

Prepoznavanje saobraćajnih znakova pomoću CNN

Jana Jovičić, Jovana Pejkić

Prezentacija seminarskog rada
u okviru kursa
Računarska inteligencija
Matematički fakultet

jana.jovicic755@gmail.com, jov4ana@gmail.com

Sadržaj

- 1 Cilj rada
- 2 Informacije o korišćenom skupu podataka
- 3 Modeli
- 4 Zaključak
- 5 Literatura

Cilj rada

- Za bazu podataka kineskih saobraćajnih znakova izvršiti što precizniju klasifikaciju
- Implementirati CNN u programskom jeziku Python uz korišćenje Keras biblioteke
- Isprobati nekoliko različitih arhitektura mreže
- Uporediti dobijene rezultate i izvesti zaključke

Informacije o korišćenom skupu podataka

- Baza sadrži **6164 slika** saobraćajnih znakova
 - podeljenih u **58 klasa**
 - pri čemu **trening skup** sadrži **4170 slika**
 - a **test skup** **1994 slika**
- Zbog nejednakog broja slika (negde 5, negde 400) po klasama, korišćen je deo baze
- Izdvojeno je **10 klasa** koje su imale približno jednak broj slika
- Dobijen je **trening skup** od **1693 slika** i **test skup** od **764 slika**

Unutrašnja struktura CNN

■ Konvolutivni sloj

- Operacija konvolucije:

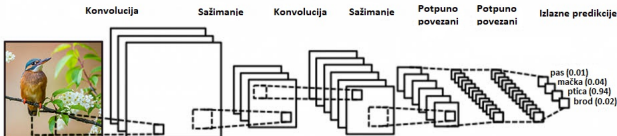
$$(f * g)_{ij} = \sum_{k=0}^{p-1} \sum_{l=0}^{q-1} f_{i-k, j-l} * g_{k,l}$$

■ Sloj agregacije

- Maxpooling / Averagepooling

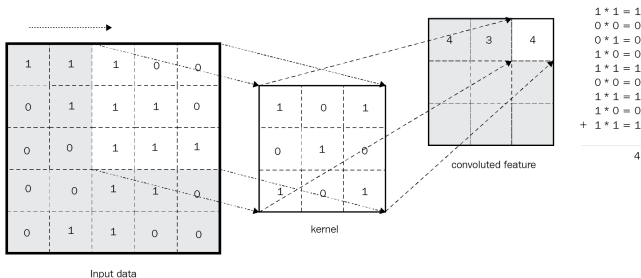
■ Potpuno povezani sloj

- Softmax funkcija
- Categorical crossentropy



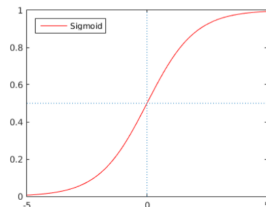
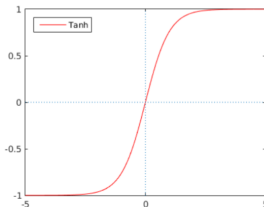
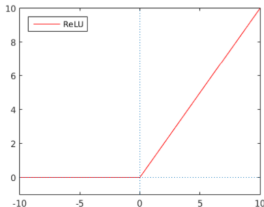
Filtriranje, proširivanje i korak

- Vrš se na konvolutivnom sloju
- **Filtriranje** je „ekstraktovanje“ karakteristika ulaza (slike)
 - tako što se izvršava operacija konvolucije
- Dimenzija mape nakon filtriranja je manja od dimenzije ulaza
- Vrš se **proširivanje** ulazne matrice
 - nulama / vrednostima koje su već na obodu
- **Korak** - za koliko piksela se filter pomera duž slike



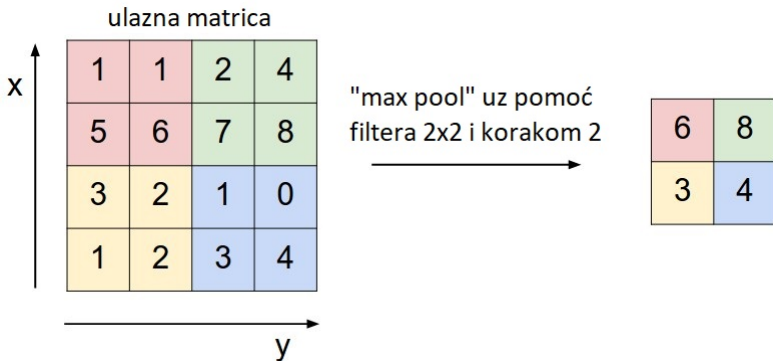
Aktivaciona funkcija

- Funkcija kojom se ograničavaju vrednosti izlaza neurona
 - opseg izlaza neurona obično je u intervalu $[0,1]$ ili $[-1,1]$ (osim za ReLU)
- Više vrsta aktivacionih funkcija:
 - **ReLU** (Rectified Linear Unit): $f(z) = \max(0, z)$
 - **tanh**: $\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$
 - **sigmoid**: $\sigma(x) = \frac{1}{1 + e^{-x}}$



Agregacija metodom Maxpool

- Jedan od metoda koji se koristi na sloju za agregaciju, najzastupljeniji
- Vraća **maksimum** dela slike prekrivene filterom
- Ideja je da se informacije o slici što više „ukrupne“



Potpuno povezani sloj

- Svaki neuron je povezan sa svakim neuronom iz prethodnog sloja
- Kao ulaz očekuje vektor, što znači da je prvo potrebno ispraviti mapu karakteristika u vektor
- Poslednji potpuno povezani sloj treba da ima onoliko neurona koliko ima i klasa
- Poslednji sloj određuje koju klasu da pridruži slici pomoću softmax aktivacione funkcije
- Softmax funkcija određuje verovatnoću pripadnosti slike svakoj od klasa
- Slika se klasifikuje u onu klasu kojoj odgovara najveća verovatnoća pripadnosti

Potpuno povezani sloj - Funkcija gubitka

- Funkcija gubitka: categorical crossentropy
- Upoređuje raspodelu verovatnoća predviđenih klasa sa raspodelom koja odgovara pravim klasama

■

$$H(y, \hat{y}) = - \sum_i y_i \cdot \log(\hat{y}_i)$$

- Služi da kvantifikuje kvalitet klasifikacije, tj. da oceni koliko je dobar skup težina koji služi za učenje mreže

Optimizacija mreže

- Cilj optimizacije: pronaći skup težina tako da se funkcija gubitka minimizuje
- SGD (Stohastički Gradijentni Spust)
 - Težine se traže u pravcu koji je suprotan pravcu gradijenta funkcije gubitka - tako se može stići do lokalnog minimuma
 - Potrebno odrediti pogodnu dužinu koraka, tj. pronaći najbolji parametar učenja (learning rate)
 - Ažuriranje težina moguće na osnovu samo dela podataka iz trening skupa
 - Stohastički - nasumično bira trening uzorke na osnovu kojih će ažurirati parametre
 - Održava jednu vrednost parametra učenja za sva ažuriranja težina i ta vrednost se ne menja tokom treninga
- Adam
 - Unapređena verzija SGD-a
 - Može da održava različite parametre učenja za različite težine i da ih menja tokom procesa učenja

Model 1

- Jedan od prvih modela koji je imao uspeha nad test podacima
- Sastoji se iz:
 - 4 konvolutivna sloja
 - 2 agregirajuća sloja
 - 2 potpuno povezana sloja
- U svim konvolutivnim slojevima:
 - velicina jezgra je 3×3
 - broj filtera u konvolutivnom sloju je 32
- U svakom sloju se koristi **ReLU** aktivaciona funkcija
- Agregacija se vrši biranjem **maksimalne vrednosti** dela mape karakteristika koji je prekriven filterom

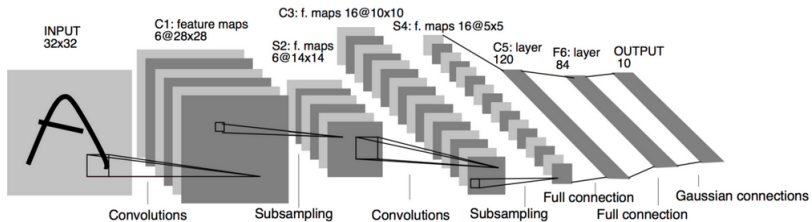
Model 1 nastavak

- Funkcijom **Dropout()** je isključivan određen broj nasumično odabranih neurona (da bi se sprečilo preprilagodjavanje)
- Nakon agregacija je isključeno 20% neurona, pre FC sloja 50%
- Poslednji potpuno povezani (FC) sloj
 - ima onoliko neurona koliko ima klasa
 - koristi softmax aktivacionu funkciju
- Učenje modela je sprovedeno u 30 epoha
- Batch size je postavljen na 32
 - sto znaci da u svakoj iteraciji uzima 32 primerka iz trening skupa koja ce biti propagirana kroz mrežu
- Optimizacija modela je izvršena pomocu **SGD**
- Sa tako podesenim parametrima preciznost je 0.88

Model 2: LeNet5

- 7 slojeva: 2 konvolutivna, 2 average pooling, 3 potpuno povezana
- Prvi konvolutivni sloj: 6 filtera veličine 3x3
- Drugi konvolutivni sloj: 16 filtera veličine 3x3
- ReLU aktivaciona funkcija
- Potpuno povezani slojevi - broj neurona redom: 120, 84, koliko ima i klasa
- Pre treninga su slike skalirane na 32x32
- Optimizacija pomoću Adam optimizatora
- Prilikom učenja u 30 epoha sa batch size 32, preciznost je bila 0.91

Model 2 - nastavak



Slika: LeNet5 arhitektura

Modeli 1 - statistike

Model 1, 10 epoha

Funkcija aktivacije	optimizator	Batch size	preciznost na test skupu	vreme iz-vrsavanja
relu	adam	32	0.848	233.333
relu	adam	64	0.767	218.867
relu	sgd	32	0.879	213.162
relu	sgd	64	0.845	257.118
sigmoid	adam	32	0.230	255.351
sigmoid	adam	64	0.230	243.452
sigmoid	sgd	32	0.230	252.360
sigmoid	sgd	64	0.230	265.278
tanh	adam	32	0.230	231.808
tanh	adam	64	0.110	228.060
tanh	sgd	32	0.908	223.549
tanh	sgd	64	0.879	228.474

Slika: Rezultati Modela 1 za 10 epoha, različite vrednosti funkcije aktivacije (relu, tanh i sigmoid), optimizatora (SGD, Adam) i Batch size (32, 64)

Modeli 2 - statistike

LeNet5, 10 epoha				
Funkcija aktivacije	optimizator	Batch size	preciznost na test skupu	vreme iz-vrsavanja
relu	adam	32	0.825	29.112
relu	adam	64	0.827	31.737
relu	adam	128	0.835	23.660
relu	sgd	32	0.830	27.591
relu	sgd	64	0.796	25.008
relu	sgd	128	0.487	22.703
sigmoid	adam	32	0.230	29.442
sigmoid	adam	64	0.230	25.430
sigmoid	adam	128	0.230	24.470
sigmoid	sgd	32	0.230	29.697
sigmoid	sgd	64	0.230	28.206
sigmoid	sgd	128	0.230	23.382
tanh	adam	32	0.866	29.580
tanh	adam	64	0.845	25.425
tanh	adam	128	0.872	24.269
tanh	sgd	32	0.843	29.354
tanh	sgd	64	0.817	26.013
tanh	sgd	128	0.772	23.323

Slika: Rezultati LeNet5 modela za 10 epoha, različite vrednosti funkcije aktivacije (relu, tanh i sigmoid), optimizatora (SGD, Adam) i Batch size (32, 64 i 128)

Modeli 2 - statistike (nastavak)

LeNet5, 100 epoha				
Funkcija aktivacije	optimizator	Batch size	preciznost na test skupu	vreme iz-vrsavanja
relu	adam	32	0.869	285.869
relu	adam	64	0.822	229.086
relu	adam	128	0.843	215.497
relu	sgd	32	0.796	255.521
relu	sgd	64	0.811	252.399
relu	sgd	128	0.730	221.506
sigmoid	adam	32	0.230	286.774
sigmoid	adam	64	0.230	249.128
sigmoid	adam	128	0.230	242.137
sigmoid	sgd	32	0.230	240.445
sigmoid	sgd	64	0.230	238.546
sigmoid	sgd	128	0.230	235.574
tanh	adam	32	0.887	274.488
tanh	adam	64	0.853	240.747
tanh	adam	128	0.866	212.645
tanh	sgd	32	0.816	274.816
tanh	sgd	64	0.780	235.054
tanh	sgd	128	0.780	219.117

Slika: Rezultati LeNet5 modela za 100 epoha, različite vrednosti funkcije aktivacije (relu, tanh i sigmoid), optimizatora (SGD, Adam) i Batch size (32, 64 i 128)

Modeli 1 i 2 - poređenje

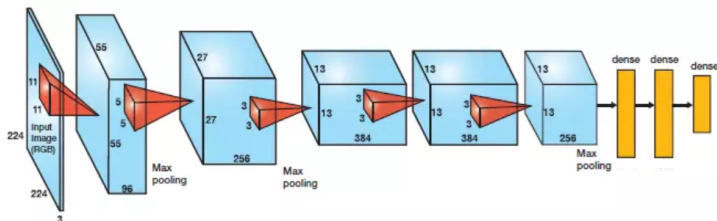
- Sigmoidnu funkciju ne treba koristiti ni u modelu 1 ni u LeNet modelu, jer sve slike klasifikuje u istu klasu (8. klasu)
- LeNet model je mnogo brži od modela 1: za približno isto vreme u modelu 1 se izvrši 10 epoha, a u LeNet modelu 100 epoha
- Kada se posmatra izvorsavanje u 10 epoha, model 1 daje malo bolje rezultate od LeNet (ako se izuzme kombinacija tangentne aktivacione funkcije i optimizacije pomocu Adama)
- Model 1 daje najveću preciznost (0.908) na test skupu u 10 epoha kada se kao aktivaciona funkcija koristi tangenta funkcija, kada se optimizuje pomoću SGD i kada je batch size = 32

Modeli 1 i 2 - poređenje (nastavak)

- Model 1 ima nešto veću preciznost kada je batch size = 32, u odnosu na to kada je batch size = 64
- Kada poredimo izvršavanje LeNet modela u 10 i 100 epoha, vidimo da je preciznost tek malo bolja u 100 epoha, nego u 10 - za 0.01 ili 0.02. A izvršavanje traje oko 10 puta duže. Zato se više isplati uzeti manji broj epoha.

Model 3: AlexNet

- AlexNet arhitektura je jedna od prvih dubokih mreža
- Sastoji se iz:
 - 5 konvolutivna sloja
 - 3 potpuno povezana sloja
- Kao aktivaciona funkcija koristi se ReLu



Slika: AlexNet arhitektura

Model 3: AlexNet - nastavak

- Mreža koju smo implementirale svaki izlaz iz konvolutivnog sloja normalizuje pre nego što ga prosledi narednom sloju
- Pre normalizacije (nakon 1., 2. i 5. konv. sloja) - agregacija
 - sa parametrom padding = 'valid' (nema proširenja)
- Poravnavajući sloj ("ispravlja" mapu karakteristika u vektor)
- 3 Dense sloja + Dropout()
 - Dropout() sprečava prilagodavanje modela
- Na kraju je izlazni sloj koji preslikava ulaz u zadati broj klasa
 - kao funkciju aktivacije koristi Softmax.

Model 3: AlexNet - statistike

- Kod AlexNet modela, vreme izvršavanja je oko 10 puta veće za 100 nego za 10 epoha
- Najmanja preciznost za AlexNet model se postiže za funkciju sigmoid, 10 epoha i Batch size 256, a iznosi 0.696
 - to je jedini slučaj da je preciznost ispod 0.71
 - uglavnom su preciznosti u intervalu [0.801, 0.934]
 - ne osciluje mnogo izvan pomenutog intervala
- Prosečna preciznost za 10 epoha je 82%, a za 100 epoha 90%
- Iz ovoga se može zaključiti da je za AlexNet model i dati skup podataka bolje trenirati model u što više epoha
- Ovo potvrđuje i činjenica da je u 1000 epoha ovaj model dostigao najveću preciznost od 95% (veću od sva tri modela)

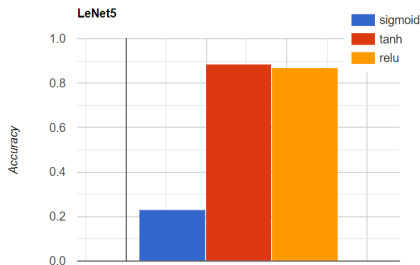
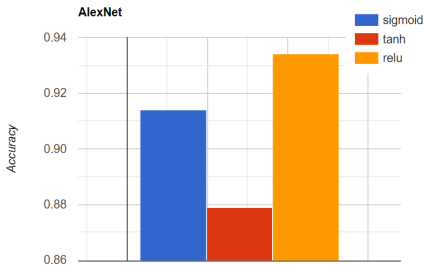
Model 3: AlexNet - statistike - nastavak

AlexNet					
Br. epoha	Funkcija aktivacije	Optimizator	Batch size	Preciznost na test skupu	Vreme izvršavanja (sekunde)
10	relu	adam	32	0.838	44.356
10	relu	adam	64	0.848	37.813
10	relu	adam	256	0.715	33.786
10	sigmoid	adam	32	0.919	43.755
10	sigmoid	adam	64	0.845	38.291
10	sigmoid	adam	256	0.696	36.811
10	tanh	adam	32	0.801	43.664
10	tanh	adam	64	0.906	39.264
10	tanh	adam	256	0.851	34.403
100	relu	adam	32	0.934	372.984
100	relu	adam	64	0.934	334.667
100	relu	adam	256	0.906	295.523
100	sigmoid	adam	32	0.914	372.971
100	sigmoid	adam	64	0.914	334.225
100	sigmoid	adam	256	0.856	292.743
100	tanh	adam	32	0.879	373.153
100	tanh	adam	64	0.921	339.918
100	tanh	adam	256	0.906	296.921

Slika: Rezultati za AlexNet model za različite vrednosti broja epoha (10 i 100), funkcije aktivacije (relu, tanh i sigmoid) i Batch size (32, 64 i 256)

AlexNet i LeNet-5

- Primetiti da se vrednosti na y-osi ova dva grafika razlikuju
- To je zato što LeNet-5 daje malu preciznost za f-ju sigmoid
- U 100 epoha maksimalna preciznost AlexNet mreže je 0.934, a LeNet-5 0.887



Slika: Grački prikaz preciznosti za mreže AlexNet i LeNet-5 za 100 epoha, za funkcije sigmoid, tanh i Relu.

Zaključak

- Za sva tri isprobana modela mreža moguće je naći parametre tako da dobro klasifikuju test podatke
- Od svega što smo probale, najbolje rezultate (preciznost od 0.95) je dala AlexNet mreža u 1000 epoha

Literatura



Mladen Nikolić, Anđelka Zečević

Mašinsko učenje, 2019.



Saad Albawi

Understanding of a Convolutional Neural Network, 2017.



Leon Bottou, Yann LeCun, Patric Haffner

Object recognition with Gradient-Based learning, 1998.