Approximation du flot de courbure moyenne et apprentissage par réseaux de neurones

Garry Terii

Institut Camille Jordan
Université Lyon 1

Travail en collaboration avec Elie Bretin (ICJ/INSA), Roland Denis (ICJ), Simon Masnou (ICJ).

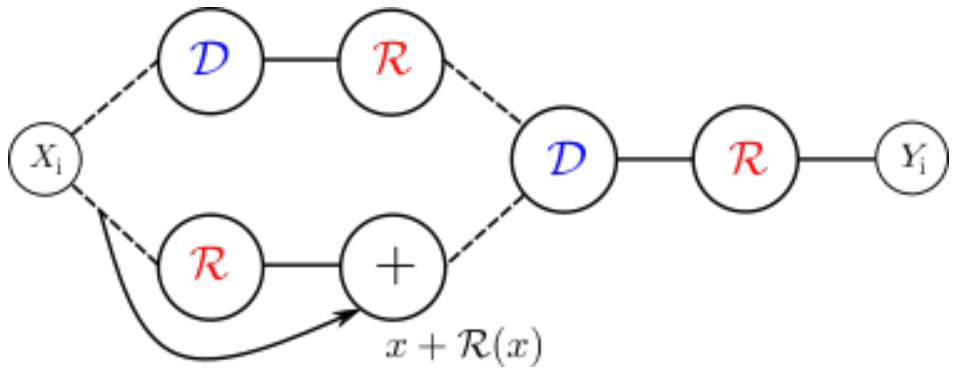
Congrès SMAI
24 juin 2021

Motivation

Point de départ : approximation champ de phase du flot de courbure moyenne

Objectifs :

- Obtenir une méthode numérique plus rapide et plus précise
- Traiter des cas pour lesquels la méthode champ de phase n'est pas adaptée (surfaces non orientées, surfaces à bord, ensembles de codimension ≥ 2)
- Apprendre un flot géométrique et ses propriétés à partir d'observations



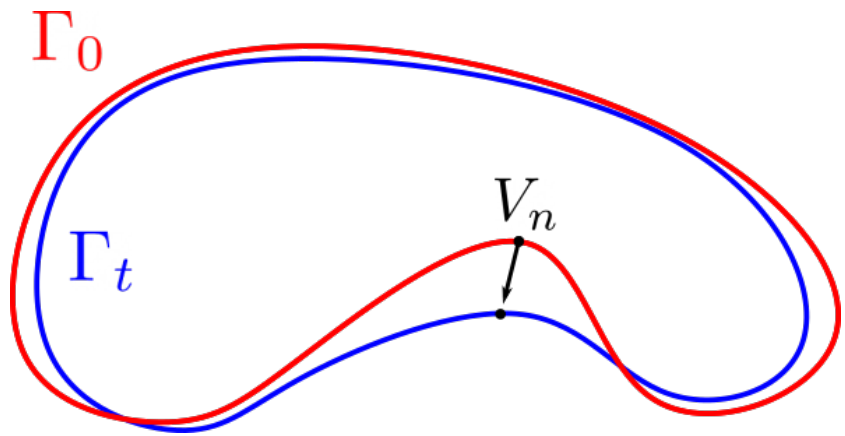
Mouvement par courbure moyenne

C'est le mouvement qui minimise le périmètre:

$$P(\Omega) = \int_{\partial\Omega} d\mathcal{H}^{d-1}$$

L'interface évolue suivant la loi:

$$V_n = H$$



Formulation champ de phase du flot par courbure moyenne

Pour approcher le flot par courbure moyenne, on introduit l'équation d'Allen-Cahn

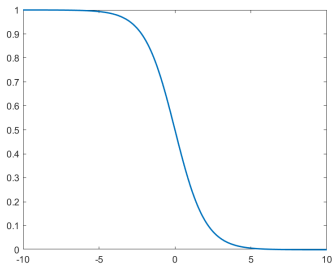
$$\partial_t u = \Delta u - \frac{1}{\varepsilon^2} W'(u)$$

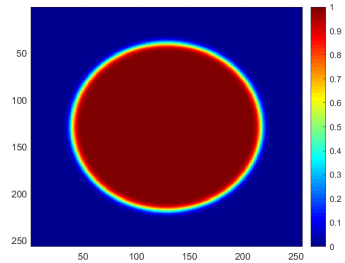
où $\varepsilon > 0$ est fixé et $W(s) = \frac{1}{2} s^2 (1-s)^2$ est le potentiel double puits.

Limite quand $\varepsilon \to 0$:

$$u_\varepsilon(t,x) = q\left(\frac{d(x,\Omega(t))}{\varepsilon}\right) + \mathcal{O}(\varepsilon^2)$$

$$V_\varepsilon = H + \mathcal{O}(\varepsilon^2)$$





où d est la fonction distance signée et q le profil optimal associé au potentiel double puits.

Discrétisation : méthode de Splitting

$$\begin{cases} \partial_t u(x,t) = \Delta u(x,t) - \dfrac{1}{\varepsilon^2} W'(u(x,t)) \\ u(x,0) = u_0 \end{cases}$$

Pour $\delta t > 0$ fixé, on obtient le schéma:

$$\frac{u^{n+1} - u^n}{\delta t} = \Delta u^{n+1} - \frac{1}{\varepsilon^2} W'(u^n)$$

qui se réécrit

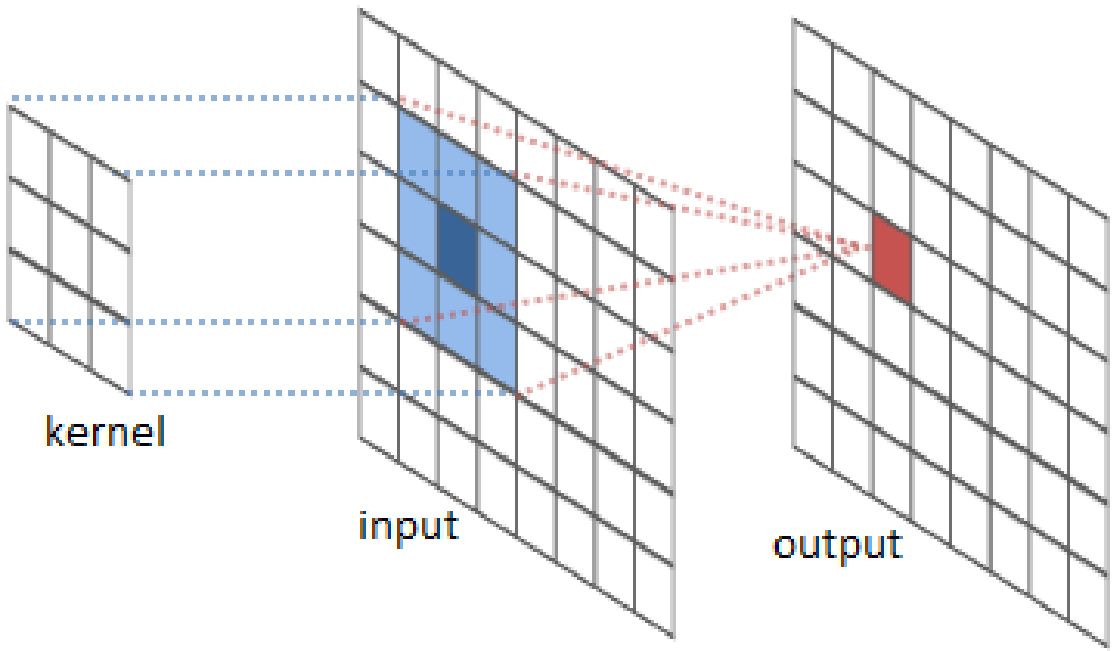
$$\begin{aligned} u^{n+1} &= (Id - \delta t \Delta)^{-1} \left(u^n - \frac{\delta t}{\varepsilon^2} W'(u^n) \right) \\ &= K_{\delta t} * \rho(u^n) \end{aligned}$$

où $K_{\delta t}(x) = \mathcal{F}^{-1}\left(\frac{1}{1+4\delta t \pi^2 |\cdot|^2}\right)(x)$ et $\rho(s) = s - \frac{\delta t}{\varepsilon^2} W'(s)$.

Approximation par les réseaux de neurones : construction

On s'inspire du splitting précédent et on construit un réseau S obtenu comme composition d'un réseau convolutif ($\mathcal{D}$) et un perceptron multicouche ($\mathcal{R}$).

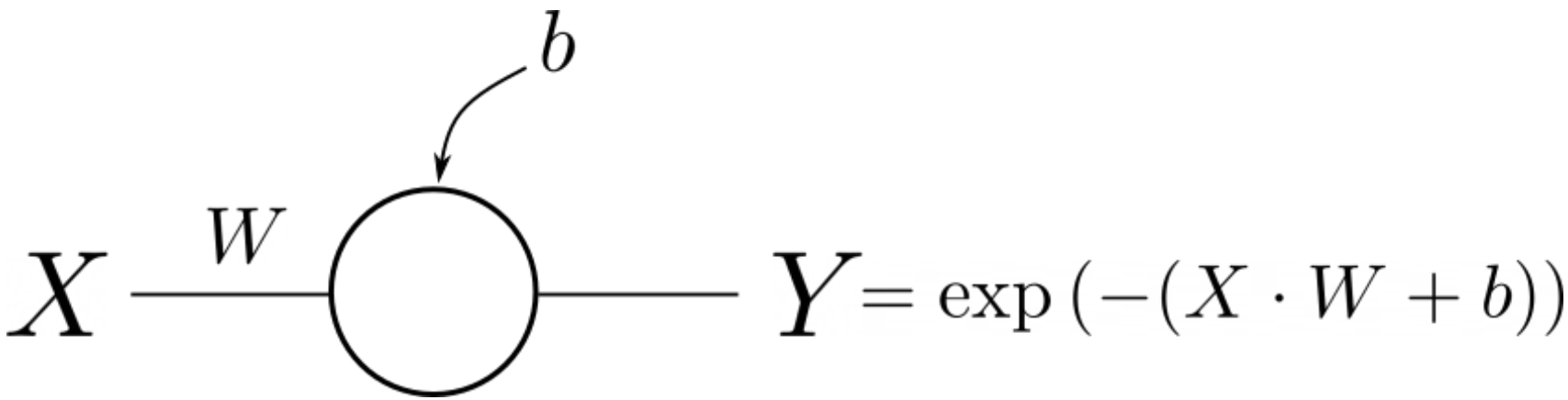
Réseau de Convolution: $u^{n+1} = \mathcal{D}u^n = K * u^n$.



Approximation par les réseaux de neurones : construction

On s'inspire du splitting précédent et on construit un réseau S obtenu comme composition d'un réseau convolutif ($\mathcal{D}$) et un perceptron multicouche ($\mathcal{R}$).

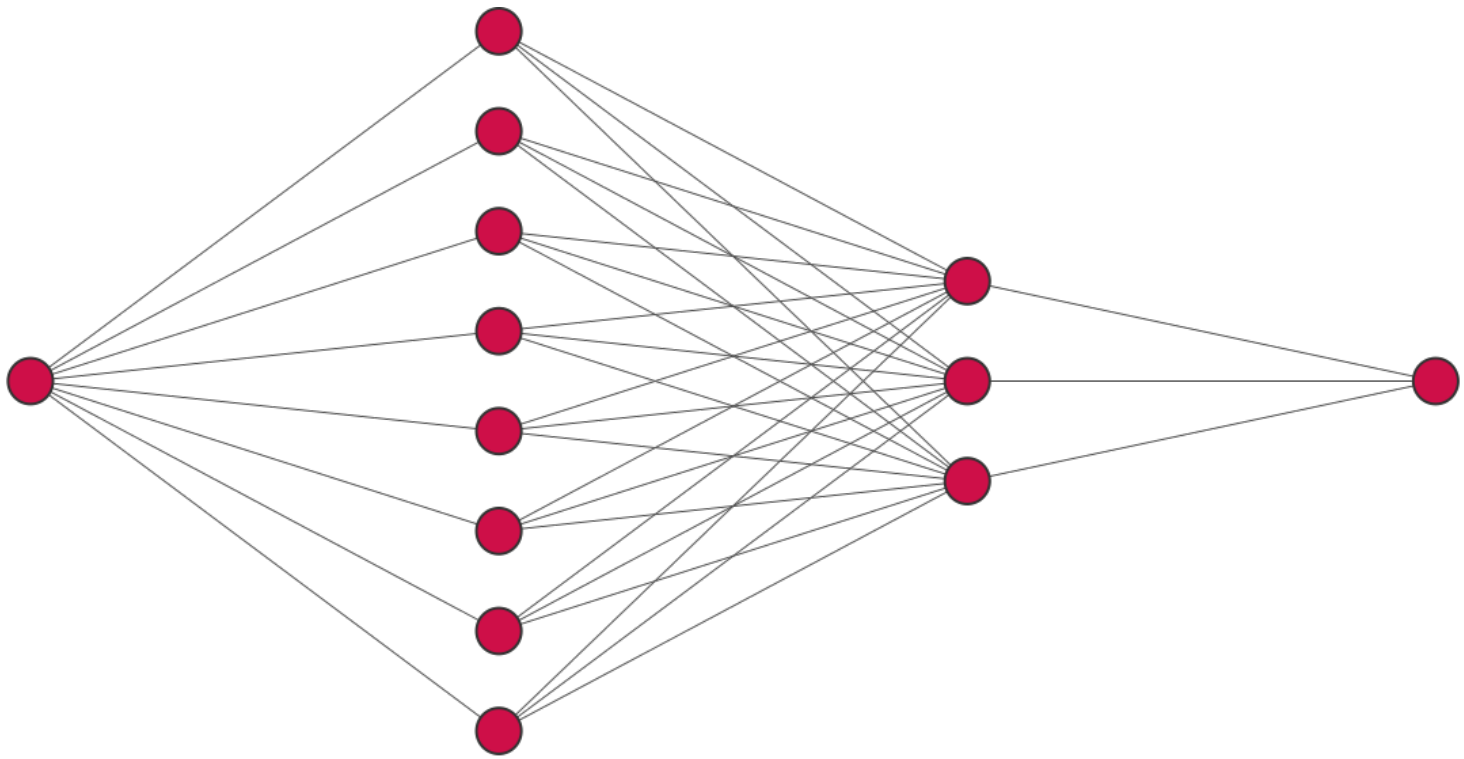
Perceptron:



Approximation par les réseaux de neurones : construction

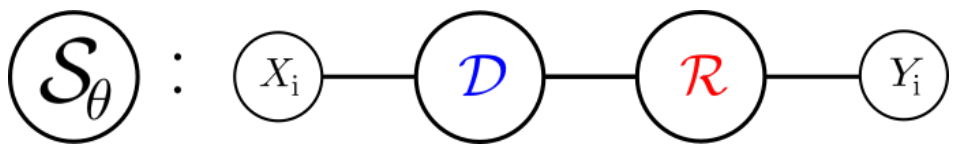
On s'inspire du splitting précédent et on construit un réseau S obtenu comme composition d'un réseau convolutif ($\mathcal{D}$) et un perceptron multicouche ($\mathcal{R}$).

Perceptron multicouche: $u^{n+1} = \mathcal{R}u^n$.



Approximation par les réseaux de neurones : construction

On construit ainsi notre premier réseau de neurones S_θ qui combine un réseau convolutif ($\mathcal{D}$) et un perceptron multicouche ($\mathcal{R}$). :



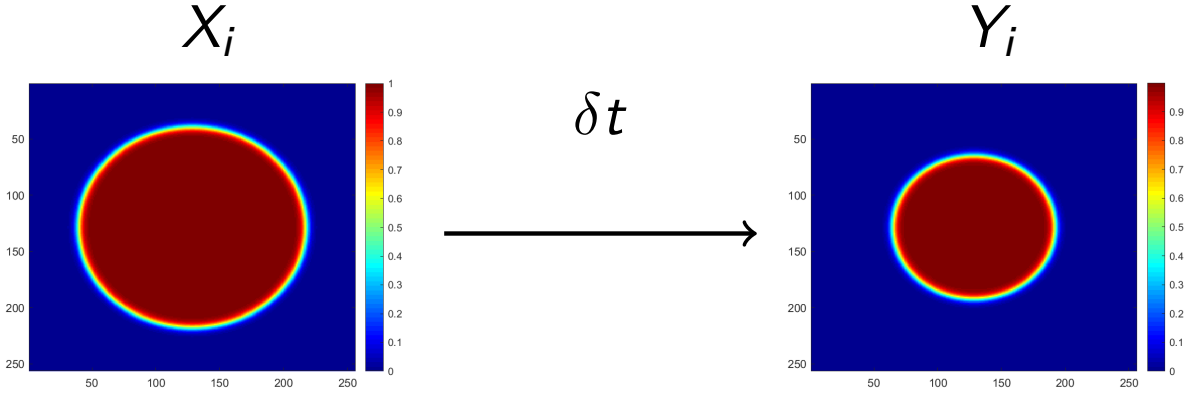
où θ est le vecteur des paramètres d'apprentissage. Il contient tous les poids des réseaux $\mathcal{D}$ et $\mathcal{R}$.

Base de données et apprentissage

Données labellisées: On fixe ε et δt.

$$(X_i, Y_i) = \left(q\left(\frac{d(.,\Omega_i(0)}{\varepsilon}\right), q\left(\frac{d(.,\Omega_i(\delta t)}{\varepsilon}\right)\right), \ i = 1, \ldots, N$$

où $\Omega_i(t)$ est un cercle de rayon $R_i(t) = \sqrt{r_i - 2t}$.

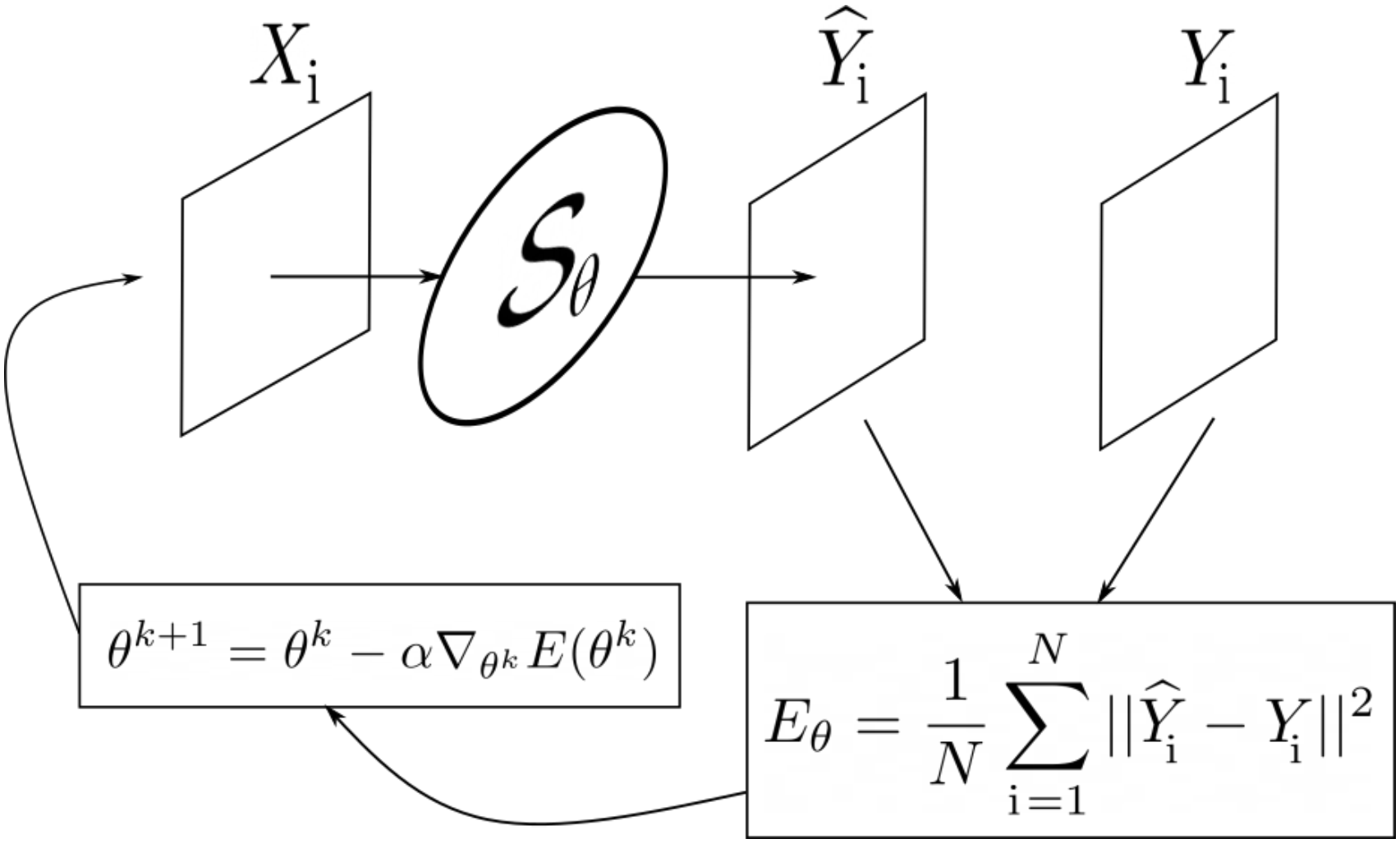


Apprentissage sur le risque empirique: Pour un réseau S_θ paramétré par θ,

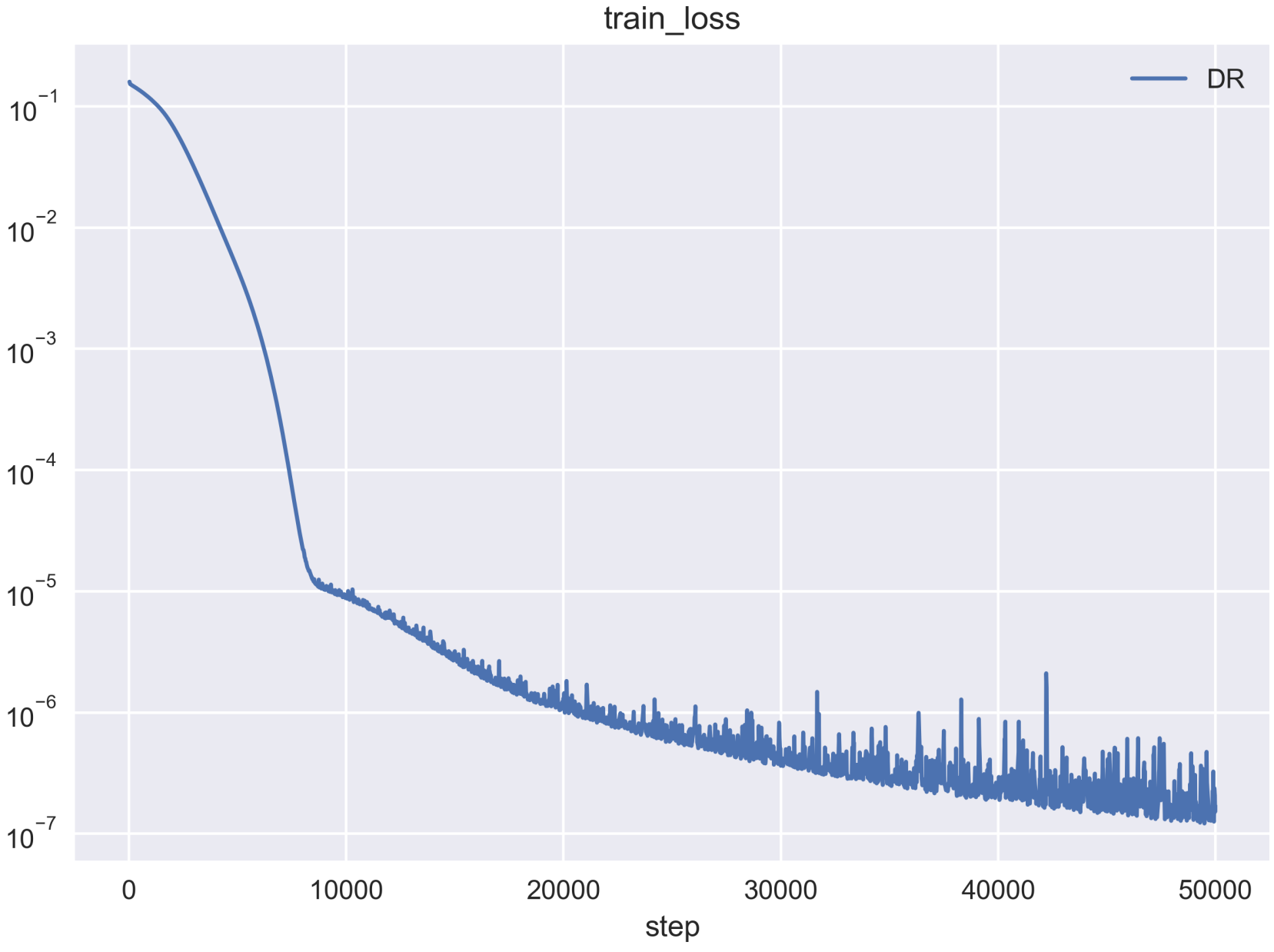
$$\inf_\theta \frac{1}{N} \sum_{i=1}^{N} \|S_\theta(X_i) - Y_i\|_{L^2}^2$$

Apprentissage : principe

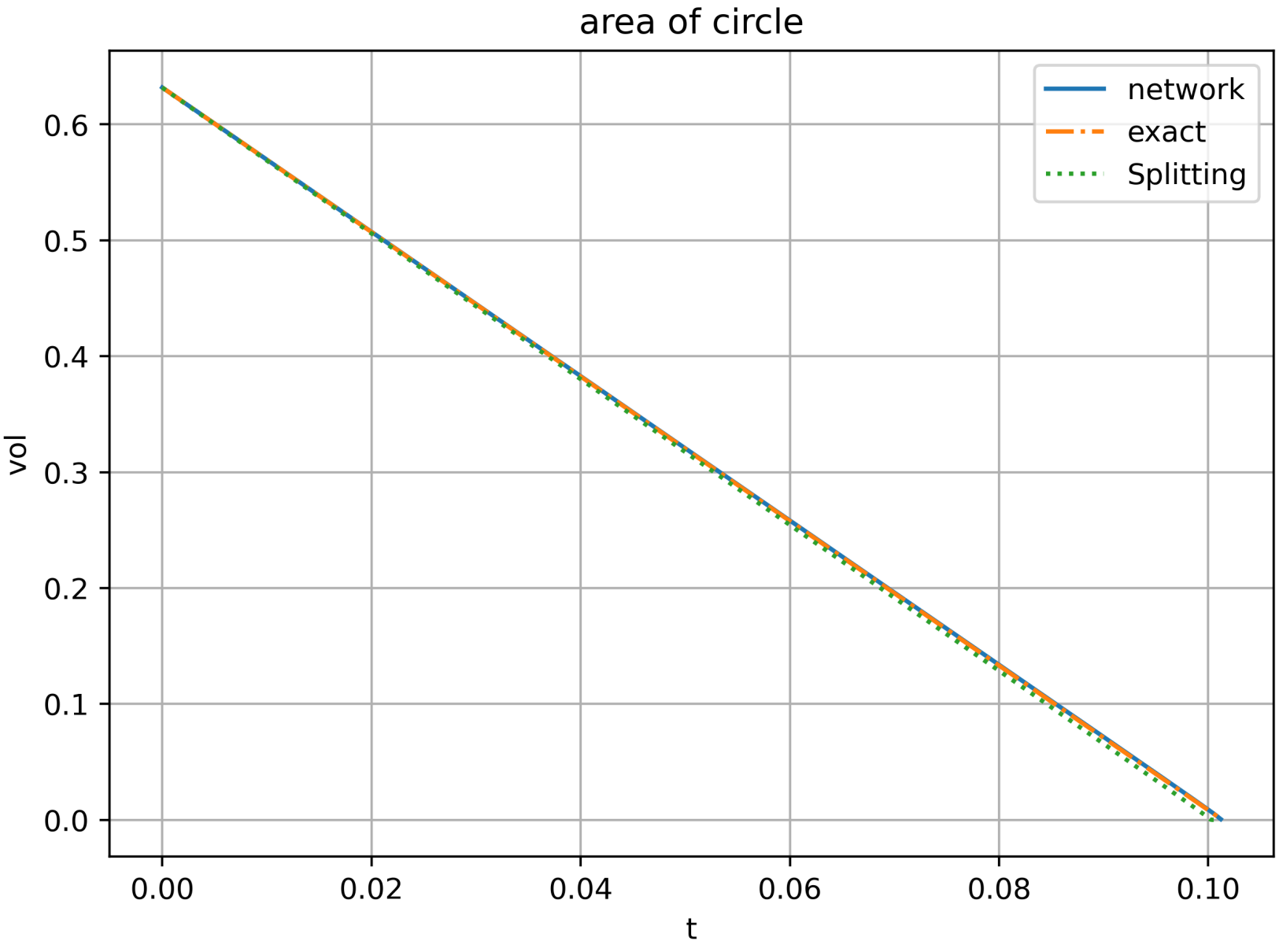
Librairie: **Pytorch** Framework: **Pytorch-Lightning**



Apprentissage avec le réseau $\mathcal{DR}$: fonctions coût



Validation de la méthode

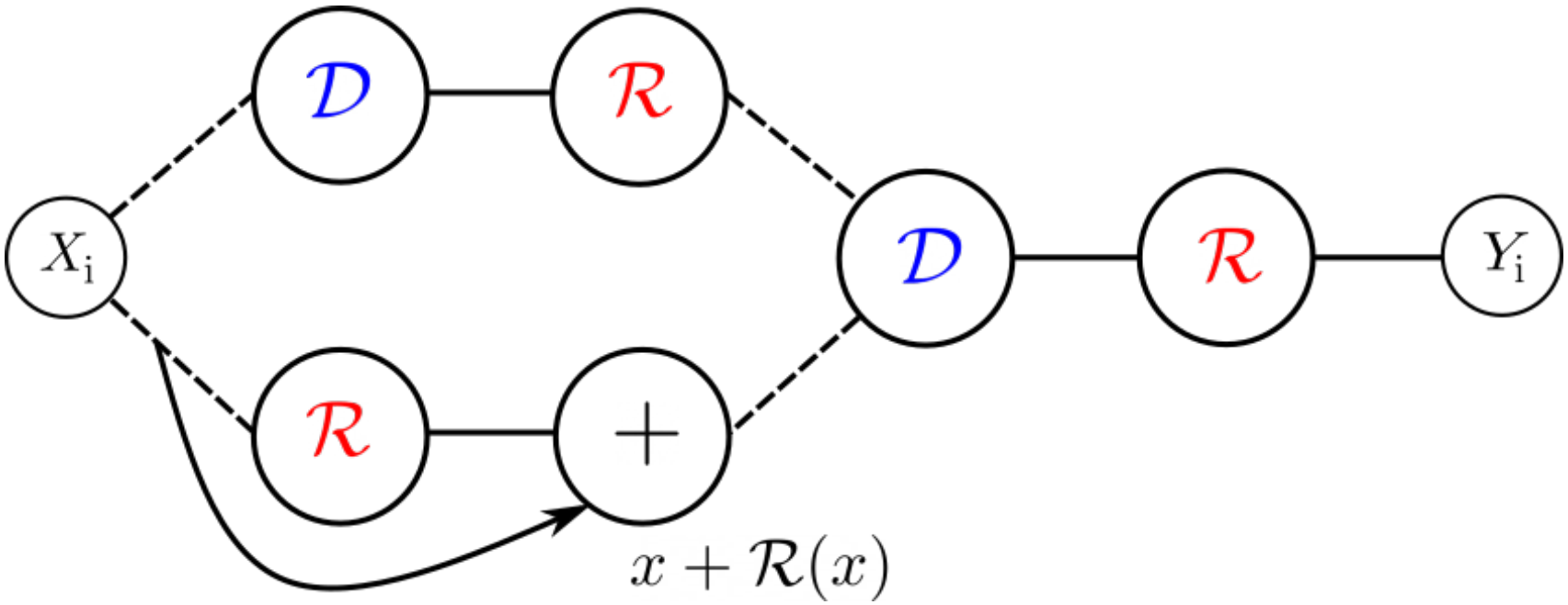


Simulations

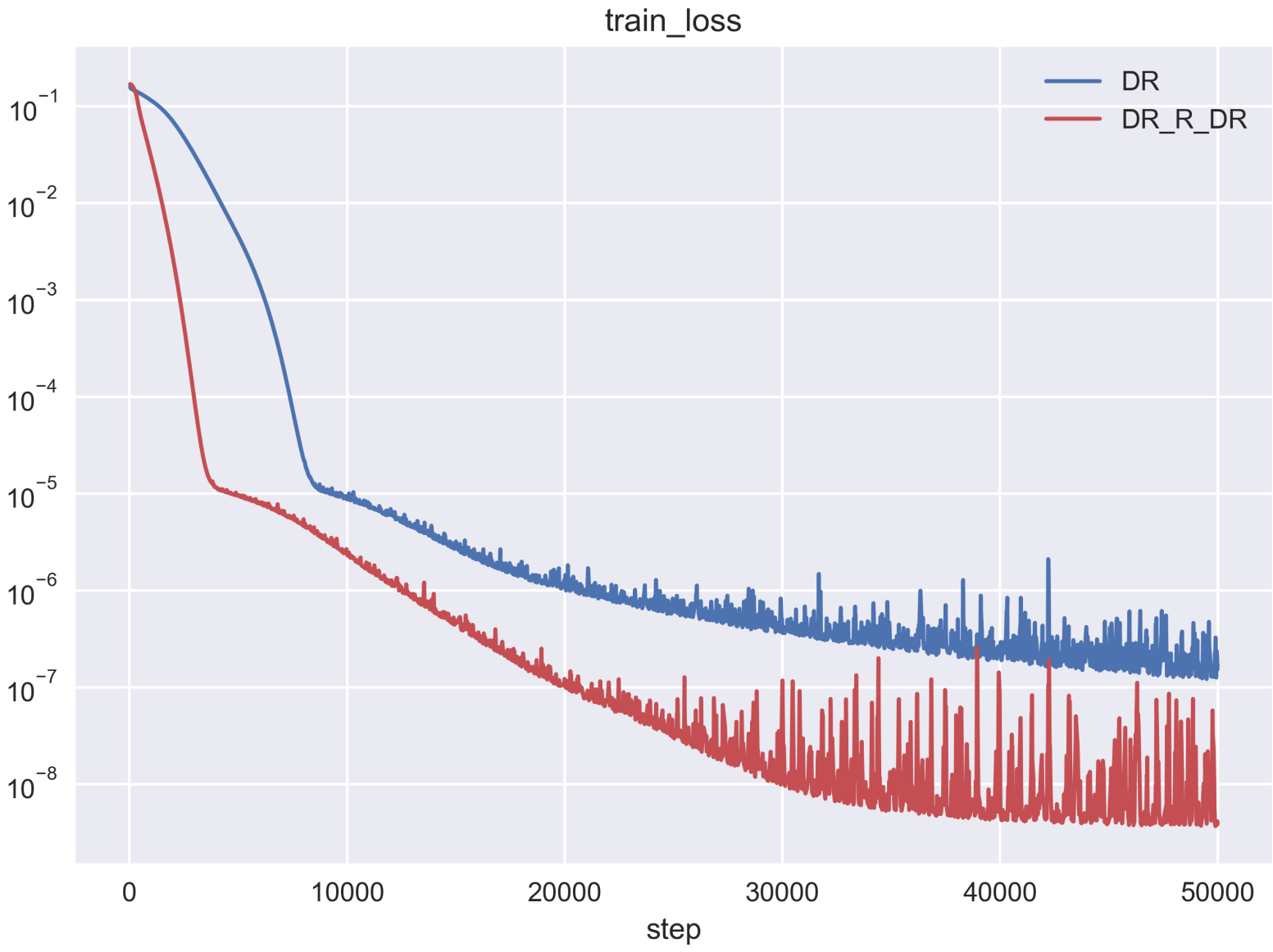
Cercle

Forme non convexe

Un réseau de neurone plus complexe



Comparaison des deux réseaux de neurones précédents



Application : flot par courbure moyenne avec conservation du volume

Le mouvement par courbure moyenne avec conservation du volume est défini par la loi :

$$V = H + \fint H$$

On adapte la méthode champ de phase associée au mouvement par courbure moyenne conservé :

$$\begin{cases} u^{n+1/2} = \mathcal{S} u^n \\ u^{n+1} = \lambda^n \sqrt{2W(u^{n+1/2})} \end{cases}$$

où λ^n est le multiplicateur de Lagrange associé à la contrainte

$$\int u^{n+1} = \int u^n$$

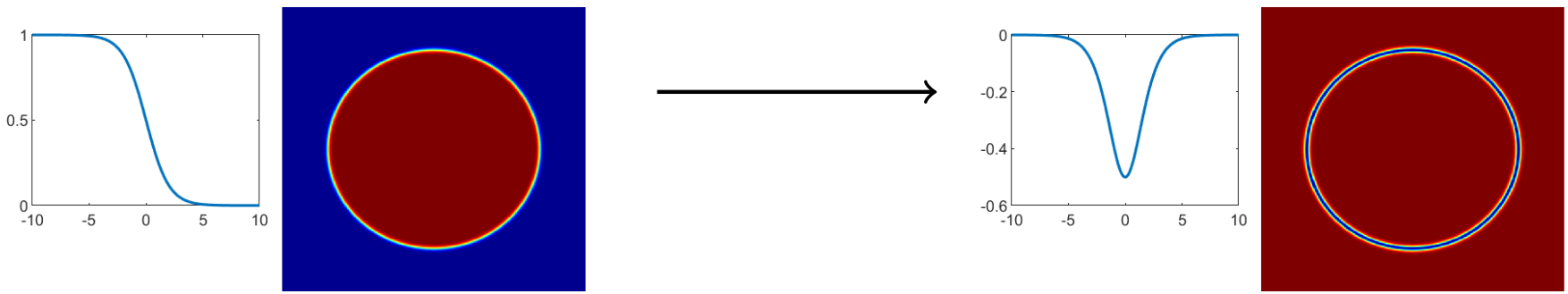
Application : simulation

Conservation sur deux cercles

Mouvement par courbure moyenne dans le cas d'interfaces non orientées

On part du même réseau $\mathcal{S}$ que précédemment et on change la base d'entraînement.

q $\quad$ $q\,(d/\varepsilon)$ $\qquad\qquad$ q' $\quad$ $q'\,(d/\varepsilon)$



Validation du modèle sur le cercle

validation sur un cercle

Simulations: points triples

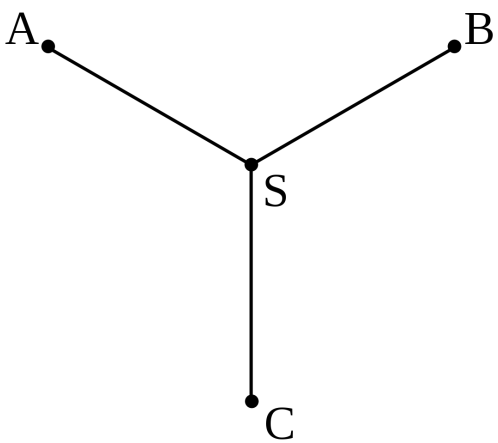
Trois cercles

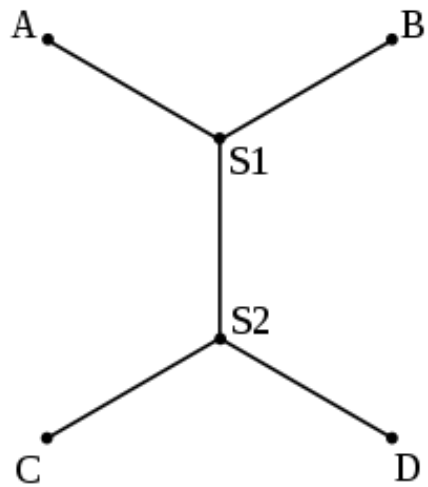
Application I : flot par courbure moyenne avec contrainte d'inclusion

Problème de Steiner: pour $x_1, x_2, \ldots, x_N \in \Omega$ fixés,

$$\min_{K \subset \Omega} P(K)$$

sous la contrainte K compact connexe contenant les $x_1, x_2, \ldots, x_N$.





Application I : flot par courbure moyenne avec contrainte d'inclusion

Problème de Steiner: pour $x_1, x_2, \ldots, x_N \in \Omega$ fixés,

$$\min_{K \subset \Omega} P(K)$$

sous la contrainte K compact connexe contenant les $x_1, x_2, \ldots, x_N$.

Steiner à 4 points

Steiner à 5 points

Steiner à 7 points

Application II : flot par courbure moyenne en codimension 2 dans $\mathbb{R}^3$

Codimension 2

Merci de votre attention!

Anisotropie