

Seminarski rad iz Računarske Inteligencije

# **Prepoznavanje cifara upotrebom neuronskih mreža**

Stefan Stevović 151/2015  
Đorđe Vučković 31/2015

# **SADRŽAJ:**

1. Uvod

2. Opis problema

3. Opis rešenja

3.1. Treniranje

3.2. Testiranje

4. Upoređivanje

5. Upotreba programa

6. Zaključak

7. Literatura

# UVOD

Seminarski rad se zasniva na problemu prepoznavanja cifara sa slika korišćenjem neuronskih mreža. Za izradu smo koristili MNIST bazu podataka, koja se sastoji od 60.000 slika za treniranje i 10.000 slika za testiranje preciznosti modela. Uz MNIST bazu podataka za pisanje samog projekta koristili smo programski jezik Python 2.7 kao i biblioteku Numpy koja nam je služila za brzu linearnu algebru. Što se tiče API-ja korišćen je KERAS koji predstavlja API za neuronske mreže visokog nivoa i koji je sposoban da radi na TensorFlow, Python biblioteci otvorenog koda koju je razvio Google Brain za potrebe deep learning-a. Za iscrtavanje slika korišćena je Python biblioteka matplotlib.

Neuronske mreže predstavljaju širok pojam i njihovo polje delovanja je jako široko, ovaj problem koji smo mi obradili je samo deo svega onog sto neuronskim mrežama može da se predstavi. Postoji više načina da se ovaj problem predstavi I reši ali svaki od njih ima skoro pa isti sastav uz neznatne izmene. Čitav posao se deli u nekoliko celina:

- 1) Odabir okruženja
- 2) Priprema baze podataka
- 3) Izgradnja mreže
- 4) Kompajliranje i trening podataka
- 5) Testiranje i ocenjivanje performansi modela

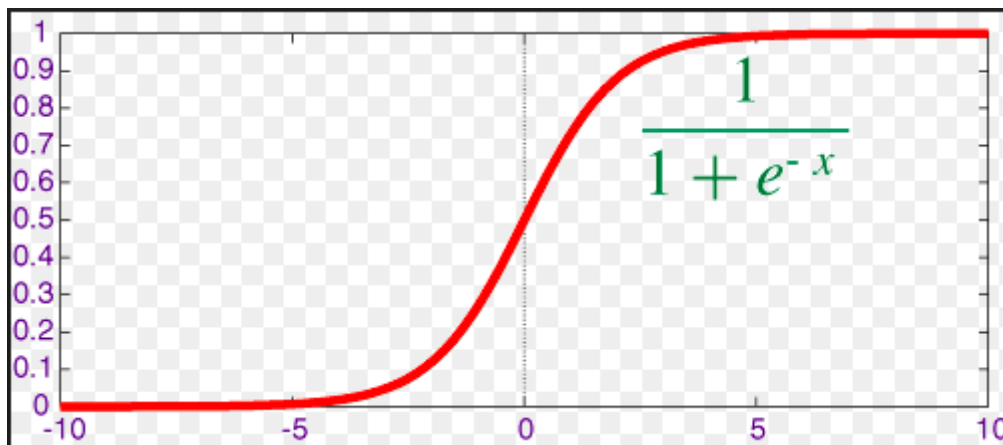
Ovih 5, gore navedenih stavki predstavlja kostur rešavanja ovog problema, postoje različitii načini da se svaki od ovih stavki implementira i izabere. I to predstavlja glavni zadatak svakog ko želi da se bavi ovim problemom.

## OPIS REŠENJA

Ovaj problem je razložen na dva dela. Prvi obuhvata proces treniranja mreže i čuvanja naučenog modela. Drugi deo obuhvata proces testiranja, odnosno učitavanja proizvoljne slike, njenu obradu i primenu modela na datu sliku.

### **Treniranje modela:**

Za treniranje modela smo koristili neuronsku mrežu sa 3 nivoa. Prvi nivo ima 128 čvorova. Drugi ima 50, a treći 10. Broj čvorova smo odredili eksperimentalno. Za aktivacionu funkciju smo koristili sigmoidnu funkciju.



Što se tiče dodatnih opcija kod treniranja modela, definisali smo da će se treniranje zaustaviti pre prolaska kroz sve iteracije(epohe) ukoliko je gubitak manji od 0.05.

Tokom svake epohe treniranja modela, u terminalu nam se ispisuje broj trenutne epohe, broj sekundi do kraja iteracije, vrednost funkcije gubitka i trenutna preciznost modela.

Nakon treniranja i čuvanja modela, primenili smo naš model na 10.000 slika koje se koriste za testiranje u MNIST bazi i ispisali preciznost treniranog modela.

### Testiranje modela:

Prilikom testiranja modela, prva stvar koju je potrebno uraditi jeste učitavanje samog modela kog smo sacuvali nakon procesa treniranja. Nakon toga sledi učitavanje slike na koju želimo da primenimo učitani model. Nakon učitavanja, sledi binarizacija, gde slike definišemo tako da cifre budu ispisane belom bojom na crnoj pozadini. Svaki piksel ima vrednost od 0 do 255, a naš prag za binarizaciju je 100. To znaci da ce svaki piksel koji ima vrednost manju od 100 biti obojen u crno, a svaki piksel koji ima vrednost veću od 100 biti obojen u belo.

Nakon binarizacije sledi promena veličine same slike jer naš model prima samo one slike koje su veličine 28x28 piksela.

## UPOREĐIVANJE MREŽA

Za određivanje najboljeg modela upoređivali smo preciznost modela nad test podacima iz MNIST baze podataka. Trenirali smo mreže koje su imale 2, 3 i 4 sloja sa promenljivim brojem čvorova u sloju. Broj epoha smo fiksirali na 50, jer smo primetili da se nezavisno od broja nivoa preciznost kod treniranja ne menja značajno nakon pedesete epohe (moguće je poboljšanje za najviše 0.05). U nastavku teksta smo definisali tabele samo za neke od najpreciznijih modela koje smo dobili:

	2 sloja	3 sloja	4 sloja
prvi sloj	50	128	130
drugi sloj	10	50	70
treći sloj		10	35
cetvrti sloj			10
preciznost	0.9687	0.9782	0.9773

Primetili smo to da za ovaj problem najveću preciznost imaju neuronske mreže sa 3 sloja. Prilikom svakog treniranja njihova preciznost je bila neznatno, ali uvek malo veća od preciznosti modela sa 4 sloja. Iz tog razloga smo više pažnje posvetili analizi mreža sa 3 sloja.

Na sledećoj tabeli smo definisali broj čvorova za svaki od slojeva i preciznosti definisanog modela kako na test podacima MNIST baze podataka, tako i na slikama koje smo mi sami definisali i koje se nalaze u direktorijumu “numbers”.

prvi sloj	128	50	25
drugi sloj	50	30	20
treći sloj	10	10	15
preciznost	0.9782	0.971	0.958
preciznost na slikama	17/20	17/20	16/20

Broj čvorova smo izabrali eksperimentalno. Sam program koji smo mi predstavili ima u prvom sloju 128 čvorova, u drugom 50 i u trećem sloju 10 čvorova, jer je kao takav vratio najbolju preciznost.

## UPOTREBA PROGRAMA

Da bismo kreirali naš model potrebno je pokrenuti program “training.py”. Nakon treniranja mreže dobićemo model “my\_model.h5”.

Mi smo definisali dva načina na koji je moguće testirati podatke. Prvi je unošenje putanje do slike koju je potrebno odraditi i to je standardni način testiranja algoritma koji bi korisnik trebao da koristi. Kada pokrenemo “test.py”, izlazi nam tekst koji nam kaže da unesemo putanju do slike. Nakon toga, prvo će nam se prikazati slika koju smo učitali. Izlaskom iz slike, učitana slika se dalje obrađuje i u terminalu nam se ispisuje predviđanje modela.

Drugi način je definisan specijalno za potrebe odbrane projekta. Taj drugi način podrazumeva obradu svih slika odjednom koji se nalaze u direktorijumu “numbers”. U suštini, ideja je potpuno ista kao kod testiranja pojedinačnih instanci (slika), samo što imamo jednu for petlju koja obilazi sve slike unutar foldera i primenjuje model na svaku od njih bez prikazivanja same slike radi uštede vremena.

## ZAKLJUČAK

Preciznost modela neuronske mreže zavisi od podešavanja parametara samog modela poput broja epoha, slojeva, čvorova u slojevima, funkcije gubitka itd. Model koji smo mi dobili nije najbolji mogući, jer se pronalaženjem odgovarajućih vrednosti ovih parametara može dobiti model koji je za nijasnu bolji od naseg, ali sa preciznošću od oko 0.97 naš model je sasvim dobar.

Što se tiče samog unapređivanja algoritma, korisno bi bilo doraditi program tako da prepozna više od jedne cifre na nekoj slici. Za to je, pored malo bolje istreniranog modela od ključne važnosti preprocesiranje samih podataka. Pod tim preprocesiranjem se najviše misli na primene različitih naprednih metoda za obradu slika.

## **LITERATURA**

[1] Neural networks, Christos Stergiou and Dimitrios Siganos

[2] Neural Networks and Deep Learning, Michael Nielsen

[3] Materijali sa predavanja profesora Aleksandra Kartelja, Matematički fakultet Beograd

[4] The MNIST database of handwritten digits, Yann LeCun and Corina Cortes and Christopher J.C. Burges