
GAN

GENERATIVE ADVERSARIAL NETWORKS

AUTORI

NATALIJA JOVANOVIĆ 226/2015

MARIJA MILIĆEVIĆ 105/2015

MATEMATIČKI FAKULTET
FEBRUAR 2019.

Uvod

Veštačke neuronske mreže su familija statističkih modela učenja inspirisana biološkim neuronskim mrežama. Koriste se u svrhu aproksimacije funkcija koje mogu zavisiti od velike količine ulaznih podataka, a koje su u principu nepoznate. Veštačke neuronske mreže su sistemi međusobno povezanih neurona koji šalju poruke jedni drugima. Veze između ovih neurona imaju numeričke težine koje mogu biti podložne promenama u zavisnosti od iskustva, što neuronske mreže čini adaptivnim i sposobnim za učenje.

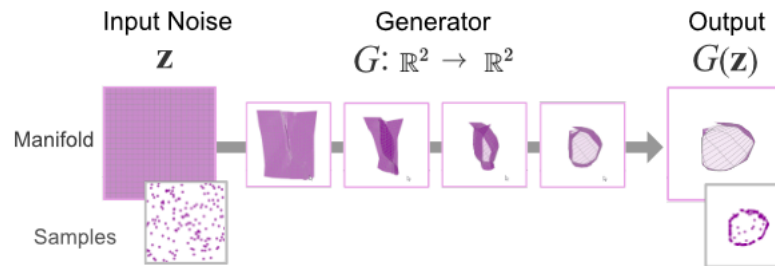
GAN su vrsta neuronskih mreža koje se sastoje od dve neuronske mreže koje se međusobno takmiče. Ova tehnika može da generiše slike tako da posmatraču izgledaju barem površno autentično time što imaju dosta realnih karakteristika.

Opis problema

Glavni cilj GAN-a je da izgenerise podatke od nule (uglavnom slike, ali može i muziku). GAN pravi dve mreže - generator, koji generiše podatke, i diskriminator, koji procenjuje da li je podatak došao iz pravih podataka ili ga je generator generisao (binarni klasifikator). GAN trenira te mreže koje se takmiče sve dok se ne uspostavi neka ravnoteža. Cilj je da se obe mreže vremenom poboljšavaju, što prouzrokuje da generator da što realniji izlaz. Ova situacija može da se modeluje minimax igrom. Dakle, postoje dva agenta, kriminalac i policajac. Ciljevi su im suprotni, kriminalac hoće da na kompleksne načine falsifikuje novac, tako da policajac ne može da prepozna da li je novac falsifikovan ili ne, dok policajac hoće da što bolje prepozna da li je novac falsifikovan ili ne. Tokom procesa i policajac i kriminalac razvijaju sve sofisticiranije tehnologije da ispune svoje ciljeve.

Prvo uzmemo neki šum z iz normalne ili uniformne raspodele. Sa z kao ulaz, koristimo generator G da napravimo sliku x , koja je slična skupu podataka koji imamo ($x = G(z)$). Konceptualno, z predstavlja skrivene karakteristike generisane slike (npr. boju, oblik). Dakle, generator je funkcija koja transformiše slučajan ulaz u sintetički izlaz, gde je slučajan ulaz neki 2D uzorak sa vrednošću (x, y) , a izlaz je 2D uzorak koji je mapiran na drugu poziciju, tj. lažni uzorak.

Zašto uopšte koristiti šumove? Ne bi bilo zanimljivoda se napravi sistem koji daje isti rezultat svaki put kad se pokrene. Takodje je bitno razmišljati o verovatnoćama, jer nam pomaže da prebacimo problem generisanja slike u "prirodnu matematiku". Zbog ovog tipa problema - modelovanja funkcije u visedimenzionom prostoru, su neuronske mreže i napravljene.



Funkcija gubitka

Diskriminator za izlaz daje vrednost $D(x)$, tj. verovatnoću da je x iz realne slike. Mi želimo da maksimizujemo verovatnoću da tačno prepozna, realnu sliku kao realnu, generisanu kao lažnu. Kao merilo gubitka, korišćemo cross-entropy ($p \log(p)$).

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

Takodje, želimo da generator što bolje prevvari diskriminatora (minimizuje se verovatnoća da prepozna lažnu sliku).

$$\min_G V(G) = \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]$$

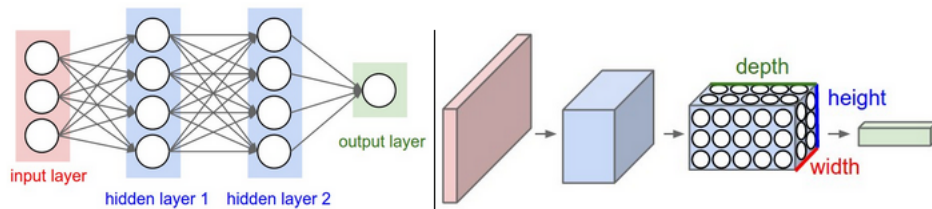
Kada su obe funkcije definisane, zajedno uče pomoću gradijentnog spusta. Popravljamo parametre generatornog modela i izvodimo jednu iteraciju gradijentnog spusta nad diskriminatorom koristeći prave i generisane slike. Zatim menjamo stranu. Popravljamo diskriminator i treniramo generator za jednu iteraciju. Treniramo obe mreže naizmeničnim koracima sve dok generator ne proizvede kvalitetnu sliku. Gradijenti se koriste za propagaciju unazad.

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

Konvolutivne neuronske mreže

Konvolutivne neuronske mreže primenjuju serije filtera na sirove podatke slike kako bi izdvojile i naučile karakteristike višeg nivoa, koje model kasnije koristi za klasifikaciju.

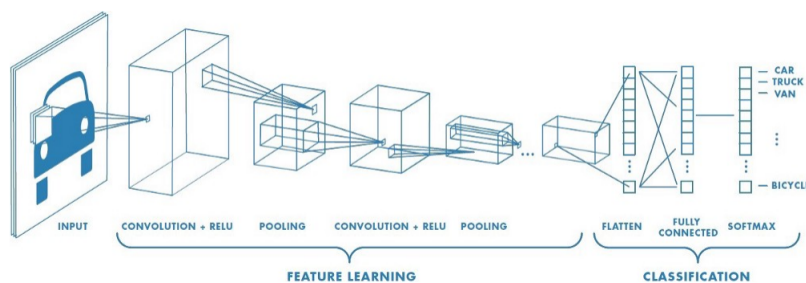
Regularne neuronske mreže za ulaz dobijaju jedan vektor i transformišu ga kroz niz skrivenih slojeva. Kada su slike dimenzija 64x64x3 (64 dužina, 64 visina, 3 kanala za boju), prvi skriveni sloj bi imao $64 \times 64 \times 3 = 12288$ težina. Ovaj broj deluje izvodljivo, ali očigledno ova potpuno povezana struktura ne može da skalira do većih slika. Kod slika realnijih veličina, npr. 200x200x3, neuroni će imati $200 \times 200 \times 3 = 120000$ težina. Štaviše, skoro sigurno bismo želeli da imamo nekoliko takvih neurona tako da bi se parametri brzo uklopili. Očigledno, ova potpuna povezanost je rasipna i veliki broj parametara bi brzo doveo do preprilagodjenosti. Za razliku od običnih neuronskih mreža, sloj konvolutivne neuronske mreže uređjuje neurone u tri dimenzije: širina, visina, dubina. Neuroni će biti povezani samo sa malim regijama iz prethodnog sloja umesto sa svim neuronima.



Konvolutivne neuronske mreže se sastoje iz pet delova:

1. Ulaz - sadrži sirove podatke slike
2. Konvolutivni sloj - primenjuje odredjeni broj konvolutivnih filtera na sliku; za svaku podoblast, sloj izvršava skup matematičkih operacija kako bi dobio jednu vrednost na izlazu.
3. *ReLU* sloj - primenjuje *ReLU* aktivacionu funkciju (*ReLU* - pozitivni deo argumenta, $f(x) = \max(0, x)$, x je ulaz).

4. Slojevi za objedinjavanje - upsample-uje podatke slike, izdvajajući ih po konvolutivnim slojevima da bi smanjio dimenzionost mape karakteristika kako bi se smanjilo vreme procesiranja. Algoritam koji se koristi je *maxpooling* koji izdvaja regione mape karakteristika (npr. 2x2 piksel ploča), čuva njihovu maksimalnu vrednost i odbacuje sve ostale vrednosti.
5. Slojevi gustine (potpuno povezani) - izvršavaju klasifikaciju nad karakteristikama koje su izdvojene konvolutivnim slojem i upsample-ovane u sloju objedinjavanja. Svaki čvor iz sloja je povezan sa svakim čvorom iz sloja predikcije.



Dobijeni rezultati

Zaključak

Literatura

- [1] Ian J. Goodfellow, Jean Pouget-Abadie , Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio *Generative Adversarial Nets*
- [2] Peter J.B. Hancock and Charlie D. Frowd *Evolutionary generation of faces*
- [3] Jon Gauthier *Conditional generative adversarial nets for convolutional face generation*
- [4] Yaohui Wang, Antitza Dantcheva, Francois Bremond *From attribute-labels to faces: face generation using a conditional generative adversarial network*
- [5] <https://fairyonice.github.io/My-first-GAN-using-CelebA-data.html>
- [6] <https://towardsdatascience.com/having-fun-with-deep-convolutional-gans-f4f8393686ed>