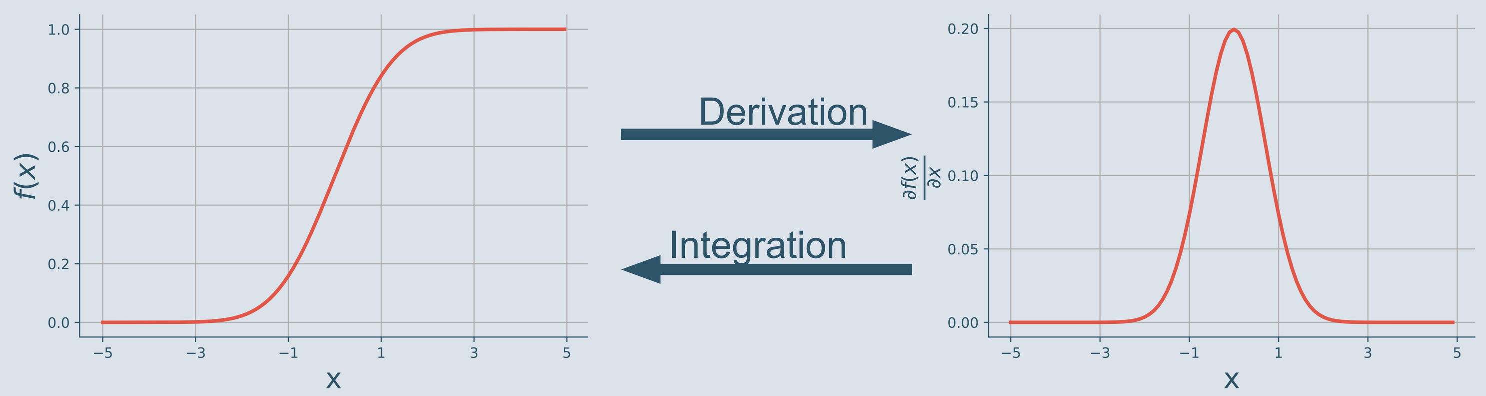
Unconstrained Monotonic Neural Networks

Antoine Wehenkel and Gilles Louppe

What ? UMNN is a **new architecture** to model **monotonic** functions.
How ? The strictly **positive** scalar output of a **neural network** is **numerically integrated**.
Applications ? We combine UMNNs with **autoregressive flows** to perform **density estimation**.

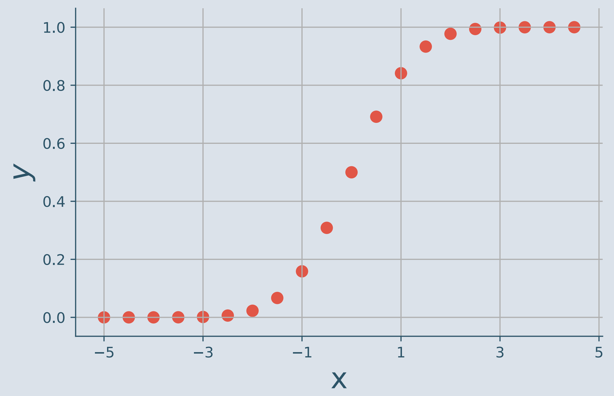
Monotonicity

In theory



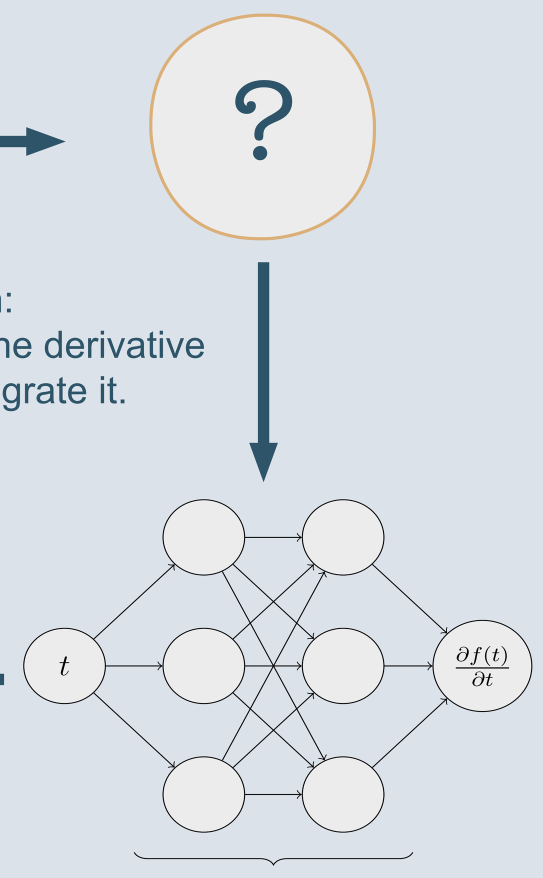
Monotonicity is hard to enforce BUT positiveness is not !

In practice



We only have a cloud of points.

Derivation



Solution: Model the derivative and integrate it.

$$\hat{y} = \int_0^x \frac{\partial f(t)}{\partial t} dt + f(0)$$

Numerical integration

UMNN

Update

$$\mathcal{L}_\theta = \sum_i l(\hat{y}_i, y_i)$$

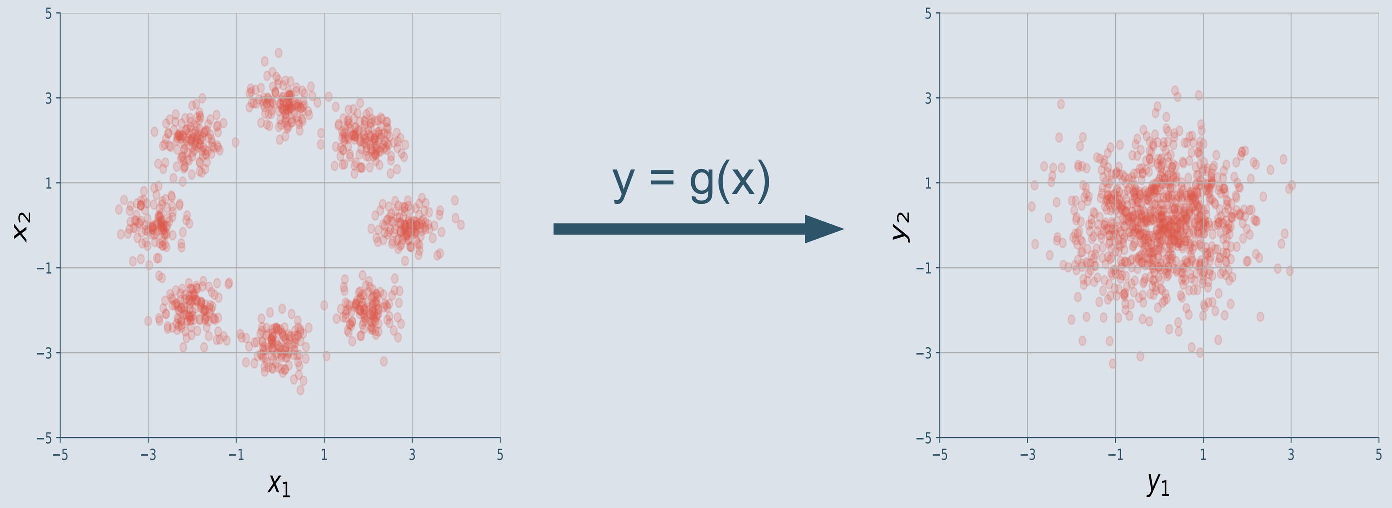
Backward

$$\nabla \mathcal{L}_\theta$$

Change of variables

Let g be a bijective function, x a random variable and let y defined as g(x). The change of variables theorem states that:

$$f_Y(y) = f_X(g^{-1}(y)) \left|\det(J_{g^{-1}})\right|$$



A bijective transformation can be built by the combination of an autoregressive architecture with a UMNN.

Autoregressivity

Autoregressive transformations are commonly used to build bijective transformations.

$$\mathbf{g}(\mathbf{x};\theta) = \left[g^1(x_1;\theta) \quad \dots \quad g^i(\mathbf{x}_{1:i};\theta) \quad \dots \quad g^d(\mathbf{x}_{1:d};\theta)\right]$$

The induced multivariate density can be expressed by the chain rule:

$$p(\mathbf{x};\theta) = p(x_1;\theta)\prod_{i=1}^{d-1} p(x_{i+1}|\mathbf{x}_{1:i};\theta).$$

We combine UMNN with autoregressive transformations as:

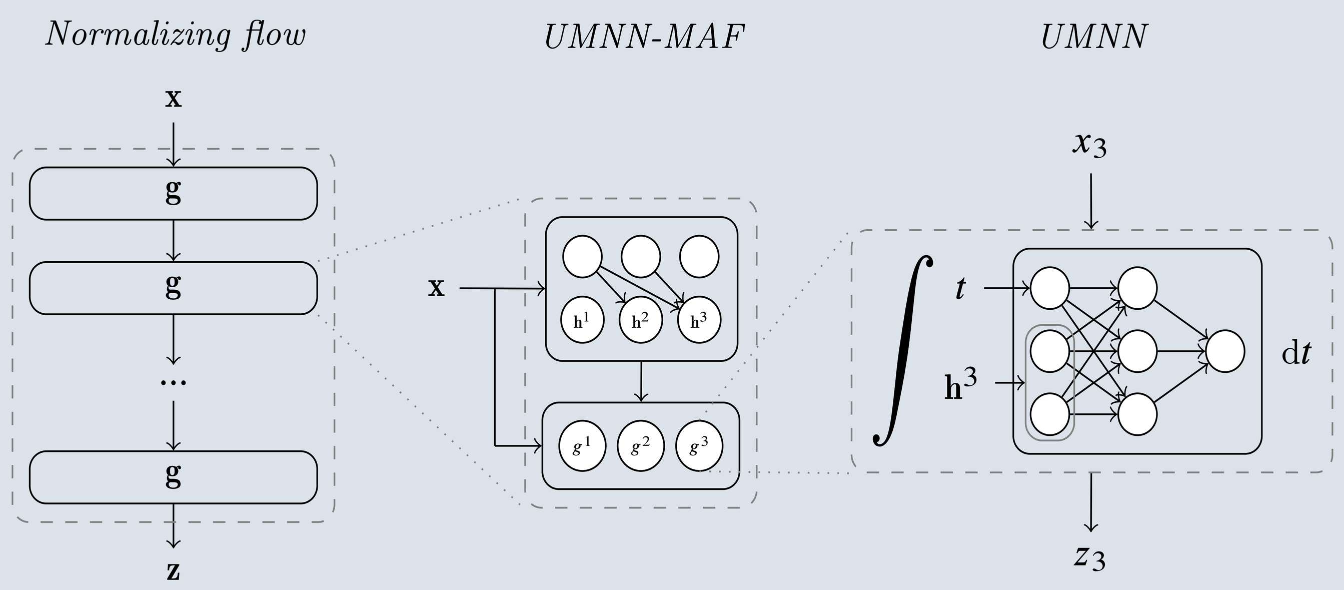
$$g^i(\mathbf{x}_{1:i};\theta) = F^i(x_i, \mathbf{h}^i(\mathbf{x}_{1:i-1};\phi^i);\psi^i) = \int_0^{x_i} f^i(t, \mathbf{h}^i(\mathbf{x}_{1:i-1};\phi^i);\psi^i) + \beta^i(\mathbf{h}^i(\mathbf{x}_{1:i-1};\phi^i))$$

UMNN-MAF leads to a simple expression of the Jacobian:

$$\log p(\mathbf{x};\theta) = \log p_Z(\mathbf{g}(\mathbf{x};\theta)) + \sum_{i=1}^{d} \log f^i(x_i, \mathbf{h}^i(\mathbf{x}_{1:i-1}))$$

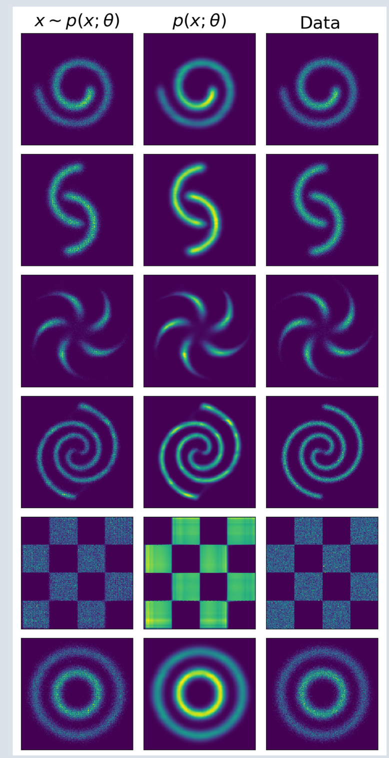
Architecture

We combine the UMNN architecture with an autoregressive network to represent multi-dimensionnal bijective transformations.



Results

Toy problems



MNIST

Density Estimation

	Dataset	POWER	GAS	HEPMASS	MINIBOONE	BSDS300	MNIST
	RealNVP - Dinh et al. [2017]	$-0.17_{\pm.01}$	$-8.33_{\pm.14}$	$18.71_{\pm.02}$	$13.55_{\pm.49}$	$-153.28_{\pm1.78}$	-
(a)	Glow - Kingma and Dhariwal [2018]	$-0.17_{\pm.01}$	$-8.15_{\pm.40}$	$19.92_{\pm.08}$	$11.35_{\pm.07}$	$-155.07_{\pm.03}$	-
	FFJORD - Grathwohl et al. [2018]	$\underline{-0.46}_{\pm.01}$	$\underline{-8.59}_{\pm.12}$	$\underline{14.92}_{\pm.08}$	$\underline{10.43}_{\pm.04}$	$\underline{-157.40}_{\pm.19}$	-
	MADE - Germain et al. [2015]	$3.08_{\pm.03}$	$-3.56_{\pm.04}$	$20.98_{\pm.02}$	$15.59_{\pm.50}$	$-148.85_{\pm.28}$	$2.04_{\pm.01}$
(b)	MAF - Papamakarios et al. [2017]	$-0.24_{\pm.01}$	$-10.08_{\pm.02}$	$17.70_{\pm.02}$	$11.75_{\pm.44}$	$-155.69_{\pm.28}$	$1.89_{\pm.01}$
	TAN - Oliva et al. [2018]	$\underline{-0.60}_{\pm.01}$	$\underline{\mathbf{-12.06}}_{\pm.02}$	$\underline{\mathbf{13.78}}_{\pm.02}$	$\underline{11.01}_{\pm.48}$	$\underline{\mathbf{-159.80}}_{\pm.07}$	$\underline{1.19}$
	NAF - Huang et al. [2018]	$-0.62_{\pm.01}$	$-11.96_{\pm.33}$	$15.09_{\pm.40}$	$\underline{\mathbf{8.86}}_{\pm.15}$	$-157.73_{\pm.30}$	-
(c)	B-NAF - De Cao et al. [2019]	$-0.61_{\pm.01}$	$\underline{-12.06}_{\pm.09}$	$14.71_{\pm.38}$	$8.95_{\pm.07}$	$-157.36_{\pm.03}$	-
	SOS - Jaini et al. [2019]	$-0.60_{\pm.01}$	$-11.99_{\pm.41}$	$15.15_{\pm.1}$	$8.90_{\pm.11}$	$-157.48_{\pm.41}$	1.81
	UMNN-MAF (ours)	$\underline{\mathbf{-0.63}}_{\pm.01}$	$-10.89_{\pm.7}$	$\underline{13.99}_{\pm.21}$	$9.67_{\pm.13}$	$\underline{-157.98}_{\pm.01}$	$\underline{\mathbf{1.13}}_{\pm.02}$

Fun Facts

1. The numerical integration is performed with static Clenshaw-Curtis method which is proven to converge for Lipschitz continuous functions.
2. The backward computation is performed by solving numerically another integral coming from the Leibniz integral rule which leads to:

$$\nabla_\psi F(x;\psi) = \int_0^x \nabla_\psi f(t;\psi) + \nabla_\psi \beta.$$