SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 6348

**VIZUALIZACIJA I INTERPRETACIJA KONVOLUCIJSKIH MODELA**

Martin Sršen

Zagreb, lipanj 2019.

Uvod

Umjetna inteligencija (engl. artificial intelligence), ponekad nazvana strojna inteligencija je područje računalne znanosti koje pokušava kreirati stroj koji će oponašati kognitivne funkcije čovjeka, kao što su učenje i rješavanje problema. Ti problem mogu biti vrlo jednostavni za čovjeka, dok su u istom trenutku za računalo vrlo teški i zahtjevni. Grana umjetne inteligencije koja proučava računalne algoritme koji se automatski poboljšavaju iskustvom bez uporabe eksplicitnih uputa, tj. uče na temelju datih podataka naziva se strojno učenje (engl. machine learning). Zanimljivo područje strojnog učenja kojeg ćemo proučavati u ovom radu je računalni vid, koji se bavi teorijom iza umjetnih sustava koji izvlače informacije iz slika. Područje inspirirano na temelju ljudskog vida koji je najvažnije osjetilo kod ljudi. Neke od pod-domena računalnog vida su rekonstrukcija prizora (proces hvatanja oblika I izgleda stvarnih objekata), otkrivanje događaja, video praćenje, prepoznavanje objekata (utvrđuje dali slika sadrži određeni object, značajku ili aktivnost), procjena 3D pozicije, učenje, indeksiranje, procjena pokreta te restauracija slika (uklanjanje smetnji na slikama).

Velika prekretnica u računalnom vidu se dogodila 2012. kada je konvolucijska neuronska mreža (tip duboke neuronske mreže) predstavljena svijetu na glavnom natjecanju računalnog vida, ImageNet natjecanje u detekciji i prepoznavanju 1000 različitih objekata na slikama. Umjetne neuronske mreže imaju začetke u 40-tim godinama 20. Stoljeća, ali nisu doživjele procvat zbog slabe razvijenosti računala te nemogućnosti izvođenja većih i kompleksnijih izračuna. Razvojem grafičkih kartica umjetne neuronske mreže doživljavaju procvat jer omogućuju veliku količinu paralelnih izračuna. Prednost korištenja umejtne neuronske mreže je što se ne trabaju koristiti kompleksni matematički modeli te ručno upisivanje filtera već ih mreža sama uči na osnovu danih podataka.

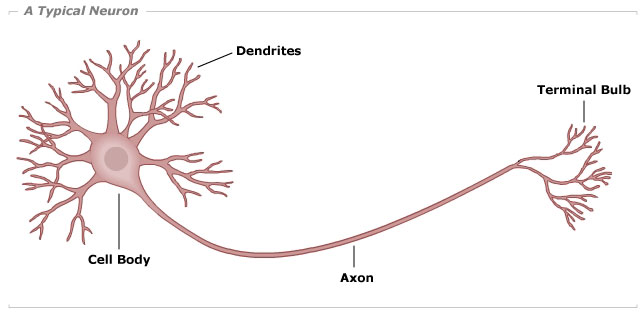
Cilj ovog rada je upoznati se s metodama vizualizacije te interpretacije konvolucijskih modela te njihovih parametara, proučiti metode vizualizacije, tj. vidjeti na koji način umjetna neuronska mreža prepoznaje objekt na slici, koji dijelovi slike igraju ključnu ulogu u klasifikaciji.

Umjetna neuronska mreža

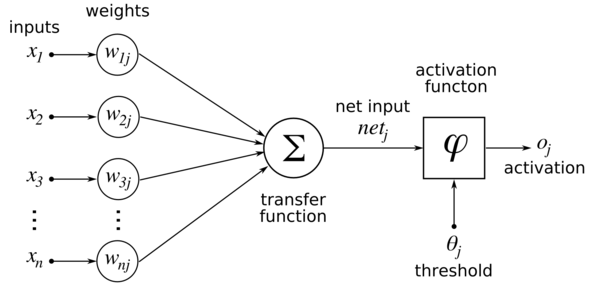
Umjetna neuronska mreža, jedan od važnijih algoritama strojnog učenja koji je motiviran na principu rada biološkog mozga. Mozak je središnji organ svakog živog bića koji mu omogućuje učenje, pamćenje te rješavanje raznih problema. Cilj umjetne inteligencije je upravo reproducirati mogućnosti mozga te ih poboljšati, pa je zbog toga razumno krenuti od funkcioniranja mozga te reporoduciranju funkcioniranja istog. Ljudski mozak se sastoji od oko 86 milijuna neurona međusobno povezanih sa 1014 - 1015 sinapsi.

Neuron

Osnovna građevna jedinica svake umjetne neuronske mreže. Inspiriran biološkim neuronom koji funkcionira na način da prikuplja informacije iz susjednih, povezanih neurona kroz dendrite, obrađuje signal te prosljeđuje kroz aksone preko sinaptičkih veza u slučaju da je signal dovoljno jak.



Umjetni neuron je elementarna jedinica u umjetnoj neuronskoj mreži koji je zapravo matematička funkcija koja pokušava imitirati rad biološkog neurona. Funkcionira na način da prima jedan ili više ulaza ,sumira ih te dodaje prag kako bi proizveo izlaz. Uloga praga je da se odredi minimalna vrijednost koju neuron mora imati da bi postao aktivan. Obično se ulaz množi sa pripadajučim težinama koje predstavljaju jačinu veze između dva neurona te se suma prenosi kroz nelinearnu aktivacijsku/prijenosnu funkciju. Izlaz aktivacijske funcije jednog neurona predstavlja ulaz idućeg neurona.



Funkcija koja predstavlja rad umjetnog neurona:

gdje je značenje varijabli iduće:

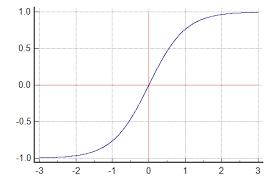
* y – izlaz neurona, tj. ulaz u idući neuron u nizu
* f – aktivacijska/prijenosna funkcija
* x – ulaz u neuron
* w – težina između prijašnjeg i trenutnog neurona
* – prag

Umjetni neuroni se razlikuju na temelju aktivacijskih/prijenosnih funkcija. Ona može biti proizvoljna te se odabire ovisno o problemu koji rješava. Kada nebi koristili aktivacijsku funkciju, mala promjena ulaza bi mogla potpuno promjeniti izlaz, te bi izlaz bio potpuno linearno ovisan o ulazu. Aktivacijske funkcije koje se najčešće koriste su:

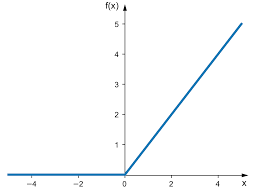
* Sigmoidalna funkcija – glavno obilježje ove funkcije je da svaki ulaz skalira na interval između 0 i 1. Jako pozitivni brojevi poprimaju vrijednost približno 1, dok jako negativni približno 0. Sigmoidalna funkcija se koristi sve manje zbog niza razloga, jedan od glavnih je što kod povratne propagacije gradijent postaje sve manji kroz mrežu, te za duboke neuronske mreže može potpuno nestati.



* Tanh – glavna razlika u odnosu na sigmoidalnu funkciju je da skalira ulaz na interval između -1 i 1. Ovo rješava problem centriranja vrijednosti oko 0, te je zbog toga preporućeno koristiti nju u odnosu na sigmoidalnu funkciju.



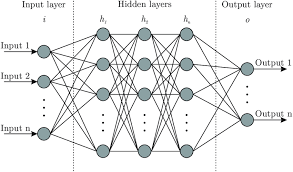
* ReLU (Rectified Linear Unit) – funkcionira na način da propušta ulaze koji su pozitivni, tj. veći od 0. Glavna prednost je brzina računanja što smanjiva vrijeme treniranja. Glavni problem je što neuroni mogu „odumrijeti” na način da preveliki gradijent izmjeni težine na način da se neuron više nikad ne aktivira. Problem se može rješiti na način da gradijent ne bude 0 nego neka mala vrijednost.



Arhitektura neuronske mreže

Neuronska mreža nastaje povezivanjem većeg broja neurona. Osnovna arhitektura mreže je podjeljena na 3 sloja:

1. Ulazni sloj – predstavlja ulazne podatke u mrežu
2. Skriveni slojevi – ne moraju postojati, ali su upravo oni ključni u rješavanju težih problema
3. Izlazni sloj – izlaz mreže



Svaki sloj neuronske mreže ima vezu sa susjednim na način da izlazi prethodnog sloja predstavljaju ulaze u idući sloj. Moguće su i arhitekture u kojem postoje veze na prijašnje slojeve, ali to nećemo razmatrati. U slučaju kada je svaki neuron prethodnog sloja povezan sa svakim neuronom idućeg sloja to nazivamo potpuno povezanom neuronskom mrežom. Neuronske mreže koje imaju 2 ili više skrivenih skojeva nazivamo dubokim neuronskim mrežama. Veći broj slojeva nam omogućuje preciznije dekomponirati problem na niz jednostavnijih dijelova te je time dekompozicija preciznija i rješenje je pouzdanije, ali je učenje dulje te složenije s povećanjem broja slojeva.

Učenje neuronske mreže

Učenje neuronske mreže predstavlja modifikaciju promjenjivih parametara mreže, težina u svrhu što boljih rezultata mreže. Razlikujemo 3 načinja učenja s obzirom na oblik podataka:

1. nadzirano učenje (engl. supervised) – za svaki ulaz u mrežu znamo očekivani izlaz
2. nenadzirano učenje (engl. unsupervised) – znamo samo ulaze u mrežu
3. polu-nadzirano učenje (engl. semi-supervised learning) – koristi manju grupu podataka za koje znamo izlaz
4. podržano učenje (engl. reinforcement) – koristi se sustav nagrađivanja, dajući mreži povratnu informaciju

Cilj neuronske mreže kod nadziranog učenja je minimizirati funkciju gubitka. Razlikujemo dvije grupe funkcija gubitka s obzirom na problem:

1. klasifikacijski problem – tražimo kategoriju kojoj pripada dani ulaz
2. regresijski problem – tražimo kontinuirani izlaz u odnosu na ulaz

Najčešće korištena funkcija gubitka za problem regresije je funkcija srednje kvadratne pogreške (engl. mean square error) ili L2 gubitak,

dok za problem klasifikacije se najčešće koristi unakrsna entropija (engl. cross entropy loss)

Sada kada znamo kako izračunati grešku treba vidjeti kako promjeniti parametre mreže tako da u budućnosti greška bude što manja. Formula kojom ažuriramo parametre/težine neuronske mreže je sljedeća:

gdje je faktor učenja, hiper-parametar koji kontrolira koliko prilagođavamo težine i pragove s obzirom na gradijent gubitka. Što veći faktor uzmemo, brže je učenje, ali riskiramo da promašimo lokalni minimum.

Sada još trebamo vidjeti kako dobiti gradijente po težinama, tj. vidjeti kako dobiti koliko koja težina te prag utječu na promjenu gubitka. Za to se koristi algoritam propagacije pogreške unazad (engl. backpropagation), glavni i neizostavni algoritam za treniranje neuronske mreže. Upravo ovaj algoritam daje mogućnost mreži da uči na svojim pogreškama te postane bolja u rješavanju određenih problema. Koristimo ga da saznamo koliko je koja težina utjecala na grešku i nakon toga ažuriramo pripadajuće težine s obzirom na faktor učenja.