Semester Project : Generalized Tensor Models for Recurrent Neural Networks

Nitya Afambo

Supervisor:
Prof. Dr. Schwab Christoph

Advisor:
Dr. Maksim Rakhuba

May 25, 2020

Contents

Abstract

The purpose of the present semester paper is to study from a theoretical and practical point of view, the so-called *depth efficiency* of neural networks using the framework provided by tensor methods. More precisely, we look at how one can relate certain network architectures to certain tensor decompositions. We focus on recurrent neural networks while briefly discussing the case of shallow feedforward networks for the sake of comparaison. The theoretical exposition follows [5] and [6] where we give more detailed proofs of the statements. Unless stated otherwise, all theoretical results come from these two papers. The practical work consists in the implementation of so-called *generalized tensor networks* and conducting numerical experimentation to verify how well theoretical results hold in practice.

1 Introduction

Recent years have seen machine learning algorithms increase in popularity due to their ability to solve complex problems. In particular, deep learning and artificial neural networks have gained attention by outperforming other machine learning algorithms when considering tasks such as image and speech recognition. Although widely used, there are still attempts today to interpret and obtain a better comprehension of how these deep learning algorithms work. Examples of such attempts consist in visualizing activation values for neurons in the network ([12]) and interpretability based on pixel-wise decompositions of images ([2]). A recent line of work tries to address the question of the *expressive power* of deep networks by establishing a connection between certain tensor decompositions and certain network architectures by seeing the weights of a network as a single tensor.

This semester paper studies approaches relying on tensor-based methods that naturally have many applications in the context of machine learning which makes extensive use of multidimensional arrays. In what follows we focus on tensor decompositions of the weight tensor and see how they lead to various neural network architectures. The paper is structured as follows. In section 2 we recall the relevant tensor-based notions as well as introduce so-called generalized tensors. Section 3 introduces the problem framework of solving a classification problem with tensor networks. Theoretical results concerning these networks are proved in section 4 and how well these results hold in practice is verified by numerical experimentations presented in section 5.

2 Generalized Tensor Decompositions

In this section, we recall all the necessary definitions and basic results concerning tensors that we will need and use in subsequent sections. In our current setting, a tensor is understood to be a multidimensional array

$$\boldsymbol{\mathcal{W}} \in \mathbb{R}^{M_1 \times \cdots \times M_T}, \tag{1}$$

where T is the *order* and $M_i \in \mathbb{N}$ is the dimension in *mode* i of the tensor. Given two tensors $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ of order P and Q respectively, their tensor product $\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{B}}$ is the tensor of order $P+Q$ defined by

$$(\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{B}})_{i_1 \ldots i_P j_1 \ldots j_Q} = \boldsymbol{\mathcal{A}}_{i_1 \ldots i_P} \cdot \boldsymbol{\mathcal{B}}_{j_1 \ldots j_Q}. \tag{2}$$

In what follows, we take tensor products between vectors $\mathbf{u} \in \mathbb{R}^{M_1}$ and $\mathbf{v} \in \mathbb{R}^{M_2}$, in which case the tensor product coincides with the standard outer product. Standard tensor decompositions are obtained by expressing a tensor as a weighted sum of tensor products. Such decompositions can thus be generalized if we are able to generalize the tensor operator. For this purpose, let $\xi : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ be an associative and commutative binary operator, i.e $\xi(\xi(x, y), z) = \xi(x, \xi(y, z))$ and $\xi(x, y) = \xi(y, x)$ $\forall x, y, z \in \mathbb{R}$ respectively. The tensor product can be generalized as follows: for tensors $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{B}}$ of order P and Q respectively, we define their *generalized outer product* $\boldsymbol{\mathcal{A}} \otimes_\xi \boldsymbol{\mathcal{B}}$ to be an $(P + Q)$ order tensor with entries

$$(\boldsymbol{\mathcal{A}} \otimes_\xi \boldsymbol{\mathcal{B}})_{i_1 \dots i_P j_1, \dots J_Q} = \xi(\boldsymbol{\mathcal{A}}_{i_1 \dots i_P}, \boldsymbol{\mathcal{B}}_{j_1 \dots J_Q}). \tag{3}$$

Given a choice of subset of axes of $\boldsymbol{\mathcal{W}}$ as in (1), $s = \{i_1, i_2, ..., i_{m_s}\}$ and its complement $t = \{j_1, j_2, ..., j_{T-m_s}\}$, the *matricization* of $\boldsymbol{\mathcal{W}}$ specified by (s, j) is a matrix

$$\boldsymbol{\mathcal{W}}^{(s,t)} \in \mathbb{R}^{M_{i_1} M_{i_2} \dots M_{i_{m_s}} \times M_{j_1} M_{j_2} \dots M_{j_{T-m_s}}} \tag{4}$$

corresponding to a reshape of $\boldsymbol{\mathcal{W}}$. For a tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{M_1 \times \dots \times M_N}$ where we assume N to be even, we will particularly be interested in the matricization where odd modes correspond to rows and even modes to columns i.e $s = \{1, 3, ..., N - 1\}$ and $t = \{2, 4, ..., N\}$. For this particular matricization, element $\boldsymbol{\mathcal{W}}_{i_1 \dots i_N}$ is mapped to index (l, j) with

$$l = 1 + \sum_{n=1}^{N/2} (i_{2n-1} - 1) \prod_{k=n+1}^{N/2} M_{2k-1}$$

and

$$j = 1 + \sum_{n=1}^{N/2} (i_{2n} - 1) \prod_{k=n+1}^{N/2} M_{2k}.$$

The *vectorization* of $\boldsymbol{\mathcal{W}}$ is a linear transformation that converts $\boldsymbol{\mathcal{W}}$ into a column vector $vec(\mathcal{W}) \in \mathbb{R}^{\prod_{n=1}^{N} M_n}$. More specificly, we stack into a vector the columns of the matrix corresponding to the matricization of the tensor along its first mode.

Tensor decompositions allow one to express a tensor as a weighted sum of tensor products. For tensors with an exponentially large number of elements, such decompositions have the benefit of providing us with a way of compactly represent them using a smaller amount of elements. The two decompositions we will focus on are the *Canonical Polyadic* (CP) decomposition and the *Tensor Train* (TT) decomposition. The CP decomposition of $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{M_1 \times \dots \times M_T}$ is given by

$$\boldsymbol{\mathcal{W}} = \sum_{r=1}^{R} \lambda_r \mathbf{v}_r^{(1)} \otimes \mathbf{v}_r^{(2)} \otimes \dots \otimes \mathbf{v}_r^{(T)} \tag{5}$$

where $\mathbf{v}_r^{(i)} \in \mathbb{R}^{M_i}$. The minimal R such that the decomposition (5) exists is called the *CP-rank* of the tensor. For a given R, finding the best rank-R approximation of a tensor $\boldsymbol{\mathcal{W}}$

$$\text{argmin}_{\boldsymbol{\mathcal{X}}: Rank_{CP}(\boldsymbol{\mathcal{X}}) \leq R} ||\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{X}}||_F \tag{6}$$

is an NP-hard and ill-posed problem. A connection between the CP rank of a tensor and its matrizications that we will need later on is given in the following lemma.

Lemma 2.1. *Let $\boldsymbol{\mathcal{W}}$ be a tensor with $rank_{CP}(\boldsymbol{\mathcal{W}}) = r$. Then for any matrizication of $\boldsymbol{\mathcal{W}}^{(s,t)}$ we have that rank$(\boldsymbol{\mathcal{W}}^{(s,t)}) \leq r$, where the ordinary matrix rank is assumed.*

The Tensor Train decomposition is given by

$$\boldsymbol{\mathcal{W}} = \sum_{r_1=1}^{R_1} ... \sum_{r_{T-1}=1}^{R_{T-1}} \mathbf{g}^{(1)}_{r_0 r_1} \otimes \mathbf{g}^{(2)}_{r_1 r_2} \otimes ... \otimes \mathbf{g}^{(T)}_{r_{T-1} r_T} \tag{7}$$

where $\mathbf{g}^{(t)}_{r_{t-1} r_t} \in \mathbb{R}^{M_t}$ and $r_0 = r_T = 1$ by definition. For a given t, if we gather the vectors $\mathbf{g}^{(t)}_{r_{t-1}, r_t}$ for all indices $r_{t-1} \in \{1, ..., R_{t-1}\}$, $r_t \in \{1, ..., R_t\}$ we obtain a three dimensional tensor $\boldsymbol{\mathcal{G}}^{(t)} \in \mathbb{R}^{M \times R_{t-1} \times R_t}$. The tensors $\{\boldsymbol{\mathcal{G}}^{(t)}\}_{t=1}^{T-1}$ are called *TT-cores* and the minimal values of $\{R_t\}_{t=1}^{T-1}$ such that the decomposition (7) exists are called *TT-ranks*.

If we replace $\otimes$ with $\otimes_\xi$ in (5) and (7) we obtain so-called *generalized tensor decompositions*.

3 Tensor Neural Networks

3.1 Machine Learning framework and setup

We consider a classification problem to be solved using a neural network on a dataset $\{(X^{(b)}, y^{(b)})\}_{b=1}^{L}$. We make the assumption that $X^{(b)}$ is a sequence of vectors :

$$X^{(b)} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(T)}),\ \mathbf{x}^{(i)} \in \mathbb{R}^N \tag{8}$$

and introduce a parametric feature map

$$f_\theta : \mathbb{R}^N \mapsto \mathbb{R}^M \tag{9}$$

where we suppose $M < N$, thus allowing us to obtain lower dimensional representations $\{f_\theta(\mathbf{x}^{(k)})\}_{k=1}^{T}$ of our features. Now, let $l(X)$ be the score functions of the network (the output of the last hidden layer) for a single class when the input data is X. We write

$$l(X) = \langle \boldsymbol{\mathcal{W}}, \boldsymbol{\Phi}(X) \rangle = (\text{vec}(\boldsymbol{\mathcal{W}}))^\top \text{vec}(\boldsymbol{\Phi}(X)) \tag{10}$$

where $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{M \times M \times ... \times M}$ is an order T tensor corresponding to the weights of our network to be trained and $\boldsymbol{\Phi}(X)$ is the so-called feature tensor:

$$\boldsymbol{\Phi}(X) = f_\theta(\mathbf{x}^{(1)}) \otimes f_\theta(\mathbf{x}^{(2)}) \otimes ... \otimes f_\theta(\mathbf{x}^{(T)}). \tag{11}$$

3.2 Tensor networks

We know show how tensor decompositions naturally appear in the previous setting by showing how different tensor decompositions of the weight tensor $\boldsymbol{\mathcal{W}}$ in (10) lead to neural networks with different architectures. We consider the two networks presented in Fig 1 implementing respectively shallow and recurrent architectures.

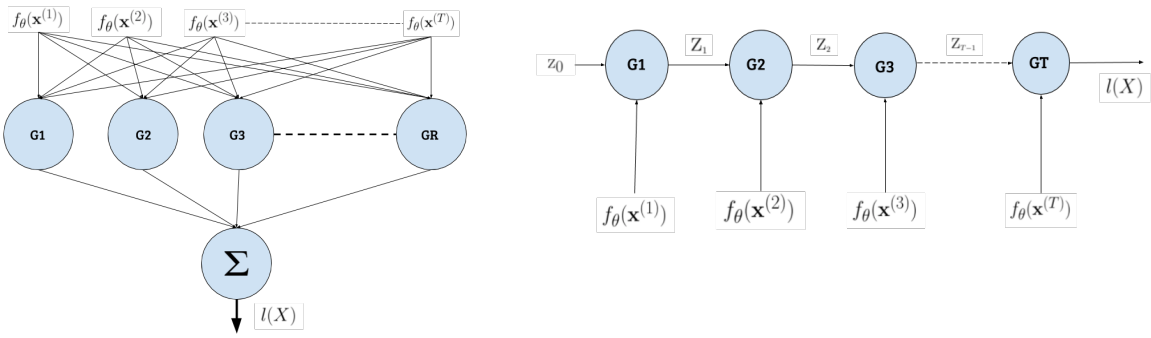

Figure 1: Representation of shallow (left) and recurrent (right) neural networks. Neurons in the hidden layer correspond to multilinear units. Arrows indicates that an element serves as input to the unit. The initial input $\mathbf{Z}_0$ is set to be equal to 1.

We start by defining *multilinear units*. For $\mathbf{x}_1 \in \mathbb{R}^{n_1}, ..., \mathbf{x}_T \in \mathbb{R}^{n_T}$, a multilinear unit $G \in \mathbb{R}^{n_1 \times \cdots \times n_T \times k}$ defines a multilinear map $G : \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_T} \mapsto \mathbb{R}^k$ as

$$G(\mathbf{x}_1, ..., \mathbf{x}_T) = \mathbf{z},$$

$$\mathbf{z}^j = \sum_{i_1,...,i_T} G^{i_1,...,i_T,j} \mathbf{x}_1^{i_1} \cdots \mathbf{x}_T^{i_T}.$$

Now consider the shallow neural network in Fig. 1 where each neuron G_r in the hidden layer is a multilinear unit taking as input the T features of our data and mapping into $\mathbb{R}$ (i.e k=1). Let us write this unit as a rank-1 tensor:

$$G_r = \mathbf{v}_r^{(1)} \otimes \cdots \otimes \mathbf{v}_r^{(T)}. \tag{12}$$

An explicit expression for the score function of this network is given by

$$\begin{aligned} l(X) &= \sum_{r=1}^{R} G\left(f_\theta(\mathbf{x}^{(1)}), ..., f_\theta(\mathbf{x}^{(T)})\right) \\ &= \sum_{r=1}^{R} \left(\sum_{i_1,...,i_T=1}^{M} [\mathbf{v}_{i_1,r}^{(1)} f_\theta(\mathbf{x}^{(1)})^{i_1}] ... [\mathbf{v}_{i_T,r}^{(T)} f_\theta(\mathbf{x}^{(T)})^{i_T}] \right) \\ &= \sum_{r=1}^{R} \langle f_\theta(\mathbf{x}^{(1)}), \mathbf{v}_r^{(1)} \rangle \otimes \cdots \otimes \langle f_\theta(\mathbf{x}^{(t)}), \mathbf{v}_r^{(T)} \rangle \\ &= \langle f_\theta(\mathbf{x}^{(1)}) \otimes \cdots \otimes f_\theta(\mathbf{x}^{(T)}), \sum_{r=1}^{R} \mathbf{v}_r^{(1)} \otimes \mathbf{v}_r^{(2)} \otimes \cdots \otimes \mathbf{v}_r^{(T)} \rangle. \end{aligned} \tag{13}$$

and hence can be represented as in (10) with $\boldsymbol{\mathcal{W}}$ given by (5). From this, it can readily be seen that the CP-decomposition of a weight tensor leads to a shallow neural network.

Taking into account potential constants, the general expression is

$$l(X) = \sum_{r=1}^{R} \lambda_r [\langle f_\theta(\mathbf{x}^{(1)}), \mathbf{v}_r^{(1)} \rangle \otimes ... \otimes \langle f_\theta(\mathbf{x}^{(t)}), \mathbf{v}_r^{(T)} \rangle]. \tag{14}$$

A similar derivation is now done for a network with recurrent-like architecture as in Fig.1. Given a vector $\mathbf{r} = (r_0, r_1, ..., r_{T-1}, r_T)$ of positive integers with $r_0 = r_T = 1$, we consider bilinear units $G_k \in \mathbb{R}^{r_{k-1} \times m \times r_k}$, taking as input $\mathbf{z}_{k-1}$ (output of the previous unit), the feature vector $f_\theta(\mathbf{x}^{(k)})$ and mapping into $\mathbb{R}^{r_k}$ for $k = 1, ..., T$. As before, we can compute the score function of such a network as

$$\begin{aligned}
l(X) &= G_T(\mathbf{z}_{T-1}, f_\theta(\mathbf{x}^{(T)})) \\
&= \sum_{r_{T-1}=1}^{R_{T-1}} \mathbf{z}_{T-1}^{r_{T-1}} \underbrace{\sum_{m=1}^{M} G_T^{r_{T-1},m} f_\theta(\mathbf{x}^{(T)})^m}_{=\langle f_\theta(\mathbf{x}^{(T)}), \mathbf{g}_{r_{T-1},r_T} \rangle} \\
&= \sum_{r_{T-1}=1}^{R_{T-1}} \mathbf{z}_{T-1}^{r_{T-1}} \langle f_\theta(\mathbf{x}^{(T)}), \mathbf{g}_{r_{T-1},r_T} \rangle \\
&= \sum_{r_{T-1}=1}^{R_{T-1}} \left[G_{T-1}\left(\mathbf{z}_{T-2}, f_\theta(\mathbf{x}^{(T-1)})\right) \right]_{r_{T-1}} \langle f_\theta(\mathbf{x}^{(T)}), \mathbf{g}_{r_{T-1},r_T} \rangle \\
&= \sum_{r_{T-1}=1}^{R_{T-1}} \left[\sum_{r_{T-2}=1}^{R_{T-2}} \sum_{m=1}^{M} G_{T-1}^{r_{T-2},m,r_{T-1}} \mathbf{z}_{T-2}^{r_{T-2}} f_\theta(\mathbf{x}^{(T-1)})^m \right] \langle f_\theta(\mathbf{x}^{(T)}), \mathbf{g}_{r_{T-1},r_T} \rangle \\
&= \sum_{r_{T-2}=1}^{R_{T-2}} \sum_{r_{T-1}=1}^{R_{T-1}} \mathbf{z}_{T-2}^{r_{T-2}} \langle f_\theta(\mathbf{x}^{(T)}), \mathbf{g}_{r_{T-1},r_T} \rangle \underbrace{\left[\sum_{m=1}^{M} G_{T-1}^{r_{T-2},m,r_{T-1}} f_\theta(\mathbf{x}^{(T-1)})^m \right]}_{\langle f_\theta(\mathbf{x}^{(T-1)}, \mathbf{g}_{r_{T-2},r_{T-1}} \rangle} \\
&= ... \\
&= \sum_{r_1=1}^{R_1} ... \sum_{r_{T-1}}^{R_{T-1}} \mathbf{z}_1^{r_1} \prod_{t=2}^{T-1} \langle f_\theta(\mathbf{x}^{(t)}), \mathbf{g}_{r_{t-1},r_t} \rangle \\
&= \sum_{r_1=1}^{R_1} ... \sum_{r_{T-1}}^{R_{T-1}} \prod_{t=2}^{T-1} \langle f_\theta(\mathbf{x}^{(t)}), \mathbf{g}_{r_{t-1},r_t} \rangle \underbrace{\left[\sum_{m=1}^{M} G_1^{r_0,m,r_1} f_\theta(\mathbf{x}^{(1)})^m \right]}_{=\langle f_\theta(\mathbf{x}^{(1)}), \mathbf{g}_{r_0,r_1} \rangle} \\
&= \sum_{r_1=1}^{R_1} ... \sum_{r_{T-1}=1}^{R_{T-1}} \prod_{t=1}^{T} \langle f_\theta(\mathbf{x}^{(t)}), \mathbf{g}_{r_{t-1}r_t}^{(t)} \rangle \\
&= \langle f_\theta(\mathbf{x}^{(1)}) \otimes \cdots \otimes f_\theta(\mathbf{x}^{(T)}), \sum_{r_1=1}^{R_1} ... \sum_{r_{T-1}=1}^{R_{T-1}} \mathbf{g}_{r_0 r_1}^{(1)} \otimes \mathbf{g}_{r_1 r_2}^{(2)} \otimes \cdots \otimes \mathbf{g}_{r_{T-1} r_T}^{(T)} \rangle.
\end{aligned} \tag{15}$$

Taking the matrices $\{\boldsymbol{\mathcal{G}}^{(t)}\}_{t=1}^{T}$ of the score function (using the same notation and as previously defined) to be TT-cores of the weight tensor of a network, we thus see that its TT-decomposition realizes the recurrent network shown in Fig.1. More concretely, a connection with recurrent neural networks can be seen in the following way. Define the hidden states

$$\mathbf{h}^{(1)} \in \mathbb{R}^{R_1} : \ \mathbf{h}^{(1)}_{r_1} = \langle f_\theta(\mathbf{x}^{(1)}), \mathbf{g}^{(1)}_{r_0 r_1} \rangle \tag{16}$$

$$\mathbf{h}^{(t)} \in \mathbb{R}^{R_t} : \ \mathbf{h}^{(t)}_{r_t} = \sum_{r_{t-1}=1}^{R_{t-1}} \langle f_\theta(\mathbf{x}^{(1)}), \mathbf{g}^{(1)}_{r_{t-1} r_t} \rangle \mathbf{h}^{(t-1)}_{r_{t-1}}, \ t = 2, ...T \tag{17}$$

Lemma 3.1.

$$l(X) = \mathbf{h}^{(T)} \in \mathbb{R}. \tag{18}$$

Proof. A direct computation gives

$$\begin{aligned}
l(X) &= \sum_{r_1=1}^{R_1} ... \sum_{r_{T-1}=1}^{R_{T-1}} \prod_{t=1}^{T} \langle f_\theta(x^{(t)}), g^{(t)}_{r_{t-1},r_t} \rangle \\
&= \sum_{r_1=1}^{R_1} ... \sum_{r_{T-1}=1}^{R_{T-1}} \prod_{t=2}^{T} \langle f_\theta(x^{(t)}), g^{(t)}_{r_{t-1},r_t} \rangle \langle f_\theta(x^{(1)}), g^{(1)}_{r_0,r_1} \rangle \\
&= \sum_{r_{T-1}=1}^{R_{T-1}} ... \sum_{r_1=0}^{R_1} \prod_{t=2}^{T} \langle f_\theta(x^{(t)}), g^{(t)}_{r_{t-1},r_t} \rangle h^{(1)}_{r_1} \\
&= \sum_{r_{T-1}=1}^{R_{T-1}} ... \sum_{r_2=0}^{R_2} \prod_{t=3}^{T} \langle f_\theta(x^{(t)}), g^{(t)}_{r_{t-1},r_t} \rangle \sum_{r_1=1}^{R_1} \langle f_\theta(x^{(2)}), g^{(2)}_{r_1,r_2} \rangle h^{(1)}_{r_1} \\
&= \sum_{r_{T-1}=1}^{R_{T-1}} ... \sum_{r_2=0}^{R_2} \prod_{t=3}^{T} \langle f_\theta(x^{(t)}), g^{(t)}_{r_{t-1},r_t} \rangle h^{(2)}_{r_2} \\
&= ... \\
&= \sum_{r_{T-1}=1}^{R_{T-1}} \prod_{t=2}^{T} \langle f_\theta(x^{(T)}), g^{(T)}_{r_{T-1},r_T} \rangle h^{(T-1)}_{r_{T-1}} \\
&= \mathbf{h}^{(T)}
\end{aligned} \tag{19}$$

□

Using the TT-cores, we can write (17) as

$$\begin{aligned}
\mathbf{h}^{(t)}_k &= \sum_{i,j} \mathcal{G}^{(t)}_{i,j,k} [f_\theta(x^{(t)}) \otimes \mathbf{h}^{(t-1)}]_{i,j} \\
&= g(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}, \Theta^{(t)}_{\boldsymbol{\mathcal{G}}}), \qquad\qquad k = 1, ..., R_t
\end{aligned} \tag{20}$$

for some function g and where $\Theta^{(t)}_{\boldsymbol{\mathcal{G}}}$ contains weights from $\boldsymbol{\mathcal{G}}$ and f_θ. We see that (17) is reminiscent of hidden states equations of recurrent neural networks.

The networks as defined so far have limited practical use since they only implement multiplicative nonlinearities. By using instead generalized tensor decompositions, replacing $\otimes$ in (14) and (20) with $\otimes_\xi$, allows for various nonlinearities. Henceforth, we consider

- **Generalized shallow networks with ξ-nonlinearity** with score function

$$l(X) = \sum_{r=1}^{R} \lambda_r \xi \left(\langle f_\theta(\mathbf{x}^{(1)}), \mathbf{v}_r^{(1)} \rangle, ..., \langle f_\theta(\mathbf{x}^{(T)}), \mathbf{v}_r^{(T)} \rangle \right) \tag{21}$$

- **Generalized RNN with ξ-nonlinearity** with hidden states

$$\mathbf{h}_k^{(t)} = \sum_{i,j} \mathcal{G}_{i,j,k}^{(t)} [\mathbf{C}^{(t)} f_\theta(x^{(t)}) \otimes_\xi \mathbf{h}^{(t-1)}]_{i,j} \tag{22}$$

  and score function

$$l(X) = \mathbf{h}^{(T)}. \tag{23}$$

where the introduced matrices $\mathbf{C}^{(t)}$ act on the input states in order to account for the possibility of generalized shallow networks being able to be represented as generalized RNNs of width 1.

Proposition 3.2. *If we replace the generalized outer product $\otimes_\xi$ in (22) with the standard outer product $\otimes$, we can subsume matrices $\mathbf{C}^t$ into tensors $\mathbf{G}^t$ without loss of generality.*

Proof.

$$\begin{aligned}
\mathbf{h}_k^{(t)} &= \sum_{i,j} \mathcal{G}_{i,j,k}^{(t)} [\mathbf{C}^{(t)} f_\theta(x^{(t)}) \otimes \mathbf{h}^{(t-1)}]_{i,j} \\
&= \sum_{i,j} \mathcal{G}_{i,j,k}^{(t)} \sum_{l} \mathbf{C}_{il}^{(t)} f_\theta(x^{(t)})_l \mathbf{h}_j^{(t-1)} \\
&= \sum_{l,j} \tilde{\mathcal{G}}_{l,j,k}^{(t)} f_\theta(x^{(t)})_l \mathbf{h}_j^{(t-1)}, \qquad \tilde{\mathcal{G}}_{l,j,k} = \sum_{i} \mathcal{G}_{i,j,k} \mathcal{C}_{i,l}^{(t)} \\
&= \sum_{l,j} \tilde{\mathcal{G}}_{l,j,k}^{(t)} [f_\theta(x^{(t)}) \otimes \mathbf{h}^{(t-1)}]_{i,j}
\end{aligned} \tag{24}$$

where we see that the TT-cores $\mathcal{G}^{(t)}$ are replaced by the cores $\tilde{\mathcal{G}^{(t)}}$. □

4 Universality and expressivity of ReLU tensor networks

The *expressive power* of a neural network is its ability to approximate functions. Depth efficiency refers to the known empirical fact that, for a fixed number of parameters, networks with deep architectures are typically more expressive then those with shallow ones. The notion of *universality* expresses the idea that, under mild conditions, neural networks are able to represent a large class of functions (in a sense to be made precise) The purpose of this section is to formalize this for generalized networks with ReLU non-linearity activation

$\xi(x, y) = \max(x, y)$. The two main results concern *universality* and *expressivity* of these networks. For this purpose, we first need to define grid tensors and prove certain statements about them.

4.1 Grid tensors

Given a *template* of fixed vectors $\mathbb{X} = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(M)}\}$, the *grid tensor* of $\mathbb{X}$ (for a given network with score function l) is defined to be the tensor of order T and dimension M in each mode with entries given by

$$\Gamma^{(l)}(\mathbb{X})_{i_1 i_2 ... i_T} = l(X), \quad X = \left(\mathbf{x}^{(i_1)}, \mathbf{x}^{(i_2)}, ..., \mathbf{x}^{(i_T)}\right) \tag{25}$$

$i_t \in \{1, ..., T\}$. Grid tensors will allow us to measure the expressiveness of generalized neural networks. This is done by asking if for a given tensor $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{M \times M ... \times M}$ of order T, there exits a generalized network with grid tensor equal to $\boldsymbol{\mathcal{H}}$. Define the matrix $\mathbf{F} \in \mathbb{R}^{M \times M}$ by

$$\mathbf{F} = [f_\theta(\mathbf{x}^{(1)}) \mid f_\theta(\mathbf{x}^{(2)}) \mid ... \mid f_\theta(\mathbf{x}^{(M)})]^\top \tag{26}$$

Using $\mathbf{F}$, we can write the grid tensor of a generalized shallow network in the following form

$$\mathbf{\Gamma}^l(\mathbb{X}) = \sum_{r=1}^{R} \lambda_r \left(\mathbf{F}\mathbf{v}_r^{(1)}\right) \otimes_\xi \left(\mathbf{F}\mathbf{v}_r^{(2)}\right) \otimes_\xi ... \otimes_\xi \left(\mathbf{F}\mathbf{v}_r^{(T)}\right) \tag{27}$$

since for $X = (\mathbf{x}^{(i_1)}, \mathbf{x}^{(i_2)}, ..., \mathbf{x}^{(i_T)})$ we have

$$\begin{aligned}
l(X) &= \sum_{r=1}^{R} \lambda_r \xi \left(\langle f_\theta(\mathbf{x}^{(i_1)}), \mathbf{v}_r^{(i_1)}\rangle, ..., \langle f_\theta(\mathbf{x}^{(i_T)}), \mathbf{v}_r^{(i_T)}\rangle\right) && (28)\\
&= \sum_{r=1}^{R} \lambda_r \left(\mathbf{F}\mathbf{v}_r^{(1)}\right)_{i_1} \otimes_\xi \left(\mathbf{F}\mathbf{v}_r^{(2)}\right)_{i_2} \otimes_\xi ... \otimes_\xi \left(\mathbf{F}\mathbf{v}_r^{(T)}\right)_{i_T} && (29)\\
&= \sum_{r=1}^{R} \lambda_r [\left(\mathbf{F}\mathbf{v}_r^{(1)}\right) \otimes_\xi ... \otimes_\xi \left(\mathbf{F}\mathbf{v}_r^{(T)}\right)]_{i_1, i_2 ... i_T} && (30)\\
&= \mathbf{\Gamma}^l(\mathbb{X})_{i_1 i_2 ... i_T}. && (31)
\end{aligned}$$

For generalized RNNs, a recursive expression of the grid tensor is given in the following proposition .

Proposition 4.1. *The grid tensor of a generalized RNN network has the following form:*

$\mathbf{\Gamma}^{(l,0)}(\mathbb{X}) = \mathbf{h}^{(0)} \in \mathbb{R}$

$\mathbf{\Gamma}^{(l,1)}(\mathbb{X})_{km_1} = \sum_{i,j} \mathcal{G}^{(1)}_{i,j,k} \left(\mathbf{C}^{(1)}\mathbf{F}^T \otimes_\xi \mathbf{\Gamma}^{l,0}\right)_{im_1 j} \in \mathbb{R}^{R_1 \times M}$

$\mathbf{\Gamma}^{(l,2)}(\mathbb{X})_{km_1 m_2} = \sum_{i,j} \mathcal{G}^{(2)}_{i,j,k} \left(\mathbf{C}^{(2)}\mathbf{F}^T \otimes_\xi \mathbf{\Gamma}^{l,1}\right)_{im_2 j m_1} \in \mathbb{R}^{R_2 \times M \times M}$

...

$$\mathbf{\Gamma}^{(l,T)}(\mathbb{X})_{km_1m_2...m_T} = \sum_{i,j} \mathcal{G}^{(T)}_{i,j,k} \left(\mathbf{C}^{(T)}\mathbf{F}^T \otimes_\xi \mathbf{\Gamma}^{l,1}\right)_{im_Tjm_1...m_{T-1}} \in \mathbb{R}^{1\times M\times M\times M\times M}$$

$$\mathbf{\Gamma}^{l}(\mathbb{X}) = \mathbf{\Gamma}^{(l,T)}(\mathbb{X})_{1,:,:,...,:}$$

Proof. Define $\mathbf{h}^{(q)}(m_1, ..., m_q) \in \mathbb{R}^{R_q}$ to be the hidden state at time step q for a generalized RNN with input $X = (\mathbf{x}^{(m_1)}, \mathbf{x}^{(m_2)}, ..., \mathbf{x}^{(m_q)})$. Then we claim that

$$\mathbf{\Gamma}^{(l,q)}(\mathbb{X})_{k,m_1,...,m_q} = \mathbf{h}^{(q)}_k(m_1, ..., m_q),\ q = 1, ..., T. \tag{32}$$

It is enough to prove (32) since then for $X = (\mathbf{x}^{i_1}, ..., \mathbf{x}^{(i_T)})$ an arbitrary sequence of templates, we will have

$$\mathbf{\Gamma}^{l}(\mathbb{X})_{i_1,...,i_T} = \mathbf{\Gamma}^{(l,T)}(\mathbb{X})_{1,i_1,...,i_T} = \mathbf{h}^{(T)}(i_1, ..., i_T) = l(X) \tag{33}$$

which is what we want to prove. We prove (32) by induction on q. For $q = 1$ we have

$$\begin{aligned}
\mathbf{\Gamma}^{(l,1)}(\mathbb{X})_{k,m_1} &= \sum_{i,j} \mathcal{G}^{(1)}_{i,j,k} \left(\mathbf{C}^{(1)}\mathbf{F}^T \otimes_\xi \mathbf{\Gamma}^{l,0}\right)_{im_1j} && (34)\\
&= \sum_{i,j} \mathcal{G}^{(1)}_{i,j,k} \xi\left([\mathbf{C}^{(1)}\mathbf{F}^T]_{im_1}, \mathbf{h}^{(0)}_j\right) && (35)\\
&= \sum_{i,j} \mathcal{G}^{(1)}_{i,j,k} \xi\left([\mathbf{C}^{(1)} f_\theta(\mathbf{x}^{(m_1)})]_i, \mathbf{h}^{(0)}_j\right) && (36)\\
&= \mathbf{h}^{(1)}_k(\mathbf{x}^{(m_1)}). && (37)
\end{aligned}$$

If (32) is true for $1 \le q < T$ then

$$\begin{aligned}
\mathbf{\Gamma}^{(l,q+1)}(\mathbb{X})_{k,m_1,...,m_{q+1}} &= \sum_{i,j} \mathcal{G}^{(q+1)}_{i,j,k} \left(\mathbf{C}^{(q)}\mathbf{F}^T \otimes_\xi \mathbf{\Gamma}^{l,q}\right)_{im_{q+1}jm_1,...,m_q} && (38)\\
&= \sum_{i,j} \mathcal{G}^{(q+1)}_{i,j,k} \xi\left([\mathbf{C}^{(q+1)}\mathbf{F}^T]_{im_{q+1}}, \mathbf{\Gamma}^{l,q}_{j,m_1,...,m_q}\right) && (39)\\
&= \sum_{i,j} \mathcal{G}^{(q+1)}_{i,j,k} \xi\left([\mathbf{C}^{(q+1)} f_\theta(\mathbf{x}^{(m_{q+1})})]_i, \mathbf{h}^{(q)}_j(m_1, ..., m_q)\right) && (40)\\
&= \mathbf{h}^{(q+1)}_k(\mathbf{x}^{(m_1)}, ..., \mathbf{x}^{(m_{q+1})}) && (41)
\end{aligned}$$

by our assumption that $\mathbf{h}^q(m_1, ..., m_q)$ is the hidden state at time step q. □

In what follows, we consider the parametric family of feature maps

$$\mathcal{F} = \{f_\theta(\mathbf{x}) = \sigma(\mathbf{w}^\top\mathbf{x} + b),\ \theta = (\mathbf{w}, b) \in \mathbb{R}^n \times \mathbb{R}\} \tag{42}$$

where $\sigma(\cdot)$ is sigmoidal (i.e, monotonic, with $\lim_{z\to-\infty}\sigma(z) = c, \lim_{z\to+\infty}\sigma(z) = C$ for some constants $c, C \in \mathbb{R}$) or the ReLU activation function, $\sigma(z) = \max(z, 0)$. Furthermore, as in [3], we consider the following conditions:

- **Continuity**: $f_\theta(\mathbf{x})$ is continuous with respect to θ and $\mathbf{x}$.
- **Non-degeneracy**: For any $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(T)} \in \mathbb{R}^n$ pairwise distinct, there exits $f_{\theta_1}, ..., f_{\theta_T} \in \mathcal{F}$ such that if we define $f(\mathbf{x}) = (f_{\theta_1}(\mathbf{x}), ..., f_{\theta_T}(\mathbf{x}))$, then $\mathbf{F} \in \mathbb{R}^{M\times M}$ defined in (26) is non-singular.

We then have the following result (claim 1, [3])

Proposition 4.2. *The parametric family $\mathcal{F}$ satisfies the non-degeneracy condition.*

The converse also holds, i.e, under mild conditions on the functions f, one can choose template vectors such that $\mathbf{F}$ is invertible (claim 2, [3]):

Proposition 4.3. *Let $f_{\theta_1}, ..., f_{\theta_M} : \mathbb{R}^n \mapsto \mathbb{R}$ be any linearly independent continuous functions. Then there exits $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(M)} \in \mathbb{R}^n$ such that the corresponding matrix $\mathbf{F}$ is non singular*

Proof. Write the determinant of $\mathbf{F}$ as a function of $(\mathbf{x}^{(1)}, ..., \mathbf{x}^{(M)})$:

$$\det F(\mathbf{x}^{(1)}, ..., \mathbf{x}^{(M)}) = \sum_{\delta\in S_M} sign(\delta) \prod_{i=1}^{M} f_{\theta_{\delta(i)}}(\mathbf{x}^{(i)}), \tag{43}$$

where S_M is the permutation group on $\{1, ..., M\}$ and $sign(\delta) \in \{+1, -1\}$. Hence $\det F$ is a linear combination of the product functions $\{(\mathbf{x}^{(1)}, ..., \mathbf{x}^{(M)}) \mapsto \prod_{i=1}^{M} f_{\theta_{d_i}}(\mathbf{x}^{(i)})\}_{d_1,...,d_M\in[M]}$ which are linearly independent since the $f_{\theta_1}, ..., f_{\theta_M}$ are linearly independent. It follows that $\det F$, seen as a function, is not identically zero i.e there exits $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(M)} \in \mathbb{R}^n$ such $\mathbf{F}$ is non singular. □

We now take f_θ to be an affine map followed by the ReLU activation : $f_\theta(\mathbf{x}) = \sigma(\mathbf{A}\mathbf{x}+b)$. The previous two proposition show that without loss of generality, we may assume $\mathbf{F}$ to be invertible. We fix some templates $\mathbb{X}$ and make the additional assumption that value of score functions outside of the grid tensor generated by $\mathbb{X}$ is irrelevant for classification. The next results shows that the set of grid tensors realized by generalized RNN is closed under taking linear combinations.

Lemma 4.4. *Given two generalized RNNs with grid tensors $\mathbf{\Gamma}^{l_A}(\mathbb{X})$, $\mathbf{\Gamma}^{l_A}(\mathbb{X})$ and arbitrary ξ-nonlinearity, there exists a generalized RNN with grid tensor $\mathbf{\Gamma}^{l_C}(\mathbb{X})$ satisfying*

$$\mathbf{\Gamma}^{l_C}(\mathbb{X}) = a\mathbf{\Gamma}^{l_A}(\mathbb{X}) + b\mathbf{\Gamma}^{l_B}(\mathbb{X}),\ \forall a, b \in \mathbb{R}. \tag{44}$$

Proof. Suppose the RNNs have respectively weight parameters

$$\Theta_A = \left(\{\mathbf{C}_A^{(t)}\}_{t=1}^T, \{\mathcal{G}_A^{(t)}\}_{t=1}^T \in \mathbb{R}^{L_A\times R_{t-1,A}\times R_{t,A}}\right), \tag{45}$$

and

$$\Theta_B = \left(\{\mathbf{C}_B^{(t)}\}_{t=1}^T, \{\mathcal{G}_A^{(t)}\}_{t=1}^T \in \mathbb{R}^{L_B\times R_{t-1,B}\times R_{t,B}}\right). \tag{46}$$

Then the network with weights

$$\boldsymbol{C}_C^{(t)} \in \mathbb{R}^{(L_A+L_B)\times M}$$

$$\boldsymbol{C}_C^{(t)} = \begin{bmatrix} \boldsymbol{C}_A^{(t)} \\ \boldsymbol{C}_B^{(t)} \end{bmatrix}$$

$$\mathcal{G}_C^{(1)} \in \mathbb{R}^{(L_A+L_B)\times 1\times (R_{1,A}+R_{1,B})}$$

$$[\mathcal{G}_C^{(1)}]_{i,:,:} = \begin{cases} \begin{bmatrix} [\mathcal{G}_A^{(1)}]_{i,:,:} & 0 \end{bmatrix} & i \in \{1, ..., L_A\} \\ \\ \begin{bmatrix} 0 & [\mathcal{G}_B^{(1)}]_{(i-L_A),:,:} \end{bmatrix} & i \in \{L_A+1, , ..., L_A+L_B\} \end{cases}$$

$$\mathcal{G}_C^{(t)} \in \mathbb{R}^{(L_A+L_B)\times (R_{t-1,A}+R_{t-1,B})\times (R_{t,A}+R_{t,B})},\ 1 < t < T$$

$$[\mathcal{G}_C^{(1)}]_{i,:,:} = \begin{cases} \begin{bmatrix} [\mathcal{G}_A^{(t)}]_{i,:,:} & 0 \\ 0 & 0 \end{bmatrix} & i \in \{1, ..., L_A\} \\ \\ \begin{bmatrix} 0 & 0 \\ 0 & [\mathcal{G}_B^{(t)}]_{i,:,:} \end{bmatrix} & i \in \{L_A+1, , ..., L_A+L_B\} \end{cases}$$

$$\mathcal{G}_C^{(T)} \in \mathbb{R}^{(L_A+L_B)\times (R_{T-1,A}+R_{T-1,B})\times 1}$$

$$[\mathrm{G}_C^{(1)}]_{i,:,:} = \begin{cases} \begin{bmatrix} a[\mathcal{G}_A^{(T)}]_{i,:,:} \\ 0 \end{bmatrix} & i \in \{1, ..., L_A\} \\ \\ \begin{bmatrix} 0 \\ b[\mathcal{G}_B^{(T)}]_{(i-L_A),:,:} \end{bmatrix} & i \in \{L_A+1, , ..., L_A+L_B\} \end{cases}$$

(47)

satisfies

$$\mathbf{h}_C^{(t)} = \begin{bmatrix} \mathbf{h}_A^{(t)} \\ \mathbf{h}_B^{(t)} \end{bmatrix},\ 0 < t < T \tag{48}$$

and

$$\mathbf{h}_C^{(T)} = a\mathbf{h}_A^{(T)} + b\mathbf{h}_A^{(T)} \tag{49}$$

which directly shows the claim. Indeed, a direct computation gives

$$
\begin{aligned}
(\mathbf{h}_C^{(1)})_k &= \sum_i (\mathcal{G}_C^{(1)})_{i,1,k} \xi \left([C_C^{(1)} f_\theta(\mathbf{x}^{(1)}]_i, (\mathbf{h}_C^{(0)})] \right) \\
&= \sum_{i=1}^{L_A} (\mathcal{G}_A^{(1)})_{i,1,k} \xi \left([C_A^{(1)} f_\theta(\mathbf{x}^{(1)})]_i, (\mathbf{h}_A^{(0)})] \right) \mathbb{1}\{k \leq R_{1,A}\} + \\
&\quad \sum_{i=L_A+1}^{L_A+L_B} (\mathcal{G}_B^{(1)})_{i,1,k} \xi \left([C_B^{(1)} f_\theta(\mathbf{x}^{(1)})]_i, (\mathbf{h}_B^{(0)})] \right) \mathbb{1}\{R_{1,A}+1 \leq k \leq R_{1,A}+R_{1,B}\} \\
&= (\mathbf{h}_A^{(1)})_k \mathbb{1}_{\{k \leq R_{1,A}\}} + (\mathbf{h}_B^{(1)})_k \mathbb{1}_{\{R_{1,A}+1 \leq k \leq R_{1,A}+R_{1,B}\}}
\end{aligned} \tag{50}
$$

and by an induction argument we get for $1 \leq t < T$

$$
\begin{aligned}
(\mathbf{h}_C^{(t)})_k &= \sum_{i,j} (\mathcal{G}_C^{(t)})_{i,t,k} \xi \left([C_C^{(t)} f_\theta(\mathbf{x}^{(t)}]_i, (\mathbf{h}_C^{(t-1)})_j] \right) \\
&= \sum_{i=1}^{L_A} \sum_{j=1}^{R_{t-1,A}} (\mathcal{G}_A^{(t)})_{i,j,k} \xi \left([C_A^{(t)} f_\theta(\mathbf{x}^{(t)})]_i, (\mathbf{h}_A^{(t-1)})_j] \right) \mathbb{1}\{k \leq R_{1,A}\} + \\
&\quad \sum_{i=L_A+1}^{L_A+L_B} \sum_{j=R_{t-1,A+1}}^{R_{t-1,A}+R_{t-1,B}} \mathcal{G}_B^{(t)})_{i,j,k} \xi \left([C_B^{(t)} f_\theta(\mathbf{x}^{(t)})]_i, (\mathbf{h}_B^{(t-1)})] \right) \mathbb{1}\{R_{1,A}+1 \leq k \leq R_{1,A}+R_{1,B}\} \\
&= (\mathbf{h}_A^{(t)})_k \mathbb{1}_{\{k \leq R_{1,A}\}} + (\mathbf{h}_B^{(t)})_k \mathbb{1}_{\{R_{1,A}+1 \leq k \leq R_{1,A}+R_{1,B}\}}
\end{aligned} \tag{51}
$$

and for $t = T$ we get (49) by the exact same argument. □

4.2 Universality of ReLU tensor networks

We now move on to proving the universality result. For this, some intermediate results will be needed.

Proposition 4.5. *For any associative and commutative binary operator ξ, an arbitrary generalized rank 1 shallow network with ξ-nonlinearity can be represented as a generalized RNN with unit ranks $R_1 = ... = R_{T-1} = 1$ and ξ-nonlinearity.*

Proof. If the parameters of the generalized shallow network are $\mathbf{\Theta} = \{\lambda, \{\mathbf{v}^{(t)}\}_{t=1}^T\}$ then the generalized RNN with weights

$$
\mathbf{C}^{(t)} = \left(\mathbf{v}^{(t)}\right)^T \in \mathbb{R}^{1 \times M} \tag{52}
$$

$$
\mathcal{G}^{(t)} = 1, \ t < T \tag{53}
$$

$$
\mathcal{G}^{(T)} = \lambda \tag{54}
$$

satisfies the desired property since

$$
\begin{aligned}
\mathbf{h}^{(t)} &= \mathcal{G}^{(t)} \xi \left([\mathbf{C}^{(t)} f_\theta(\mathbf{x}^{(t)}), \mathbf{h}^{(t-1)}] \right) \\
&= \xi \left(\langle f_\theta(\mathbf{x}^{(t)}), \mathbf{v}^{(t)} \rangle, \mathbf{h}^{(t-1)} \right), \ t = 1, ..., T-1
\end{aligned} \tag{55}
$$

and the corresponding score function

$$\begin{aligned}
\mathbf{h}^{(T)} &= \lambda\xi\left(\langle f_\theta(\mathbf{x}^{(T)}, \mathbf{v}^{(T)}\rangle, \mathbf{h}^{(T-1)}\right) \\
&= \lambda\xi\left(\langle f_\theta(\mathbf{x}^{(T)}), \mathbf{v}^{(T)}\rangle, [\xi\left(\langle f_\theta(\mathbf{x}^{(T-1)}), \mathbf{v}^{(T-1)}\rangle, \mathbf{h}^{(T-2)}\right)]\right) \\
&= \lambda\xi\left(\langle f_\theta(\mathbf{x}^{(T)}), \mathbf{v}^{(T)}\rangle, \langle f_\theta(\mathbf{x}^{(T-1)}), \mathbf{v}^{(T-1)}\rangle, \mathbf{h}^{(T-2)}\right) \\
&= ... \\
&= \lambda\xi\left(\langle f_\theta(\mathbf{x}^{(T)}), \mathbf{v}^{(T)}\rangle, ..., \langle f_\theta(\mathbf{x}^{(1)}), \mathbf{v}^{(1)}\rangle\right)
\end{aligned} \tag{56}$$

which equals the score function realized by a generalized shallow network with weights $\mathbf{\Theta}$. □

Lemma 4.6. *Let $\boldsymbol{\varepsilon}^{(j_1,j_2,...,j_T)}$ be an arbitrary one-hot tensor defined as*

$$\boldsymbol{\varepsilon}^{j_1,j_2,...,j_T}_{(i_1,i_2,...,i_T)} = \begin{cases} 1 & j_t = i_t \ \forall t \in \{1, ..., T\}, \\ 0 & \textit{otherwise.} \end{cases} \tag{57}$$

Then there exits a generalized RNN with rectifier nonlinearities such that its grid tensor satisfies

$$\mathbf{\Gamma}^l(\mathbb{X}) = \boldsymbol{\varepsilon}^{(j_1,j_2,...,j_T)}. \tag{58}$$

Proof. We first prove the result for generalized shallow networks by explicitly constructing construction such a network for $\mathbf{F} = \mathbf{I}$, the identity. Consider a generalized shallow network defined by the weights

$$\mathbf{\Theta} = \left(\{\lambda_r\}_{r=1}^{R=2}, \{\mathbf{v}_r^{(t)}\}_{r=1,t=1}^{R=2,T} \in \mathbb{R}^m\right) \tag{59}$$

with $\lambda_1 = 1, \lambda_2 = -1, \mathbf{v}_1^{(1)} = \mathbf{v}_1^{(2)} = ... = \mathbf{v}_1^{(T)} = \mathbb{1}$ the vector with all values equal to 1 and $\mathbf{v}_2^{(t)} = e_{j_t}$, the vector with all values equal to 1 except the value in position j_t equal to 0. The grid tensor realized by this network is then given by

$$\begin{aligned}
\Gamma^l(\mathbb{X}) &= \sum_{r=1}^{2} \lambda_r \left(\mathbf{F}\mathbf{v}_r^{(1)}\right) \otimes_\xi ... \otimes_\xi \left(\mathbf{F}\mathbf{v}_r^{(T)}\right) \\
&= [(\mathbf{F}\mathbb{1}) \otimes_\xi ... \otimes_\xi (\mathbf{F}\mathbb{1})] - [(\mathbf{F}e_{j_1}) \otimes_\xi ... \otimes_\xi (\mathbf{F}e_{j_T})].
\end{aligned} \tag{60}$$

Hence,

$$\begin{aligned}
\Gamma^l(\mathbb{X})_{i_1,...,i_T} &= \max\{\mathbb{1}_{i_1}, ..., \mathbb{1}_{i_T}, 0\} - \max\{(e_{j_1})_{i_1}, ..., (e_{j_T})_{i_T}, 0\} \\
&= \mathbb{1}_{\{(i_1,...,i_T)=(j_1,...,j_T)\}},
\end{aligned} \tag{61}$$

which proves the claim for generalized shallow networks. Clearly, the constructed rank 2 generalized shallow network can be written as a linear combination of rank 1 network generalized shallow networks. By proposition (4.5), each of these rank 1 shallow networks can be represented in a form of generalized RNN. Lemma 4.4 gives us the existence of the desired generalized RNN. □

Now that we have these two results, we can state and prove the following theorem.

Theorem 4.7. *Let $\mathcal{H} \in \mathbb{R}^{M\times\ldots\times M}$ be an arbitrary tensor of order T. Then there exists a generalized shallow network and a generalized RNN with rectifier non linearity $\xi(x,y) = max(x,y,0)$ such that the grid tensor of each of the networks coincides with $\mathcal{H}$*

Proof. By lemma 4.6, for each basis tensor $\boldsymbol{\varepsilon}^{(i_1,i_2,\ldots,i_T)}$ there exists a generalized RNN with rectifier nonlinearities such that its grid tensor equals $\boldsymbol{\varepsilon}$. By lemma (4.4) we can construct a RNN with grid tensor

$$\Gamma^{(l)}(\mathbb{X}) = \sum_{i_1,i_2,\ldots,i_T} \mathcal{H}_{i_1,i_2,\ldots,i_T}\boldsymbol{\varepsilon}^{(i_1,i_2,\ldots,i_T)} = \mathcal{H}. \tag{62}$$

Using 4.6 and the fact that the collection of grid tensors of shallow generalized networks is closed under taking linear combinations, the exact same proof as above proves the claim for generalized shallow networks. □

Theorem (4.7) also holds for product nonlinearities.

Proposition 4.8. *Theorem (4.7) holds with product nonlinearity $\xi(x,y) = xy$*

Proof. Fix a template of vectors $\mathbb{X} = \{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(T)}\}$ such that $\mathbb{F}$ is invertible and consider a network with weight tensor

$$\hat{\boldsymbol{\mathcal{H}}}_{i_1,i_2,\ldots,i_T} = \sum_{j_1,\ldots,j_T} \boldsymbol{\mathcal{H}}_{j_1,\ldots,j_T}\mathbf{F}^{-1}_{j_1,i_1}\ldots\mathbf{F}^{-1}_{j_T,i_T}. \tag{63}$$

Then a direct computation shows that the grid tensor realized by this network using non multiplicative nonlinearity is given by

$$\begin{aligned}
\Gamma(\mathbb{X})_{i_1,\ldots,i_T} &= l((\mathbf{x}^{(i_1)},\ldots,\mathbf{x}^{(i_T)})) \\
&= \sum_{l_1,\ldots,l_T=1}^{M} \hat{\boldsymbol{\mathcal{H}}}_{l_1,l_2,\ldots,l_T}\boldsymbol{\Phi}_{l_1,l_2,\ldots,l_T}((\mathbf{x}^{(i_1)},\ldots,\mathbf{x}^{(i_T)})) \\
&= \sum_{l_1,\ldots,l_T=1}^{M} \left(\sum_{j_1,\ldots,j_T} \boldsymbol{\mathcal{H}}_{j_1,\ldots,j_T}\mathbf{F}^{-1}_{j_1,l_1}\ldots\mathbf{F}^{-1}_{j_T,l_T}\right) f_\theta(\mathbf{x}^{(i_1)})_{l_1}\ldots f_\theta(\mathbf{x}^{(i_T)})_{l_T} \\
&= \boldsymbol{\mathcal{H}}_{i_1,\ldots,i_T}\left(\sum_{l_1,\ldots,l_T=1}^{M} \mathbf{F}^{-1}_{i_1,l_1}\ldots\mathbf{F}^{-1}_{i_T,l_T} f_\theta(\mathbf{x}^{(i_1)})_{l_1}\ldots f_\theta(\mathbf{x}^{(i_T)})_{l_T}\right) + \underbrace{[\text{remainder terms}]}_{=0} \\
&= \boldsymbol{\mathcal{H}}_{i_1,\ldots,i_T}\underbrace{\sum_{l_1=1}^{M}\mathbf{F}^{-1}_{i_1,l_1}\mathbf{F}_{l_1,i_1}\ldots\left(\sum_{l_T=1}^{M}\mathbf{F}^{-1}_{i_T,l_T}\mathbf{F}_{l_T,i_T}\right)}_{=1} = \boldsymbol{\mathcal{H}}_{i_1,\ldots,i_T}
\end{aligned} \tag{64}$$

where the remainder terms corresponds to a double sum over indices $(j_1,\ldots,j_T) \neq (i_1,\ldots,i_T)$ and is thus equal to 0 (using the fact that $\mathbf{F}\mathbf{F}^{-1} = \mathbf{I}$). Thus the network with weight tensor $\hat{\boldsymbol{\mathcal{H}}}$ and multiplicative nonlinearity has grid tensor equal to $\boldsymbol{\mathcal{H}}$. By taking the CP-decomposition and TT-decomposition of $\hat{\boldsymbol{\mathcal{H}}}$, we respectively get generalized CP and generalized TT networks with multiplicative nonlinearity which shows the claim. □

4.3 Expressivity of ReLU tensor networks

The next objectif is to prove two expressivity results. The first says generalized RNN are more expressive then their shallow counterparts.

Theorem 4.9. *For every value of R there exits a generalized RNN with ranks $\leq R$ and rectifier nonlinearity which is exponentially more efficient than shallow networks, i.e, the corresponding grid tensor may be realized only by a shallow network with rectifier nonlinearity of width at least $\frac{2}{MT} min(M,R)^{T/2}$.*

Proving theorem 4.9 will require the following two lemmas, the first given without proof (claim 9, [3]):

Lemma 4.10. *Let $\mathbf{\Gamma}^l(\mathbb{X})$ be a grid tensor generated by a generalized shallow network of rank R and $\xi(x,y) = max(x,y,0)$. Then*

$$rank[\mathbf{\Gamma}^l(\mathbb{X})]^{(s_{odd},t_{even})} \leq R\frac{TM}{2}, \tag{65}$$

where the ordinary matrix rank is assumed

Lemma 4.11. *Without loss of generality, assume that $\mathbf{x_i} = \mathbf{e}_i$. Let $\mathbf{1}^{(p,q)}$ denote the matrix of size $p \times q$ with each entry being 1, $\mathbf{I}^{(p,q)}$ the matrix of size $p \times q$ with $\mathbf{I}^{(p,q)}_{ij} = \delta_{ij}$ and $\mathbf{b} = [1 - min(M,R), \mathbf{0}^T_{R-1}] \in \mathbb{R}^1$. Consider the generalized RNN with $\xi(x,y) = max(x,y,0)$*

$$\mathbf{C}^{(t)} = \begin{cases} \mathbf{1}^{(M,M)} - \mathbf{I}^{M,M} & t \text{ odd} \\ \mathbf{1}^{(M+1,M)} - \mathbf{I}^{(M+1,M)} & t \text{ even} \end{cases} \tag{66}$$

$$\boldsymbol{\mathcal{G}}^{(t)} = \begin{cases} \mathbf{I}^{(M,R)} \in \mathbb{R}^{(M\times 1\times R)} & t \text{ odd} \\ \begin{bmatrix} \mathbf{1}^{(M,R)} \\ \mathbf{b} \end{bmatrix} \in \mathbb{R}^{(M+1)\times R\times 1} & t \text{ even} \end{cases} \tag{67}$$

Then the corresponding grid tensor satisfies

$$rank[\mathbf{\Gamma}^l(\mathbb{X})]^{(s_{odd},t_{even})} \geq min(M,R)^{T/2}\frac{TM}{2} \tag{68}$$

where the ordinary matrix rank is assumed.

Proof. First we show that for inputs of the form $X = (\mathbf{x}_{i_1}, \mathbf{x}_{i_1}, ..., \mathbf{x}_{i_{T/2}}, \mathbf{x}_{i_{T/2}})$ with $i_1 \leq R, ..., i_{T/2} \leq R$ we have $l(X) = 0$ and $l(X) = 1$ in all other cases. Let $\mathbf{e}_{j_1}, \mathbf{e}_{j_2}$ be respectively the first and second input vectors of the network. For the first input vector $\mathbf{e}_{j_1}$ at time step 1 we have $\left[\mathbf{C}^{(1)}\mathbf{e}_{j_1}\right]_i = \mathbb{1}_{\{i\neq j_1\}}$ and so the first hidden state is given by

$$\mathbf{h}^{(1)}_k = \sum_{i=1}^{\min(R,M)} \mathbb{1}_{\{i=k\}}\xi(\left[\mathbf{C}^{(1)}\mathbf{e}_{j_1}\right]_i, 0) = \max(\mathbb{1}_{\{k\neq j_1\}}, 0) = \mathbb{1}_{\{k\neq j_1\}}, \quad k = 1, ..., R \tag{69}$$

i.e the first hidden state corresponds to the bitwise negative of the input. Similarly, for the second time step we have $\left[\mathbf{C}^{(2)}\mathbf{e}_{j_2}\right]_i = \mathbb{1}_{\{i\neq j_2\}}$ for $i = 1, ..., M$, $\left[\mathbf{C}^{(2)}\mathbf{e}_{j_2}\right]_{M+1} = \mathbb{1}_{\{j_2\neq M\}}$ and the hidden state at time 2 is

$$\begin{aligned}\mathbf{h}^{(2)} &= \sum_{i=1}^{\min(M,R)} \xi(\left[\mathbf{C}^{(2)}\mathbf{e}_{j_2}\right]_i, \mathbf{h}_i^{(1)}) + (1-\min(R,M))\xi(\left[\mathbf{C}^{(2)}\mathbf{e}_{j_2}\right]_{M+1}, \mathbf{h}_1^{(1)}) \\ &= \sum_{i=1}^{\min(M,R)} \max(\mathbb{1}_{\{i\neq j_2\}}, \mathbb{1}_{\{i\neq j_1\}}, 0) + (1-\min(R,M))\max(\mathbb{1}_{\{M\neq j_2\}}, \mathbb{1}_{\{1\neq j_1\}}, 0)\end{aligned} \tag{70}$$

If $j_1 = j_2$ (i.e $\mathbf{e}_{j_1} = \mathbf{e}_{j_2}$) then the previous line is equal to $(\min(M,R)-1)+(1-\min(M,R) = 0$ and the process continues. Otherwise, we get $\min(M,R) + (1-\min(M,R) = 1$ and it is not hard to see that this will be the final output of the network. In other words, this networks measures pairwise similarity and the claim mentionned at the beginning of the proof is verified. If follows that $[\Gamma(\mathbb{X})^l]^{(s_{\text{odd}},t_{\text{even}})}$ is a matrix with $\min(M,R)^{T/2}$ entries on the diagonal equal to 0 and the remaining entries equal to 1. The rank of such a matrix is $R^{T/2}+1$ if $R < M$ and $M^{T/2}$ otherwise, proving the lemma. □

The proof of theorem 4.9 can now be completed and uses the matricization into even and odd modes of the grid tensors. It relies on the idea that for architectures realizing the same grid tensor, upper and lower bounds on the matrix rank of a matricization allow to compare the sizes of the corresponding architectures.

Proof. If we consider the example constructed in the proof of lemma 4.11, then lemma 4.10 tells us that the rank if the shallow network with rectifier nonlinearity which is able to represent the same grid tensor is at least $\frac{2}{TM}\min(M,R)^{T/2}$ □

The next theorem shows that this expressivity property of generalized RNN is however limited.

Theorem 4.12 (Expressivity 2)**.** *For every value of R, there exists an open set of generalized RNNs with rectifier non-linearity $\xi(x,y) = max(x,y,0)$, such that for each RNN in this open set, the corresponding grid tensor can be realized by a rank 1 shallow network with rectifier non-linearity.*

Proof. Denote by $\boldsymbol{I}^{(p,q)}$ the matrix of size $p\times q$ such that $\boldsymbol{I}^{(p,q)} = \delta_{i,j}$ and $\boldsymbol{a}^{(p_1,...,p_d)}$ the tensor of shape $p_1 \times ... \times p_d$ with every entry being $\boldsymbol{a}$. Consider a generalized RNN with weights

$$\mathbf{C}^{(t)} = \left(\mathbf{F}^{(T)}\right)^{-1},\ \boldsymbol{\mathcal{G}}^{(t)} = \begin{cases}\mathbf{2}^{(M,1,R)} & t=1\\ \mathbf{1}^{(M,R,R)} & t=2,...,T-1\\ \mathbf{1}^{(M,R,1)} & t=T\end{cases} \tag{71}$$

The grid tensor realized by this generalized RNN is $\mathbf{\Gamma}^{(l)}(\mathbb{X}) = \mathbf{2}(\boldsymbol{MT})^{\boldsymbol{T}-\mathbf{1}(M,...,M)}$. Indeed, we have

$$\mathbf{\Gamma}^{(l,0)}(\mathbb{X}) = 0$$

$$\mathbf{\Gamma}^{(l,1)}(\mathbb{X})_{km_1} = \sum_i \mathcal{G}^{(1)}_{i,1,k}\left(\mathbf{I}^{(M,M)}\otimes_\xi 0\right)_{im_1j} = \sum_i 2\times\max\{\mathbf{I}^{(M,M)}_{im_1}, 0\} = 2$$

$$\mathbf{\Gamma}^{(l,2)}(\mathbb{X})_{km_1m_2} = \sum_{i,j} (\mathbf{I}^{(M,M)} \otimes_\xi \mathbf{\Gamma}^{(l,1)})_{im_2jm_1} = \sum_{i,j} 2 = 2(MR)$$

...

and so on where we see the claim clearly holds. This tensor can be represented by a rank 1 generalized shallow network and we show that this still holds when adding a perturbation collectively denoted by $\boldsymbol{\varepsilon}$.

$$\mathbf{\Gamma}^{(l,0)}(\mathbb{X}) = 0 \in \mathbb{R},$$

$$\mathbf{\Gamma}^{(l,1)}(\mathbb{X})_{km_1} = \sum_{i,j} \mathcal{G}^{(1)}_{i,j,k} \left((\mathbf{I}^{(M,M)} + \boldsymbol{\varepsilon}) \otimes_\xi 0 \right)_{im_1j} \in \mathbb{R}^{R_1 \times M} = \mathbf{1} \otimes (\mathbf{2} + \boldsymbol{\varepsilon}),$$

$$\mathbf{\Gamma}^{(l,2)}(\mathbb{X})_{km_1m_2} = \sum_{i,j} \mathcal{G}^{(2)}_{i,j,k} \left((\mathbf{I}^{(M,M)} + \boldsymbol{\varepsilon}) \otimes_\xi \mathbf{\Gamma}^{l,1} \right)_{im_2jm_1} = \mathbf{1} \otimes (\mathbf{2MR} + \boldsymbol{\varepsilon}) \otimes \mathbf{1},$$

...

$$\mathbf{\Gamma}^{(l,T)}(\mathbb{X})_{km_1m_2...m_T} = 1 \otimes (\mathbf{2}(\mathbf{MR})^{\mathbf{T-1}} + \boldsymbol{\varepsilon}) \otimes \mathbf{1} ... \otimes \mathbf{1},$$

$$\mathbf{\Gamma}^{l}(\mathbb{X}) = \mathbf{\Gamma}^{(l,T)}(\mathbb{X})_{1,:,:,...,:} = (\mathbf{2}(\mathbf{MR})^{\mathbf{T-1}} + \boldsymbol{\varepsilon}) \otimes \mathbf{1} ... \otimes \mathbf{1},$$

using the fact that $\boldsymbol{\mathcal{A}} \otimes_\xi \boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{A}} \otimes \mathbf{1}$ when each element of $\boldsymbol{\mathcal{A}}$ is greater or equal to $\boldsymbol{\mathcal{B}}$. We claim that this grid tensor can be represented by a rank 1 shallow network with weight setting

$$\lambda = 1, \ \mathbf{v}_t = \begin{cases} \mathbf{F}^{-1}(\mathbf{2}(\mathbf{MR})^{\mathbf{T-1}} + \boldsymbol{\varepsilon}) & t = 1 \\ 0 & t > 1 \end{cases} \tag{72}$$

Indeed, the corresponding grid tensor $\tilde{\mathbf{\Gamma}}(\mathbb{X})$ satisfies

$$\begin{aligned} \tilde{\mathbf{\Gamma}}(\mathbb{X})_{i_1,...,i_T} &= \lambda \cdot \max\{\langle f_\theta(\mathbf{x})^{(i_1)}, \mathbf{v}_1^{(1)} \rangle, 0\} \\ &= \max\{\langle f_\theta(\mathbf{x})^{(i_1)}, \mathbf{F}^{-1}(\mathbf{2}(\mathbf{MR})^{\mathbf{T-1}} + \boldsymbol{\varepsilon}) \rangle, 0\} \\ &= \sum_{m=1}^{M} \mathbf{F}_{i_1m} \left(\sum_{j=1}^{M} (\mathbf{F}^{-1})_{mj} \left(\mathbf{2}(\mathbf{MR})^{\mathbf{T-1}} + \boldsymbol{\varepsilon} \right)_j \right) \\ &= \left(\mathbf{2}(\mathbf{MR})^{\mathbf{T-1}} + \boldsymbol{\varepsilon} \right)_{i_1} \\ &= \mathbf{\Gamma}(\mathbb{X})_{i_1,...,i_T}. \end{aligned} \tag{73}$$

□

5 Numerical Experiments

The numerical experiment conducted mainly consisted in implementing the generalized networks and attempting to reproduce the results in [5] and [6]. Part of the work done consisted in implementing the non-generalized versions of these networks. Since they are less general and of less interest, we only present the numerical results for the generalized networks. The purpose of these experiments is to asses both the performance and expressivity of these networks.

5.1 Methodology

We used four datasets for our experiments: toy datasets "Moons" and "Circles" ([10]), visual datasets MNIST ([8]), CIFAR10 ([7]) and sentiment analysis dataset IMDB ([9]). Proceeding as in [6] we modify our datasets (after performing standard feature prepossessing such as standardization) so that they have the desired sequential structure (this modification is only applied to the visual datsets). One natural approach consists in extracting patches from the image along each chanel and stacking them into a matrix. Adopting this point of vue, each column corresponds to a time step. Figure (2) illustrates this process. As feature map, we take f_θ to be an affine map followed by a ReLU activation.

Generalized RNNs require to specify the choice of the initial hidden state $\mathbf{h}^{(0)}$.We initialize $\mathbf{h}^{(0)}$ as unit of the considered non-linearity, i.e an element u such that $\xi(x, y, u) = \xi(x, y) \forall x, y \in \mathbb{R}$. For instance, $u = 0$ for $\xi(x, y) = \max\{x, y, 0\}$.

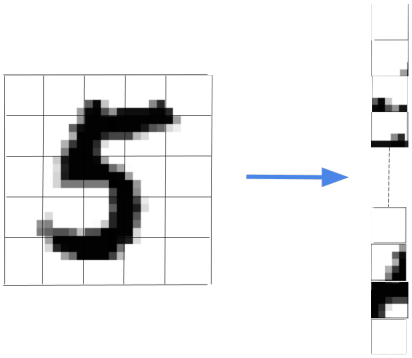

Figure 2: Illustration of extraction procedure producing, turning a (32×32) MNIST digit into a (64×16) image and a $(32 \times 32 \times 3)$ CIFAR10 image into a (64×48) image.

We implement the networks using Tensorflow ([1]). The parameters to be optimized for the shallow networks are

$$\Theta = \left(\{\lambda_r\}_{r=1}^{R} \in \mathbb{R}, \{\mathbf{v}_r^{(t)}\}_{r=1,t=1}^{R,T} \in \mathbb{R}^M \right)$$

and for generalized RNN

$$\Theta = \left(\{\mathbf{C}^{(t)}\}_{r=1}^{R} \in \mathbb{R}^{L \times M}, \{\boldsymbol{\mathcal{G}}^{(t)}\}_{t=1}^{T} \in \mathbb{R}^{L \times R_{t-1} \times R_t} \right).$$

In both cases we use the Adam optimizer and categorical cross-entropy loss function. In order to achieve the best test accuracy, the choice of most hyper-parameters such as the learning rate, choice of rank of the networks and number of training epochs depended on the specific choice of non linearity being used.

The choice of the initial initialization has a significant impact on the performance of the networks. A widely applied heuristic is for the distribution of the weights to be normally distributed with mean 0 and variance 1. While obtaining the distribution of the weight tensor is not straightforward, with certain non-linearities we can still initialize the weights such as to obtain the desired expectation and variance. We derive the following results.

Proposition 5.1. *Assume product non-linearity $\xi(x,y) = xy$. Then the following initialization schemes*

- ***Generalized shallow network***: $\lambda_r \overset{iid}{\sim} \mathcal{N}(0,1)$, $\mathbf{v}_r^{(t)} \overset{iid}{\sim} \mathcal{N}_M(0, (\frac{1}{R})^{1/T} I_M)$
- ***Generalized RNN***: $\mathbf{g}^{(t)}_{r_{t-1}r_t} \overset{iid}{\sim} \mathcal{N}_L(0, (1/\prod_{t=1}^{T} R_t)^{1/T} I_L)$

give weight tensors with zero expectation and unit variance.

Proof. For generalized shallow networks, an element of the weight tensor is given by

$$\boldsymbol{\mathcal{W}}_{i_1 i_2 \dots i_T} = \sum_{r=1}^{R} \lambda_r \mathbf{v}^{(1)}_{r_{i_1}} \dots \mathbf{v}^{(T)}_{r_{i_T}},$$

hence using standard formulas for the expectation and variance of products of independent random variables we get

$$\mathbb{E}[\boldsymbol{\mathcal{W}}_{i_1 i_2 \dots i_T}] = \sum_{r=1}^{R} \underbrace{\mathbb{E}[\lambda_r]}_{=0} \mathbb{E}[\mathbf{v}^{(1)}_{r_{i_1}}] \dots \mathbb{E}[\mathbf{v}^{(T)}_{r_{i_T}}] = 0$$

$$\begin{aligned}
\mathrm{Var}\left(\boldsymbol{\mathcal{W}}_{i_1 i_2 \dots i_T}\right) &= \sum_{r=1}^{R} \left[\left(\mathrm{Var}(\lambda_r) + \mathbb{E}[\lambda_r]^2\right) \prod_{t=1}^{T} \left(\mathrm{Var}(\mathbf{v}^{(t)}_{r_{i_t}}) + \mathbb{E}[\mathbf{v}^{(t)}_{r_{i_t}}]^2\right) - \mathbb{E}[\lambda_r]^2 \prod_{t=1}^{T} \mathbb{E}[\mathbf{v}^{(t)}_{r_{i_t}}]^2 \right] \\
&= \sum_{r=1}^{R} \mathrm{Var}(\lambda_r) \prod_{t=1}^{T} \mathrm{Var}(\mathbf{v}^{(t)}_{r_{i_t}}) \\
&= R(\frac{1}{R})^{\frac{1}{T}T} = 1.
\end{aligned} \tag{74}$$

For generalized RNN, one can show by a similar computation that the result does indeed hold. □

Proposition 5.2. *Assume additive non-linearity $\xi(x,y) = x + y$. Then the following initialization schemes*

- ***Generalized shallow network***: $\lambda_r \overset{iid}{\sim} \mathcal{N}(0, \frac{1}{R})$, $\mathbf{v}_r^{(t)} \overset{iid}{\sim} \mathcal{N}_M(0, \frac{1}{T} I_M)$
- ***Generalized RNN***: $\mathbf{g}^{(t)}_{r_{t-1}r_t} \overset{iid}{\sim} \mathcal{N}_L(0, \frac{1}{TR_1 \dots R_{T-1}} I_L)$

give weight tensors with zero expectation zero and variance one.

Proof. In the first case we get

$$\boldsymbol{\mathcal{W}}_{i_1 i_2 \dots i_T} = \sum_{r=1}^{R} \lambda_r \left(\sum_{t=1}^{T} \mathbf{v}^{(t)}_{r_{i_t}} \right),$$

hence similarly as in proposition (5.1) we get

$$\mathbb{E}[\boldsymbol{\mathcal{W}}_{i_1 i_2 \dots i_T}] = \sum_{r=1}^{R} \sum_{t=1}^{T} \underbrace{\mathbb{E}[\lambda_r]}_{=0} \mathbb{E}[\mathbf{v}_{r_{i_t}}^{(t)}] = 0$$

$$\begin{aligned}
\mathrm{Var}\left(\boldsymbol{\mathcal{W}}_{i_1 i_2 \dots i_T}\right) &= \sum_{r=1}^{R} \sum_{t=1}^{T} \left[\left(\mathrm{Var}(\lambda_r) + \mathbb{E}[\lambda_r]^2\right) \left(\mathrm{Var}(\mathbf{v}_{r_{i_t}}^{(t)}) + \mathbb{E}[\mathbf{v}_{r_{i_t}}^{(t)}]^2\right) - \mathbb{E}[\lambda_r]^2 \mathbb{E}[\mathbf{v}_{r_{i_t}}^{(t)}]^2 \right] \\
&= \sum_{r=1}^{R} \mathrm{Var}(\lambda_r) \sum_{t=1}^{T} \mathrm{Var}(\mathbf{v}_{r_{i_t}}^{(t)}) \qquad (75) \\
&= \sum_{r=1}^{R} \frac{1}{R} \sum_{t=1}^{T} \frac{1}{T} = 1.
\end{aligned}$$

For generalized RNN, one can show by a similar computation that the result does indeed hold. □

These initialization did indeed lead to slightly better performance and worked well with the other non linearities as well.

The purpose of the second numerical experiment is to further asses the expressivity of generalized RNNs compared to shallow networks for rectifier non-linearity. This is done by generating a number of generalized RNNs with different vales of TT-ranks and computing a lower bound on the rank of shallow networks necessary to realize the same grid tensor. Concretely, we proceed as follows:

1. Randomly generate parameters of a generalized RNN according to a $\mathcal{N}(0, 1)$ distribution as well as the matrix $\mathbf{F}$ followed by an element-wise ReLU activation.
2. Compute the corresponding grid tensor $\boldsymbol{\Gamma}(\mathbb{X})$ using the recursive formula given in proposition (4.1).
3. Compute the matricization $\mathrm{rank}[\boldsymbol{\Gamma}^l(\mathbb{X})]^{(s_{odd}, t_{even})}$ and estimate the lower bound R using lemma (4.10).

We now present the results of our experimentations in the following section.

5.2 Results

Test accuracies of the generalized networks with various nonlinearities can be found in Table 1. Classification of the toy datasets is a relatively easy problem and serves as a sanity check as to the performance of the networks which achieve an accuracy of 1. Furthermore, as we can see in Figure 3, the networks are able to implement non trivial decision boundaries for two dimensional datasets and various non-linearities.

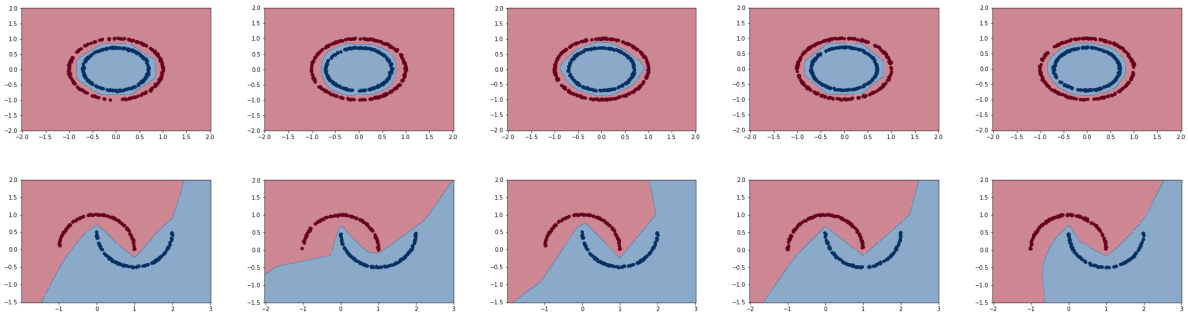

Figure 3: Decision boundaries for two dimensional datasets produced by a generalized RNN for various non linearites (from left to right : $\xi(x, y) = xy$, $\xi(x, y) =$max$(x, y, 0)$, $\xi(x, y) =$ln$(\mathrm{e}^x + e^y)$,$\xi(x, y) = x + y$, $\xi(x, y) = \sqrt{x^2 + y^2}$).

Fig 4 shows test accuracy on the IMDB dataset for rectifier nonlinearity, highlighting the fact that generalized shallow network of much higher rank is needed to achieve the level of performance of generalized RNNs.

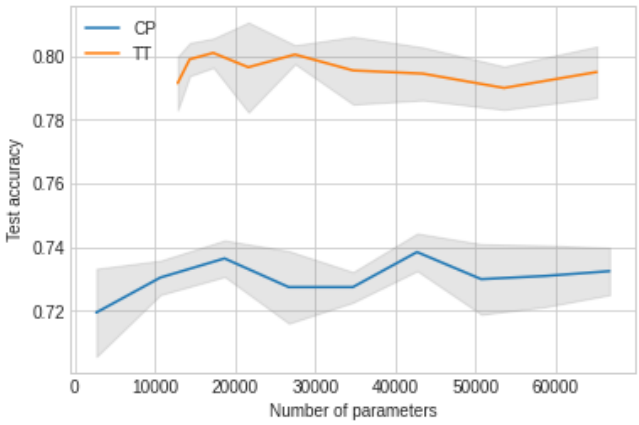

Figure 4: Test accuracy on the IMDB dataset for generalized RNNs and generalized shallow networks with respect to the total number of parameter with non linearity $\xi(x, y) = \max(x, y, 0)$. For simplicity, the same number of training epochs was used each time, which explains the slight drop in accuracy when considering large number of parameters. Nonetheless, the graph still shows that generalized RNN are more expressive then their shallow counterparts.

$\xi(x, y)$	xy	$\max(x, y, 0)$	$\ln(e^x + e^y)$	$x + y$	$\sqrt{x^2 + y^2}$
Generalized CP Network					
Moon	1.00	1.00	1.00	1.00	1.00
Circle	1.00	1.00	1.00	1.00	1.00
MNIST	0.9727	0.9426	0.9127	0.9503	0.9557
CIFAR10	0.4398	0.4522	0.455	0.5001	0.509
IMDB	0.738	0.756	0.754	0.738	0.754
Generalized TT Network					
Moon	1.00	1.00	1.00	1.00	1.00
Circle	1.00	1.00	1.00	1.00	1.00
MNIST	0.9593	0.9632	0.9154	0.9503	0.9598
CIFAR10	0.4355	0.4463	0.4220	0.4726	0.4336
IMDB	0.814	0.748	0.82	0.814	0.794

Table 1: Test accuracy of the generalized networks with various nonlinearities. Reported accuracies correspond to best accuracy obtained across multiple runs of training. In particular, different hyper parameters are used for different data sets and nonlinearities.

Figure 5 presents the results of our second experiment where we randomly generate 20 RNNs for each rank of interest. For TT-ranks equal to 1, 2, 4 and 8, the mean lower bound rank of generalized shallow networks obtained were 1.3, 7.6, 11.4 and 16.2 respectively with standard deviations equal to 0.7, 7.8, 8.5, 8.0. We see that indeed, as the rank of the generalized RNN increases, a much greater rank is necessary for an equivalent shallow network. We note hower that our results differ from those presented by the authors, in particular for R=8 we do not obtain nearly as many lower bounds greater then 30. We found that the method of initialization of the tensor cores greatly affect the obtained lower bounds and this might be reason for this.

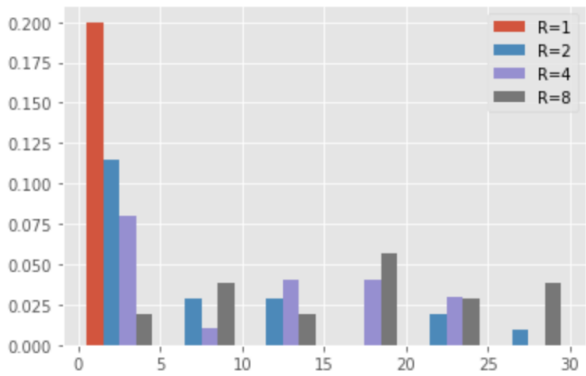

Figure 5: Distribution of lower bounds on the rank of generalized shallow networks equivalent to randomly generated generalized RNNs of ranks 1, 2, 4, 8 (M = 10, T = 6) for rectifier non-linearity (bars side by side).

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7), 2015.

[3] Nadav Cohen and Amnon Shashua. Convolutional rectifier networks as generalized tensor decompositions. In *International Conference on Machine Learning*, pages 955–963, 2016.

[4] IU S Iliashenko, IU S Iliashenko, Julij S Iljašenko, Yulij Ilyashenko, Ûlij S Il'âšenko, and S Yakovenko. *Lectures on analytic differential equations*, volume 86. American Mathematical Soc., 2008.

[5] Valentin Khrulkov, Oleksii Hrinchuk, and Ivan Oseledets. Generalized tensor models for recurrent neural networks. *arXiv preprint arXiv:1901.10801*, 2019.

[6] Valentin Khrulkov, Alexander Novikov, and Ivan Oseledets. Expressive power of recurrent neural networks. *arXiv preprint arXiv:1711.00811*, 2017.

[7] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[8] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[9] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[10] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[11] Igor Rostislavovich Shafarevich and Miles Reid. *Basic algebraic geometry*, volume 2. Springer, 1994.

[12] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

GENERALIZED TENSOR MODELS FOR RECURRENT NEURAL NETWORKS

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
AFAMBO	Nitya

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zurich, 25/05/2020	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.