Advanced Topics in Machine Learning
(deep generative modelling)

Lecture 4: Models with tractable exact density

Nikolay Malkin

3 February 2026

Outline of Lecture 4

Autoregressive models and normalising flows:

- Review
  - Warmup: Autoregressive models
- Noise outsourcing and pushforward measures
- Normalising flows

- Review
  - Warmup: Autoregressive models
- Noise outsourcing and pushforward measures
- Normalising flows

Review: Generative modelling as divergence minimisation

Setting:

- We have a **data distribution** π_{data} over $\mathbb{R}^d$ (from which we can sample, but we do not know its density function)
  - It could be the empirical distribution of a dataset
- We have a class of **model distributions** $\{\pi_\theta\}$ (with densities p_θ)
  - θ are the parameters of the model (*e.g.*, neural network weights, Gaussian mixture parameters)
  - Note that we do not necessarily know the density functions p_θ
- We seek θ such that π_θ approximates π_{data} well:

$$\theta^* = \arg\min_\theta D(\pi_\theta, \pi_{\text{data}})$$

Review: Generative modelling as divergence minimisation

The KL divergence

$$\mathrm{KL}(p\|q) = \mathbb{E}_{X\sim p}\left[\log \frac{p(X)}{q(X)}\right] = \int_{\mathbb{R}^d} p(x) \log \frac{p(x)}{q(x)}\,\mathrm{d}x$$

- Minimising $\mathrm{KL}(\pi_{\text{data}}\|\pi_\theta) \equiv$ maximising sample log-likelihood $\mathbb{E}_{X\sim\pi_{\text{data}}}[\log p_\theta(X)]$
- Algorithm to fit θ using stochastic gradient descent:
  - Sample a minibatch $x_1, \ldots, x_m \sim \pi_{\text{data}}$
  - Compute gradient estimate:

$$g = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta[-\log p_\theta(x_i)]$$

  - Update parameters: $\theta \leftarrow \theta - \eta g$ (or using your favourite optimiser)

Autoregressive models

Simplest models with tractable exact density: **autoregressive models**

- Factor a density $p(x_1, \ldots, x_d)$ as

$$p(x_1, \ldots, x_d) = p(x_1)p(x_2|x_1) \cdots p(x_d|x_1, \ldots, x_{d-1})$$

- Model each factor using a neural network

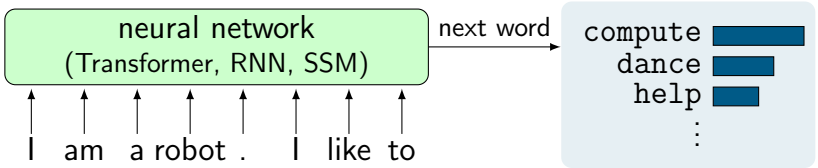
Autoregressive models

Simplest models with tractable exact density: **autoregressive models**

- Factor a density $p(x_1, \ldots, x_d)$ as

$$p(x_1, \ldots, x_d) = p(x_1)p(x_2|x_1) \cdots p(x_d|x_1, \ldots, x_{d-1})$$

- Model each factor using a neural network



- Can also randomise order (related to discrete diffusion models)

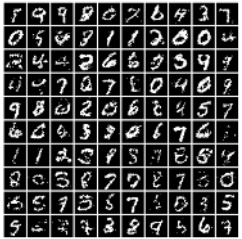
Autoregressive models

Simplest models with tractable exact density: **autoregressive models**

- Factor a density $p(x_1, \ldots, x_d)$ as

$$p(x_1, \ldots, x_d) = p(x_1)p(x_2|x_1) \cdots p(x_d|x_1, \ldots, x_{d-1})$$

- Model each factor using a neural network
- Common in language modelling, but also possible for images



MADE [Germain et al., 2015]

PixelRNN [van den Oord et al., 2016]

- Can also randomise order (related to discrete diffusion models)

- Review
  - Warmup: Autoregressive models
- Noise outsourcing and pushforward measures
- Normalising flows

Pushforward measure

If π is a probability distribution over $\mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a measurable function, then the **pushforward distribution** $f_{\#}\pi$ is a distribution over $\mathbb{R}^{d'}$ that is sampled by the procedure:

- Sample $x \sim \pi$
- Output $f(x)$

Pushforward measure

If π is a probability distribution over $\mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a measurable function, then the **pushforward distribution** $f_{\#}\pi$ is a distribution over $\mathbb{R}^{d'}$ that is sampled by the procedure:

- Sample $x \sim \pi$
- Output $f(x)$

Examples:

- $\pi = \text{Unif}([0, 1])$, $f(x) = 2x + 1$.

Pushforward measure

If π is a probability distribution over $\mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a measurable function, then the **pushforward distribution** $f_{\#}\pi$ is a distribution over $\mathbb{R}^{d'}$ that is sampled by the procedure:

- Sample $x \sim \pi$
- Output $f(x)$

Examples:

- $\pi = \text{Unif}([0, 1])$, $f(x) = 2x + 1$. $f_{\#}\pi = \text{Unif}([1, 3])$.

Pushforward measure

If π is a probability distribution over $\mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a measurable function, then the **pushforward distribution** $f_{\#}\pi$ is a distribution over $\mathbb{R}^{d'}$ that is sampled by the procedure:

- Sample $x \sim \pi$
- Output $f(x)$

Examples:

- $\pi = \text{Unif}([0, 1])$, $f(x) = 2x + 1$. $f_{\#}\pi = \text{Unif}([1, 3])$.
- $\pi = \delta_{17}$, $f(x) = 2x + 1$.

Pushforward measure

If π is a probability distribution over $\mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a measurable function, then the **pushforward distribution** $f_{\#}\pi$ is a distribution over $\mathbb{R}^{d'}$ that is sampled by the procedure:

- Sample $x \sim \pi$
- Output $f(x)$

Examples:

- $\pi = \text{Unif}([0, 1])$, $f(x) = 2x + 1$. $f_{\#}\pi = \text{Unif}([1, 3])$.
- $\pi = \delta_{17}$, $f(x) = 2x + 1$. $f_{\#}\pi = \delta_{35}$.

Pushforward measure

If π is a probability distribution over $\mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a measurable function, then the **pushforward distribution** $f_{\#}\pi$ is a distribution over $\mathbb{R}^{d'}$ that is sampled by the procedure:

- Sample $x \sim \pi$
- Output $f(x)$

Examples:

- $\pi = \text{Unif}([0, 1])$, $f(x) = 2x + 1$. $f_{\#}\pi = \text{Unif}([1, 3])$.
- $\pi = \delta_{17}$, $f(x) = 2x + 1$. $f_{\#}\pi = \delta_{35}$.
- $\pi = \mathcal{N}(0, 1)$, $f(x) = 2x + 1$.

Pushforward measure

If π is a probability distribution over $\mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a measurable function, then the **pushforward distribution** $f_{\#}\pi$ is a distribution over $\mathbb{R}^{d'}$ that is sampled by the procedure:

- Sample $x \sim \pi$
- Output $f(x)$

Examples:

- $\pi = \text{Unif}([0, 1])$, $f(x) = 2x + 1$. $f_{\#}\pi = \text{Unif}([1, 3])$.
- $\pi = \delta_{17}$, $f(x) = 2x + 1$. $f_{\#}\pi = \delta_{35}$.
- $\pi = \mathcal{N}(0, 1)$, $f(x) = 2x + 1$. $f_{\#}\pi = \mathcal{N}(1, 4)$.

Pushforward measure

If π is a probability distribution over $\mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a measurable function, then the **pushforward distribution** $f_{\#}\pi$ is a distribution over $\mathbb{R}^{d'}$ that is sampled by the procedure:

- Sample $x \sim \pi$
- Output $f(x)$

Examples:

- $\pi = \text{Unif}([0, 1])$, $f(x) = 2x + 1$. $f_{\#}\pi = \text{Unif}([1, 3])$.
- $\pi = \delta_{17}$, $f(x) = 2x + 1$. $f_{\#}\pi = \delta_{35}$.
- $\pi = \mathcal{N}(0, 1)$, $f(x) = 2x + 1$. $f_{\#}\pi = \mathcal{N}(1, 4)$.
- $\pi = \text{Unif}(0, 2\pi)$, $f(x) = (\cos x, \sin x)$.

Pushforward measure

If π is a probability distribution over $\mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a measurable function, then the **pushforward distribution** $f_{\#}\pi$ is a distribution over $\mathbb{R}^{d'}$ that is sampled by the procedure:

- Sample $x \sim \pi$
- Output $f(x)$

Examples:

- $\pi = \text{Unif}([0, 1])$, $f(x) = 2x + 1$. $f_{\#}\pi = \text{Unif}([1, 3])$.
- $\pi = \delta_{17}$, $f(x) = 2x + 1$. $f_{\#}\pi = \delta_{35}$.
- $\pi = \mathcal{N}(0, 1)$, $f(x) = 2x + 1$. $f_{\#}\pi = \mathcal{N}(1, 4)$.
- $\pi = \text{Unif}(0, 2\pi)$, $f(x) = (\cos x, \sin x)$. $f_{\#}\pi$ is uniform on the circle.

Short review of variable change formula

A special case:

- π has a density $p : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$
- $f : \mathbb{R}^d \to \mathbb{R}^d$ is **invertible** and **differentiable**, with differentiable inverse f^{-1} (a diffeomorphism)

Short review of variable change formula

A special case:

- π has a density $p : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$
- $f : \mathbb{R}^d \to \mathbb{R}^d$ is **invertible** and **differentiable**, with differentiable inverse f^{-1} (a diffeomorphism)

Change of variables formula: if $y = f(x)$, then

density of the pushforward at y

$$\overbrace{(f_{\#}p)(y)} = p(x) \left| \det \left(\frac{\partial f^{-1}}{\partial y}(y) \right) \right|$$

$$= p(x) \underbrace{\left| \det \left(\frac{\partial f}{\partial x}(x) \right) \right|^{-1}}$$

determinant of Jacobian of f at $x = f^{-1}(y)$

Short review of variable change formula

$$\overbrace{(f_{\#}p)(y)}^{\text{density of the pushforward at } y} = p(x)\left|\det\left(\frac{\partial f^{-1}}{\partial y}(y)\right)\right| = p(x)\underbrace{\left|\det\left(\frac{\partial f}{\partial x}(x)\right)\right|}_{\text{determinant of Jacobian of } f \text{ at } x = f^{-1}(y)}^{-1}$$

Why?

- Linear approximation:

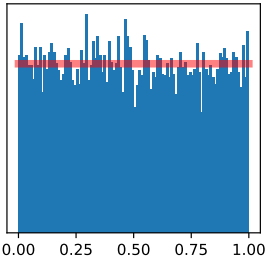
$$f(x+\Delta x) = f(x) + \underbrace{\frac{\partial f}{\partial x}(x)}_{d \times d \text{ matrix}} \overbrace{\Delta x}^{\text{vector in } \mathbb{R}^d} + O(\|\Delta x\|^2)$$

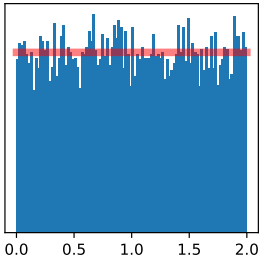
- Volumes are scaled by the determinant under a linear transformation

Short review of variable change formula

Examples:

- $\pi = \mathrm{Unif}([0,1])$ ($p(x) = 1$ for $x \in [0,1]$), $f(x) = 2x$
  - $|f'(x)| = 2$, so $(f_{\#}\pi)(y) = \frac{1}{2}$ for $y \in [0,2]$

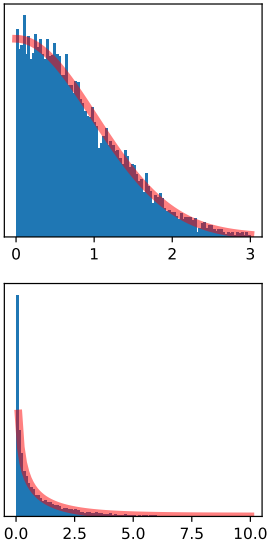




Short review of variable change formula

Examples:

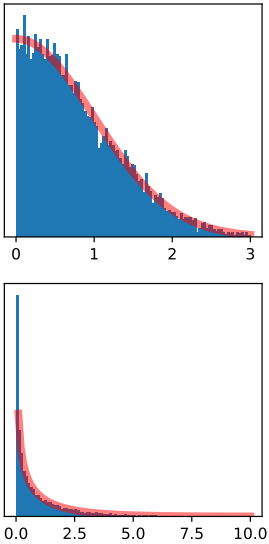
- $\pi = \text{Unif}([0,1])$ ($p(x) = 1$ for $x \in [0,1]$), $f(x) = 2x$
  - $|f'(x)| = 2$, so $(f_{\#}\pi)(y) = \frac{1}{2}$ for $y \in [0,2]$
- $\pi = \mathcal{N}(0,1)$ restricted to $[0,\infty)$ ($p(x) = \frac{2}{\sqrt{2\pi}} \exp(\frac{-x^2}{2})$), $f(x) = x^2$
  - $|f'(x)| = 2|x|$, so $(f_{\#}\pi)(y) = \frac{1}{\sqrt{2\pi y}} \exp(\frac{-y}{2})$ for $y \geq 0$



Short review of variable change formula

Examples:

- $\pi = \mathrm{Unif}([0,1])$ ($p(x) = 1$ for $x \in [0,1]$), $f(x) = 2x$
  - $|f'(x)| = 2$, so $(f_{\#}\pi)(y) = \frac{1}{2}$ for $y \in [0,2]$
- $\pi = \mathcal{N}(0,1)$ restricted to $[0,\infty)$ ($p(x) = \frac{2}{\sqrt{2\pi}} \exp(\frac{-x^2}{2})$), $f(x) = x^2$
  - $|f'(x)| = 2|x|$, so $(f_{\#}\pi)(y) = \frac{1}{\sqrt{2\pi y}} \exp(\frac{-y}{2})$ for $y \geq 0$ (χ^2 distribution with one degree of freedom).



Short review of variable change formula

Examples:

- $\pi = \mathrm{Unif}([0,1])$ ($p(x) = 1$ for $x \in [0,1]$), $f(x) = 2x$
  - $|f'(x)| = 2$, so $(f_{\#}\pi)(y) = \frac{1}{2}$ for $y \in [0,2]$
- $\pi = \mathcal{N}(0,1)$ restricted to $[0,\infty)$ ($p(x) = \frac{2}{\sqrt{2\pi}} \exp(\frac{-x^2}{2})$), $f(x) = x^2$
  - $|f'(x)| = 2|x|$, so $(f_{\#}\pi)(y) = \frac{1}{\sqrt{2\pi y}} \exp(\frac{-y}{2})$ for $y \geq 0$ (χ^2 distribution with one degree of freedom).
- $\pi = \mathrm{Unif}([0,1] \times [0,2\pi])$, $f(r,\theta) = (r\cos\theta, r\sin\theta)$

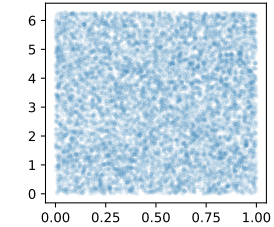
Short review of variable change formula

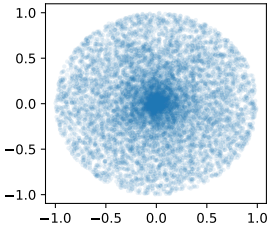
Examples:

- $\pi = \text{Unif}([0,1])$ ($p(x) = 1$ for $x \in [0,1]$), $f(x) = 2x$
  - $|f'(x)| = 2$, so $(f_{\#}\pi)(y) = \frac{1}{2}$ for $y \in [0,2]$
- $\pi = \mathcal{N}(0,1)$ restricted to $[0,\infty)$ ($p(x) = \frac{2}{\sqrt{2\pi}} \exp(\frac{-x^2}{2})$), $f(x) = x^2$
  - $|f'(x)| = 2|x|$, so $(f_{\#}\pi)(y) = \frac{1}{\sqrt{2\pi y}} \exp(\frac{-y}{2})$ for $y \geq 0$ (χ^2 distribution with one degree of freedom).
- $\pi = \text{Unif}([0,1] \times [0,2\pi])$, $f(r,\theta) = (r\cos\theta, r\sin\theta)$

$$\det\left(\frac{\partial f}{\partial(r,\theta)}\right) = \det\begin{pmatrix}\cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta\end{pmatrix} = r$$

  - $(f_{\#}\pi)(x,y) = \frac{1}{2\pi r}$, where $r = \sqrt{x^2+y^2}$, for $r \in [0,1]$





Noise outsourcing

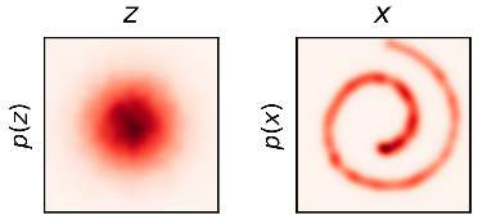
Noise outsourcing lemma: Any distribution π over $\mathbb{R}^d$ can be represented as the pushforward of a simple distribution (*e.g.*, $\mathcal{N}(0, I_d)$ or $\mathrm{Unif}([0,1]^d)$) by some function

- Not very useful, does not tell us how to find such a function or guarantee invertibility and differentiability

Noise outsourcing

Noise outsourcing lemma: Any distribution π over $\mathbb{R}^d$ can be represented as the pushforward of a simple distribution (*e.g.*, $\mathcal{N}(0, I_d)$ or $\mathrm{Unif}([0,1]^d)$) by some function

- Not very useful, does not tell us how to find such a function or guarantee invertibility and differentiability
- But if π has a density and satisfies other basic conditions, then such a smooth invertible function **does** exist (related to Brenier's theorem, optimal transport)



Noise outsourcing

Noise outsourcing lemma: Any distribution π over $\mathbb{R}^d$ can be represented as the pushforward of a simple distribution (*e.g.*, $\mathcal{N}(0, I_d)$ or $\text{Unif}([0,1]^d)$) by some function

- Not very useful, does not tell us how to find such a function or guarantee invertibility and differentiability
- But if π has a density and satisfies other basic conditions, then such a smooth invertible function **does** exist (related to Brenier's theorem, optimal transport)

Implication for generative modelling: To approximate π_{data}:

- Parametrise a class of smooth invertible functions $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ that are easy to invert and differentiate
- Define $\pi_\theta = (f_\theta)_{\#}\pi_{\text{base}}$, where π_{base} is a simple distribution (*e.g.*, $\mathcal{N}(0, I_d)$)
- Fit θ by maximising data log-likelihood (given by variable change formula)

- Review
  - Warmup: Autoregressive models
- Noise outsourcing and pushforward measures
- Normalising flows

Normalising flows

Parametrise a class of smooth invertible functions $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ that are easy to invert and differentiate

Normalising flows

Parametrise a class of smooth invertible functions $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ that are easy to invert and differentiate How do we build such functions? Compose simple invertible layers!

- If $f : \mathbb{R}^d \to \mathbb{R}^d$ and $g : \mathbb{R}^d \to \mathbb{R}^d$ are diffeomorphisms, then so is $g \circ f$, and the inverse is $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$

Normalising flows

Parametrise a class of smooth invertible functions $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ that are easy to invert and differentiate How do we build such functions? Compose simple invertible layers!

- If $f : \mathbb{R}^d \to \mathbb{R}^d$ and $g : \mathbb{R}^d \to \mathbb{R}^d$ are diffeomorphisms, then so is $g \circ f$, and the inverse is $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$
- Chain rule: if $y = f(x)$ and $z = g(y)$, then

$$\det\left(\frac{\partial z}{\partial x}\right) = \det\left(\frac{\partial z}{\partial y}\frac{\partial y}{\partial x}\right) = \det\left(\frac{\partial z}{\partial y}\right) \cdot \det\left(\frac{\partial y}{\partial x}\right)$$

- Compose more such layers to get more complex functions

Normalising flows

Parametrise a class of smooth invertible functions $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ that are easy to invert and differentiate How do we build such functions? Compose simple invertible layers!

- If $f : \mathbb{R}^d \to \mathbb{R}^d$ and $g : \mathbb{R}^d \to \mathbb{R}^d$ are diffeomorphisms, then so is $g \circ f$, and the inverse is $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$
- Chain rule: if $y = f(x)$ and $z = g(y)$, then

$$\det\left(\frac{\partial z}{\partial x}\right) = \det\left(\frac{\partial z}{\partial y}\frac{\partial y}{\partial x}\right) = \det\left(\frac{\partial z}{\partial y}\right) \cdot \det\left(\frac{\partial y}{\partial x}\right)$$

- Compose more such layers to get more complex functions

Next: a library of simple layers

Linear layer

$f(x) = Ax + b$, where A is an invertible $d \times d$ matrix and $b \in \mathbb{R}^d$

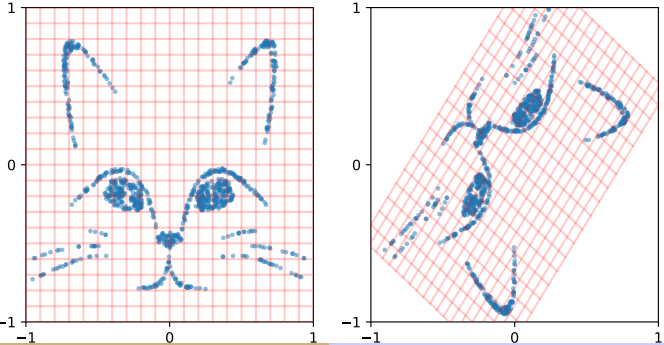
- Inverse: $f^{-1}(y) = A^{-1}(y - b)$
- Jacobian determinant: $\left|\det\left(\frac{\partial f}{\partial x}\right)\right| = |\det A|$

Linear layer

$f(x) = Ax + b$, where A is an invertible $d \times d$ matrix and $b \in \mathbb{R}^d$

- Inverse: $f^{-1}(y) = A^{-1}(y - b)$
- Jacobian determinant: $\left|\det\left(\frac{\partial f}{\partial x}\right)\right| = |\det A|$

Can we build a model out of just linear layers?

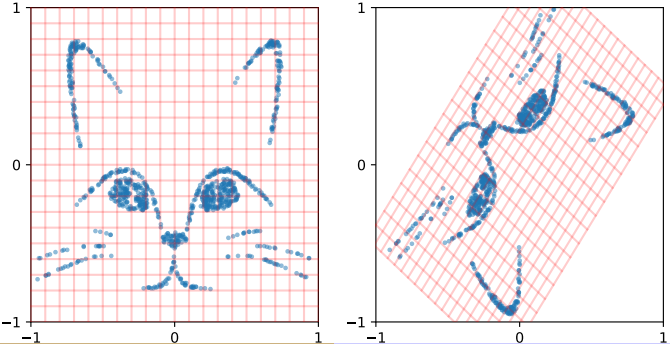


Linear layer

$f(x) = Ax + b$, where A is an invertible $d \times d$ matrix and $b \in \mathbb{R}^d$

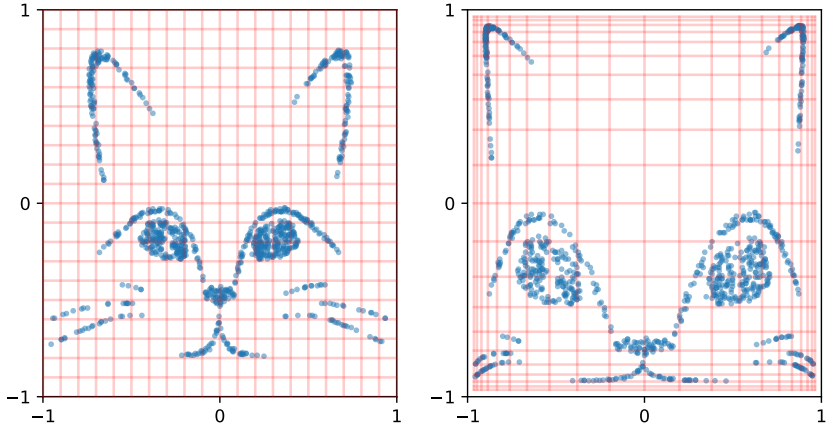
- Inverse: $f^{-1}(y) = A^{-1}(y - b)$
- Jacobian determinant: $\left|\det\left(\frac{\partial f}{\partial x}\right)\right| = |\det A|$

Can we build a model out of just linear layers? The composition is again linear, not very expressive!



Elementwise nonlinearity layer

$f(x_1, \ldots, x_d) = (\sigma(x_1), \ldots, \sigma(x_d))$, where $\sigma : \mathbb{R} \to \mathbb{R}$ is a smooth invertible function (*e.g.*, ELU, leaky ReLU, softplus, tanh)



Elementwise nonlinearity layer

$f(x_1, \ldots, x_d) = (\sigma(x_1), \ldots, \sigma(x_d))$, where $\sigma : \mathbb{R} \to \mathbb{R}$ is a smooth invertible function (*e.g.*, ELU, leaky ReLU, softplus, tanh)

- Inverse: $f^{-1}(y_1, \ldots, y_d) = (\sigma^{-1}(y_1), \ldots, \sigma^{-1}(y_d))$
- Jacobian determinant:

$$\left|\det\left(\frac{\partial f}{\partial x}\right)\right| = \left|\det\begin{pmatrix} \sigma'(x_1) & 0 & \cdots & 0 \\ 0 & \sigma'(x_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma'(x_d) \end{pmatrix}\right| = \prod_{i=1}^{d} |\sigma'(x_i)|$$

Elementwise nonlinearity layer

$f(x_1, \ldots, x_d) = (\sigma(x_1), \ldots, \sigma(x_d))$, where $\sigma : \mathbb{R} \to \mathbb{R}$ is a smooth invertible function (*e.g.*, ELU, leaky ReLU, softplus, tanh)

- Inverse: $f^{-1}(y_1, \ldots, y_d) = (\sigma^{-1}(y_1), \ldots, \sigma^{-1}(y_d))$
- Jacobian determinant:

$$\left|\det\left(\frac{\partial f}{\partial x}\right)\right| = \left|\det\begin{pmatrix} \sigma'(x_1) & 0 & \cdots & 0 \\ 0 & \sigma'(x_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma'(x_d) \end{pmatrix}\right| = \prod_{i=1}^{d} |\sigma'(x_i)|$$

Can we build a model out of just elementwise nonlinearities?

Elementwise nonlinearity layer

$f(x_1, \ldots, x_d) = (\sigma(x_1), \ldots, \sigma(x_d))$, where $\sigma : \mathbb{R} \to \mathbb{R}$ is a smooth invertible function (*e.g.*, ELU, leaky ReLU, softplus, tanh)

- Inverse: $f^{-1}(y_1, \ldots, y_d) = (\sigma^{-1}(y_1), \ldots, \sigma^{-1}(y_d))$
- Jacobian determinant:

$$\left|\det\left(\frac{\partial f}{\partial x}\right)\right| = \left|\det\begin{pmatrix} \sigma'(x_1) & 0 & \cdots & 0 \\ 0 & \sigma'(x_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma'(x_d) \end{pmatrix}\right| = \prod_{i=1}^{d} |\sigma'(x_i)|$$

Can we build a model out of just elementwise nonlinearities? The composition is again elementwise, not very expressive!

What about linear layers + elementwise nonlinearities?

Elementwise nonlinearity layer

$f(x_1, \ldots, x_d) = (\sigma(x_1), \ldots, \sigma(x_d))$, where $\sigma : \mathbb{R} \to \mathbb{R}$ is a smooth invertible function (*e.g.*, ELU, leaky ReLU, softplus, tanh)

- Inverse: $f^{-1}(y_1, \ldots, y_d) = (\sigma^{-1}(y_1), \ldots, \sigma^{-1}(y_d))$
- Jacobian determinant:

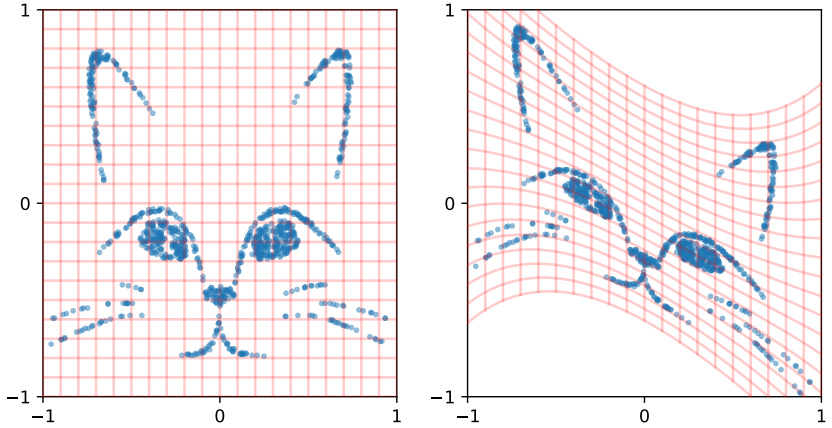
$$\left|\det\left(\frac{\partial f}{\partial x}\right)\right| = \left|\det\begin{pmatrix} \sigma'(x_1) & 0 & \cdots & 0 \\ 0 & \sigma'(x_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma'(x_d) \end{pmatrix}\right| = \prod_{i=1}^{d} |\sigma'(x_i)|$$

Can we build a model out of just elementwise nonlinearities? The composition is again elementwise, not very expressive!

What about linear layers + elementwise nonlinearities? Expressive in theory, but inefficient, since dimension of all intermediate representations is d.

Coupling layer

In two dimensions: $f(x, y) = (x, g_\theta(x, y))$, where $g : \mathbb{R}^2 \to \mathbb{R}$ is invertible in y, *e.g.*, $g_\theta(x, y) = \exp(s_\theta(x))y + t_\theta(x)$, where $s, t : \mathbb{R} \to \mathbb{R}$



Coupling layer

In two dimensions: $f(x, y) = (x, g_\theta(x, y))$, where $g : \mathbb{R}^2 \to \mathbb{R}$ is invertible in y, *e.g.*, $g_\theta(x, y) = \exp(s_\theta(x))y + t_\theta(x)$, where $s, t : \mathbb{R} \to \mathbb{R}$

- Inverse: $f^{-1}(u, v) = (u, g_\theta^{-1}(u, v))$, where $g_\theta^{-1}(u, v) = \exp(-s_\theta(u))(v - t_\theta(u))$
- Jacobian determinant: if $(u, v) = f(x, y)$, then:

$$\left| \det \left(\frac{\partial f}{\partial (x, y)} \right) \right|$$

Coupling layer

In two dimensions: $f(x, y) = (x, g_\theta(x, y))$, where $g : \mathbb{R}^2 \to \mathbb{R}$ is invertible in y, *e.g.*, $g_\theta(x, y) = \exp(s_\theta(x))y + t_\theta(x)$, where $s, t : \mathbb{R} \to \mathbb{R}$

- Inverse: $f^{-1}(u, v) = (u, g_\theta^{-1}(u, v))$, where $g_\theta^{-1}(u, v) = \exp(-s_\theta(u))(v - t_\theta(u))$
- Jacobian determinant: if $(u, v) = f(x, y)$, then:

$$\left|\det\left(\frac{\partial f}{\partial(x, y)}\right)\right| = \left|\det\begin{pmatrix}\frac{\partial u}{\partial x} & \frac{\partial u}{\partial y}\\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y}\end{pmatrix}\right|$$

Coupling layer

In two dimensions: $f(x, y) = (x, g_\theta(x, y))$, where $g : \mathbb{R}^2 \to \mathbb{R}$ is invertible in y, *e.g.*, $g_\theta(x, y) = \exp(s_\theta(x))y + t_\theta(x)$, where $s, t : \mathbb{R} \to \mathbb{R}$

- Inverse: $f^{-1}(u, v) = (u, g_\theta^{-1}(u, v))$, where $g_\theta^{-1}(u, v) = \exp(-s_\theta(u))(v - t_\theta(u))$
- Jacobian determinant: if $(u, v) = f(x, y)$, then:

$$\begin{aligned}\left|\det\left(\frac{\partial f}{\partial(x, y)}\right)\right| &= \left|\det\begin{pmatrix}\frac{\partial u}{\partial x} & \frac{\partial u}{\partial y}\\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y}\end{pmatrix}\right| \\ &= \left|\det\begin{pmatrix}1 & 0\\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y}\end{pmatrix}\right|\end{aligned}$$

Coupling layer

In two dimensions: $f(x, y) = (x, g_\theta(x, y))$, where $g : \mathbb{R}^2 \to \mathbb{R}$ is invertible in y, *e.g.*, $g_\theta(x, y) = \exp(s_\theta(x))y + t_\theta(x)$, where $s, t : \mathbb{R} \to \mathbb{R}$

- Inverse: $f^{-1}(u, v) = (u, g_\theta^{-1}(u, v))$, where $g_\theta^{-1}(u, v) = \exp(-s_\theta(u))(v - t_\theta(u))$
- Jacobian determinant: if $(u, v) = f(x, y)$, then:

$$\left|\det\left(\frac{\partial f}{\partial(x, y)}\right)\right| = \left|\det\begin{pmatrix}\frac{\partial u}{\partial x} & \frac{\partial u}{\partial y}\\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y}\end{pmatrix}\right|$$

$$= \left|\det\begin{pmatrix}1 & 0\\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y}\end{pmatrix}\right| \qquad = \left|\frac{\partial v}{\partial y}\right|$$

Coupling layer

In two dimensions: $f(x, y) = (x, g_\theta(x, y))$, where $g : \mathbb{R}^2 \to \mathbb{R}$ is invertible in y, *e.g.*, $g_\theta(x, y) = \exp(s_\theta(x))y + t_\theta(x)$, where $s, t : \mathbb{R} \to \mathbb{R}$

- Inverse: $f^{-1}(u, v) = (u, g_\theta^{-1}(u, v))$, where $g_\theta^{-1}(u, v) = \exp(-s_\theta(u))(v - t_\theta(u))$
- Jacobian determinant: if $(u, v) = f(x, y)$, then:

$$\left|\det\left(\frac{\partial f}{\partial(x, y)}\right)\right| = \left|\det\begin{pmatrix}\frac{\partial u}{\partial x} & \frac{\partial u}{\partial y}\\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y}\end{pmatrix}\right|$$

$$= \left|\det\begin{pmatrix}1 & 0\\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y}\end{pmatrix}\right| \qquad = \left|\frac{\partial v}{\partial y}\right| = |\exp(s_\theta(x))|$$

Coupling layer

Higher-dimensional version (transforming $x \in \mathbb{R}^d$):

- Split x in half: $x = \overbrace{x_\mathsf{L} \oplus x_\mathsf{R}}^{\text{concatenation}}$ with $x_\mathsf{L} \in \mathbb{R}^{d_\mathsf{L}}$, $x_\mathsf{R} \in \mathbb{R}^{d_\mathsf{R}}$ $(d_\mathsf{L} + d_\mathsf{R} = d)$

Coupling layer

Higher-dimensional version (transforming $x \in \mathbb{R}^d$):

- Split x in half: $x = \overbrace{x_\mathrm{L} \oplus x_\mathrm{R}}^{\text{concatenation}}$ with $x_\mathrm{L} \in \mathbb{R}^{d_\mathrm{L}}$, $x_\mathrm{R} \in \mathbb{R}^{d_\mathrm{R}}$ $(d_\mathrm{L} + d_\mathrm{R} = d)$
- Define

$$f(x_\mathrm{L}, x_\mathrm{R}) = \underbrace{x_\mathrm{L}}_{u} \oplus \underbrace{g_\theta(x_\mathrm{L}, x_\mathrm{R})}_{v},$$

where $g_\theta(x_\mathrm{L}, x_\mathrm{R}) = \underbrace{\exp(s_\theta(x_\mathrm{L})) \odot x_\mathrm{R}}_{\text{elementwise product}} + t_\theta(x_\mathrm{L})$ with $s_\theta, t_\theta : \mathbb{R}^{d_\mathrm{L}} \to \mathbb{R}^{d_\mathrm{R}}$

Coupling layer

Higher-dimensional version (transforming $x \in \mathbb{R}^d$):

- Split x in half: $x = \overbrace{x_\mathrm{L} \oplus x_\mathrm{R}}^{\text{concatenation}}$ with $x_\mathrm{L} \in \mathbb{R}^{d_\mathrm{L}}$, $x_\mathrm{R} \in \mathbb{R}^{d_\mathrm{R}}$ $(d_\mathrm{L} + d_\mathrm{R} = d)$
- Define

$$f(x_\mathrm{L}, x_\mathrm{R}) = \underbrace{x_\mathrm{L}}_{u} \oplus \underbrace{g_\theta(x_\mathrm{L}, x_\mathrm{R})}_{v},$$

where $g_\theta(x_\mathrm{L}, x_\mathrm{R}) = \underbrace{\exp(s_\theta(x_\mathrm{L})) \odot x_\mathrm{R}}_{\text{elementwise product}} + t_\theta(x_\mathrm{L})$ with $s_\theta, t_\theta : \mathbb{R}^{d_\mathrm{L}} \to \mathbb{R}^{d_\mathrm{R}}$

- Jacobian has a block structure, determinant simplifies in a similar way:

$$\left|\det\left(\frac{\partial f}{\partial(x_\mathrm{L}, x_\mathrm{R})}\right)\right| = \left|\det\begin{pmatrix} I_{d_\mathrm{L}} & 0 \\ \frac{\partial v}{\partial x_\mathrm{L}} & \frac{\partial v}{\partial x_\mathrm{R}} \end{pmatrix}\right| = \left|\det\left(\frac{\partial v}{\partial x_\mathrm{R}}\right)\right| = \prod_{i=1}^{d_\mathrm{R}} |\exp(s_\theta(x_\mathrm{L})_i)|$$

Coupling layer

Higher-dimensional version (transforming $x \in \mathbb{R}^d$):

- Split x in half: $x = \overbrace{x_\mathsf{L} \oplus x_\mathsf{R}}^{\text{concatenation}}$ with $x_\mathsf{L} \in \mathbb{R}^{d_\mathsf{L}}$, $x_\mathsf{R} \in \mathbb{R}^{d_\mathsf{R}}$ $(d_\mathsf{L} + d_\mathsf{R} = d)$
- Define

$$f(x_\mathsf{L}, x_\mathsf{R}) = \underbrace{x_\mathsf{L}}_{u} \oplus \underbrace{g_\theta(x_\mathsf{L}, x_\mathsf{R})}_{v},$$

  where $g_\theta(x_\mathsf{L}, x_\mathsf{R}) = \underbrace{\exp(s_\theta(x_\mathsf{L})) \odot x_\mathsf{R}}_{\text{elementwise product}} + t_\theta(x_\mathsf{L})$ with $s_\theta, t_\theta : \mathbb{R}^{d_\mathsf{L}} \to \mathbb{R}^{d_\mathsf{R}}$

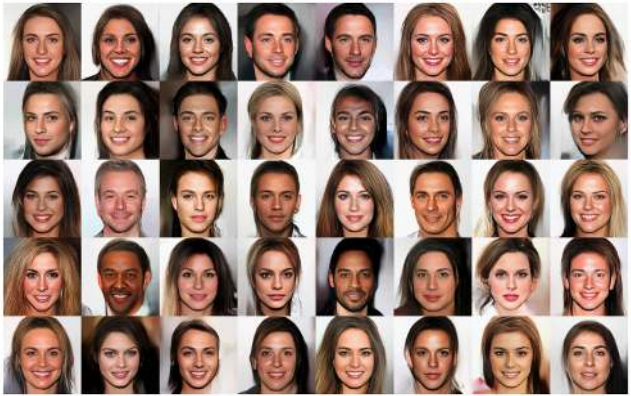
- Jacobian has a block structure, determinant simplifies in a similar way:

$$\left|\det\left(\frac{\partial f}{\partial(x_\mathsf{L}, x_\mathsf{R})}\right)\right| = \left|\det\begin{pmatrix} I_{d_\mathsf{L}} & 0 \\ \frac{\partial v}{\partial x_\mathsf{L}} & \frac{\partial v}{\partial x_\mathsf{R}} \end{pmatrix}\right| = \left|\det\left(\frac{\partial v}{\partial x_\mathsf{R}}\right)\right| = \prod_{i=1}^{d_\mathsf{R}} |\exp(s_\theta(x_\mathsf{L})_i)|$$

- By stacking such layers with different splits, get expressive models

Scaling to image data

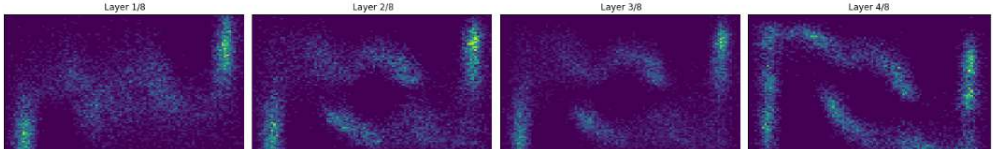
With a few domain-specific layers, normalising flows can scale to high-dimensional data (such as images)



Glow [Kingma and Dhariwal, 2018] (coupling layers with convolutions)

Some challenges of normalising flows

- Limited capacity $\rightsquigarrow$ need many layers, mode connectivity issue



[from Murphy, §23.2]

Some challenges of normalising flows

- Limited capacity $\rightsquigarrow$ need many layers, mode connectivity issue
- Less scalable to high-dimensional data than other models
- Requires a noise of the same dimension as the data (no latent compression)
  - If f_θ is a normalising flow and $x \sim \pi_{\text{data}}$, can we think of $z = f_\theta^{-1}(x)$ as a latent code for x?

Some challenges of normalising flows

- Limited capacity $\rightsquigarrow$ need many layers, mode connectivity issue
- Less scalable to high-dimensional data than other models
- Requires a noise of the same dimension as the data (no latent compression)
  - If f_θ is a normalising flow and $x \sim \pi_{\text{data}}$, can we think of $z = f_\theta^{-1}(x)$ as a latent code for x?
  - Not really, since z has the same dimension as x and is rarely "simpler" than x
  - But we have more control with continuous normalising flows (last lectures) – invertible maps given by simulating a dynamical system

Conclusion and looking ahead

- Models with tractable exact density allow direct maximum likelihood training and evaluation
- Less scalable than other families, so fell out of favour for high-dimensional generative modelling
  - But also used as variational posteriors (in latent-variable models / for Bayesian inference)
- Next time: GANs (another pushforward model family, but trained very differently)