Nikita Kazeev

Generative Adversarial Network (GAN)

2021

Generative models in general

- Want:
  - A neural network generator. When applied to noise z, samples the target distribution: $G(z) \sim P(x)$
  - OR a neural network approximation of probability density
    - Not covered now, Artem Ryzhikov will teach them in a few sections
- Training:
  - Minimize some divergence $D\left(P_{\text{train}} || Q_{\text{generated}}\right)$
- Challenges:
  - Both the training and generated distributions are not known explicitly

Consider an f-divergence

Can be computed directly

$$D_f(P\|Q) = \underbrace{\int dx \cdot q(x)}_{\text{Can be estimated by sampling}} \cdot f\left(\overbrace{\frac{p(x)}{q(x)}}^{\text{Can be estimated by a classifier}}\right)$$

Spherical generative adversarial network in vacuum

Repeat until convergence:

- Fit a classifier to estimate $\frac{p(x)}{q(x)}$
- Compute the value of an f-divergence between the real and generated data via Monte-Carlo sampling
- Train the generator: take a gradient descent step to minimize the distance

Challenges:

- Finite training data
- Finite computing resources

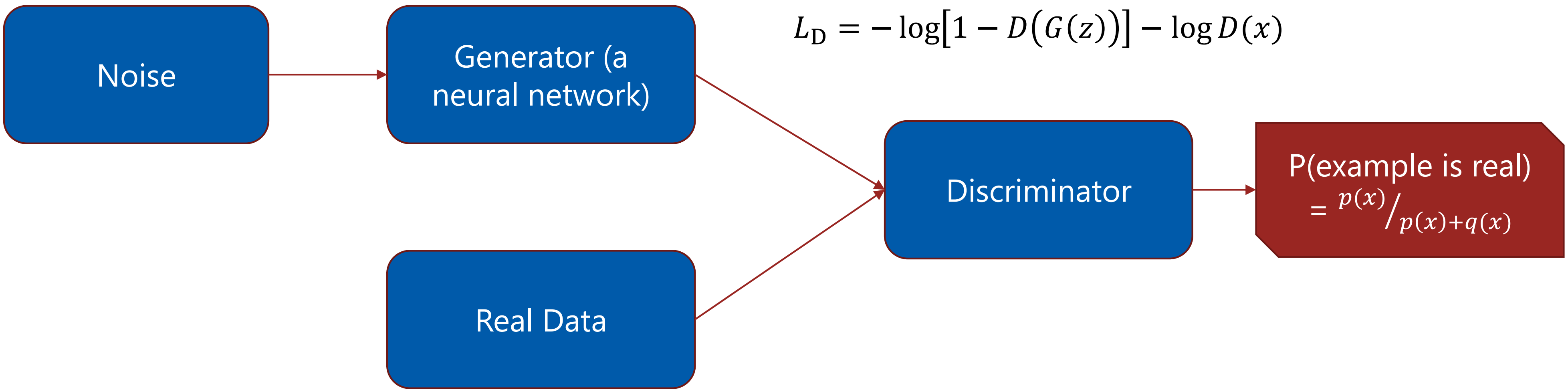
Divergence selection

- Not KL, it is infinite if $\exists x: p(x) \neq 0, q(x) = 0$
  - For a complex distribution (e. g. images), it is almost certain to happen
- Most of the papers on GANs with f-divergencies use JS
  - It works well enough
  - No tangible benefit from using other divergencies
  - See a review paper: https://arxiv.org/pdf/1606.00709.pdf

JS GAN scheme

$$L_{\mathrm{G}} = -\log D(G(z))$$

$$L_{\mathrm{D}} = -\log\left[1 - D\big(G(z)\big)\right] - \log D(x)$$

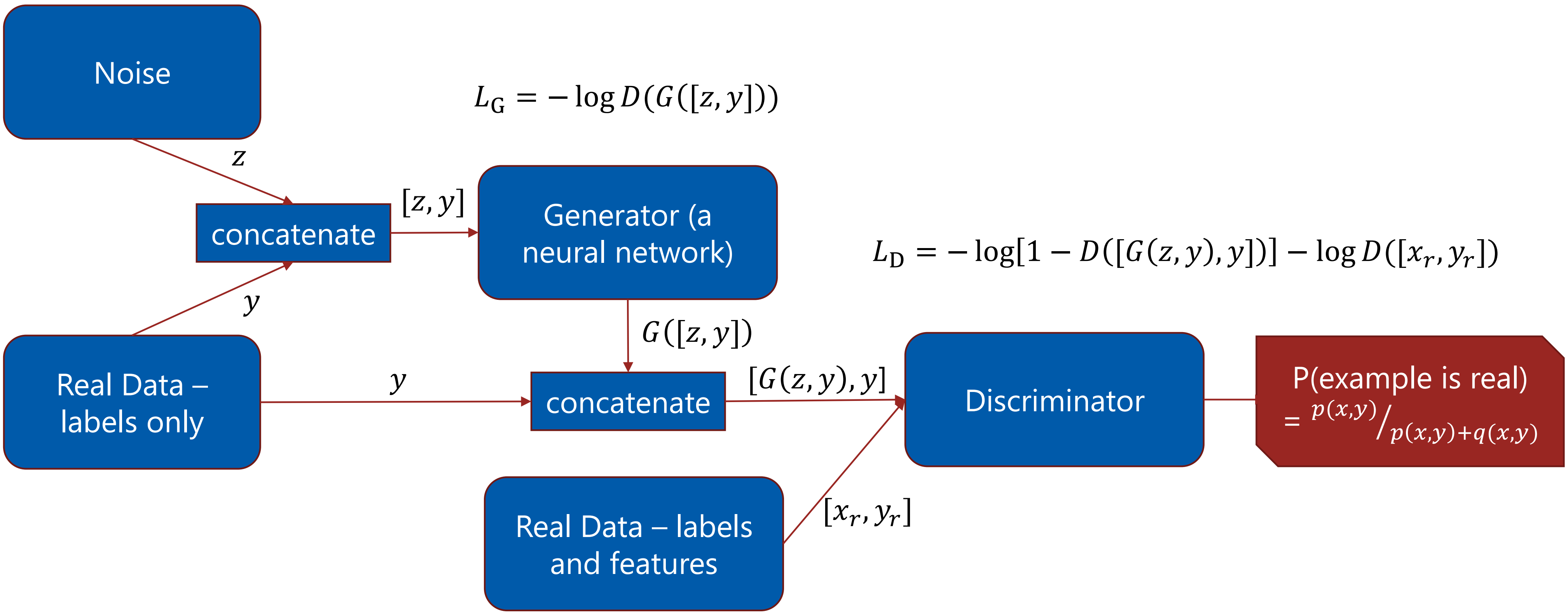


Training algorithm

Repeat while not good enough

- Train discriminator for N_{d} iterations:
  - sample a batch of training data x_r
  - sample a batch of noise z
  - compute discriminator loss $L_{\mathrm{D}} = -\log\left[1 - D\left(G(z)\right)\right] - \log D(x_r)$
  - take an optimization step for discriminator, minimizing L_{D}
- Train generator
  - sample a batch of noise z
  - compute generator loss $L_{\mathrm{G}} = -\log D(G(z))$
  - take an optimization step for generator, minimizing L_{G}

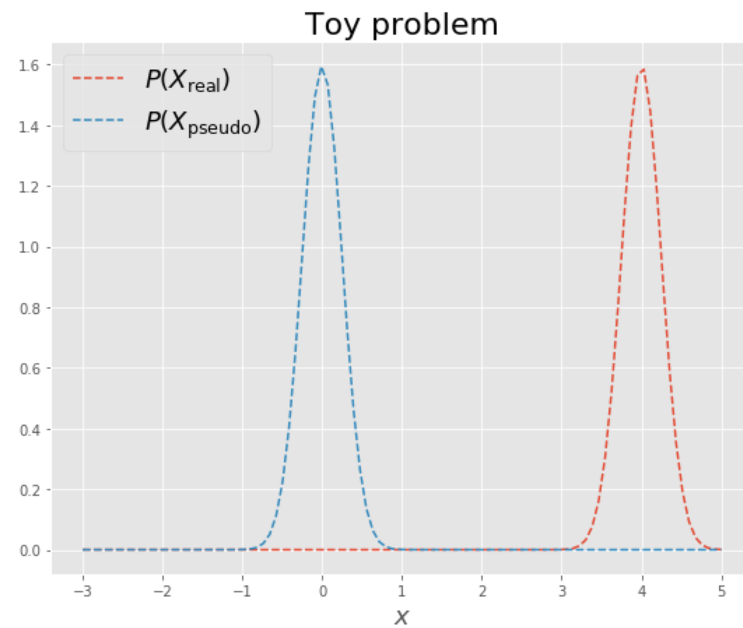
Conditional GAN scheme



Problem #1: vanishing gradients

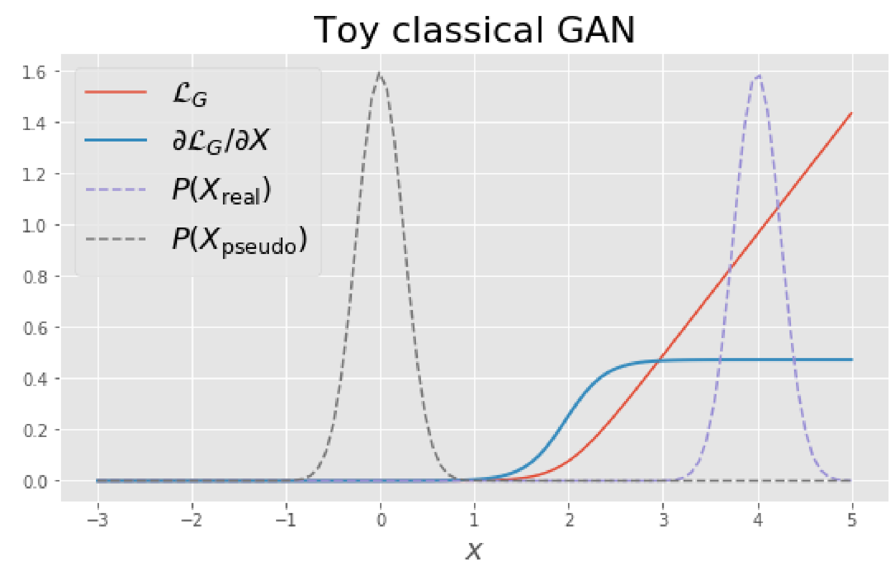
- Consider the case of disjoint support of real and generated data
- An ideal discriminator can perfectly tell the real and generated data apart:

$$D(G(z)) \approx 0$$



Problem #1: vanishing gradients

- $L_{\mathrm{G}} = -\log D(G(z))$
- ${dD(x)}/{dx} \approx 0$ for generated x
- ${dL_{\mathrm{G}}(x)}/{dx} \approx 0$ for generated x
- Generator can't train!



Fight for the gradients

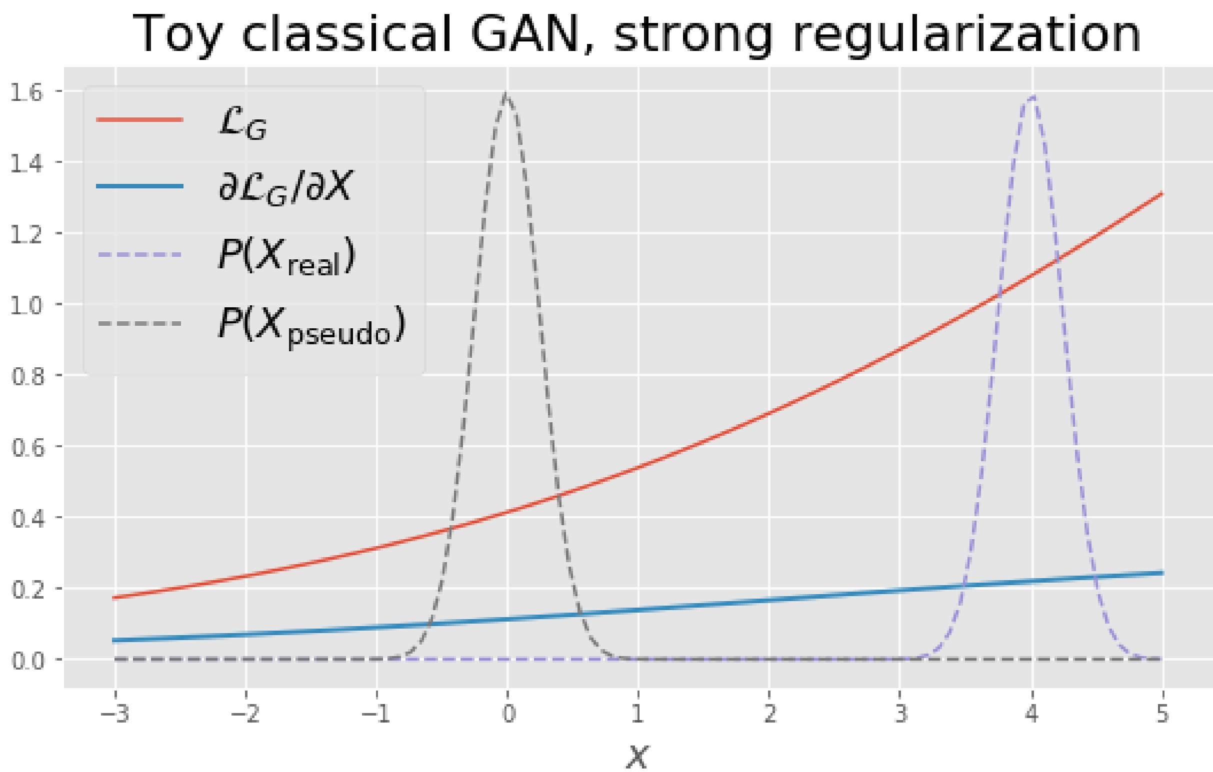
Start with heavily restricted discriminator:

- don't train discriminator fully
- add noise to the samples:
  - nicely works for target on low-dimensional manifolds;
  - easy to control.
- discriminator regularization:
  - might interfere with the convergence.

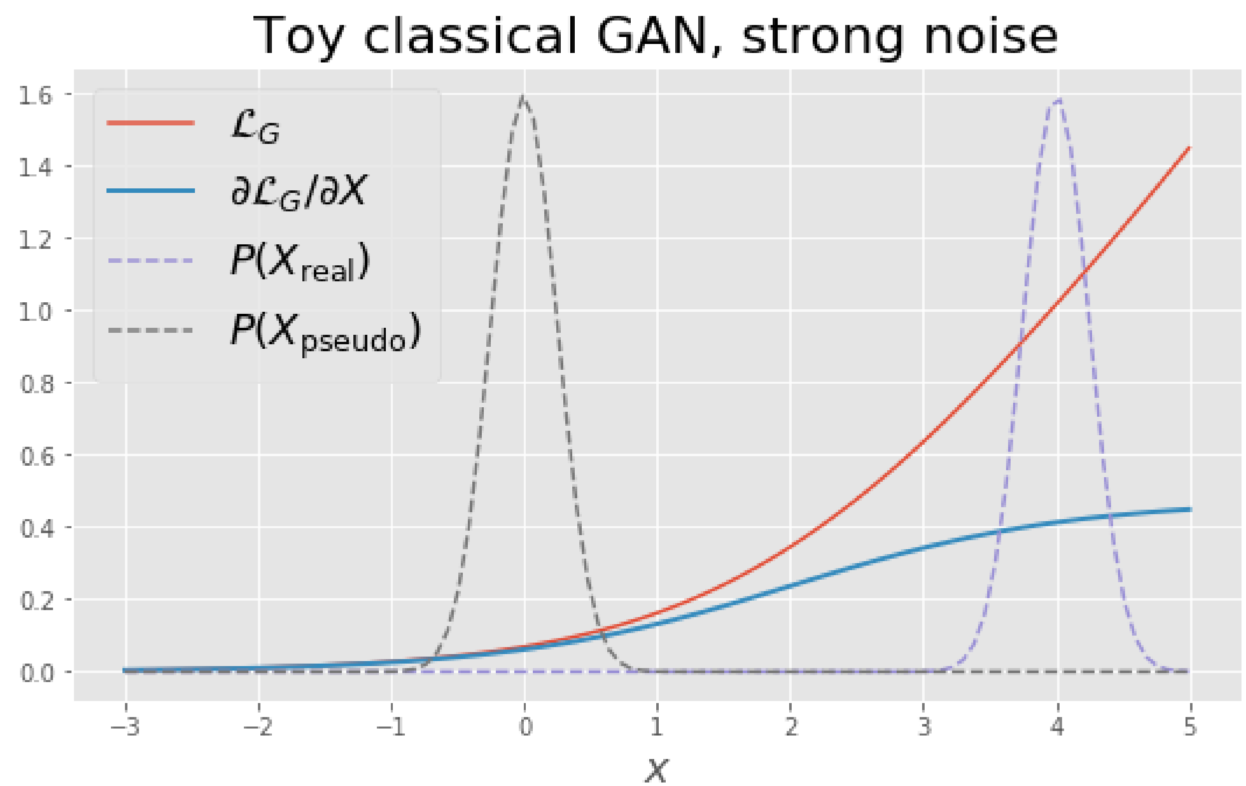
As learning progresses gradually relax restrictions.

(or use a different class of divergencies – see the next lecture)

Fight for the gradients: regularization



Fight for the gradients: noise



Problem #2: mode collapse

- Generator only trains on the objects it generates:
$$L_{\mathrm{G}} = -\log D(G(z))$$
- If there are different modes in the data distribution, the generator can get stuck in a local minimum of the discriminator
- Example. If a generator has learned to generate only “1”, but perfectly, it might fail to learn to generate “5”

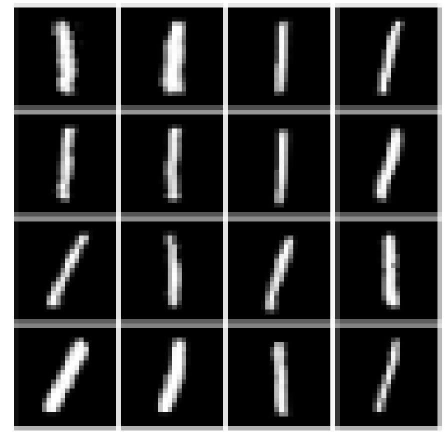


Image: Adiga, Sudarshan, et al. "On the tradeoff between mode collapse and sample quality in generative adversarial networks." 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP). IEEE, 2018.

Summary

- Classic generative adversarial networks with JS loss
  - Powerful method for generative modelling
  - Hard to train
    - Need to balance discriminator and generator power
    - Vanishing gradients
    - Mode collapse
  - Are the starting point of many image-specific advances
- See the next lecture for a newer approach based on integral probability measures

Thank you!

nkazeev@hse.ru

kazeevn

hse_lambda

Nikita Kazeev

Acknowledgements

The presentation is based on the previous MLHEP editions: vanishing gradients slides my Maxim Borisyak