Research Proposal
Three Bodies Recurrent Neural Network

Gilad Altshuler
supervised by Omri Barak

July 11, 2024

Abstract

In recent years, there have been several examples of artificial neural networks that have been trained to perform simple cognitive tasks. Remarkably, the internal activity of the artificial networks resembled aspects of the biological ones, despite the fact that the model neurons are very far removed from their biological counterparts. This work sheds some light on these unique resemblance patterns by devising a temporal network that is fundamentally different from the classic recurrent neural networks (RNN) and by thorough investigation of similarities and differences lies between them. The proposed network model, Three Bodies Recurrent Neural Network (TBRNN)[1] implements the idea that some natural systems, such as protein-enzyme networks, develop in time by three-bodies (or even more) interaction properties, as opposed to the classic modeled RNN, which develops in time by two-bodies interactions manner. Though they are very different, this work reveals that the aforementioned network maintains low-rank patterns and universal approximation properties. To this end, this network has been trained to perform two simple computational neuroscience cognitive tasks and a new task - the protein interaction network (PIN) task. This work analyzes the theoretical and practical properties of this kind of network, with reference to the performance on these tasks.

[1] Github code implementation is in `https://github.com/gilad-altshuler/Three_Bodies_RNN`

1 Introduction

Some may say that AI in general and artificial neurons in particular meant, at first at least, to mimic their natural counterparts, which in the case of the latter are the real-life neurons. To that mission, a nonlinear function applied to each neuron where on its turn linearly combines the outputs from all the previous neuron layer. Among the myriad neural networks that can be defined, recurrent neural networks (RNN) are typically used as a tool to model dynamic networks, and in the context of neuroscience, to explain neurobiological phenomena [1]. Unfolded in time for $t = 1, 2, ...$ The vanilla RNN uses the following formula:

$$h_t = \phi(W \cdot h_{t-1} + I \cdot u_t) \tag{1}$$

Where h is the hidden state, developing in time, u is the network input, W is the connectivity matrix, I is the input weight matrix and ϕ is some nonlinear function. Commonly used, there are two forms to represent neurobiological data as an ordinary differential equation, the fire rate or the voltage notations. The fire rate notation is:

$$\frac{\tau dr}{dt} = -r + \phi(Wr + I) \tag{2}$$

While the voltage form is:

$$\frac{\tau dv}{dt} = -v + W\phi(v) + I \tag{3}$$

So it is natural to discretize those notations onto RNN update rule to learn the dynamics. Although different, the two forms are somewhat equivalent to each other [10], with the intuitive map $v = Wr + I$.

Recently, many attempts have been made to make network connectivity more interpretable and comprehensible [2, 4, 18]. While full-rank RNNs are hard to explain, it is claimed that many RNNs are, in fact, lying in a low-rank connectivity, which are easier and more intuitive to analyze, in order to understand the prevailing dynamics. The low-rank recurrent neural network (lrRNN) group is a subset of the RNN group. The assumption behind this group, is that the connectivity matrix is low-ranked, i.e., say that is rank $R \ll N$, then exists $\mathbf{m}, \mathbf{n} \in \mathbb{R}^{R \times N}$ such that:

$$W = \sum_{r=1}^{R} \mathbf{m}^{(r)} \cdot \mathbf{n}^{(r)^T}$$

One way to infer the low ranked pattern, suggested by [18], is to train a lrRNN to learn the trajectories of a full rank RNN. Their work showed that this method reconstructs, by reverse engineering, the full-rank RNN better than the SVD truncation method. In addition, their work showed that W includes some unimportant information to dynamics, thus defined W_{eff} to be:

$$W_{eff} = \frac{1}{N}\sum_{r=1}^{R} \mathbf{m}^{(r)} \cdot \mathbf{n}_{||}^{(r)^T}$$

where $\mathbf{n}_{||}^{(r)}$ vectors are the orthogonal projection on the linear space spanned by $Span\left\{\mathbf{m}^{(r)}, \mathbf{I}^{(r)}\right\}$.

The above (1) network induce in fact two bodies interactions in time, i.e. between the "source" neuron layer, h_{t-1} and the updated "destination" neuron layer h_t. Some of the biological network systems, however, do have complex functional activities, yet do not comply to this two-bodies rules. An example is protein interaction networks (PINs). PIN are protein networks that combine three types of protein bodies: reactants, enzymes and products. The enzyme group is a subset of the protein group and affects the kinetics of the reaction, although they do not change during the reaction. The basic theory of enzymes is called the Michaelis-Menten (MM) theory of enzyme-catalyzed reaction. MM formulate those reactions such that if S, E, P is the substrate, enzyme, and product, respectively, then the following reaction generally holds [15]:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} E : S \overset{k_2}{\rightarrow} E + P \tag{4}$$

Where $E : S$ is enzyme-substrate complex, k_1 and k_{-1} are the equilibrium rates and k_2 is the fast reaction rate. Equation 4 above, provides some intuition and incentive to diverge from the classic RNN two-bodies update rule into a new model that refers to this three body interactions structure. There have been several attempts to devise second order RNN in the past [9, 16]. Those attempts referred as second order by the general idea of also multiplying the hidden state by the input, at each time t, and not just them as stand-alone parameters. For example, [9] proposed the formula

$$h_t = \phi(\mathbf{A}(\mathbf{B}x_t * \mathbf{C}h_{t-1}) + \mathbf{D}x_t + \mathbf{E}h_{t-1} + \mathbf{f})$$

Which does not implement quadratic connection with h_{t-1} to itself.

2 Methods

2.1 TBRNN model

Referring to the aforementioned equation 4, we define Three Bodies Network RNN model (TBRNN). Given that the hidden state size is N, then for all $i \in [N]$, the element $h_t^{(i)}$ in the hidden state vector $\mathbf{h_t}$ is:

$$h_t^{(i)} = \phi(h_{t-1}^{\mathbf{T}} \cdot W_i \cdot h_{t-1} + I_i \cdot u_t) \tag{5}$$

such that $W_i \in \mathbb{R}^{N \times N}$ is the connectivity matrix with refer to element i. The general formulation, by concatenating $\mathcal{W} = [W_1; \ldots; W_N] \in \mathbb{R}^{N \times N \times N}$ is:

$$h_t = \phi(h_{t-1}^{\mathbf{T}} \otimes \mathcal{W} \otimes h_{t-1} + I \cdot u_t) \tag{6}$$

This is a recursive formula that semi-generalizes RNN equation 1 to second order form, yet does not allow first order computations with h, in purpose of total separation from first-order solutions. In addition, it is emphasized the three bodies interactions between h to itself in the TBRNN model. To our current knowledge, this is the first attempt to analyze such network. When ϕ is invertible (e.g. sigmoid, tanh), then the form of equation 6 equivalent to the form:

$$h_t = \phi^{\mathbf{T}}(h_{t-1}) \otimes \mathcal{W} \otimes \phi(h_{t-1}) + I \cdot u_t$$

Which induce the following dynamics:

$$\tau \frac{dx}{dt} = -x + \phi^T(x) \otimes \mathcal{W} \otimes \phi(x) + I \cdot u(t) \tag{7}$$

2.2 lr-TBRNN model

In this work, we define a subgroup of TBRNN, namely low-rank three bodies networks (lrTBRNN) which indicate low rank three bodies dynamics. The contributions for this low rank inference are two:

1. Rank inference of some task, that assumed to be with low-rank properties.

2. Rank inference ability of some full rank TBRNN, that assumed to be with low-rank properties, that eases the black-box comprehension.

3. To compare the fitness of TBRNN with RNN to perform some task (see 2.3 subsection) with vanilla RNN.

Similarly to the work of [8], we define lrTBRNN as a TBRNN with some low-rank connectivity tensor $\mathcal{W}$, such that there exists $\mathbf{l}, \mathbf{m}, \mathbf{n} \in \mathbb{R}^{R \times N}$ that makes up it:

$$\mathcal{W} = \frac{1}{N^2} \cdot \sum_{r=1}^{R} \mathbf{l}^{(r)} \otimes \mathbf{m}^{(r)} \otimes \mathbf{n}^{(r)}$$

Low rank inference

To address the rank inference ability of some full rank TBRNN issue, we train student lrTBRNN trajectories to be the same as teacher TBRNN trajectories on the same tasks (see Subsection 2.3. Formally, the loss is the same as [18] defined:

$$\mathcal{L} = \sum_{c=1}^{C} \sum_{i=1}^{N} \sum_{t=1}^{T} \left(\phi(x_i^{(c)}(t)) - \phi(\hat{x}_i^{(c)}(t)) \right)^2 \tag{8}$$

Where $x_i^{(c)}(t)$ represents the target teacher model trajectory in condition c, for neuron i and timestep t and $\hat{x}_i^{(c)}(t)$ the corresponding trajectory produced by the low rank model.

2.3 Tasks

To assess the capability of the new network, we used two basic tasks that have been previously studied in the neuroscience literature [7] and one new task based on our protein network analysis (PIN task).

2.3.1 K Bit Flip Flop

Following [7], TBRNNs as well as RNNs, were provided K inputs taking discrete values in $\{-1, 0, +1\}$. The RNN has K outputs, each of which is trained to remember the last non-zero input on its corresponding input. Here we set $K = 3$, so e.g. output 2 remembers the last non-zero state of input 2 ($+1$ or -1), but ignores inputs 1 and 3. We set the number of time steps, T, to 100, and the flip probability (the probability of any input flipping on a particular time step) to 5%.

2.3.2 Frequency-cued sine wave

Following [7], TBRNNs as well as the RNNs received a static input, $x \sim Uniform(0, 1)$, and were trained to produce a unit-amplitude sine wave, $\sin(2\pi\omega t)$, whose frequency is proportional to the input: $\omega = 0.04x + 0.01$. We set $T = 500$ and $dt = 0.01$ (5 simulated seconds total).

2.3.3 Protein Interaction Network

To describe the task definition, first we give a short formalities:
Following the MM equation described in 4, consider network of proteins such that any protein permitted to be at any role of one of the 3 among substrate,enzyme,product, in any reaction exist within the network. Moreover, we permit enzyme-substrate complexes to be catalyzed (fast reaction) to any product protein (plus the enzyme). Here we give some notations:

- Let N be the number of proteins in the network.
- $\forall i \in \{1, \ldots, N\}$, denote $[P_i]$ as the concentration of protein i.
- $\forall i, j \in \{1, \ldots, N\}$, denote $[P_i : P_j]$ as the concentration of the complex $P_i : P_j$ where P_i is the enzyme and P_j is the substrate.
- $\forall i, j \in \{1, \ldots, N\}$, denote:
  - $K_{ij}^{(1)}$ as the rate for P_i and P_j to become complex $P_i : P_j$.
  - $K_{ij}^{(-1)}$ the rate of the complex reversed reaction $P_i : P_j \to P_i + P_j$.
  - $\forall k \in \{1, \ldots, N\}\,; K_{ijk}^{(2)}$ the catalyzed rate of $P_i : P_j \to P_i + P_k$

The work of [3] implies that the following is approximately true:

1. $\forall i \in [N]$:

$$\frac{d[P_i]}{dt} = \left(\sum_{j=1}^{N} K_{ij}^{(-1)} \cdot [P_i : P_j] + K_{ji}^{(-1)} \cdot [P_j : P_i]\right) - \left(\sum_{j=1}^{N} \left(K_{ij}^{(1)} + K_{ji}^{(1)}\right) \cdot [P_i] \cdot [P_j]\right) + \left(\sum_{\substack{1 \le j,k \le N; \\ j \ne i}} K_{jki}^{(2)} \cdot [P_j : P_k]\right) + \left(\sum_{1 \le j,k \le N} K_{ijk}^{(2)} \cdot [P_i : P_j]\right)$$

2. $\forall i, j \in [N]$:

$$\frac{d[P_i : P_j]}{dt} = K_{ij}^{(1)} \cdot [P_i] \cdot [P_j] - K_{ij}^{(-1)} \cdot [P_i : P_j] - \sum_{k=1}^{N} K_{ijk}^{(2)} \cdot [P_i : P_j]$$

Note that even at first glance, those first order PDEs have three bodies interactions properties. To generate PIN data from those PDEs, we follow these steps:

1. Set values for $K^{(1)}, K^{(-1)}, K^{(2)}$.
2. Define the concentration of the initial state of the proteins ($[P_i]$) and complexes ($[P_i : P_j]$) state concentration.
3. Use the Euler method with $\delta t \leq 0.01$ to generate the PIN data from the above PDEs. The appropriate total generation time T depends on the data.

3 Experiments

3.1 Model TBRNN basic capabilities

It is not immediately that three way based dynamical network, would be capable to learn, even simple, cognitive tasks. To assess the ability TBRNN model to perform those kind of tasks, we conducted series of experiments on K-bit Flip Flop and on frequency-cued sine wave tasks. Each task has been learned with different hyperparameters.

K-bit Flip Flop

TBRNN models were trained to learn the K-bit Flip-Flop with alternating parameters, $K \in \{1, 2, 3, 4\}$, activation function $\phi \in \{Relu, Tanh, Sigmoid\}$, hidden size $N \in \{30, 60, 90, 120\}$, batch size 128, Adam optimization algorithm with learning rate $lr \in \{10^{-3}, 10^{-4}, 10^{-5}\}$, and **MSE** loss.

Frequency-cued sine wave

TBRNN models were trained to learn the frequency-cued sine wave with

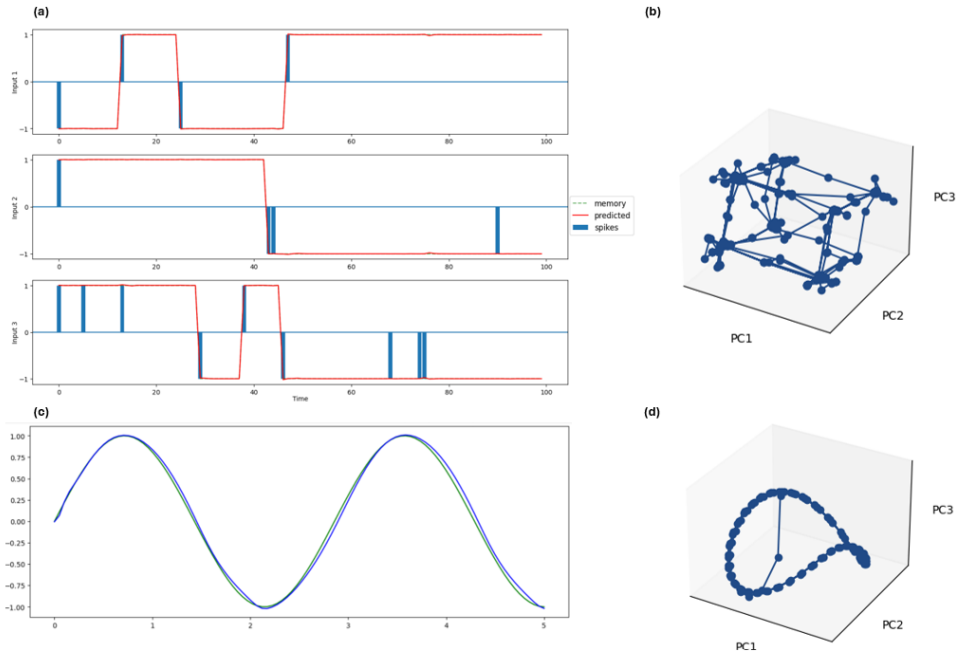

Figure 1: TBRNN train on 3-bit Flip Flop and Frequency-cued sine wave tasks. **(a)** 3-bit Flip Flop performance on random example, demonstrates the capability of TBRNN to learn basic cognitive task, even with only three-way connections. **(b)** Neurons PCA analysis with 3PCs, with 0.965 explained variance. Fix point analysis leads to the same result as [17]. **(c-d)** Same as **(a-b)** for sine wave task.

parameter set from , activation function $\phi \in \{Relu, Tanh, Sigmoid\}$, hidden size $N \in \{30, 60, 90, 120\}$, batch size 128, Adam optimization algorithm with learning rate $lr \in \{10^{-3}, 10^{-4}, 10^{-5}\}$, and **MSE** loss. Figure 1 shows the results for both tasks, both reached negligible loss and high accuracy on the test set, and both appeared with the same trajectories pattern as regular RNN, when plotting the first 3 PCs in principal component analysis.

3.2 Model lrTBRNN validation

Increasing ranks ($r \in \{1 \dots 6\}$) lrTBRNN where trained to mimic trained full rank TBRNN trajectories, with the loss presented in equation (8) on the K-bit flip flop and sine prediction tasks. The threshold used to determine the rank of the corresponding lrTBRNN is the formula:

$$\min_r \left\{ r | Accuracy^{(task)}(\text{rank-r lrTBRNN}) \geq 95\% \right\}$$

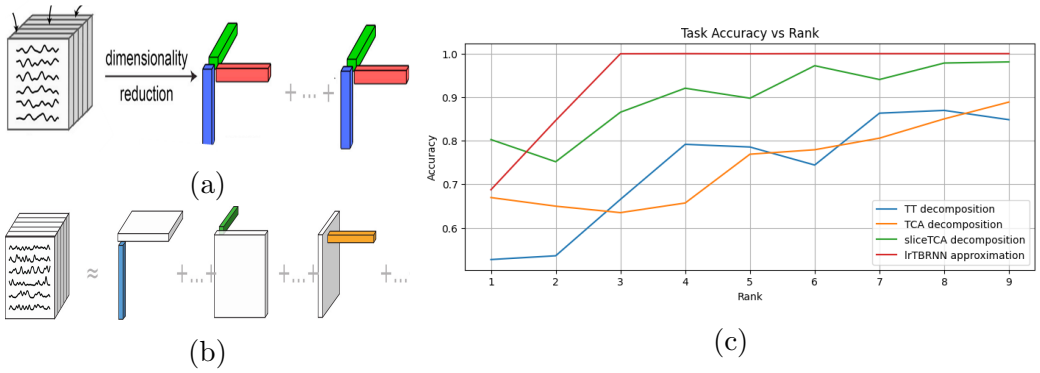

Figure 2: Connectivity decomposition of tensor $\mathcal{W}$, referred to the 3-bit flip flop task. **(a)** Demonstrates tensor component analysis, also known as PARAFAC analysis, $\mathcal{W}$ is sum of 1-rank tensors. **(b)** spliceTCA pipeline, taken from [12], does not impose a low rank on the other two dimensions. **(c)** Comparison between lrTBRNN method to other truncation methods, shows that ours vastly outperforms the exist methods.

As in the figure we found that rank-3 lrTBRNN is enough to exhibit the full-rank trajectories, while when fitting lrRNN, at least rank 4 is needed, shows that the trajectories lies in multilinear connection space. In addition, we compared lrTBRNN method with truncating the connectivaty tensor $\mathcal{W}$ in three known tensor decomposition methods - Tensor Component Analysis (TCA), Tensor Train (TT) [11] and spliceTCA [12]. Figure 2c show that our method vastly outperforms the other decomposition method to reverse engineer the full-rank TBRNN on 3-bit flip flop task.

4 Work plan

We divide the work plan into three consecutive stages:

1. **Low-rank tensor inference**
   The authors in [14] demonstrated that in certain neuroscience tasks, connectivity changes are low-rank even when the initial connectivity is full-rank. Specifically, ΔW is a low-rank matrix, where W_0 represents the initial full-rank random connectivity matrix, and W is the learned connectivity matrix obtained through a learning algorithm, such that $W = W_0 + \Delta W$. Our goal is to conduct a fair comparison between our low-rank inference method and the other methods illustrated in Figure 2c, Notably, our method is the only learning-based approach, whereas

the others are static methods. To achieve this comparison, instead of truncating the entire $\mathcal{W}$ tensor, we plan to incrementally truncate $\Delta\mathcal{W}$ and evaluate whether any truncation method can effectively reverse-engineer some portion of the connectivity tensor.

2. **Universality**
   It has been proven that any open dynamical system of the form

$$s_{t+1} = g(s_t, u_t)$$
$$y_t = h(s_t)$$

   can be approximated by an element of RNN class with equation 1 form [13]. Moreover, the class of rank-R RNNs are universal approximators for R-dimensional dynamical systems [2]. We aim to extend these concepts to our framework and demonstrate that TBRNNs are universal approximators of any dynamical system in general. In addition, we hope to show that the class of rank-R TBRNNs are universal approximators for R-dimensional dynamical systems, or alternatively, to provide evidence to the contrary.

3. **TBRNN superiority over RNN**
   We hypothesize that a natural three-body dynamical system, such as a protein network defined in 2.3.3, will exhibit low-rank connectivity patterns. To test this hypothesis, we use a Three-Bodies Recurrent Neural Network (TBRNN) to predict the evolution of the Protein Interaction Network (PIN) over time based on the defined Partial Differential Equations (PDEs). To validate our hypothesis, we can apply a low-rank inference method to determine the rank of either $\mathcal{W}$ or $\Delta\mathcal{W}$, which are learned to predict protein states at each time step. If $\mathcal{W}$ is indeed a low-rank tensor, and if the connectivity matrix W of a vanilla RNN has a significantly higher rank on the same task, it would imply the justification of the use of TBRNNs for studying three-body dynamical systems. This is because TBRNNs offer higher interpretability, despite their greater computational cost in terms of time and space.

A Low rank three bodies RNN

Below, detailed the theoretical link between connectivity and low-dimensional dynamics in low rank TBRNNs, and outline its consequences for the main results.

A.1 Low-dimensional dynamics

First, we will start from three-bodies recurrent neural network, as from equation 7, with N neurons, each activation - fire rate variable notated by $x_i; 1 \leq i \leq N$ follows the dynamics:

$$\begin{aligned} \tau \frac{dx_i}{dt} &= -x_i + \phi^T(x) \cdot \mathcal{W}_i \cdot \phi(x) + I_i \cdot u(t) + \eta_i(t) \\ &= -x_i + \sum_{1 \leq j,k \leq N} \mathcal{W}_{ijk} \cdot \phi(x_j) \cdot \phi(x_k) + \sum_{s=1}^{N_{in}} I_i^{(s)} \cdot u_s(t) + \eta_i(t) \end{aligned} \tag{9}$$

Here, we assume that the connectivity tensor $\mathcal{W}$, is low ranked, as in [2, 4], so the effective total network dynamics that will be defined later, will be with low rank properties. According to Tensor Rank decomposition, and because $\mathcal{W}$ is low ranked tensor, exists some low R, such that:

$$\mathcal{W} = \frac{1}{N^2} \cdot \sum_{r=1}^{R} \mathbf{l}^{(r)} \otimes \mathbf{m}^{(r)} \otimes \mathbf{n}^{(r)} \tag{10}$$

Where, $\forall r \in [R], \mathbf{l}^{(r)}, \mathbf{m}^{(r)}, \mathbf{n}^{(r)} \in \mathbb{R}^N$ are constant apriori and known. Hence, get that:

$$\begin{aligned} \tau \dot{\mathbf{x}} = -\mathbf{x} &+ \frac{1}{N^2} \sum_{r=1}^{R} \mathbf{l}^{(r)} \cdot \left(\mathbf{m}^{(r)^T} \cdot \phi(\mathbf{x}) \right) \cdot \left(\mathbf{n}^{(r)^T} \cdot \phi(\mathbf{x}) \right) \\ &+ \sum_{s=1}^{N_{in}} \mathbf{I}^{(s)} \cdot u_s(t) + \eta(t) \end{aligned} \tag{11}$$

So, it can be noticed that x must evolve in the $R + N_{in}$ subspace, spanned by the vectors $l^{(r)}, I^{(s)}$.
Therefore, we can define $\kappa(t), \kappa_I(t)$ vectors which will describe the whole

network dynamics (and not individual neurons). Then it can be written that:

$$x(t) = \sum_{r=1}^{R} \kappa_r(t) \cdot \mathbf{l}^{(r)} + \sum_{s=1}^{N_{in}} \kappa_{I_s}(t) \cdot \mathbf{I}^{(s)} \tag{12}$$

Where:

$$\kappa_r(t) \coloneqq \frac{\left\langle \mathbf{l}^{(r)},\ \mathbf{x}(t) \right\rangle}{\|\mathbf{l}^{(r)}\|_2^2}$$

For each r, projecting equation 11 on $\mathbf{l}^{(r)}$ we get the dynamics of κ_r:

$$\tau \frac{d\kappa_r(t)}{dt} = -\kappa_r(t) + \frac{1}{N^2} \cdot \Big({\mathbf{m}^{(r)}}^T \cdot \phi(\mathbf{x}) \Big) \cdot \Big({\mathbf{n}^{(r)}}^T \cdot \phi(\mathbf{x}) \Big) \tag{13}$$

Where as in [2] assume that $\mathbf{l}^{(r)} \perp \mathbf{I}^{(s)}$. In addition, $\kappa_{I_s}(t)$ follows the dynamics:

$$\tau \frac{d\kappa_{I_s}(t)}{dt} = -\kappa_{I_s}(t) + u_s(t) \tag{14}$$

Now, denote:

$$\kappa_r^{rec}(t) \coloneqq \frac{1}{N^2} \cdot \Big({\mathbf{m}^{(r)}}^T \cdot \phi(\mathbf{x}) \Big) \cdot \Big({\mathbf{n}^{(r)}}^T \cdot \phi(\mathbf{x}) \Big)$$

Inserting equation 12 to $\kappa_r^{rec}(t)$ equation, we get that:

$$\begin{aligned} \kappa_r^{rec}(t) &= \frac{1}{N^2} \cdot \left\langle \mathbf{m}^{(r)},\ \phi(\mathbf{x}) \right\rangle \cdot \left\langle \mathbf{n}^{(r)},\ \phi(\mathbf{x}) \right\rangle \\ &= \frac{1}{N^2} \cdot \left\langle \mathbf{m}^{(r)},\ \phi\Big(\sum_{r'=1}^{R} \kappa_{r'}(t) \cdot \mathbf{l}^{(r')} + \sum_{s=1}^{N_{in}} \kappa_{I_s}(t) \cdot \mathbf{I}^{(s)} \Big) \right\rangle \\ &\cdot \left\langle \mathbf{n}^{(r)},\ \phi\Big(\sum_{r'=1}^{R} \kappa_{r'}(t) \cdot \mathbf{l}^{(r')} + \sum_{s=1}^{N_{in}} \kappa_{I_s}(t) \cdot \mathbf{I}^{(s)} \Big) \right\rangle \end{aligned} \tag{15}$$

Hence, $\kappa_r^{rec}(t)$ depends only on k_r and κ_{I_s} parameters, so we can conclude that three bodies tensor RNN with rank R tensor $\mathcal{W}$, induces rank $\sim R$ dynamics, but with quadratic dependency at each time t, and there is some nonlinear function $\mathfrak{F} : \mathbb{R}^{R+N_{in}} \to \mathbb{R}^R$ such that:

$$\tau \frac{d\kappa_r(t)}{dt} = \mathfrak{F}\Big(\kappa(t), u(t) \Big) \tag{16}$$

A.2 Mean-field approximation

From now on, the derivations are meant to and find explicitly an approximation properties of $\mathfrak{F}$, using the **DMF** (dynamic mean field) theorem. Here are our assumptions to DMF:

1. $N \gg R$ and $N \to \infty$.
2. Each neuron $i \in [N]$ is a point in a loading space of dimension $(3R + N_{in})$ which has coordinates:

$$\left(\{l_i^r\}_{r=1}^R, \{m_i^r\}_{r=1}^R, \{n_i^r\}_{r=1}^R, \{I_i^s\}_{s=1}^{N_{in}}\right) := (\underline{\mathrm{l}}_i, \underline{\mathrm{m}}_i, \underline{\mathrm{n}}_i, \underline{\mathrm{I}}_i)$$

3. The whole neurons set is i.i.d. with the distribution of gaussian mixture for each neuron i:
   - Each neuron $i \in [N]$ assigned to some population $p \in [P]$ with probability of α_p.
   - Within population p the joint distribution $P^{(p)}(l, m, n, I)$ is multivariate gaussian with the follow mean and covariance:

$$a^{(p)} = \left(a_{l_1}^{(p)}, \ldots, a_{l_R}^{(p)}, a_{m_1}^{(p)}, \ldots, a_{m_R}^{(p)}, a_{n_1}^{(p)}, \ldots, a_{n_R}^{(p)}, a_{I_1}^{(p)}, \ldots, a_{I_{N_{in}}}^{(p)}\right) \in \mathbb{R}^{3R+N_{in}}$$
$$\sigma_{xy}^{(p)} = \mathbb{E}\left[\left(x^{(p)} - a_x^{(p)}\right) \cdot \left(y^{(p)} - a_y^{(p)}\right)\right]$$

4. As in [4], we assume that all the variables are zero centered, that is:

$$a^{(p)} \equiv 0$$

So, start from equation 15, and use the big N assumption, we will show the derivation for one factor, but the other is equivalently derived.

First, we know by the central limit theorem that:

$$\frac{1}{N}\left\langle \mathbf{n}^{(r)},\ \phi\Big(\sum_{r'=1}^{R}\kappa_{r'}(t)\cdot\mathbf{l}^{(r')}+\sum_{s=1}^{N_{in}}\kappa_{I_s}(t)\cdot\mathbf{I}^{(s)}\Big)\right\rangle$$

$$\overset{N\to\infty}{\to}$$

$$\sum_{p=1}^{P}\alpha_p\int d_l\cdot d_m\cdot d_n\cdot d_I\cdot P^{(p)}\left(\underline{\mathrm{l}},\underline{\mathrm{m}},\underline{\mathrm{n}},\underline{\mathrm{I}}\right)\cdot\mathbf{n}_r^{(p)}\phi\left(\sum_{r'=1}^{R}\kappa_{r'}(t)\cdot\mathbf{l}^{(r')}+\sum_{s=1}^{N_{in}}\kappa_{I_s}(t)\cdot\mathbf{I}^{(s)}\right)$$

Where we used for shorthand:

$$d_l\cdot d_m\cdot d_n\cdot d_I=\prod_{r'=1}^{R}dl^{(r')}dm^{(r')}dn^{(r')}\prod_{s'=1}^{N_{in}}dI^{(s')}$$

And because $\int P^{(p)}\left(\underline{\mathrm{l}},\underline{\mathrm{m}},\underline{\mathrm{n}},\underline{\mathrm{I}}\right)d_m=P^{(p)}\left(\underline{\mathrm{l}},\underline{\mathrm{n}},\underline{\mathrm{I}}\right)$, we can rewrite the last derivation as:

$$\sum_{p=1}^{P}\alpha_p\int d_l\cdot d_n\cdot d_I\cdot P^{(p)}\left(\underline{\mathrm{l}},\underline{\mathrm{n}},\underline{\mathrm{I}}\right)\cdot\mathbf{n}_r^{(p)}\phi\left(\sum_{r'=1}^{R}\kappa_{r'}(t)\cdot\mathbf{l}^{(r')}+\sum_{s=1}^{N_{in}}\kappa_{I_s}(t)\cdot\mathbf{I}^{(s)}\right)$$

We know from [4], that the last equation is equal to:

$$\sum_{p=1}^{P}\alpha_p\left(\sum_{r'=1}^{R}\sigma^{(p)}_{n^{(r)}l^{(r')}}\kappa_{r'}+\sum_{s=1}^{N_{in}}\sigma^{(p)}_{n^{(r)}I^{(s)}}\kappa_{I_s}\right)\int Dz\phi'\left(\Delta^{(p)}z\right)$$

With:

$$\Delta^{(p)}=\sqrt{\sum_{r'=1}^{R}\left(\sigma^{(p)}_{n^{(r)}l^{(r')}}\right)^2\kappa^2_{r'}+\sum_{s=1}^{N_{in}}\left(\sigma^{(p)}_{n^{(r)}I^{(s)}}\right)^2\kappa^2_{I_s}}$$

Let us denote:

$$\langle\Phi'\rangle\coloneqq\int Dz\phi'\left(\Delta^{(p)}z\right)$$

$$\tilde{\sigma}_{ab}\coloneqq\sum_{p=1}^{P}\sigma^{(p)}_{ab}\langle\Phi'\rangle$$

$\langle\Phi'\rangle$ is the average gains over ϕ, and $\tilde{\sigma}_{ab}$ are the effective coupling as in [4].

So, we eventually got:

$$\frac{1}{N}\left\langle \mathbf{n}^{(r)},\ \phi\Big(\sum_{r'=1}^{R}\kappa_{r'}(t)\cdot\mathbf{l}^{(r')}+\sum_{s=1}^{N_{in}}\kappa_{I_s}(t)\cdot\mathbf{I}^{(s)}\Big)\right\rangle$$

$$\overset{N\to\infty}{\to}$$

$$\sum_{r'=1}^{R}\tilde{\sigma}^{(p)}_{n^{(r)}l^{(r')}}\kappa_{r'}+\sum_{s=1}^{N_{in}}\tilde{\sigma}^{(p)}_{n^{(r)}I^{(s)}}\kappa_{I_s} \tag{17}$$

Therefore, plug equation 17 in κ_r^{rec} definition, it holds that:

$$\kappa_r^{rec}(t)=\frac{1}{N^2}\cdot\left\langle \mathbf{m}^{(r)},\ \phi\Big(\sum_{r'=1}^{R}\kappa_{r'}(t)\cdot\mathbf{l}^{(r')}+\sum_{s=1}^{N_{in}}\kappa_{I_s}(t)\cdot\mathbf{I}^{(s)}\Big)\right\rangle$$
$$\cdot\left\langle \mathbf{n}^{(r)},\ \phi\Big(\sum_{r'=1}^{R}\kappa_{r'}(t)\cdot\mathbf{l}^{(r')}+\sum_{s=1}^{N_{in}}\kappa_{I_s}(t)\cdot\mathbf{I}^{(s)}\Big)\right\rangle$$

$$\overset{N\to\infty}{\to}$$

$$\left(\sum_{r'=1}^{R}\tilde{\sigma}^{(p)}_{m^{(r)}l^{(r')}}\kappa_{r'}+\sum_{s=1}^{N_{in}}\tilde{\sigma}^{(p)}_{m^{(r)}I^{(s)}}\kappa_{I_s}\right)\cdot\left(\sum_{r'=1}^{R}\tilde{\sigma}^{(p)}_{n^{(r)}l^{(r')}}\kappa_{r'}+\sum_{s=1}^{N_{in}}\tilde{\sigma}^{(p)}_{n^{(r)}I^{(s)}}\kappa_{I_s}\right)$$

$$=\sum_{1\le r',r''\le R}\tilde{\sigma}^{(p)}_{m^{(r)}l^{(r')}}\kappa_{r'}\tilde{\sigma}^{(p)}_{n^{(r)}l^{(r'')}}\kappa_{r''}+\sum_{\substack{1\le r'\le R;\\1\le s\le N_{in}}}^{R}\tilde{\sigma}^{(p)}_{m^{(r)}l^{(r')}}\kappa_{r'}\tilde{\sigma}^{(p)}_{n^{(r)}I^{(s)}}\kappa_{I_s}$$
$$+\sum_{\substack{1\le r'\le R;\\1\le s\le N_{in}}}^{R}\tilde{\sigma}^{(p)}_{n^{(r)}l^{(r')}}\kappa_{r'}\tilde{\sigma}^{(p)}_{m^{(r)}I^{(s)}}\kappa_{I_s}+\sum_{1\le s,s'\le N_{in}}^{R}\tilde{\sigma}^{(p)}_{m^{(r)}I^{(s)}}\kappa_{I_s}\tilde{\sigma}^{(p)}_{n^{(r)}I^{(s')}}\kappa_{I_{s'}}$$

For conveniency we denote the matrices:

$$\Sigma^{(r)}_{RR}\coloneqq\tilde{\sigma}_{m^{(r)}\mathbf{l}}\cdot\tilde{\sigma}^{T}_{n^{(r)}\mathbf{l}}\in\mathbb{R}^{R\times R}$$
$$\Sigma^{(r)}_{R_mI_n}\coloneqq\tilde{\sigma}_{m^{(r)}\mathbf{l}}\cdot\tilde{\sigma}^{T}_{n^{(r)}\mathbf{I}}\in\mathbb{R}^{R\times N_{in}}$$
$$\Sigma^{(r)}_{R_nI_m}\coloneqq\tilde{\sigma}_{n^{(r)}\mathbf{l}}\cdot\tilde{\sigma}^{T}_{m^{(r)}\mathbf{I}}\in\mathbb{R}^{R\times N_{in}}$$
$$\Sigma^{(r)}_{II}\coloneqq\tilde{\sigma}_{m^{(r)}\mathbf{I}}\cdot\tilde{\sigma}^{T}_{n^{(r)}\mathbf{I}}\in\mathbb{R}^{N_{in}\times N_{in}}$$

Then we get:

$$\kappa_r^{rec} = \kappa^T \Sigma_{RR}^{(r)} \kappa + \kappa^T \left(\Sigma_{R_m I_n}^{(r)} + \Sigma_{R_n I_m}^{(r)} \right) \kappa_I + \kappa_I^T \Sigma_{II}^{(r)} \kappa_I$$

Thus:

$$\tau \frac{d\kappa_r}{dt} = -\kappa_r + \kappa^T \Sigma_{RR}^{(r)} \kappa + \kappa^T \left(\Sigma_{R_m I_n}^{(r)} + \Sigma_{R_n I_m}^{(r)} \right) \kappa_I + \kappa_I^T \Sigma_{II}^{(r)} \kappa_I \tag{18}$$

Which realizes explicitly the function $\mathfrak{F}$, and illustrates the quadratic dependencies within the computation of κ itself. Note that from the structure of equation 18 it can be noted that that the latent space dynamics of κ is from dimension R, while its inner dependencies are quadratic in $R + N_{in}$, i.e., $(R + N_{in})^2$.

A.3 Effective low rank connectivity

Deducted from equation 18, notably the low-dimensional dynamics depend directly on $l^{(r)}$ and $I^{(s)}$ vectors, but not on the exact entries of $m^{(r)}, n^{(r)}$ vectors. Instead, $m^{(r)}, n^{(r)}$ vectors influence the dynamics only through their covariances with $l^{(r)}, I^{(s)}$ vectors. A way to see that is directly from Σ's matrices definition. For example, the part of $m^{(r)}$ orthogonal to $l^{(r)}, I^{(s)}$ vectors, has zero influence on $\tilde{\sigma}_{m^{(r)}\mathbf{l}}, \tilde{\sigma}_{m^{(r)}\mathbf{I}}$, thus has no contribution to any among $\left\{ \Sigma_{RR}^{(r)}, \Sigma_{R_m I_n}^{(r)}, \Sigma_{R_n I_m}^{(r)}, \Sigma_{II}^{(r)} \right\}$, so the contribution to the development of $\kappa_{(r)}$ is 0. The same holds for $n^{(r)}$. Therefore, all components of $m^{(r)}, n^{(r)} \perp SP\left\{ l^{(r)}, I^{(s)} \right\}$ are irrelevant for the dynamics, and removing them leads to an effective connectivity tensor $\mathcal{W}_{eff}$ which captures the minimal features required for obtaining particular dynamics.
Hence, for a rank R TBRNN, $\mathcal{W}_{eff}$ is computed as follows:

$$\mathcal{W}_{eff} = \frac{1}{N^2} \cdot \sum_{r=1}^{R} \mathbf{l}^{(r)} \otimes \mathbf{m}_{||}^{(r)} \otimes \mathbf{n}_{||}^{(r)}$$

where each $\mathbf{m}_{||}^{(r)}, \mathbf{n}_{||}^{(r)}$ is defined as the orthogonal projection of $\mathbf{m}^{(r)}, \mathbf{n}^{(r)}$ on the subspace spanned by $l^{(r)}, I^{(s)}$ respectively. Note that there is no such limitation between $\mathbf{m}^{(r)}$ and $\mathbf{n}^{(r)}$.

B TBRNNs are universal approximators

The universal approximation is a theorem that claims that some family of functions in some domain can approximate any function in the domain. It is well known that wide-enough one hidden layer feed-forward network model of the form $f(x) = a^T \phi\left(Wx + b\right); a \in \mathbb{R}^m, W \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$, can approximate any function $g : \mathbb{R}^n \to \mathbb{R}$ with an arbitrary accuracy [6]. Assisted by the last fact, [13] showed that RNN element of the form:

$$\begin{aligned} s_{t+1} &= \phi\left(W \cdot s_t + B \cdot u_t + \theta\right) \\ y_t &= Cs_t \end{aligned}$$

Can approximate any open dynamical system of the form:

$$\begin{aligned} s_{t+1} &= g\left(s_t, u_t\right) \\ y_t &= h(s_t) \end{aligned}$$

With network inputs $u_t \in \mathbb{R}^I$, hidden states $s_t \in \mathbb{R}^H$, outputs $y_t \in \mathbb{R}^n$ and $\phi : \mathbb{R}^H \to \mathbb{R}^H$ non-linear activation function, for $t = 1, ..., T$.
We will now define the $TBRNN$ functions class, as well as the three-bodies feed forward function class, then show in two steps that $TBRNN$ are universal approximators to any open dynamical system of the aforementioned form.

Definition 1. *For any continuous measurable function $\phi : \mathbb{R}^H \to \mathbb{R}^H$ and $I, n \in \mathbb{N}$ be $TBRNN^{I,n}(\phi)$ the class of functions:*

$$\begin{aligned} s_{t+1} &= \phi\left(s_t^T \otimes \mathcal{W} \otimes s_t + B \cdot u_t - \theta\right) \\ y_t &= Cs_t \end{aligned}$$

Such that $u_t \in \mathbb{R}^I, s_t \in \mathbb{R}^H$ and $y_t \in \mathbb{R}^n$, for $t = 1, ...T$. In addition, let the weights be $\mathcal{W} \in \mathbb{R}^{H^3}, B \in \mathbb{R}^{H \times I}, C \in \mathbb{R}^{n \times H}$ and bias $\theta \in \mathbb{R}^H$.

Definition 2. *Consider unbounded width Three-Bodies Feed-Forward Network with one hidden layer function group:*

$$\mathcal{F}_{\phi,d,m} \coloneqq \mathcal{F}_{d,m} \coloneqq \{x \mapsto a^T \phi\left(x \otimes \mathcal{W} \otimes x + b\right) : a \in \mathbb{R}^m, \mathcal{W} \in \mathbb{R}^{m \times d \times d}, b \in \mathbb{R}^m\}$$

$$\mathcal{F}_{\phi,d} \coloneqq \mathcal{F}_d \coloneqq \bigcup_{m \geq 0} \mathcal{F}_{d,m}$$

B.1 Three-Bodies Feed-Forward universality

Let $\mathcal{C}(\mathcal{D}, \mathcal{X})$ be the family class of functions from domain $\mathcal{D}$ to $\mathcal{X}$.
In this part, we will show that for any sigmoidal activation function ϕ, $\mathcal{F}_{\phi,d}$ is a universal approximator of $\mathcal{C}\left([0,1]^d, \mathbb{R}\right)$. This will be derived from the two lemmas in the following.

Lemma B.1. *$\mathcal{F}_{cos,d}$ is universal approximator of $\mathcal{C}\left([0,1]^d, \mathbb{R}\right)$*

Proof. We will show that the four criterion of the Stone-Weierstrass theorem [5] are met:

1. *Continuity.* Each $f \in \mathcal{F}_{cos,d}$ is continuous.

2. *Identity.* For every $x \in [0,1]^d$, take $cos\left(x^T \cdot \mathbf{0} \cdot x\right) = cos(0) = 1 \neq 0$.

3. *Separation.* For every $\mathbf{x} \neq \mathbf{x}' \in [0,1]^d$, exists $1 \leq i \leq d$ such that $x_i \neq x'_i$. Let $\mathcal{W}_i = diag\left(0, \ldots, 0, \underset{i}{1}, 0, \ldots, 0\right)$, and define $f(z) = cos(z^T \mathcal{W}_i z) = cos(z_i^2)$. Because $x_i, x'_i \in [0,1]$ then $f(\mathbf{x}) = cos(x_i^2) \neq cos(x_i'^2) = f(\mathbf{x}')$.

4. *Closure.* Let $f_1, f_2 \in \mathcal{F}_{cos,d}$. $f_1 + f_2 \in \mathcal{F}_{cos,d}$ by adding more neurons to the network (and using a block matrix). Now, assume $f_k(x) = a_k^T cos(x \otimes \mathcal{W}_k \otimes x + b_k)$, for $k \in \{1,2\}$. Therefore,

   $$f_1 \cdot f_2(x) = a_1^T \cdot cos(x \otimes \mathcal{W}_1 \otimes x + b_1) \cdot cos(x \otimes \mathcal{W}_2 \otimes x + b_2)^T \cdot a_2$$

   But,

   $$\begin{aligned} &cos(x^T W_1 x + b_1) \cdot cos(x^T W_2 x + b_2) \\ &= cos(x^T (W_1 + W_2) x + (b_1 + b_2)) + cos(x^T (W_1 - W_2) x + (b_1 - b_2)) \end{aligned}$$

   Therefore, $f_1 \cdot f_2 \in \mathcal{F}_{cos,d}$

□

Lemma B.2. *Let $g(x) = a^T cos(x \otimes \mathcal{W} \otimes x + b), x \in [0,1]^d$. For an arbitrary sigmoidal function ϕ, for arbitrary compact $K \subset [0,1]^d$, and for arbitrary $\epsilon > 0$, there is an $f \in \mathcal{F}_{\phi,d}$ such that $\sup_{x \in K} |g(x) - f(x)| < \epsilon$.*

Proof. Pick an integer $M > 0$, such that $\forall i \in [Q], \forall x \in [0,1]^d, x^T \mathcal{W}_i x + b_i \in [-M, M]$. Since Q is finite, K is compact and $x^T \mathcal{W}_i x + b_i$ continuous, such M can be found. Let, $Q' = Q \sum_{i=1}^{Q} |a_i|$. By Lemma A.3 presented in Hornick et. al. [6], for all $x \in K$ exists affine function $A(x) = \hat{a}x + \hat{b}$, so $cos_{M,\frac{\epsilon}{Q'}} \equiv \phi \circ A$ such that:

$$\left| \sum_{i=1}^{Q} a_i cos_{M,\frac{\epsilon}{Q'}}(x^T \mathcal{W}_i x + b_i) - g(x) \right| < \epsilon$$

We see that:

$$\begin{aligned} f(x) &= \sum_{i=1}^{Q} a_i cos_{M,\frac{\epsilon}{Q'}}(x^T \mathcal{W}_i x + b_i) \\ &= \sum_{i=1}^{Q} a_i \phi(A(x^T \mathcal{W}_i x + b_i)) \\ &= \sum_{i=1}^{Q} a_i \phi(x^T (\hat{a}\mathcal{W}_i) x + \hat{a} b_i + \hat{b})) \\ &= a^T \phi(x \otimes (\hat{a}\mathcal{W}) \otimes x + (\hat{a}\mathbf{b} + \hat{\mathbf{b}})) \in \mathcal{F}_{\phi,d} \end{aligned}$$

□

Theorem B.3. *Let $\phi : \mathbb{R}^d \to \mathbb{R}^d$ be a continuous sigmoidal function, such that $\lim_{x\to\infty} \phi(x) = 1$, $\lim_{x\to-\infty} \phi(x) = 0$. Then, $\mathcal{F}_{\phi,d}$ is a universal approximator of $\mathcal{C}\left([0,1]^d, \mathbb{R}\right)$.*

Proof. Let $\epsilon > 0$ and $g : [0,1]^d \to \mathbb{R}$. From Lemma B.1, we know that $\mathcal{F}_{cos,d}$ is a universal approximator of $\mathcal{C}\left([0,1]^d, \mathbb{R}\right)$. Then for $\frac{\epsilon}{2}$, let $a^T cos(x \otimes \mathcal{W} \otimes x + b) \in \mathcal{F}_{cos,d}$ such that:

$$\sup_{x \in [0,1]^d} \left| g(x) - a^T cos(x \otimes \mathcal{W} \otimes x + b) \right| < \frac{\epsilon}{2}$$

From Lemma B.2, for $\frac{\epsilon}{2}$, exists $f \in \mathcal{F}_{\phi,d}$, such that:

$$\sup_{x \in [0,1]^d} \left| a^T cos(x \otimes \mathcal{W} \otimes x + b) - f(x) \right| < \frac{\epsilon}{2}$$

From triangle inequality the theorem is proven. □

Note that by Theorem 1 of [13], $\mathcal{F}_{\phi,d}$ can be extended to multi-output functions:

$$\mathcal{F}_{\phi,d}^{n} \coloneqq \bigcup_{m\geq 0} \{x \mapsto A\phi\left(x \otimes \mathcal{W} \otimes x + b\right) : A \in \mathbb{R}^{n\times m}, \mathcal{W} \in \mathbb{R}^{m\times d\times d}, b \in \mathbb{R}^{m}\}$$

that $\mathcal{F}_{\phi,d}^{n}$ is a universal approximator of $\mathcal{C}\left([0,1]^{d}, \mathbb{R}^{n}\right)$.

B.2 TBRNNs are universal approximators with any sigmoidal activation function

Lemma B.4. *Let $g : \mathbb{R}^{H} \times \mathbb{R}^{I} \to \mathbb{R}^{H}$ measurable function, and $h : \mathbb{R}^{H} \to \mathbb{R}^{n}$ be continuous function; the external inputs $u_t \in \mathbb{R}^{I}$, hidden states $s_t \in \mathbb{R}^{H}$ and system outputs $y_t \in \mathbb{R}^{n}$, $\forall t \in [T]$. Then, for any open dynamical system of the form:*

$$\begin{aligned} s_{t+1} &= g(s_t, u_t) \\ y_t &= h(s_t) \end{aligned}$$

Exists $g' : [0,1]^{H} \times \mathbb{R}^{I} \to [0,1]^{H}$, measurable, and $h' : [0,1]^{H} \to \mathbb{R}^{n}$ continuous function such that:

$$\begin{aligned} s'_{t+1} &= g'(s'_t, u_t) \\ y_t &= h'(s'_t) \end{aligned}$$

Proof. Since $\mathbb{R}$ and $[0,1]$ are equinumerous, then exists bijection (and reversible) function $a : \mathbb{R}^{H} \to [0,1]^{H}$. Denote:

$$s'_t \coloneqq a(s_t)$$

then

$$s'_{t+1} = a(s_{t+1}) = a\left(g(s_t, u_t)\right) = a\left(g(a^{-1}(s'_t), u_t)\right)$$

Now enlarge a^{-1} to be $\tilde{a} : [0,1]^{H} \times \mathbb{R}^{I} \to \mathbb{R}^{H} \times \mathbb{R}^{I}$ such that:

$$\tilde{a}(s'_t, u_t) = (a^{-1}(s'_t), u_t)$$

So:

$$s'_{t+1} = a\left(g\left(\tilde{a}(s'_t, u_t)\right)\right)$$

Now denote $g' : [0,1]^H \times \mathbb{R}^I \to [0,1]^H$ to be $g' \coloneqq a \circ g \circ \tilde{a}$ measurable function, and $h' : [0,1]^H \to \mathbb{R}^n$ to be $h' \coloneqq h \circ a^{-1}$ continuous, so we get:

$$\begin{aligned} s'_{t+1} &= g'(s'_t, u_t) \\ y_t &= h'(s'_t) \end{aligned}$$

□

Theorem B.5. *(TBRNN universal approximation). Let $g : \mathbb{R}^H \times \mathbb{R}^I \to \mathbb{R}^H$ measurable function, and $h : \mathbb{R}^H \to \mathbb{R}^n$ be continuous function; the external inputs $u_t \in \mathbb{R}^I$, hidden states $s_t \in \mathbb{R}^H$ and system outputs $y_t \in \mathbb{R}^n$, $\forall t \in [T]$. Then, any open dynamical system of the form:*

$$\begin{aligned} s_{t+1} &= g(s_t, u_t) \\ y_t &= h(s_t) \end{aligned}$$

can be approximated by an element of the function class $TBRNN^{I,n}(f)$, with an arbitrary accuracy, where f is a continuous sigmoidal activation function.

Proof. First, from Lemma B.4, it is sufficient to approximate $g : [0,1]^H \times \mathbb{R}^I \to [0,1]^H$ and $h : [0,1]^H \to \mathbb{R}^n$. From now on, following the proof presented in "RNNs are universal approximator" [13].
Let $\epsilon > 0$. Then for arbitrary $\delta > 0$, from Theorem B.3, a function

$$TBNN(s_t, u_t) \coloneqq V \cdot f(s_t \otimes \mathcal{W} \otimes s_t + Bu_t - \bar{\theta})$$

with weights $V \in \mathbb{R}^{H \times \bar{H}}, \mathcal{W} \in \mathbb{R}^{\bar{H} \times H \times H}, B \in \mathbb{R}^{\bar{H} \times H}, \bar{\theta} \in \mathbb{R}^{\bar{H}}$ exists, that:

$$\sup_{s_t, u_t \in K} |g(s_t, u_t) - TBNN(s_t, u_t)| < \delta; \forall t = 1, \ldots, T$$

Then exists $\delta_\epsilon > 0$ that from $\delta - \epsilon$ criterion, for the dynamics:

$$\bar{s}_{t+1} = Vf\left(\bar{s}_t \otimes \mathcal{W} \otimes \bar{s}_t + Bu_t - \bar{\theta}\right)$$

the following holds:

$$|s_t - \bar{s}_t| < \epsilon; \forall t = 1, \ldots, T$$

Furthermore, define

$$s'_{t+1} \coloneqq f\left(\bar{s}_t \otimes \mathcal{W} \otimes \bar{s}_t + Bu_t - \bar{\theta}\right)$$

So,

$$\hat{s}_t = V s'_t$$

Therfore,

$$\begin{aligned} s'_{t+1} &= f\left((V s'_t) \otimes \mathcal{W} \otimes (V s'_t) + B u_t - \bar{\theta}\right) \\ &= f\left(s'_t \otimes (V \otimes \mathcal{W} \otimes V) \otimes s'_t + B u_t - \bar{\theta}\right) \end{aligned}$$

Take $\mathcal{A} \coloneqq V \otimes \mathcal{W} \otimes V \in \mathbb{R}^{\bar{H}^3}$, we get that:

$$s'_{t+1} = f\left(s'_t \otimes \mathcal{A} \otimes s'_t + B u_t - \bar{\theta}\right)$$

Note that $s' \in \mathbb{R}^{\bar{H}}$, while $\bar{s} \in \mathbb{R}^H$.
To finish the proof, we need to show that $y_t = h(s_t)$ can be approximated by some $\bar{y} = C\bar{s}_t$.
Let $\tilde{\epsilon} > 0$. Since h is continuous, then exists $\epsilon > 0$ that for $|s_t - \bar{s}_t| < \epsilon$, $|h(s_t) - h(\bar{s}_t)| = |y_t - \hat{y}_t| < \tilde{\epsilon}$. Hence, $\hat{y}_t$ will instead be approximated and, with triangle inequality, it holds. Because $h : [0,1]^H \to \mathbb{R}^n$, by Theorem B.3, it can be approximated by: $\bar{y}_t = N f(\bar{s}_t \otimes \mathcal{M} \otimes \bar{s}_t - \hat{\theta})$, with $N \in \mathbb{R}^{n \times \hat{H}}, \mathcal{M} \in \mathbb{R}^{\hat{H} \times H \times H}, \hat{\theta} \in \mathbb{R}^{\hat{H}}$. Then, by insertion we get:

$$\begin{aligned} \bar{y}_t &= N f\left(\bar{s}_t \otimes \mathcal{M} \otimes \bar{s}_t - \hat{\theta}\right) \\ &= N f\left((V s'_t) \otimes \mathcal{M} \otimes (V s'_t) - \hat{\theta}\right) \\ &= N f\left(s'_t \otimes (V \otimes \mathcal{M} \otimes V) \otimes s'_t - \hat{\theta}\right) \\ &= N f\big(f\left(\bar{s}_{t-1} \otimes \mathcal{W} \otimes \bar{s}_{t-1} + B u_{t-1} - \bar{\theta}\right) \otimes (V \otimes \mathcal{M} \otimes V) \\ &\otimes f\left(\bar{s}_{t-1} \otimes \mathcal{W} \otimes \bar{s}_{t-1} + B u_{t-1} - \bar{\theta}\right) - \hat{\theta}\big) \coloneqq \hat{f}(\bar{s}_{t-1}, u_{t-1}) \end{aligned}$$

Using again Theorem B.3, $\hat{f}$ can be approximated by:

$$\tilde{y}_t = D f(s'_{t-1} \otimes \mathcal{E} \otimes s'_{t-1} + F u_{t-1} - \tilde{\theta})$$

with $D \in \mathbb{R}^{n \times \bar{\bar{H}}}, \mathcal{E} \in \mathbb{R}^{\bar{\bar{H}} \times H \times H}, F \in \mathbb{R}^{\bar{\bar{H}} \times I}, \tilde{\theta} \in \mathbb{R}^{\bar{\bar{H}}}$.
Now, set:

$$r_{t+1} \coloneqq f(s'_t \otimes \mathcal{E} \otimes s'_t + F u_t - \tilde{\theta}) \left(\in \mathbb{R}^{\bar{\bar{H}}}\right)$$

And enlarge the system equations, we get:

$$\begin{pmatrix} s'_{t+1} \\ r_{t+1} \end{pmatrix} = f\left(\begin{pmatrix} s'_t \\ r_t \end{pmatrix} \otimes \mathcal{X} \otimes \begin{pmatrix} s'_t \\ r_t \end{pmatrix} + \begin{pmatrix} B \\ F \end{pmatrix} u_t - \begin{pmatrix} \bar{\theta} \\ \tilde{\theta} \end{pmatrix}\right)$$
$$\tilde{y}_t = \begin{pmatrix} 0 & D \end{pmatrix} \cdot \begin{pmatrix} s'_t \\ r_t \end{pmatrix}$$

With:

$$\mathcal{X}_i = \begin{cases} \begin{pmatrix} \mathcal{A}_i & 0 \\ 0 & 0 \end{pmatrix}, & \text{for } 1 \le i \le \bar{H} \\ \begin{pmatrix} \mathcal{E}_{i-\bar{H}} & 0 \\ 0 & 0 \end{pmatrix}, & \text{for } \bar{H} \le n \le \bar{H} + \bar{\bar{H}} \end{cases}$$

So,

$$\begin{pmatrix} s'_t \\ r_t \end{pmatrix} \otimes \mathcal{X} \otimes \begin{pmatrix} s'_t \\ r_t \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} s'_t & r_t \end{pmatrix} \cdot \mathcal{X}_1 \cdot \begin{pmatrix} s'_t \\ r_t \end{pmatrix} \\ \vdots \\ \begin{pmatrix} s'_t & r_t \end{pmatrix} \cdot \mathcal{X}_{\bar{H}+\bar{\bar{H}}} \cdot \begin{pmatrix} s'_t \\ r_t \end{pmatrix} \end{pmatrix}$$
$$= \begin{pmatrix} s_t'^T \cdot \mathcal{A}_1 \cdot s'_t \\ \vdots \\ s_t'^T \cdot \mathcal{A}_{\bar{H}} \cdot s'_t \\ s_t'^T \cdot \mathcal{E}_1 \cdot s'_t \\ \vdots \\ s_t'^T \cdot \mathcal{E}_{\bar{\bar{H}}} \cdot s'_t \end{pmatrix} = \begin{pmatrix} s'_t \otimes \mathcal{A} \otimes s'_t \\ s'_t \otimes \mathcal{E} \otimes s'_t \end{pmatrix}$$

Finally, denote:

$$\tilde{H} \coloneqq \bar{H} + \bar{\bar{H}}$$
$$\tilde{s}_t \coloneqq \begin{pmatrix} s'_t \\ r_t \end{pmatrix} \in \mathbb{R}^{\tilde{H}}$$
$$\mathcal{X} \in \mathbb{R}^{\tilde{H}^3}$$
$$\tilde{B} \coloneqq \begin{pmatrix} B \\ F \end{pmatrix} \in \mathbb{R}^{\tilde{H} \times I}, \tilde{C} \coloneqq \begin{pmatrix} 0 & D \end{pmatrix} \in \mathbb{R}^{n \times \tilde{H}}, \theta \coloneqq \begin{pmatrix} \bar{\theta} \\ \tilde{\theta} \end{pmatrix} \in \mathbb{R}^{\tilde{H}}$$

We get:

$$\tilde{s}_{t+1} = f\left(\tilde{s}_t \otimes \mathcal{X} \otimes \tilde{s}_t + \tilde{B}u_t - \theta\right)$$
$$\tilde{y}_t = \tilde{C}\tilde{s}_t$$

Obviously, it is element of $TBRNN^{I,n}(f)$, and $\tilde{y}_t = \hat{y}_t$; and from triangle inequality it can be concluded that:

$$|y_t - \tilde{y}_t| < \tilde{\epsilon}$$

The theorem is proven. □

References

[1] Omri Barak. "Recurrent neural networks as versatile tools of neuroscience research". In: *Current Opinion in Neurobiology* 46 (2017), pp. 1–6. URL: https://api.semanticscholar.org/CorpusID:46360000.

[2] Manuel Beirán et al. "Shaping Dynamics With Multiple Populations in Low-Rank Recurrent Networks". In: *Neural Computation* 33 (2020), pp. 1572–1615. URL: https://api.semanticscholar.org/CorpusID:220364304.

[3] Andrea Ciliberto, Fabrizio Capuani, and John J. Tyson. "Modeling Networks of Coupled Enzymatic Reactions Using the Total Quasi-Steady State Approximation". In: *PLoS Computational Biology* 3 (2007). URL: https://api.semanticscholar.org/CorpusID:3197479.

[4] Alexis M. Dubreuil et al. "The role of population structure in computations through neural dynamics". In: *Nature Neuroscience* 25 (2022), pp. 783–794. URL: https://api.semanticscholar.org/CorpusID:256838997.

[5] Gerald B. Folland. "Real Analysis: Modern Techniques and Their Applications". In: 1984. URL: https://api.semanticscholar.org/CorpusID:260492294.

[6] Kurt Hornik, Maxwell B. Stinchcombe, and Halbert L. White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2 (1989), pp. 359–366. URL: https://api.semanticscholar.org/CorpusID:2757547.

[7] Niru Maheswaranathan et al. "Universality and individuality in neural dynamics across large populations of recurrent networks". In: *Advances in neural information processing systems* 2019 (2019), pp. 15629–15641. URL: https://api.semanticscholar.org/CorpusID:197935221.

[8] Francesca Mastrogiuseppe and Srdjan Ostojic. "Linking Connectivity, Dynamics, and Computations in Low-Rank Recurrent Neural Networks". In: *Neuron* 99 (2017), 609–623.e29. URL: https://api.semanticscholar.org/CorpusID:51866519.

[9] Diego Maupomé and Marie-Jean Meurs. "Language Modeling with a General Second-Order RNN". In: *International Conference on Language Resources and Evaluation*. 2020. URL: https://api.semanticscholar.org/CorpusID:218973949.

[10] Kenneth D. Miller and Francesco Fumarola. "Mathematical Equivalence of Two Common Forms of Firing Rate Models of Neural Networks". In: *Neural Computation* 24 (2012), pp. 25–31. URL: https://api.semanticscholar.org/CorpusID:5107843.

[11] I. Oseledets. "Tensor-Train Decomposition". In: *SIAM J. Sci. Comput.* 33 (2011), pp. 2295–2317. URL: https://api.semanticscholar.org/CorpusID:207059098.

[12] Arthur Pellegrino, Heike Stein, and N. Alex Cayco-Gajic. "Disentangling Mixed Classes of Covariability in Large-Scale Neural Data". In: *bioRxiv* (2023). URL: https://api.semanticscholar.org/CorpusID:257335265.

[13] Anton Maximilian Schäfer and Hans-Georg Zimmermann. "Recurrent Neural Networks Are Universal Approximators". In: *International journal of neural systems* 17 4 (2006), pp. 253–63. URL: https://api.semanticscholar.org/CorpusID:2238422.

[14] Friedrich Schuessler et al. "The interplay between randomness and structure during learning in RNNs". In: *arXiv: Neurons and Cognition* (2020). URL: https://api.semanticscholar.org/CorpusID:219955885.

[15] Lee A. Segel. "On the validity of the steady state assumption of enzyme kinetics." In: *Bulletin of mathematical biology* 50 6 (1988), pp. 579–93. URL: https://api.semanticscholar.org/CorpusID:39052729.

[16] Shreyansh Singh and John Hewitt. "Second-order RNNs: Resurgence of Recurrence". In: 2020. URL: https://api.semanticscholar.org/CorpusID:215716777.

[17] David Sussillo and Omri Barak. "Opening the Black Box: Low-Dimensional Dynamics in High-Dimensional Recurrent Neural Networks". In: *Neural Computation* 25 (2013), pp. 626–649. URL: https://api.semanticscholar.org/CorpusID:3082835.

[18] Adrian Valente, Jonathan W. Pillow, and Srdjan Ostojic. "Extracting computational mechanisms from neural data using low-rank RNNs". In: *Neural Information Processing Systems*. 2022. URL: https://api.semanticscholar.org/CorpusID:258508970.