

Deep Generative Models

Lecture 6

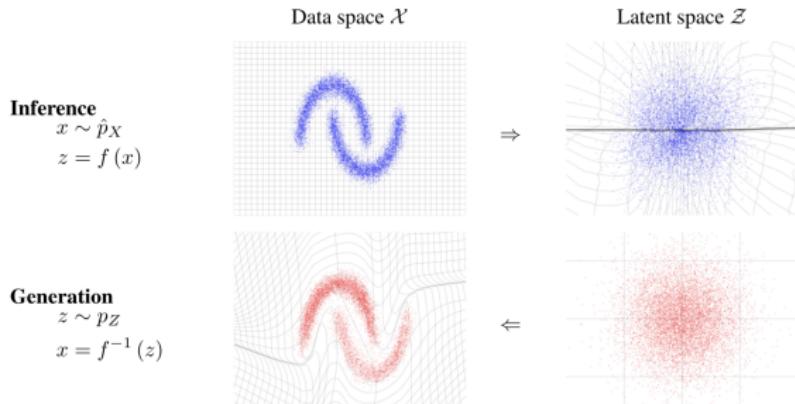
Roman Isachenko



Ozon Masters

Spring, 2021

Recap of previous lecture



Flow likelihood

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

What we want

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

Recap of previous lecture

Planar flow

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

Sylvester flow

$$g(\mathbf{z}, \theta) = \mathbf{z} + \mathbf{A} h(\mathbf{B}\mathbf{z} + \mathbf{b}).$$

NICE/RealNVP: Affine coupling law

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

Glow: invertible 1x1 conv

$$\mathbf{W} = \mathbf{P}\mathbf{L}(\mathbf{U} + \text{diag}(\mathbf{s})).$$

Rezende D. J., Mohamed S. *Variational Inference with Normalizing Flows*, 2015

Berg R. et al. *Sylvester normalizing flows for variational inference*, 2018

Dinh L., Krueger D., Bengio Y. *NICE: Non-linear Independent Components Estimation*, 2014

Dinh L., Sohl-Dickstein J., Bengio S. *Density estimation using Real NVP*, 2016

Kingma D. P., Dhariwal P. *Glow: Generative Flow with Invertible 1x1 Convolutions*, 2018

Recap of previous lecture

ELBO

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

- ▶ Normal variational distribution
 $q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\mu_\phi(\mathbf{x}), \sigma_\phi^2(\mathbf{x}))$ is poor (e.g. has only one mode).
- ▶ Flows models convert a simple base distribution to a complex one using invertible transformation with simple Jacobian.

Flow model in latent space

$$\log q_K(\mathbf{z}_K|\mathbf{x}, \phi_*) = \log q(\mathbf{z}_0|\mathbf{x}, \phi) - \sum_{k=1}^K \log \left| \det \left(\frac{\partial g_k(\mathbf{z}_{k-1}, \phi_k)}{\partial \mathbf{z}_{k-1}} \right) \right|.$$

Let use $q_K(\mathbf{z}_K|\mathbf{x}, \phi_*)$, $\phi_* = \{\phi, \phi_1, \dots, \phi_K\}$ as a variational distribution. Here ϕ – encoder parameters, $\{\phi_k\}_{k=1}^K$ – flow parameters.

Recap of previous lecture

Variational distribution

$$\log q_K(\mathbf{z}_K | \mathbf{x}, \phi_*) = \log q(\mathbf{z}_0 | \mathbf{x}, \phi) - \sum_{k=1}^K \log \left| \det \left(\frac{\partial g_k(\mathbf{z}_{k-1}, \phi_k)}{\partial \mathbf{z}_{k-1}} \right) \right|.$$

ELBO objective

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

- ▶ Obtain samples \mathbf{z}_0 from the encoder.
- ▶ Apply flow model $\mathbf{z}_K = g(\mathbf{z}_0, \{\phi_k\}_{k=1}^K)$.
- ▶ Compute likelihood for \mathbf{z}_K using the decoder, base distribution for \mathbf{z}_0 and the Jacobian.
- ▶ We do not need inverse flow function, if we use flows in variational inference.

Dequantization

- ▶ Images are discrete data, pixels lies in the $[0, 255]$ integer domain (the model is $P(\mathbf{x}|\theta) = \text{Categorical}(\boldsymbol{\pi}(\theta))$).
- ▶ Flow is a continuous model (it works with continuous data \mathbf{x}).

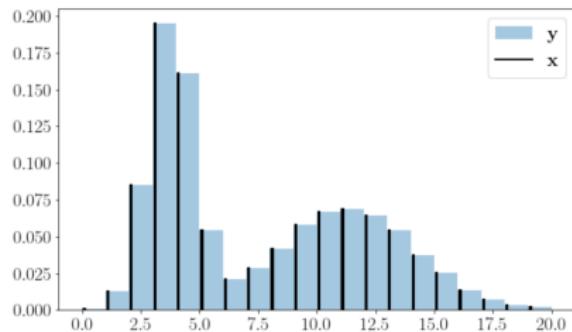
Fitting a continuous density model to discrete data, could produce a degenerate solution with all probability mass on discrete values.
How to convert discrete data distribution to the continuous one?

Uniform dequantization

$$\mathbf{x} \sim \text{Categorical}(\boldsymbol{\pi})$$

$$\mathbf{u} \sim U[0, 1]$$

$$\mathbf{y} = \mathbf{x} + \mathbf{u} \sim \text{Continuous}$$



Uniform dequantization

Statement

Fitting continuous model $p(\mathbf{y}|\theta)$ on uniformly dequantized data $\mathbf{y} = \mathbf{x} + \mathbf{u}$, $\mathbf{u} \sim U[0, 1]$ is equivalent to maximization of a lower bound on the log-likelihood for a discrete model:

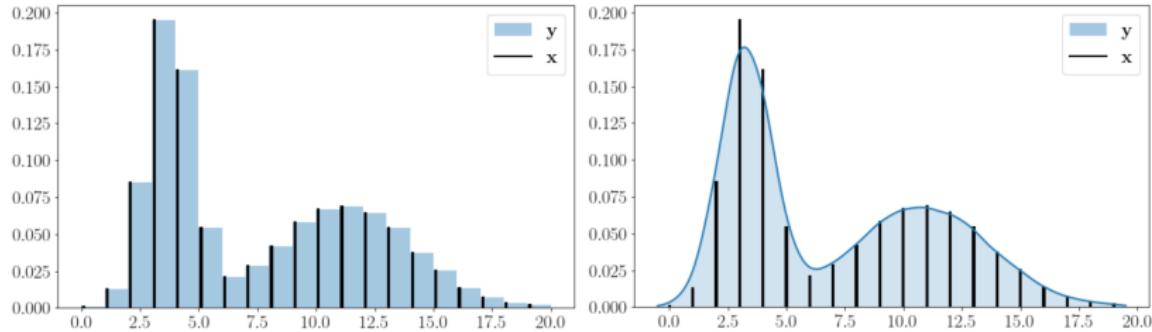
$$P(\mathbf{x}|\theta) = \int_{U[0,1]} p(\mathbf{x} + \mathbf{u}|\theta) d\mathbf{u}$$

Thus, maximizing the log-likelihood of the continuous model on \mathbf{y} cannot lead to the collapsing onto the discrete data (objective is bounded above by the log-likelihood of a discrete model).

Proof

$$\begin{aligned} \log P(\mathbf{x}|\theta) &= \log \int_{U[0,1]} p(\mathbf{x} + \mathbf{u}|\theta) d\mathbf{u} \geq \\ &\geq \int_{U[0,1]} \log p(\mathbf{x} + \mathbf{u}|\theta) d\mathbf{u} = \log p(\mathbf{y}|\theta). \end{aligned}$$

Variational dequantization



- ▶ $p(y|\theta)$ assign uniform density to unit hypercubes $x + U[0, 1]$ (left fig).
- ▶ Neural network density models is a smooth function approximator (right fig).
- ▶ Smooth dequantization is more natural.

How to make the smooth dequantization?

Flow++

Variational dequantization

Introduce variational dequantization noise distribution $q(\mathbf{u}|\mathbf{x})$ and treat it as an approximate posterior.

Variational lower bound

$$\begin{aligned}\log P(\mathbf{x}|\theta) &= \left[\log \int q(\mathbf{u}|\mathbf{x}) \frac{p(\mathbf{x} + \mathbf{u}|\theta)}{q(\mathbf{u}|\mathbf{x})} d\mathbf{u} \right] \geq \\ &\geq \int q(\mathbf{u}|\mathbf{x}) \log \frac{p(\mathbf{x} + \mathbf{u}|\theta)}{q(\mathbf{u}|\mathbf{x})} d\mathbf{u} = \mathcal{L}(q, \theta).\end{aligned}$$

Uniform dequantization bound

$$\begin{aligned}\log P(\mathbf{x}|\theta) &= \log \int_{U[0,1]} p(\mathbf{x} + \mathbf{u}|\theta) d\mathbf{u} \geq \\ &\geq \int_{U[0,1]} \log p(\mathbf{x} + \mathbf{u}|\theta) d\mathbf{u} = \log p(\mathbf{y}|\theta).\end{aligned}$$

Flow++

Variational lower bound

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

Let $\mathbf{u} = h(\epsilon, \phi)$ is a flow model with base distribution $\epsilon \sim p(\epsilon) = \mathcal{N}(0, \mathbf{I})$:

$$q(\mathbf{u}|\mathbf{x}) = p(h^{-1}(\mathbf{u}, \phi)) \cdot \left| \det \frac{\partial h^{-1}(\mathbf{u}, \phi)}{\partial \mathbf{u}} \right|.$$

Then

$$\log P(\mathbf{x}|\theta) \geq \mathcal{L}(\phi, \theta) = \int p(\epsilon) \log \left(\frac{p(\mathbf{x} + h(\epsilon, \phi)|\theta)}{p(\epsilon) \cdot \left| \det \frac{\partial h(\epsilon, \phi)}{\partial \epsilon} \right|^{-1}} \right) d\epsilon.$$

Variational lower

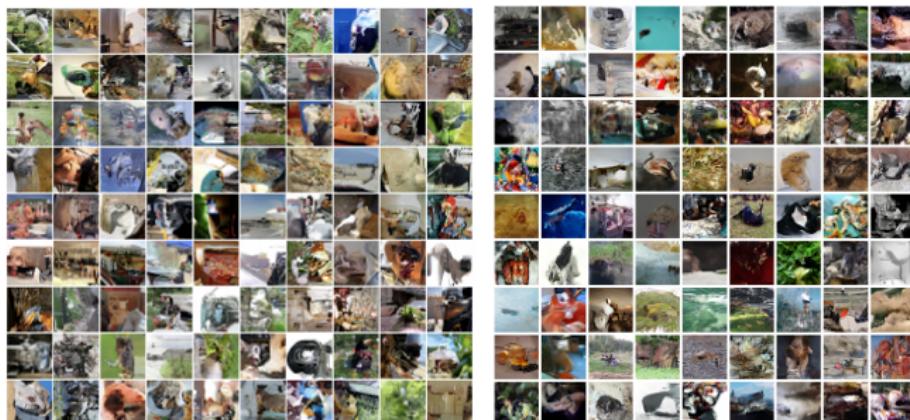
$$\log P(\mathbf{x}|\theta) \geq \int p(\epsilon) \log \left(\frac{p(\mathbf{x} + h(\epsilon, \phi))}{p(\epsilon) \cdot \left| \det \frac{\partial h(\epsilon, \phi)}{\partial \epsilon} \right|^{-1}} \right) d\epsilon.$$

- ▶ If $p(\mathbf{x} + \mathbf{u}|\theta)$ is also a flow model, it is straightforward to calculate stochastic gradient of this ELBO.
- ▶ Uniform dequantization is a special case of variational dequantization ($q(\mathbf{u}|\mathbf{x}) = U[0, 1]$). The gap between $\log P(\mathbf{x}|\theta)$ and the derived ELBO is $KL(q(\mathbf{u}|\mathbf{x})||p(\mathbf{u}|\mathbf{x}))$.
- ▶ In the case of uniform dequantization the model unnaturally places uniform density over each hypercube $\mathbf{x} + U[0, 1]$ due to inexpressive distribution q .

Flow++

Table 1. Unconditional image modeling results in bits/dim

Model family	Model	CIFAR10	ImageNet 32x32	ImageNet 64x64
Non-autoregressive	RealNVP (Dinh et al., 2016)	3.49	4.28	—
	Glow (Kingma & Dhariwal, 2018)	3.35	4.09	3.81
	IAF-VAE (Kingma et al., 2016)	3.11	—	—
	Flow++ (ours)	3.08	3.86	3.69
Autoregressive	Multiscale PixelCNN (Reed et al., 2017)	—	3.95	3.70
	PixelCNN (van den Oord et al., 2016b)	3.14	—	—
	PixelRNN (van den Oord et al., 2016b)	3.00	3.86	3.63
	Gated PixelCNN (van den Oord et al., 2016c)	3.03	3.83	3.57
	PixelCNN++ (Salimans et al., 2017)	2.92	—	—
	Image Transformer (Parmar et al., 2018)	2.90	3.77	—
	PixelSNAIL (Chen et al., 2017)	2.85	3.80	3.52



(a) PixelCNN

(b) Flow++

Gaussian autoregressive model

Consider autoregressive model

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

with conditionals

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

Sampling

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

Sampling from autoregressive model is sequential.

Note that we could interpret this sampling as a transformation $\mathbf{x} = g(\mathbf{z}, \boldsymbol{\theta})$, where \mathbf{z} comes from base distribution $\mathcal{N}(0, 1)$.

Gaussian autoregressive model

Sampling

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

Jacobian

$$\log \left| \det \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}} \right) \right| = - \log \left| \det \left(\frac{\partial g(\mathbf{z}, \boldsymbol{\theta})}{\partial \mathbf{z}} \right) \right| = - \sum_{i=1}^m \log \hat{\sigma}_i(\mathbf{x}_{1:i-1}).$$

Inverse transform

$$z_i = (x_i - \hat{\mu}_i(\mathbf{x}_{1:i-1})) \cdot \frac{1}{\hat{\sigma}_i(\mathbf{x}_{1:i-1})}.$$

We get an autoregressive model with tractable (triangular) Jacobian, which is easily invertible. It is a flow!

Inverse autoregressive flow (IAF)

Gaussian autoregressive model ($\mathbf{z} \rightarrow \mathbf{x}$)

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

$$z_i = (x_i - \hat{\mu}_i(\mathbf{x}_{1:i-1})) \cdot \frac{1}{\hat{\sigma}_i(\mathbf{x}_{1:i-1})}.$$

This process is sequential.

Let use the following reparametrization: $\sigma = \frac{1}{\hat{\sigma}}$; $\mu = -\frac{\hat{\mu}}{\hat{\sigma}}$.

Inverse transform ($\mathbf{x} \rightarrow \mathbf{z}$)

$$z_i = \sigma_i(\mathbf{x}_{1:i-1}) \cdot x_i + \mu_i(\mathbf{x}_{1:i-1}).$$

$$x_i = (z_i - \mu_i(\mathbf{x}_{1:i-1})) \cdot \frac{1}{\sigma_i(\mathbf{x}_{1:i-1})}.$$

This process is **not** sequential.

Inverse autoregressive flow (IAF)

Inverse transform ($\mathbf{x} \rightarrow \mathbf{z}$)

$$z_i = \sigma_i(\mathbf{x}_{1:i-1}) \cdot x_i + \mu_i(\mathbf{x}_{1:i-1}).$$

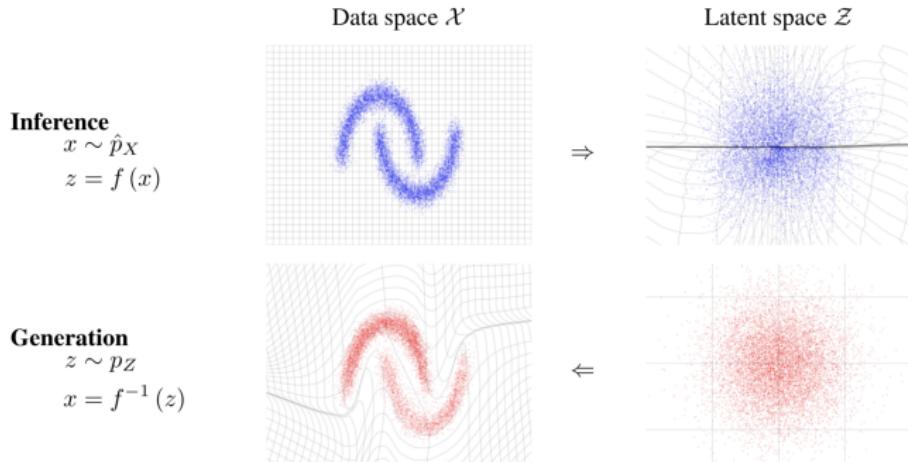
$$x_i = (z_i - \mu_i(\mathbf{x}_{1:i-1})) \cdot \frac{1}{\sigma_i(\mathbf{x}_{1:i-1})}.$$

Inverse autoregressive flow use such inverted autoregressive model as a flow in VAE:

$$\mathbf{z}_0 = \boldsymbol{\sigma}(\mathbf{x}) \cdot \epsilon + \boldsymbol{\mu}(\mathbf{x}), \quad \epsilon \sim \mathcal{N}(0, 1); \quad \sim q(\mathbf{z}_0 | \mathbf{x}, \boldsymbol{\phi}).$$

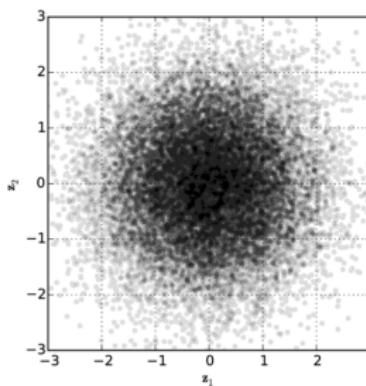
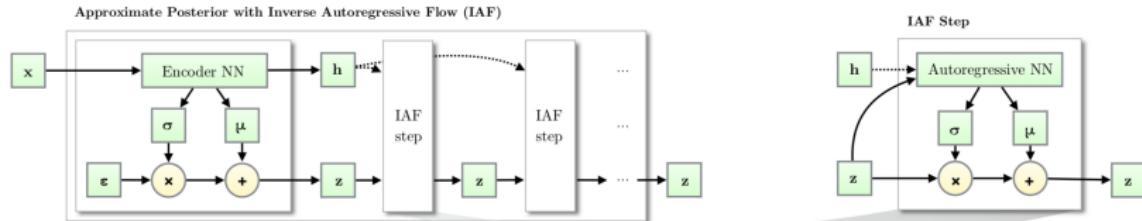
$$\mathbf{z}_k = \boldsymbol{\sigma}_k(\mathbf{z}_{k-1}) \cdot \mathbf{z}_{k-1} + \boldsymbol{\mu}_k(\mathbf{z}_{k-1}), \quad k \geq 1; \quad \sim q_k(\mathbf{z}_k | \mathbf{x}, \boldsymbol{\phi}, \{\boldsymbol{\phi}_j\}_{j=1}^k).$$

Flows

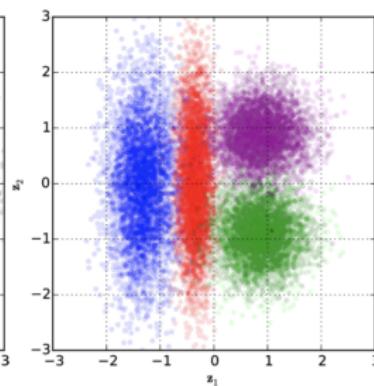


- ▶ Inference mode in autoregressive flows is used for density estimation tasks.
- ▶ Generation mode in autoregressive flows (IAF) is used for stochastic variational inference to get a more flexible posterior distribution.

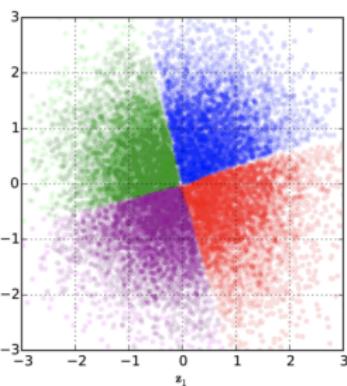
Inverse autoregressive flow (IAF)



(a) Prior distribution



(b) Posteriors in standard VAE



(c) Posteriors in VAE with IAF

Inverse autoregressive flow (IAF)

Gaussian autoregressive model

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

Inverse transform

$$z_i = (x_i - \hat{\mu}_i(\mathbf{x}_{1:i-1})) \cdot \frac{1}{\hat{\sigma}_i(\mathbf{x}_{1:i-1})};$$

$$z_i = \sigma_i(\mathbf{x}_{1:i-1}) \cdot x_i + \mu_i(\mathbf{x}_{1:i-1}).$$

Inverse autoregressive flow

$$x_i = \sigma_i(\mathbf{z}_{1:i-1}) \cdot z_i + \mu_i(\mathbf{z}_{1:i-1}).$$

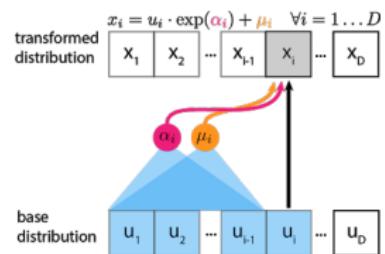
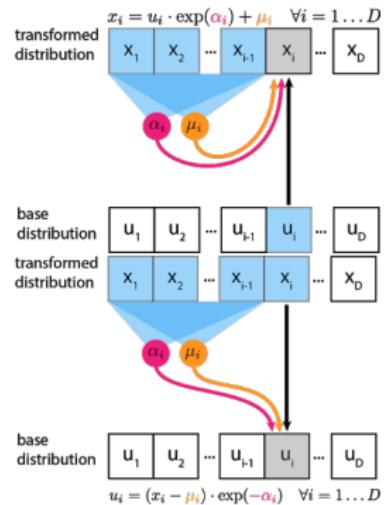


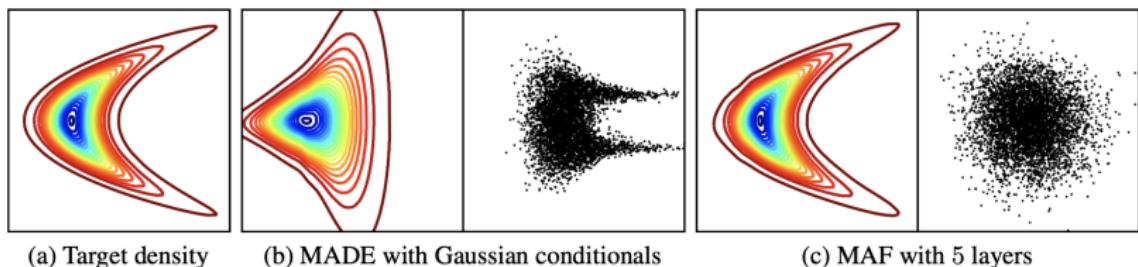
image credit: <https://blog.evjang.com/2018/01/nf2.html>

Masked autoregressive flow (MAF)

Gaussian autoregressive model

$$p(\mathbf{x}|\theta) = \prod_{i=1}^m p(x_i|\mathbf{x}_{1:i-1}, \theta) = \prod_{i=1}^m \mathcal{N}(x_i|\mu_i(\mathbf{x}_{1:i-1}), \sigma_i^2(\mathbf{x}_{1:i-1})).$$

We could use MADE (masked autoencoder) as conditional model.
The sampling order could be crucial.

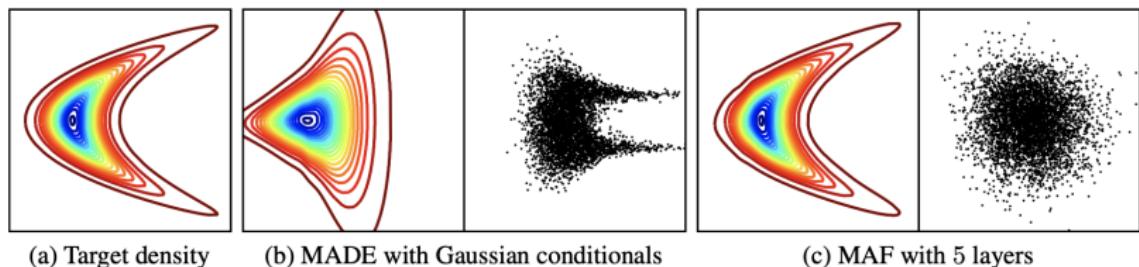


Samples from the base distribution could be an indicator of how good the flow was fitted.

Masked autoregressive flow (MAF)

Gaussian autoregressive model

$$p(\mathbf{x}|\theta) = \prod_{i=1}^m p(x_i|\mathbf{x}_{1:i-1}, \theta) = \prod_{i=1}^m \mathcal{N}(x_i | \mu_i(\mathbf{x}_{1:i-1}), \sigma_i^2(\mathbf{x}_{1:i-1})) .$$



MAF is just a stacked MADE model.

MAF vs IAF

Sampling and inverse transform in MAF

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

$$z_i = (x_i - \hat{\mu}_i(\mathbf{x}_{1:i-1})) \cdot \frac{1}{\hat{\sigma}_i(\mathbf{x}_{1:i-1})}.$$

- ▶ Sampling is slow (sequential).
- ▶ Density estimation is fast.

Sampling and inverse transform in IAF

$$x_i = \sigma_i(\mathbf{z}_{1:i-1}) \cdot z_i + \mu_i(\mathbf{z}_{1:i-1}).$$

$$z_i = (x_i - \mu_i(\mathbf{z}_{1:i-1})) \cdot \frac{1}{\sigma_i(\mathbf{z}_{1:i-1})}.$$

- ▶ Sampling is fast.
- ▶ Density estimation is slow (sequential).

MAF vs IAF

Theorem

Training a MAF with maximum likelihood corresponds to fitting an implicit IAF with stochastic variational inference where the posterior is taken to be the base density $\pi(\mathbf{z})$:

$$\max_{\theta} p(\mathbf{X}|\theta) \Leftrightarrow \min_{\theta} KL(p(\mathbf{z}|\theta)||\pi(\mathbf{z})).$$

- ▶ $\pi(\mathbf{z})$ is a base distribution; $\pi(\mathbf{x})$ is a data distribution.
- ▶ $\mathbf{z} = f(\mathbf{x}, \theta)$ – MAF model; $\mathbf{x} = g(\mathbf{z}, \theta)$ – IAF model.

$$\log p(\mathbf{z}|\theta) = \log \pi(g(\mathbf{z}, \theta)) + \log \left| \det \left(\frac{\partial g(\mathbf{z}, \theta)}{\partial \mathbf{z}} \right) \right|$$

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

MAF vs IAF

Theorem

Training a MAF with maximum likelihood corresponds to fitting an implicit IAF with stochastic variational inference where the posterior is taken to be the base density $\pi(\mathbf{z})$:

$$\max_{\theta} p(\mathbf{X}|\theta) \Leftrightarrow \min_{\theta} KL(p(\mathbf{z}|\theta)||\pi(\mathbf{z})).$$

Proof

$$\begin{aligned}KL(p(\mathbf{z}|\theta)||\pi(\mathbf{z})) &= \mathbb{E}_{p(\mathbf{z}|\theta)} [\log p(\mathbf{z}|\theta) - \log \pi(\mathbf{z})] = \\&= \mathbb{E}_{p(\mathbf{z}|\theta)} \left[\log \pi(g(\mathbf{z}, \theta)) + \log \left| \det \left(\frac{\partial g(\mathbf{z}, \theta)}{\partial \mathbf{z}} \right) \right| - \log \pi(\mathbf{z}) \right] = \\&= \mathbb{E}_{\pi(\mathbf{x})} \left[\log \pi(\mathbf{x}) - \log \left| \det \left(\frac{\partial f(\mathbf{x}, \theta)}{\partial \mathbf{x}} \right) \right| - \log \pi(f(\mathbf{x}, \theta)) \right].\end{aligned}$$

MAF vs IAF

Proof (continued)

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

$$\begin{aligned} \arg \min_{\theta} KL(\pi(\mathbf{x})||p(\mathbf{x}|\theta)) &= \arg \min_{\theta} \mathbb{E}_{\pi(\mathbf{x})} [\log \pi(\mathbf{x}) - \log p(\mathbf{x}|\theta)] \\ &= \arg \max_{\theta} \mathbb{E}_{\pi(\mathbf{x})} \log p(\mathbf{x}|\theta) \end{aligned}$$

Unbiased estimator is MLE:

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

MAF vs IAF vs RealNVP

MAF

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

- ▶ Calculating the density $p(\mathbf{x}|\theta)$ - 1 pass.
- ▶ Sampling - m passes.

IAF

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

- ▶ Calculating the density $p(\mathbf{x}|\theta)$ - m passes.
- ▶ Sampling - 1 pass.

RealNVP

$$\mathbf{x}_{1:d} = \mathbf{z}_{1:d};$$

$$\mathbf{x}_{d:m} = \mathbf{z}_{d:m} \odot \exp(c_1(\mathbf{z}_{1:d}, \theta)) + c_2(\mathbf{x}_{1:d}, \theta).$$

MAF vs IAF vs RealNVP

RealNVP

$$\mathbf{x}_{1:d} = \mathbf{z}_{1:d};$$

$$\mathbf{x}_{d:m} = \mathbf{z}_{d:m} \odot \exp(c_1(\mathbf{z}_{1:d}, \theta)) + c_2(\mathbf{x}_{1:d}, \theta).$$

- ▶ Calculating the density $p(\mathbf{x}|\theta)$ - 1 pass.
- ▶ Sampling - 1 pass.

RealNVP is a special case of MAF and IAF:

MAF

$$\begin{cases} \hat{\mu}_i = \hat{\sigma}_i = 0, i = 1, \dots, d; \\ \hat{\mu}_i, \hat{\sigma}_i - \text{functions of } \mathbf{x}_{1:d}, i = d + 1, \dots, m. \end{cases}$$

IAF

$$\begin{cases} \mu_i = \sigma_i = 0, i = 1, \dots, d; \\ \mu_i, \sigma_i - \text{functions of } \mathbf{z}_{1:d}, i = d + 1, \dots, m. \end{cases}$$

MAF/IAF pros and cons

MAF

- ▶ Sampling is slow.
- ▶ Likelihood evaluation is fast.

IAF

- ▶ Sampling is fast.
- ▶ Likelihood evaluation is slow.

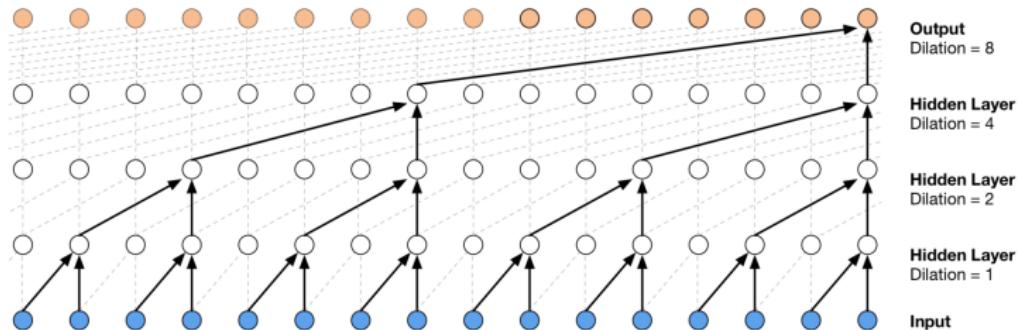
How to take the best of both worlds?

WaveNet (2016)

Autoregressive model for raw audio waveforms generation

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

The model uses causal dilated convolutions.



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) → increase dilated convolution filter size from 2 to 3;
- ▶ 16-bit signals → mixture of logistics instead of categorical distribution.

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

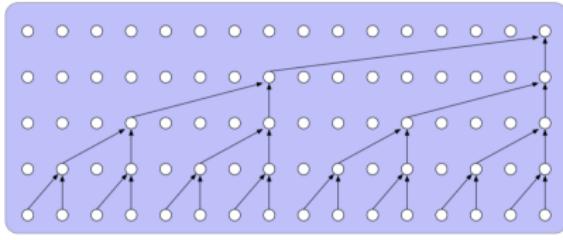
$$KL(p_s || p_t) = H(p_s, p_t) - H(p_s).$$

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

Parallel WaveNet, 2017

WaveNet Teacher

Linguistic features \dashrightarrow



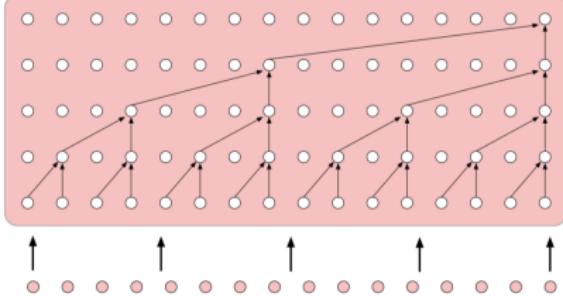
Teacher Output
 $P(x_i | x_{<i})$

Generated Samples
 $x_i = g(z_i | z_{<i})$

Student Output
 $P(x_i | z_{<i})$

WaveNet Student

Linguistic features \dashrightarrow



Input noise
 z_i

Summary

- ▶ To apply continuous model to discrete distribution the standard practice is to dequantize data at first.
- ▶ Uniform dequantization is the simplest form of dequantization. Variational dequantization is more natural type that was proposed in Flow++ model.
- ▶ Gaussian autoregressive model is a special type of flow (RealNVP model is a special type of this autoregressive model).
- ▶ MAF is an example of such model which is suitable for density estimation tasks.
- ▶ IAF used the inverse autoregressive transformation for variational inference task.