Generative Adversarial Nets

Radek Bartyzal

Let’s Talk ML

15. 2. 2017

Introduction

- deep learning successes
  - mainly **discriminative** models

    = mapping high dimensional input to classes

    = using backpropagation, dropout … + units with nice gradient

- deep learning difficulties
  - **generative** models

    = difficult computations during maximum likelihood estimation and similar strategies

    -> GAN to the rescue

Adversarial nets framework

- generative model G - discriminative model D

- G(z = noise) -> data space
- D(data sample x) -> [0-1] = prob. that x is from training data

- goal
  = G recovers training data distribution: **p_g = p_d**
  = D(x) = 0.5 everywhere = cannot tell generated data from training ones

Adversarial nets framework

- two-player minimax game with value function V(G, D):

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

- D(x) = prob. that x from training data is from training data
  - G wants to minimize this while D maximize
- D(G(z)) = prob. that sample generated by G is from training data
  - G wants to maximize this => minimize [1 - D(G(z))]

Training

- both models = MLP

  => training can be done using only backprop and dropout

- k steps of training the D -> 1 step of training the Generator

  => Discriminator is always near its optimal solution if Generator improves slowly enough

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

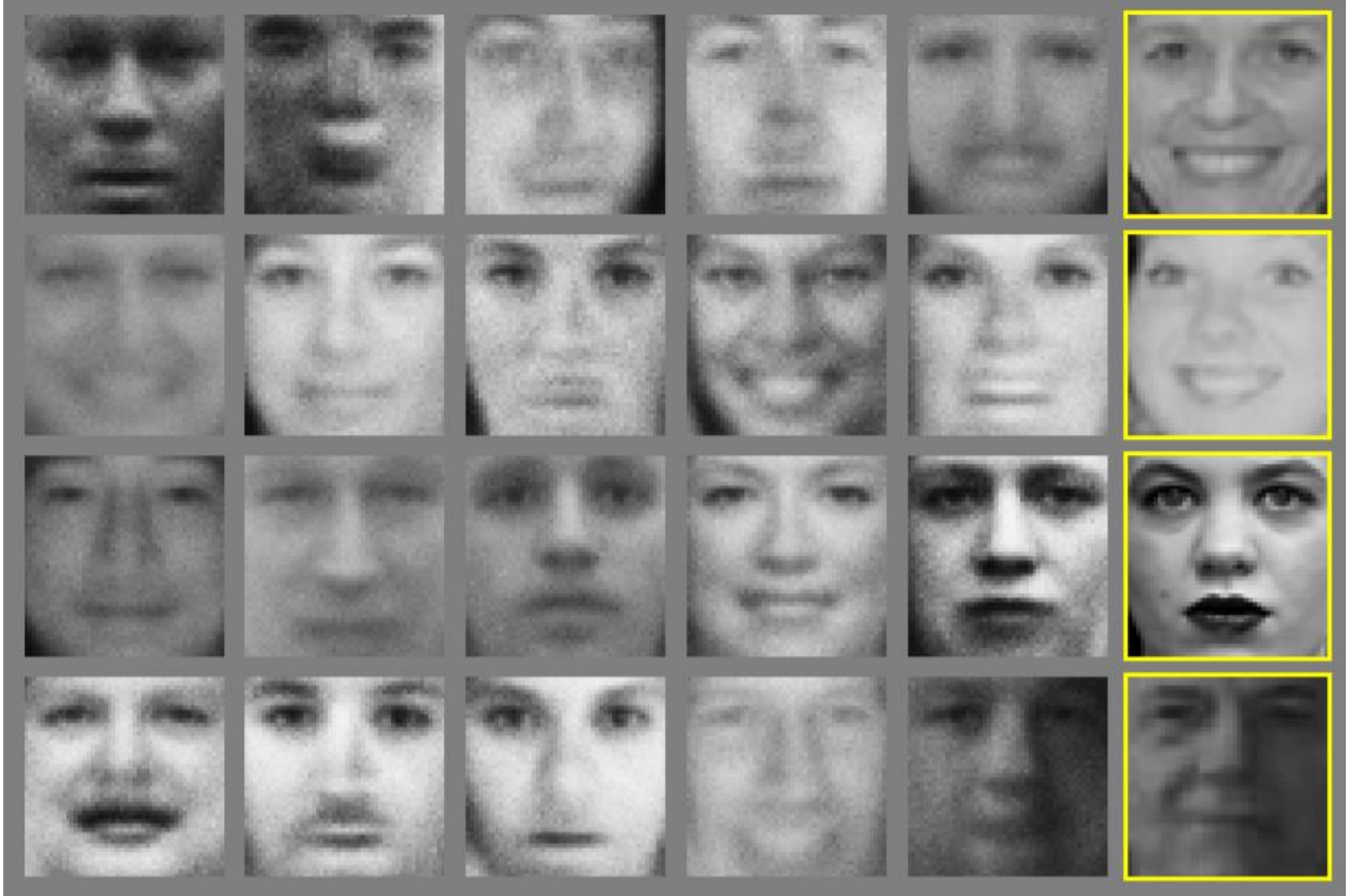
$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for



Sources

- https://arxiv.org/abs/1406.2661 = original GAN paper by Ian Goodfellow from 2014