

Modelos Generativos Profundos para Imágenes

Normalizing Flows (parte 2)

Pablo Musé
pmuse@fing.edu.uy

Instituto de Ingeniería Eléctrica
Facultad de Ingeniería



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Resumen de lo visto hasta aquí

Los *Normalizing flows*:

- Transforman distribuciones simples en complejas mediante una secuencia de transformaciones invertibles.
- Son modelos de variables latentes dirigidos con verosimilitud marginal dada por la fórmula de cambio de variables.
- El Jacobiano triangular permite una evaluación eficiente de las log-verosimilitudes.
- Inferencia: al ser invertibles, permiten al mismo tiempo generar muestras y evaluar la verosimilitud de un dato.

En lo que sigue

- Transformaciones invertibles con Jacobianos diagonales (NICE, Real-NVP).
- Modelos autorregresivos como modelos de flujo normalizante.
- Caso de Estudio: Destilación de densidad de probabilidad para aprendizaje e inferencia eficientes en Parallel Wavenet.

Diseño de transformaciones invertibles

- NICE: Nonlinear Independent Components Estimation (Dinh et al., 2014). Compone dos tipos de transformaciones invertibles:
 - Additive coupling layers
 - Rescaling layers
- Real-NVP: Real-valued Non-Volume Preserving (Dinh et al., 2017)
- Inverse Autoregressive Flow (Kingma et al., 2016)
- Masked Autoregressive Flow (Papamakarios et al., 2017)
- I-resnet (Behrmann et al, 2018)
- Glow (Kingma et al, 2018)
- MintNet (Song et al., 2019)
- Y muchos más.

Agenda

① Modelos con matriz Jacobiana diagonal

NICE

Real-NVP

② Modelos autorregresivos continuos como modelos de flujo

Masked Autorregressive Flow

Inverse Autorregressive Flow

③ Destilación de densidad de probabilidad y Parallel WaveNet

① Modelos con matriz Jacobiana diagonal

NICE

Real-NVP

② Modelos autorregresivos continuos como modelos de flujo

Masked Autorregressive Flow

Inverse Autorregressive Flow

③ Destilación de densidad de probabilidad y Parallel WaveNet

① Modelos con matriz Jacobiana diagonal

NICE

Real-NVP

② Modelos autorregresivos continuos como modelos de flujo

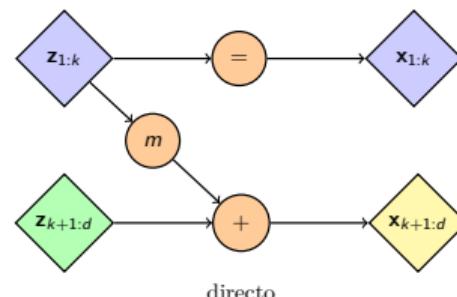
Masked Autorregressive Flow

Inverse Autorregressive Flow

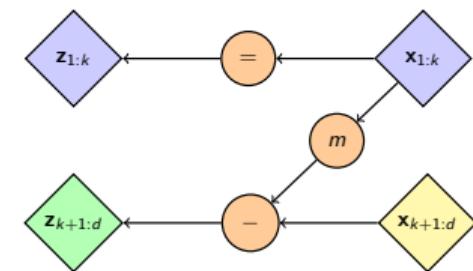
③ Destilación de densidad de probabilidad y Parallel WaveNet

NICE: additive coupling layer

- Partitiona las variables \mathbf{z} en dos subconjuntos disjuntos, $\mathbf{z}_{1:k}$ y $\mathbf{z}_{k+1:d}$, $1 \leq k < d$
- Se define el mapeo directo $\mathbf{z} \mapsto \mathbf{x}$:
 - $\mathbf{x}_{1:k} = \mathbf{z}_{1:k}$ (transformación de identidad)
 - $\mathbf{x}_{k+1:d} = \mathbf{z}_{k+1:d} + m_\theta(\mathbf{z}_{1:k})$ ($m_\theta : \mathbb{R}^k \rightarrow \mathbb{R}^{d-k}$ es una NN con parámetros θ)
- Mapeo inverso $\mathbf{x} \mapsto \mathbf{z}$:
 - $\mathbf{z}_{1:k} = \mathbf{x}_{1:k}$
 - $\mathbf{z}_{k+1:d} = \mathbf{x}_{k+1:d} - m_\theta(\mathbf{x}_{1:k})$



directo



inverso

NICE: additive coupling layer

- Jacobiano del mapeo directo

$$\begin{cases} \mathbf{x}_{1:k} = \mathbf{z}_{1:k} \\ \mathbf{x}_{k+1:d} = \mathbf{z}_{k+1:d} + m_{\theta}(\mathbf{z}_{1:k}). \end{cases}$$

$$J = \frac{\partial \mathbf{x}}{\partial \mathbf{z}} = \begin{pmatrix} I_k & 0 \\ \frac{\partial \mathbf{x}_{k+1:d}}{\partial \mathbf{z}_{1:k}} & I_{d-k} \end{pmatrix}, \quad \det(J) = 1.$$

- Observaciones:

- Transformación que **preserva el volumen** (determinante 1).
- La inversa se puede calcular para cualquier m_{θ} .
- El determinante es independiente de m_{θ} .

NICE: rescaling layers

- Las capas de acoplamiento aditivo se componen juntas (con particiones arbitrarias de variables en cada capa).
- La capa final de NICE aplica una transformación de reescalado:
- Mapeo directo $\mathbf{z} \mapsto \mathbf{x}$:

$$x_i = s_i z_i, \quad s_i > 0.$$

- Mapeo inverso $\mathbf{x} \mapsto \mathbf{z}$:

$$z_i = \frac{x_i}{s_i}.$$

- Jacobiano del mapeo directo:

$$J = \text{diag}(\mathbf{s})$$

$$\det(J) = \prod_{i=1}^d s_i.$$

NICE: muestras generadas

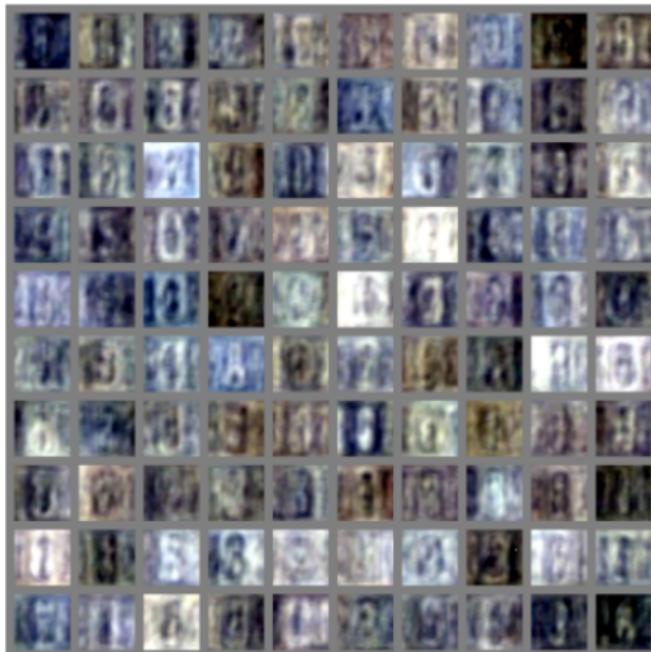


(a) Model trained on MNIST



(b) Model trained on TFD

NICE: muestras generadas



(c) Model trained on SVHN



(d) Model trained on CIFAR-10

① Modelos con matriz Jacobiana diagonal

NICE

Real-NVP

② Modelos autorregresivos continuos como modelos de flujo

Masked Autorregressive Flow

Inverse Autorregressive Flow

③ Destilación de densidad de probabilidad y Parallel WaveNet

Real-NVP: Extensión de NICE sin preservar el volumen

Mapeo directo $\mathbf{z} \mapsto \mathbf{x}$:
$$\begin{cases} \mathbf{x}_{1:k} = \mathbf{z}_{1:k} \\ \mathbf{x}_{k+1:d} = \mathbf{z}_{k+1:d} \odot \exp(\mathbf{s}_\theta(\mathbf{z}_{1:k})) + \mathbf{t}_\theta(\mathbf{z}_{1:k}). \end{cases}$$

- $\mathbf{t}_\theta : \mathbb{R}^k \rightarrow \mathbb{R}^{d-k}$, $\mathbf{s}_\theta : \mathbb{R}^k \rightarrow \mathbb{R}^{d-k}$ son NNs (redes de escalado y de translación).

Mapeo inverso $\mathbf{x} \mapsto \mathbf{z}$:
$$\begin{cases} \mathbf{z}_{1:k} = \mathbf{x}_{1:k} \\ \mathbf{z}_{k+1:d} = (\mathbf{x}_{k+1:d} - \mathbf{t}_\theta(\mathbf{x}_{1:k})) \odot \exp(-\mathbf{s}_\theta(\mathbf{x}_{1:k})). \end{cases}$$

Jacobiano mapeo directo:

$$J = \frac{\partial \mathbf{x}}{\partial \mathbf{z}} = \begin{pmatrix} I_k & 0 \\ \frac{\partial \mathbf{x}_{k+1:d}}{\partial \mathbf{z}_{1:k}} & \text{diag}(\exp(\mathbf{s}_\theta(\mathbf{z}_{1:k}))) \end{pmatrix}$$

$$\det(J) = \prod_{i=k+1}^d \text{diag}(\exp(\mathbf{s}_\theta(\mathbf{z}_{1:k})_i)).$$

⇒ No preserva el volumen en general (el determinante puede ser distinto de 1).

Real-NVP: otras capas y operaciones

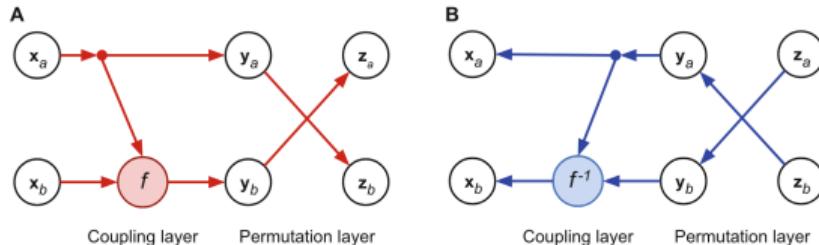
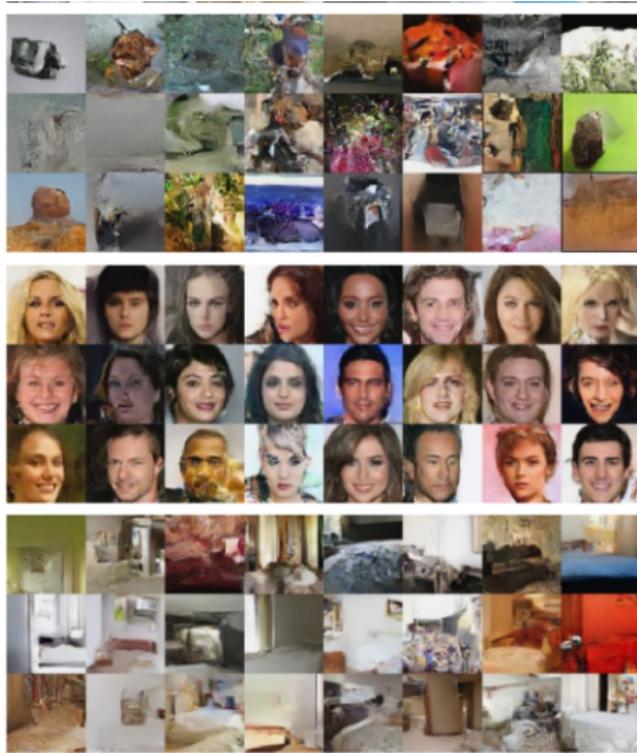
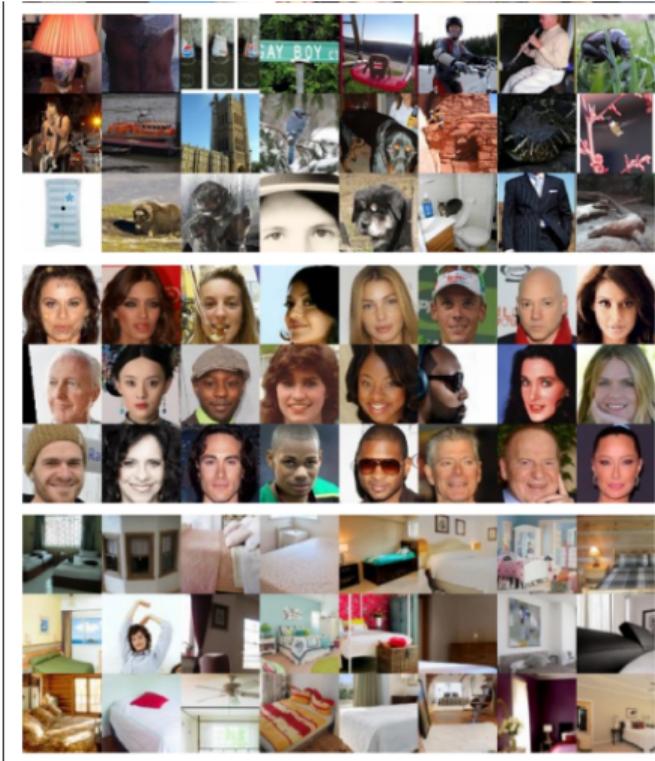


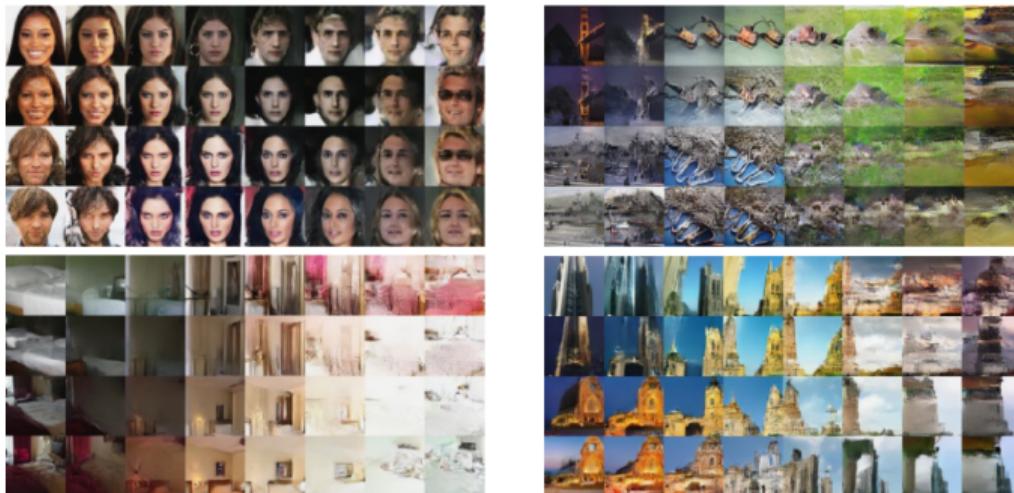
Fig. 4.4 A combination of a coupling layer and a permutation layer that transforms $[x_a, x_b]$ to $[z_a, z_b]$. (a) A forward pass through the block. (b) An inverse pass through the block.

- **Additive coupling layers**
- **Capas de permutación:** transformación inversible y de determinante 1.
- **Decuantización:** los NF asumen que $\mathbf{X} \in \mathbb{R}^d$ es continua. Muchas veces, con imágenes, $\mathbf{x} \in \{0, \dots, 255\}^d$. Una forma de obtener densidades decuantizadas es considerar $\mathbf{x} + \mathbf{u}$, con $\mathbf{u} \sim \mathcal{U}[-0,5, 0, 5]^d$.
- **Arquitectura multiescala (con squeezing y splitting), bloques residuales, batch normalization.**

Muestras generadas con Real-NVP



Interpolaciones del espacio latente a través de Real-NVP



Usando cuatro ejemplos de validación $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{z}^{(3)}, \mathbf{z}^{(4)}$, defina \mathbf{z} interpolado como:

$$\mathbf{z} = \cos\phi(\mathbf{z}^{(1)}\cos\phi' + \mathbf{z}^{(2)}\sin\phi') + \sin\phi(\mathbf{z}^{(3)}\cos\phi' + \mathbf{z}^{(4)}\sin\phi'),$$

con manifold parametrizado por ϕ y ϕ' .

① Modelos con matriz Jacobiana diagonal

NICE

Real-NVP

② Modelos autorregresivos continuos como modelos de flujo

Masked Autorregressive Flow

Inverse Autorregressive Flow

③ Destilación de densidad de probabilidad y Parallel WaveNet

Modelos autorregresivos continuos como modelos de flujo

- Consideramos un modelo autorregresivo gaussiano:

$$p(\mathbf{x}) = \prod_{i=1}^d p(x_i | \mathbf{x}_{<i}), \quad p(x_i | \mathbf{x}_{<i}) = \mathcal{N}(\mu_i(\mathbf{x}_{<i}), \exp(\alpha_i(\mathbf{x}_{<i}))).$$

Aquí, $\mu_i(\cdot)$ y $\alpha_i(\cdot)$ son NNs para $i > 1$ y constantes para $i = 1$.

- Muestreador para este modelo:

- ① Muestrear $z_i \sim \mathcal{N}(0, 1)$ para $i = 1, \dots, d$.
- ② $x_1 := \exp(\alpha_1)z_1 + \mu_1$. Calcular $\mu_2(x_1), \alpha_2(x_1)$
- ③ $x_2 := \exp(\alpha_2)z_2 + \mu_2$. Calcular $\mu_3(x_1, x_2), \alpha_3(x_1, x_2)$
- ④ $x_3 := \exp(\alpha_3)z_3 + \mu_3$

- Interpretación como flujo:** transforma muestras de la normal (z_1, z_2, \dots, z_d) a muestras del modelo (x_1, x_2, \dots, x_d) via transformaciones invertibles (parametrizadas por $\mu_i(\cdot), \alpha_i(\cdot)$)

① Modelos con matriz Jacobiana diagonal

NICE

Real-NVP

② Modelos autorregresivos continuos como modelos de flujo

Masked Autorregressive Flow

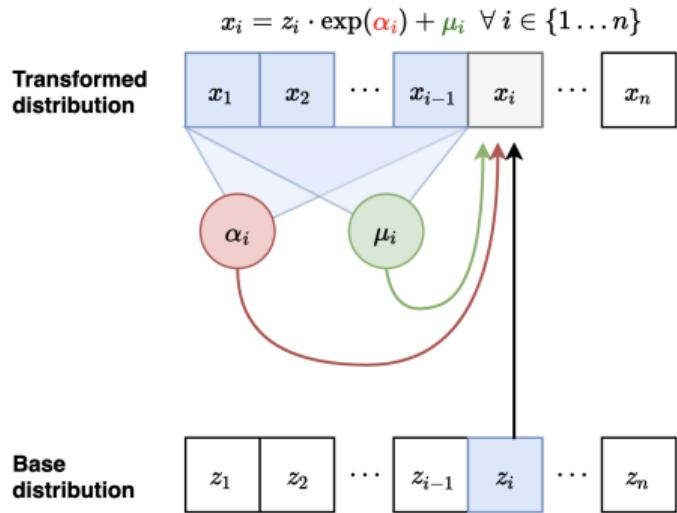
Inverse Autorregressive Flow

③ Destilación de densidad de probabilidad y Parallel WaveNet

Masked Autoregressive Flow (MAF)

Mapeo directo $\mathbf{z} \mapsto \mathbf{x}$:

- ① Muestrear $z_i \sim \mathcal{N}(0, 1)$ para $i = 1, \dots, d$.
- ② $x_1 := \exp(\alpha_1)z_1 + \mu_1$. Calcular $\mu_2(x_1), \alpha_2(x_1)$
- ③ $x_2 := \exp(\alpha_2)x_1 + \mu_2$. Calcular $\mu_3(x_1, x_2), \alpha_3(x_1, x_2)$
- ④ $x_3 := \exp(\alpha_3)x_2 + \mu_3$

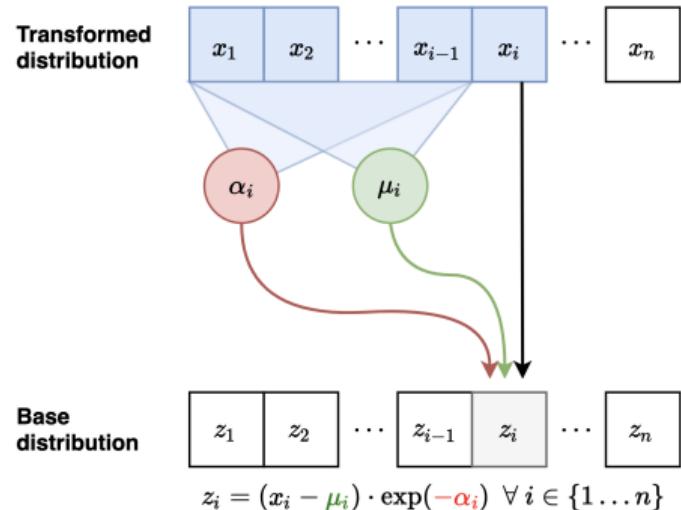


⇒ El muestreo es secuencial y lento (proceso autorregresivo): tiempo $O(d)$

Masked Autoregressive Flow (MAF)

Mapeo inverso $\mathbf{x} \mapsto \mathbf{z}$:

- Calcular todos los $\mu_i(\mathbf{x}_{<i})$, $\alpha_i(\mathbf{x}_{<i})$ (en paralelo, como MADE)
- $z_i := (x_i - \mu_i) / \exp(\alpha_i)$, $i = 1, \dots, d$.



- El Jacobiano es triangular inferior \Rightarrow cálculo eficiente del determinante.
- **Mapeo inverso (i.e. evaluación de la verosimilitud) es fácil y paralelizable.**
- Se pueden componer varias capas con distintos ordenes para las variables.

Papamakarios et al., *Masked Autoregressive Flow for Density Estimation*, NeurIPS 2017.

Figura adaptada de *Normalizing Flows tutorial*, Eric Jang, 2018.

① Modelos con matriz Jacobiana diagonal

NICE

Real-NVP

② Modelos autorregresivos continuos como modelos de flujo

Masked Autorregressive Flow

Inverse Autorregressive Flow

③ Destilación de densidad de probabilidad y Parallel WaveNet

Inverse Autoregressive Flow (IAF)

El mapeo $f : \mathbf{X} \mapsto \mathbf{Z}$ es autorregresivo en \mathbf{z} (a diferencia del MAF que es autorregresivo en \mathbf{x} .)

Mapeo directo $\mathbf{z} \mapsto \mathbf{x}$ (paralelo):

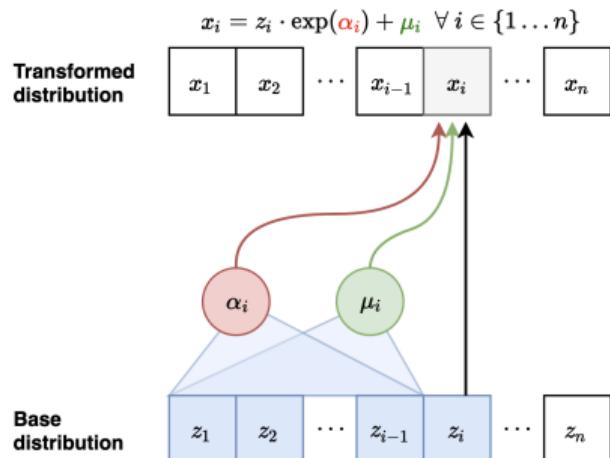
- Muestrear $z_i \sim \mathcal{N}(0, 1)$ para $i = 1, \dots, d$.
- Calcular todos los $\mu_i(\mathbf{z}_{<i})$, $\alpha_i(\mathbf{z}_{<i})$
- $x_i := \exp(\alpha_i)z_i + \mu_i$, $i = 1, \dots, d$.

Mapeo inverso $\mathbf{x} \mapsto \mathbf{z}$ (secuencial):

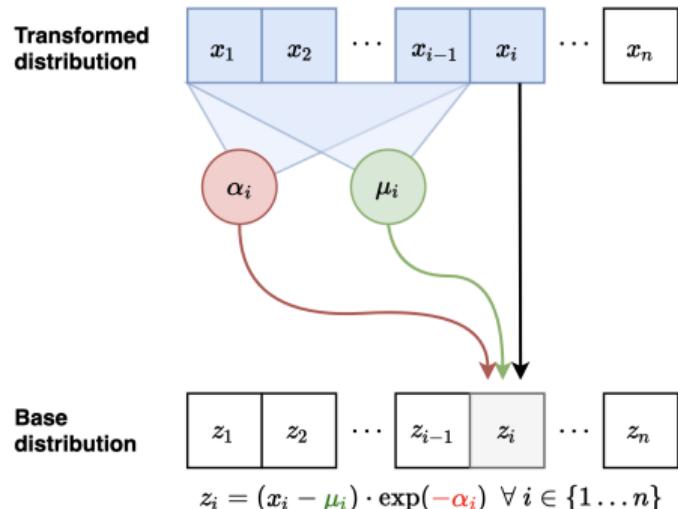
- ① $z_1 := (x_1 - \mu_1) / \exp(\alpha_1)$. Calcular $\mu_2(z_1), \alpha_2(z_1)$
- ② $z_2 := (x_2 - \mu_2) / \exp(\alpha_2)$. Calcular $\mu_3(z_1, z_2), \alpha_3(z_1, z_2)$

...

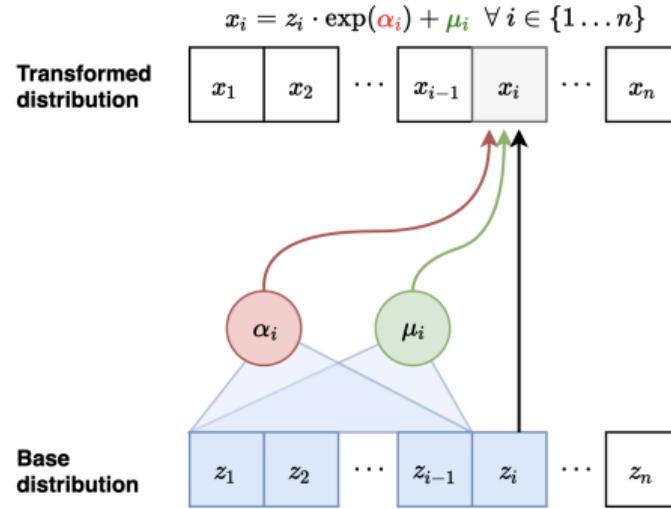
⇒ Rápido para muestrear, lento para evaluar verosimilitudes de los datos (entrenamiento)
Nota: Rápido para evaluar verosimilitudes de una muestra generada (caché z_1, z_2, \dots, z_d).



IAF es el transpuesto de MAF



Paso inverso de MAF



Paso directo de IAF

- Intercambiar \mathbf{z} y \mathbf{x} en el mapeo inverso de MAF da el mapeo directo de IAF
- Intercambiar \mathbf{z} y \mathbf{x} en el mapeo directo de MAF da el mapeo inverso de IAF.

IAF vs. MAF

- IAF y MAF son modelos expresivos, con diferentes compromisos computacionales:
 - MAF: Evaluación rápida de la verosimilitud, muestreo lento
 - IAF: Muestreo rápido, evaluación lenta de la verosimilitud
- MAF más adaptado para el entrenamiento basado en MLE y la estimación de densidad.
- IAF más adaptado para la generación en tiempo real.
- ¿Podemos obtener lo mejor de ambos mundos?

① Modelos con matriz Jacobiana diagonal

NICE

Real-NVP

② Modelos autorregresivos continuos como modelos de flujo

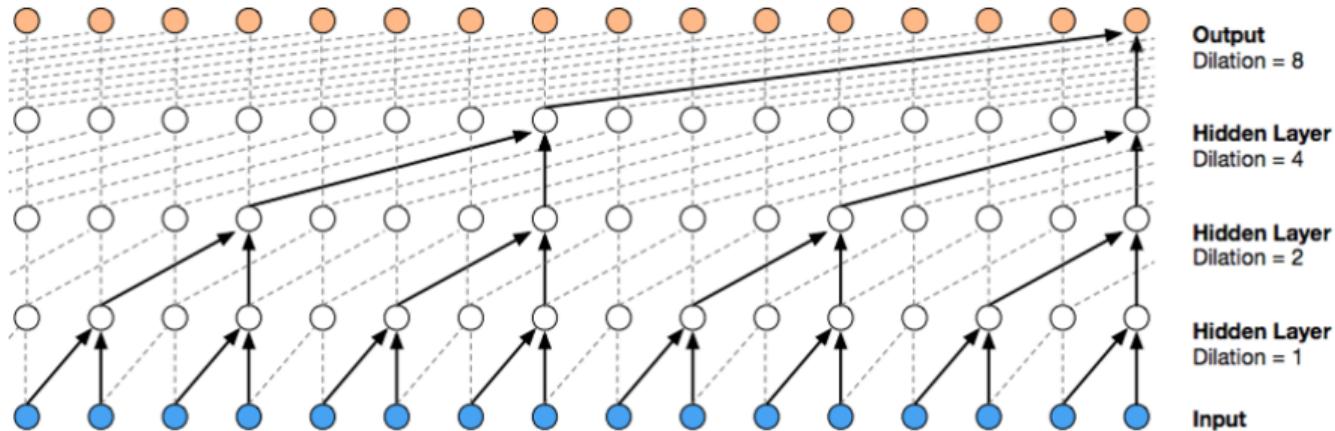
Masked Autorregressive Flow

Inverse Autorregressive Flow

③ Destilación de densidad de probabilidad y Parallel WaveNet

Reaso: WaveNet

- Modelo muy efectivo para síntesis de audio (estado del arte):



- Modelo autorregresivo \Rightarrow generación lenta (audio se muestrea a decenas de kHz).

Destilación de modelos

- **Destilación de densidad de probabilidad:** la distribución del estudiante se entrena para minimizar la divergencia KL entre el estudiante (p_s) y el maestro (p_t):

$$KL(p_s\|p_t) = E_{\mathbf{x} \sim p_s}[\log p_s(\mathbf{x}) - \log p_t(\mathbf{x})].$$

- La evaluación y optimización de las estimaciones Monte Carlo de $KL(p_s\|p_t)$ requieren:
 - Muestras \mathbf{x} del modelo estudiante (IAF)
 - Evaluaciones de la densidad del estudiante en \mathbf{x}
 - Evaluaciones de la densidad del maestro en \mathbf{x} (MAF).
- Todas las operaciones anteriores se pueden implementar de manera eficiente

Parallel Wavenet

Entrenamiento en dos partes con un modelo maestro y un modelo estudiante:

- ① Entrenar un modelo maestro WaveNet parametrizado por un MAF eficientemente (MLE)
- ② Ajustar un modelo estudiante parametrizado por un IAF
 - El modelo estudiante IAF puede generar muestras eficientemente
 - El modelo estudiante IAF puede evaluar eficientemente la densidad de sus muestras (via caching)
 - El modelo maestro MAF también puede evaluar eficientemente la densidad de las muestras generadas por el modelo estudiante, haciendo posible la destilación.

Inferencia: se genera con el modelo estudiante IAF \Rightarrow acelera WaveNet 1000x.

Resumen de Normalizing Flows

- Transformar distribuciones simples en complejas mediante **cambio de variables**.
- **Pros:**
 - La verosimilitud marginal exacta $p(\mathbf{x})$ es fácil de calcular y optimizar.
 - La inferencia posterior exacta $p(\mathbf{z}|\mathbf{x})$ es tratable.
- **Contras:**
 - La dimensión de \mathbf{z} y \mathbf{x} **debe ser la misma** (puede plantear desafíos computacionales).
 - Impone restricciones importantes sobre qué **familia de modelos** podemos usar.
- **Estrategias para Construir Flujos:**
 - Composición de bijecciones simples.
 - Jacobiano triangular (simplifica cálculos).
 - Compromisos computacionales en la evaluación de las transformaciones directa e inversa.

Referencias

-  C. M. Bishop, *Pattern Recognition and Machine Learning*.
Springer, 2006.
-  Stanford, "CS236 Deep Generative Models." <https://deepgenerativemodels.github.ioLecture>, 2024.
-  J. M. Tomczak, *Deep Generative Modeling*.
Springer Cham, 2024.