Approximation Theory of Convolutional Architectures for Time Series Modelling

Haotian Jiang [1] **Zhong Li** [2] **Qianxiao Li** [1 3]

Abstract

We study the approximation properties of convolutional architectures applied to time series modelling, which can be formulated mathematically as a functional approximation problem. In the recurrent setting, recent results reveal an intricate connection between approximation efficiency and memory structures in the data generation process. In this paper, we derive parallel results for convolutional architectures, with WaveNet being a prime example. Our results reveal that in this new setting, approximation efficiency is not only characterised by memory, but also additional fine structures in the target relationship. This leads to a novel definition of spectrum-based regularity that measures the complexity of temporal relationships under the convolutional approximation scheme. These analyses provide a foundation to understand the differences between architectural choices for time series modelling and can give theoretically grounded guidance for practical applications.

1. Introduction

While deep learning has evolved to be a powerful tool to model temporal relationships, the choice of architectures significantly affects performance. Classical recurrent neural networks (RNNs) and recent adaptation of convolutional neural networks (CNNs) to time series data represent two popular classes of architectural choices. Empirical works show that depending on the application setting, either one can have an advantage over the other (Yin et al., 2017; Bai et al., 2018; Banerjee et al., 2019). However, a concrete theoretical understanding of how such differences arise remains largely unexplored.

[1]Department of Mathematics, National University of Singapore [2]School of Mathematical Science, Peking University [3]Institute of High Performance Computing, A*STAR, Singapore. Correspondence to: Qianxiao Li <qianxiao@nus.edu.sg>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

The recent work of Li et al. (2021) formulated the temporal modelling task into a functional analysis problem, and uncovered relationships between approximation efficiency of RNNs and memory structures of the target relationship. In this paper, we develop parallel results for CNNs in the functional approximation setting. On the one hand, these results provide approximation guarantees and rate estimations for CNNs applied to time series modelling. On the other hand, they allow us to concretely understand the differences between CNNs and RNNs in terms of their approximation capabilities for temporal relationships, serving as a starting point to bridge theories and applications.

Our main contributions are:

1. We develop universal approximation results for CNNs when applied to model temporal relationships in the linear setting, which shows that the approximation efficiency is characterised by both memory structures and a certain spectrum-based regularity of the target relationship.

2. We make rigorous comparisons between CNNs and RNNs in the approximation setting, where we show that the targets that can be easily approximated are completely different for these two architectures. This provides a theoretical foundation for building principled model selection strategies for deep learning in dynamical settings.

The paper is organised as follows. We discuss related work in Section 2. In Section 3, we formulate the approximation problem precisely and introduce the RNN and CNN hypothesis spaces under this formulation. The approximation results of CNNs are presented in Section 4. In Section 5, we make a concrete comparison between CNNs and RNNs in terms of approximation capabilities for different classes of target temporal relationships. The proofs of all presented results are found in the appendix.

1.1. Notations and definitions.

We use boldface letters $\boldsymbol{x}$ to denote discrete temporal sequences, and $x(t)$ to denote the vector/scalar value of the

sequence at the time index t. We use $\boldsymbol{x}_{[t_1,t_2]}$ to denote values of the sequence $\boldsymbol{x}$ from t_1 to t_2, including the end points. Subscripts such as $\boldsymbol{x}_i$, $\boldsymbol{x}_m$ are used to specify dimensional indices. All other dependencies are written as superscripts, e.g. $\boldsymbol{x}^{(K)}$ means that the temporal sequence $\boldsymbol{x}$ depends on K. For a discrete sequence $\boldsymbol{\rho} : \mathbb{N} \to \mathbb{R}^d$, define the radius of $\boldsymbol{\rho}$ by $r(\boldsymbol{\rho}) = \sup\{s : \rho(s) \neq 0\}$. We refer $[0, r(\boldsymbol{\rho})]$ as its support. We use $|\cdot|$ to denote the Euclidean norm of a vector.

We set the notation for the central operation under study, namely the dilated convolution for discrete sequences.

Definition 1. *Let $\boldsymbol{f} : \mathbb{Z} \to \mathbb{R}^d$, $\boldsymbol{g} : \mathbb{Z} \to \mathbb{R}^d$ be two discrete sequences. Define the following discrete dilated convolution:*

$$(\boldsymbol{f} *_l \boldsymbol{g})(t) = \sum_{s\in\mathbb{Z}} f(s)^\top g(t - ls). \tag{1}$$

When $l = 1$, this is the usual convolution, and we denote it by $*$.

2. Related work

In this section, we discuss some related work to the results presented here. On the application side, convolution-based models are gradually becoming popular for a variety of time series applications. For example, WaveNet (van den Oord et al., 2016) is a generative model for processing and generating raw audio. Since its emergence, WaveNet and its variants (van den Oord et al., 2018; Jin et al., 2018; Kalchbrenner et al., 2018; Prenger et al., 2019; Paine et al., 2016) have been successfully applied to a variety of time series modelling problems in natural language processing. A more detailed review can be found in Boilard et al. (2019). While convolutional architectures have become an important alternative to classical recurrent neural networks, these investigations focus on applications, lacking theoretical results to understand the origin of their superior performance on practical tasks.

On the theoretical side, a number of universal approximation results for CNNs have been obtained in the image processing setting. For example, Zhou (2020a;b) prove that (simplified) deep CNNs are universal approximators, and the approximation rate is characterised by Sobolev norms of target functions. Bao et al. (2019) gives a theoretical interpretation on the reason why the state-of-the-art CNNs can achieve high classification accuracy. It is shown that the hierarchical compositional structures in target functions can help to (exponentially) reduce the parameters needed compared with fully connected neural networks achieving the same approximation accuracy. Beyond the approximation theory, Oono & Suzuki (2019) studies the estimation error rates of a ResNet-type of CNN applied to the Barron and Hölder classes. All of these works share one thing in common, namely the targets to be approximated are functions defined in finite space domains, i.e. images. By contrast, our approximation results here are for the targets defined in infinite time domains, where the memory of data plays an important role. We will see that this leads to very different approximation problems.

On the recurrent front, the approximation properties of RNNs have been investigated in a number of works, including both in discrete time (Matthews, 1993; Doya, 1993; Schäfer & Zimmermann, 2006; 2007) and continuous time (Funahashi & Nakamura, 1993; Chow & Xiao-Dong Li, 2000; Li et al., 2005; Maass et al., 2007; Nakamura & Nakagawa, 2009). Most of these study settings where the target relationships are generated from hidden dynamical systems (in the form of difference or differential equations). Recently, Li et al. (2021) studies the general setting where the target relationships are represented as functionals, and derived the approximation theory of RNNs that revealed the connection between approximation and memory: it takes an exponentially large number of neurons to approximate the target with memory that decays slowly. This sheds some light on the interaction of the structure of RNNs and the nature of relationships to be captured. Following this line of enquiry, the purpose of the present paper is to develop parallel approximation results for CNNs to highlight such interactions. In particular, this work complements previous theoretical analyses in the recurrent setting and allows one to characterise the key difference between recurrent and convolutional approaches for time series modelling.

3. Problem Formulation

3.1. Functional formulation of supervised learning for temporal data

The temporal supervised learning task can be mathematically formulated as follows. Suppose we are given an input sequence $\boldsymbol{x}$ indexed by a time parameter $t \in \mathbb{Z}$. We want to predict the corresponding output sequence $\boldsymbol{y}$ at each time step, or up to some terminal time step $y(T)$. The mapping between $\boldsymbol{x}$ and $\boldsymbol{y}$ can be described by a sequence of functionals:

$$\{y(t) = H_t(\boldsymbol{x}) : t \in \mathbb{Z}\}. \tag{2}$$

That is, the output at each time step may potentially depend on the entire input sequence through a time-dependent functional. The goal is to learn this sequence of functionals $\boldsymbol{H} = \{H_t : t \in \mathbb{Z}\}$.

As an illustrative example, we can consider a standard benchmark example known as the adding problem (Hochreiter & Schmidhuber, 1997): each output $y(t)$ is equal to the weighted cumulative sum of inputs $x(s)$ for $s \leq t$, i.e. $y(t) = \sum_{-\infty}^{t} \alpha(s)x(s)$, where α denotes the weights. In this case, the sequence of functionals $\{H_t(\boldsymbol{x}) =$

$\sum_{-\infty}^{t} \alpha(s)x(s) : t \in \mathbb{Z}\}$ can be viewed as the target temporal relationship, or ground truth.

In supervised learning, we need to define the input space, output space and concept space precisely. We note that vector-valued discrete temporal sequences can also be understood as functions from a time index set $\mathcal{I}$ into the real vectors. We denote this set of functions by $\boldsymbol{c}(\mathcal{I}, \mathbb{R}^d)$. In this paper, $\mathcal{I}$ is taken to be $\mathbb{Z}$ or $\mathbb{N}$ depending on the specific setting. For $\boldsymbol{x} \in \boldsymbol{c}(\mathcal{I}, \mathbb{R}^d)$, we define its norm by $\|\boldsymbol{x}\|_2 := \sqrt{\sum_{s\in\mathcal{I}} |x(s)|^2}$. We use $\boldsymbol{c}_0(\mathbb{N}, \mathbb{R}^d)$ to denote the sequence that converges to 0 at infinity.

Define the input space by

$$\mathcal{X} = \Big\{\boldsymbol{x} \in \boldsymbol{c}(\mathbb{Z}, \mathbb{R}^d) : \sum_{s\in\mathbb{Z}} |x(s)|^2 < \infty\Big\}. \tag{3}$$

This is the usual ℓ^2 sequence space, which is a Banach space with the norm $\|\boldsymbol{x}\|_{\mathcal{X}} := \|\boldsymbol{x}\|_2 = \sqrt{\sum_{s\in\mathbb{Z}} |x(s)|^2}$.

Since vector-valued outputs can be handled by considering each dimension individually, we can restrict our attention to the case where the output time series are real-valued. That is, define the output space

$$\mathcal{Y} = \boldsymbol{c}(\mathbb{Z}, \mathbb{R}). \tag{4}$$

3.2. RNN and CNN hypothesis spaces

We now introduce the RNN and CNN architectures in the functional approximation language, which leads to different types of hypothesis spaces. We start with the recurrent setting. The simplest recurrent neural network with the linear readout layer is given by

$$\begin{aligned} h(t+1) &= \sigma(Wh(t) + Ux(t)), \\ \hat{y}(t) &= c^\top h(t), \\ \text{with } c \in \mathbb{R}^m, W &\in \mathbb{R}^{m\times m}, U \in \mathbb{R}^{m\times d}. \end{aligned} \tag{5}$$

Here, $h \in \mathbb{R}^m$ is the hidden state and m denotes the width of RNNs, which determines the model complexity. Note that we do not include a bias term here because it can be absorbed into the hidden state h.

This dynamics defines a family of functionals

$$\begin{aligned} \mathcal{H}_{\text{RNN}}^{(m)} := \Big\{ &\hat{\boldsymbol{H}} : \hat{H}_t = \hat{y}(t) \text{ solves (5)} \\ &\text{with } c \in \mathbb{R}^m, W \in \mathbb{R}^{m\times m}, U \in \mathbb{R}^{m\times d}\Big\}. \end{aligned} \tag{6}$$

The hypothesis space for one layer RNNs with arbitrary depth is defined by

$$\mathcal{H}_{\text{RNN}} := \bigcup_{m\in\mathbb{N}_+} \mathcal{H}_{\text{RNN}}^{(m)}. \tag{7}$$

In essence, RNNs can be understood as a way to represent functionals by introducing hidden dynamical systems, with the predicted outputs being observations.

Recent approaches based on convolutional architectures present an entirely different class of methods to parameterise the functionals. Concretely, a convolution based temporal sequence model with K layers and M_k channels at layer k is given by

$$\begin{aligned} \boldsymbol{h}_{0,i} &= \boldsymbol{x}_i, \\ \boldsymbol{h}_{k+1,i} &= \sigma\left(\sum_{j=1}^{M_k} \boldsymbol{w}_{kji} \ast_{d_k} \boldsymbol{h}_{k,j}\right), \\ \hat{\boldsymbol{y}} &= \boldsymbol{h}_K. \end{aligned} \tag{8}$$

Here, $\boldsymbol{x}_i$ is the ith dimension of $\boldsymbol{x}$, and $\boldsymbol{w}_{kji}$ is the filter from channel j at layer k to the channel i at layer $k+1$. All the filters have a size $l \geq 2$, i.e. $r(\boldsymbol{w}_{kji}) = l \geq 2$. In the following discussion, we assume that the dilation rate satisfies $d_k = l^k$. That is, the dilation rate increases exponentially for each layer to achieve an exponentially large receptive field. This is the standard practice for dilated convolutional structures (Oord et al., 2016; Yu & Koltun, 2016).

The CNN model (8) also defines a family of functionals

$$\begin{aligned} \mathcal{H}_{\text{CNN}}^{(l,K,\{M_k\})} := \Big\{ &\hat{\boldsymbol{H}} : \hat{H}_t = \hat{y}(t) \text{ according to (8)} \\ &\text{with } \boldsymbol{w}_{kji} \in \mathbb{R}^l\Big\}. \end{aligned} \tag{9}$$

For any $l \geq 2$ and $d_k = l^k$, the hypothesis space for CNNs with arbitrary depth and number of channels is defined as

$$\mathcal{H}_{\text{CNN}}^{(l)} = \bigcup_{K\in\mathbb{N}_+} \bigcup_{\{M_k\}\in\mathbb{N}_+^K} \mathcal{H}_{\text{CNN}}^{(l,K,\{M_k\})}. \tag{10}$$

4. Approximation theory for convolutional structures

In this section, we study the approximation properties of $\mathcal{H}_{\text{CNN}}^{(l)}$. Our main results consist of two parts. First, we prove that $\mathcal{H}_{\text{CNN}}^{(l)}$ is dense in appropriate concept spaces. Next, we derive explicit upper and lower bounds for the approximation rate, which depends on the depth and width of CNNs, together with appropriate notions of complexity of the target functional family. This is an important result that characterises the kind of targets that can be well-approximated by CNNs using a small number of parameters.

To make analysis amenable, we consider the linear setting. This is non-trivial for two reasons: first, the key feature we would like to study for time series applications is the dependence on time, for which non-linearity still plays an

important role even in the case of linear activations. For example, we will see that the Riesz representation of a linear functional is nonlinear in time. The approximation results for RNNs (Li et al., 2021) has already demonstrated this. Second, since the approximation results for linear RNNs have been derived in Li et al. (2021), by adopting the same setting we can study explicitly the differences between RNNs and CNNs. Under the assumption that the activation σ is linear, the RNN and CNN hypothesis spaces can be further simplified as

$$\mathcal{H}_{\text{RNN}}^{(m)} := \Big\{ \hat{\boldsymbol{H}} : \hat{H}_t(\boldsymbol{x}) = \sum_{s\in\mathbb{N}} c^\top W^{s-1} U x(t-s), \quad c \in \mathbb{R}^m, W \in \mathbb{R}^{m\times m}, U \in \mathbb{R}^{m\times d} \Big\}, \tag{11}$$

$$\begin{aligned}\mathcal{H}_{\text{CNN}}^{(l,K,\{M_k\})} := \Big\{ \hat{\boldsymbol{H}} : \hat{H}_t(\boldsymbol{x}) &= \sum_{s\in\mathbb{N}} \rho^{(\hat{\boldsymbol{H}})}(s)^\top x(t-s), \\ \boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}(s) &= \sum_{i_{K-1}=1}^{M_{K-1}} \boldsymbol{w}_{K-1,i_{K-1},t} \ast_{l^{K-1}} \\ &\sum_{i_{K-2}=1}^{M_{K-2}} \boldsymbol{w}_{K-2,i_{K-2},i_{K-1}} \ast_{l^{K-2}} \\ &\cdots \ast_{l^2} \sum_{i_1=1}^{M_1} \boldsymbol{w}_{1i_1i_2} \ast_{l^1} \boldsymbol{w}_{0i_1}(s) \Big\}.\end{aligned} \tag{12}$$

We will also restrict to a target functional family (concept space) that is consistent with the linear activation setting. The following definition is introduced in Li et al. (2021).

Definition 2. *Let $\boldsymbol{H} = \{H_t : t \in \mathbb{Z}\}$ be a sequence of functionals which satisfies*

1. *$\boldsymbol{H}$ is causal if it does not depend on the future inputs: for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}$ and any $t \in \mathbb{Z}$ such that $x_1(s) = x_2(s)$ for all $s \le t$, the output satisfies $H_t(\boldsymbol{x}_1) = H_t(\boldsymbol{x}_2)$;*

2. *$H_t \in \boldsymbol{H}$ is a continuous linear functional if for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}$ and $\lambda_1, \lambda_2 \in \mathbb{R}$,*

$$\begin{aligned} H_t(\lambda_1 \boldsymbol{x}_1 + \lambda_2 \boldsymbol{x}_2) &= \lambda_1 H_t(\boldsymbol{x}_1) + \lambda_2 H_t(\boldsymbol{x}_2), \\ \|H_t\| &:= \sup_{\boldsymbol{x}\in\mathcal{X}, \|\boldsymbol{x}\|_{\mathcal{X}} \le 1} |H_t(\boldsymbol{x})| < \infty, \end{aligned} \tag{13}$$

   where $\|H_t\|$ denotes the induced functional norm. Furthermore, the norm of a sequence of functionals is defined by $\|\boldsymbol{H}\| := \sup_{t\in\mathbb{Z}} \|H_t\|$.

3. *A sequence of functionals $\boldsymbol{H}$ is time-homogeneous if for any $t, \tau \in \mathbb{Z}$, $H_t(\boldsymbol{x}) = H_{t+\tau}(\boldsymbol{x}^{(\tau)})$ where $x^{(\tau)}(s) := x(s-\tau)$ for all $s \in \mathbb{Z}$.*

Following the Definition 2, we consider the concept space

$$\mathcal{C} = \{ \boldsymbol{H} : H_t \in \boldsymbol{H} \text{ is linear continuous, } \boldsymbol{H} \text{ is causal and time-homogeneous} \}. \tag{14}$$

Note that $\mathcal{H}_{\text{RNN}} \subset \mathcal{C}$ and $\mathcal{H}_{\text{CNN}}^{(l)} \subset \mathcal{C}$, which follows from the fact that any sequences in these two spaces have convolutional (thus linear) representations. For a given $\hat{\boldsymbol{H}}$, we denote its corresponding representation by $\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}$. We see that RNNs and CNNs are different in the sense that for RNNs, the representation $\rho^{(\hat{\boldsymbol{H}})}(s) = c^\top W^{s-1} U$, which has an infinite support in time. However, for CNNs, the representation $\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}$ is a finitely supported sequence with radius $r(\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}) = l^K - 1$. This can be understood as the maximal memory length of the CNN model, which is usually called the receptive field.

Next we present a key lemma regarding the hypothesis space, which shows that any target in the concept space has a convolutional representation.

Lemma 1. *For any $\boldsymbol{H} \in \mathcal{C}$, there exists a unique ℓ^2 sequence $\boldsymbol{\rho}^{(\boldsymbol{H})} : \mathbb{N} \to \mathbb{R}^d$ such that*

$$H_t(\boldsymbol{x}) = \sum_{s=0}^{\infty} \rho^{(\boldsymbol{H})}(s)^\top x(t-s), \quad t \in \mathbb{Z}. \tag{15}$$

For a target $\boldsymbol{H}$, the corresponding representation is denoted as $\boldsymbol{\rho}^{(\boldsymbol{H})}$. The approximation of $\boldsymbol{H}$ by RNNs or CNNs is equivalent to the approximation of $\boldsymbol{\rho}^{(\boldsymbol{H})}$ using the respective $\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}$.

Remark 1. *The hypothesis space is related with linear time-invariant system (LTI system). In fact, a causal LTI system will induce a linear functional which satisfies Definition 2. However, we note that not every linear functional satisfying Definition 2 corresponds to an LTI system. Our aim is to investigate general funtionals, and does not assume the data is generated from a hidden linear system.*

4.1. Summary of approximation results for RNNs

In order to facilitate the presentation of CNN results and subsequent comparisons, we first review the main approximation results for RNNs proved in Li et al. (2021) [1]. It was shown that under fairly general conditions, $\mathcal{H}_{\text{RNN}}$ is dense in $\mathcal{C}$. With further assumptions that $\boldsymbol{\rho}^{(\boldsymbol{H})}$ decays at least exponentially in time, an explicit bound of the approximation rate can be obtained. That is, for any $\boldsymbol{H} \in \mathcal{C}$, there exists $\hat{\boldsymbol{H}} \in \mathcal{H}_{\text{RNN}}^{(m)}$ such that

$$\|\boldsymbol{H} - \hat{\boldsymbol{H}}\| \equiv \sup_{t\in\mathbb{R}} \|H_t - \hat{H}_t\| \le \frac{C\gamma d}{\beta m^\alpha}, \tag{16}$$

[1] Note that the continuous time setting was considered there, but the main conclusions remain the same under discretization.

where $C > 0$ is a universal constant, β denotes the (exponential) decay rate of $\boldsymbol{\rho}^{(\boldsymbol{H})}$ and γ measures the smoothness of $\boldsymbol{H}$. This result shows that a target functional family generating the temporal relationship can be well approximated by RNN models if it is smooth and decays sufficiently fast. For a target with long memory, say a power law decay, the number of parameters sufficient to approximate it using RNNs may increase exponentially. This is the so-called "curse of memory" in approximation of RNNs (Li et al., 2021).

4.2. Approximation results for CNNs

In this section, we present the main approximation results of this paper, which can be viewed as parallel results to those discussed in section 4.1, but for CNNs. This allows us to understand precisely the similarity and differences of CNNs and RNNs with respect to their approximation capabilities, when applied to time series modelling.

First, we prove a density result. Recall that $\mathcal{H}^{(l)}_{\text{CNN}} \subset \mathcal{C}$. The density result here shows that $\mathcal{H}^{(l)}_{\text{CNN}}$ is in fact dense in $\mathcal{C}$. It is also understood as a universal approximation property of linear CNNs applied to linear functionals.

Theorem 2. (UAP for CNNs) *Let $\boldsymbol{H} \in \mathcal{C}$. Then for any $\epsilon > 0$, there exist $\hat{\boldsymbol{H}} \in \mathcal{H}^{(l)}_{CNN}$ such that*

$$\|\boldsymbol{H} - \hat{\boldsymbol{H}}\| \equiv \sup_{t\in\mathbb{Z}} \|H_t - \widehat{H}_t\| < \epsilon. \tag{17}$$

Now we discuss some basic understanding of Theorem 2. For any $\boldsymbol{x}$ such that $\|\boldsymbol{x}\|_{\mathcal{X}} \leq 1$, we have

$$\begin{aligned} |H_t(\boldsymbol{x}) - \hat{H}_t(\boldsymbol{x})|^2 &\leq \sum_{s=0}^{\infty} \left|\boldsymbol{\rho}^{(\boldsymbol{H})}(s) - \boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}(s)\right|^2 \qquad (18)\\ &= \sum_{s=0}^{l^K-1} \left|\boldsymbol{\rho}^{(\boldsymbol{H})}(s) - \boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}(s)\right|^2 + \sum_{s=l^K}^{\infty} \left|\boldsymbol{\rho}^{(\boldsymbol{H})}(s)\right|^2, \end{aligned}$$

where we use the fact that $r(\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}) = l^K - 1$. Consider the approximation error on two intervals. On $[l^K, \infty]$, $\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}$ is always zero, thus the error only depends on the tail sum of $\boldsymbol{\rho}^{(\boldsymbol{H})}$. That is, the long term memory of the target. Since $\boldsymbol{\rho}^{(\boldsymbol{H})} \in \ell^2$, this error converges to zero when we choose a deep model (with K appropriately large). On $[0, l^K - 1]$, we can show that there exits a $\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}$ with a sufficient number of channels such that $\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})} = \boldsymbol{\rho}^{(\boldsymbol{H})}$ in this range.

The error on $[l^K, \infty]$ is easy to analyse as it only involves the decay of $\boldsymbol{\rho}^{(\boldsymbol{H})}$, which enters in the approximation of both RNNs and CNNs. The more interesting phenomenon occurs on the interval $[0, l^K - 1]$. A natural question is, given a target $\boldsymbol{H}$, how does the approximation error on this interval depend on the number of channels and depth of CNNs? This question is important since it reveals the structures in the target functional family which facilitate efficient approximation using CNNs besides the decay of memory. In other words, it characterises theoretically the type of temporal modelling tasks for which CNNs are naturally suited.

Interestingly, it turns out that the approximation rate depends on the spectrum of $\boldsymbol{\rho}^{(\boldsymbol{H})}$ under a suitable tensorisation procedure. We motivate this idea by simple examples where $d = 1$ and $l = K = 2$.

Example 1. *Recall that a target $\boldsymbol{H}$ has the representation $\boldsymbol{\rho}^{(\boldsymbol{H})}$. Since $r(\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}) = 4$ here, the CNN performs approximation only on $[0, 3]$, hence we restrict the target as $\boldsymbol{\rho}^{(\boldsymbol{H})}_{[0,3]} \in \mathbb{R}^4$. Denote the rearrangement operator by T, and*

$$T\left(\boldsymbol{\rho}^{(\boldsymbol{H})}_{[0,3]}\right) = \begin{pmatrix} \rho_0^{(\boldsymbol{H})} & \rho_2^{(\boldsymbol{H})} \\ \rho_1^{(\boldsymbol{H})} & \rho_3^{(\boldsymbol{H})} \end{pmatrix}. \tag{19}$$

We study the singular value decomposition (SVD) of the matrix $T\left(\boldsymbol{\rho}^{(\boldsymbol{H})}_{[0,3]}\right)$. If $T\left(\boldsymbol{\rho}^{(\boldsymbol{H})}_{[0,3]}\right)$ has only one non-zero singular value, then $\operatorname{rank} T\left(\boldsymbol{\rho}^{(\boldsymbol{H})}_{[0,3]}\right) = 1$. It is straightforward to deduce that a CNN with 1 channel in both layers is sufficient to represent it. In fact, let $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ be the filter on the first and second layer respectively. The resulting representation is

$$\begin{aligned} \boldsymbol{\rho}^{(\hat{\boldsymbol{H}})} = \boldsymbol{w}_2 *_2 \boldsymbol{w}_1 &= (w_{11}, w_{12}) *_2 (w_{21}, w_{22}) \qquad (20)\\ &= (w_{11}w_{21}, w_{12}w_{21}, w_{11}w_{22}, w_{12}w_{22}), \\ T(\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}) &= \begin{pmatrix} w_{11}w_{21} & w_{11}w_{22} \\ w_{12}w_{21} & w_{12}w_{22} \end{pmatrix} \qquad (21)\\ &= \begin{pmatrix} w_{11} \\ w_{12} \end{pmatrix} \begin{pmatrix} w_{21} & w_{22} \end{pmatrix}. \qquad (22) \end{aligned}$$

This can represent any 2 by 2 rank 1 matrix. Thus, any $\boldsymbol{\rho}^{(\boldsymbol{H})}$ such that $T\left(\boldsymbol{\rho}^{(\boldsymbol{H})}_{[0,3]}\right)$ with rank no more than 1 can be represented.

In order to represent a rank 2 target, we need to increase the number of channels. Let the first layer have 2 channels. Then the resulting representation is

$$\begin{aligned} \boldsymbol{\rho}^{(\hat{\boldsymbol{H}})} &= \boldsymbol{w}_2 *_2 \boldsymbol{w}_1 + \boldsymbol{v}_2 *_2 \boldsymbol{v}_1, \\ T(\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}) &= \begin{pmatrix} w_{11}w_{21} & w_{11}w_{22} \\ w_{12}w_{21} & w_{12}w_{22} \end{pmatrix} + \begin{pmatrix} v_{11}v_{21} & v_{11}v_{22} \\ v_{12}v_{21} & v_{12}v_{22} \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{\sqrt{\sigma_1}}w_{11} & \frac{1}{\sqrt{\sigma_2}}v_{11} \\ \frac{1}{\sqrt{\sigma_1}}w_{12} & \frac{1}{\sqrt{\sigma_2}}v_{12} \end{pmatrix} \begin{pmatrix} \sigma_1 & \\ & \sigma_2 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{\sigma_1}}w_{21} & \frac{1}{\sqrt{\sigma_2}}v_{21} \\ \frac{1}{\sqrt{\sigma_1}}w_{22} & \frac{1}{\sqrt{\sigma_2}}v_{22} \end{pmatrix}^{\top}. \end{aligned} \tag{23}$$

That is, $T(\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})})$ can represent the SVD for any 2 by 2 matrix with appropriate choices of $\boldsymbol{w}_1, \boldsymbol{w}_2$ and $\boldsymbol{v}_1$, which implies that $T\left(\boldsymbol{\rho}^{(\boldsymbol{H})}_{[0,3]}\right)$ with rank no more than 2 can be represented.

The above two examples show that the number of channels needed for an exact representation ($\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}_{[0,3]} = \boldsymbol{\rho}^{(\boldsymbol{H})}_{[0,3]}$)

depends on the rank of targets, i.e. the number of non-zero singular values of $T(\boldsymbol{\rho}^{(\boldsymbol{H})}_{[0,3]})$. Furthermore, when there is no such exact representations, we need to decide the approximation error. Suppose the target is with rank 2, but we use a CNN with only 1 channel to approximate it. By the Eckart–Young–Mirsky theorem, the best approximation error is equal to the smallest singular value of $T(\boldsymbol{\rho}^{(\boldsymbol{H})}_{[0,3]})$. This inspires us to relate the approximation error with the decay of singular values.

The above examples illustrate the basic approach on how to determine the number of channels needed to achieve a given approximation accuracy. Although the main tool used in above examples, SVD, is only valid for $K = 2$ where we can reshape the vector representation of $\boldsymbol{\rho}^{(\boldsymbol{H})}$ into a matrix, in general we can extend the above techniques to any K using higher order singular value decomposition (HOSVD) (De Lathauwer et al., 2000). Now we introduce some basic facts about HOSVD.

HOSVD basics. For a discrete sequence $\boldsymbol{\rho}$, we denote the tensorisation of $\boldsymbol{\rho}_{[0,l^K-1]}$ by $T_{l^K}(\boldsymbol{\rho})$. This is an order K tensor, with all the dimensions equal to l i.e. $T_{l^K}(\boldsymbol{\rho}) \in \mathbb{R}^{l\times l\times\cdots\times l}$. The tensorisation follows column major ordering. The singular values of $T_{l^K}(\boldsymbol{\rho})$ is defined by applying the usual SVD for matrices to the K matrices with size $\mathbb{R}^{l\times l^{K-1}}$ obtained by mode-k flattening of the tensor. This gives rise to at most lK singular values. Define the rank of the tensor by its number of non-zero singular values. We defer the precise but involved definition of HOSVD to the appendix.

Similar to the SVD low rank approximation for matrices, we have the following error estimation for tensors (De Lathauwer et al., 2000).

Lemma 3. *Denote the singular values of $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{l\times l\times\cdots\times l}$ by $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{lK} \geq 0$, we have*

$$\inf_{\hat{\boldsymbol{H}}} \left\| \boldsymbol{\mathcal{A}} - T_{l^K}(\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}) \right\| \leq \left(\sum_{i=K'+1}^{\operatorname{rank}\boldsymbol{\mathcal{A}}} \sigma_i^2 \right)^{\frac{1}{2}}, \tag{24}$$

where the infimum is taken over all $\hat{\boldsymbol{H}} \in \mathcal{H}^{(l,K,\{M_k\})}_{CNN}$ such that $K' < \operatorname{rank}\boldsymbol{\mathcal{A}}$, where $K' = \operatorname{rank} T_{l^K}(\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})})$.

Based on Lemma 3, one can measure the complexity of a target $\boldsymbol{H}$ by the decay rate of singular values of $T_{l^K}(\boldsymbol{\rho}^{(\boldsymbol{H})})$.

Complexity measure and approximation rates. Now we can get down to define appropriate complexity measures and prove approximation rates of CNNs. Based on previous discussions on HOSVD, we have the following key definition.

Definition 3. *Consider a sequence of functionals $\boldsymbol{H}$ with associated representation $\boldsymbol{\rho}^{(\boldsymbol{H})}$. For any $l, K \in \mathbb{N}_+$, let $\sigma_1^{(K)} \geq \sigma_2^{(K)} \geq \cdots \geq \sigma_{lK}^{(K)} \geq 0$ be the singular values of $T_{l^K}(\boldsymbol{\rho}^{(\boldsymbol{H})})$. Let $g \in \boldsymbol{c}_0(\mathbb{N}, \mathbb{R}_+)$ be a non-increasing function with zero limit at infinity. Define the complexity measure of $\boldsymbol{H}$ by*

$$\mathrm{C}^{(l,g)}(\boldsymbol{H}) = \inf\left\{ c : \left(\sum_{i=s+K}^{lK} |\sigma_i^{(K)}|^2 \right)^{\frac{1}{2}} \leq cg(s), \quad s \geq 0, K \geq 1 \right\}. \tag{25}$$

Based on this complexity measure, we define a concept space with certain spectral regularity (measured by the decay of singular values)

$$\mathcal{C}^{(l,g)} := \left\{ \boldsymbol{H} \in \mathcal{C} : \mathrm{C}^{(l,g)}(\boldsymbol{H}) < \infty \right\}. \tag{26}$$

Next, we discuss some facts and examples to enhance the understanding of the space $\mathcal{C}^{(l,g)}$.

Remark 2. *Suppose the function g is monotonously decreasing and strictly positive. Then for any $\boldsymbol{H} \in \mathcal{C}$ such that $\boldsymbol{\rho}^{(\boldsymbol{H})}$ is finitely supported, we have $\boldsymbol{H} \in \mathcal{C}^{(l,g)}$.*

Remark 3. *Suppose $\boldsymbol{H} \in \mathcal{C}$ with $\boldsymbol{\rho}^{(\boldsymbol{H})}$ finitely supported. Then there exists a finitely supported decreasing g such that $\boldsymbol{H} \in \mathcal{C}^{(l,g)}$.*

The details can be found in appendix.

The reason why the summation in (25) starts at $s+K$ is that, when K increases, there will be additional singular values resulting from the additional matrix SVDs. These singular values are all equal to the 2-norm of $\boldsymbol{\rho}^{(\boldsymbol{H})}$. We show this by the following example.

Example 2. *Suppose the filter size $l = 2$ and $\boldsymbol{\rho}^{(\boldsymbol{H})} = (1, 0, 0, 1, 0, 0, 0, \cdots)$. The following table shows the corresponding singular values of $T_{l^K}(\boldsymbol{\rho}^{(\boldsymbol{H})})$ for different K:*

K	*Singular Values of* $T_{l^K}(\boldsymbol{\rho}^{(\boldsymbol{H})})$
1	$(1, 1)$
2	$(1, 1, 1, 1)$
3	$(\sqrt{2}, 1, 1, 1, 1, 0)$
4	$(\sqrt{2}, \sqrt{2}, 1, 1, 1, 1, 0, 0)$

When $K \geq 2$, $[0, l^K - 1]$ covers the support of $\boldsymbol{\rho}^{(\boldsymbol{H})}$. If we further increase K, there will be additional singular values all equal to $\|\boldsymbol{\rho}^{(\boldsymbol{H})}\|_2 = \sqrt{2}$.

For $K \geq 2$, we have

$$\sum_{i=s+K}^{lK} |\sigma_i^{(K)}|^2 = \begin{cases} 2, & s = 1 \\ 1, & s = 2. \\ 0, & \textit{otherwise} \end{cases} \tag{27}$$

Then given g one can compute $\mathrm{C}^{(l,g)}(\boldsymbol{H})$ based on (25) and (27).

Based on Definition 3, we can now present our main result on the approximation rate of CNNs.

Theorem 4. (Approximation rate for CNNs) *Fix $l \geq 2$ and $g \in \boldsymbol{c}_0(\mathbb{N}, \mathbb{R}_+)$. For any $\boldsymbol{H} \in \mathcal{C}^{(l,g)}$ and any set of parameters $(K, \{M_k\})$, we have*

$$\frac{1}{\sqrt{d}} \sup_{t \in [l^K, \infty]} |\boldsymbol{\rho}^{(\boldsymbol{H})}(t)| \leq \inf_{\hat{\boldsymbol{H}} \in \mathcal{H}_{CNN}^{(l,K,\{M_k\})}} \|\boldsymbol{H} - \hat{\boldsymbol{H}}\| \leq d\, g(KM^{\frac{1}{K}} - K)\, \mathrm{C}^{(l,g)}(\boldsymbol{H}) + \|\boldsymbol{\rho}^{(\boldsymbol{H})}_{[l^K,\infty]}\|_2, \tag{28}$$

where $M := \frac{1}{d}(\sum_{k=2}^{K} M_k M_{k-1} - lK)$ denotes the effective number of filters.

As discussed earlier, the term $\|\boldsymbol{\rho}^{(\boldsymbol{H})}_{[l^K,\infty]}\|_2$ is the error on $[l^K, \infty]$, due to the limited support of depth-K CNNs. This error can be reduced by increasing the number of layers K, and it is less important as l^K increases exponentially fast.

Hence, the subsequent discussion will focus on the more interesting term, i.e. the approximation error on the interval $[0, l^K - 1]$: $d\, g(KM^{\frac{1}{K}} - K)\, \mathrm{C}^{(l,g)}(\boldsymbol{H})$. This is related to the complexity measure of $\boldsymbol{H}$, which is small if there exists effective low rank structures in the target. With the fact that $\operatorname{rank} T_{l^K}(\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})})$ is at least $KM^{\frac{1}{K}}$ (see details in appendix), combining Lemma 3 and (25) gives the result. Given a target $\boldsymbol{H}$, this error can be reduced by either increasing the number of filters M or the number of layers K.

Whether a target $\boldsymbol{H}$ can be easily approximated depends on $\mathrm{C}^{(l,g)}(\boldsymbol{H})$, which is determined by the decay of singular values of $T_{l^K}(\boldsymbol{\rho}^{(\boldsymbol{H})})$. For $T_{l^K}(\boldsymbol{\rho}^{(\boldsymbol{H})})$ with fast decaying singular values, even if the rank is large or $\boldsymbol{\rho}^{(\boldsymbol{H})}$ decays slowly, one can still have a good approximation with CNNs. This is very different from previous results of RNNs.

We now introduce an example to illustrate this observation. Consider 3 targets with the following representations

$$\rho_1(t) = \begin{cases} \frac{\pi^2}{12}, & t = 17, 18, 25, 26 \\ 0, & \text{otherwise} \end{cases},$$

$$\rho_2(t) = \begin{cases} \frac{\pi^2}{12}, & t = 9, 15, 19, 26 \\ 0, & \text{otherwise} \end{cases}, \quad \rho_3(t) = \frac{1}{t}.$$

Note that all of them are instances of the adding problem. Moreover, they all have the same magnitude, $\|\boldsymbol{\rho}_1\|_2 = \|\boldsymbol{\rho}_2\|_2 = \|\boldsymbol{\rho}_3\|_2$. The support of both $\boldsymbol{\rho}_1, \boldsymbol{\rho}_2$ are finite, while $\boldsymbol{\rho}_3$ have an infinite support. Furthermore, $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$ have the same number of non-zero entries. We plot the error bound of each targets in Figure 1.

First, from the figure we conclude that the approximation can be reduced by increasing K or M. For a fixed K, when M is sufficiently large such that the rank of the model is no less than the rank of the target, the error will no longer decrease if we further increase M. At this point the error only comes from the tail which can only be reduced by increasing K (hence increasing the support, or receptive field of the CNN model).

Next, we compare $\boldsymbol{\rho_1}$ and $\boldsymbol{\rho_2}$. From Figure 1, we conclude that $\boldsymbol{\rho_1}$ is easier to approximate than $\boldsymbol{\rho_2}$. The only difference between these two targets is the position of their non-zero entries, which results in $T_{2^5}(\boldsymbol{\rho_1})$ being rank 5 while $T_{2^5}(\boldsymbol{\rho_2})$ being rank 10. This is the case where low rank structures facilitates the approximation by CNNs.

Finally, we make a comparison between $\boldsymbol{\rho_2}$ and $\boldsymbol{\rho_3}$. We conclude that $\boldsymbol{\rho_3}$ is easier to approximate as the overall error is smaller than $\boldsymbol{\rho_2}$, in spite of the fact that $\boldsymbol{\rho_2}$ is finitely supported and has a simple form. The reason is that the tail sum of singular values of $T_{l^K}(\boldsymbol{\rho_3})$ decays faster than that of $T_{l^K}(\boldsymbol{\rho}_2)$. This illustrates the case where a target with fast decaying singular values (i.e. low effective rank) can be easily approximated.

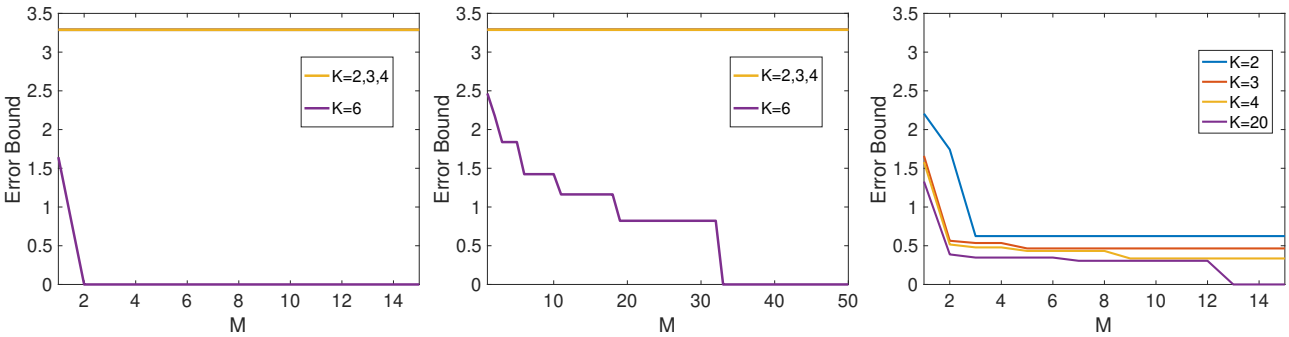
5. Comparison of CNNs and RNNs in terms of approximation

As motivated earlier, how to judiciously choose architectures for time series modelling is an important practical problem, and prior works only address it empirically. Combining the approximation theory for CNNs developed here and those for RNNs in Li et al. (2021), we are in a position to provide theoretical answers to such problems, albeit in specific settings. This is the purpose of the present section.

In short, our theoretical analyses show that neither CNNs nor RNNs are always better than the other. Whether one out-performs its counterpart depends on the properties of the target relationship. Our results make this statement precise, and provide general guidelines on how to select architectures based on applications. We begin by presenting two representative examples for the two aforementioned cases, from which we can infer key differences between RNNs and CNNs for time series modelling.

Example where RNNs out-perform CNNs. We assume a scalar input with $d = 1$. Consider a target $\boldsymbol{H} \in \mathcal{C}$ with the representation $\rho^{\boldsymbol{H}}(t) = \gamma^t$, where $0 < \gamma < 1$. It is easy for RNNs to approximate this target, since the representation has a power form. In fact, we have $\boldsymbol{H} \in \mathcal{H}^{(1)}_{\text{RNN}}$, i.e. a RNN with one hidden unit is sufficient to achieve an exact representation, with 0 approximation error for any γ.

For a CNN $\hat{\boldsymbol{H}} \in \mathcal{H}^{(l)}_{\text{CNN}}$, based on the lower bound of the approximation error from Theorem 4, we have that $\|\boldsymbol{H} - \hat{\boldsymbol{H}}\|^2 \geq \sup_{t \in [l^K, \infty]} \|\boldsymbol{\rho}^{(\boldsymbol{H})}(t)\|_2$. Thus, in order to achieve an approximation error with $\|\boldsymbol{H} - \hat{\boldsymbol{H}}\|^2 < \epsilon$, we have $\sup_{t \in [l^K, \infty]} \|\boldsymbol{\rho}^{(\boldsymbol{H})}(t)\|_2 = \gamma^{l^K} < \epsilon$. This implies $l^K \geq$



(a) $\boldsymbol{\rho}_1$ (b) $\boldsymbol{\rho}_2$ (c) $\boldsymbol{\rho}_3$

Figure 1. Let $l = 2$. We plot the upper bound of the approximation errors against the effective number of filters. Each line corresponds to a fixed K. The error bound is calculated based on Theorem 4, where the tail term is calculated directly and the rank term is determined by numerically compute the singular values of $T_{lK}(\boldsymbol{\rho}_i)$. These curves describe, according to Theorem 4, the decay of approximation errors of CNNs as one increases the number of filters. Observe that the decay depends on the low rank structures of the targets.

$\frac{\log(\epsilon)}{\log(\gamma)}$. That is to say, the necessary number of layers to achieve an approximation error smaller than ϵ diverges to infinity as γ approaches 1.

Example where CNNs out-perform RNNs. Still assume a scalar input with $d = 1$. Consider a target $\boldsymbol{H} \in \mathcal{C}$ with representation $\rho^{(\boldsymbol{H})}(t) = \delta(t - 2^K)$ where $K \in \mathbb{N}_+$. This is an discrete time impulse at $t = 2^K$ and can be regarded as the copying memory problem (Bai et al., 2018): given an input $\boldsymbol{x}$, the output $\boldsymbol{y}$ satisfies $y(t) = x(t - 2^K)$. We have $\boldsymbol{H} \in \mathcal{H}_{\text{CNN}}^{(2,K,\{1\})}$. Thus, a K-layer CNN with one channel per layer is sufficient to achieve an exact representation.

Recall that RNN approximates the target $\boldsymbol{\rho}^{(\boldsymbol{H})}$ with a power sum $\boldsymbol{\rho}^{(\hat{\boldsymbol{H}})}(s) = c^\top W^{s-1} U$. Suppose here $W \in \mathbb{R}^{m\times m}$ is a diagonalisable matrix with negative eigenvalues. The latter has a special structure, which makes it difficult to approximate certain functions. Suppose u is a m-term power sum, i.e. $u(t) = c_0 + \sum_{i=1}^m c_i\, \gamma_i^t$. Then according to (Borwein & Erdélyi, 1996), we have

$$|u(t+1) - u(t)| \leq \frac{2m}{t} \sup_{s\geq 0} u(s). \tag{29}$$

For a fixed number of terms m, the changes between $u(t+1)$ and $u(t)$ approaches zero as t approaches infinity. This means that if there is a sudden change in u far from origin, the number of terms n must be large.

For a RNN $\hat{\boldsymbol{H}} \in \mathcal{H}_{\text{RNN}}^{(m)}$, assume $\|\boldsymbol{H} - \hat{\boldsymbol{H}}\| < \epsilon$. Let $u(s) = c^\top W^{s-1} U$ denote the power sum corresponding to $\hat{\boldsymbol{H}}$. Since W is a $m \times m$ matrix, $u(s)$ is a m^2-term power sum. We have that $|u(2^K + 1) - u(2^K)| > 1 - 2\epsilon$, and hence (29) implies $m^2 > 2^{K-1}\frac{1-2\epsilon}{1+\epsilon}$. As k increases, the number of parameters needed for RNNs to achieve an error less than ϵ increases exponentially, while this increment is linear for CNNs.

For these examples, the targets actually come from the other respective hypothesis spaces. This means that the functionals these two hypothesis spaces can represent are quite different. The difference between RNNs and CNNs comes from their underlying structure: the RNN uses a power sum to approximate the target, while the CNN uses a finitely supported function. Therefore, RNNs are good at approximating targets with exponentially decaying structures, but it is not efficient to handle targets with sudden changes. On the other hand, CNN works well for targets with low ranks or fast decaying singular values. However, it can become inefficient if the tail error term is significant, and at the same time, the truncated term does not possess low rank structures.

In practice, the property of decay and sparsity can be checked by querying the target relationship with proper inputs. However, for a more general situation when there is only access to sampled data for input/output, inferring these underlying properties of targets remains challenging.

6. Conclusion

In this paper, we studied the approximation properties of convolutional architectures when applied to time series modelling. We considered the simple but representative linear setting. The approximation error is characterised by both the memory and the spectrum of the target relationship based on a tensorisation argument. We concluded that a target with a low rank property or fast decaying spectrum can be efficiently approximated by a CNN model.

In particular, these results for CNNs, together with previous results for RNNs, provide basic theoretical insights into the problem of architectural choice for time series modelling: the RNN exploits exponential decaying structures, whereas

the CNN exploits low-rank structures. This forms the first basic step towards a concrete understanding of the interplay between different neural network architectures and the structures of temporal relationships under investigation.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. HJ is supported by Nationa University of Singapore under PGF scholarship. QL is supported by the National Research Foundation, Singapore, under the NRF fellowship (NRF-NRFF13-2021-0005).

References

Bai, S., Kolter, J. Z., and Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. 2018. URL http://arxiv.org/abs/1803.01271.

Banerjee, I., Ling, Y., Chen, M. C., Hasan, S. A., Langlotz, C. P., Moradzadeh, N., Chapman, B., Amrhein, T., Mong, D., Rubin, D. L., Farri, O., and Lungren, M. P. Comparative effectiveness of convolutional neural network (cnn) and recurrent neural network (rnn) architectures for radiology text report classification. Artificial Intelligence in Medicine, 97:79 – 88, 2019. ISSN 0933-3657. doi: https://doi.org/10.1016/j.artmed.2018.11.004. URL http://www.sciencedirect.com/science/article/pii/S0933365717306255.

Bao, C., Li, Q., Shen, Z., Tai, C., Wu, L., and Xiang, X. Approximation analysis of convolutional neural networks a**. 2019.

Boilard, J., Gournay, P., and Lefebvre, R. A literature review of wavenet: theory, application, and optimization. Journal of the Audio Engineering Society, march 2019.

Borwein, P. and Erdélyi, T. A sharp Bernstein-type inequality for exponential sums. Journal fur die Reine und Angewandte Mathematik, 476(476):127–141, 1996. ISSN 00754102. doi: 10.1515/crll.1996.476.127.

Bramwell, M. C. and Kreyszig, E. Introductory Functional Analysis with Applications. The Mathematical Gazette, 1979. ISSN 00255572. doi: 10.2307/3616033.

Chow, T. W. S. and Xiao-Dong Li. Modeling of continuous time dynamical systems with input by recurrent neural networks. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 47(4):575–578, 2000.

De Lathauwer, L., De Moor, B., and Vandewalle, J. A multilinear singular value decomposition. SIAM Journal on Matrix Analysis and Applications, 21(4): 1253–1278, 2000. ISSN 08954798. doi: 10.1137/S0895479896305696.

Doya, K. Universality of fully connected recurrent neural networks. Dept. of Biology, UCSD, Tech. Rep, 1993.

Funahashi, K. and Nakamura, Y. Approximation of dynamical systems by continuous time recurrent neural networks. Neural Networks, 6(6):801 – 806, 1993. ISSN 0893-6080.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Jin, Z., Finkelstein, A., Mysore, G. J., and Lu, J. Fftnet: A real-time speaker-dependent neural vocoder. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2251–2255, 2018. doi: 10.1109/ICASSP.2018.8462431.

Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., Stimberg, F., van den Oord, A., Dieleman, S., and Kavukcuoglu, K. Efficient neural audio synthesis. In Dy, J. and Krause, A. (eds.), Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pp. 2410–2419, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/kalchbrenner18a.html.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications, 2009. ISSN 00361445.

Li, X.-D., Ho, J. K. L., and Chow, T. W. S. Approximation of dynamical time-variant systems by continuous-time recurrent neural networks. IEEE Transactions on Circuits and Systems II Analog and Digital Signal Processing, 52: 656–660, 10 2005.

Li, Z., Han, J., E, W., and Li, Q. On the curse of memory in recurrent neural networks: Approximation and optimization analysis. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=8Sqhl-nF50.

Maass, W., Joshi, P., and Sontag, E. D. Computational aspects of feedback in neural circuits. PLOS Computational Biology, 3(1):e165, 2007.

Matthews, M. B. Approximating nonlinear fading-memory operators using neural network models. Circuits, Systems and Signal Processing, 12:279–307, 1993.

Nakamura, Y. and Nakagawa, M. Approximation capability of continuous time recurrent neural networks for

non-autonomous dynamical systems. In Artificial Neural Networks – ICANN 2009, pp. 593–602, 2009.

Oono, K. and Suzuki, T. Approximation and non-parametric estimation of ResNet-type convolutional neural networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pp. 4922–4931. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/oono19a.html.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 2016.

Paine, T. L., Khorrami, P., Chang, S., Zhang, Y., Ramachandran, P., Hasegawa-Johnson, M. A., and Huang, T. S. Fast wavenet generation algorithm. CoRR, abs/1611.09482, 2016. URL http://arxiv.org/abs/1611.09482.

Prenger, R., Valle, R., and Catanzaro, B. Waveglow: A flow-based generative network for speech synthesis. In ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3617–3621, 2019. doi: 10.1109/ICASSP.2019.8683143.

Schäfer, A. M. and Zimmermann, H. G. Recurrent neural networks are universal approximators. In Artificial Neural Networks – ICANN 2006, pp. 632–640, 2006.

Schäfer, A. M. and Zimmermann, H.-G. Recurrent neural networks are universal approximators. International journal of neural systems, 17(04):253–263, 2007.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. WaveNet: A Generative Model for Raw Audio. pp. 1–15, 2016. URL http://arxiv.org/abs/1609.03499.

van den Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., van den Driessche, G., Lockhart, E., Cobo, L., Stimberg, F., Casagrande, N., Grewe, D., Noury, S., Dieleman, S., Elsen, E., Kalchbrenner, N., Zen, H., Graves, A., King, H., Walters, T., Belov, D., and Hassabis, D. Parallel WaveNet: Fast high-fidelity speech synthesis. In Dy, J. and Krause, A. (eds.), Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pp. 3918–3926, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/oord18a.html.

Yin, W., Kann, K., Yu, M., and Schütze, H. Comparative study of CNN and RNN for natural language processing. CoRR, abs/1702.01923, 2017. URL http://arxiv.org/abs/1702.01923.

Yu, F. and Koltun, V. Multi-scale context aggregation by dilated convolutions, 2016.

Zhou, D.-X. Theory of deep convolutional neural networks: Downsampling. Neural Networks, 124:319 – 327, 2020a. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2020.01.018. URL http://www.sciencedirect.com/science/article/pii/S0893608020300204.

Zhou, D.-X. Universality of deep convolutional neural networks. Applied and Computational Harmonic Analysis, 48(2):787 – 794, 2020b. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2019.06.004. URL http://www.sciencedirect.com/science/article/pii/S1063520318302045.