SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1160

Algoritmi za brzo učenje na neprijateljskim primjerima

Dominik Jambrović

Zagreb, svibanj 2023.

SADRŽAJ

1. Uvod

Velik broj problema s kojima se danas susrećemo takve su prirode da ne znamo kako ih riješiti koristeći klasičan, algoritamski pristup. Razlog tome često leži u činjenici da ne znamo ni kako mi sami rješavamo te probleme, a jedan od najčešćih primjera za to je raspoznavanje tj. klasifikacija slika. Jedno od mogućih rješenja takvih problema je korištenje umjetnih neuronskih mreža - mreža sastavljenih od velikog broja povezanih jedinica (neurona) koje obavljaju veoma jednostavne operacije.

Razvojem dubokih neuronskih mreža došlo je do ubrzanog napretka u području računalnog vida. Računalni vid područje je umjetne inteligencije koje se bavi problemima poput klasifikacije 2D slika. Sve većom popularizacijom i korištenjem dubokih modela u sustavima različitih namjena, u pitanje se dovodi sigurnost takvih modela - ako model želimo koristiti u automobilima s ciljem detekcije pješaka i vozila, model mora moći dobro generalizirati, kao i biti robustan. Takav model ne bi smio mijenjati svoje odluke na temelju veoma malih promjena na ulazu - na primjeru prometa, želimo da model točno detektira pješaka, bez obzira nosi li on kapu ili ne.

Kako bi ostvarili robustnost modela, predložene su brojne tehnike, a jedna od najpopularnijih je robusno učenje tj. učenje na neprijateljskim primjerima. Kada su u pitanju modeli koji brzo uče, robusno učenje prihvatljivo je rješenje za postizanje robusnih modela otpornih na napade. Ipak, kada su u pitanju veći modeli za koje učenje traje veoma dugo, obično robusno učenje često je neprihvatljivo. U tu svrhu, razvijene su metode koje ubrzavaju robusno učenje. U ovome radu, razmatrat ćemo tri takve metode: besplatno robusno učenje [6], brzo robusno učenje [7] i nadogradnju na brzo robusno učenje (FastAdv+ i FastAdvW, [4]).

Uz to, razmatrat ćemo i otpornost naučenih modela na zatrovane podatke. Zatrovani podatci ulazi su izmijenjeni s ciljem navođenja modela na neočekivano ponašanje. Uvođenjem takve ranjivosti u model, napadači mogu neprimjećeno postići proizvoljne ciljeve poput izbjegavanja detekcije ili pogrešne klasifikacije.

2. Neuronske mreže

2.1. Općenito o neuronskim mrežama

Umjetne neuronske mreže veoma su popularan alat za rješavanje kompleksnih problema za koje je teško modelirati ili formalizirati znanje. Predstavnik su konektivističkog pristupa umjetnoj inteligenciji [1] koji se zasniva na oblikovanju sustava inspiriranih građom mozga.

Problemi koje rješavamo umjetnim neuronskim mrežama svrstavaju se u dvije glavne kategorije:

1. klasifikacija
2. regresija

Kada su u pitanju klasifikacijski problemi, cilj nam je svrstati ulaz u jedan od mogućih razreda. Pritom je na izlazu često korišteno jednojedinično kodiranje - ako ulaze svrstavamo u 10 razreda, u izlaznom sloju mreže bit će 10 neurona, a aktivacija jednog od njih predstavljat će rezultat klasifikacije.

S druge strane, kod regresijskih problema cilj nam je što bolje aproksimirati neku, modelu nepoznatu, funkciju. Za ovakve probleme, često nam je dovoljan jedan neuron u izlaznom sloju. Taj neuron na izlazu bi trebao davati predviđenu vrijednost funkcije za neki, do sada neviđeni, ulaz.

Da bi neuronska mreža mogla rješavati takve probleme, važno nam je da može učiti na temelju predočenih podataka. Učenje neuronske mreže odvija se izmjenom težina pojedinih neurona (time znanje implicitno ugrađujemo u našu mrežu). Kako bismo detaljnije mogli govoriti o učenju i arhitekturama neuronskih mreža, važno je ukratko opisati neuron - osnovnu gradivnu jedinicu svake mreže.

2.2. Umjetni neuron

Umjetni neuroni predstavljaju jednostavne procesne jedinice koje modeliraju ponašanje prirodnih neurona. Osnovni neuron akumulira vrijednosti na ulazu pomnožene težinama, akumuliranoj vrijednosti dodaje pomak te na kraju istu propušta kroz prijenosnu (aktivacijsku) funkciju. Ponašanje jednog neurona možemo modelirati jednadžbom:

$$o = f(\sum_{i=1}^{n} x_i w_i + b) \tag{2.1}$$

pri čemu x označava pojedine ulaze, w težine na pripadnim ulazima, b pomak te f prijenosnu funkciju.

2.3. Prijenosne funkcije

Neki od najranijih modela umjetnog neurona kao prijenosnu funkciju koristili su funkciju identiteta (ADELINE-neuron) te funkciju skoka (TLU-perceptron). S vremenom su korištene i razvijene brojne druge prijenosne funkcije poput sigmoidalne funkcije definirane kao:

$$sigmoid(net) = \frac{1}{1 + e^{-net}} \tag{2.2}$$

i zglobnice (ReLU - engl. *Rectified Linear Unit*) definirane kao:

$$relu(net) = max(0, net) \tag{2.3}$$

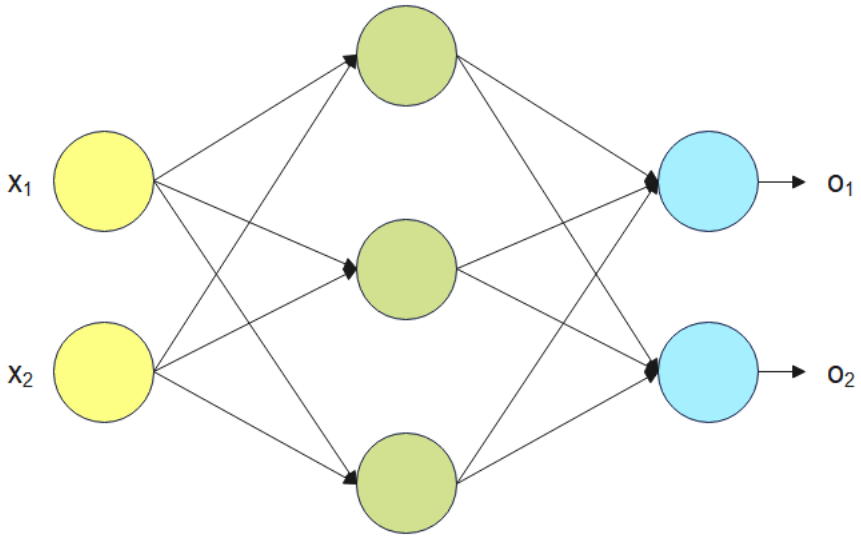
Važno je primijetiti da ako je prijenosna funkcija linearna, cijeli neuron može postići isključivo linearnu transformaciju. Kako bi umjetnim neuronima mogli modelirati kompleksnije funkcije, koristimo nelinearne prijenosne funkcije poput sigmoidalne funkcije i zglobnice. Pritom je za duboke neuronske mreže s velikim brojem slojeva često korištena upravo zglobnica - sigmoidalna funkcija za takve mreže nije prikladna zbog problema nestajućeg gradijenta (engl. *vanishing gradient problem*) koji nastaje tijekom učenja temeljenog na gradijentnim metodama.

2.4. Arhitektura umjetne neuronske mreže

Kada je za neki problem potrebno koristiti više od jednog osnovnog neurona, neurone povezujemo u mrežu. Pritom kažemo da se neuronska mreža sastoji od nekolicine slojeva:

1. ulazni sloj
2. skriveni slojevi
3. izlazni sloj

Iako ulazni sloj predstavljamo neuronima, oni, za razliku od neurona u skrivenim slojevima i izlaznom sloju, ne obavljaju nikakve transformacije - možemo reći da predstavljaju ulazni podatak. Veličina ulaznog sloja govori nam o dimenzionalnosti ulaznih podataka, a veličina izlaznog sloja u slučaju problema klasifikacije često nam govori o broju razreda u koje klasificiramo ulaz.



Slika 2.1: 2x3x2 arhitektura umjetne neuronske mreže

Na slici 2.1 moguće je vidjeti primjer arhitekture umjetne neuronske mreže. Neuroni označeni žutom bojom predstavljaju ulazni sloj, neuroni označeni zelenom bojom skriveni sloj, a neuroni označeni plavom bojom izlazni sloj. Ovakvu arhitekturu mreže skraćeno možemo označiti kao 2x3x2 neuronsku mrežu. Pritom brojke označavaju broj neurona u pojedinom sloju (ulazni sloj je prvi sloj mreže).

Za ovakvu mrežu kažemo da je unaprijedna potpuno-povezana mreža. Pojam unaprijedna mreža označava to da ne postoje veze iz dubljih slojeva prema plićim slojevima, a pojam potpuno-povezana mreža označava to da svaki neuron ima vezu sa svakim neuronom iz prethodnog sloja. Uz to, svi neuroni imaju i dodatnu težinu zvanu pomak (nije prikazano na slici 2.1). Djelovanje jednog sloja mreže sažeto možemo prikazati kao:

$$\boldsymbol{h}_i = f(\boldsymbol{W}_i \boldsymbol{h}_{i-1} + \boldsymbol{b}_i) \tag{2.4}$$

pri čemu W_i predstavlja težine trenutnog sloja, b_i pomake trenutnog sloja, h_{i-1} izlaz iz prethodnog sloja, f prijenosnu funkciju primijenjenu na svaki element te h_i izlaz iz trenutnog sloja. Korištenjem takve formule za svaki sloj mreže, na kraju ćemo dobiti izlaz mreže za neki proizvoljni ulaz. Ovo nazivamo unaprijednim prolazom.

2.5. Učenje neuronske mreže

Kako bismo mogli koristiti proizvoljnu mrežu za probleme klasifikacije ili regresije, potrebno ju je prvo naučiti. Kao što je već prethodno rečeno, učenje neuronske mreže odgovara izmjeni težina pojedinih neurona, a najčešće se postiže algoritmom propagacije pogreške unatrag [2]. Da bismo mogli znati kako trebamo izmijeniti težine neurona, prvo trebamo znati koliko naš model griješi. Mjera greške naziva se gubitak, a računa se na temelju izlaza modela i očekivanog (točnog) izlaza. Za izračun gubitka često je korištena unakrsna entropija koju možemo definirati kao:

$$H(P^*|P) = -\sum_i P^*(i) \log P(i) \tag{2.5}$$

pri čemu $P^*(i)$ označava distribuciju očekivanog izlaza, a P(i) distribuciju izlaza modela.

Jednom kada znamo iznos gubitka, koristeći optimizatore poput stohastičkog gradijentnog spusta (SGD) ili Adam optimizatora [3] možemo poboljšati naš model. Pritom nam je za optimizacijski postupak veoma često potreban gradijent funkcije gubitka po parametrima modela, a izračunavamo ga uzastopnom primjenom pravila ulančavanja koje u svojem najjednostavnijem obliku možemo definirati kao:

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx} \tag{2.6}$$

U slučaju stohatičkog gradijentnog spusta, iterativno ažuriramo težine modela na temelju iznosa gradijenta funkcije gubitka, kao i stope učenja. Sažeto postupak stohastičkog gradijentnog spusta možemo prikazati jednadžbom:

$$w_{i+1} := w_i - \eta \nabla_w L(w_i, x) \tag{2.7}$$

pri čemu w_i označava jednu od težina modela u trenutnoj iteraciji, w_{i+1} istu težinu u sljedećoj iteraciji, $L(w_i, x)$ funkciju gubitka, a η stopu učenja. Stopa učenja mali je pozitivni broj pomoću kojeg možemo kontrolirati iznos promjene težina po iteracijama. Iterativnim ponavljanjem ovakvog postupka minimiziramo iznos funkcije gubitka, time dobivajući što bolju mrežu. Kod stohastičkog gradijentnog spusta, iteracije mogu biti predstavljene pojedinačnim ulazima ili mini-grupama. Zbog činjenice da SGD pohranjuje gradijent za male ulaze, ovaj optimizacijski postupak zahtijeva malo memorije.

Nakon učenja mreže na skupu za učenje, evaluirat ćemo performanse mreže na neviđenom skupu zvanom skup za testiranje (engl. *test set*). Često korištena mjera za kvalitetu modela je točnost definirana kao:

$$accuracy = \frac{correct}{total} \tag{2.8}$$

pri čemu correct označava broj točno klasificiranih primjera, total ukupan broj primjera, a accuracy točnost. Uz točnost, postoje i brojne druge mjere kvalitete modela. Neke od njih su preciznost (engl. *precision*), odziv (engl. *recall*) i matrica zabune (engl. *confusion matrix*).

2.6. Duboke neuronske mreže

Ako želimo rješavati složenije probleme koristeći umjetne neuronske mreže sa samo jednim skrivenim slojem, suočit ćemo se s problemom - broj neurona potreban kako bi umjetna neuronska mreža mogla obavljati svoju zadaću bit će prevelik. Uz to, korištenjem širokog modela s velikim brojem neurona u skrivenom sloju teško ćemo postići svojstvo generalizacije jer će se model lako prenaučiti i zapamtiti ulazne podatke.

Zbog tih razloga, veoma su popularne duboke neuronske mreže [2]. Duboke neuronske mreže, za razliku od mreža sa samo jednim skrivenim slojem, imaju nekolicinu skrivenih slojeva. Pritom za rješavanje složenijih problema duboke mreže trebaju imati značajno manje neurona po sloju naspram mreže sa samo jednim skrivenim slojem. Zasebni slojevi mreže naučit će prepoznavati zasebne značajke ulaza, a njihovom kombinacijom mreža će moći postići uspješnu klasifikaciju.

Ipak, postoje i određene mane dubokih neuronskih mreža. Jedna od mana činjenica je da je za propagaciju pogreške unazad kod ovakvih mreža potrebno množiti gradijente. U slučaju da kao prijenosnu funkciju koristimo sigmoidalnu funkciju, ovo lako vodi do problema nestajućeg gradijenata zbog kojega težine neurona u plitkim slojevima nećemo moći ispravno izmijeniti. Uz problem nestajućeg gradijenta, postoji i problem eksplodirajućeg gradijenta (engl. *exploding gradient problem*) koji se pojavljuje kod nekih drugih prijenosnih funkcija od kojih je najpoznatija upravo zglobnica (ReLU). Još jedna mana dubokih neuronskih mreža činjenica je da kako bismo kvalitetno naučili duboku mrežu moramo imati veoma velik skup podataka.

2.7. Konvolucijske mreže

Za probleme s velikom dimenzionalnošću ulaza, potpuno-povezane mreže imaju izuzetno velik broj parametara tj. težina neurona. Učenje ovakvih modela zahtijeva veliku količinu memorije, a i općenito je sporo. Uz to, potpuno-povezane mreže osjetljive su na translaciju ulaza: ako učimo mrežu da klasificira slike vozila te nakon učenja mreži predočimo translatiranu sliku iz skupa za učenje, mreža tu sliku neće nužno moći ispravno klasificirati jer je za nju to potpuno novi podatak. Ovo svojstvo proizlazi iz činjenice da je svaki neuron ovisan o svakom neuronu iz prethodnog sloja.

Kako bi doskočili ovim problemima, uvedene su konvolucijske mreže [5]. Za razliku od potpuno-povezanih mreža, ovdje aktivacija neurona ne ovisi o svim neuronima iz prethodnog sloja, već samo o malom rasponu neurona iz prethodnog sloja. Time konvolucijska mreža ima značajno manje parametara naspram potpuno-povezane mreže iste dubine, a postiže i svojstvo translacijske invarijantnosti.

Ova svojstva konvolucijska mreža postiže zamjenom standardnog množenja matrica operacijom konvolucije s jezgrom (engl. *kernel*). Općenito govoreći, operaciju konvolucije za dvije funkcije *f, g* možemo definirati kao:

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{2.9}$$

Operacija konvolucije opisuje nam kako se izgled jedne funkcije mijenja pod utjecajem druge funkcije, a definirana je kao integral umnoška funkcija nakon što je jedna reflektirana i translatirana. Uz operaciju konvolucije postoji i unakrsna korelacija definirana kao:

$$(f \star g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t+\tau)d\tau \tag{2.10}$$

Važno je primijetiti da je glavna razlika između te dvije operacije izostajanje reflektiranja jedne od funkcija u slučaju operacije unakrsne korelacije. Kada kod konvolucijskih mreža govorimo o konvoluciji, gotovo uvijek se zapravo misli na unakrsnu korelaciju.

3. Zaključak

Zaključak.

LITERATURA

[1] Bojana Dalbelo Bašić, Marko Čupić, i Jan Šnajder. Umjetne neuronske mreže, prezentacija, 2020.

[2] Ian Goodfellow, Yoshua Bengio, i Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[3] Diederik P Kingma i Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[4] Bai Li, Shiqi Wang, Suman Jana, i Lawrence Carin. Towards understanding fast adversarial training. *arXiv preprint arXiv:2006.03089*, 2020.

[5] Keiron O'Shea i Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[6] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, i Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019.

[7] Eric Wong, Leslie Rice, i J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

Algoritmi za brzo učenje na neprijateljskim primjerima

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: Ključne riječi, odvojene zarezima.

Algorithms for fast robust training on adversarial examples

Abstract

Abstract.

Keywords: Keywords.