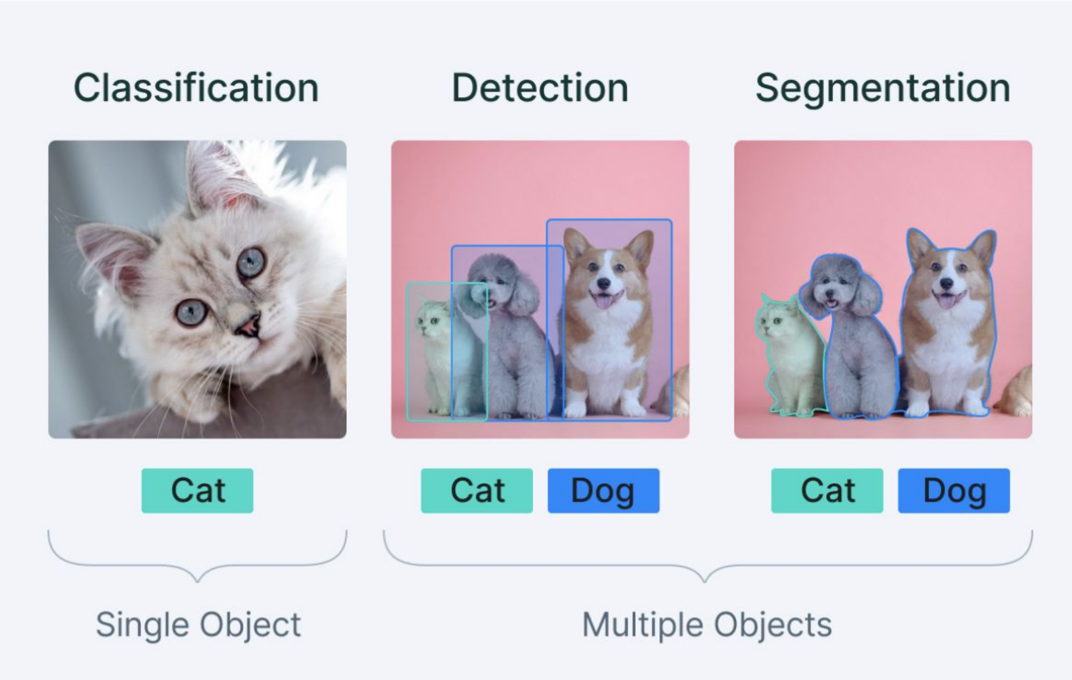
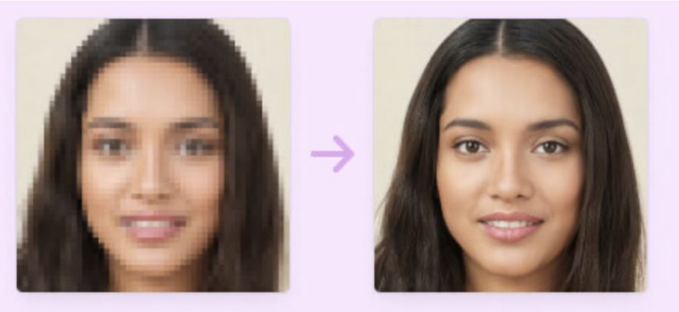
# cv and ml

## generative models

Владимир Глазачев
cv в rosebud.ai

# Умеем решать supervised задачи



- Традиционные задачи
- немного img2img если есть парный датасет
- умеем разумно сравнивать изображения (perceptual loss)

# Discriminative vs Generative Models

**Discriminative Model:**
Learn a probability distribution p(y|x)

**Generative Model**:
Learn a probability distribution p(x)

**Conditional Generative Model:** Learn p(x|y)

**Data**: x

**Label**: y

Cat

# Discriminative vs Generative Models

**Discriminative Model:**
Learn a probability distribution p(y|x)

**Generative Model**:
Learn a probability distribution p(x)

**Conditional Generative Model:** Learn p(x|y)

**Data**: x

**Label**: y
Cat

**Probability Recap:**

**Density Function**
p(x) assigns a positive number to each possible x; higher numbers mean x is more likely

Density functions are **normalized**:

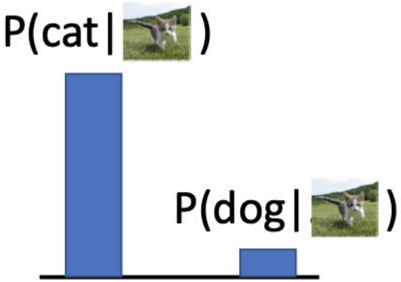
$$\int_X p(x)dx = 1$$

Different values of x **compete** for density

# Discriminative vs Generative Models

**Discriminative Model:** Learn a probability distribution p(y|x)

**Data**: x




**Generative Model**: Learn a probability distribution p(x)

**Density Function**
p(x) assigns a positive number to each possible x; higher numbers mean x is more likely

Density functions are **normalized**:

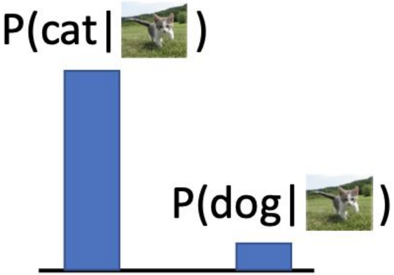
$$\int_X p(x)dx = 1$$

Different values of x **compete** for density

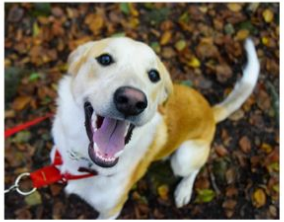
**Conditional Generative Model:** Learn p(x|y)

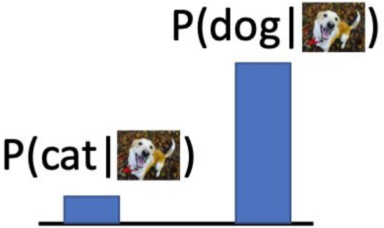
# Discriminative vs Generative Models

**Discriminative Model:**
Learn a probability distribution $p(y|x)$




**Generative Model**:
Learn a probability distribution $p(x)$



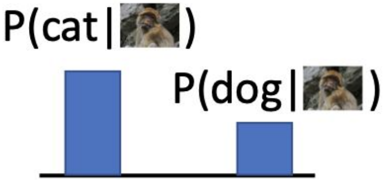



**Conditional Generative Model:** Learn $p(x|y)$

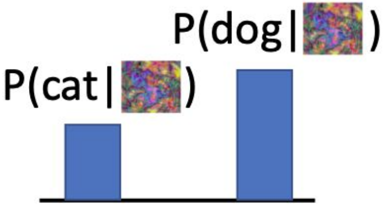
Discriminative model: the possible labels for each input "compete" for probability mass. But no competition between **images**

# Discriminative vs Generative Models

**Discriminative Model:** Learn a probability distribution p(y|x)




**Generative Model**: Learn a probability distribution p(x)




**Conditional Generative Model:** Learn p(x|y)

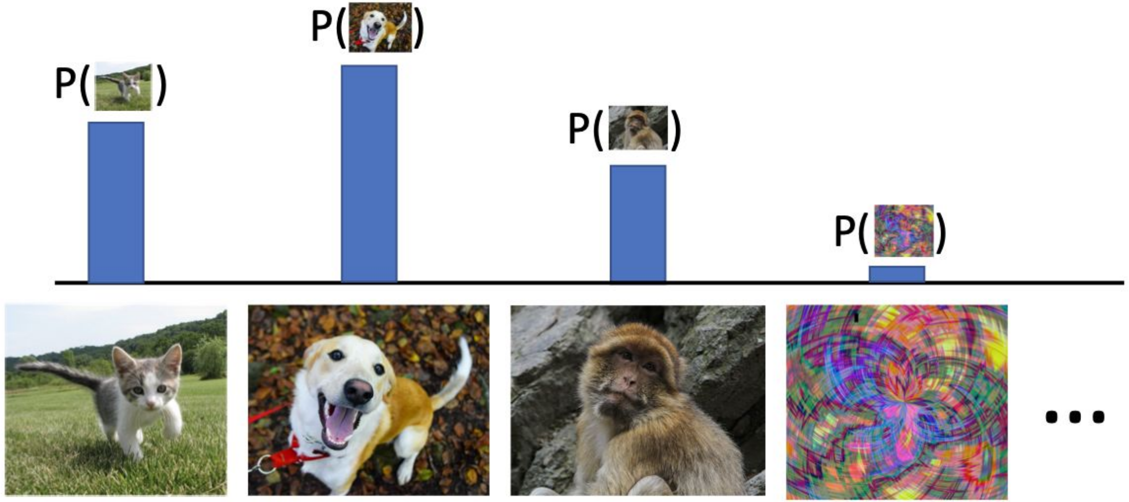
Discriminative model: No way for the model to handle unreasonable inputs; it must give label distributions for all images

# Discriminative vs Generative Models

**Discriminative Model:** Learn a probability distribution p(y|x)

**Generative Model**: Learn a probability distribution p(x)

**Conditional Generative Model:** Learn p(x|y)



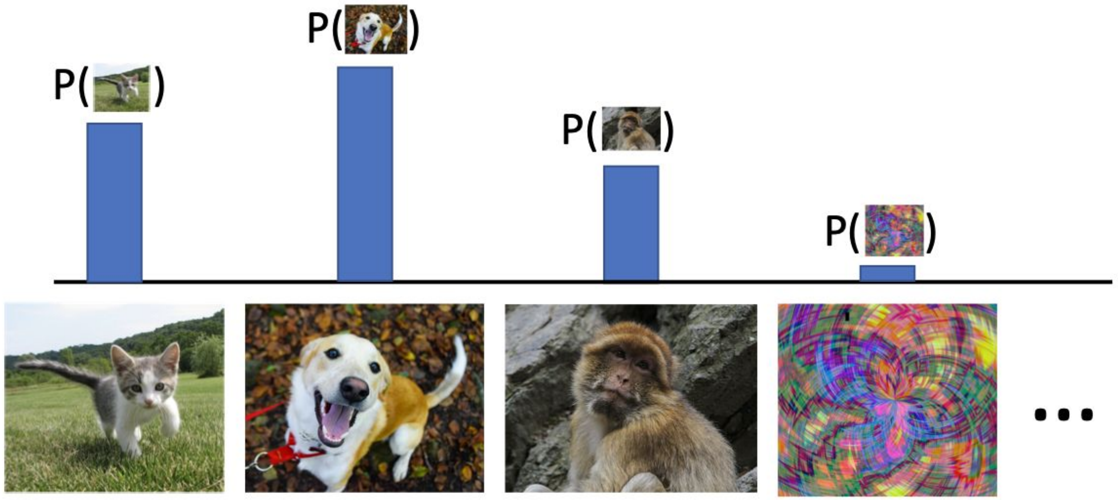

Generative model: All possible images compete with each other for probability mass

# Discriminative vs Generative Models

**Discriminative Model:**
Learn a probability distribution p(y|x)

**Generative Model:**
Learn a probability distribution p(x)

**Conditional Generative Model:** Learn p(x|y)




Generative model: All possible images compete with each other for probability mass

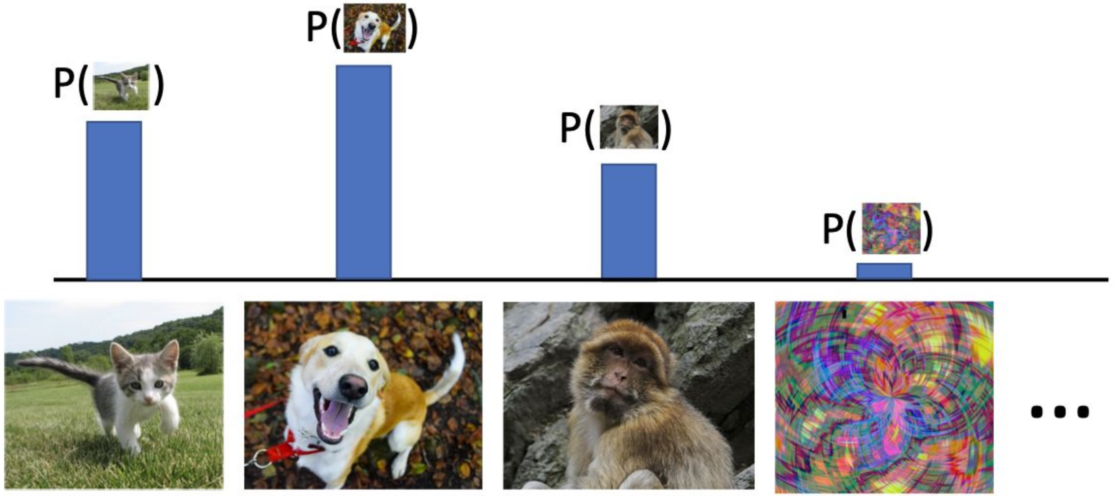
Requires deep image understanding! Is a dog more likely to sit or stand? How about 3-legged dog vs 3-armed monkey?

# Discriminative vs Generative Models

**Discriminative Model:** Learn a probability distribution p(y|x)

**Generative Model**: Learn a probability distribution p(x)

**Conditional Generative Model:** Learn p(x|y)



Generative model: All possible images compete with each other for probability mass

Model can "reject" unreasonable inputs by assigning them small values

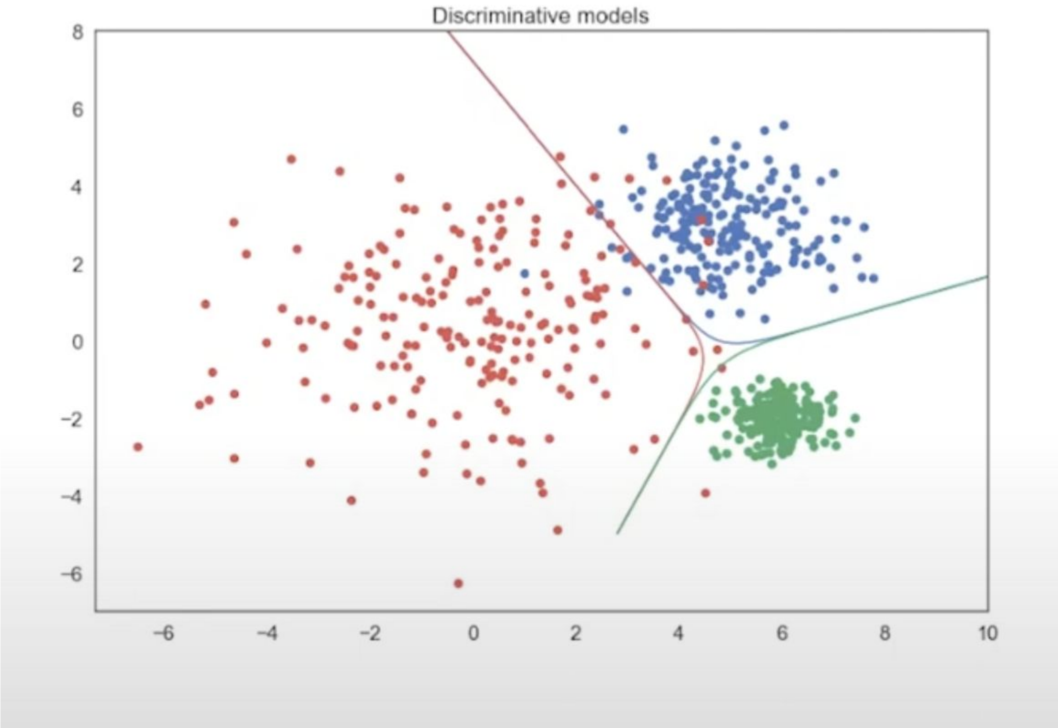
# What can we do with a generative model?

**Discriminative Model:**
Learn a probability distribution $p(y|x)$ → Assign labels to data
Feature learning (with labels)

**Generative Model**:
Learn a probability distribution $p(x)$ → Detect outliers
Feature learning (without labels)
**Sample** to generate new data

**Conditional Generative Model:** Learn $p(x|y)$

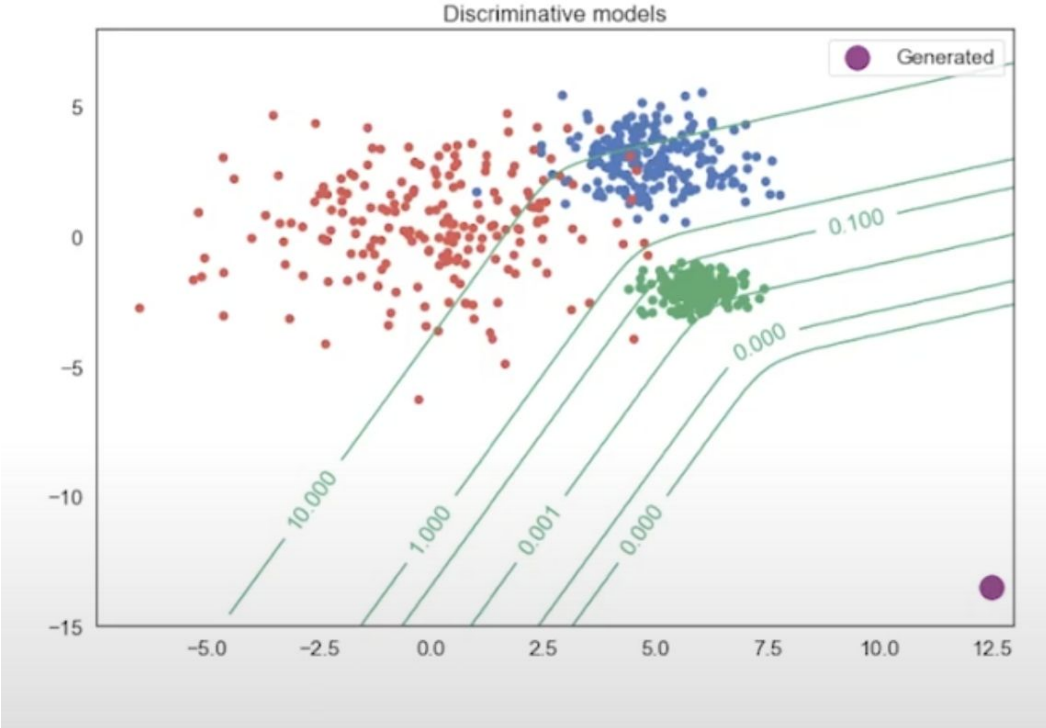
# У нас есть дискрииминативная модель



# У нас есть дискриминативная модель

Классификация на 3 класса

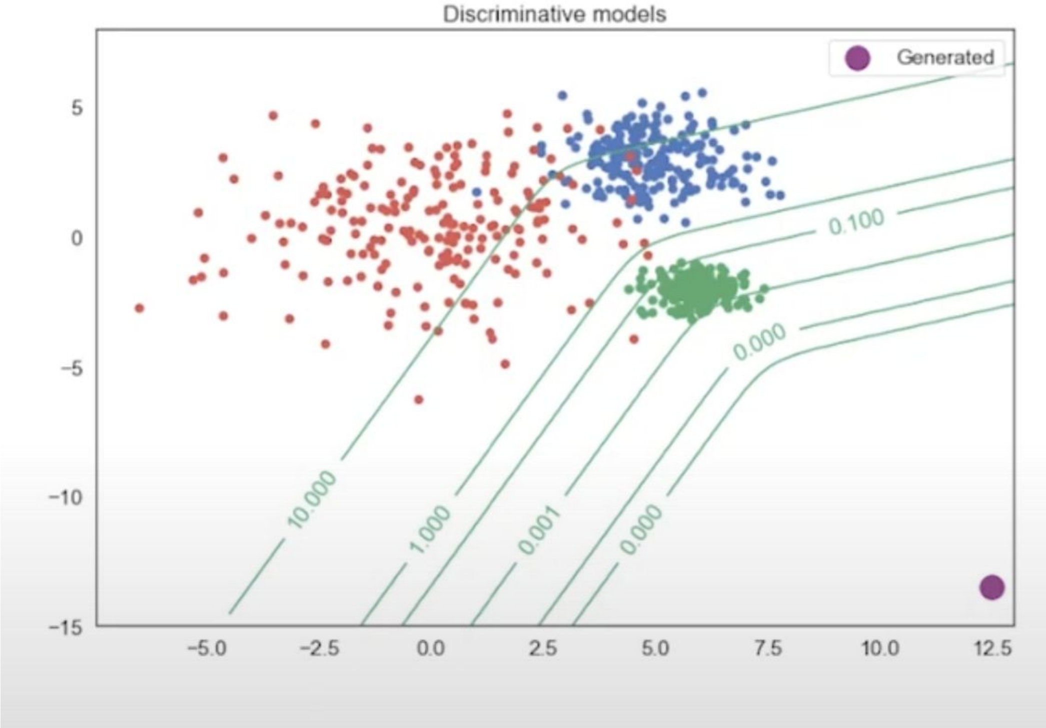
- Хотим сгенерировать точку зеленого класса
- Инициализируем рандомную точку (x, y)
- Считаем предсказание, считаем -log(p(y | x)) и делаем градиентый шаг по нему - обновляем (x, y)



# У нас есть дискрииминативная модель

Классификация на 3 класса
- Хотим сгенерировать точку зеленого класса
- Инициализируем рандомную точку (x, y)
- Считаем предсказание, считаем -log(p(y | x)) и делаем градиентый шаг по нему - обновляем (x, y)

- максимизируем принадлежность к классу



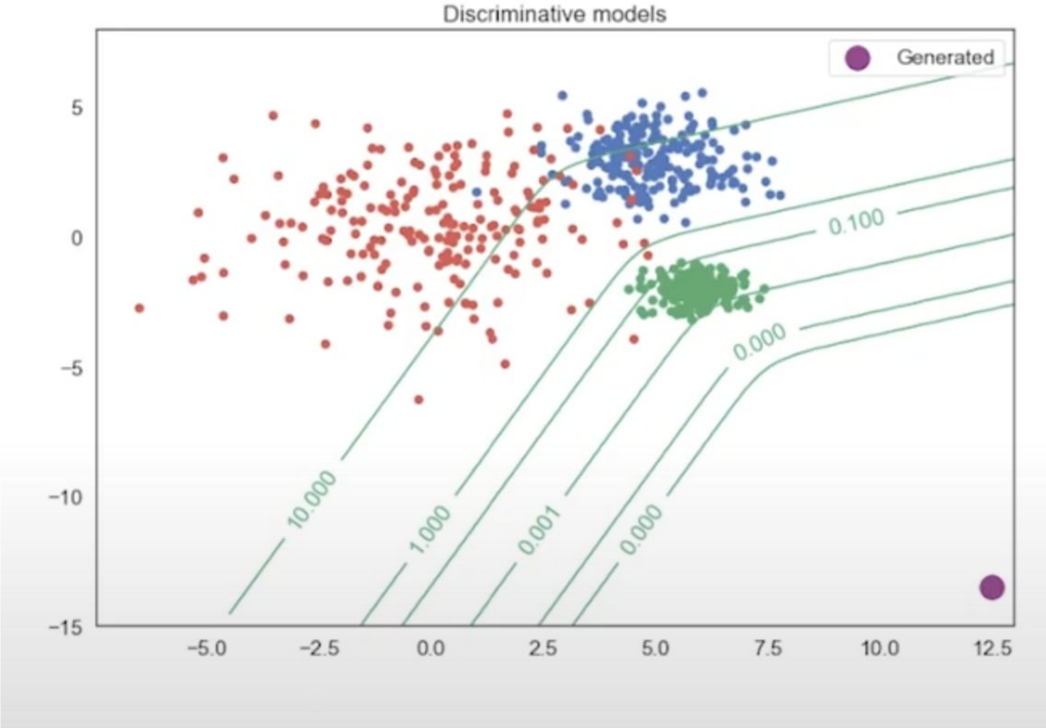
# У нас есть дискрииминативная модель

Классификация на 3 класса

- Хотим сгенерировать точку зеленого класса
- Инициализируем рандомную точку (x, y)
- Считаем предсказание, считаем -log(p(y | x)) и делаем градиентый шаг по нему - обновляем (x, y)

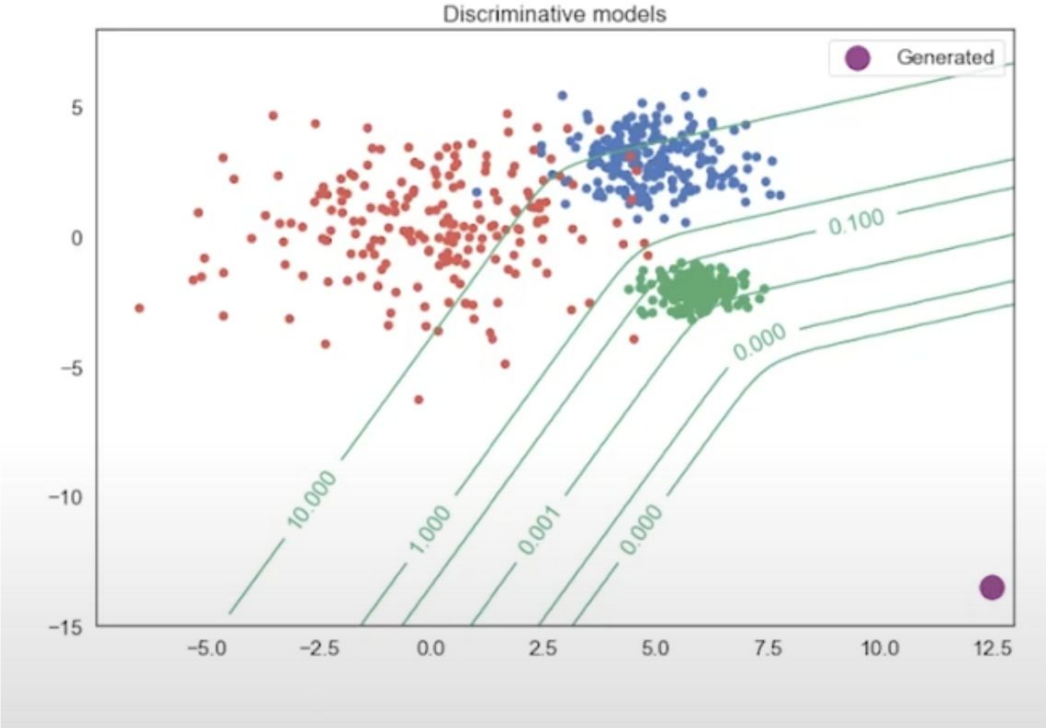
- максимизируем принадлежность к классу

- точка уехала куда то далеко, хоть и правильно разделена :)



# У нас есть дискрииминативная модель

Для картинок можно попробовать сделать тоже самое, но картинки не получится - будет прост какой то шум который модель распознает как нужный класс

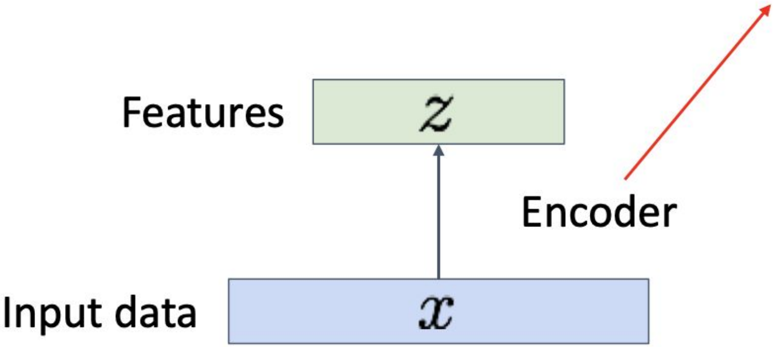


# Autoencoder

Unsupervised method for learning feature vectors from raw data x, without any labels

Features should extract useful information (maybe object identities, properties, scene type, etc) that we can use for downstream tasks

**Originally**: Linear + nonlinearity (sigmoid)
**Later**: Deep, fully-connected
**Later**: ReLU CNN





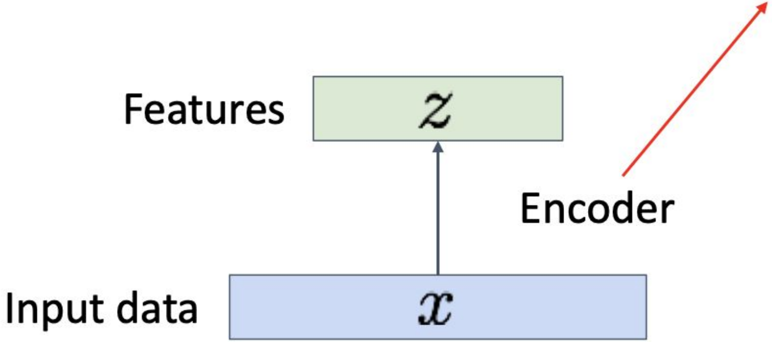
Input Data

# Autoencoder

**Problem**: How can we learn this feature transform from raw data?

Features should extract useful information (maybe object identities, properties, scene type, etc) that we can use for downstream tasks
But we can't observe features!

**Originally**: Linear + nonlinearity (sigmoid)
**Later**: Deep, fully-connected
**Later**: ReLU CNN



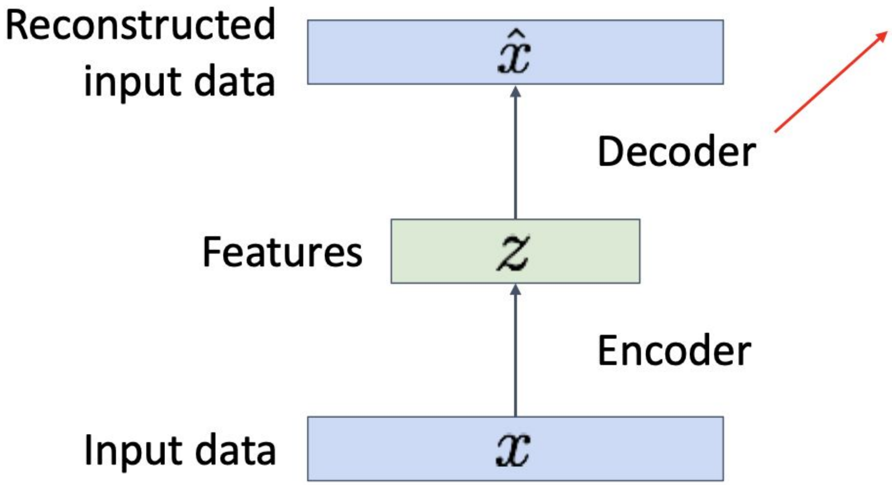

Input Data

# Autoencoder

**Problem**: How can we learn this feature transform from raw data?

**Idea**: Use the features to reconstruct the input data with a **decoder**

“Autoencoding” = encoding itself




**Originally**: Linear + nonlinearity (sigmoid)

**Later**: Deep, fully-connected

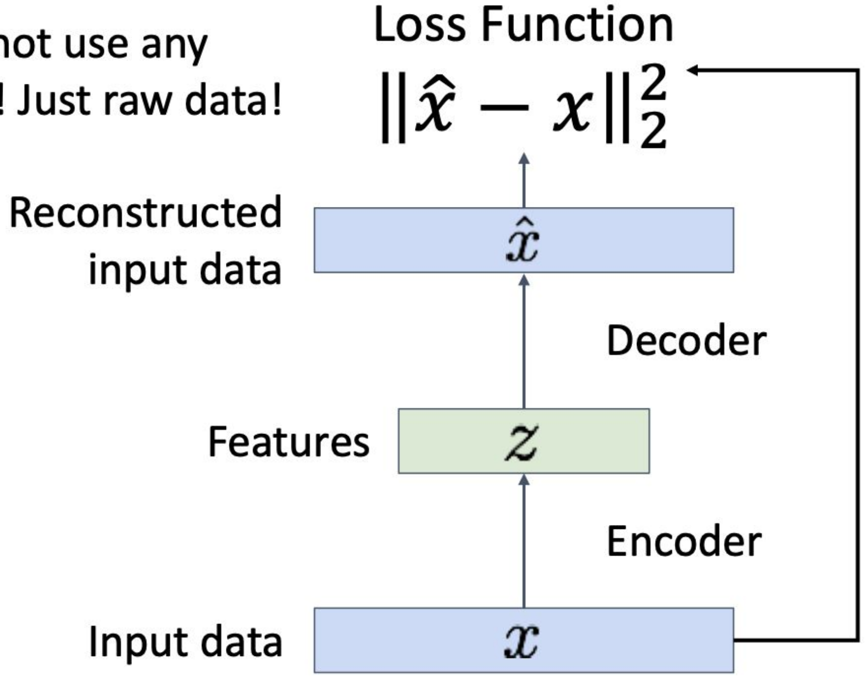
**Later**: ReLU CNN (upconv)

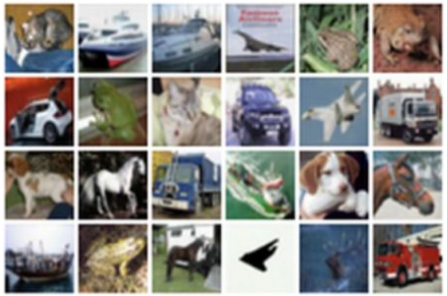

Input Data

# Autoencoder

**Loss**: L2 distance between input and reconstructed data.

Does not use any labels! Just raw data!



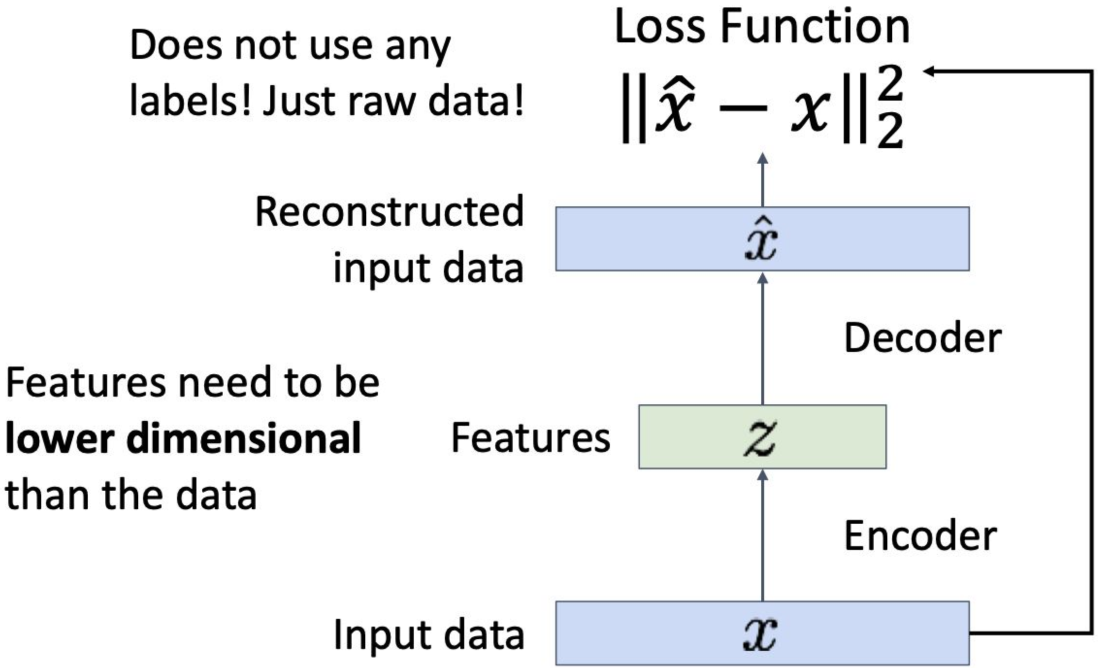



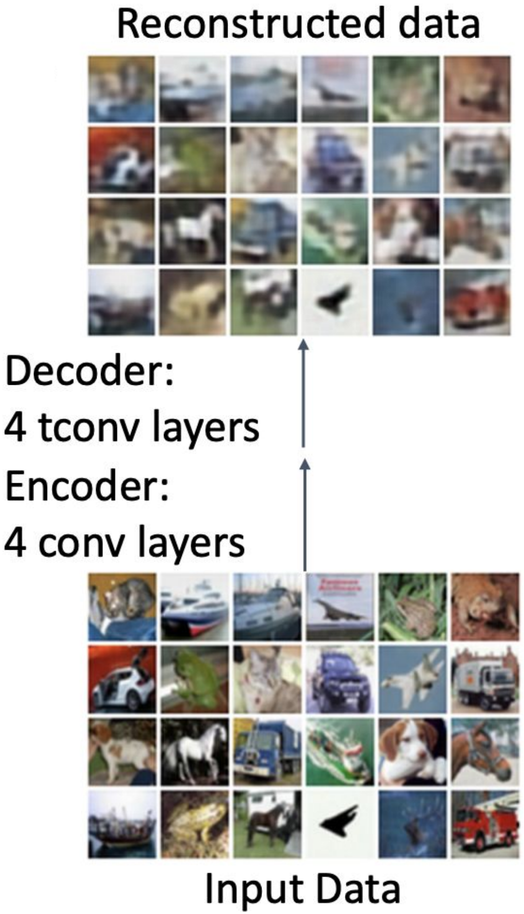
Input Data

# Autoencoder

Можно добавить сюда еще наш perceptual loss и станет получше

**Loss**: L2 distance between input and reconstructed data.



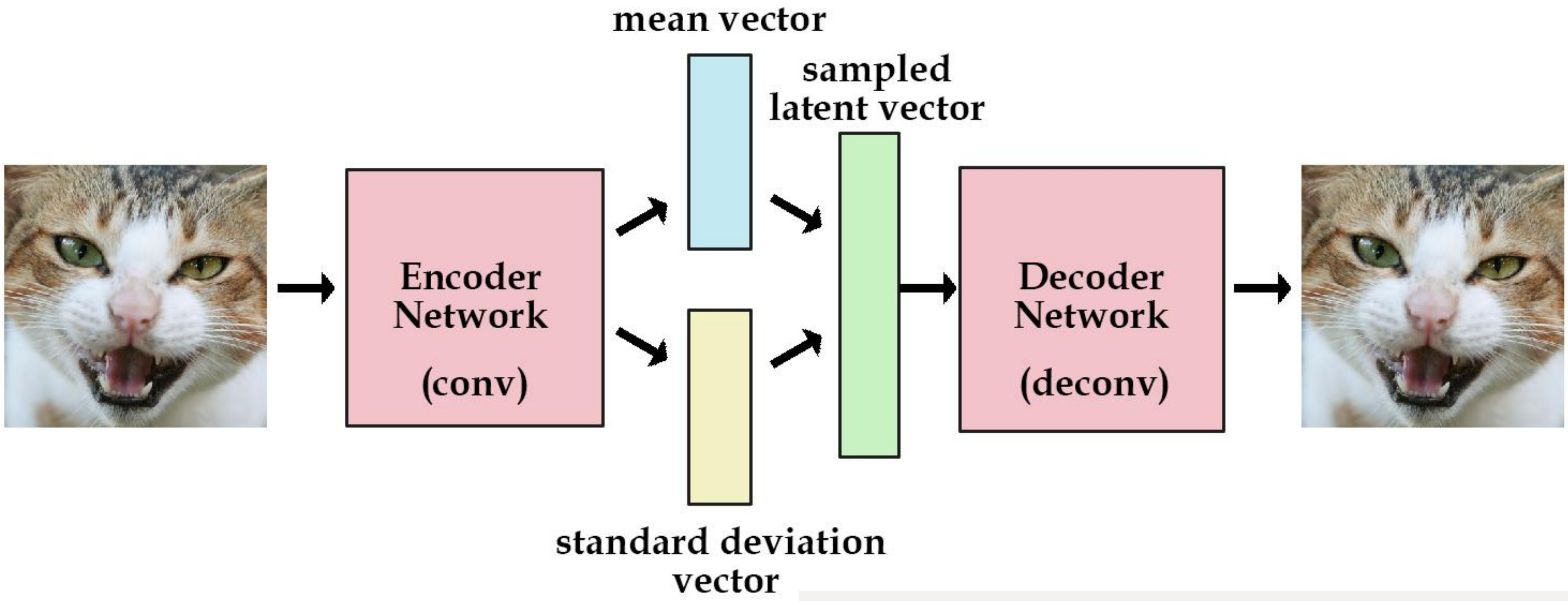
# Autoencoder

- Что то напоминает - UNET это тоже автоенкодер но мы к нему пришли через другую задачу
- Через автоенкодер можно решать задачи поиска аномалий:
  - обучаем на данных без аномалий, они должны хорошо реконструироваться
  - если пришла аномалия на инференсе - у нее будет большая ошибка реконструкции

- Задачу генерации это нам пока решать не помогает - у автоенкодера нет способа семплировать новые данные

- Идеи?
  - Мы упростили задачу, нам надо семплировать не всю картинку (входные данные) а только компактное пространство фич
  - Можно как нибудь обусловить его, чтобы у фич были вероятностные свойства

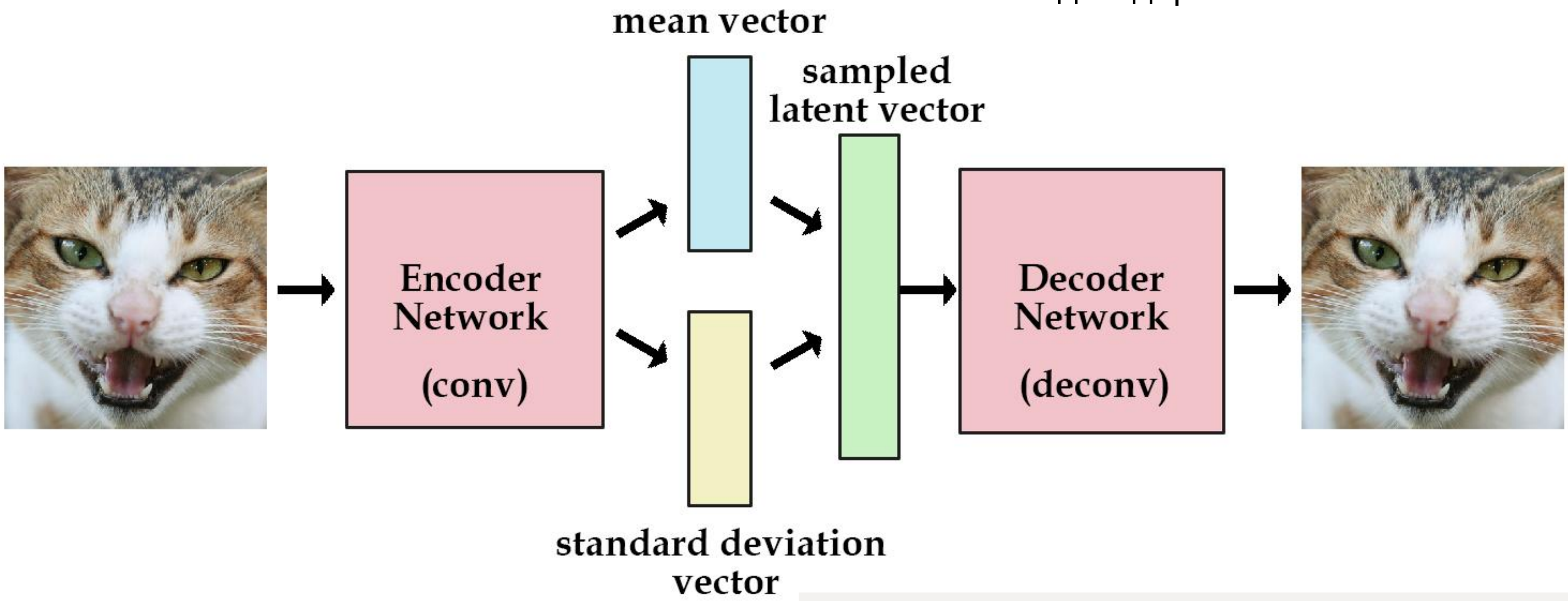
# Variational Autoencoder




```
generation_loss = mean(square(generated_image - real_image))
latent_loss = KL-Divergence(latent_variable, unit_gaussian)
loss = generation_loss + latent_loss
```

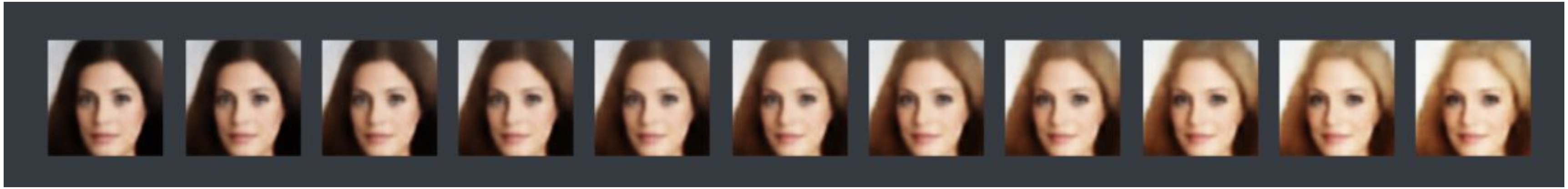
# Variational Autoencoder

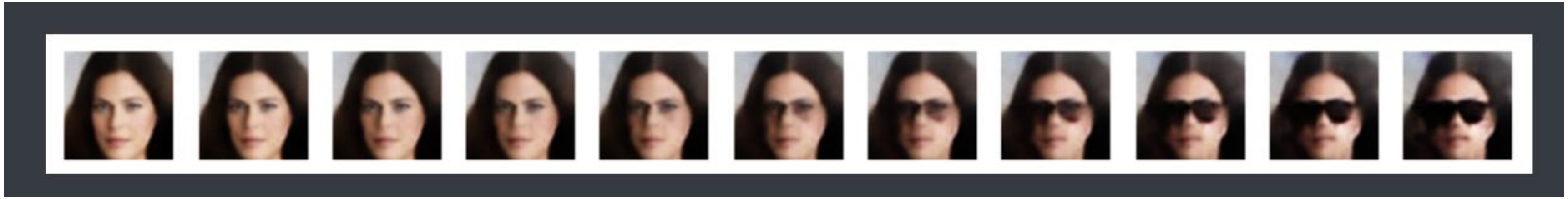
- Форсим фичи жить в нормальном распределении
- В момент генерации - семплируем из нормального распределения и отдаем в декодер




```
generation_loss = mean(square(generated_image - real_image))
latent_loss = KL-Divergence(latent_variable, unit_gaussian)
loss = generation_loss + latent_loss
```

# Variational Autoencoder



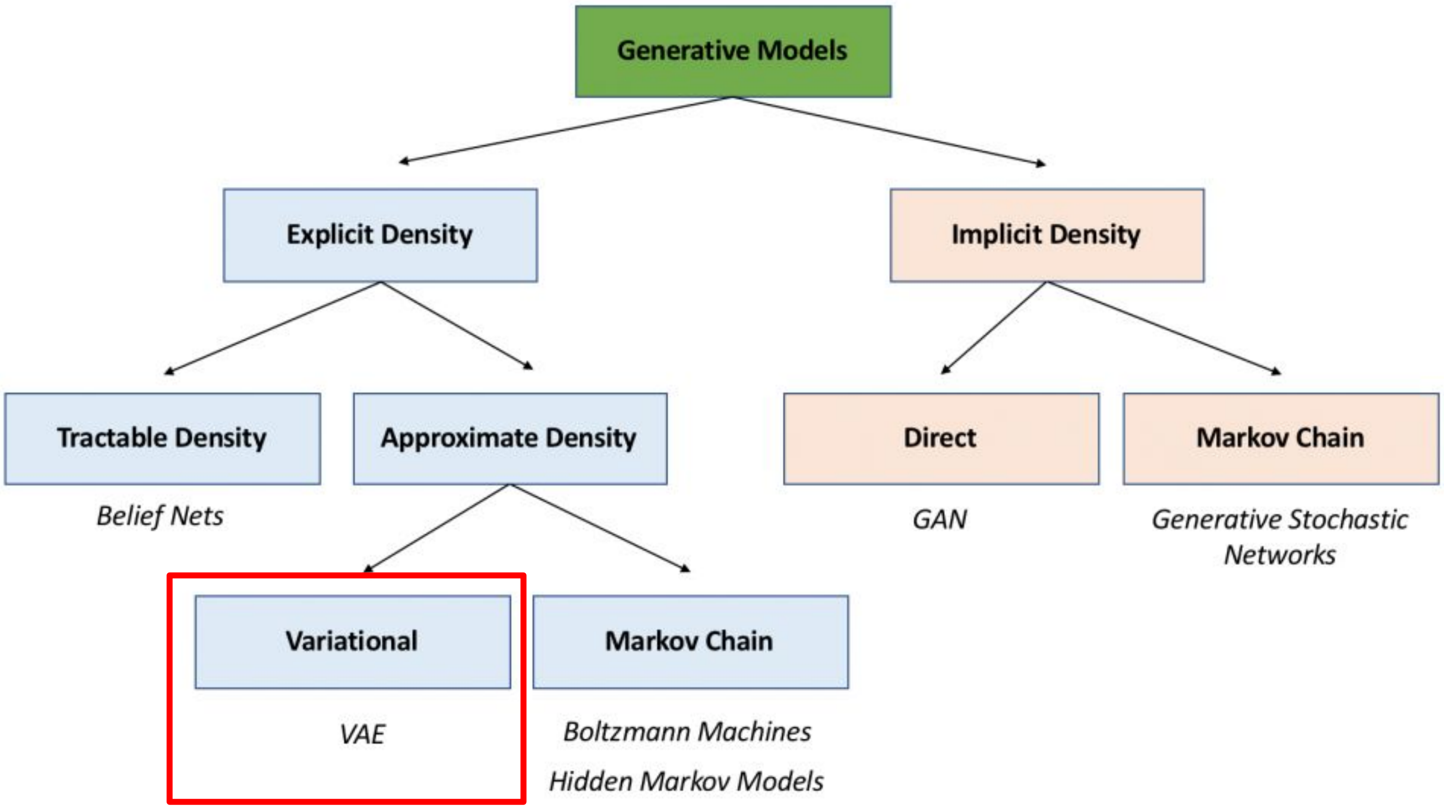


- Можно делать арифметику в пространстве фич!
- Можно искать направления фич (где находятся люди в очках или блондины)
- Все что работает для норм распределения - работает и тут (сумма - нормальное)

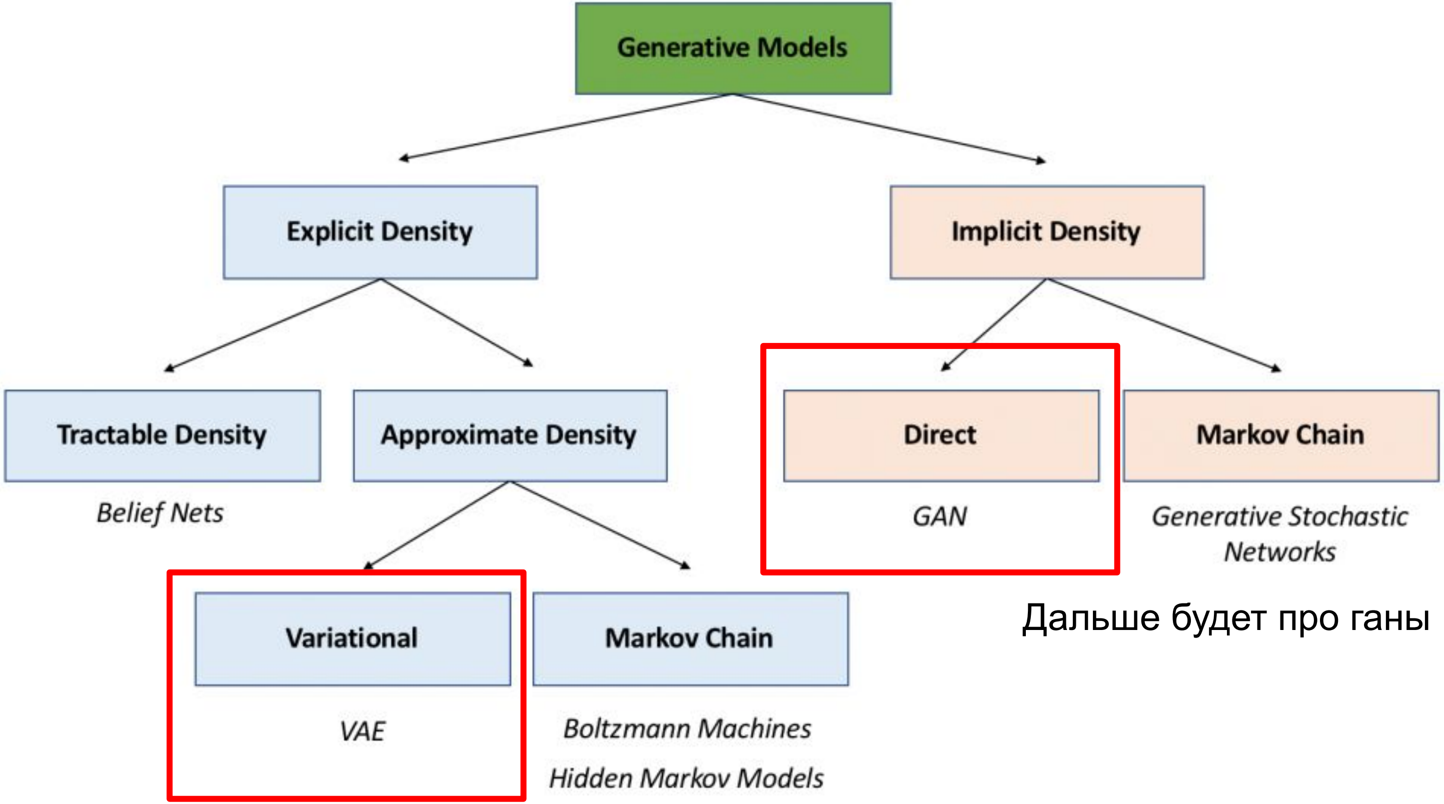
# Variational Autoencoder

## Combining VAE + Autoregressive: VQ-VAE2

1024 x 1024 generated faces, trained on FFHQ



Generative Models
Explicit Density
Implicit Density
Tractable Density
Approximate Density
Direct
Markov Chain
Belief Nets
GAN
Generative Stochastic Networks
Variational
Markov Chain
VAE
Boltzmann Machines
Hidden Markov Models



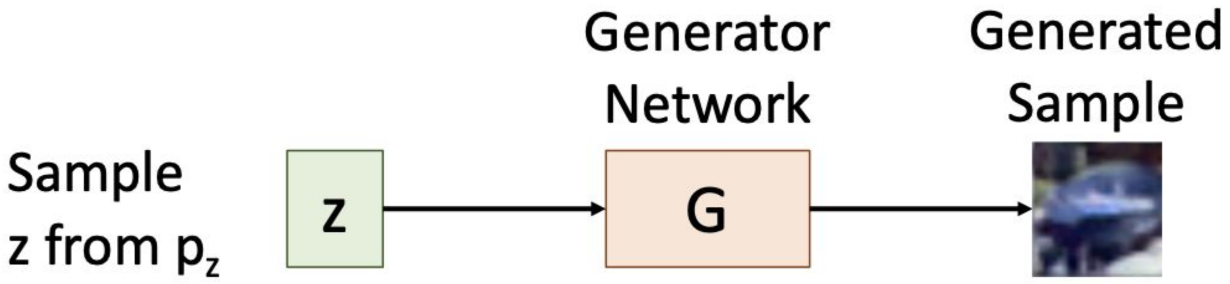

Дальше будет про ганы

# GAN

- Generative Adversarial Network
- У нас есть X данные (изображения) из распределения p
- Мы хотим научиться семплировать из p новые данные
- Мы не хотим знать/учить p, мы просто хотим новые данные

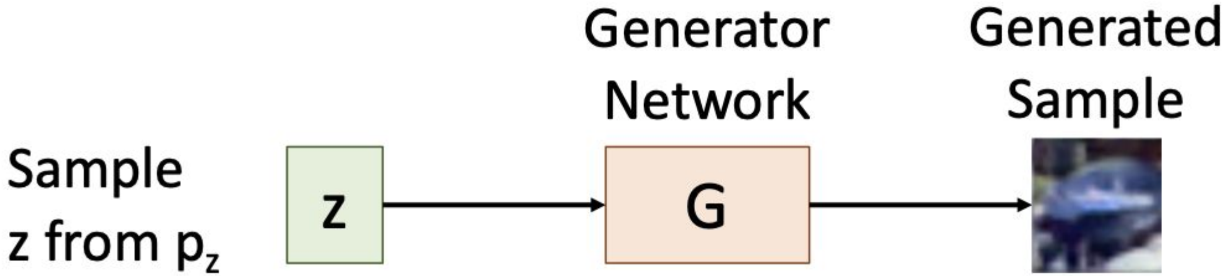
# GAN

- Generative Adversarial Network
- У нас есть X данные (изображения) из распределения  p
- Мы хотим научиться семплировать из p новые данные
- Мы не хотим знать/учить p, мы просто хотим новые данные



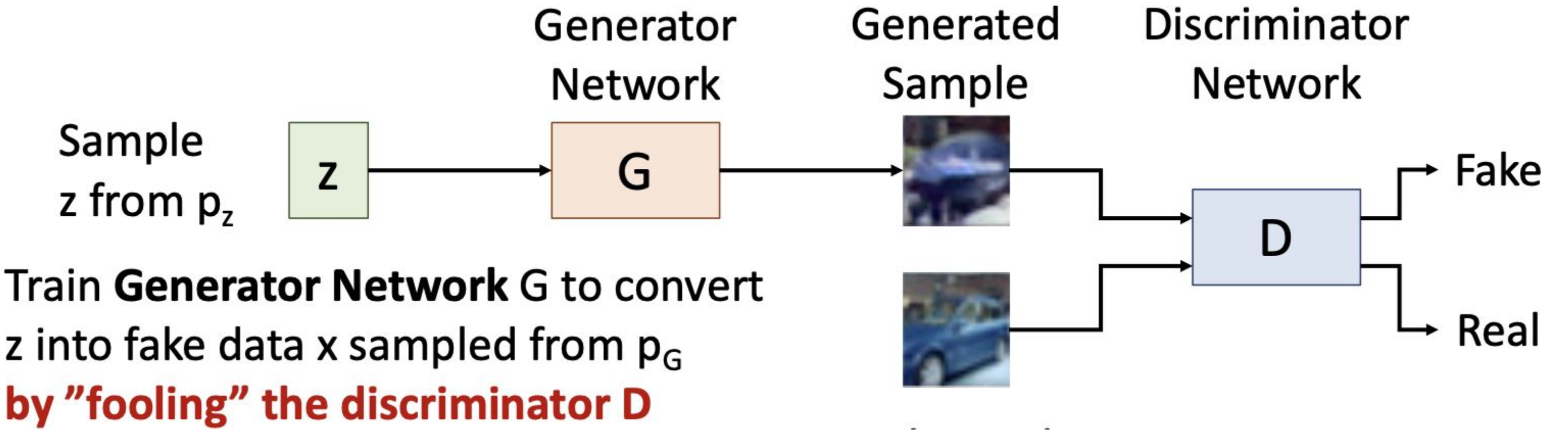
# GAN

- Generative Adversarial Network
- У нас есть X данные (изображения) из распределения  p
- Мы хотим научиться семплировать из p новые данные
- Мы не хотим знать/учить p, мы просто хотим новые данные



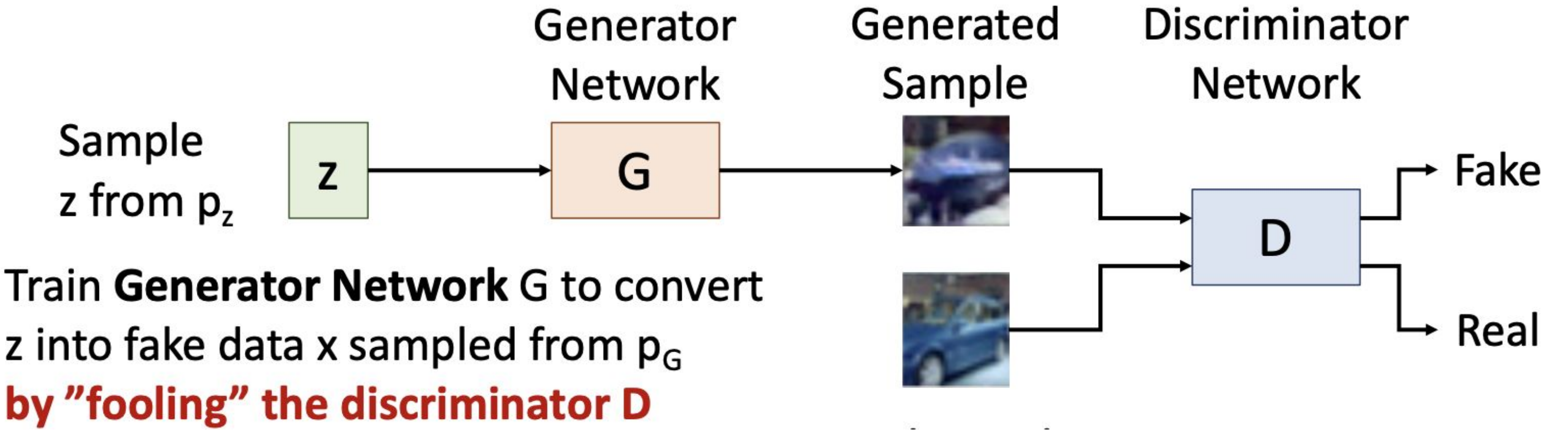

чего не хватает чтобы это обучить?

# GAN



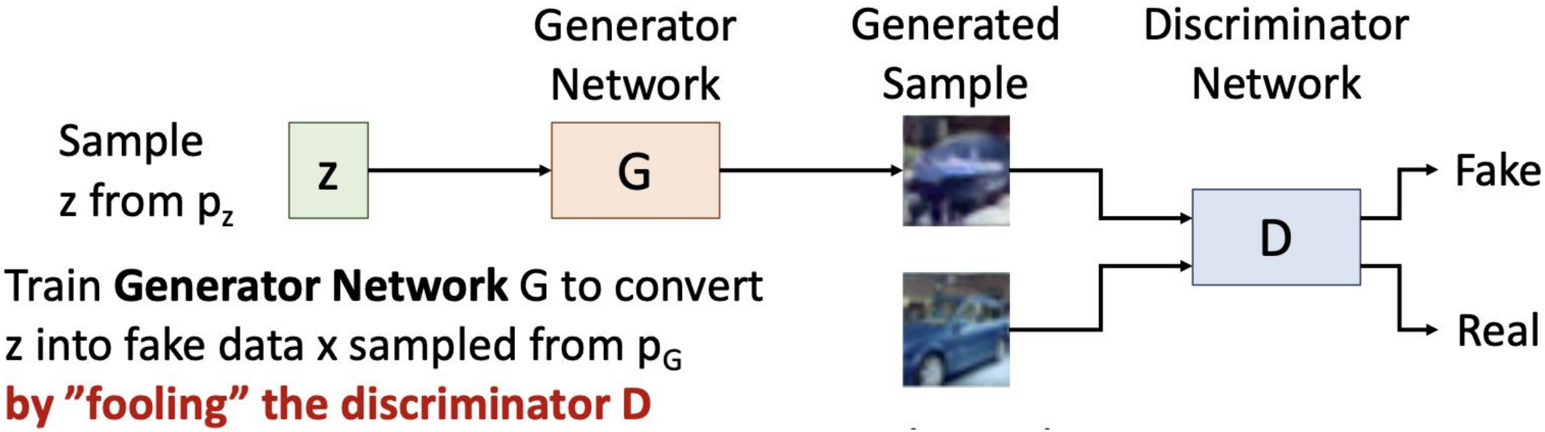

Не хватало loss function
Давайте добавим модель классификатор (в терминологии ганов - дискриминатор)
И будем обучать его параллельно генератору, фейковая картинка или настоящая

# GAN




Не хватало loss function
Давайте добавим модель классификатор (в терминологии ганов - дискриминатор)
И будем обучать его параллельно генератору, фейковая картинка или настоящая
Надеемся что они что то выучат :)

# GAN




Не хватало loss function
Давайте добавим модель классификатор (в терминологии ганов - дискриминатор)
И будем обучать его параллельно генератору, фейковая картинка или настоящая
Надеемся что они что то выучат :)

# GAN Loss

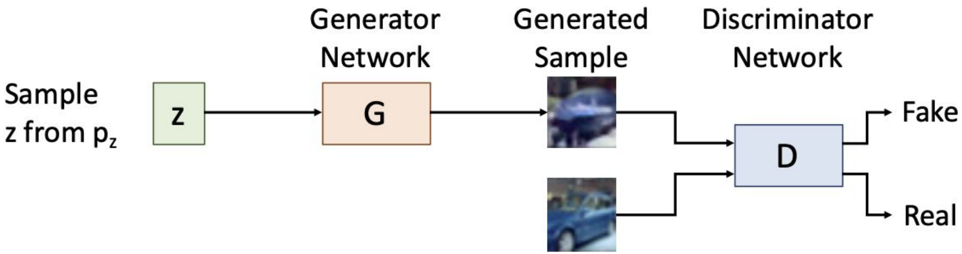
Jointly train generator G and discriminator D with a **minimax game**

$$\min_{G} \max_{D} \left( E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[ \log \left( 1 - D\big(G(z)\big) \right) \right] \right)$$

# GAN Loss

Jointly train generator G and discriminator D with a **minimax game**

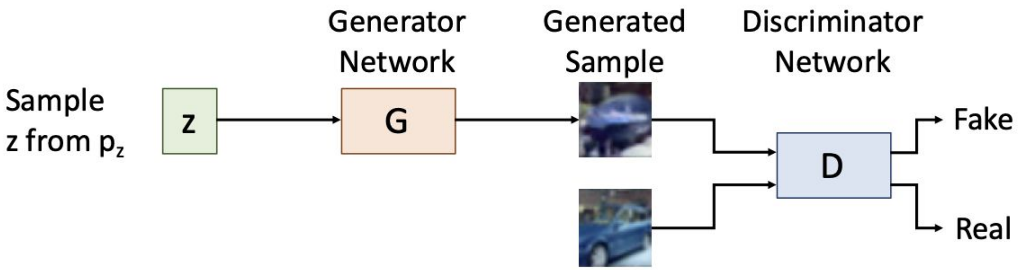
$$\min_{G} \max_{D} \left( E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[ \log \left( 1 - D(G(z)) \right) \right] \right)$$



# GAN Loss

Jointly train generator G and discriminator D with a **minimax game**

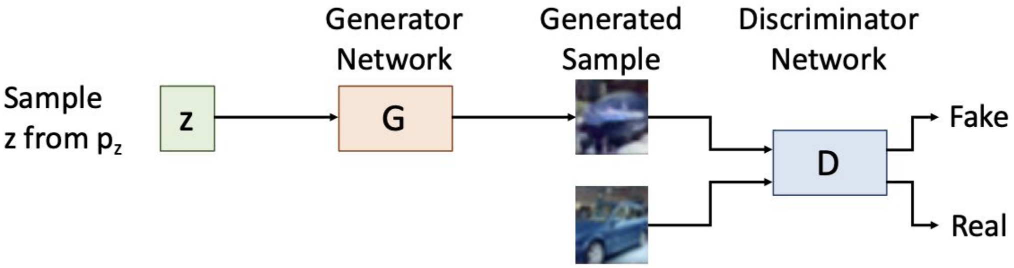
$$\min_{G} \max_{D} \left( \underbrace{E_{x \sim p_{data}} [\log D(x)]}^{\text{Discriminator wants } D(x) = 1 \text{ for real data}} + E_{z \sim p(z)} \left[ \log \left( 1 - D(G(z)) \right) \right] \right)$$



# GAN Loss

Jointly train generator G and discriminator D with a **minimax game**

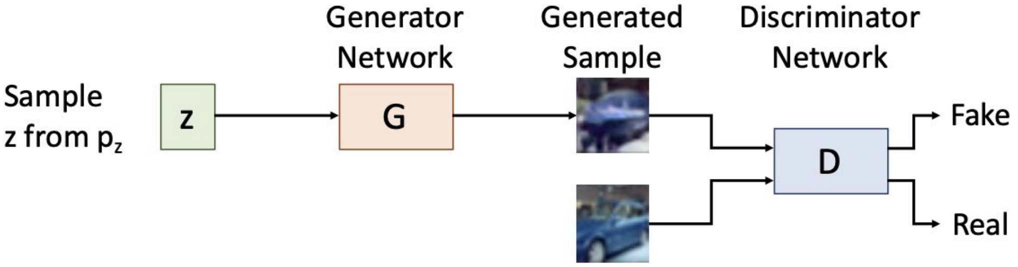
$$\min_{G} \max_{D} \left( \overbrace{E_{x\sim p_{data}}[\log D(x)]}^{\text{Discriminator wants } D(x) = 1 \text{ for real data}} + E_{z\sim p(z)} \overbrace{\left[\log\left(1 - D(G(z))\right)\right]}^{\text{Discriminator wants } D(x) = 0 \text{ for fake data}} \right)$$



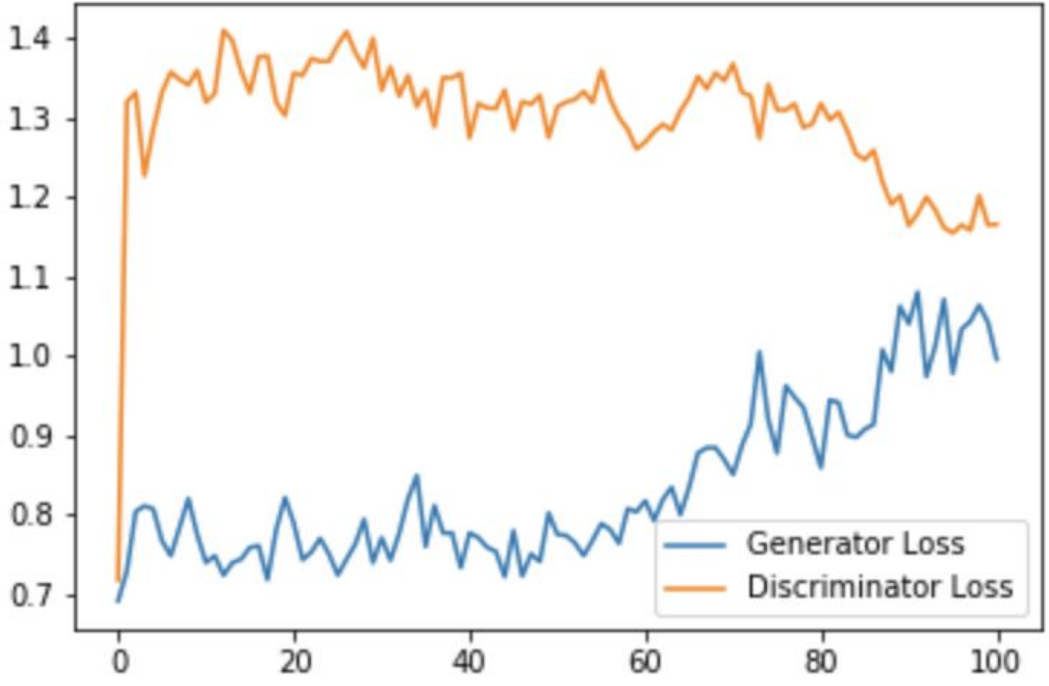
# GAN Loss

Jointly train generator G and discriminator D with a **minimax game**

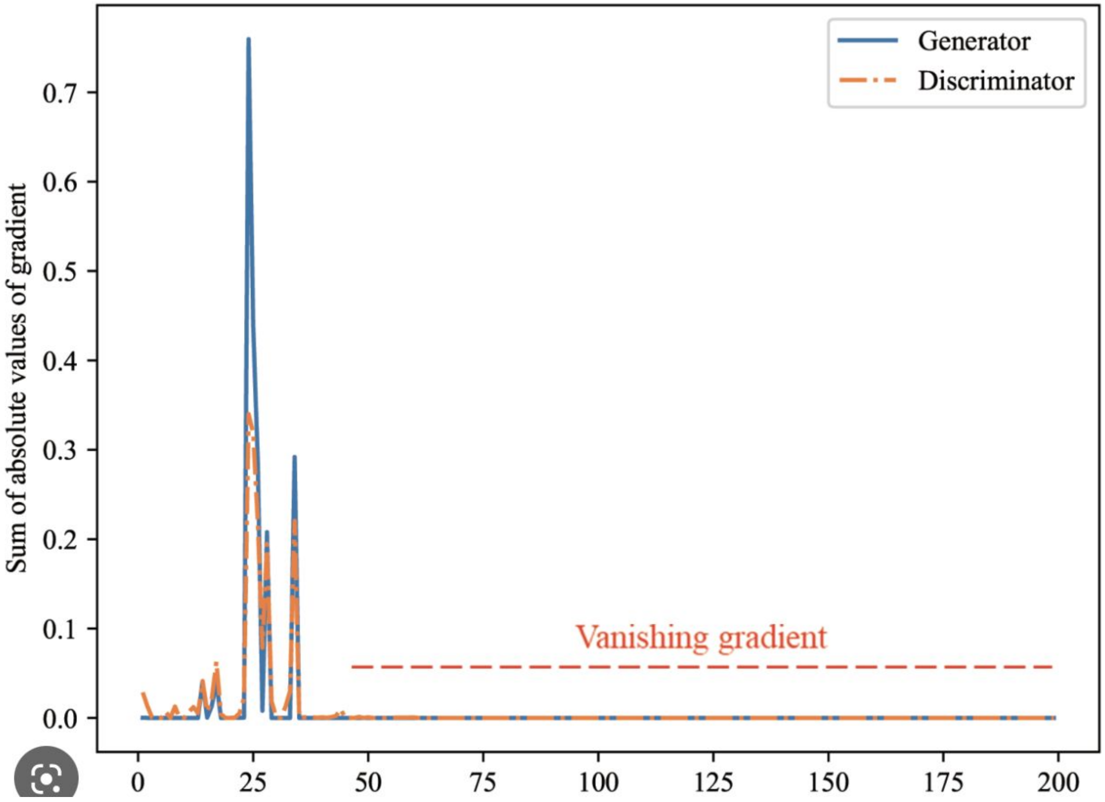
$$\min_{G} \max_{D} \left( \overbrace{E_{x\sim p_{data}}[\log D(x)]}^{\text{Discriminator wants } D(x) = 1 \text{ for real data}} + E_{z\sim p(z)} \overbrace{\underbrace{\left[\log\left(1 - D(G(z))\right)\right]}_{\text{Generator wants } D(x) = 1 \text{ for fake data}}}^{\text{Discriminator wants } D(x) = 0 \text{ for fake data}} \right)$$



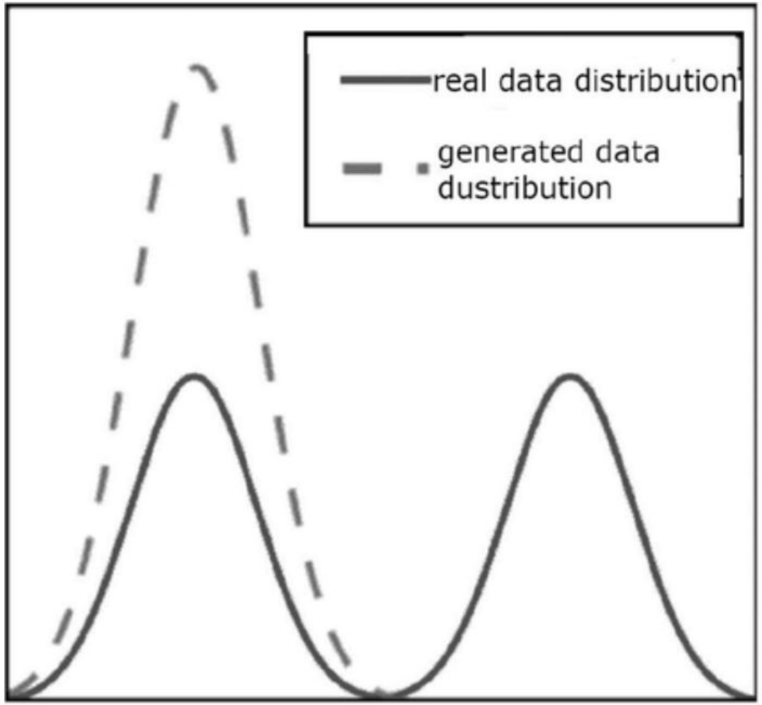
# Проблемы - нет одного лосса



# Проблемы - vanishing gradient



# Проблемы - mode collapse



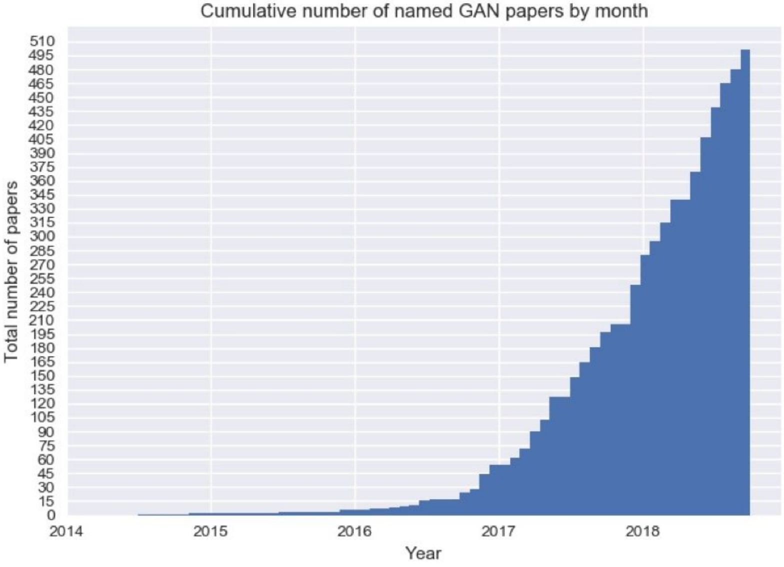

(a)

(b)

512 x 384 cars

1024 x 1024 faces

# 2017 to present: Explosion of GANs




https://github.com/hindupuravinash/the-gan-zoo

- 3D-ED-GAN - Shape Inpainting using 3D Generative Adversarial Network and Recurrent Convolutional Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling (github)
- 3D-IWGAN - Improved Adversarial Systems for 3D Object Generation and Reconstruction (github)
- 3D-PhysNet - 3D-PhysNet: Learning the Intuitive Physics of Non-Rigid Object Deformations
- 3D-RecGAN - 3D Object Reconstruction from a Single Depth View with Adversarial Learning (github)
- ABC-GAN - ABC-GAN: Adaptive Blur and Control for improved training stability of Generative Adversarial Networks (github)
- ABC-GAN - GANs for LIFE: Generative Adversarial Networks for Likelihood Free Inference
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- ACGAN - Coverless Information Hiding Based on Generative adversarial networks
- acGAN - On-line Adaptative Curriculum Learning for GANs
- ACtuAL - ACtuAL: Actor-Critic Under Adversarial Learning
- AdaGAN - AdaGAN: Boosting Generative Models
- Adaptive GAN - Customizing an Adversarial Example Generator with Class-Conditional GANs
- AdvEntuRe - AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples
- AdvGAN - Generating adversarial examples with adversarial networks
- AE-GAN - AE-GAN: adversarial eliminating with GAN
- AE-OT - Latent Space Optimal Transport for Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AF-DCGAN - AF-DCGAN: Amplitude Feature Deep Convolutional GAN for Fingerprint Construction in Indoor Localization System
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AIM - Generating Informative and Diverse Conversational Responses via Adversarial Information Maximization
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference (github)
- AlignGAN - AlignGAN: Learning to Align Cross-Domain Images with Conditional Generative Adversarial Networks
- AlphaGAN - AlphaGAN: Generative adversarial networks for natural image matting
- AM-GAN - Activation Maximization Generative Adversarial Nets
- AmbientGAN - AmbientGAN: Generative models from lossy measurements (github)
- AMC-GAN - Video Prediction with Appearance and Motion Conditions
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- APD - Adversarial Distillation of Bayesian Neural Network Posteriors
- APE-GAN - APE-GAN: Adversarial Perturbation Elimination with GAN
- ARAE - Adversarially Regularized Autoencoders for Generating Discrete Structures (github)
- ARDA - Adversarial Representation Learning for Domain Adaptation
- ARIGAN - ARIGAN: Synthetic Arabidopsis Plants using Generative Adversarial Network
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorial GANs
- ASDL-GAN - Automatic Steganographic Distortion Learning Using a Generative Adversarial Network
- ATA-GAN - Attention-Aware Generative Adversarial Networks (ATA-GANs)
- Attention-GAN - Attention-GAN for Object Transfiguration in Wild Images
- AttGAN - Arbitrary Facial Attribute Editing: Only Change What You Want (github)
- AttnGAN - AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks (github)
- AVID - AVID: Adversarial Visual Irregularity Detection
- B-DCGAN - B-DCGAN:Evaluation of Binarized DCGAN for FPGA
- b-GAN - Generative Adversarial Nets from a Density Ratio Estimation Perspective
- BAGAN - BAGAN: Data Augmentation with Balancing GAN
- Bayesian GAN - Deep and Hierarchical Implicit Models
- Bayesian GAN - Bayesian GAN (github)
- BCGAN - Bayesian Conditional Generative Adverserial Networks
- BCGAN - Bidirectional Conditional Generative Adversarial networks
- BEAM - Boltzmann Encoded Adversarial Machines
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BEGAN-CS - Escaping from Collapsing Modes in a Constrained Space
- Bellman GAN - Distributional Multivariate Policy Evaluation and Exploration with the Bellman

- BGAN - Binary Generative Adversarial Networks for Image Retrieval (github)
- Bi-GAN - Autonomously and Simultaneously Refining Deep Neural Network Parameters by a Bi-Generative Adversarial Network Aided Genetic Algorithm
- BicycleGAN - Toward Multimodal Image-to-Image Translation (github)
- BiGAN - Adversarial Feature Learning
- BinGAN - BinGAN: Learning Compact Binary Descriptors with a Regularized GAN
- BourGAN - BourGAN: Generative Networks with Metric Embeddings
- BranchGAN - Branched Generative Adversarial Networks for Multi-Scale Image Manifold Learning
- BRE - Improving GAN Training via Binarized Representation Entropy (BRE) Regularization (github)
- BridgeGAN - Generative Adversarial Frontal View to Bird View Synthesis
- BS-GAN - Boundary-Seeking Generative Adversarial Networks
- BubGAN - BubGAN: Bubble Generative Adversarial Networks for Synthesizing Realistic Bubbly Flow Images
- BWGAN - Banach Wasserstein GAN
- C-GAN - Face Aging with Contextual Generative Adversarial Nets
- C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training (github)
- CA-GAN - Composition-aided Sketch-realistic Portrait Generation
- CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks (github)
- CAN - CAN: Creative Adversarial Networks, Generating Art by Learning About Styles and Deviating from Style Norms
- CapsGAN - CapsGAN: Using Dynamic Routing for Generative Adversarial Networks
- CapsuleGAN - CapsuleGAN: Generative Adversarial Capsule Network
- CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CatGAN - CatGAN: Coupled Adversarial Transfer for Domain Generation
- CausalGAN - CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training
- CC-GAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks (github)
- cd-GAN - Conditional Image-to-Image Translation
- CDcGAN - Simultaneously Color-Depth Super-Resolution with Conditional Generative Adversarial Network
- CE-GAN - Deep Learning for Imbalance Data Classification using Class Expert Generative Adversarial Network
- CFG-GAN - Composite Functional Gradient Learning of Generative Adversarial Models
- CGAN - Conditional Generative Adversarial Nets
- CGAN - Controllable Generative Adversarial Network
- Chekhov GAN - An Online Learning Approach to Generative Adversarial Networks
- ciGAN - Conditional Infilling GANs for Data Augmentation in Mammogram Classification
- CinCGAN - Unsupervised Image Super-Resolution using Cycle-in-Cycle Generative Adversarial Networks
- CipherGAN - Unsupervised Cipher Cracking Using Discrete GANs
- ClusterGAN - ClusterGAN : Latent Space Clustering in Generative Adversarial Networks
- CM-GAN - CM-GANs: Cross-modal Generative Adversarial Networks for Common Representation Learning
- CoAtt-GAN - Are You Talking to Me? Reasoned Visual Dialog Generation through Adversarial Learning
- CoGAN - Coupled Generative Adversarial Networks
- ComboGAN - ComboGAN: Unrestrained Scalability for Image Domain Translation (github)
- ConceptGAN - Learning Compositional Visual Concepts with Mutual Consistency
- Conditional cycleGAN - Conditional CycleGAN for Attribute Guided Face Image Generation
- constrast-GAN - Generative Semantic Manipulation with Contrasting GAN
- Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- CorrGAN - Correlated discrete data generation using adversarial training
- Coulomb GAN - Coulomb GANs: Provably Optimal Nash Equilibria via Potential Fields
- Cover-GAN - Generative Steganography with Kerckhoffs' Principle based on Generative Adversarial Networks
- cowboy - Defending Against Adversarial Attacks by Leveraging an Entire GAN
- CR-GAN - CR-GAN: Learning Complete Representations for Multi-view Generation
- Cramèr GAN - The Cramer Distance as a Solution to Biased Wasserstein Gradients
- Cross-GAN - Crossing Generative Adversarial Networks for Cross-View Person Re-identification
- crVAE-GAN - Channel-Recurrent Variational Autoencoders
- CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CSG - Speech-Driven Expressive Talking Lips with Conditional Sequential Generative Adversarial Networks
- CT-GAN - CT-GAN: Conditional Transformation Generative Adversarial Network for Image Attribute Modification
- CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks