

Deep Generative Models

Lecture 2

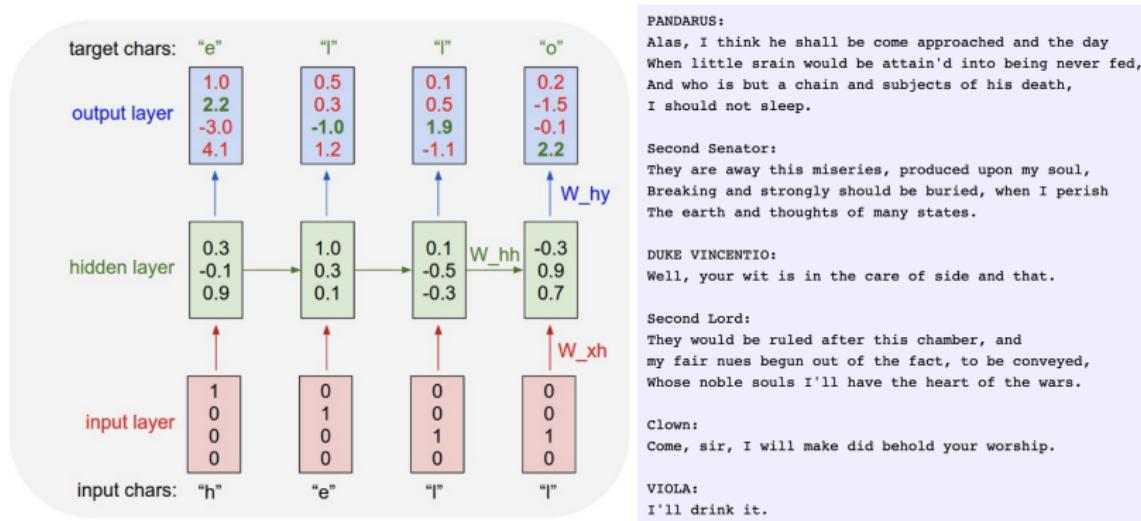
Roman Isachenko

Moscow Institute of Physics and Technology

Autumn, 2021

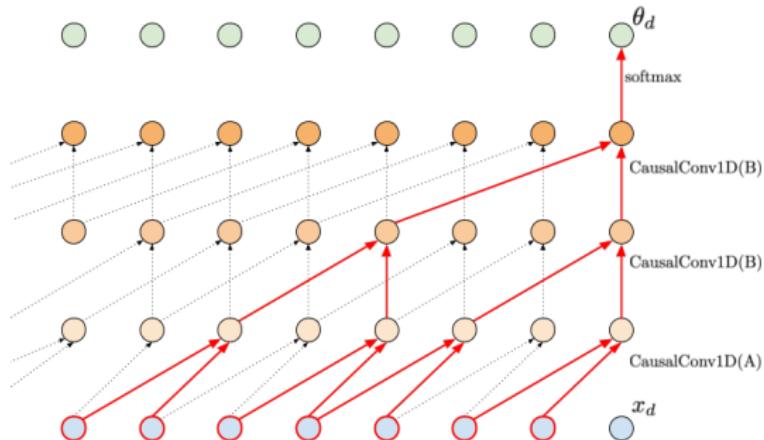
Char RNN

Model tries to predict the next token (single letter) from previous context.



Autoregressive models

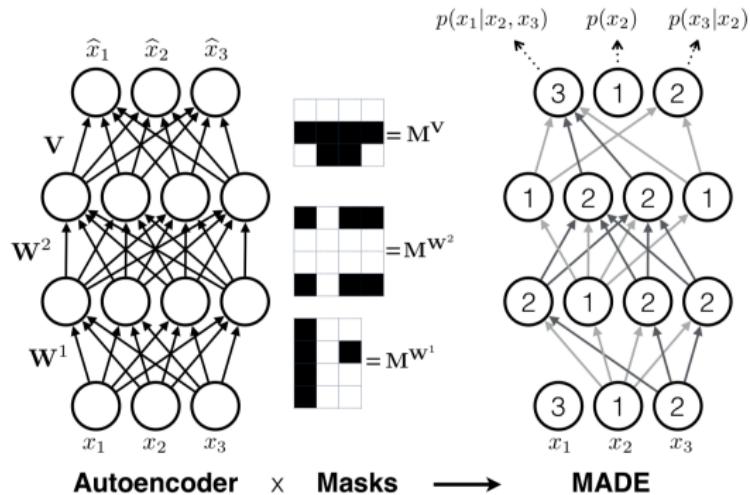
- ▶ Convolutions could be used for autoregressive models, but they have to be **causal**.
- ▶ Try to find and understand the difference between Conv A/B.



- ▶ Could learn long-range dependencies.
- ▶ Do not suffer from gradient issues.
- ▶ Easy to estimate probability for given input, but hard generation of new samples (the sequential process).

MADE

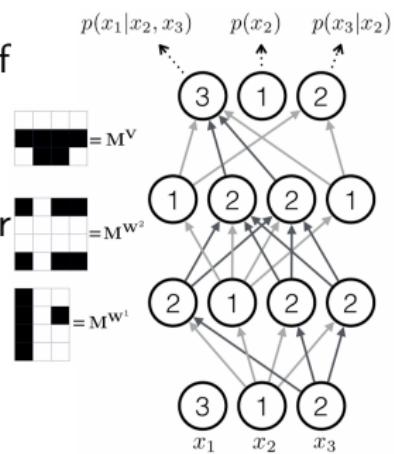
- ▶ Vanila autoencoder is not a generative model. Why?
- ▶ Let mask the weight matrices to make the model generative:
 $\mathbf{W}_M = \mathbf{W} \cdot \mathbf{M}$.



- ▶ The question is how to create matrices \mathbf{M} which produce the autoregressive property?

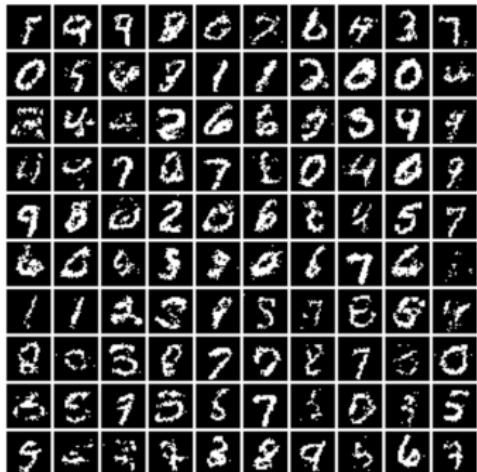
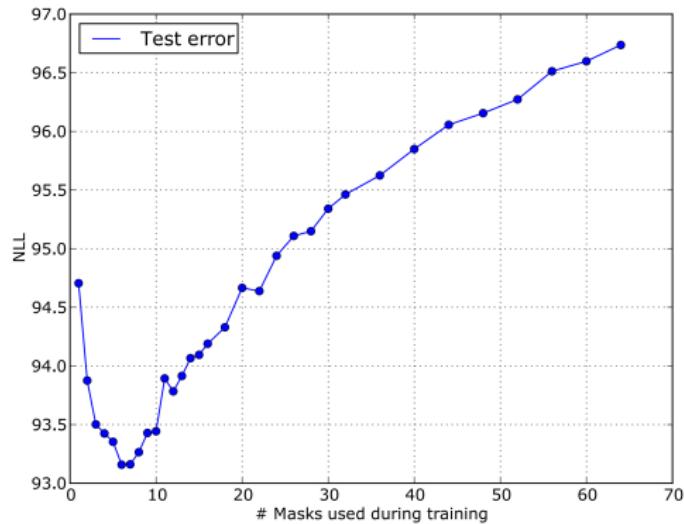
Masks generation

- ▶ Define the ordering of input elements from 1 to m .
- ▶ Assign the random number k from 1 to $m - 1$ to each hidden unit. The number gives the maximum number of input units to which the unit can be connected.
- ▶ Connect each hidden unit with number k with the previous layer units which has the number is **less or equal** than k .
- ▶ Connect each output unit with number k with the previous layer units which has the number is **less** than k .



Possible variations

- ▶ Order agnostic training (missing values in partially observed input vectors can be imputed efficiently);
- ▶ Connectivity-agnostic training (cheap ensembling).



WaveNet

Goal

Efficient generation of raw audio waveforms with natural sounds.



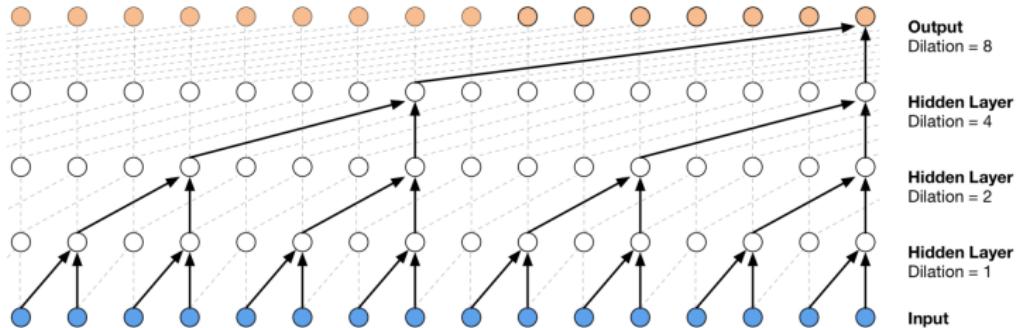
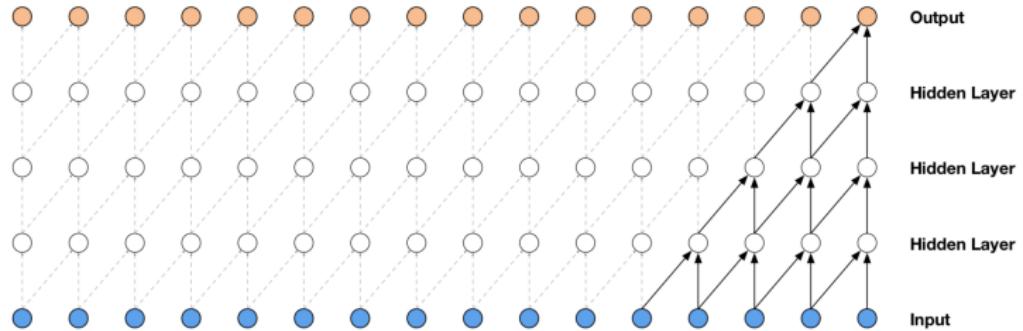
Solution

Autoregressive model

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

- ▶ Each conditional $p(x_t|\mathbf{x}_{1:t-1}, \theta)$ models the distribution for the timestamp t .
- ▶ The model uses **causal** dilated convolutions.

WaveNet



PixelCNN

Goal

Model a distribution of natural images.

Solution

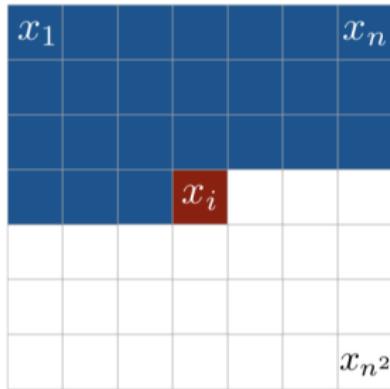
Autoregressive model on 2D pixels

$$p(\mathbf{x}|\theta) = \prod_{i=1}^{\text{width} \times \text{height}} p(x_i | \mathbf{x}_{1:i-1}, \theta).$$

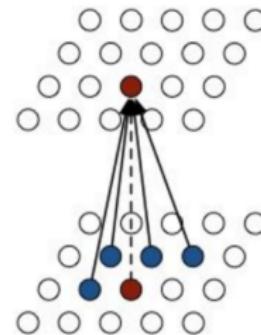
- ▶ We need to introduce the ordering of image pixels.
- ▶ The convolution should be **masked** to make them causal.
- ▶ The image has RGB channels, these dependencies could be addressed.

PixelCNN

Raster ordering



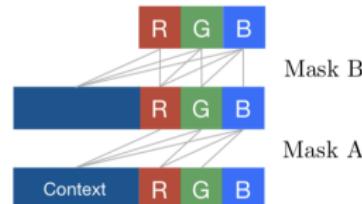
Dependencies between pixels



PixelCNN

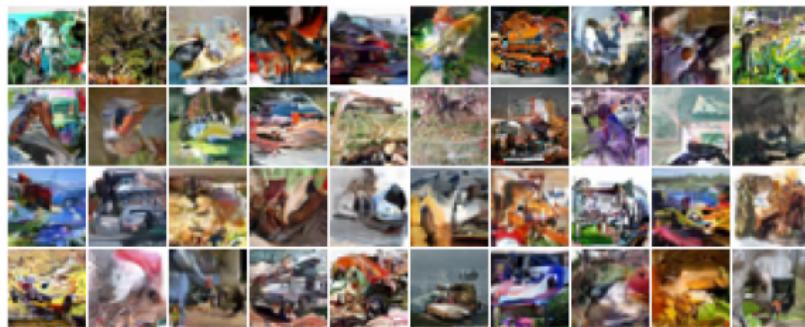
Masked convolution kernel

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 0 | 0 |



PixelCNN

CIFAR-10 generated samples

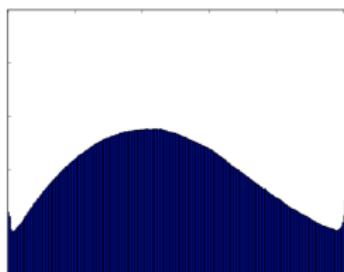


CIFAR-10 performance

| Model | NLL Test (Train) |
|------------------------|--------------------|
| Uniform Distribution: | 8.00 |
| Multivariate Gaussian: | 4.70 |
| NICE [1]: | 4.48 |
| Deep Diffusion [2]: | 4.20 |
| Deep GMMs [3]: | 4.00 |
| RIDE [4]: | 3.47 |
| PixelCNN: | 3.14 (3.08) |
| Row LSTM: | 3.07 (3.00) |
| Diagonal BiLSTM: | 3.00 (2.93) |

PixelCNN++

CIFAR-10 pixel values distribution



- ▶ Standard PixelCNN outputs softmax probabilities for values $\{0, 255\}$ (256 outputs feature maps).
- ▶ Categorical distribution do not know anything about numerical relationships (220 is close to 221 and far from 15).
- ▶ If pixel value is not presented in the training dataset , it won't be predicted.
- ▶ (Look at the edges of the distributions: they have higher probability mass).

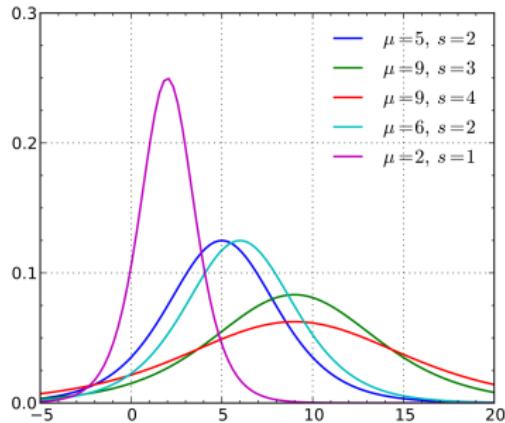
Salimans T. et al. PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications, 2017

PixelCNN++

Mixture of logistic distributions

$$p(\nu|\mu, s) = \frac{\exp^{-(\nu-\mu)/s}}{s(1 + \exp^{-(\nu-\mu)/s})^2};$$

$$p(\nu|\boldsymbol{\mu}, \mathbf{s}, \boldsymbol{\pi}) = \sum_{i=1}^K \pi_k p(\nu|\mu_k, s_k);$$



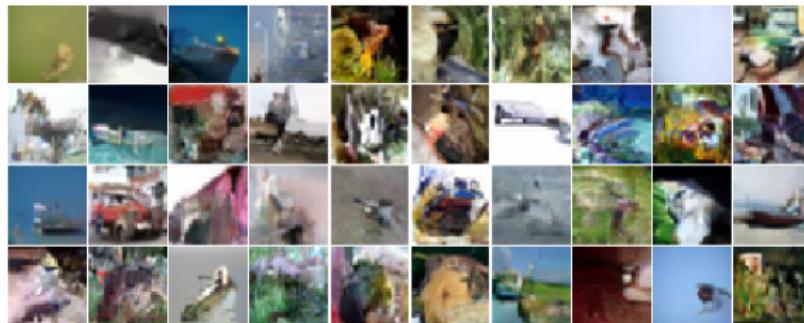
To adopt probability calculation to discrete values:

$$P(x|\boldsymbol{\mu}, \mathbf{s}, \boldsymbol{\pi}) = P(x + 0.5|\boldsymbol{\mu}, \mathbf{s}, \boldsymbol{\pi}) - P(x - 0.5|\boldsymbol{\mu}, \mathbf{s}, \boldsymbol{\pi})$$

For the edge case of 0, replace $x - 0.5$ by $-\infty$, and for 255 replace $x + 0.5$ by $+\infty$.

PixelCNN++

CIFAR-10 generated samples



CIFAR-10 performance

| Model | Bits per sub-pixel |
|--|--------------------|
| Deep Diffusion (Sohl-Dickstein et al., 2015) | 5.40 |
| NICE (Dinh et al., 2014) | 4.48 |
| DRAW (Gregor et al., 2015) | 4.13 |
| Deep GMMs (van den Oord & Dambre, 2015) | 4.00 |
| Conv DRAW (Gregor et al., 2016) | 3.58 |
| Real NVP (Dinh et al., 2016) | 3.49 |
| PixelCNN (van den Oord et al., 2016b) | 3.14 |
| VAE with IAF (Kingma et al., 2016) | 3.11 |
| Gated PixelCNN (van den Oord et al., 2016c) | 3.03 |
| PixelRNN (van den Oord et al., 2016b) | 3.00 |
| PixelCNN++ | 2.92 |

Salimans T. et al. *PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications*, 2017

Recap of previous lecture

We are given i.i.d. samples $\{\mathbf{x}_i\}_{i=1}^n \in X$ (e.g. $X = \mathbb{R}^m$) 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 (how likely to get object \mathbf{x} ?);
- ▶ sampling from $\pi(\mathbf{x})$ (to get new objects $\mathbf{x} \sim \pi(\mathbf{x})$).

Challenge

Data is complex and high-dimensional.

MLE problem

Fix probabilistic model $p(\mathbf{x}|\theta)$ – a set of parameterized distributions, such that $p(\mathbf{x}|\theta) \approx \pi(\mathbf{x})$.

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

Recap of previous lecture

Likelihood as product of conditionals

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

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

MLE problem for autoregressive model

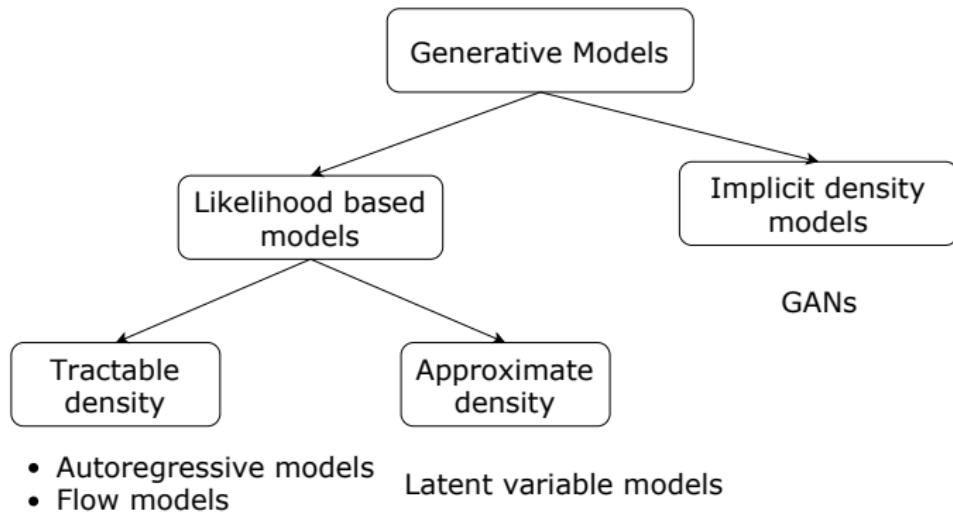
$$\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta}} p(\mathbf{X}|\boldsymbol{\theta}) = \arg \max_{\boldsymbol{\theta}} \sum_{i=1}^n \sum_{j=1}^m \log p(x_{ij}|\mathbf{x}_{i,1:j-1}\boldsymbol{\theta}).$$

Sampling

$$\hat{x}_1 \sim p(x_1|\boldsymbol{\theta}), \quad \hat{x}_2 \sim p(x_2|\hat{x}_1, \boldsymbol{\theta}), \dots, \quad \hat{x}_m \sim p(x_m|\hat{\mathbf{x}}_{1:m-1}, \boldsymbol{\theta})$$

New generated object is $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m)$.

Generative models zoo



Bayesian framework

Bayes theorem

$$p(\mathbf{t}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{t})p(\mathbf{t})}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\mathbf{t})p(\mathbf{t})}{\int p(\mathbf{x}|\mathbf{t})p(\mathbf{t})d\mathbf{t}}$$

- ▶ \mathbf{x} – observed variables, \mathbf{t} – unobserved variables (latent variables/parameters);
- ▶ $p(\mathbf{x}|\mathbf{t})$ – likelihood;
- ▶ $p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{t})p(\mathbf{t})d\mathbf{t}$ – evidence;
- ▶ $p(\mathbf{t})$ – prior distribution, $p(\mathbf{t}|\mathbf{x})$ – posterior distribution.

Meaning

We have unobserved variables \mathbf{t} and some prior knowledge about them $p(\mathbf{t})$. Then, the data \mathbf{x} has been observed. Posterior distribution $p(\mathbf{t}|\mathbf{x})$ summarizes the knoweldge after the obbservations.

Bayesian framework

Let consider the case, where the unobserved variables \mathbf{t} is our model parameters θ .

- ▶ $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ – observed samples;
- ▶ $p(\theta)$ – prior parameters distribution (we treat model parameters θ as random variables).

Posterior distribution

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{\int p(\mathbf{X}|\theta)p(\theta)d\theta}$$

Bayesian inference

$$p(\mathbf{x}|\mathbf{X}) = \int p(\mathbf{x}|\theta)p(\theta|\mathbf{X})d\theta$$

Note the difference from

$$p(\mathbf{x}) = \int p(\mathbf{x}|\theta)p(\theta)d\theta.$$

Bayesian framework

Posterior distribution

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})} = \frac{p(\mathbf{X}|\theta)p(\theta)}{\int p(\mathbf{X}|\theta)p(\theta)d\theta}$$

Bayesian inference

$$p(\mathbf{x}|\mathbf{X}) = \int p(\mathbf{x}|\theta)p(\theta|\mathbf{X})d\theta$$

If evidence $p(\mathbf{X})$ is intractable (due to multidimensional integration), we can't get posterior distribution and perform the precise inference.

Maximum a posteriori (MAP) estimation

$$\theta^* = \arg \max_{\theta} p(\theta|\mathbf{X}) = \arg \max_{\theta} (\log p(\mathbf{X}|\theta) + \log p(\theta))$$

Bayesian framework

MAP estimation

$$\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} | \mathbf{X}) = \arg \max_{\boldsymbol{\theta}} (\log p(\mathbf{X} | \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}))$$

Estimated $\boldsymbol{\theta}^*$ is a deterministic variable, but we could treat it as a random variable with density $p(\boldsymbol{\theta} | \mathbf{X}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$.

Dirac delta function

$$\delta(x) = \begin{cases} +\infty, & x = 0; \\ 0, & x \neq 0; \end{cases} \quad \int f(x)\delta(x - y)dx = f(y).$$

MAP inference

$$p(\mathbf{x} | \mathbf{X}) = \int p(\mathbf{x} | \boldsymbol{\theta}) p(\boldsymbol{\theta} | \mathbf{X}) d\boldsymbol{\theta} \approx p(\mathbf{x} | \boldsymbol{\theta}^*).$$

Latent variable models (LVM)

MLE problem

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

Challenge

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

Extend probabilistic model

Introduce latent variable \mathbf{z} for each sample \mathbf{x}

$$p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z}); \quad \log p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \log p(\mathbf{z}).$$

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})d\mathbf{z} = \int p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z})d\mathbf{z}.$$

Motivation

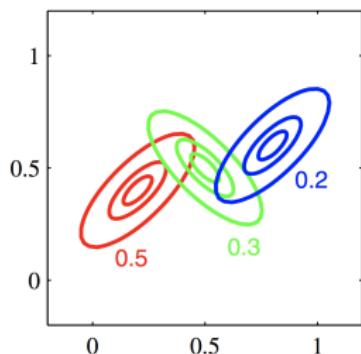
The distributions $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ and $p(\mathbf{z})$ could be quite simple.

Latent variable models (LVM)

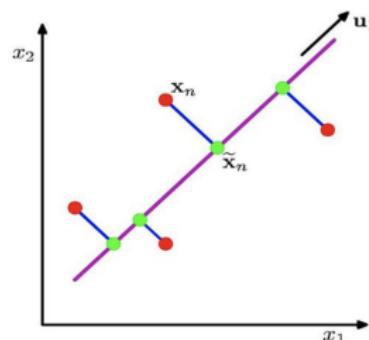
$$\log p(\mathbf{x}|\theta) = \log \int p(\mathbf{x}|\mathbf{z}, \theta)p(\mathbf{z})d\mathbf{z} \rightarrow \max_{\theta}$$

Examples

Mixture of gaussians



PCA model

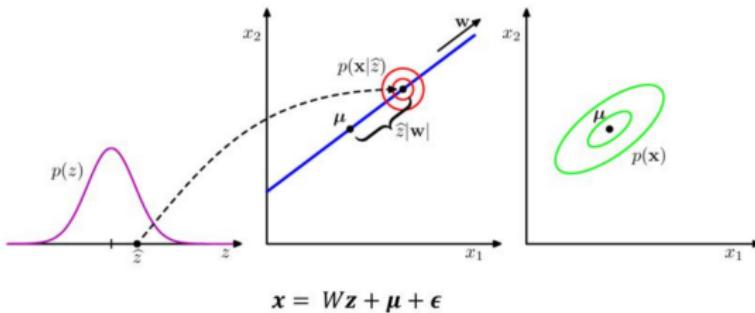


- ▶ $p(\mathbf{x}|\mathbf{z}, \theta) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{z}})$
- ▶ $p(\mathbf{z}) = \text{Categorical}(\mathbf{z}|\boldsymbol{\pi})$
- ▶ $p(\mathbf{x}|\mathbf{z}, \theta) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Sigma}_{\mathbf{z}})$
- ▶ $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \mathbf{I})$

Latent variable models (LVM)

$$\log p(\mathbf{x}|\theta) = \log \int p(\mathbf{x}|\mathbf{z}, \theta)p(\mathbf{z})d\mathbf{z} \rightarrow \max_{\theta}$$

PCA goal: Project original data \mathbf{X} onto a low dimensional latent space while maximizing the variance of the projected data.



- ▶ $p(\mathbf{x}|\mathbf{z}, \theta) = \mathcal{N}(\mathbf{x}|W\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Sigma}_z)$
- ▶ $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \mathbf{I})$

Incomplete likelihood

MLE

$$\begin{aligned}\theta^* &= \arg \max_{\theta} p(\mathbf{X}, \mathbf{Z} | \theta) = \arg \max_{\theta} \prod_{i=1}^n p(\mathbf{x}_i, \mathbf{z}_i | \theta) = \\ &= \arg \max_{\theta} \sum_{i=1}^n \log p(\mathbf{x}_i, \mathbf{z}_i | \theta).\end{aligned}$$

Since \mathbf{Z} is unknown, maximize **incomplete likelihood**.

MILE problem

$$\begin{aligned}\theta^* &= \arg \max_{\theta} \log p(\mathbf{X} | \theta) = \arg \max_{\theta} \sum_{i=1}^n \log p(\mathbf{x}_i | \theta) = \\ &= \arg \max_{\theta} \sum_{i=1}^n \log \int p(\mathbf{x}_i, \mathbf{z}_i | \theta) d\mathbf{z}_i = \\ &= \arg \max_{\theta} \log \sum_{i=1}^n \int p(\mathbf{x}_i | \mathbf{z}_i, \theta) p(\mathbf{z}_i) d\mathbf{z}_i.\end{aligned}$$

Variational lower bound

$$\begin{aligned}\log p(\mathbf{x}|\boldsymbol{\theta}) &= \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})} = \\&= \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})} d\mathbf{z} = \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})q(\mathbf{z})} d\mathbf{z} = \\&= \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} + \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})} d\mathbf{z} = \\&= \mathcal{L}(q, \boldsymbol{\theta}) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \geq \mathcal{L}(q, \boldsymbol{\theta}).\end{aligned}$$

Kullback-Leibler divergence

- ▶ $KL(q||p) = \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z})} d\mathbf{z};$
- ▶ $KL(q||p) \geq 0;$
- ▶ $KL(q||p) = 0 \Leftrightarrow q \equiv p.$

Variational lower bound

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \geq \mathcal{L}(q, \boldsymbol{\theta}).$$

Evidence Lower Bound (ELBO)

$$\begin{aligned}\mathcal{L}(q, \boldsymbol{\theta}) &= \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} = \\ &= \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) d\mathbf{z} + \int q(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z} \\ &= \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z}))\end{aligned}$$

Instead of maximizing incomplete likelihood, maximize ELBO
(equivalently minimize KL)

$$\max_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta}) \quad \rightarrow \quad \max_{q, \boldsymbol{\theta}} \mathcal{L}(q, \boldsymbol{\theta}) \equiv \min_{q, \boldsymbol{\theta}} KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

EM-algorithm

$$\mathcal{L}(q, \theta) = \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} + \int q(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z}.$$

Block-coordinate optimization

- ▶ Initialize θ^* ;
- ▶ E-step

$$q(\mathbf{z}) = \arg \max_q \mathcal{L}(q, \theta^*) = \arg \min_q KL(q||p) = p(\mathbf{z}|\mathbf{x}, \theta^*);$$

- ▶ M-step

$$\theta^* = \arg \max_{\theta} \mathcal{L}(q, \theta);$$

- ▶ Repeat E-step and M-step until convergence.

Ugly pic

Amortized variational inference

E-step

$$q(\mathbf{z}) = \arg \max_q \mathcal{L}(q, \boldsymbol{\theta}^*) = \arg \min_q KL(q||p) = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*).$$

- ▶ $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*)$ could be **intractable**;
- ▶ $q(\mathbf{z})$ is different for each object \mathbf{x} .

Idea

Restrict a family of all possible distributions $q(\mathbf{z})$ to a particular parametric class $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$ conditioned on samples \mathbf{x} with parameters $\boldsymbol{\phi}$.

Variational Bayes

- ▶ E-step

$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} + \eta \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}_{k-1})|_{\boldsymbol{\phi}=\boldsymbol{\phi}_{k-1}}$$

- ▶ M-step

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\phi}_k, \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}}$$

Variational EM-algorithm

ELBO

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \geq \mathcal{L}(q, \boldsymbol{\theta}).$$

- ▶ E-step

$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} + \eta \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}_{k-1})|_{\boldsymbol{\phi}=\boldsymbol{\phi}_{k-1}},$$

where $\boldsymbol{\phi}$ – parameters of variational distribution $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$.

- ▶ M-step

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\phi}_k, \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}},$$

where $\boldsymbol{\theta}$ – parameters of the generation function $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.

Now all we have to do is to obtain two gradients $\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$,
 $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$.

Difficulty: number of samples n .

Summary

- ▶ Bayesian inference is a generalization of most common machine learning tasks. It allows to construct MLE, MAP and bayesian inference, to compare models complexity and many-many more cool stuff.
- ▶ LVM introduce latent representation of observed samples to make model more interpretable.
- ▶ LVM maximizes variational evidence lower bound to find MLE of model parameters.
- ▶ ELBO maximization is performed by the general variational EM algorithm.