XLI ENCUENTRO DE ECONOMISTAS DEL BANCO CENTRAL DE RESERVA DEL PERÚ

STUDY OF ARTIFICIAL INTELLIGENCE TOPOLOGIES FOR FORECASTING

DESIGNING NEURAL NETWORKS FOR MACROECONOMIC FORECASTING *

Presenter: Andrew Garcia, Ph.D.
Collaborator: Juan Aquino, Ph.D.

October 24, 2023

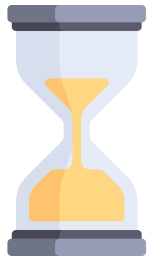
* The views expressed here are those of the authors and not necessarily of the BCRP

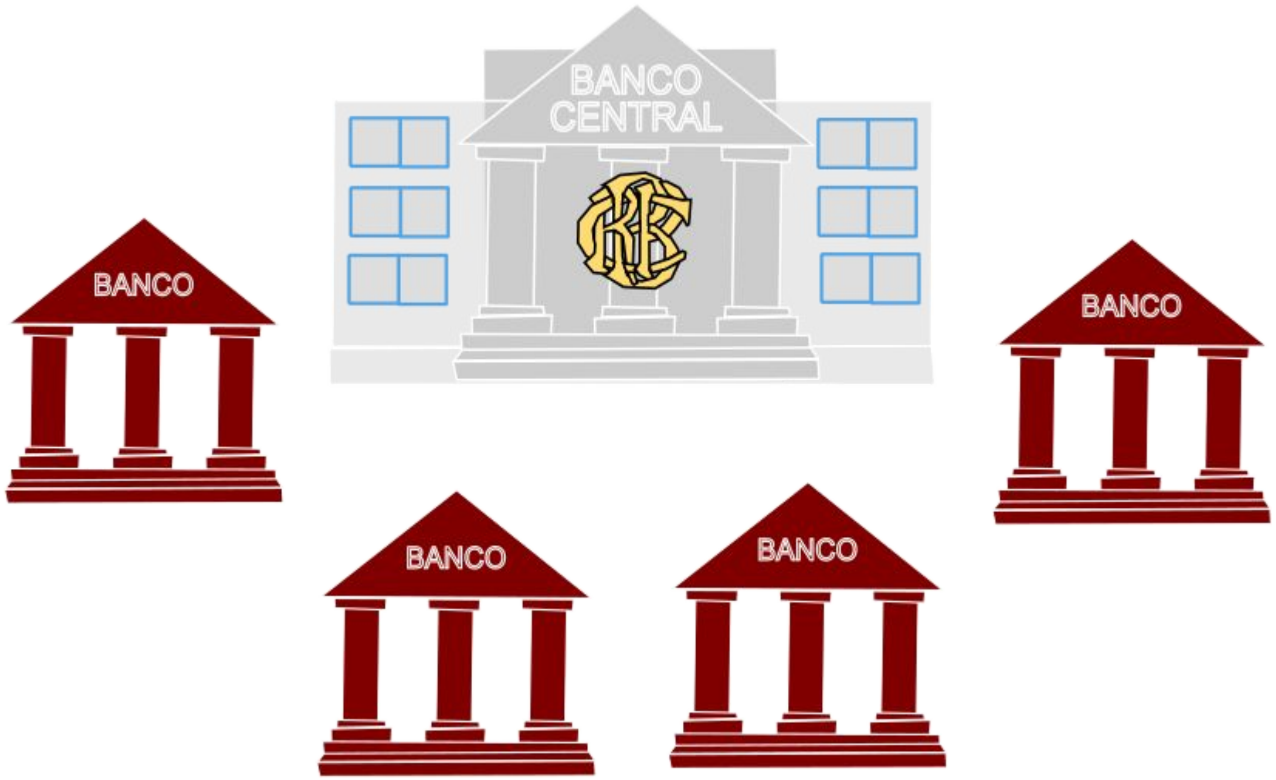
Objectives

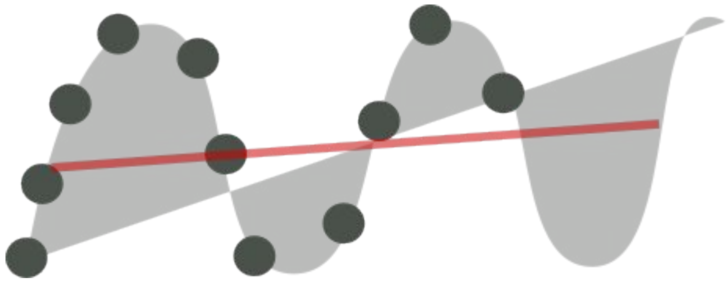
- Develop specialized neural network architectures for forecasting.
- Explore the impact of different architectural paradigms on forecasting performance.
- Assess the influence of components like Time2Vec and non-local blocks.
- Conduct empirical evaluations using real-world data

Why Use Neural Networks for Macroeconomic Forecasting?

- **Timely and Accurate Decisions:** Central banks shape economic policies, requiring precise and timely decisions.
- **Challenges in Traditional Methods:** Traditional forecasting struggles with complex financial systems.
- **Information Overload:** Central banks manage massive datasets, requiring advanced tools.
- **Policy Formulation:** AI-driven forecasting aids adaptive policy formulation.
- **Competitive Advantage:** Central banks using AI gain a policy-making edge.







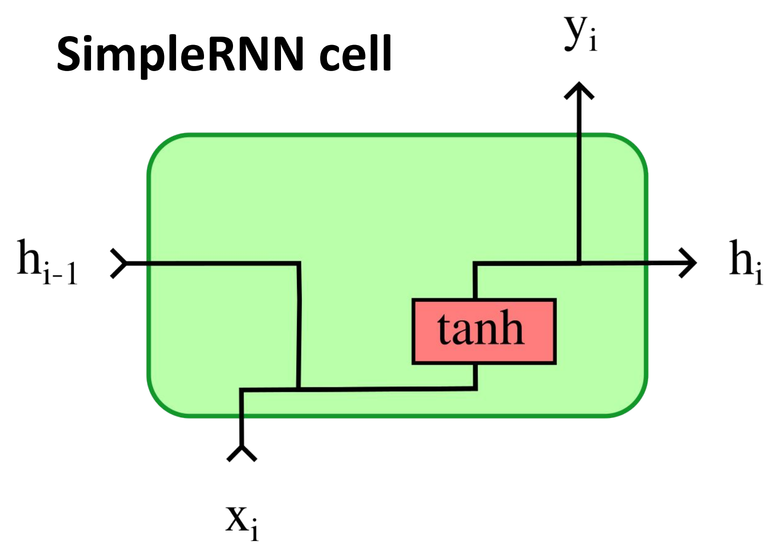
Agenda

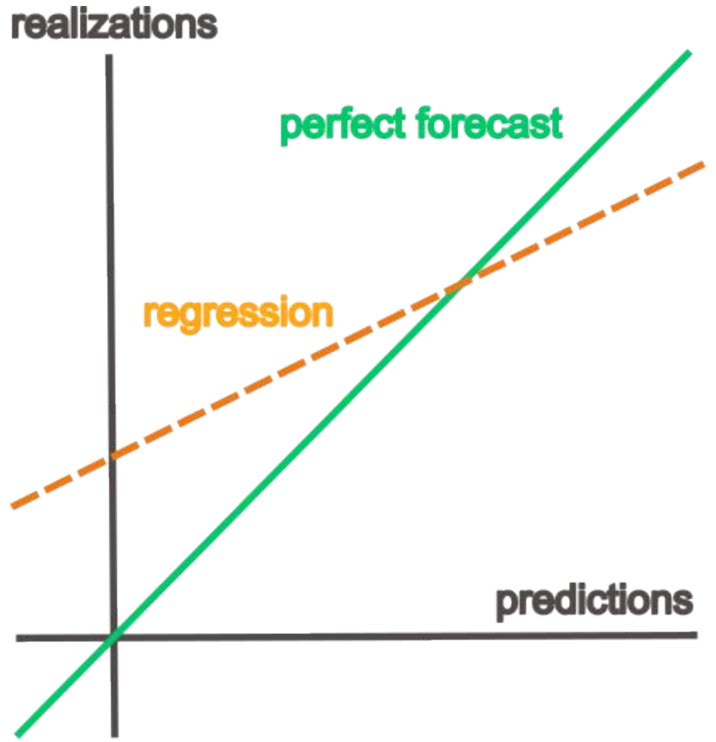
- Recurrent Neural Network Architectures
  - Overview
  - FixedSeq and Seq2Seq Networks
  - Component Integrations

- Research Methodology
  - Data Collection and Processing

- Model Performance Evaluation
  - Learning Curves and Methodology
  - Forecast Evaluation
    - Horizon MSE between Realizations and
      - E|Random Walk|, E|Autoregression|
      - Neural Network Predictions

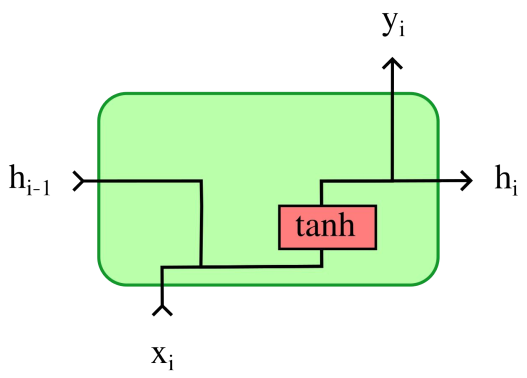
- Results
  - Key Findings



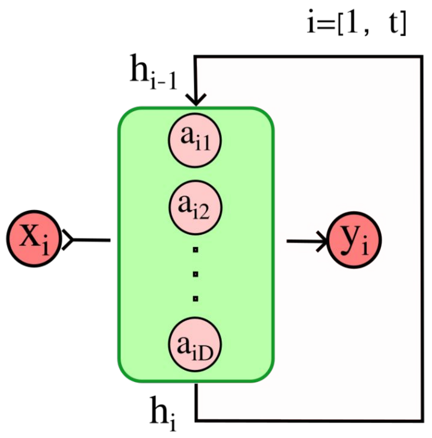


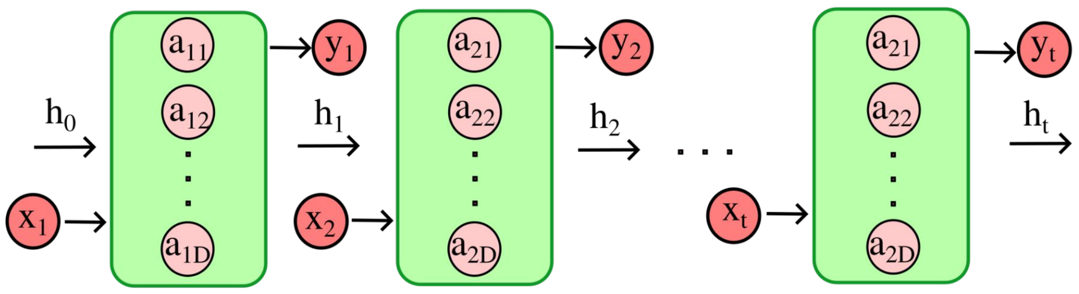
Recurrent Neural Networks

- **Specialized to process sequential data**
- **Key Features**
  - **Cell States**
    - Memory of network
  - **Hidden States**
    - Output sent from one cell to the next
  - **Outputs**
    - Each cell state gives an output
  - **Sequential Propagation**
    - Continuum of information from one cell to the next



recurrent unit representation

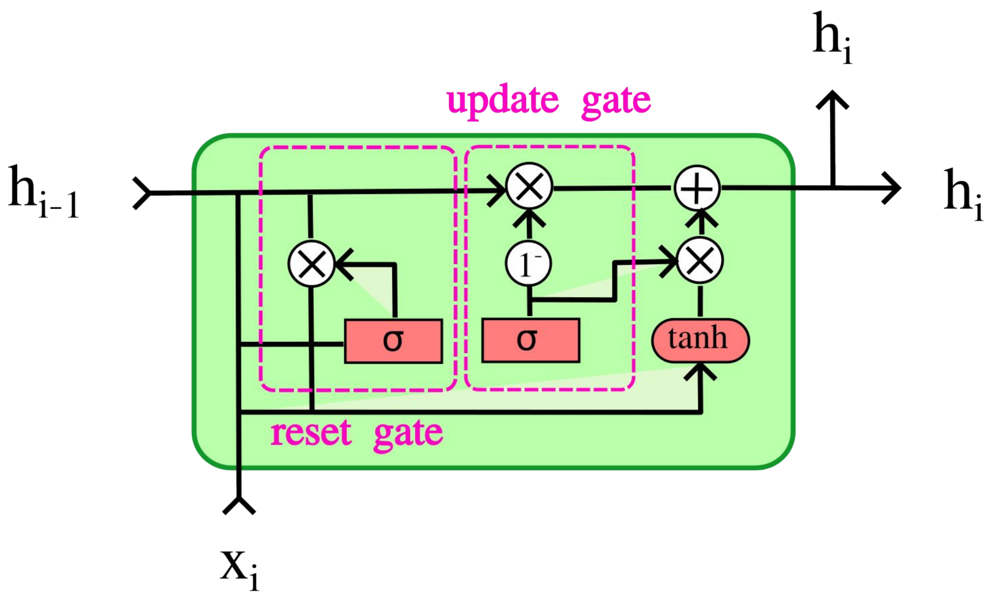


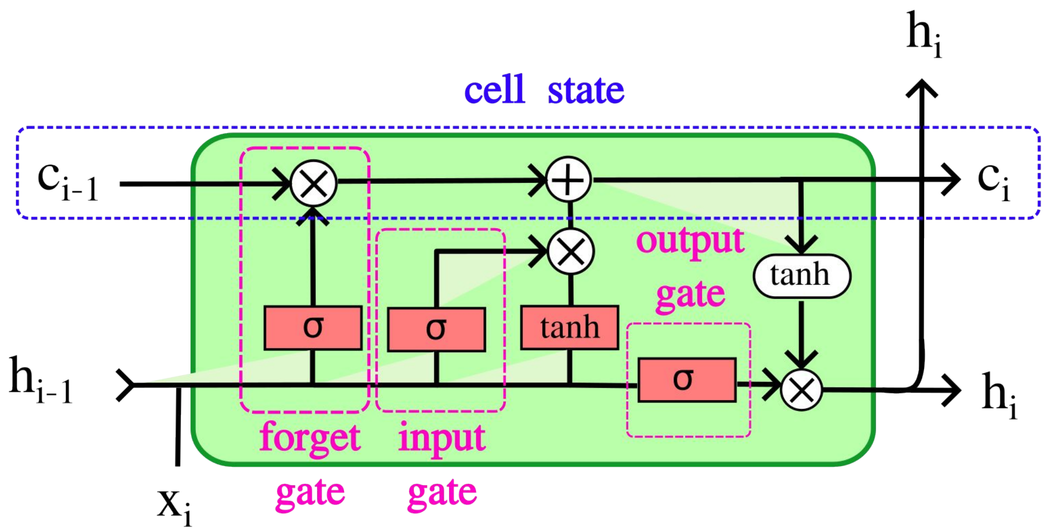


unfolded network

Advanced RNN Units

- **Recurrent neural networks with gates:** regulate information flow to capture and retain long-term dependencies
- **Two Explored**
  - **Long Short-Term Memory (LSTM):** Fundamental components of the LSTM include the input, forget, and output gates, and cell state.
    - Used in Seq2Seq networks
  - **Gated Recurrent Unit (GRU):** merges the cell state and hidden state and uses fewer gates.
    - Used in FixedSeq networks

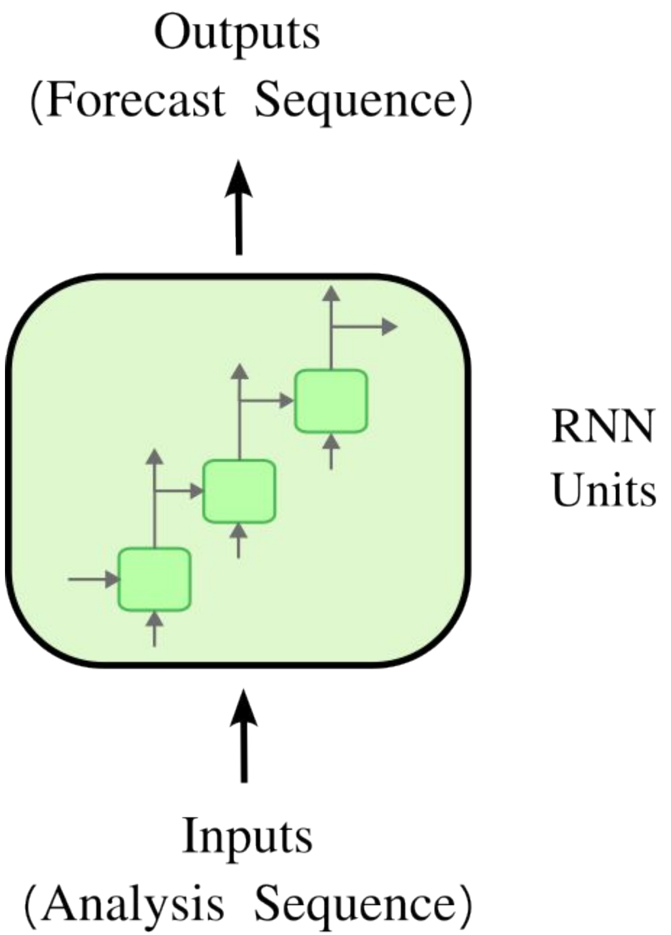




FixedSeq Network

- **FixedSeq is composed of a set of GRU units alone**
- **Fixed length constraint:** Inputs (X) must have the length as Outputs (y)

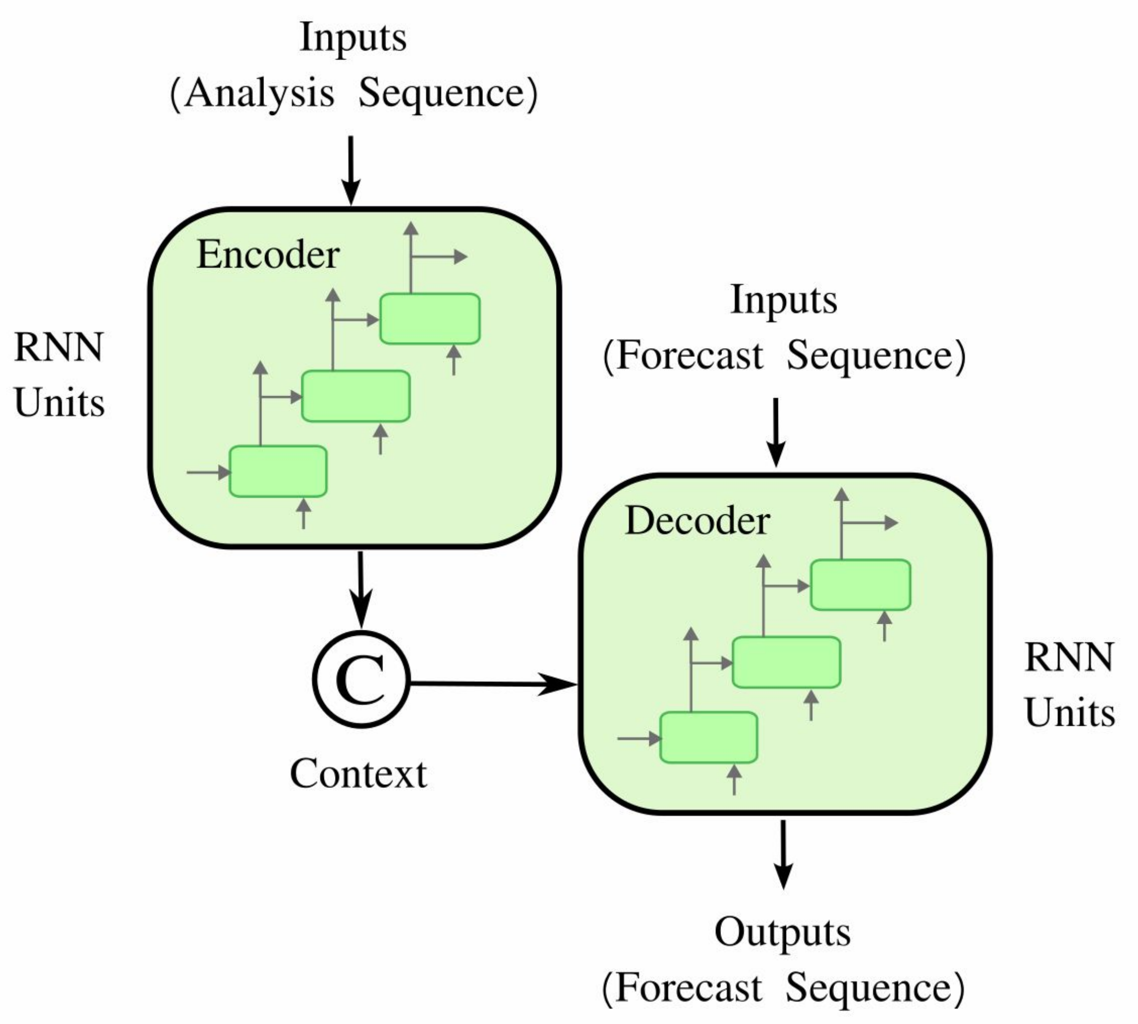
- **Inputs (X) :** Sequential data to infer from
  - **Analysis time windows**
- **Outputs (y):** Sequential data to fit / predict
  - **Forecast time windows**



Seq2Seq Network

- **Seq2Seq is consists of an encoder-decoder mechanism**
- **Flexible length features:** Inputs (X) can have different lengths as Outputs (y)

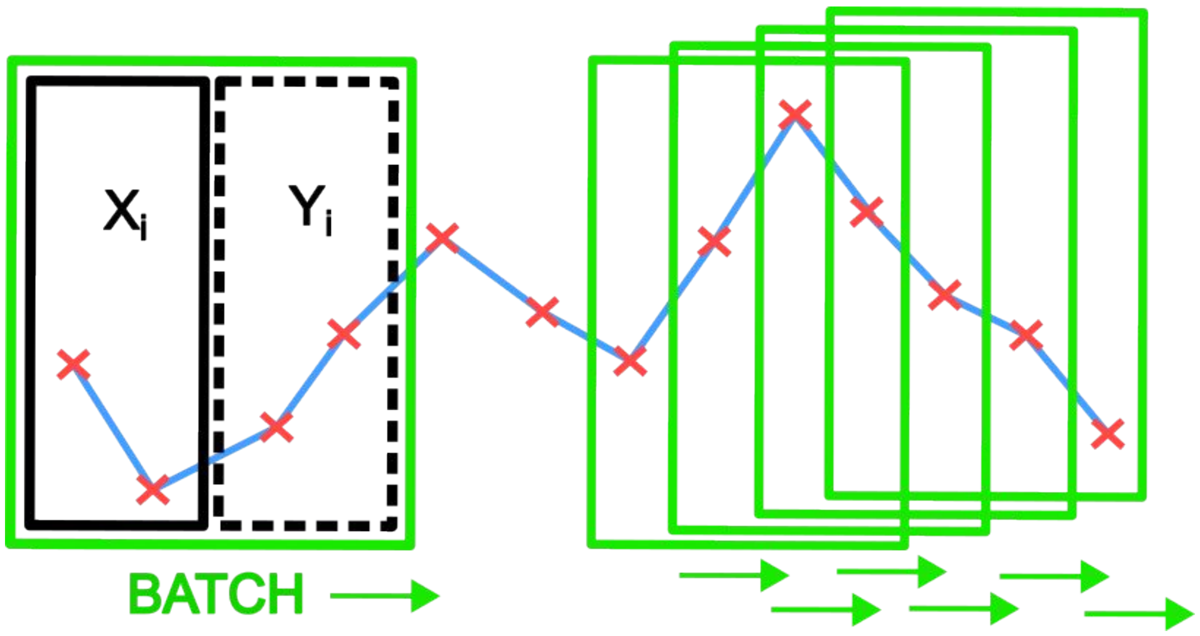
- **Encoder**: Composed of LSTM units
  - Inputs are fed and Cell State is output as “Context”
- **Decoder**: Composed of LSTM units
  - Context from Encoder block is fed and final Outputs (y) are released



Data Collection and Processing

- **Baseline: 20-year dataset of monthly inflation index data from Peru, spanning until October 2023**

- **Sliding window batches:** Multiple sequential sector pairs
  - **Sector 1: analysis time window**
  - **Sector 2: forecast time window**
- Random shuffling of above batches for training and testing sets




```
import bcrpy

banco = bcrpy.Marco()

banco.codigos = ['PN01273PM']

banco.fechaini = '2003-10'
banco.fechafin = '2023-10'

df = banco.GET()
```

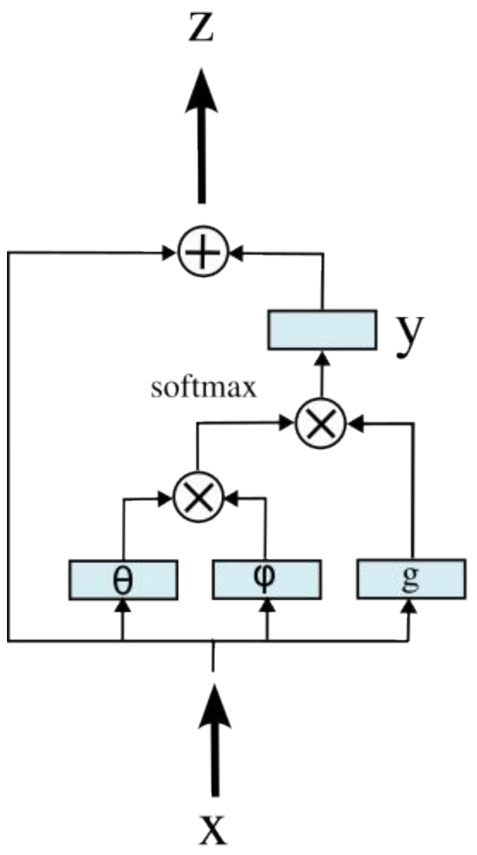
Non-Local Blocks (Component Integrations)

- **Inspired by Transformer architecture's Attention Layers**
  - Multihead attention layers are specialized to process embedded string information ('words')
- **Non-local blocks process numerical data**
  - Developed by Facebook Research (2017) to capture **long-range dependencies of spatial information**
- Here used to prove whether they can further improve capturing long-range dependencies in time series

$$f(\mathbf{x}_i, \mathbf{x}_j) = e^{\theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j)}$$

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j).$$

$$\mathbf{z}_i = W_z \mathbf{y}_i + \mathbf{x}_i$$

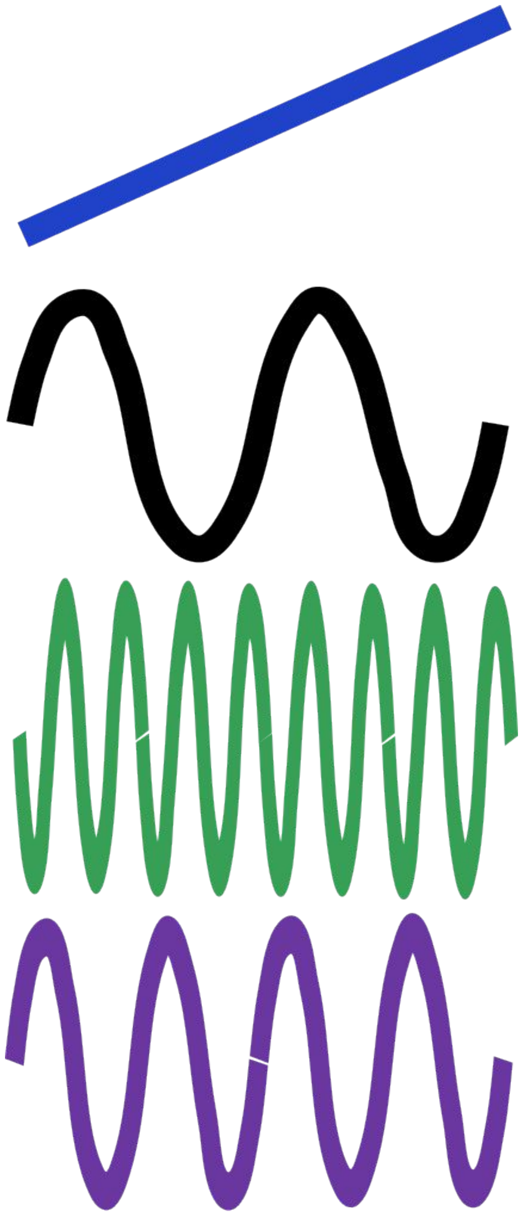


Non-Local Block

Time2Vec Time Encoding (Component Integrations)

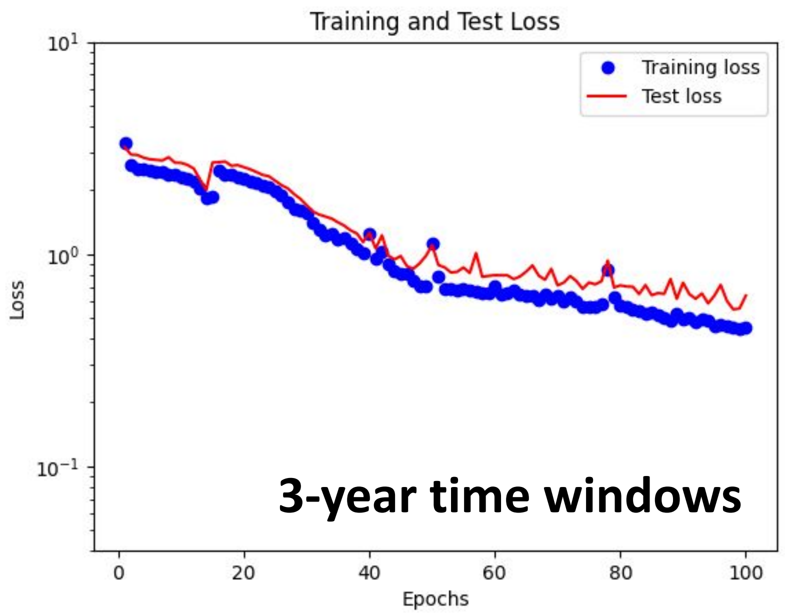
- Developed by Borealis AI (2019)
- Reported to learn features of time (progression, periodicity, scale) through time encoding
- The Time2Vec encoder may take a time series (or scalar notion of time) and encode it to a series of vectors with the same length:
  - Trend Component (one vector)
  - Periodic Components (custom number of vectors)

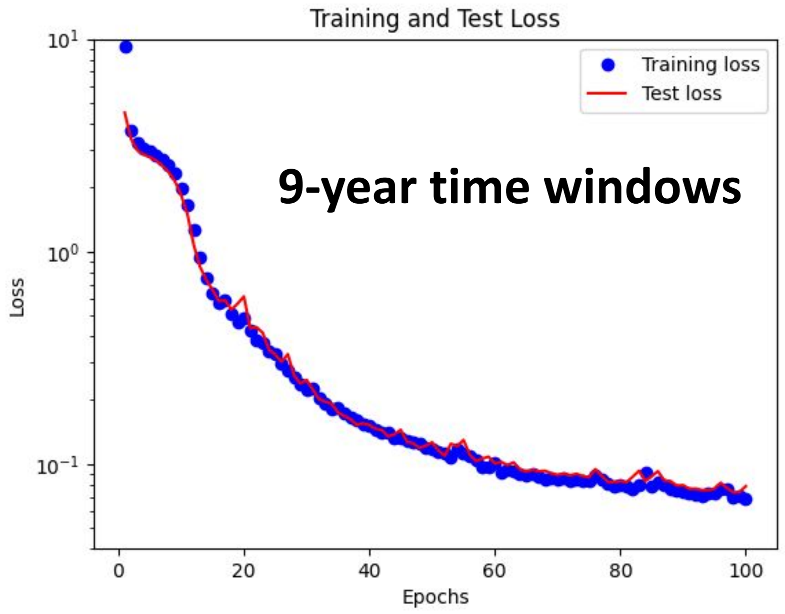
$$\mathbf{t2v}(\tau)[i] = \begin{cases} \omega_i \tau + \phi_i, & \text{if} \quad i = 0. \\ \mathcal{F}(\omega_i \tau + \phi_i), & \text{if} \quad 1 \leq i \leq k. \end{cases}$$



Bare FixedSeq

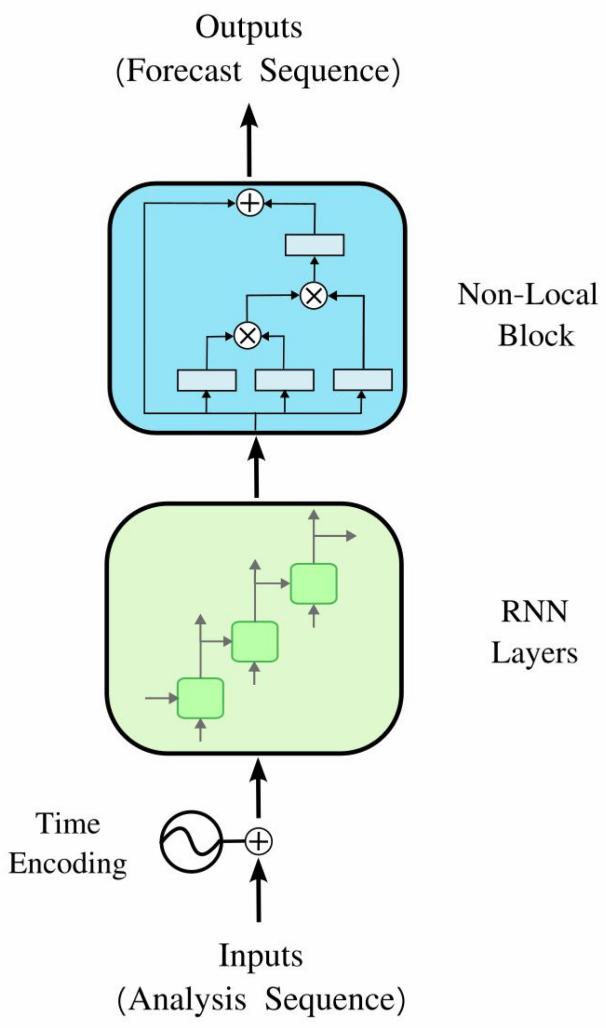
- **Effect of analysis [time] window lengths**
- **Hypothesis:** Providing more information to neural network will result in improved data inferences
- **FixedSeq network using the longest monthly analysis windows top-performing according to learning curves.**
    - Lowest overall MSE loss
    - More narrow training-testing loss gap
        - Implies better generalization from unseen data
- **Shortcomings of long windows:**
    - More Information is demanded
    - Less training and testing batches

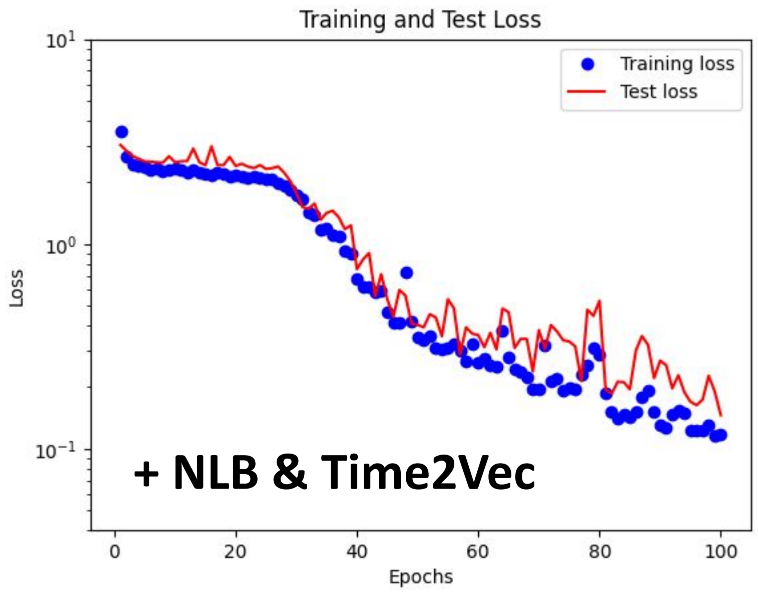




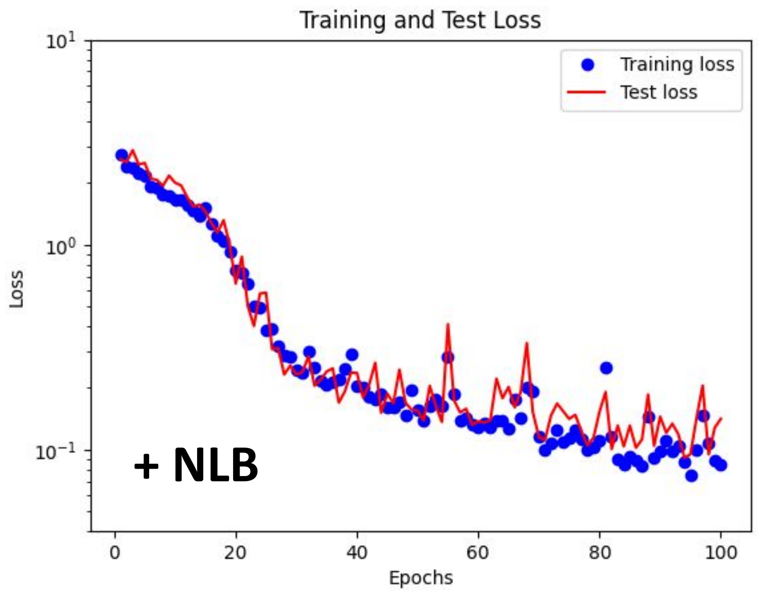
FixedSeq with Component Integrations

- **Lesser performing RNN with non-local block and time encoding**
  - FixedSeq-FULL (top)
- **And with nonlocal block alone**
  - FixedSeq-NLB (bottom)
- Both show dramatic improvement, though FixedSeq-NLB shows superior performance



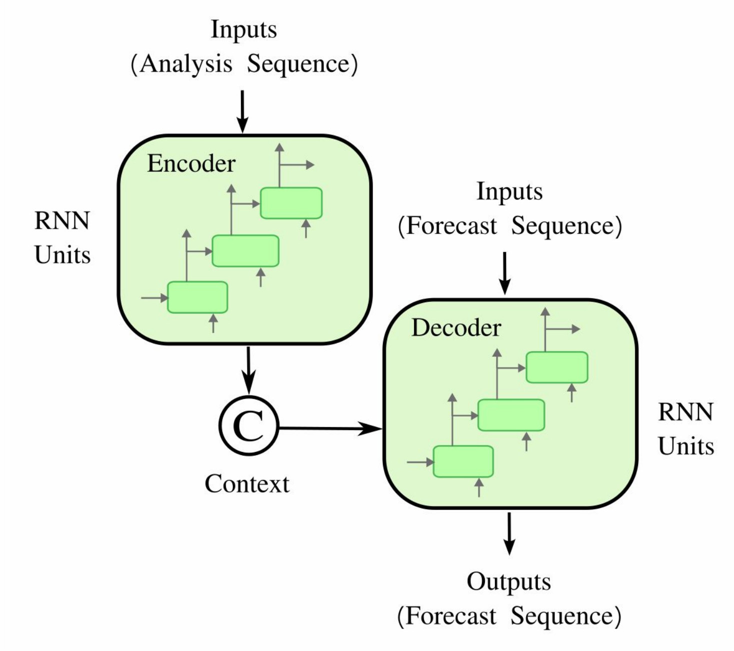


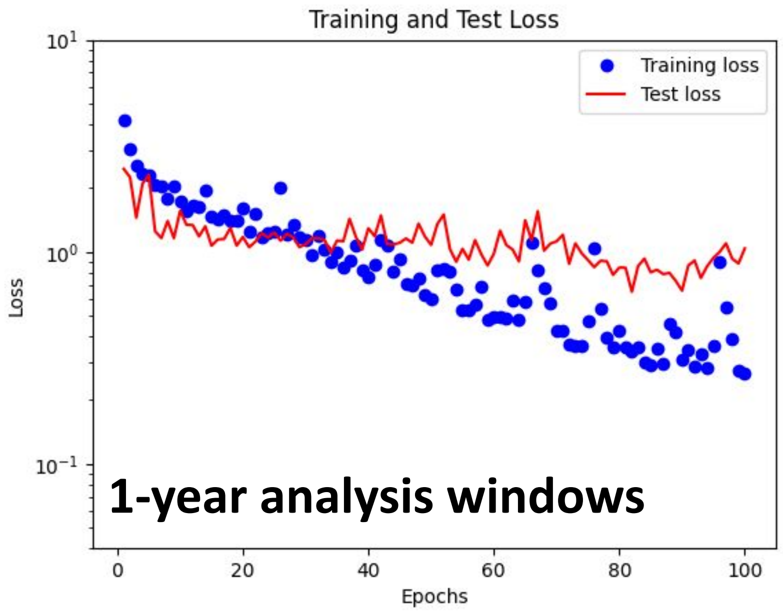
3-year time windows

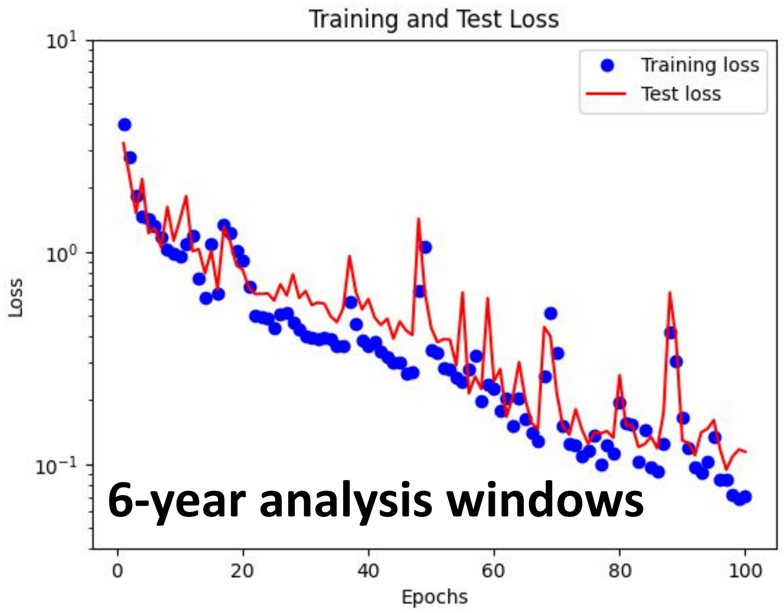


Bare Seq2Seq - 1-year forecast windows

- **Reducing the forecast time windows** to 12 months (1-year) while changing the analysis time window size.
- **Both cases show significant improvement to the analogous FixedSeq results**
- **Longer analysis time windows give better learning curve performance**
  - Lower errors
  - Better generalization





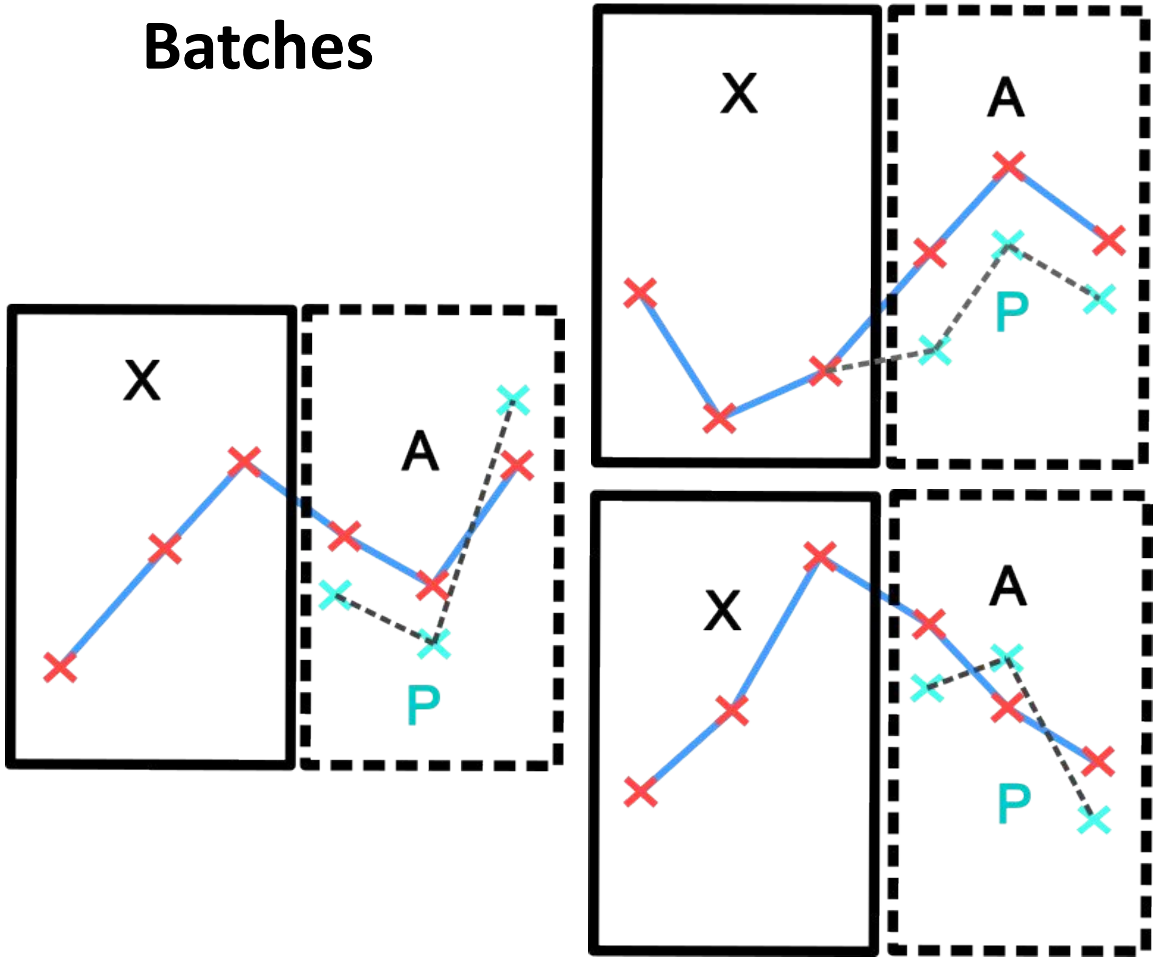


Forecast Evaluations

- **Develop a methodology inspired by Mincer and Zarnowitz (1969) to compare forecasting error of RNN against benchmarks**
  - **Across-time comparison of errors between realizations (ground-truth) and predictions**

$$g(e) = \frac{1}{n}\sum_{i=1}^{n} E(A_i - P_i)^2$$

Batches



Random Walk

$$A_t = A_{t-1} + \epsilon_t$$

$$E(A_t^{rw}) = A_{t-1}$$

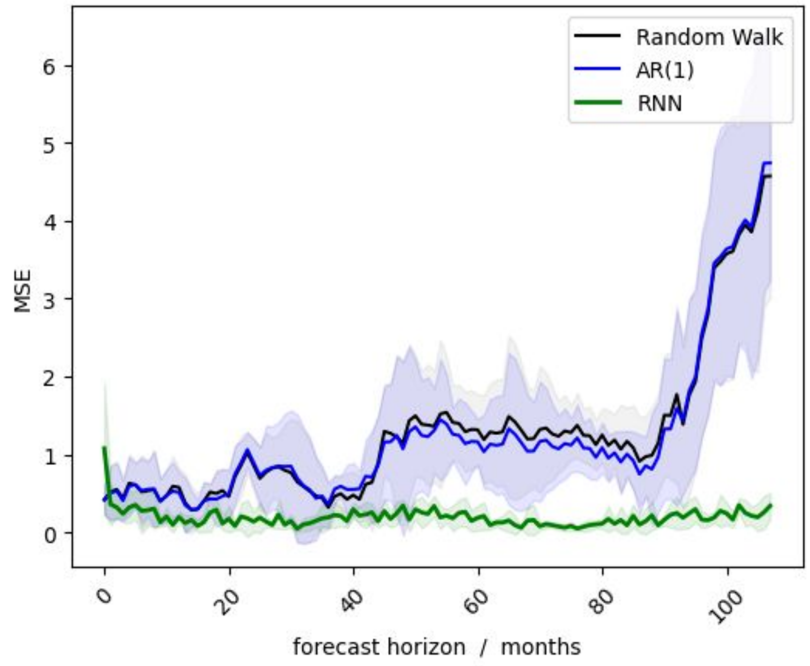
Autoregression

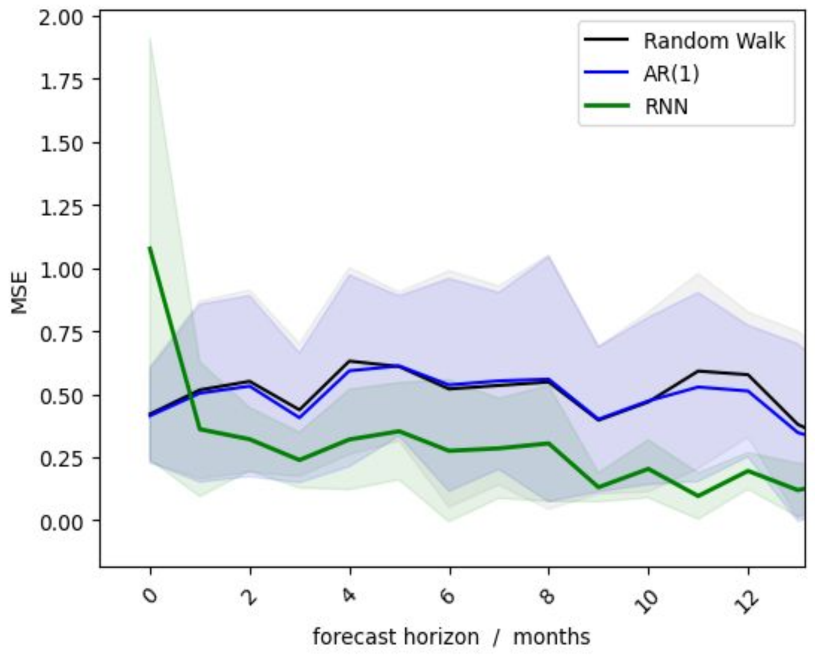
$$A_t = \alpha + \beta_1 A_{t-1} + \epsilon_t$$

$$E(A_t^{ar1}) = \alpha + \beta_1 A_{t-1}$$

Forecast Evaluations

- Most information-intensive model
  - Bare FixedSeq, 9-year (108 months) analysis & forecast windows
- Beats the random walk and AR(1) benchmarks at longer times
- Fails to do so at earlier forecast time regimes
  - Compare against other better-performing designs

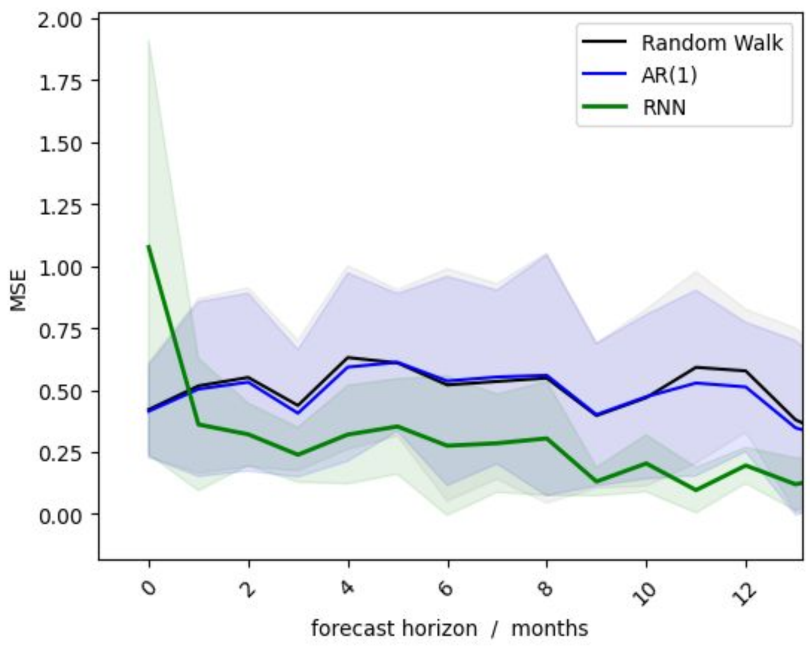




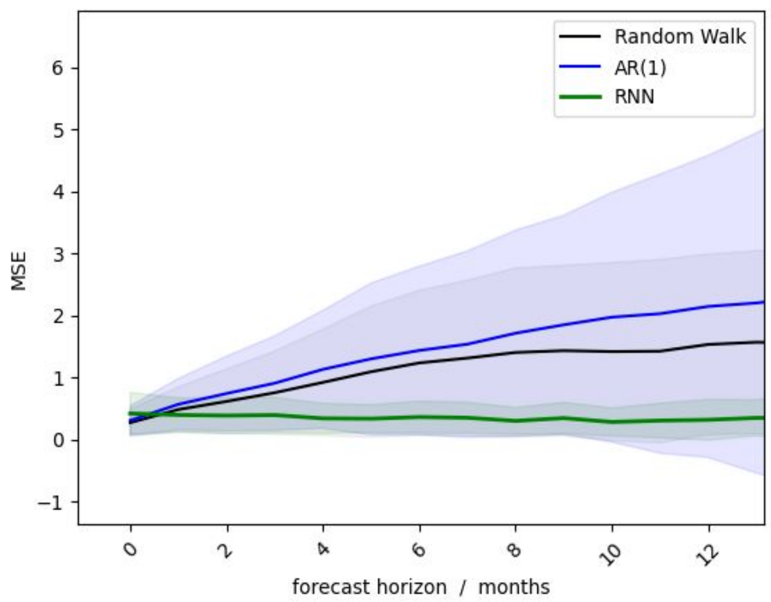
Forecast Evaluations

- Compare the latter model,
  - bare FixedSeq (top), to a
  - FixedSeq-NLB with 3-year analysis / forecast windows (bottom)
- bare FixedSeq noise attributed to significantly lower batch density
  - Less power to report accurate forecast
- Model # 2 surpasses random walk and AR(1) with greater confidence
  - Mean and deviation significantly lower to benchmarks

FixedSeq 9-year windows



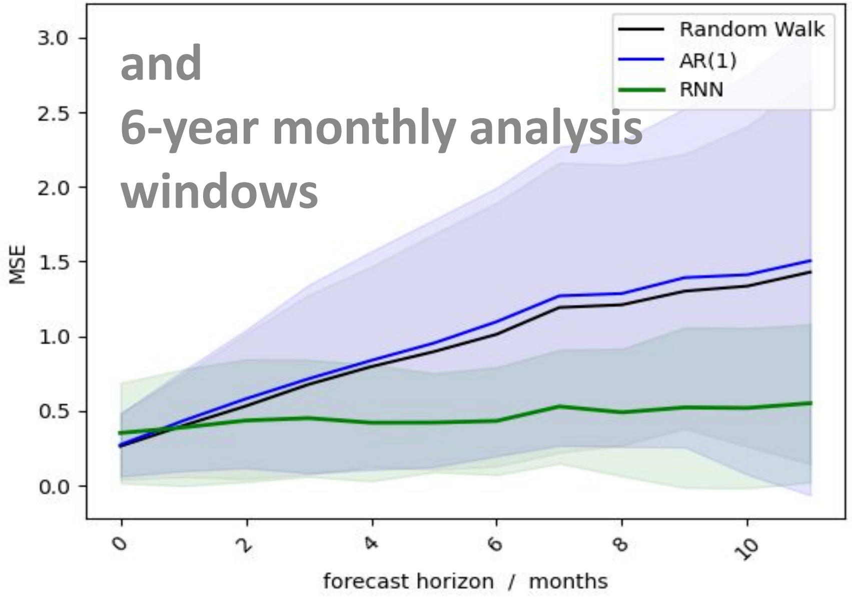
FixedSeq-NLB 3-year windows



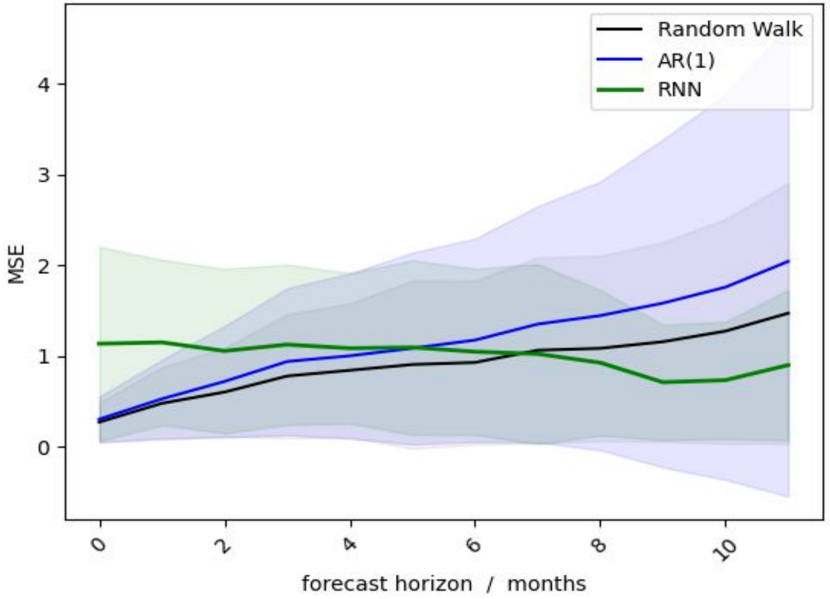
Forecast Evaluations - Seq2Seq

- **Best performing Seq2Seq along with FixedSeq counterpart**
  - **Same Forecast Window Size (1-year)**
- **FixedSeq cannot take longer Analysis Windows than its Forecast Length**
- **Significant overperformance of Seq2Seq**
  - **Beats benchmarks after 2 months**
  - **FixedSeq unable to beat benchmarks**

Seq2Seq 1-year forecast windows



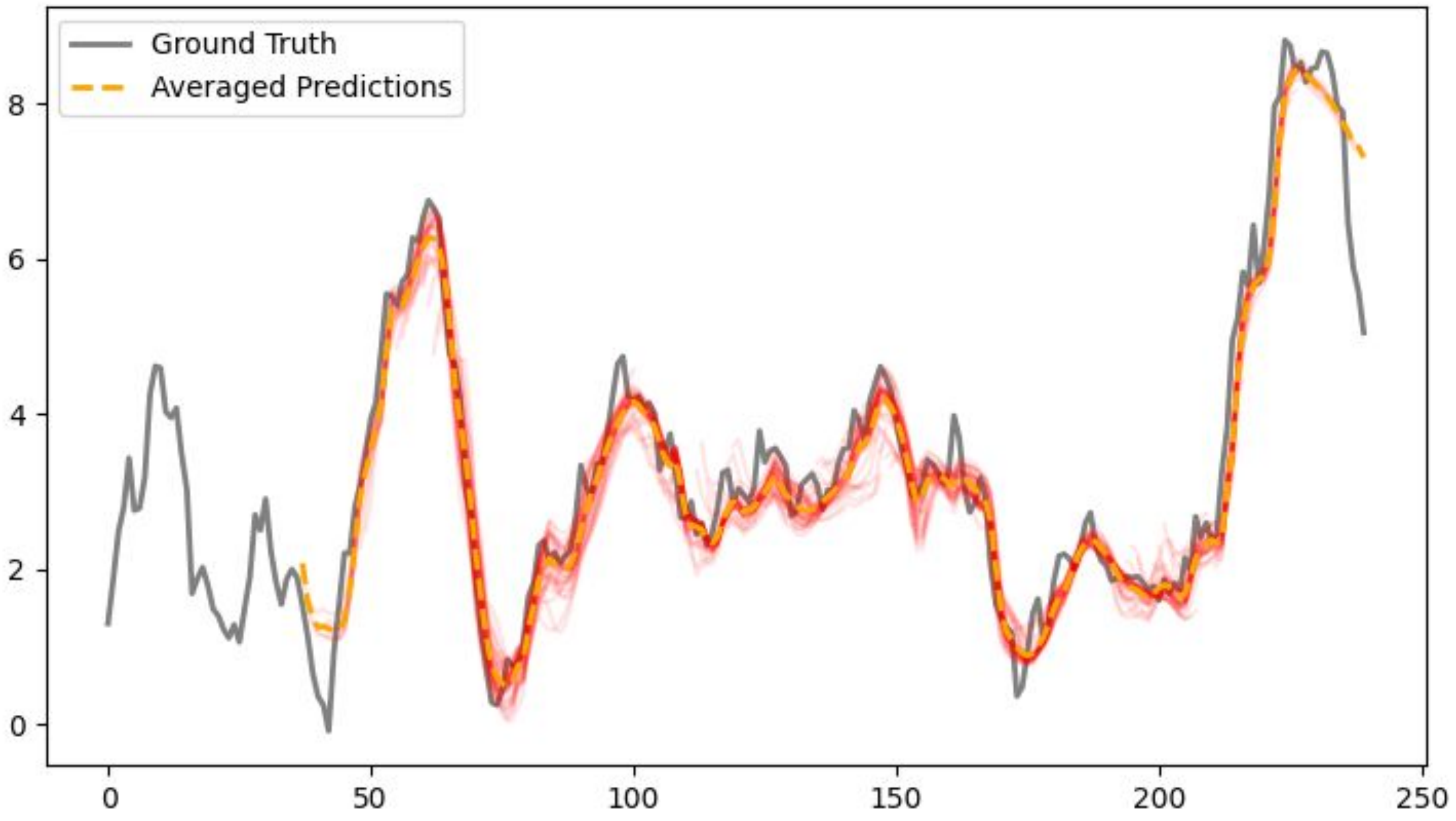
FixedSeq 1-year forecast windows



Conclusion

- Developed a methodology for assessing forecasting accuracy for neural networks
- Choice of neural network architecture and input characteristics significantly impacts forecasting performance
- Seq2Seq models demonstrate superior performance for cases where data availability is limited
- Adoption of specific architectural components can optimize forecasting accuracy

THANK YOU


Ground Truth
Averaged Predictions
0
2
4
6
8
0
50
100
150
200
250