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:

\[
\begin{align*}
h_t &= \sigma(W_{xh} \cdot x_t + W_h \cdot h_{t-1} + b_h) \\
y_t &= \sigma(W_{hy} \cdot h_t + b_y)
\end{align*}
\]

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

\[
\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T} \mathcal{L}(\hat{y}_t, y_t)
\]

\[
\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial W}
\]

\[
\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}
\]

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.

\[
i_t = \sigma \left( x_t \cdot W^i_x + h_{t-1} \cdot W^i_h \right)
\]
\[
f_t = \sigma \left( x_t \cdot W^f_x + h_{t-1} \cdot W^f_h \right)
\]
\[
o_t = \sigma \left( x_t \cdot W^o_x + h_{t-1} \cdot W^o_h \right)
\]
\[
\tilde{C}_t = \tanh \left( x_t \cdot W^g_x + h_{t-1} \cdot W^g_h \right)
\]
\[
C_t = \sigma \left( f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \right)
\]
\[
h_t = \tanh \left( C_t \right) \cdot o_t
\]

From Nonlinear dynamic soft sensor modeling with supervised long short-term memory network [2]
PANDARUS:
Alas, I think he shall be come approached and the day
When little sain 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."
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

\[
\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(L)
\]

Proof. This is an algebraic space with the composition of sheaves \( \mathcal{F} \) on \( X_{\text{etale}} \) we have

\[
\mathcal{O}_X(\mathcal{F}) = \{\text{morph} \times \mathcal{O}_X \ (\mathcal{G}, \mathcal{F})\}
\]

where \( \mathcal{G} \) defines an isomorphism \( \mathcal{F} \rightarrow \mathcal{F} \) 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 \( \mathcal{U} \subset X \) 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

\[
b : X \rightarrow Y \rightarrow Y \rightarrow Y' \times_X Y \rightarrow X.
\]

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

\[
\begin{array}{ccc}
X & \rightarrow & \mathcal{O}_X^{
abla} \\
\downarrow & & \downarrow \\
X & \rightarrow & \mathcal{O}_X
\end{array}
\]

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 \( X = \text{Spec}(R) \) and \( \mathcal{F} \) is a finite type representable by algebraic space. The property \( \mathcal{F} \) is a finite morphism of algebra 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 \( C \). The functor \( \mathcal{F} \) is a "field"

\[
\mathcal{O}_{\mathcal{X}_X} \rightarrow \mathcal{F}_X \rightarrow \mathcal{F}_{\mathcal{X}_X} \rightarrow \mathcal{O}_{\mathcal{X}_X}(\mathcal{O}_{\mathcal{X}_X})
\]

is an isomorphism of covering of \( \mathcal{O}_{\mathcal{X}_X} \). 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 \( \mathcal{O}_{\mathcal{X}_X} \) 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:**

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

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

Where \( \tilde{v}_i \) denotes the intermediate approximations used in the computation of \( \tilde{u}_n \) from \( \tilde{u}_{n-1} \).
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:

\[
\frac{2ty}{t^2 + 1} - (2 - \ln(t^1 + 1))y' = 0 \quad y(5) = 0
\]

2nd Order Systems:

a.)

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

b.)

\[
\frac{1}{2}y'' + 8y = 10\cos(\pi t) \quad y(0) = 0, \quad y'(0) = 0
\]
Coupled ODE Systems of Study

Lotka-Volterra Equations:
\[
\begin{align*}
\frac{dx}{dt} &= x(a - by) \quad x(0) = 1.5 \\
\frac{dy}{dt} &= -y(c - dx) \quad y(0) = 1
\end{align*}
\]

Chemical Reaction System
\[
\begin{align*}
\frac{dA}{dt} &= -k_1AB \quad A(0) = 1 \\
\frac{dB}{dt} &= -k_1AB - k_2BC \quad B(0) = 1 \\
\frac{dC}{dt} &= k_1AB - k_2BC \quad C(0) = 0 \\
\frac{dD}{dt} &= k_2BC \quad D(0) = 0
\end{align*}
\]
Model 1 Results

1st and 2nd Order ODEs

Coupled Systems of ODEs
Model 2 Results

1st and 2nd Order ODEs

Coupled Systems of ODEs

Chemical Reaction System One-Shot Predictions
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
References


