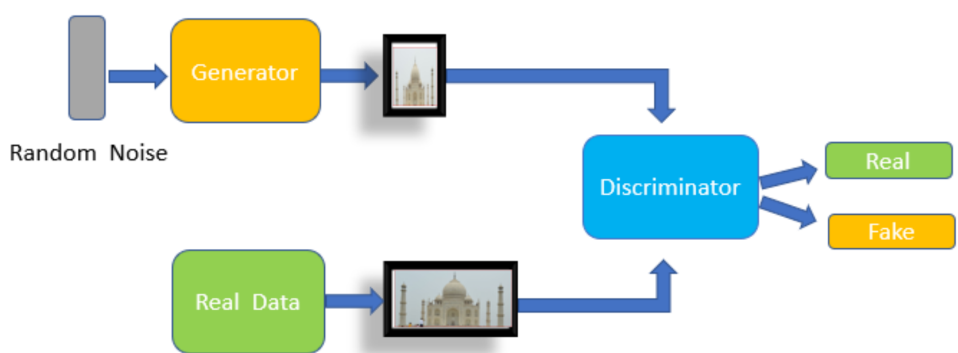


## Uvod

Za generisanje slika koriste se Generativne adverzalne mreže (GAN). Ovakva tehnika nam omogućava da treniramo mrežu da generiše slike slične ulaznim podacima. U ovom projektu su se generisali jednostavni crteži mačaka. GAN je vrsta minimax algortima, gde imamo generativnu funkciju i diskriminator koji čine dve neuronske mreže.

Generatorna funkcija transformiše ulaz koji je vrsta šuma u nešto što liči na realne podatke. Podaci iz generatorna funkcije se ubacuju u diskriminator zajedno sa pravim podacima. Tada diskriminator pokušava da odredi da li je slika prava ili je generisana od strane generatora.

U projektu se koristi slična tehnika, ubacujemo prave podatke koji se sastoje od crteža mačaka i pokušavamo da treniramo generator tako da na izlazu iz mreže imamo slike slične originalnim.



## Podaci

Podaci koji su korišćeni kao ulazu u generator su dobijeni sa Google Quick Draw! projekta. Podaci su uzeti u formatu pogodnom za obradu tako velikog seta podataka.

Fajl u sebi ima 75000 crteža i podeljen je na 70000 za trening, 2500 za validaciju i 2500 za test.

Ulazni podatak u mrežu u jednoj istanci prilikom treniranja izgleda ovako:



## Metodologija

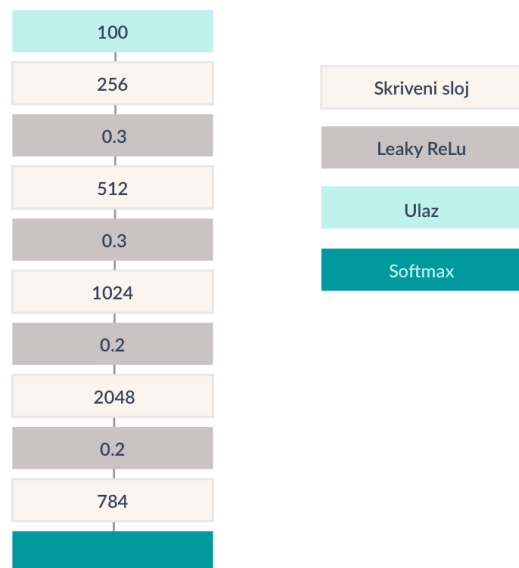
Generativne adverzalne mreže se sastoje iz dve neuronske mreže koje se "takmiče" jedna sa drugom. Zbog toga se zovu adverzalne mreže. Prva mreža je generatorna koja pokušava da ulazni nasumični šum napravi što sličniji pravim podacima. Slike koje koristimo su višedimenzionalni podaci.

Uzimamo jedan uzorak iz šuma i ubacujemo ga u generatornu funkciju tj. generatornu neuronsku mrežu. Generator pokušava da nauči funkciju distribucije podataka iz realnih slika. Generator i diskriminator igraju jednu minimax igru. Generator pokušava da maksimizira vrednost funkcija dok diskriminator pokušava da je minimizira.

$$V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

### Generatorna neuronska mreža

Projekat u sebi ima generatornu mrežu koja izgleda kao slika desno. Jedan od slojeva je Leaky ReLu funkcija. Ona sprečava problem kod neuronskih mreža kada vrednost skrivenog sloja vrati negativnu vrednost. Funkcija vraća nulu ako vrednost bude negativna. Softmax na kraju funkcionira slično kao sigmoidna funkcija, tj. vraća 0 ili 1.



### Diskriminatorna neuronska mreža

Pored generatorne mreže postoji i diskriminator koji pokušava da odredi da li je slika prava ili ne. Ulazu u ovu mrežu je izlaz iz generatora i prava slika iz našeg dataset-a.

Diskriminator koristi stohastički gradijentni spust.

U mreži se koristi Dropout funkcija koja sprečava da neuroni postanu previše zavisni jedni od drugih.

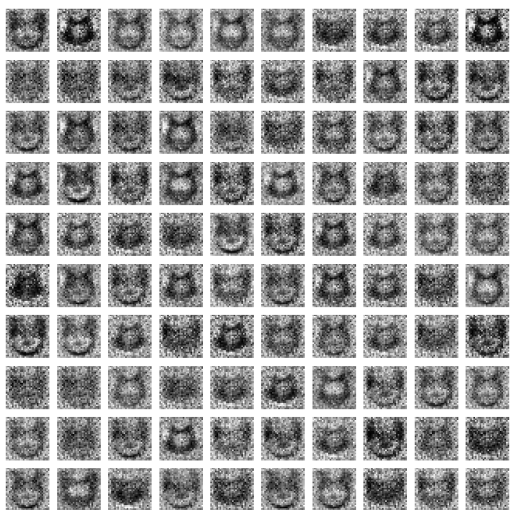


## Rezultati

Ulazni podatak



1. epoha



50. epoha



200. epoha



## Zaključak i vreme izvršavanja

Generativne adverzalne mreže su veoma primenjive u mnogim oblastima i veoma su fleksibilne. Treniranje ove mreže na računaru sa integrisanom grafičkom je pokazivalo rezultate ne toliko brzo. Broj iteracija po sekundi je bilo sedam, što je značilo da je za 100 epoha trebalo oko sat vremena. Isto treniranje je rađeno i na GTX 2080 grafičkoj što je dovelo do određenog ubrzanja u treniranju. Rezultati nisu bili toliko različiti pošto kod nije bio optimizovan da radi na grafičkoj. Za neki dalji i budući rad promenila bih i optimizovala projekat tako da se može iskoristiti veća paralelizacija, koju ova grafička podržava.