učenja je smanjena 10 puta. Nakon svake epohe provedena je evaluacija nad skupom za validaciju, a najbolji parametri modela nad skupom za validaciju korišteni su kao konačni parametri predloženog pristupa.

Srednja kvadratna pogreška (*MSE*) najčešće se koristi kao regresijska, a ne klasifikacijska funkcija gubitka. U ovom radu pokazalo se kako korištenje srednje kvadratne pogreške kao funkcije gubitka rezultira sličnom točnošću naučenog modela kao i korištenje negativne log-izglednosti. Srednja kvadratna pogreška (*MSE*) opisana je izrazom u nastavku (30):

$$MSE = \frac{1}{N} \sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2,$$

pri čemu N označava broj slika iz promatranog skupa podataka, y_i označava izlaz algoritma strojnog učenja za i -tu sliku, a označava \hat{y}_i očekivani izlaz za i -tu sliku.

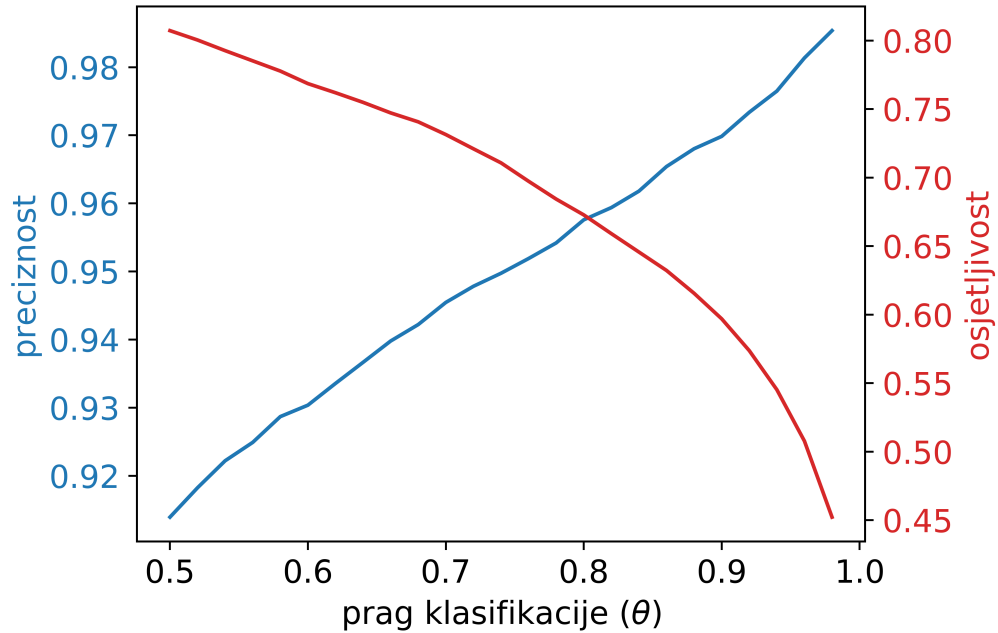
Augmentacija skupa podataka za učenje rezultira boljom točnošću od preko 5% za obje funkcije gubitka. Proces augmentacije sastojao se od horizontalnog i vertikalnog preokretanja slika te izmjene svjetline do 30%.

Metoda predložena u ovom radu sastoji se od modela prikazanog na Slici 5.1, a koristi negativnu log-izglednost kao funkciju gubitka te prethodno opisanu augmentaciju skupa podataka za učenje. Tablica 5.2 prikazuje usporedbu predložene metode s najboljim pronađenim objavljenim metodama.

Povećanjem praga klasifikacije (θ) moguće je povećati preciznost predložene metode, ali na račun osjetljivosti. Slika 5.2 prikazuje ovisnost preciznosti i osjetljivosti predložene metode o odabiru praga klasifikacije. Predložena metoda učena je na malim regijama patohistološkog preparata. Korištenje predložene metode nad cijelim preparatom uz viši prag klasifikacije omogućio bi veliku preciznost detekcije metastatskog tkiva te potencijalno uštedio vrijeme potrebno za manualnu klasifikaciju.

Tablica 5.2: Usporedba točnosti s drugim modelima.

| Metoda | NLL | Točnost (%) |
|-------------------|-------|--------------|
| predložena metoda | 0,336 | 86,57 |
| (31) | - | 88,10 |
| (17) | 0,260 | 89,80 |
| (32) | - | 90,36 |



Slika 5.2: Ovisnost preciznosti i osjetljivosti o pragu klasifikacije.

6. Zaključak

U ovom radu predložena je metoda za detekciju metastatskog tkiva na patohistološkim slikama presjeka limfnih čvorova, a temeljena je na dubokim neuronskim mrežama. Duboka neuronska mreža učena je na javno dostupnom skupu podataka te je stoga usporediva s drugim metodama.

Proces učenja predložene duboke neuronske mreže računalno je vrlo zahtjevan, no izvođenje je vrlo brzo, čak i na prosječnom računalu. Prijenosno računalo, opisano u poglavlju 3, može obraditi oko 250 ulaznih slika u sekundi.

Predložena metoda ne može zamijeniti rad patologa, niti je tome namijenjena. Ova metoda mogla bi ubrzati rad patologa označavanjem regija preparata za koje je vrlo sigurna da su metastatskog podrijetla, odnosno građeni od stanica koje nisu inače dio normalnog tkiva limfnog čvora.

ZAHVALE

Zahvaljujem svim djelatnicima Medicinskog fakulteta Sveučilišta u Zagrebu na pruženoj edukaciji, a posebno svojoj mentorici doc. dr. sc. Gorani Aralici. Posebno hvala mojoj obitelji i prijateljima koji su mi bili velika podrška tijekom studija.

LITERATURA

- [1] Kurzweil R. The age of intelligent machines. MIT press Cambridge, MA; 1990.
- [2] Searle JR. Minds, brains, and programs. *Behav Brain Sci.* 1980 Nov;3(3):417–424. doi: 10.1017/S0140525X00005756.
- [3] Sasaki M, Shinnou H. Spam detection using text clustering. U: 2005 International Conference on Cyberworlds (CW'05). IEEE; 2005. str. 316–319. doi: 10.1109/CW.2005.83.
- [4] Ghosh S, Reilly DL. Credit card fraud detection with a neural-network. U: System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on. sv. 3. IEEE; 1994. str. 621–630. doi: 10.1109/HICSS.1994.323314.
- [5] Chang PC, Liu CH. A TSK type fuzzy rule based system for stock price prediction. *Expert Syst Appl.* 2008 Jan;34(1):135–144. doi: 10.1016/j.eswa.2006.08.020.
- [6] Viola P, Jones MJ. Robust real-time face detection. *Int J Comput Vis.* 2004 May;57(2):137–154. doi: 10.1023/B:VISI.0000013087.49260.fb.
- [7] McCarthy J, Minsky ML, Rochester N, Shannon CE. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI Mag.* 2006 Dec;27(4):12–12. doi: 10.1609/aimag.v27i4.1904.
- [8] Alpaydin E. Introduction to machine learning. MIT press; 2009.
- [9] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. Cite-seer; 2009.
- [10] Huang Y, Cheng Y, Chen D, Lee H, Ngiam J, Le QV, i sur. GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism. *CoRR.* 2018;abs/1811.06965. Dostupno na: <http://arxiv.org/abs/1811.06965>.
- [11] Obermeyer Z, Emanuel EJ. Predicting the Future - Big Data, Machine Learning, and Clinical Medicine. *N Engl J Med.* 2016 Sep;375(13):1216–1219. doi: 10.1056/NEJMp1606181.

- [12] Bishop CM. Pattern recognition and machine learning. springer; 2006.
- [13] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, i sur. Mastering the game of Go with deep neural networks and tree search. *Nature*. 2016 Jan;529(7587):484–489. doi: 10.1038/nature16961.
- [14] Damjanov I, Jukić S, Nola M. Patologija. Medicinska naklada; 2011.
- [15] Cybenko G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*. 1989 Dec;2(4):303–314. doi: 10.1007/BF02551274.
- [16] Jaderberg M, Dalibard V, Osindero S, Czarnecki WM, Donahue J, Razavi A, i sur. Population Based Training of Neural Networks. *CoRR*. 2017;abs/1711.09846. Dostupno na: <http://arxiv.org/abs/1711.09846>.
- [17] Veeling BS, Linmans J, Winkens J, Cohen T, Welling M. Rotation equivariant CNNs for digital pathology. U: Frangi AF, Schnabel JA, Davatzikos C, Alberola-López C, Fichtinger G, urednici. *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Springer International Publishing; 2018. str. 210–218. doi: 10.1007/978-3-030-00934-2_24.
- [18] Bejnordi BE, Veta M, Van Diest PJ, Van Ginneken B, Karssemeijer N, Litjens G, i sur. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*. 2017 Dec;318(22):2199–2210. doi: 10.1001/jama.2017.14585.
- [19] Koranne S. Hierarchical data format 5: HDF5. U: *Handbook of Open Source Tools*. Springer; 2011. str. 191–200. doi: 10.1007/978-1-4419-7719-9_10.
- [20] Russell S, Norvig P. *Artificial Intelligence: A Modern Approach*. 3rd izd. Prentice Hall; 2009.
- [21] Goodfellow I, Bengio Y, Courville A. *Deep learning*. MIT press; 2016. <http://www.deeplearningbook.org>.
- [22] Kluyver T, Ragan-Kelley B, Pérez F, Granger BE, Bussonnier M, Frederic J, i sur. Jupyter Notebooks - a publishing format for reproducible computational workflows. U: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*; 2016. str. 87–90. doi: 10.3233/978-1-61499-649-1-87.
- [23] Oliphant TE. Python for scientific computing. *Comput Sci Eng*. 2007;9(3):10–20. doi: 10.1109/MCSE.2007.58.

- [24] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. TensorFlow: A system for large-scale machine learning. U: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). {USENIX} Association; 2016. str. 265–283.
- [25] Ketkar N. Introduction to Keras. U: Deep Learning with Python. Springer; 2017. str. 97–111. doi: 10.1007/978-1-4842-2766-4_7.
- [26] Hunter JD. Matplotlib: A 2D graphics environment. Comput Sci Eng. 2007 Jun;9(3):90–95. doi: 10.1109/MCSE.2007.55.
- [27] Azevedo FA, Carvalho LR, Grinberg LT, Farfel JM, Ferretti RE, Leite RE, et al. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. J Comp Neurol. 2009 Apr;513(5):532–541. doi: 10.1002/cne.21974.
- [28] Delalleau O, Bengio Y. Shallow vs. deep sum-product networks. U: Advances in Neural Information Processing Systems 24; 2011. str. 666–674.
- [29] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. CoRR. 2015;abs/1502.03167. Dostupno na: <http://arxiv.org/abs/1502.03167>.
- [30] Shcherbakov MV, Brebels A, Shcherbakova NL, Tyukov AP, Janovsky TA, Kamaev VA. A survey of forecast error measures. World Appl Sci J. 2013 Sep;24(24):171–176.
- [31] Worrall DE, Welling M. Deep Scale-spaces: Equivariance Over Scale. CoRR. 2019;abs/1905.11697. Dostupno na: <http://arxiv.org/abs/1905.11697>.
- [32] Teh EW, Taylor GW. Metric Learning for Patch Classification in Digital Pathology. U: International Conference on Medical Imaging with Deep Learning – Extended Abstract Track. London, United Kingdom; 2019. Dostupno na: <https://openreview.net/forum?id=BJgtl1V6FN>.

ŽIVOTOPIS

Andrija Miličević rođen je 25. lipnja 1990. godine u Zagrebu. XV. gimnaziju (MIOC) završio je 2009. godine, a iste godine upisao je Medicinski fakultet Sveučilišta u Zagrebu. Fakultet elektrotehnike i računarstva Sveučilišta u Zagrebu upisao je 2011. godine. Preddiplomski studij završio je 2016. godine, a diplomski 2018. godine. Od veljače 2018. zaposlen je kao istraživač na Zavodu za osnovne elektrotehnike i električna mjerenja. Tijekom studija radio je 6 mjeseci u razvojnom centru tvrtke Microsoft u Oslu, Norveška.