Apprentissage de flot

Emmanuel Franck[*],

September 16-20, 2024
Master CMSI, M2, Strasbourg

[*] MACARON project-team, Université de Strasbourg, CNRS, Inria, IRMA, France

Outline

Apprentissage de flot

Apprentissage de flot discret

Résumé

Jusqu'a présent on a regardé comment approcher une EDO qu'on discrétise ensuite pour obtenir un flot discret. On peut aussi apprendre directement le flot.

- On se donne l'EDO:

$$\frac{d\mathbf{x}(t)}{dt} = F(\mathbf{x}(t); \boldsymbol{\mu})$$

  avec $\boldsymbol{\mu}$ des paramètres physiques?
- La première solution est d'apprendre le flot discret

$$\Phi_{\theta,\Delta t}(\mathbf{x}, \boldsymbol{\mu})$$

- Fonction de coût:

$$\min_\theta \sum_{i=1}^{N} \sum_{t=1}^{T} \| \mathbf{x}_{n+1}^i - \Phi_{\theta,\Delta t}(\mathbf{x}_n^i, \boldsymbol{\mu}_i) \|_2^2$$

Flot discret et déroulement

- Ce flot discret peut être appris pour un Δt donné. On peut aussi essayer d'apprendre une dépendance en Δt avec

$$\Phi_{\theta,\Delta t}(\mathbf{x}, \boldsymbol{\mu}) = I_d + \Delta t MLP_\theta(\mathbf{x}, \boldsymbol{\mu}, \Delta t)$$

- En inférence on compose le flot appris T fois. Pas d'assurance de stabilité.
- En utilisant la programmation différentiable et le mode VJP on peut minimiser et "dérouler l'apprentissage":

$$\min_\theta \sum_{i=1}^{N} \sum_{t=1}^{T} \| \mathbf{x}^i_{n+k} - \Phi_{\theta,\Delta t} \circ ... \circ \Phi_{\theta,\Delta t}(\mathbf{x}^i_n, \boldsymbol{\mu}_i) \|_2^2$$

  avec $k > 1$.
- Pour des problèmes autonome on peut choisir $k < T$. En effet on a pas besoin de considérer des trajectoires complètes.

Apprentissage de flot continue

Flot continue vs flot discret

Les flots discrets peuvent être limités par les instabilités issues de leurs composition. Le déroulement de l'apprentissage peut être une solution. Une autre est d'apprendre directement:

$$\varphi_\theta(\mathbf{x}; t; \boldsymbol{\mu})$$

afin d'éviter la composition.

- Fonction de coût:

$$\min_\theta \sum_{i=1}^{N} \sum_{t=1}^{T} \| \mathbf{x}_{n+1}^i - \varphi_\theta(\mathbf{x}_n^i; t_n, \boldsymbol{\mu}_i) \|_2^2$$

- Le flot est une application avec une structure particulière qu'on ne respect pas ici.
- **Exemple**: La structure de semi groupe est plus ou moins imposer lorsqu'on compose des flots discrets. Ici on en tient nullement compte.

Inversibilité

Couche inversible

Soit la paramétrisation suivante

$$\varphi_{loc,\theta}(\mathbf{x}; t, \boldsymbol{\mu}) = I_d + \phi(t) g_\theta(\mathbf{x}, \boldsymbol{\mu})$$

avec $\phi(0) = 0$ et g_θ une couche de constante de Lipchitz < 1. Cette paramétrisation est inversible

- Soit la transformation direct $\mathbf{y} = \mathbf{x} + \phi(t) g_\theta(\mathbf{x}, \boldsymbol{\mu})$
- Si on sait résoudre le point fixe $\mathbf{y} - \phi(t) g_\theta(\mathbf{x}, \boldsymbol{\mu}) = \mathbf{x}$ on a une transformation inverse qui passe de $\mathbf{y}$ à $\mathbf{x}$
- Le point fixe peut être résolu si g_θ est contractante.

Flot inversible

La composition de fonctions inversible étant inversible

$$\varphi_\theta(\mathbf{x}; t, \boldsymbol{\mu}) = \varphi^L_{loc,\theta} \circ \ldots \circ \varphi^1_{loc,\theta}$$

donnera un réseau inversible.

Réseau inversible

- Pour avoir notre bloc $\varphi_{loc,\theta}$ inversible il faut donc avoir une bonne constante de Lipchitz (par rapport à $\mathbf{x}$) pour g_θ
- Comment construire g_θ. Supposons

$$g_\theta = \sigma \circ L_3 \circ \sigma \circ L_2 \circ \sigma \circ L_1$$

avec

$$L(\mathbf{x}, \boldsymbol{\mu}) = W_x \mathbf{x} + W_\mu \boldsymbol{\mu} + \mathbf{b}$$

- Si σ Lipchitz (comme Tanh) alors on a:

$$C_{Lip}(g_\theta) \leqslant \| W_{x,3} \|_2 \| W_{x,2} \|_2 \| W_{x,1} \|_2$$

Solution

On remplace les matrices de poids W_x par

$$\hat{W}_x = c \frac{W_x}{\sigma_1}$$

avec $0 < c < 1$ et σ_1 une estimation de la plus grande valeur singulière obtenue par une méthode de la puissance différentiable.

Structure de semi-groupe

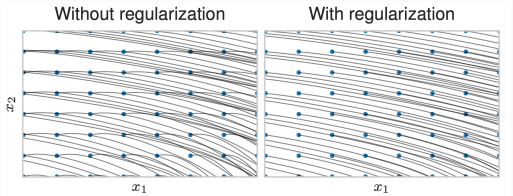
Un flot associé à une EDO autonome satisfait:

$$\varphi(\mathbf{x}_0; t+s, \boldsymbol{\mu}) = \varphi(\varphi(\mathbf{x}_0; t, \boldsymbol{\mu}); s, \boldsymbol{\mu})$$

- On souhaiterait que le flot appris satisfasse cette propriété fondamentale.
- Pas de solution pour l'imposer dans le réseau.
- On utilise donc une fonction de coût additionnelle:

$$\gamma \frac{1}{n} \sum_i \left(\varphi_\theta(\mathbf{x}_i; t_i, \boldsymbol{\mu}) - \varphi_\theta\left(\varphi_\theta\left(\mathbf{x}_i; t_i^{(1)}, \boldsymbol{\mu} \right); t_i^{(2)}, \boldsymbol{\mu} \right) \right)^2$$

avec $t_i = t_i^{(1)} + t_i^{(2)}$



Apprentissage de flot géométrique

Apprentissage de flot géométrique

Flot hamiltonien

Comme introduit dans le cours numéro 2, le flot d'une EDO Hamiltonienne est symplectique ce qui pour conséquence la conservation du volumes de l'ensemble des conditions initiales.

- Il est donc naturel et intéressant et construire un flot symplectique.
- Comment faire ?
- **1er solution**: par pénalisation. On ajoute la fonction de coût:

$$\sum_{i=1}^{N}\sum_{t=1}^{T} \parallel \nabla_{\mathbf{x}}\varphi_\theta(\mathbf{x}_n^i; t_n, \boldsymbol{\mu}_i)^t \mathcal{J} \nabla_{\mathbf{x}}\varphi_\theta(\mathbf{x}_n^i; t_n, \boldsymbol{\mu}_i) - \mathcal{J} \parallel_2^2$$

- **2ème solution**: en imposant la structure symplectique dans le réseau.
- Comment construire des réseaux de neurones symplectique ?

Flot symplectique et splitting

Solution

En s'inspirant de la construction des schémas symplectiques qui nous donne des flots discrets.

- On cherche a construire un symplectomorphisme $S(\mathbf{x})$. La théorie nous dit qu'il existe un Hamiltonien H tel que le flot $\varphi_H = S$.
- On cherche donc a approcher le flot d'un Hamiltonien H. Comment ? par une méthode de splitting.
- Supposons qu'on peut écrire $H = \sum_{i=1}^{K} H_i$ alors

$$\varphi_{H,\Delta t} \approx \varphi_{\hat{H},\Delta t} = \varphi_{H_K,\Delta t} \circ \ldots \circ \varphi_{H_1,\Delta t}$$

avec

$$\hat{H} = \sum_{i=1}^{K} H_i + \frac{\Delta t}{2}[H_i, H_j] + O(\Delta t^2)$$

et $[H_i, H_j] = (\nabla_{\mathbf{x}} H_i)^t \mathcal{J} (\nabla_{\mathbf{x}} H_j)$.

- On obtient ce résultat par analyse des schémas de splitting et la formule BCH.

Couche Symplectique et splitting

Réseaux de neurones symplectique

Soit une fonction élémentaire $\varphi_{H_{\theta_i},\Delta t}$ qui est le flot exact associé a un Hamiltonien paramétrique H_{θ_i}. La composition de ses fonction élémentaires

$$\varphi_{\hat{H},\Delta t} = \varphi_{H_K,\Delta t} \circ \ldots \circ \varphi_{H_1,\Delta t}$$

nous donnera donc un réseau de neurones symplectique associé à l'Hamiltonien

$$\hat{H} = \sum_{i=1}^{K} H_i + \sum_{I<j\leqslant K} \frac{\Delta t}{2}[H_i, H_j] + O(\Delta t^2).$$

On appellera $\varphi_{H_{\theta_i},\Delta t}$ une couche symplectique.

- On peut démontrer la densité de ses réseaux symplectiques dans l'espace des symplectomorphisme.
- Comment construire les couches symplectiques ?

SympNet I

- Pour construire une couche il faut donc construire le flot exact associé à H_{θ_i}.
- Comment faire ? **Solution 1**: encore par par splitting.
- Supposons

$$H_{\theta_i}(\mathbf{x}) = H_{\theta_i}(\mathbf{q}, \mathbf{p}) = K_{\theta_i}(\mathbf{p}) + U_{\theta_i}(\mathbf{q})$$

- Comme vu dans le cours 2: si on applique un splitting les flots d'Euler de chaque sous système sont exacts et donc symplectique.
- En paramétrant K_{θ_i}, U_{θ_i} + splitting et flot d'Euler on a des couches symplectiques.

Couche symplectique linéaire

La couche linéaire est obtenue en choisissant $K_{\theta_i}(\mathbf{p}) = (\mathbf{p}, A\mathbf{p})+$ et $U_{\theta_i}(\mathbf{q}) = (\mathbf{q}, A\mathbf{q})$. Cela donne la couche $\phi_{La}(\mathbf{q}, \mathbf{p}) = L^2 \circ L^1$ avec

$$L^2 = \begin{pmatrix} I_d & S \\ 0 & Id \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ \mathbf{p} \end{pmatrix} + \mathbf{b}, \quad L^1 = \begin{pmatrix} I_d & 0 \\ S & Id \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ \mathbf{p} \end{pmatrix} + \mathbf{b}$$

avec $\mathbf{b}$ et A les paramètres entrainables. $\mathbf{b}$ est ajouté après.

Couche symplectique d'activation

La couche d'activation est obtenue en choisissant $K_{\theta_i}(\mathbf{p}) = diag(\mathbf{a})\Sigma(\mathbf{p})$ et $U_{\theta_i}(\mathbf{q}) = diag(\mathbf{a})\Sigma(\mathbf{q})$ avec Σ la primitive de σ. Cela donne la couche $\phi_{La}(\mathbf{q},\mathbf{p}) = L^2 \circ L^1$ avec

$$L^2 = \begin{pmatrix} \mathbf{q} \\ \mathbf{p} + diag(\mathbf{a})\sigma(\mathbf{q}) + \mathbf{b}, \end{pmatrix} \quad L^1 = \begin{pmatrix} \mathbf{q} + diag(\mathbf{a})\sigma(\mathbf{p}) + \mathbf{b}, \\ \mathbf{p} \end{pmatrix}$$

avec **b** et **a** les paramètres entrainables. **b** est ajouté après.

Couche symplectique gradient

La couche est obtenue en choisissant $K_{\theta_i}(\mathbf{p}) = diag(\mathbf{a})\Sigma(K\mathbf{p} + \mathbf{b})$ et $U_{\theta_i}(\mathbf{q}) = diag(\mathbf{a})\Sigma(K\mathbf{q} + \mathbf{b})$. Cela donne la couche $\phi_{La}(\mathbf{q},\mathbf{p}) = L^2 \circ L^1$ avec

$$L^2 = \begin{pmatrix} \mathbf{q} \\ \mathbf{p} + K^t diag(\mathbf{a})\sigma(K\mathbf{q} + \mathbf{b}), \end{pmatrix} \quad L^1 = \begin{pmatrix} \mathbf{q} + K^t diag(\mathbf{a})\sigma(K\mathbf{p} + \mathbf{b}), \\ \mathbf{p} \end{pmatrix}$$

avec K, **b** et **a** les paramètres entrainables. **b** est ajouté après.

- Le caractère symplectique de chaque couche est obtenu car de composition de flots exacts pour des sous systèmes.
- En enchainant les couches de linéaire et d'activation on obtient des réseaux dense dans les symplectomorphismes aux moins C^1 appelé LA SympNet.
- Idem pour les couches Gradient qui donne un G-SympNet

Autre solution

Une autre solution pour construire des flots exacte et donc de nouvelles couches est de prendre des Hamiltoniens nilpotent de degré 2 ce qui vaut a une dérivée seconde nulle

$$\ddot{\mathbf{x}}(t) = \frac{d}{dt}\mathcal{J}^{-1}\nabla H(\mathbf{x}) = \mathcal{J}^{-1}\nabla^2 H(x)\mathcal{J}^{-1}\nabla H(x) = 0.$$

En effet les flot discret d'Euler explicite sont exacte si les dérivées supérieures à deux sont nulles.

(Feng and Wang, 1998)

Une fonction Hamiltonienne $H : \mathbb{R}^{2n} \to \mathbb{R}$ est nilpotent de degré deux si

$$H(\mathbf{x}) = K(C\mathbf{x}), \text{ avec } C\mathcal{J}C^T = 0$$

avec $K : \mathbb{R}^n \to \mathbb{R}$ et $C \in \mathbb{R}^{n \times 2n}$.

- Dans ce cas le flot discret d'Euler est exacte donc symplectique
$$\varphi(\mathbf{x}) = \mathbf{x} + h\mathcal{J}^{-1}C^t K(C\mathbf{x})$$
- Paramétrisation possible:
$$C = \begin{pmatrix} \vdots & 0 & \mathbf{w} & 0 & \vdots \end{pmatrix}, \quad H(\mathbf{x}) = \alpha(\langle \mathbf{w}, \mathbf{x} \rangle)$$

R-P SympNet

Une couche R-P sympNet est de la forme

$$\varphi_\theta(\mathbf{x}) = \mathbf{x} + h\alpha'_{\theta_a}(\langle \mathbf{w}, \mathbf{x} \rangle)\mathcal{J}^{-1}\mathbf{w}$$

avec α un perceptron à une couche (R-SympNet) ou α un polynôme (P-SympNet) et $\theta = (\mathbf{w}, \theta_a)$.

Flots symplectiques continues

- Le flot généré est associé a un Hamiltonien de la forme:

$$\hat{H} = \sum_{i=1}^{K} H_i + O(\Delta t)$$

- Les méthodes G et LA sympNet vont générer des Hamiltoniens $\hat{H}$ séparables contrairement aux approches P et R.

Flots continus

On peut transformer ses architectures pour obtenir des flots continues en temps. Pas besoin d'imposer l'inversibilité car une symplectomorphisme est inversible. On peut

- Exemple: G-SympNet

$$L^2 = \begin{pmatrix} \mathbf{q} \\ \mathbf{p} + \beta(t) K^t diag(\mathbf{a}) \sigma(K\mathbf{q} + M(t,\mu)^t + \mathbf{b}) \end{pmatrix} \quad L^1 = \begin{pmatrix} \mathbf{q} + \beta(t) K^t diag(\mathbf{a}) \sigma(K\mathbf{p} + M(t,\mu)^t + \mathbf{b}) \\ \mathbf{p} \end{pmatrix}$$

avec $\beta(0) = 0$.

Apprentissage de flot

Apprentissage de flot géométrique

Conclusion

Conclusion

Apprentissage d'EDO

- On peut considère trois approches pour l'apprentissage de séries temporelle issue d'un processus régulier :
  - Apprentissage supervisé de modèle: on apprend le flux de l'EDO $\mathbf{f}$ en estimant la dérivée localement (qui correspond à la sortie de $\mathbf{f}$. Cela peut être fait avec des modèles de type réseaux de neurones/méthodes à noyaux ou des modèles parcimonieux type Sindy.
  - Apprentissage par comparaison de trajectoires: ici, on apprend le modèle en cherchant à minimiser l'écart entre les trajectoires de références et celles produites par le modèle. On a deux approches de calcul de gradient: OpD (OdeNet) et DpO qui peuvent être combinées avec des réseaux ou des modèles parcimonieux.
  - Apprentissage de flots: on apprend directement un flot discret ou continue. Le flot discret revient à apprendre l'équation et le schéma
- Imposer une structure a priori (Hamiltonienne, Lagrangienne ou symplectique) donne de meilleurs résultats notamment en temps long.

Apprentissage partiel

Principe

Jusqu'a présent on a appris la totalité de l'équation donc de $\mathbf{f}$. On peut aussi apprendre une partie de l'équation ou du flot voir juste un paramètre.

Exemple: Méthode Aphynity

La méthode propose d'apprendre $\mathbf{f}_{\theta,a}$ en minimisant

$$\min_\theta (J_{min} + \lambda J_{tra})$$

avec

$$J_{tra} = \left(\sum_{i=1}^{M} \sum_{j=1}^{T-1} \| \mathbf{x}_{\theta,i}(t_j) - \mathbf{x}_i^j \|_2^2 \right)$$

et

$$J_{min} = \left(\sum_{i=1}^{M} \sum_{j=1}^{T-1} \| \mathbf{f}_{a,\theta}(\mathbf{x}_i^j) \|_2^2 \right)$$

$\mathbf{x}_\theta$ solution de $\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}_s(\mathbf{x}) + \mathbf{f}_{\theta,a}(\mathbf{x})$ et λ qui croit a chaque itération.