Chaos Modeling

Comparison of Classical and Quantum Reservoir Computers

TJHSST Quantum Research Lab

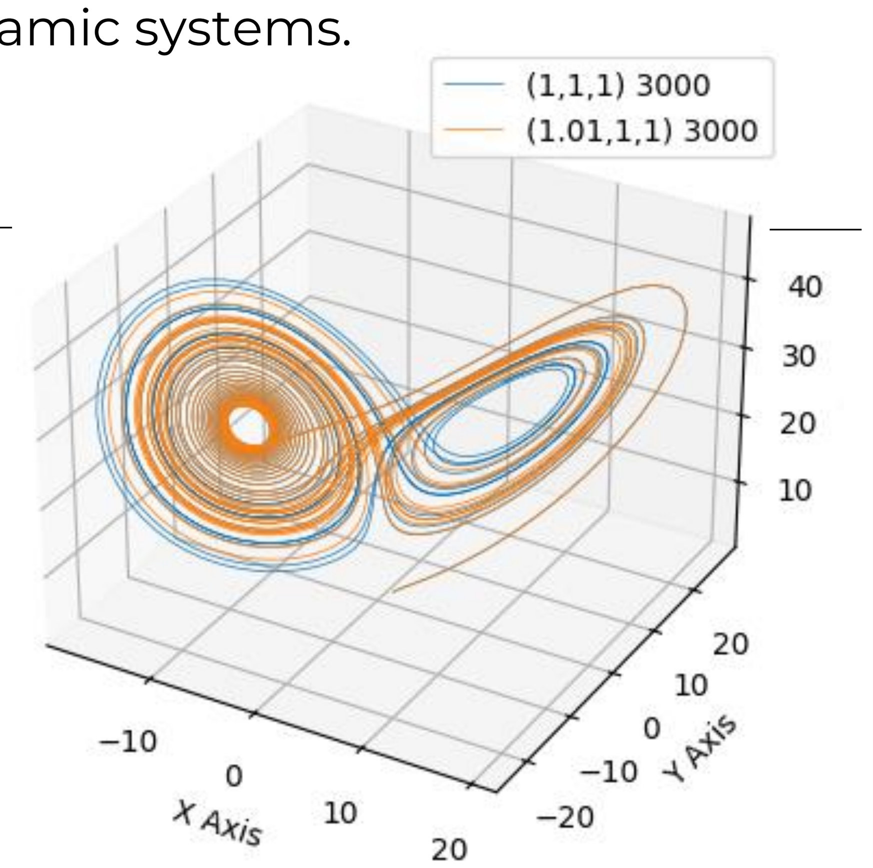
Kanjonavo Sabud
Arjun Bhat

Introduction

Systems in which a small difference in the initial state grow exponentially over time are said to be chaotic or dynamic. From **stock market prediction to weather forecasting**, many different models are being explored for modeling such dynamic systems.

Dataset

$$\frac{dx}{dt} = \sigma \cdot (y - x)$$
$$\frac{dy}{dt} = x \cdot (\rho - z) - y$$
$$\frac{dz}{dt} = x \cdot y - \beta \cdot z ,$$



The **Lorenz 1963** dataset was created using the above differential equations with **σ=10, ρ=28, β=8/3 and initial position being (1,1,1)**. The graph to the right demonstrates the dynamic nature of this system where only a 0.01 difference in the x initial position results in completely separate paths.

ANN

Control Model

Structure: **3x100x100x100x3.** We used the **Relu** activation function for all layers except for the last one for which we used **SoftMax**.

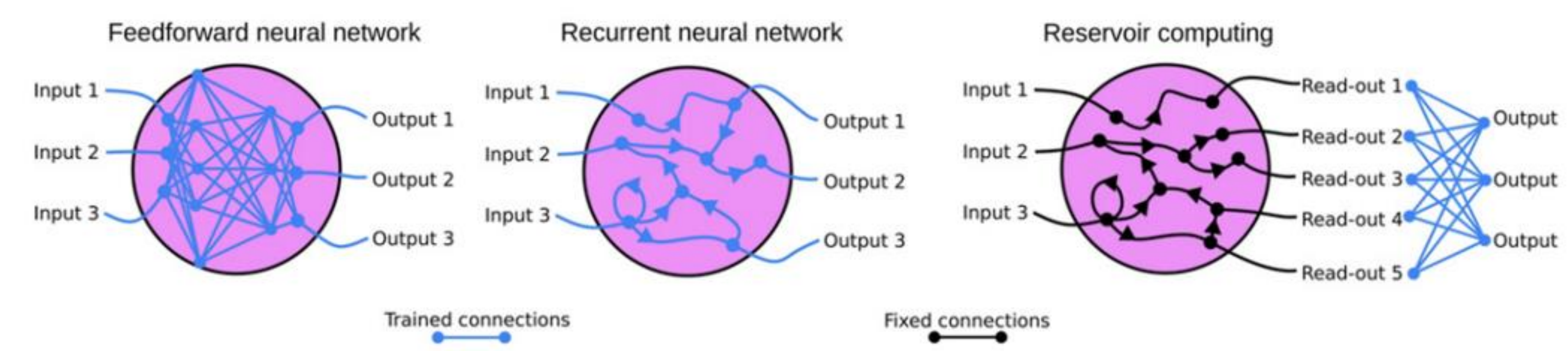
Inputs were the raw positions, and the outputs were the change in positions during training.

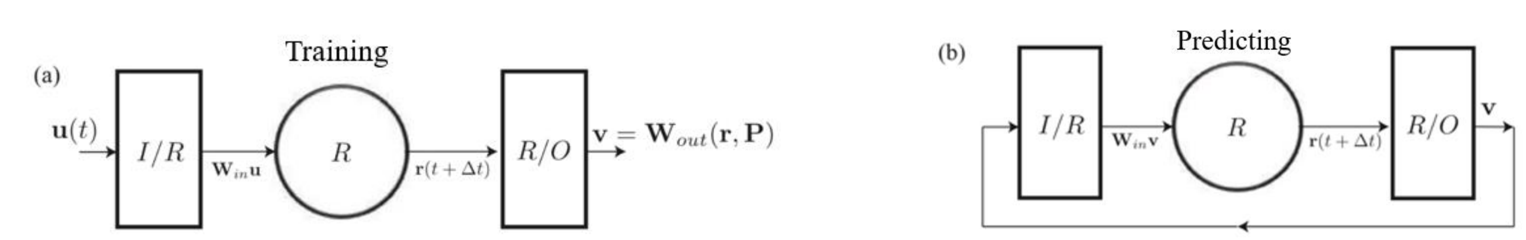
The **Adams Bashford** scheme (below) was used to calculate the input for the next time stamp during prediction.

$$X(t+\Delta t)^{\text{test}} = X(t)^{\text{test}} + \frac{1}{2}\left[3\Delta X(t)^{\text{test}} - \Delta X(t-\Delta t)^{\text{test}}\right].$$

Reservoir Computer

The Reservoir Computer (RC) is a subset of a class of ML Algorithms called RNNs. It has many similarities such as having memory of previous inputs and randomly connected nodes. However, the **weights connecting the nodes of the reservoir cannot be trained**. Only the weights connecting the readout layer and the output layer are trained using linear regression. This drastically reduces time and computational complexity and results in better performance.

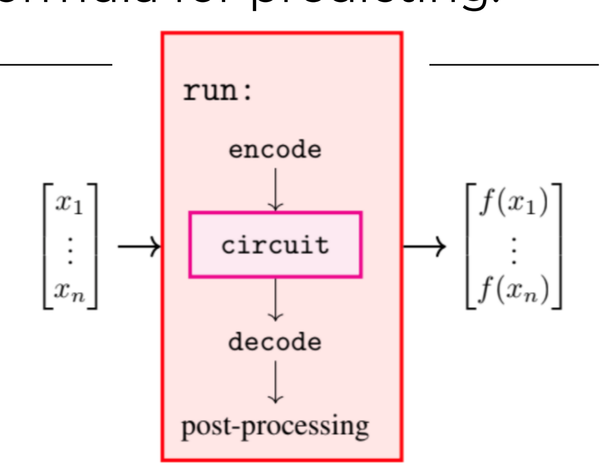


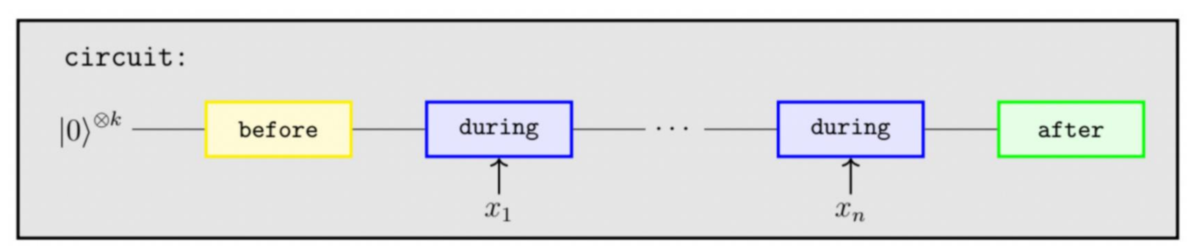


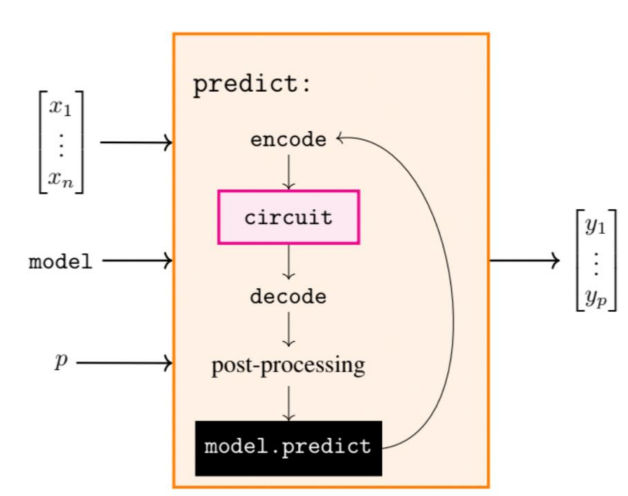
1. $\mathbf{r}(t+\Delta t) = \tanh[\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{in}\mathbf{u}(t)]$
2. $W^{out} = Y^{target}X^{T}\left(XX^{T} + \beta I\right)^{-1}$
3. $\mathbf{r}(t+\Delta t) = \tanh[\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{in}\mathbf{W}_{out}(\mathbf{r}(t), \mathbf{P})]$

Eq 1, 2 show the reservoir update and final weight out formulas for training.
Eq 3 shows the looped reservoir update formula for predicting.

Quantum Reservoir Computer



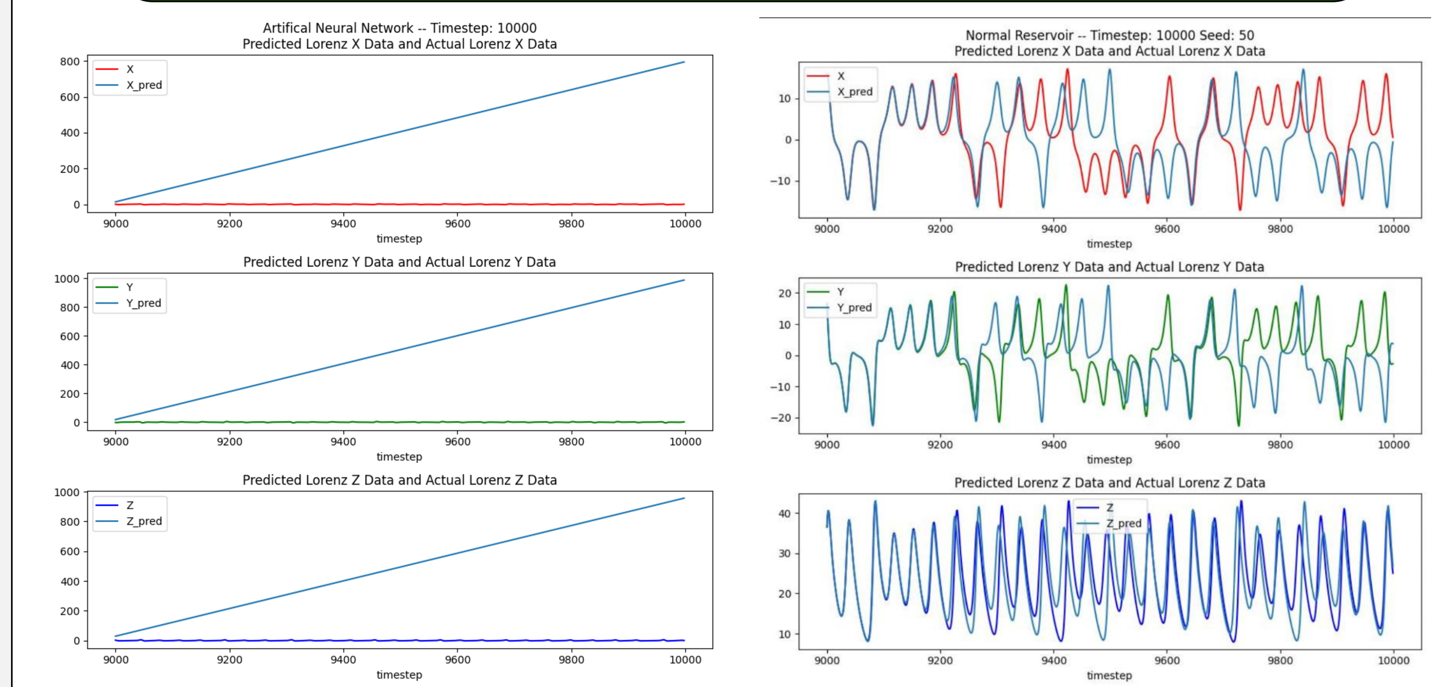


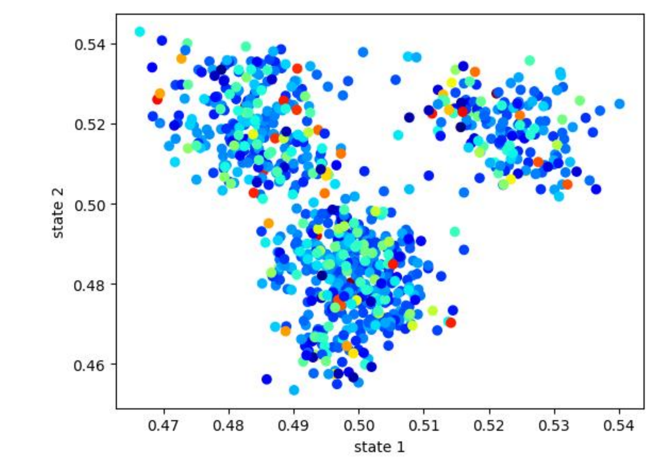


Differences between Quantum and Classical:

1. **Encoder-decoder coupling** to transform from classical to quantum and vice verse
2. **An ML model** (analogous to the single layer MLP in classical) is trained using the output of the run (train) method and then passed onto the predict (test) method.

Results





The transformed feature vector (output of QRC run method) spread is shown to the left. Our NN model obtained a 0% accuracy. **Our QRC was unable to model the Lorenz system.**

Discussion

As shown by the graphs of ANN and RC in the results, the **RC performed much better than ANN**. This was expected and conformed to the results of previous research.
Research on QRC and its dynamic systems modeling capabilities is still in its infancy. Therefore, it is highly likely that our implementation of QRC was not prepared to model a system as complex as Lorenz 63.

References

Chattopadhyay, A., Hassanzadeh, P., & Subramanian, D. (2020). Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Processes in Geophysics*, *27*(3), 373–389. https://doi.org/10.5194/npg-27-373-2020

Cucchi, M., Abreu, S., Ciccone, G., Brunner, D., & Kleemann, H. (2022). Hands-on reservoir computing: a tutorial for practical implementation. *Neuromorphic Computing and Engineering*, *2*(3), 032002. https://doi.org/10.1088/2634-4386/ac7db7

Ma, H., Prosperino, D., & Räth, C. (2023). A novel approach to minimal reservoir computing. *Scientific Reports*, *13*(1). https://doi.org/10.1038/s41598-023-39886-w

Morales, G. B., Mirasso, C. R., & Soriano, M. C. (2021). Unveiling the role of plasticity rules in reservoir computing. *Neurocomputing, 461*, 705–715. https://doi.org/10.1016/j.neucom.2020.05.127

Pathak, J., Lu, Z., Hunt, B. R., Girvan, M., & Ott, E. (2017). Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *27*(12). https://doi.org/10.1063/1.5010300

Acknowledgements

Thank you to Mr. Hannum, Dr. Chattopadhyay, Hriday Sainathuni, Aarya Vijayaraghavan, and the reservoir.py and quantumreservoir.py teams.