



ОНЛАЙН-ОБРАЗОВАНИЕ

Генеративные Состязательные Сети

Я тебя породил, я тебя и... отличу!

Артур Кадурин
Преподаватель

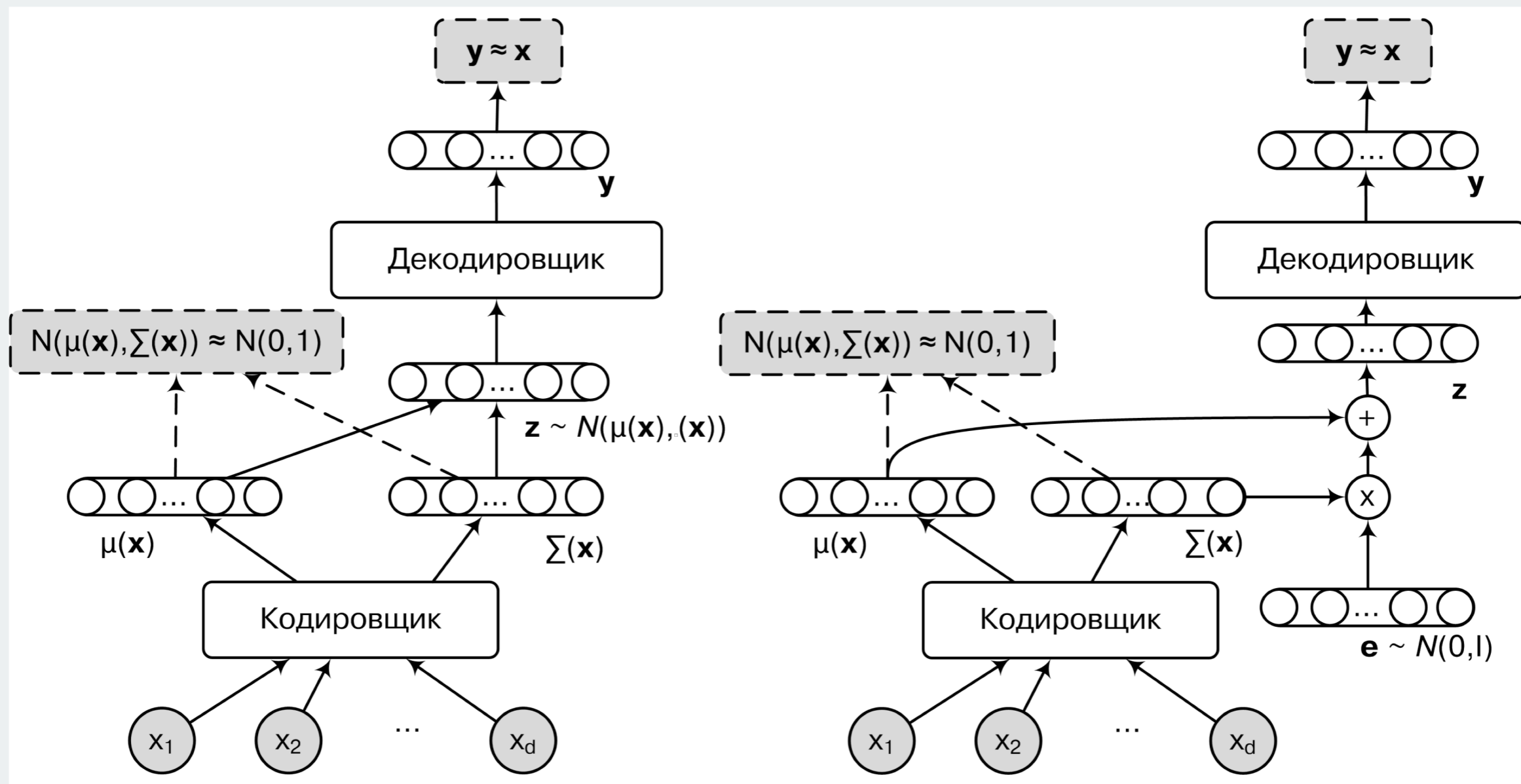


1. **Репараметризация**
2. Формула Байеса
3. Генеративные Состязательные Сети
4. DCGAN
5. Практика
6. Состязательный Автокодировщик



$$D_{KL}(N[0,1] \parallel q(x)) = \frac{1}{2} [tr(\Sigma_q) + \mu_q^T \mu_q - k - \log \det \Sigma_q] =$$
$$\frac{1}{2} \sum_{i=1}^k [\sigma^2(x) + \mu^2(x) - \mathbf{1} - \log \sigma^2(x)]$$





$$D_{KL}(N[0,1] \parallel q(x)) = \frac{1}{2} [tr(\Sigma_q) + \mu_q^T \mu_q - k - \log \det \Sigma_q] =$$
$$\frac{1}{2} \sum_{i=1}^k [e^{\log var(x)} + \mu^2(x) - \mathbf{1} - \log var(x)]$$



1. Репараметризация
2. **Формула Байеса**
3. Генеративные Состязательные Сети
4. DCGAN
5. Практика
6. Состязательный Автокодировщик



$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Теорема Байеса позволяет перейти от априорного распределения к апостериорному, но при этом еще позволяет связать между собой условные распределения.



$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

$p(y)$ — априорное распределение над классами:

Какова вероятность выпадения орла/решки

Какая доля людей болеет раком груди

$p(x)$ — априорное распределение над объектами:

Вероятность выбора монеты достоинством 1р./2р./5р. ...

Вероятность выбора конкретного человека для анализа



$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

$p(x|y)$ — условное распределение над объектами:

Если выпала решка, то какова вероятность что выбрали 5р.

Если рак груди обнаружен, то какова вероятность что это Иван Иванович



$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

$p(y|x)$ — условное распределение над классами:

Если выбрали 5р., то какова вероятность орла/решки

Если обследуем Ивана Ивановича, то какова вероятность обнаружить рак груди



$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Дискриминативная модель — используя обучающую выборку аппроксимируем $p(y|x)$ с помощью нейросети $q_{\theta}(y|x)$



$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Дискриминативная модель — используя обучающую выборку аппроксимируем $p(y|x)$ с помощью нейросети $q_{\theta}(y|x)$

$$p(y|x) \sim p(x|y)p(y)$$

Генеративная модель — используя обучающую выборку оцениваем $p(y)$ и аппроксимируем $p(y|x)$ с помощью нейросети $q_{\theta}(x|y)$



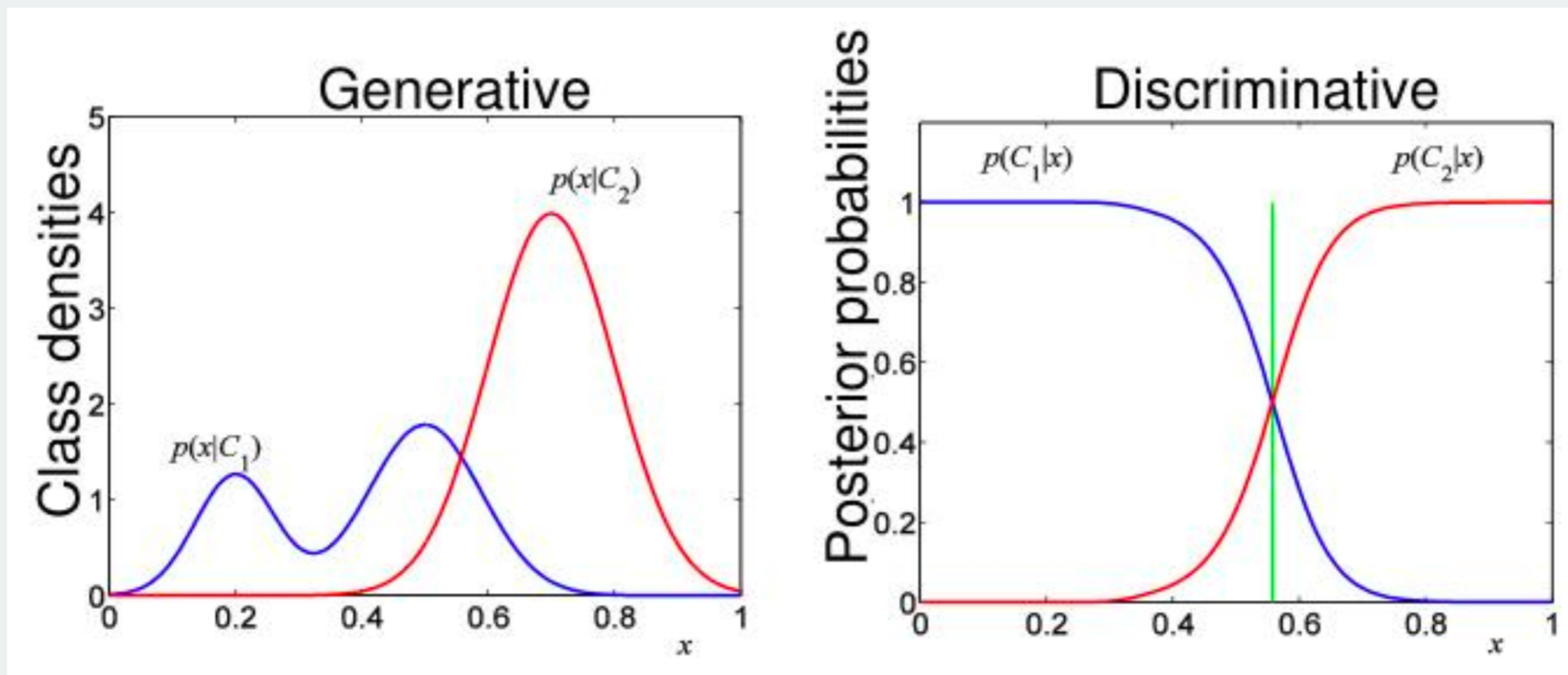
$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Дискриминативная модель — используя обучающую выборку аппроксимируем $p(y|x)$ с помощью нейросети $q_{\theta}(y|x)$

$$p(y|x) \sim p(x|y)p(y)$$

Генеративная модель — используя обучающую выборку оцениваем $p(y)$ и аппроксимируем $p(y|x)$ с помощью нейросети $q_{\theta}(x|y)$





1. Репараметризация
2. Формула Байеса
- 3. Генеративные Состязательные Сети**
4. DCGAN
5. Практика
6. Состязательный Автокодировщик



А теперь забудьте то о чем мы говорили в предыдущей секции.

Генеративные Состязательные Сети называются так не потому, что они относятся к генеративным моделям, а потому, что они позволяют генерировать новые объекты. Т.е. наша цель не обучить сеть $q_{\theta}(x|y)$ предсказывать вероятность объекта по какому-то условию, а обучить сеть $G_{\theta}(y): Y \rightarrow X$, которая по заданному условию порождает новый объект.



Нашей целью будет сеть, которая может порождать объекты похожие на то что есть в обучающей выборке. В простейшем случае, в качестве условия, мы можем использовать нормальный шум и все что нам остается сделать, это задать подходящую функцию ошибки.

Как можно задавать функции ошибки?

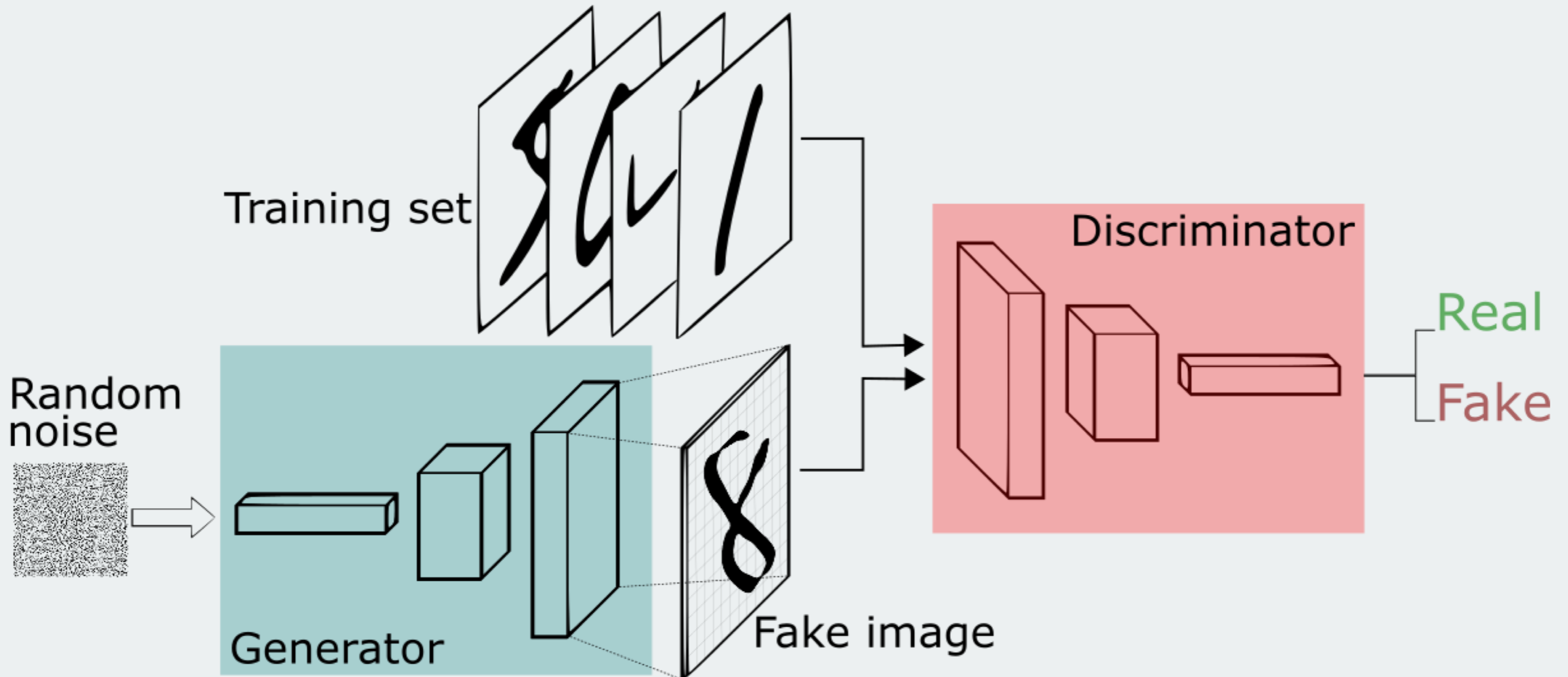


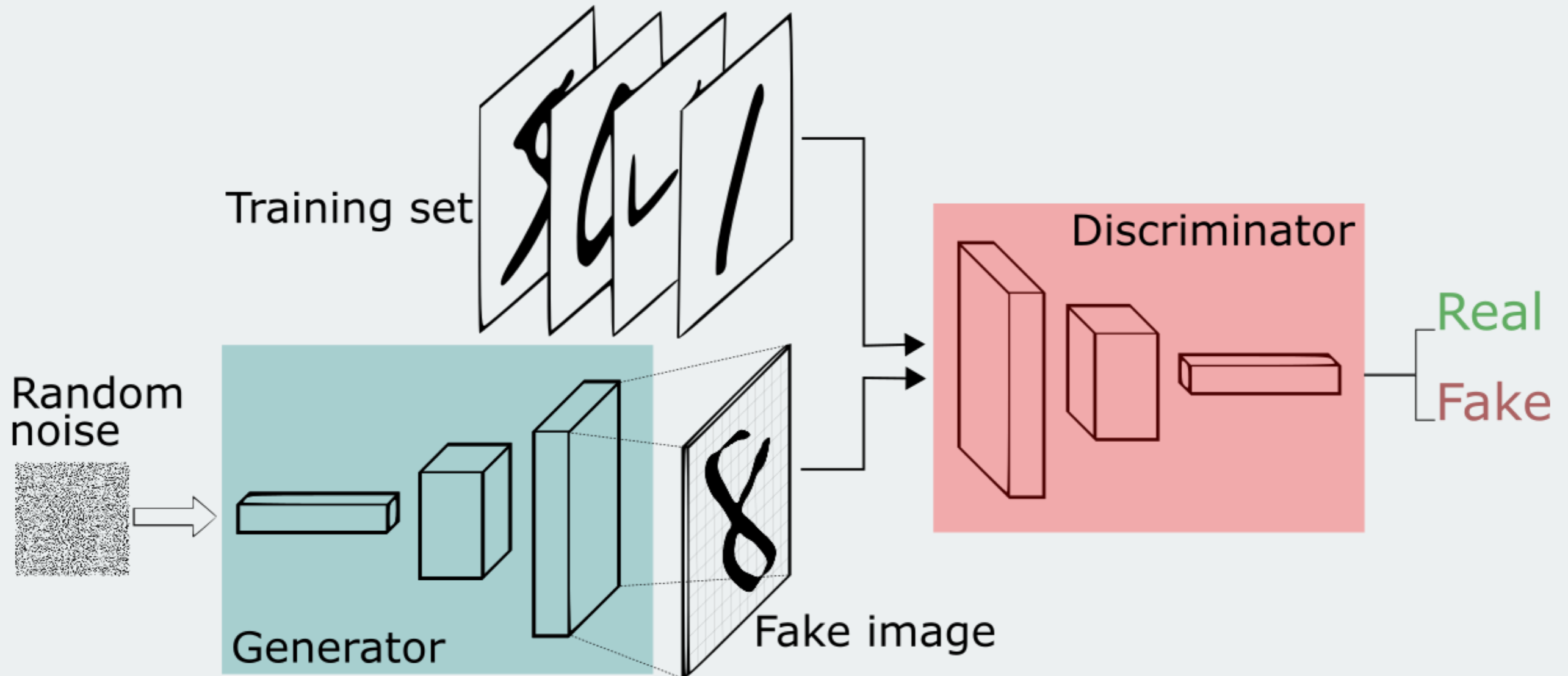
Нашей целью будет сеть, которая может породить объекты похожие на то что есть в обучающей выборке. В простейшем случае, в качестве условия, мы можем использовать нормальный шум и все что нам остается сделать, это задать подходящую функцию ошибки.

Как можно задавать функции ошибки?

Нейросетью!







$$\mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{x \sim p_{gen}(x)} [\log(1 - D(x))]$$



$$\mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{x \sim p_{gen}(x)} [\log(1 - D(x))]$$

$x \sim p_{gen}(x)$ — это то же самое, что и $G(z)$, $z \sim p_z$



$$\mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{x \sim p_{gen}(x)} [\log(1 - D(x))]$$

$x \sim p_{gen}(x)$ — это то же самое, что и $G(z)$, $z \sim p_z$

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

С одной стороны мы хотим максимизировать точность

Дискриминатора, с другой — максимизировать шанс Генератора

«обмануть» Дискриминатор.



Устойчивость: минимаксная игра — поиск седловой точки

Mode collapse: часто Генератору оказывается достаточно генерировать только часть датасета

Насыщение Дискриминатора: градиенты от хорошего Дискриминатора могут быть нулевыми



Подбор гиперпараметров: баланс глубины, количества шагов обучения, оптимизаторов и т.д.

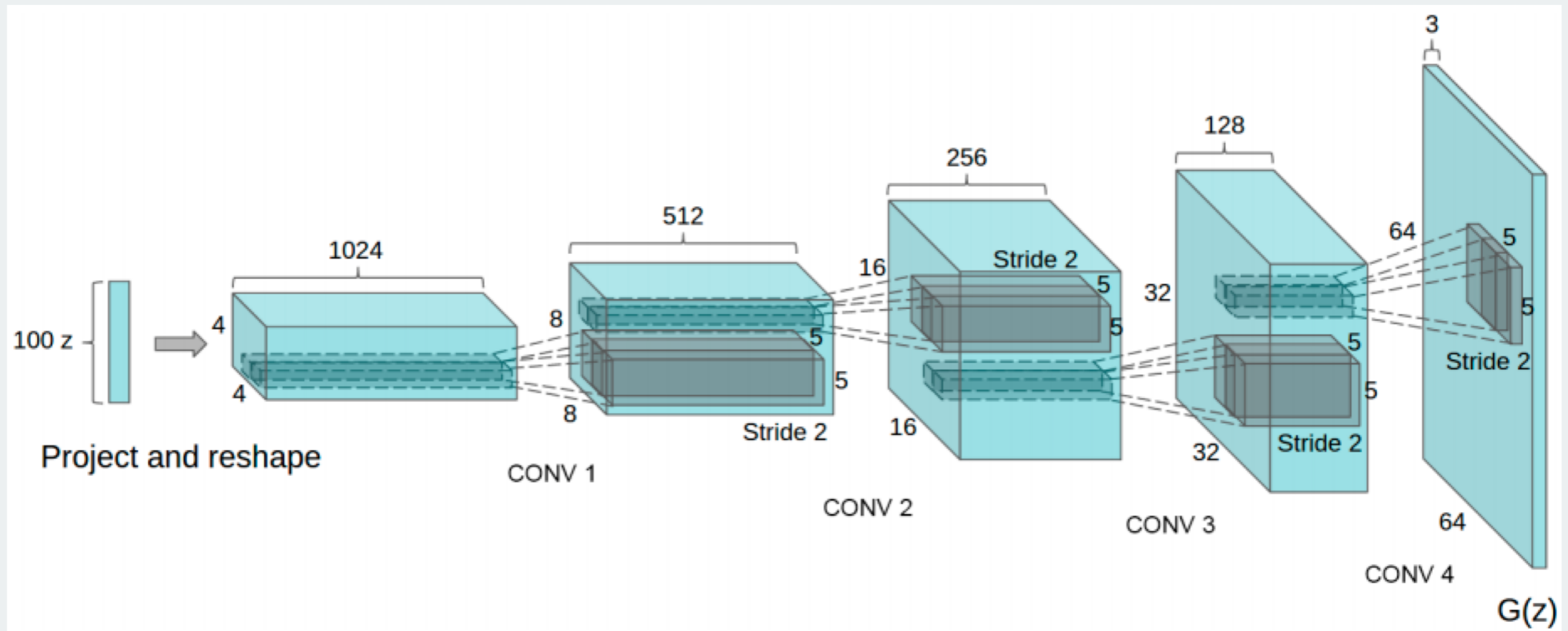
Другие функции потерь: *Wassershtein loss*, *statistics matching*

Слои: дропаут, батч нормализация и т.д.



1. Репараметризация
2. Формула Байеса
3. Генеративные Состязательные Сети
4. **DCGAN**
5. Практика
6. Состязательный Автокодировщик





<https://arxiv.org/abs/1511.06434>



1. Ни каких пуллинг слоев как в Генераторе, так и в Дискриминаторе
2. Нет полносвязных слоев
3. Batch Normalization на всех слоях кроме выхода G и входа D
4. ReLU в Генераторе и LeakyReLU в Дискриминаторе

<https://arxiv.org/abs/1511.06434>



DCGAN: Примеры



<https://arxiv.org/abs/1511.06434>





<https://arxiv.org/abs/1511.06434>



1. Репараметризация
2. Формула Байеса
3. Генеративные Состязательные Сети
4. DCGAN
- 5. Практика**
6. Состязательный Автокодировщик



1. Репараметризация
2. Формула Байеса
3. Генеративные Состязательные Сети
4. DCGAN
5. Практика
6. **Состязательный Автокодировщик**



Мы уже умеем использовать нейросеть в качестве функции ошибки.
Что если заменить KL-дивергенцию в Вариационном Автокодировщике на нейросеть?

Чему надо учить Дискриминатор?



Мы уже умеем использовать нейросеть в качестве функции ошибки.
Что если заменить KL-дивергенцию в Вариационном Автокодировщике на нейросеть?

Чему надо учить Дискриминатор?

Отличать выходы Энкодера от сэмплов из $N[0,1]$





Спасибо
за внимание!