Time and State Dependent Neural Delay Differential Equations

Thibault Monsel, Onofrio Semeraro, Lionel Mathelin, Guillaume Charpiat

Time and State Dependent Neural Delay Differential Equations

T.Monsel, O.Semeraro, L.Mathelin, G.Charpiat
thibaultmonsel@inria.fr

ML-DE Workshop at ECAI 2024

Motivation

➡ Discontinuities and delayed terms are encountered in many domains (physics, engineering, medicine, economics, etc…).
➡ These systems cannot be modelled and simulated with ODEs.
➡ Latent variables are introduced to solve this issue which lack physical interpretability.
➡ Delay Differential Equations (DDEs) naturally appear as good candidates for such systems.

We revisit the recently proposed Neural DDE by introducing **Neural State-Dependent DDE (SDDDE)**, a general and flexible framework that can model multiple and state- and time-dependent delays.

Modelling choice

Neural ODE

$$\frac{d\mathbf{y}(t)}{dt} = \mathbf{f}_\theta(t, \mathbf{y}(t)), \quad \mathbf{y}(0) = \mathbf{y}_0.$$

ANODE

$$\frac{d}{dt}\begin{bmatrix}\mathbf{y}(t)\\ \mathbf{a}(t)\end{bmatrix} = \mathbf{f}_\theta\left(t, \begin{bmatrix}\mathbf{y}(t)\\ \mathbf{a}(t)\end{bmatrix}\right), \quad \begin{bmatrix}\mathbf{y}(0)\\ \mathbf{a}(0)\end{bmatrix} = \begin{bmatrix}\mathbf{y}_0\\ \mathbf{0}\end{bmatrix}.$$

Neural DDE

$$\frac{d\mathbf{y}(t)}{dt} = \mathbf{f}_\theta(t, \mathbf{y}(t), \mathbf{y}(t-\tau)), \quad \tau \in \mathbb{R}^+$$
$$\mathbf{y}(t<0) = \boldsymbol{\phi}(t),$$

Neural Laplace

$$\mathbf{p} = \mathbf{h}_\gamma((\mathbf{y}(t_1), t_1), \ldots, (\mathbf{y}(t_T), t_T))$$
$$\mathbf{F}(\mathbf{s}) = v\left(\mathbf{g}_\beta(\mathbf{p}, u(\mathbf{s}))\right)$$

Neural SDDDE

$$\frac{d\mathbf{y}(t)}{dt} = \mathbf{f}_\theta(t, \mathbf{y}(t), \mathbf{y}(t-\tau_1(t, \mathbf{y}(t))), \ldots, \mathbf{y}(t-\tau_k(t, \mathbf{y}(t))))$$
$$\mathbf{y}(t<0) = \phi(t)$$

Our method can handle :
➡ multiple delays
➡ constant, state and time dependent delays

It cannot handle delays defined through integrals as typically found in integro-differential equations.

Delay types	τ's Definition	Neural DDE	NPCDDEs	Neural Laplace	Neural SDDDE
References	-	[1]	[2]	[3]	This work
Constant	$\tau = a$	√	×	√	√
Piece-wise constant	$\tau = \lfloor \frac{t-a}{a} \rfloor a$	×	√	√	√
Time-dependent	$\tau = g(t)$	×	×	√	√
State-dependent	$\tau = g(t, y(t))$	×	×	×	√
Continuous	$\tau = \int_0^t g(s, y(s))\mathrm{d}s$	×	×	×	×

Experiments & Results

Time Dependent DDE

$$\tau(t) = 2 + \sin(t)$$
$$\frac{dy(t)}{dt} = y(t)\left[1 - y(t-\tau(t))\right],$$

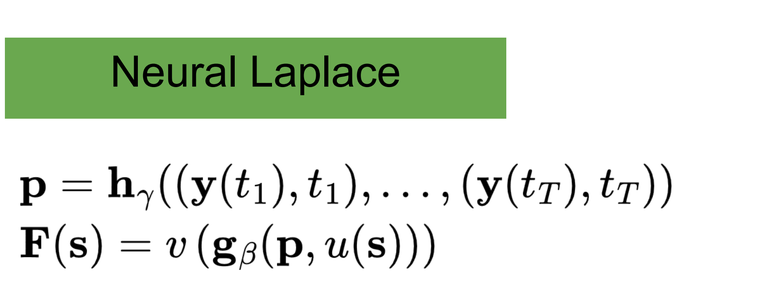
State Dependent DDE

$$\tau(y) = \tfrac{1}{2}\cos(y(t))$$
$$\frac{dy(t)}{dt} = -\alpha(t)y(t) + \beta(t)\frac{y^2(t-\tau(y))}{1+y^2(t-\tau(y))} + \gamma(t)$$

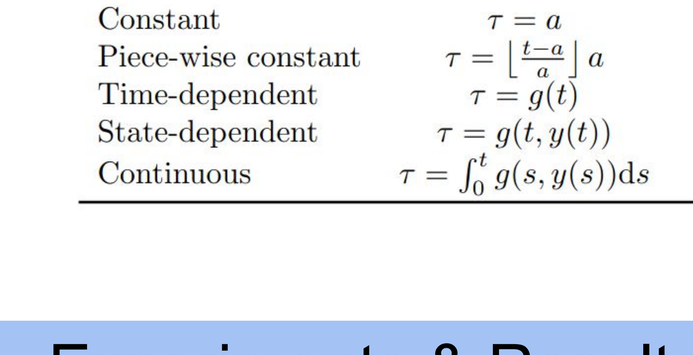
Delay Diffusion

$$\tau = 2$$
$$\frac{\partial u}{\partial t}(x,t) = D\frac{\partial^2 u}{\partial x^2}(x,t) + ru(x,t)\left(1 - u(x, t-\tau)\right),$$

Test dataset

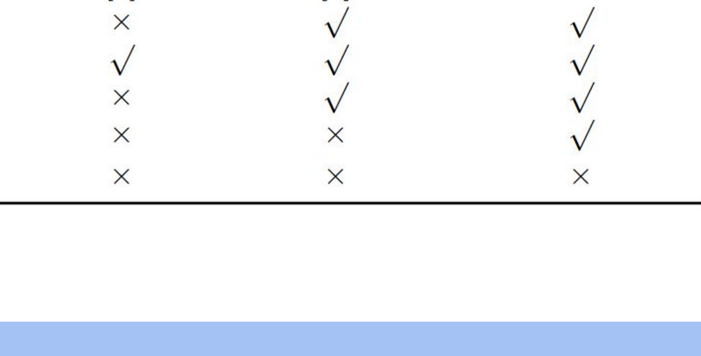


Changing history function dataset



State dependent sample with new history function

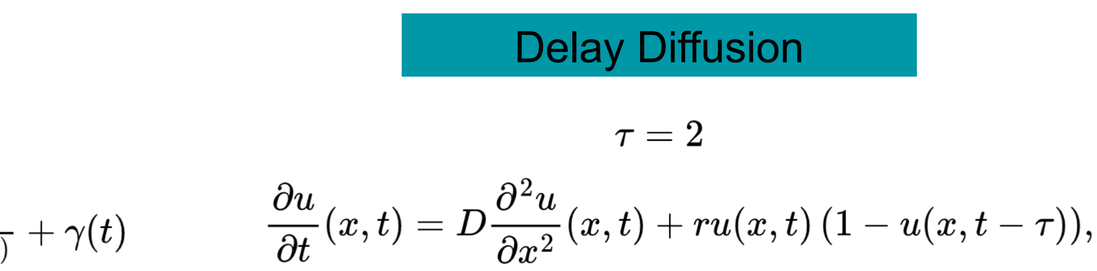
Out of distribution (OOD) dataset



Time dependent OOD sample

Conclusion

➡ More experiments and discussions are in the paper !
➡ Neural SDDDE almost consistently outperforms all other models.
➡ With the OOD and changing history function dataset, we show SDDDE's robustness.



Delay Diffusion OOD sample