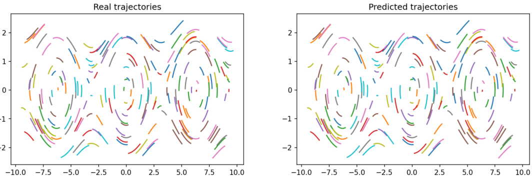
Learning the Hamiltonian of some classes of mechanical systems

Davide Murari
davide.murari@ntnu.no
Norwegian University of Science and Technology, Trondheim

NUMDIFF-16,Halle, 09-09-2021

Part of an ongoing project with Elena Celledoni, Ergys Çokaj, Andrea Leone, Brynjulf Owren.

Definition of the problem



Goal: to approximate the dynamics of a Hamiltonian vector field $X_H \in \mathfrak{X}(T^*Q)$, $T^*Q \subset \mathbb{R}^{2n}$, starting from a set of given trajectories.

Some solutions proposed in the literature:

- Learning the vector field: $f_\theta(z) \approx X_H(z)$,
- Learning the Hamiltonian: $f_\theta(z) \approx H(z) \Longrightarrow X_H(z) \approx \mathbb{J}\nabla_z f_\theta(z)$.
  1. Hamiltonian Neural Networks [1] (Greydanus et al., 2019)
  2. Symplectic Recurrent Neural Networks [2] (Chen et al., 2020)
- Learning the symplectic flow map $\Phi^{\Delta t}_{X_H}$, e.g. SympNets [3] (Jin et al., 2020)

What are Recurrent Neural Networks (RNNs)?

Made by multiple copies of the same network, each passing a message to a successor.

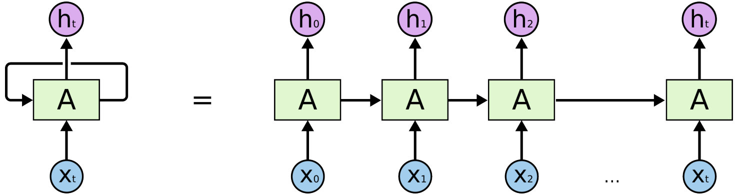


Figure 1: Source:
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

When to use RNNs?

"Whenever there is a sequence of data and that temporal dynamics that connects the data is more important than the spatial content of each individual frame." – Lex Fridman (MIT).

A Neural Network as the Hamiltonian

- **Separable Hamiltonian to learn:**

$$H(q,p) = K(p) + U(q), \quad (q,p) \in \mathbb{R}^{2n}$$

A Neural Network as the Hamiltonian

- **Separable Hamiltonian to learn:**

$$H(q,p) = K(p) + U(q), \quad (q,p) \in \mathbb{R}^{2n}$$

- **Approximation of the kinetic energy**

$$K(p) \approx K_A(p) = \frac{1}{2}\|Ap\|^2 = \frac{1}{2}p^T(A^TA)p$$

A Neural Network as the Hamiltonian

- **Separable Hamiltonian to learn:**

$$H(q,p) = K(p) + U(q), \quad (q,p) \in \mathbb{R}^{2n}$$

- **Approximation of the kinetic energy**

$$K(p) \approx K_A(p) = \frac{1}{2}\|Ap\|^2 = \frac{1}{2}p^T(A^TA)p$$

- **Approximation of the potential energy**

$$U(q) \approx U_\theta(q) = f_{\theta_m} \circ \ldots \circ f_{\theta_1}(q),$$

$$\theta_i = (W_i, b_i) \in \mathbb{R}^{n_i \times n_{i-1}} \times \mathbb{R}^{n_i}, \ \theta := [\theta_1, \ldots, \theta_m]$$

$$f_{\theta_i}(q) := \Sigma(qW_i^T + b_i), \ \mathbb{R}^n \ni z \mapsto \Sigma(z) = [\sigma(z_1), \ldots, \sigma(z_n)] \in \mathbb{R}^n,$$

and for example $\sigma(x) = \tanh(x)$.

Learning a better Hamiltonian

- **Approximation of the Hamiltonian:** $H_{A,\theta}(q,p) = K_A(p) + U_\theta(q)$,

Learning a better Hamiltonian

- **Approximation of the Hamiltonian:** $H_{A,\theta}(q,p) = K_A(p) + U_\theta(q)$,
- **Training trajectories:**

$$\{(x_i, y_i^1, ..., y_i^M)\}_{i=1,...,N},\ y_i^j = \Phi_{X_H}^{j\Delta t}(x_i)$$

Learning a better Hamiltonian

- **Approximation of the Hamiltonian:** $H_{A,\theta}(q,p) = K_A(p) + U_\theta(q)$,
- **Training trajectories:**

$$\{(x_i, y_i^1, ..., y_i^M)\}_{i=1,...,N},\ y_i^j = \Phi_{X_H}^{j\Delta t}(x_i)$$

- **Numerical Δt-flow of $X_{H_{A,\theta}}$:**

$$\hat{y}_i^j(A,\theta) := \Psi^{\Delta t}(\hat{y}_i^{j-1}(A,\theta)), \quad \hat{y}_i^0(A,\theta) := x_i$$

Learning a better Hamiltonian

- **Approximation of the Hamiltonian:** $H_{A,\theta}(q,p) = K_A(p) + U_\theta(q)$,
- **Training trajectories:**

$$\{(x_i, y_i^1, ..., y_i^M)\}_{i=1,...,N},\ y_i^j = \Phi_{X_H}^{j\Delta t}(x_i)$$

- **Numerical Δt-flow of $X_{H_{A,\theta}}$:**

$$\hat{y}_i^j(A,\theta) := \Psi^{\Delta t}(\hat{y}_i^{j-1}(A,\theta)), \quad \hat{y}_i^0(A,\theta) := x_i$$

- **Loss function:**

$$\mathcal{L}(A,\theta) := \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{m} \|\hat{y}_i^j(A,\theta) - y_i^j\|^2$$

- **Improving the approximation:**

$$\bar{A}, \bar{\theta} := \arg\min_{A,\theta} \mathcal{L}(A,\theta) \to H \approx H_{\bar{A},\bar{\theta}}.$$

Numerical methods and RNNs

- RNNs have two main features:
  1. They exploit the temporal relations in the data
  2. They are based on weight sharing between the layers

Numerical methods and RNNs

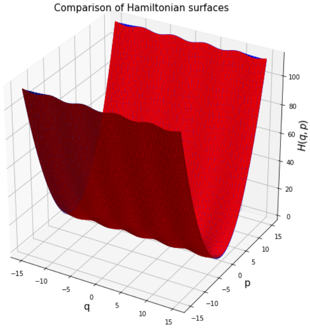
- RNNs have two main features:
  1. They exploit the temporal relations in the data
  2. They are based on weight sharing between the layers
- In this learning framework we use some numerical integrator $\Psi^{\Delta t}$ and obtain:
  - $\hat{y}_i^1 := \Psi^{\Delta t}(\hat{y}_i^0) = \Psi^{\Delta t}(x_i)$
  - $\hat{y}_i^2 := \Psi^{\Delta t}(\hat{y}_i^1)$
  - ...
  - $\hat{y}_i^M := \Psi^{\Delta t}(\hat{y}_i^{M-1})$

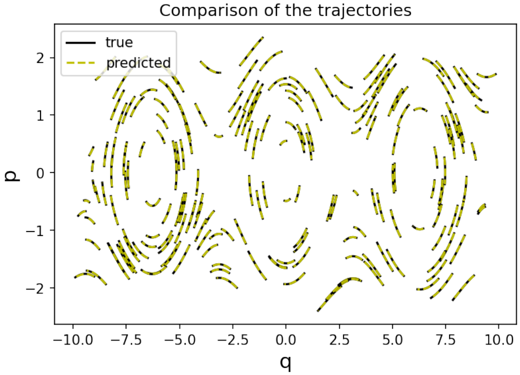
Numerical methods and RNNs

- RNNs have two main features:
  1. They exploit the temporal relations in the data
  2. They are based on weight sharing between the layers
- In this learning framework we use some numerical integrator $\Psi^{\Delta t}$ and obtain:
  - $\hat{y}_i^1 := \Psi^{\Delta t}(\hat{y}_i^0) = \Psi^{\Delta t}(x_i)$
  - $\hat{y}_i^2 := \Psi^{\Delta t}(\hat{y}_i^1)$
  - ...
  - $\hat{y}_i^M := \Psi^{\Delta t}(\hat{y}_i^{M-1})$
- The weights are shared since this time stepping refers always to $H_{A,\theta}$,
- The time updates to compute the $\hat{y}_i^j$ can be interpreted similarly to layer updates in RNNs.

Numerical experiment: Mathematical pendulum

$$H(q,p) = \frac{p^2}{2} - \cos(q), \quad (q,p) \in \mathbb{R}^2$$



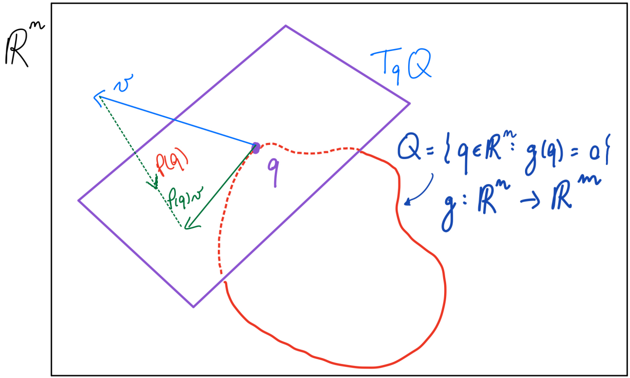


Learning some constrained Hamiltonian systems

Formulation of the dynamics ([5] Lee et al., 2017)

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0_n & P(q) \\ -P(q)^T & M(q,p) \end{bmatrix} \nabla H(q,p)$$

$$M(q,p) = P(q)^T \left(\frac{\partial P(q)^T p}{\partial q} \right)^T P(q) + \left(\frac{\partial P(q)^T p}{\partial q} \right) P(q) - P(q)^T \left(\frac{\partial P(q)^T p}{\partial q} \right)^T .$$



Hamiltonian systems on homogeneous manifolds

Homogeneous manifold

A smooth manifold M is homogeneous if there is a transitive Lie group action $\psi : G \times M \to M$.

- We restrict to the case $T^*Q = M$ is homogeneous.
- Here the Δt flow of $X \in \mathfrak{X}(M)$, reads

$$\Phi^{\Delta t}(x) = \psi(exp(\sigma_x(\Delta t)), x)$$

$$\dot{\sigma}_x = dexp_{\sigma_x}^{-1} \circ f \circ \psi(exp(\sigma_x), x),\ \sigma_x(0) = 0 \in \mathfrak{g},$$

for some $f : M \to \mathfrak{g} = T_e G$.

Hamiltonian systems on homogeneous manifolds

Homogeneous manifold

A smooth manifold M is homogeneous if there is a transitive Lie group action $\psi : G \times M \rightarrow M$.

- We restrict to the case $T^*Q = M$ is homogeneous.
- Here the Δt flow of $X \in \mathfrak{X}(M)$, reads

$$\Phi^{\Delta t}(x) = \psi(exp(\sigma_x(\Delta t)), x)$$

$$\dot{\sigma}_x = dexp_{\sigma_x}^{-1} \circ f \circ \psi(exp(\sigma_x), x), \ \sigma_x(0) = 0 \in \mathfrak{g},$$

  for some $f : M \rightarrow \mathfrak{g} = T_e G$.
- If $X = X_H$ is Hamiltonian, then f just depends on H and M, i.e. $f = F[H]$ for some F depending on M.

Learning the Hamiltonian of these systems

- **Neural network as the Hamiltonian**

$$Net(q,p) = H_{A,\theta}(q,p) \approx H(q,p),$$

- **Approximation of the Δt flow with some numerical method**
  For example with a Runge–Kutta–Munthe–Kaas method:

$$\hat{y}_i^{j+1} = \psi(exp(\bar{\sigma}), \hat{y}_i^j)$$

  where $\bar{\sigma}$ is the Δt solution of

$$\begin{cases} \dot{\sigma} = dexp_{\sigma}^{-1} \circ F[Net] \circ \psi(exp(\sigma), \hat{y}_i^j), \\ \sigma(0) = 0 \in \mathfrak{g} \end{cases}$$

  approximated with some Runge-Kutta method.

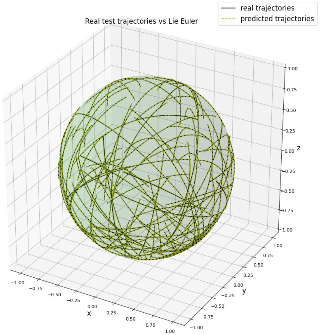
Case $M = T^*S^2$

- A transitive action on M [4] (Celledoni et al., 2021) is

$$\psi : SE(3) \times M \to M, \;\; \psi((R,r),(q,p)) = (Rq, Rp + r \times Rq),$$

- The Hamiltonian is a function $H : M \to \mathbb{R}$, and

$$f = F[H] = \begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix} \nabla H.$$



References

- Greydanus, S. J., Dzumba, M., Yosinski, J. (2019). Hamiltonian neural networks.
- Chen, Z., Zhang, J., Arjovsky, M., Bottou, L. (2019). Symplectic recurrent neural networks. International Conference on Learning Representations
- Jin, P., Zhang, Z., Zhu, A., Tang, Y., Karniadakis, G. E. (2020). SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems. Neural Networks, 132, 166-179.
- Celledoni, E., Çokaj, E., Leone, A., Murari, D., Owren, B. (2021) Lie Group integrators for mechanical systems. International Journal of Computer Mathematics.
- Lee, T., Leok, M., McClamroch, N. H. (2017). Global formulations of Lagrangian and Hamiltonian dynamics on manifolds. Springer, 13, 31.

Thanks for the attention