An analysis of the LSTM's ability to learn and reproduce Chaotic Systems.

Sheil Kumar
University of Illinois Urbana-Champaign
sk17@illinois.edu

***Abstract*— a-**

I. Introduction

The constantly evolving and diversely applicable field of machine learning has grown exponentially as both the magnitude and accessibility of computing power have improved in recent times. Accompanying this growth, great strides have been made in testing the limits of the various machine learning frameworks. The frameworks have also been used to try and understand chaotic systems, while previous methodologies would suggest a knowledge based approach to model the equations governing these systems, machine learning methods aim to predict future states of the system by relying just on past time series data, providing a model-free method of accurately predicting these systems.

The Long-Short Term Memory (universally abbreviated LSTM) cell is a hidden unit belonging to the family of artificial Recurrent Neural Networks (RNNs) which was designed to be capable of remembering long term dependencies[1] which previous RNNs found difficult to address due to the vanishing or exploding gradients problem that occurs while back-propagating in time during the training of said RNNs. Networks utilizing LSTMs have already demonstrated their capacity to solve problems with long term dependencies as they have been used in analysis of audio[2] and video[3] data, speech recognition[4], and predictions of traffic flow[5].

There have been many attempts at simulating and reproducing the dynamics of the 1963 Lorenz system[6]. Using machine learning schemes to reproduce the Lorenz system is not a novel task in the field, as various machine learning frameworks have been used to tackle this problem, attempts using Reservoir Computing schemes[7–10] have been thoroughly explored with varying degrees of success, especially in the short term. Even the use of LSTMs in modelling the Lorenz has been explored to some extent, [11] explores the effects of precision of the training data in predicting future time steps of the Lorenz system compared to the true values. In this paper we explore the effect of the LSTMs hyper-parameters in predicting the dynamics of the system, we demonstrate each designed networks ability to predict the system by evaluating the errors between the predicted and true values of the attractor, as well as how well the dynamics are modelled by comparing the maximal Lyapunov exponents of a predicted system to that of the literature value.

II. Long Short Term Memory (LSTM)

The LSTM belongs to the family of RNN cells, it was designed in order to address the vanishing/exploding gradients problem that plagues traditional RNN cells. The structure of a regular RNN layer and the schematics of an LSTM cell can be seen in 1 and it's governing equations are provided in I.

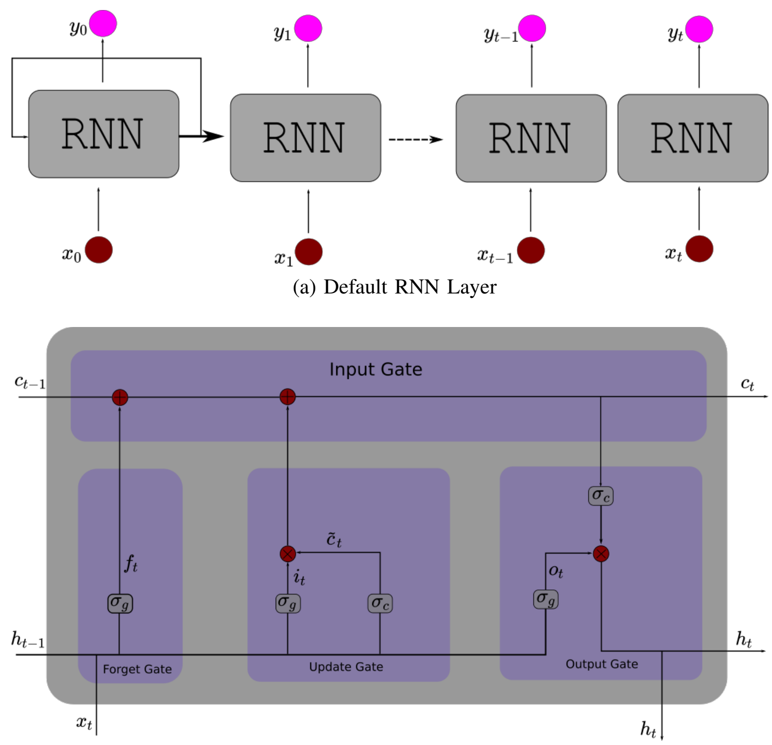
TABLE I: LSTM Equations

Forward Pass Equations
$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$
$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$
$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$
$\widetilde{c_t} = \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$
$c_t = f_t \circ c_{t-1} + i_t \circ \widetilde{c_t}$
$h_t = \sigma_t \circ \sigma_h(c_t)$
Notation
$x_t \in \mathbb{R}^d$: input vector to LSTM cell
$f_t \in \mathbb{R}^h$: forget gate function
$i_t \in \mathbb{R}^h$: input gate function
$o_t \in \mathbb{R}^h$: output gate function
$h_t \in \mathbb{R}^h$: hidden state of LSTM cell
$\widetilde{c_t} \in \mathbb{R}^h$: cell input function
$c_t \in \mathbb{R}^h$: cell state
$W \in \mathbb{R}^{h\times d}$: weight matrix for input vector $*^0$
$U \in \mathbb{R}^{h\times h}$: weight matrix for hidden state vector $*^0$
$b \in \mathbb{R}^h$: bias vectors
Activation Functions
σ_g : sigmoid function
σ_c : hyperbolic tangent function

* the values d and h refer to the number of input features (in our case 3 (x,y,z) components of the time step) and the number of hidden units respectively.

A. Network Structure

In this paper, we evaluated the effectiveness of a variety of hyperparameters while keeping the founding architecture of the network the same throughout. We evaluate networks with a number of LSTM cells ranging from $[16-256]$ units.



(a) Default RNN Layer

(b) LSTM Cell Schematic

Fig. 1: Schematic of a LSTM Cell and the structure of a normal RNN layer.

III. Data

A. Data Generation

We generate data from the Lorenz attractor defined by the equations:

$$\begin{aligned} \dot{x} &= \sigma(y-x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z \end{aligned} \tag{1}$$

Where $\sigma, \rho,$ and β are constants.To generate the data, we make use of the Forward Euler Integration Scheme which works recursively by solving

$$y'(t) = f(t, y(t)) \tag{2}$$

$$\begin{aligned} y(t_0) &= y_0 \\ t_n &= t_0 + nh \end{aligned}$$

$$\therefore y_{n+1} = y_n + hf(t_n, y_n) \tag{3}$$

Where h is the **step size** and t_i is the i^{th} **time step**. Identical equations exist for both x and z as well. We begin with the initial value $(x_0, y_0, z_0) = (0,0,0)$ and solve recursively as defined above to obtain $\mathbf{X} = X_0, X_1, ..., X_n$, where $X_i = (x_i, y_i, z_i)$ for $n = 16000$.

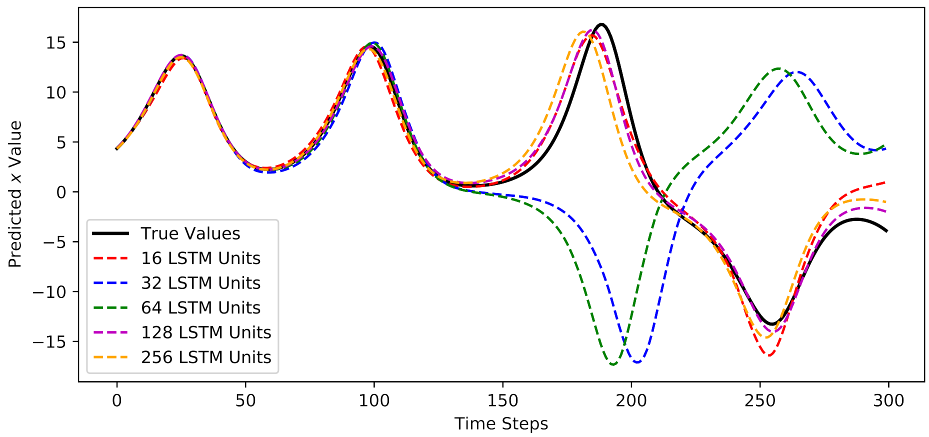
B. Training and Testing

Let us define the target vector, $\mathbf{Y} = Y_0, Y_1, ..Y_{n-1}$, where $Y_i = (x_{i+1} - x_i, y_{i+1} - y_i, z_{i+1} - z_i) = X_{i+1} - X_i$. The target vector $\mathbf{Y}$ is the vector we will **train the model to predict**. As such the model will be fed the true $\mathbf{X}$ and true $\mathbf{Y}$ and be asked to predict a $\hat{Y}_i$ for each time step. That is, each X_i will be mapped to a $\hat{Y}_i$ as predicted by the model; $X_i \longmapsto \hat{Y}_i$. The $\hat{Y}_i$ determined by the model are predicted based on a set of weights $\mathbf{W}, \mathbf{U}$ and a set of bias terms $\mathbf{B}$ which are determined by the model while training to minimize the loss function, $\mathbf{L} = (\mathbf{Y} - \hat{\mathbf{Y}})^2$, this loss function is simply the **mean squared error**, where $\hat{\mathbf{Y}} = \hat{Y}_0, \hat{Y}_1, ..., \hat{Y}_{n-1}$. The specific method through which the weights, $\mathbf{W}, \mathbf{U}$ and bias terms, $\mathbf{B}$ are determined is not imperative to understand, and these will change depending on the size and architecture of the model, however **the model itself** will be discussed to some extent in the following section.

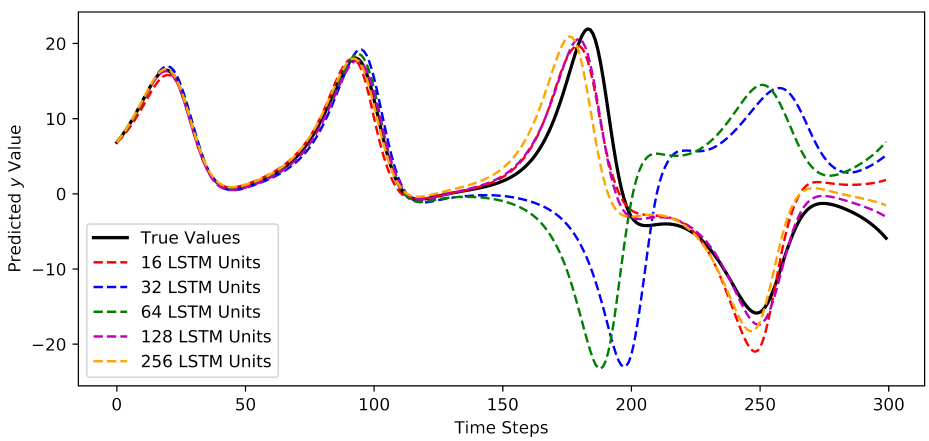
IV. Model

A. Network Structure

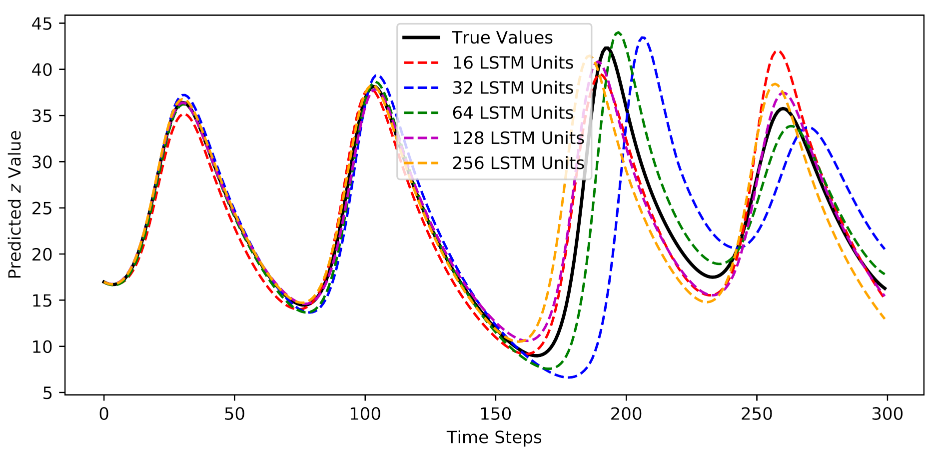
We have analyzed the Lorenz attractor through many different neural networks. However, the structure of the neural networks remains the same throughout and they all recieved the same data as described in **2.1**. Each neural network is first composed of an LSTM layer composed of a certain number of LSTM units, this is then fed into a dense layer composed of three units, respectively the (x, y, z) components of the attractor, that is then returned to the user by the network. We evaluated networks LSTM units ranging between 8 and 256 units, while the number of epochs were also varied.



(a) X-Coordinate Predictions



(b) Y-Coordinate Predictions



(c) Z-Coordinate Predictions

Fig. 2: Predictions of the various networks over 300 time steps.

V. Results

Each network was evaluated according to how well it was able to reproduce the literature Lyapunov exponent for it's given attractor. The networks were trained with data pertaining to a Lorenz attractor with $\beta = \mathbf{8/2}, \rho = \mathbf{28}, \sigma = \mathbf{10}$, the literature value of the Lyapunov exponent corresponding to these parameters is **0.9056**.

References

[1]K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: a search space odyssey", IEEE

Transactions on Neural Networks and Learning Systems **28**, 2222–2232 (2017).
[2]E. Marchi, G. Ferroni, F. Eyben, L. Gabrielli, S. Squartini, and B. Schuller, “Multi-resolution linear prediction based features for audio onset detection with bidirectional lstm neural networks”, in 2014 ieee international conference on acoustics, speech and signal processing (icassp) (IEEE, 2014), pp. 2164–2168.
[3]J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description”, in Proceedings of the ieee conference on computer vision and pattern recognition (2015), pp. 2625–2634.
[4]A. Graves, N. Jaitly, and A.-r. Mohamed, “Hybrid speech recognition with deep bidirectional lstm”, in 2013 ieee workshop on automatic speech recognition and understanding (IEEE, 2013), pp. 273–278.
[5]Y. Tian and L. Pan, “Predicting short-term traffic flow by long short-term memory recurrent neural network”, in 2015 ieee international conference on smart city/socialcom/sustaincom (smartcity) (IEEE, 2015), pp. 153–158.
[6]E. N. Lorenz, “Deterministic nonperiodic flow”, Journal of atmospheric sciences **20**, 130–141 (1963).
[7]J. Pathak, A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott, “Hybrid forecasting of chaotic processes: using machine learning in conjunction with a knowledge-based model”, Chaos: An Interdisciplinary Journal of Nonlinear Science **28**, 041101 (2018).
[8]J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, “Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data”, Chaos: An Interdisciplinary Journal of Nonlinear Science **27**, 121102 (2017).
[9]Z. Lu, B. R. Hunt, and E. Ott, “Attractor reconstruction by machine learning”, Chaos: An Interdisciplinary Journal of Nonlinear Science **28**, 061104 (2018).
[10]D. Canaday, A. Pomerance, and D. J. Gauthier, “Model-free control of dynamical systems with deep reservoir computing”, arXiv preprint arXiv:2010.02285 (2020).
[11]S. Bompas, B. Georgeot, and D. Guéry-Odelin, “Accuracy of neural networks for the simulation of chaotic dynamics: precision of training data vs precision of the algorithm”, Chaos: An Interdisciplinary Journal of Nonlinear Science **30**, 113118 (2020).