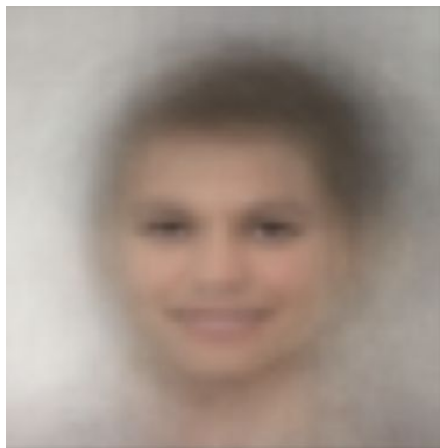




**ІТМО**

**Современные архитектуры  
нейронных сетей**

**GAN, WGAN, CGAN**

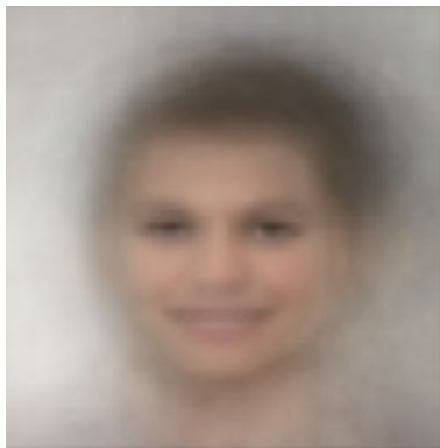


Генерація AE

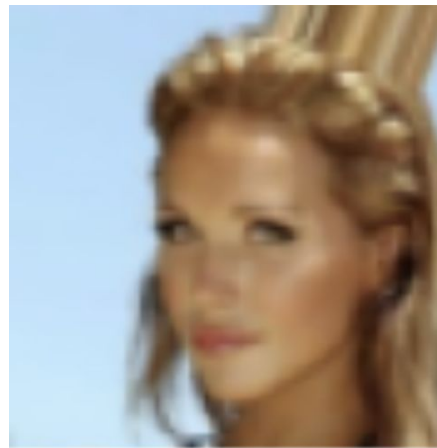


Генерація VAE





Сгенерированное  
изображение



Реальное  
изображение

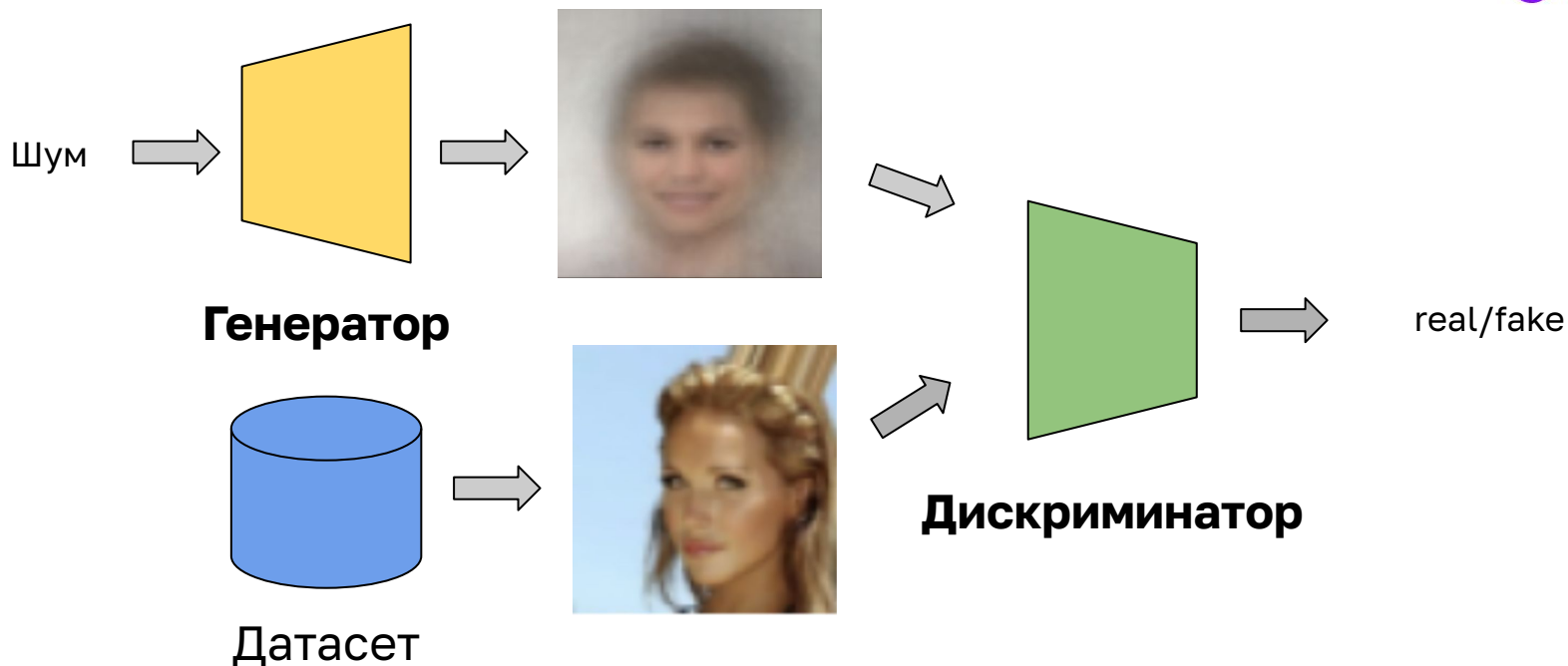


# Stack more layers - плохое решение ИТМО

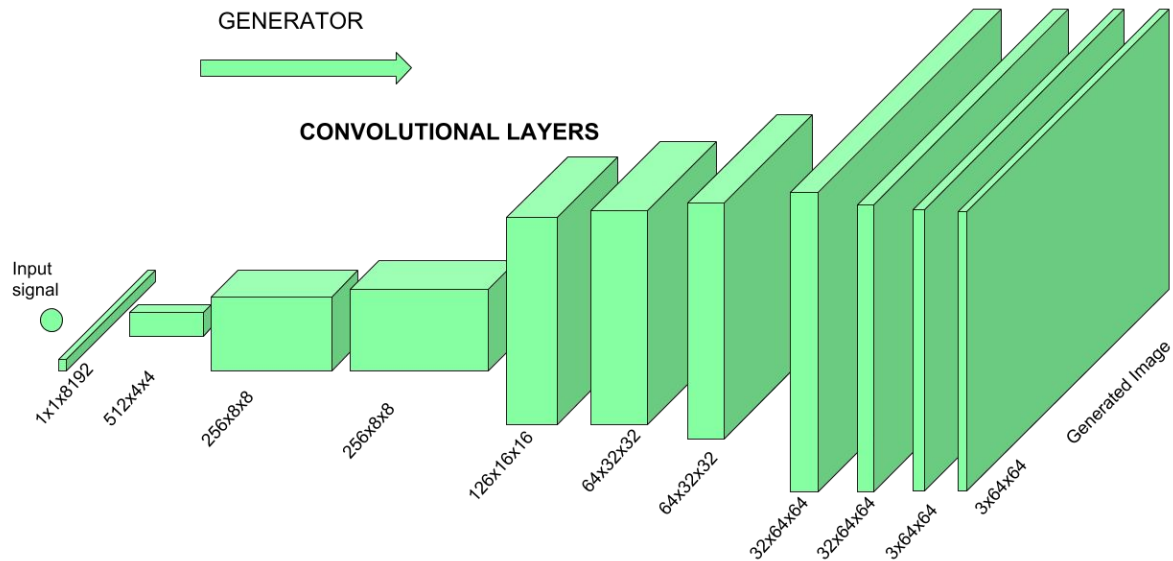


# Классификация сгенерированных изображений

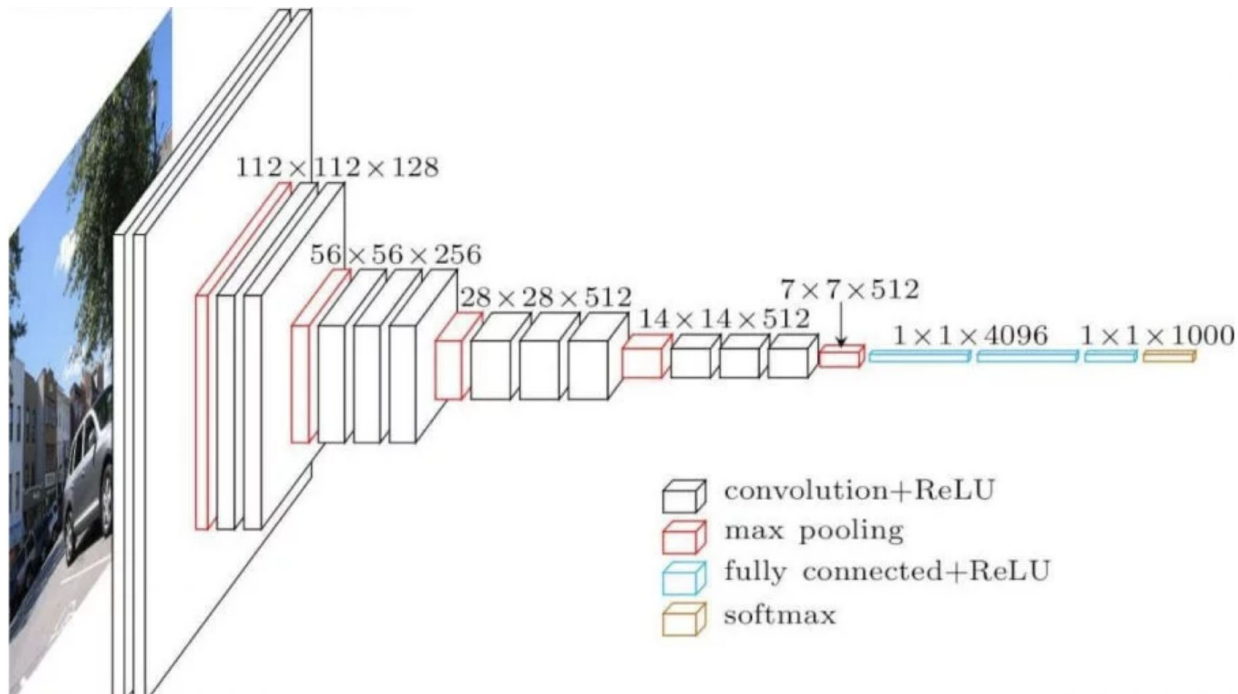
ІТМО



# Генератор

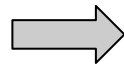
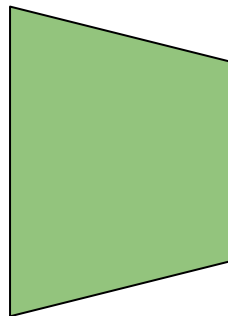
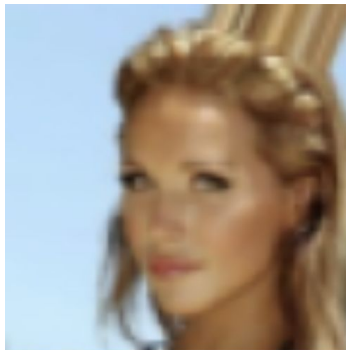


# Дискриминатор



# Обучение дискриминатора. Шаг 1

ІІТМО



BCELoss,  
label = 1

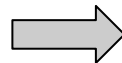
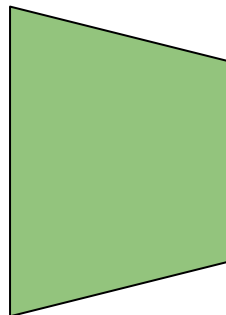
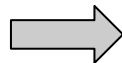
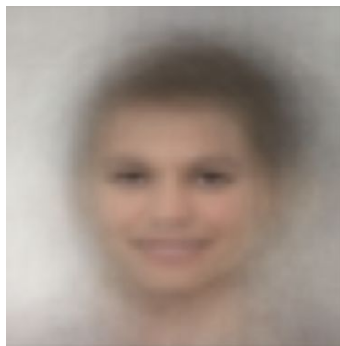
**Дискриминатор**





# Обучение дискриминатора. Шаг 2

ІІТМО



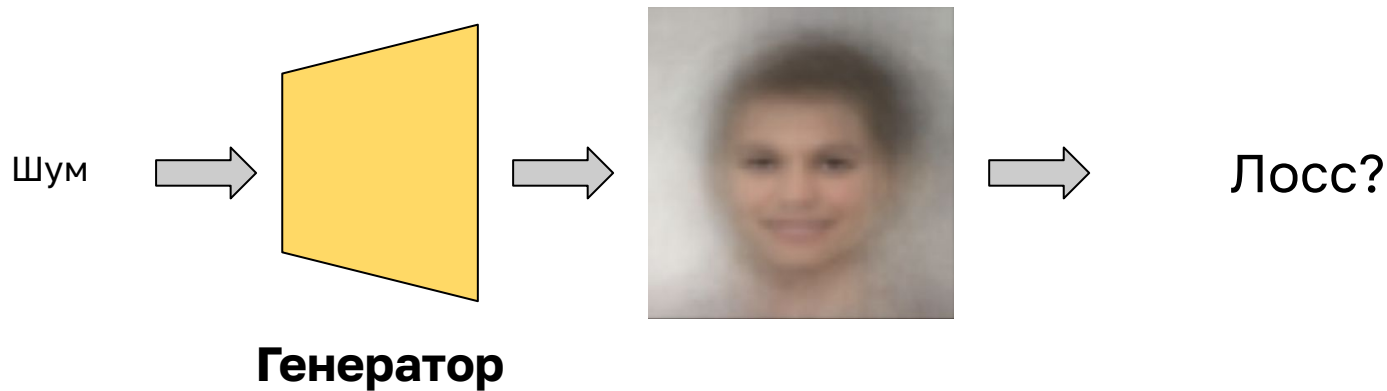
BCELoss,  
label = 0

**Дискриминатор**



# Обучение генератора

ІТМО



# Обучение генератора



Формула бинарной кросс-энтропии:



$$L(y_{true}, y_{pred}) = -\mathbb{E}[y_{true} \log(y_{pred}) + (1 - y_{true}) \log(1 - y_{pred})]$$

Первый шаг обучения дискриминатора:

$$L(y_{true} = 1, y_{pred} = D(x_{real})) = -\mathbb{E}[\log(D(x_{real}))]; \Leftrightarrow L_1 = -\mathbb{E}_{x \sim p(x)}[\log(D(x))],$$

Второй шаг обучения дискриминатора:

$$L(y_{true} = 0, y_{pred} = D(x_{gen})) = -\mathbb{E}[\log(1 - D(x_{gen}))];$$

$$L_2 = -\mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))],$$

# Adversarial Loss. Minimax-game

Запишем функцию V:



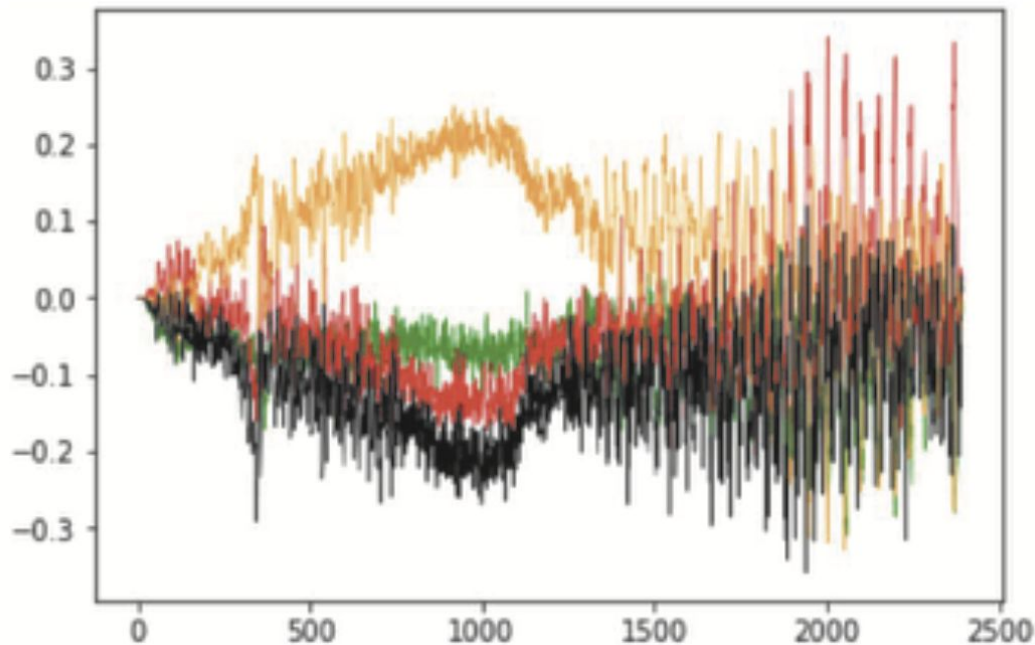
$$V_{D,G} = -L_1 - L_2 = \mathbb{E}_{x \sim p(x)}[\log(D(x))] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]$$

Сыграем в минимаксную игру:

$$\min_G \max_D \{V_{D,G}\} = \min_G \max_D \left\{ \mathbb{E}_{x \sim p(x)}[\log(D(x))] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))] \right\}$$

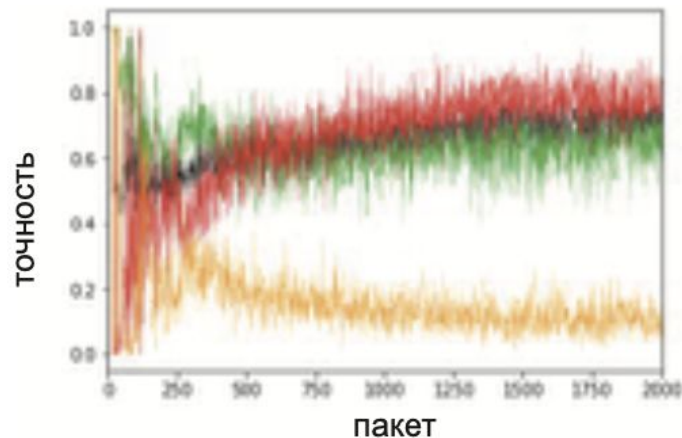
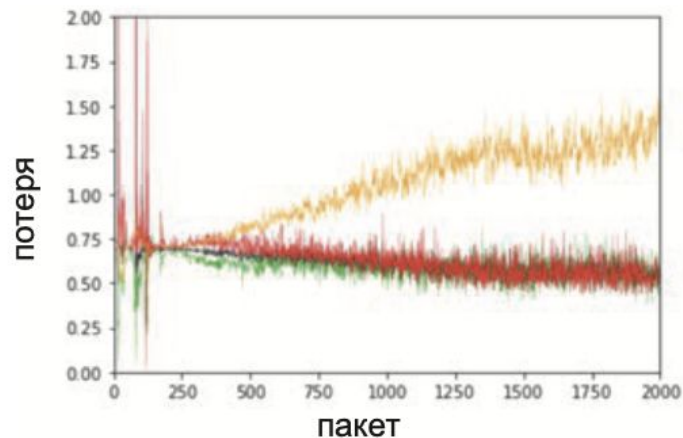
# Проблемы GAN. Колебания лосса

ІІТМО



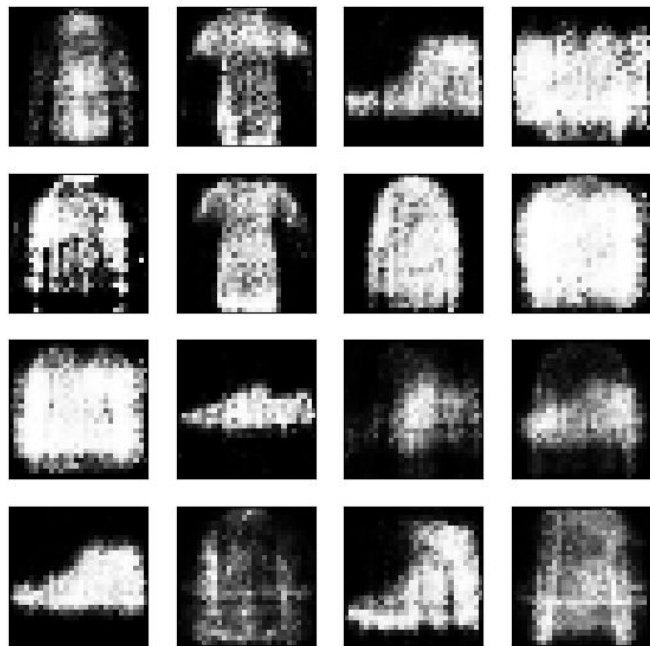
# Проблемы GAN.

## Неинформативность лосса

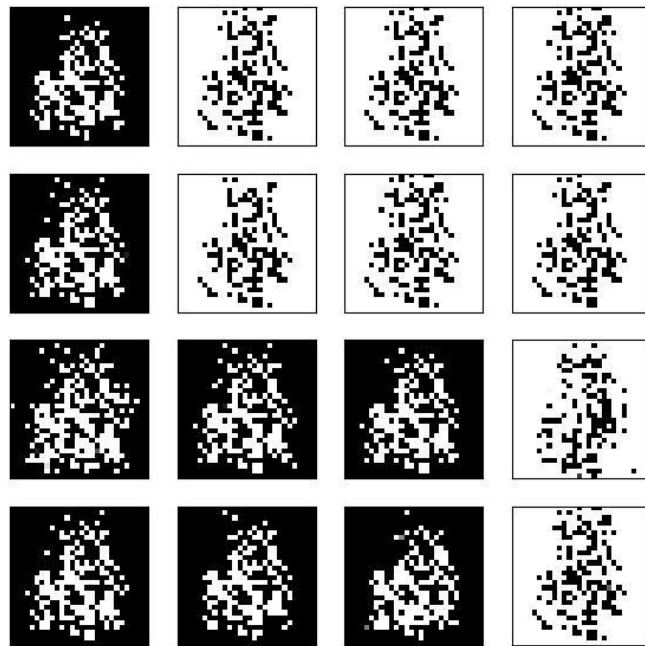


- дискриминатор (истинное)
- дискриминатор (поддельное)
- дискриминатор (среднее по истинным и поддельным)
- генератор

# Проблемы GAN. Коллапс модели



хочу



могу





# Функция потерь Вассерштайна

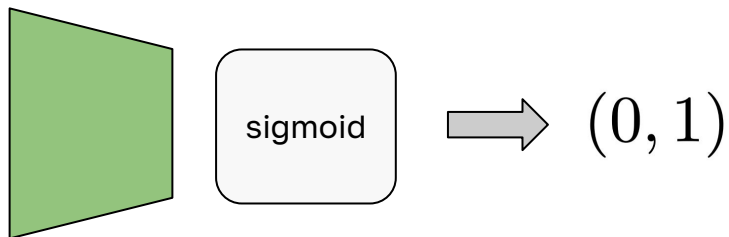
Функция потерь:

$$L_W = -\frac{1}{n} \sum_{i=1}^n (y_i p_i).$$

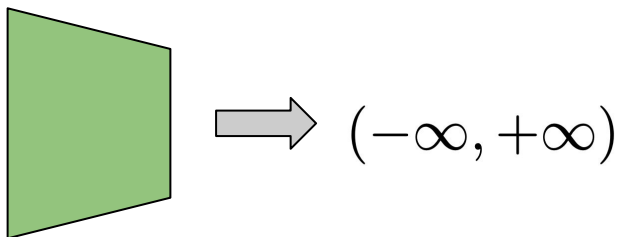
label\_false = -1

label\_true = 1

Дискриминатор:



Критик:



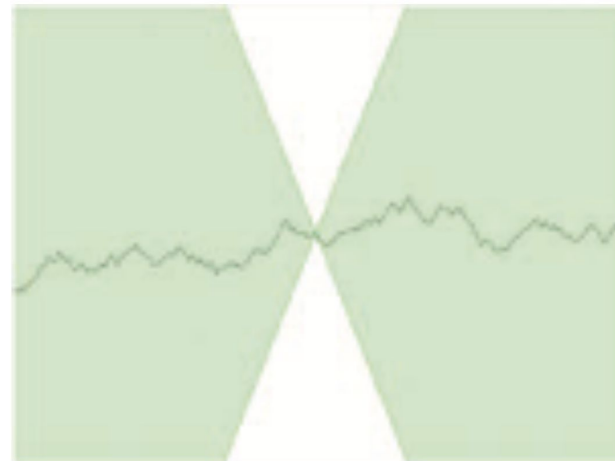
# Простое решение проблемы неограниченности выхода



Критик должен быть одномерной Липшицевой функцией:

$$\frac{|D(x_1) - D(x_2)|}{\sum_i |x_{1_i} - x_{2_i}|} \leq 1,$$

Поэтому урежем веса критика до  $[-0.01, 0.01]$



# Хорошее решение проблемы неограниченности выхода



Большие значения выхода приводят к большим значениям градиента.

Тогда для уменьшения выхода добавим в функцию потерь штраф за градиент:

$$L = \underbrace{\mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g} [D(\tilde{\mathbf{x}})] - \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r} [D(\mathbf{x})]}_{\text{Original critic loss}} + \lambda \underbrace{\mathbb{E}_{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}} [(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2]}_{\text{Our gradient penalty}} .$$

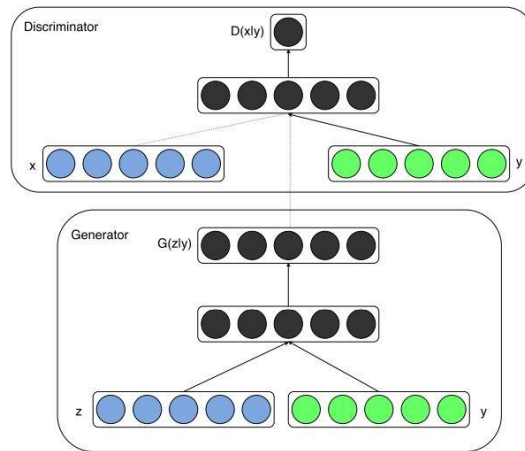
# Conditional GAN

Добавим контроль в генерацию:



$$\min_G \max_D \left\{ V_{G,D} \right\} = \min_G \max_D \left\{ \mathbb{E}_{x \sim p(x)} [\log D(x|y)] + \mathbb{E}_{z \sim p(z)} [1 - D(G(z|y))] \right\}.$$

Генератор аналогичен  
Conditional AE/VAE.  
Дискриминатор  
дополнительно получает  
информацию о контроле.





Задачи генеративного моделирования:

- Генерация изображений (Image generation)
- Перенос домена (Image2Image translation, Domain Transfer)
- Увеличение разрешения (Super Resolution)
- Генерация текста (Text Generation)
- Описание изображения (Image Captioning)
- Генерация изображения по тексту (Text2Image)

# Progressive GAN

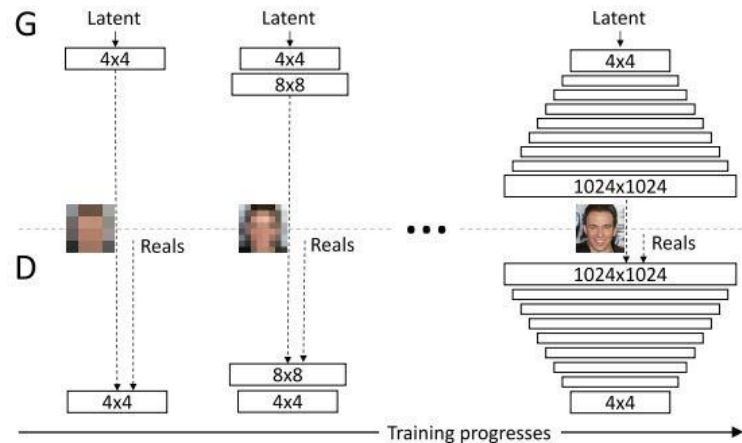


Figure 1: We start with a low resolution of  $4 \times 4$  pixels. As the training advances, we incrementally introduce new layers to increase the resolution.

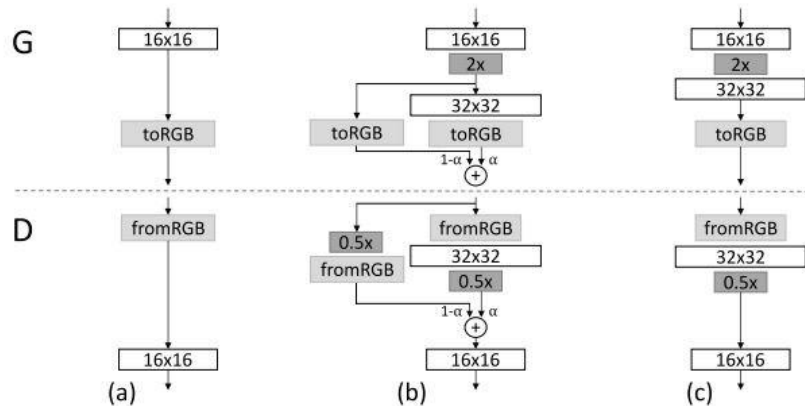


Figure 2: When doubling the resolution, we faded in the new layers smoothly by treating them like a residual block, whose weight increases linearly from 0 to 1.

# Progressive GAN

VITMO







Labels to Street Scene

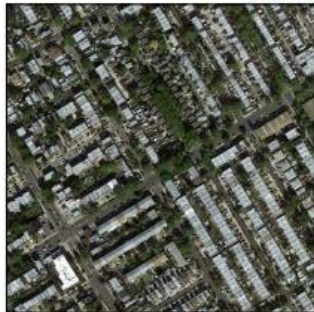


input



output

Aerial to Map

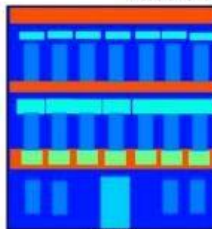


input



output

Labels to Facade



input



output

Day to Night



input



output

BW to Color



input



output

Edges to Photo



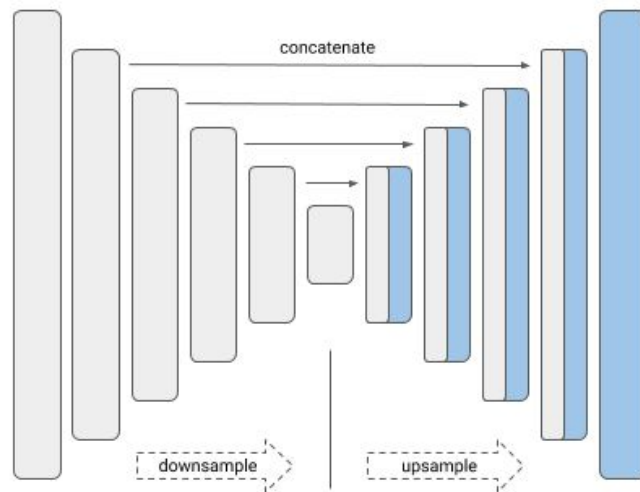
input



output



# Pix2pix. Генератор



# Pix2pix. Сравнение лоссов

ІІТМО



# Pix2pix. Дискриминатор

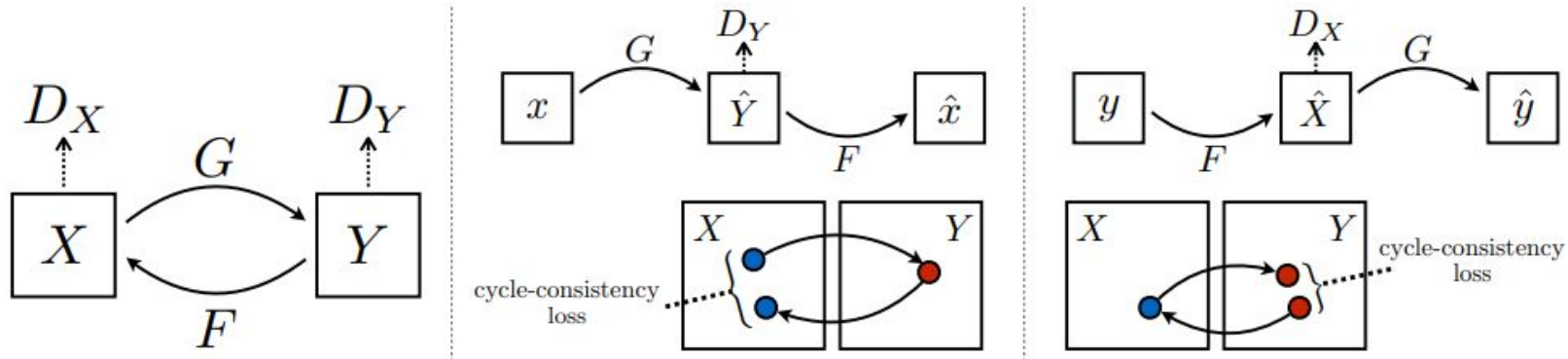


Дискриминатор улучшает моделирование высокочастотных признаков, но ухудшает моделирование низкочастотных.

Решение: ограничим дискриминатор “окном”, чтобы он не мог влиять на низкочастотные признаки



# CycleGAN. Архитектура



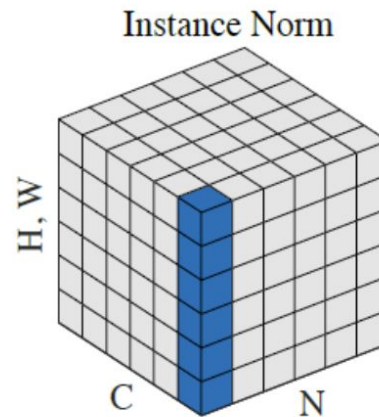
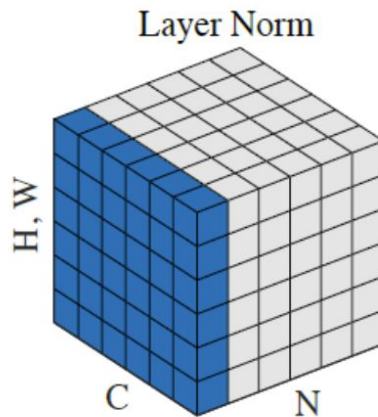
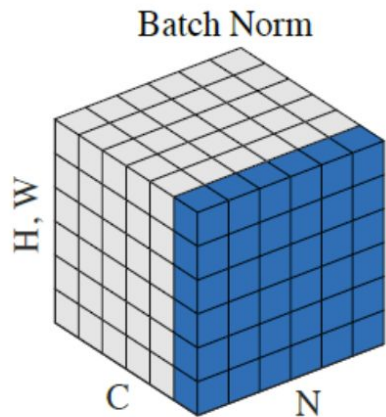
$$G : X \rightarrow Y$$

$$F : Y \rightarrow X$$

# CycleGAN. InstanceNorm

Формула общая с пакетной нормализацией

$$y = \frac{x - E[x]}{\sqrt{\text{Var}[x] + \epsilon}} * \gamma + \beta$$





Трансфер в одну сторону:

$$L_{GAN,G,D_Y}(x,y) = \mathbb{E}_{y \sim p(y)} [\log D_Y(y)] + \mathbb{E}_{x \sim p(x)} [\log(1 - D_Y(G(x)))]$$

Трансфер в другую сторону:

$$L_{GAN,F,D_X}(y,x)$$

Мы хотим сохранить возможность циклично преобразовывать данные:

$$x \rightarrow G(x) \rightarrow F(G(x)) \approx x,$$

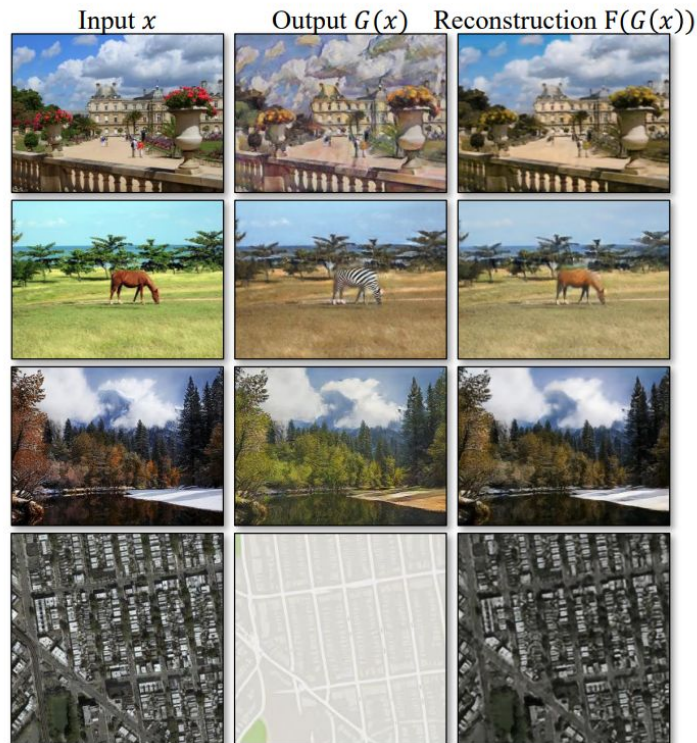
$$y \rightarrow F(y) \rightarrow G(F(y)) \approx y$$

Добавим компоненту лосса - cycle consistency loss:

$$L_{cyc,G,F}(x, y) = \mathbb{E}_{x \sim p(x)} \left[ \|F(G(x)) - x\|_1 \right] + \mathbb{E}_{y \sim p(y)} \left[ \|G(F(y)) - y\|_1 \right]$$



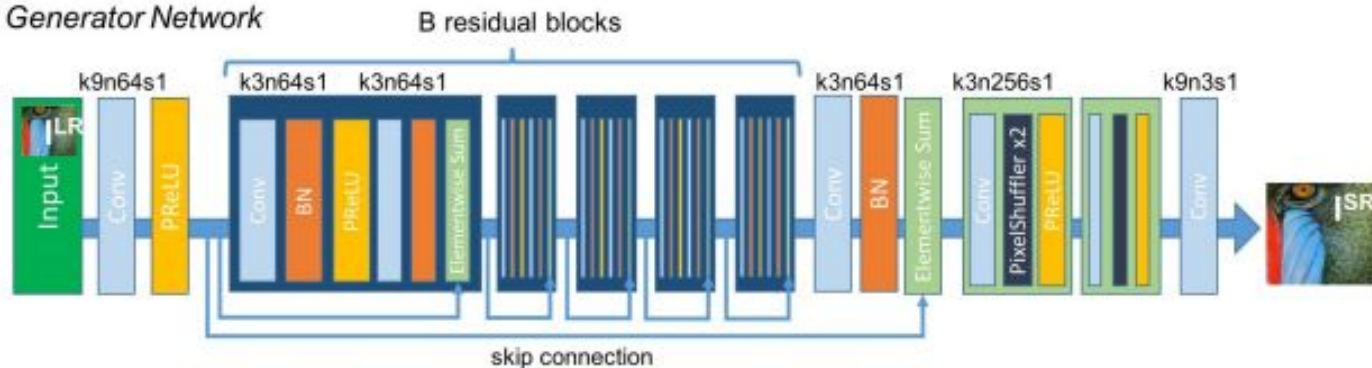
# CycleGAN. Результаты генерации



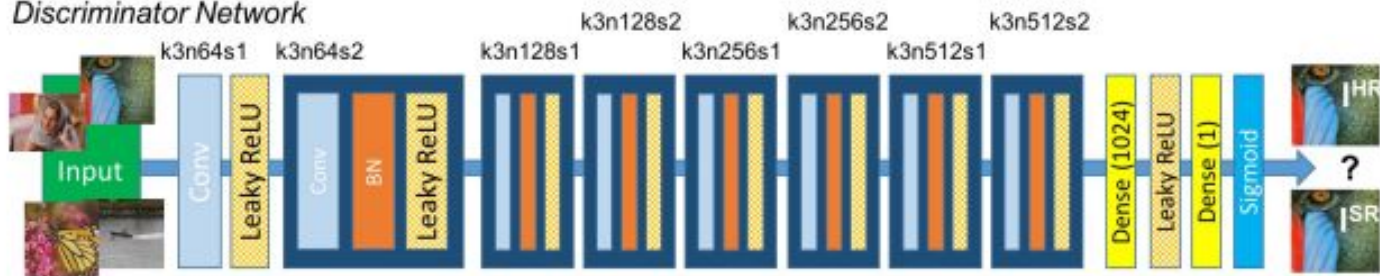


# Super Resolution GAN. Архитектура ИИТМО

Generator Network



Discriminator Network



# Super Resolution GAN. Архитектура



Функция активации - PReLU:

$$PReLU(x) = \begin{cases} x, & \text{если } x \geq 0, \\ ax, & \text{иначе,} \end{cases}$$

Pixel Shuffle - увеличение размерности за счет количества каналов:

$$(4 \cdot C) \times H \times W \rightarrow C \times (2 \cdot H) \times (2 \cdot W)$$

# SRGAN. Perceptual loss

Функция потерь:

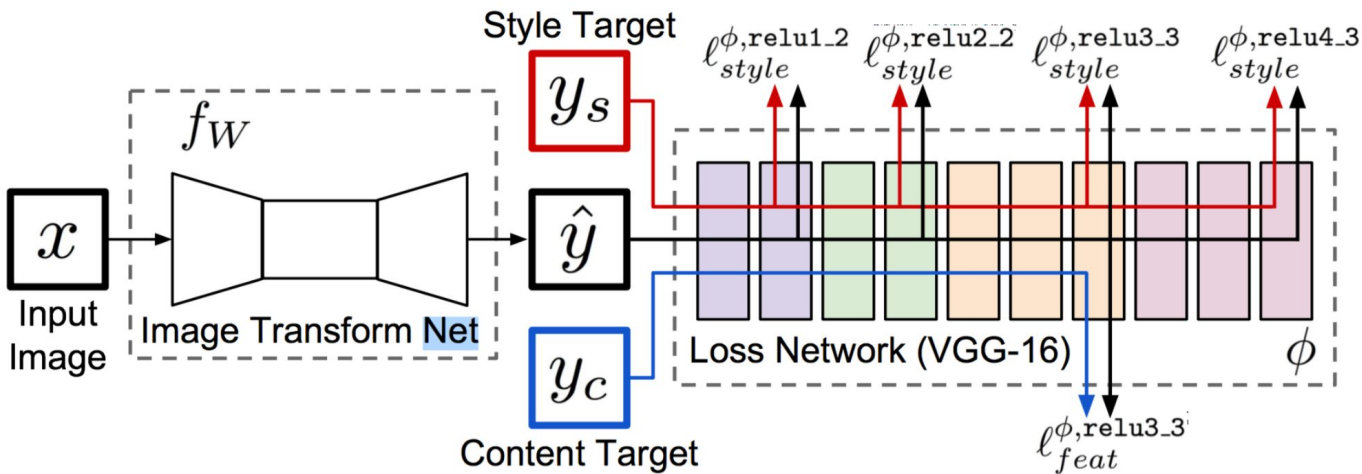
$$L_{SR} = L_{VGG/i,j} + 10^{-3} L_{gen}.$$

где:

$$L_{VGG/i,j} = \|VGG_{i,j}(x_{SR}) - VGG_{i,j}(G(x_{LR}))\|_2$$

$$L_{gen} = - \sum_{n=1}^N \log D(G(x_{LR}))$$

# Perceptual loss для Style Transfer



# SRGAN. Сравнение результатов

