Episodic Memory Theory of Recurrent Neural Networks: Insights into Long-Term Information Storage and Manipulation

Anonymous Authors[1]

Abstract

Recurrent neural networks (RNNs) have emerged as powerful models capable of storing and manipulating external information over long periods in various domains. Yet, the mechanisms that underly this behavior remain a mystery due to the black-box nature of these models. This paper addresses this question by proposing an episodic memory theory of RNN dynamics, enabling a more comprehensive understanding of the RNN weights as memories and inter-memory interactions. This approach sheds light on the inner workings of RNNs and connects to existing research on memory representation and organization. The theory extends the current linearization approaches by providing alternative interpretations of the eigenspectrum and its connection to the long-term storage and manipulation of information. We discuss how the segregation, representation, and composition of the variable binding problem—a fundamental question in cognitive science and artificial intelligence—can be mechanistically interpreted within the theory. Using an elementary task - repeat copy, we demonstrate the validity of the theory in experimental settings. Our work represents a step towards opening the black box of RNNs, offering new insights into their functionality and bridging the gap between recurrent neural networks and memory models.

1. Introduction

Recurrent neural networks (RNNs) are unique neural models capable of comprehending time-dependent connections thanks to their feedback loops. However, their stochastic training process (Rumelhart et al., 1986; Sussillo & Abbott, 2009; Martens & Sutskever, 2011) often makes them inscrutable, with essential task-oriented calculations concealed within multidimensional dynamics. Current methods of deciphering RNNs view them as non-linear dynamic systems, applying linear approximations at static or slow-evolving points to illuminate their functions (Sussillo & Barak, 2013).

The first step to understanding an RNN's functionality is pinpointing the fixed and slowly variable points using optimization techniques. Then, the phase space flow is assembled from each linearly approximated area. A detailed examination of each area's long-term behavior is carried out through spectral analysis of the related linear dynamical systems (Strogatz, 1994). This evaluation discloses vital information about the dynamics, such as their patterns of convergence, divergence, stability, or cyclic behavior. Despite the general applicability of this approach, it is still challenging to comprehend when the task spans multiple dimensions with non-converging dynamics.

2. Background

The required background for our theory includes two distinct but complementary strands of research—variable binding and neural memory models, which until now were independently studied.

2.1. Variable Binding

In humans, the generalization phenomenon is attributed to the ability to ”bind” information together and symbolically compose it to create novel outputs (Johnson-Laird, 2010). This ability for variable binding enables applying what is learned in one situation to a future, novel situation (Whitehead). The role of this type of generalization motivated symbolic AI approaches to model intelligence through formal logic and symbol manipulation. The early stages of symbolic AI can be traced back to the 1950s and 1960s, with the work of pioneers like Allen Newell and Herbert A. Simon, who developed the Logic Theorist (Newell & Simon, 1956) and General Problem Solver (Newell & Simon, 1995). In the 1970s and 1980s, symbolic AI saw significant advancements with the development of expert systems, such as MYCIN (Shortliffe & Buchanan, 1990; Buchanan

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

& Shortliffe, 1984) and DENDRAL (Lindsay et al., 1980; 1993) which were rule-based systems designed to mimic the decision-making ability of a human expert. The systems used if-then rules to navigate large databases of knowledge and make decisions. However, symbolic AI started to face criticism and challenges in the late 1980s. One of the main criticisms was its inability to handle uncertain or incomplete information effectively, unlike humans (Kautz, 2022). As a result, interest shifted towards alternative approaches, such as connectionist models (neural networks) (Rosenblatt, 1958; Rumelhart et al., 1986) and probabilistic methods (Russell & Norvig, 1995), which offered better handling of uncertainty and could learn from data. In the late 2010s, integrating symbolic reasoning into RNNs started gaining traction. One significant early approach was the Neural Turing Machine (Graves et al., 2014), which integrated memory manipulation capabilities into neural networks, allowing them to read and write to memory in a manner similar to how a Turing machine operates (Turing, 2021). Similar to this approach, models like the Differentiable Neural Computer (Graves et al., 2016), Transformer Networks (Vaswani et al., 2017), and Neural Symbolic Machines (Liang et al., 2016) were developed. These models aimed to allow neural networks to handle symbolic data better and perform tasks requiring symbolic manipulation, such as question answering and program synthesis.

Symbolic RNN approaches assume that RNNs learn only surface statistics without symbolically binding information (Greff et al., 2020). This assumption is sometimes contrary to what is found in practice (Vankov & Bowers, 2019), where RNNs trained on certain temporal input-output relationships have demonstrated symbolic generalization to inputs and timescales not experienced during training (Panigrahi & Goyal, 2021).

2.2. Neural Memory Models

Memory, the backbone of intelligence, is pivotal in our capacity to learn, reason, and make decisions (Dasgupta & Gershman, 2021). It functions as a fundamental element of cognition, facilitating information storage, retention, and retrieval over time. In its simplest form, memory serves as a repository of past experiences, allowing us to learn by establishing associations between various information pieces.

A notable milestone in understanding the neural basis of memory was achieved in 1982 when Hopfield (Hopfield, 1982) introduced an energy-based model for associative memories—memories retrieved by querying an input ”associated” with one of the stored memories (Dennis et al., 2015). Initially, Hopfield's model was equipped with binary states and discrete transition rules, limiting its applicability. However, subsequent research broadened this approach to include continuous states and transition rules (Hopfield, 1984; Koiran, 1994; Ramsauer et al., 2020; Krotov & Hopfield, 2020). This generalization started to reveal the link between deep neural networks and memory models. The first investigation of duality explored the relationship between Dense Associative Memory and Multi-Layer Perceptron (MLP) with various activation functions (Ramsauer et al., 2020). Later studies extended this duality to explain the practical computational benefits observed in neural architectures like transformers (Ramsauer et al., 2020). Until now, this relationship has been explored only in neural models without any recurrent behavior.

The traditional associative memory model was recently expanded to an episodic memory model capable of retrieving sequences of stored memories besides just single memory retrieval (Karuvally et al., 2022). This expansion allows memories that previously did not interact in the single memory retrieval context to interact and produce complex temporal behavior (Kleinfeld, 1986; Kleinfeld & Sompolinsky, 1988). A fundamental assumption in memory modeling (in both single and sequence retrieval) is that the network's memories are predetermined and stored in the synapses. This assumption limits the models' applicability to the variable binding problem, which requires the storage of external information provided during inference.

We will demonstrate that the episodic memory model can be utilized to support the binding of external information during inference, and the fixed memory assumption can be lifted to explain how RNNs enable symbolic variable binding.

3. RNN as Episodic Memory

We next show that Elman RNNs can be viewed as a discrete-time analog of the episodic memory model called General Sequential Episodic Memory Model (GSEMM) (Karuvally et al., 2022). One of the crucial components of GSEMM that drives its dynamic behavior is the inter-memory interactions represented by a matrix Φ'. We modify the GSEMM formulation with a pseudoinverse learning rule instead of the Hebbian learning rule for the synapses. This modification allows us to deal with more general (linearly independent vectors) memories than orthogonal vectors (Personnaz et al., 1986). The dynamical equations for our modified GSEMM are given below.

$$\begin{cases} \mathcal{T}_f \dfrac{\mathrm{d}V_f}{\mathrm{d}t} = \sqrt{\alpha_s}\, \Xi\, \sigma_h(V_h) - V_f, \\ \mathcal{T}_h \dfrac{\mathrm{d}V_h}{\mathrm{d}t} = \sqrt{\alpha_s}\, \Xi^{\dagger}\, \sigma_f(V_f) + \alpha_c \Phi'^{\top} \Xi^{\dagger} V_d - V_h, \\ \mathcal{T}_d \dfrac{\mathrm{d}V_d}{\mathrm{d}t} = \sigma_f(V_f) - V_d, \end{cases} \tag{1}$$

The neural state variables of the dynamical system are $V_f \in \mathbb{R}^{N_f \times 1}, V_h \in \mathbb{R}^{N_h \times 1}, V_d \in \mathbb{R}^{N_f \times 1}$. The interactions are represented by $\Xi \in \mathbb{R}^{N_f \times N_h}$ and $\Phi \in \mathbb{R}^{N_h \times N_h}$. $\Xi^{\dagger}$ is the Moore-Penrose pseudoinverse of Ξ. To derive the connection between the continuous time model and discrete updates of RNNs, we discretize the continuous time model using forward Euler approximation under the conditions that $\mathcal{T}_f = 1, \mathcal{T}_h = 0$, and $\mathcal{T}_d = 0$ (See Appendix A for details). The final discrete system is $V_f(t+1) = \Xi\,(I + \Phi'^{\top})\,\Xi^{\dagger}\,\sigma_f(V_f(t))$. The columns of Ξ are the *stored memories* of the model, and $(I + \Phi'^{\top}) = \Phi^{\top}$ is the matrix representing sequential memory interactions between these stored memories. The discrete system we derived is topologically conjugate to the update equations of an Elman RNN under the homeomorphic transformation σ_f if the norm of the matrix is bounded by 1. That is, if $||\Xi\,\Phi^{\top}\,\Xi^{\dagger}|| \leq 1$, we can consider a new state variable $h = \sigma_f(V_f)$ such that

$$h(t) = \sigma_f(\Xi\,\Phi^{\top}\,\Xi^{\dagger} h(t-1))\,. \tag{2}$$

This conjugate system has equations that are equivalent to an Elman RNN hidden state update equation without bias $h(t+1) = \sigma_f(W_{hh} h(t))$. A corollary to the equivalence between the episodic memory model and Elman RNNs is that if we decompose the weight matrix of the RNN in terms of the memories such that $W_{hh} = \Xi\,\Phi^{\top}\,\Xi^{\dagger}$, the RNN computations can be interpreted as the retrieval of episodic memories temporally transitioning according to the rules encoded in Φ. Unfortunately, such a decomposition is not unique as each configuration of the memories Ξ can lead to a different Φ and hence a different interpretation of the dynamics. This can become more complicated if the stored memories can take complex values. However, an invariant property across all these interpretations is the eigenspectrum of Φ. Thus instead of having a single interpretation of the RNN dynamics as episodic memory, we obtain an isospectral continuum of interpretations that depends on the choice of basis (the set of stored memories), the spectral analysis of RNN being the particular case with eigenvectors as the basis. This new perspective of the RNN dynamics enables us to identify mechanisms of variable binding in RNNs by defining an appropriate basis.

4. Mathematical Preliminaries

The core concept of the episodic memory theory is basis change, the appropriate setting of the stored memories. Current notations lack the ability to adequately capture the nuances of basis change. Hence, we introduce abstract algebra notations typically used in theoretical physics literature to formally explain the variable binding mechanisms. We treat a *vector* as an abstract mathematical object invariant to basis changes. Vectors have *vector components* that are associated with the respective basis under consideration. We use Dirac notations to represent vector v as - $|v\rangle = \sum_i v^i\,|e_i\rangle$. Here, the linearly independent collection of vectors $|e_i\rangle$ is the *basis* with respect to which the vector $|v\rangle$ has component $v^i \in \mathbb{R}$. Linear algebra states that a collection of basis vectors $|e_i\rangle$ has an associated collection of *basis covectors* $\langle e^i|$ defined such that $\langle e^i|e_j\rangle = \delta_{ij}$, where δ_{ij} is the Kronecker delta. This allows us to reformulate the vector components in terms of the vector itself as $|v\rangle = \sum_i \langle e^i|v\rangle\,|e_i\rangle$. We use the Einstein summation convention to omit the summation symbols wherever the summation is clear. Therefore, vector $|v\rangle$ written in basis $|e_i\rangle$ is

$$\begin{aligned} |v\rangle &= v^i\,|e_i\rangle \\ &= \langle e^i|v\rangle\,|e_i\rangle\,. \end{aligned} \tag{3}$$

The set of all possible vectors $|v\rangle$ is a *vector space* spanned by the basis vectors $|e_i\rangle$. A *subspace* of this space is a vector space that is spanned by a subset of basis vectors $\{|e'_j\rangle : |e'_j\rangle \subseteq \{|e_i\rangle\}\}$.

The RNN dynamics presented in Equation 2 represented in the new notation is reformulated as:

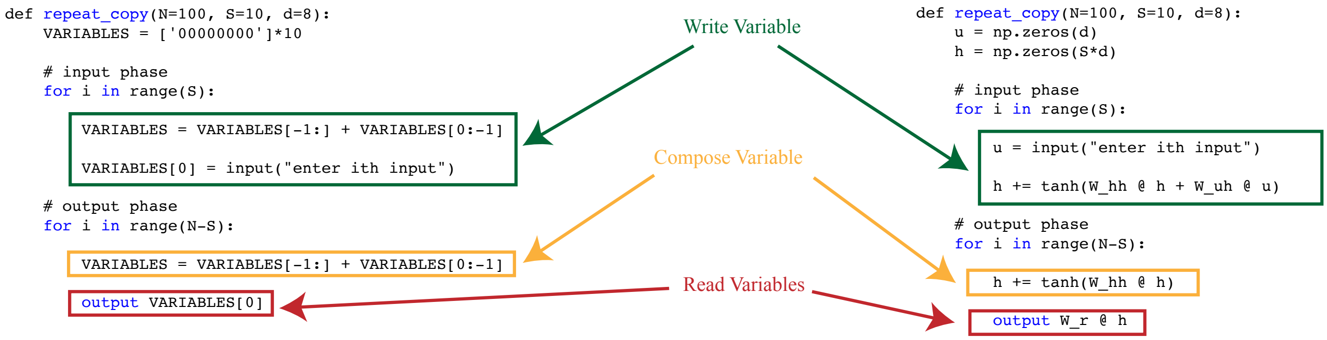
$$\begin{aligned} |h(t)\rangle &= \sigma_f\left(\left(\xi^i_\mu\,\Phi^\mu_\nu\,(\xi^{\dagger})^\nu_j\;\;|e_i\rangle\,\langle e^j|\right)|h(t-1)\rangle\right) \\ &= \sigma_f\left(W_{hh}\,\vec{h}(t-1)\right)\,. \end{aligned} \tag{4}$$

The greek indices iterate over memory space dimensions $\{1, 2, \ldots, N_h\}$, alpha numeric indices iterate over feature dimension indices $\{1, 2, \ldots, N_f\}$. Typically, we use the standard basis in our simulations. For the rest of the paper, the standard basis will be represented by the collection of vectors $|e_i\rangle$ and the covectors $\langle e^i|$. The hidden state at time t in the standard basis is denoted as $|h(t)\rangle = \langle e^j|h(t)\rangle\,|e_i\rangle$. $\langle e^j|h(t)\rangle$ are the *vector components* of $|h(t)\rangle$ we obtain from simulations.

5. Theoretical Model of Variable Binding in RNN

5.1. Problem Setup

We consider RNNs trained on tasks with distinct input and output phases. At each timestep of the input phase, external information is provided *to* the network, and during the output phase, the network needs to utilize this external information to synthesize novel outputs at each time step that are read *from* the network. Formally, the input phase consists of s total timesteps where at each timestep t, a vector of κ dimensions $u(t) = \left(u^1(t), u^2(t), \ldots, u^{\kappa}(t)\right)^{\top}$ is input to the model. We call the vector components $u^i(t)$ the external information that needs to be *stored* in the RNN. After



```
def repeat_copy(N=100, S=10, d=8):
    VARIABLES = ['00000000']*10

    # input phase
    for i in range(S):
        VARIABLES = VARIABLES[-1:] + VARIABLES[0:-1]

        VARIABLES[0] = input("enter ith input")
    # output phase
    for i in range(N-S):
        VARIABLES = VARIABLES[-1:] + VARIABLES[0:-1]
        output VARIABLES[0]
```

```
def repeat_copy(N=100, S=10, d=8):
    u = np.zeros(d)
    h = np.zeros(S*d)

    # input phase
    for i in range(S):
        u = input("enter ith input")

        h += tanh(W_hh @ h + W_uh @ u)
    # output phase
    for i in range(N-S):
        h += tanh(W_hh @ h)
        output W_r @ h
```

Figure 1: **Algorithm-RNN Equivalence**: Pseudocode for an algorithm that solves the repeat copy task and its equivalent in RNNs. The three components of external information processing in RNNs (write, compose, and read) are encoded in the learned weight matrices of an RNN trained to solve the repeat copy task.

the input phase is complete, the zero vector is continually passed as input to the model, so we say the RNN evolves autonomously (without any external input) during the output phase. The Elman RNN is trained to approximate the dynamical system of the given by the following equation.

$$u(t) = f(u(t-1), u(t-2), \dots u(t-s))\,. \tag{5}$$

The following equations govern the Elman RNN.

$$\begin{cases} h(t+1) = \tanh(W_{hh}h(t) + W_{uh}u(t))\,, \\ y(t) = W_r\, h(t+1)\,. \end{cases} \tag{6}$$

where W_{hh}, W_{uh}, W_r are linear operators, $h(t)$ is the hidden state, $u(t)$ is the input, and $y(t)$ is the output. We use a simplifying assumption that W_{hh} has sufficient capacity to represent all the variables required to estimate the dynamical system. We further assume that $h(0)$ is the zero vector.

5.2. Variable Binding Mechanisms

To formalize the basis change suggested by the episodic memory view of RNNs we use Dirac and Einstein notations from abstract algebra. Formally, we write any vector v as $|v\rangle = \langle \epsilon^i | v \rangle |\epsilon_j\rangle = v^i |\epsilon_j\rangle$. $|\epsilon_j\rangle$ is the basis in which the vector has vector components $v^i = \langle \epsilon^i | v \rangle$. $\langle \epsilon^i |$ are the basis *covectors* defined such that $\langle \epsilon^i | \epsilon_j \rangle = \delta_{ij}$, and δ_{ij} is the Dirac delta function.

The Elman ENN is a non-linear dynamical system that is difficult to interpret analytically. However, linearization approaches have shown great promise in their analysis in the neighborhood of fixed points. Thus, we define variable binding mechanisms on a simplified linear dynamical system.

$$|h(t)\rangle = \left(\xi^i_\mu\, \Phi^\mu_\nu\, (\xi^\dagger)^\nu_j \;\; |e_i\rangle \langle e^j|\right) |h(t-1)\rangle \tag{7}$$

Here $|e_i\rangle$ is the standard basis vector. To simplify this system further, we use a new basis $|\psi_\mu\rangle = \xi^i_\mu |e_i\rangle$.

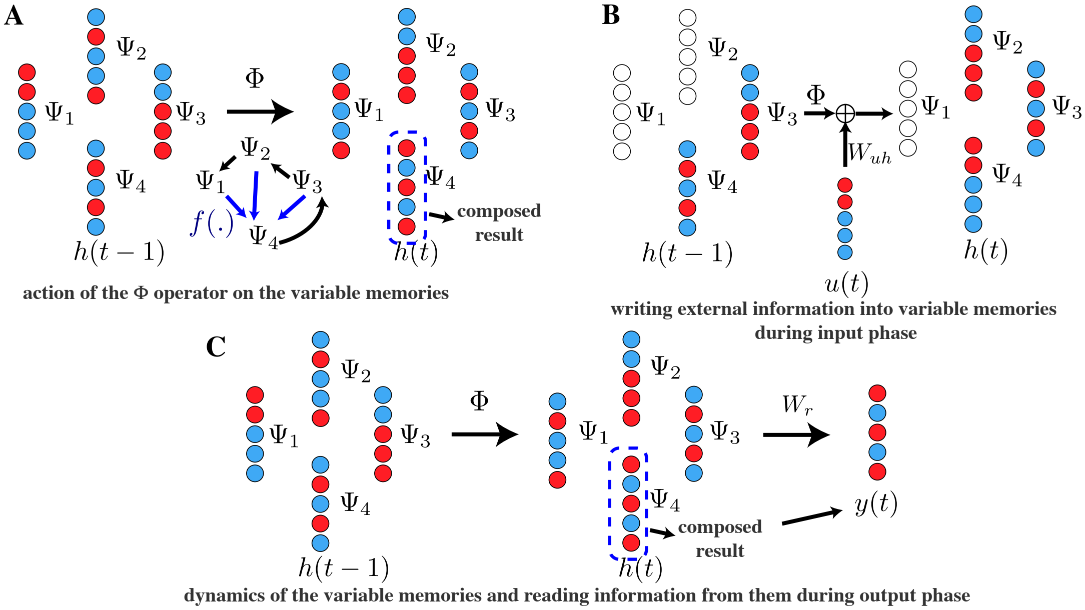
$$|h(t+1)\rangle = (\Phi^\mu_\nu\, |\psi_\mu\rangle \langle \psi^\nu|)\, |h(t)\rangle \tag{8}$$

This new basis allows us to treat the linearized RNN as applying a single linear operator as opposed to three in the original formulation. In the new basis, the hidden state vector is $|h(t)\rangle = h^{\psi_\mu} |\psi_\mu\rangle$. We pose that these components h^{ψ_μ} can be set to any external information for solving the task by appropriately interacting with the hidden state, thus behaving like variables in computation. The collection of vectors $\{\Psi_i\} = \{|\psi_\mu\rangle : \mu \in \{(i-1)\kappa, \dots, i\kappa\}\}$ that defines a subspace where the variable is stored is called the i^{th} *variable memory*. The contents of this variable memory can be extracted from the hidden state vector by applying the linear operator $\Psi_i^* = \sum_{\mu=(i-1)\kappa+1}^{i\kappa} |e_\mu\rangle \langle \psi^\mu|$, on the hidden vector $|h(t)\rangle$. We propose the linear operator Φ has the following general formulation.

$$\Phi = \sum_{\mu=1}^{(N-1)\kappa} |\psi_\mu\rangle \langle \psi^{\mu+\kappa}| + \underbrace{\sum_{\mu=(N-1)\kappa}^{N\kappa} \Phi^\mu_\nu\, |\psi_\mu\rangle \langle \psi^\nu|}_{f(u(t-1),u(t-2),\dots,u(t-N))}\,. \tag{9}$$

The action of the operator on the hidden state is illustrated in Figure 2A. For variable memories with index $i \in \{2, 3, 4, \dots N\}$, the information contents are copied to the variable memory with index $i-1$. The operator then implements the function f defined in Equation 5 and stores the result in the N^{th} subspace. It is easy to see that any *linear* function f of the history can be represented in this framework.

Reading Variables: Once RNN has performed its computation, the computed information needs to be extracted.



action of the Φ operator on the variable memories

writing external information into variable memories during input phase

dynamics of the variable memories and reading information from them during output phase

Figure 2: **Theoretical mechanisms for variable binding in an illustrative task of four variables, each with five dimensions**: **A**. The hidden state at time t has subspaces capable of storing external information in their activities. The colors depict the vector components of the hidden state in the variable memory basis. The linear operator Φ acts on the hidden state such that these activities are copied between variable memories except for Ψ_4, which implements the linear operation f. **B**. The input phase writes external information by adding activity to an empty variable memory. $W_{uh} = \Psi_N$ so that the external information is added to the N^{th} variable memory. **C**. The N^{th} variable contents are read during the output phase using the appropriate linear operator $W_r = \Psi_N^*$.

RNNs have a linear operator W_r, which facilitates the reading of information from $|h(t)\rangle$ at consecutive time steps. We propose that W_r has the following equation.

$$\begin{aligned} W_r &= \Psi_N^* \\ &= \sum_{\mu=(N-1)\kappa+1}^{N\kappa} \left|e_{\mu-(N-1)\kappa}\right\rangle \langle\psi^\mu| \end{aligned} \tag{10}$$

It can be easily seen that the reading operation reads the contents of the N^{th} subspace and outputs them in the standard basis (Figure 2C).

Writing Variables: External information can be written to the hidden state of the RNN by interacting with the variable memories appropriately. Typically, RNNs have a linear operator W_{uh} facilitating the interaction of external information with the RNN. We propose that the linear operator W_{uh} has the following equation.

$$\begin{aligned} W_{uh} &= \Psi_N \\ &= \left|\psi_{(N-1)\kappa+j}\right\rangle \left\langle e^j\right| . \end{aligned} \tag{11}$$

Example: Repeat Copy - Repeat Copy is a task typically used to evaluate the memory storage characteristics of RNNs. The task has deterministic evolution represented by a simple algorithm that stores all input vectors in memory for later retrieval. Repeat copy provides an elementary framework to explore the variable binding mechanisms we proposed in action. We propose the linear operator Φ for the repeat copy task has the following equation.

$$\Phi = \sum_{\mu=1}^{(s-1)\kappa} |\psi_\mu\rangle \left\langle\psi^{\mu+\kappa}\right| + \sum_{\mu=(s-1)\kappa+1}^{s\kappa} |\psi_\mu\rangle \left\langle\psi^{\mu-(s-1)\kappa}\right| \tag{12}$$

This construction of Φ can be imagined as a linear operator that copies the activity of the subspaces cyclically. That is, the content of the i^{th} subspace is copied to the $(i-1)^{\text{th}}$ subspace with the first subspace being copied to the s^{th} subspace (Appendix B). The dynamical evolution of the RNN at time step 1 is, $|h(1)\rangle = \left|\psi_{(s-1)\kappa+j}\right\rangle \left\langle e^j\right| u^i(1) \left|e_i\right\rangle = u^i(1) \left|\psi_{(s-1)\kappa+i}\right\rangle$. At the final step of the input phase, after all the variables are input, $|h(s)\rangle$ is $|h(s)\rangle = \sum_{\mu=1}^{s} u^i(\mu) \left|\psi_{(\mu-1)\kappa+i}\right\rangle$. For t timesteps after s, the general equation for $|h(s+t)\rangle$ is:

$$|h(s+t)\rangle = \sum_{\mu=1}^{s} u^i(\mu) \left|\psi_{[((\mu-t-1 \mod s)+1)\kappa+i]}\right\rangle \tag{13}$$

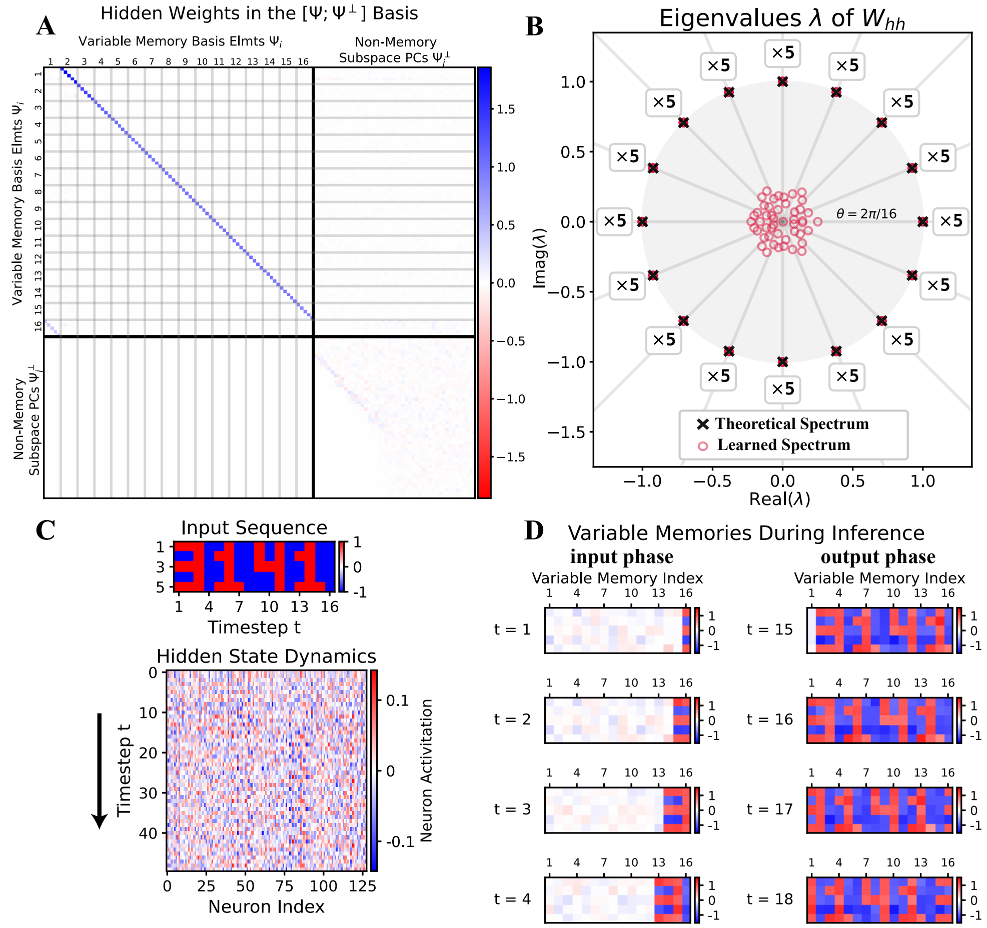


Figure 3: **Experimental Results of Repeat Copy Task with 16 vectors, each of 5 dimensions**: **A**. W_{hh} visualized in the variable memory basis reveals the variable memories and their interactions. **B**. After training, the eigenspectrum of W_{hh} with a magnitude ≥ 1 overlaps with the theoretical Φ. The boxes show the number of eigenvalues in each direction. **C**. During inference, ”3141” is inserted into the network in the form of binary vectors. This input results in the hidden state evolving in the standard basis, as shown. How this hidden state evolution is related to the computations cannot be interpreted easily in this basis. **D**. When projected on the variable memories, the hidden state evolution reveals the contents of the variables over time. In the variable memory basis, it is clear that the input is stored in the hidden state activities and is recursively accessed to produce outputs.

From this equation for the hidden state vector, it can be easily seen that the μ^{th} variable is stored in the $[(\mu - t - 1 \mod s) + 1]^{\text{th}}$ variable memory at time step t. The equivalence between the pseudocode for an algorithm solving repeat copy along with its equivalent neural mechanisms is illustrated in Figure 1.

6. Algorithm for Computing Variable Memories

To compare the learned hidden weights W_{hh} to the interaction matrix Φ of the theoretical model in experimental settings, we view the hidden state space in a new basis consisting of the $\Psi = [\Psi_1; ...; \Psi_s]$ defining the variable memories and an orthogonal basis $\Psi^{\perp}$ for the remaining dimensions of the space. Once this basis is defined, the action of the learned weights on the variable memories can be extracted from the learned W_{hh} using the operator $\Phi_{\text{learned}}^{\top} = \Psi^* W_{hh} \Psi$. Any interaction between the variable memories and the non-memory space will be encoded in the matrices $\Psi^* W_{hh} \Psi^{\perp}$ and $(\Psi^{\perp})^* W_{hh} \Psi$.

We define the variable memories Ψ similar to the linear model. The write weights $\Psi_s = W_{uh}$ define one of the variable memory, and all other subspaces can be found simply by propagating these dimensions *forward* in time: $\Psi_k = \Phi^{s-k} \Psi_s = W_{hh}^{s-k} W_{uh}$. Similarly, the dual of the read weights W_r^* will also define Ψ_s, and propagating this subspace *backwards* in time will yield the other variable

Algorithm 1 Algorithm for computing variable memories of trained RNNs

```
$0 \leq \alpha \leq 1$
$s$ {number of time-steps in the input phase}
$W_{hh}, W_{uh}, W_r$ {learned parameters of the RNN}
$\Psi_s \leftarrow \alpha W_{uh} + (1-\alpha) W_r^*$
for $k \in \{s-1, s-2, \dots 1\}$ do
    $\Psi_k \leftarrow \alpha W_{hh}^{s-k} W_{uh} + (1-\alpha) \left( \left( W_{hh}^\top \right)^k W_r^* \right)$
end for
$\Psi \leftarrow [\Psi_1; \dots; \Psi_s]$
$\Psi^\perp \leftarrow$ PC($\{h\tilde{(t)}\} - \Psi^* \{\tilde{h}(t)\}$) {Principle Components
  of $\tilde{h}$ from simulations} =0
```

memory subspaces: $\Psi_k = (\Phi^{-1})^k \Psi_s = (\Phi^\top)^k \Psi_s = (W_{hh}^\top)^k W_r^*$. The RNN is unlikely to perfectly learn the relationships $W_{uh} = W_r^*$ and $W_{hh}\Psi_k = \Psi_{k-1}$ due to the stochastic training and non-linearity. That being said, we found that the method of power iterating W_{hh} was effective in defining a variable memory space for the nonlinear model. We specifically selected to define the variable memory dimensions Ψ_k by taking a weighted average of the results obtained by propagating W_{uh} forward in time and W_r backwards in time:

$$\Psi_s = \alpha W_{uh} + (1-\alpha) W_r^* \quad , \tag{14}$$

$$\Psi_k = \alpha \, W_{hh}^{s-k} W_{uh} + (1-\alpha)(W_{hh}^\top)^k W_r^*) \text{ for } k < s \tag{15}$$

We also removed the projection of each Ψ_k onto the eigenvectors of W_{hh} whose eigenvalues were less than 1 in magnitude since they do not contribute to the long-term behavior. To obtain the orthogonal basis $\Psi^\perp$ for the rest of the hidden space, we applied principal component analysis (PCA) to hidden state dynamics during inference after removing the projection onto the variable memory space. The algorithm to compute variable memories from trained Elman RNN is summarized in Algorithm 1.

7. Experimental Results

To test the validity of the theoretical models, we trained Elman RNNs (c.f. Equation 6) via gradient descent to perform the repeat copy and compose copy tasks, and compared the structure of the learned weights to that of the theoretical models.

Repeat Copy: We present results for repeat copy with sequence length $s = 16$ and $\kappa = 5$ ($s \cdot \kappa = 80$ total bits), but similar results are obtained for all networks we trained, regardless of the sequence length, vector dimension κ, and experiment seed as long as W_{hh} has sufficient dimensions. First, we show the structure of the learned weights when viewed in the basis $[\Psi; \Psi^\perp]$ in Figure 3A. We observe that the variable memory interactions defined by the upper left block of the matrix form the theoretical structure. The other blocks of the weight matrix show much smaller interactions, indicating that the variable memories interact weakly with the orthogonal subspace. By construction, $\Psi^\perp$ is orthogonal to the eigenvectors of W_{hh} with eigenvalue magnitude ≥ 1, so the interactions of the non-memory subspace with itself decays to zero as we iterate this matrix. This computation structure is also reflected in the eigenspectrum of the learned W_{hh} showing agreement with the theoretical values (Figure 3B). Figure 3D depict the projection of the hidden state sequence onto the variable memories during inference for an example input resembling the text “3141” (Figure 3C). At timestep 1, a majority of the variable memories store values close to 0 (white, as indicated by the colorbar), and subspace s stores the first input vector. This vector is moved to the next variable memory subspace on the subsequent timestep, and the next input vector is copied to the Ψ_s subspace. This process continues iteratively until $t = s$, with all loaded variables being shifted over by one subspace and the following input being loaded into the Ψ_s subspace. At $t = s$, all variable memories have been assigned values. Afterward, the network autonomously shifts these values to one subspace per timestep to periodically recall the input sequence by reading from $W_s = W_r^*$.

8. Discussion

We have introduced the Episodic Memory Theory of Recurrent Neural Networks (RNNs), portraying these networks as dynamic systems executing episodic memory retrieval. We presented the notion of ”variable memory,” a linear subspace that can connect and recursively compute data symbolically. Our methodology circumvents the obstacles posed by conventional techniques in comprehending RNNs, specifically their obscurity as ’black boxes’ and the intricacy of spectral analysis in tasks involving multiple dimensions.

The episodic memory theory and the theoretical model of variable binding challenge the basic presumptions made in the two research directions delineated in Section 2. Addressing the ”variable binding problem,” we demonstrated how RNNs could symbolically bind and synthesize data using variable memories. As for the fixed memory assumption in memory models, we demonstrated how it could be bypassed by interacting suitably with the memory model.

References

Buchanan, B. G. and Shortliffe, E. H. Rule based expert systems: The mycin experiments of the stanford heuristic programming project (the addison-wesley series in artificial intelligence). 1984.

Dasgupta, I. and Gershman, S. J. Memory as a computational resource. *Trends in Cognitive Sciences*, 25:240–

251, 2021.

Dennis, N. A., Turney, I. C., Webb, C. E., and Overman, A. A. The effects of item familiarity on the neural correlates of successful associative memory encoding. *Cognitive, Affective, & Behavioral Neuroscience*, 15:889–900, 2015.

Graves, A., Wayne, G., and Danihelka, I. Neural turing machines. *ArXiv*, abs/1410.5401, 2014.

Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwinska, A., Colmenarejo, S. G., Grefenstette, E., Ramalho, T., Agapiou, J. P., Badia, A. P., Hermann, K. M., Zwols, Y., Ostrovski, G., Cain, A., King, H., Summerfield, C., Blunsom, P., Kavukcuoglu, K., and Hassabis, D. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538:471–476, 2016.

Greff, K., van Steenkiste, S., and Schmidhuber, J. On the binding problem in artificial neural networks. *ArXiv*, abs/2012.05208, 2020.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79 8:2554–8, 1982.

Hopfield, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 81 10:3088–92, 1984.

Johnson-Laird, P. N. Mental models and human reasoning. *Proceedings of the National Academy of Sciences*, 107: 18243 – 18250, 2010.

Karuvally, A., Sejnowski, T. J., and Siegelmann, H. T. Energy-based general sequential episodic memory networks at the adiabatic limit. *ArXiv*, abs/2212.05563, 2022.

Kautz, H. A. The third ai summer: Aaai robert s. engelmore memorial lecture. *AI Mag.*, 43:93–104, 2022.

Kleinfeld, D. Sequential state generation by model neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 83 24:9469–73, 1986.

Kleinfeld, D. and Sompolinsky, H. Associative neural network model for the generation of temporal patterns. theory and application to central pattern generators. *Biophysical journal*, 54 6:1039–51, 1988.

Koiran, P. Dynamics of discrete time, continuous state hopfield networks. *Neural Computation*, 6:459–468, 1994.

Krotov, D. and Hopfield, J. J. Large associative memory problem in neurobiology and machine learning. *ArXiv*, abs/2008.06996, 2020.

Liang, C., Berant, J., Le, Q. V., Forbus, K. D., and Lao, N. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. *ArXiv*, abs/1612.01197, 2016.

Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., and Lederberg, J. Applications of artificial intelligence for organic chemistry: The dendral project. 1980.

Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., and Lederberg, J. Dendral: A case study of the first expert system for scientific hypothesis formation. *Artif. Intell.*, 61:209–261, 1993.

Martens, J. and Sutskever, I. Learning recurrent neural networks with hessian-free optimization. In *International Conference on Machine Learning*, 2011.

Newell, A. and Simon, H. A. The logic theory machine-a complex information processing system. *IRE Trans. Inf. Theory*, 2:61–79, 1956.

Newell, A. and Simon, H. A. Gps, a program that simulates human thought. 1995.

Panigrahi, A. and Goyal, N. Learning and generalization in rnns. In *Neural Information Processing Systems*, 2021.

Personnaz, L., Guyon, I. I., and Dreyfus, G. Collective computational properties of neural networks: New learning mechanisms. *Phys Rev A Gen Phys*, 34(5):4217–4228, November 1986.

Ramsauer, H., Schafl, B., Lehner, J., Seidl, P., Widrich, M., Gruber, L., Holzleitner, M., Pavlovi'c, M., Sandve, G. K., Greiff, V., Kreil, D. P., Kopp, M., Klambauer, G., Brandstetter, J., and Hochreiter, S. Hopfield networks is all you need. *ArXiv*, abs/2008.02217, 2020.

Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

Russell, S. J. and Norvig, P. Artificial intelligence: A modern approach. 1995.

Shortliffe, E. H. and Buchanan, B. G. A model of inexact reasoning in medicine. *Bellman Prize in Mathematical Biosciences*, 23:259–275, 1990.

Strogatz, S. H. Nonlinear dynamics and chaos: With applications to physics, biology, chemistry and engineering. *Physics Today*, 48:93–94, 1994.

Sussillo, D. and Abbott, L. F. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63: 544–557, 2009.

Sussillo, D. and Barak, O. Opening the black box: Low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Computation*, 25:626–649, 2013.

Turing, A. On computable numbers, with an application to the entscheidungsproblem. *Proc. London Math. Soc.*, s2-42:230–265, 2021.

Vankov, I. I. and Bowers, J. S. Training neural networks to encode symbols enables combinatorial generalization. *Philosophical Transactions of the Royal Society B*, 375, 2019.

Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NIPS*, 2017.

Whitehead, A. N. Symbolism: its meaning and effect.

A. RNN - Episodic Memory Equivalence

From a given time t, the update equations are given as

$$\begin{cases} \mathcal{T}_f(V_f(t+1) - V_f(t)) = & \Xi\,\sigma_h(V_h(t)) - V_f(t)\,, \\ V_h(t) = & \Xi^\top\,\sigma_f(V_f(t)) + \Phi^\top\Xi^\top V_d(t)\,, \\ V_d(t) = & \sigma_f(V_f(t))\,. \end{cases} \tag{16}$$

$$\begin{cases} \mathcal{T}_f(V_f(t+1) - V_f(t)) = & \Xi\,\sigma_h(V_h) - V_f(t)\,, \\ V_h(t) = & \Xi^\top\,\sigma_f(V_f(t)) + \Phi^\top\Xi^\top\sigma_f(V_f)\,, \end{cases} \tag{17}$$

$$\begin{cases} \mathcal{T}_f(V_f(t+1) - V_f(t)) = & \Xi\,V_h - V_f(t)\,, \\ V_h(t) = & (I + \Phi^\top)\Xi^\top\sigma_f(V_f)\,, \end{cases} \tag{18}$$

$$\mathcal{T}_f(V_f(t+1) - V_f(t)) = \Xi\,(I + \Phi^\top)\Xi^\top\sigma_f(V_f) - V_f(t) \tag{19}$$

Final discrete upate equation

$$V_f(t+1) = \Xi\,(I + \Phi^\top)\Xi^\top\sigma_f(V_f) \tag{20}$$

Restrict the norm of matrix $||\Xi\,(I + \Phi^\top)\Xi^\top|| \leq 1$.
This allows us to consider the transformation $V_f' = \sigma_f(V_f)$, so for invertible σ_f,

$$\sigma_f^{-1}(V_f'(t+1)) = \Xi\,(I + \Phi^\top)\Xi^\top V_f' \tag{21}$$

$$\sigma_f^{-1}(V_f'(t+1)) = \Xi\,(I + \Phi^\top)\Xi^\top V_f' \tag{22}$$

$$V_f'(t+1) = \sigma_f(\Xi\,(I + \Phi^\top)\Xi^\top V_f') \tag{23}$$

this is a general update equation for an RNN without bias. The physical interpretation of this equation is that the columns of Ξ stores the individual *memories* of the system and the linear operator $(I + \Phi)$ is the temporal interaction between the stored *memories*. In the memory modeling literature, it is typical to consider memories as a fixed collection instead of a variable collection that shares a common interaction behavior. We will show how in the next sections how the dynamics as a result of fixed collection can be used to store variable information.

A.1. Topological Conjugacy with RNNs

Proof that dynamical systems governed by Equations 20 and 23 are topological conjugates.

Consider $f(x) = \Xi\,(I + \Phi^\top)\Xi^\top\sigma_f(x)$ for Equation 20 and $g(x) = \sigma_f(\Xi\,(I + \Phi^\top)\Xi^\top x)$ for Equation 23. Consider a homeomorphism $h(y) = \sigma_f(y)$ on g. Then,

$$\begin{aligned} (h^{-1} \circ g \circ h)(x) &= \sigma_f^{-1}(\sigma_f(\Xi\,(I + \Phi^\top)\Xi^\top\sigma_f(x))) \\ &= \Xi\,(I + \Phi^\top)\Xi^\top\sigma_f(x) \\ &= f(x) \end{aligned} \tag{24}$$

So, for the homeomorphism h on g, we get that $h^{-1} \circ g \circ h = f$ proving that f and g are topological conjugates. Therefore all dynamical properties of f and g are shared.

B. Repeat Copy: Mathematical Theory

Repeat Copy is a task typically used to evaluate the memory storage characteristics of RNNs since the task has a deterministic evolution represented by a simple algorithm that stores all input vectors in memory for later retrieval. Although elementary, repeat copy provides a simple framework to imagine the variable binding mechanisms we theorized in action. For the repeat copy task, the linear operators of the RNN has the following equations.

$$\begin{cases} \Phi = \sum_{\mu=1}^{(s-1)\kappa} \left|\psi_\mu\right\rangle \left\langle\psi^{\mu+\kappa}\right| + \sum_{\mu=(s-1)\kappa+1}^{s\kappa} \left|\psi_\mu\right\rangle \left\langle\psi^{\mu-(s-1)\kappa}\right| \\ W_{uh} = \Psi_s \\ W_r = \Psi_s^* \end{cases} \tag{25}$$

This ϕ can be imagined as copying the contents of the subspaces in a cyclic fashion. That is, the content of the i^{th} subspace goes to $(i-1)^{\text{th}}$ subspace with the first subspace being copied to the N^{th} subspace. The dynamical evolution of the RNN is represented at the time step 1 as,

$$|h(1)\rangle = \left|\psi_{(s-1)\kappa+j}\right\rangle \left\langle e^j\right| u^i(1) \left|e_i\right\rangle \tag{26}$$

$$|h(1)\rangle = u^i(1) \left|\psi_{(s-1)\kappa+j}\right\rangle \left\langle e^j \middle| e_i\right\rangle \tag{27}$$

$$|h(1)\rangle = u^i(1) \left|\psi_{(s-1)\kappa+j}\right\rangle \delta_{ij} \tag{28}$$

Kronecker delta index cancellation

$$|h(1)\rangle = u^i(1) \left|\psi_{(s-1)\kappa+i}\right\rangle \tag{29}$$

At time step 2,

$$|h(2)\rangle = u^i(1)\, \Phi \, \left|\psi_{(s-1)\kappa+i}\right\rangle + u^i(2) \, \left|\psi_{(s-1)\kappa+i}\right\rangle \tag{30}$$

Expanding Φ

$$|h(2)\rangle = u^i(1) \left(\sum_{\mu=1}^{(s-1)\kappa} \left|\psi_\mu\right\rangle \left\langle\psi^{\mu+\kappa}\right| + \sum_{\mu=(s-1)\kappa+1}^{s\kappa} \left|\psi_\mu\right\rangle \left\langle\psi^{\mu-(s-1)\kappa}\right| \right) \left|\psi_{(s-1)\kappa+i}\right\rangle + u^i(2) \left|\psi_{(s-1)\kappa+i}\right\rangle \tag{31}$$

$$|h(2)\rangle = u^i(1) \, \left|\psi_{(s-2)\kappa+i}\right\rangle + u^i(2) \, \left|\psi_{(s-1)\kappa+i}\right\rangle \tag{32}$$

At the final step of the input phase when $t = s$, $|h(s)\rangle$ is defined as:

$$|h(s)\rangle = \sum_{\mu=1}^{s} u^i(\mu) \, \left|\psi_{(\mu-1)\kappa+i}\right\rangle \tag{33}$$

For t timesteps after s, the general equation for $|h(s+t)\rangle$ is:

$$|h(s+t)\rangle = \sum_{\mu=1}^{s} u^i(\mu) \, \left|\psi_{[((\mu-t-1 \mod s)+1)\kappa+i]}\right\rangle \tag{34}$$

From this equation for the hidden state vector, it can be easily seen that the μ^{th} variable is stored in the $[(\mu-t-1 \mod s)+1]^{\text{th}}$ subspace at time step t. The readout weights $W_r = \Psi_s^*$ reads out the contents of the s^{th} subspace.

C. Non-linear RNN

The linear RNNs we discussed are powerful in terms of the content of variables that can be stored and reliably retrieved. The variable contents, u^i, can be any real number and this information can be reliably retrieved in the end using the appropriate readout weights. However, learning such a system is difficult using gradient descent procedures. To see this, setting the components of Φ to anything other than unity might result in dynamics that is eventually converging or diverging resulting in a loss of information in these variables. Additionally, linear systems are not used in the practical design of RNNs. The main difference is now the presence of the nonlinearity. In this case, our theory can still be used. To illustrate this, consider a general RNN evolving according to $h(t+1) = g(W_{hh}h(t) + b)$ where b is a bias term. Suppose $h(t) = h^*$ is a fixed point of the system. We can then linearize the system around the fixed point to obtain the linearized dynamics in a small region around the fixed point.

$$h(t+1) - h^* = \mathcal{J}(g)|_{h^*} \, W_{hh} \, (h(t+1) - h^*) + O((h(t+1) - h^*)^2) \tag{35}$$

where $\mathcal{J}$ is the jacobian of the activation function g. If the RNN had an additional input, this can also be incorporated into the linearized system by treating the external input as a control variable

$$h(t+1) - h^* = \mathcal{J}(g)|_{h^*} \, W_{hh} \, (h(t) - h^*) + \mathcal{J}(g)|_{h^*} \, W_{uh}u(t) \tag{36}$$

Substituting $h(t) - h^* = h'(t)$

$$h'(t+1) = \mathcal{J}(g)|_{h^*} \, W_{hh} \, h'(t) + \mathcal{J}(g)|_{h^*} \, W_{uh}u(t) \tag{37}$$

which is exactly the linear system which we studied where instead of $W_{hh} = \Xi\Phi\Xi^\dagger$, we have $J(g)|_{h^*}W_hh = \Xi\Phi\Xi^\dagger$. With this result, we will analyse Elman RNN models that have the general update equations $h(t+1) = \tanh(W_{hh}h(t) + W_{uh}u(t) + b)$.