ТЕХНОСФЕРА

Generative adversarial networks

Part 2

Галков Михаил

22 апреля 2017 г.

План лекции

Применения GAN

- DCGAN
- Domain transfer network
- SRGAN
- Text to image (StackedGAN)
- Image to image (cGAN)

Проблема плохих градиентов и подходы к решению

- LSGAN
- WGAN
- BEGAN

DCGAN

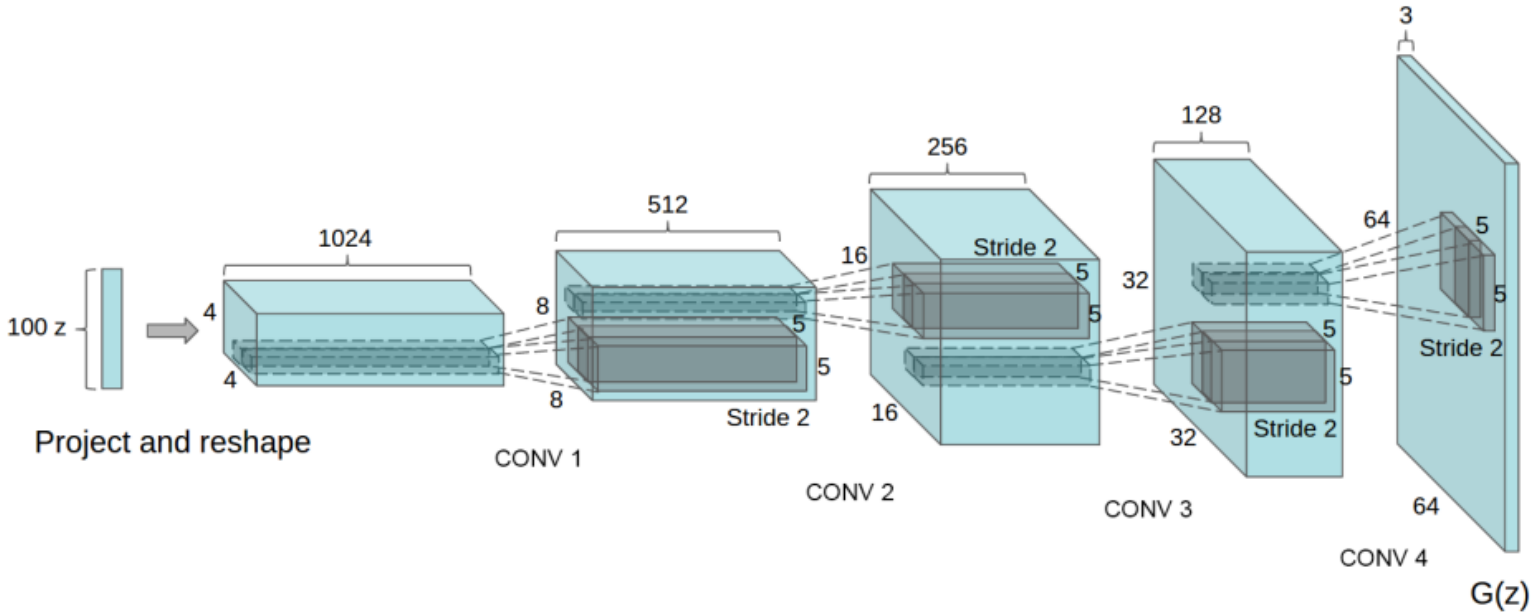
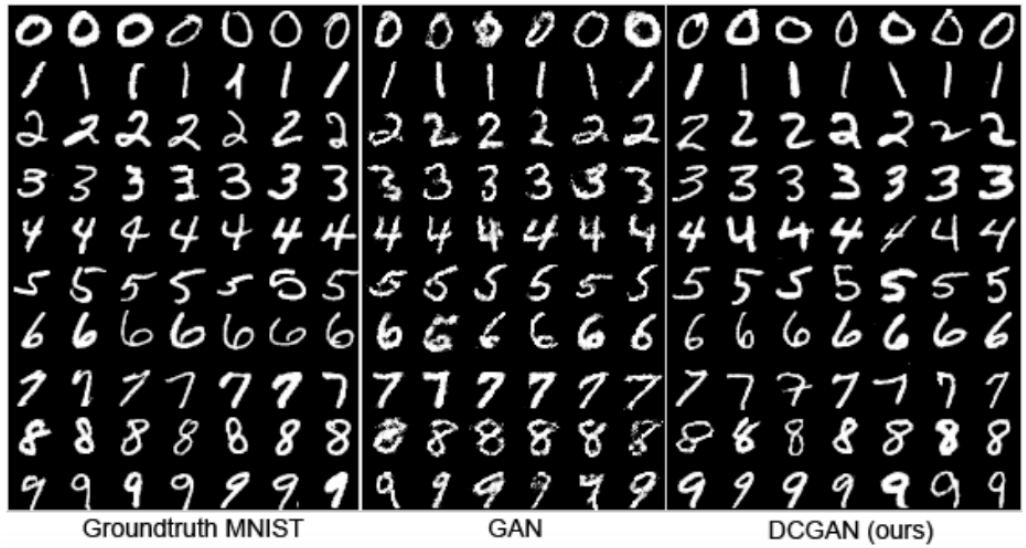


Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64×64 pixel image. Notably, no fully connected or pooling layers are used.

Architecture guidelines for stable Deep Convolutional GANs

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

DCGAN results



DCGAN interpolation

Почему интерполяция возможна?

Domain transfer network

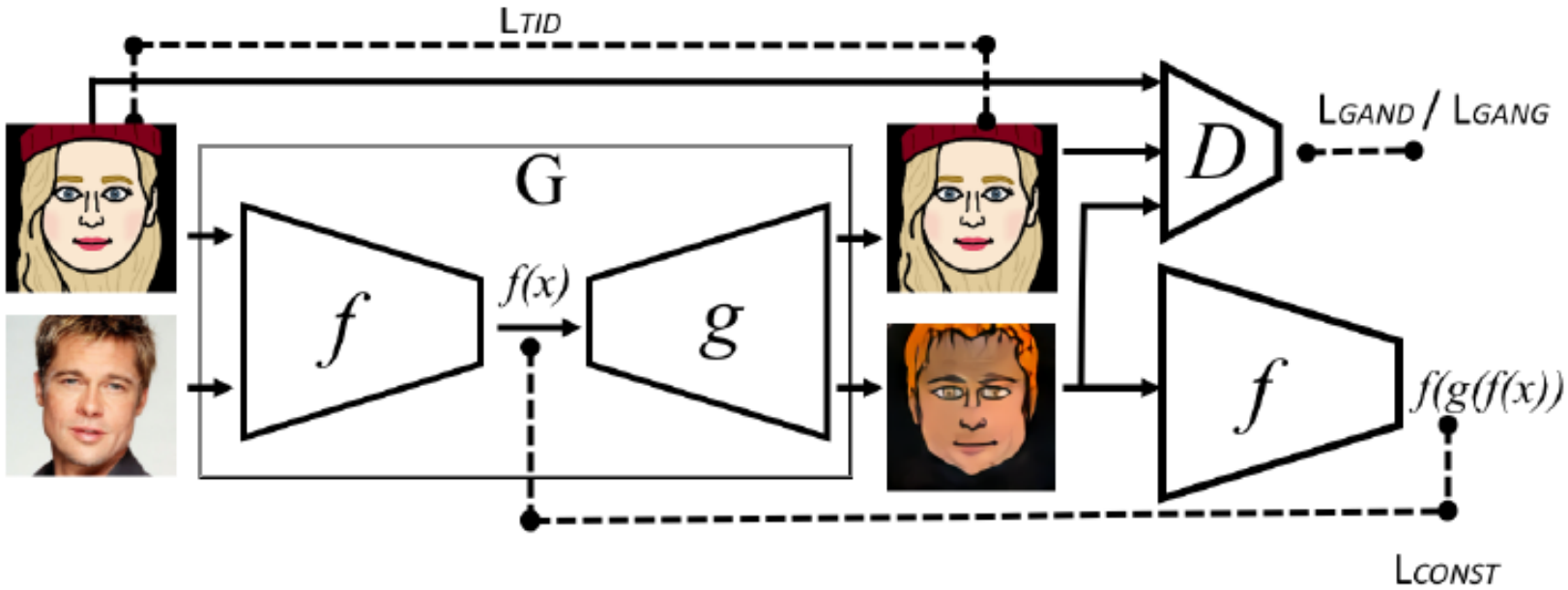


Figure 1: The Domain Transfer Network. Losses are drawn with dashed lines, input/output with solid lines. After training, the forward model G is used for the sample transfer.

$$L_{\mathrm{D}} = -\mathbb{E}_{x\in\mathbf{s}} \log D_1(g(f(x))) - \mathbb{E}_{x\in\mathbf{t}} \log D_2(g(f(x))) - \mathbb{E}_{x\in\mathbf{t}} \log D_3(x)$$

$$L_{\mathrm{GANG}} = -\mathbb{E}_{x\in\mathbf{s}} \log D_3(g(f(x))) - \mathbb{E}_{x\in\mathbf{t}} \log D_3(g(f(x)))$$

$$L_{\mathrm{CONST}} = \sum_{x\in\mathbf{s}} d(f(x), f(g(f(x))))$$

$$L_{\mathrm{TID}} = \sum_{x\in\mathbf{t}} d_2(x, G(x))$$

Results

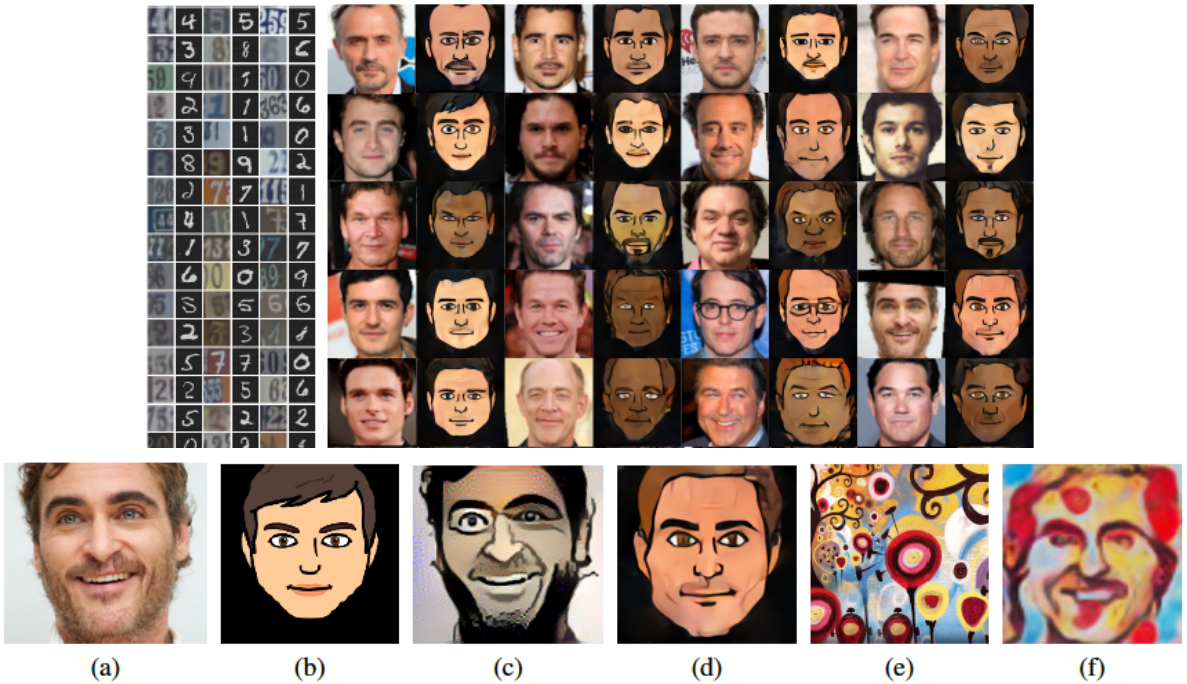
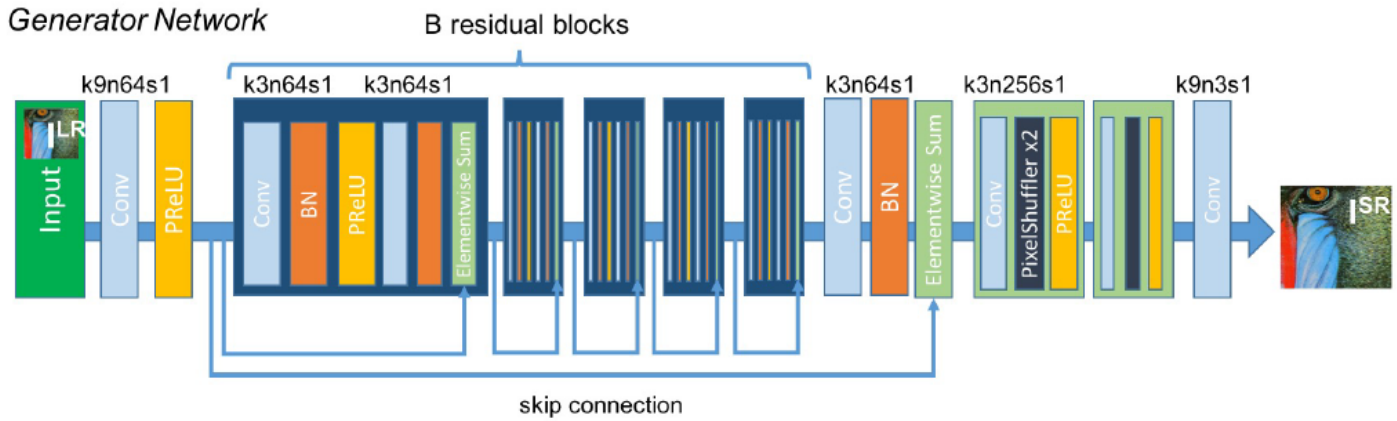
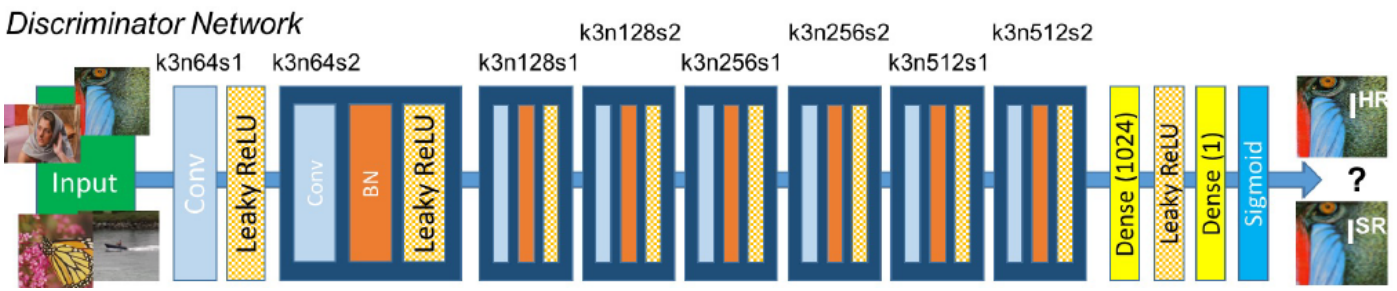


Figure 5: Style transfer as a specific case of Domain Transfer. (a) The input content photo. (b) An emoji taken as the input style image. (c) The result of applying the style transfer method of Gatys et al. (2016). (d) The result of the emoji DTN. (e) Source image for style transfer. (f) The result, on the same input image, of a DTN trained to perform style transfer.

Лекция по Style Transfer следующая, вы поймете почему GAN потенциально более "сильная"модель.

Super resolution GAN





$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I^{HR} \sim p_{\text{train}}(I^{HR})}[\log D_{\theta_D}(I^{HR})] +$$

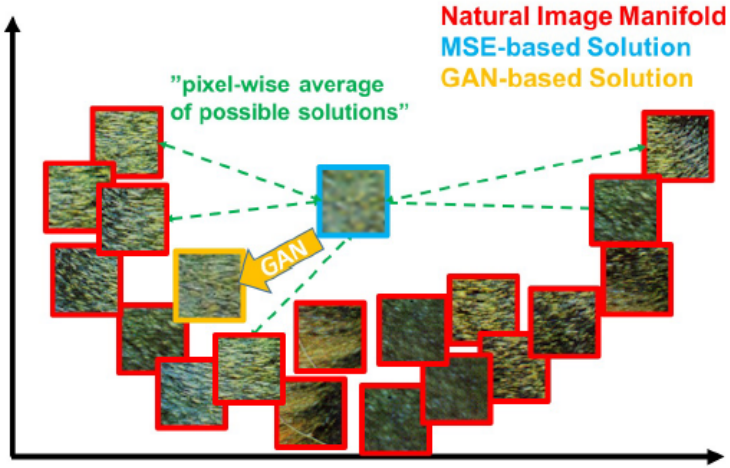
$$\mathbb{E}_{I^{LR} \sim p_G(I^{LR})}[\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR}))]$$

Можно ли решить как supervised задачу?

SRGAN

SRGAN
(21.15dB/0.6868)

original



Text to image

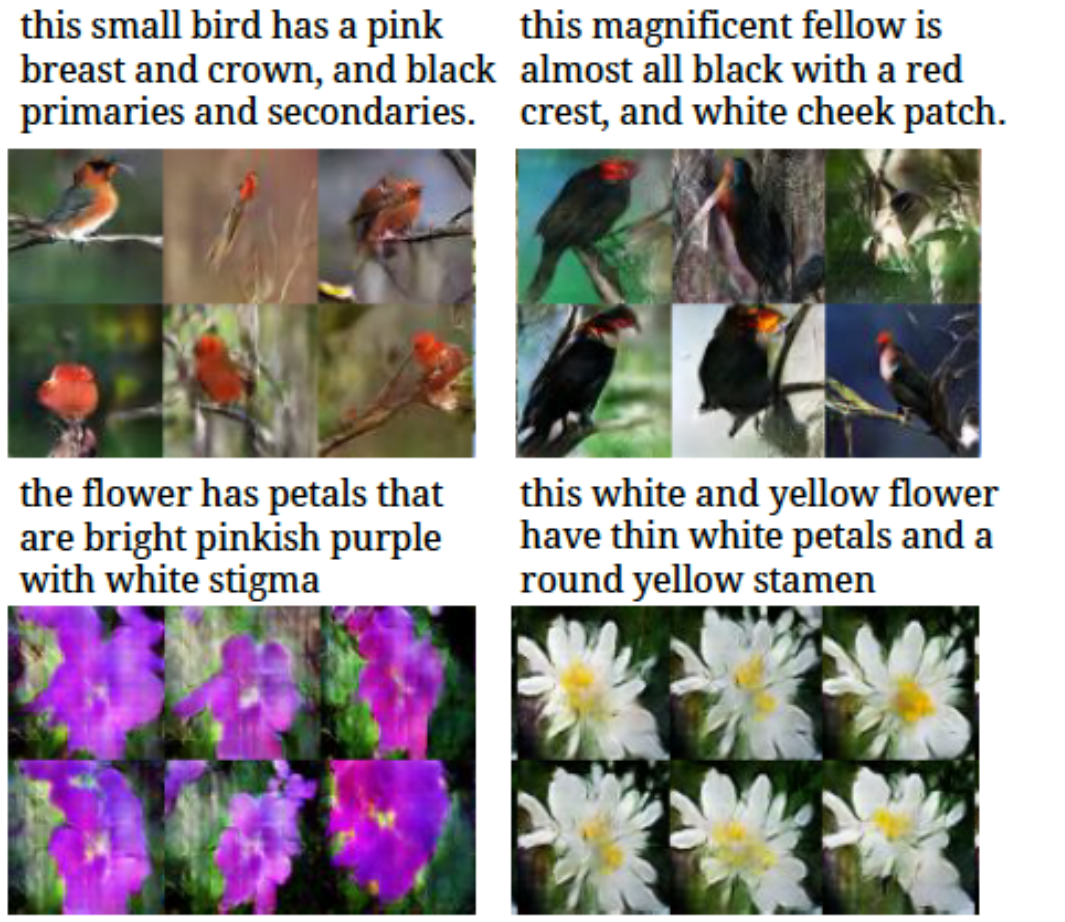
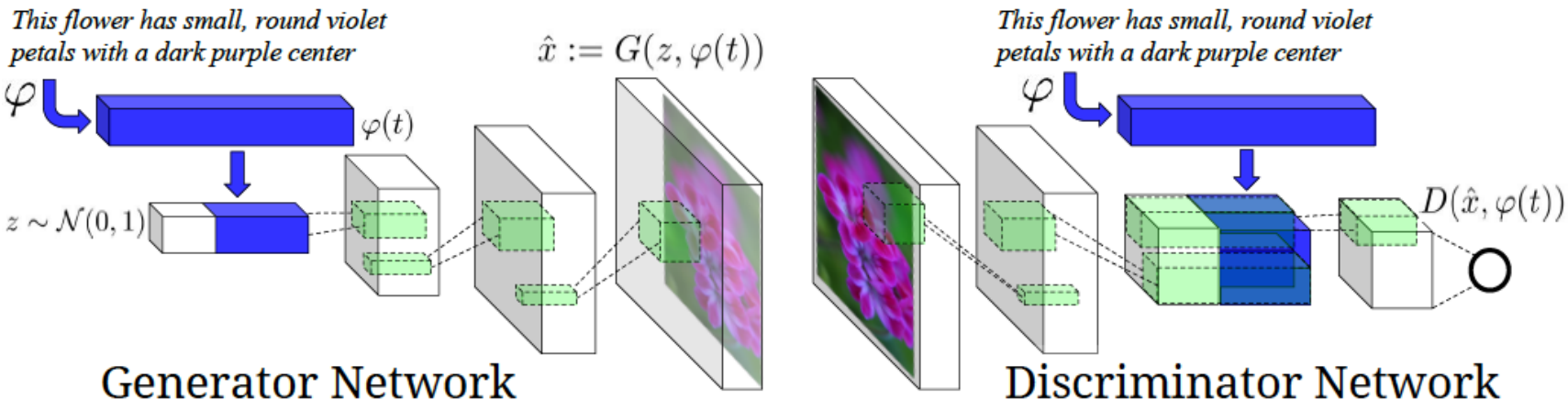


Figure 1. Examples of generated images from text descriptions. Left: captions are from zero-shot (held out) categories, unseen text. Right: captions are from the training set.

Text to image model



Text to image algo

Algorithm 1 GAN-CLS training algorithm with step size α, using minibatch SGD for simplicity.

Input: minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
for $n = 1$ **to** S **do**
  $h \leftarrow \varphi(t)$ {Encode matching text description}
  $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
  $z \sim \mathcal{N}(0, 1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z, h)$ {Forward through generator}
  $s_r \leftarrow D(x, h)$ {real image, right text}
  $s_w \leftarrow D(x, \hat{h})$ {real image, wrong text}
  $s_f \leftarrow D(\hat{x}, h)$ {fake image, right text}
  $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
  $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ {Update discriminator}
  $\mathcal{L}_G \leftarrow \log(s_f)$
  $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ {Update generator}
end for

Image to image

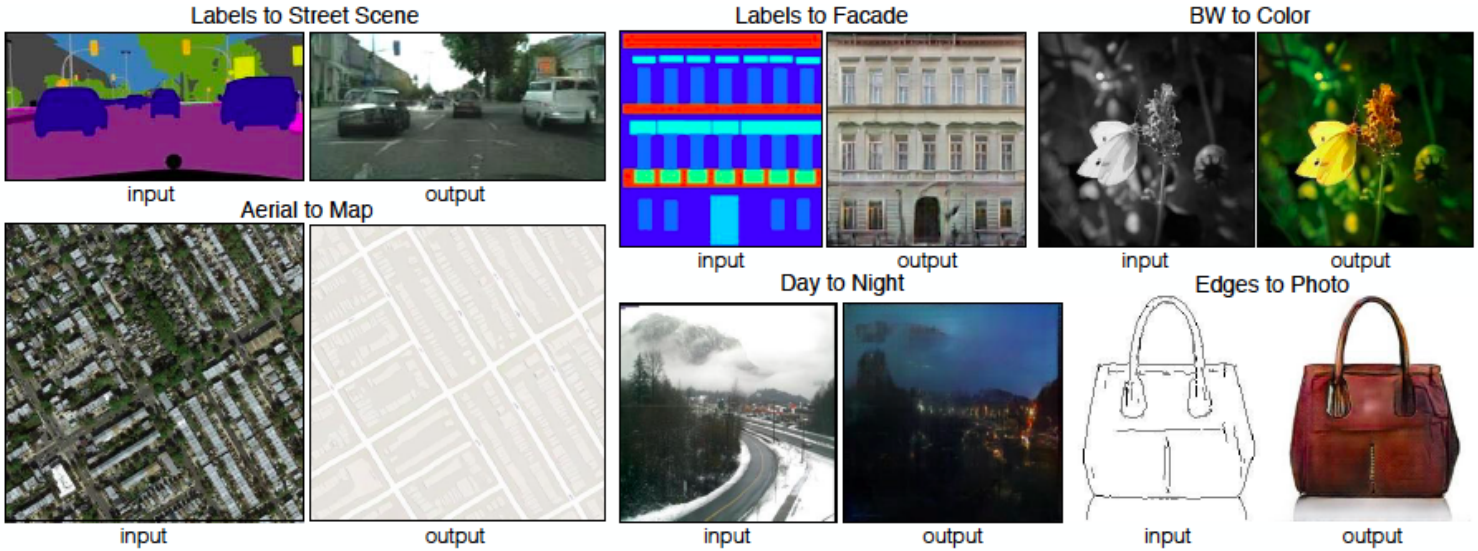


Image to image model

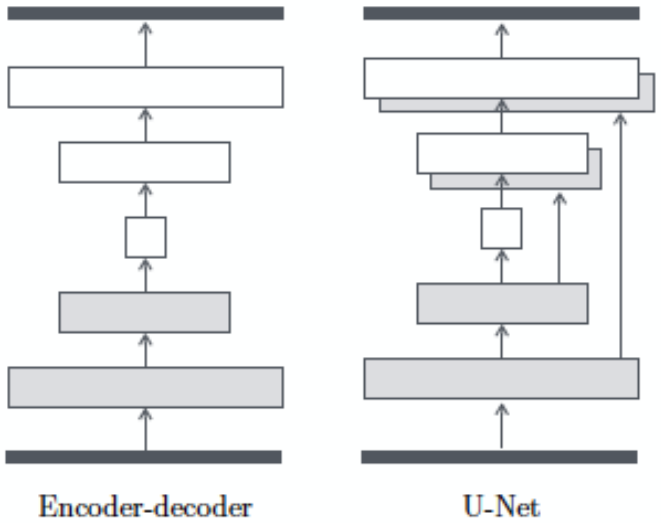
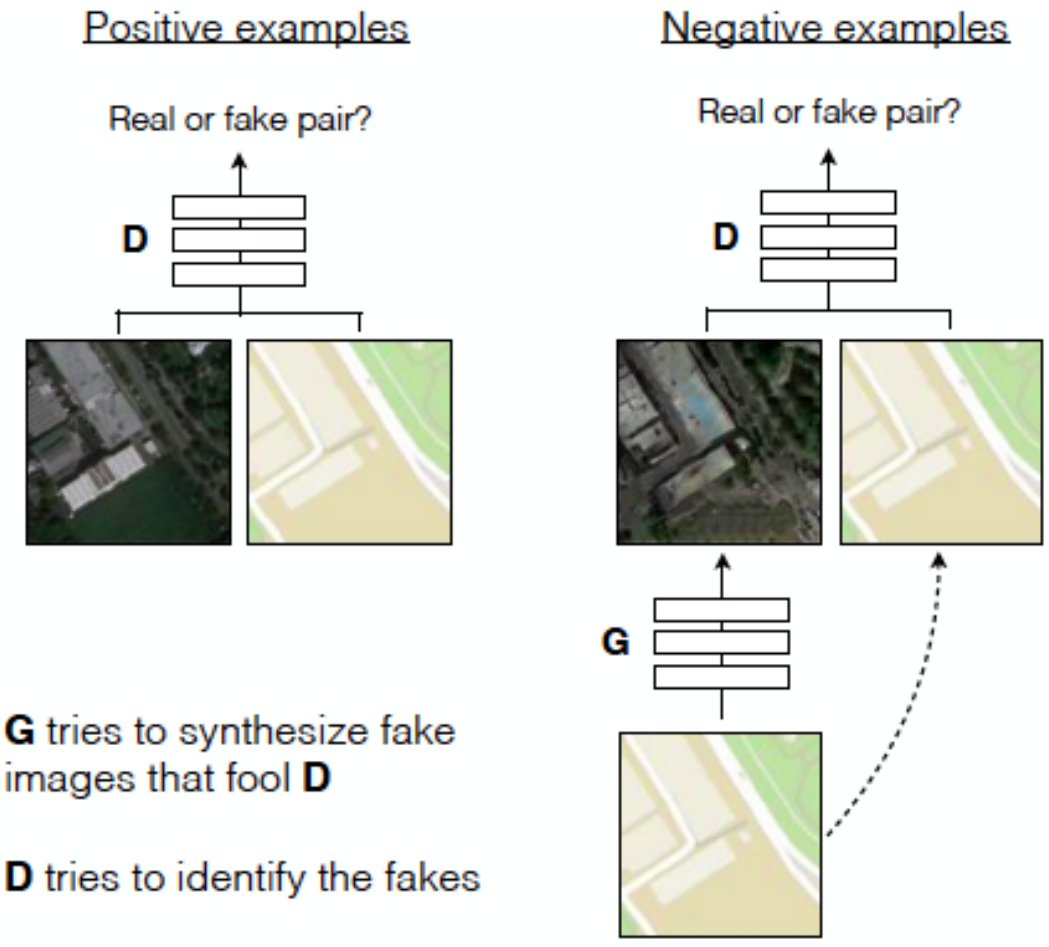
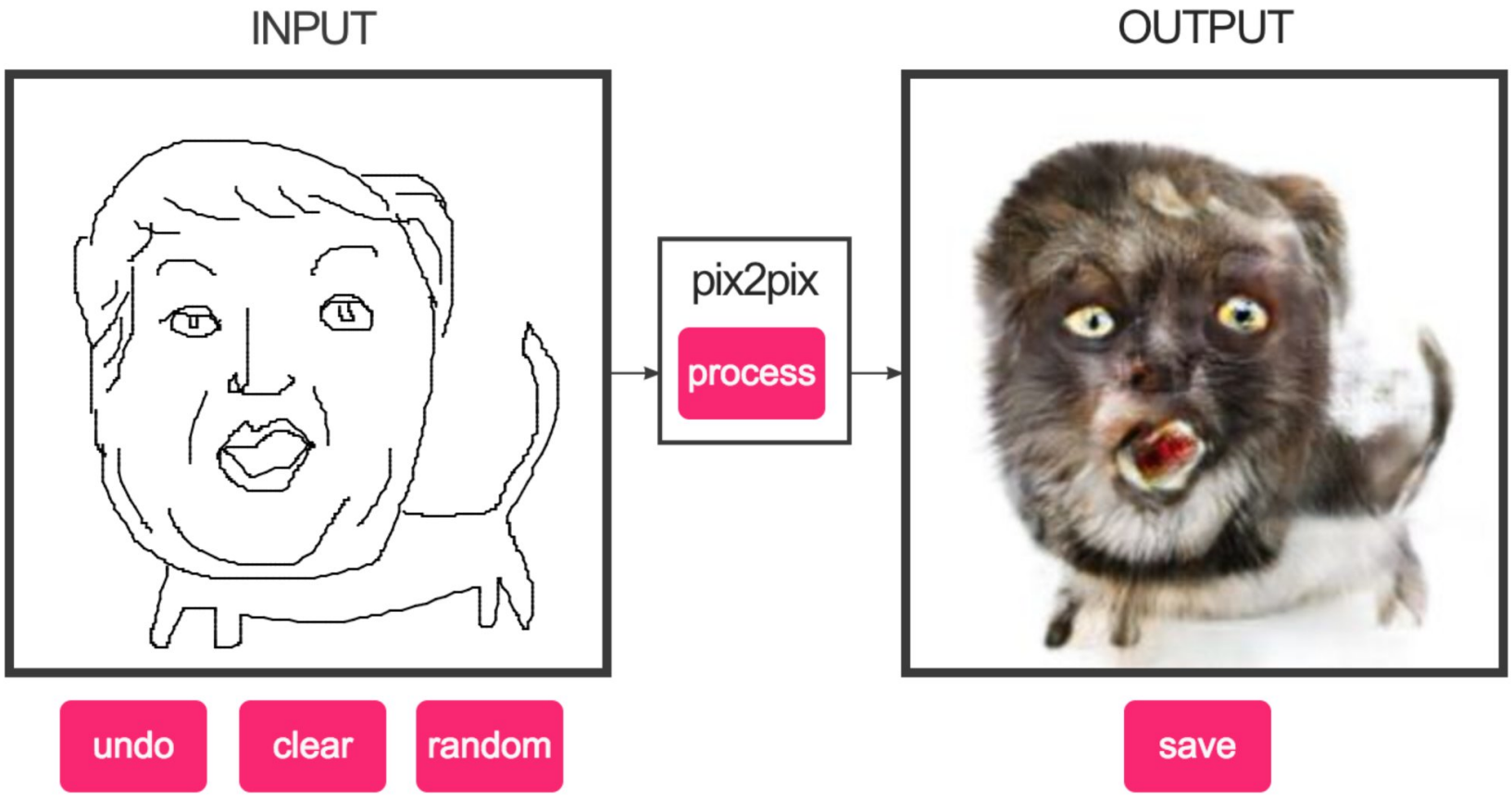


Figure 3: Two choices for the architecture of the generator. The "U-Net" [34] is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks.

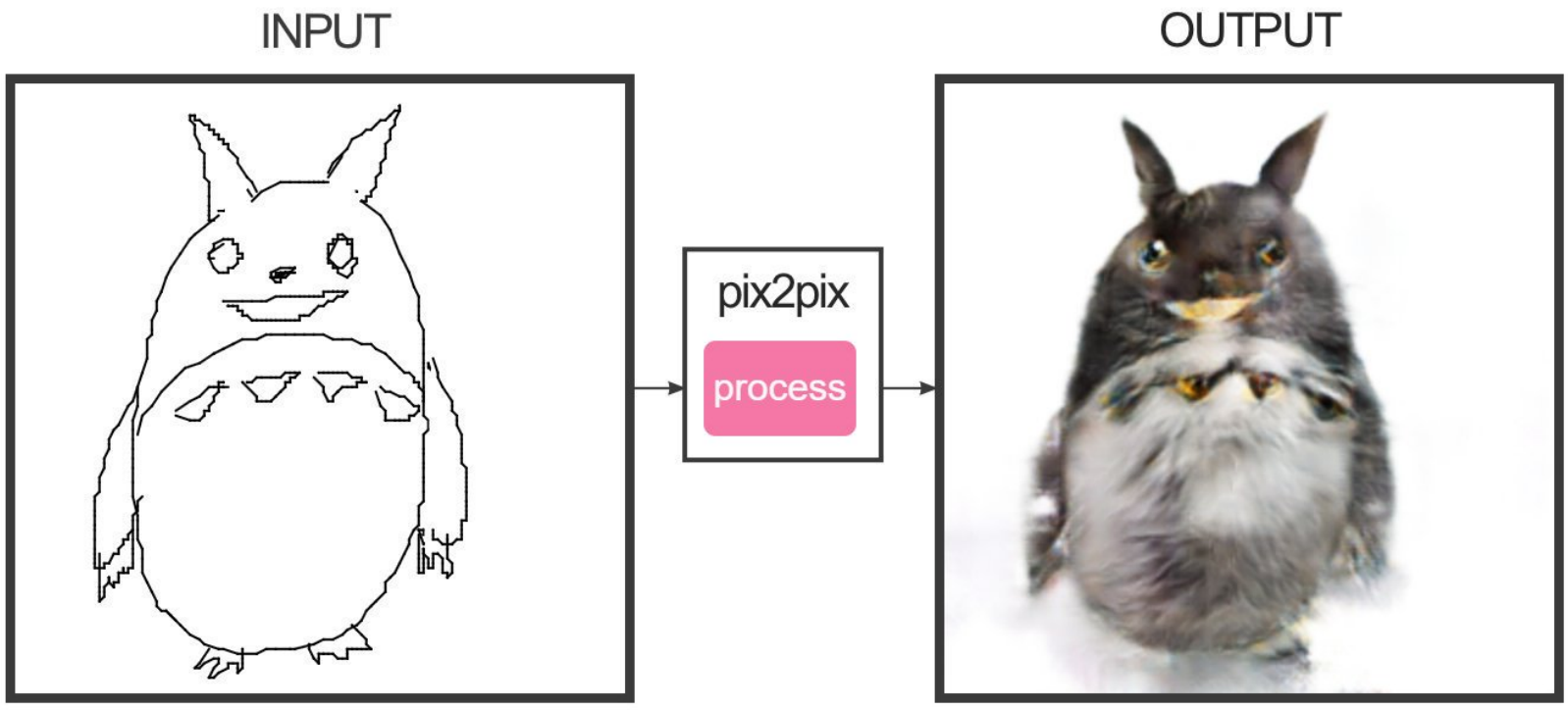
Image to image algo



MOAR IMAGES



MOAR IMAGES 2



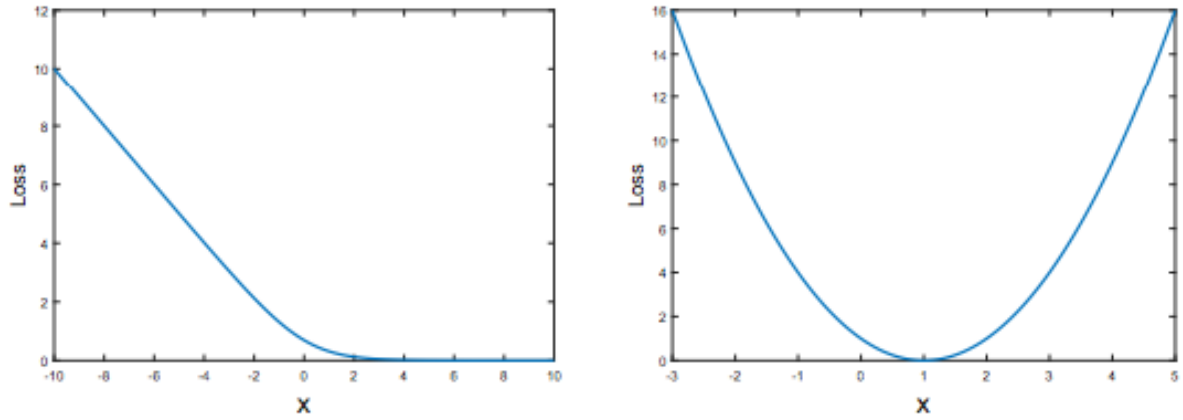
Problems

Visual

- Mode-collapsing
- Instability (long run)
- Noise generation (L1 лучше по началу)
- Bad converging (подбор гиперпараметров - боль)

Возможно, проблема в том, что мы оптимизируем что-то не то?

LSGAN



(a) (b)

Figure 2: (a): The sigmoid cross entropy loss function. (b): The least squares loss function.

$$\frac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x)) \tag{1}$$

LSGAN model

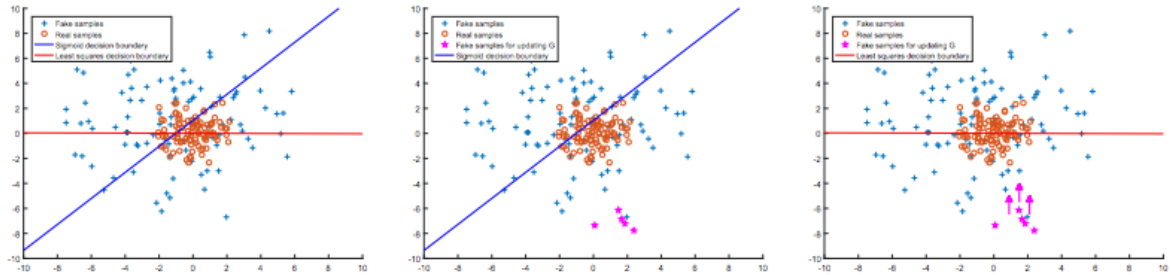


Figure 1: Illustration of different behaviors of two loss functions. (a): Decision boundaries of two loss functions. Note that the decision boundary should go across the real data distribution for a successful GANs learning. Otherwise, the learning process is saturated. (b): Decision boundary of the sigmoid cross entropy loss function. It gets very small errors for the fake samples (in magenta) for updateing G as they are on the correct side of the decision boundary. (c): Decision boundary of the least squares loss function. It penalize the fake samples (in magenta), and as a result, it forces the generator to generate samples toward decision boundary.

$$\min_D V_{\text{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\left[(D(\boldsymbol{x})-b)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z}))-a)^2\right]$$

$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z}))-c)^2\right],$$

LSGAN results

(a) Church outdoor.

(b) Dining room.

(c) Kitchen.

(d) Conference room.

LSGAN problems

Model collapsing

Optimizer	BN_G Adam	BN_G RMSProp	BN_{GD} Adam	BN_{GD} RMSProp
Regular GANs	YES	NO	YES	YES
LSGANs	**NO**	NO	YES	**NO**

Wasserstein distance

- The *Jensen-Shannon* (JS) divergence

$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r \| \mathbb{P}_m) + KL(\mathbb{P}_g \| \mathbb{P}_m) \ ,$$

  where $\mathbb{P}_m$ is the mixture $(\mathbb{P}_r + \mathbb{P}_g)/2$. This divergence is symmetrical and always defined because we can choose $\mu = \mathbb{P}_m$.

- The *Earth-Mover* (EM) distance or Wasserstein-1

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \big[\, \|x - y\| \, \big] \ , \tag{1}$$

  where $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_g$. Intuitively, $\gamma(x, y)$ indicates how much "mass" must be transported from x to y in order to transform the distributions $\mathbb{P}_r$ into the distribution $\mathbb{P}_g$. The EM distance then is the "cost" of the optimal transport plan.

Wasserstein > KL

Example 1 (Learning parallel lines). Let $Z \sim U[0,1]$ the uniform distribution on the unit interval. Let $\mathbb{P}_0$ be the distribution of $(0, Z) \in \mathbb{R}^2$ (a 0 on the x-axis and the random variable Z on the y-axis), uniform on a straight vertical line passing through the origin. Now let $g_\theta(z) = (\theta, z)$ with θ a single real parameter. It is easy to see that in this case,

- $W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|$,
- $JS(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0\ , \\ 0 & \text{if } \theta = 0\ , \end{cases}$

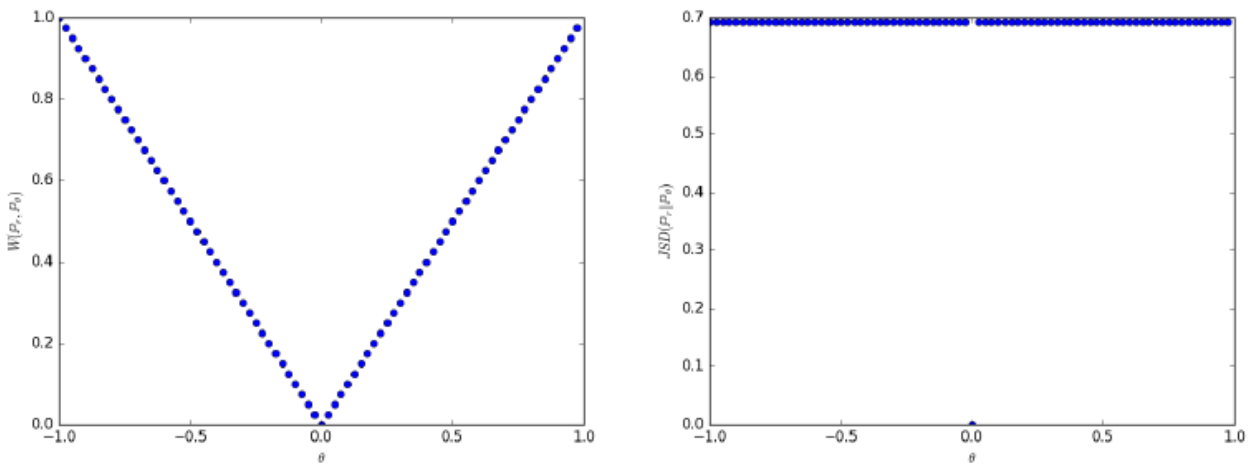


Figure 1: These plots show $\rho(\mathbb{P}_\theta, \mathbb{P}_0)$ as a function of θ when ρ is the EM distance (left plot) or the JS divergence (right plot). The EM plot is continuous and provides a usable gradient everywhere. The JS plot is not continuous and does not provide a usable gradient.

Счастье, радость, сходимость

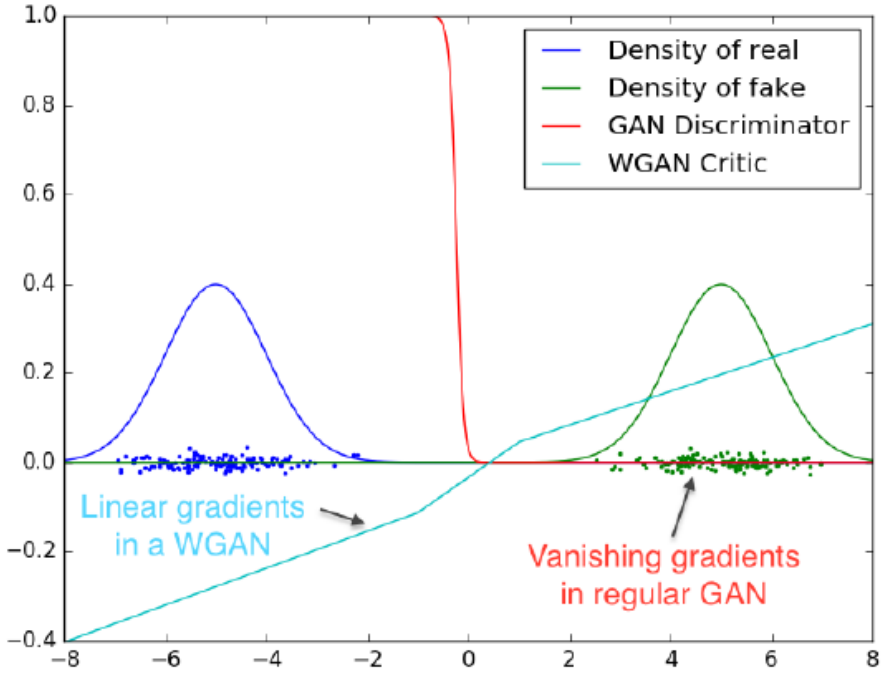


Figure 2: Optimal discriminator and critic when learning to differentiate two Gaussians. As we can see, the traditional GAN discriminator saturates and results in vanishing gradients. Our WGAN critic provides very clean gradients on all parts of the space.

Счастье, радость, сходимость 2

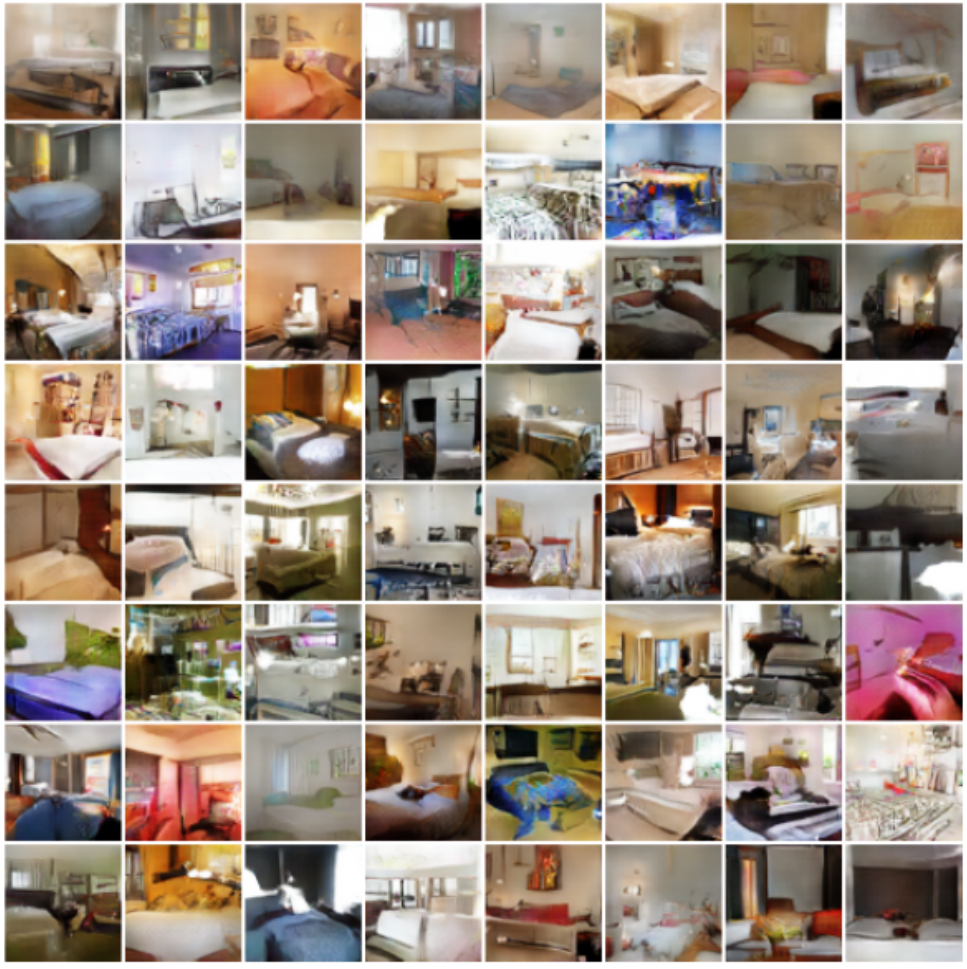


Figure 9: WGAN algorithm: generator and critic are DCGANs.

Вопросы