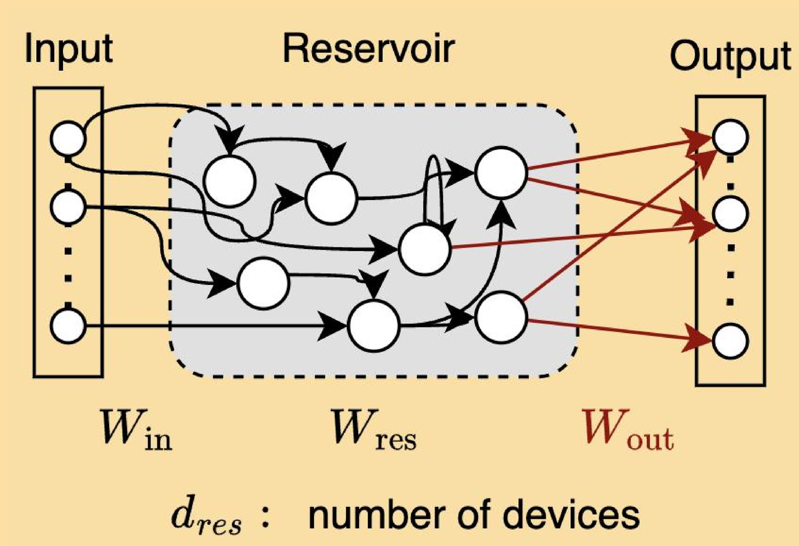
Towards Design Guidance of Reservoir Computing — An Empirical Study

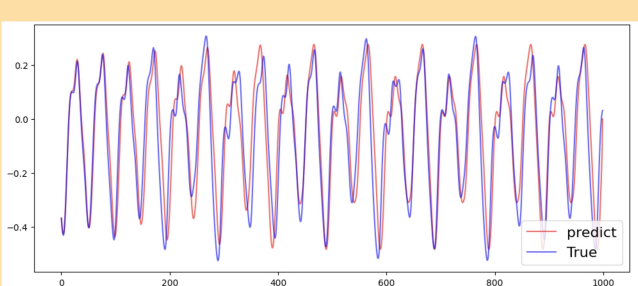
Christina Lee[1,3], Xiangxiang Xu[2], Prof. Lizhong Zheng[2]
[1] ESBNU, Xicheng, Beijing 100032, China,
[2] Dept. EECS, MIT, Cambridge, MA 02139, USA, [3] Boston University, Boston, MA 02215,USA

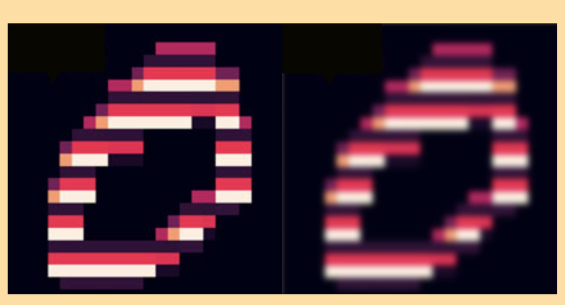
Backgroud

- **Reservoir Computing (RC)** has emerged as a versatile machine learning technique.
  - Recurrent Neural Network (RNN)
  - Reservoir Setup: input layer, reservoir, and output layer.
  - Reservoir Dynamics: The reservoir neurons create complex temporal representations of the input.
  - Training: Set the reservoir's weights, only the weights connecting the reservoir to the output layer are trained.
  - Testing: Task-specific learning



- Finding applications in a wide range of domains,
  - e.g., Time Series Prediction, Pattern Recognition and Image Processing, Wireless Network, Brain-Computer Interfaces (BCI)...





Motivation

Nowadays algorithm designers DON'T care about .... Design

While Reservoir Computing has gained widespread popularity for its versatility in diverse applications, there remains a lack of

- Limited understanding of the algorithm's setting
- The intricate interplay between various model components.
- Comprehensive investigations into the impact of key design choices on model performance.
- Design Factors for more conprehensive analysis, including **activation functions**, **initialization techniques**, **parameter settings**, and **data types**. Our aim is to provide valuable guidance to algorithm designers who are considering the adoption of RC in their applications.

Experiment

To facilitate our study, we create a robust testing environment capable of evaluating RC performance under diverse parameter configurations. Specifically, data is generated from a compact **Echo State Network (ESN)** model to provide an in-depth examination of how different activation functions affect the reservoir's capabilities.

- ESN_0: small scale linear reservoir model; data generator
- x_in: random input signal
- y_out: target signal
- Study how error change with different parameters

Activation Function   Randomly Given

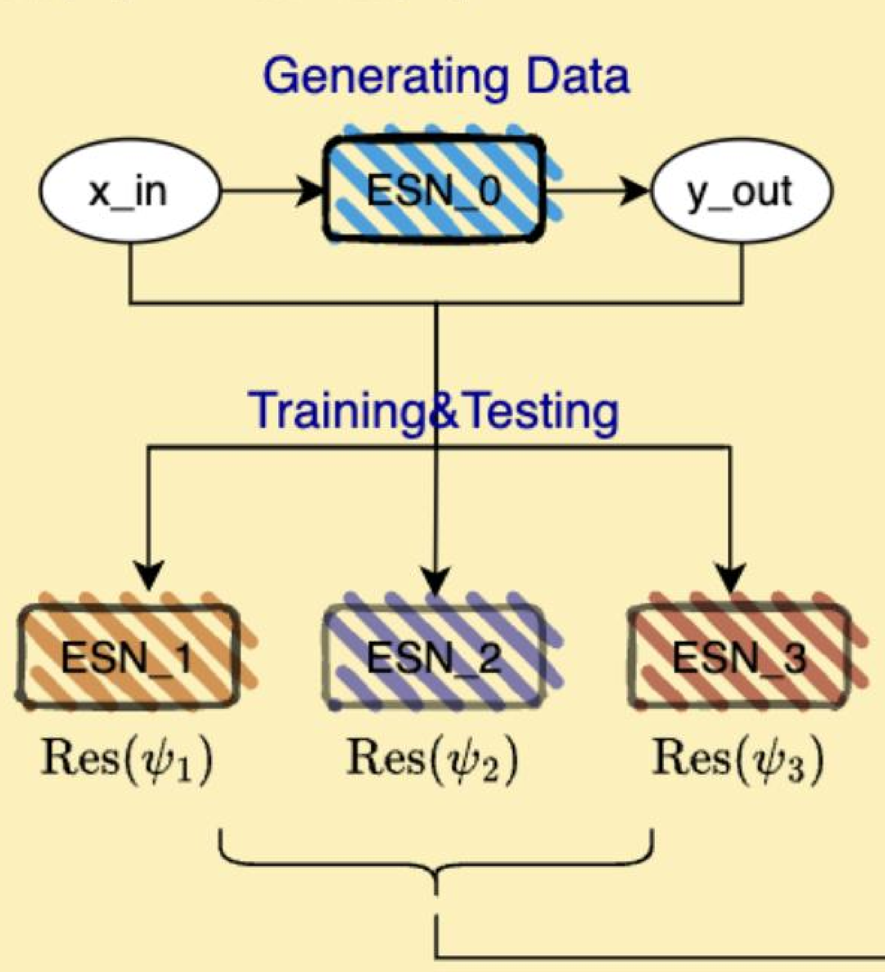
$$X_{\text{res}}[m] = \psi(W_{\text{res}} X_{\text{res}[m-1]} + W_{\text{in}} X_{\text{in}}[m-1] + b_{\text{in}})$$

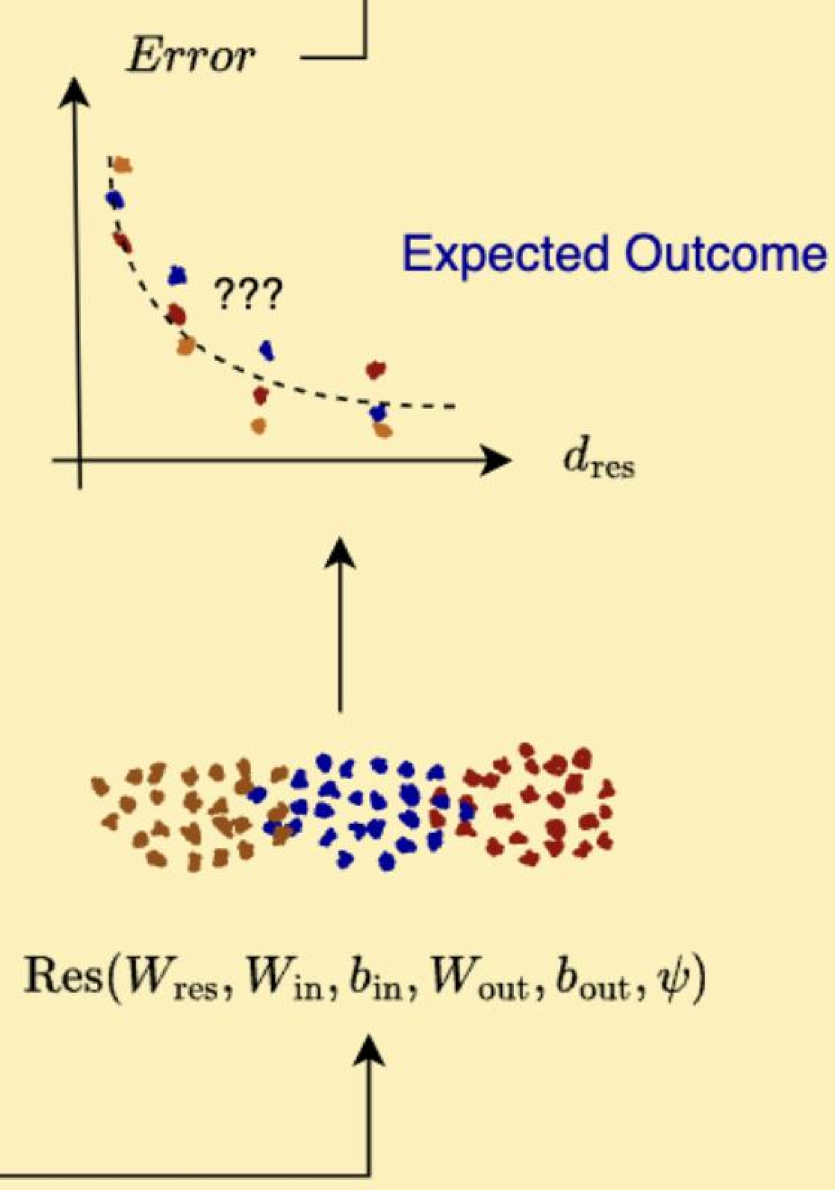
$$\psi_1 : x \mapsto x$$
$$\psi_2 : x \mapsto tanh(x)$$
$$\psi_3 : x \mapsto relu(x)$$
$$relu(x) = max(x, 0)$$

$$X_{\text{out}}[m] = W_{\text{out}} X_{\text{res}}[m]$$

$$\min \frac{1}{N} \sum_{l=1}^{N} \|Y^{(i)} - W_{\text{res}} X^{(i)}\|^2$$

Error

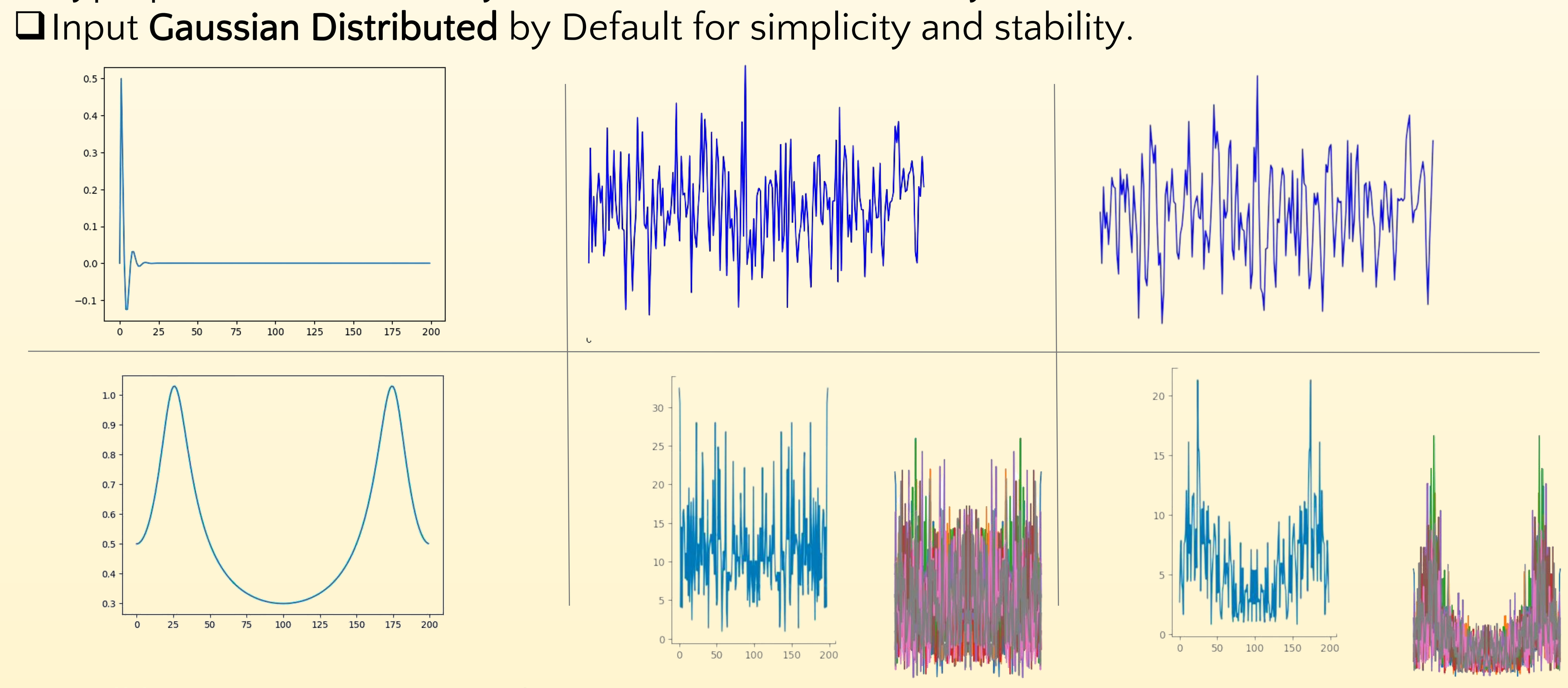




Results

❖ESN_0, Impulse Response & Frequency Response

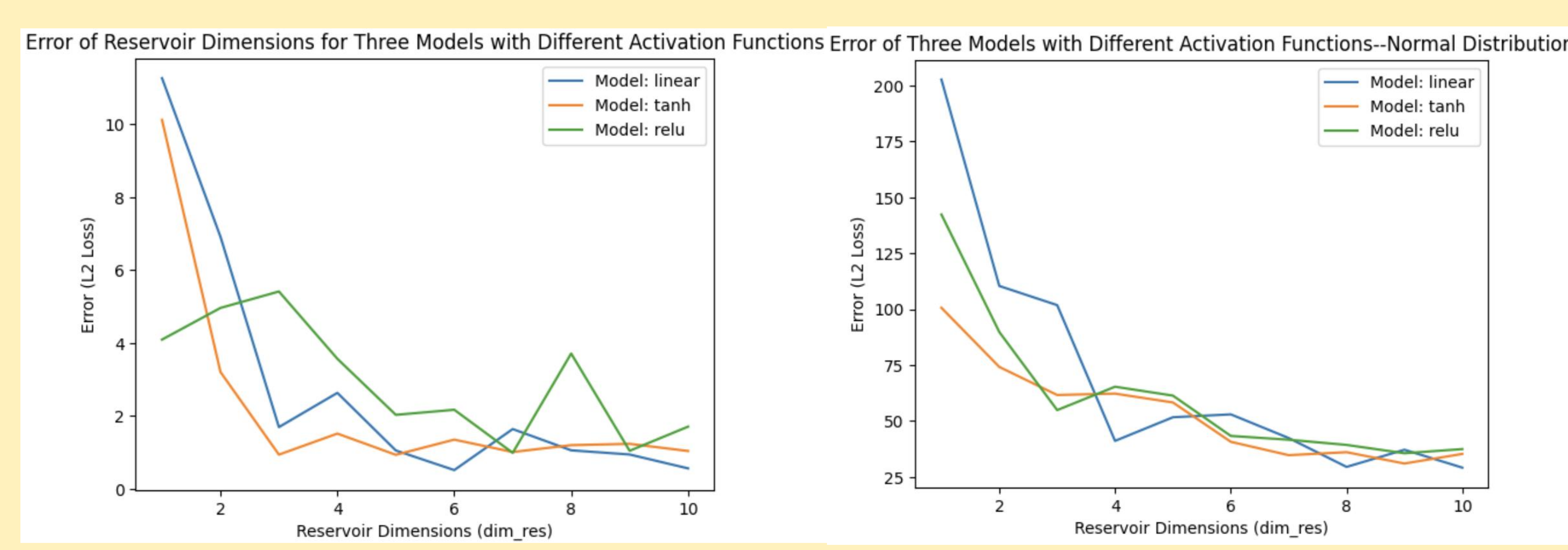
- Here we use default dimension of 1 for input and output.
- Using **Linear Input Transformation (LIT)** as ESN_0 simplified initialization,reduced hyperparameter sensitivity and maintained Consistency Across Tasks:
- Input **Gaussian Distributed** by Default for simplicity and stability.



- In all linear system, an **impulse response** represents how ESN_0 reacts to an instantaneous change or impulse in the input signal
  ⟹ how the system's behavior evolves over time.
- **impulse_response = ESN_0(impulse_input)**
  - The impulse response inherently contains frequency components involved in the system's response.
- **freq_response = FFT(impulse_response)**
  - The **spectrum** of a signal refers to the distribution of its frequency components.
  - Utilize **Fast Fourier Transform (FFT)** to uncover the spectrum of the impulse response of ESN_0.

❖Experiment of Different Parameters (Activation Function, Weight Initialization, Reservoir Dimension)

- 10,000 of total Test Cases of Length 1,000; 10 Epochs
- controling variable: generated training data from ESN_0
- Dimension of reservoir (d_res) from 1-10



- We set weights and freeze the value of W_res through normal distribution and redo the experiment; we devided the result according to the dimension of reservoir and sqeezing its size.

Conclusion

- The error is decreasing as dimension of the reservoir (d_res) increases, which indicates **Capacity Increase** and **Better Representation.**
- The optimal choice of activation function relies on on the complexity of the reservoir.
  - **Linear activation** might be well-suited for **lower dimensions** where simpler transformations suffice.
  - **ReLU** might be more effective for **larger dimensions** to capture more intricate patterns.
  - The **consistence** of **tanh** function could indicate its **balanced performance** across dimensions -> tanh might be a reliable choice when the optimal activation function isn't apparent.
- The significant error expansion due to normal distribution initialization underscores the importance of **weight initialization** in reservoir computing.
- This platform we build can be a paradigm for further investigation about Reservoir Computing, including impacts of data scale, optimization methods and generalization errors.

Acknowledgements

I express my sincere appreciation to Prof. Lizhong Zheng and Dr. Xiangxiang Xu my dedicated advisors, whose unwavering encouragement, expertise, and mentorship have been instrumental throughout this research journey. I extend my gratitude to MIT Research Laboratory of Electronics (RLE) and Research in Science & Engineering (RISE) Internship track by Boston University, for providing the necessary resources. Finally, I'd like to thank our families and friends for their unwavering support, understanding, and encouragement throughout the research process. Their moral support and discussions greatly enriched the conceptualization and interpretation of results. This research was supported in part by the National Science Foundation(NSF) under Award CNS-2002908.

Reference

[1] 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.
[2] Oppenheim, A. V., & Verghese, G. C. (2018). Signals, systems and inference. Pearson Higher Ed.