Recurrent Neural Network Models for Predicting ODE Dynamics

Moncrief Summer Internship

Camille Renaud
Dr. Tan Bui-Thanh
Probabilistic and High Order Inference, Computation, Estimation, and Simulation
(Pho-Ices) Group
January 23, 2021

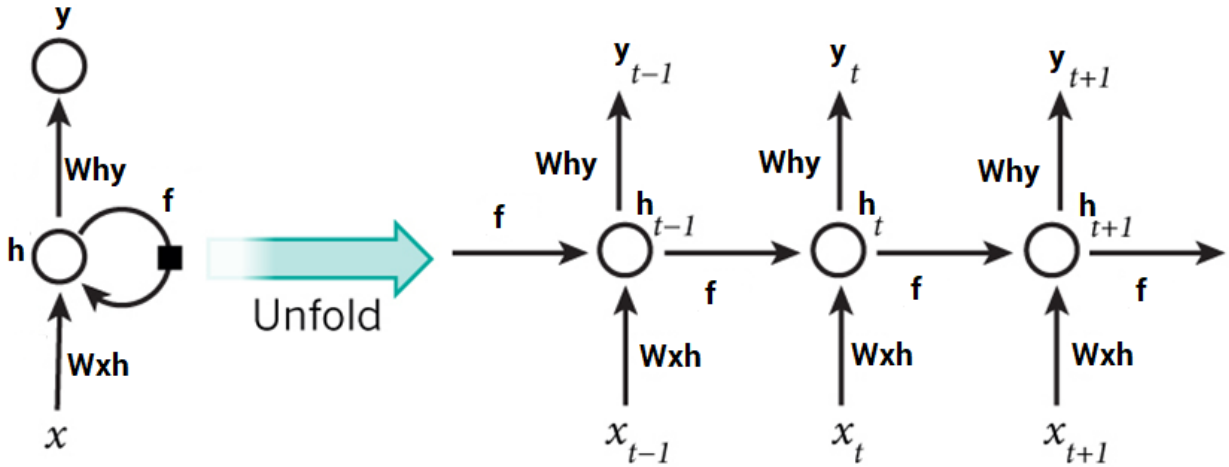
Outline

- Background on RNNs
- Applications and Examples
- Mathematical Motivation
- Research and Results
- Conclusions and Future Directions

Recurrent Neural Networks

- Recurrent Neural Network is a powerful neural network architecture that models sequential data
- Mathematical Structure of a Vanilla RNN:

$$h_t = \sigma(W_{xh} \cdot x_t + W_h \cdot h_{t-1} + b_h)$$
$$y_t = \sigma(W_{hy} \cdot h_t + b_y)$$



From *Fundamentals of Deep Learning – Introduction to Recurrent Neural Networks* [1]

BPTT and Vanishing/Exploding Gradients

- Loss function calculates difference between true and estimated values
- Backpropagation through time (BPTT) calculates the change in error with respect to the change in weights
- Weights are updated, and cycle repeats

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{t=1}^{T} \mathcal{L}(\hat{\mathbf{y}}_t, \mathbf{y}_t)$$

$$\frac{\partial \mathbf{E}}{\partial \mathbf{W}} = \sum_{t=1}^{T} \frac{\partial \mathbf{E}_t}{\partial \mathbf{W}}$$

$$\frac{\partial \mathbf{E}}{\partial \mathbf{W}} = \sum_{t=1}^{T} \frac{\partial \mathbf{E}}{\partial \mathbf{y}_t} \frac{\partial \mathbf{y}_t}{\partial \mathbf{h}_t} \overbrace{\frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_k}}^{\star} \frac{\partial \mathbf{h}_k}{\partial \mathbf{W}}$$

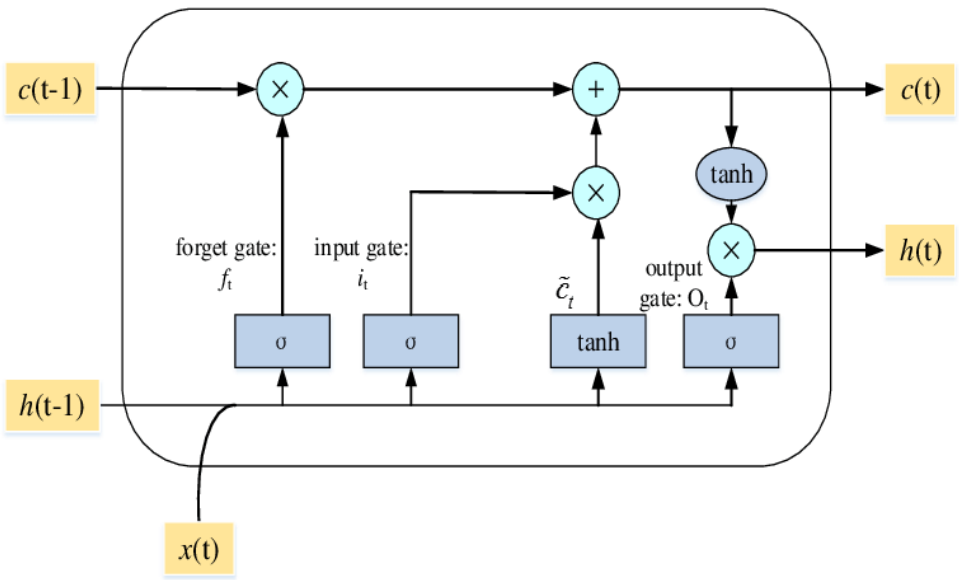
From *Fundamentals of Deep Learning – Introduction to Recurrent Neural Networks* [1]

Long Short-Term Memory RNNs

LSTM RNN is a class of RNNs that works with a cell state to create an information highway to be able to back propagate through time without the gradients vanishing.

$$i_t = \sigma\left(x_t \cdot W_x^i + h_{t-1} \cdot W_h^i\right)$$
$$f_t = \sigma\left(x_t \cdot W_x^f + h_{t-1} \cdot W_h^f\right)$$
$$o_t = \sigma\left(x_t \cdot W_x^o + h_{t-1} \cdot W_h^o\right)$$
$$\tilde{C}_t = \tanh\left(x_t \cdot W_x^g + h_{t-1} \cdot W_h^g\right)$$

$$C_t = \sigma\left(f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t\right)$$
$$h_t = \tanh\left(C_t\right) \cdot o_t$$



From *Nonlinear dynamic soft sensor modeling with supervised long short-term memory network* [2]

Shakespeare Writing

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

Image Captioning

"little girl is eating piece of cake."

"baseball player is throwing ball in game."

"woman is holding bunch of bananas."

"black cat is sitting on top of suitcase."

"a young boy is holding a baseball bat."

"a cat is sitting on a couch with a remote control."

"a woman holding a teddy bear in front of a mirror."

"a horse is standing in the middle of a road."

Algebraic Geometry Latex Textbook

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram

$$\begin{tikzcd}
\mathcal{S} \arrow[r] \arrow[d] & & & \\
\xi \arrow[r] & \mathcal{O}_{X'} \arrow[dr] & & \\
\mathrm{gor}_s \arrow[u] & & & \\
 & = \alpha' \arrow[r] \arrow[d, leftrightarrow] & & \\
 & = \alpha' \arrow[r] & \alpha & X \arrow[d] \\
 & \mathrm{Spec}(K_\psi) & & \mathrm{Mor}_{Sets} \quad \mathrm{d}(\mathcal{O}_{X_{X/k}}, \mathcal{G})
\end{tikzcd}$$

is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

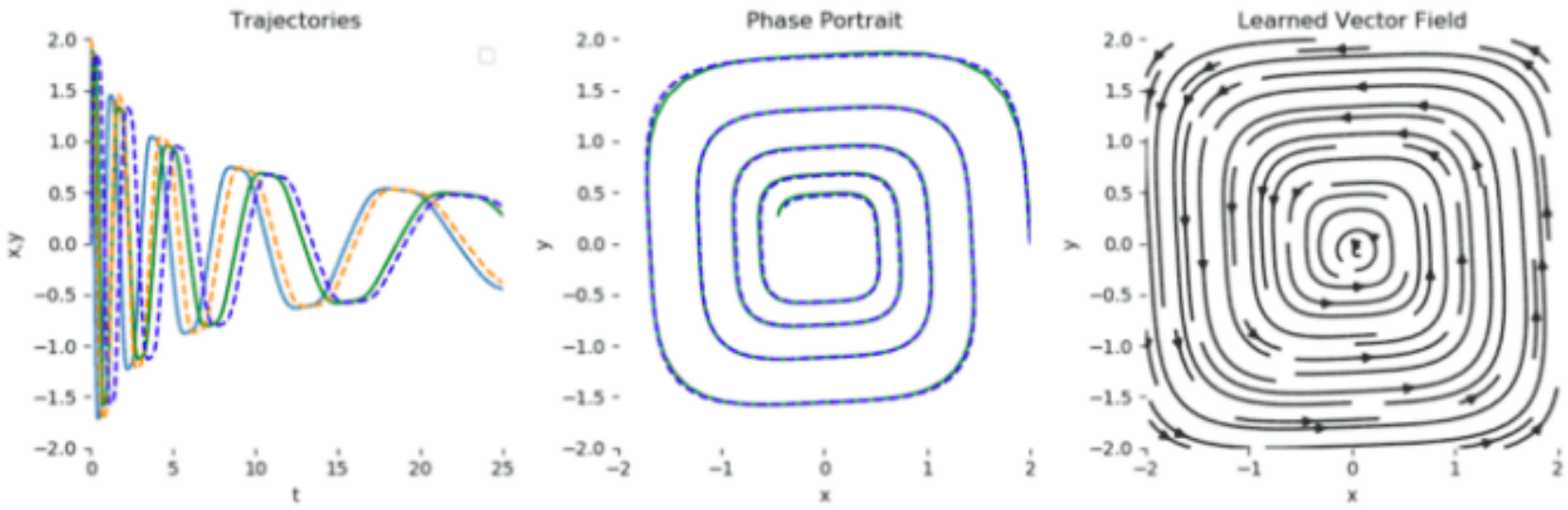
$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \quad \text{-1}(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_t}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{\eta}})$$

is an isomorphism of covering of $\mathcal{O}_{X_t}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

Connections to ODEs

Family of RNNs called the ODERNNs that are claimed to be a renaming of variables for general implicit runge kutta numerical method



Visualization of the Neural ODE learning the dynamical system

From *Neural Ordinary Differential Equation Based Recurrent Neural Network Model* [3]

BN-Stability analysis of ODERENNs

- Claims of BN-stability of ODERNNs through its mapping to discretization methods for ODEs.

General Implicit Runge-Kutta Numerical Method:

$$\vec{u}_n = \vec{u}_{n-1} + \delta \sum_{j=1}^{m} b_j f(t_{n-1} + c_j \delta, \vec{v}_j)$$

$$\vec{v}_i = \vec{u}_{n-1} + \delta \sum_{j=1}^{m} a_{ij} f(t_{n-1} + c_j \delta, \vec{v}_j) \qquad \text{for } n, i = 1 \ldots m$$

Where $\vec{v}_i$ denotes the intermediate approximations used in the computation of $\vec{u}_n$ from $\vec{u}_{n-1}$.

BN-Stability analysis of ODERENNs

BN-Stability

Implicit Runge-Kutta method is BN-stable if for two solution sequences $\cdots, \hat{u}_{n-1}, \hat{u}_n \cdots$ and $\cdots, u_{n-1}, u_n \cdots$ applied to any problem (1) where (2) holds, $||\hat{u}_n - u_n|| \leq ||\hat{u}_{n-1} - u_{n-1}||$.

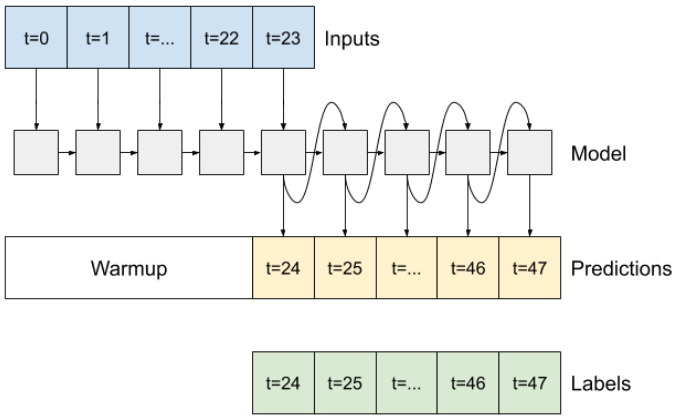
$$M = m_{ij} = b_i a_{ij} + b_j a_{ij} - b_i b_j$$

Claim: If the implicit Runge-Kutta method satisfies $b_1, b_2, \cdots b_m \geq 0$, and M is positive semi-definite, then the method is BN-stable.

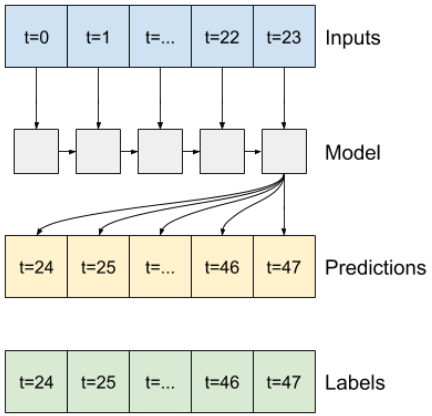
Our Approach

Consider two LSTM architectures and analyze their abilities in producing predictions of future solutions of systems of ODEs.

- **Model 1**: Recursive Multi-Step LSTM
- Predicts one time step using the prior time step as the input



- **Model 2**: One-Shot Multi-Step LSTM
- Predicts an entire range of future values in a one-shot manner



From *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems* [4]

ODE Systems of Study

1st Order System:

$$\frac{2ty}{t^2+1} - (2 - \ln(t^1+1))y' = 0 \qquad y(5) = 0$$

2nd Order Systems:

a.)

$$y'' + 2y' + 17y = -2\sin(3t) \qquad y(0) = 2, \quad y'(0) = 2$$

b.)

$$\frac{1}{2}y'' + 8y = 10\cos(\pi t) \qquad y(0) = 0, \quad y'(0) = 0$$

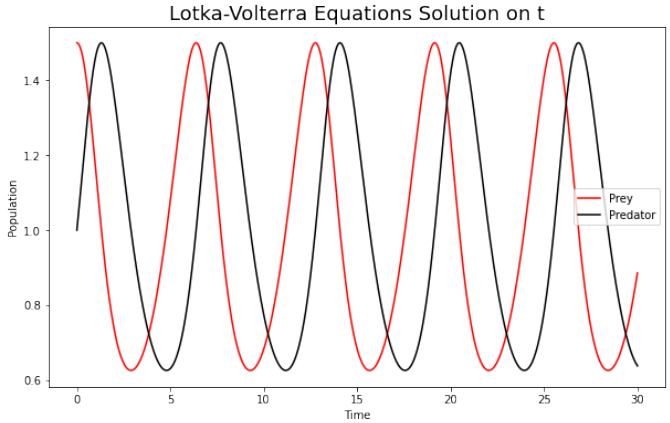
Coupled ODE Systems of Study

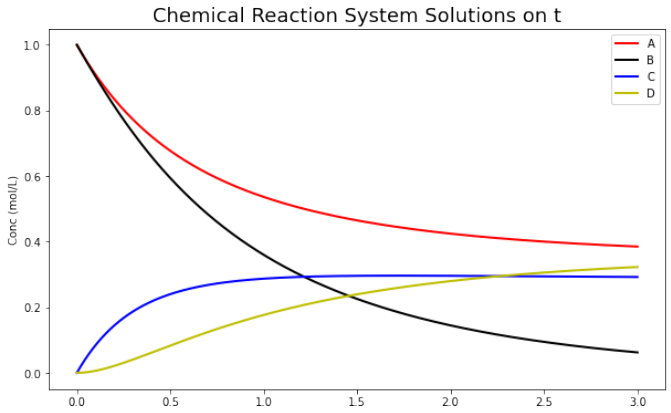
Lotka-Volterra Equations:

$$\frac{dx}{dt} = x(a - by) \qquad x(0) = 1.5$$
$$\frac{dy}{dt} = -y(c - dx) \qquad y(0) = 1$$

Chemical Reaction System

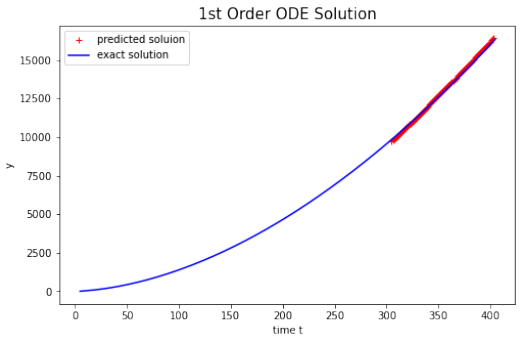
$$\frac{dA}{dt} = -k1AB \qquad A(0) = 1$$
$$\frac{dB}{dt} = -k1AB - k2BC \qquad B(0) = 1$$
$$\frac{dC}{dt} = k1AB - k2BC \qquad C(0) = 0$$
$$\frac{dD}{dt} = k2BC \qquad D(0) = 0$$

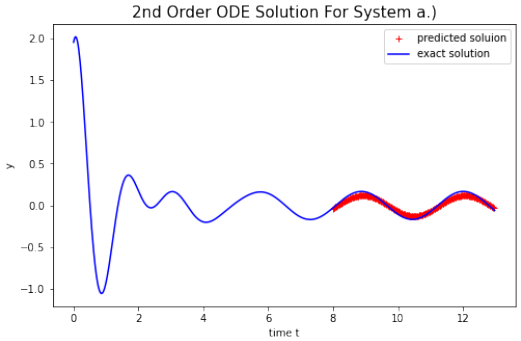


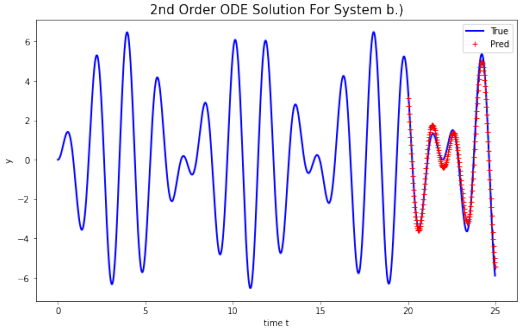


Model 1 Results

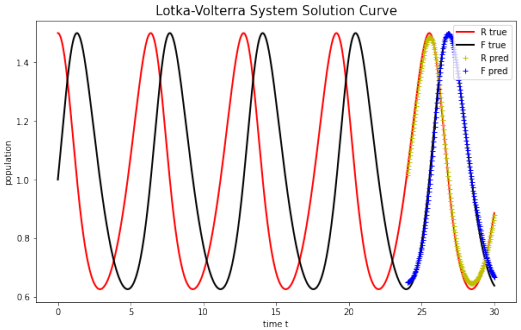
1st and 2nd Order ODEs

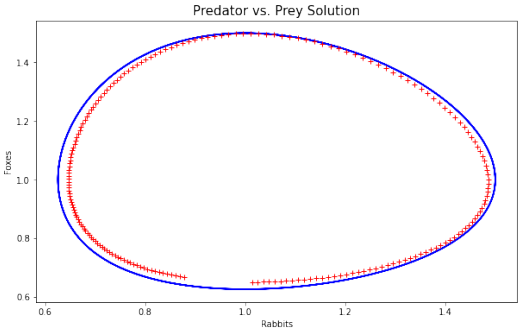


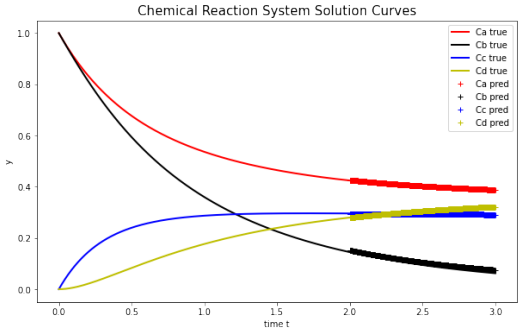




Coupled Systems of ODEs

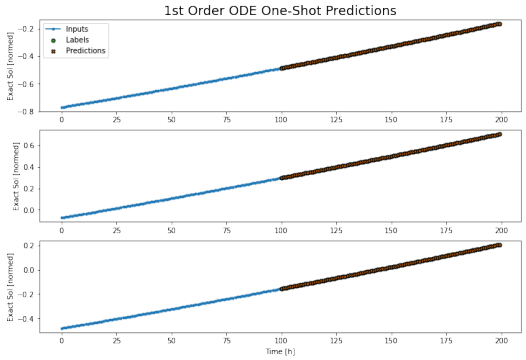


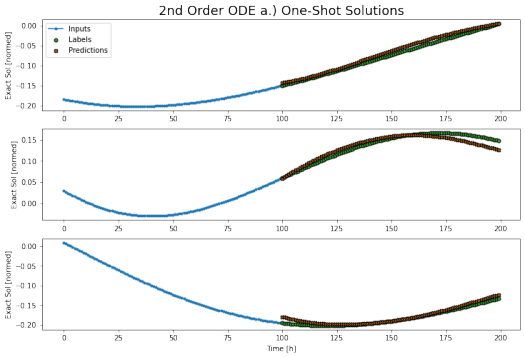


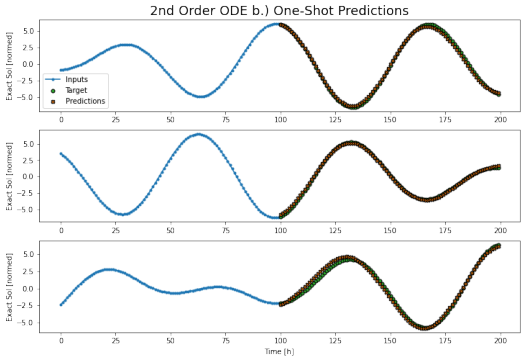


Model 2 Results

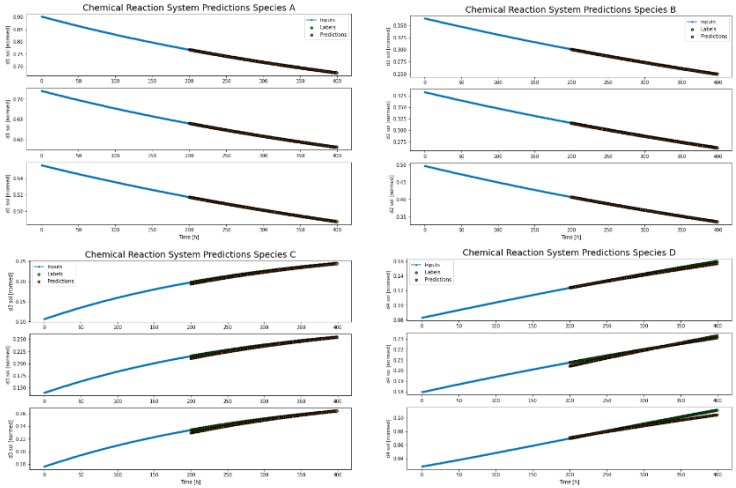
1st and 2nd Order ODEs

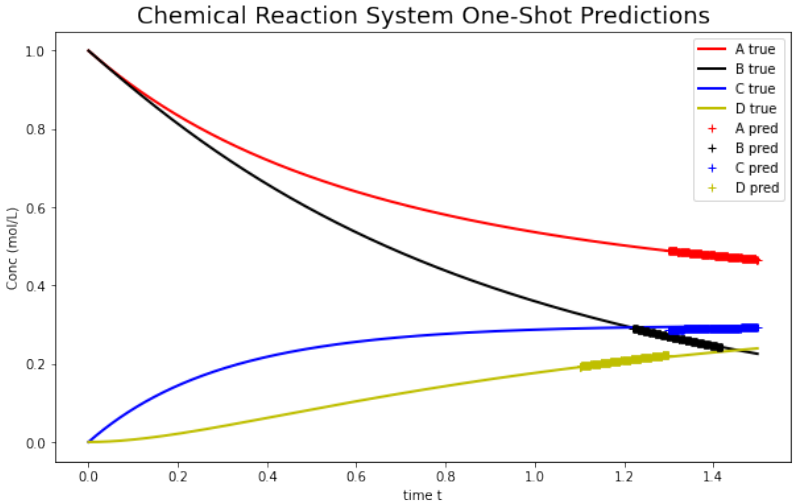






Coupled Systems of ODEs





Conclusions and Future Directions

- Adding a one-shot, multi-input LSTM model that can take vectors solutions
- Exploring real world applications and partial differential equations with the models
- Connecting the math theory of stability by creating a Runge-Kutta inspired RNN architecture to solve ODEs

The University of Texas at Austin

Oden Institute for Computational Engineering and Sciences

References

[1] Disheshree Gupta. *Fundamentals of Deep Learning – Introduction to Recurrent Neural Networks*. Analytics Vidhya, 2017. https://www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/

[2] Xiaofeng Yuan, Lin Li, and Yalin Wang. *Nonlinear dynamic soft sensor modeling with supervised long short-term memory network*. IEEE Transactions on Industrial Informatics PP(99):1-1, 2019.

[3] Mansure Habiba and Barak Pearlmutter. *Neural Ordinary Differential Equation Based Recurrent Neural Network Model*. Neural Information Processing Systems (NeurIPS), 2018. arXiv:2005.09807.

[4] TensorFlow Authors.Tensorflow timeseries forecasting. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*, 2015. https://www.tensorflow.org/tutorials/structured$_data/time_series$

[5] Muphy Yeuzhen Nui, Isaac Chuang, and Lior Horesh.*Recurrent Neural Networks in the Eye of Differential Equations*. arXiv:1904.12933, 2019.

[6] Thushan Ganegedara. *Stock Market Predictions with LSTM in Python*. DataCamp, 2020. https://www.datacamp.com/community/tutorials/lstm-python-stock-market

[7] Thomas Judson. *The Ordinary Differential Equations Project*. Department of Mathematics and Statistics, Stephan F. Austin State University, 2018.