

Deep Generative Models

Lecture 1

Roman Isachenko

Ozon Masters

2021

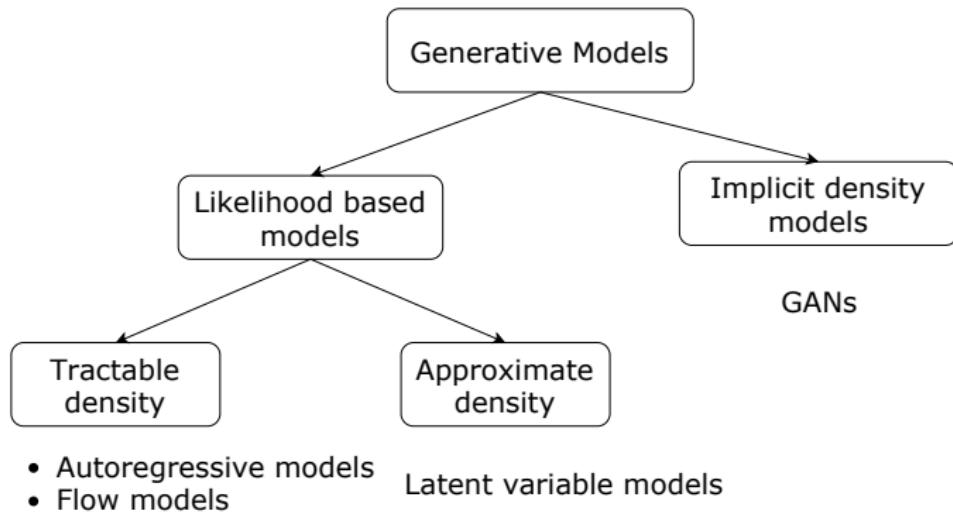
Logistics

- ▶ homeworks: 30 points
 - ▶ hw1: autoregressive models
 - ▶ hw2: latent variable models
 - ▶ hw3: flow models
 - ▶ hw4: adversarial models
- ▶ exam: 30 points
- ▶ final project: 40 points

Last year course page: [link](#)

Admission: [link](#)

Generative models zoo



Motivation

■ "Pure" Reinforcement Learning (cherry)

- ▶ The machine predicts a scalar reward given once in a while.
- ▶ **A few bits for some samples**

■ Supervised Learning (icing)

- ▶ The machine predicts a category or a few numbers for each input
- ▶ Predicting human-supplied data
- ▶ **10→10,000 bits per sample**

■ Unsupervised/Predictive Learning (cake)

- ▶ The machine predicts any part of its input for any observed part.
- ▶ Predicts future frames in videos
- ▶ **Millions of bits per sample**



Applications: Image generation (VAE)



Applications: Image generation (DCGAN)



Radford A., Metz L., Chintala S. *Unsupervised representation learning with deep convolutional generative adversarial networks*, 2015

Applications: SuperResolution (SRGAN)



Ledig C. et al. Photo-realistic single image super-resolution using a generative adversarial network, 2016

Applications: Domain translation (CycleGAN)



Applications: Face generation (StyleGAN)



Karras T., Laine S., Aila T. A style-based generator architecture for generative adversarial networks, 2018

Applications: Face generation (VQ-VAE-2)



Razavi A., Oord A., Vinyals O. Generating Diverse High-Fidelity Images with VQ-VAE-2, 2019

Applications

- ▶ Audio Generation (WaveNet, ...)
- ▶ Video Generation (DVD-GAN)
- ▶ NLP (Transformer, BERT, GPT-3, ...)
- ▶ Compression

Problem Statement

We are given samples $\{\mathbf{x}_i\}_{i=1}^n \in X$ from unknown distribution $\pi(\mathbf{x})$.

Goal

We would like to learn a distribution $\pi(\mathbf{x})$ for

- ▶ evaluating $\pi(\mathbf{x})$ for new samples;
- ▶ sampling from $\pi(\mathbf{x})$.

Challenge

Data is complex and high-dimensional (curse of dimensionality).

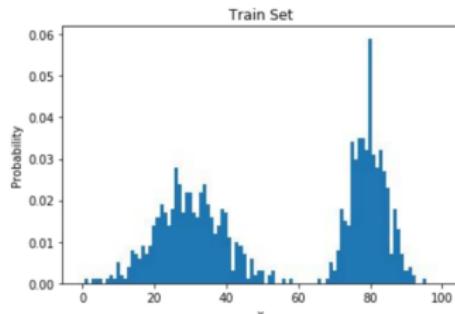
Histogram as a generative model

The histogram is totally defined by

$$p_k = p(x = k) = \frac{\sum_{i=1}^k [x_i = k]}{n}.$$

Problem: curse of dimensionality.

MNIST: 28x28 gray-scaled images
 $2^{28 \times 28} - 1$ parameters to specify $p(\mathbf{x})$



$$p(\mathbf{x}) = p(x_1) \cdot p(x_2|x_1) \cdot \dots \cdot p(x_m|x_{m-1}, \dots, x_1).$$

Question: How many parameters do we need in these cases?

$$p(\mathbf{x}) = p(x_1) \cdot p(x_2) \cdot \dots \cdot p(x_m).$$

$$p(\mathbf{x}) = p(x_1) \cdot p(x_2|x_1) \cdot \dots \cdot p(x_m|x_{m-1}).$$

Maximum likelihood

Fix probabilistic model $p(\mathbf{x}|\theta)$ – the set of parameterized distributions .

Instead of searching true $\pi(\mathbf{x})$ over all probability distributions, learn function approximation $p(\mathbf{x}|\theta) \approx \pi(\mathbf{x})$.

MLE problem

$$\boldsymbol{\theta}^* = \arg \max_{\theta} p(\mathbf{X}|\theta) = \arg \max_{\theta} \prod_{i=1}^n p(\mathbf{x}_i|\theta) = \arg \max_{\theta} \sum_{i=1}^n \log p(\mathbf{x}_i|\theta).$$

The problem is solved with SGD.

Requirements

- ▶ efficiently compute $\log p(\mathbf{x}|\theta)$;
- ▶ efficiently compute gradient of $\log p(\mathbf{x}|\theta)$.

Autoregressive model

MLE problem

$$\theta^* = \arg \max_{\theta} p(\mathbf{X}|\theta) = \arg \max_{\theta} \prod_{i=1}^n p(\mathbf{x}_i|\theta) = \arg \max_{\theta} \sum_{i=1}^n \log p(\mathbf{x}_i|\theta).$$

Challenge

$p(\mathbf{x}|\theta)$ could be intractable.

Likelihood as product of conditionals

Let $\mathbf{x} = (x_1, \dots, x_m)$, $\mathbf{x}_{1:i} = (x_1, \dots, x_i)$. Then

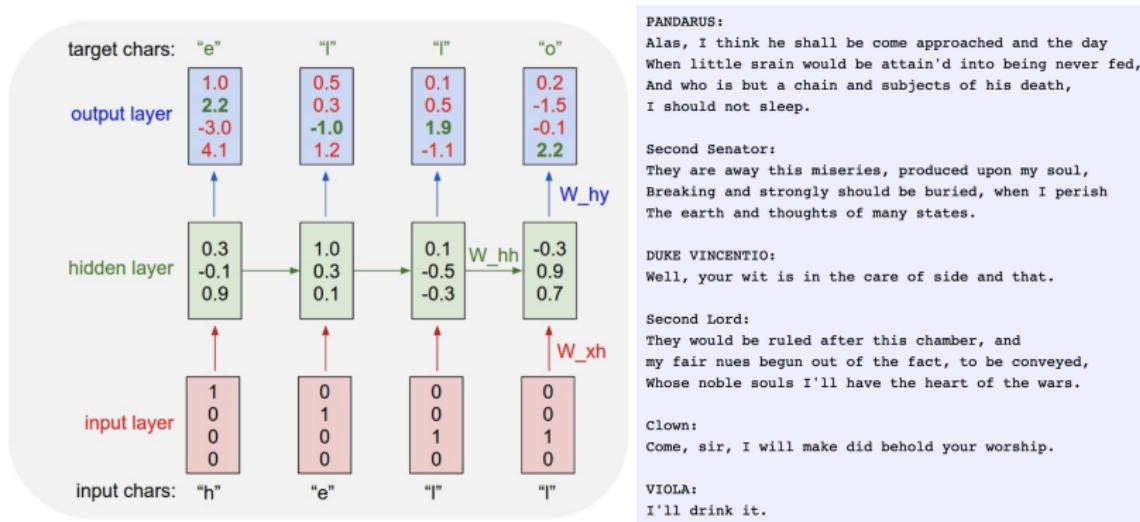
$$p(\mathbf{x}|\theta) = \prod_{i=1}^m p(x_i|\mathbf{x}_{1:i-1}, \theta); \quad \log p(\mathbf{x}|\theta) = \sum_{i=1}^m \log p(x_i|\mathbf{x}_{1:i-1}, \theta).$$

Autoregressive models

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{i=1}^m \log p(x_i|\mathbf{x}_{1:i-1}, \boldsymbol{\theta})$$

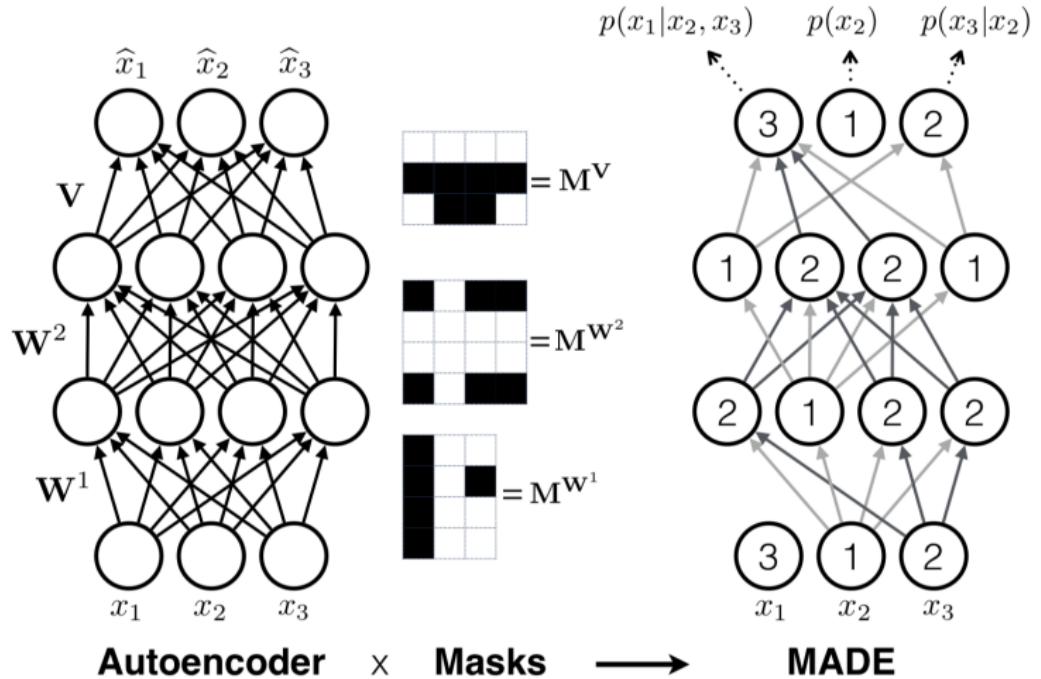
- ▶ Each conditional could be modelled by neural network.
- ▶ To extend to high dimensions share parameters across conditionals.
- ▶ Sampling is sequential.

Char RNN



Drawback

Sequential computation of all conditionals $p(x_i | \mathbf{x}_{1:i-1}, \theta)$.



WaveNet

Goal

Efficient generation of raw audio waveforms with natural sounds.

Solution

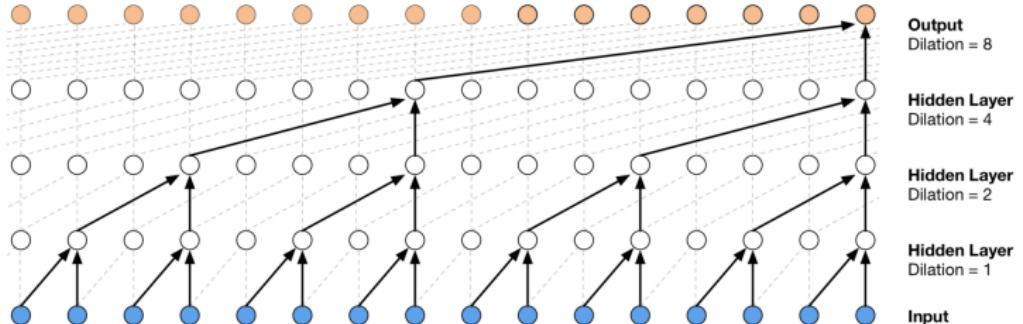
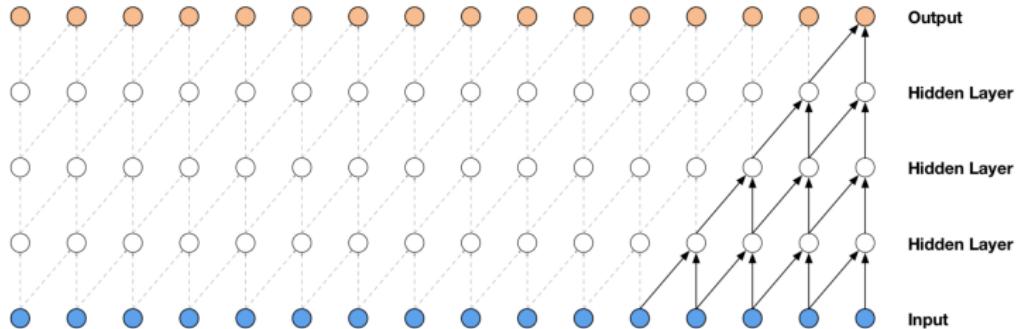
Autoregressive model

$$p(\mathbf{x}|\theta) = \prod_{t=1}^T p(x_t|\mathbf{x}_{1:t-1}, \theta).$$

The model uses causal dilated convolutions.



WaveNet (2016)



PixelCNN

Goal

Model a distribution of natural images.

Solution

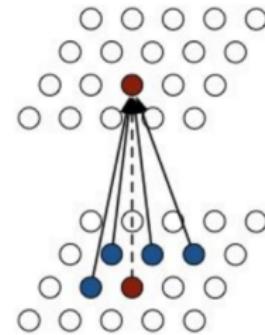
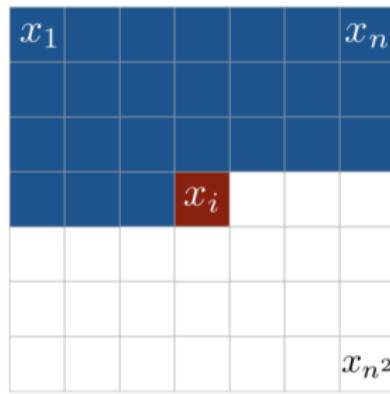
Autoregressive model

$$p(\mathbf{x}|\theta) = \prod_{i=1}^{n^2} p(x_i|\mathbf{x}_{1:i-1}, \theta).$$

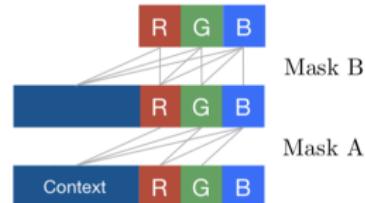
- ▶ masked convolutions;
- ▶ dependencies over RGB channels.

PixelCNN (2016)

1	1	1
1	0	0
0	0	0



PixelCNN



Summary

- ▶ Sampling from autoregressive models is trivial, but sequential
 - ▶ sample $x_0 \sim p(x_0)$;
 - ▶ sample $x_1 \sim p(x_1|x_0)$;
 - ▶
- ▶ Estimating probability:

$$p(\mathbf{x}) = \prod_{i=1}^m p(x_i|\mathbf{x}_{1:i-1}).$$

- ▶ Work on both continuous and discrete data.
- ▶ There is no natural way to do unsupervised learning.

Summary