Aprendizado Profundo

Normalizing Flows

Professor: Lucas Silveira Kupssinskü

Agenda

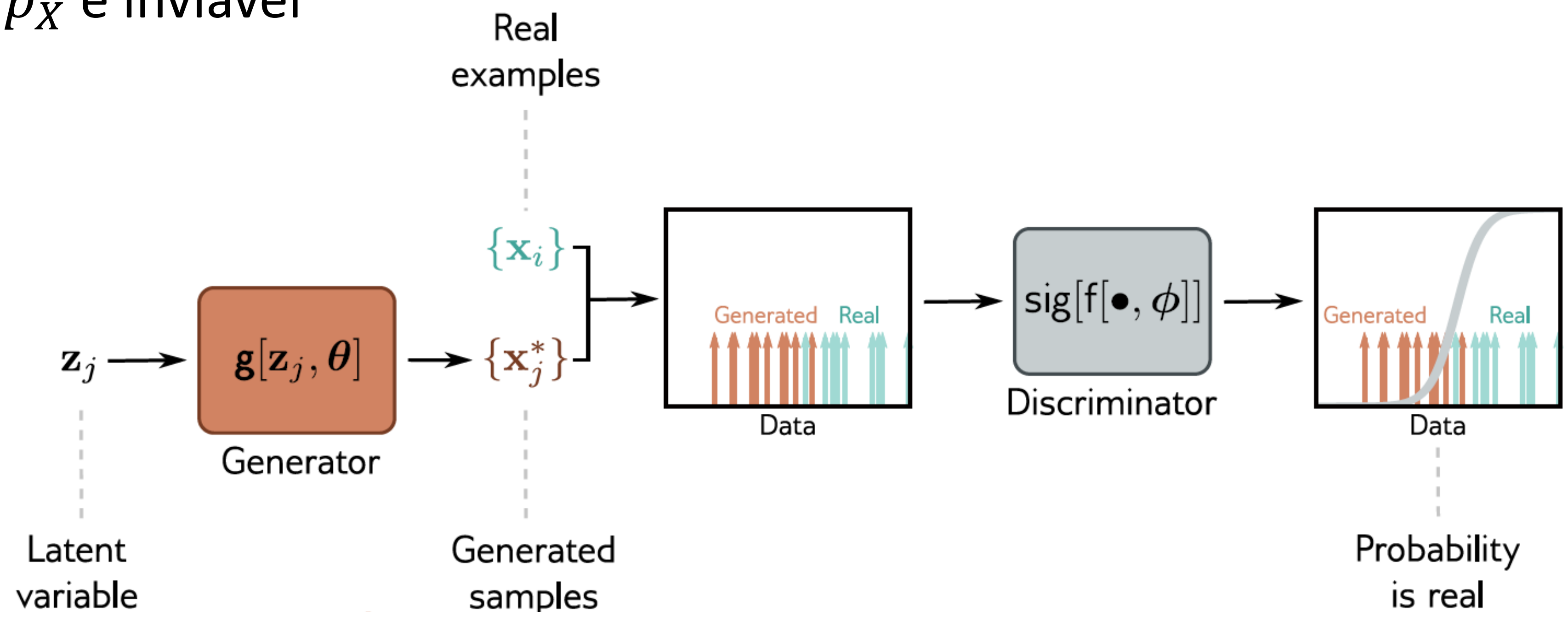
- Modelos Geradores – revisão
- Visão geral de *Normalizing Flows*
- Mudança de Variável em distribuições de probabilidade
- Loss
- O que são *Flows*
    - Composição de *Flows*
    - *Linear Flows*
    - *Coupling Flows*
    - *Afine Coupling Flows*

Modelos Geradores

- O objetivo de um modelo gerador é aprender uma distribuição de probabilidade sobre uma variável aleatória $\boldsymbol{X}$ usando um conjunto de dados observados $\{x^{(i)}\}_{i=1}^{N}$ com uma densidade de probabilidade $p_X(x)$ parametrizada por θ
- Repare que, podemos ter interesse tanto em amostrar quanto em avaliar $p_X(x)$

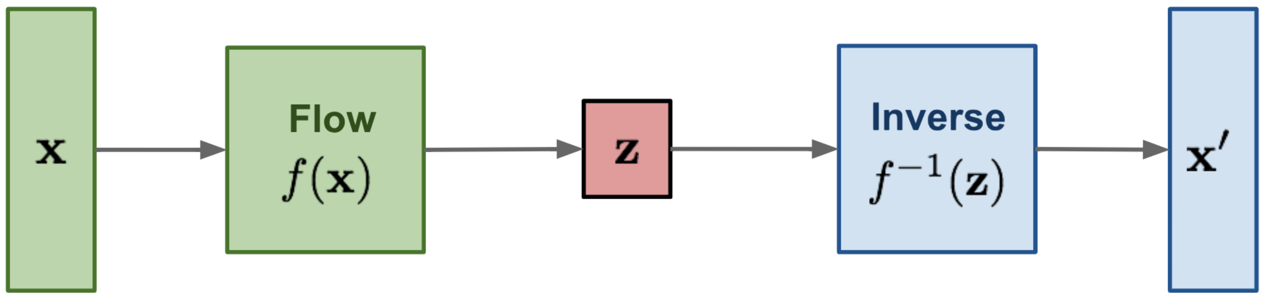
Modelos Geradores

- Vamos olhar novamente para GANs
  - Conseguimos amostrar p_X
  - Avaliar o valor de p_X é inviável

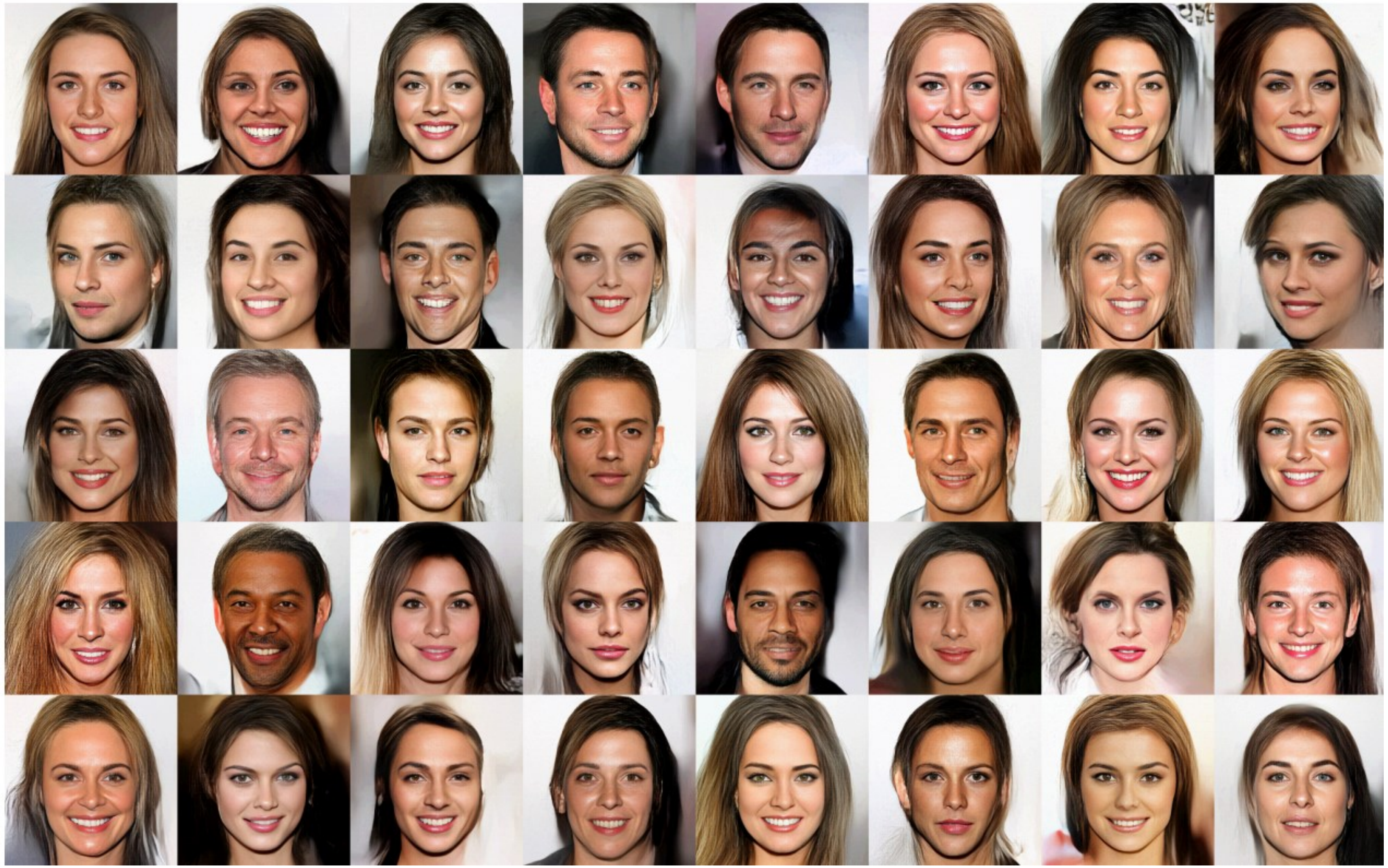


Modelos Geradores

- *Normalizing Flows* vão são modelos geradores probabilísticos que permitem tanto amostrar quanto avaliar $p_X(x)$
- Consistem em transformar uma distribuição tratável base (e.g. $\mathrm{z} \sim \mathcal{N}(0,1)$) em uma outra distribuição usando funções inversíveis



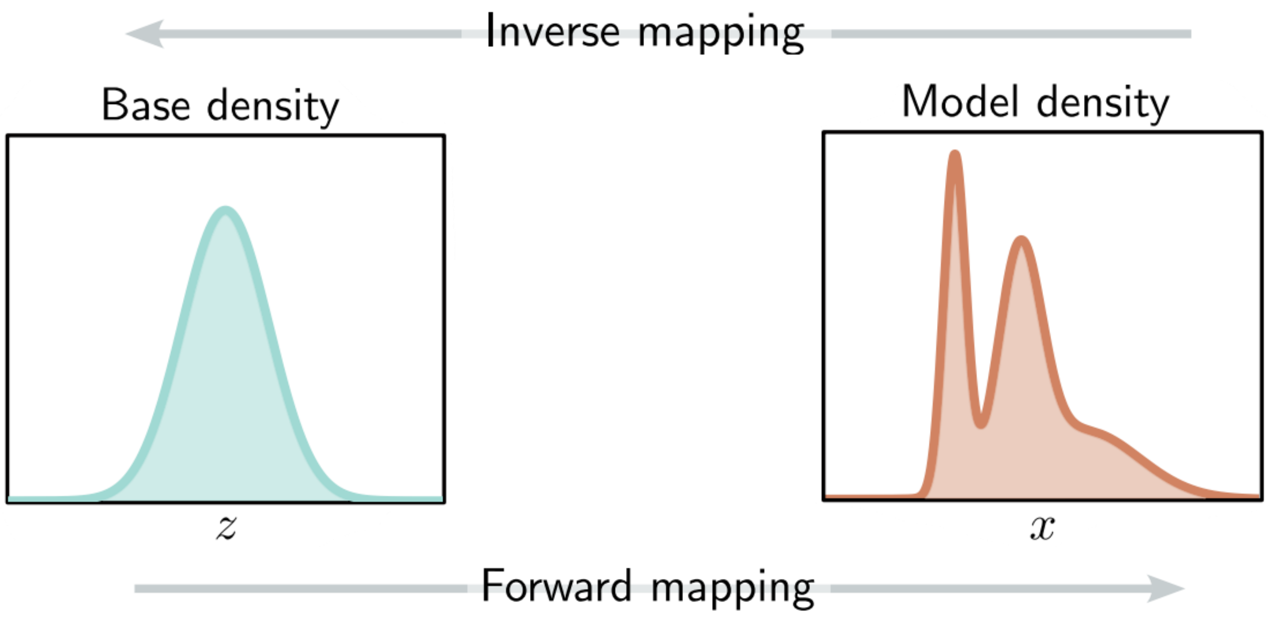
Modelos Geradores



KINGMA, Durk P.; DHARIWAL, Prafulla. Glow: Generative flow with invertible 1x1 convolutions. **Advances in neural information processing systems**, v. 31, 2018.

Modelos Geradores

- Abaixo um exemplo de transformação de densidades de probabilidades 1D

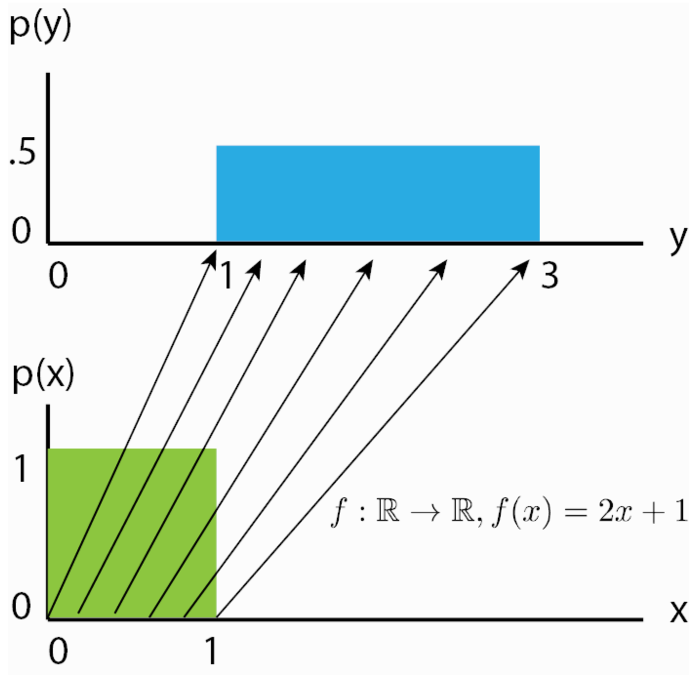


Mudança de Variável

- Caso discreto
  - Realizar mudança de variável no caso discreto é trivial
  - Vamos imaginar o lançamento de uma moeda
    - $p_X(x = 1) = \frac{1}{6}$
  - Se aplicarmos uma função inversível, por exemplo $f(x) = x^2$, o espaço amostral do nosso experimento muda
    - $\{1,2,3,4,5,6\} \rightarrow \{1,4,9,16,25,36\}$
  - Logo, para calcularmos $p_Z(z = 9)$ temos que perceber que $z = 9 \equiv x = 3$, por consequência as probabilidades são iguais

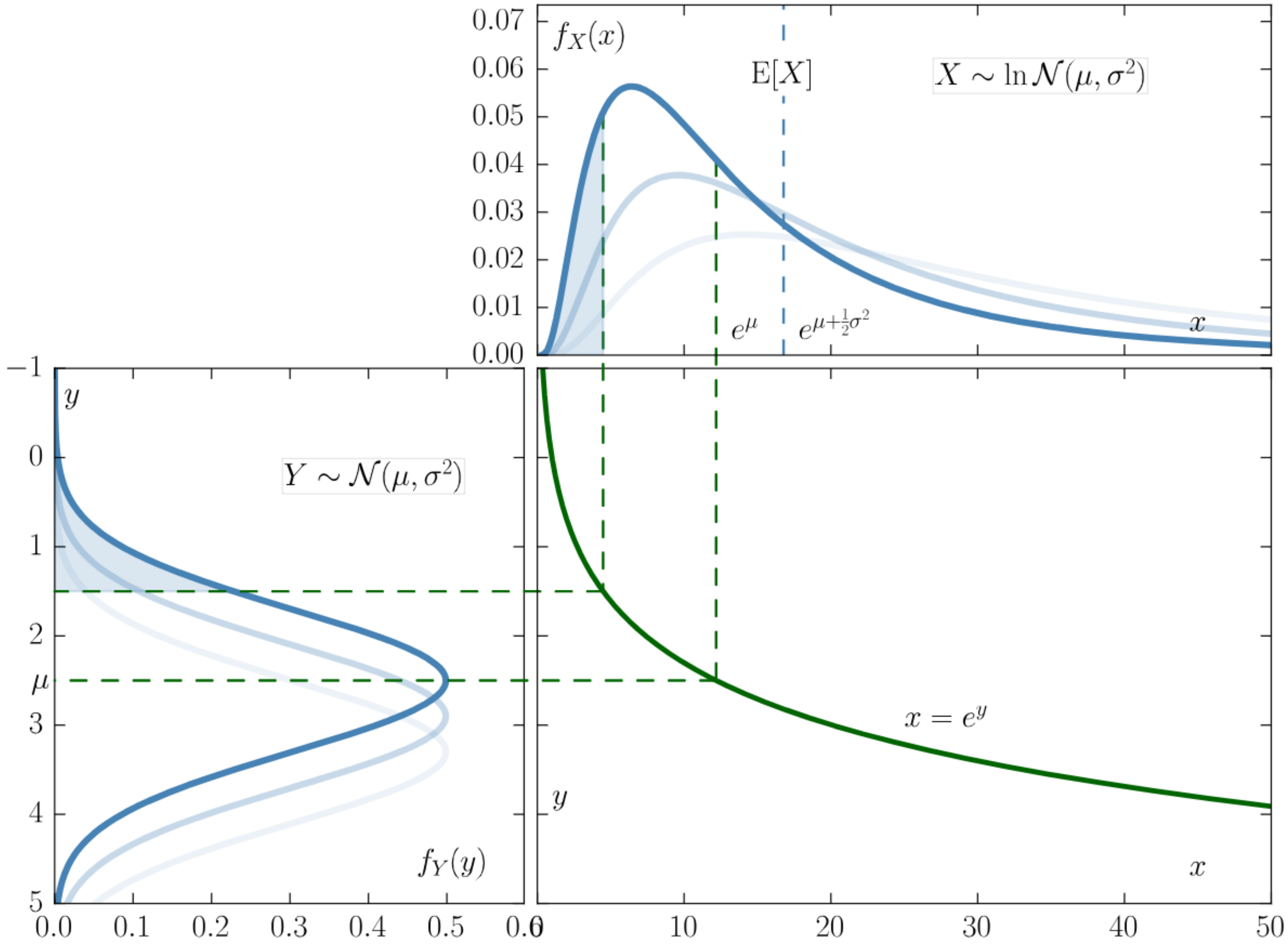
Mudança de Variável

- Caso contínuo
  - Aqui nossa noção fundamental passa a ser um intervalo de probabilidades, não um evento pontual
  - Ao aplicar essa função (inversível e derivável) a densidade sofre uma deformação



Mudança de Variável

- Repare as duas distribuições ao lado
- Perceba que, um determinado volume de uma das distribuições pode ser achatado ou alongado ao efetuar a transformação



Mudança de Variável

- $X \sim \mathcal{N}(0,1)$
- $p_X(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$
- $z = \ln(x)$
- $p_Z(z) = ?$

Mudança de Variável

- $X \sim \mathcal{N}(0,1)$
- $p_X(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$
- $z = \exp(x)$
- $p_Z(z) = ?$
- $x = \ln(z)$
- $P(Z \leq z) = P(X \leq \ln(z))$
  - $CDF(z) = CDF(\ln(z))$
  - $p_Z(z) = p_X(\ln(z)) \frac{\partial(\ln x)}{\partial dz}$
  - $p_Z(z) = \frac{1}{z\sqrt{2\pi}} \exp\left(-\frac{\ln(z)^2}{2}\right)$

Mudança de Variável

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

Mudança de Variável

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

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

O módulo é colocado para garantir que o resultado seja positivo e que não tenhamos que nos preocupar com direções de integrações e derivadas ☺

Loss

- Para realizar o treinamento do nosso modelo, seguimos usando o framework de maximizar a verossimilhança dos dados

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \left[\prod_{i=1}^{N} P\left(x^{(i)} | \theta\right) \right]$$

Loss

- Para realizar o treinamento do nosso modelo, seguimos usando o framework de maximizar a verossimilhança dos dados

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \left[\prod_{i=1}^{N} P\left(x^{(i)} | \theta\right) \right]$$

$$= \underset{\theta}{\operatorname{argm}in} \left[\sum_{i=1}^{N} -\log\left(P\left(x^{(i)} | \theta\right)\right) \right]$$

$$= \underset{\theta}{\operatorname{argm}in} \left[\sum_{i=1}^{N} -\log\left(P\left(z^{(i)}\right) \left| \frac{\partial f\left(z^{(i)}, \theta\right)}{\partial z^{(i)}} \right|^{-1} \right) \right]$$

$$= \underset{\theta}{\operatorname{argm}in} \left[\sum_{i=1}^{N} \left[\log \left| \frac{\partial f\left(z^{(i)}, \theta\right)}{\partial z^{(i)}} \right| - \log P\left(z^{(i)}\right) \right] \right]$$

Flows

- Um flow é uma função $f(x)$
  - Paramétrica (queremos definir parâmetros)
  - Inversível (queremos calcular o flow nos dois sentidos)
  - Diferenciável (queremos otimizar via backprop)
  - Conseguimos computar a inversa e o determinante da Jacobiana de maneira eficiente

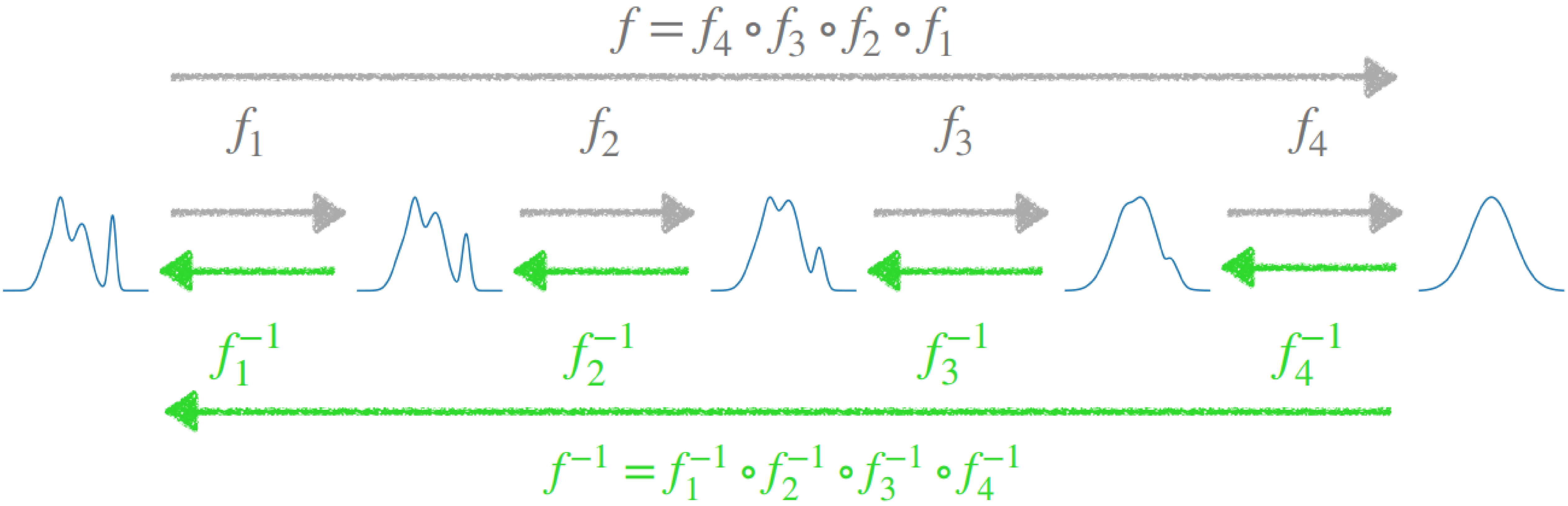
Flows

- A classe de funções diferenciáveis e inversíveis é fechada em composição

$$f = f_k \circ f_{k-1} \circ \cdots \circ f_2 \circ f_1$$

- Isso nos possibilita projetar camadas que computam um flow e criar redes com múltiplas dessas camadas

Flows


$f = f_4 \circ f_3 \circ f_2 \circ f_1$
f_1
f_2
f_3
f_4
f_1^{-1}
f_2^{-1}
f_3^{-1}
f_4^{-1}
$f^{-1} = f_1^{-1} \circ f_2^{-1} \circ f_3^{-1} \circ f_4^{-1}$

Flows

$$f = f_k \circ f_{k-1} \circ \cdots \circ f_2 \circ f_1$$

$$f^{-1} = f_1^{-1} \circ f_2^{-1} \circ \cdots \circ f_{k-1}^{-1} \circ f_k^{-1}$$

$$\left|\frac{\partial f[z,\theta]}{\partial z}\right| = \left|\frac{\partial f_k[f_{k-1},\theta_k]}{\partial f_{k-1}}\right| \cdot \left|\frac{\partial f_{k-1}[f_{k-2},\theta_{k-1}]}{\partial f_{k-2}}\right| \cdots \left|\frac{\partial f_2[f_1,\theta_2]}{\partial f_1}\right| \cdot \left|\frac{\partial f_1[z,\theta_1]}{\partial z}\right|$$

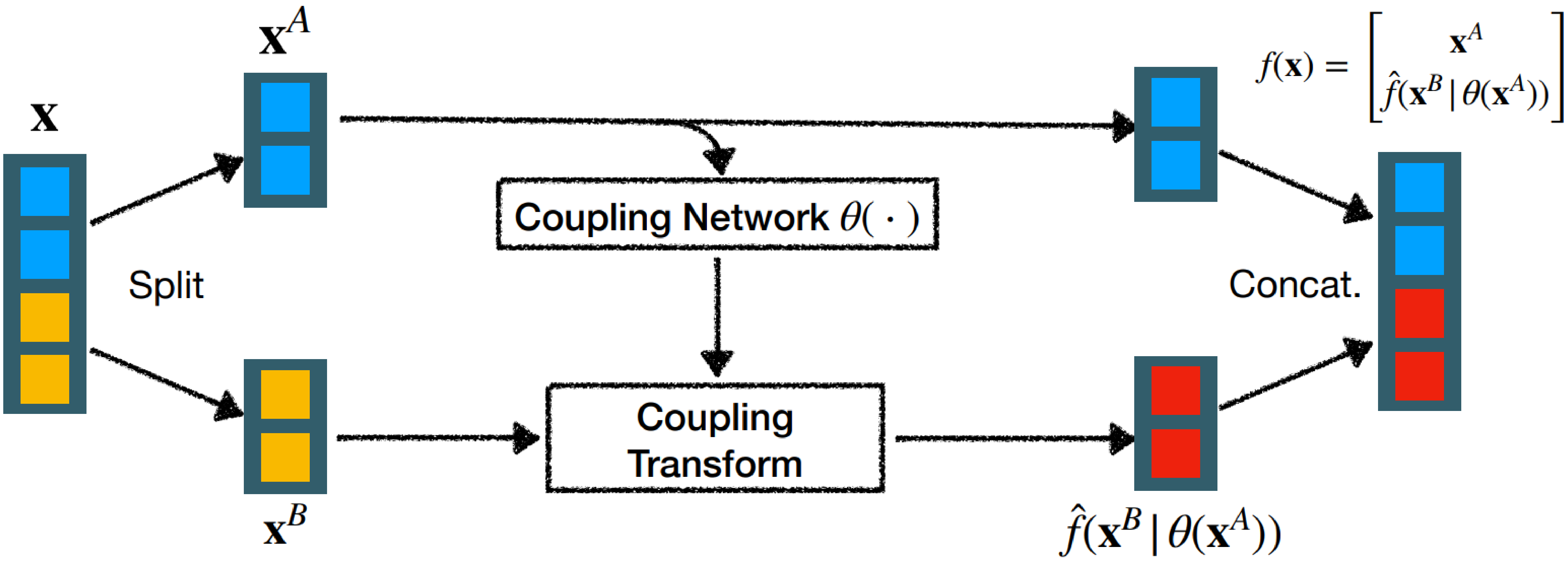
Linear Flow

- Uma transformação linear pode ser um Linear Flow se a matriz admitir inversa
$$f(x) = \theta x + b$$
- Inversa: $f^{-1}(z) = \theta^{-1}(z - b)$
- Problemas:
  - Pouco expressivas
  - Determinante e inversa podem ter custo $O(d^3)$

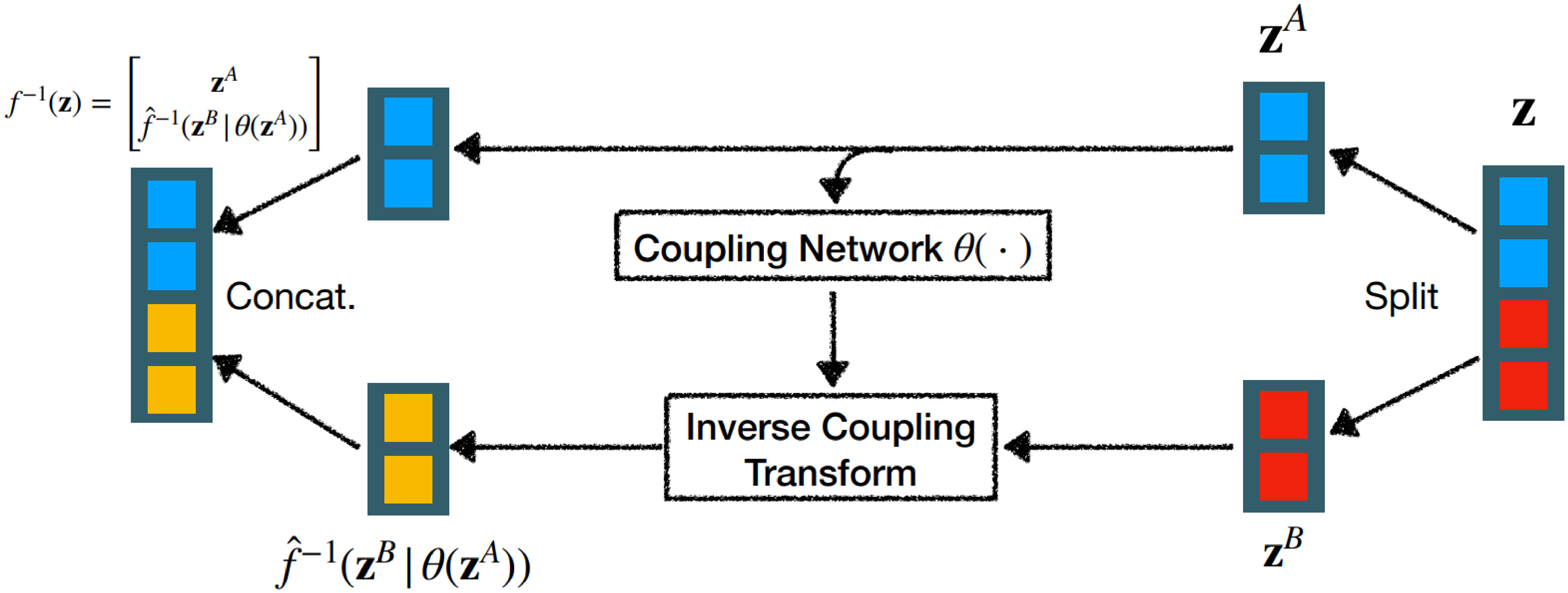
Linear Flow

	Inverse	Determinant
Full	$O(d^3)$	$O(d^3)$
Diagonal	$O(d)$	$O(d)$
Triangular	$O(d^2)$	$O(d)$
Block Diagonal	$O(c^3 d)$	$O(c^3 d)$
LU Factorized *[Kingma and Dhariwal 2018]*	$O(d^2)$	$O(d)$
Spatial Convolution *[Hoogeboom et al 2019; Karami et al., 2019]*	$O(d \log d)$	$O(d)$
1x1 Convolution *[Kingma and Dhariwal 2018]*	$O(c^3 + c^2 d)$	$O(c^3)$

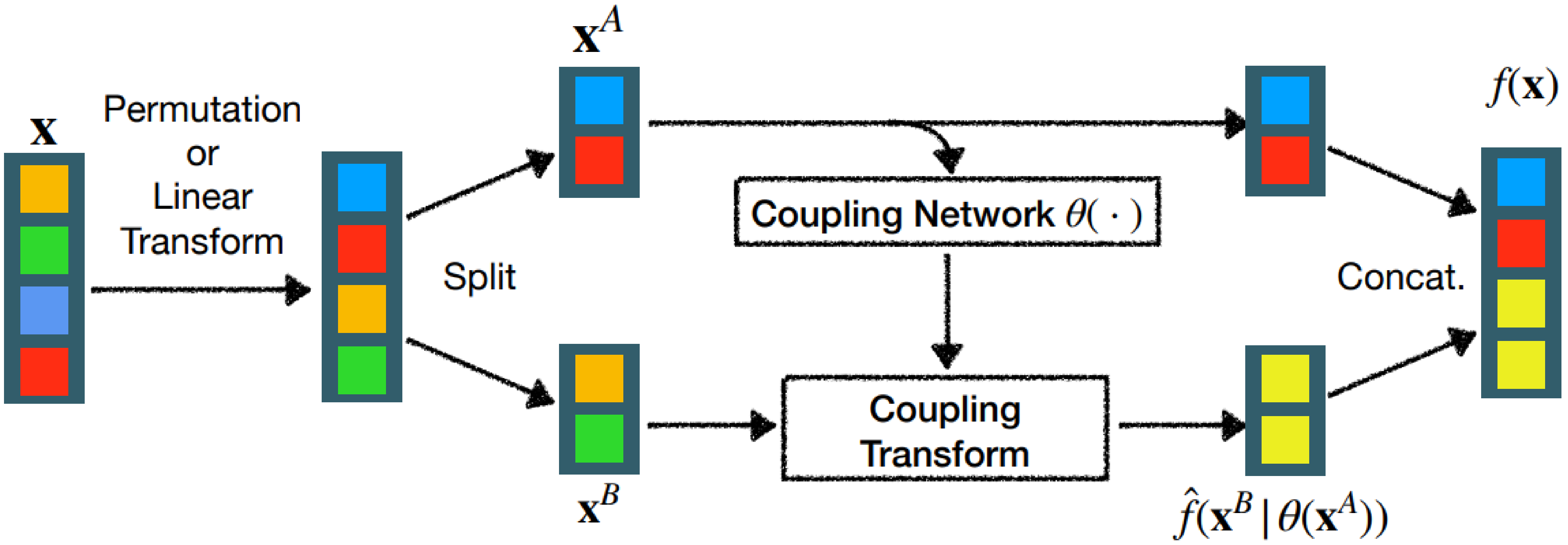
Coupling Flows



Coupling Flows



Coupling Flows



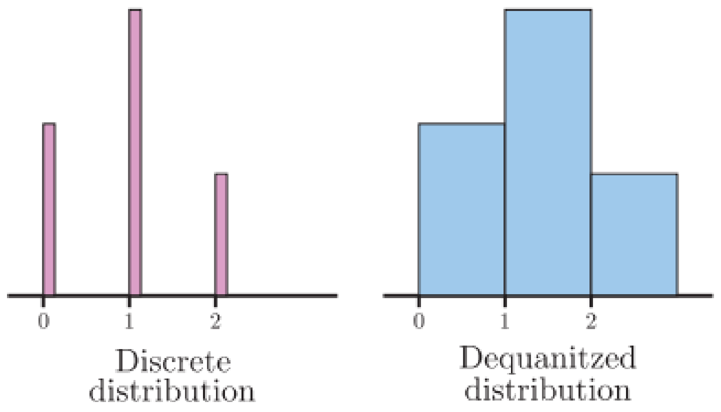
Coupling Flows

- Jacobiana: $Df(x) = \begin{bmatrix} I & 0 \\ \frac{\partial \hat{f}\left(x^B | \theta(x^A)\right)}{\partial x^A} & \hat{f}\left(x^B | \theta(x^A)\right) \end{bmatrix}$

- Determinante da Jacobiana:
  - $\hat{f}\left(x^B | \theta(x^A)\right)$

Discreto vs Contínuo

- Como imagens são quantizadas, podemos ter problemas de singularidades ao tentar ajustar uma distribuição contínua
- Para atenuar esse tipo de problema é possível adicionar um ruído (usualmente uniforme) a distribuição modelada



TLDR

- Normalizing Flows são modelos geradores capazes de amostragem e avaliação da probabilidade
- São construídos com base em uma série de transformações invertíveis
- Ainda não possuem a mesma qualidade de GANs ou modelos difusores

Referências

- KOBYZEV, Ivan; PRINCE, Simon JD; BRUBAKER, Marcus A. Normalizing flows: An introduction and review of current methods. **IEEE transactions on pattern analysis and machine intelligence**, v. 43, n. 11, p. 3964-3979, 2020.
- Capítulo 16 – Understanding Deep Learning – Simon J. Prince