Introduction to Normalizing Flow

by Sylvester Li

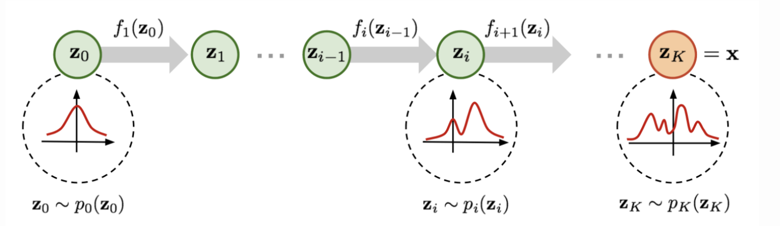
April 2020

Outline

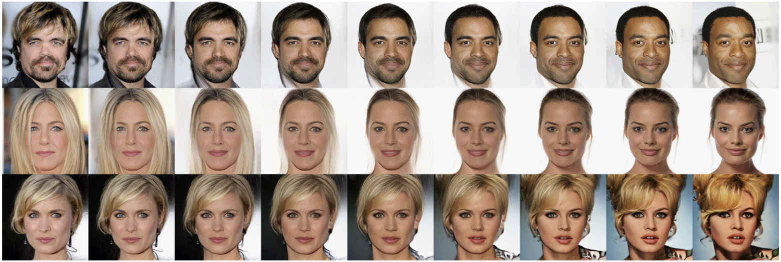
- Motivation
- Basics
  - Change of Varibles Formula
  - Triangular matrix
  - Autoregressive structure
- Models
  - NICE
  - Real NVP
  - NAF
  - IAF
  - Sum-of-Squares Polynomial Flow
  - Glow

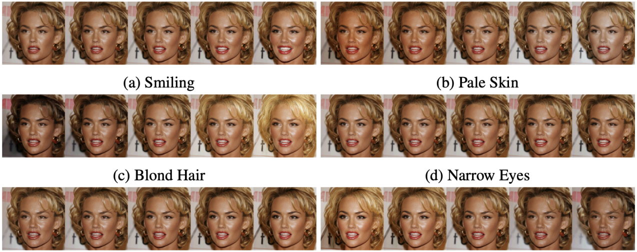
What is normalizing flow

Normizing flow is a transformation of a simple probability distribution into a more complex distribution by a sequence of invertable and differentiable mappings.



Motivation





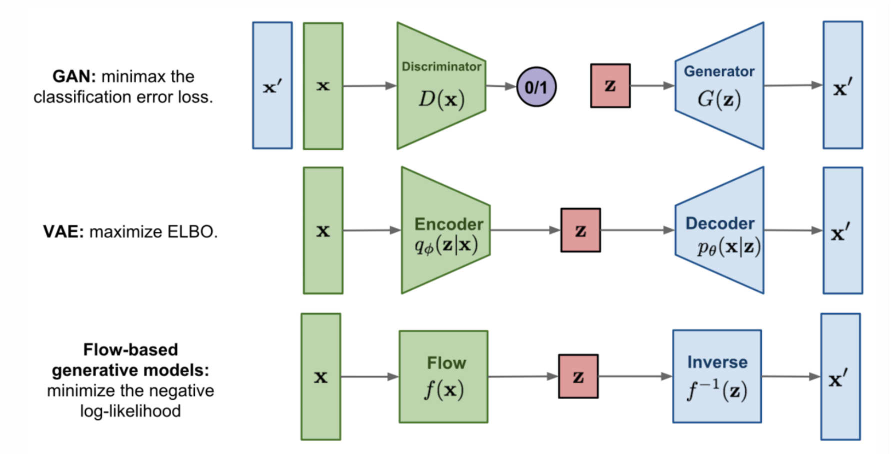
(a) Smiling (b) Pale Skin

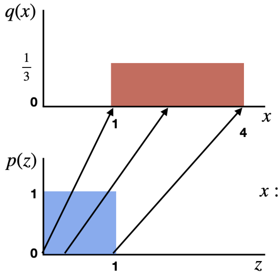
(c) Blond Hair (d) Narrow Eyes

(e) Young (f) Male

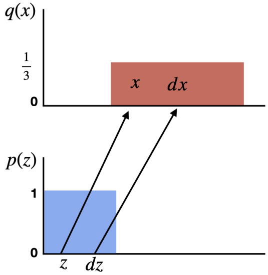
Motivation

Comparison between Normalizing flow and other generative methods





$x := T(z) = 3z + 1$



$$p(z)dz = q(x)dx$$

$$q(x) = p(z)\left|\frac{dz}{dx}\right|$$

$$q(x) = p(z)\left|\frac{dT(x)}{dz}\right|^{-1}$$

For one dimensional case, let $T : Z \mapsto X, x, z \in \mathbb{R}$, and we have

$$p_X(x) = p_Z(z)|\frac{dT(z)}{dz}|^{-1}$$

For D dimensional case, let $T : \boldsymbol{Z} \mapsto \boldsymbol{X}, \boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^D$, and we have

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = p_{\boldsymbol{Z}}(\boldsymbol{z})|det(\nabla_{\boldsymbol{Z}} T(\boldsymbol{z}))|^{-1} = p_{\boldsymbol{Z}}(\boldsymbol{z})|det(J(T(\boldsymbol{z}))^{-1})|$$

Where the $\nabla_{\boldsymbol{Z}} T(\boldsymbol{z})$ is the Jacobian matrix shown below:

$$\nabla_{\boldsymbol{Z}} T(\boldsymbol{z}) = J = \begin{bmatrix} \frac{\partial T_1}{\partial z_1} & \frac{\partial T_1}{\partial z_2} & \dots & \frac{\partial T_1}{\partial z_d} \\ \dots & \dots & \dots & \dots \\ \frac{\partial T_d}{\partial z_1} & \frac{\partial T_d}{\partial z_2} & \dots & \frac{\partial T_d}{\partial z_d} \end{bmatrix}$$

Take the log likelihood, we have

$$\log p_{\boldsymbol{X}}(\boldsymbol{x}) = \log p_{\boldsymbol{Z}}(\boldsymbol{z}) + \log \left| det(J(T(\boldsymbol{z}))^{-1}) \right|$$

Notice: if the Jacobian matrix is triangular, then we have

$$\begin{aligned}
\log p_{\boldsymbol{X}}(\boldsymbol{x}) &= \log p_{\boldsymbol{Z}}(\boldsymbol{z}) + \sum_{i=1}^{d} \log(\frac{\partial T}{\partial z_i})^{-1} \\
&= \log p_{\boldsymbol{Z}}(\boldsymbol{z}) - \sum_{i=1}^{d} \log(\frac{\partial T}{\partial z_i})
\end{aligned}$$

We want the Jacobian matrix to be

$$J = \begin{bmatrix} \frac{\partial T_1}{\partial z_1} & \frac{\partial T_1}{\partial z_2} & \dots & \frac{\partial T_1}{\partial z_d} \\ \dots & \dots & \dots & \dots \\ \frac{\partial T_d}{\partial z_1} & \frac{\partial T_d}{\partial z_2} & \dots & \frac{\partial T_d}{\partial z_d} \end{bmatrix} = \begin{bmatrix} \frac{\partial T_1}{\partial z_1} & 0 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ \frac{\partial T_d}{\partial z_1} & \frac{\partial T_d}{\partial z_2} & \dots & \frac{\partial T_d}{\partial z_d} \end{bmatrix}$$

Is this form generalizable? How do we construct such matrix?

By Bogachev 2005, they proved "From any two density p,q over $Z = X = \mathbb{R}^d$, there exists a unique trangular map $T : Z \mapsto X$ such that q=Tp."

If we construct a structure as below,

$$
\begin{aligned}
x_1 &= T_1(z_1) \\
x_2 &= T_2(z_1, z_2) \\
&\dots \\
x_d &= T_d(z_1, z_2, \dots, z_d))
\end{aligned}
$$

The structure is known as autoregressive structure. We have

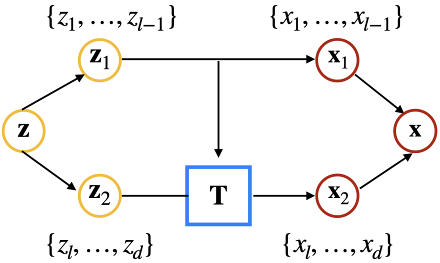
$$\frac{\partial T_i}{\partial z_j} = 0, j > i$$

Basics

Autoregressive structure

$$
\begin{aligned}
x_1 &= T_1(z_1) && \sim p_{x_1}(x_1) \\
x_2 &= T_2(z_1, z_2) = T_2(T_1^{-1}(x_1), z_2) && \sim p_{x_2}(x_2|x_1) \\
&\dots && \dots \\
x_d &= T_d(z_1, z_2, \dots, z_d) = T_d(f(x_{<d}), z_d) && \sim p_{x_d}(x_d|x_{<d})
\end{aligned}
$$

where f is some function of $x_1, x_2, x_3, \dots, x_{d-1}$

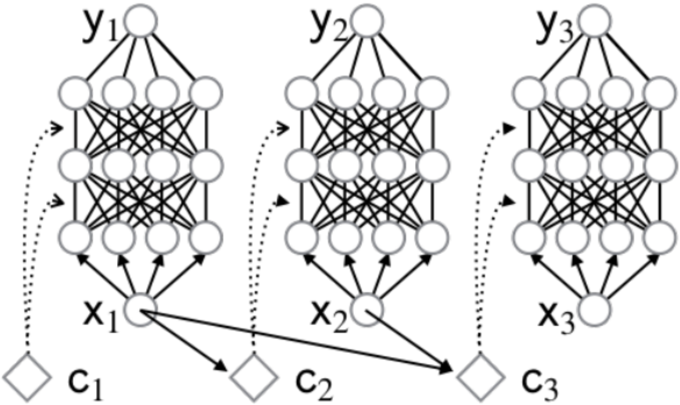


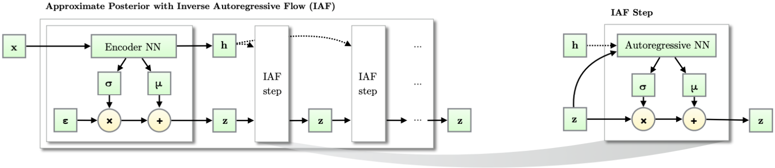
$$\begin{cases} \mathbf{y}_{1:d} &= \mathbf{x}_{1:d} \\ \mathbf{y}_{d+1:D} &= \mathbf{x}_{d+1:D} + m(\mathbf{x}_{1:d}) \end{cases} \Leftrightarrow \begin{cases} \mathbf{x}_{1:d} &= \mathbf{y}_{1:d} \\ \mathbf{x}_{d+1:D} &= \mathbf{y}_{d+1:D} - m(\mathbf{y}_{1:d}) \end{cases}$$

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

Models

NAF: Neural Autoregressive Flows

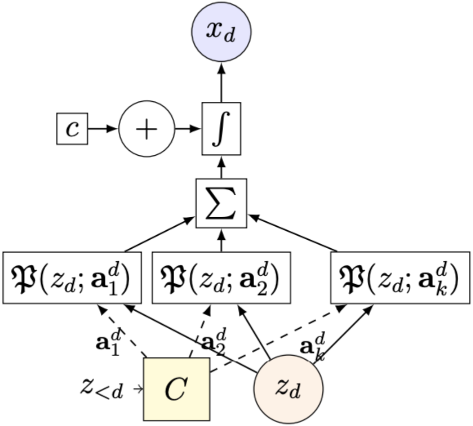


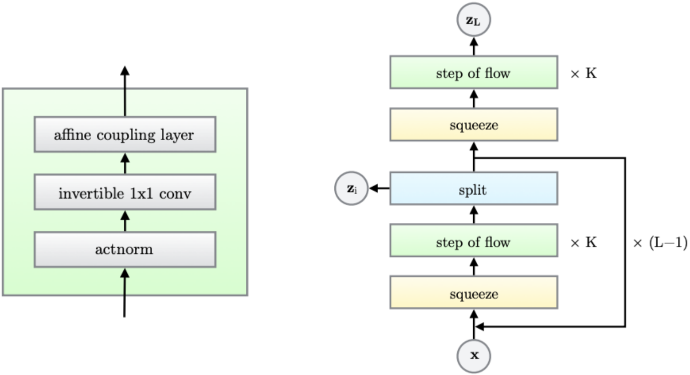


Approximate Posterior with Inverse Autoregressive Flow (IAF)
x
Encoder NN
h
σ
μ
ε
×
+
z
IAF step
IAF step
z
...
z
IAF Step
h
Autoregressive NN
σ
μ
z
×
+
z

Models

Sum-of-Squares Polynomial Flow





(a) One step of our flow.

(b) Multi-scale architecture (Dinh et al., 2016).

Reference and Further Reading

Overview of normalizing flow:
Normalizing Flows Tutorial Part 1 & 2
Papers:
NICE: Non-linear Independent Components Estimation
Density estimation using Real NVP
Neural Autoregressive Flows
Improving Variational Inference with Inverse Autoregressive Flow
Sum-of-Squares Polynomial Flow
Glow: Generative Flow with Invertible 1x1 Convolutions
Normalizing Flows: An Introduction and Review of Current Methods
Variational Inference with Normalizing Flows
Masked Autoregressive Flow for Density Estimation