**Neuralne mreže**

**Projekat iz dubokog učenja**

Andrej Davidovic 0302/2020

Djurdja Joksimović 0281/2020

Izveštaj

Dataset koji je korišćen: <https://www.kaggle.com/datasets/olafkrastovski/handwritten-digits-0-9/code>.

Korišćena je baza podataka rukom pisanih cifara. Slike su klasifikovane u 10 klasa(brojevi od 0 do 9).

Prve CNN mreže (LeNet) su se bavile problemom klasifikovanja rukom pisanih cifara I eksperimentalno je dokazano da su konvolucione neuralne mreže pokazale mnogo bolje performanse u klasifikaciji u odnosu na ‘obične’ neuralne mreže.

**Grafik date baze podataka:**

Chart, bar chart

Description automatically generated

Na x osi su obeležene vrednosti klasa:broj\_odabiraka\_klase da bi se preciznije broj odabiraka svake klase video.

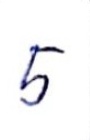
Sa grafika možemo videti da su vrednosti za svaku klasu približne, tj. da je razlika u broju primeraka klase procentualno mala pa mozemo smatrati da su podaci balansirani.

**Primeri iz svake klase:**

A picture containing insect

Description automatically generated

Klasa 0 Klasa 1 Klasa 2 Klasa 3 Klasa 4

Kasa 5 Klasa 6 Klasa 7 Klasa 8 Klasa 9

Podaci se dele na 2 skupa: **train i validation**.

Podela na train i validation skup se vrsi zbog provere tacnosti mreze, posto je moguce da se mreza preobuci samo za zadati skup nad kojim se vrsi treniranje i da izgubi mogucnost generalizacije. Zbog toga se ubacuje validirajući skup podataka. Bitno je da se podaci nad kojima se vrši treniranje uzimaju iz različitih klasa da se mreža ne bi naučila samo na jednu klasu (npr. broj 0). Primer za to je neuralna mreža za bolnicu koja se trenira isključivo nad skupom bolesnih pacijenata, ona neće efikasno raditi kada se pojavi zdrav pacijent. Za validaciju se uzima 20%, a za treniranje preostalih 80%.

Dimenzije slika su namestene na 64x64, batch je podesen na 64, tj. ne vrsi se odmah update velicina i podataka, vec se posle svake 64.obradjene slike vrsi azuriranje.

Dodali smo augmentaciju(umnozavanje podataka) pri cemu je nad podacima vrseno uvecavanje i rotacija podataka, da bismo dobili na raznovrsnosti podataka i sprecili mrezu da se navikne na dati skup podataka.Takodje smo dodali i skaliranje podataka, to se vrsi deljenjem sa 255 posto pikseli uzimaju vrednost od 0 do 255 u zavisnosti od svoje svetline. Deljenjem sa 255 dobijamo vrednosti od 0 do 1 pri cemu ce 1 imati vrednost koja je prethodno iznosila 255 (maksimum).

Koriscena je aktivaciona funkcija reLu, to je funkcija koja je danas najcesce koriscenja zbog svojih osobina da ne dolazi do saturacije(zasicenja) neurona za odredjene input podatke, funkcija je zero-centrirana i nece imati problem da sledeci sloj prima samo pozitivne podatke (tj.izlaz iz datog sloja ce sadrzati i negativne parametre) i samim time promena nece biti samo inkrementalna ili samo dekrementalna. Poslednji sloj je softmax, funkcija za klasifikaciju vise klasa. Izlazi predstavljaju verovatnoce da izlaz sloja pripada nekoj od datih klasa I u zbiru iznose 1.

Optimizator koji koristimo je Adamov optimizator, optimizator sluzi za pronalazenje minimuma kriterijumske funkcije, dok kao kriterijumsku funkciju koristimo SparseCategoricalCrossentropy.

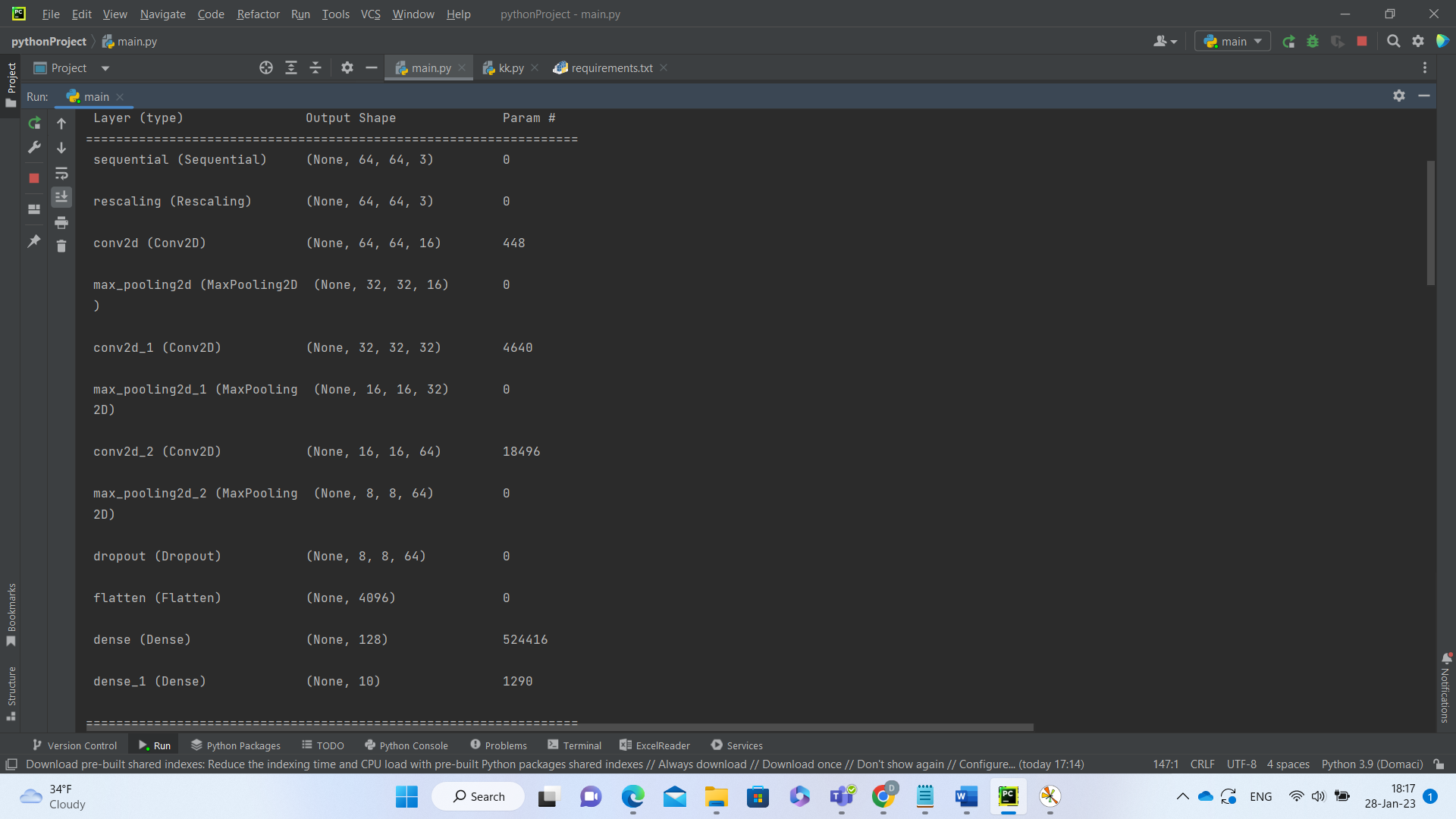
**Arhitektura mreze:**

Pooling sloj koji se koristi je max pooling 2x2, sto je danas najcesce korisceno (prve neuralne mreze kao sto su leNet su koristili average pooling), on dolazi posle svakog Conv sloja da bi izvrsio izvlacenje esencijalnih osobina.

Dopunjavanje (padding) je same padding, tj.podesavanje dimenzija izlaza da se poklapaju sa dimenzijama ulaza. Ovo se radi da se ne bi smanjivale dimenzije slika posle svakog primenjivanja filtera, pritom se i ivicni pikseli koriste samo jednom.

Flatten sloj se nalazi iznad poslednja dva sloja koja imaju ulogu slicnu obicnim neuralnim mrezama, flatten sloj ulazne podatke transformise u jednodimenzionalni niz sa kojim radi poslednji sloj(softmax).

**Struktura mreze**:



Total params: 549,290

Trainable params: 549,290

Non-trainable params: 0

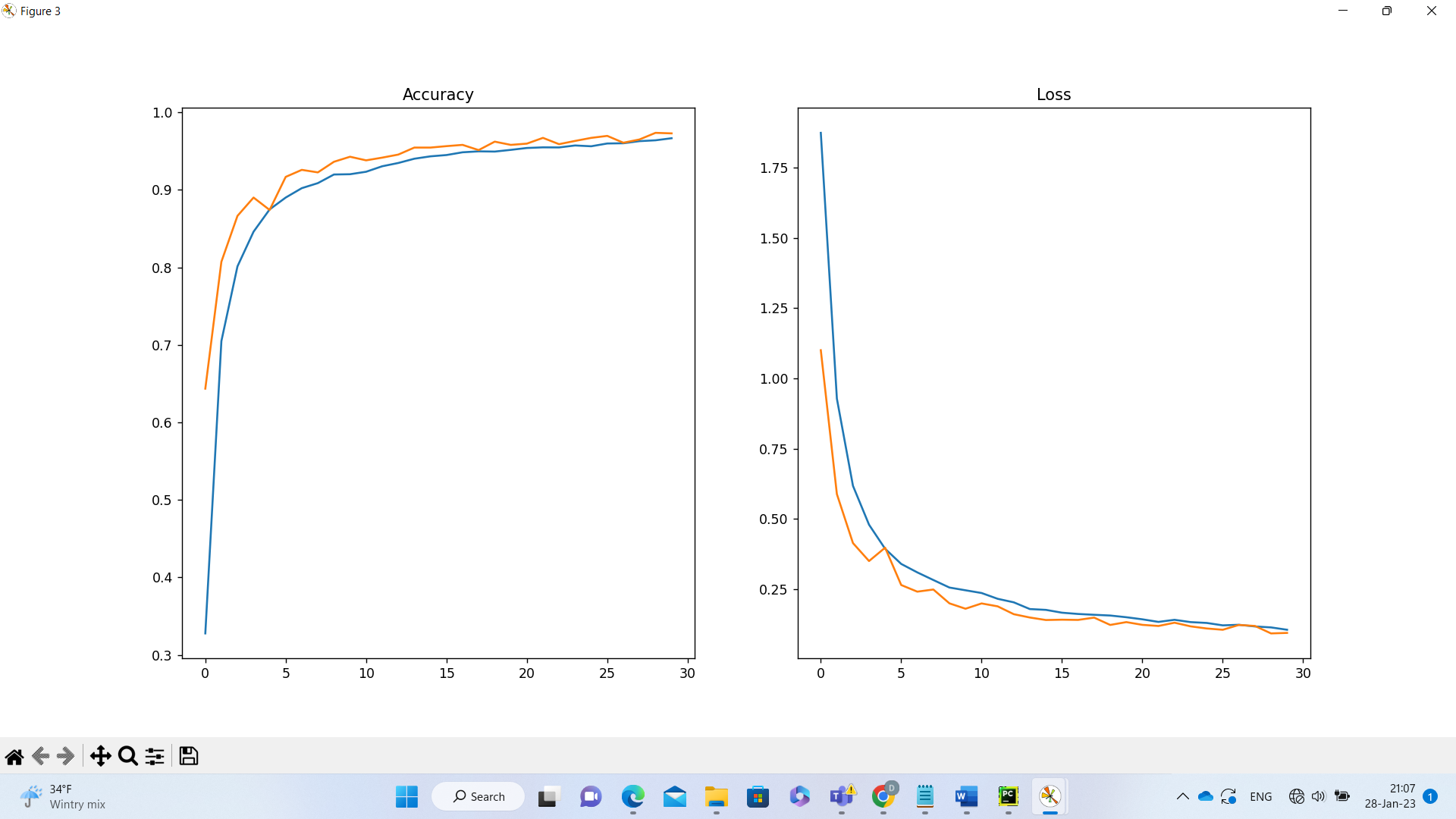
**Preobucavanje neuralne mreze** se javlja kod mreza koje imaju veliki broj parametara u odnosu na broj trening podataka. Kod preobucene mreze, podaci nad Kojima je vrseno treniranje pokazuju dosta veliki procenat tacnosti, dok je procenat tacnosti nad podacima koje mreza prvi put vidi dosta los (gore objasnjeno na primeru pacijenata).

Koristimo dropout sloj za zastitu od preobucavanja , verovatnoca za odbacivanje neurona iz mreze jeste 0.2, pri cemu svaki ulaz u mrezu prolazi kroz razlicitu mrezu neurona i to je jedan od nacina za sprecavanje preobucavanja zbog prevelikog broja parametara koji se koriste. Tokom faze obucavanja neuralne mreze, deo skrivenih neurona je nasumicno izostavljen u svakoj iteraciji sa odredjenom verovatnocom.

Takodje, koristimo I early stopping, u slucaju da dodje do preobucavanja pre zavrsetka zadatog broja epoha.

**Grafik performanse neuralne mreze kroz epohe obucavanja:**

Kroz 30 epoha:



**Matrica konfuzije za skup validacije:**

Tacnost na skupu validacije 95.94061702621202%

Graphical user interface

Description automatically generated

**Matrica konfuzije za skup za treniranje:**

Graphical user interface

Description automatically generated

Tacnost na skupu trening 97.6687543493389%

Vidimo da su metode za sprecavanje preobucavanja bile uspesne, posto su razlike izmedju tacnosti u trening I val skupu zanemarljive, I tacnost je visoka u oba skupa.

Dobro klasifikovani podatak:



Predvidjeno: 0 Ocekivano: 0

Lose klasifikovan podatak:



Predvidjeno: 9 Ocekivano: 8

Podaci su skalirani na velicinu 64x64, posto je nasa mreza obucena nad slikama date dimenzije, druga slika je posebno ubacena zbog dobro obucene mreze koja je odlicno raspoznavala slike iz dataset-a.