Završno izvješće

Original MNIST & Fashion-MNIST

Ivana Lubar, Katarina Mikulić, Kristina Udovičić

Prirodoslovno – matematički fakultet
Matematički odsjek

Sažetak: U ovom završnom izvješću predstavljeni su rezultati projekta u sklopu kolegija Strojno učenje na Prirodoslovno - matematičkom fakultetu u Zagrebu. Tema je klasifikacija slika iz dva skupa podataka od kojih svaki ima po točno deset klasa.
Ključne riječi: MNIST dataset, Fashion-MNIST, klasifikacija slika, kategorizacija, strojno učenje, neuronske mreže

1. Uvod

Klasifikacija slike je dobro istražen problem u kojem je slika identificirana kao pripadnost jednoj od već poznatih klasa. Istraživači nastoje izdvojiti određene značajke iz kojih mogu odrediti uzorke koji sadrže sliku. Algoritmi za određivanje tih bitnih značajki uključuju statističke metode kao što su sljedeće: grupiranje na osnovi centroida, povezivanje/grupiranje na temelju grafa, grupiranje temeljeno na distribuciji i grupiranje na temelju gustoće, kao i algoritmi učenja (linearna diskriminatorna analiza, vektorski strojevi za podršku, neuronske mreže).

Segmentacija i klasifikacija objekata na slikama je jedan od najvažnijih, ali i najsloženijih problema u računalnom vidu. Složenost problema je naročito izražena kod slika prirodnih scena gdje postoji velika varijacija samog izgleda objekata zbog različite perspektive, osvjetljenja i tipično velikog broja djelomično vidljivih objekata. Ljudski vid se dobro snalazi u scenama prirodnog okoliša jer se prirodno oslanja na strategije koje se oslanjaju na više različitih izvora informacija, na prethodna iskustva i naučena pravila fizičkog svijeta. Zbog toga se u području računalnog vida često pokušavaju replicirati strategije korištene kod ljudskog vida kako bi se poboljšala uspješnost sustava na tipičnim zadacima detekcije, lokacije i klasifikacije objekata na slici. Jedan od važnijih izvora informacija je kontekst slike koji je složen od više razina i izvora informacija dobivenih iz scene ili izvan scene. Zbog heterogenosti izvora kontekstualnih informacija postoji veliki broj pristupa koji za klasifikaciju objekata na slici integriraju dio kontekstualnih informacija i informacije dobivene iz izgleda objekta.

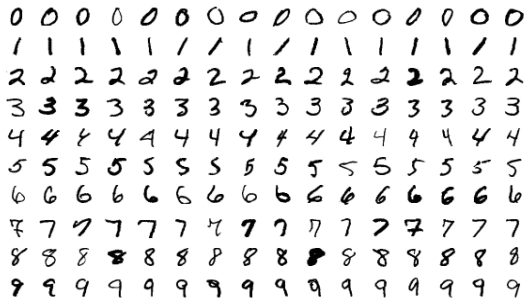
2. Opis skupova podataka i problema

Skupovi podataka koje koristimo su već poznati skupovi podataka nad kojima su već rađena mnoga istraživanja. Radi se o

MNIST (*engl. Modified National Institute of Standards and Technology*) skupovima podataka od kojih je jedan originalni dataset koji se sastoji od 70 000 slika ručno ispisanih znamenki, a drugi dataset – Fashion-MNIST – je skup slika koje predstavljaju odjevne predmete. Oba skupa podataka podijeljena su na skup za treniranje koji sadrži 60 000 slika i na skup za testiranje koji sadrži 10 000 slika. Skupovi za treniranje i skupovi za testiranje opisani su CSV datotekama koje se sastoje od 785 stupaca. Prvi stupac, naziva label, označava znamenkom kojoj klasi slika pripada. Ćelije ostalih stupaca označavaju vrijednost piksela pripadne slike.

2.1. MNIST skup podataka

Originalni MINST dataset se sastoji od slika ručno napisanih znamenaka.



Slika 1 Originalni MNIST dataset

Svaka slika ima visinu i širinu od 28 piksela što daje 784 piksela ukupno. Svaki piksel ima jedinstvenu vrijednost piksela koja je povezana s njim i ukazuje na svjetlinu tog piksela. Veća vrijednost piksela znači tamniji piksel. Vrijednost piksela je cijeli broj između 0 i 255.

Trenirani i testni primjeci imaju 785 kolona. Prva kolona sadrži ime klase kojoj primjerak pripada i označava neki odjevni predmet. Ostale kolone označavaju vrijednosti piksela pripadne slike.

Za lociranj piksela na slici, pretpostavimo da imamo vrijednost x kao

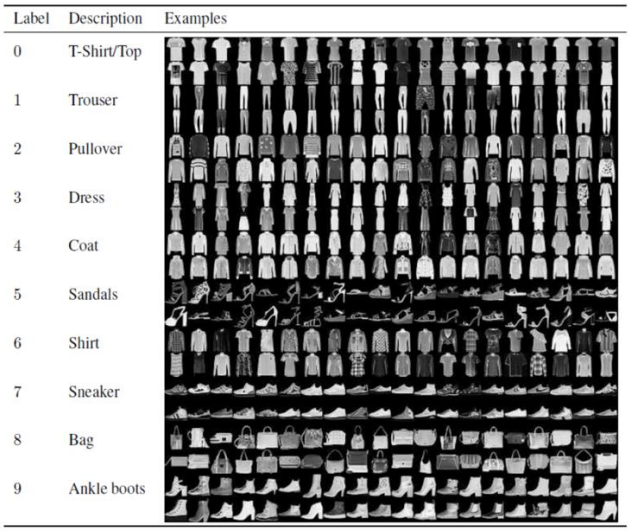
$$x = i * 28 + j$$

gdje su i i j cijeli brojevi između 0 i 27. Piksel je smješten u red i i stupac j matrice 28 x 28.

Na primjer, piksel 31 ukazuje na piksel koji je u četvrtom redu slijeva i drugom stupcu.

2.2. Fashion-MNIST skup podataka

Oznake prve kolone koje su ujedno klase našeg dataseta. Fashion-MNIST skup podataka je skup slika Zalandova članka koji se sastoji od treniranog seta od 60 000 primjeraka i testnog seta od 10 000 primjeraka. Svaki primjerak je crno-bijela slika dimenzija 28 x 28 povezana s nazivom jedne od 10 klasa.



Slika 2 Fashion-MNIST dataset

Zadatak je klasificirati slike iz testnog skupa tako da svaka pripada ispravnoj klasi.

3. Dosadašnja istraživanja

Općenito, kod algoritama za binarno klasificiranje, npr. kod prepoznavanja smije li se osoba ili ne, prepoznavanje lica općenito, za sliku koja se želi klasificirati se

navode ograničenja – mora biti jako osvjetljenje, slika ne smije biti mutna itd., dok su podaci za treniranje dosta slični – sve osobe su slikane sa svijetlom pozadinom iza, iz istog kuta, zauzimajući isti postotak slike, slika je crno-bijela, veličina slike (širina i visina) je ograničena. Kod ovakvih podataka često može doći do *overfit*-anja – jer su podaci za treniranje slični, a ima ih jako puno.

S obzirom na to da je problem klasificiranja aktualan, probao se riješiti raznim pristupima – koristeći logističku regresiju, stroj s potpornim vektorima, neuralne mreže, slučajne šume i tako dalje. Kod algoritama za klasificiranje u više kategorija, koriste se novije i složenije metode – konvolucijske neuralne mreže te kombinaciju stroja s potpornim vektorima, konvolucijskih neuralnih mreža i *transfer learning*-a.

S obzirom na razne algoritme, javljaju se i razni nedostaci – npr. konvolucijske neuralne mreže zahtijevaju jake računalne resurse.

4. Postupak izrade

Programski jezik koji smo koristili za izradu projektnog zadatka je Python. Napisani kod izvršavan je u Jupyter Notebooku. Potrebne biblioteke koje smo importali za analizu skupova podataka su:

- **pandas** – za manipulaciju podataka pomoću dataframes
- **numpy** – za statističku analizu podataka
- **matplotlib.pyplot** – za vizualizaciju podataka
- **seaborn** – za vizualizaciju podataka na temelju matplotliba. Pruža sučelje visoke razine za crtanje atraktivne i informativne statističke grafike.
- **random** – za nasumičan izbor iz odabranog skupa

4.1. Analiza podataka

Kreirali smo dataframeove za train i test skupove podataka na način:

```
train_df = pd.read_csv(lokacija_na_kojoj_se_nalazi_datoteka, sep=',')
test_df = pd.read_csv(lokacija_na_kojoj_se_nalazi_datoteka, sep = ',')
```

Nakon što smo učitali skupove podataka, bilo je važno provjeriti s kakvim podacima upravljamo.
Naredbom:

```
train_df.groupby('label').size()
```

možemo prijmjetiti da u Fashion-MNIST skupu imamo ravnomjerno raspoređen popis slika po klasama tj. u svakoj klasi se nalazi po 6000 slika dok su u originalnom MNIST skupu podataka slike raspoređene neravnomjerno - to bi moglo utjecati na rezultate testiranja.

```
label
0    5923
1    6742
2    5958
3    6131
4    5842
5    5421
6    5918
7    6265
8    5851
9    5949
dtype: int64
```

Slika 3 Kardinalnost klasa u MNIST skupu podataka

4.2. Kreiranje modela

Izgradnja konvolucijske neuronske mreže (engl. convolutional neural network - CNN), zahtijeva konfiguriranje slojeva modela, a zatim sastavljanje modela. Za kreiranje modela smo koristili biblioteku TensorFlow. Ova biblioteka olakšava proces stjecanja podataka, obučavanja modela, predviđanja i unapređivanja budućih rezultata.

Danas postoji mnogo izbora za CNN arhitekturu no pitanje je kako odabrati najbolju. Najbolja arhitektura bi bila možda najjednostavnija ili pak najučinkovitija za proizvodnju točnosti uz minimizaciju složenosti računanja. Za naše skupove podataka ćemo napraviti eksperimente kako bismo pronašli najučinkovitiju i najpreciznu CNN arhitekturu za klasifikaciju slika znamenki iz MNIST skupa te za klasifikaciju slika odjeće iz Fashion-MNIST skupa.

4.3. Osnovna struktura CNNa

Koristimo tipičan CNN, dakle dizajn započinje ekstrakcijom značajki i završava klasifikacijom. Ekstrakcija značajki provodi se naizmjeničnim konvolucijskim slojevima s podslojevima.
Konvolucijski sloj dodajemo na način:

```
model.add(Conv2D(filters=48,kernel_size=5,strides=1,padding='same',activation='relu'))
```

gdje su redom:

- feature – broj karata značajki (*engl. feature maps*)
- kernel_size – veličina konvolucijske jezgre. Broj 5 znači konvolucija 5 x 5
- strides – broj karata značajki idućeg sloja, imat će veličinu jednaku veličini prethodne karte značajki podijeljene sa strides. Defaultna vrijednost je 1.
- padding – postavlja se na vrijednost *same* ili *valid.* Defaultna vrijednost je *valid.* Ako je padding postavljen na valid, onda se veličina novog sloja umanjuje za kernel_size – 1. Ako je padding postavljen na same, veličina se ne smanjuje.
- activation – primjenjuje se prilikom širenja prema naprijed.

Notacija koju ćemo dalje koristiti:

- 24C5 - konvolucijski sloj s 24 karakteristične karte (*engl. feature maps*) filtra 5x5 i strides 1
- 24C5S2 - konvolucijski sloj s 24 karakteristične karte (engl. feature maps) veličine 5x5 i strides 2
- P2 - maksimalno udruživanje pomoću 2x2 filtra i strides 2
- 256 - potpuno povezani gusti sloj s 256 jedinica

Prvo pitanje koje nam se postavlja je, koliko podskupinskih slojeva koristiti.

4.4. Testiranje broja slojeva

Prva provjera koju provodimo je testiranje koliko slojeva koristiti. Provjere koje radimo su za jedan, dva ili tri para konvolucijskih slojeva. Ne radimo četiri para konvolucijskih slojeva jer bi u tom slučaju slika bila previše umanjena. Ulazna slika u našim skupovima podataka je veličine 28 x 28. Nakon jednog para je 14 x 14. Nakon dva para dolazimo do 7 x 7. Nakon tri je 4 x 4 (ili 3 x 3 ako ne koristimo padding = „same“). Iz navedenog slijedi da nema smisla raditi četvrtu kovoluciju.

Zaključak testiranja

Iz eksperimenta vidimo da su 3 para konvolucijskih slojeva nešto bolja od 2 para. No, zbog učinkovitosti, poboljšanje ne opravdava dodatne računske troškove pa koristimo 2 para konvolucijskih slojeva.

4.5.Testiranje broja karata značajki

Primjeri karata značajki koje smo testirali su redom:

- 784 - [8C5-P2] - [16C5-P2] - 256 - 10
- 784 - [16C5-P2] - [32C5-P2] - 256 - 10
- 784 - [24C5-P2] - [48C5-P2] - 256 - 10
- 784 - [32C5-P2] - [64C5-P2] - 256 - 10
- 784 - [48C5-P2] - [96C5-P2] - 256 - 10
- 784 - [64C5-P2] - [128C5-P2] - 256 – 10

Zaključak testiranja

Iz testiranja proizlazi da su 32 mape u prvom konvolucijskom sloju i 64 karte u drugom konvolucijskom sloju najbolje. Arhitekture s više karata značajki rade samo malo bolje i ne vrijede dodatne troškove izračuna.

4.6. Testiranje veličine gustog sloja (engl. dense layer)

Primjeri koje smo testirali su

- 784 - [32C5-P2] - [64C5-P2] - 0 - 10
- 784 - [32C5-P2] - [64C5-P2] - 32 - 10
- 784 - [32C5-P2] - [64C5-P2] - 64 - 10
- 784 - [32C5-P2] - [64C5-P2] - 128 -10
- 784 - [32C5-P2] - [64C5-P2] - 256 - 10
- 784 - [32C5-P2] - [64C5-P2] - 512 -10
- 784 - [32C5-P2] - [64C5-P2] - 1024 - 10
- 784 - [32C5-P2] - [64C5-P2] - 2048 – 10

Zaključak testiranja

Pokazalo se da je 128 jedinica najbolje. Gusti slojevi s više jedinica djeluju samo malo bolje i ne vrijede dodatne računske troškove. Ispitivali samo i dva uzastopna gusta sloja umjesto jednog, ali to nije pokazalo veći benefit nad jednim gustim slojem.

4.7. Droupout testiranje

Dropout sprječava prekomjerna namještanja mreže i na taj način pomaže da se mreža generalizira bolje.
Testirali smo vrijednosti: 0%, 10%, 20%, 30%, 40%, 50%, 60% i 70%.

Zaključak testiranja

Dropout postavljen na 40% se pokazao kao najuspješniji.

4.8. Testiranje naprednih značajki

Umjesto jednog sloja veličine 5x5, možemo oponašati sloj 5x5 pomoću dva uzastopna sloja 3x3 i bit će nelinearniji. Isprobali smo iduće značajke:

- zamijeniti '32C5' sa '32C3-32C3'
- zamijeniti 'P2' sa '32C5S2'
- dodati batch normalizaciju
- dodati data augmentation

Zaključak testiranja

Iz ovog eksperimenta vidimo da svaka od četiri napredne značajke poboljšava točnost. Prvi model ne koristi napredne značajke. Drugi koristi samo trik s dvostrukim savijanjem. Treći koristi samo trik sloja podskupa koji se može naučiti. Treći model koristi obje ove tehnike plus serijsku normalizaciju. Posljednji model koristi sve ove tri tehnike plus povećanje podataka i postiže najbolju točnost.

5. Usporedba rezultata

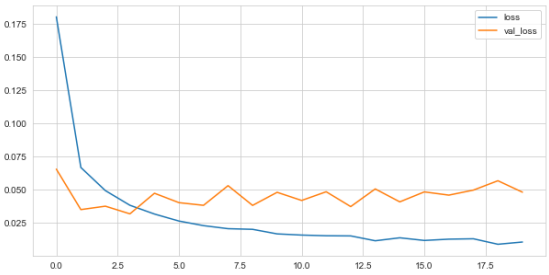
Funkcija gubitka mjeri odstupanje izlaznog rezultata jedinice od stvarne vrijednosti pridružene stvarnom objektu. Gubitak, odnosno, greška će biti veća ako smo objekt pogrešno klasificirali. Dakle, cilj je da funkcija gubitka bude što manja. Funkcija gubitka pokazuje koliko dobro ili loše se model ponaša nakon svake iteracije u optimizacji, a prikaz točnosti mjeri koliko je točno predviđanje modela u usporedbi sa stvarnim podacima. Prikaz točnosti je izražen u postotcima za razliku od funkcije gubitka.
Promatrale smo slučaj na train setu od 60 000 slika i validate setu od 10 000 slika, koristeći 20 iteracija.
Ako promatramo grafove za MINSET (slika 4 i slika 6) možemo zaključiti da skup testnih podataka ima dosta visoku točnost od 99.05% za train set podataka i 98.84% za test set podataka, dok su funkcije

gubitka niske, odnosno za train setu 3.03 i za test set 4.52).

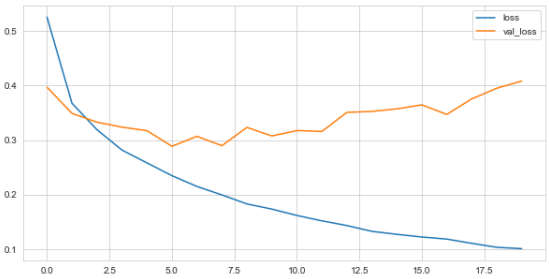
Za Fashion-MNIST (slika 5 i slika 7) smo dobile malo lošije rezultate u odnosnu na klasični MNIST dataset, dakle za točnost je 92.39% za training set i 89.30% za test set, gubitci za Fashion-MINST za train set su 20.144 dok su za test set 34,103.

U sljedećim potpoglavljima grafički su prikazani prethodno opisani podaci.

5.1. Funkcija gubitka

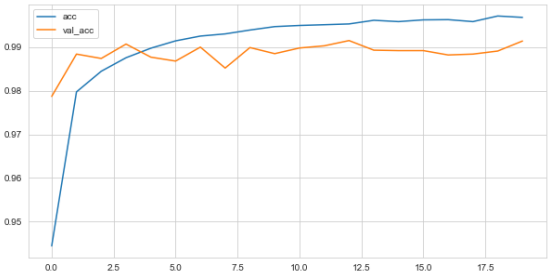


Slika 4 Funkcija gubitka za MNIST dataset

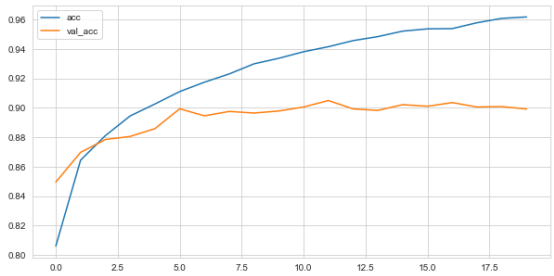


Slika 5 Funkcija gubitka za Fashion MNIST dataset

5.2. Prikaz točnosti

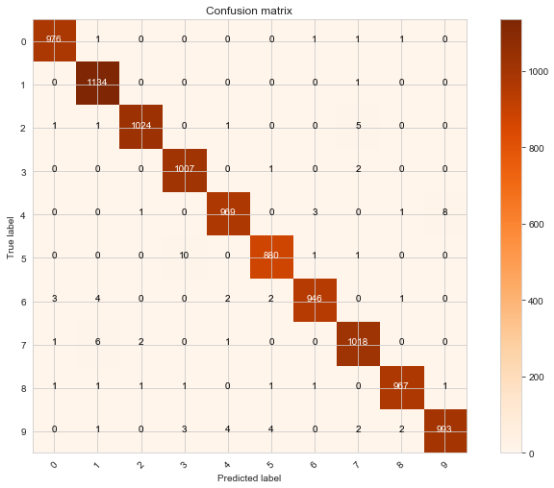


Slika 6 Točnost po iteraciji za MNIST dataset

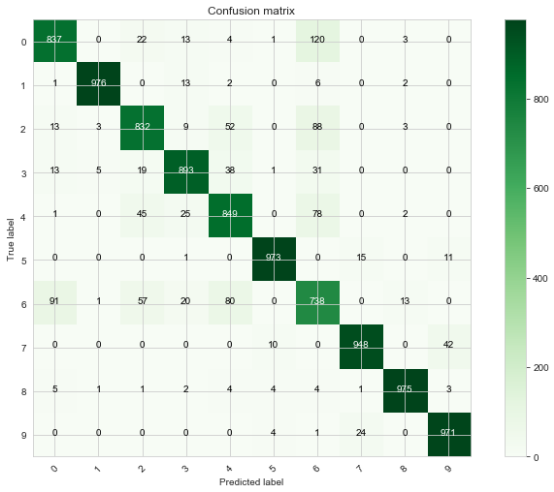


Slika 7 Točnost po iteraciji za Fashion MNIST dataset

5.3. Matrica konfuzije



Slika 8 Matrica konfuzije za MNIST dataset



Slika 9 Matrica konfuzije za Fashion-MNIST dataset

6. Zaključak

Rješavanjem problema klasifikacije znamenaka i odjevnih predmeta dobili smo uvid o mogućnosti TensorFlowa kao biblioteke za izradu modela neuronske mreže. Prije izrade modela bilo je potrebno

temeljito proučiti dokumentaciju i same skupove podataka kako bi odgovarali kreiranom modelu. Također, za rješavanje problema bila je potrebna velika količina predznanja iz programiranja općenito, programskog jezika Python i matematike, a kao pomoć i razumijevanju neuronskih mreža bilo je potrebno proužiti dokumentaciju CNN modela.

Rezultati koje smo dobili su s očekivanim točnostima. Budući da su neke slike iz naših skupova podataka bile teže za rasvrstati u odredenu kategoriju, bilo je očekivano da se neće moći postići stopostotna točnost. Ovaj rad nam je bio prvi doticaj sa ovakvim načinom programiranja i zadovoljne smo rezultatima koje smo dobili.

7. Literatura

[1]https://research.zalando.com/welcome/mission/research-projects/fashion-mnist/

[2] https://www.tensorflow.org

[3] https://www.kaggle.com