Normalizing Flows
Invertible Neural Networks (INNs)

Jérémie Cabessa

Laboratoire DAVID, UVSQ

Introduction

- On s'intéresse maintenant aux **modèles de flots génératifs (flow-based models)**.
- On possède des data d'apprentissages (e.g. des images) et on aimerait générer de nouvelles data "ressemblantes" à partir de ces dernières.
- **Idée générale:**

1. On définit un réseau de neurones inversible qui transporte la distribution des data sur une loi normale centrée réduite.
2. Pour générer de nouvelles data, on sample la loi normale et on applique la transformation inverse.

Introduction

- On s'intéresse maintenant aux **modèles de flots génératifs (flow-based models)**.
- On possède des data d'apprentissages (e.g. des images) et on aimerait générer de nouvelles data "ressemblantes" à partir de ces dernières.
- **Idée générale:**

1. On définit un réseau de neurones inversible qui transporte la distribution des data sur une loi normale centrée réduite.
2. Pour générer de nouvelles data, on sample la loi normale et on applique la transformation inverse.

Introduction

- On s'intéresse maintenant aux **modèles de flots génératifs (flow-based models)**.
- On possède des data d'apprentissages (e.g. des images) et on aimerait générer de nouvelles data "ressemblantes" à partir de ces dernières.
- **Idée générale:**

1. On définit un réseau de neurones inversible qui transporte la distribution des data sur une loi normale centrée réduite.
2. Pour générer de nouvelles data, on sample la loi normale et on applique la transformation inverse.

Introduction

- On s'intéresse maintenant aux **modèles de flots génératifs (flow-based models)**.
- On possède des data d'apprentissages (e.g. des images) et on aimerait générer de nouvelles data "ressemblantes" à partir de ces dernières.
- **Idée générale:**

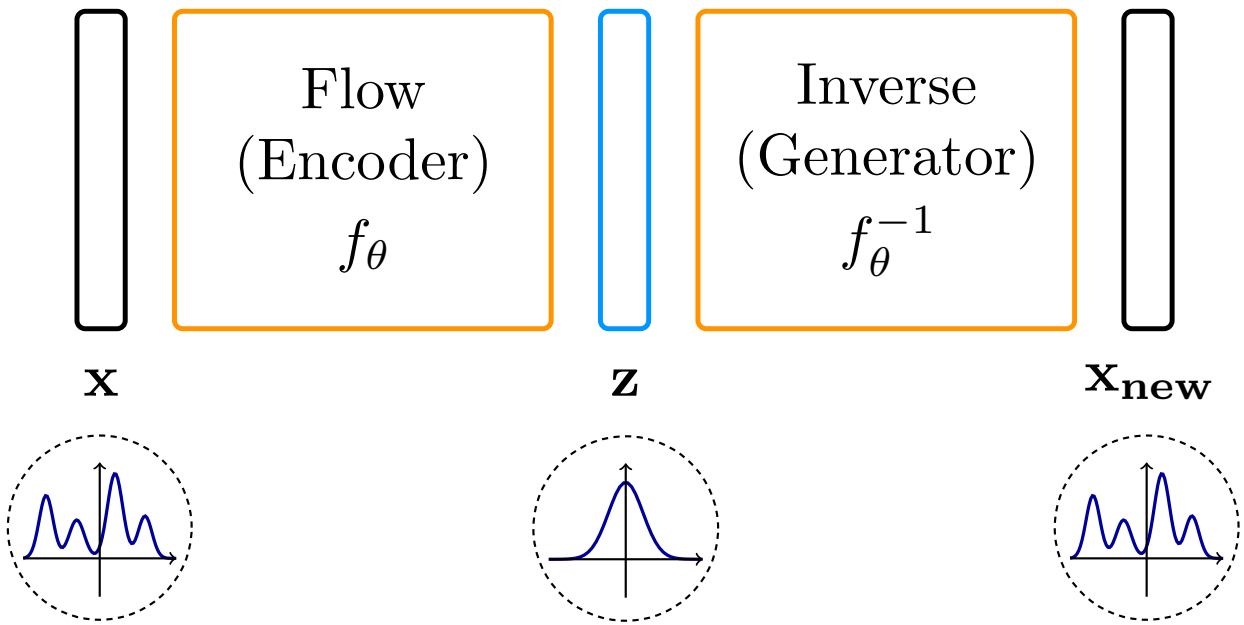
1. On définit un réseau de neurones inversible qui transporte la distribution des data sur une loi normale centrée réduite.
2. Pour générer de nouvelles data, on sample la loi normale et on applique la transformation inverse.

Introduction

- On s'intéresse maintenant aux **modèles de flots génératifs (flow-based models)**.
- On possède des data d'apprentissages (e.g. des images) et on aimerait générer de nouvelles data "ressemblantes" à partir de ces dernières.
- **Idée générale:**

1. On définit un réseau de neurones inversible qui transporte la distribution des data sur une loi normale centrée réduite.
2. Pour générer de nouvelles data, on sample la loi normale et on applique la transformation inverse.

Introduction



Introduction

- Les concepts clés qui sous-tendent ces approches génératives sont:

1. L'apprentissage d'une distribution de data.
   $\rightarrow$ **Density estimation**
2. Le transport de mesures.
   $\rightarrow$ **Transportation of measures**

Introduction

- Les concepts clés qui sous-tendent ces approches génératives sont:

1. L'apprentissage d'une distribution de data.
   $\rightarrow$ **Density estimation**
2. Le transport de mesures.
   $\rightarrow$ Transportation of measures

Introduction

- Les concepts clés qui sous-tendent ces approches génératives sont:

1. L'apprentissage d'une distribution de data.
   $\rightarrow$ **Density estimation**
2. Le transport de mesures.
   $\rightarrow$ **Transportation of measures**

Density estimation

- Soit un dataset $\mathcal{D} = \{\mathbf{x_i} \in \mathbb{R}^d : i = 1, \ldots, n\}$ dont les éléments proviennent d'une distribution empirique inconnue p, i.e.,

$$\mathbf{x_1}, \ldots, \mathbf{x_n} \sim p \quad \text{(i.i.d)}$$

- *Apprendre la distribution* p (density estimation) signifie chercher un réseau de neurones $p_\theta : \mathbb{R}^d \to \mathbb{R}$ de paramètres θ tel que

$$p_\theta(\mathbf{x}) \simeq p(\mathbf{x}), \ \forall x \in \mathbb{R}^d$$

- On cherche un réseau de neurones qui, pour tout élément $\mathbf{x}$, prédit la probabilité empirique $p(\mathbf{x})$.

Density estimation

- Soit un dataset $\mathcal{D} = \{\mathbf{x_i} \in \mathbb{R}^d : i = 1, \ldots, n\}$ dont les éléments proviennent d'une distribution empirique inconnue p, i.e.,

$$\mathbf{x_1}, \ldots, \mathbf{x_n} \sim p \quad \text{(i.i.d)}$$

- *Apprendre la distribution* p (density estimation) signifie chercher un réseau de neurones $p_\theta : \mathbb{R}^d \to \mathbb{R}$ de paramètres θ tel que

$$p_\theta(\mathbf{x}) \simeq p(\mathbf{x}), \ \forall x \in \mathbb{R}^d$$

- On cherche un réseau de neurones qui, pour tout élément $\mathbf{x}$, prédit la probabilité empirique $p(\mathbf{x})$.

Density estimation

- Soit un dataset $\mathcal{D} = \{\mathbf{x_i} \in \mathbb{R}^d : i = 1, \ldots, n\}$ dont les éléments proviennent d'une distribution empirique inconnue p, i.e.,

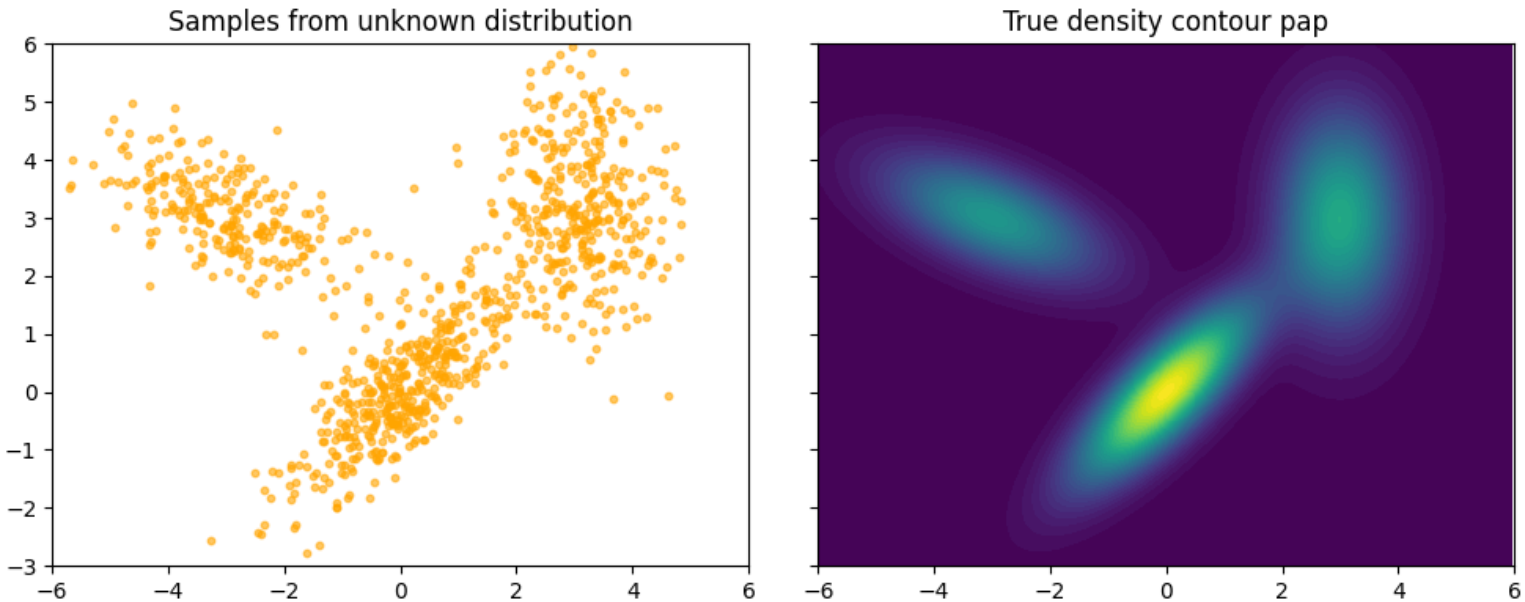
$$\mathbf{x_1}, \ldots, \mathbf{x_n} \sim p \quad \text{(i.i.d)}$$

- *Apprendre la distribution* p (density estimation) signifie chercher un réseau de neurones $p_\theta : \mathbb{R}^d \to \mathbb{R}$ de paramètres θ tel que

$$p_\theta(\mathbf{x}) \simeq p(\mathbf{x}), \ \forall x \in \mathbb{R}^d$$

- On cherche un réseau de neurones qui, pour tout élément $\mathbf{x}$, prédit la probabilité empirique $p(\mathbf{x})$.

Density estimation



Density estimation

- On cherche un réseau de neurones p_θ de paramètres θ tel que

$$p_\theta(\mathbf{x}) \simeq p(\mathbf{x}),\ \forall x \in \mathbb{R}^d$$

- Pour cela, on cherche à minimiser la distance de Kullback-Leibler entre les lois p et p_θ

$$\begin{aligned} D_{\mathsf{KL}}(p \parallel p_\theta) &= \int_{-\infty}^{\infty} p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{p_\theta(\mathbf{x})}\right) d\mathbf{x} \\ &= \int_{-\infty}^{\infty} p(\mathbf{x})[\log(p(\mathbf{x})) - \log(p_\theta(\mathbf{x}))]\, dx \\ &= \mathbb{E}_p[\log(p(\mathbf{x}))] - \mathbb{E}_p[\log(p_\theta(\mathbf{x}))] \end{aligned}$$

Density estimation

- On cherche un réseau de neurones p_θ de paramètres θ tel que

$$p_\theta(\mathbf{x}) \simeq p(\mathbf{x}),\ \forall x \in \mathbb{R}^d$$

- Pour cela, on cherche à minimiser la distance de Kullback-Leibler entre les lois p et p_θ

$$\begin{aligned} D_{\mathsf{KL}}(p \parallel p_\theta) &= \int_{-\infty}^{\infty} p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{p_\theta(\mathbf{x})}\right) d\mathbf{x} \\ &= \int_{-\infty}^{\infty} p(\mathbf{x}) \big[\log(p(\mathbf{x})) - \log(p_\theta(\mathbf{x}))\big] d\mathbf{x} \\ &= \mathbb{E}_p\big[\log(p(\mathbf{x}))\big] - \mathbb{E}_p\big[\log(p_\theta(\mathbf{x}))\big] \end{aligned} \tag{1}$$

Density estimation

- On cherche un réseau de neurones p_θ de paramètres θ tel que

$$p_\theta(\mathbf{x}) \simeq p(\mathbf{x}),\ \forall x \in \mathbb{R}^d$$

- Pour cela, on cherche à minimiser la distance de Kullback-Leibler entre les lois p et p_θ

$$\begin{aligned} D_{\mathsf{KL}}(p \parallel p_\theta) &= \int_{-\infty}^{\infty} p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{p_\theta(\mathbf{x})}\right) d\mathbf{x} \\ &= \int_{-\infty}^{\infty} p(\mathbf{x}) \big[\log(p(\mathbf{x})) - \log(p_\theta(\mathbf{x}))\big] d\mathbf{x} \\ &= \mathbb{E}_p\big[\log(p(\mathbf{x}))\big] - \mathbb{E}_p\big[\log(p_\theta(\mathbf{x}))\big] \end{aligned} \tag{1}$$

Density estimation

- On cherche un réseau de neurones p_θ de paramètres θ tel que

$$p_\theta(\mathbf{x}) \simeq p(\mathbf{x}),\ \forall x \in \mathbb{R}^d$$

- Pour cela, on cherche à minimiser la distance de Kullback-Leibler entre les lois p et p_θ

$$\begin{aligned} D_{\mathsf{KL}}(p \,\|\, p_\theta) &= \int_{-\infty}^{\infty} p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{p_\theta(\mathbf{x})}\right) d\mathbf{x} \\ &= \int_{-\infty}^{\infty} p(\mathbf{x}) \big[\log(p(\mathbf{x})) - \log(p_\theta(\mathbf{x}))\big] d\mathbf{x} \\ &= \mathbb{E}_p\big[\log(p(\mathbf{x}))\big] - \mathbb{E}_p\big[\log(p_\theta(\mathbf{x}))\big] \qquad (1) \end{aligned}$$

Density estimation

- On cherche un réseau de neurones p_θ de paramètres θ tel que

$$p_\theta(\mathbf{x}) \simeq p(\mathbf{x}),\ \forall x \in \mathbb{R}^d$$

- Pour cela, on cherche à minimiser la distance de Kullback-Leibler entre les lois p et p_θ

$$\begin{aligned} D_{\mathsf{KL}}(p \parallel p_\theta) &= \int_{-\infty}^{\infty} p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{p_\theta(\mathbf{x})}\right) d\mathbf{x} \\ &= \int_{-\infty}^{\infty} p(\mathbf{x}) \big[\log(p(\mathbf{x})) - \log(p_\theta(\mathbf{x}))\big] d\mathbf{x} \\ &= \mathbb{E}_p\big[\log(p(\mathbf{x}))\big] - \mathbb{E}_p\big[\log(p_\theta(\mathbf{x}))\big] \end{aligned} \tag{1}$$

Density estimation

- Les paramètres $\hat{\theta}$ qui minimisent $D_{\mathsf{KL}}(p \,\|\, p_\theta)$ sont donc donnés par

$$\hat{\theta} = \arg\min_\theta \Big\{ D_{\mathsf{KL}}(p \,\|\, p_\theta) \Big\}$$

$$\text{by (1)} \quad = \arg\min_\theta \Big\{ \mathbb{E}_p\big[\log(p(\mathbf{x}))\big] - \mathbb{E}_p\big[\log(p_\theta(\mathbf{x}))\big] \Big\}$$

$$\text{term indep of } \theta \quad = \arg\min_\theta \Big\{ - \mathbb{E}_p\big[\log(p_\theta(\mathbf{x}))\big] \Big\} \qquad (2)$$

Density estimation

- Les paramètres $\hat{\theta}$ qui minimisent $D_{\mathsf{KL}}(p \,\|\, p_\theta)$ sont donc donnés par

$$\begin{aligned}
\hat{\theta} &= \arg\min_\theta \Big\{ D_{\mathsf{KL}}(p \,\|\, p_\theta) \Big\} \\
\text{by (1)} \quad &= \arg\min_\theta \Big\{ \mathbb{E}_p\big[\log(p(\mathbf{x}))\big] - \mathbb{E}_p\big[\log(p_\theta(\mathbf{x}))\big] \Big\} \\
\text{term indep of } \theta \quad &= \arg\min_\theta \Big\{ -\mathbb{E}_p\big[\log(p_\theta(\mathbf{x}))\big] \Big\} \qquad (2)
\end{aligned}$$

Density estimation

- Les paramètres $\hat{\theta}$ qui minimisent $D_{\mathsf{KL}}(p \,\|\, p_\theta)$ sont donc donnés par

$$\begin{aligned}
\hat{\theta} &= \arg\min_\theta \Big\{ D_{\mathsf{KL}}(p \,\|\, p_\theta) \Big\} \\
\text{by (1)} \quad &= \arg\min_\theta \Big\{ \mathbb{E}_p\big[\log(p(\mathbf{x}))\big] - \mathbb{E}_p\big[\log(p_\theta(\mathbf{x}))\big] \Big\} \\
\text{term indep of } \theta \quad &= \arg\min_\theta \Big\{ - \mathbb{E}_p\big[\log(p_\theta(\mathbf{x}))\big] \Big\} \qquad (2)
\end{aligned}$$

Density estimation

- Les paramètres $\hat{\theta}$ qui minimisent $D_{\mathsf{KL}}(p \,\|\, p_\theta)$ sont donc donnés par

$$\begin{aligned}
\hat{\theta} &= \arg\min_\theta \Big\{ D_{\mathsf{KL}}(p \,\|\, p_\theta) \Big\} \\
\text{by (1)} \quad &= \arg\min_\theta \Big\{ \mathbb{E}_p\big[\log(p(\mathbf{x}))\big] - \mathbb{E}_p\big[\log(p_\theta(\mathbf{x}))\big] \Big\} \\
\text{term indep of } \theta \quad &= \arg\min_\theta \Big\{ -\mathbb{E}_p\big[\log(p_\theta(\mathbf{x}))\big] \Big\} \qquad (2)
\end{aligned}$$

Density estimation

- On ne connaît pas la loi p mais on peut l'estimer de manière discrète à partir du dataset $\mathcal{D}$

$$p(\mathbf{x}) = \begin{cases} 1/n & \text{si } \mathbf{x} = \mathbf{x_i} \text{ pour } i = 1, \dots, n \text{ (i.e. si } \mathbf{x} \in \mathcal{D}) \\ 0 & \text{sinon} \end{cases}$$

- Ainsi, l'espérance cherchée est estimée par

$$\mathbb{E}_p\big[\log(p_\theta(x))\big] \simeq \sum_{i=1}^{n} p(\mathbf{x_i}) \log(p_\theta(\mathbf{x_i})) = \sum_{i=1}^{n} \frac{1}{n} \log(p_\theta(\mathbf{x_i}))$$

et les paramètres optimaux $\hat{\theta}$ sont donc donnés par (by (2))

$$\hat{\theta} = \arg\min_\theta \Big\{ -\sum_{i=1}^{n} \frac{1}{n} \log(p_\theta(\mathbf{x_i})) \Big\} \tag{3}$$

Density estimation

- On ne connaît pas la loi p mais on peut l'estimer de manière discrète à partir du dataset $\mathcal{D}$

$$p(\mathbf{x}) = \begin{cases} 1/n & \text{si } \mathbf{x} = \mathbf{x_i} \text{ pour } i = 1, \dots, n \text{ (i.e. si } \mathbf{x} \in \mathcal{D}) \\ 0 & \text{sinon} \end{cases}$$

- Ainsi, l'espérance cherchée est estimée par

$$\mathbb{E}_p\big[\log(p_\theta(x))\big] \quad \simeq \quad \sum_{i=1}^{n} p(\mathbf{x_i}) \log(p_\theta(\mathbf{x_i})) = \sum_{i=1}^{n} \frac{1}{n} \log(p_\theta(\mathbf{x_i}))$$

et les paramètres optimaux $\hat{\theta}$ sont donc donnés par (by (2))

$$\hat{\theta} \quad = \quad \arg\min_\theta \Big\{ -\sum_{i=1}^{n} \frac{1}{n} \log(p_\theta(\mathbf{x_i})) \Big\} \tag{3}$$

Density estimation

- On obtient alors la fonction de loss, appelée **negative log likelihood (NLL)** ou **negative log density**, qui permet d'apprendre la distribution empirique p (by (3))

$$\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\log(p_\theta(\mathbf{x_i})) = -\frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}}\log(p_\theta(\mathbf{x})) \quad (4)$$

- On peut donc entraîner un réseau de neurones multicouches p_θ avec la loss $\mathcal{L}(\theta)$ pour tout problème de density estimation.

Density estimation

- On obtient alors la fonction de loss, appelée **negative log likelihood (NLL)** ou **negative log density**, qui permet d'apprendre la distribution empirique p (by (3))

$$\mathcal{L}(\theta) \quad = \quad -\frac{1}{n}\sum_{i=1}^{n}\log(p_\theta(\mathbf{x_i})) \quad = \quad -\frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}}\log(p_\theta(\mathbf{x})) \tag{4}$$

- On peut donc entraîner un réseau de neurones multicouches p_θ avec la loss $\mathcal{L}(\theta)$ pour tout problème de density estimation.

Density estimation

- En pratique, la density estimation via réseaux de neurones classiques et minimisation de la NLL fonctionne mal...
- Distributions multimodales difficiles à apprendre
- Distributions de haute dimension très difficiles à apprendre
  $\rightarrow$ curse of dimensionality
- Il existe beaucoup d'autres méthodes très performantes.

Density estimation

- En pratique, la density estimation via réseaux de neurones classiques et minimisation de la NLL fonctionne mal...
- Distributions multimodales difficiles à apprendre
- Distributions de haute dimension très difficiles à apprendre
  $\rightarrow$ curse of dimensionality
- Il existe beaucoup d'autres méthodes très performantes.

Density estimation

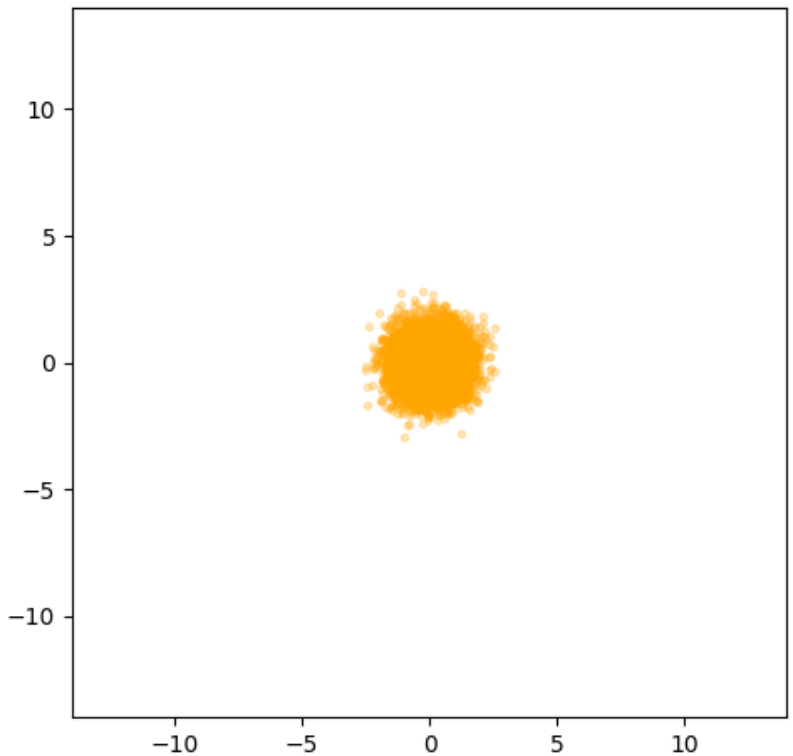
- En pratique, la density estimation via réseaux de neurones classiques et minimisation de la NLL fonctionne mal...
- Distributions multimodales difficiles à apprendre
- Distributions de haute dimension très difficiles à apprendre
  $\rightarrow$ curse of dimensionality
- Il existe beaucoup d'autres méthodes très performantes.

Density estimation

- En pratique, la density estimation via réseaux de neurones classiques et minimisation de la NLL fonctionne mal...
- Distributions multimodales difficiles à apprendre
- Distributions de haute dimension très difficiles à apprendre
  $\rightarrow$ curse of dimensionality
- Il existe beaucoup d'autres méthodes très performantes.

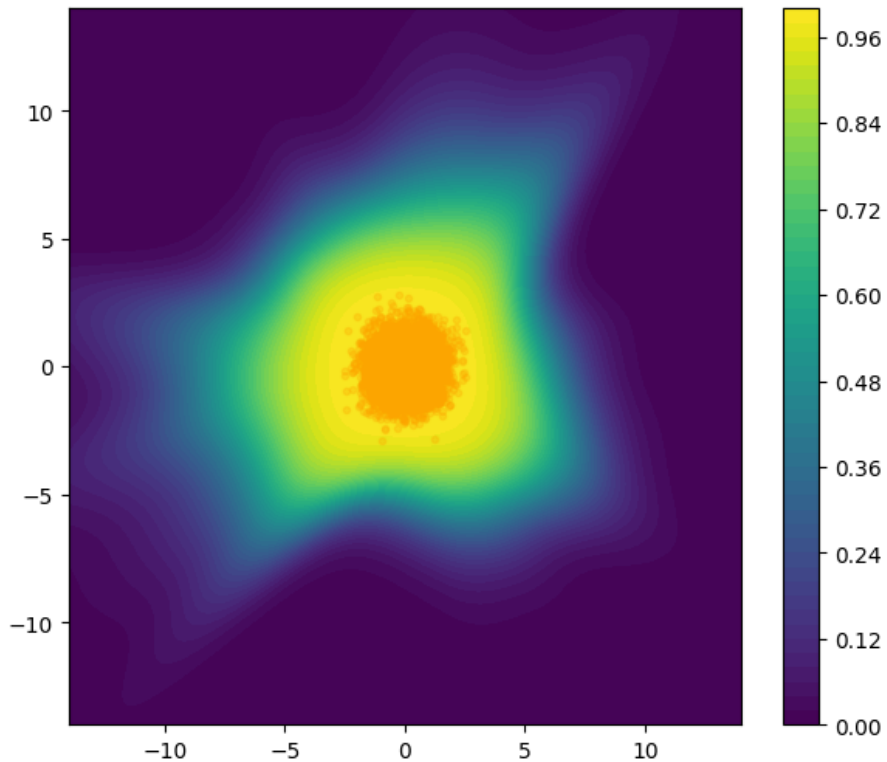
Density estimation

- Apprentissage de distributions multimodales (1, 2, 3 et 4 modes) par un MLP.
- Le MLP n'arrive pas à capturer les différents modes de manière discontinue.



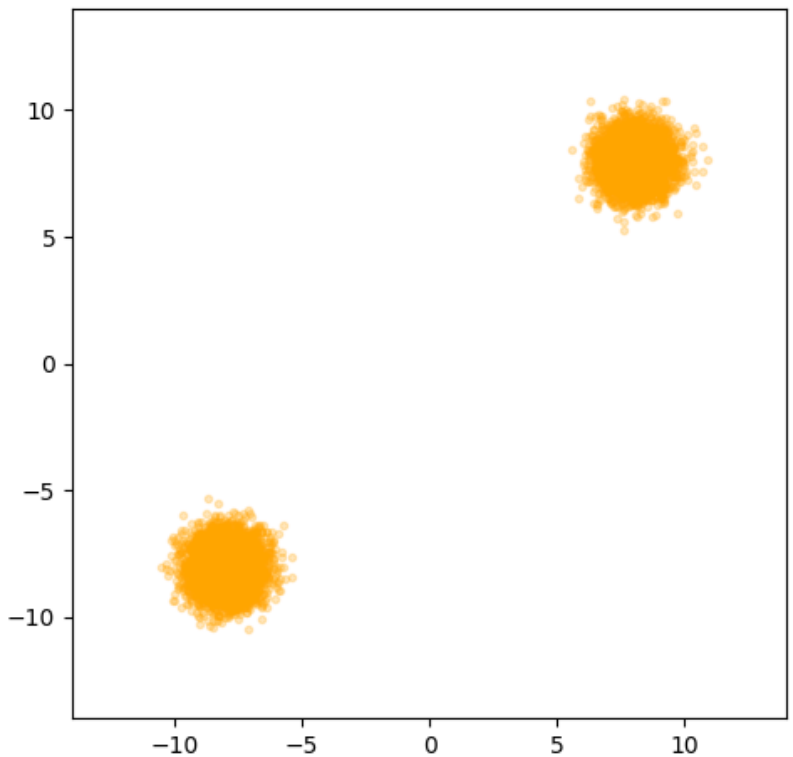
Density estimation

- Apprentissage de distributions multimodales (1, 2, 3 et 4 modes) par un MLP.
- Le MLP n'arrive pas à capturer les différents modes de manière discontinue.



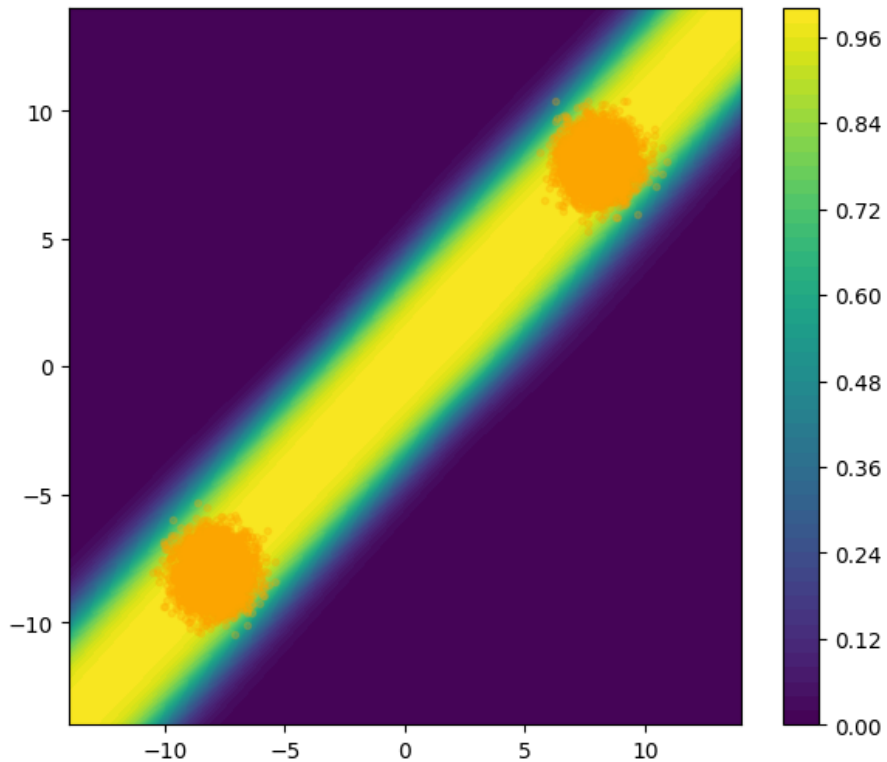
Density estimation

- Apprentissage de distributions multimodales (1, 2, 3 et 4 modes) par un MLP.
- Le MLP n'arrive pas à capturer les différents modes de manière discontinue.



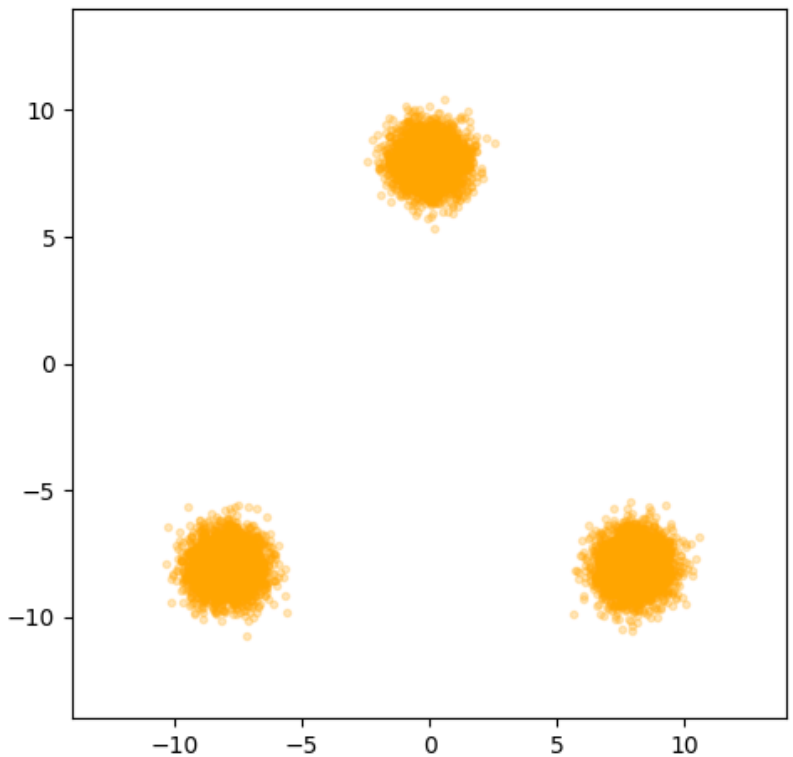
Density estimation

- Apprentissage de distributions multimodales (1, 2, 3 et 4 modes) par un MLP.
- Le MLP n'arrive pas à capturer les différents modes de manière discontinue.



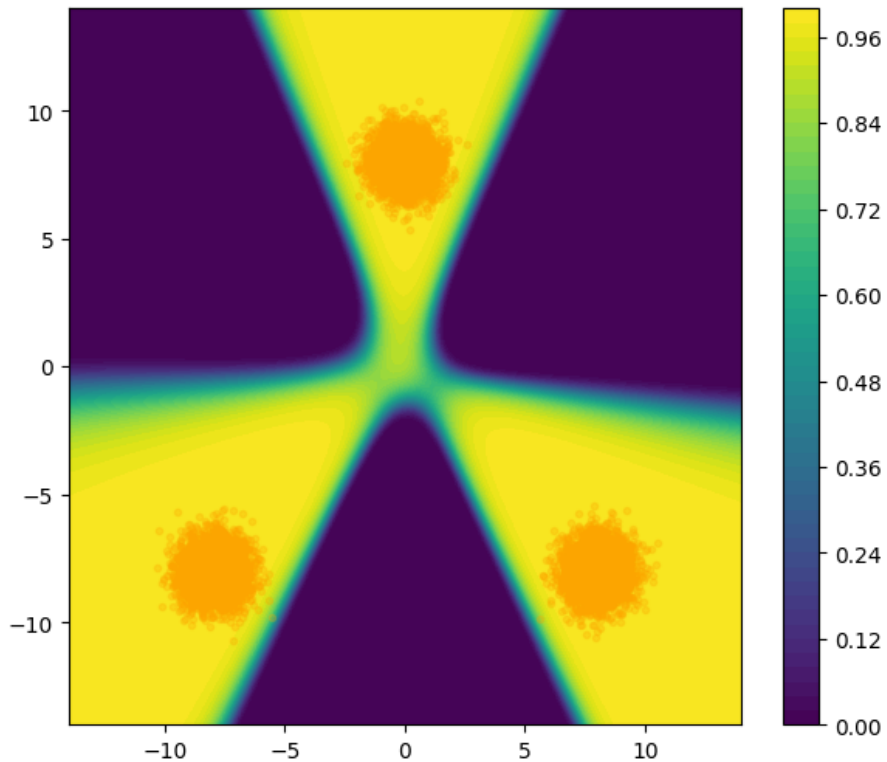
Density estimation

- Apprentissage de distributions multimodales (1, 2, 3 et 4 modes) par un MLP.
- Le MLP n'arrive pas à capturer les différents modes de manière discontinue.



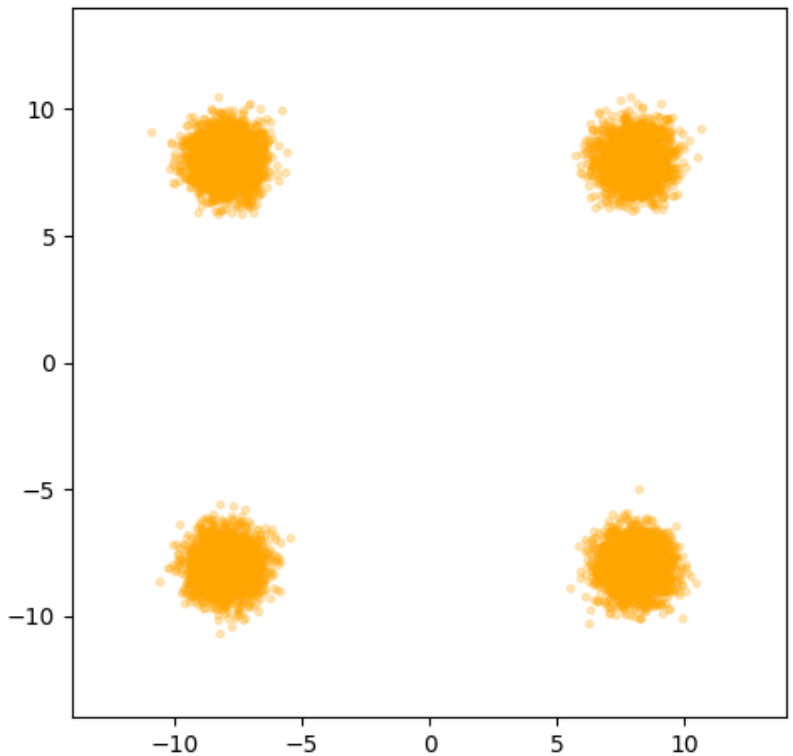
Density estimation

- Apprentissage de distributions multimodales (1, 2, 3 et 4 modes) par un MLP.
- Le MLP n'arrive pas à capturer les différents modes de manière discontinue.



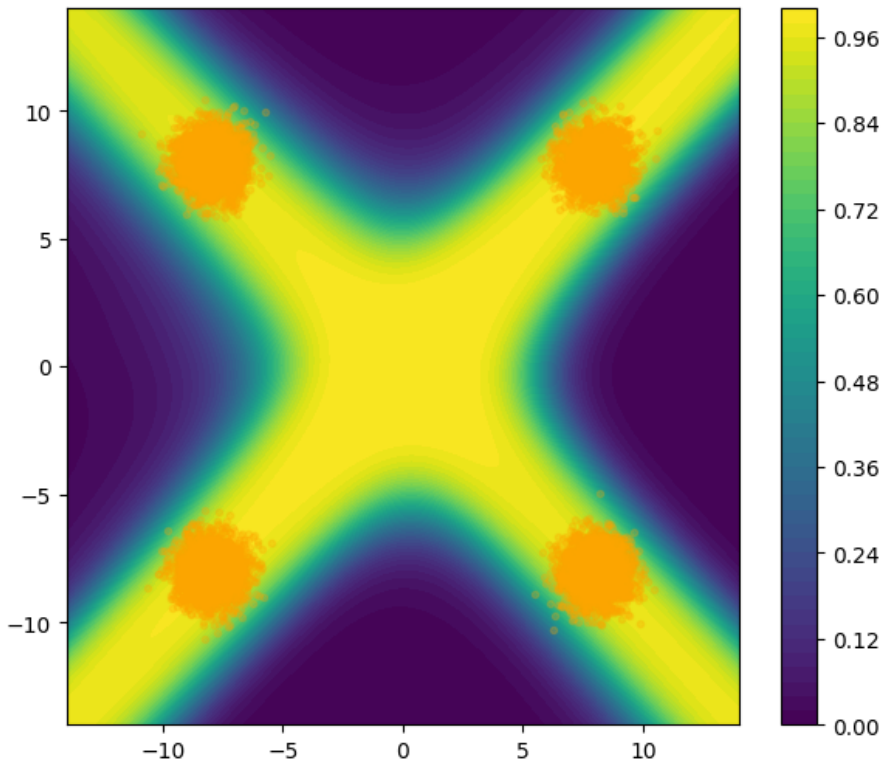
Density estimation

- Apprentissage de distributions multimodales (1, 2, 3 et 4 modes) par un MLP.
- Le MLP n'arrive pas à capturer les différents modes de manière discontinue.



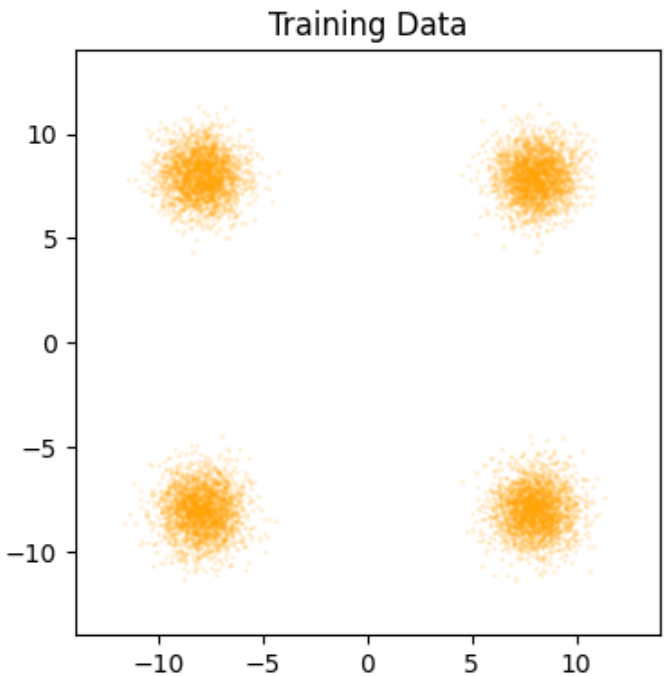
Density estimation

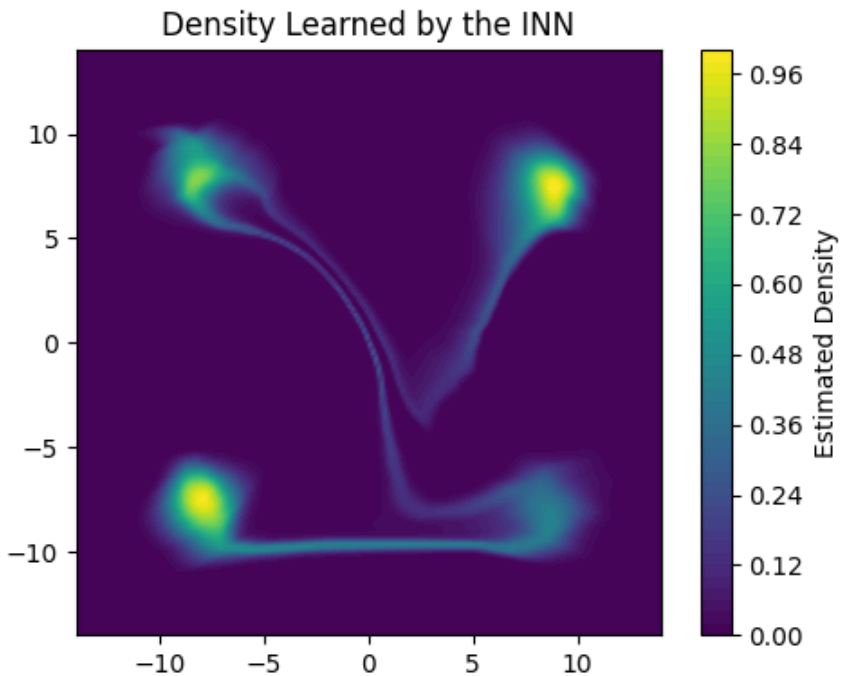
- Apprentissage de distributions multimodales (1, 2, 3 et 4 modes) par un MLP.
- Le MLP n'arrive pas à capturer les différents modes de manière discontinue.



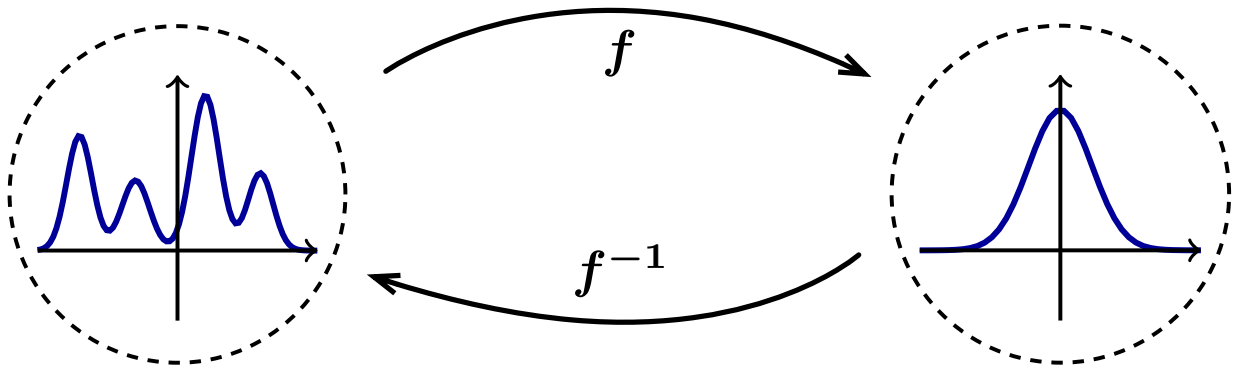
Density estimation

- Apprentissage d'une distributions multimodales (4 modes) par un réseau de neurones inversible (INN) (cf. slides suivants).
- C'est beaucoup mieux ! (malgré quelques filaments de continuité entre les modes)





Transport de mesures



Random variable $\mathbf{X}$
with density $p_{\mathbf{X}}$

Random variable $\mathbf{Z} = f(\mathbf{X})$
with density $p_{\mathbf{Z}}$

Transport de mesures (dim. 1)

- Soit X une variable aléatoire sur $\mathbb{R}$ de fonction de densité p_X (PDF) et de répartition P_X (CDF).

  Rappel: $P_X(a) = Pr(X \leq a) = \int_{-\infty}^{a} p_X(x)dx$

- Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction différentiable bijective.

  Remarque: f continue et bijective $\Rightarrow$ f monotone croissante ou monotone décroissante (sinon, on a des oscillations qui cassent l'injectivité et donc la bijectivité).

- Soit $Z = f(X)$ la variable aléatoire obtenue par transformation de X via f. On cherche à exprimer la fonction de densité p_Z de Z en fonction de p_X.

Transport de mesures (dim. 1)

- Soit X une variable aléatoire sur $\mathbb{R}$ de fonction de densité p_X (PDF) et de répartition P_X (CDF).

  Rappel: $P_X(a) = Pr(X \leq a) = \int_{-\infty}^{a} p_X(x)dx$

- Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction différentiable bijective.

  Remarque: f continue et bijective $\Rightarrow$ f monotone croissante ou monotone décroissante (sinon, on a des oscillations qui cassent l'injectivité et donc la bijectivité).

- Soit $Z = f(X)$ la variable aléatoire obtenue par transformation de X via f. On cherche à exprimer la fonction de densité p_Z de Z en fonction de p_X.

Transport de mesures (dim. 1)

- Soit X une variable aléatoire sur $\mathbb{R}$ de fonction de densité p_X (PDF) et de répartition P_X (CDF).

  Rappel: $P_X(a) = Pr(X \leq a) = \int_{-\infty}^{a} p_X(x)dx$

- Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction différentiable bijective.

  Remarque: f continue et bijective $\Rightarrow$ f monotone croissante ou monotone décroissante (sinon, on a des oscillations qui cassent l'injectivité et donc la bijectivité).

- Soit $Z = f(X)$ la variable aléatoire obtenue par transformation de X via f. On cherche à exprimer la fonction de densité p_Z de Z en fonction de p_X.

Transport de mesures (dim. 1)

- Soit X une variable aléatoire sur $\mathbb{R}$ de fonction de densité p_X (PDF) et de répartition P_X (CDF).

  Rappel: $P_X(a) = Pr(X \leq a) = \int_{-\infty}^{a} p_X(x)dx$

- Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction différentiable bijective.

  Remarque: f continue et bijective $\Rightarrow$ f monotone croissante ou monotone décroissante (sinon, on a des oscillations qui cassent l'injectivité et donc la bijectivité).

- Soit $Z = f(X)$ la variable aléatoire obtenue par transformation de X via f. On cherche à exprimer la fonction de densité p_Z de Z en fonction de p_X.

Transport de mesures (dim. 1)

- Soit X une variable aléatoire sur $\mathbb{R}$ de fonction de densité p_X (PDF) et de répartition P_X (CDF).

  Rappel: $P_X(a) = Pr(X \leq a) = \int_{-\infty}^{a} p_X(x)dx$

- Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction différentiable bijective.

  Remarque: f continue et bijective $\Rightarrow$ f monotone croissante ou monotone décroissante (sinon, on a des oscillations qui cassent l'injectivité et donc la bijectivité).

- Soit $Z = f(X)$ la variable aléatoire obtenue par transformation de X via f. On cherche à exprimer la fonction de densité p_Z de Z en fonction de p_X.

Transport de mesures (dim. 1)

- Soit X une variable aléatoire sur $\mathbb{R}$ de fonction de densité p_X (PDF) et de répartition P_X (CDF).

  Rappel: $P_X(a) = Pr(X \leq a) = \int_{-\infty}^{a} p_X(x)dx$

- Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction différentiable bijective.

  Remarque: f continue et bijective $\Rightarrow$ f monotone croissante ou monotone décroissante (sinon, on a des oscillations qui cassent l'injectivité et donc la bijectivité).

- Soit $Z = f(X)$ la variable aléatoire obtenue par transformation de X via f. On cherche à exprimer la fonction de densité p_Z de Z en fonction de p_X.

Transport de mesures (dim. 1)

- Soit X une variable aléatoire sur $\mathbb{R}$ de fonction de densité p_X (PDF) et de répartition P_X (CDF).

  Rappel: $P_X(a) = Pr(X \leq a) = \int_{-\infty}^{a} p_X(x)dx$

- Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction différentiable bijective.

  Remarque: f continue et bijective $\Rightarrow$ f monotone croissante ou monotone décroissante (sinon, on a des oscillations qui cassent l'injectivité et donc la bijectivité).

- Soit $Z = f(X)$ la variable aléatoire obtenue par transformation de X via f. On cherche à exprimer la fonction de densité p_Z de Z en fonction de p_X.

Transport de mesures (dim. 1)

Si f est monotone croissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \leq f^{-1}(z) \tag{5}$$

$$(f^{-1}(z))' \geq 0 \quad \Rightarrow \quad \left|(f^{-1}(z))'\right| = (f^{-1}(z))' \tag{6}$$

Ainsi

$$\begin{aligned} P_Z(z) &= Pr(Z \leq z) = Pr(f(X) \leq z) \\ &\overset{(5)}{=} Pr(X \leq f^{-1}(z)) = P_X(f^{-1}(z)) \\ p_Z(z) &= \frac{dP_Z(z)}{dz} = \frac{dP_X(f^{-1}(z))}{dz} = \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{dx}{dz} \\ &= \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{d(f^{-1}(z))}{dz} \overset{(6)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{d(f^{-1}(z))}{dz}\right| \end{aligned}$$

Transport de mesures (dim. 1)

Si f est monotone croissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \leq f^{-1}(z) \tag{5}$$

$$\left(f^{-1}(z)\right)' \geq 0 \quad \Rightarrow \quad \left|\left(f^{-1}(z)\right)'\right| = \left(f^{-1}(z)\right)' \tag{6}$$

Ainsi

$$\begin{aligned} P_Z(z) &= Pr(Z \leq z) = Pr\big(f(X) \leq z\big) \\ &\overset{(5)}{=} Pr\big(X \leq f^{-1}(z)\big) = P_X\big(f^{-1}(z)\big) \\ p_Z(z) &= \frac{dP_Z(z)}{dz} = \frac{dP_X(f^{-1}(z))}{dz} = \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{dx}{dz} \\ &= \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{d(f^{-1}(z))}{dz} \overset{(6)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{d(f^{-1}(z))}{dz}\right| \end{aligned}$$

Transport de mesures (dim. 1)

Si f est monotone croissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \leq f^{-1}(z) \tag{5}$$

$$\left(f^{-1}(z)\right)' \geq 0 \quad \Rightarrow \quad \left|\left(f^{-1}(z)\right)'\right| = \left(f^{-1}(z)\right)' \tag{6}$$

Ainsi

$$\begin{aligned} P_Z(z) &= Pr(Z \leq z) = Pr\big(f(X) \leq z\big) \\ &\overset{(5)}{=} Pr\big(X \leq f^{-1}(z)\big) = P_X\big(f^{-1}(z)\big) \\ p_Z(z) &= \frac{dP_Z(z)}{dz} = \frac{dP_X(f^{-1}(z))}{dz} = \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{dx}{dz} \\ &= \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{d(f^{-1}(z))}{dz} \overset{(6)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{d(f^{-1}(z))}{dz}\right| \end{aligned}$$

Transport de mesures (dim. 1)

Si f est monotone croissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \leq f^{-1}(z) \tag{5}$$

$$\left(f^{-1}(z)\right)' \geq 0 \quad \Rightarrow \quad \left|\left(f^{-1}(z)\right)'\right| = \left(f^{-1}(z)\right)' \tag{6}$$

Ainsi

$$P_Z(z) \quad = \quad Pr(Z \leq z) \; = \; Pr\big(f(X) \leq z\big)$$

$$\overset{(5)}{=} \quad Pr\big(X \leq f^{-1}(z)\big) \; = \; P_X\big(f^{-1}(z)\big)$$

$$p_Z(z) \quad = \quad \frac{dP_Z(z)}{dz} \; = \; \frac{dP_X(f^{-1}(z))}{dz} \; = \; \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{dx}{dz}$$

$$= \quad \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{d(f^{-1}(z))}{dz} \quad \overset{(6)}{=} \quad p_X(f^{-1}(z)) \cdot \left|\frac{d(f^{-1}(z))}{dz}\right|$$

Transport de mesures (dim. 1)

Si f est monotone croissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \leq f^{-1}(z) \tag{5}$$

$$\left(f^{-1}(z)\right)' \geq 0 \quad \Rightarrow \quad \left|\left(f^{-1}(z)\right)'\right| = \left(f^{-1}(z)\right)' \tag{6}$$

Ainsi

$$\begin{aligned} P_Z(z) &= Pr(Z \leq z) = Pr\big(f(X) \leq z\big) \\ &\overset{(5)}{=} Pr\big(X \leq f^{-1}(z)\big) = P_X\big(f^{-1}(z)\big) \end{aligned}$$

$$\begin{aligned} p_Z(z) &= \frac{dP_Z(z)}{dz} = \frac{dP_X(f^{-1}(z))}{dz} = \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{dx}{dz} \\ &= \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{d(f^{-1}(z))}{dz} \overset{(6)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{d(f^{-1}(z))}{dz}\right| \end{aligned}$$

Transport de mesures (dim. 1)

Si f est monotone croissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \leq f^{-1}(z) \tag{5}$$

$$\left(f^{-1}(z)\right)' \geq 0 \quad \Rightarrow \quad \left|\left(f^{-1}(z)\right)'\right| = \left(f^{-1}(z)\right)' \tag{6}$$

Ainsi

$$\begin{aligned} P_Z(z) &= Pr(Z \leq z) = Pr\big(f(X) \leq z\big) \\ &\overset{(5)}{=} Pr\big(X \leq f^{-1}(z)\big) = P_X\big(f^{-1}(z)\big) \end{aligned}$$

$$\begin{aligned} p_Z(z) &= \frac{dP_Z(z)}{dz} = \frac{dP_X(f^{-1}(z))}{dz} = \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{dx}{dz} \\ &= \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{d(f^{-1}(z))}{dz} \overset{(6)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{d(f^{-1}(z))}{dz}\right| \end{aligned}$$

Transport de mesures (dim. 1)

Si f est monotone croissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \leq f^{-1}(z) \tag{5}$$

$$\left(f^{-1}(z)\right)' \geq 0 \quad \Rightarrow \quad \left|\left(f^{-1}(z)\right)'\right| = \left(f^{-1}(z)\right)' \tag{6}$$

Ainsi

$$\begin{aligned} P_Z(z) &= Pr(Z \leq z) = Pr\big(f(X) \leq z\big) \\ &\overset{(5)}{=} Pr\big(X \leq f^{-1}(z)\big) = P_X\big(f^{-1}(z)\big) \end{aligned}$$

$$\begin{aligned} p_Z(z) &= \frac{dP_Z(z)}{dz} = \frac{dP_X(f^{-1}(z))}{dz} = \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{dx}{dz} \\ &= \frac{dP_X(f^{-1}(z))}{dx} \cdot \frac{d(f^{-1}(z))}{dz} \overset{(6)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{d(f^{-1}(z))}{dz}\right| \end{aligned}$$

Transport de mesures (dim. 1)

Si f est monotone décroissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \geq f^{-1}(z) \tag{7}$$

$$(f^{-1}(z))' \leq 0 \quad \Rightarrow \quad \left|(f^{-1}(z))'\right| = -(f^{-1}(z))' \tag{8}$$

Ainsi

$$\begin{aligned} P_Z(z) &= Pr(Z \leq z) = Pr(f(X) \leq z) \\ &\overset{(7)}{=} Pr(X \geq f^{-1}(z)) = 1 - P_X(f^{-1}(z)) \end{aligned}$$

$$\begin{aligned} p_Z(z) &= \frac{dP_Z(z)}{dz} = \frac{d(1 - P_X(f^{-1}(z)))}{dz} \\ &= \frac{d(1 - P_X(f^{-1}(z)))}{dx} \cdot \frac{dx}{dz} = \frac{d(1 - P_X(f^{-1}(z)))}{dx} \cdot \frac{df^{-1}(z)}{dz} \\ &= -p_X(f^{-1}(z)) \cdot \frac{df^{-1}(z)}{dz} \overset{(8)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{df^{-1}(z)}{dz}\right| \end{aligned}$$

Transport de mesures (dim. 1)

Si f est monotone décroissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \geq f^{-1}(z) \tag{7}$$

$$\left(f^{-1}(z)\right)' \leq 0 \quad \Rightarrow \quad \left|\left(f^{-1}(z)\right)'\right| = -\left(f^{-1}(z)\right)' \tag{8}$$

Ainsi

$$\begin{aligned} P_Z(z) &= Pr(Z \leq z) = Pr\big(f(X) \leq z\big) \\ &\overset{(7)}{=} Pr\big(X \geq f^{-1}(z)\big) = 1 - P_X\big(f^{-1}(z)\big) \end{aligned}$$

$$\begin{aligned} p_Z(z) &= \frac{dP_Z(z)}{dz} = \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dz} \\ &= \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dx} \cdot \frac{dx}{dz} = \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dx} \cdot \frac{d\,f^{-1}(z)}{dz} \\ &= -p_X(f^{-1}(z)) \cdot \frac{d\,f^{-1}(z)}{dz} \overset{(8)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{d\,f^{-1}(z)}{dz}\right| \end{aligned}$$

Transport de mesures (dim. 1)

Si f est monotone décroissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \geq f^{-1}(z) \tag{7}$$

$$\left(f^{-1}(z)\right)' \leq 0 \quad \Rightarrow \quad \left|\left(f^{-1}(z)\right)'\right| = -\left(f^{-1}(z)\right)' \tag{8}$$

Ainsi

$$\begin{aligned}
P_Z(z) &= Pr(Z \leq z) = Pr\left(f(X) \leq z\right) \\
&\overset{(7)}{=} Pr\left(X \geq f^{-1}(z)\right) = 1 - P_X\left(f^{-1}(z)\right) \\
p_Z(z) &= \frac{dP_Z(z)}{dz} = \frac{d\left(1 - P_X(f^{-1}(z))\right)}{dz} \\
&= \frac{d\left(1 - P_X(f^{-1}(z))\right)}{dx} \cdot \frac{dx}{dz} = \frac{d\left(1 - P_X(f^{-1}(z))\right)}{dx} \cdot \frac{df^{-1}(z)}{dz} \\
&= -p_X(f^{-1}(z)) \cdot \frac{df^{-1}(z)}{dz} \overset{(8)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{df^{-1}(z)}{dz}\right|
\end{aligned}$$

Transport de mesures (dim. 1)

Si f est monotone décroissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \geq f^{-1}(z) \tag{7}$$

$$\left(f^{-1}(z)\right)' \leq 0 \quad \Rightarrow \quad \left|\left(f^{-1}(z)\right)'\right| = -\left(f^{-1}(z)\right)' \tag{8}$$

Ainsi

$$P_Z(z) = Pr(Z \leq z) = Pr\big(f(X) \leq z\big)$$

$$\overset{(7)}{=} Pr\big(X \geq f^{-1}(z)\big) = 1 - P_X\big(f^{-1}(z)\big)$$

$$p_Z(z) = \frac{dP_Z(z)}{dz} = \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dz}$$

$$= \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dx} \cdot \frac{dx}{dz} = \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dx} \cdot \frac{df^{-1}(z)}{dz}$$

$$= -p_X(f^{-1}(z)) \cdot \frac{df^{-1}(z)}{dz} \overset{(8)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{df^{-1}(z)}{dz}\right|$$

Transport de mesures (dim. 1)

Si f est monotone décroissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \geq f^{-1}(z) \tag{7}$$

$$\left(f^{-1}(z)\right)' \leq 0 \quad \Rightarrow \quad \left|\left(f^{-1}(z)\right)'\right| = -\left(f^{-1}(z)\right)' \tag{8}$$

Ainsi

$$\begin{aligned} P_Z(z) &= Pr(Z \leq z) = Pr\big(f(X) \leq z\big) \\ &\overset{(7)}{=} Pr\big(X \geq f^{-1}(z)\big) = 1 - P_X\big(f^{-1}(z)\big) \end{aligned}$$

$$\begin{aligned} p_Z(z) &= \frac{dP_Z(z)}{dz} = \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dz} \\ &= \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dx} \cdot \frac{dx}{dz} = \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dx} \cdot \frac{d\,f^{-1}(z)}{dz} \\ &= -p_X(f^{-1}(z)) \cdot \frac{d\,f^{-1}(z)}{dz} \overset{(8)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{d\,f^{-1}(z)}{dz}\right| \end{aligned}$$

Transport de mesures (dim. 1)

Si f est monotone décroissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \geq f^{-1}(z) \tag{7}$$

$$\left(f^{-1}(z)\right)' \leq 0 \quad \Rightarrow \quad \left|\left(f^{-1}(z)\right)'\right| = -\left(f^{-1}(z)\right)' \tag{8}$$

Ainsi

$$\begin{aligned} P_Z(z) &= Pr(Z \leq z) = Pr\big(f(X) \leq z\big) \\ &\overset{(7)}{=} Pr\big(X \geq f^{-1}(z)\big) = 1 - P_X\big(f^{-1}(z)\big) \end{aligned}$$

$$\begin{aligned} p_Z(z) &= \frac{dP_Z(z)}{dz} = \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dz} \\ &= \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dx} \cdot \frac{dx}{dz} = \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dx} \cdot \frac{d\,f^{-1}(z)}{dz} \\ &= -p_X(f^{-1}(z)) \cdot \frac{d\,f^{-1}(z)}{dz} \overset{(8)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{d\,f^{-1}(z)}{dz}\right| \end{aligned}$$

Transport de mesures (dim. 1)

Si f est monotone décroissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \geq f^{-1}(z) \tag{7}$$

$$\left(f^{-1}(z)\right)' \leq 0 \quad \Rightarrow \quad \left|\left(f^{-1}(z)\right)'\right| = -\left(f^{-1}(z)\right)' \tag{8}$$

Ainsi

$$\begin{aligned} P_Z(z) &= Pr(Z \leq z) = Pr\big(f(X) \leq z\big) \\ &\overset{(7)}{=} Pr\big(X \geq f^{-1}(z)\big) = 1 - P_X\big(f^{-1}(z)\big) \\ p_Z(z) &= \frac{dP_Z(z)}{dz} = \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dz} \\ &= \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dx} \cdot \frac{dx}{dz} = \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dx} \cdot \frac{d\,f^{-1}(z)}{dz} \\ &= -p_X(f^{-1}(z)) \cdot \frac{d\,f^{-1}(z)}{dz} \overset{(8)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{d\,f^{-1}(z)}{dz}\right| \end{aligned}$$

Transport de mesures (dim. 1)

Si f est monotone décroissante, alors f^{-1} également, et on a

$$f(X) \leq z \quad \Leftrightarrow \quad X \geq f^{-1}(z) \tag{7}$$

$$\left(f^{-1}(z)\right)' \leq 0 \quad \Rightarrow \quad \left|\left(f^{-1}(z)\right)'\right| = -\left(f^{-1}(z)\right)' \tag{8}$$

Ainsi

$$\begin{aligned}
P_Z(z) &= Pr(Z \leq z) = Pr\big(f(X) \leq z\big) \\
&\overset{(7)}{=} Pr\big(X \geq f^{-1}(z)\big) = 1 - P_X\big(f^{-1}(z)\big) \\
p_Z(z) &= \frac{dP_Z(z)}{dz} = \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dz} \\
&= \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dx} \cdot \frac{dx}{dz} = \frac{d\big(1 - P_X(f^{-1}(z))\big)}{dx} \cdot \frac{d\,f^{-1}(z)}{dz} \\
&= -p_X(f^{-1}(z)) \cdot \frac{d\,f^{-1}(z)}{dz} \overset{(8)}{=} p_X(f^{-1}(z)) \cdot \left|\frac{d\,f^{-1}(z)}{dz}\right|
\end{aligned}$$

Transport de mesures (dim. 1)

En résumé, on a le **théorème de changement de variable** suivant:

Theorem

Soient X une variable aléatoire sur $\mathbb{R}$, $f : \mathbb{R} \to \mathbb{R}$ une fonction différentiable bijective et $Z = f(X)$. La fonction de densité de Z est donnée par:

$$p_Z(z) = p_X(f^{-1}(z)) \cdot \left| \frac{d f^{-1}(z)}{dz} \right| \tag{9}$$

Transport de mesures (dim. 1)

En résumé, on a le **théorème de changement de variable** suivant:

Theorem

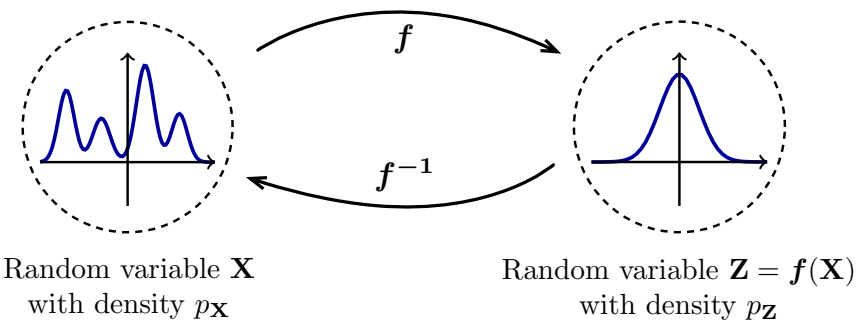
Soient X une variable aléatoire sur $\mathbb{R}$, $f : \mathbb{R} \to \mathbb{R}$ une fonction différentiable bijective et $Z = f(X)$. La fonction de densité de Z est donnée par:

$$p_Z(z) = p_X\left(f^{-1}(z)\right) \cdot \left| \frac{d\, f^{-1}(z)}{dz} \right| \tag{9}$$

Transport de mesures (dim. 1)

- Dans notre cas, on cherchera plutôt à exprimer p_X en fonction de p_Z, qui sera une loi normale centrés réduite (simple).
- En remplaçant X, Z et f par Z, X et f^{-1}, respectivement, dans le théorème, on a:

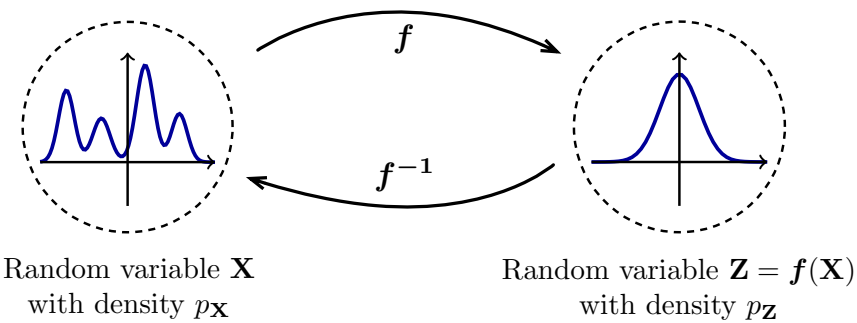
$$p_X(x) = p_Z\big(f(x)\big) \cdot \left|\frac{d\,f(x)}{dx}\right| = p_Z\big(f(x)\big) \cdot |J_f(x)| \tag{10}$$



Transport de mesures (dim. 1)

- Dans notre cas, on cherchera plutôt à exprimer p_X en fonction de p_Z, qui sera une loi normale centrés réduite (simple).
- En remplaçant X, Z et f par Z, X et f^{-1}, respectivement, dans le théorème, on a:

$$p_X(x) = p_Z\big(f(x)\big) \cdot \left| \frac{d\, f(x)}{dx} \right| = p_Z\big(f(x)\big) \cdot |J_f(x)| \tag{10}$$



Transport de mesures (dim. $d > 1$)

Dans le cas multidimensionnel, le **théorème de changement de variable** se généralise ainsi:

Theorem

Soient $\mathbf{X}$ *une variable aléatoire sur* $\mathbb{R}^d$, $f : \mathbb{R}^d \to \mathbb{R}^d$ *une fonction différentiable bijective et* $\mathbf{Z} = f(\mathbf{X})$. *La fonction de densité de* $\mathbf{Z}$ *est donnée par:*

$$p_{\mathbf{Z}}(\mathbf{z}) = p_{\mathbf{X}}(f^{-1}(\mathbf{z})) \cdot \left| \det \left(\frac{\partial f^{-1}(\mathbf{z})}{\partial \mathbf{z}} \right) \right|$$

où $\frac{\partial f^{-1}(\mathbf{z})}{\partial \mathbf{z}} = \begin{bmatrix} \frac{\partial f_1^{-1}(\mathbf{z})}{\partial z_1} & \cdots & \frac{\partial f_1^{-1}(\mathbf{z})}{\partial z_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_d^{-1}(\mathbf{z})}{\partial z_1} & \cdots & \frac{\partial f_d^{-1}(\mathbf{z})}{\partial z_d} \end{bmatrix}$ *est le Jacobien de* $f^{-1}(\mathbf{z})$.

Transport de mesures (dim. $d > 1$)

Dans le cas multidimensionnel, le **théorème de changement de variable** se généralise ainsi:

Theorem

Soient $\mathbf{X}$ *une variable aléatoire sur* $\mathbb{R}^d$, $f : \mathbb{R}^d \to \mathbb{R}^d$ *une fonction différentiable bijective et* $\mathbf{Z} = f(\mathbf{X})$. *La fonction de densité de* $\mathbf{Z}$ *est donnée par:*

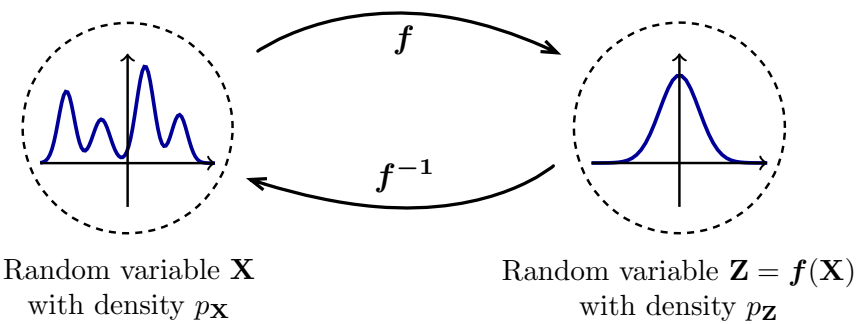
$$p_{\mathbf{Z}}(\mathbf{z}) \;=\; p_{\mathbf{X}}\left(f^{-1}(\mathbf{z})\right) \cdot \left|\det\left(\frac{\partial\, f^{-1}(\mathbf{z})}{\partial\, \mathbf{z}}\right)\right|$$

où $\frac{\partial\, f^{-1}(\mathbf{z})}{\partial\, \mathbf{z}} = \begin{bmatrix} \frac{\partial f_1^{-1}(\mathbf{z})}{\partial z_1} & \cdots & \frac{\partial f_1^{-1}(\mathbf{z})}{\partial z_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_d^{-1}(\mathbf{z})}{\partial z_1} & \cdots & \frac{\partial f_d^{-1}(\mathbf{z})}{\partial z_d} \end{bmatrix}$ *est le Jacobien de* $f^{-1}(\mathbf{z})$.

Transport de mesures (dim. $d > 1$)

- Dans notre cas, on cherchera plutôt à exprimer $p_{\mathbf{X}}$ en fonction de $p_{\mathbf{Z}}$, qui sera une loi normale centrée réduite (simple).
- En remplaçant $\mathbf{X}$, $\mathbf{Z}$ et f par $\mathbf{Z}$, $\mathbf{X}$ et f^{-1}, respectivement, dans le théorème, on a:

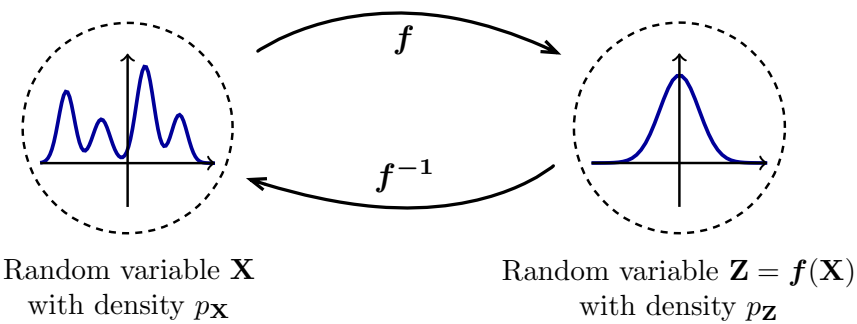
$$p_{\mathbf{X}}(\mathbf{x}) = p_{\mathbf{Z}}\big(f(\mathbf{x})\big) \cdot \left|\det\left(\frac{d\, f(\mathbf{x})}{d\, \mathbf{x}}\right)\right| = p_{\mathbf{Z}}\big(f(\mathbf{x})\big) \cdot \left|\det\left(J_f(\mathbf{x})\right)\right| \quad (11)$$



Transport de mesures (dim. $d > 1$)

- Dans notre cas, on cherchera plutôt à exprimer $p_{\mathbf{X}}$ en fonction de $p_{\mathbf{Z}}$, qui sera une loi normale centrée réduite (simple).
- En remplaçant $\mathbf{X}$, $\mathbf{Z}$ et f par $\mathbf{Z}$, $\mathbf{X}$ et f^{-1}, respectivement, dans le théorème, on a:

$$p_{\mathbf{X}}(\mathbf{x}) \;=\; p_{\mathbf{Z}}\big(f(\mathbf{x})\big) \cdot \left|\det\left(\frac{d\, f(\mathbf{x})}{d\, \mathbf{x}}\right)\right| \;=\; p_{\mathbf{Z}}\big(f(\mathbf{x})\big) \cdot \left|\det\left(J_f(\mathbf{x})\right)\right| \qquad (11)$$



Normalizing Flows

- Les modèles de **flots génératifs (normalizing flows, flow-based generative models)**, en particulier les **réseaux de neurones inversibles**, utilisent le *théorème du changement de variable* pour:

1. apprendre la distribution des data plus efficacement ;
2. générer des data.

Réseaux de neurones inversibles (INNs)

- Un **réseau de neurones inversible (invertible neural networks, INN)** est un réseau de neurones dont la fonction associée

$$f_\theta : \mathbb{R}^d \to \mathbb{R}^d$$

satisfait les propriétés suivantes:

1. f_θ est bijective ;
2. l'inverse f_θ^{-1} est facile à calculer ;
3. le Jacobien $\frac{\partial f_\theta}{\partial \mathbf{x}}$ et son déterminant sont faciles à calculer.

Réseaux de neurones inversibles (INNs)

- Un **réseau de neurones inversible (invertible neural networks, INN)** est un réseau de neurones dont la fonction associée

$$f_\theta : \mathbb{R}^d \to \mathbb{R}^d$$

satisfait les propriétés suivantes:

1. f_θ est bijective ;
2. l'inverse f_θ^{-1} est facile à calculer ;
3. le Jacobien $\frac{\partial f_\theta}{\partial \mathbf{x}}$ et son déterminant sont faciles à calculer.

Réseaux de neurones inversibles (INNs)

- Un **réseau de neurones inversible (invertible neural networks, INN)** est un réseau de neurones dont la fonction associée

$$f_\theta : \mathbb{R}^d \to \mathbb{R}^d$$

satisfait les propriétés suivantes:

1. f_θ est bijective ;
2. l'inverse f_θ^{-1} est facile à calculer ;
3. le Jacobien $\frac{\partial f_\theta}{\partial \mathbf{x}}$ et son déterminant sont faciles à calculer.

Réseaux de neurones inversibles (INNs)

- Un **réseau de neurones inversible (invertible neural networks, INN)** est un réseau de neurones dont la fonction associée

$$f_\theta : \mathbb{R}^d \to \mathbb{R}^d$$

satisfait les propriétés suivantes:

1. f_θ est bijective ;
2. l'inverse f_θ^{-1} est facile à calculer ;
3. le Jacobien $\frac{\partial f_\theta}{\partial \mathbf{x}}$ et son déterminant sont faciles à calculer.

INN – RealNVP

- Un **real-valued non-volume preserving network (RealNVP)** est un INNs composé de couches bijectives (inversibles) $l_i : \mathbb{R}^d \to \mathbb{R}^d$ appelées "coupling layers" (dim. d conservée).

  [Dinh et al., 2017]

- Le réseau est entraîné de telle sorte que la composition de toutes ses couches $f_\theta = l_n \circ \cdots \circ l_1$ réalise pas à pas (interpolation) un *transport de mesure inversible* :

  les data originales sont transportées sur une distribution normale centrée réduite.

- Ensuite, pour *générer* de nouvelles data, on sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ et on applique la transformation inverse

$$\mathbf{x_{new}} = f_\theta^{-1}(\mathbf{z}) = l_1^{-1} \circ \cdots \circ l_n^{-1}(\mathbf{z})$$

INN – RealNVP

- Un **real-valued non-volume preserving network (RealNVP)** est un INNs composé de couches bijectives (inversibles) $l_i : \mathbb{R}^d \to \mathbb{R}^d$ appelées "coupling layers" (dim. d conservée).

  [Dinh et al., 2017]

- Le réseau est entraîné de telle sorte que la composition de toutes ses couches $f_\theta = l_n \circ \cdots \circ l_1$ réalise pas à pas (interpolation) un *transport de mesure inversible* :

  les data originales sont transportées sur une distribution normale centrée réduite.

- Ensuite, pour *générer* de nouvelles data, on sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ et on applique la transformation inverse

$$\mathbf{x_{new}} = f_\theta^{-1}(\mathbf{z}) = l_1^{-1} \circ \cdots \circ l_n^{-1}(\mathbf{z})$$

INN – RealNVP

- Un **real-valued non-volume preserving network (RealNVP)** est un INNs composé de couches bijectives (inversibles) $l_i : \mathbb{R}^d \to \mathbb{R}^d$ appelées "coupling layers" (dim. d conservée).

  [Dinh et al., 2017]

- Le réseau est entraîné de telle sorte que la composition de toutes ses couches $f_\theta = l_n \circ \cdots \circ l_1$ réalise pas à pas (interpolation) un *transport de mesure inversible* :

  les data originales sont transportées sur une distribution normale centrée réduite.

- Ensuite, pour *générer* de nouvelles data, on sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ et on applique la transformation inverse

$$\mathbf{x_{new}} = f_\theta^{-1}(\mathbf{z}) = l_1^{-1} \circ \cdots \circ l_n^{-1}(\mathbf{z})$$

INN – RealNVP

- Un **real-valued non-volume preserving network (RealNVP)** est un INNs composé de couches bijectives (inversibles) $l_i : \mathbb{R}^d \to \mathbb{R}^d$ appelées "coupling layers" (dim. d conservée).

  [Dinh et al., 2017]

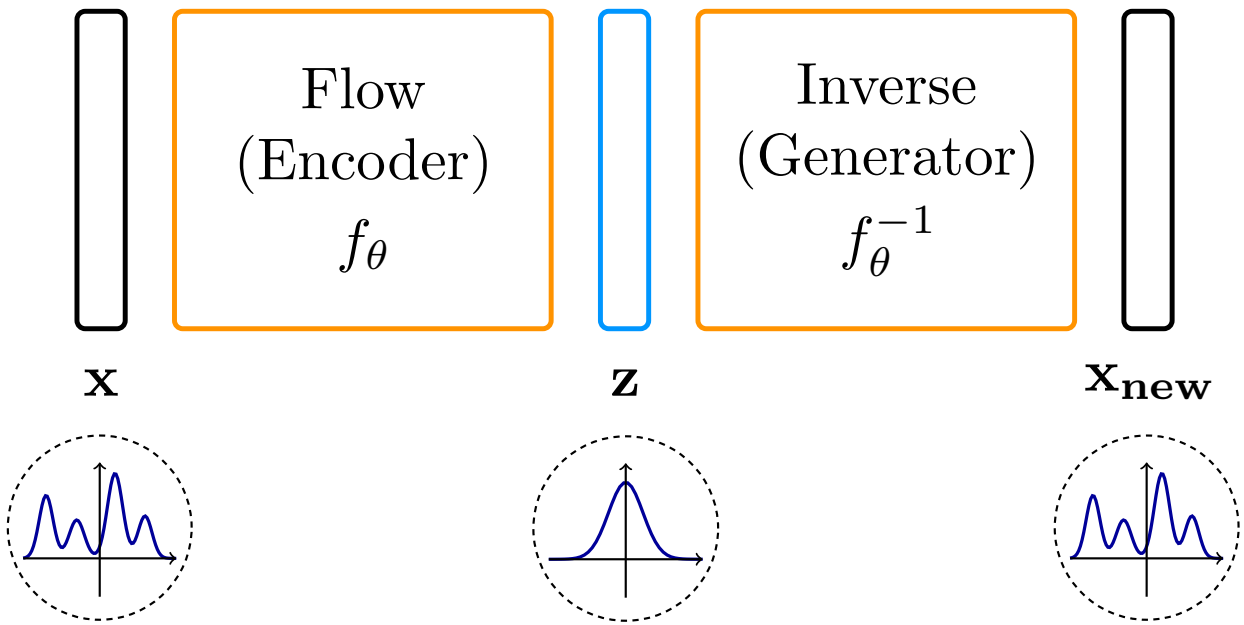
- Le réseau est entraîné de telle sorte que la composition de toutes ses couches $f_\theta = l_n \circ \cdots \circ l_1$ réalise pas à pas (interpolation) un *transport de mesure inversible* :

  les data originales sont transportées sur une distribution normale centrée réduite.

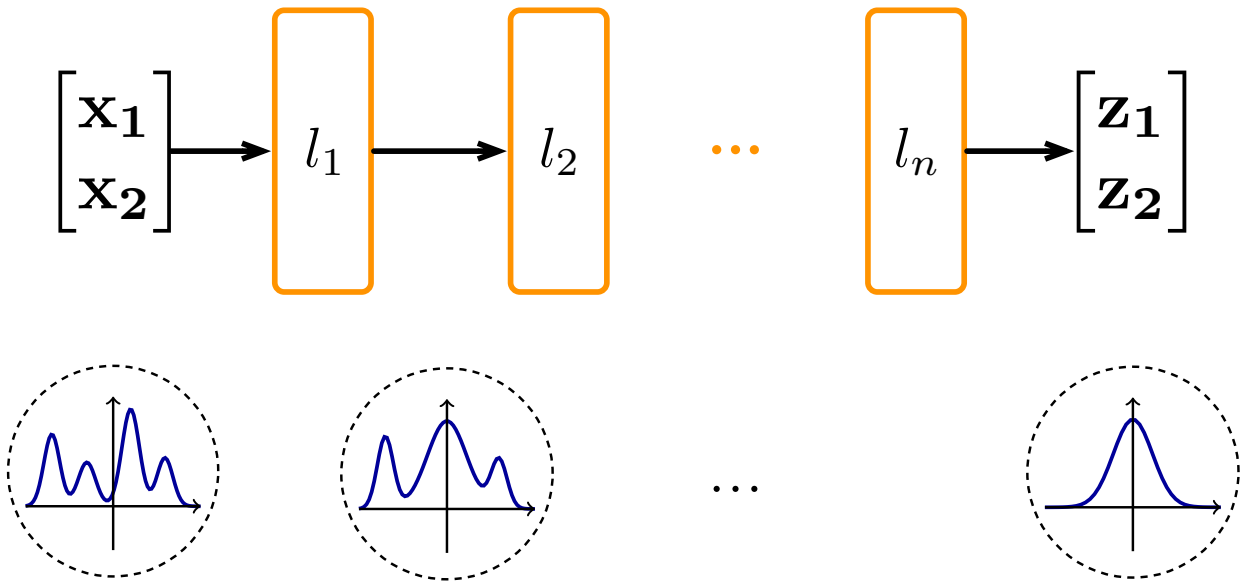
- Ensuite, pour *générer* de nouvelles data, on sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ et on applique la transformation inverse

$$\mathbf{x_{new}} = f_\theta^{-1}(\mathbf{z}) = l_1^{-1} \circ \cdots \circ l_n^{-1}(\mathbf{z})$$

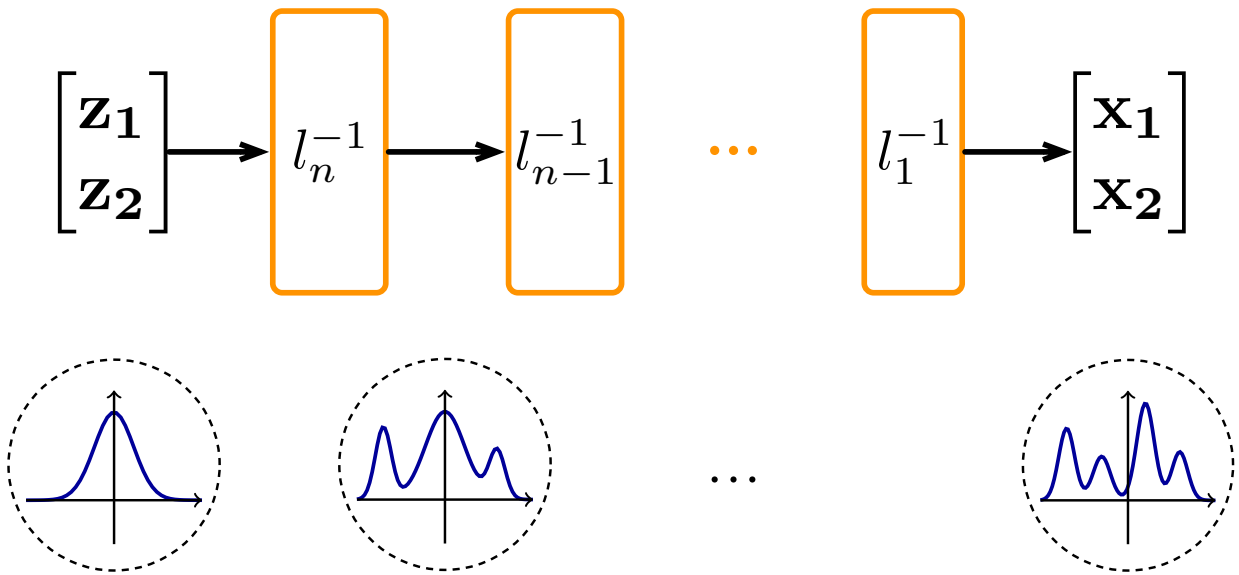
INN – RealNVP


Flow
(Encoder)
f_θ
Inverse
(Generator)
f_θ^{-1}
$\mathbf{x}$
$\mathbf{z}$
$\mathbf{x_{new}}$

INN – RealNVP



INN – RealNVP



INN – RealNVP: layer

- Chaque couche $l_i : \mathbb{R}^d \to \mathbb{R}^d$ réalise une transformation bijective facile à inverser.
- L'idée est de séparer l'input en deux parties $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}] \in \mathbb{R}^d$, où $\mathbf{x_1} \in \mathbb{R}^{k_i}$ et $\mathbf{x_2} \in \mathbb{R}^{d-k_i}$.
- La couche $l_i : \mathbb{R}^d \to \mathbb{R}^d$ implémente la transformation affine suivante:

$$\begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} \overset{l_i}{\longmapsto} \begin{bmatrix} \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix} = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \odot \exp\left(s_i(\mathbf{x_1})\right) + t_i(\mathbf{x_1}) \end{bmatrix}$$

où s_i (scale function) et t_i (translation function) sont des réseaux de neurones simples.

INN – RealNVP: layer

- Chaque couche $l_i : \mathbb{R}^d \to \mathbb{R}^d$ réalise une transformation bijective facile à inverser.
- L'idée est de séparer l'input en deux parties $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}] \in \mathbb{R}^d$, où $\mathbf{x_1} \in \mathbb{R}^{k_i}$ et $\mathbf{x_2} \in \mathbb{R}^{d-k_i}$.
- La couche $l_i : \mathbb{R}^d \to \mathbb{R}^d$ implémente la transformation affine suivante:

$$\begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} \overset{l_i}{\longmapsto} \begin{bmatrix} \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix} = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \odot \exp\left(s_i(\mathbf{x_1})\right) + t_i(\mathbf{x_1}) \end{bmatrix}$$

où s_i (scale function) et t_i (translation function) sont des réseaux de neurones simples.

INN – RealNVP: layer

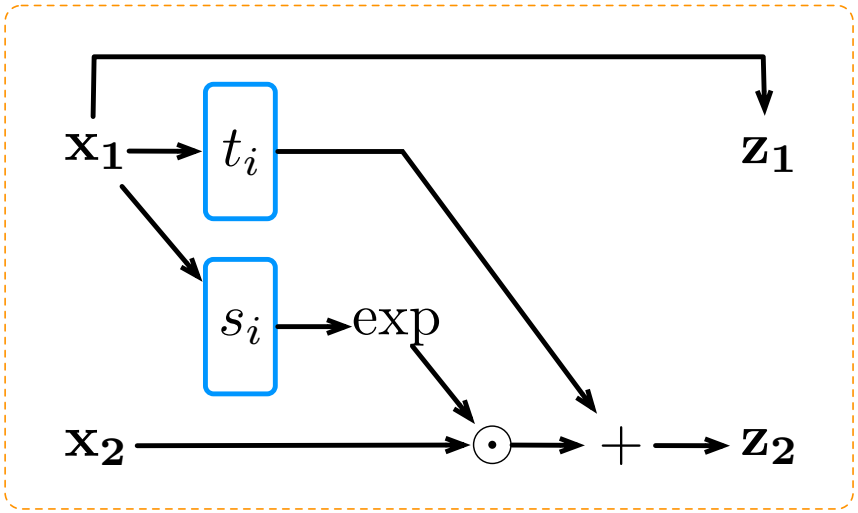
- Chaque couche $l_i : \mathbb{R}^d \to \mathbb{R}^d$ réalise une transformation bijective facile à inverser.
- L'idée est de séparer l'input en deux parties $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}] \in \mathbb{R}^d$, où $\mathbf{x_1} \in \mathbb{R}^{k_i}$ et $\mathbf{x_2} \in \mathbb{R}^{d-k_i}$.
- La couche $l_i : \mathbb{R}^d \to \mathbb{R}^d$ implémente la transformation affine suivante:

$$\begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} \overset{l_i}{\longmapsto} \begin{bmatrix} \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix} = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \odot \exp\big(s_i(\mathbf{x_1})\big) + t_i\big(\mathbf{x_1}\big) \end{bmatrix}$$

où s_i (scale function) et t_i (translation function) sont des réseaux de neurones simples.

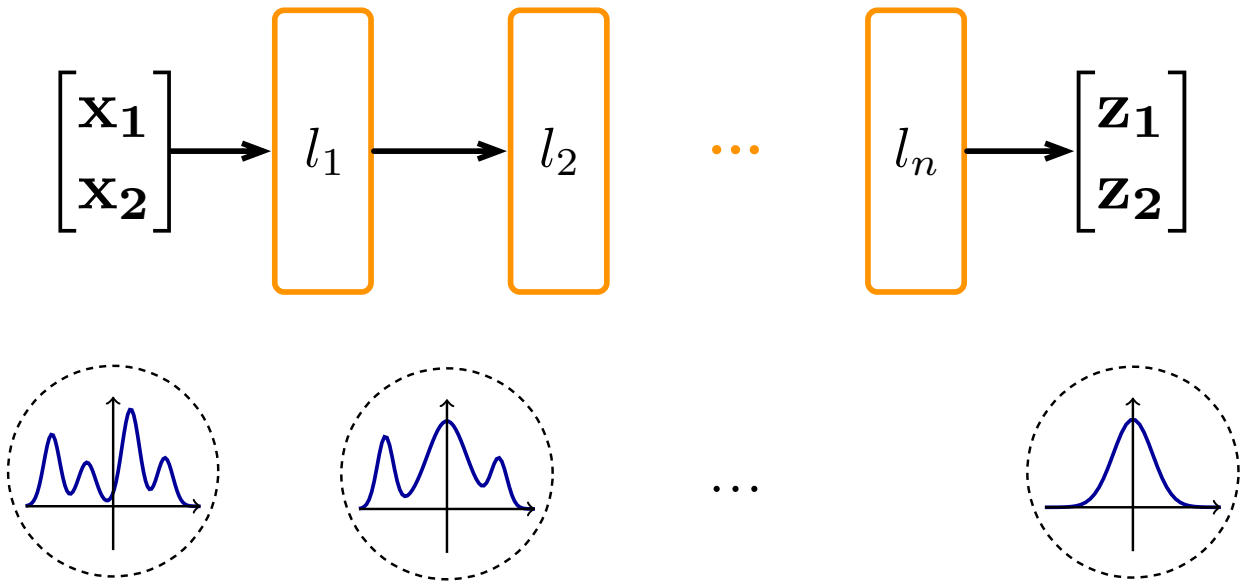
INN – RealNVP: layer

$$\begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} \overset{l_i}{\longmapsto} \begin{bmatrix} \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix} = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \odot \exp\left(s_i(\mathbf{x_1})\right) + t_i(\mathbf{x_1}) \end{bmatrix}$$



layer l_i

INN – RealNVP: full network



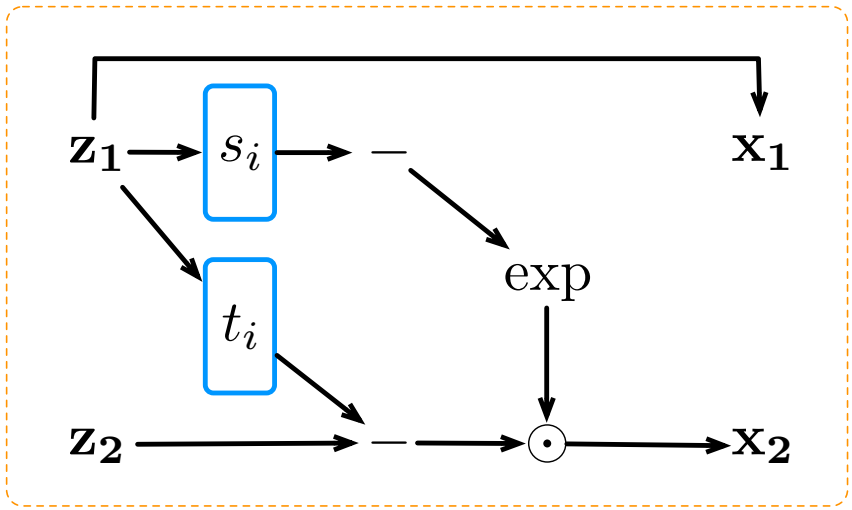
INN – RealNVP: inverse layer

- Grâce à cette architecture, la transformation inverse de chaque couche l_i, notée $l_i^{-1} : \mathbb{R}^d \to \mathbb{R}^d$, peut se calculer simplement:

$$\begin{bmatrix} \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix} \overset{l_i^{-1}}{\longmapsto} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} = \begin{bmatrix} \mathbf{y_1} \\ \big(\mathbf{z_2} - t_i(\mathbf{z_1})\big) \odot \exp\big(-s_i(\mathbf{z_1})\big) \end{bmatrix}$$

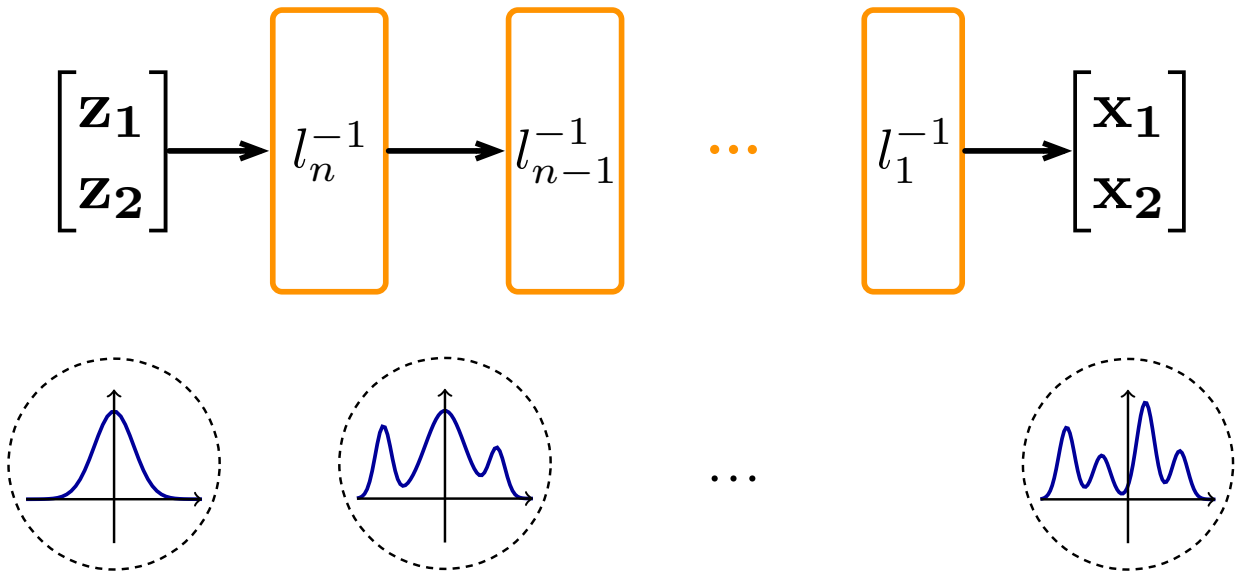
INN – RealNVP: inverse layer

$$\begin{bmatrix} \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix} \overset{l_i^{-1}}{\longmapsto} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} = \begin{bmatrix} \mathbf{z_1} \\ \left(\mathbf{z_2} - t_i(\mathbf{z_1})\right) \odot \exp\left(-s_i(\mathbf{z_1})\right) \end{bmatrix}$$



layer l_i^{-1}

INN – RealNVP: full inverse network



Calcul du Jacobien (1 couche)

- Le Jacobien associé à chaque couche l_i est une matrice triangulaire inférieure :

$$\mathbf{z} \quad = \quad l_i(\mathbf{x}) = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \odot \exp\left(s_i(\mathbf{x_1})\right) + t_i(\mathbf{x_1}) \end{bmatrix}$$

$$J_{l_i} \quad := \quad \frac{\partial\, l_i(\mathbf{x})}{\partial\, \mathbf{x}} = \frac{\partial\, \mathbf{z}}{\partial\, \mathbf{x}} = \begin{bmatrix} \mathbf{I}_{k_i} & \mathbf{0} \\ \frac{\partial\, \mathbf{z_2}}{\partial\, \mathbf{x_1}} & \operatorname{diag}\left[\exp\left(s_i(\mathbf{x_1})\right)\right] \end{bmatrix}$$

où le terme $\frac{\partial\, \mathbf{z_2}}{\partial\, \mathbf{x_1}}$ est non trivial (ce qui ne pose aucun problème).

- Ainsi, le déterminant du Jacobien se calcule simplement :

$$|\det(J_{l_i})| = \left|\det\left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right)\right| = \exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x_1})\right) \tag{12}$$

Calcul du Jacobien (1 couche)

- Le Jacobien associé à chaque couche l_i est une matrice triangulaire inférieure :

$$\mathbf{z} \quad = \quad l_i(\mathbf{x}) = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \odot \exp\left(s_i(\mathbf{x_1})\right) + t_i(\mathbf{x_1}) \end{bmatrix}$$

$$J_{l_i} \quad := \quad \frac{\partial\, l_i(\mathbf{x})}{\partial\, \mathbf{x}} = \frac{\partial\, \mathbf{z}}{\partial\, \mathbf{x}} = \begin{bmatrix} \mathbf{I}_{k_i} & \mathbf{0} \\ \frac{\partial\, \mathbf{z_2}}{\partial\, \mathbf{x_1}} & \operatorname{diag}\left[\exp\left(s_i(\mathbf{x_1})\right)\right] \end{bmatrix}$$

où le terme $\frac{\partial\, \mathbf{z_2}}{\partial\, \mathbf{x_1}}$ est non trivial (ce qui ne pose aucun problème).

- Ainsi, le déterminant du Jacobien se calcule simplement :

$$|\det(J_{l_i})| = \left|\det\left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right)\right| = \exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x_1})\right) \tag{12}$$

Calcul du Jacobien (réseau total)

- La transformation totale du RealNVP est donnée par

$$\begin{aligned} f_\theta(\mathbf{x}) &= l_n \circ l_{n-1} \circ \cdots \circ l_1(\mathbf{x}) = \mathbf{z} \\ \mathbf{x}_0 &= \mathbf{x}, \quad \mathbf{x}_1 = l_1(\mathbf{x}_0), \ \ldots \ , \mathbf{z} = \mathbf{x}_n = l_n(\mathbf{x}_{n-1}) \end{aligned}$$

- Pour cette transformation totale, en appliquant la 'chain rule', le Jacobien et son déterminant sont donnés par :

$$\begin{aligned} J_{f_\theta}(\mathbf{x}) &= J_{l_n}(\mathbf{x}_{n-1}) \cdots J_{l_2}(\mathbf{x}_1) J_{l_1}(\mathbf{x}_0) \\ |\det J_{f_\theta}(\mathbf{x})| &= \prod_{i=1}^{n} |\det J_{l_i}(\mathbf{x}_{i-1})| \overset{(12)}{=} \prod_{i=1}^{n} \exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x}_1) \right) \quad (13) \end{aligned}$$

Calcul du Jacobien (réseau total)

- La transformation totale du RealNVP est donnée par

$$\begin{aligned} f_\theta(\mathbf{x}) &= l_n \circ l_{n-1} \circ \cdots \circ l_1(\mathbf{x}) = \mathbf{z} \\ \mathbf{x}_0 &= \mathbf{x}, \quad \mathbf{x}_1 = l_1(\mathbf{x}_0), \ \ldots \ , \mathbf{z} = \mathbf{x}_n = l_n(\mathbf{x}_{n-1}) \end{aligned}$$

- Pour cette transformation totale, en appliquant la 'chain rule', le Jacobien et son déterminant sont donnés par :

$$\begin{aligned} J_{f_\theta}(\mathbf{x}) &= J_{l_n}(\mathbf{x}_{n-1}) \cdots J_{l_2}(\mathbf{x}_1) J_{l_1}(\mathbf{x}_0) \\ |\det J_{f_\theta}(\mathbf{x})| &= \prod_{i=1}^{n} |\det J_{l_i}(\mathbf{x}_{i-1})| \overset{(12)}{=} \prod_{i=1}^{n} \exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x_1}) \right) \quad (13) \end{aligned}$$

INN – Entraînement

- Soit un dataset $\mathcal{D} = \{\mathbf{x_i} \in \mathbb{R}^d : i = 1, \ldots, n\}$ dont les éléments proviennent d'une distribution empirique p.
- Soit la distribution normale centrée réduite $p_{\mathbf{Z}}$.
- Soit un réseau de neurones inversible $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$.
- Par le *théorème de changement de variable*, la distribution $p_{\mathbf{X}}$ qui est *transportée* par f_θ sur $p_{\mathbf{Z}}$ est donnée par :

$$p_{\mathbf{X}}(\mathbf{x}) \overset{(11)}{=} p_{\mathbf{Z}}\left(f_\theta(\mathbf{x})\right) \cdot \left|\det\left(J_{f_\theta}(\mathbf{x})\right)\right|$$
$$\overset{(13)}{=} p_{\mathbf{Z}}\left(f_\theta(\mathbf{x})\right) \cdot \prod_{i=1}^{n} \exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x_1})\right) \tag{14}$$

INN – Entraînement

- Soit un dataset $\mathcal{D} = \{\mathbf{x_i} \in \mathbb{R}^d : i = 1, \ldots, n\}$ dont les éléments proviennent d'une distribution empirique p.
- Soit la distribution normale centrée réduite $p_{\mathbf{Z}}$.
- Soit un réseau de neurones inversible $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$.
- Par le *théorème de changement de variable*, la distribution $p_{\mathbf{X}}$ qui est *transportée* par f_θ sur $p_{\mathbf{Z}}$ est donnée par :

$$p_{\mathbf{X}}(\mathbf{x}) \overset{(11)}{=} p_{\mathbf{Z}}\big(f_\theta(\mathbf{x})\big) \cdot \left|\det\left(J_{f_\theta}(\mathbf{x})\right)\right|$$
$$\overset{(13)}{=} p_{\mathbf{Z}}\big(f_\theta(\mathbf{x})\big) \cdot \prod_{i=1}^{n} \exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x_1})\right) \tag{14}$$

INN – Entraînement

- Soit un dataset $\mathcal{D} = \{\mathbf{x_i} \in \mathbb{R}^d : i = 1, \ldots, n\}$ dont les éléments proviennent d'une distribution empirique p.
- Soit la distribution normale centrée réduite $p_{\mathbf{Z}}$.
- Soit un réseau de neurones inversible $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$.
- Par le *théorème de changement de variable*, la distribution $p_{\mathbf{X}}$ qui est *transportée* par f_θ sur $p_{\mathbf{Z}}$ est donnée par :

$$p_{\mathbf{X}}(\mathbf{x}) \overset{(11)}{=} p_{\mathbf{Z}}\big(f_\theta(\mathbf{x})\big) \cdot \left|\det\left(J_{f_\theta}(\mathbf{x})\right)\right|$$

$$\overset{(13)}{=} p_{\mathbf{Z}}\big(f_\theta(\mathbf{x})\big) \cdot \prod_{i=1}^{n} \exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x_1})\right) \tag{14}$$

INN – Entraînement

- Soit un dataset $\mathcal{D} = \{\mathbf{x_i} \in \mathbb{R}^d : i = 1, \ldots, n\}$ dont les éléments proviennent d'une distribution empirique p.
- Soit la distribution normale centrée réduite $p_{\mathbf{Z}}$.
- Soit un réseau de neurones inversible $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$.
- Par le *théorème de changement de variable*, la distribution $p_{\mathbf{X}}$ qui est *transportée* par f_θ sur $p_{\mathbf{Z}}$ est donnée par :

$$\begin{aligned} p_{\mathbf{X}}(\mathbf{x}) &\overset{(11)}{=} p_{\mathbf{Z}}\big(f_\theta(\mathbf{x})\big) \cdot \left|\det\left(J_{f_\theta}(\mathbf{x})\right)\right| \\ &\overset{(13)}{=} p_{\mathbf{Z}}\big(f_\theta(\mathbf{x})\big) \cdot \prod_{i=1}^{n} \exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x_1})\right) \end{aligned} \tag{14}$$

INN – Entraînement

- On entraîne f_θ pour que la distribution $p_{\mathbf{X}}$ qui est transportée par f_θ sur $p_{\mathbf{Z}}$ soit le plus proche possible de la distribution empirique p, i.e., $p_{\mathbf{X}}(\mathbf{x}) \simeq p(\mathbf{x}),\ \forall x \in \mathbb{R}^d$.
- On se ramène alors à un problème de d'*apprentissage de la distribution* p via $p_{\mathbf{X}}$.
- Pour cela, on minimise la negative log likelihood (NLL) :

$$\begin{aligned}
\mathcal{L}(\theta) &\overset{(4)}{=} -\frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} \log\left(p_{\mathbf{X}}(\mathbf{x})\right) \\
&\overset{(14)}{=} -\frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} \log\left(p_{\mathbf{Z}}(f_\theta(\mathbf{x})) \cdot \prod_{i=1}^{n} \exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x_1})\right)\right) \\
&= -\frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} \left[\log\left(p_{\mathbf{Z}}(f_\theta(\mathbf{x}))\right) + \sum_{i=1}^{n} \log\left(\exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x_1})\right)\right)\right]
\end{aligned}$$

INN – Entraînement

- On entraîne f_θ pour que la distribution $p_{\mathbf{X}}$ qui est transportée par f_θ sur $p_{\mathbf{Z}}$ soit le plus proche possible de la distribution empirique p, i.e., $p_{\mathbf{X}}(\mathbf{x}) \simeq p(\mathbf{x}),\ \forall x \in \mathbb{R}^d$.
- On se ramène alors à un problème de d'*apprentissage de la distribution* p via $p_{\mathbf{X}}$.
- Pour cela, on minimise la negative log likelihood (NLL) :

$$\begin{aligned}
\mathcal{L}(\theta) &\overset{(4)}{=} -\frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} \log\left(p_{\mathbf{X}}(\mathbf{x})\right) \\
&\overset{(14)}{=} -\frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} \log\left(p_{\mathbf{Z}}(f_\theta(\mathbf{x})) \cdot \prod_{i=1}^{n} \exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x_1})\right)\right) \\
&= -\frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} \left[\log\left(p_{\mathbf{Z}}(f_\theta(\mathbf{x}))\right) + \sum_{i=1}^{n} \log\left(\exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x_1})\right)\right)\right]
\end{aligned}$$

INN – Entraînement

- On entraîne f_θ pour que la distribution $p_{\mathbf{X}}$ qui est transportée par f_θ sur $p_{\mathbf{Z}}$ soit le plus proche possible de la distribution empirique p, i.e., $p_{\mathbf{X}}(\mathbf{x}) \simeq p(\mathbf{x}),\ \forall x \in \mathbb{R}^d$.
- On se ramène alors à un problème de d'*apprentissage de la distribution* p via $p_{\mathbf{X}}$.
- Pour cela, on minimise la negative log likelihood (NLL) :

$$
\begin{aligned}
\mathcal{L}(\theta) &\overset{(4)}{=} -\frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} \log\left(p_{\mathbf{X}}(\mathbf{x})\right) \\
&\overset{(14)}{=} -\frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} \log\left(p_{\mathbf{Z}}\left(f_\theta(\mathbf{x})\right)\cdot\prod_{i=1}^{n}\exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x_1})\right)\right) \\
&= -\frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} \left[\log\left(p_{\mathbf{Z}}\left(f_\theta(\mathbf{x})\right)\right) + \sum_{i=1}^{n}\log\left(\exp\left(\sum_{i=1}^{d-k_i} s_i(\mathbf{x_1})\right)\right)\right]
\end{aligned}
$$

INN – Génération

- Le réseau de neurones inversible $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ peut ensuite être utilisé pour la *génération* de data.
- Pour générer une data $\mathbf{x_{new}}$:

1. On sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
2. On applique la transformation inverse

$$\mathbf{x_{new}} = f_\theta^{-1}(\mathbf{z}) = l_1^{-1} \circ \cdots \circ l_n^{-1}(\mathbf{z})$$

INN – Génération

- Le réseau de neurones inversible $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ peut ensuite être utilisé pour la *génération* de data.
- Pour générer une data $\mathbf{x_{new}}$:

1. On sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
2. On applique la transformation inverse

$$\mathbf{x_{new}} = f_\theta^{-1}(\mathbf{z}) = l_1^{-1} \circ \cdots \circ l_n^{-1}(\mathbf{z})$$

INN – Génération

- Le réseau de neurones inversible $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ peut ensuite être utilisé pour la *génération* de data.
- Pour générer une data $\mathbf{x_{new}}$:

1. On sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
2. On applique la transformation inverse

$$\mathbf{x_{new}} = f_\theta^{-1}(\mathbf{z}) = l_1^{-1} \circ \cdots \circ l_n^{-1}(\mathbf{z})$$

INN – Génération

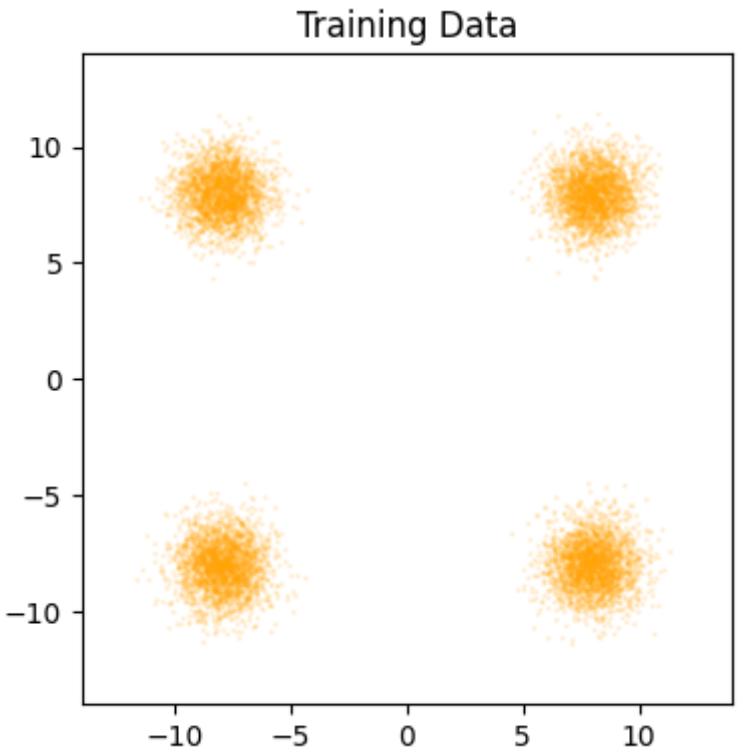
- Le réseau de neurones inversible $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ peut ensuite être utilisé pour la *génération* de data.
- Pour générer une data $\mathbf{x_{new}}$:

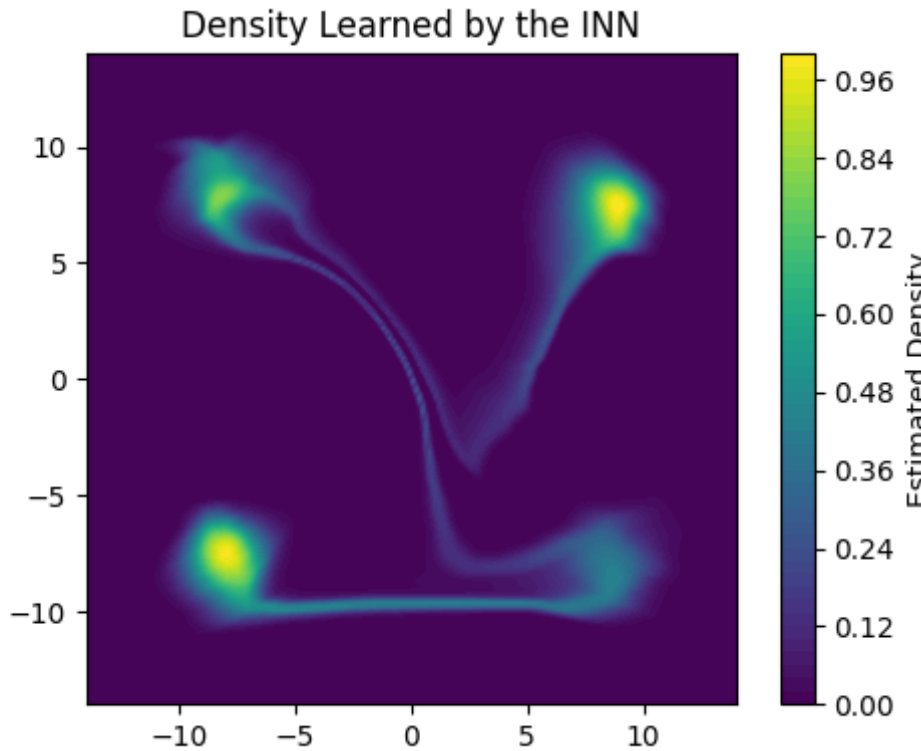
1. On sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
2. On applique la transformation inverse

$$\mathbf{x_{new}} = f_\theta^{-1}(\mathbf{z}) = l_1^{-1} \circ \cdots \circ l_n^{-1}(\mathbf{z})$$

Exemple

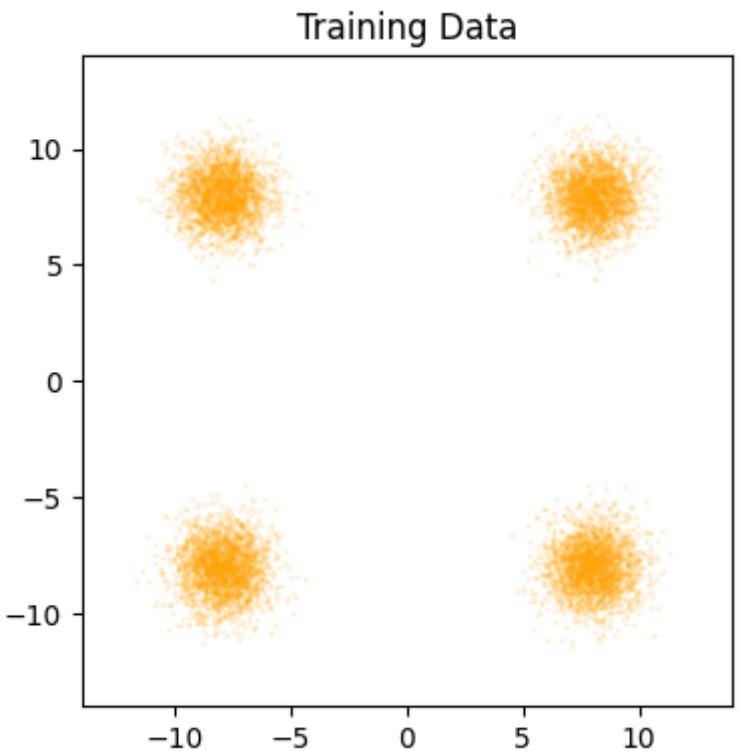
- Apprentissage d'une distributions multimodales (4 modes) par un RealNVP.
- On entraîne f_θ comme décrit précédemment et on utilise ensuite la formule (14) pour évaluer la densité des points du plan.

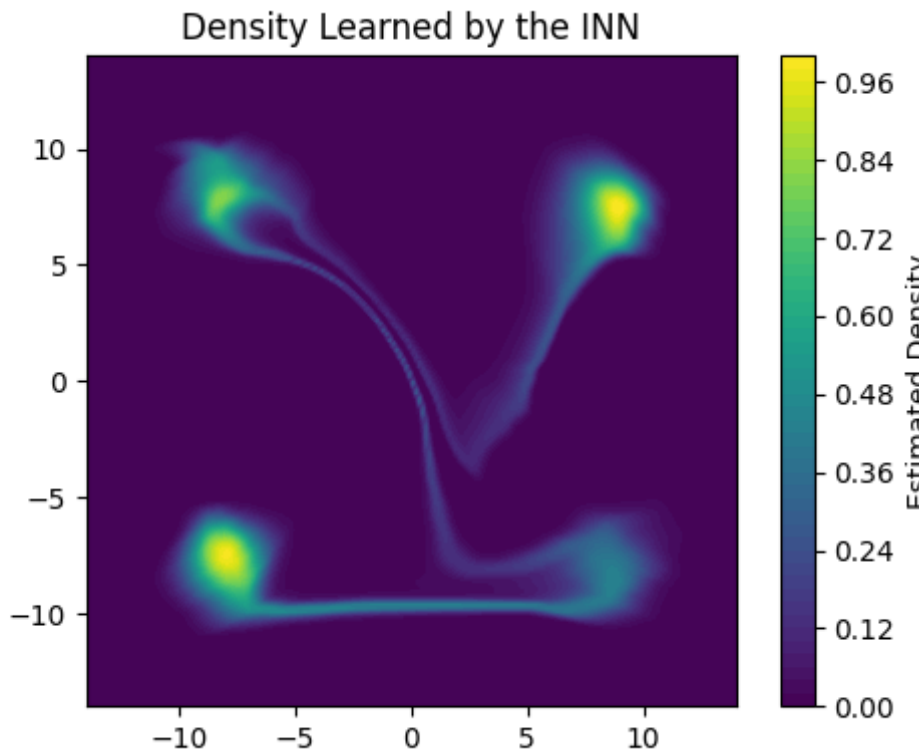




Exemple

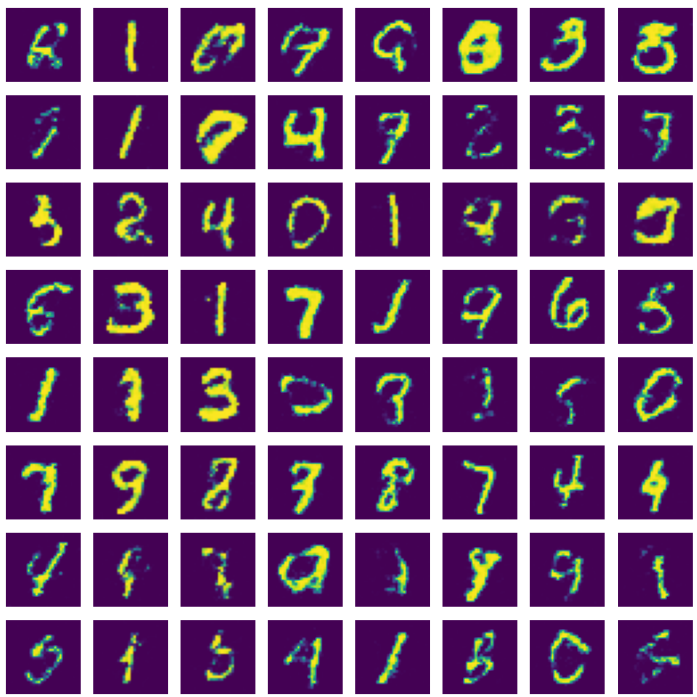
- Apprentissage d'une distributions multimodales (4 modes) par un RealNVP.
- On entraîne f_θ comme décrit précédemment et on utilise ensuite la formule (14) pour évaluer la densité des points du plan.





Exemple

- Entraînement d'un RealNVP sur le dataset MNIST et génération de data...
- On pourrait utiliser d'autres architectures inversibles plus appropriées aux images.



Exemple

- Entraînement d'un RealNVP sur le dataset MNIST et génération de data...
- On pourrait utiliser d'autres architectures inversibles plus appropriées aux images.

Bibliographie


Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017).
Density estimation using real NVP.
In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.


Ermon, S. and Grover, A. (2023).
Normalizing flows.
https://deepgenerativemodels.github.io/notes/flow/.
Accessed: 2025-04-28.


Weng, L. (2018).
Flow-based deep generative models.
lilianweng.github.io.


Wikipedia contributors (2024).
Flow-based generative model.
https://en.wikipedia.org/wiki/Flow-based_generative_model#cite_note-27.
Accessed: 2025-04-28.