

SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Luka Baković

**Homogenizacija elektromagnetskih
kompozitnih materijala metodama
dubokog učenja**

Zagreb, 2020.

Ovaj rad izrađen je na Zavodu za radiokomunikacije pod vodstvom doc. dr. sc. Daria Bojanjca i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2019./2020.

Sadržaj

Popis slika	iii
1 Motivacija	1
2 Postavljanje problema	5
2.1 Homogenizacija kompozitnih materijala	5
2.2 Duboko učenje	7
2.3 Svrha primjene dubokog učenja na homogenizaciju	8
3 Metode	10
3.1 Računalno okruženje	10
3.2 Priprema baze primjera	10
3.2.1 Gmsh	11
3.2.2 Effective2D	12
3.3 Duboki model	13
3.3.1 Konvolucijska neuronska mreža	14
3.3.2 PyTorch	15
3.3.3 Korišteni model	15
4 Rezultati	18
4.1 Pretraživanje parametara	18
4.2 Najbolji model	21
4.3 Test na <i>dva sa dva</i> skupu	21
5 Zaključak i daljnji rad	26
6 Literatura	30

Popis slika

1	Skica kompozita sastavljenog od tri materijala	5
2	Kvadrat unutar jedinične ćelije dobiven Gmsh programom	11
3	Kompliciranija geometrija jedinične ćelije dobiven Gmsh programom	12
4	Primjeri binarnih slika kakve su korištene kao ulazni podatci	16
5	Pogreška predikcije na minigrupi veličine 16 tokom faze testiranja	19
6	Pogreška predikcije na minigrupi veličine 32 tokom faze testiranja	19
7	Pogreška predikcije na minigrupi veličine 16 tokom faze validacije	20
8	Pogreška predikcije na minigrupi veličine 32 tokom faze validacije	20
9	3x3 geometrije s najboljim performansama	22
10	3x3 geometrije s najgorim performansama	23
11	2x2 geometrije s najboljim performansama	25
12	2x2 geometrije s najgorim performansama	26

1 Motivacija

Sadržaj ovog rada nalazi se unutar presjeka matematike, fizike i inženjerstva. Poanta je računarskim idejama nadopuniti matematičke metode koje pak koriste fizičarima i inženjerima u njihovim proučavanjima bilo u znanosti ili industriji. Mada zvuči apstraktno, ideja rada počinje s vrlo konkretnim, dapače, opipljivim problemom. Današnja industrija, pri čemu najviše ciljamo na proizvodnju, davno je nadišla staru kinesku ideju o pet elemenata koji tvore svijet. Kameni zidovi već dugo vremena ne sadrže samo kamen - danas od građevine zahtjevamo bolju izolaciju i poželjna mehanička svojstva. Čisti metali davno su zamijenjeni legurama u nadi da se spriječe razne neželjene reakcije od hrđanja nadalje. Također, postaje skoro nemoguće pronaći staklo koje nema u najmanju ruku premaz protiv grebanja. Očito, rodila se ideja izrade materijala koji svojim svojstvima nadilaze one prirodne. Tako nastaje pojam metamaterijala [1] koji označava materijal sa svojstima koja ne možemo naći u prirodi. Tako nastaju krutine iznimno niske gustoće, poput aerogela, ili pak materijali s negativnim indeksom loma [2] kakve proučava nelinearna optika. Osim izrade *neobičnih* materijala od nule, jako poželjna svojstva mogu se dobiti i stvaranjem kompozitnih materijala. Pojam kompozitnog materijala (ili kompozita) odnosi se na materijal sačinjen od dva ili više materijala s različitim fizikalnim svojstvima koji u cjelini daju potpuno novi materijal sa svojim skupom fizikalnih svojstava. Neki od poznatijih primjera uključuju armirani beton, stakloplastika (*fiberglas*) i karbonska vlakna. Takvi materijali čine temelj današnje proizvodnje, gdje zahtjevi na fizikalna svojstva postaju sve rigorozniji kako bi nam omogućili gradnju sve viših nebodera, proizvodnju što lakših aviona te operacije pod uvjetima u rasponu od najdubljeg oceana do svemirskog prostranstva.

Istraživanje metamaterijala i kompozita usko je povezano s matematičkim modeliranjem. Postavljaju se pitanja o svojstvima takvih materijala te o njihovom ponašanju u raznolikim okruženjima. Područja fizike i inženjerstva navikla su davati odgovore na takva pitanja, pritom koristeći matematičko modeliranje. Pojam matematičkog modeli-

ranja podrazumijeva prevođenje zbilje u jezik matematike - dakle opis stvarnih objekata pomoću matematičkih objekata. Takav prijevod često zahtjeva razna pojednostavljenja ili dodatne pretpostavke o fizikalnoj situaciji, pošto je priroda rijetko dovoljno pristojna da dozvoli direktan, *jedan na jedan*, zapis modela fizikalnog fenomena. Takva pojednostavljenja potrebna su i u onim sustavima koji su na prvi pogled najjednostavniji, recimo modeliranje obične nogometne lopte koja pada sa visine. Nekad ta potreba za pojednostavljenjem dolazi iz pedagoških razloga. Tako svaki učenik srednje škole, a i jedan dio studenata na fakultetu, u svojim udžbenicima iz fizike pronađe naputke "zakružite gravitacijsku konstantu na 10", "zanemarite otpor zraka" ili "smatrajte da je plin idealan". Sakupljanjem više znanja i savladavanjem težih metoda ova pojednostavljenja nestaju, pa gravitacijska konstanta počinje iznositi 9.81, otpor zraka biva modeliran članom proporcionalnim kubu brzine, a proučavanje idealnih plinova odvede studente na dugačak put kroz statističku i kvantnu fiziku. Ali nekad ta pojednostavljenja proizlaze iz potrebe da se dobije rješenje problema u situaciji gdje egzaktno modeliranje vodi u propast, što iz teoretskih (problem je nekorektno postavljen) a što iz praktičnih razloga (na primjer, prevelik broj jednadžbi).

Konkretan problem kojime se ovaj rad bavi jedan je mali podskup upravo opisanih problema matematičkog modeliranja kompozitnih materijala. Radi se o homogenizaciji elektromagnetskih kompozitnih materijala. Elektromagnetski kompozit u ovom je slučaju materijal sastavljen od dva različita materijala, i to spajanjem nekog broja jediničnih ćelija koje se periodično ponavljaju. Pitanje koje se postavlja odnosi se na ponašanje ovakvog kompozita u interakciji sa elektromagnetskim valovima. Odgovor uobičajeno dolazi iz grane numeričke matematike, to jest materijal se modelira uz pomoć metode konačnih ili rubnih elemenata. Takve metode uključuju dijeljenje jedinične ćelije na male poddomene, te rješavanje sustava na tim manjim komadićima kako bi se dobilo rješenje na cjelokupnom materijalu. No, zbog malih dimenzija komadića ovog metamaterijala u odnosu na valnu duljinu, zahtjev na točnost rješenja sa sobom povlači

jako gustu mrežu (eng. *mesh*) na kojoj se problem rješava. Jako gusta mreža za sobom pak povlači velik broj nepoznanica, pa i velik broj jednažbi koje treba riješiti kako bi se dobio konačan rezultat. To često predstavlja velik zahtjev na količinu računalnih resursa potrebnih za rad na takvom problemu. Jedan od načina za smanjenje broja jednažbi i pojednostavljenje modela zove se homogenizacija - proces zamjene parametara kompozitnog materijala s jednim, efektivnim parametrom koji vrijedi na jediničnoj ćeliji uz kontrolu promjene rješenja prijelaskom iz jednog modela u drugi. Zanimljiv problem koji tu nastaje je da sama homogenizacija opet uključuje rješavanje sustava diferencijalnih jednažbi te sa sobom opet nosi određene zahtjeve na računalne resurse potrebne za rješavanje problema. Cilj je ovog rada doskočiti tom problemu, te umjesto rješavanja sustava diferencijalnih jednažbi pokušati iskoristiti metode dubokog učenja za homogenizaciju elektromagnetskih kompozitnih materijala.

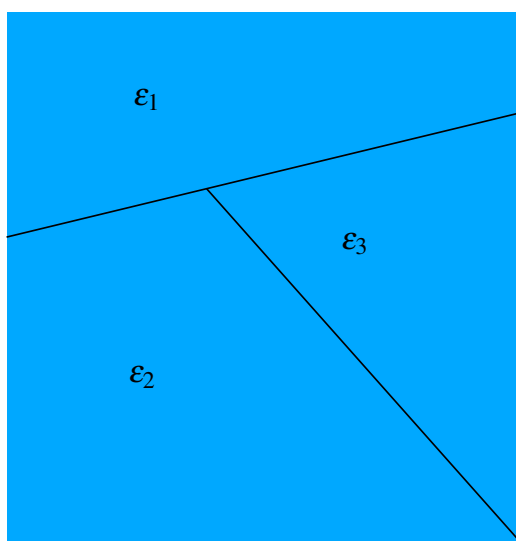
Zadnja točka motivacije problema očito se onda mora odnositi na opravdavanje korištenja baš modela dubokog učenja u ovu svrhu. Odgovor na to pitanje možemo pružiti na više načina. Pragmatično, modeli dubokog učenja rade malo drugačiji raspored resursa u odnosu na tipično rješavanje jednažbi. Postoji inicijalni utrošak na treniranje modela, koji je značajan, ali jednom kada se ta energija utroši - model je spreman davati predikcije uz minimalne resurse. Jednom kada je model gotov, dobivanje rješenja za problem svodi se na svega nekoliko množenja matrica, što je neznatno u odnosu na utrošak rješavanje sustava nekoliko tisuća jednažbi. Dakle, duboki model je svojevrсно ulaganje u budućnost - velik utrošak na početku za veliku uštedu u nastavku. Nepragmatično, korištenje modela dubokog učenja u ove svrhe interdisciplinarna je ideja, i kao takva potiče daljni razvoj novih područja znanosti te novih pristupa postojećim problemima, vidljivo iz [3]. Dugi niz godina, strojno učenje svodilo se na statističke postupke nalaženja parametara distribucije, ili računanja regresije nad podacima. Danas, strojno učenje evoluiralo je u granu dubokog učenja s modelima koji sami pronalaze optimalne značajke za opis velikih skupova podataka, nude univerzalne aproksimatore za funkcije

te u zadnje vrijeme nude i razne generativne funkcije. Ti modeli koriste se u većinom u svijetu *big data*, ali nude obećavajuće rezultate u primjenama na fizikalne probleme. Tako očekujemo da u budućnosti napredniji modeli dubokog učenja budu iskorišteni u svijetu matematičkog modeliranja. Kao jedan primjer, u elektromagnetizmu pojavljuju se inverzni problemi - ako znamo parametar, kako izgleda materijal? Ili pak, ako znamo kako izgleda odbijeni val, kako je izgledao ulazni val? Koji je optimalan oblik jedinične ćelije za zadane parametre? Duboko učenje danas odgovara na ta pitanja u malo drugačijim formulacijama, iskorištavanjem GAN-ova za generalizaciju objekata na slikama, ili pak korištenjem varijabilnih autoenkodera za proizvodnju novih primjera iz zadanih značajki. Ovaj rad čini prvi korak ka rješavanju dubljih problema, jer svaki ulazak u interdisciplinarnost počinje sa pronalaženjem zajedničkih problema te opisa zajedničkim jezikom. Rješavanje problema homogenizacije elektromagnetskih kompozitnih materijala dubokim učenjem djeluje kao motivacija za daljnji ulazak u presjek područja dubokog učenja i matematičkog modeliranja [4], te na kraju krajeva radi prvi korak prema industrijskoj primjeni takvih metoda te olakšavanju ili pak unaprijeđenju procesa izrade samih kompozita koji žive u većini današnjih proizvoda.

2 Postavljanje problema

2.1 Homogenzacija kompozitnih materijala

Proces matematičkog modeliranja kompozitnog materijala sa sobom nosi određene poteškoće. Većina fizikalnih procesa modelira se diferencijalnim jednačbama, koje zatim u sebi imaju funkcije parametara materijala - jer ako ponašanje stvarnog materijala ovisi o njegovim svojstvima, onda rješenje modela mora ovisiti o nekakvim parametrima koji modeliraju ta svojstva. Za ilustraciju, promatrajmo kompozit sačinjen od tri materijala. Pretpostavimo da je problem koji rješavamo elektromagnetske prirode, recimo da nas zanima električno polje u materijalu na slici 1. Upute za analitičko rješavanje ovog problema mogu se naći u [5].



Slika 1: Skica kompozita sastavljenog od tri materijala

Rješavamo jednačbu

$$\nabla \cdot [\boldsymbol{\varepsilon}(\mathbf{r}) \mathbf{E}(\mathbf{r})] = \rho(\mathbf{r}).$$

Rješenje sustava očito ovisi o funkciji $\boldsymbol{\varepsilon}(\mathbf{r})$. Intuitivno, što kompliciranija funkcija $\boldsymbol{\varepsilon}(\mathbf{r})$ - to više analize je potrebno za doći do rješenja. Dakle, jedan od problema proizlazi iz kompleksnosti funkcije $\boldsymbol{\varepsilon}(\mathbf{r})$, čija vrijednost ne ovisi samo o poziciji \mathbf{r} , već i o komadu materijala unutar kojeg se nalazimo. Funkcija $\boldsymbol{\varepsilon}(\mathbf{r})$ iznosi $\boldsymbol{\varepsilon}_1(\mathbf{r})$ unutar prvog materijala, $\boldsymbol{\varepsilon}_2(\mathbf{r})$ unutar drugog materijala, a $\boldsymbol{\varepsilon}_3(\mathbf{r})$ unutar trećeg materijala. Za rješavanje ovakvih sustava uobičajeno se koristi metoda konačnih elemenata, koja zahtjeva dijeljenje domene na manje komade (eng. *meshing*) kako bi se problem podijelio na lakše podprobleme. Kako bi problem na manjoj domeni stvarno bio lakši od onog na cijeloj domeni, moramo osigurati da parametri na poddomenama imaju manju varijancu od onih na cijeloj domeni. Dakle, veličina najvećeg pojedinačnog komada (poddomene) ograničen je promjenom parametara. S obzirom da imamo tri funkcije parametra $\boldsymbol{\varepsilon}$ omotane unutar jedne, intuicija nalaže da će ti komadići morati biti jako maleni kako bi izbjegli diskontinuitete pri prijelazu granice materijala unutar kompozita.

Drugi problem proizlazi iz činjenice da je veličina najvećeg komada ograničena i valnom duljinom koju istražujemo. Komad poddomene mora biti dovoljno malen kako bi na njemu bili vidljivi svi efekti. Pre velik komad, i rubni uvjeti na krajevima navodit će nas na pogrešna rješenja. Kao rezultat svega toga, veličina najmanjeg komada na koji smijemo razdijeliti domenu postaje jako malen. To stvara problem jer svaka poddomena zahtjeva svoje rješenje, dakle jedan matrični sustav oblika

$$\mathbf{Ax} = \mathbf{b}.$$

Jedno od rješenja problema ukazuje se u obliku homogenizacije. U tom postupku, kompozitni materijal biva zamijenjen homogenim materijalom uz uvjet da razlika rješenja problema u takvom materijalu ne odstupa previše od rješenja u kompozitu. Postavlja se uvjet

$$\|\mathbf{E}_{kom} - \mathbf{E}_{hom}\| \ll .$$

Proces homogenizacije na jediničnoj ćeliji kompozitnog materijala kao rezultat daje efektivni parametar koji vrijedi unutar cijele ćelije te čije korištenje daje rješenje problema koje je dovoljno točno (gdje je definicija dovoljne točnosti prepuštena rješavaču problema). Ovakvo rješavanje problema manje je računalno zahtjevno - materijal je homogen, parametar ε mijenja se manje nego u kompozitu jer sada ima jednu umjesto nekoliko domena, samim time najveći dopušteni komad poddomene veći je nego prije pa je broj matričnih sustava manji.

2.2 Duboko učenje

Pojam duboko učenje odnosi se na granu strojnog učenja koja se većinom temelji na korištenju neuronskih mreža za uočavanje značajki (eng. *representation learning*) nad velikim skupovima podataka. Neuronska je mreža struktura inspirirana izgledom neurona u ljudskom mozgu. Sastoji se od slojeva umjetnih neurona koji primaju ulazni signal (podatak), procesuiraju ga (primjenjuju funkciju na njega) te ga šalju dalje jednome ili više neurona u idućem sloju. Mogu se učiti procesima nadziranog ili nenadziranog učenja. U nadziranom učenju, postoji skup ulaznih podataka te skup željenih izlaza iz mreže koji odgovaraju ulaznim podacima. Proces se sastoji od optimizacijskog problema minimizacije razlike izlaza iz mreže za dani ulazni podatak i zadanog željenog izlaza. Druga mogućnost je nenadzirano učenje, gdje je dostupan jedino skup ulaznih podataka bez ikakvih zadanih oznaka. Uloga je modela u tom slučaju prepoznati uzorke u podacima i na neki način ih grupirati.

Ovaj rad koristiti će metodu nadziranog učenja. Rješavanje nekog problema dubokog učenja na taj način sastoji se od nekoliko tipičnih koraka. Prvi od njih je priprema ulaznih podataka i njihovih rješenja. Podatci moraju biti u pravom obliku kako bi mogli ući u duboki model, a rješenja (ili oznake) moraju biti takve da ih je moguće dobiti kao izlaz iz neuronske mreže. Ulazi i izlazi, dakle, moraju biti u matričnom obliku. Potrebno je paziti i na svojstva samih podataka, ne samo na oblik. Uobičajen je postupak norma-

lizacije podataka, tj. uprosječivanju i skaliranju podataka kako bi njihova distribucija imala nulto očekivanje i jediničnu varijancu. Zatim slijedi dizajniranje samog modela. Potrebno je izabrati dovoljno slojeva da bi se mogle prepoznati sve bitne značajke unutar podataka, ali ne previše slojeva kako bi se izbjeglo nepotrebno kompliciranje podela i prenaučnost (eng. *overfitting*). Zatim, potrebno je razmisliti o prikladnom optimizacijskom postupku te istrenirati model. Kao zadnji korak, potrebno je napraviti analizu pogreške.

2.3 Svrha primjene dubokog učenja na homogenizaciju

Iduće logično pitanje glasi: ako prijašnje opisana numerička metoda radi, i daje dovoljno točno rješenje, čemu duboko učenje? Odgovor je da je situacija puno zanimljivija od samog dobivanja efektivnog parametra. Dok numerička metoda daje *ispravne brojeve*, duboka metoda u ovom slučaju daje dodatan uvid u sam problem. Većina zanimljivih (iz današnje perspektive) fizikalnih problema zahtjevaju modeliranje parcijalnim diferencijalnim jednadžbama, koje su generalno nelinearne. Recimo da znamo rješenje problema za neku zadanu geometriju, i neku drugu zadanu geometriju. Zatim postavimo pitanje o rješenju na zbroju te dvije geometrije. Nažalost, konzultacijom sa sustavom parcijalnim diferencijalnih jednadžbi saznajemo da rješenje zbroja nije jednako zbroju rješenja i tu priča staje. Nadalje, možemo se zapitati je li rješenje problema na pravokutniku na neki način povezano s rješenjem na kvadratu. Opet, numerički postupak ne daje odgovor na to pitanje, jedino nam nudi krajnje rješenje. Ukratko, nedostaje nam generalizacija. Čak i nakon rješavanja stotinu problema, stotinu i prvi jednako je težak i njegovo rješenje jednako nam je nepoznato.

Duboki modeli trenirani nadziranim učenjem pokušavaju postići točno to - generalizaciju. Oni obrađuju podatke, u ovom slučaju velik broj zadataka i velik broj rješenja, te pokušavaju dati predikcije za konačno rješenje na nepoznatim problemima. Ukoliko se pokaže da je duboki model uspješan, postoji neka doza moguće generalizacije

u problemu. To je odgovor koji se ne može dobiti iz numeričkog postupka. Ukoliko on stvarno bude potvrđan, otvaraju se daljnje mogućnosti. Kao prvo, takav model moguće je koristiti kao grubu procjenu u procesu projektiranja. Ukoliko nam nije potrebno numeričko rješenje već samo estimacija, duboki model nam to može pružiti s manje utroška vremena nego numerički postupak (jer zahtjeva samo nekoliko množenja matrica, a numerika zahtjeva množenje i invertiranje većeg broja matrica). Nadalje, otvara se mogućnost pokušaja zaključivanja o parametru kompleksnih geometrija na osnovu njihovih sastavnih dijelova. Numerički postupak dopušta samo direktno računanje, ali potpuno je legitimno trenirati duboki model na jednostavnim sastavnicama geometrija i zatim ga zatražiti procjenu parametra kompleksne geometrije sastavljene od puno manjih jednostavnih dijelova. U daljnjem istraživanju, moguće je i izgraditi generativne modele te pokušati riješiti inverzni problem - za zadani efektivni parametar, kako bi mogla izgledati geometrija?

3 Metode

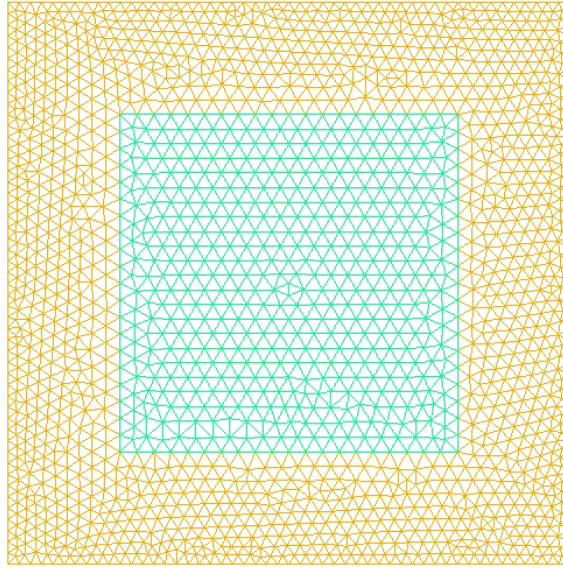
3.1 Računalno okruženje

Osnovno okruženje za rad sastojalo se od dva računala različitih specifikacija. Testiranje skripti, programiranje i računanje parametara provedeno je pod okruženjem Void Linux, na računalu s Intel i5-4200U procesorom te bez grafičke kartice. Samo treniranje modela napravljeno je pod okruženjem Ubuntu 18.04 LTS, na računalu s Intel i7-5960X procesorom te Nvidia GeForce GTX Titan Black grafičkom karticom. Razlog imanja dva računala leži u mogućnosti iskorištavanja grafičke kartice za brže treniranje dubokog modela. Cijeli opisani postupak bilo je moguće provesti i na jednom računalu bez grafičke kartice, samo bi cijeli postupak bio vremenski zahtjevniji.

Većina kodova napisano je u programskom jeziku Python, u verziji 3.8.2. Sortiranje datoteka u mape, preimenovanje datoteka te čuvanje ispisa programa postignuto je pomoću skriptnog jezika Bash kroz komandnu liniju operativnog sustava.

3.2 Priprema baze primjera

Prvi korak rada s modelom, u slučaju nadziranog učenja, uvijek je pribavljanje podataka za učenje. U ovome slučaju, to je značilo stvaranje baze jediničnih ćelija s različitim geometrijama te računanje njihovog efektivnog parametra, zatim spremanje rezultata u prikladnom formatu. U ovom slučaju, kao ulaz u neuronsku mrežu želimo predati sliku geometrije u obliku matrice, a zatim na izlazu očekujemo dvodimenzionalni vektor koji predstavlja efektivni parametar. Geometrija će imati označena dva materijala, te ćemo za parametar uzeti električnu permitivnost materijala.



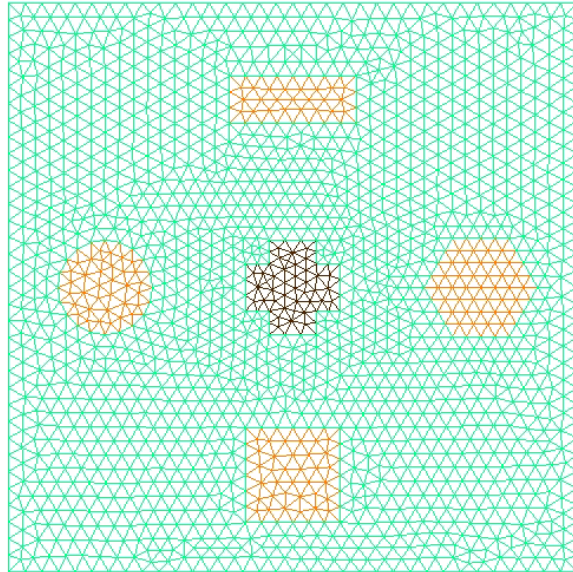
Slika 2: Kvadrat unutar jedinične ćelije dobiven Gmsh programom

3.2.1 Gmsh

Za stvaranje baze geometrija koristio se Gmsh [6], generator trodimenzionalnih geometrija i njihovih podjela na poddomene (eng. *meshing*). Gmsh implementira svoj vlastiti skriptni jezik unutar kojeg se definiraju elementi geometrije kao što su točke, linije, zatvorene krivulje, plohe i volumeni. Primjer jednostavne kvadratne domene unutar jedinične ćelije vidljiv je na slici 2, a primjer kompliciranije geometrije na slici 3.

U ovom radu, primjeri su generirani na idući način. Prvo je generirana jedinična ćelija veličine jedan sa jedan. Zatim je unutar te ćelije određeno devet ekvidistantnih pozicija, dakle tri sa tri mreža unutar jedinične ćelije. Definirani su oblici pravokutnika, kvadrata, kruga, šesterokuta te križa. U bazu su ubačene sve kombinacije i permutacije svakog pojedinačnog elementa sa nul-elementom (praznim poljem). Ukupni broj primjera dobiven ovom metodom je

$$5 \cdot 2^9 = 2560.$$



Slika 3: Kompliciranija geometrija jedinične ćelije dobiven Gmsh programom

Zatim je trebalo u bazu ugodati kombinacije različitih oblika međusobno, a ne samo s nul elementom. Ako bi se uzele sve kombinacije, broj primjera porastao bi na

$$6^9 = 10077696$$

što je prevelik broj podataka za obradu, i u prostornom i u vremenskom smislu. Tom problemu doskočili smo nasumičnim uzorkovanjem dodatnih 6000 primjera iz baze svih mogućih kombinacija, kako bi ukupni broj primjera došao na

$$2560 + 6000 = 8560.$$

3.2.2 Effective2D

Za računanje efektivnih parametara svih geometrija korišten je kod Effective2D [7] baziran na bibliotekama FEniCS [8] i DOLFIN [9], koji na temelju metode konačnih elemenata računa efektivnu permitivnost kompozita dva dielektrika. On kao ulaz prima

Gmsh geometriju, a na izlazu daje matricu koja na svojoj dijagonali ima efektivne parametre. Pomoću bash skripte ovaj algoritam upogonjen je za svih 8560 geometrija te su tako izračunati efektivni parametri spremljeni u tekstualnu datoteku. Dodatno, u kod je ubačena preinaka tako da za svaku učitane Gmsh geometriju kod generira i sprema sliku same geometrije kako bi se slika kasnije koristila na ulazu neuronske mreže.

3.3 Duboki model

Duboki modeli generalizacija su klasičnog modela regresije. U klasičnom modelu regresije, postavlja se problem pronalaska matrice \mathbf{A} takve da norma $\|\mathbf{Ax} - \mathbf{y}\|$ bude minimalna za neki zadani skup vektora značajki $\{\mathbf{x}\}$ i skup pripadajućih vektora oznaka $\{\mathbf{y}\}$. Unutar tog modela, svaki član vektora \mathbf{y} može se zapisati kao linearna kombinacija članova vektora \mathbf{x} , na način

$$y_i = \sum_j A_{ij}x_j.$$

Generalizacija ovog modela radi se na način da se svaki član $\sum_j A_{ij}x_j$ shvati ne kao konačan izlaz, već kao novi vektor značajki, koji se zatim opet linearno kombinira kako bi se dobio izlaz. Sada, dodavajući još jedan sloj, problem postaje minimizirati $\|\mathbf{BAx} - \mathbf{y}\|$. Nadalje, uvođenje ideje aktivacijske funkcije omogućava izradu nelinearnog modela (s obzirom da su sve operacije do sad bile isključivo linearne). Neka je f sada neka nelinearna funkcija. Optimizacijski problem postaje minimizirati

$$\|f(\mathbf{BAx}) - \mathbf{y}\|.$$

Ovakvi duboki nelinearni modeli u stanju su aproksimirati raznolike funkcije, točnije one koje se mogu prikazati kao kompozicija jednostavnijih funkcija. Više o navedenome može se naći u [10].

3.3.1 Konvolucijska neuronska mreža

Prijašnje opisan model primjeren je za podatke vektorskog tipa, ili one koji se daju elegantno prikazati kao vektori. No nastaje problem u trenutku pokušaja ubacivanja slike kao ulaza u standardni duboki model. Slika u računalu obično je prikazana u formatu koji u pozadini krije tri matrice sa intenzitetima crvene, zelene i plave boje. Zbog jednostavnosti objašnjenja, pretpostavimo da imamo crno-bijelu sliku prikazanu kao matricu u kojoj svaki član odgovara intenzitetu jednog piksela slike (tzv. binarna slika). Neka je slika standardne, ali za današnje pojmove malene, rezolucije od 640×480 . Razvlačenje slike u vektor daje vektor dimenzije 307200, što rezultira velikom matricom težina \mathbf{A} koja mora množiti zadani vektor te pretvara svaki član vektora \mathbf{y} u linearnu kombinaciju 307200 članova - previše za praktične svrhe.

Jedan od načina doskakanja tom problemu leži u prepoznavanju značajki (eng. *feature extraction*), gdje se umjesto slike kao ulaz stavlja vektor koji sadrži određene značajke prepoznate na slici - recimo broj prepoznatih linija, zasićenost slike ili pak momenti. Problem s ovim pristupom je zahtjevnost samog postupka prepoznavanja značajki - često je potreban cijeli dodatni model kako bi prepoznao značajke na slici a u tom slučaju opet vrijedi prvi problem.

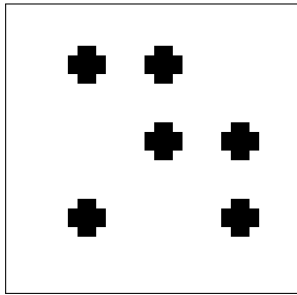
Drugi način je korištenje jezgrenih konvolucija (eng. *kernel convolutions*). Sliku shvatimo kao matricu, zatim na njene članove primjenjujemo jezgenu konvoluciju - konvoluciju između slike i jezgre koja je generalno matrica. Rezultat te konvolucije generalno je matrica manja od početne matrice slike. Konvolucija se primjenjuje nekoliko puta, s različitim veličinama jezgre i različitim vrijednostima u matrici jezgre. Izlazi se zatim koriste kao značajke za duboki model. Takav pristup zove se konvolucijska neuronska mreža i to je pristup korišten u ovom radu.

3.3.2 PyTorch

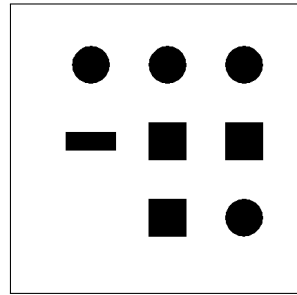
Duboki model iz ovog rada izveden je u programskom paketu PyTorch. Programski paket PyTorch biblioteka je za programski jezik Python koja sadrži funkcije i klase za jednostavno izrađivanje raznolikih modela strojnog učenja, pa tako i dubokog učenja. Uz same modele i njihove sastavne dijelove, PyTorch biblioteka sadrži i klase za pripremanje skupova podataka za učenje. Funkcije Dataset i DataLoader korištene su kao omotač oko podataka dobivenih korištenjem geometrija dobivenih Gmsh-om i parametara izračunatih Effective2D kodom.

3.3.3 Korišteni model

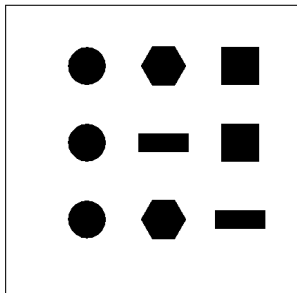
Duboki model korišten u ovom radu sastoji se od dvije konvolucije i tri potpuno povezana sloja, aktivirana ReLu funkcijama. Točna arhitektura može se isčitati iz koda 1, a detaljnija rasprava o aktivacijskoj funkciji može se pronaći u [11]. Ulazni podaci bili su pretvoreni u binarne slike (vidljivo iz slike 4), skupljeni na veličinu 224×224 (proizvoljno), pretvoreni u PyTorch tenzore te grupirani (eng. *mini-batching*). Duboki model optimiziran je korištenjem Adam [12] postupka, uz varijabilnu stopu učenja postignutu optimizatorom ReduceLRonPlateau. Optimizator Adam mijenja stopu učenja tokom procesa učenja. U trenutcima kada učenje ne postiže velike promjene, stopa učenja se povećava. Ako pak učenje počinje uzrokovati pogrešku, stopa se smanjuje. Nadalje, optimizator ReduceLRonPlateau smanjuje temeljnu stopu učenja koju Adam koristi svaki puta kada se na skupu podataka za validaciju ne postigne poboljšanje unutar nekoliko koraka učenja. Kao mjera pogreške korišteno je kvadratno odstupanje od oznaka pomoću funkcije MSELoss. U izradi modela korišteni su razni komadi PyTorch dokumentacije, većinom [13] i [14]. Također, korišten je dio koda iz rasprave [15] unutar StackOverflow foruma.



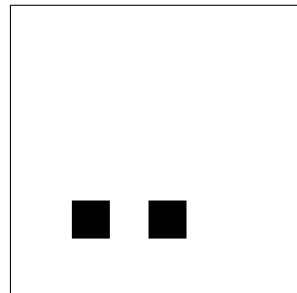
(a) Efektivni parametar: [10.0684, 10.0694]



(b) Efektivni parametar: [9.3746, 9.2356]



(c) Efektivni parametar: [9.2467, 8.9264]



(d) Efektivni parametar: [11.0386, 10.9152]

Slika 4: Primjeri binarnih slika kakve su korištene kao ulazni podatci

```

class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(44944, 120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 2)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = x.view(x.size(0), -1)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x

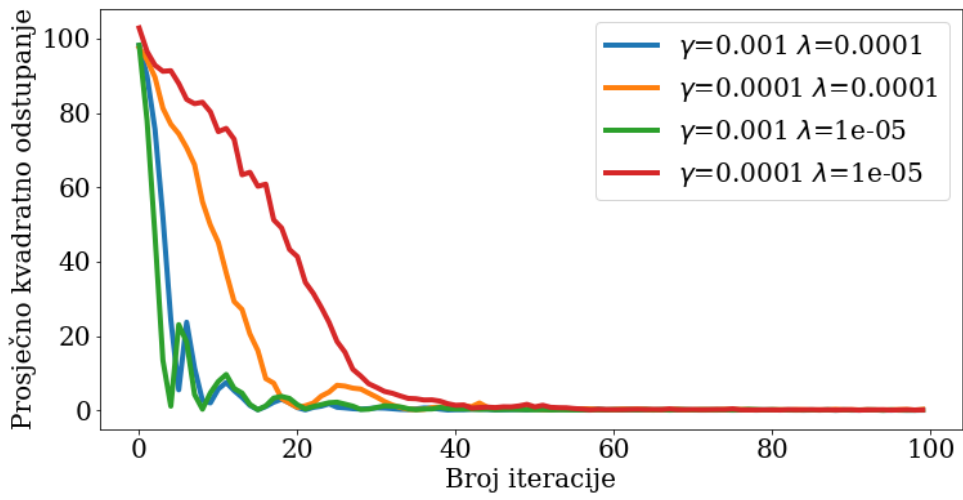
```

Kod 1: Python kod klase korištenog dubokog modela

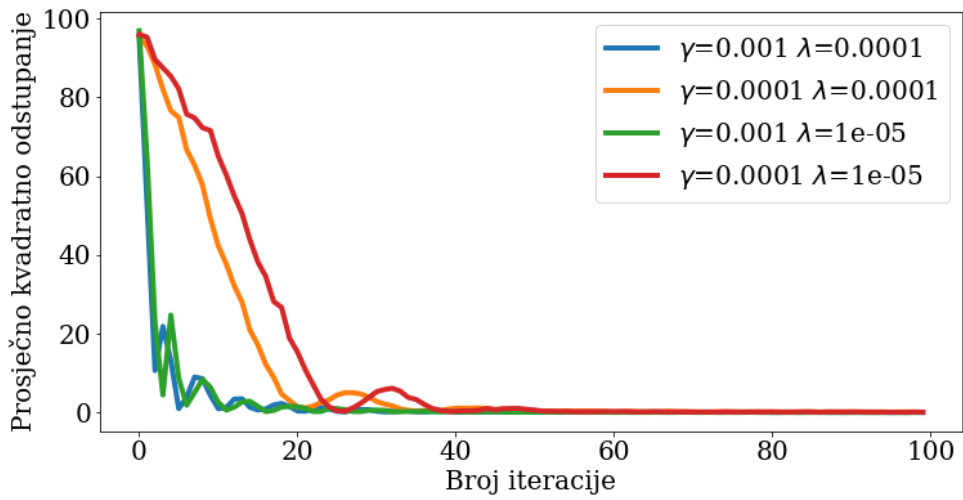
4 Rezultati

4.1 Pretraživanje parametara

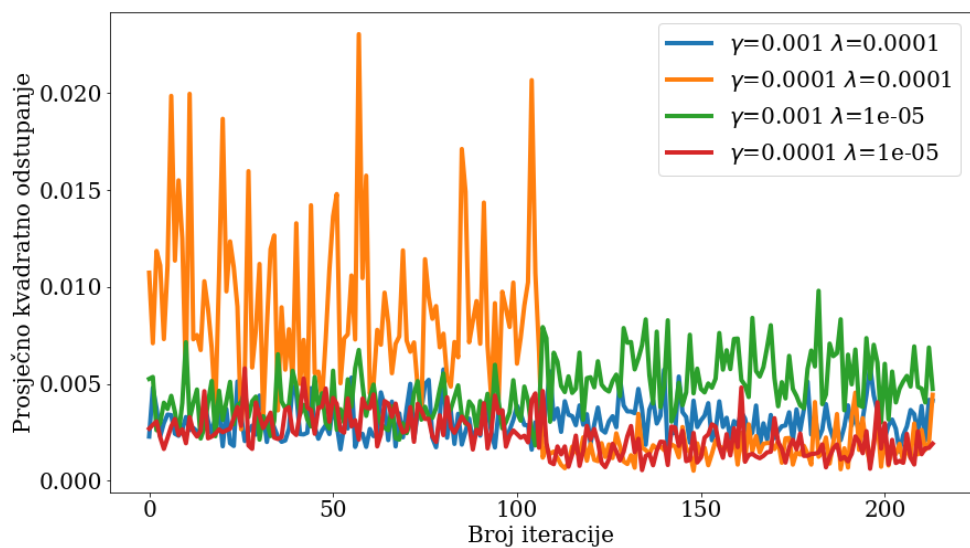
Opisani model treniran je Adam optimizatorom te je provedeno pretraživanje parametara po rešetci. Isprobane su kombinacije sa stopom učenja $\gamma \in \{0.001, 0.0001\}$, sa stopom regularizacije $\lambda \in \{0.0001, 0.00001\}$ te sa veličinom grupe (eng. *batch size*) $bs \in \{16, 32\}$. Sve kombinacije parametara daju zadovoljavajuće performanse, pogreška modela u svakom slučaju teži prema nuli nakon malog broja iteracija. Svaki model treniran je kroz dvije epohe, s obzirom da već unutar prve epohe greška dolazi vrlo blizu nule. Skup podataka podijeljen je na skup za učenje te skup za validaciju omjerom 80 : 20 u korist skupa za učenje. Na slikama 5 i 7 vidljive su performanse za minigrupe veličine 16, dok su na slikama 6 i 8 iste vidljive za minigrupe veličine 32. Grafovi su odvojeni jer zbog različite veličine minigrupe broj iteracija nije isti - veća minigrupa rezultira manjim brojem iteracija potrebnim kako bi se prošlo kroz sve primjere u skupu za učenje.



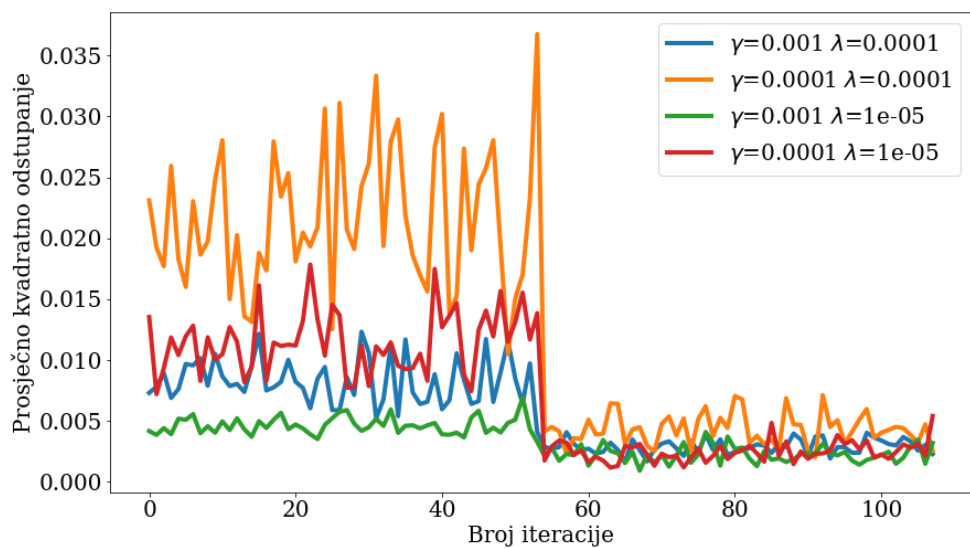
Slika 5: Pogreška predikcije na minigrupi veličine 16 tokom faze testiranja



Slika 6: Pogreška predikcije na minigrupi veličine 32 tokom faze testiranja



Slika 7: Pogreška predikcije na minigrupi veličine 16 tokom faze validacije



Slika 8: Pogreška predikcije na minigrupi veličine 32 tokom faze validacije

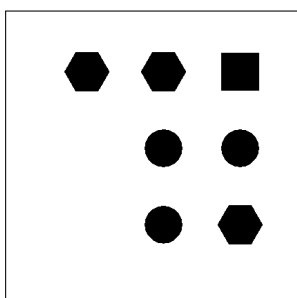
4.2 Najbolji model

Iako svi modeli postižu dobre performanse, model s konačno najmanjom greškom na skupu za validaciju bio je model treniran stopom učenja 0.001, stopom regularizacije od 0.00001 te sa minigrupama od po 32 primjera. Pet slika iz skupa za validaciju na kojima je model pokazao najtočniju predikciju vidljive su na slici 9, a pet slika na kojima je dao najlošiju predikciju vidljive su na slici 10. Iz uvida u primjere može se primjetiti kako model lošije procjenjuje jednolične geometrije od onih raznovrsnih.

4.3 Test na *dva sa dva* skupu

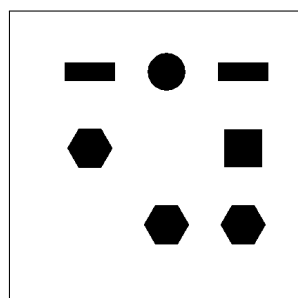
Posljednja provjera napravljena na modelu bila je ona sa potpuno drugačijim skupom podataka. Za potrebe testiranja, generirane su geometrije na mreži *dva sa dva*. Korištenjem metoda opisanih u poglavlju 3, izgrađen je skup podataka koji sadrži sve moguće kombinacije i permutacije pet geometrijskih oblika. Ovaj put, zbog manje mreže na koju se primjeri stavljaju, izgenerirana je svaka moguća kombinacija umjesto podskupa kombinacija kao što je to bilo potrebno na *tri sa tri* skupu. Rezultat je bio skup od 1296 geometrija. Elementarni oblici izgledali su isto kao i u prijašnjem skupu, ali su bili veći i bili su postavljeni na drugim mjestima unutar slike. Najbolji model iztreniran na originalnom *tri sa tri* skupu iskorišten je za predikcije na ovom skupu. Prosječna greška kroz cijeli skup iznosila je 1.4935. Ta pogreška puno je veća od one dobivene na originalnom skupu. Dapače, ta greška ukazuje na nemogućnost točne predikcije parametara u ovom slučaju. Taj zaključak donosimo provjerom parametara dobivenih na *dva sa dva* skupu. Očekivanje vektora parametra iznosi $[9.703, 9.405]$ uz standardnu devijaciju od $[0.552, 0.589]$. Dakle, nasumično izvlačenje iz normalne Gaussove distribucije s navedenim parametrima točnije je od računanja koristeći model.

Ipak, vrijedi pogledati najbolje i najgore predikcije koje model ostvaruje na promijenjenom skupu. Najbolje predikcije, vidljive na slici 11 dobivamo na geometrijama



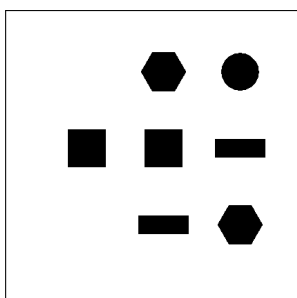
Predikcija: [9.495138,9.527328]

Stvarni parametar: [9.493496,9.527988]



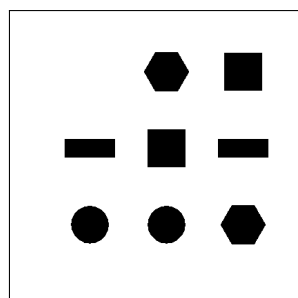
Predikcija: [9.737828,9.419177]

Stvarni parametar: [9.740836,9.418423]



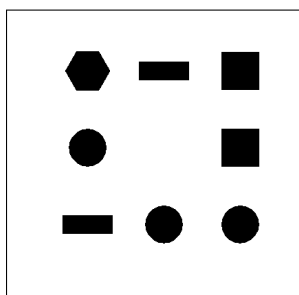
Predikcija: [9.71744,9.40456]

Stvarni parametar: [9.718775,9.406456]



Predikcija: [9.452213,9.149256]

Stvarni parametar: [9.450932,9.149215]

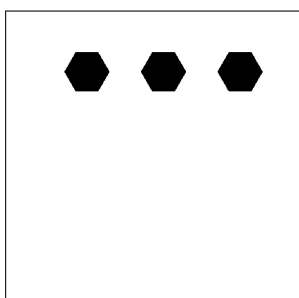


Predikcija: [9.507225,9.199731]

Stvarni parametar: [9.508434,9.2012005]

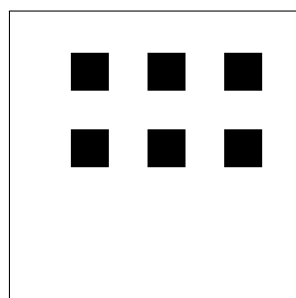
Slika 9: 3x3 geometrije s najboljim performansama

iznimno sličnima onima iz originalnog skupa podataka. Model u originalnom skupu ima geometrija jako sličnih zadanima, uz malu promjenu položaja i skaliranje. Sve najbolje geometrije sadrže točno jedan geometrijski lik. Primjećujemo da svaka od tih



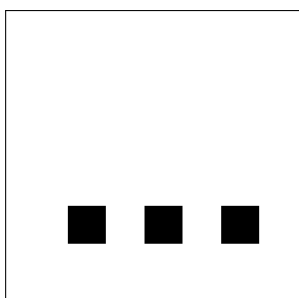
Predikcija: [10.636538, 10.669748]

Stvarni parametar: [10.803831, 10.49987]



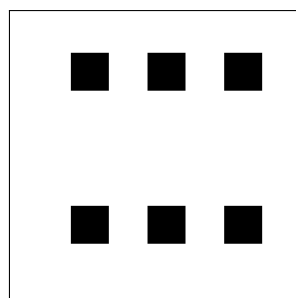
Predikcija: [9.513198, 9.563392]

Stvarni parametar: [9.700901, 9.399222]



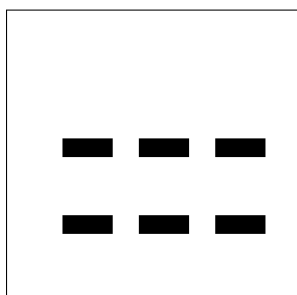
Predikcija: [10.556707, 10.614277]

Stvarni parametar: [10.755921, 10.403635]



Predikcija: [9.510986, 9.574716]

Stvarni parametar: [9.705782, 9.39971]



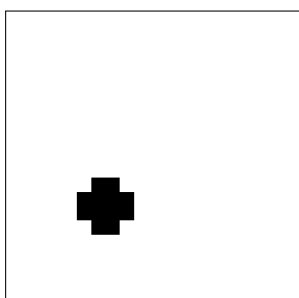
Predikcija: [10.741583, 9.715953]

Stvarni parametar: [10.821038, 9.522667]

Slika 10: 3x3 geometrije s najgorim performansama

geometrija imaju svoj ekvivalent u velikom skupu. Promatrajući parametre tih ekvivalenata, vidimo da je model u svakom slučaju bliži parametru ekvivalenta nego zadanom parametru. Najgore klasificirane geometrije sve sadrže pravokutnike i križeve, te su os-

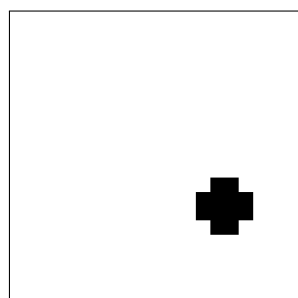
tvarene greške od nekoliko standardnih devijacija. Teške geometrije i njihovi parametri vidljivi su na 12. Iako postoje ekvivalenti i za ovaj skup geometrija, nije nužno da su ti ekvivalenti ušli u originalni skup za treniranje, s obzirom da su u *tri sa tri* skupu podataka geometrije s kombinacijama različitih oblika bile uzorkovane jer bi pokrivanje svih slučajeva zahtijevalo 6^9 slika.



Predikcija: [11.396071, 11.376707]

Stvarni parametar: [11.017323, 11.017544]

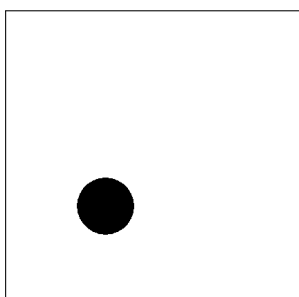
Ekvivalent: [11.4859344, 11.486590]



Predikcija: [11.398898, 11.368677]

Stvarni parametar: [11.017275, 11.017952]

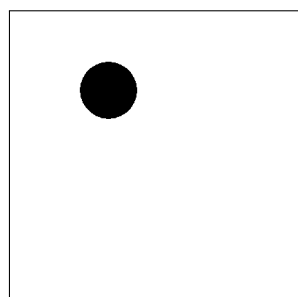
Ekvivalent: [11.4860039, 11.486362]



Predikcija: [11.36917, 11.351578]

Stvarni parametar: [11.037147, 11.037162]

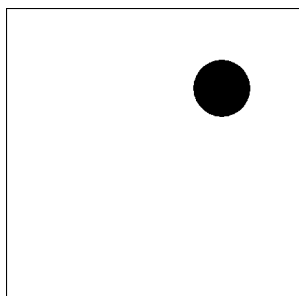
Ekvivalent: [11.4794554, 11.4792876]



Predikcija: [11.362464, 11.3375025]

Stvarni parametar: [11.027093, 11.027081]

Ekvivalent: [11.4851947, 11.4853597]

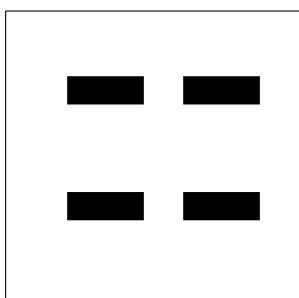


Predikcija: [11.364088, 11.335599]

Stvarni parametar: [11.037129, 11.037213]

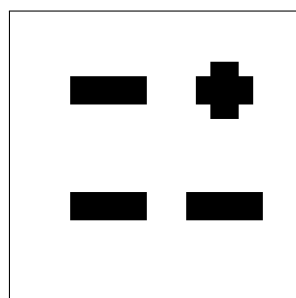
Ekvivalent: [11.4795191, 11.4794269]

Slika 11: 2x2 geometrije s najboljim performansama



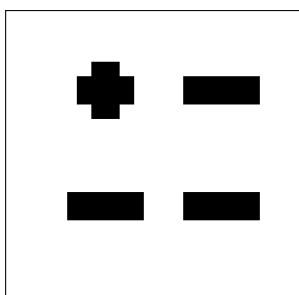
Predikcija: [10.767843, 10.808615]

Stvarni parametar: [10.080499, 8.338756]



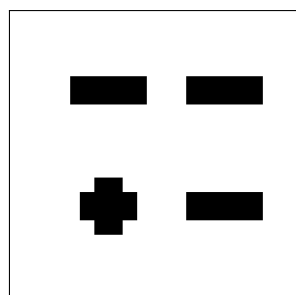
Predikcija: [10.725192, 10.7431965]

Stvarni parametar: [9.804124, 8.500903]



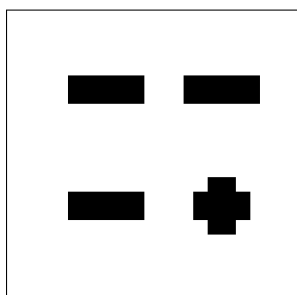
Predikcija: [10.729884, 10.745892]

Stvarni parametar: [9.8059845, 8.505385]



Predikcija: [10.727821, 10.74163]

Stvarni parametar: [9.806415, 8.508761]



Predikcija: [10.728396, 10.727259]

Stvarni parametar: [9.807047, 8.509377]

Slika 12: 2x2 geometrije s najgorim performansama

5 Zaključak i daljnji rad

Iz prikazanih rezultata može se zaključiti da se metode dubokog učenja mogu koristiti u računanju efektivnih parametara heterogenih periodičnih materijala. Treniranjem na

skupu od 6800 primjera i zatim provjerom na skupu od 1700 primjera primjećena je sposobnost modela da sa zadovoljavajućom točnošću predviđa parametre neviđenih primjera, predviđajući parametre s točnošću od cijelog dijela broja u najgorem slučaju do tri decimale u najboljem slučaju. Treniranje je uspješno provedeno sa različitim kombinacijama hiperparametara, iz čega se može zaključiti da je konstruirana mreža relativno stabilna te da daje konzistentne rezultate (greška treniranja i validacije teži u nulu u svim slučajevima).

Provjerom na potpuno različitom, *dva sa dva* skupu podataka uviđeno je da model ne može generalizirati na geometrije sa drugačije postavljenom mrežom. Dobiveni rezultati lošiji su nego obično izvlačenje iz normalne distribucije s parametrima proračunatim iz novog skupa podataka. Također, vidljivo je da u tom slučaju model bolje predviđa na primjerima koji su iznimno slični već viđenima - što naglašava potrebu za pažljivim modeliranjem skupa za učenje kako bi se postigle najbolje performanse na zadanom zadatku.

Daljnji rad moguć je u više smjerova. Prva mogućnost je uzeti identičan problem, ali promijeniti skup podataka za učenje. Trenutni skup podataka uključuje samo malen dio mogućnosti pri generiranju geometrija. Moguće je napraviti i geometrije koje uključuju rotacije, translacije i skaliranja oblika, te korištenje još više kombinacija i permutacija istih. Isto tako, moguće je promijeniti mrežu na kojoj se geometrija generira, kako bi uključivala geometrije *dva sa dva* ili *četiri sa četiri* oblika. Trenutna konstrukcija mreže invarijantna je na translacije i rotacije, s obzirom da je konvolucijska, te bi u tom slučaju trebalo razmisliti o samoj arhitekturi mreže kao i o skupu podataka. Na takvoj mreži mogla bi se dalje ispitivati moć generalizacije ovakvih modela. Druga je mogućnost promijeniti problem te rješavati inverzni problem. Korištenjem GAN-ova, varijacijskih autoenkodera ili drugih generativnih modela moguće je postaviti pitanje - kako izgleda tipična geometrija za neki zadani efektivni parametar. Također je moguće proučavati značajke koje zadani modeli prepoznaju kao bitne za generiranje primjera.

Sažetak

Autor: Luka Baković

Naslov: Homogenizacija elektromagnetskih kompozitnih materijala metodama dubokog učenja

Pojam kompozitnih materijala odnosi se na one kompozite sastavljene od nekoliko materijala s različitim fizikalnim svojstvima. Kompozitne materijale u današnjici srećemo u skoro svakom stupnju proizvodnje i u većini proizvoda koje koristimo, od armiranog betona pa do plastike. U procesu modeliranja procesa s takvim materijalima javlja se problem numeričkog rješavanja matematičkog modela u kojem se sureće velik broj jednažbi i velik zahtjev na računalnu snagu potrebnu za rješavanje zadanog sustava. Ovaj rad fokusira se na homogenizaciju kao rješenje navedenog problema, konkretno u slučaju elektromagnetizma. Proučava se primjena dubokog učenja na homogenizacijski problem, kao zamjena za rješavanje sustava jednažbi kako bi se došlo do homogenog parametra. Dobivaju se zadovoljavajući rezultati te točnost i do tri decimalna mjesta u izračunatom parametru. Provodi se i analiza na potpuno neviđenom skupu podataka s drugačijom postavom geometrijskih elemenata, te tako postavlja temelj za daljnji razvoj modela.

Ključne riječi rada: kompozitni materijali, elektromagnetizam, matematičko modeliranje, diferencijalne jednažbe, duboko učenje, konvolucijska mreža

Summary

Author: Luka Baković

Title: Homogenization of electromagnetic composite materials using deep learning

The term composite material refers to a material composed of several materials with different physical properties. In the modern world, they are encountered in every step of the production process, especially in the final product. Examples range from armed concrete to plastics. In the process of mathematical modeling of such systems, one encounters a problem in the numerical model - a very big number of equations which need to be solved for a solution. This puts a huge requirement on the computational resources required to solve such a problem. This paper focuses on homogenization as a solution, and looks at the particular case of an electromagnetic composite. We consider a homogenization method using deep learning as opposed to the solving of a system of differential equations. The results are promising, showing up to three decimal places of precision in some cases. Also, an analysis is made on the generalization abilities of the model. This was done by testing the model on a completely new data set, and this can be interpreted as laying down the basis for future research.

Keywords: composite materials, electromagnetism, mathematical modeling, differential equations, deep learning, convolutional networks

6 Literatura

- [1] D. Felbacq, G. Bouchitte, *Metamaterials Modelling and Design*, Pan Stanford Publishing, 2017
- [2] Krowne, Clifford M., Zhang, Yong, *Physics of Negative Refraction and Negative Index Materials*, 2007, ISBN 978-3-540-72132-1
- [3] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naf-tali Tishby, Leslie Vogt-Maranto and Lenka Zdeborová. Machine learning and the physical sciences, *Rev. Mod. Phys.* 91, 045002, 2019 arXiv:1903.10563.
- [4] Thomas Christensen, Charlotte Loh, Stjepan Picek, Domagoj Jakobović, Li Jing, Sophie Fisher, Vladimir Ceperic, John D. Joannopoulos i Marin Soljačić Predictive and generative machine learning models for photonic crystals, *Nanophotonics*, DOI: <https://doi.org/10.1515/nanoph-2020-0197>
- [5] Andrew Zangwill, *Modern Electrodynamics*, Cambridge University Press, 2020, ISBN 978-0521896979
- [6] C. Geuzaine i J.-F. Remacle., Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, *International Journal for Numerical Methods in Engineering* 79(11), 2009.
- [7] Dario Bojanjac i Darko Janeković, Effective2D, github repozitorij, <https://github.com/dbojanjac/effective2D/>
- [8] M. S. Alnaes, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes i G. N. Wells, The FEniCS Project Version 1.5, *Archive of Numerical Software*, vol. 3, 2015
- [9] A. Logg and G. N. Wells, DOLFIN: Automated Finite Element Computing, *ACM Transactions on Mathematical Software*, vol. 37, 2010

- [10] Ian Goodfellow i Yoshua Bengio i Aaron Courville, Deep Learning, MIT Press, 2016
- [11] Ovidiu Calin, Deep Learning Architectures: A Mathematical Approach, Springer, 2020, ISBN 978-3030367206
- [12] Diederik P. Kingma i Jimmy Ba., Adam: A Method for Stochastic Optimization, 2014, arXiv:1412.6980.
- [13] Upute za rad sa skupovima podataka unutar PyTorch dokumentacije, https://pytorch.org/tutorials/beginner/data_loading_tutorial.html
- [14] Upute za treniranje mreža unutar PyTorch dokumentacije, https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html
- [15] Rasprava unutar pitanja na StackOverflow forumu, <https://stackoverflow.com/questions/50544730/how-do-i-split-a-custom-dataset-into-training-and-test-datasets/>