A Conceptual Analysis of Recurrence Relations in Recurrent Neural Networks for Sequential Data Prediction

Gabriella Botimada Lubis - 13524006
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
lubis.gabriella47@gmail.com, 13524006@std.stei.itb.ac.id

***Abstract*— Sequential data prediction is essential in various fields such as finance, weather forecasting, and natural language processing. Recurrent Neural Networks (RNNs) are widely used to address these problems due to their ability to model dependencies over time. The core mechanism that enables this capability is the recurrence relation, which allows each hidden state in an RNN to depend on both the current input and the previous hidden state. This paper presents a conceptual analysis of the application of recurrence relations in RNNs for sequential data prediction. Instead of building new predictive models, this study focuses on examining how recurrence relations operate within existing RNN structures and how they influence the information flow and stability of predictions. The analysis also highlights the similarities between RNNs and classical statistical models, such as the Autoregressive Moving Average (ARMA) model, particularly in how both handle time-dependent data. The findings of this study show that the recurrence relation is the foundation of RNN performance, directly affecting its ability to process sequences, retain memory, and predict future values. Understanding the mathematical role of recurrence relations provides valuable insight into the strengths and limitations of RNNs in sequential modeling.**

***Keywords*—recurrence relations, Recurrent Neural Networks, sequential data prediction**

I. Introduction

Sequential data prediction is a fundamental task in various fields such as finance, natural language processing, signal processing, and weather forecasting. Many real-world phenomena evolve over time, forming patterns that can only be effectively modeled when past information is appropriately considered. Recurrent Neural Networks (RNNs) have emerged as one of the most prominent machine learning models capable of processing sequential data by capturing dependencies across time steps.

The core principle that enables RNNs to handle sequential data is the application of recurrence relations. Through these relations, each hidden state in the RNN is computed as a function of the current input and the hidden state from the previous time step. This recursive dependency forms the backbone of RNN architecture, allowing it to retain memory and learn temporal patterns. Unlike feedforward neural networks that process independent data points, RNNs are explicitly designed to handle ordered sequences by leveraging this internal recurrence mechanism.

This paper aims to analyze the application of recurrence relations within the structure of RNNs for sequential data prediction. The focus of the study is not to develop new predictive models but rather to examine how the recurrence relation operates within existing RNN frameworks, how it governs the flow of information over time, and how it affects the stability and accuracy of predictions.

By analyzing the mathematical foundations and behavioral patterns of recurrence relations in RNNs, this paper provides a deeper understanding of why RNNs are effective for certain types of sequence learning and prediction tasks, as well as the limitations that arise from their recursive structures. This analysis contributes to a conceptual appreciation of the strengths and challenges of RNNs in sequential modeling.

II. Theoretical Framework

A. *Recurrence Relation*

A recurrence relation is a mathematical expression that defines each element in a sequence based on its preceding elements. It describes the dependency of a current term on one or more previous terms, forming a chain of relations across the sequence. This is a simple form of a recurrence function:

$$f(n) = A\,f(n-1) + B\,f(n-2)$$

This indicates that the current term $f(n)$ is computed from its immediate predecessors $f(n-1)$ and $f(n-2)$.

In time-dependent systems, recurrence relations are essential to capture the temporal dynamics of sequential processes. They provide a mathematical framework for understanding how the current state is influenced by historical information.

B. *Recurrent Neural Networks (RNN)*

RNN is a class of networks that process input with various information that has been obtained previously. The determination of decisions or results given from a particular input will be influenced by the information system that already exists. This happens because the Recurrent neural network has an internal memory that can remember a collection of information.

The way the recurrent neural network (RNN) works is not directly proportional to the feed-forward neural network, but rather passes through a loop that contains some of the previous information. Thus, the recurrent neural network not only considers the input at that time, but also considers other things that have been obtained previously.

Unlike traditional feedforward neural networks, which assume independence between input data points, RNNs are capable of modeling dependencies across time by maintaining an internal state known as the hidden state. The hidden state in an RNN is updated at each time step based on a recurrence relation:

$$h_t = f(w_{xh}x_t + w_{hh}h_{t-1} + b)$$

h_t: the hidden state at time step t
x_t: input at time step t
w_{xh}, $w_{hh}h_{t-1}$: weight matrices
b: bias vector
f: activation function, typically tanh or ReLU

The recurrence in this equation lies in the dependency of the current hidden state on the previous hidden state h_{t-1}, enabling the network to pass information through time.

C. ARMA

The Autoregressive Moving Average (ARMA) model is a classical method used in time series analysis to model and predict sequential data. It is built upon the idea that the current value of a data sequence is influenced by its own past values and by past prediction errors.

The ARMA model consists of two main components, Autoregressive Component and Moving Average. Autoregressive (AR) Component assumes that the current observation is directly related to a certain number of previous observations. In simple terms, it suggests that what happens now is strongly dependent on what happened recently.

Moving Average (MA) Component considers the effect of past errors or shocks on the current observation. It captures the random influences or deviations that affected previous predictions and incorporates them into the current calculation.

The ARMA model is widely used for forecasting in fields such as economics, finance, and engineering. It is especially effective when the time series data shows consistent, predictable patterns over time.

The ARMA model is one of the foundational tools in statistical forecasting because of its ability to describe sequential dependencies using relatively simple calculations. It is particularly useful when the sequence is stable or stationary, meaning its statistical properties, like mean and variance, remain constant over time.

Although the ARMA model is based on linear relationships, it shares an important conceptual similarity with Recurrent Neural Networks (RNNs). Both models rely on the idea that the present depends on the past.

The difference lies in complexity. ARMA uses fixed linear formulas to capture sequential patterns, while RNNs use more flexible and adaptive structures that can learn non-linear patterns from data.

In this sense, RNNs can be seen as a more advanced, modern extension of the ARMA concept, where the sequential influence is maintained but with the added ability to model complex and dynamic patterns that are beyond the capability of classical linear models.

This connection shows that understanding ARMA provides a helpful foundation for analyzing how RNNs process and predict sequential data.

III. Analysis

A. Application of Recurrence Relation in RNN

The core of Recurrent Neural Network (RNN) is the recurrence relation that governs the computation of the hidden state across time steps. The RNN that we deal with is

$$y_t = w_1 h_t + b_y,$$

t: time
y_t: predicted value
w_1: real value
h_t: hidden layer

The hidden layer is computed by

$$h_t = \tanh(w_2 x_t + w_3 h_{t-1} + b_h)$$

x_t: input data
w_2, w_3: real values
h_{t-1}: previous hidden layer

In the context of machine learning, let LS be the learning dataset, and k>2 be the size of LS. If the first departure time of the learning data is 1, then it can be said that $LS = \{x1, x2, \ldots, x\kappa\}$. Assuming that the initial condition of the hidden layer is 0 (h_0=0), we can calculate y_t for each time t. Here, x_t is the data at time t and yt is the predicted value, so we want to satisfy $y_t = xt+1$.

This formula explicitly represents a recurrence relation, where the current hidden state depends on both the current input and the hidden state from the previous time step. The recurrence ensures that the RNN retains information from past sequences, enabling it to model time-dependent patterns.

This relation structurally mimics a mathematical sequence recurrence, where each element is a function of its predecessors. In this RNN model, the recurrence relation acts as a memory chain, passing the influence of previous time steps forward into future computations.

B. Relationship Between RNN and ARMA Model

The recurrence relation in RNNs shares structural similarities with the Autoregressive Moving Average (ARMA) models commonly used in time series analysis. In ARMA models, future values are predicted as linear combinations of past observations:

$$\hat{x}_{\kappa+1} = C_0 x_\kappa + C_1 x_{\kappa-1} + C_2 x_{\kappa-2} + \cdots + C_l x_0 + C^* e$$

$x_0, \cdots, x_\kappa$: given data

However, the presence of a non-linear activation function such as tanh in RNNs introduces non-linearity, which distinguishes RNNs from traditional linear models. When the activation function is expanded using a Taylor series, the structure of the RNN can be viewed as an extension of the ARMA model with non-linear components and more flexible learning capabilities.

This non-linearity enables RNNs to capture more complex patterns, but it also introduces sensitivity to parameter changes and potential instability in learning sequences with irregular fluctuations.

C. *Sensitivity to Initial Conditions*

An important property of the recurrence relation in RNNs is the tendency of the hidden state sequence to converge toward fixed points under specific parameter configurations. For example, when certain weight parameters result in a product less than one, the hidden state sequence often stabilizes at a unique solution regardless of initial conditions.

Conversely, when the weight configuration exceeds a critical threshold, the hidden state sequence may have multiple fixed points. In this situation, the final outcome is influenced by the initial hidden state and input trajectory. The system can converge to different solutions depending on the starting point, which highlights the importance of initial conditions in RNN-based prediction tasks.

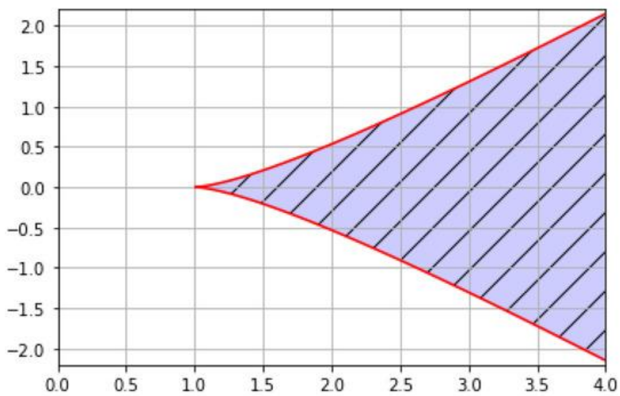


Figure 2.1
Source: https://www.mdpi.com/2073-8994/12/4/615

Figure 2.1 shows that if point (θ,b) is in the white region, there is one solution. If point (θ,b) lies in the red curve, there are two solutions. There are three solutions if point (θ,b) is in the blue region.

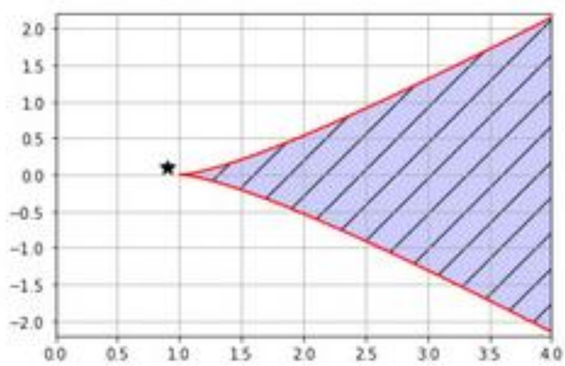


Figure 2.2 Example of One-Solution Case
Source: https://www.mdpi.com/2073-8994/12/4/615

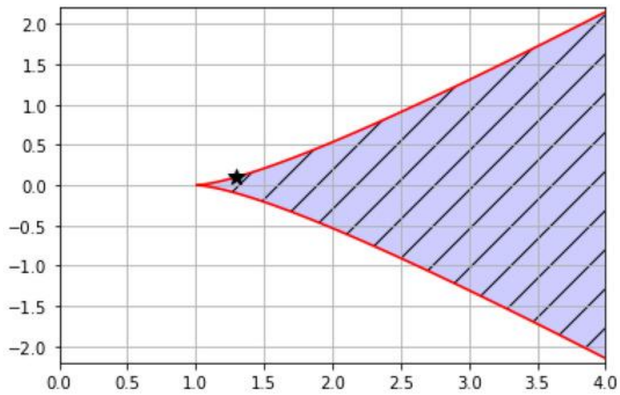


Figure 2.3 Example of Two Solutions Case
Source: https://www.mdpi.com/2073-8994/12/4/615

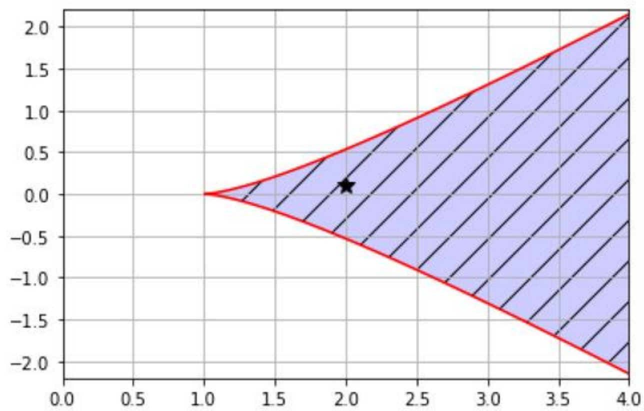


Figure 2.4 Example of Three Solutions Case
Source: https://www.mdpi.com/2073-8994/12/4/615

The fixed-point convergence behavior illustrates that the recurrence relation not only governs the short-term sequential computation but also controls the long-term stability and memory retention of the network.

The recurrence structure in RNNs inherently carries sensitivity to initial hidden state values. In certain parameter settings, small variations in initial conditions can lead the network to converge to different prediction trajectories or fixed points.

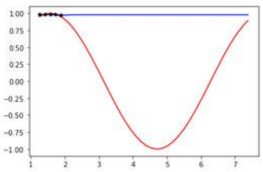
This sensitivity is particularly evident in cases where multiple solutions are possible. Such behavior emphasizes that the initial hidden state and the sequence of early inputs can significantly influence the prediction path and the final output.

Understanding this sensitivity is crucial when applying RNNs to real-world problems, as improper initialization or noisy starting data can steer the network toward unintended or suboptimal predictive outcomes.

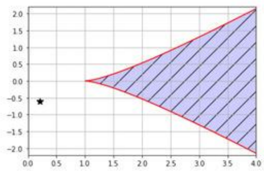
D. Sequential Pattern Learning Based on Recurrence

Through analysis of recurrence behavior, it is observed that RNNs can effectively learn and predict sequential patterns such as:

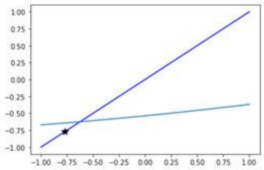
- Monotonic sequences



(a) Plot of real and predicted values.



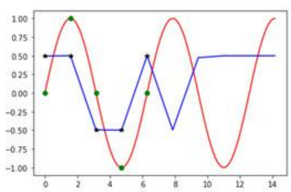
(b) Solution number region in Case 5.



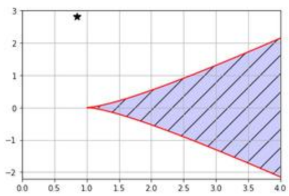
(c) Plot of *z* in Equation (17).

Figure 2.5
Source: https://www.mdpi.com/2073-8994/12/4/615

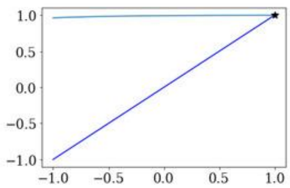
- Periodic or oscillating sequences



(a) Plot of real and predicted values.



(b) Solution number region in Case 7.



(c) Plot of *z* in Equation (17).

Figure 2.6
Source: https://www.mdpi.com/2073-8994/12/4/615

However, RNNs with simple recurrence structures tend to struggle when learning highly fluctuating or irregular sequences, particularly those with rapid or unpredictable variations. In such cases, the network may fail to fully capture the sequence dynamics and may instead converge prematurely to a stable but incorrect solution.

This limitation is consistent with the known challenge in RNNs regarding vanishing gradients and memory loss over long sequences. The basic recurrence relation, while powerful, is not always sufficient to maintain complex sequential information, especially over extended time steps.

E. Implications of Recurrence Relations in Sequential Prediction

The application of recurrence relations within RNN architectures plays a critical role in enabling sequential prediction. The recurrence structure:

- Establishes a direct link between past and present computations
- Controls how information from previous time steps is retained and propagated
- Influences the stability and convergence of prediction paths

Understanding the mathematical behavior of these recurrence relations provides valuable insights into why RNNs perform well on certain tasks but may falter in others. The selection of weight parameters, activation functions, and initial conditions all contribute to the network's ability to model and predict time-dependent data effectively.

In practical terms, when the recurrence relation is appropriately configured, RNNs can serve as robust tools for sequence modeling. However, to overcome the limitations in learning more complex patterns, deeper architectures or advanced variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) may be required to enhance the memory capacity and learning stability of the model.

IV. Conclusion

The analysis of recurrence relations within Recurrent Neural Networks (RNNs) reveals that recurrence is a fundamental mathematical structure that enables the network to process and predict sequential data. The hidden state updates in RNNs, which depend on both current inputs and previous hidden states, form a recurrence chain that effectively captures temporal dependencies.

This recurrence structure allows RNNs to retain information across time steps, making them suitable for learning sequential patterns such as trends, cycles, and dependencies in time-series data. The mathematical similarity between RNNs and classical autoregressive models, with the added benefit of non-linear transformations, highlights the flexibility of RNNs in modeling more complex sequence behaviors.

Through detailed analysis, it is evident that the recurrence relation significantly influences the stability, convergence, and predictive capacity of the network. Specific parameter settings can lead the network to converge toward one or more fixed points, and the sensitivity to initial conditions may affect prediction paths. These characteristics emphasize the importance of carefully understanding and configuring the recurrence structure in sequential learning tasks.

However, simple RNNs also have notable limitations, particularly when dealing with highly fluctuating or complex sequences. In such cases, the basic recurrence mechanism may not adequately capture the necessary patterns, and more advanced architectures such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) may be required to enhance performance and stability.

In conclusion, the recurrence relation is not only the defining feature of RNNs but also the key element that determines their effectiveness in sequential prediction tasks. A thorough comprehension of this mathematical structure is essential for optimizing RNN-based models and improving their practical application in various fields.

V. Acknowledgment

First and foremost, the author would like to express my deepest gratitude to God for the guidance, strength, and blessings that have enabled me to complete this paper successfully. Without His grace, this work would not have been possible. The author would also like to sincerely thank the lecturer of ITB Discrete Mathematics, Dr. Rinaldi Munir and Arrival Dwi Sentosa, M.T, who has provided invaluable support, knowledge, and encouragement throughout the process of writing this paper. The guidance and insights shared have been instrumental in deepening my understanding of the topic. Additionally, the author is truly grateful to family and friends for their endless support, patience, and motivation. Their unwavering belief in me has been a constant source of inspiration.

References

[1] J. Park, “Analysis of Recurrent Neural Network and Predictions”. Accessed on June 20th 2025 from https://www.mdpi.com/2073-8994/12/4/615.

[2] Informatika.stei.itb.ac.id., “Deretan, rekursi, dan relasi rekurens (Bagian 2)”. Accessed on June 20th 2025 from https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/11-Deretan,%20rekursi-dan-relasi-rekurens-(Bagian2)-2024.pdf.

[3] I, Farisi, “Penerapan Model Recurrent Neural Network (RNN) untuk Prediksi Curah Hujan Berbasis Data Historis”, 2024, INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS Vol. 9, No. 2.

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Juni 2025

Gabriella Botimada Lubis
13524006