

# Введение в генеративные модели

Денисеко Наталья

# План выступления

- Общее определение генеративных моделей
- Примеры моделей
  - VAE
  - PixelRNN
  - PixelCNN
- Расскажу о их плюсах и минусах

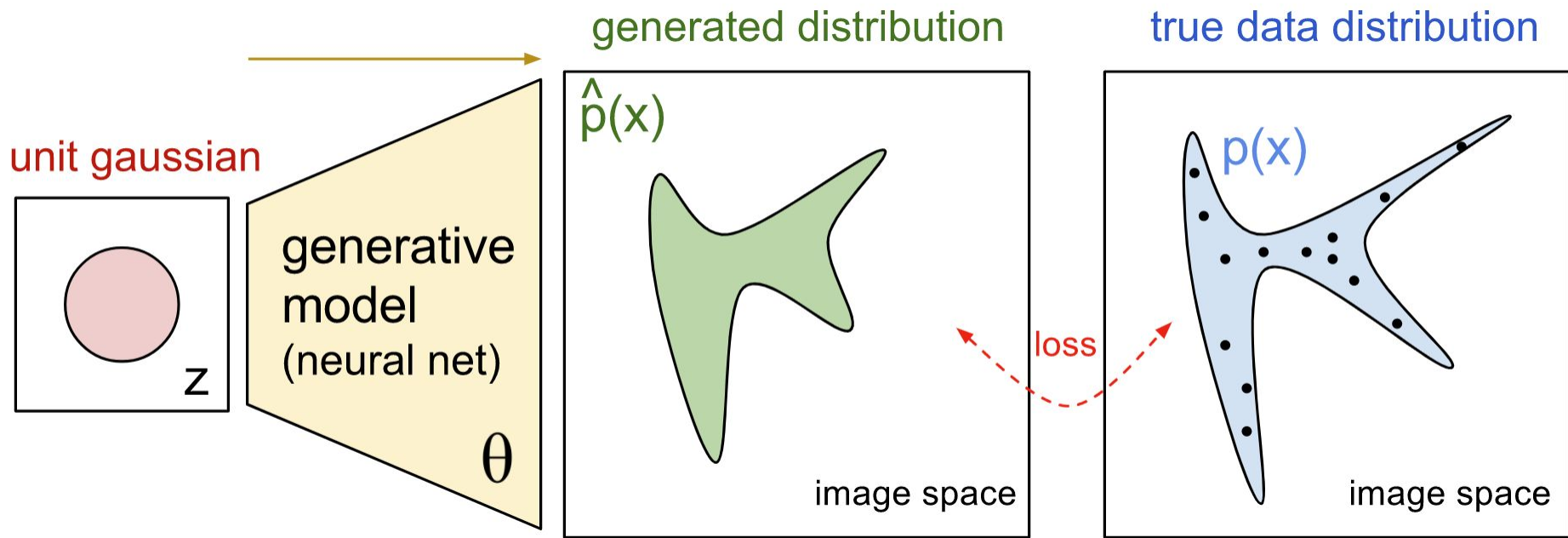
# Общее определение



# Основная цель

Мы хотим на основе собранных данных, научиться генерировать новые данные из выборки.

Так как параметров модели намного меньше, чем предоставленных примеров из набора данных, то она будет вынуждена найти самые важные признаки данных.



# Две модели

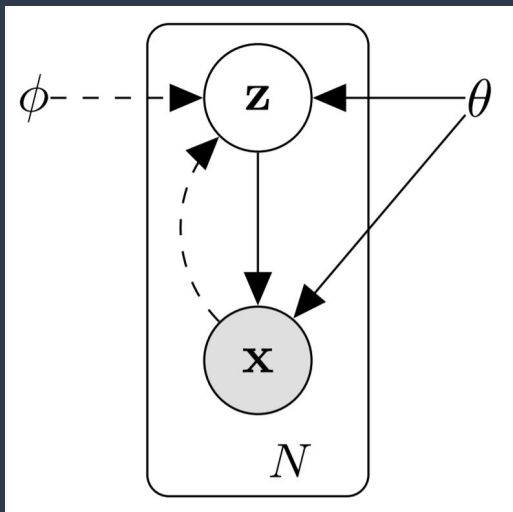
Можно выделить два основных подхода:

- Генеративная модель
  - модель условной вероятности  $X$ , по целевой переменной  $y$
  - $P(X|Y=y)$
- Дискриминативная модель
  - модель условной вероятности  $Y$ , по наблюдаемой переменной  $x$
  - $P(Y|X=x)$

# Примеры



# VAE



Мы предполагаем, что элементы датасета могут быть получены с помощью некоторого процесса, с использованием неизвестной непрерывной случайной величины  $z$ .

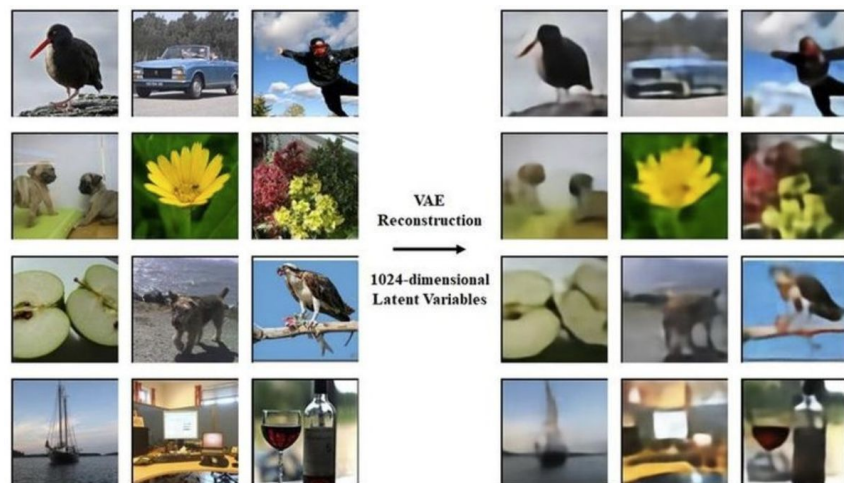
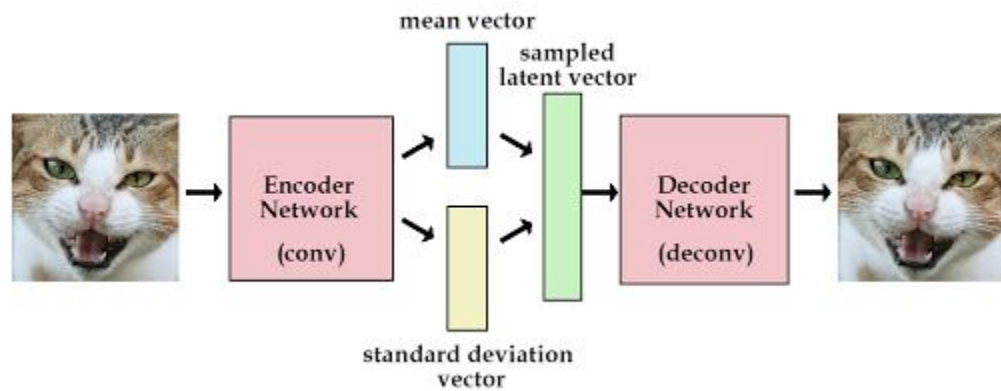
Процесс делится на два этапа:

- новое  $z$  получается из предыдущего распределения  $p_{\theta^*}(z)$  (1)
- значение  $x$  получается из некоторого условного распределения  $p_{\theta^*}(x|z)$  (2)

Считается что и (1) и (2) принадлежат параметрическим распределениям

$$p_{\theta^*}(z) \quad p_{\theta^*}(x|z)$$





Предельная вероятность отдельного значения может быть записана как  $\log p_\theta(x^{(i)}) = D_{KL}(q_\varphi(z|x^{(i)})||p_\theta(z|x^{(i)})) + \mathcal{L}(\theta, \varphi; x^{(i)})$

$$\mathcal{L}(\theta, \varphi; x) = -D_{KL}(q_\varphi(z|x)||p_\theta(z)) + \mathbb{E}_{q_\varphi(z|x)}[\log p_\theta(x|z)]$$

дословно переводится как нижняя граница предельной вероятности. Её мы и пытаемся оптимизировать

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})} [f(\mathbf{z})] = \mathbb{E}_{q_\phi(\mathbf{z})} [f(\mathbf{z}) \nabla_{q_\phi(\mathbf{z})} \log q_\phi(\mathbf{z})] \simeq \frac{1}{L} \sum_{l=1}^L f(\mathbf{z}) \nabla_{q_\phi(\mathbf{z}^{(l)})} \log q_\phi(\mathbf{z}^{(l)})$$

SGVB приближение нижней границы

$$\begin{aligned}\tilde{\mathcal{L}}^A(\theta, \varphi; x^{(i)}) &= \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x^{(i)}, z^{(i,l)}) - \log q_{\varphi}(z^{(i,l)} | x^{(i)}) \\ z^{(i,l)} &= g_{\varphi}(\epsilon^{(i,l)}, x^{(i)}) \\ \epsilon^{(l)} &\sim p(\epsilon)\end{aligned}$$

$$\tilde{\mathcal{L}}^B(\theta, \phi; \mathbf{x}^{(i)}) = -D_{KL}(q_{\phi}(\mathbf{z} | \mathbf{x}^{(i)}) || p_{\theta}(\mathbf{z})) + \frac{1}{L} \sum_{l=1}^L (\log p_{\theta}(\mathbf{x}^{(i)} | \mathbf{z}^{(i,l)}))$$

приближение с использованием небольшого набора данных (здесь  $\mathbf{X}$  содержит  $N$  элементов, а  $\mathbf{X}^M$  соответственно  $M$ )

$$\mathcal{L}(\theta, \phi; \mathbf{X}) \simeq \tilde{\mathcal{L}}^M(\theta, \phi; \mathbf{X}^M) = \frac{N}{M} \sum_{i=1}^M \tilde{\mathcal{L}}(\theta, \phi; \mathbf{x}^{(i)})$$

$\theta, \phi \leftarrow$  Initialize parameters

**repeat**

$\mathbf{X}^M \leftarrow$  Random minibatch of  $M$  datapoints (drawn from full dataset)

$\epsilon \leftarrow$  Random samples from noise distribution  $p(\epsilon)$

$\mathbf{g} \leftarrow \nabla_{\theta, \phi} \tilde{\mathcal{L}}^M(\theta, \phi; \mathbf{X}^M, \epsilon)$  (Gradients of minibatch estimator (8))

$\theta, \phi \leftarrow$  Update parameters using gradients  $\mathbf{g}$  (e.g. SGD or Adagrad [DHS10])

**until** convergence of parameters  $(\theta, \phi)$

**return**  $\theta, \phi$

Для VAE мы считаем, что распределение  $p$  это либо многомерное нормальное, либо бернулли.

В этом случае, мы так же считаем, что и распределение  $q$  многомерное нормальное

$$\log q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)}) = \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\mathbf{I})$$

Мы подбираем  $\mathbf{z}$  с использованием MLP (Многослойный перцептрон Румельхарта) и  $\boldsymbol{\mu}^{(i)}$   $\boldsymbol{\sigma}^{(i)}$  будут получены на выходе данной модели

Как ранее описано  $\mathbf{z}^{(i,l)} \sim q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})$  а значит и

$$g_{\phi}(\mathbf{x}^{(i)}, \boldsymbol{\epsilon}^{(l)}) = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)} \text{ where } \boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

А значит, оценщик нашей модели

$$\mathcal{L}(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) \simeq \frac{1}{2} \sum_{j=1}^J \left( 1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2 \right) + \frac{1}{L} \sum_{l=1}^L \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})$$

# Плюсы и минусы

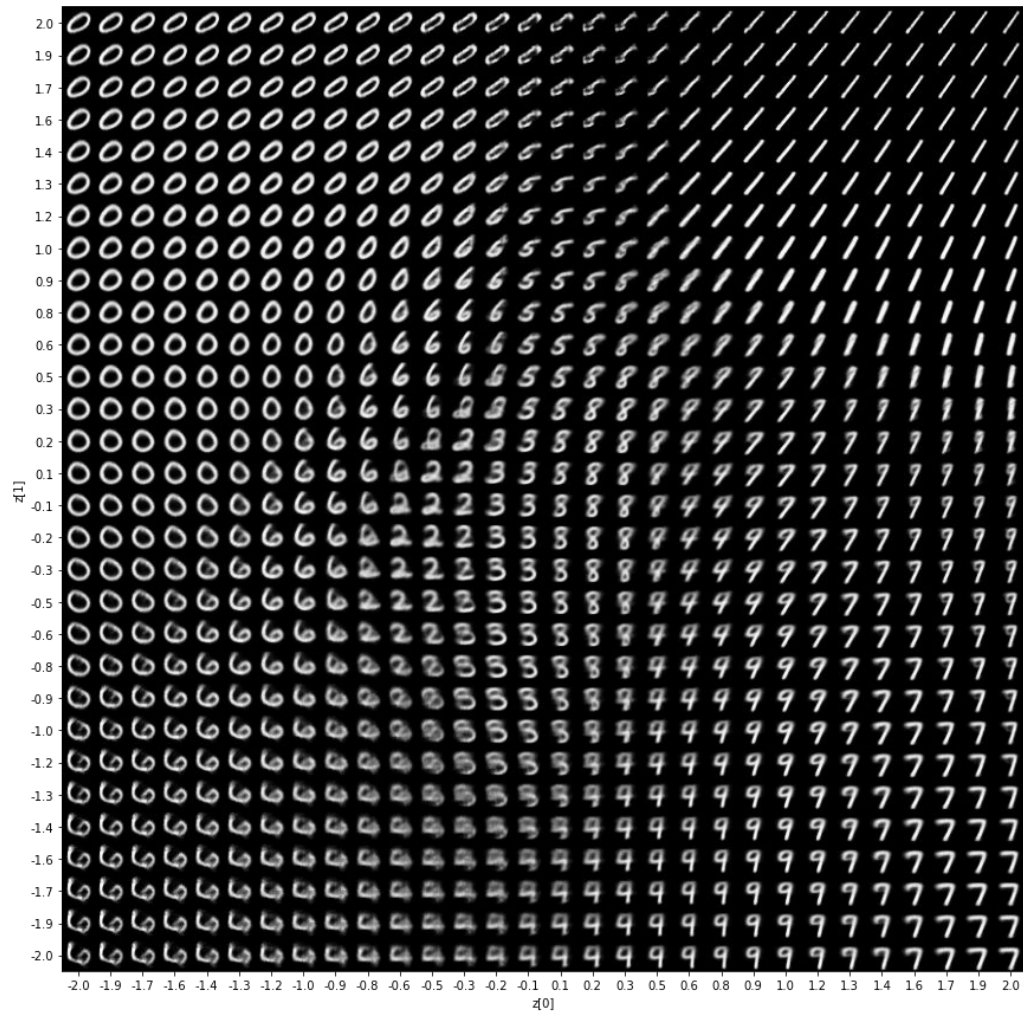
+

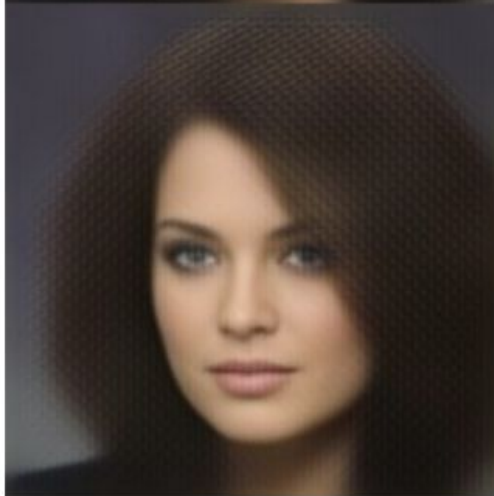
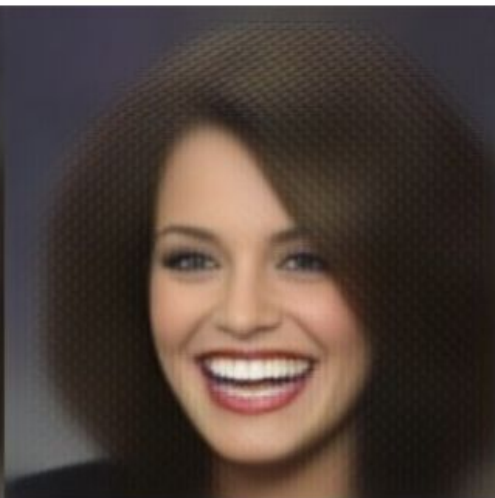
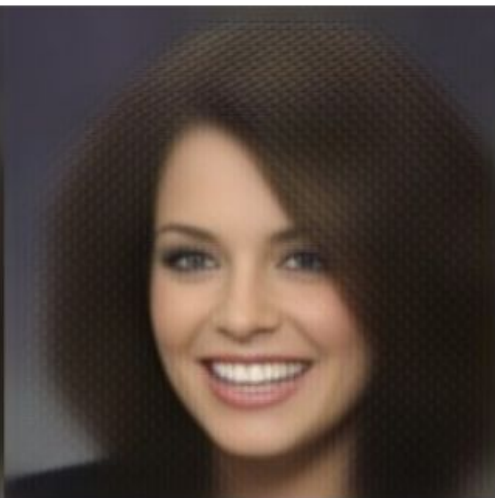
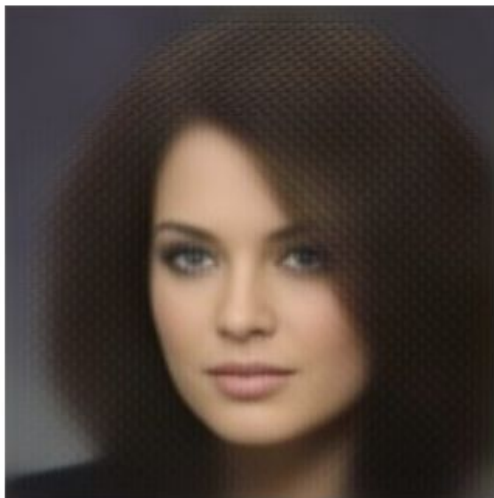
- просто определить качество модели без визуализации сгенерированных данных

-

- полученные на выход данные размыты в случае изображений

# Примеры

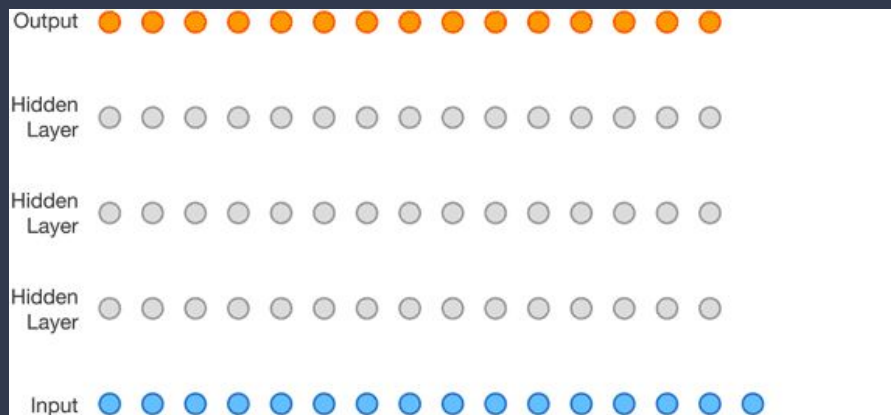




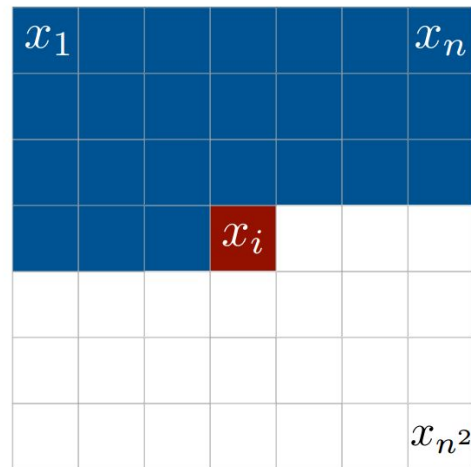


# PixelRNN/PixelCNN

Данные модели предсказывают каждый пиксель изображения на основе ранее предсказанных.

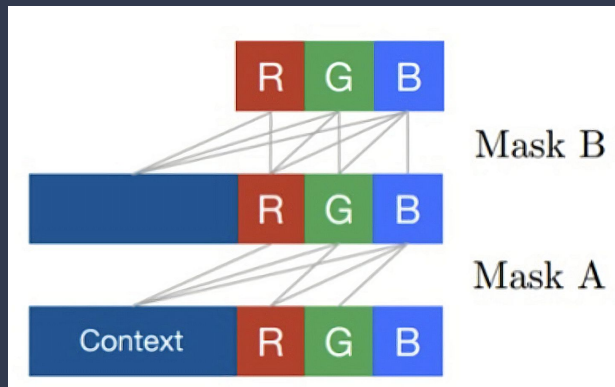


# Основная идея



$$p(x) = \prod_{i=1}^{n^2} p(x_i | x_1, \dots, x_{i-1})$$

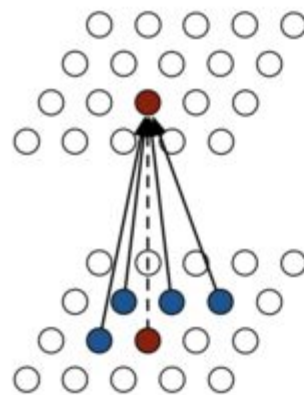
# Виды масок



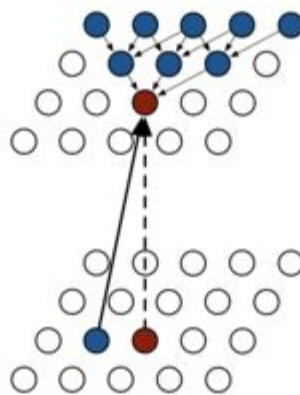
Маска первого типа **не** соединяет каналы с самими собой (mask A)

Маска второго типа соединяет каналы с самими собой (mask B)

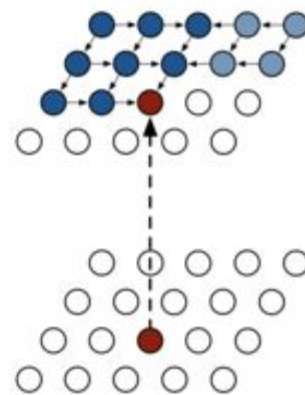
# Виды архитектур



PixelCNN



Row LSTM



Diagonal BiLSTM

PixelCNN	Row LSTM	Diagonal BiLSTM
7x7 conv mask A		
3x3 conv mask B	Row LSTM	Diagonal BiLSTM
	3x1 conv mask B	1x1 conv mask B
	3x1 conv no mask	1x2 conv no mask
ReLU + 1x1 conv mask B		
256 softmax for each RGB color		

# Сравнение

PixelCNN	Row LSTM	Diagonal BiLSTM
абсолютная зависимость	треугольная зависимость	абсолютная зависимость
самое быстрое	медленное	самое медленное
худшее качество	so so	лучшее качество

# Плюсы и минусы

+

- хорошо работает с изображениями
- просто определить качество модели без визуализации сгенерированных данных
- работает как для дискретных, так и для непрерывных данных

-

- не умеет создавать изображение “с нуля”
- долго обучается

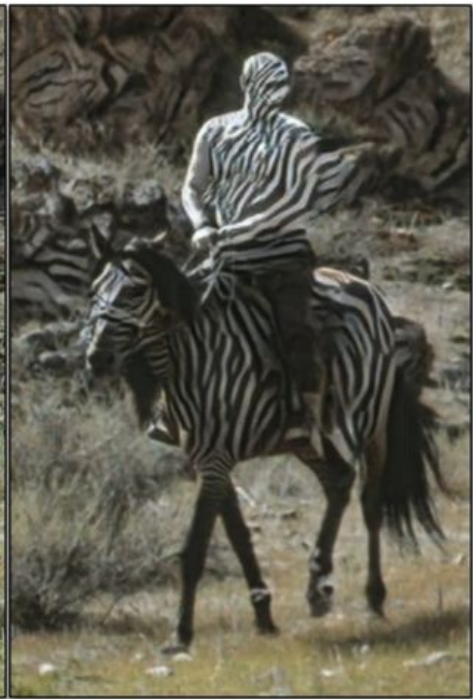
# Примеры

occluded

completions

original







# Список использованных медиа

- “Auto-Encoding Variational Bayes” Diederik P Kingma, Max Welling
- <https://openai.com/blog/generative-models/>
- “Stochastic gradient variational bayes for deep learning-based asr” Andros Tjandra1 , Sakriani Sakti2 , Satoshi Nakamura2 , Mirna Adriani
- <https://towardsdatascience.com/auto-regressive-generative-models-pixelrnn-pixelcnn-32d192911173>
- <https://habr.com/ru/post/331552/>
- “Pixel Recurrent Neural Networks” Aaron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu