A Tensor Decomposition Perspective on Second-order RNNs

Maude Lizaire [1] **Michael Rizvi-Martel** [1] **Marawan Gamal Abdel Hameed** [1] **Guillaume Rabusseau** [1 2]

Abstract

Second-order Recurrent Neural Networks (2RNNs) extend RNNs by leveraging second-order interactions for sequence modelling. These models are provably more expressive than their first-order counterparts and have connections to well-studied models from formal language theory. However, their large parameter tensor makes computations intractable. To circumvent this issue, one approach known as MIRNN consists in limiting the type of interactions used by the model. Another is to leverage tensor decomposition to diminish the parameter count. In this work, we study the model resulting from parameterizing 2RNNs using the CP decomposition, which we call CPRNN. Intuitively, the rank of the decomposition should reduce expressivity. We analyze how rank and hidden size affect model capacity and show the relationships between RNNs, 2RNNs, MIRNNs, and CPRNNs based on these parameters. We support these results empirically with experiments on the Penn Treebank dataset which demonstrate that, with a fixed parameter budget, CPRNNs outperforms RNNs, 2RNNs, and MIRNNs with the right choice of rank and hidden size.

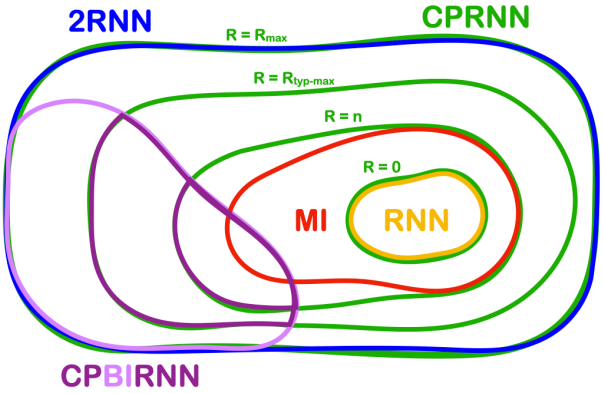

Figure 1. Overview of expressivity relationships among Recurrent Neural Network architectures. 2RNNs (blue) encompasses all other models. BIRNN (purple) and RNN (yellow) are a subclasses of 2RNNs as they only have first-order and second-order interactions respectively. MIRNN (red) includes element wise multiplicative interactions. CPRNNs expressive power varies with the rank R, reaching the same capacity as 2RNNs when $R = R_{\max}$.

1. Introduction

Recurrent neural networks (RNNs) have been pivotal to the deep learning revolution (Elman, 1990; Hopfield, 1982). Given their strong inductive bias, they are a natural choice when it comes to sequence modelling. Although transformers (Vaswani et al., 2017) remain the de facto choice for language modelling, there exists many domains in which RNNs still excel. Moreover, with the advent of state space models (Gu et al., 2020; 2021), such architectures have seen an increase in popularity (Orvieto et al., 2023), especially with regard to long-range dependencies. Second-order Recurrent Neural Networks (2RNNs) are a generalization of RNNs which integrates second-order interactions between hidden states and inputs. These second-order interactions make 2RNNs strictly more expressive than their first-order counterparts. However, the tensors parameterizing the second-order term quickly become very large as the hidden state and input dimensions grow (Giles et al., 1989; Goudreau et al., 1994), thus making inference with such models intractable. A simple but drastic approach to reduce the parameter count is to keep only the second-order interactions from the component-wise product between input and hidden state (Wu et al., 2016). The resulting model, Multiplicative Integration RNN (MIRNN), was shown to alleviate the vanishing/exploding gradient problem and to improve performance on language modelling tasks. A linear version of this model without first-order interactions was also introduced under the name *Recurrent Arithmetic Circuits* to analyze the benefits of depth in RNNs (Levine et al., 2018).

A more sophisticated approach would be to leverage tensor

[1]Mila & DIRO, Université de Montréal, Montreal, Canada [2]CIFAR AI Chair. Correspondence to: Maude Lizaire <maude.lizaire@umontreal.ca>, Guillaume Rabusseau <grabus@iro.umontreal.ca>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

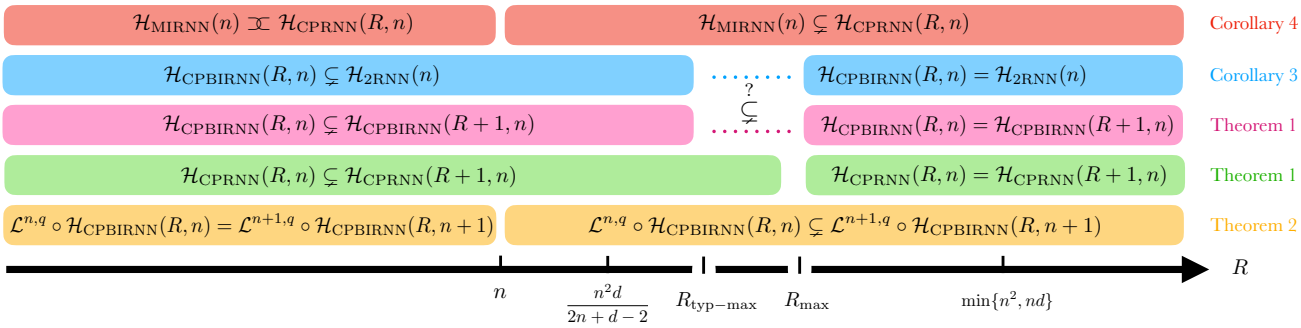

Figure 2. Relations of expressivity between CP(BI)RNNs, 2RNNs and MIRNNs as a function of the rank R of the CP(BI)RNN; n denotes the hidden dimension, d the input dimension and $R_{\max}$ (resp. $R_{\text{typ-max}}$) the maximal CP rank (resp. maximal typical CP rank). Theorems 1&2 and Corollaries 3&4 detail these results.

decomposition to compress the third-order tensor parameterizing the 2RNN. While several decompositions could be used to reduce parameter count in 2RNNs, in this work we focus on the popular CP decomposition (Kiers, 2000). We call the resulting model CPRNN. Sutskever et al. (2011) empirically demonstrated the potential of this architecture by successfully deploying it on one of the largest RNN applications. However, there was no theoretical analysis of this new model, a gap we propose to fill with this work.

It is interesting to note that while the model proposed in Sutskever et al. (2011) is indeed a CPRNN—that is a 2RNN with second-order weights parameterized by a CP decomposition— there is no mention of tensor decomposition or of the CP decomposition in their work.

Similarly to matrices, the sets of tensors of CP rank at most R, for increasing values of R, form a nested family in the space of tensors. Intuitively, an increase in rank strictly increases the capacity up to a point of saturation. This naturally leads to the following questions:

- How does increasing the capacity of the tensor parameter relate to the expressivity of CPRNNs?
- How does the point of saturation in the tensor parameter space translate in the function space of CPRNNs? In particular, how is it affected by the hidden dimension?
- More generally, how do the rank and the hidden dimension interplay in controlling the capacity of CPRNNs?

To the best of our knowledge, our work is the first to formally address these questions. We start from the observation that the rank in CPRNNs acts as a hyper-parameter interpolating between first-order and second-order RNNs. We formalize this observation by showing that CPRNNs are equivalent to 2RNNs above some maximal rank. We also show that MIRNNs are a special case of CPRNNs. More precisely, we show that MIRNNs correspond to a specific point on this interpolation line where the rank and hidden dimension of a CPRNN are both equal. In fact, we show that CPRNNs are strictly more expressive than MIRNNs past this threshold, i.e., when rank is greater than hidden size. Tensor decompositions thus prove to be an effective approach to characterize the expressivity of RNNs with different degrees of multiplicative interactions. Figure 1 illustrates the hierarchy established by the rank of CPRNNs, where the threshold values between different classes explored in this paper are directly connected to the hidden size.

We investigate to which extent theory holds in practice for models trained on real data by conduction a set of experiments on the Penn Tree Bank dataset*. We first show that, as expected, the performance of CPRNNs increases as either the rank or hidden dimension grows, and that the rank of CPRNNs naturally controls the parameter/expressivity tradeoff, interpolating between RNNs and 2RNNs in a fine-grained fashion. More interestingly, our experimental results show that for any model size budget there always exists a choice of hidden size and rank that fits the parameter budget and outperforms all 2RNNs, RNNs and MIRNNs with the same parameter count. Another interesting behavior of CPRNNs illustrated by our experiments is that, since rank and hidden size are tied when the number of parameters is fixed, there are two underfitting regimes: one when the rank is too small (where the rank acts as the bottleneck) and one when it is too large (where it is the hidden dimension that is too small and acts as a bottleneck).

Summary of contributions We present a formal study of CPRNNs: a natural approach to reducing parameter count in second-order RNNs. While CPRNNs have been empirically considered previously, to the best of our knowledge, this work is the first to thoroughly and rigorously analyze it from a formal perspective. Our analysis establishes several novel and non-trivial results related to the expressivity of

*Code base for this paper can be found at https://github.com/MaudeLiz/cprnn

CPRNNs, how the rank and hidden size of CPRNN interact, and how CPRNNs encompass previous second-order recurrent models. Beyond this theoretical analysis, we design several experiments on real data showcasing various properties and empirical behaviours of CPRNNs. In particular, we show that CPRNNs always offer a better tradeoff between expressivity and size than RNNs, 2RNNs and MIRNNs.

Related work Extensive work has been done on the expressive power of RNNs and how they relate to formal languages (see e.g., Siegelmann and Sontag (1994); Chen et al. (2017); Weiss et al. (2018); Korsky and Berwick (2019); Merrill (2019); Merrill et al. (2020); Delétang et al. (2022)). In the case of 2RNNS there is a direct correspondence with finite states machines. Indeed, the linear form of 2RNNs was shown to generalize weighted finite automaton (WFA) to non-discrete inputs (Li et al., 2022). From a language modelling perspective, multiplicative interactions draw interest for their potential to represent more complex dependencies such as compositional semantics (Irsoy and Cardie, 2014; Jayakumar et al., 2020). This motivated the design of various multiplicative RNNs architectures (Tjandra et al., 2016; Krause et al., 2017; Sutskever et al., 2011; Wu et al., 2016; Su et al., 2024), among which CPRNNs and MIRNNs stand out for their parameter efficiency. At the same time, tensor decomposition methods have been used to compress sequential models (Yang et al., 2017; Ye et al., 2018; Ma et al., 2021; Wang et al., 2021; Tjandra et al., 2018) and tackle optimization problems involving multivariate higher-order polynomial functions (Yu et al., 2017; Ayvaz and De Lathauwer, 2022; Dubey et al., 2022). More broadly, tensor decompositions have been used to study the expressivity of different neural network architectures, in particular to explore the benefits of depth (Cohen and Shashua, 2016a; Cohen et al., 2016; Cohen and Shashua, 2016b; Sharir and Shashua, 2017; Levine et al., 2018; Alexander et al., 2024; Razin et al., 2024) and structural alignment between data and model (Balda et al., 2018; Khrulkov et al., 2017; 2019). The analysis in most of these work relies on comparing different tensor decomposition (or tensor network) structures, in contrast, our analysis relies on studying the effect of the rank of a fixed tensor network structure. This strategy of leveraging the rank of a tensor decomposition to characterize the expressive power of models has also been used for deep polynomial networks in (Kileel et al., 2019).

Finally, it is worth mentioning that the popular family of state space models (Hamilton, 1994; Gu et al., 2021) also have second-order counterparts (Sattar et al., 2022). These model variants appear mostly in dynamical systems literature and are not quite as well known in the machine learning community.

2. Preliminaries

We first introduce notations, tensor decomposition and the various RNN models studied in this work.

2.1. Notation

Vectors, matrices and tensors are respectively represented in bold $\mathbf{v} \in \mathbb{R}^{d_1}$, uppercase bold $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$ and calligraphic $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times \cdots \times d_p}$. The n-mode product of a tensor with a *vector*[†] is defined by: $(\boldsymbol{\mathcal{T}} \times_n \mathbf{v})_{i_1,\ldots,i_{n-1},i_{n+1},\ldots,i_p} = \sum_{i_n=1}^{d_n} \boldsymbol{\mathcal{T}}_{i_1,\ldots,i_p} \mathbf{v}_{i_n}$. The Hadamard product, or element-wise product, between vectors of the same size is noted $\mathbf{u} \odot \mathbf{v}$. The outer product noted $\mathbf{u} \circ \mathbf{v}$ is given by $(\mathbf{u} \circ \mathbf{v})_{ij} = \mathbf{u}_i \mathbf{v}_j$.

2.2. CP decomposition

A *CP decomposition* of rank R factorizes a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ into a sum of rank one tensors, $\boldsymbol{\mathcal{T}} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \equiv [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, where $\mathbf{a}_r \in \mathbb{R}^{d_1}$, $\mathbf{b}_r \in \mathbb{R}^{d_2}$ and $\mathbf{c}_r \in \mathbb{R}^{d_3}$, and the factors matrices $\mathbf{A} \in \mathbb{R}^{d_1 \times R}, \mathbf{B} \in \mathbb{R}^{d_2 \times R}, \mathbf{C} \in \mathbb{R}^{d_3 \times R}$ have as columns the vectors $\mathbf{a}_r, \mathbf{b}_r$ and $\mathbf{c}_r$, respectively. Using a CP decomposition reduces the number of parameters from $\mathcal{O}(d^3)$ to $\mathcal{O}(Rd)$ where $d = max\{d_1, d_2, d_3\}$.

The *CP rank* of a tensor, noted $\mathrm{rank}_{\mathrm{CP}}(\boldsymbol{\mathcal{T}})$, is the minimal value R for which an exact CP decomposition of rank R exists. Note that what we call a CP decomposition may not be minimal: writing a CP decomposition $\boldsymbol{\mathcal{T}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ does not imply that $\mathrm{rank}_{\mathrm{CP}}(\boldsymbol{\mathcal{T}}) = R$, but only that $\mathrm{rank}_{\mathrm{CP}}(\boldsymbol{\mathcal{T}}) \leq R$ (i.e., there may exist a smaller CP decomposition of $\boldsymbol{\mathcal{T}}$).

2.3. Models

We now define the different RNN variants considered in this work and briefly explain how they relate to one another. Figure 3 summarizes the set models presented.

Definition 1 (RNN)**.** *A Recurrent Neural Network $\mathcal{R} = \langle \mathbf{h}^0, \mathbf{U}, \mathbf{V}, \mathbf{b}, \sigma \rangle$ of hidden size n is parameterized by an initial hidden state $\mathbf{h}^0 \in \mathbb{R}^n$, weight matrices $\mathbf{U} \in \mathbb{R}^{n \times d}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$, a bias term $\mathbf{b} \in \mathbb{R}^n$ and an activation function $\sigma : \mathbb{R}^n \to \mathbb{R}^n$. Given an input sequence $(\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^T)$, a RNN computes, for each time steps t, the following hidden state:*

$$\mathbf{h}^t = \sigma(\mathbf{V}\mathbf{h}^{t-1} + \mathbf{U}\mathbf{x}^t + \mathbf{b}).$$

Second-order RNNs extend RNNs by incorporating bilinear interactions between input and (previous) hidden state.

Definition 2 (2RNN)**.** *A Second-order Recurrent Neural*

[†]Note that this notation slightly differs from the one introduced in (Kolda and Bader, 2009) where $\times_n$ denotes the n-mode product of a tensor with a *matrix*.

Network $\mathcal{A} = \langle \mathbf{h}^0, \boldsymbol{\mathcal{A}}, \mathbf{U}, \mathbf{V}, \mathbf{b}, \sigma \rangle$ *of hidden size* n *is parameterized by a recurrent weight tensor* $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n \times d \times n}$ *and the other terms as in Definition 1. Given an input sequence* $(\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^T)$*, a 2RNN computes the following hidden state for each time step* t *:*

$$\mathbf{h}^t = \sigma(\boldsymbol{\mathcal{A}} \times_1 \mathbf{h}^{t-1} \times_2 \mathbf{x}^t + \mathbf{V}\mathbf{h}^{t-1} + \mathbf{U}\mathbf{x}^t + \mathbf{b}).$$

In the absence of the second-order term ($\boldsymbol{\mathcal{A}} = 0$), we recover the definition of a (first-order) RNN. When the model is restricted to the bilinear term (*i.e.* $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{b}$ are null), it is referred to as a BIRNN. It is worth mentioning that the class of linear BIRNNs exactly corresponds to weighted languages recognized by weighted finite automata (Li et al., 2022). The next class of models, CPRNNs, parameterizes the second-order weight tensor of 2RNNs using the CP decomposition.

Definition 3 (CPRNN)**.** *A CP Recurrent Neural Network* $\mathcal{A}_{CP} = \langle \mathbf{h}^0, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{U}, \mathbf{V}, \mathbf{b}, \sigma \rangle$ *with hidden size* n *and rank* R *is a 2RNN whose second-order term is parameterized by a CP decomposition* $[\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ *with* $\mathbf{A} \in \mathbb{R}^{n \times R}$*,* $\mathbf{B} \in \mathbb{R}^{d \times R}$ *and* $\mathbf{C} \in \mathbb{R}^{n \times R}$*. Given an input sequence* $(\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^T)$*, a CPRNN computes for each time steps* t *the following hidden state:*

$$\mathbf{h}^t = \sigma([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \times_1 \mathbf{h}^{t-1} \times_2 \mathbf{x}^t + \mathbf{V}\mathbf{h}^{t-1} + \mathbf{U}\mathbf{x}^t + \mathbf{b}).$$

When the first-order terms are null, the model is referred to as a CPBIRNN ($\mathcal{A}_{CPBI}$). Note that even though the hyper-parameter R is called the *rank* of the CPRNN (or CPBIRNN), the CP rank of the tensor $[\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ is only upper-bounded by R (it can be strictly smaller, e.g. choosing all weights to be ones, in which case the CP rank of $[\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ would be one for any value of R). The last model we introduce is the MIRNN, which was proposed in (Wu et al., 2016) as a way to drastically reduce the number of parameters of 2RNNs.

Definition 4 (MIRNN)**.** *A Recurrent Neural Network with Multiplicative Integration* $\mathcal{A}_{MI} = \langle \mathbf{h}^0, \boldsymbol{\alpha}, \boldsymbol{\beta_1}, \boldsymbol{\beta_2}, \mathbf{U}, \mathbf{V}, \sigma \rangle$ *of hidden size* n *is parameterized as Definition 1 with additional gate vectors* $\boldsymbol{\alpha}, \boldsymbol{\beta_1}, \boldsymbol{\beta_2} \in \mathbb{R}^n$*. The hidden state at time step* t *computed by a MIRNN given an input sequence* $(\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^T)$ *is:*

$$\mathbf{h}^t = \sigma(\boldsymbol{\alpha} \odot \mathbf{V}\mathbf{h}^{t-1} \odot \mathbf{U}\mathbf{x}^t + \boldsymbol{\beta_1} \odot \mathbf{V}\mathbf{h}^{t-1} + \boldsymbol{\beta_2} \odot \mathbf{U}\mathbf{x}^t + \mathbf{b}).$$

The activation functions considered in this work are either bijective (e.g. linear and $\tanh$) or the Rectified Linear Unit (ReLU) (Nair and Hinton, 2010) applied element-wise. We use the notation $\mathbf{a}^t$ to refer to the pre-activation vectors at each time step, that is the hidden state prior the activation function, $\mathbf{h}^t = \sigma(\mathbf{a}^t)$.

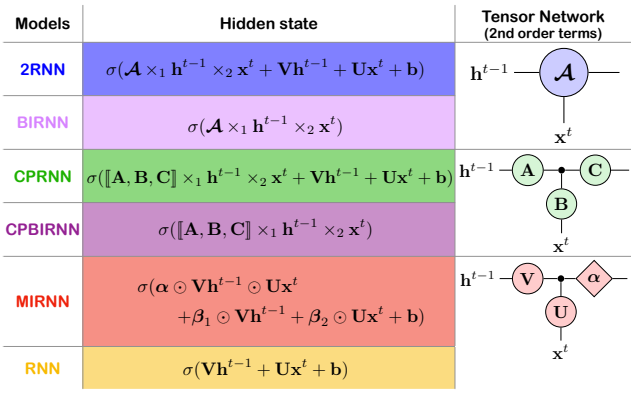

Figure 3. Different recurrent models considered in this work with their hidden state computation and the tensor network representation of their second-order term (the diamond shape represents a diagonal matrix).

3. Theoretical results

In order to compare the expressive power of CPRNNs for different ranks and hidden sizes (and with other models), we formally define the set of functions they can represent.

Definition 5. *We denote by* $\mathcal{H}_{\text{CPRNN}}(R, n)$ *the set of functions* h *mapping input sequences to the hidden states sequences computed by CPRNNs of rank* R *and hidden size* n*:* $h(\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^T) = (\mathbf{h}^1, \mathbf{h}^2, ..., \mathbf{h}^T)$*, where the hidden state vectors* $\mathbf{h}^t$ *are defined in Def. 3,*

In the same way, we define the classes of functions $\mathcal{H}_{\text{CPBIRNN}}(R, n)$, $\mathcal{H}_{\text{2RNN}}(n)$ and $\mathcal{H}_{\text{MIRNN}}(n)$ respectively computed by CPBIRNNs, 2RNNs and MIRNNs of hidden size n, respectively. The theoretical findings presented in this section are summarized in Figure 2.

3.1. Expressive power of CPRNNs: Relating tensor space to function space through rank

Intuitively, the rank of a CPRNN is an hyper-parameter controlling the tradeoff between expressive power and parameter efficiency. We thus expect the expressivity to increase with rank, up to some potential saturating point. Let $R_{\max}$ be the maximal CP rank for a family of tensors sharing the same dimensions:

$$R_{\max}^{d_1, d_2, d_3} = \max\{\text{rank}_{\text{CP}}(\boldsymbol{\mathcal{T}}) \mid \boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times d_2 \times d_3}\}.$$

$R_{\max}$ represents the saturation point in tensor space, which translates into a saturation in expressivity of CPRNNs for ranks above that threshold. This comes from the fact that *an inclusion over the space of tensors parameterizing CPRNNs directly implies an inclusion in the space of functions computed by CPRNNs*.

Indeed, to see this, consider CPRNNs of ranks R and $R+1$. The second-order terms computed by CPRNNs of rank R

are parameterized by CP decompositions that are less expressive than the ones from CPRNNs with rank $R+1$ (*i.e.* there is inclusion in the tensor space). Looking at the computation of their hidden state $\mathbf{h}^t$, it is easy to see that any function computed by a CPRNNs $\mathcal{A}_{CP}$ of rank R can be computed by a CPRNNs of rank $R+1$. It suffices to use the same weights as $\mathcal{A}_{CP}$ and to pad the extra dimensions with zeros. When $R > R_{\max}$, CPRNNs of ranks R and $R+1$ become equally expressive. Indeed, the expressive power of the CP decompositions parameterizing their second-order terms saturates, therefore they both can be parameterized by a minimal CP decomposition padded with zeros.

Now, to show that a class of CPRNNs is *strictly* more expressive than another one, the direct implication from the expressivity of the tensor space does not hold. Indeed, under the argument for inclusion discussed above, lies the assumption that the computation at each time step $\mathbf{h}^t$ is a function of two variables, $(\mathbf{x}^t, \mathbf{h}^{t-1}) \mapsto \mathbf{h}^t$ and that $\mathbf{h}^{t-1}$ is "free". But it is not, it is a function of $\mathbf{h}^0$ and the previous inputs. This is crucial when trying to prove *strict inclusion* results: *a strict inclusion of sets of tensor parameters does not imply strict inclusion of the corresponding classes of recurrent functions*. In other words, the recurrent nature of the function h cannot be ignored. The relation between tensor space and function space thus becomes non-trivial when considering strict inclusion, raising the question: under which conditions does the expressive power of CPRNNs *strictly* increase with rank?

To address this question we consider the notion of *typical* ranks. These are CP rank values that have a non-zero probability of occurring in random tensors (*i.e.* ranks R's such that the set of tensors of CP rank R have positive Lebesgue measure). Our first theoretical result states that the expressivity of CPBIRNNs increases with the rank up to the maximal *typical* rank $R_{\text{typ}-\max}$. The result generalizes to CPRNNs with linear activation function. Theorem 1 also provides $R_{\max}$ as a bound above which the expressive power of CPRNNs saturates. Whether the inclusion is strict or not in the gap between $R_{\text{typ}-\max}$ and $R_{\max}$ for CPBIRNNs remains an open question.

Theorem 1. *The following hold for any n and d:*

- $\mathcal{H}_{\text{CPRNN}}(R,n) \subseteq \mathcal{H}_{\text{CPRNN}}(R+1,n)$ *for any R.*
- $\mathcal{H}_{\text{CPRNN}}(R,n) = \mathcal{H}_{\text{CPRNN}}(R+1,n)$ *for any $R \geq R_{max}$.*

Moreover, assuming $n \leq d$:

- $\mathcal{H}_{\text{CPBIRNN}}(R,n) \subsetneq \mathcal{H}_{\text{CPBIRNN}}(R+1,n)$ *for any $R < R_{\text{typ}-\max}$ and any real analytic invertible activation function.*
- $\mathcal{H}_{\text{CPRNN}}(R,n) \subsetneq \mathcal{H}_{\text{CPRNN}}(R+1,n)$ *for a linear activation function and any $R < R_{\text{typ}-\max}$.*

Note that the first two points also hold for CPBIRNNs (since they are CPRNNs restricted to second-order interactions). We conjecture that the third and fourth points of this theorem generalize to any common activation function (hyperbolic tangent, ReLU and linear). For the strict inclusions (third and fourth points), the assumption $n \leq d$ is made in order to preserve the emphasis of the results and the proofs on the interplay between rank and hidden size, as opposed to a limitation coming from the input size.

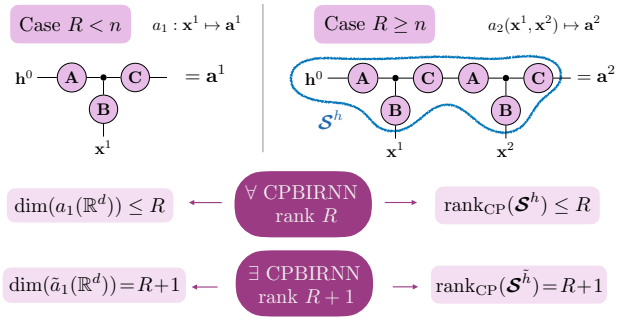

Figure 4. Elements of proof for strict inclusion of CPBIRNNs in Theorem 1. Details of the proof can be found in the Appendix.

Sketch of proof The method to prove inclusions and saturation consists in finding explicit parameterizations such that there is equality between the latent vectors $\mathbf{h}^t$'s, as outlined in the beginning of this section.

To show strict inclusion, that is $\mathcal{H}_{\text{CPBIRNN}}(R+1,n) \not\subset \mathcal{H}_{\text{CPBIRNN}}(R,n)$, two cases are considered: $R < n$ and $R \geq n$. Figure 4 illustrated the key elements of these proofs. The idea behind the proof when $R < n$ is to consider the linear mapping of the first pre-activation vector $a_1 : \mathbf{x}^1 \mapsto \mathbf{a}^1 \in \mathbb{R}^n$. For CPBIRNNs of rank R the dimension of the image of a_1 is upper bounded by R. At the same time, we show that there exists a $\tilde{h}$ computed by a CPBIRNN of rank $R+1$ for which the pre-activation map $\tilde{a}_1$ has an image of dimension equal to $R+1$. This implies that the dimension of the manifold formed by the hidden vectors $\tilde{\mathbf{h}}^1 = \sigma(\tilde{\mathbf{a}}^1)$ of this CPBIRNNs is $R+1$ (for invertible σ). This cannot be the case of any CPBIRNN of rank R, as this dimension is upper bounded by R, which concludes the case $R < n$.

In the case $R \geq n$, the rank does not act as a bottleneck on the dimension of the space of first hidden states.Therefore, we turn to the computation of the second hidden state $\mathbf{h}^2$, looking at the mapping of the pre-activation vector $a_2(\mathbf{x}^1, \mathbf{x}^2) \mapsto \mathbf{a}^2$. We consider the tensor $\boldsymbol{\mathcal{S}}^h \in \mathbb{R}^{d \times d \times n}$ defined by $\boldsymbol{\mathcal{S}}^h_{ijk} = [a_2(\mathbf{e}_i, \mathbf{e}_j)]_k$, where $\mathbf{e}_i$ denotes the ith vector of the canonical basis of $\mathbb{R}^d$. Intuitively, $\boldsymbol{\mathcal{S}}^h$ gathers all the second hidden states obtained by applying the CPBIRNN to length 2 sequences of one-hot encodings and, crucially, is a witness to the low CP rank structure of the model.

More precisely, we show that (i) the CP rank of this tensor is upper bounded by the rank of the CPBIRNN and (ii) there exists a parameterization such that this limit can be reached when the rank is smaller or equal to $R_{\text{typ}-\max}$. Therefore, for $R < R_{\text{typ}-\max}$, we conclude that there exists a function computed by a CPBIRNN of rank $R+1$ whose tensor $\boldsymbol{\mathcal{S}}^h$ has CP rank $R+1$, and thus cannot be computed by a CPBIRNN of rank R.

For CPRNNs, we again look at the computation of the second hidden state vector, $h_2 : (\mathbf{x}^1, \mathbf{x}^2) \mapsto \mathbf{h}^2$. This time however, because the activation function is linear, h_2 can be decomposed in four terms: $h_2(\mathbf{x}^1, \mathbf{x}^2) = \alpha(\mathbf{x}^1, \mathbf{x}^2) + \beta(\mathbf{x}^1) + \gamma(\mathbf{x}^2) + \delta$ where α is a bilinear map containing only second-order terms. This decomposition is such that given two functions h and $\hat{h}$, if $\alpha \neq \hat{\alpha}$ then $h \neq \hat{h}$. Therefore, we define the tensor $\boldsymbol{\mathcal{S}}^{\alpha}_{ijk} = [\alpha(\mathbf{e}_i, \mathbf{e}_j)]_k$ and apply a similar argument as for CPBIRNNs. The complete proof can be found in Appendix.

Explicit bounds Note that unlike matrices, for higher order tensors $R^{d_1,d_2,d_3}_{\max}$ is not given by the smallest dimension $\min\{d_1, d_2, d_3\}$; it can even be greater than the largest dimension. For a third-order tensor, a loose upper-bound is given by $R^{d_1,d_2,d_3}_{\max} \leq \min\{d_1 d_2, d_1 d_3, d_2 d_3\}$, but the exact value is in general unknown and bounding it is a non-trivial problem (see e.g. (Howell, 1978)). In the context of Theorem 1, this upper bound implies that the saturation result is valid for $R \geq \min\{nd, n^2\}$. Similarly, while the notion of *typical* rank for matrices leads to a unique value that coincides with $R_{\max}$, for higher-order tensors there can be a set of typical ranks that does not necessarily include $R_{\max}$. The characterization of *typical* ranks is also a complex problem, but there is a known lower bound on the smallest typical rank which guarantees that the last two bullets of Theorem 1 hold when $R < \frac{n^2 d}{(2n+d-2)}$ (Strassen, 1983; Brockett, 1976; Comon et al., 2009).

3.2. Expressive power of CPRNNs: interplay between rank and hidden size

We have established that the rank plays a key role in the expressive power of CPRNNs. The following natural question is to what extent increasing hidden size impacts the models expressivity for a fixed rank. Comparing the classes of functions $\mathcal{H}_{\text{CPBIRNN}}(R, n)$ and $\mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ is, however, an ill-defined problem because their output space is different. It nonetheless makes sense to compare the expressiveness of these classes after being composed with another family of simple functions bringing them to a common space. We naturally consider linear maps. Let $\mathcal{L}^{n,q}$ denote the space of linear maps from $\mathbb{R}^n$ to $\mathbb{R}^q$ with $q \geq 1$. To formally study the impact of hidden size in CPRNNs, we compare the composition of functions : $\mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n)$ which can be thought of as an output layer. The following theorem shows that the rank of the CP decomposition acts as a bottleneck to the expressive power of CPRNNs: for hidden sizes below the CPRNN rank, the expressive power of CPRNNs improves with its hidden size; however, once the hidden size gets larger than the rank, the expressivity of the model saturates.

Theorem 2. *The following hold for any d and n:*

- $\mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n) \subseteq \mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ *for any R and n.*
- $\mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n) = \mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ *for any $n \geq R$ and linear activation function.*

Moreover, assuming $n \leq d$:

- $\mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n) \subsetneq \mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ *for any $n < R$ and any invertible activation function satisfying $\sigma(0) = 0$.*

We conjecture that the second point (saturation) and the third point (strict inclusion) of the theorem also hold for any real analytic activation function. Note that this theorem is intentionally stated only for CPBIRNNs. In CPRNNs the expressivity would also be impacted by the difference in hidden size of the first-order and bias terms, while our focus is to compare the second-order terms.

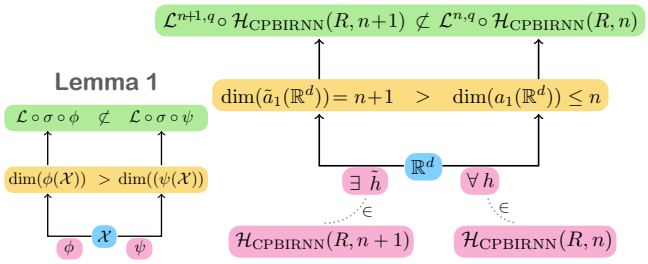

Figure 5. Schematic representation of Lemma 1 and its application in the context of Theorem 2.

Sketch of proof The main technical challenge of the proof is to show the strict inclusion (details for inclusion and saturation can be found in appendix). We want to show that, when $n < R$, $\mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n)$ is strictly included in $\mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$. The key idea is to leverage the following two facts:

- the dimension of the linear space formed by all first hidden states pre-activation computed by a function $h \in \mathcal{H}_{\text{CPBIRNN}}(R, n)$ is bounded by n, i.e. $\dim(a_1(\mathbb{R}^d)) \leq n$,
- there exists a function $\tilde{h} \in \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ for which the corresponding linear space is of dimension $n+1$, i.e. $\dim(\tilde{a}_1(\mathbb{R}^d)) = n+1$.

While these two facts are obvious (when $n < R$), it is not trivial to show that they imply the existence of a function $\ell \circ h \in \mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ that cannot be computed by any function in $\mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n)$, since (in some very loose sense) the strictly greater intrinsic linear dimension of $\tilde{h}$ is collapsed into a q-dimensional space through an activation function and a linear projection (note that if $q > n$ and the activation is linear, the result is somehow trivial).

This difficulty is addressed by the following lemma showing that as long as the activation σ is a homeomorphism, the linear bottleneck on the space of pre-activations implies a strict inclusion of the function classes after non-linear transformation through σ and linear projection. This result is schematically illustrated in Figure 5 as well as how it is applied to demonstrate the strict inclusion of Theorem 2.

Lemma 1. *Let V be a vector space of dimension d, $\phi, \psi : \mathcal{X} \to V$ two maps whose images $\phi(\mathcal{X})$ and $\psi(\mathcal{X})$ are subspaces of V and σ an homeomorphism. Lastly, let $\mathcal{L}(V)$ denote the set of all linear forms on V (i.e. $\mathcal{L}(V)$ is the dual space V^*). If $\dim(\phi(\mathcal{X})) > \dim(\psi(\mathcal{X}))$, then $\mathcal{L}(V) \circ \sigma \circ \phi \not\subset \mathcal{L}(V) \circ \sigma \circ \psi$.*

Note that this lemma could be applied more broadly to other neural network architectures.

3.3. Comparison with RNNs, 2RNNs and MIRNNs

First, it is trivial to check that CPRNNs are strictly more expressive than RNNs since both models share the same first-order terms. In fact, one can view RNNs as CPRNNs with rank $R = 0$. Similarly, one can easily check that there is no inclusion relation (one way or the other) between RNNs and CPBIRNNs since the latter only contain second-order terms while the former only captures first-order ones.

Turning to the comparison with 2RNNs, Theorem 1 naturally implies that (i) 2RNNs are strictly more expressive than CPBIRNNs for ranks smaller than the maximal typical rank and (ii) the expressivity of CPRNNs saturates when they become as expressive as 2RNNs, that is at $R_{\max}$, leading to the following corollary.

Corollary 3. *For any d and n, we have:*

- $\mathcal{H}_{\text{CPRNN}}(R, n) = \mathcal{H}_{\text{2RNN}}(n)$ *for any $R \geq R_{max}$ (for any activation function)*

Moreover, assuming $n \leq d$:

- $\mathcal{H}_{\text{CPBIRNN}}(R, n) \subsetneq \mathcal{H}_{\text{2RNN}}(n)$ *for any $R < R_{\text{typ-max}}$ and any real analytic invertible activation function.*

Note that the second point (strict inclusion) also holds for CPRNNs with linear activation functions (as a corollary of Theorem 1) and the first point also holds for CPBIRNNs (trivially).

Another approach suggested to introduce second-order terms to RNNs while preserving a similar number of parameters is the MIRNN. Inspired by the gated mechanisms of LSTM and GRU, MIRNNs limit second-order interactions to terms that are linear in the component-wise/Hadamard product of the hidden state and input (see Def. 4):

$$\text{diag}(\boldsymbol{\alpha})(\mathbf{V}\mathbf{h}^{t-1} \odot \mathbf{U}\mathbf{x}^t).$$

At the same time, the CP decomposition of the second-order term in CPRNN can also be expressed in terms of a Hadamard product:

$$[\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \times_1 \mathbf{h}^{t-1} \times_2 \mathbf{x}^t = \mathbf{C}(\mathbf{A}^\top \mathbf{h}^{t-1} \odot \mathbf{B}^\top \mathbf{x}^t).$$

This illustrates that MIRNNs are in fact CPRNNs of rank n whose matrix $\mathbf{C}$ is constrained to be diagonal. Consequently, $\mathcal{H}_{\text{MIRNN}}(n) \subseteq \mathcal{H}_{\text{CPRNN}}(n, n)$. The constraint on the matrix $\mathbf{C}$ to be diagonal means that each component of the second-order term of a MIRNN hidden state is computed by a rank one matrix while for a CPRNN it is rank R. This suggests that CPRNNs are strictly more expressive than MIRNNs for ranks greater than the hidden size. To prove it, it suffices to use the strict inclusion of Theorem 1 for linear CPRNNs. Indeed, in this case and when $R > n$ we have $\mathcal{H}_{\text{MIRNN}}(n) \subseteq \mathcal{H}_{\text{CPRNN}}(n, n) \subsetneq \mathcal{H}_{\text{CPRNN}}(R, n)$. When the rank is smaller than the hidden dimension, there is no clear inclusion relation (in one way or the other) between the two model families. The following corollary (whose proof can be found in appendix) formalizes this result.

Corollary 4. *Assuming $n \leq d$, for any $R > n$,*

- $\mathcal{H}_{\text{MIRNN}}(n) \subseteq \mathcal{H}_{\text{CPRNN}}(R, n)$
- $\mathcal{H}_{\text{MIRNN}}(n) \subsetneq \mathcal{H}_{\text{CPRNN}}(R, n)$ *for linear activation function*

4. Experiments

We investigate to which extent our theoretical results hold empirically in the context of character level sequence modelling subject to gradient based training. We perform experiments on the Penn Treebank dataset (Marcus et al., 1993) measuring bits-per-character (BPC) using the same train/valid/test partition as in Mikolov et al. (2012). All models were trained using truncated back propagation through time (Werbos, 1990) with sequence length of 50, batch size of 128 and using the Adam optimizer (P. Kingma and Ba, 2015) to minimize the negative log likelihood. Initial weights were drawn from a uniform random distribution $\mathcal{U}[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$. For all experiments, we use the $\tanh$ activation function. For training, we use early stopping and a scheduler to reduce the learning rate (initialized at 0.001) by half on plateaus of the validation loss. All the results presented are on the test set. Each point on the plots is an average over 10 experiments run with different seeds and the variance is represented by a shaded area.

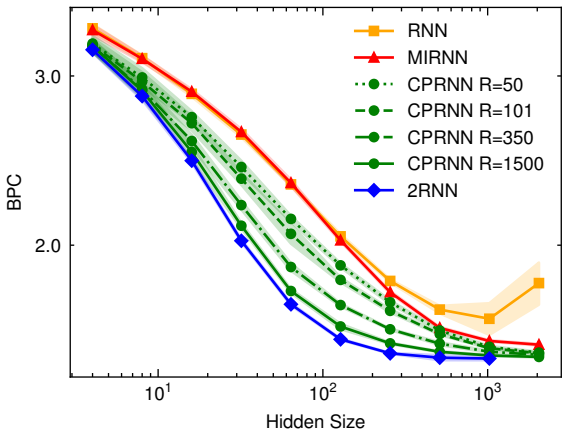

Figure 6. BPC as a function of hidden size of RNN, MIRNN, 2RNN and CPRNN with ranks 50, 101, 350 and 1500. As rank increases, CPRNNs approaches 2RNNs having greater capacity for fixed hidden sizes.

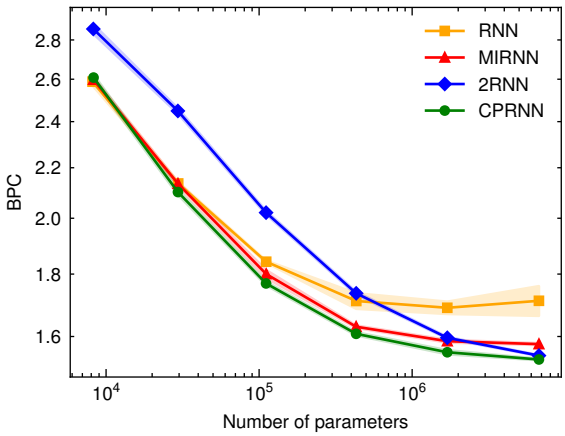

Figure 7. BPC as a function of the number of parameters in RNN, MIRNN, 2RNN and CPRNN. Tuning rank and hidden size allows to find a CPRNN that outperforms all other models of same size.

Importance of hidden size We first assess how the hidden size affects performance for RNNs, MIRNNs, 2RNNs and CPRNNs of various ranks. Figure 6 shows that the performance of all models increases with the hidden dimension. We note a slight inflexion point past a hidden size corresponding to the alphabet size, $d = 101$. Interestingly, this is also where MIRNN starts to separate from RNN, showing the advantage of multiplicative interactions. As the hidden size increases, we note that the CPRNN's performance becomes closer and closer to the 2RNN's. This is also the case for rank values; as the rank increases, the CPRNN's performance gradually matches the 2RNN's. Both these observations are coherent with our theoretical results. More so, it is interesting to note that the 2RNN still outperforms all other models. This is also consistent with theory as 2RNNs encompass all other RNN models considered in this study. It should be noted that since the hidden size (n) directly impacts the number of parameters, the memory requirements of these models are $\mathcal{O}(nd + n^2)$ for RNNs and MIRNNs, $\mathcal{O}(Rn + nd)$ for CPRNNs and $\mathcal{O}(n^2 d)$ for 2RNNs. This difference in model size is also reflected in training time. Reaching early stopping with a hidden size of 512 takes on average 12mins for a RNN, 25mins to 1h for CPRNNs of ranks 1500 to 50, and 1.5h for a 2RNN.

Comparing models of the same size The observations made in the previous paragraph could simply be explained by the respective parameter count of each model, as the number of parameters directly affects expressivity. As such, we conduct an analysis where we keep model size *constant*. Figure 7 gives the BPC of all considered models as a function of parameter count. For CPRNNs, as many configurations of rank and hidden size can correspond to the same parameter count, we choose the best configuration (w.r.t. the validation set) for each fixed model size. The key takeaway of this figure is that it is *always* possible to find a CPRNN which outperforms the other models, regardless of the number of parameters. We also notice that 2RNNs have rather poor performance below the 1M parameter threshold. This can be explained by their small hidden size values (no larger than $n = 64$). Interestingly, 2RNNs match the performance of the other models at a parameter count of 1.7M, corresponding to a hidden size of 128, which is just above the input size of $d = 101$. This is equally the threshold after which vanilla RNNs start overfitting, hence demonstrating the limited capacity of first-order interactions.

Hidden size vs. rank in CPRNNs Going back to the interplay between rank and hidden size, Figure 8 compares the BPC of CPRNNs as a function of the rank for different hidden sizes. We include the BPC values for RNNs and 2RNNs of corresponding hidden sizes on the left and right ends of the figure, respectively. This illustrates how CPRNNs naturally interpolate between first and second-order RNNs. We observe that for high values of hidden size ($n = 1024$), the performance gain is substantial between RNNs and CPRNNs, even for very small ranks. Indeed, with a CPRNN of rank 4, which only increases the number of parameters by 2%, the BPC already decreases by 59% of the relative difference between RNN and 2RNN. Similarly to Figures 6 and 7, we observe a slight inflection point when the rank gets greater then the input size $d = 101$.

Bias-variance tradeoff in CPRNNs Figures 6 and 8 illustrate how increasing the rank and hidden size improves the capacity of CPRNNs, which is in line with our theoretical results. When it comes to comparing models at a fixed size, as done in Figure 7, those two parameters are tied together and vary in opposite directions. Figure 4 presents the BPC of

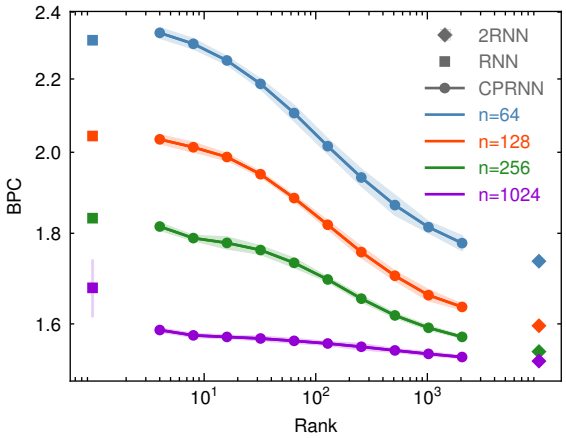

Figure 8. BPC of CPRNN as a function of the rank alongside RNN and 2RNN for hidden sizes 64, 128, 256 and 1024. Variation of the rank in CPRNN interpolates between the performance of the first and second order model.

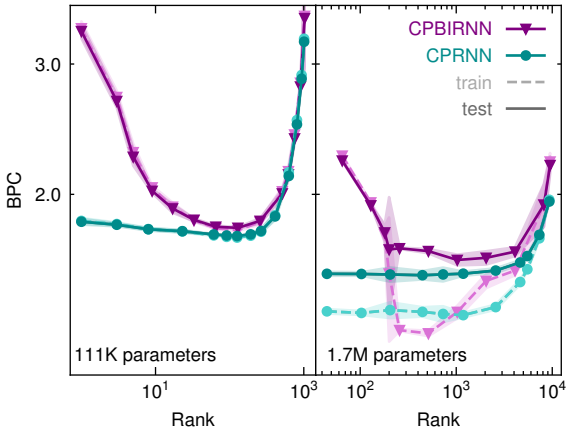

Figure 9. BPC of CPRNN and CPBIRNN for 111K parameters (left) and 1.7M parameters (right). CPBIRNN underfits when rank or hidden size gets low, while in CPRNN the first-order term maintain performance at low rank (i.e. high hidden size). Training and testing curves are approximately the same for 111K parameters, while they show overfitting at 1.7M of parameters.

CPRNNs and CPBIRNNs as a function of rank, keeping the number of parameters fixed at 1.7M and 111K. We observe two underfitting regimes for CPBIRNNs, one corresponding to the rank being too low and the other corresponding to the hidden size being too low (which corresponds to high values of the rank). The optimal combination of rank and hidden size thus lies in between, maximizing neither of these parameters. When the number of parameters is sufficiently large (1.7M), the overfitting regime is more sensitive to the hidden size. By comparing CPBIRNNs and CPRNNs, we see the importance of first-order terms in the interplay between rank and hidden size. In CPRNNs, there is almost no underfitting for small rank. Increasing the hidden size makes the first-order term gain importance in the total number of parameters. Thus, the resulting model is similar to a RNN, preserving some expressive power, whilst without this term, the limitation of small rank is clear.

5. Conclusion

In this work, we formally and empirically characterize the expressivity of RNN models with various levels of second-order interactions and show how they relate to one another. In particular, we use the rank of CPRNNs as a means of interpolation between RNNs (only first-order interactions), MIRNNs (restricted second-order interactions) and 2RNNs (unrestricted second-order interactions). Our analysis sheds light on how of the rank and hidden size affects the expressivity of CPRNNs by providing thresholds above which it saturates. We determine in which measure CPRNNs are strictly more powerful than MIRNNs and when they are equivalent to 2RNNs. We corroborate our theoretical results with language modelling experiments on the Penn Treebank dataset. These empirical results demonstrate that for a fixed number of parameters, we can always find CPRNNs that outperform 2RNNs and MIRNNs. This supports the hypothesis that tensor decompositions are an effective approach to optimize the bias-variance tradeoff in machine learning and motivates us to extend our study to other decompositions such as Tensor-Train (Oseledets, 2011) and tensor ring (Zhao et al., 2016). Such study could be extended to other classes of models, including state space models and their second-order variants. Additionally, it could be applied to other types of problems, as multiplicative interactions have proven beneficial in contexts beyond sequential learning (Cheng et al., 2024). Furthermore, exploring neural network architectures based on tensor network structures opens the possibility to introduce higher-order interactions as well as studying the impact of depth on expressivity along the lines of (Cohen and Shashua, 2016a; Cohen et al., 2016; Cohen and Shashua, 2016b; Sharir and Shashua, 2017; Levine et al., 2018; Alexander et al., 2024; Razin et al., 2024). Another interesting avenue for future work is to explore how these model classes compare in terms of approximation instead of exact realization of the functions they compute, similarly to what was done in, e.g., (Cohen et al., 2016).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgement

M. Lizaire's research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC, Vanier Scholarship) and IVADO (PhD Excellence Scholarship). G. Rabusseau's research is supported by the CIFAR AI Chair program and NSERC. In addition, we acknowledge material support from NVIDIA Corporation in the form of computational resources.

References

Yotam Alexander, Nimrod De La Vega, Noam Razin, and Nadav Cohen. What makes data suitable for a locally connected neural network? a necessary and sufficient condition based on quantum entanglement. *Advances in Neural Information Processing Systems*, 36, 2024.

Muzaffer Ayvaz and Lieven De Lathauwer. Cpd-structured multivariate polynomial optimization. *Frontiers in Applied Mathematics and Statistics*, 8:836433, 2022.

Emilio Rafael Balda, Arash Behboodi, and Rudolf Mathar. A tensor analysis on dense connectivity via convolutional arithmetic circuits. 2018.

R. W. Brockett. On the generic degree of a 3-tensor. *unpublished manuscript*, 1976.

Yining Chen, Sorcha Gilroy, Andreas Maletti, Jonathan May, and Kevin Knight. Recurrent neural networks as weighted language recognizers. *arXiv preprint arXiv:1711.05408*, 2017.

Yixin Cheng, Grigorios Chrysos, Markos Georgopoulos, and Volkan Cevher. Multilinear operator networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=bbCL5aRjUx.

Nadav Cohen and Amnon Shashua. Convolutional rectifier networks as generalized tensor decompositions. In *International conference on machine learning*, pages 955–963. PMLR, 2016a.

Nadav Cohen and Amnon Shashua. Inductive bias of deep convolutional networks through pooling geometry. *arXiv preprint arXiv:1605.06743*, 2016b.

Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on learning theory*, pages 698–728. PMLR, 2016.

Pierre Comon, Jos MF Ten Berge, Lieven De Lathauwer, and Josephine Castaing. Generic and typical ranks of multi-way arrays. *Linear Algebra and its Applications*, 430(11-12):2997–3007, 2009.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, et al. Neural networks and the chomsky hierarchy. *arXiv preprint arXiv:2207.02098*, 2022.

Abhimanyu Dubey, Filip Radenovic, and Dhruv Mahajan. Scalable interpretability via polynomials. *Advances in neural information processing systems*, 35:36748–36761, 2022.

Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

C. Giles, Guo-Zheng Sun, Hsing-Hen Chen, Yee-Chun Lee, and Dong Chen. Higher order recurrent networks and grammatical inference. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989.

Mark W Goudreau, C Lee Giles, Srimat T Chakradhar, and Dong Chen. First-order versus second-order single-layer recurrent neural networks. *IEEE Transactions on Neural Networks*, 5(3):511–513, 1994.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33:1474–1487, 2020.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

James D Hamilton. State-space models. *Handbook of econometrics*, 4:3039–3080, 1994.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

Thomas D Howell. Global properties of tensor rank. *Linear Algebra and its Applications*, 22:9–23, 1978.

Ozan Irsoy and Claire Cardie. Modeling compositionality with multiplicative recurrent neural networks. *CoRR*, abs/1412.6577, 2014.

Siddhant M. Jayakumar, Wojciech M. Czarnecki, Jacob Menick, Jonathan Schwarz, Jack Rae, Simon Osindero, Yee Whye Teh, Tim Harley, and Razvan Pascanu. Multiplicative interactions and where to find them. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rylnK6VtDH.

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

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

Henk AL Kiers. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 14(3):105–122, 2000.

Joe Kileel, Matthew Trager, and Joan Bruna. On the expressive power of deep polynomial neural networks. *Advances in neural information processing systems*, 32, 2019.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Samuel A Korsky and Robert C Berwick. On the computational power of rnns. *arXiv preprint arXiv:1906.06349*, 2019.

Ben Krause, Iain Murray, Steve Renals, and Liang Lu. Multiplicative LSTM for sequence modelling. *ICLR Workshop track*, 2017. URL https://openreview.net/forum?id=SJCS5rXFl.

Yoav Levine, Or Sharir, and Amnon Shashua. Benefits of depth for long-term memory of recurrent networks. 2018.

Tianyu Li, Doina Precup, and Guillaume Rabusseau. Connecting weighted automata, tensor networks and recurrent neural networks through spectral learning. *Machine Learning*, pages 1–35, 2022.

Ruixin Ma, Chuang Wang, and Xin Li. Cp decomposition for fast training of bi-lstm. In *2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 244–248. IEEE, 2021.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. 1993.

William Merrill. Sequential neural networks as automata. *arXiv preprint arXiv:1906.01615*, 2019.

William Merrill, Gail Weiss, Yoav Goldberg, Roy Schwartz, Noah A Smith, and Eran Yahav. A formal hierarchy of rnn architectures. *arXiv preprint arXiv:2004.08500*, 2020.

Tomás Mikolov, Ilya Sutskever, Anoop Deoras, Hai-Son Le, Stefan Kombrink, and Jan Cernocky. Subword language modeling with neural networks. *preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf)*, 8(67), 2012.

Boris Samuilovich Mityagin. The zero set of a real analytic function. *Mathematical Notes*, 107(3-4):529–530, 2020.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. *arXiv preprint arXiv:2303.06349*, 2023.

Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Noam Razin, Tom Verbin, and Nadav Cohen. On the ability of graph neural networks to model interactions between vertices. *Advances in Neural Information Processing Systems*, 36, 2024.

Yahya Sattar, Samet Oymak, and Necmiye Ozay. Finite sample identification of bilinear dynamical systems. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 6705–6711. IEEE, 2022.

Or Sharir and Amnon Shashua. On the expressive power of overlapping architectures of deep learning. *arXiv preprint arXiv:1703.02065*, 2017.

Hava T Siegelmann and Eduardo D Sontag. Analog computation via neural networks. *Theoretical Computer Science*, 131(2):331–360, 1994.

Volker Strassen. Rank and optimal computation of generic tensors. *Linear algebra and its applications*, 52:645–685, 1983.

Zhan Su, Yuqin Zhou, Fengran Mo, and Jakob Grue Simonsen. Language modeling using tensor trains. *arXiv preprint arXiv:2405.04590*, 2024.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1017–1024, 2011.

Andros Tjandra, Sakriani Sakti, Ruli Manurung, Mirna Adriani, and Satoshi Nakamura. Gated recurrent neural tensor network. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 448–455. IEEE, 2016.

Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. Tensor decomposition for compressing recurrent neural network. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Dingheng Wang, Bijiao Wu, Guangshe Zhao, Man Yao, Hengnu Chen, Lei Deng, Tianyi Yan, and Guoqi Li. Kronecker cp decomposition with fast multiplication for compressing rnns. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Gail Weiss, Yoav Goldberg, and Eran Yahav. On the practical computational power of finite precision rnns for language recognition. *arXiv preprint arXiv:1805.04908*, 2018.

Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10): 1550–1560, 1990.

Yuhuai Wu, Saizheng Zhang, Ying Zhang, Yoshua Bengio, and Russ R Salakhutdinov. On multiplicative integration with recurrent neural networks. *Advances in neural information processing systems*, 29, 2016.

Yinchong Yang, Denis Krompass, and Volker Tresp. Tensor-train recurrent neural networks for video classification. In *International Conference on Machine Learning*, pages 3891–3900. PMLR, 2017.

Jinmian Ye, Linnan Wang, Guangxi Li, Di Chen, Shandian Zhe, Xinqi Chu, and Zenglin Xu. Learning compact recurrent neural networks with block-term tensor decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9378–9387, 2018.

Rose Yu, Stephan Zheng, Anima Anandkumar, and Yisong Yue. Long-term forecasting using higher order tensor rnns. *arXiv preprint arXiv:1711.00073*, 2017.

Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.

A. Proofs

In this section, we present the proofs of Theorems 1, 2 and Corollary 4.

A.1. Proof of Theorem 1

Theorem 1. *The following hold for any n and d:*

- $\mathcal{H}_{\text{CPRNN}}(R,n) \subseteq \mathcal{H}_{\text{CPRNN}}(R+1,n)$ *for any R.*
- $\mathcal{H}_{\text{CPRNN}}(R,n) = \mathcal{H}_{\text{CPRNN}}(R+1,n)$ *for any $R \geq R_{max}$.*

Moreover, assuming $n \leq d$:

- $\mathcal{H}_{\text{CPBIRNN}}(R,n) \subsetneq \mathcal{H}_{\text{CPBIRNN}}(R+1,n)$ *for any $R < R_{\text{typ-max}}$ and any real analytic invertible activation function.*
- $\mathcal{H}_{\text{CPRNN}}(R,n) \subsetneq \mathcal{H}_{\text{CPRNN}}(R+1,n)$ *for a linear activation function and any $R < R_{\text{typ-max}}$.*

Proof.

Inclusion for CP(BI)RNN (all bullet points) The proof for the inclusion is the same whether we consider CPRNN or CPBIRNN. We show that for any $h \in \mathcal{H}_{\text{CPRNN}}(R,n)$ there exists $\tilde{h} \in \mathcal{H}_{\text{CPRNN}}(R+1,n)$ such that $h = \tilde{h}$ to conclude that $\mathcal{H}_{\text{CPRNN}}(R,n) \subseteq \mathcal{H}_{\text{CPRNN}}(R+1,n)$. Let h be parameterized by $\mathcal{A}_{CP} = \langle \mathbf{h}^0, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{U}, \mathbf{V}, \mathbf{b}, \sigma \rangle$. Now consider $\tilde{h}$ parameterized by $\tilde{\mathcal{A}}_{CP} = \langle \mathbf{h}^0, \tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \mathbf{U}, \mathbf{V}, \mathbf{b}, \sigma \rangle$ using the same activation function, initial hidden state, first-order and bias terms as $\mathcal{A}_{CP}$, and padding the extra dimension of the factor matrices with 0s, ie. $\tilde{\mathbf{A}}_{ij} = \mathbf{A}_{ij}$ and $\tilde{\mathbf{A}}_{i,R+1} = 0$ for all $i = 1, \cdots, n$ and all $j \leq R$, likewise for the matrices $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{C}}$. One can easily check that with these parameterizations $h = \tilde{h}$.

Saturation (second bullet point) We show that $\mathcal{H}_{\text{CPRNN}}(R,n) \supseteq \mathcal{H}_{\text{CPRNN}}(R+1,n)$ for all $R \geq R_{max}$. This is easy, since any $\tilde{h} \in \mathcal{H}_{\text{CPRNN}}(R+1,n)$ can be computed by a CPRNN whose second-order term is parameterized by a minimal CP decomposition of rank R_{max}. Thus, a function $h \in \mathcal{H}_{\text{CPRNN}}(R,n)$ computed by a CPRNN using this parameterization with zero padding for the extra dimensions is such that $h = \tilde{h}$. We conclude from this and the previous inclusion result ($\mathcal{H}_{\text{CPRNN}}(R,n) \subseteq \mathcal{H}_{\text{CPRNN}}(R+1,n)$) that $\mathcal{H}_{\text{CPRNN}}(R,n) = \mathcal{H}_{\text{CPRNN}}(R+1,n) \quad \forall R \geq R_{max}$.

Strict inclusion for CPBIRNN (third bullet point) The fact that the inclusion is strict for $R < R_{\text{typ-max}}$, i.e. $\mathcal{H}_{\text{CPBIRNN}}(R+1,n) \not\subset \mathcal{H}_{\text{CPBIRNN}}(R,n)$, is the key technical element of the proof of this theorem.
We begin with the easiest case $R < n$, where we can show that not all maps $\mathbf{x}^1 \mapsto \mathbf{h}^1$ computed by CPBIRNNs of rank $R+1$ can be computed by CPBIRNNs of rank R. Consider the linear mapping leading to the first pre-activation vector, $a_1 : \mathbf{x}^1 \mapsto \mathbf{a}^1 \in \mathbb{R}^n$. One can easily check that:

(i) the dimension of the image of a_1 is upper-bounded by R for CPBIRNNs of rank R, i.e. $\dim(a_1(\mathbb{R}^d)) \leq R$,

(ii) as long as $R < n$, there exist parameters $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{h}^0$ of a CPBIRRN of rank $R+1$ such that the image $a_1(\mathbb{R}^d)$ has dimension exactly equal to $R+1$ (e.g. taking $\text{diag}(\mathbf{A}^\top \mathbf{h}^0), \mathbf{B}, \mathbf{C}$ full rank), and

(iii) the dimension of the manifold formed by the hidden vectors $\mathbf{h}^1 = \sigma(\mathbf{a}^1)$ is the same as $a_1(\mathbb{R}^d)$ for any invertible activation functions, since they are homeomorphisms.

Therefore, any function computed by a CPBIRNN of rank $R+1$ for which the manifold of $\sigma(a_1(\mathbb{R}^d))$ is of dimension $R+1$ can not be computed by a CPBIRNN of rank R. This concludes the case $R < n$ ($\leq R_{\text{typ-max}}$): $\mathcal{H}_{\text{CPBIRNN}}(R+1,n) \not\subset \mathcal{H}_{\text{CPBIRNN}}(R,n)$ for any invertible activation function.

We now turn to the case $R \geq n$. Here an argument over the dimension of the first hidden state space can not be made as it is limited by the hidden size n and not the rank R. Therefore, we turn to the computation of the second hidden state $\mathbf{h}^2$, looking at the mapping of the second pre-activation vector $a_2(\mathbf{x}^1, \mathbf{x}^2) \mapsto \mathbf{a}^2$ ($\mathbf{h}^2 = \sigma(\mathbf{a}^2)$). For any $h \in \mathcal{H}_{\text{CPBIRNN}}(R,n)$

parameterized by $\mathcal{A}_{CPBI} = \langle \mathbf{h}^0, \mathbf{A}, \mathbf{B}, \mathbf{C}, \sigma \rangle$, consider the tensor defined by $\boldsymbol{\mathcal{S}}^h_{ijk} = [a_2(\mathbf{e}_i, \mathbf{e}_j)]_k$. For intuition, when the activation function is linear, this tensor computes the second hidden state vector $\mathbf{h}^2 = \boldsymbol{\mathcal{S}}^h \times_1 \mathbf{x}^1 \times_2 \mathbf{x}^2$.

First, one can check that $\boldsymbol{\mathcal{S}}^h = [\![\sigma(\mathbf{B}\mathrm{diag}(\mathbf{A}^\top \mathbf{h}^0)\mathbf{C}^\top)\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ (where σ is applied component-wise) and therefore

$$\mathrm{rank}_{\mathrm{CP}}(\boldsymbol{\mathcal{S}}^h) \leq \mathrm{rank}_{\mathrm{CP}}([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]) \leq R.$$

Second, we show that there exists $\tilde{h} \in \boldsymbol{\mathcal{H}}_{\mathrm{CPBIRNN}}(R+1, n)$ parameterized by $\tilde{\mathcal{A}}_{CPBI} = \langle \tilde{\mathbf{h}}^0, \tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \sigma \rangle$ such that $\mathrm{rank}_{\mathrm{CP}}(\boldsymbol{\mathcal{S}}^{\tilde{h}}) = R+1$. This part of the proof relies on the following lemma, whose proof is in Appendix A.1.1 and leverages the fact that the probability of drawing $\mathbf{A}, \mathbf{B}, \mathbf{C}$ (from a distribution which is continuous w.r.t. the Lebesgues measure) such that $\mathrm{rank}_{\mathrm{CP}}([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]) = R_{\mathrm{typ-max}}$ is strictly positive.

Lemma 2. *For all $n \leq d \leq r \leq R_{\mathrm{typ-max}}(n, d, n)$, there exist matrices $\mathbf{A}, \mathbf{C} \in \mathbb{R}^{n \times r}, \mathbf{B} \in \mathbb{R}^{d \times r}$ and a vector $\mathbf{h}_0 \in \mathbb{R}^n$ such that $\mathrm{rank}_{\mathrm{CP}}([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]) = r$ and $\mathrm{rank}(\mathbf{B}\mathrm{diag}(\mathbf{A}^\top \mathbf{h}_0)\mathbf{C}^\top) = \mathrm{rank}(\tanh(\mathbf{B}\mathrm{diag}(\mathbf{A}^\top \mathbf{h}_0)\mathbf{C}^\top)) = n$ (where $\tanh$ is applied component-wise).*

It follows from this lemma that there exist $\tilde{\mathbf{A}} \in \mathbb{R}^{n \times (R+1)}$, $\tilde{\mathbf{B}} \in \mathbb{R}^{d \times (R+1)}$, $\tilde{\mathbf{C}} \in \mathbb{R}^{n \times (R+1)}$ and $\tilde{\mathbf{h}}_0 \in \mathbb{R}^n$ such that $\mathrm{rank}_{\mathrm{CP}}([\![\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}]\!]) = R+1$ and $\mathrm{rank}(\tilde{\mathbf{B}}\mathrm{diag}(\tilde{\mathbf{A}}^\top \tilde{\mathbf{h}}_0)\tilde{\mathbf{C}}^\top) = \mathrm{rank}(\tanh(\tilde{\mathbf{B}}\mathrm{diag}(\tilde{\mathbf{A}}^\top \tilde{\mathbf{h}}_0)\tilde{\mathbf{C}}^\top)) = n$. We thus have that $\boldsymbol{\mathcal{S}}^{\tilde{h}} = [\![\mathbf{M}\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}]\!]$ where $\mathbf{M} \in \mathbb{R}^{d \times n}$ is a left invertible matrix (both when σ is the hyperbolic tangent and the identity), which in turn implies that (see Lemma 4)

$$\mathrm{rank}_{\mathrm{CP}}(\boldsymbol{\mathcal{S}}^{\tilde{h}}) = \mathrm{rank}_{\mathrm{CP}}([\![\mathbf{M}\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}]\!]) = \mathrm{rank}_{\mathrm{CP}}([\![\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}]\!]) = R+1.$$

We conclude that $\tilde{h} \notin \mathcal{H}_{\mathrm{CPBIRNN}}(R, n)$ since $\mathrm{rank}_{\mathrm{CP}}(\boldsymbol{\mathcal{S}}^{\tilde{h}}) = R+1$ while $\forall h \in \boldsymbol{\mathcal{H}}_{\mathrm{CPBIRNN}}(R, n)$ we have $\mathrm{rank}_{\mathrm{CP}}(\boldsymbol{\mathcal{S}}^h) \leq R$.

Strict inclusion for linear CPRNNs (fourth bullet point) In the case of linear CPRNNs, we again consider the function computing the second hidden state, but this time we only look at the term with second-order interactions between $\mathbf{x}^1$ and $\mathbf{x}^2$ and use a similar argument to the one used for the strict inclusion of CPBIRNNs. The argument is however slightly simpler in this case and can be used for both cases $R < n$ and $R \geq n$.

For any $h \in \mathcal{H}_{\mathrm{CPRNN}}(R, n)$ parameterized by $\mathcal{A}_{CP} = \langle \mathbf{h}^0, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{U}, \mathbf{V}, \mathbf{b}, \sigma = \mathbf{I} \rangle$, let $h_2 : (\mathbf{x}^1, \mathbf{x}^2) \mapsto \mathbf{h}^2$ be the associated function mapping the two first inputs to the second hidden state. First observe that since $\sigma = \mathbf{I}$, the associated h_2 can be decomposed in four terms: $h_2(\mathbf{x}^1, \mathbf{x}^2) = \alpha(\mathbf{x}^1, \mathbf{x}^2) + \beta(\mathbf{x}^1) + \gamma(\mathbf{x}^2) + \delta$ where α is the bilinear map containing only the second order terms (i.e. α is a linear map of the Kronecker product of $\mathbf{x}^1$ and $\mathbf{x}^2$). Note that this decomposition is such that given two functions $h, \hat{h}$, if $\alpha \neq \hat{\alpha}$ then $h \neq \hat{h}$. One can check that

$$\alpha(\mathbf{x}^1, \mathbf{x}^2) = [\![(\mathbf{U}^\top + \mathbf{B}\mathrm{diag}(\mathbf{A}^\top \mathbf{h}^0)\mathbf{C}^\top)\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \times_1 \mathbf{x}_1 \times_2 \mathbf{x}_2$$

It follows that defining the tensor $\boldsymbol{\mathcal{S}}^\alpha \in \mathbb{R}^{d \times d \times n}$ by $\boldsymbol{\mathcal{S}}^\alpha_{ijk} = [\alpha(\mathbf{e}_i, \mathbf{e}_j)]_k$, we have

$$\mathrm{rank}_{\mathrm{CP}}(\boldsymbol{\mathcal{S}}^\alpha) \leq \mathrm{rank}_{\mathrm{CP}}([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]) \leq R.$$

Now, since $R < R_{\mathrm{typ-max}}$, there exist parameters of a function $\tilde{h} \in \mathcal{H}_{\mathrm{CPRNN}}(R+1, n)$ such that $\mathrm{rank}_{\mathrm{CP}}([\![\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}]\!]) = R+1$. Moreover, since $n \leq d$, $\tilde{\mathbf{U}} \in \mathbb{R}^{n \times d}$ can always be chosen such that $(\tilde{\mathbf{U}}^\top + \tilde{\mathbf{B}}\mathrm{diag}(\tilde{\mathbf{A}}^\top \tilde{\mathbf{h}}^0)\tilde{\mathbf{C}}^\top) \in \mathbb{R}^{d \times n}$ is left invertible, in which case (see again Lemma 4)

$$\mathrm{rank}_{\mathrm{CP}}(\boldsymbol{\mathcal{S}}^{\tilde{\alpha}}) = \mathrm{rank}_{\mathrm{CP}}([\![(\tilde{\mathbf{U}}^\top + \tilde{\mathbf{B}}\mathrm{diag}(\tilde{\mathbf{A}}^\top \tilde{\mathbf{h}}^0)\tilde{\mathbf{C}}^\top))\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}]\!]) = \mathrm{rank}_{\mathrm{CP}}([\![\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}]\!]) = R+1.$$

By construction we thus have $\mathrm{rank}_{\mathrm{CP}}(\boldsymbol{\mathcal{S}}^{\tilde{\alpha}}) = R+1$ while $\mathrm{rank}_{\mathrm{CP}}(\boldsymbol{\mathcal{S}}^\alpha) \leq R$ for all $h \in \mathcal{H}_{\mathrm{CPRNN}}(R, n)$, hence $\tilde{h} \notin \mathcal{H}_{\mathrm{CPRNN}}(R, n)$.

□

A.1.1. Proof of Lemma 2

Lemma 2. *For all $n \leq d \leq r \leq R_{\mathrm{typ-max}}(n, d, n)$, there exist matrices $\mathbf{A}, \mathbf{C} \in \mathbb{R}^{n \times r}, \mathbf{B} \in \mathbb{R}^{d \times r}$ and a vector $\mathbf{h}_0 \in \mathbb{R}^n$ such that $\mathrm{rank}_{\mathrm{CP}}([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]) = r$ and $\mathrm{rank}(\mathbf{B}\mathrm{diag}(\mathbf{A}^\top \mathbf{h}_0)\mathbf{C}^\top) = \mathrm{rank}(\tanh(\mathbf{B}\mathrm{diag}(\mathbf{A}^\top \mathbf{h}_0)\mathbf{C}^\top)) = n$ (where $\tanh$ is applied component-wise).*

The proof of the lemma relies on the following result whose proof can be found in (Mityagin, 2020)[‡].

Lemma 3. *Let $A(x)$ be a real analytic function on (a connected open domain U of) $\mathbb{R}^d$. If A is not identically zero, then its zero set*

$$Z(A) := \{x \in U \mid A(x) = 0\}$$

has a zero measure.

Proof of Lemma 2. Let $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{h}_0$ be randomly drawn from a distribution which is continuous w.r.t. the Lebesgue measure. On the one hand, since $r \leq R_{\text{typ}-\max}(n, d, n)$, the measure of the set of matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ such that $\text{rank}_{\text{CP}}(\llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket) = r$ is stricly positive, and thus the probability that the random matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are such that $\text{rank}_{\text{CP}}(\llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket) = r$ is strictly greater than 0. On the other hand, consider the real analytic function

$$F(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{h}_0) = \det[(\mathbf{B}\text{diag}(\mathbf{A}^\top \mathbf{h}_0)\mathbf{C}^\top)_{1:n,:}] \cdot \det[\tanh(\mathbf{B}\text{diag}(\mathbf{A}^\top \mathbf{h}_0)\mathbf{C}^\top)_{1:n,:}]$$

where the notation $\mathbf{M}_{1:n,:}$ denotes the matrix containing the first n rows of $\mathbf{M}$. One can check that by letting $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}$ be identity matrices padded with 0's and $\tilde{\mathbf{h}}_0$ be a vector full of ones, we have $F(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \tilde{\mathbf{h}}_0) \neq 0$, and thus F is not identically zero. It then follows from Lemma 3 that the set of $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{h}_0$ such that $F(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{h}_0) = 0$ has measure 0. Hence, with probability one, both $\mathbf{B}\text{diag}(\mathbf{A}^\top \mathbf{h}_0)\mathbf{C}^\top$ and $\tanh(\mathbf{B}\text{diag}(\mathbf{A}^\top \mathbf{h}_0)\mathbf{C}^\top)$ are full rank, i.e. of rank n (since $n \leq d \leq r$). Combining these two observations lead to the existence of matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and a vector $\mathbf{h}_0$ satisfying the claim of the lemma. □

A.2. Proof of Theorem 2

Theorem 2. *The following hold for any d and n:*

- $\mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n) \subseteq \mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ *for any R and n.*
- $\mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n) = \mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ *for any $n \geq R$ and linear activation function.*

Moreover, assuming $n \leq d$:

- $\mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n) \subsetneq \mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ *for any $n < R$ and any invertible activation function satisfying $\sigma(0) = 0$.*

Proof.

Inclusion (all bullet points) We first show that $\mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n) \subseteq \mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ (both when $n < R$ and $n \geq R$). Consider a function $l \circ h \in \mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n)$ with l defined by $l(\mathbf{v}) = \mathbf{W}\mathbf{v}$ for some $\mathbf{W}$ in $\mathbb{R}^{q \times n}$ and h parameterized by the CPBIRNN $\mathcal{A}_{CPBI} = \langle \mathbf{h}^0, \mathbf{A}, \mathbf{B}, \mathbf{C}, \sigma \rangle$. Now consider the function $\tilde{l} \circ \tilde{h}$ in $\mathcal{L}^{n+1} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ with $\tilde{l}$ defined by $\tilde{l}(\mathbf{v}) = \tilde{\mathbf{W}}\mathbf{v}$ where $\tilde{\mathbf{W}}^\top = \begin{pmatrix} \mathbf{W}^\top \\ 0 \cdots 0 \end{pmatrix}$ and $\tilde{h}$ parameterized by $\tilde{\mathcal{A}}_{CPBI} = \langle \tilde{\mathbf{h}}^0, \tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \sigma \rangle$ where $\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{A} \\ 0 \cdots 0 \end{pmatrix}$, $\tilde{\mathbf{C}} = \begin{pmatrix} \mathbf{C} \\ 0 \cdots 0 \end{pmatrix}$ and $\tilde{\mathbf{h}}_0 = \begin{pmatrix} \mathbf{h}_0 \\ 0 \end{pmatrix}$. One can check that $\tilde{l} \circ \tilde{h} = l \circ h$. This construction thus shows that for any function in $\mathcal{L}^n \circ \mathcal{H}_{\text{CPBIRNN}}(R, n)$ we can find an equivalent function in $\mathcal{L}^{n+1} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$, i.e. $\mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n) \subseteq \mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$.

Saturation (second bullet point) Since we already showed that $\mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n) \subseteq \mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ for any invertible activation function (including linear ones), we just need to show that $\mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1) \subseteq \mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n)$. Let $\tilde{h} \in \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ parameterized by $\tilde{\mathcal{A}}_{CPBI} = \langle \tilde{\mathbf{h}}^0, \tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \sigma = \mathbf{I} \rangle$ and $\tilde{l} : \mathbf{v} \mapsto \tilde{\mathbf{W}}\mathbf{v} \in \mathcal{L}^{n+1,q}$. Since $R < n+1$, we can factorize $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{C}}$ into $\tilde{\mathbf{A}} = \mathbf{P}\mathbf{A}$ and $\tilde{\mathbf{C}} = \mathbf{C}\mathbf{Q}$ where $\mathbf{A}, \mathbf{C} \in \mathbb{R}^{n \times R}$ and $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{n+1 \times n}$ are rank n. The functions $h \in \mathcal{H}_{\text{CPBIRNN}}(R, n)$ parameterized by $\mathcal{A}_{CPBI} = \langle \mathbf{h}^0 = \mathbf{P}^\top \tilde{\mathbf{h}}^0, \mathbf{A}, \tilde{\mathbf{B}}, \mathbf{C}, \sigma = \mathbf{I} \rangle$ and $l : \mathbf{v} \mapsto \mathbf{Q}^\top \tilde{\mathbf{W}}\mathbf{v} \in \mathcal{L}^{n,q}$ are such that $\tilde{l} \circ \tilde{h} = l \circ h$ which shows that $\mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1) \subseteq \mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n)$. Note that without the restriction of a linear activation function is key here as it is what allows the factorization matrix $\mathbf{Q}^\top$ to be introduced in the parameterization of l ($\mathbf{Q}^\top \tilde{\mathbf{W}}$).

[‡]Mityagin, Boris Samuilovich. "The zero set of a real analytic function." Mathematical Notes 107.3-4 (2020): 529-530.

Strict inclusion (third bullet point) We now show that the inclusion is strict for $n < R$, i.e., that

$$\mathcal{L}^{n+1,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1) \not\subset \mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n).$$

It suffices to show the result for $q = 1$ which we will do. Indeed, suppose the result is true for $q = 1$ and let $\tilde{l} \circ \tilde{h} \in \mathcal{L}^{n+1,1} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ be s.t. $\tilde{l} \circ \tilde{h} \neq l \circ h$ for all $l \circ h \in \mathcal{L}^{n,1} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n)$. It is easy to see that, for any $q \geq 1$, the map $\bar{l} : \mathbf{v} \mapsto (\tilde{l}(\mathbf{v})\ 0\ \cdots\ 0)^\top \in \mathcal{L}^{n+1,q}$ is such that $\bar{l} \circ \tilde{h} \notin \mathcal{L}^{n,q} \circ \mathcal{H}_{\text{CPBIRNN}}(R, n)$. Then the proof goes as follow:

(i) First, observe that for any $h \in \mathcal{H}_{\text{CPBIRNN}}(R, n)$ parameterized by $\mathcal{A}_{CPBI} = \langle \mathbf{h}^0, \mathbf{A}, \mathbf{B}, \mathbf{C}, \sigma \rangle$, the image of the first pre-activation mapping $a_1 : \mathbf{x}^1 \mapsto \mathbf{a}^1 = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \times_1 \mathbf{h}^0 \times_2 \mathbf{x}^1$ is a linear space of dimension at most n, i.e. $\dim(a_1(\mathbb{R}^d)) \leq n$.

(ii) Second, consider $\tilde{h} \in \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ parameterized by $\tilde{\mathcal{A}}_{CPBI} = \langle \tilde{\mathbf{h}}^0, \tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \sigma \rangle$ such that $\text{diag}(\tilde{\mathbf{A}}^\top \tilde{\mathbf{h}}_0)$, $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{C}}$ are full rank matrices (e.g. $\tilde{\mathbf{A}}_{1:}$ is a row of 1's and $\tilde{\mathbf{h}}^0 = \mathbf{e}_1$). Since $n < R \leq d$, one can check that the image of $\tilde{a}_1 : \mathbf{x}^1 \mapsto \tilde{\mathbf{a}}^1$ has dimension $n+1$, i.e. $\dim(\tilde{a}_1(\mathbb{R}^d)) = n+1$.

(iii) Third, let $z : \mathbf{v} \mapsto (\mathbf{v}\ 0)^\top$ be a mapping that augments the dimension while leaving the original transformation intact. We have that $z \circ a_1(\mathbb{R}^d)$ and $\tilde{a}_1(\mathbb{R}^d)$ are both linear subspaces of $\mathbb{R}^{n+1}$ and that $\dim(z \circ a_1(\mathbb{R}^d)) < \dim(\tilde{a}_1(\mathbb{R}^d))$.

It thus follows from Lemma 1 that $\mathcal{L}^{n+1,1} \circ \sigma \circ \tilde{a}_1 \not\subset \mathcal{L}^{n+1,1} \circ \sigma \circ z \circ a_1$. Since for any activation function such that $\sigma(0) = 0$ we have $\mathcal{L}^{n,1} \circ \sigma = \mathcal{L}^{n+1,1} \circ \sigma \circ z$, we conclude that $\mathcal{L}^{n+1,1} \circ \sigma \circ \tilde{a}_1 \not\subset \mathcal{L}^{n,1} \circ \sigma \circ a_1$. We thus have shown that there exists $\tilde{h} \in \mathcal{H}_{\text{CPBIRNN}}(R, n+1)$ such that for any $h \in \mathcal{H}_{\text{CPBIRNN}}(R, n)$ we have $\mathcal{L}^{n+1,1} \circ \tilde{h}_1 \not\subset \mathcal{L}^{n,1} \circ h_1$ (where h_1 and $\tilde{h}_1$ denote the function mapping the first input vector to the first hidden state of the respective CPRNNs), which concludes the proof.

□

A.2.1. Proof of Lemma 1

Lemma 1. *Let V be a vector space of dimension d, $\phi, \psi : \mathcal{X} \to V$ two maps whose images $\phi(\mathcal{X})$ and $\psi(\mathcal{X})$ are subspaces of V and σ an homeomorphism. Lastly, let $\mathcal{L}(V)$ denote the set of all linear forms on V (i.e. $\mathcal{L}(V)$ is the dual space V^*).*

If $\dim(\phi(\mathcal{X})) > \dim(\psi(\mathcal{X}))$, then $\mathcal{L}(V) \circ \sigma \circ \phi \not\subset \mathcal{L}(V) \circ \sigma \circ \psi$.

Proof. We suppose $\mathcal{L}(V) \circ \sigma \circ \phi \subseteq \mathcal{L}(V) \circ \sigma \circ \psi$ and show it leads to a contradiction. Let $l_1, \ldots l_d$ be d linear forms from $\mathcal{L}(V)$ that are linearly independent. By hypothesis, there exist $\tilde{l}_1, \ldots \tilde{l}_d$ such that $l_i \circ \sigma \circ \phi = \tilde{l}_i \circ \sigma \circ \psi$ for all $i = 1, \ldots, d$. Consider the mappings $\nu, \tilde{\nu} : V \to V$ given by $\nu : u \mapsto \sum_{i=1}^d l_i(u) v_i$ and $\tilde{\nu} : u \mapsto \sum_{i=1}^d \tilde{l}_i(u) v_i$ where $v_1, \ldots, v_d$ is an arbitrary basis of V. The previous equality ($l_i \circ \sigma \circ \phi = \tilde{l}_i \circ \sigma \circ \psi$) implies that $\nu \circ \sigma \circ \phi = \tilde{\nu} \circ \sigma \circ \psi$. On the one hand, since ν is invertible by construction, $\nu \circ \sigma$ is an homeomorphism and the dimension of the manifold (noted $\dim_{\mathcal{M}}$) of the image of $\nu \circ \sigma \circ \phi$ is the same as the dimension of the image of ϕ, i.e.

$$\dim_{\mathcal{M}}(\nu \circ \sigma \circ \psi(\mathcal{X})) = \dim(\phi(\mathcal{X})).$$

On the other hand, since $\tilde{\nu}$ is a linear transformation it can not increase the dimension of the manifold, thus

$$\dim_{\mathcal{M}}(\tilde{\nu} \circ \sigma \circ \psi(\mathcal{X})) \leq \dim(\psi(\mathcal{X})) < \dim(\phi(\mathcal{X})).$$

This is a contradiction since the hypothesis led to $\nu \circ \sigma \circ \phi = \tilde{\nu} \circ \sigma \circ \psi$ which implies $\dim_{\mathcal{M}}((\tilde{\nu} \circ \sigma \circ \psi)(\mathcal{X})) = \dim_{\mathcal{M}}((\tilde{\nu} \circ \sigma \circ \phi)(\mathcal{X}))$. □

A.3. Proof of Corollary 4

Corollary 4. *Assuming $n \leq d$, for any $R > n$,*

- $\mathcal{H}_{\text{MIRNN}}(n) \subseteq \mathcal{H}_{\text{CPRNN}}(R, n)$
- $\mathcal{H}_{\text{MIRNN}}(n) \subsetneq \mathcal{H}_{\text{CPRNN}}(R, n)$ *for linear activation function*

Proof.

Inclusion We show that for any $\mathcal{H}_{\text{MIRNN}}(n) \subseteq \mathcal{H}_{\text{CPRNN}}(R, n)$ for $R \geq n$. Note the inclusion is valid for CPRNNs and without restrictions on the first order and biais terms of MIRNNs. First observe that an equality between the second-order terms of a CPRNN and a MIRNN corresponds to $\sum_{r=1}^{R} \mathbf{A}_{ir}\mathbf{B}_{jr}\mathbf{C}_{kr} = \boldsymbol{\alpha}_k \mathbf{V}_{ki}\mathbf{U}_{kj}$ for all $i, k = 1, \cdots, n$ and $j = 1, \cdots, d$. Thus, for any $\bar{h} \in \mathcal{H}_{\text{MIRNN}}(n)$ parameterized by $\mathcal{A}_{MI} = \langle \mathbf{h}^0, \bar{\mathbf{U}}, \bar{\mathbf{V}}, \boldsymbol{\alpha}, \boldsymbol{\beta_1}, \boldsymbol{\beta_2}, \mathbf{b}, \sigma \rangle$ a CPRNN given by $\mathcal{A}_{CP} = \langle \mathbf{h}^0, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{U} = \boldsymbol{\beta_1} \odot \bar{\mathbf{U}}, \mathbf{V} = \boldsymbol{\beta_2} \odot \bar{\mathbf{V}}, \mathbf{b}, \sigma \rangle$ with

$$\mathbf{A}_{ir} = \begin{cases} \bar{\mathbf{V}}_{ir} & \text{if } r, i = 1 \dots n \\ \mathbf{0} & \text{else} \end{cases} \quad \mathbf{B}_{jr} = \begin{cases} \bar{\mathbf{U}}_{jr} & \text{if } r = 1 \dots n \; j = 1 \dots d \\ 0 & \text{else} \end{cases} \quad \mathbf{C}_{kr} = \begin{cases} \boldsymbol{\alpha}_k & \text{if } k = r \\ 0 & \text{else} \end{cases}$$

computes a function h that is equal to $\bar{h}$.

Strict inclusion The strict inclusion comes from the observation that since we have $\mathcal{H}_{\text{MIRNN}}(n) \subseteq \mathcal{H}_{\text{CPRNN}}(R, n)$. As Theorem 1 states that for linear activation function and $R < R_{\max}$ $\mathcal{H}_{\text{CPRNN}}(R, n) \subsetneq \mathcal{H}_{\text{CPRNN}}(R + 1, n)$, it follows that for any $R > n$ and a linear activation function $\mathcal{H}_{\text{MIRNN}}(n) \subsetneq \mathcal{H}_{\text{CPRNN}}(R, n)$. □

A.4. Additional Lemmas

Lemma 4. *Let $\boldsymbol{\mathcal{T}}$ be a tensor of CP rank R and let $\boldsymbol{\mathcal{T}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ be a minimal (i.e. rank R) CP decomposition of $\boldsymbol{\mathcal{T}}$. Then, for any left invertible matrix $\mathbf{M}$, $\text{rank}_{\text{CP}}(\boldsymbol{\mathcal{T}}) = \text{rank}_{\text{CP}}([\![\mathbf{MA}, \mathbf{B}, \mathbf{C}]\!])$.*

Proof. Let $\mathbf{a}_1, \cdots, \mathbf{a}_R$ be the columns of $\mathbf{A}$, $\mathbf{b}_1, \cdots, \mathbf{b}_R$ the columns of $\mathbf{B}$ and $\mathbf{c}_1, \cdots, \mathbf{c}_R$ be the columns of $\mathbf{C}$.

First, it is trivial to show that $R \geq \text{rank}_{\text{CP}}([\![\mathbf{MA}, \mathbf{B}, \mathbf{C}]\!])$ since $[\![\mathbf{MA}, \mathbf{B}, \mathbf{C}]\!]$ is a rank R CP decomposition. To show the result, we need to show that this CP decomposition is minimal. Suppose it is not the case, i.e., there exists a rank $S < R$ CP decomposition $[\![\mathbf{MA}, \mathbf{B}, \mathbf{C}]\!] = [\![\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}]\!]$ where $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}$ all have S columns. Then, one can easily check that $\boldsymbol{\mathcal{T}} = [\![\mathbf{M}^+\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}]\!]$, where $\mathbf{M}^+$ is the left inverse of $\mathbf{M}$, i.e. $\boldsymbol{\mathcal{T}}$ admits a CP decomposition of rank $S < R$, a contradiction.

□