Normalising Flow Models (Part 1)

Hao Dong

Peking University

Normalising Flow Models (Part 1)

Hao Dong

Peking University

Where we are?

- Autoregressive Models
  $p(x_1, x_2, \dots, x_n) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \dots p(x_n|x_1, \dots, x_{n-1})$
  - Provide tractable likelihoods
  - No direct mechanism for learning features
  - Slow generation – Wavenet: 1 second audio takes 90 mins (200K samples)

- Variational Autoencoders
  $p(X) = \sum_Z p(X|Z)p(Z)$ or $p(X) = \int_Z p(X|Z)p(Z)dZ$
  - Can learn feature representations (via latent variables Z)
  - Have intractable marginal likelihoods.
  - Optimising a lower bound – it is not maximising the likelihood … we don't know the gap.

Question: Can we design a latent variable model with tractable likelihoods?
Yes! We can use normalising flow models. (Today)

Reference slides

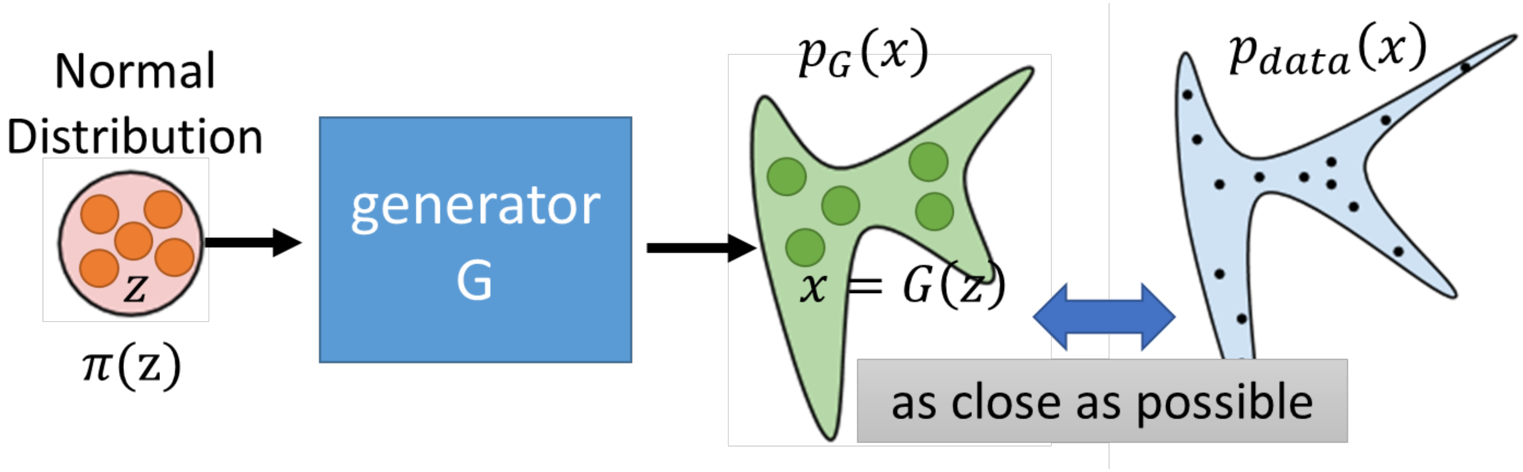
- Hung-yi Li. Flow-based Generative Model
- Stanford “Deep Generative Models”. Normalising Flow Models

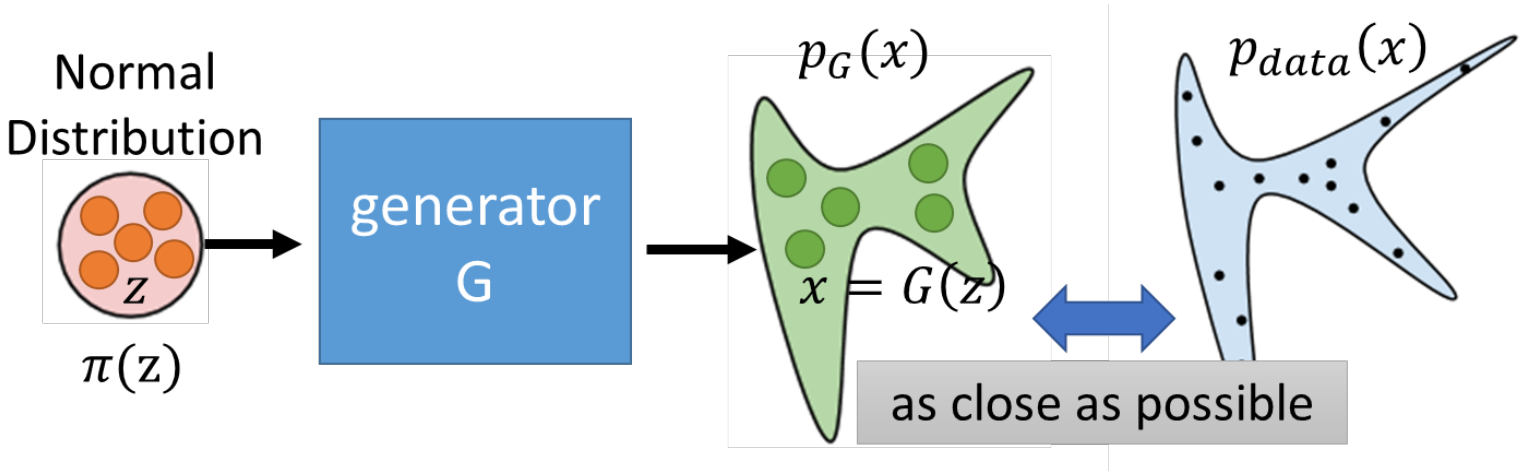
- Background
  - Generator
  - Change of variable theorem (1D)
  - Jacobian Matrix & Determinant
  - Change of variable theorem
- Normalising Flow
  - Flow-based model
  - Learning and inference
  - Desiderata

- Background
  - Generator
  - Change of variable theorem (1D)
  - Jacobian Matrix & Determinant
  - Change of variable theorem
- Normalising Flow
  - Flow-based model
  - Learning and inference
  - Desiderata

Generator

- A generator G is a network. The network maps a simple distribution (for example, normal distribution) $\pi(z)$ to a complex data distribution $p_G(x)$, which aims to be as close to real data distribution $p_{data}(x)$ as possible.





- $G^* = arg \max_G \sum_{i=1}^{m} log P_G(x^i)$
- Normalising flow models directly optimise the objective function!
- **Key idea:** Map simple distributions (easy to sample and evaluate densities) to complex distributions (learned via data) using **change of variables.**

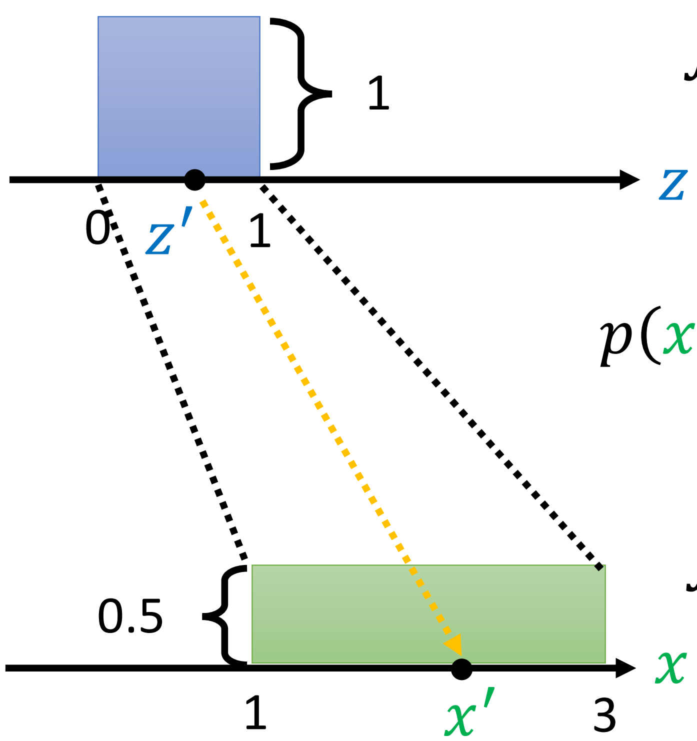
- Background
  - Generator
  - Change of variable theorem (1D)
  - Jacobian Matrix & Determinant
  - Change of variable theorem
- Normalising Flow
  - Flow-based model
  - Learning and inference
  - Desiderata

Change of Variable Theorem (1D)

- Let Z be a uniform random variable $U[0,1]$ with density π_Z. What is $\pi_Z(1/2)$?
  - 1
- Let $X = f(Z) = 2Z + 1$ and let p_X be its density. What is $p_X(2)$?
  - When Z = 1/2, X = 2Z + 1 = 2, so does $p_X(2) = \pi_Z\left(\frac{1}{2}\right) = 1$?
  - No
- Clearly, X is uniform in [1,3], so $p_X(2) = 1/2$

Change of Variable Theorem (1D)

$\pi(z)$



$p(x)$

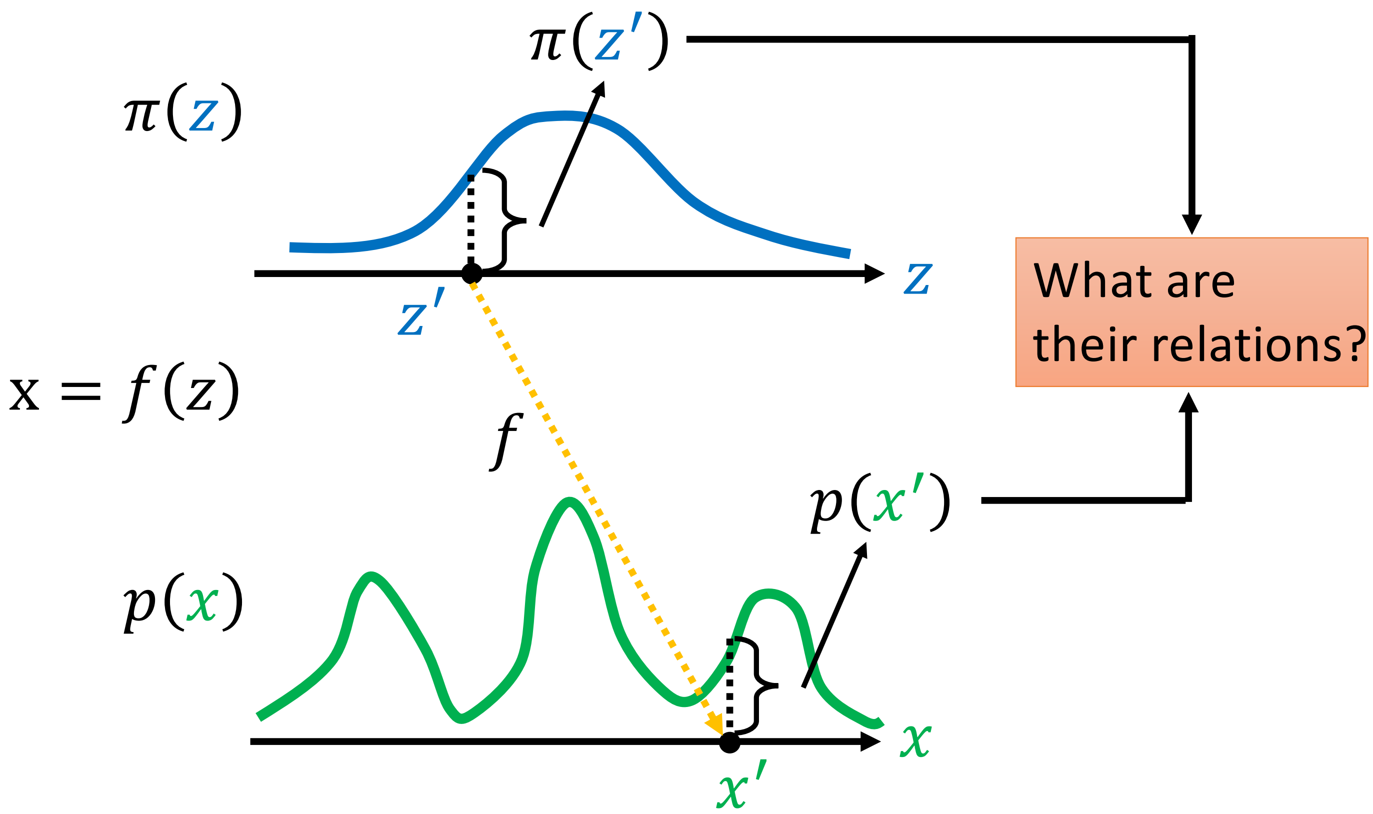
$$\int \pi(z)dz = 1$$

$$\mathrm{x} = f(z)$$
$$= 2z + 1$$

$$p(x') = \frac{1}{2}\pi(z')$$

$$\int p(x)dx = 1$$

Change of Variable Theorem (1D)



Change of Variable Theorem (1D)

When $x = f(z)$ and function f is **invertible** and **differentiable**.

If f is monotonically increasing, we have $Pr(z' \le z \le (z' + \Delta z)) = Pr\big(f(z') \le f(z) \le f(z' + \Delta z)\big) = Pr(x' \le x \le (x' + \Delta \mathrm{x})\ )$

If f is monotonically decreasing, we can get the same result.

So we get

$$\left| \int_{z'}^{z'+\Delta z} \pi(z)\, dz \right| = \left| \int_{x'}^{x'+\Delta x} p(x)\, dx \right|$$

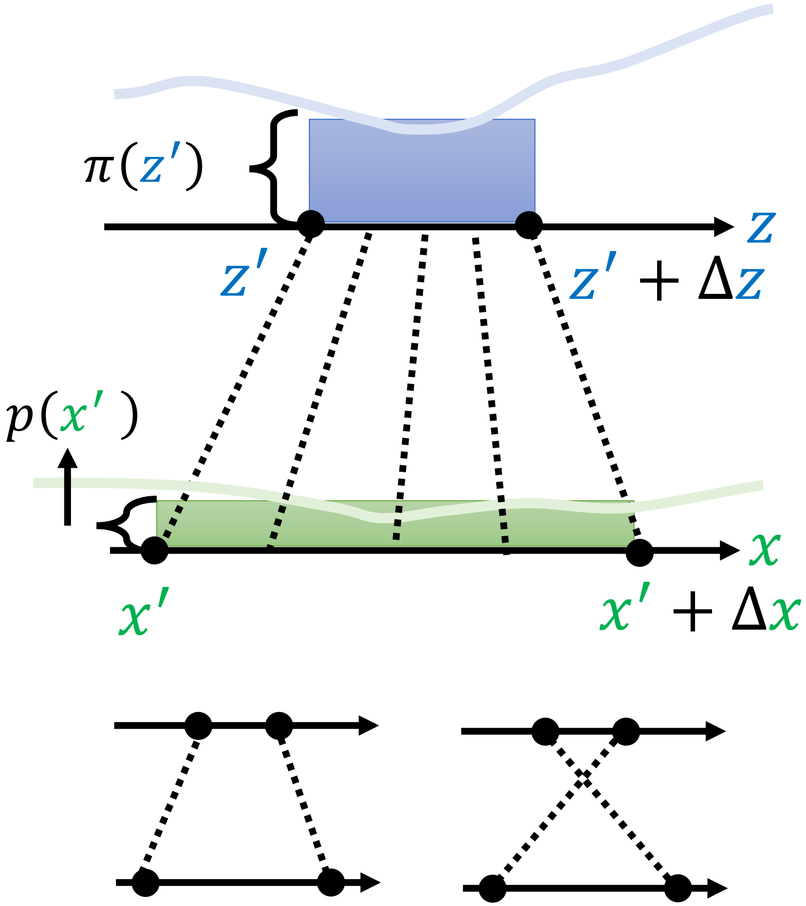
Change of Variable Theorem (1D)

- $\left|\int_{z'}^{z'+\Delta z} \pi(z)\, dz\right| = \left|\int_{x'}^{x'+\Delta x} p(x)\, dx\right|$

- Use laGrange's Mean Value Theorem, we get $\pi(\tilde{z})|\Delta z| = p(\tilde{x})|\Delta \mathrm{x}|$, where $z' \leq \tilde{z} \leq z' + \Delta z, x' \leq \tilde{x} \leq x' + \Delta x$

- When $\Delta z \to 0$, we have $p(x') = \pi(z') \left|\frac{\Delta z}{\Delta x}\right|_{x=x'} = \pi(z') \left|\frac{\partial z}{\partial x}\right|_{x=x'}$

Change of Variable Theorem (1D)



The blue square and the green square should be equal in area

$$p(x')|\Delta x| = \pi(z')|\Delta z|$$

$$p(x') = \pi(z')\left|\frac{\partial z}{\partial x}\right|$$

Change of Variable Theorem (1D)

- **change of variable theorem (1-D case):** if $x = f(z)$ and function f is invertible and differentiable, then $\boxed{p(x) = \pi(z) \left|\frac{\partial z}{\partial x}\right| = \pi(z) \left|\frac{\partial f^{-1}(x)}{\partial x}\right|}$

- How about multi-dimension cases?
  - We need more math background.

- Background
  - Generator
  - Change of variable theorem (1D)
  - Jacobian Matrix & Determinant
  - Change of variable theorem
- Normalising Flow
  - Flow-based model
  - Learning and inference
  - Desiderata

Jacobian Matrix (2D case)

1)

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$x = f(z) \quad z = f^{-1}(x)$$

2)

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} z_1 + z_2 \\ 2z_1 \end{bmatrix} = f\left(\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}\right)$$

$$\begin{bmatrix} x_2/2 \\ x_1 - x_2/2 \end{bmatrix} = f^{-1}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right)$$

3)

$$J_f = \begin{array}{c} \text{input} \\ \begin{bmatrix} \partial x_1/\partial z_1 & \partial x_1/\partial z_2 \\ \partial x_2/\partial z_1 & \partial x_2/\partial z_2 \end{bmatrix} \end{array} \ \text{output}$$

$$J_{f^{-1}} = \begin{bmatrix} \partial z_1/\partial x_1 & \partial z_1/\partial x_2 \\ \partial z_2/\partial x_1 & \partial z_2/\partial x_2 \end{bmatrix}$$

4)

$$J_f = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}$$

$$J_{f^{-1}} = \begin{bmatrix} 0 & 1/2 \\ 1 & -1/2 \end{bmatrix}$$

$$J_f J_{f^{-1}} = I$$

Determinant

The determinant of a **square matrix** is a **scalar** that provides information about the matrix.

- 2 X 2

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$det(A) = ad - bc$$

- 3 x 3

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix}$$

$$det(A) =$$

$$a_1a_5a_9 + a_2a_6a_7 + a_3a_4a_8$$

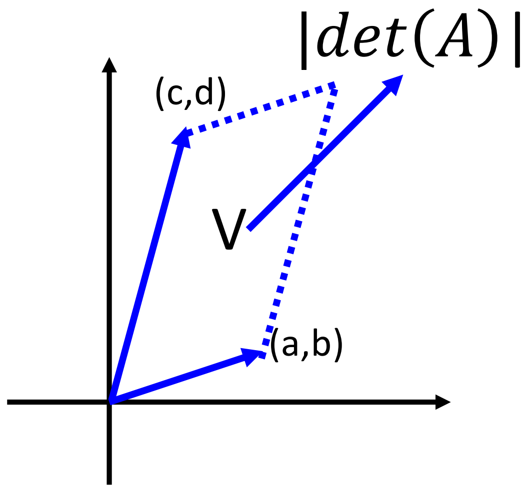
$$-a_3a_5a_7 - a_2a_4a_9 - a_1a_6a_8$$

$$\boxed{\begin{aligned} det(\mathrm{A}) &= 1/det(A^{-1}) \\ det(J_f) &= 1/det(J_{f^{-1}}) \end{aligned}}$$

Determinant

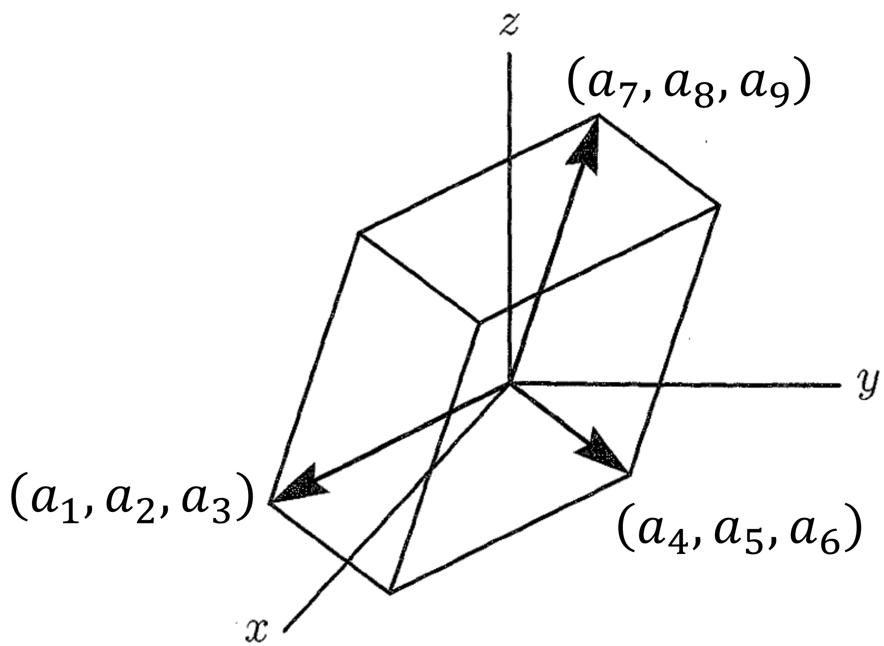
- 2 X 2

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$



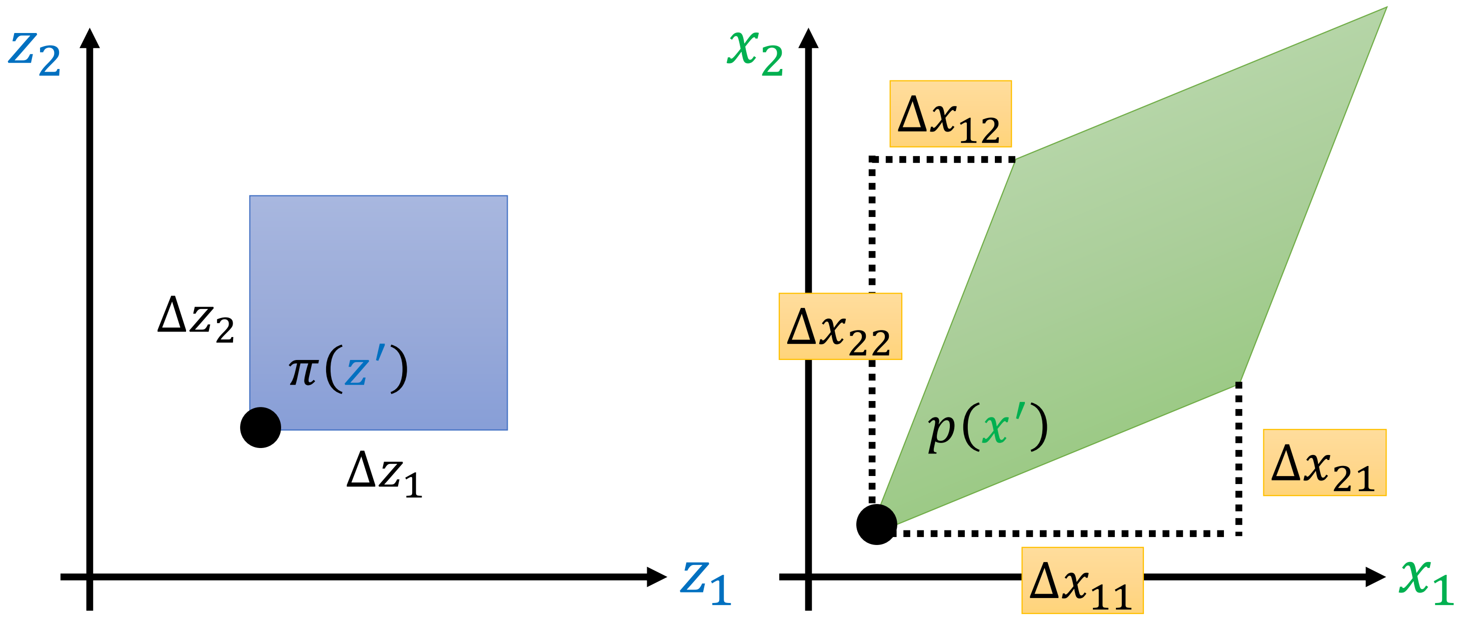
- 3 x 3

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix}$$



- Background
  - Generator
  - Change of variable theorem (1D)
  - Jacobian Matrix & Determinant
  - Change of variable theorem
- Normalising Flow
  - Flow-based model
  - Learning and inference
  - Desiderata

Change of Variable Theorem (2D case)



$$p(x') \left| det \begin{bmatrix} \Delta x_{11} & \Delta x_{21} \\ \Delta x_{12} & \Delta x_{22} \end{bmatrix} \right| = \pi(z') \Delta z_1 \Delta z_2$$

$$p(x')\left|det\begin{bmatrix}\Delta x_{11} & \Delta x_{21}\\ \Delta x_{12} & \Delta x_{22}\end{bmatrix}\right| = \pi(z')\Delta z_1 \Delta z_2 \qquad \mathrm{x} = f(z)$$

$$p(x')\left|\frac{1}{\Delta z_1 \Delta z_2}det\begin{bmatrix}\Delta x_{11} & \Delta x_{21}\\ \Delta x_{12} & \Delta x_{22}\end{bmatrix}\right| = \pi(z')$$

$$p(x')\left|det\begin{bmatrix}\Delta x_{11}/\Delta z_1 & \Delta x_{21}/\Delta z_1\\ \Delta x_{12}/\Delta z_2 & \Delta x_{22}/\Delta z_2\end{bmatrix}\right| = \pi(z')$$

$$p(x')\left|det\begin{bmatrix}\partial x_1/\partial z_1 & \partial x_2/\partial z_1\\ \partial x_1/\partial z_2 & \partial x_2/\partial z_2\end{bmatrix}\right| = \pi(z')$$

$$p(x')\left|det\begin{bmatrix}\partial x_1/\partial z_1 & \partial x_1/\partial z_2\\ \partial x_2/\partial z_1 & \partial x_2/\partial z_2\end{bmatrix}\right| = \pi(z') \quad \text{(transpose)}$$

$$\boxed{\begin{array}{ll} p(x')\left|det(J_f)\right| = \pi(z') & \multirow{2}{*}{$p(x') = \pi(z')\left|\dfrac{1}{det(J_f)}\right|$} \\ p(x') = \pi(z')\left|det(J_{f^{-1}})\right| & \end{array}}$$

北京大學 PEKING UNIVERSITY

Change of Variable Theorem (General case)

- **Change of Variable Theorem (General case):** if the mapping function between Z and X, given by $f: R^n \to R^n$, is differentiable and invertible such that $X = f^{-1}(Z)$ and $Z = f(X)$, then

$$p(\boldsymbol{x}) = \pi(\boldsymbol{z}) \left| \det(\frac{\partial f^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}}) \right| = \pi(\boldsymbol{z}) \left| det\left(J_{f^{-1}}\right) \right|$$

- Note 1: **x** and **z** need to be continuous and have the same dimension
- Note 2: since for any invertible matrix A, $\det(A^{-1}) = \det(A)^{-1}$

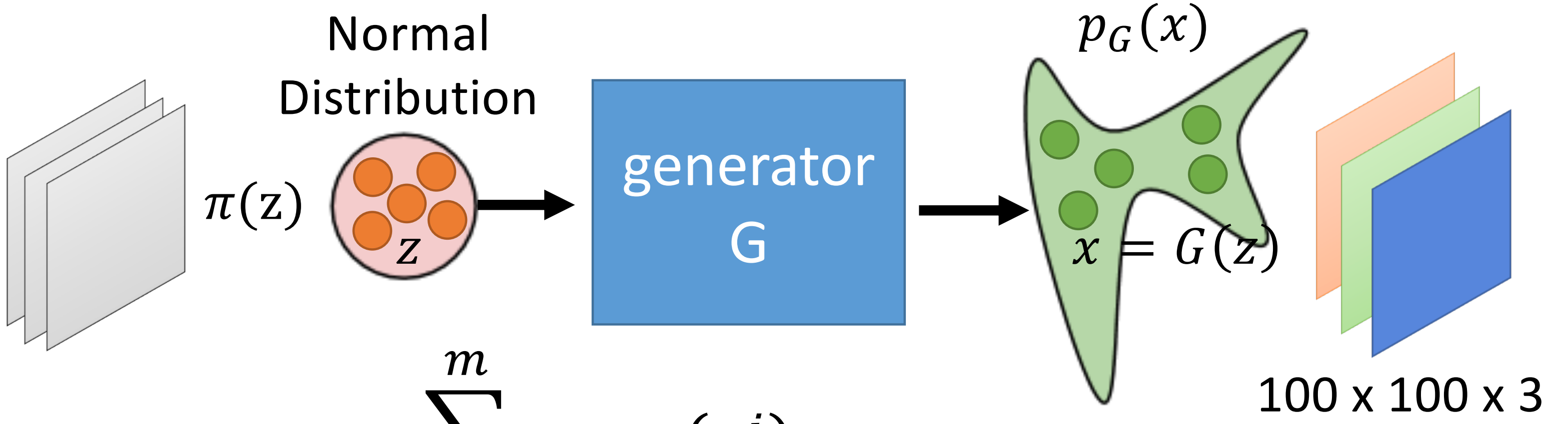
$$\boxed{p(\boldsymbol{x}) = \pi(\boldsymbol{z}) \left| \frac{1}{det\left(J_f\right)} \right|}$$

- Background
  - Generator
  - Change of variable theorem (1D)
  - Jacobian Matrix & Determinant
  - Change of variable theorem
- Normalising Flow
  - Flow-based model
  - Learning and inference
  - Desiderata

Flow-based Model

$$p(x')\left|det(J_f)\right| = \pi(z')$$

$$p(x') = \pi(z')\left|det(J_{f^{-1}})\right|$$



$$G^* = arg \max_G \sum_{i=1}^{m} log p_G(x^i)$$

$$p_G(x^i) = \pi(z^i)\left|det(J_{G^{-1}})\right|$$

$$z^i = G^{-1}(x^i)$$

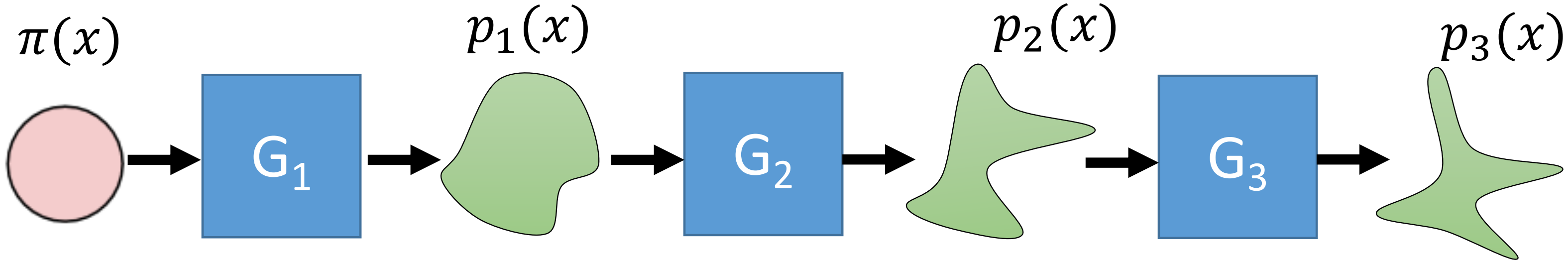
You can compute $det(J_G)$

You know G^{-1}

G has limitation

$$\boxed{log p_G(x^i) = log\pi\left(G^{-1}(x^i)\right) + log\left|det(J_{G^{-1}})\right|}$$

G is limited. We need more generators



$$p_1(x^i) = \pi(z^i)\left(\left|det\left(J_{G_1^{-1}}\right)\right|\right) \qquad z^i = G_1^{-1}\left(\cdots G_K^{-1}(x^i)\right)$$

$$p_2(x^i) = \pi(z^i)\left(\left|det\left(J_{G_1^{-1}}\right)\right|\right)\left(\left|det\left(J_{G_2^{-1}}\right)\right|\right)$$

$$\vdots$$

$$p_K(x^i) = \pi(z^i)\left(\left|det\left(J_{G_1^{-1}}\right)\right|\right)\cdots\left(\left|det\left(J_{G_K^{-1}}\right)\right|\right)$$

$$log p_K(x^i) = log\pi(z^i) + \sum_{h=1}^{K} log\left|det\left(J_{G_K^{-1}}\right)\right|$$ Maximise

What you actually do?

$$z \rightarrow \boxed{G} \rightarrow x$$

$$\log p_G(x^i) = \underbrace{\log \pi\left(G^{-1}(x^i)\right)} + \underbrace{\log|det(J_{G^{-1}})|}$$

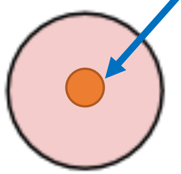
If z is zero, this term will be -inf

This term: make z^i become zero vector

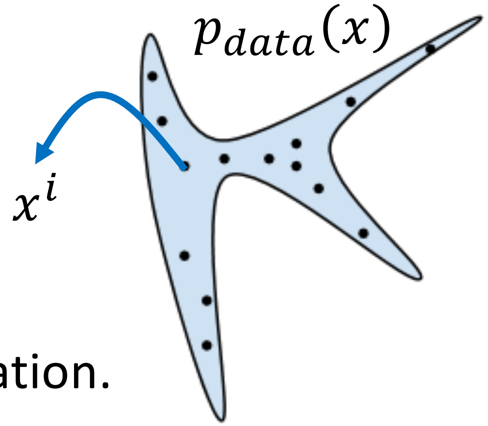
If z^i is always zero:

$J_{G^{-1}}$ would be zero matrix

$det(J_{G^{-1}}) = 0$



$$z^i = G^{-1}(x^i) \leftarrow \boxed{G^{-1}} \leftarrow x^i$$



Actually, we train G^{-1}, but we use G for generation.

北京大學 PEKING UNIVERSITY

- Background
  - Generator
  - Change of variable theorem (1D)
  - Jacobian Matrix & Determinant
  - Change of variable theorem
- Normalising Flow
  - Flow-based model
  - Learning and inference
  - Desiderata

Learning and inference

- Learning via **maximum likelihood** over the dataset D

$$\max_{\theta} logp(D;\theta) = \sum_{x\in D} log\pi\left(G_\theta^{-1}(x)\right) + log\left|det\left(\frac{\partial G_\theta^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)\right|$$

- **1） Exact likelihood evaluation** via inverse transformation and change of variables formula
- **2） Sampling** via forward transformation $G_\theta: Z \to X$

$$z \sim \pi(z), x = G_\theta(z)$$

- **3） Latent representations** inferred via inverse transformation (no inference network required!)

$$z = G_\theta^{-1}(x)$$

Normalising Flow

- “Normalising” means that the change of variables gives a normalised density after applying an invertible transformation.
- “Flow” means that the invertible transformations can be composed with each other to create more complex invertible transformations.

- Background
  - Generator
  - Change of variable theorem (1D)
  - Jacobian Matrix & Determinant
  - Change of variable theorem
- Normalising Flow
  - Flow-based model
  - Learning and inference
  - Desiderata

Desiderata for flow models

- Simple prior $\pi(z)$ that allows for efficient sampling and tractable likelihood evaluation. E.g., Gaussian
- Invertible transformations
- Computing likelihoods also requires the evaluation of determinants of $n \times n$ Jacobian matrices, where n is the data dimensionality
  - Computing the determinant for an $n \times n$ matrix is $O(n^3)$: prohibitively expensive within a learning loop!
  - **Key idea**: Choose transformations so that the resulting Jacobian matrix has special structure. For example, the determinant of a triangular matrix is the product of the diagonal entries, i.e., an $O(n)$ operation

Triangular Jacobian Matrix

$$\mathbf{x} = (x_1, \cdots, x_n) = \mathbf{f}(\mathbf{z}) = (f_1(\mathbf{z}), \cdots, f_n(\mathbf{z}))$$

$$J = \frac{\partial \mathbf{f}}{\partial \mathbf{z}} = \begin{pmatrix} \frac{\partial f_1}{\partial z_1} & \cdots & \frac{\partial f_1}{\partial z_n} \\ \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial z_1} & \cdots & \frac{\partial f_n}{\partial z_n} \end{pmatrix}$$

Suppose $x_i = f_i(\mathbf{z})$ only depends on $\mathbf{z}_{\leq i}$. Then

$$J = \frac{\partial \mathbf{f}}{\partial \mathbf{z}} = \begin{pmatrix} \frac{\partial f_1}{\partial z_1} & \cdots & 0 \\ \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial z_1} & \cdots & \frac{\partial f_n}{\partial z_n} \end{pmatrix}$$

has lower triangular structure. Determinant can be computed in **linear time**. Similarly, the Jacobian is upper triangular if x_i only depends on $\mathbf{z}_{\geq i}$

Thanks