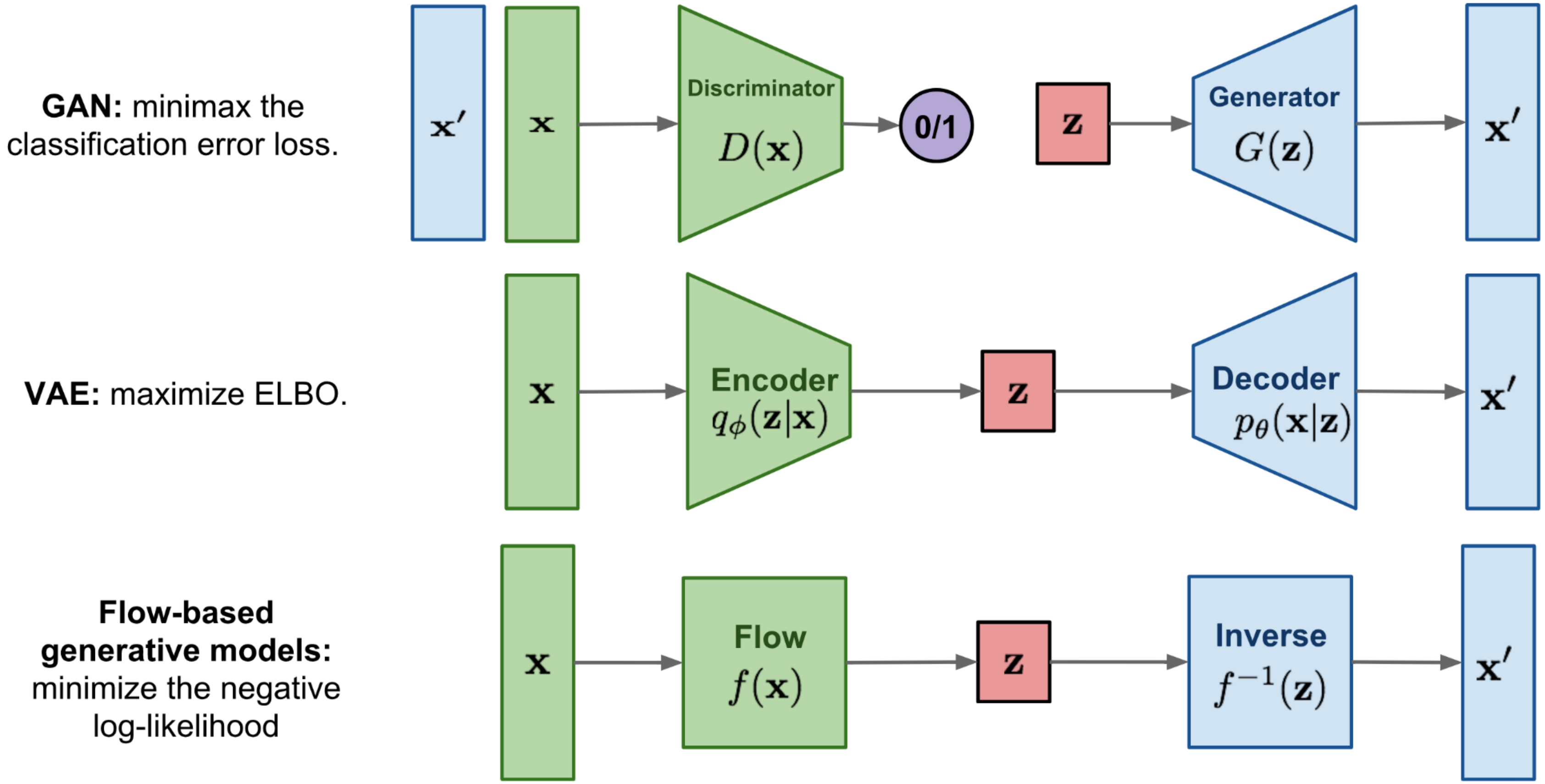
Wheels for Flow Model

Zhiyuan Tang

2020.4.8

Flow model



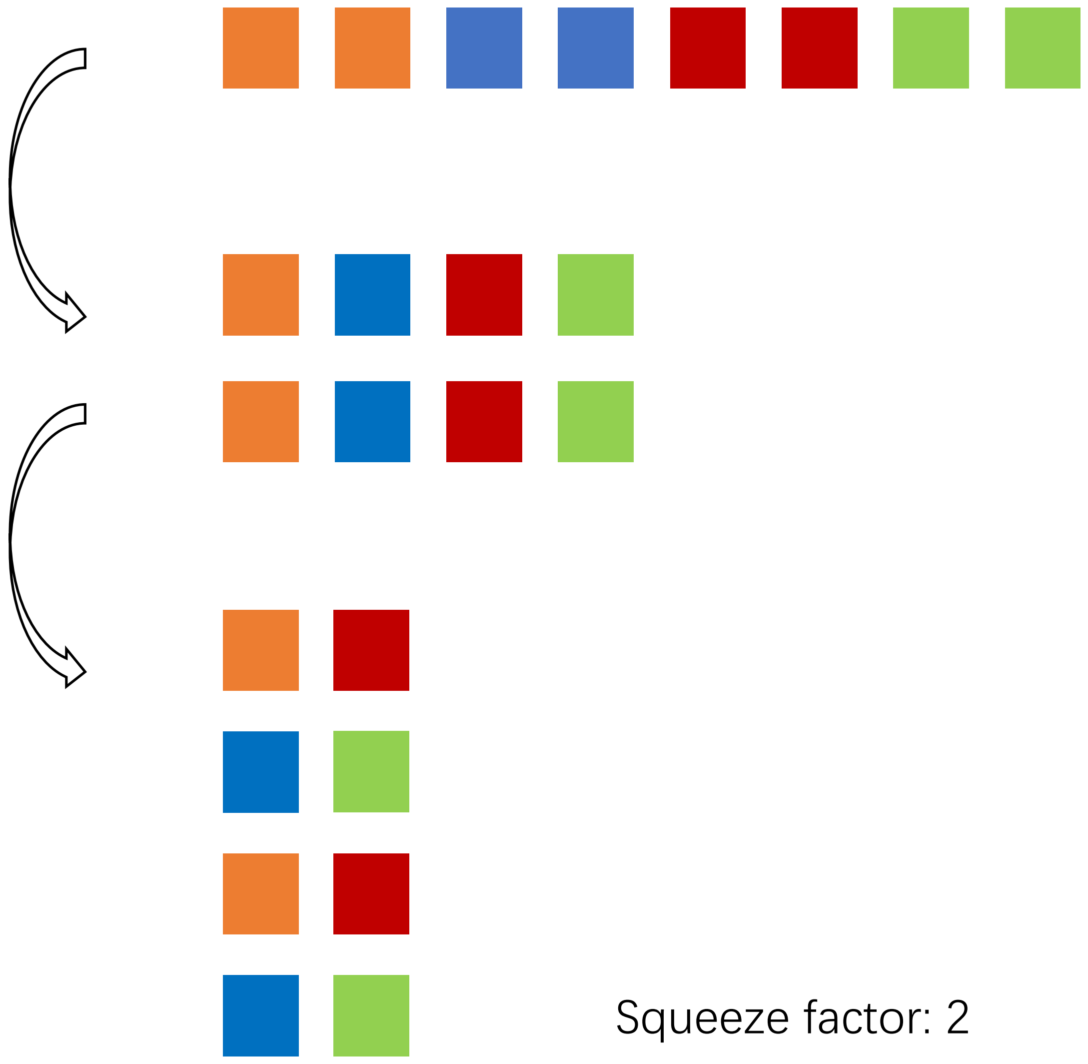
https://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html

Wheels for Flow

For practice:

- Easily invertible
- Easily computation of Jacobian determinant

Squeezer



Squeeze factor: 2

Additive/affine coupling layer

$$\begin{cases} \mathbf{y}_{1:d} & = \mathbf{x}_{1:d} \\ \mathbf{y}_{d+1:D} & = \mathbf{x}_{d+1:D} + m(\mathbf{x}_{1:d}) \end{cases} \Leftrightarrow \begin{cases} \mathbf{x}_{1:d} & = \mathbf{y}_{1:d} \\ \mathbf{x}_{d+1:D} & = \mathbf{y}_{d+1:D} - m(\mathbf{y}_{1:d}) \end{cases}$$

$$\begin{cases} \mathbf{y}_{1:d} & = \mathbf{x}_{1:d} \\ \mathbf{y}_{d+1:D} & = \mathbf{x}_{d+1:D} \odot \exp(s(\mathbf{x}_{1:d})) + t(\mathbf{x}_{1:d}) \end{cases} \Leftrightarrow \begin{cases} \mathbf{x}_{1:d} & = \mathbf{y}_{1:d} \\ \mathbf{x}_{d+1:D} & = (\mathbf{y}_{d+1:D} - t(\mathbf{y}_{1:d})) \odot \exp(-s(\mathbf{y}_{1:d})) \end{cases}$$

Density estimation using real nvp

Nice: Non-linear independent components estimation

Invertible convolution

Table 1: The three main components of our proposed flow, their reverses, and their log-determinants. Here, $\mathbf{x}$ signifies the input of the layer, and $\mathbf{y}$ signifies its output. Both $\mathbf{x}$ and $\mathbf{y}$ are tensors of shape $[h \times w \times c]$ with spatial dimensions (h, w) and channel dimension c. With (i, j) we denote spatial indices into tensors $\mathbf{x}$ and $\mathbf{y}$. The function $\texttt{NN}()$ is a nonlinear mapping, such as a (shallow) convolutional neural network like in ResNets (He et al., 2016) and RealNVP (Dinh et al., 2016).

Description	Function	Reverse Function	Log-determinant
Actnorm. See Section 3.1.	$\forall i, j : \mathbf{y}_{i,j} = \mathbf{s} \odot \mathbf{x}_{i,j} + \mathbf{b}$	$\forall i, j : \mathbf{x}_{i,j} = (\mathbf{y}_{i,j} - \mathbf{b}) / \mathbf{s}$	$h \cdot w \cdot \texttt{sum}(\log \lvert\mathbf{s}\rvert)$
Invertible 1×1 convolution. $\mathbf{W} : [c \times c]$. See Section 3.2.	$\forall i, j : \mathbf{y}_{i,j} = \mathbf{W}\mathbf{x}_{i,j}$	$\forall i, j : \mathbf{x}_{i,j} = \mathbf{W}^{-1}\mathbf{y}_{i,j}$	$h \cdot w \cdot \log \lvert\det(\mathbf{W})\rvert$ or $h \cdot w \cdot \texttt{sum}(\log \lvert\mathbf{s}\rvert)$ (see eq. (10))
Affine coupling layer. See Section 3.3 and (Dinh et al., 2014)	$\mathbf{x}_a, \mathbf{x}_b = \texttt{split}(\mathbf{x})$ $(\log \mathbf{s}, \mathbf{t}) = \texttt{NN}(\mathbf{x}_b)$ $\mathbf{s} = \exp(\log \mathbf{s})$ $\mathbf{y}_a = \mathbf{s} \odot \mathbf{x}_a + \mathbf{t}$ $\mathbf{y}_b = \mathbf{x}_b$ $\mathbf{y} = \texttt{concat}(\mathbf{y}_a, \mathbf{y}_b)$	$\mathbf{y}_a, \mathbf{y}_b = \texttt{split}(\mathbf{y})$ $(\log \mathbf{s}, \mathbf{t}) = \texttt{NN}(\mathbf{y}_b)$ $\mathbf{s} = \exp(\log \mathbf{s})$ $\mathbf{x}_a = (\mathbf{y}_a - \mathbf{t}) / \mathbf{s}$ $\mathbf{x}_b = \mathbf{y}_b$ $\mathbf{x} = \texttt{concat}(\mathbf{x}_a, \mathbf{x}_b)$	$\texttt{sum}(\log(\lvert\mathbf{s}\rvert))$

Glow: Generative flow with invertible 1x1 convolutions

Masked Autoregressive Flow (MAF)

$$x_i = u_i \exp \alpha_i + \mu_i \quad \text{where} \quad \mu_i = f_{\mu_i}(\mathbf{x}_{1:i-1}), \;\; \alpha_i = f_{\alpha_i}(\mathbf{x}_{1:i-1}) \;\; \text{and} \;\; u_i \sim \mathcal{N}(0, 1). \tag{3}$$

$$u_i = (x_i - \mu_i) \exp(-\alpha_i) \quad \text{where} \quad \mu_i = f_{\mu_i}(\mathbf{x}_{1:i-1}) \;\; \text{and} \;\; \alpha_i = f_{\alpha_i}(\mathbf{x}_{1:i-1}). \tag{4}$$

$$\left| \det \left(\frac{\partial f^{-1}}{\partial \mathbf{x}} \right) \right| = \exp\left(-\sum_i \alpha_i \right) \quad \text{where} \quad \alpha_i = f_{\alpha_i}(\mathbf{x}_{1:i-1}). \tag{5}$$

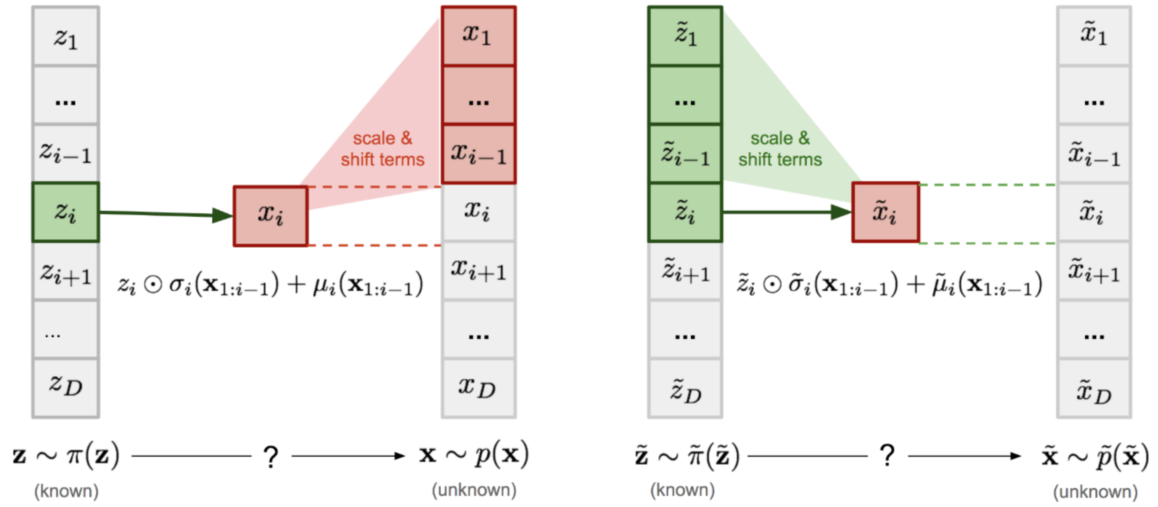
Masked autoregressive flow for density estimation

Inverse Autoregressive Flow (IAF)

$$x_i = u_i \exp \alpha_i + \mu_i \quad \textbf{where} \quad \mu_i = f_{\mu_i}(\mathbf{u}_{1:i-1}) \ \ \textbf{and} \ \ \alpha_i = f_{\alpha_i}(\mathbf{u}_{1:i-1}).$$

Improved variational inference with inverse autoregressive flow

MAF vs. IAF



Masked Autoregressive Flow (MAF) Inverse Autoregressive Flow (IAF)

Both MAF and IAF can be seen as more flexible (but different) generalizations of **coupling layer** which can both generate data and estimate densities with one forward pass only.

	Base distribution	Target distribution	Model	Data generation	Density estimation
MAF	$\mathbf{z} \sim \pi(\mathbf{z})$	$\mathbf{x} \sim p(\mathbf{x})$	$x_i = z_i \odot \sigma_i(\mathbf{x}_{1:i-1}) + \mu_i(\mathbf{x}_{1:i-1})$	Sequential; slow	One pass; fast
IAF	$\tilde{\mathbf{z}} \sim \tilde{\pi}(\tilde{\mathbf{z}})$	$\tilde{\mathbf{x}} \sim \tilde{p}(\tilde{\mathbf{x}})$	$\tilde{x}_i = \tilde{z}_i \odot \tilde{\sigma}_i(\tilde{\mathbf{z}}_{1:i-1}) + \tilde{\mu}_i(\tilde{\mathbf{z}}_{1:i-1})$	One pass; fast	Sequential; slow

https://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html

MAF vs. IAF

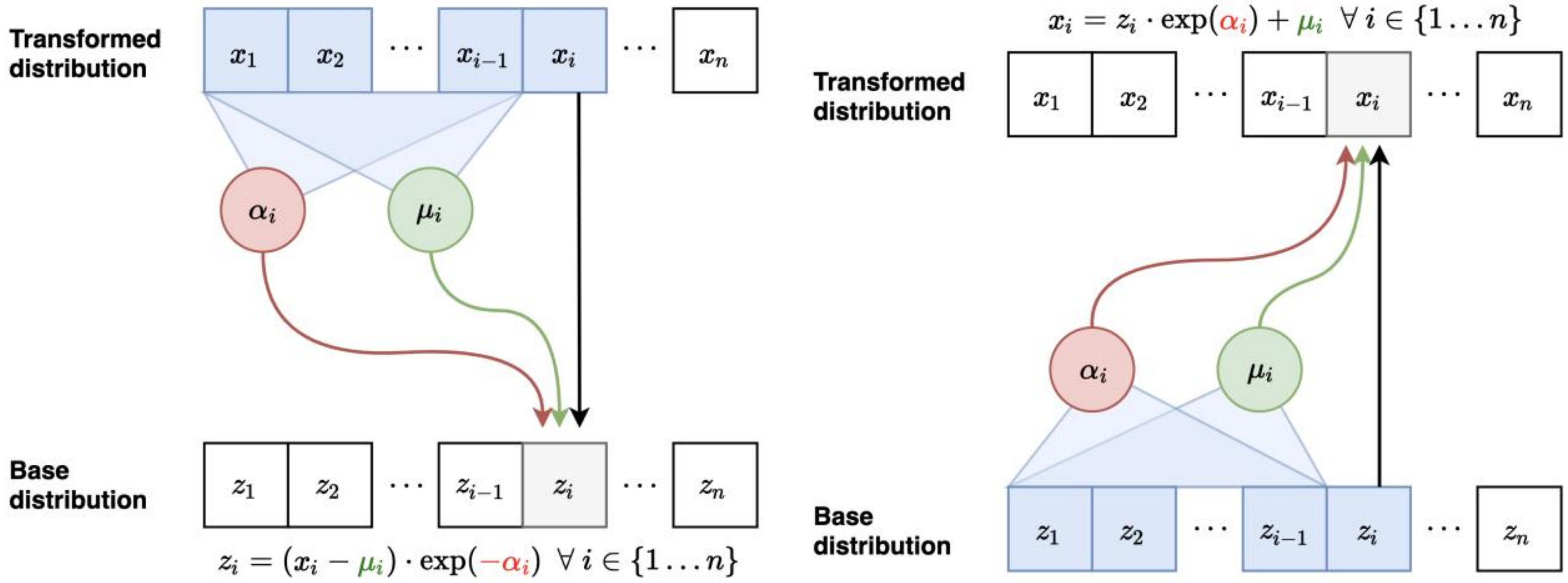


Figure: Inverse pass of MAF (**left**) vs. Forward pass of IAF (**right**)

https://deepgenerativemodels.github.io/syllabus.html

Invertible ResNet

Theorem (sufficient condition for invertible residual layer):

Let $F_\theta^t(x) = x + g_\theta^t(x)$ be a residual layer, then it is invertible if

$$\mathrm{Lip}(g_\theta^t) < 1$$

where

$$\|g(x) - g(y)\|_2 \leq \mathrm{Lip}(g)\|x - y\|_2$$

How to build i-ResNets

- Satisfy Lip-condition: data-independent upper bound

$$g = W_3 \circ \phi \circ W_2 \circ \phi \circ W_1 \circ \phi$$

$$\mathrm{Lip}(g) \leq \|W_3\|_2 \cdot \|W_2\|_2 \cdot \|W_1\|_2$$

- Spectral normalization (Miyato et al. 2018, Gouk et al. 2018)

$$\tilde{W} = c\frac{W}{\hat{\sigma}_1}, \quad 0 < c < 1$$

$\hat{\sigma}_1$ approx of largest singular value via power-iteration

```
def invertible_residual_block(self):
    layers = []
    layers.append(nn.ReLU)
    layers.append(spectral_norm(nn.Linear(in_dim, hidden_dim)))
    layers.append(nn.ReLU)
    layers.append(spectral_norm(nn.Linear(hidden_dim, in_dim)))
```


Invertible residual networks

http://www.cs.toronto.edu/~duvenaud/talks/iresnet-slides.pdf

Spectral Normalization for Generative Adversarial Networks

Invertible ResNet

Invertible Residual Networks

Method	ResNet	NICE/ i-RevNet	Real-NVP	Glow	FFJORD	i-ResNet
Free-form	✓	✗	✗	✗	✓	✓
Analytic Forward	✓	✓	✓	✓	✗	✓
Analytic Inverse	N/A	✓	✓	✗	✗	✗
Non-volume Preserving	N/A	✗	✓	✓	✓	✓
Exact Likelihood	N/A	✓	✓	✓	✗	✗
Unbiased Stochastic Log-Det Estimator	N/A	N/A	N/A	N/A	✓	✗

Table 1. Comparing i-ResNet and ResNets to NICE (Dinh et al., 2014), Real-NVP (Dinh et al., 2017), Glow (Kingma & Dhariwal, 2018) and FFJORD (Grathwohl et al., 2019). Non-volume preserving refers to the ability to allow for contraction and expansions and exact likelihood to compute the change of variables (3) exactly. The unbiased estimator refers to a stochastic approximation of the log-determinant, see section 3.2.

Invertible residual networks

Latent prior

$$p_{\boldsymbol{\theta}}(\mathbf{z}) = \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{z}_t | \mathbf{z}_{<t})$$

$$p(x_1, \ldots, x_n) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \cdots p(x_n \mid x_1, \cdots, x_{n-1})$$

VideoFlow: A Conditional Flow-Based Model for Stochastic Video Generation