Deep Generative Models

Roman Isachenko

Moscow Institute of Physics and Technology

2019

Likelihood-based models so far...

Autoregressive models

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

- tractable likelihood,
- no inferred latent factors.

Latent variable models

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

- latent feature representation,
- intractable likelihood.

How to build model with latent variables and tractable likelihood?

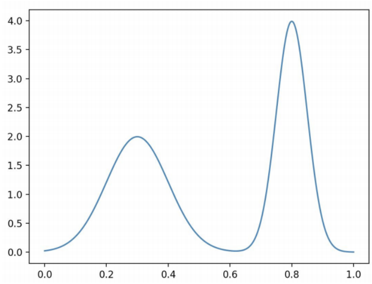
Flows intuition

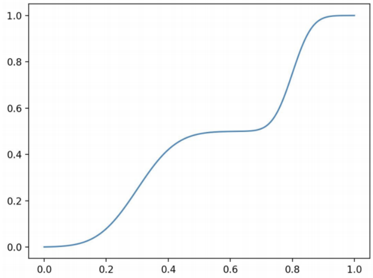
Let X be a random variable with density $p_X(x)$. Then

$$Z = F(X) = \int_{-\infty}^{x} p(t)dt \sim U[0,1].$$

Hence

$$Z \sim U[0,1]; \quad X = F^{-1}(Z) \quad X \sim p(x).$$





https://sites.google.com/view/berkeley-cs294-158-sp19/home

Change of variables

Theorem

Let

- $\mathbf{x}$ is a random variable,
- $f : \mathbb{R}^m \to \mathbb{R}^m$ is a differentiable, invertible function,
- $\mathbf{z} = f(\mathbf{x})$, $\mathbf{x} = f^{-1}(\mathbf{z}) = g(\mathbf{z})$.

Then

$$p(\mathbf{x}) = p(\mathbf{z}) \left| \det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right| = p(f(\mathbf{x})) \left| \det \left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right) \right| .$$

Note

- $\mathbf{x}$ and $\mathbf{z}$ have the same dimensionality;
- $\left| \det \left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right) \right| = \left| \det \left(\frac{\partial g^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right) \right| = \left| \det \left(\frac{\partial g(\mathbf{z})}{\partial \mathbf{z}} \right) \right|^{-1}.$

Fitting flows

MLE problem

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

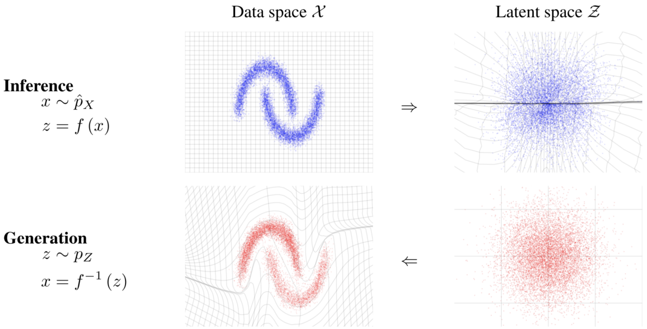
Challenge

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

Fitting flow to solve MLE

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(f(\mathbf{x}, \boldsymbol{\theta})) \left| \det \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}} \right) \right|$$

Flows



- Likelihood is given by $\mathbf{z} = f(\mathbf{x}, \boldsymbol{\theta})$ and change of variables.
- Sampling of $\mathbf{x}$ is performed by sampling from a base distribution $p(\mathbf{z})$ and applying $\mathbf{x} = f^{-1}(\mathbf{z}, \boldsymbol{\theta}) = g(\mathbf{z}, \boldsymbol{\theta})$.
- Latent representation is given by $\mathbf{z} = f(\mathbf{x}, \boldsymbol{\theta})$.

https://arxiv.org/pdf/1605.08803.pdf

Flows

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f(\mathbf{x}, \boldsymbol{\theta})) + \log \left| \det \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}} \right) \right|$$

Definition

Normalizing flow is a *differentiable, invertible* mapping from data $\mathbf{x}$ to the noise $\mathbf{z}$.

- Normalizing - convert data distribution to *noise*.
- Flow - sequence of such mapping is also a flow

$$\mathbf{z} = f_K \circ \cdots \circ f_1(\mathbf{x}); \quad \mathbf{x} = f_1^{-1} \circ \cdots \circ f_K^{-1}(\mathbf{z}) = g_1 \circ \cdots \circ g_K(\mathbf{z})$$

$$p(\mathbf{x}) = p(f_K \circ \cdots \circ f_1(\mathbf{x})) \left| \det \left(\frac{\partial f_K \circ \cdots \circ f_1(\mathbf{x})}{\partial \mathbf{x}} \right) \right| =$$

$$= p(f_K \circ \cdots \circ f_1(\mathbf{x})) \prod_{k=1}^{K} \left| \det \left(\frac{\partial \mathbf{f}_k}{\partial \mathbf{f}_{k-1}} \right) \right| .$$

Flows

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f(\mathbf{x},\boldsymbol{\theta})) + \log \left|\det\left(\frac{\partial f(\mathbf{x},\boldsymbol{\theta})}{\partial \mathbf{x}}\right)\right|$$

What we want

- Efficient computation of Jacobian $\frac{\partial f(\mathbf{x},\boldsymbol{\theta})}{\partial \mathbf{x}}$;
- Efficient sampling from the base distribution $p(\mathbf{z})$;
- Easy to invert $f(\mathbf{x},\boldsymbol{\theta})$.

Planar Flows, 2015

$$g(\mathbf{z}, \boldsymbol{\theta}) = \mathbf{z} + \mathbf{u}h(\mathbf{w}^T\mathbf{z} + b).$$

- $\boldsymbol{\theta} = \{\mathbf{u}, \mathbf{w}, b\}$;
- h is a smooth element-wise non-linearity.

$$\begin{aligned}\left|\det\left(\frac{\partial g(\mathbf{z}, \boldsymbol{\theta})}{\partial \mathbf{z}}\right)\right| &= \left|\det\left(\mathbf{I} + h'(\mathbf{w}^T\mathbf{z} + b)\mathbf{w}\mathbf{u}^T\right)\right| \\ &= \left|1 + h'(\mathbf{w}^T\mathbf{z} + b)\mathbf{w}^T\mathbf{u}\right|\end{aligned}$$

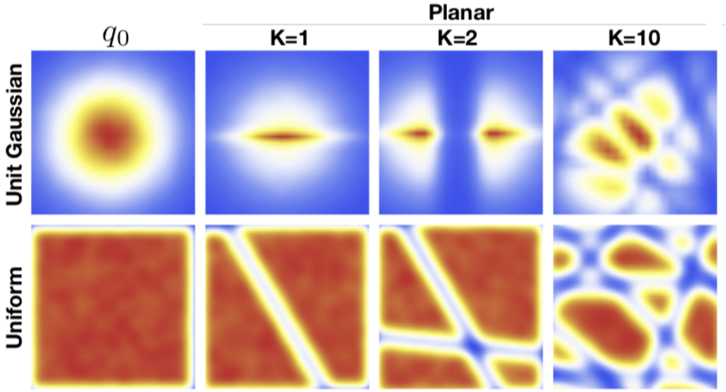
The transformation is invertible if (just one of example)

$$h = \tanh; \quad h'(\mathbf{w}^T\mathbf{z} + b)\mathbf{u}^T\mathbf{w} \geq -1.$$

https://arxiv.org/pdf/1505.05770.pdf

Planar Flows, 2015

$$\mathbf{z}_K = g_1 \circ \cdots \circ g_K(\mathbf{z}); \quad g_k = g(\mathbf{z}_k, \boldsymbol{\theta}_k).$$



https://arxiv.org/pdf/1505.05770.pdf

Jacobian structure

- What is the determinant of a diagonal matrix?

$$\mathbf{z} = f(\mathbf{x}, \boldsymbol{\theta}) = (f_1(x_1, \boldsymbol{\theta}), \dots, f_m(x_m, \boldsymbol{\theta})).$$

$$\log \left| \det \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}} \right) \right| = \log \left| \prod_{i=1}^{m} f_i'(x_i, \boldsymbol{\theta}) \right| = \sum_{i=1}^{m} \log \left| f_i'(x_i, \boldsymbol{\theta}). \right|$$

- What is the determinant of a triangular matrix?
  Let z_i depends only on $\mathbf{x}_{1:i}$ (or without loss of generality x_i depends on $\mathbf{z}_{1:i}$).
  What is the inverse of such transformations?

NICE, 2014

Coupling layer

$$\begin{cases} \mathbf{z}_{1:d} = \mathbf{x}_{1:d} \\ \mathbf{z}_{d:m} = \tau(\mathbf{x}_{d:m}, c(\mathbf{x}_{1:d})) \end{cases} \qquad \begin{cases} \mathbf{x}_{1:d} = \mathbf{z}_{1:d} \\ \mathbf{x}_{d:m} = \tau^{-1}(\mathbf{z}_{d:m}, c(\mathbf{z}_{1:d})) \end{cases}$$

- $c : \mathbb{R}^d \to \mathbb{R}^k$ – coupling function;
- $\tau : \mathbb{R}^{m-d} \times c(\mathbb{R}^d) \to \mathbb{R}^{m-d}$ – coupling law.
- $$\det\left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right) = \det\begin{pmatrix} \mathbf{I}_d & 0_{d\times m-d} \\ \frac{\partial \mathbf{z}_{d:m}}{\partial \mathbf{x}_{1:d}} & \frac{\partial \mathbf{z}_{d:m}}{\partial \mathbf{x}_{d:m}} \end{pmatrix} = \det\left(\frac{\partial \mathbf{z}_{d:m}}{\partial \mathbf{x}_{d:m}}\right)$$

https://arxiv.org/pdf/1410.8516.pdf

NICE, 2014

Coupling layer

$$\begin{cases} \mathbf{z}_{1:d} = \mathbf{x}_{1:d}; \\ \mathbf{z}_{d:m} = \tau(\mathbf{x}_{d:m}, c(\mathbf{x}_{1:d})); \end{cases} \quad \Rightarrow \quad \begin{cases} \mathbf{x}_{1:d} = \mathbf{z}_{1:d}; \\ \mathbf{x}_{d:m} = \tau^{-1}(\mathbf{z}_{d:m}, c(\mathbf{z}_{1:d})). \end{cases}$$

Coupling function $c(\cdot)$

Any complex function (without restrictions). For example, neural network.

Coupling law $\tau(\cdot, \cdot)$

- $\tau(x, c) = x + c$ – *additive*;
- $\tau(x, c) = x \odot c, \ c \neq 0$ – multiplicative;
- $\tau(x, c) = x \odot c_1 + c_2, \ c_1 \neq 0$ – affine.

To obtain more flexible class of dictributions, stack more coupling layers (with different ordering of components!).

https://arxiv.org/pdf/1410.8516.pdf

NICE, 2014

$$\det\left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right) = \det\begin{pmatrix} \mathbf{I}_d & 0_{d\times m-d} \\ \frac{\partial \mathbf{z}_{d:m}}{\partial \mathbf{x}_{1:d}} & \frac{\partial \mathbf{z}_{d:m}}{\partial \mathbf{x}_{d:m}} \end{pmatrix} = \det\left(\frac{\partial \mathbf{z}_{d:m}}{\partial \mathbf{x}_{d:m}}\right)$$

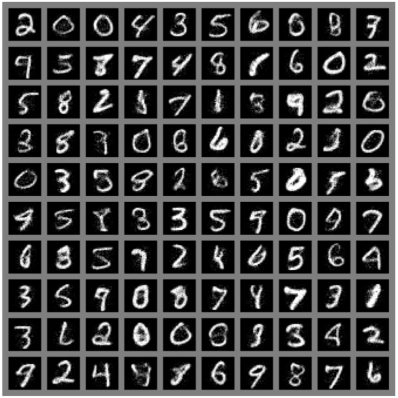
What is the Jacobian for the additive coupling law $\tau(x + c) = x + c$?
In this case the transformation is *volume preserving*.
The last layer is rescaling:

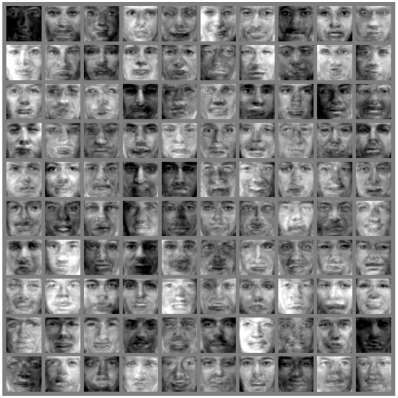
$$z_i = s_i x_i; \quad x_i = z_i/s_i.$$

What is the Jacobian of the last layer?

https://arxiv.org/pdf/1410.8516.pdf

NICE, 2014


(a) Model trained on MNIST


(b) Model trained on TFD

https://arxiv.org/pdf/1410.8516.pdf

RealNVP, 2016

Affine coupling law

$$\begin{cases} \mathbf{z}_{1:d} = \mathbf{x}_{1:d}; \\ \mathbf{z}_{d:m} = \mathbf{x}_{d:m} \odot \exp\left(c_1(\mathbf{x}_{1:d}, \boldsymbol{\theta})\right) + c_2(\mathbf{x}_{1:d}, \boldsymbol{\theta}). \end{cases}$$

$$\begin{cases} \mathbf{x}_{1:d} = \mathbf{z}_{1:d}; \\ \mathbf{x}_{d:m} = (\mathbf{z}_{d:m} - c_2(\mathbf{x}_{1:d}, \boldsymbol{\theta})) \odot \exp(-c_1(\mathbf{x}_{1:d}, \boldsymbol{\theta})). \end{cases}$$

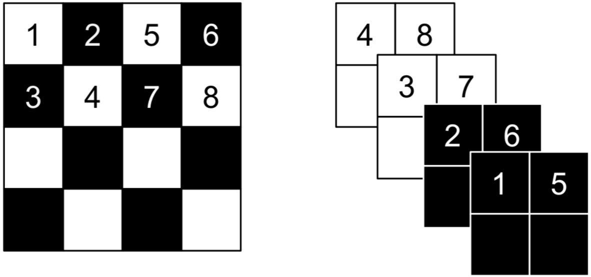
Jacobian

$$\det\left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right) = \det\begin{pmatrix} \mathbf{I}_d & 0_{d \times m-d} \\ \frac{\partial \mathbf{z}_{d:m}}{\partial \mathbf{x}_{1:d}} & \frac{\partial \mathbf{z}_{d:m}}{\partial \mathbf{x}_{d:m}} \end{pmatrix} = \prod_{i=1}^{m-d} \exp(c_1(\mathbf{x}_{1:d}, \boldsymbol{\theta})_i).$$

Non-Volume Preserving.

https://arxiv.org/pdf/1605.08803.pdf

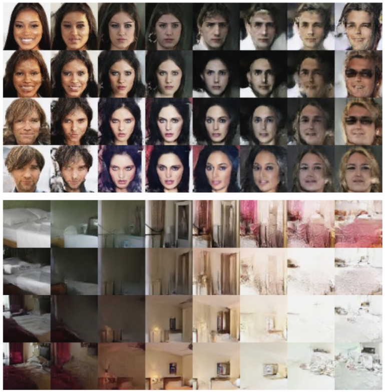
RealNVP, 2016

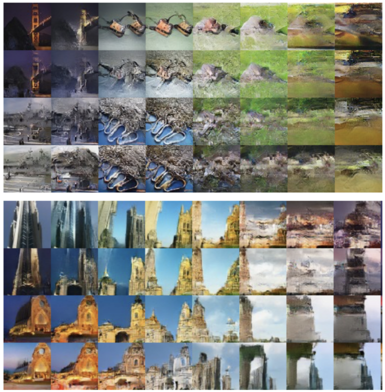


Masked convolutions are used to define ordering.

https://arxiv.org/pdf/1605.08803.pdf

RealNVP, 2016





https://arxiv.org/pdf/1605.08803.pdf

References

- Bishop, C. Pattern recognition and machine learning. 2006.
  Chapter 10.
- **NICE**: Non-linear Independent Components Estimation
  https://arxiv.org/abs/1410.8516
  Summary: Uses flows to model complex high-dimensional densities. Introduce the ways to compute determinant of Jacobian in a simple way. Triangular Jacobian, coupling layers, factorized distribution.
- Variational Inference with Normalizing Flows
  https://arxiv.org/abs/1505.05770
  Summary: Propose to use normalizing flows in variational inference. Discuss finite and infinitesimal flows. Uses invertible flows: planar, radial. Comparison with NICE.
- **RealNVP**: Density estimation using Real NVP
  https://arxiv.org/pdf/1605.08803.pdf
  Summary: Authors of NICE. The same idea and architecture, more practical. Lots of experiments and images. Coupling layers with checkerboard and channel-wise permutations.

Likelihood-based models

Exact likelihood evaluation

- Autoregressive models (PixelCNN, WaveNet);
- Flow models (NICE, RealNVP).

Approximate likelihood evaluation

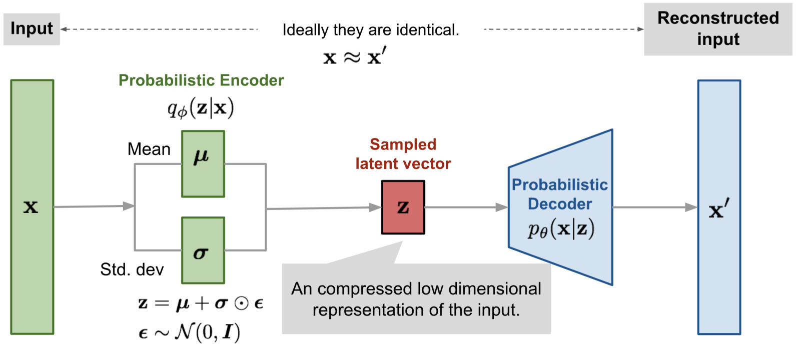
- Latent variable models (VAE).

What are the pros and cons of each of them?

How are they connected?

VAE recap

$$p(\mathbf{x}|\boldsymbol{\theta}) \geq \mathcal{L}(\phi, \boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\phi)} \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x}, \phi)} \to \max_{\phi,\boldsymbol{\theta}} .$$



Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
Probabilistic Encoder
$q_\phi(\mathbf{z}|\mathbf{x})$
$\mathbf{x}$
Mean
$\boldsymbol{\mu}$
$\boldsymbol{\sigma}$
Std. dev
$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
Sampled latent vector
$\mathbf{z}$
An compressed low dimensional representation of the input.
Probabilistic Decoder
$p_\theta(\mathbf{x}|\mathbf{z})$
$\mathbf{x}'$

https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

VAE limitations

- Poor variational posterior distribution (encoder)

$$q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\phi}(\mathbf{x}), \boldsymbol{\sigma}^2_{\phi}(\mathbf{x})).$$

- Poor prior distribution

$$p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I}).$$

- Poor probabilistic model (decoder)

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \boldsymbol{\sigma}^2_{\boldsymbol{\theta}}(\mathbf{z})).$$

- Loose lower bound

$$p(\mathbf{x}|\boldsymbol{\theta}) - \mathcal{L}(q, \boldsymbol{\theta}) = (?).$$

Variational posterior

We wish $KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) = 0$.
(In this case the lower bound is tight $p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta})$).

Normal variational distribution $q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}^2_\phi(\mathbf{x}))$ is poor (e.g. has only one mode).

Flows models transform simple base distribution to compex one using invertible transformation with simple Jacobian.

How to use flows in VAE?

Flows in VAE

Apply the sequence of transformations to the random variables

$$\mathbf{z}_0 \sim q_0(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}^2_\phi(\mathbf{x})).$$

Here, $q_0(\mathbf{z}|\mathbf{x}, \phi)$ plays the role of a base distribution.

$$\mathbf{z}_0 \xrightarrow{g_1} \mathbf{z}_1 \xrightarrow{g_2} \dots \xrightarrow{g_K} \mathbf{z}_K.$$

Each g_k is a flow transformation (e.g. planar, radial, coupling layer).

$$\log q_K(\mathbf{z}_K) = \log q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \log \left| \det \left(\frac{\partial g_k(\mathbf{z}_{k-1})}{\partial \mathbf{z}_{k-1}} \right) \right|.$$

https://arxiv.org/pdf/1505.05770.pdf

Flows in VAE

$$\log q_K(\mathbf{z}_K) = \log q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \log \left| \det \left(\frac{\partial g_k(\mathbf{z}_{k-1})}{\partial \mathbf{z}_{k-1}} \right) \right| .$$

Now the variational posterior is $q_K(\mathbf{z}_K|\mathbf{x}, \phi)$.

$$\begin{aligned}
\mathcal{L}(\phi, \boldsymbol{\theta}) &= \mathbb{E}_{q_K(\mathbf{z}_K|\mathbf{x},\phi)} \log \frac{p(\mathbf{x}, \mathbf{z}_K|\boldsymbol{\theta})}{q_K(\mathbf{z}_K|\mathbf{x}, \phi)} \\
&= \mathbb{E}_{q_K(\mathbf{z}_K|\mathbf{x},\phi)} \left[\log p(\mathbf{x}, \mathbf{z}_K|\boldsymbol{\theta}) - \log q_K(\mathbf{z}_K|\mathbf{x}, \phi) \right] \\
&= \mathbb{E}_{q_0(\mathbf{z}_0|\mathbf{x},\phi)} \left[\log p(\mathbf{x}, \mathbf{z}_K|\boldsymbol{\theta}) - \log q_K(\mathbf{z}_K|\mathbf{x}, \phi) \right] \\
&= \mathbb{E}_{q_0(\mathbf{z}_0|\mathbf{x},\phi)} \Bigg[\log p(\mathbf{x}, \mathbf{z}_K|\boldsymbol{\theta}) - \log q_0(\mathbf{z}_0|\mathbf{x}, \phi) - \\
&\quad - \sum_{k=1}^{K} \log \left| \det \left(\frac{\partial g_k(\mathbf{z}_{k-1})}{\partial \mathbf{z}_{k-1}} \right) \right| \Bigg] .
\end{aligned}$$

https://arxiv.org/pdf/1505.05770.pdf

MAF, 2017

Consider autoregressive model

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

with conditionals

$$p(x_i|\mathbf{x}_{1:i-1}, \boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{\mu}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1}), \boldsymbol{\sigma}^2_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1})\right).$$

Sampling

$$x_i = \boldsymbol{\sigma}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1}) \cdot z_i + \boldsymbol{\mu}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1}), \quad z_i \sim \mathcal{N}(0, 1).$$

Inverse transform

$$z_i = \left(x_i - \boldsymbol{\mu}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1})\right) \cdot \frac{1}{\boldsymbol{\sigma}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1})}.$$

https://arxiv.org/pdf/1705.07057.pdf

MAF, 2017

Sampling

$$\mathbf{x} = g(\mathbf{z}, \boldsymbol{\theta}) = \boldsymbol{\sigma}_{\boldsymbol{\theta}}(\mathbf{x}) \odot \mathbf{z} + \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}), \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I}).$$

Inverse transform

$$\mathbf{z} = f(\mathbf{x}, \boldsymbol{\theta}) = (\mathbf{x} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x})) \odot \frac{1}{\boldsymbol{\sigma}_{\boldsymbol{\theta}}(\mathbf{x})}.$$

What is the Jacobian of such flow?

- Sampling is slow (sequential).
- Likelihood evaluation is fast (e.g. MADE).

Suitable for density evaluation task.

https://arxiv.org/pdf/1705.07057.pdf

IAF, 2016

Inverse transform in MAF

$$
\begin{aligned}
z_i &= \left(x_i - \boldsymbol{\mu}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1})\right) \cdot \frac{1}{\boldsymbol{\sigma}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1})} \\
&= \frac{x_i}{\boldsymbol{\sigma}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1})} - \frac{\boldsymbol{\mu}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1})}{\boldsymbol{\sigma}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1})} \\
&= \hat{\boldsymbol{\sigma}}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1}) \cdot x_i + \hat{\boldsymbol{\mu}}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1}).
\end{aligned}
$$

How to make this transform to be efficient for sampling?

https://arxiv.org/pdf/1606.04934.pdf

IAF, 2016

Sampling and inverse transform in MAF

$$x_i = \boldsymbol{\sigma}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1}) \cdot z_i + \boldsymbol{\mu}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1}).$$

$$z_i = \left(x_i - \boldsymbol{\mu}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1})\right) \cdot \frac{1}{\boldsymbol{\sigma}_{i,\boldsymbol{\theta}}(\mathbf{x}_{1:i-1})}.$$

Sampling and inverse transform in IAF

$$x_i = \hat{\boldsymbol{\sigma}}_{i,\boldsymbol{\theta}}(\mathbf{z}_{1:i-1}) \cdot z_i + \hat{\boldsymbol{\mu}}_{i,\boldsymbol{\theta}}(\mathbf{z}_{1:i-1}).$$

$$z_i = \left(x_i - \hat{\boldsymbol{\mu}}_{i,\boldsymbol{\theta}}(\mathbf{z}_{1:i-1})\right) \cdot \frac{1}{\hat{\boldsymbol{\sigma}}_{i,\boldsymbol{\theta}}(\mathbf{z}_{1:i-1})}.$$

MAF and IAF are inverse to each other up to reparametrization

$$\hat{\boldsymbol{\sigma}}_i = \frac{1}{\boldsymbol{\sigma}_i}; \quad \hat{\boldsymbol{\mu}}_i = \frac{\boldsymbol{\mu}_i}{\boldsymbol{\sigma}_i}.$$

---

https://arxiv.org/pdf/1606.04934.pdf

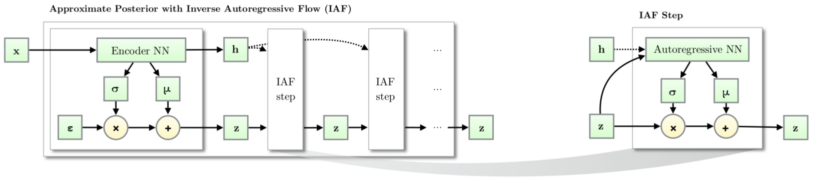
IAF, 2016

Sampling

$$x_i = \hat{\boldsymbol{\sigma}}_{i,\theta}(\mathbf{z}_{1:i-1}) \cdot z_i + \hat{\boldsymbol{\mu}}_{i,\theta}(\mathbf{z}_{1:i-1}).$$

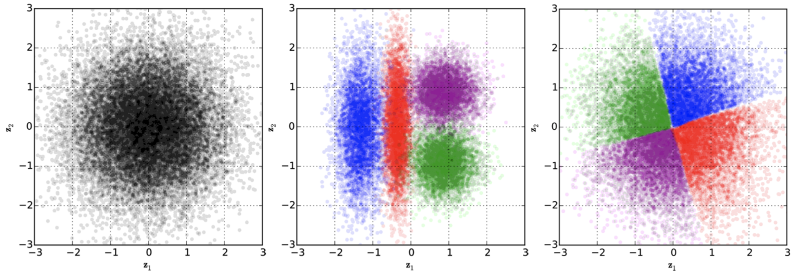
Approximate posterior

$$z_i = \left(x_i - \hat{\boldsymbol{\mu}}_{i,\theta}(\mathbf{z}_{1:i-1})\right) \cdot \frac{1}{\hat{\boldsymbol{\sigma}}_{i,\theta}(\mathbf{z}_{1:i-1})}.$$



https://arxiv.org/pdf/1606.04934.pdf

IAF, 2016



(a) Prior distribution (b) Posteriors in standard VAE (c) Posteriors in VAE with IAF

https://arxiv.org/pdf/1606.04934.pdf

MAF vs IAF

Theorem

Training a MAF with maximum likelihood corresponds to fitting an implicit IAF to the base density with stochastic variational inference:

$$\max_{\boldsymbol{\theta}} p(\mathbf{X}|\boldsymbol{\theta}) \quad \Leftrightarrow \quad \min_{\boldsymbol{\theta}} KL\left(p(\mathbf{z}|\boldsymbol{\theta})||\pi(\mathbf{z})\right)$$

(Here, $\pi(\mathbf{z})$ is a base distribution, $\pi(\mathbf{x})$ is a data distribution).

Proof

$$KL\left(p(\mathbf{z}|\boldsymbol{\theta})||p(\mathbf{z})\right) = \mathbb{E}_{p(\mathbf{z}|\boldsymbol{\theta})}\left[\log p(\mathbf{z}|\boldsymbol{\theta}) - \log \pi(\mathbf{z})\right] =$$

$$= \mathbb{E}_{p(\mathbf{z}|\boldsymbol{\theta})}\left[\log \pi(g(\mathbf{z})) + \log\left|\det\left(\frac{\partial g(\mathbf{z})}{\partial \mathbf{z}}\right)\right| - \log \pi(\mathbf{z})\right] =$$

$$= \mathbb{E}_{\pi(\mathbf{x})}\left[\log \pi(\mathbf{x}) - \log\left|\det\left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right)\right| - \log \pi(f(\mathbf{x}))\right].$$

https://arxiv.org/pdf/1705.07057.pdf

MAF vs IAF

Proof (continued)

$$KL\left(p(\mathbf{z}|\boldsymbol{\theta})||p(\mathbf{z})\right) = \mathbb{E}_{\pi(\mathbf{x})}\left[\log \pi(\mathbf{x}) - \log \left|\det\left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right)\right| - \log \pi(f(\mathbf{x}))\right] =$$
$$= \mathbb{E}_{\pi(\mathbf{x})}\left[\log \pi(\mathbf{x}) - \log p(\mathbf{x}|\boldsymbol{\theta})\right] = KL(\pi(\mathbf{x})||p(\mathbf{x}|\boldsymbol{\theta})).$$

$$\arg\min_{\boldsymbol{\theta}} KL(\pi(\mathbf{x})||p(\mathbf{x}|\boldsymbol{\theta})) = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\pi(\mathbf{x})}\left[\log \pi(\mathbf{x}) - \log p(\mathbf{x}|\boldsymbol{\theta})\right]$$
$$= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\pi(\mathbf{x})} \log p(\mathbf{x}|\boldsymbol{\theta})$$

Unbiased estimator is MLE:

$$\mathbb{E}_{\pi(\mathbf{x})} \log p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{i=1}^{n} p(\mathbf{x}_i|\boldsymbol{\theta}).$$

https://arxiv.org/pdf/1705.07057.pdf

MAF vs IAF vs RealNVP

RealNVP

$$\begin{aligned}\mathbf{x}_{1:d} &= \mathbf{z}_{1:d};\\ \mathbf{x}_{d:m} &= \mathbf{z}_{d:m} \odot \exp\left(c_1(\mathbf{z}_{1:d}, \boldsymbol{\theta})\right) + c_z(\mathbf{x}_{1:d}, \boldsymbol{\theta})\end{aligned}$$

MAF

$$\mathbf{x} = \boldsymbol{\sigma}_{\boldsymbol{\theta}}(\mathbf{x}) \odot \mathbf{z} + \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}).$$

IAF

$$\mathbf{x} = \hat{\boldsymbol{\sigma}}_{\boldsymbol{\theta}}(\mathbf{z}) \odot \mathbf{z} + \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}(\mathbf{z}).$$

How they are connected? Which flow is the most flexible?

https://arxiv.org/pdf/1705.07057.pdf

MAF/IAF pros and cons

MAF

- Sampling is slow.
- Likelihood evaluation is fast.

IAF

- Sampling is fast.
- Likelihood evaluation is slow.

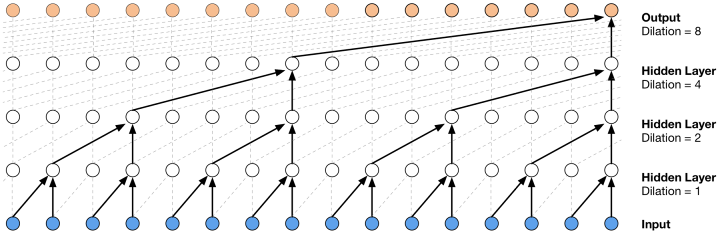
How to take the best of both worlds?

WaveNet (2016)

Autoregressive model for raw audio waveforms generation

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

The model uses causal dilated convolutions.



https://arxiv.org/pdf/1609.03499.pdf

Parallel WaveNet, 2017

Previous WaveNet model

- raw audio is high-dimensional (e.g. 16000 samples per second for 16kHz audio);
- WaveNet encodes 8-bit signal with 256-way categorical distribution.

Goal

- improved fidelity (24kHz instead of 16kHz) $\rightarrow$ increase dilated convolution filter size from 2 to 3;
- 16-bit signals $\rightarrow$ mixture of logistics instead of categorical distribution.

https://arxiv.org/pdf/1711.10433.pdf

Parallel WaveNet, 2017

Probability density distillation

1. Train usual WaveNet (MAF) via MLE (teacher network).
2. Train IAF WaveNet model (student network), which attempts to match the probability of its own samples under the distribution learned by the teacher.

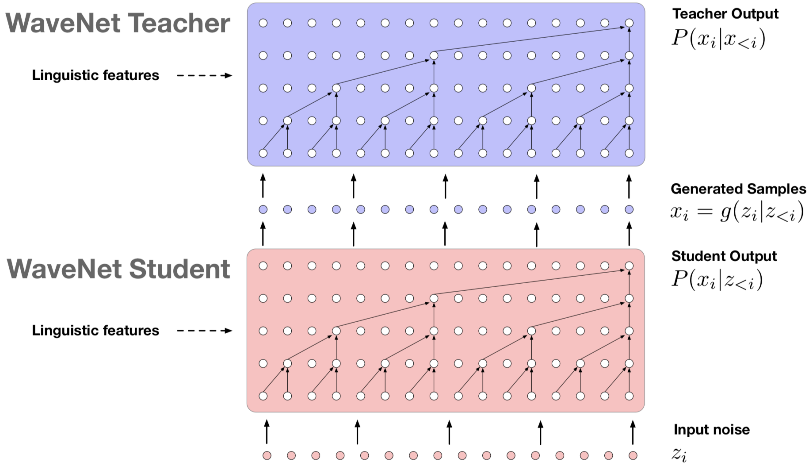
Student objective

$$KL(p_s||p_t) = H(p_s, p_t) - H(p_s).$$

More than 1000x speed-up relative to original WaveNet!

https://arxiv.org/pdf/1711.10433.pdf

Parallel WaveNet, 2017



https://arxiv.org/pdf/1711.10433.pdf

References

- **NICE**: Non-linear Independent Components Estimation
  https://arxiv.org/abs/1410.8516
  Summary: Uses flows to model complex high-dimensional densities. Introduce the ways to compute determinant of Jacobian in a simple way. Triangular Jacobian, coupling layers, factorized distribution.
- Variational Inference with Normalizing Flows
  https://arxiv.org/abs/1505.05770
  Summary: Propose to use normalizing flows in variational inference. Discuss finite and infinitesimal flows. Uses invertible flows: planar, radial. Comparison with NICE.
- **RealNVP**: Density estimation using Real NVP
  https://arxiv.org/pdf/1605.08803.pdf
  Summary: Authors of NICE. The same idea and architecture, more practical. Lots of experiments and images. Coupling layers with checkerboard and channel-wise permutations.
- **IAF**: Improving Variational Inference with Inverse Autoregressive Flow
  https://arxiv.org/abs/1606.04934
  Summary: Introduce inverse autoregressive flow (IAF). Models each autoregressive conditional as gaussian with autoregressive means and covariances. Inverse transformation allows to parallelize sampling.
- **MAF**: Masked Autoregressive Flow for Density Estimation
  https://arxiv.org/pdf/1705.07057.pdf
  Summary: Similar to IAF. Give comprehensive overview with link to IAF and RealNVP. MAF is suitable for density estimation, IAF as a recognition network.
- **Parallel WaveNet**: Fast High-Fidelity Speech Synthesis
  https://arxiv.org/pdf/1711.10433.pdf
  Summary: WaveNet is MAF (sequential generation). To exploit IAF fast sampling, knowledge distillation used. Teacher network is large WaveNet, student - is a IAF small WaveNet (generate samples from noise is parallel). The loss is KL divergence between student and teacher distributions. The additional perceptual, contrastive and power losses used to create more natural sounds.