

Polunadzirana klasifikacija rukom pisanih znakova generativnim suparničkim modelima

Marko Jelavić

Fakultet Elektrotehnike i Računarstva

July 20, 2017

- 1 Generativne suparničke mreže
- 2 Duboke konvolucijske generativne suparničke mreže
- 3 Klasifikatorska generativna suparnička mreža
- 4 Zaključak

Generativne suparničke mreže

- Ideja suprotstavljanja dvaju višeslojnih perceptrona treniranih unazadnom propagacijom
- Suprotstavljeni su generator i diskriminator
- Generator generira podatke nalik stvarnima
- Diskriminator određuje klasu podatka (stvaran ili generiran)
- Igraju minimax game po sljedećoj formuli:

Minimax igra

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

Generativne suparničke mreže

$$\begin{aligned} V(G, D) &= \int_x p_{data}(x) \log D(x) dx + \int_z p_z(z) \log(1 - D(G(z))) dz \\ &= \int_x (p_{data}(x) \log D(x) + p_g(x) \log(1 - D(x))) dx \end{aligned} \quad (1)$$

- Optimalan diskriminator:

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \quad (2)$$

Generativne suparničke mreže

$$f(y) = a \log y + b \log(1 - y) \quad (3)$$

- gdje je $a = p_{data}(x)$, $b = p_g(x)$, $y = D(x)$.

$$f'(y) = \frac{a}{y} - \frac{b}{1 - y} \quad (4)$$

$$\frac{a}{y} - \frac{b}{1 - y} = 0 \quad (5)$$

$$a - ay = by$$

$$a = ay + by$$

$$a = y(a + b)$$

$$y' = \frac{a}{a + b}$$

$$f''(y) = -\frac{a}{y^2} - \frac{b}{(1-y)^2} \quad (6)$$

$$f''\left(\frac{a}{a+b}\right) = -\frac{a}{\left(\frac{a}{a+b}\right)^2} - \frac{b}{\left(1 - \frac{a}{a+b}\right)^2}$$

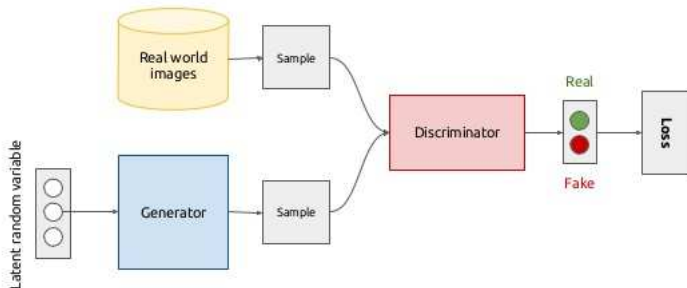
- a i b su distribucijske funkcije, stoga je navedeni ekstrem maksimum funkcije

Generativne suparničke mreže

$$\begin{aligned} C(G) &= \max_{\mathbf{D}} V(D, G) \\ &= V(D^*, G) \\ &= \mathbb{E}_{x \sim p_{data}(x)} [\log D^*(x)] + \mathbb{E}_{x \sim p_g(x)} [\log(1 - D^*(x))] \\ &= \int_x p_{data}(x) \log D^*(x) dx + \int_x p_g(x) \log(1 - D^*(x)) dx \\ &= \int_x p_{data}(x) \log \left(\frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right) dx + \int_x p_g(x) \log \left(1 - \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right) dx \\ &= \int_x p_{data}(x) \log \left(\frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right) dx + \int_x p_g(x) \log \left(\frac{p_g(x)}{p_{data}(x) + p_g(x)} \right) dx \\ &= \int_x p_{data}(x) \log \left(\frac{p_{data}(x)}{\frac{p_{data}(x) + p_g(x)}{2}} \right) dx - \int_x p_{data}(x) \log 2 dx \\ &\quad + \int_x p_g(x) \log \left(\frac{p_g(x)}{\frac{p_{data}(x) + p_g(x)}{2}} \right) dx - \int_x p_g(x) \log 2 dx \\ &= \int_x p_{data}(x) \log \left(\frac{p_{data}(x)}{\frac{p_{data}(x) + p_g(x)}{2}} \right) dx - \log 2 \\ &\quad + \int_x p_g(x) \log \left(\frac{p_g(x)}{\frac{p_{data}(x) + p_g(x)}{2}} \right) dx - \log 2 \\ &= KL \left(p_{data} \parallel \frac{p_{data} + p_g}{2} \right) + KL \left(p_g \parallel \frac{p_{data} + p_g}{2} \right) - 2 \log 2 \\ &= -\log 4 + 2JSD(p_{data} \parallel p_g) \\ &= \log \frac{1}{4} + 2JSD(p_{data} \parallel p_g) \end{aligned} \tag{3.11}$$

Generativne suparničke mreže

Generative adversarial networks (conceptual)



5

Figure: Generativna suparnička mreža

Algoritam

for number of training iterations do

for ksteps do

Sample minibatch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from $p_z(z)$

Sample minibatch of m examples $\{x^{(1)}, \dots, x^{(m)}\}$ from real data

Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m [\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))] \quad (7)$$

end

Sample minibatch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from $p_z(z)$

Update the generator by ascending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log D(G(z^{(i)})) \quad (8)$$

end

Algorithm 1: GAN

Implementacija

- Uzorkovanje šuma - $\mathcal{N}(0, 1)$. Vektor veličine 100.
- Inicijalizacija svih težina `tf.truncated_normal` s $\mathcal{N}(0, 0.01)$, pristranosti na 0.1
- Generator
 - 100x200x784
 - Skriveni sloj ReLU, izlazni sloj sigmoida
 - Adam s predefiniranim parametrima
- Diskriminator
 - 784x200x1
 - Skriveni sloj ReLU, izlazni sloj sigmoida
 - Stohastički gradijentni spust - stopa učenja 0.01
- 100000 iteracija, batch veličina 500

Rezultati



Figure: Uzorci nakon 0, 25000, 50000, 100000 iteracija

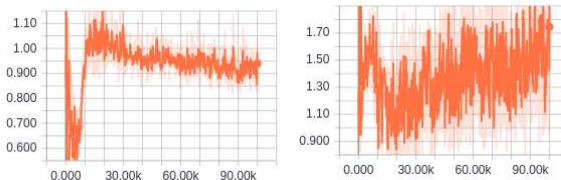


Figure: Gubitak diskriminatora (lijevo) i generatora (desno)

Duboke konvolucijske generativne suparničke mreže

- Umjesto višeslojnog perceptrona - konvolucijske arhitekture
- Uvedena još 3 bitna faktora u treniranju
 - Nema slojeva sažimanja - generator i diskriminator uče prostorna povećanja/smanjenja preko koraka (najčešće 2)
 - Izostavljanje potpuno povezanih slojeva osim po 1 na početku (generator) i kraju (diskriminator)
 - Grupna normalizacija (osim na ulazu diskriminatora, te izlazu generatora)
- Također se savjetuje korištenje ReLU (generator) i LeakyReLU (diskriminator)

Implementacija

- Šum je ovaj put uzorkovan iz $\mathcal{U}(-1, 1)$ (nema vidne razlike)
- Implementirana su 4 modela, interesantni su nam 1. i 4.
- Model 1 generator
 - Potpuno povezani sloj 100×6272 koji se zatim preoblikuje u $7 \times 7 \times 128$
 - Prvi dekonvolucijski sloj $7 \times 7 \times 128$, drugi $14 \times 14 \times 1$
- Model 4 generator
 - Potpuno povezani sloj 100×2048 koji se zatim preoblikuje u $2 \times 2 \times 512$
 - Prvi dekonvolucijski sloj $2 \times 2 \times 256$, drugi $4 \times 4 \times 128$, treći $7 \times 7 \times 64$, četvrti $14 \times 14 \times 1$
- Diskriminator sva 4 modela
 - Prvi konvolucijski sloj $5 \times 5 \times 16$, drugi $5 \times 5 \times 32$, treći $5 \times 5 \times 64$, četvrti $5 \times 5 \times 128$
 - Izlaz konvolucija je $2 \times 2 \times 128$ koji se zatim izravna te preda sigmoidalnom neuronu

Implementacija

- Inicijalizacija svih težina `tf.truncated_normal` s $\mathcal{N}(0, 0.02)$, pristranosti na 0.1
- Generator i diskriminator se treniraju Adam optimizatorom
 - Stopa učenja 0.0002
 - $\beta_1 = 0.5$
 - Ostalo predefinirano
- Veličina grupe 64
- Postupak se ponavlja 100000
- Za svako treniranje diskriminatora, generator se trenira dvaput!

Rezultati - model 1

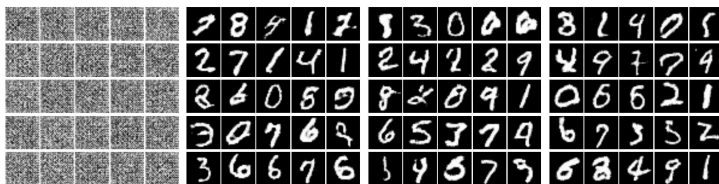


Figure: Uzorci nakon 0, 25000, 50000, 100000 iteracija

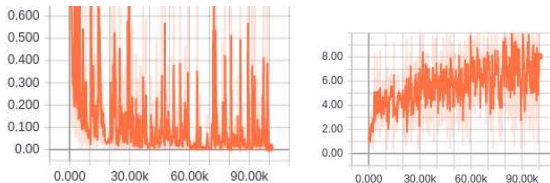


Figure: Gubitak diskriminatora (lijevo) i generatora (desno)

Rezultati - model 4

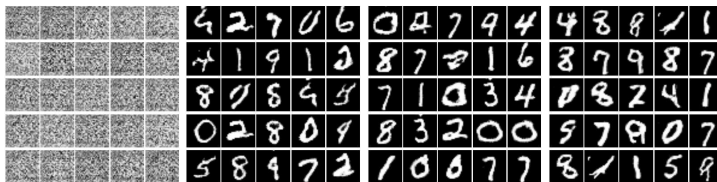


Figure: Uzorci nakon 0, 25000, 50000, 100000 iteracija

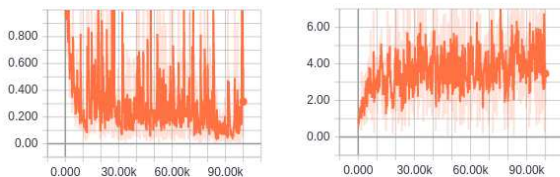


Figure: Gubitak diskriminatora (lijevo) i generatora (desno)

- Šetnja latentnim prostorom (kolinearni vektori)
 - Provjera kakve slike generator vraća za $[-0.05, 0.05, \dots -0.05]$, $[0, 0, \dots 0]$, $[0.05, 0.05, \dots 0.05]$
 - U gornjem primjeru očigledno je pomak 0.05, stoga se svi vektori razlikuju za $k * 0.05$
 - Drugi način je generiranje vektora iste veličine iz distribucije $\mathcal{U}(-0.05, 0.05)$ i korištenje njega kao pomak
- Izvlačenje značajki
 - Izvuku se značajke koje je diskriminator naučio
 - Proslijede se linearnom klasifikatoru te se uspoređuju s drugim modelima

Šetnja latentnim prostorom

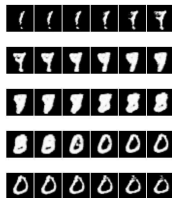


Figure: Primjer šetnje skrivenim prostorom



Figure: Primjer druge šetnje skrivenim prostorom

Table: Rezultati mjerenja za Linearni SVM

Klasifikator	Cijeli MNIST	1000 primjera	100 primjera
LSVM (pikseli)	91.75% \pm 0%	83.13% \pm 0%	72.2% \pm 0%
LSVM (RND značajke)	93.86% \pm 0.48%	83.55% \pm 1.15%	66.84% \pm 1.7%
LSVM (DCGAN značajke)	97.68% \pm 0.21%	92.99% \pm 0.45%	75.36% \pm 2%
Diskriminator	99.21% \pm 0.07%	91.14% \pm 2.6%	71.27% \pm 1.64%

Table: Rezultati mjerenja za Logističku regresiju

Klasifikator	Cijeli MNIST	1000 primjera	100 primjera
LR (pikseli)	91.98% \pm 0%	85.85% \pm 0%	73.15% \pm 0%
LR (RND značajke)	93.98% \pm 0.51%	85.11% \pm 1%	67.69% \pm 1.86%
LR (DCGAN značajke)	97.93% \pm 0.21%	93.74% \pm 0.44%	74.16% \pm 2.24%
Diskriminator	99.21% \pm 0.07%	91.14% \pm 2.6%	71.27% \pm 1.64%

Primjena na FM3 datasetu - model 4



Figure: Uzorci nakon 0, 25000, 50000, 100000 iteracija

Primjena na FM3 datasetu - model 4

Table: Rezultati mjerenja preciznosti

	0	1	2	3	4	5	6	7
pikseli	91.05%	63.52%	95.56%	71.43%	72.50%	33.33%	65.96%	66.67%
značajke	95.74%	69.41%	95.90%	92.31%	76.92%	78.05%	72.00%	65.38%

Table: Rezultati mjerenja odziva

	0	1	2	3	4	5	6	7
pikseli	95.78%	68.77%	97.42%	89.74%	70.80%	24.56%	55.17%	55.93%
značajke	97.11%	65.80%	98.06%	92.31%	75.60%	56.14%	62.07%	55.93%

Klasifikatorska generativna suparnička mreža

- Identična priča, promjena što diskriminator ima 2 izlaza
- Stoga postoje i 2 gubitka
- Funkcije izglednosti izgledaju:

$$\begin{aligned} L_S &= \mathbb{E}[\log P(S = \textit{real}|X_{\textit{real}})] + \mathbb{E}[\log P(S = \textit{fake}|X_{\textit{fake}})] \quad (9) \\ L_C &= \mathbb{E}[\log P(C = c|X_{\textit{real}})] + \mathbb{E}[\log P(C = c|X_{\textit{fake}})] \end{aligned}$$

- L_S predstavlja log izglednost uzorkovanja diskriminatora
- L_C klasifikacijsku log izglednost
- Diskriminator maksimizira vrijednost $L_S + L_C$ dok generator maksimizira $L_C - L_S$.

Klasifikatorska generativna suparnička mreža

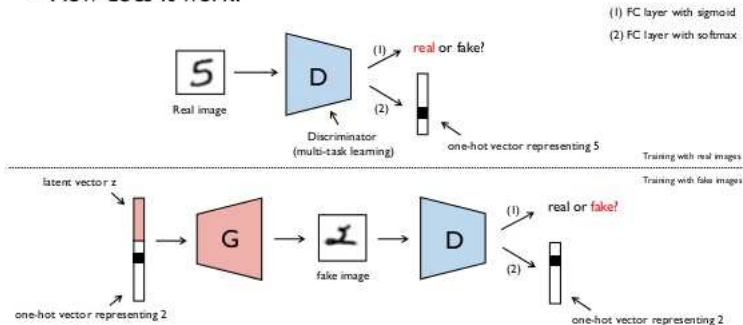
ACGAN



Variants of
GAN



- How does it work?



49

Figure: Klasifikatorska generativna suparnička mreža

Rezultati

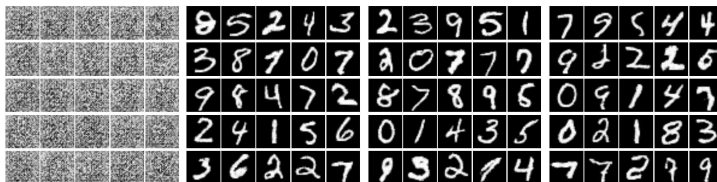


Figure: Uzorci nakon 0, 25000, 50000, 100000 iteracija

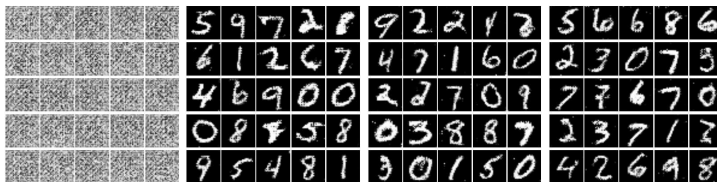


Figure: Uzorci nakon 0, 25000, 50000, 100000 iteracija (otežan)

Table: Rezultati mjerenja za klasifikatore

Klasifikator	Cijeli MNIST	1000 primjera	100 primjera
AC-GAN	98.56%	75.48%	16.4%
Weighted AC-GAN	99.21%	88.65%	22.13%
Diskriminator	99.2%	91.89%	75.4%

- Interesantno područje
- Moguće primjene kod zadataka s malo označenih primjera
- Nove varijante na tjednoj bazi
- Prema stabilnijim treniranjima arhitektura (WGAN)
- Teoretske analize nestabilnosti
- Kriteriji konvergencije

Hvala na pozornosti!