Prepoznavanje saobraćajnih znakova koristeći CNN

Seminarski rad u okviru kursa
Računarska inteligencija
Matematički fakultet

Jana Jovičić ???/2015, Jovana Pejkić 435/2016
jana.jovicic755@gmail.com, jov4ana@gmail.com

16. maj 2019.

Sažetak

Sadržaj

1 Uvod

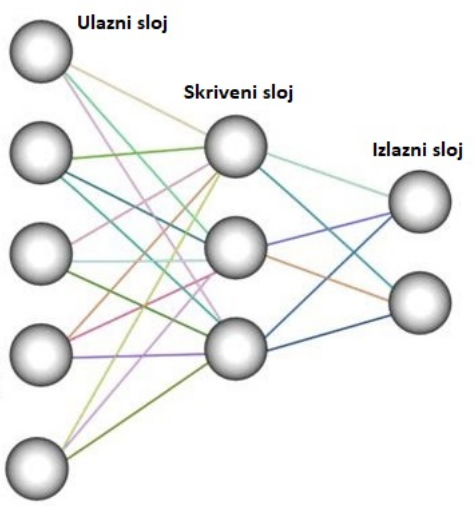
tekst tekst

2 Neuronske mreže

Neuronske mreže predstavljaju skup statističkih modela učenja inspirisana biološkim neuronima, za rešavanje klasifikacionih[1] i regresijskih problema[2]. Njihove primene su mnogobrojne, a neke od njih su: kategorijzacija teksta, raspoznavanje i sinteza govora, autonomna voznja, igranje video igara, masinsko prevodenje prirodnih jezika i slicno. Kljucna prednost neuronskih mreza je da same mogu da konstruisu nove atribute nad sirovom reprezentacijom podataka[3] i odlicno baratanje sa velikim kolicinama podataka.

2.1 Arhitektura

Osnovnu jedinicu gradje neuronske mreze predstavljaju neuroni, koji su medjusobno povezani vezama sa tezinama koje se podesavaju tokom ucenja mreze. Povezani neuroni prosledjuju signale jedni drugima, a organizovani su u slojeve. Najjednostavniji oblik neuronske mreze je perceptron, koji sadrzi samo jedan ulazni i jedan izlazni sloj. Medjutim, kako on sluzi samo za ucenje linearnih modela, a u praksi se javlja potreba da i slozeniji modeli mogu da se nauce, osim perceptrona, koriste se viseslojne neuronske mreze. Viseslojna neuronska mreza osim ulaznog i izlaznog sloja, ima jedan ili vise skrivenih slojeva (slika 1). Kako se neuronske mreze uglavnom koriste za klasifikaciju uzorka u razlicite kategorije, ulazni sloj se sastoji od onoliko neurona kolika je dimenzionalnost ulaznog prostora, a broj neurona na izlazu jednak je broju klasa. Samo ucenje neuronske mreze je zapravo podesavanje tezina sve dok se ne dobije zadovoljavajuca aproksimacija izmedju ulaznih i izlaznih velicina.



Slika 1: Višeslojna neuronska mreža sa jednim skrivenim slojem

[1] Klasifikacioni problem - ako je izlazna promenljiva kategorickog tipa, npr. ,,zdrav"i ,,bolestan".

[2] Regresioni problem - ako je izlazna promenjiva neprekidnog tipa, npr. ,,plata"ili ,,tezina".

[3] Iako se nekad mogu pretpostaviti koji su atributi najinformativniji za predvidanje ciljne promenljive, izbori tih atributa su neretko losiji od onoga sto bi algoritam ucenja mogao da detektuje u sirovoj informaciji.

2.2 Razlog uvodjenja CNN

Jednostavni zadaci prepoznavanja mogu se dosta dobro resiti skupovima podataka malih velicina, na primer desetine hiljada slika. Medjutim, objekti u realisticnim postavkama pokazuju znacajnu varijabilnost, stoga, da bi bilo moguce nauciti prepoznati ih, potrebno je koristiti mnogo vece skupove za treniranje. Da bismo naucili o hiljadama objekata iz miliona slika, potreban nam je model sa velikim kapacitetom ucenja. Konvolutivne neuronske mreze (CNN), koje ce biti detaljno obradjene u ostatku rada, cine jednu takvu klasu modela. Njihov kapacitet se moze kontrolisati variranjem njihove dubine i sirine, a oni takodje daju jake i uglavnom ispravne pretpostavke o prirodi slika. Tako, u poredjenju sa standardnim (feedforward) neuronskim mrezama sa slojevima slicnih velicina, CNN imaju mnogo manje veza i parametara, tako da ih je lakse trenirati, dok je njihov teoretski najbolji ucinak verovatno samo nesto losiji.

3 Konvolutivne neuronske mreže

Konvolutivne neuronske mreze (eng. Convolutional neural network, CNN) predstavljaju podklasu neuronskih mreza koja ima najmanje jedan konvolutivni sloj (a moze ih imati i vise). Ova vrsta neuronskih mreza je inspirisana vizuelnim korteksom. Svaki put kada nešto vidimo, aktivira se niz slojeva neurona, i svaki sloj otkriva skup karakteristika kao što su linije, ivice itd. Visi nivoi slojeva otkritivaju složenije karakteristike kako bi prepoznali ono što smo videli. Konvolutivne mreze rade po istom principu i prakticno su uvek duboke neuronske mreze, upravo zbog toga sto je potrebno od sitnijih detalja, poput uspravnih, kosih i horizontalnih linija, koji obicno bivaju detektovani u nizim slojevima mreze, konstruisati slozenije oblike poput delova lica. Konvolutivne neuronske mreze se koriste u obradi signala (slike, zvuka), ali i teksta. U odnosu na ostale vrste neuronskih mreza, isticu se u prikupljanju lokalnih informacija (na primer o susednim pikselima na slici ili ,,okruzujucim"(eng. surrounding) recima u tekstu) i smanjenju slozenosti modela (brze treniranje, potrebno je manje izoraka, manja sansa da dodje do preprilagodjavanja (eng. overfitting)). Konvulativne neuronske mreze se zasnivaju na sposobnosti mreza da iz sirovog signala konstruisu atribute. Nazivaju se konvolutivnim zato sto uce **filtere** (pojam objasnjen u tabeli 1), cijom konvolutivnom primenom detektuju odredjena svojstva signala. U narednoj tabeli su opisani neki od bitnijih parametara koje treba podesiti pri ucenju mreze.

tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst tekst tekst

3.1 Parametri

U ovoj sekciji je dat tabelarni prikaz parametara koji su najznacajniji za implementaciju konvolutivne mreze. U tabeli 1 je dat samo kratak opis svakog od njih radi boljeg razumevanja teksta koji sledi. Ipak, u nastavku je svaki detaljnije opisan.

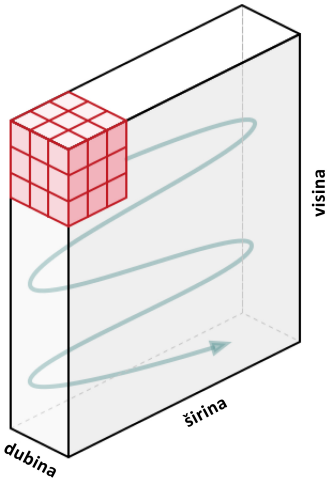
Tabela 1: Primer tabele

Filter (jezgro, kernel)	- matrica sa tezinama za konvoluciju ulaza - daje meru koliko deo ulaza liči na karakteristiku - tezine u matricama filtera su izvedene za vreme treniranja podataka
Padding	- koristi se za dodavanje kolona i redova nula da bi se odrzala konstantna velicina matrice (mape) nakon konvolucije - ovaj parametar moze da unapredi performanse tako sto zadrzava informacije u okvirima
Stride	- broj piksela koji želite preskočiti, dok prelazite ulaz vodoravno i uspravno tokom konvolucije, nakon mnozenja svakog elementa iz ulazne matrice težina s onima u filtru
Number of Channels	- It is the equal to the number of color channels for the input but in later stages is equal to the number of filters we use for the convolution operation. - The more the number of channels, more the number of filters used, more are the features learnt, and more is the chances to over-fit and vice-versa.
Pooling-layer Parameters	- ima iste parametre kao i konvolutivni sloj - uglavnom se koristi Max-Pooling opcija - The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality by keeping the max value(activated features) in the sub-regions binned.

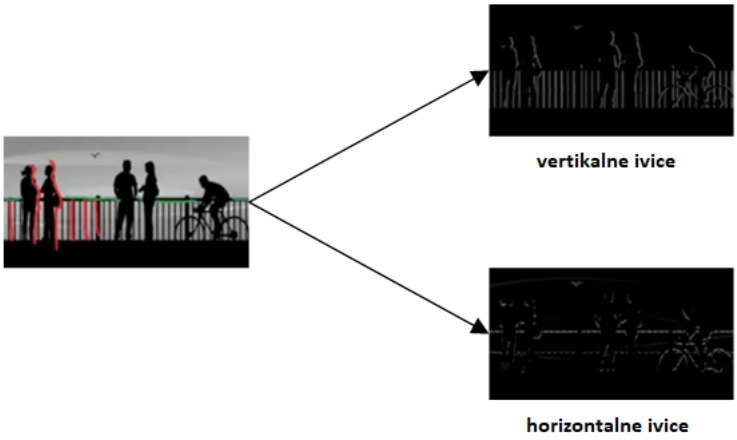
3.2 Konvolucija slika preko CNN

Da bi izvrsile klasifikaciju slika, konvolutivne mreze (preciznije, konvolutivni sloj, detaljnije u poglavlju ?????) obavljaju neku vrstu pretrage. Ovo se moze zamisliti kao mali pokretni (ili klizni) prozor (prikazano na

slici 2) koji klizi s leva na desno preko vece slike, i nastavlja s leve strane kada dodje do kraja jednog prelaza (kao kod pisace masine). Taj pokretni (klizni) prozor - koji ustvari predstavlja filter, moze da prepozna samo jednu stvar, recimo kratku vertikalnu liniju (tri tamna piksela naslagana jedan na drugi). Slicno, neki drugi filter moze da sluzi za prepoznavanje horizontalnih linija, i on se takodje pomera preko piksela slike, trazeci podudaranja. Rezultat koji se postize filterima koji prepoznaju vertikalne i horizontalne linijije prikazan je na slici 3.



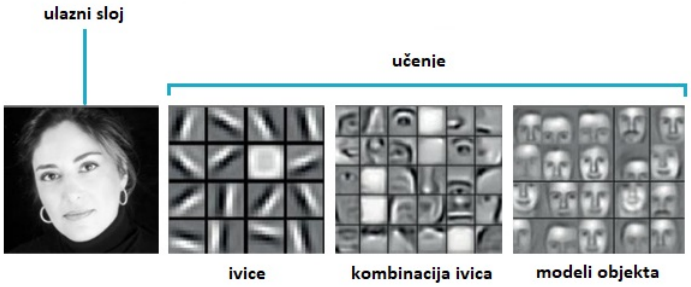
Slika 2: Kretanje filtera



Slika 3: Detektovanje ivica

Svaki put kada dodje do poklapanja (filtera sa ulazom), ono se mapira u prostor sa karakteristikama - koji se zove **mapa karakteristika** (eng. feature maps), koji je specifican za taj vizuelni element. U tom prostoru (tj. mapi) se cuva (odnosno belezi) lokacija svakog poklapanja

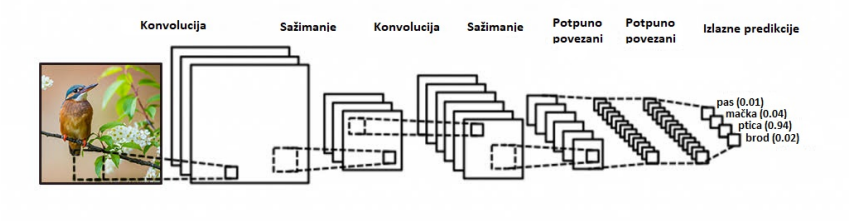
sa vertikalnom linijom. Konvolutivna mreza pokrece mnogo pretraga nad jednom istom slikom. Na pocetnom sloju mreze koriste se filteri koji prepoznaju horizontalnu liniju, vertikalnu liniju i dijagonalnu liniju, da bi kreirali mapu ivica slike. U narednim koracima (odnosno slojevima) posmatraju se kombinacije ovih ivica i tako prepoznaju slozeniji oblici. Ovo je demonstrirano na slici 4.



Slika 4: Učenje mreže

3.3 Unutrasnja struktura CNN

Unutrasnja struktura konvolutivne mreže se sastoji od nekoliko naizmeničnih konvolutivnih slojeva (eng. convolution layer) i slojeva agregacije (eng. pooling layer), pri cemu je dozvoljeno pojavljivanje iste vrste sloja vise puta (prikazano na slici 5). U dosad opisanoj strukturi neurona (neuronskih mreza) izlaz iz svakog od njih je bio skalarna veličina. Izlazi konvolutivnog sloja su dvodimenzionalni i nazivaju se mapama karakteristika (eng. feature maps). Oni se transformisu nelinearnom aktivacionom funkcijom (na primer tanh), koja ce prevesti ulazne vrednosti u opseg izmedju -1 i 1.



Slika 5: Arhitektura konvolutivne neuronske mreže

3.4 Konvolucija

Konvolutivni sloj je glavni deo konvolutivne neuronske mreze koji radi najvise izracunavanja u mrezi. Njegova uloga je konstrukcija novih atributa. To je prvi sloj koji ekstraktuje karakteristike iz ulazne slike. Konvolucija je matematicka operacija koja uzima dva ulaza - dve matrice f i

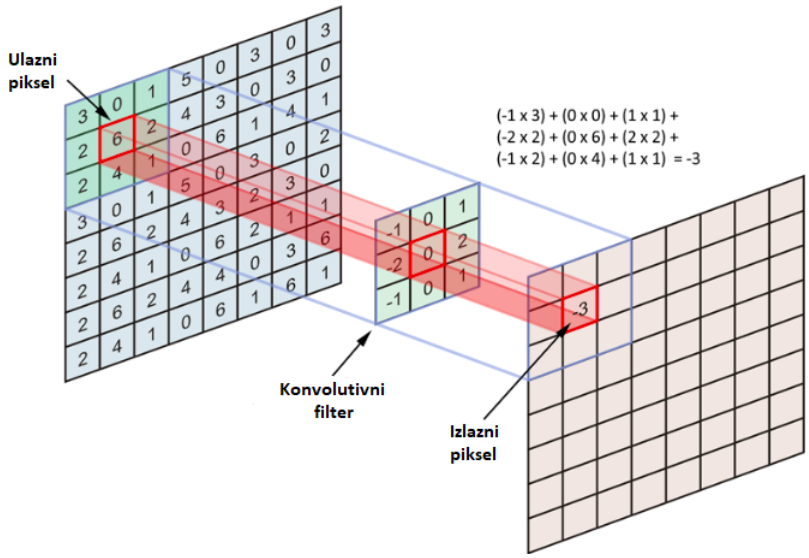
g, dimenzija m x n i p x q, a definisana je na sledeci nacin:

$$(f * g)_{ij} = \sum_{k=0}^{p-1} \sum_{l=0}^{q-1} f_{i-k,i-l} g_{k,l}$$

Matrica f je obicno ulaz, poput slike, dok je matrica g filter. Određena ulazna reprezentacija podatka (originalna slika) konvolvira se sa filterom sa određenim parametrima (ti parametri predstavljaju i parametre konvolutivne mreze). Procesom učenja ovi parametri (težine koje je potrebno nauciti kako bi mreža davala dobre rezultate) se podešavaju i bivaju nauceni. Filteri su najcešce manjih prostornih dimenzija od ulaza, ali uvek su jednake dubine kao i ulaz. Kao sto je vec receno, konvolucijska jezgra (filteri) filtriraju sliku, tj. mapu karakteristika kako bi izlučila neku korisnu informaciju kao što je recimo određeni oblik, boja ili ivica.

Tokom prvog prolaza svaki filter se pomera po širini i visini i računa se skalarni proizvod ulaza i vrednosti filtera (prikazano u formuli iznad). Izlaz jednog filtera će biti dvodimenzioni niz. Ako imamo više filtera, izlaz iz konvolucionog sloja će biti rezultati svakog filtera poredjanih po dubini. Operacijom konvolucije dobija se transformirana slika dimenzije I*K, gde je I dimenzija ulazne matrice slike, a K je dimenzija filtera koji je primenjen nad tom slikom. Takodje, dobijena transformacija je lokalna tj. pikseli izlaza zavise od lokalnih, susednih piksela ulaza. Ceo ovaj proces je prikazan na slici 6.

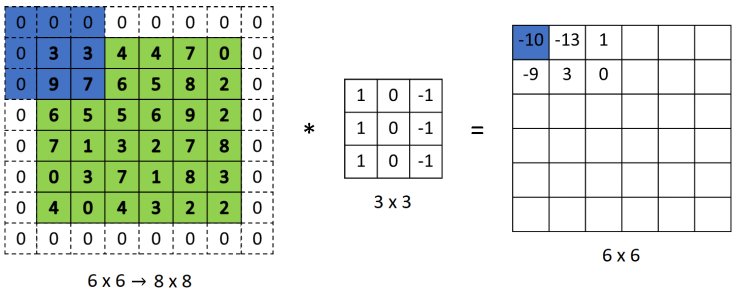
Slika 6: Konvolucija primenom filtera dimenzije 3x3 na matricu dimenzije 8x8



3.4.1 Prosirivanje

Formula konvolucije koja je data u poglavlju 3.4 nije definisana za sve indekse i=0, m - 1 i j=0, n - 1. Na primer, ako je i, j = 0 i k, l ¿ 0, vrednost fi - k,i - l nije definisana. Ukoliko bi se u obzir uzele samo definisane vrednosti, dimenzija konvolucije bi bila manja od dimenzije matrice f. Medjutim, to nije uvek pozeljno, i moze se izbeci tako sto se vrsi

prosirivanje (eng. padding) matrice f, na primer nulama ili vrednostima koje su vec na obodu, tako da velicina rezultujuce matrice bude jednaka velicini matrice f pre prosirivanja. Ovo je prikazano na slici 7. Takode, prilikom racunanja konvolucije, filter se duz slike ne mora pomerati za jedan piksel, vec za neki veci **korak** (eng. stride).



Slika 7: Primer prosirivanja

3.5 Agregacija

Uloga sloja za agregaciju je smanjenje broja parametara kada su slike prevelike, kao i smanjenje broja racunskih operacija u visim slojevima smanjuje. Agregacija smanjuje dimenzionalnost svake mape, ali zadržava važne informacije. Sve to rezultuje smanjenjem racunske zahtevnosti pri optimizaciji i pomaze u kontroli overfitinga. Zato zelimo da slojevi za agregaciju prate konvolucione slojeve kako bismo postepeno smanjili prostornu veličinu (širinu i visinu) prikaza podataka.

Cesto nas konačni zadatak postavlja neko globalno pitanje o slici, npr., Da li sadrži mačku? Tako da čvorovi našeg zadnjeg sloja moraju biti osetljivi na celi ulaz. Postepenom agregacijom informacija, stvarajuci grublje i grublje mape karakteristika, postize se taj cilj da se na kraju nauci globalna reprezentacija, zadrzavajuci sve prednosti konvolucijskih slojeva na srednjim slojevima obrade.

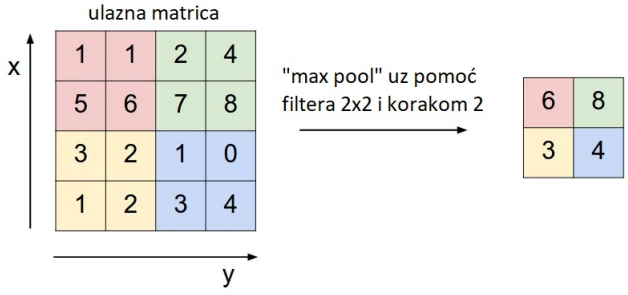
Sloj agregacije ukrupnjuje informacije, tako sto racuna neku jednostavnu funkciju agregacije susednih jedinica prethodnog sloja, poput maksimuma (eng. Max pooling) [koji vraca maksimalnu vrednost dela slike pokrivene filterom] ili proseka (eng. Average pooling) [koji vraca prosecnu vrednost dela slike pokrivene filterom]. Ukoliko agregira, na primer, 3 x 3 piskela, onda je broj izlaza ovog sloja 9 puta manji od broja izlaza prethodnog. Kada se racuna maksimum, dolazi do zanemarivanje informacije o tome gde je precizno neko svojstvo (poput uspravne linije) pronadeno, ali se ne gubi informacija da je pronadeno. Ovakva vrsta zanemarivanja informacije cesto ne steti cilju koji treba postici. Na primer, ako su na slici pronadeni kljun i krila, informacija o tacnoj pozicjiji najverovatnije nije bitna za odlucivanje da li se na slici nalazi ptica. Ipak, ukoliko je potrebno napraviti mrezu koja igra igru u kojoj su pozicije objekata na ekranu bitne, nije pozeljno koristiti agregaciju.

Average pooling was often used historically but has recently fallen out of favor compared to the max pooling operation, which has been shown to work better in practice.

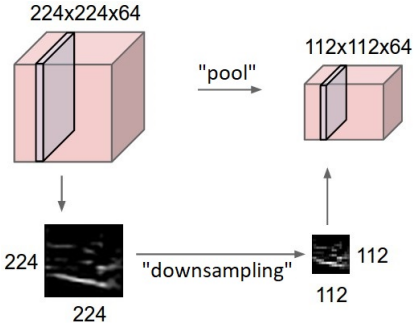
Pooling layer downsamples the volume spatially, independently in each depth slice of the input volume.

Prva slika (pool): In this example, the input volume of size [224x224x64] is pooled with filter size 2, stride 2 into output volume of size [112x112x64]. Notice that the volume depth is preserved.

Druga slika (maxpool): The most common downsampling operation is max, giving rise to max pooling, here shown with a stride of 2. That is, each max is taken over 4 numbers (little 2x2 square).



Slika 8: Maxpool



Slika 9: Pool

4 Implementacija i eksperimentalni rezultati

Kod 1 demonstrira...

```
table[key] = value
```

Kod 1: Primer koda

5 Zaključak

A Dodatak

A.1 Podnaslov 1

Dodatna objasnjenja

By visualizing the output from different convolution layers in this manner, the most crucial thing that you will notice is that the layers that are deeper in the network visualize more training data specific features, while the earlier layers tend to visualize general patterns like edges, texture, background etc.

This knowledge is very important when you use Transfer Learning whereby you train some part of a pre-trained network (pre-trained on a different dataset, like ImageNet in this case) on a completely different dataset. The general idea is to freeze the weights of earlier layers, because they will anyways learn the general features, and to only train the weights of deeper layers because these are the layers which are actually recognizing your objects.

Convolved Feature, Activation Map or Feature Map is the output volume formed by sliding the filter over the image and computing the dot product. Perceptrons come first in 1950s, and it uses a brittle activation function to do classification, so if w*x is greater than some value it predicts positive, otherwise negative.

Neurons uses a softer activation function by introducing a sigmoid function, a tanh function or other activation functions to pass on values to other neurons in the network.

So perceptrons do not use in a network setting, they do classification on their own, hence they can't classify XOR, however neurons can because they all contribute forward to the final output, using more complicated structure(i.e. multiple layers in network), they are able to classify XOR and other complicated problems.

A CNN, in specific, has one or more layers of convolution units. A convolution unit receives its input from multiple units from the previous layer which together create a proximity. Therefore, the input units (that form a small neighborhood) share their weights.

The convolution units (as well as pooling units) are especially beneficial as:

They reduce the number of units in the network (since they are many-to-one mappings). This means, there are fewer parameters to learn which reduces the chance of overfitting as the model would be less complex than a fully connected network. They consider the context/shared information in the small neighborhoods. This future is very important in many applications such as image, video, text, and speech processing/mining as the neighboring inputs (eg pixels, frames, words, etc) usually carry related information.