New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG IOP Institute of Physics

PAPER • OPEN ACCESS

Explainable natural language processing with matrix product states

To cite this article: Jirawat Tangpanitanon *et al* 2022 *New J. Phys.* **24** 053032

View the article online for updates and enhancements.

You may also like

- Predicting Coronal Mass Ejections Using *SDO*/HMI Vector Magnetic Data Products and Recurrent Neural Networks
Hao Liu, Chang Liu, Jason T. L. Wang et al.

- 'When' and 'what' did you see? A novel fMRI-based visual decoding framework
Chong Wang, Hongmei Yan, Wei Huang et al.

- Inferring population dynamics in macaque cortex
Ganga Meghanath, Bryan Jimenez and Joseph G Makin

This content was downloaded from IP address 147.27.12.155 on 13/02/2024 at 14:10

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

CrossMark

OPEN ACCESS

RECEIVED
16 December 2021

REVISED
23 March 2022

ACCEPTED FOR PUBLICATION
29 March 2022

PUBLISHED
16 May 2022

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

PAPER

Explainable natural language processing with matrix product states

Jirawat Tangpanitanon[1,2,8], **Chanatip Mangkang**[3,8], **Pradeep Bhadola**[4], **Yuichiro Minato**[5], **Dimitris G Angelakis**[6,7] **and Thiparat Chotibut**[3,*]

[1] Quantum Technology Foundation (Thailand), Bangkok, Thailand
[2] Thailand Center of Excellence in Physics, Ministry of Higher Education, Science, Research and Innovation, Bangkok, Thailand
[3] Chula Intelligent and Complex Systems, Department of Physics, Faculty of Science, Chulalongkorn University, Bangkok, Thailand
[4] Centre for Theoretical Physics & Natural Philosophy, Nakhonsawan Studiorum for Advanced Studies, Mahidol University, Nakhonsawan Campus, Thailand
[5] Blueqat Inc., Japan
[6] School of Electrical and Computer Engineering, Technical University of Crete, Chania, Greece
[7] Centre for Quantum Technologies, National University of Singapore, Singapore
[*] Author to whom any correspondence should be addressed.
[8] Equal contributions.

E-mail: thiparatc@gmail.com

Keywords: matrix product state, entanglement entropy, entanglement spectrum, quantum machine learning, natural language processing, recurrent neural networks

Abstract

Despite empirical successes of recurrent neural networks (RNNs) in natural language processing (NLP), theoretical understanding of RNNs is still limited due to intrinsically complex non-linear computations. We systematically analyze RNNs' behaviors in a ubiquitous NLP task, the sentiment analysis of movie reviews, via the mapping between a class of RNNs called recurrent arithmetic circuits (RACs) and a matrix product state. Using the von-Neumann entanglement entropy (EE) as a proxy for information propagation, we show that single-layer RACs possess a maximum information propagation capacity, reflected by the saturation of the EE. Enlarging the bond dimension beyond the EE saturation threshold does not increase model prediction accuracies, so a *minimal model* that best estimates the data statistics can be inferred. Although the saturated EE is smaller than the maximum EE allowed by the area law, our minimal model still achieves $\sim$99% training accuracies in realistic sentiment analysis data sets. Thus, low EE is not a warrant against the adoption of single-layer RACs for NLP. Contrary to a common belief that long-range information propagation is the main source of RNNs' successes, we show that single-layer RACs harness high expressiveness from the subtle interplay between the information propagation and the word vector embeddings. Our work sheds light on the phenomenology of learning in RACs, and more generally on the explainability of RNNs for NLP, using tools from many-body quantum physics.

1. Introduction

The study of many-body quantum physics prompts the development of theoretical and numerical techniques to compactly represent and analyze quantum states living in an exponentially large Hilbert space. One of the most prominent examples is a compact representation of a ground state of a one-dimensional gapped quantum lattice system with local interaction as a matrix product state (MPS), which can efficiently parametrize an appropriate ground state using resources that grow only linearly with the system size [1]. Compact representations generalizing MPS to higher dimensional systems include projected entangled pair state [2], multiscale entanglement renormalization ansatz, which efficiently parametrize critical high-dimensional systems [3], and, more generally, tensor network (TN) states [4]. These compact state representations drastically reduce the number of parameters from exponential to at most polynomial in the system size, rendering the analysis and the simulation of many-body quantum

© 2022 The Author(s). Published by IOP Publishing Ltd on behalf of the Institute of Physics and Deutsche Physikalische Gesellschaft

systems computationally tractable. Dimensionality reduction has enabled insights into a broad range of many-body quantum phenomena, ranging from quantum phase transitions to topological phases of matter.

On the other hand, machine learning (ML) has also benefitted from algorithms that extract a compact representation of complex data of interests. In supervised ML, many algorithms efficiently convert a gigantic set of data-label pair $\{(\mathbf{X}_i, l_i)\}$, where $\mathbf{X}_i \in \mathbb{R}^d$ is the ith data vector and l_i is the scalar label of that data vector, into a compact representation encoding the relationship between data-label as a conditional probability $P(l|\mathbf{X}, \boldsymbol{\theta})$ parametrized by a set of parameters $\boldsymbol{\theta}$. With the advent of deep learning (DL), a modern paradigm of ML that imitates computational models of biological neural networks, probabilistic relationships between data-label pairs as complex and extensive as picture-name matching, sound-text pairing, or text-speech generation, can be efficiently represented [5]. In fact, the ability of DL to extract a compact representation of complex data has fueled modern artificial intelligence technologies, including image recognition, speech recognition, and language translation, to name a few.

Since many-body quantum physics and supervised ML both benefit from a compact representation of high-dimensional mathematical objects, applying successful techniques from one discipline to the other has led to a fruitful cross-fertilization. For example, variational ansatz based on artificial neural networks offers a useful, though less interpretable, representation of complex many-body quantum states [6–8]. Automatic classification of quantum phases of matter also benefits from supervised ML approaches [9]. In the opposite direction, techniques from many-body quantum physics can offer novel computational paradigms for supervised ML. References [10–13] propose TNs as a quantum-inspired supervised ML ansatz that can achieve high performance in image recognition tasks. Furthermore, entanglement entropy (EE) in quantum TNs can shed light on the information propagation in the dual artificial neural networks [14, 15]. Such duality between quantum TNs and artificial neural networks can help scientists scrutinize the inner working of complex neural network algorithms, providing new tools to tackle the explainability aspects of black-box DL approaches. Recent work applies quantum techniques to tackle the explainability of image recognition tasks [12], though the analysis for realistic natural language processing (NLP) tasks is still lacking.

With the goal of investigating the inner working of DL in NLP, we study the behaviors of single-layer recurrent arithmetic circuits (RACs), a class of recurrent neural networks (RNNs) that can be mapped to an MPS [14], in a ubiquitous NLP task, the sentiment analysis of movie reviews. The objective of sentiment analysis is to classify each written review into an appropriate category such as 'like' or 'dislike'. We show that, by using the EE of the dual MPS as a measure of information propagation in the networks, single-layer RACs achieve highest prediction accuracies when the information propagation saturates. By saturation we mean there exists a critical model size χ^* (measured by the number of hidden neurons) such that larger models have prediction accuracies and the EE as high as those of the model with size χ^*. Thus, there is a *minimal* single-layer RACs model that can best estimate the statistics of sentiment analysis data. The prediction accuracies are excellent though the saturated EE is below the maximum EE restricted by the area law of an MPS.

In NLP, another crucial component of successful DL models is a word embedding, a vector representation of a word that encapsulates word semantics. While the EE analysis reveals the behaviors of information propagation within single-layer RACs, it disregards the role of word embedding. In this work, we also analyze the interplay between information propagation and the word embedding. We report that, with a trainable word embedding, single-layer RACs can achieve higher prediction accuracies at a smaller EE, compared to those of a model with a fixed embedding. As the EE drops down from its maximum value to saturation as the model size increases, the word embedding becomes more meaningful (as measured by the behavior of the cosine similarity between word vector representations). Hence, single-layer RACs have a trade-off between achieving long-range information propagation and attaining a meaningful word embedding in sentiment analysis tasks.

Although a long-range correlation in an MPS is bounded above by the area law [16], our results demonstrate that an MPS can still serve as a useful variational ansatz in a realistic NLP task, provided the input word embedding is well designed. Recently, TN models for sequence modeling have been proposed [17–21]; however, most work focus on setting up new TN-based variational ansatz that may not have an exact mapping to the RNNs counterpart. Albeit interesting, these models do not meet our goal to scrutinize the inner working of RNNs.

The manuscript is organized as follow. We begin with a self-contained background on a probabilistic sequence modeling and statistical language modeling in section 2. Sequence modeling in the era of RNNs and how to represent a word meaningfully as a vector are provided in sections 2.1 and 2.2. The mapping between RACs and MPS as well as the meaning of EE as a proxy for information propagation are discussed in section 2.3. Section 3 reports numerical methods and the results of RACs performance on sentiment analysis for the IMDb data set, a standard movie review data set, revealing the information propagation capacity in RACs. Comments on RACs' behaviors and the role of word embedding are provided in details in

the same section. Similar results for a smaller movie review data set [Rotten Tomatoes (RT)] are reported in the appendix. Finally, we conclude with the discussion and outlook in section 4.

2. Statistical language modeling with RNNs and MPS

Since language is a sequential phenomenon, in which a sequence of words (or alphabets) dictates its meaning, we first review a statistical approach to model sequences. A central mathematical object for statistical sequence modeling is the joint probability distribution $P(X_{1:T}) \equiv P(X_1, X_2, \ldots, X_T)$ where a discrete random variable X_t with $t \in \{1, \ldots, T\}$ can take a value x_t from a finite set S_N with N elements. One can regard the list of *correlated* random variables $X_{1:T}$ as a discrete time-series. Using Bayes' rules, the joint distribution can be factorized into the product of conditional probabilities, conditioned on the knowledge of the past, as

$$P(X_{1:T}) = P(X_1)\, P\left(X_2|X_1\right) P(X_3|X_{1:2}) \ldots P\left(X_T|X_{1:T-1}\right), \tag{1}$$

where $X_{1:t}$ denotes the sequence of random variables in the first t steps; i.e. $X_{1:t} \equiv (X_1, X_2, \ldots, X_t)$. However, given a time series data, inferring the conditional distribution $P\left(X_T|X_{1:T-1}\right)$ from occurrence frequency of a long data sequence can be impractical, as each realization $X_{1:T-1} = (x_1, x_2, \ldots, x_{T-1})$ typically occurs with a relative frequency $\sim N^{1-T}$ (assuming X_t is uniformly distributed over S_N), which is exponentially small in T.

One may assume a short temporal correlation in the sequence, so that the conditional probabilities depend only on the n previous steps

$$P(X_T|X_{1:T-1}) \approx P(X_T|X_{T-n:T-1}). \tag{2}$$

For a small n, a realization $X_{T-n:T-1} = (x_{T-n}, \ldots, x_{T-1})$ now occurs with a non-negligible frequency $\sim N^{-n} \gg N^{1-T}$, rendering the estimation of (2) manageable. For $n = 1$, (2) is the familiar Markov assumption, which gives a Markovian approximation of a stochastic process $X_{1:T}$ in (1).

In the context of NLP, a probabilistic model that prescribes probabilities to sequences of words (or alphabets) is called a *language model* [22]. Predicting the next word (or alphabet), given a sequence of previous words (or alphabets) is one important example with myriad applications. A model that predicts the next word based only on the last $n-1$ words according to (2) is called an *n-gram* language model.

However, even with a small $n = 4$, constructing a 4-*gram* model from a gigantic text, such as all the Wikipedia's English articles, can be impractical. Consider a random variable X_t in (1) which now takes a realization as a word w_t from S_N, a dictionary with N words. Oxford English dictionary contains $N = 171,476 \approx 10^5$ English words that are currently used [23]. In this case, the frequency of occurrence of a sequence of four words $(w_{t-4}, w_{t-3}, w_{t-2}, w_{t-1})$ can be vanishingly small $\sim N^{-4} \approx \left(10^5\right)^{-4} = 10^{-20}$, rendering the estimation of $P(X_t|X_{t-4:t-1})$ impractical. In addition, accurate prediction of the next word often depends on the words that appear in the far past. For instance, the prediction accounting for the subject-verb agreement in 'this example that we demonstrate for you …' can be syntactically wrong if the model retains only the last few words, whose prediction would be 'are'. It requires a 7-*gram* model to correctly predict 'is' in this example. Therefore, to construct a useful language model, one needs to devise a computational approach that can encapsulate a long-range correlation in a sequence of words, while also circumventing the sparsity of a long sequence problem. This can be achieved with RNNs, which we now discuss.

2.1. Statistical language modeling with RNNs

Rather than conditioning the prediction task on a window of size n as in an *n-gram* language model, RNNs allow conditioning the prediction on *all* previous words that appear in a text, approximating (1) while requiring only finite computational resources. RNNs can also perform the next item prediction as it is widely used to estimate the conditional probability $P(X_T|X_{1:T-1})$; however, one ubiquitous yet simpler task in NLP is to estimate $P(\sigma|w_{1:T})$, where σ is a discrete quantity that characterizes a sequence of words $w_{1:T}$ of length T. We will focus on characterizing the sentiment of written sentences, a task termed *sentiment analysis* in NLP, in which σ is a *binary* variable that takes a value 0 if a given sequence has a negative sentiment, and a value 1 if a given sequence has a positive sentiment. This task can be used, for example, to automatically rate product reviews or analyze news sentiment.

We now describe the simplest (Elman's/vanilla) RNN that is typically adopted to approximate conditional probabilities. Figure 1 shows a recurrent computational unit that estimates $P(\sigma|w_{1:T})$. At every time step, except the first and the last, this recurrent computational unit computes from an input $\mathbf{\Phi}(w_t)$ of dimension d_{I} and a *hidden* or *latent* vector $\boldsymbol{h}_{t-1}$ of dimension d_{H} a non-linear output function f, called

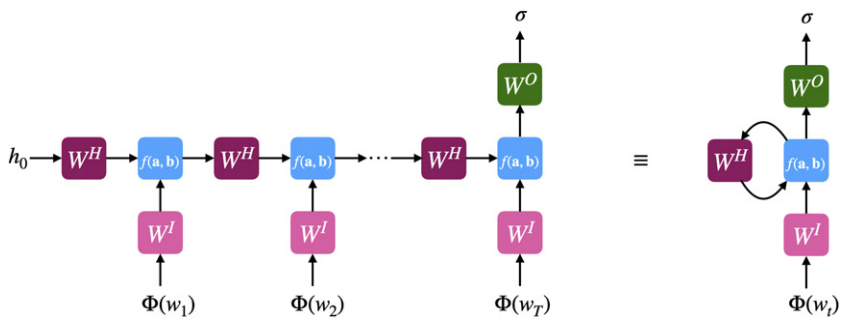

Figure 1. A schematic for the sentiment analysis (binary classification) of a word sequence $w_{1:T}$ performed by a vanilla RNN, which outputs the probability that the sequence has a positive sentiment $\sigma \in [0, 1]$. Each word w_t is embedded as a vector $\mathbf{\Phi}(w_t) \in \mathbb{R}^{d_\mathrm{I}}$, with d_I of around 300–500 for other large language modeling tasks [24, 25]. The recurrent computation is iterated from a dynamical system $\boldsymbol{h}_t = f\left(W^\mathrm{H}\boldsymbol{h}_{t-1}, W^\mathrm{I}\mathbf{\Phi}\left(w_t\right)\right) \in \mathbb{R}^{d_\mathrm{H}}$, with some non-linear map (activation function) $f : \mathbb{R}^{d_\mathrm{H}} \times \mathbb{R}^{d_\mathrm{I}} \to \mathbb{R}^{d_\mathrm{H}}$. In the last time step $t = T$, when sentiment classification is performed, one computes the sigmoid function $\sigma\left(W^\mathrm{O}\boldsymbol{h}_T + b^\mathrm{O}\right)$ which assigns the probability that the sequence $w_{1:T}$ has a positive sentiment.

output hidden vector $\boldsymbol{h}_t$ of dimension d_H,

$$\boldsymbol{h}_t \equiv f(W^\mathrm{I}\mathbf{\Phi}(w_t), W^\mathrm{H}\boldsymbol{h}_{t-1}, \boldsymbol{b}). \tag{3}$$

Here W^I is the input weight matrix with dimension $d_\mathrm{H} \times d_\mathrm{I}$ that aggregate signals from the input vector $\mathbf{\Phi}(w_t)$, W^H is the weight matrix aggregating the signals from the hidden vector whose dimension is $d_\mathrm{H} \times d_\mathrm{H}$, and $\boldsymbol{b}$ is the so called bias with dimension d_H. The non-linear *activation function* f imitates the behaviors of biological neurons, such that the weighted input and the weighted hidden vector are summed together (mimicking aggregation of potentials), while the bias (representing the background neuron's potential) is added to the aggregated weighted sum. In Elman's RNNs, each component $\boldsymbol{s}_t^{(i)}$ of the aggregated sum $\boldsymbol{s}_t = W^\mathrm{I}\mathbf{\Phi}(w_t) + W^\mathrm{H}\boldsymbol{h}_{t-1} + \boldsymbol{b}$ represents a total potential each hidden neuron $i \in \{1, \ldots, d_\mathrm{H}\}$ experiences, which triggers each hidden neuron to be activated and outputs a corresponding component of the hidden vector into the next time step as $\boldsymbol{h}_t^{(i)} = f(\boldsymbol{s}_t^{(i)})$. The notation in the last equality and (3) signifies that the *same* activation function f is applied identically to every hidden neuron. Standard non-linear activation function f motivated by neurobiology is a sigmoid function or tanh function, whereas modern ML typically employs rectified linear unit defined by $f(x) = \mathrm{Max}\{0, x\}$ and its variants [5, 26].

With (3) as a computational building block, one can iterate the computation recursively taking into account all the inputs in the sequence $\mathbf{\Phi}_{1:T} \equiv (\mathbf{\Phi}(w_1), \ldots, \mathbf{\Phi}(w_T))$ of size T, provided a hidden vector $\boldsymbol{h}_0$ was initialized. Due to the recursive structure, it is plausible that information in the far past can influence the output vector at the last step $\boldsymbol{h}_T$. This manifestation of long-term temporal dependencies through a recursive computation circumvents the problem of an astronomical number of parameters needed to model a long sequence encountered in the previous section. Here one only requires to store the bias vector $\boldsymbol{b}$ of dimension d_H, W^I of dimension $d_\mathrm{H} \times d_\mathrm{I}$, and W^H of dimension $d_\mathrm{H} \times d_\mathrm{H}$, all of which are independent of the sequence length T.

The simplest sentiment analysis task, which we focus on, is a binary classification task where there are only two sentiments $\sigma \in \{0, 1\}$. In such case, the final hidden vector will be passed to the classification neuron with the output weight W^O whose dimension is $1 \times d_\mathrm{H}$ together with the added scalar bias b^O as the aggregated signal of the classification neuron $s_\mathrm{O} = W^\mathrm{O}\boldsymbol{h}_T + b^\mathrm{O}$ before the classification neuron predicts a number $\hat{\sigma}_\theta \in [0, 1]$ computed from the sigmoid activation function

$$\hat{\sigma}_\theta \equiv \frac{1}{1 + \exp(-s_\mathrm{O})}, \tag{4}$$

where we denote the set of all parameters in this RNN that influences the value of this last neuron as $\theta \equiv \{\boldsymbol{b}, W^\mathrm{I}, W^\mathrm{H}, W^\mathrm{O}, b^\mathrm{O}\}$.

To train the model, one adjusts parameters $\theta \equiv \{\boldsymbol{b}, W^\mathrm{I}, W^\mathrm{H}, W^\mathrm{O}, b^\mathrm{O}\}$ to minimize the cost (loss) function C which accumulates the amount of mismatches between the true sentiment $\sigma(\boldsymbol{w}_m)$ associated with the mth sequence $\boldsymbol{w}_m \equiv w_{1:T,m} = (w_1, w_2, \ldots, w_T)_m$ and the RNNs' sentiment prediction $\hat{\sigma}_\theta(\boldsymbol{w}_m)$, for all sequences in the training sample $m \in \{1, \ldots, M\}$. For a binary classification task with the probabilistic

prediction given by (4), the cost function is typically taken as the binary cross-entropy

$$C \equiv \frac{1}{M}\sum_{m=1}^{M}\left(\sigma(\boldsymbol{w}_m)\log\left[\hat{\sigma}_\theta(\boldsymbol{w}_m)\right] + (1-\sigma(\boldsymbol{w}_m))\log\left[1-\hat{\sigma}_\theta(\boldsymbol{w}_m)\right]\right). \tag{5}$$

In a movie review task, for example, M can be the number of written reviews with predetermined sentiments from M different reviewers that encapsulates a reasonable relationship between word sequences and their associated sentiments.

Note that minimizing the cross-entropy $C \equiv H_{\mathrm{RNN}_\theta}(P)$ between the empirical distribution $P(\sigma|w_{1:T})$ constructed from the training data and the distribution predicted by the RNN parametrized by θ, denoted by $\mathrm{RNN}_\theta(\sigma|w_{1:T})$, is equivalent to minimizing the Kullback–Leibler (KL) divergence $D_{\mathrm{KL}}\left(P\|\mathrm{RNN}_\theta\right)$ [27, 28]. Since the KL divergence reflects the dissimilarity between the two distributions, the optimization (minimization) procedure of the cost function (5) would search for a vanilla RNN parametrized by θ^* that estimates well the empirical distribution $P(\sigma|w_{1:T})$. Provided the training data is properly curated and the optimization procedure (e.g., gradient methods and their modern variants [5, 26]) is reliable, one shall arrive at a reasonable statistical relationship between a *long* sequence of words and its associated sentiment parametrized by an RNN with a *finite* number of parameters θ^*. In other words,

$$P(\sigma|w_{1:T}) \approx \mathrm{RNN}_{\theta^*}(\sigma|w_{1:T}). \tag{6}$$

This is the main philosophy behind statistical language modeling using RNNs.

2.2. On the word vector embedding $\boldsymbol{\Phi}$

Suppose one randomly assigns or 'tokenizes' each word with a unique integer $w_i \in \{1,\ldots,N\}$, where $1 \leqslant i \leqslant N$ with N being the size of the dictionary. Then each written review is represented by a sequence of integers $w_{1:T} = (w_1,\ldots,w_T)$. Here, the length of each review is forced to be T, by padding 0's at the beginning of the review if its length is less than T, or by selecting only the first T words if its length is greater than T. For example, for $T = 6$, the sentence 'Physics is beautiful' can be encoded as $w_{1:T} = (0,0,0,532,3,46)$, where 'Physics' $= 532$, 'is' $= 3$, and 'beautiful' $= 46$. The tokenization process, however, artificially introduces the notion of distance between two words that does not encode word semantics.

How shall one mathematically represent words so that their semantics are encoded? A widely-adopted solution is to embed a word w as a vector $\boldsymbol{\Phi}(w) \in \mathbb{R}^{d_{\mathrm{I}}}$. By representing a word as a vector embedded in d_{I} dimensions, words with similar meanings that co-occur frequently in the same context can be assigned unique vectors such that their pairwise Euclidean distance are small. Also, a negative *cosine similarity* of the embeddings of the two words w_a, w_b computed from

$$\mathrm{sim}\left(\boldsymbol{\Phi}(w_a),\boldsymbol{\Phi}(w_b)\right) = \frac{\boldsymbol{\Phi}(w_a)}{\|\boldsymbol{\Phi}(w_a)\|_2}\cdot\frac{\boldsymbol{\Phi}(w_b)}{\|\boldsymbol{\Phi}(w_b)\|_2} \tag{7}$$

can signify that w_a and w_b rarely co-occur in the same context, and hence could have opposite meanings.

There are a few methods to numerically obtain an embedding Φ that effectively represents word semantics [24, 25, 29]. A simple yet classic Word2vec method [24], which is also adopted in our numerical experiments, is to assign the embedding function Φ as a matrix of size $d_{\mathrm{I}} \times N$, so that the ith column of Φ corresponds to the word vector $\boldsymbol{\Phi}(w_i)$ of the word w_i. The embedding dimension d_{I} is a hyper-parameter that can be tuned to best suit the problem. The matrix elements in Φ are treated as variational parameters to be optimized along with the optimization of the RNN for a language modeling task of interests. For example, to perform a sentiment analysis using a vanilla RNN without knowing *a priori* the embedding matrix, one would add the matrix elements of Φ into the trainable parameters $\tilde{\theta} \equiv \{\theta,\Phi\} = \{\boldsymbol{b}, W^{\mathrm{I}}, W^{\mathrm{H}}, W^{\mathrm{O}}, b^{\mathrm{O}}, \Phi\}$. In this way, training the RNN according to section 2.1 will not only yield the network parameters, but also the word vector embedding. With a sufficiently large and well curated training data set, one expects that the embedding matrix Φ would effectively encapsulate word semantics in the dictionary of interests.

Despite the empirical success of statistical language modeling using vanilla RNNs together with the well-trained word embedding as explained above, highly-nonlinear iterations of (3) by standard activation functions render the analysis of how RNNs approximate empirical sequence distributions very challenging. In the following, we review recent attempts to analyze the expressiveness of RNNs (i.e. the set of function that can be effectively parametrized by RNNs) with a specific activation function, through the mapping to their dual the TN counterparts.

2.3. Recurrent arithmetic circuit (RAC) and the mapping to matrix product state (MPS)

Consider the activation function defined by the Hadamard product

$$f_{\mathrm{RAC}}(\boldsymbol{a},\boldsymbol{b}) = \boldsymbol{a} \odot \boldsymbol{b}, \tag{8}$$

which is the element-wise multiplication $f^{(i)}_{\mathrm{RAC}}(\boldsymbol{a},\boldsymbol{b}) \equiv \boldsymbol{a}^{(i)} \cdot \boldsymbol{b}^{(i)}$. RNNs with RAC activation function, known as RACs, have recently received increasing attention and share computational paradigm similar to the multiplicative RNNs [30–33]. More importantly, references [14, 15] show that a single-layer RAC can be mapped to the dual MPS, taking the inspiration from the tensor train (TT) decomposition of [34]. By studying RACs, the analysis of learning in RNNs for temporal data can thus be performed from many-body quantum physics perspectives. For instance, one can compute the EE of the dual MPS, which is a measure of the amount of temporal correlation that can be supported by the network [14, 15]. The larger the EE means that the output of network computation crucially depends on the temporal data in the further past, enabling the network to have a longer-range memory.

The TN diagrams in figure 2 summarize the equivalence between the computation of the standard RNNs-based sentiment analysis with RAC activation function and that of the dual MPS. By defining the tensor of rank 3 of the form

$$A^{s_t}_{\alpha_t\alpha_{t-1}} \equiv \sum_{\tilde{\alpha}_{t-1},\tilde{s}_t=1}^{d_{\mathrm{H}}} W^{\mathrm{H}}_{\tilde{\alpha}_{t-1}\alpha_{t-1}} \delta_{\alpha_t\tilde{\alpha}_{t-1}\tilde{s}_t} W^{\mathrm{I}}_{\tilde{s}_t s_t}, \tag{9}$$

where δ_{jkl} is 1 if $j = k = l$ and is 0 otherwise, the state evolution by one time step can be computed by the tensor contraction between the hidden vector $\boldsymbol{h}_{t-1}$, the tensor $A^{s_t}_{\alpha_t\alpha_{t-1}}$, and the input word vector $\boldsymbol{\Phi}(w_t)$, resulting in the tensor of rank 1 describing the hidden vector of the next time step whose component α_t is given by

$$\begin{aligned}
\boldsymbol{h}^{(\alpha_t)}_t &= \sum_{\alpha_{t-1}=1}^{d_{\mathrm{H}}} \sum_{s_t=1}^{d_{\mathrm{I}}} \boldsymbol{h}^{(\alpha_{t-1})}_{t-1} A^{s_t}_{\alpha_t\alpha_{t-1}} \boldsymbol{\Phi}^{(s_t)}(w_t) \\
&= \sum_{\tilde{\alpha}_{t-1},\tilde{s}_t=1}^{d_{\mathrm{H}}} \sum_{\alpha_{t-1}=1}^{d_{\mathrm{H}}} \sum_{s_t=1}^{d_{\mathrm{I}}} \left(W^{\mathrm{H}}_{\tilde{\alpha}_{t-1}\alpha_{t-1}} \boldsymbol{h}^{(\alpha_{t-1})}_{t-1} \right) \delta_{\alpha_t\tilde{\alpha}_{t-1}\tilde{s}_t} \left(W^{\mathrm{I}}_{\tilde{s}_t s_t} \boldsymbol{\Phi}^{(s_t)}(w_t) \right) \\
&= \left(W^{\mathrm{H}} \boldsymbol{h}_{t-1} \right)^{(\alpha_t)} \cdot \left(W^{\mathrm{I}} \boldsymbol{\Phi}(w_t) \right)^{(\alpha_t)} \\
&= f^{(\alpha_t)}_{\mathrm{RAC}}(W^{\mathrm{H}} \boldsymbol{h}_{t-1}, W^{\mathrm{I}} \boldsymbol{\Phi}(w_t)).
\end{aligned}$$

Therefore, given a sequence $(\boldsymbol{\Phi}(w_1), \ldots, \boldsymbol{\Phi}(w_T))$ and the initialization of the hidden vector $\boldsymbol{h}_0$ with dimension d_{H}, the output hidden vector at time T can be computed from the contraction between the translational invariant MPS

$$\Psi^{s_T\ldots s_2 s_1}_{\alpha_T\alpha_0} \equiv \sum_{\alpha_1,\alpha_2,\ldots,\alpha_{T-1}=1}^{d_{\mathrm{H}}} A^{s_T}_{\alpha_T\alpha_{T-1}} \ldots A^{s_2}_{\alpha_2\alpha_1} A^{s_1}_{\alpha_1\alpha_0}, \tag{10}$$

the tensor of rank T constructed from the input sequence

$$\Phi^{s_T\ldots s_2 s_1} \equiv \boldsymbol{\Phi}^{(s_T)}(w_T) \ldots \boldsymbol{\Phi}^{(s_2)}(w_2) \boldsymbol{\Phi}^{(s_1)}(w_1), \tag{11}$$

and the initial hidden vector $\boldsymbol{h}_0$ as follows

$$\boldsymbol{h}^{(\alpha_T)}_T = \sum_{\alpha_0=1}^{d_{\mathrm{H}}} \sum_{s_1,s_2,\ldots,s_T=1}^{d_{\mathrm{I}}} \left(\Phi^{s_T\ldots s_2 s_1} \right) \left(\Psi^{s_T\ldots s_2 s_1}_{\alpha_T\alpha_0} \right) \boldsymbol{h}^{(\alpha_0)}_0. \tag{12}$$

The last equality is compactly represented by the standard TN graphical notation as shown in figure 2(b), whose building block is the tensor A of (9) represented graphically in figure 2(c). For sentiment analysis using binary classification, the final contraction (12) will then be used to compute the probability that the input sequence $w_{1:T}$ has a positive sentiment through the usual sigmoid activation function as in (4). Note that, in many-body quantum physics language, the dimension of the hidden unit d_{H} is in fact the *bond dimension* $\chi \equiv d_{\mathrm{H}}$ of the MPS.

2.4. Entanglement entropy of the MPS as a proxy for information propagation in RAC

Since the fundamental building block of the computation is the translational invariant MPS, we can compute the EE by partitioning the MPS into two subsystems through the standard

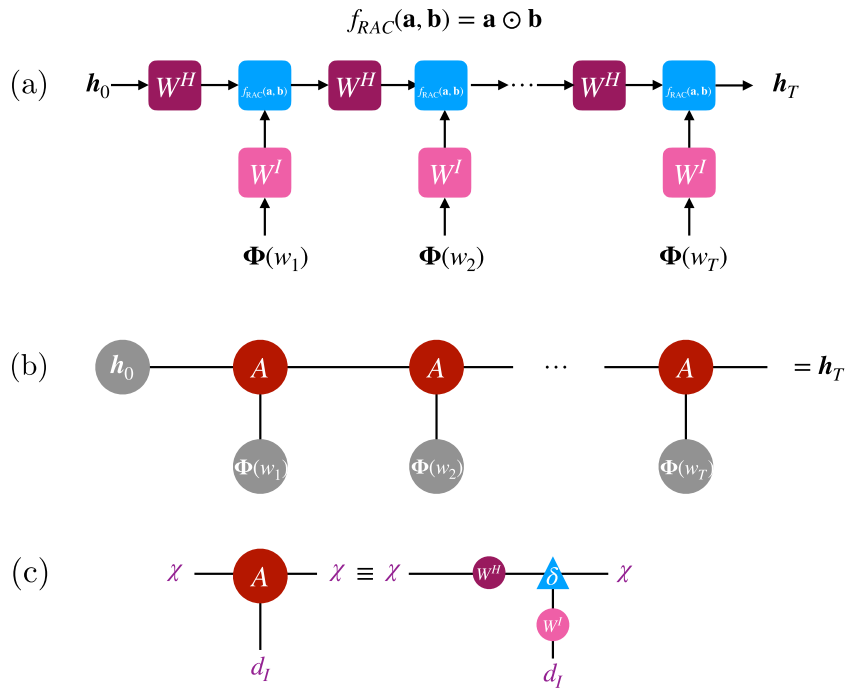

Figure 2. The graphical representation of the mapping between a single-layer RACs for sentiment analysis task (a) to the dual MPS (b). As a fundamental building block, the translational invariant MPS (without the contraction by boundary vector $\boldsymbol{h}_0$) consists of the rank-3 tensor $A^{s_t}_{\alpha_t\alpha_{t-1}} \equiv \sum_{\tilde{\alpha}_{t-1},\tilde{s}_t=1}^{d_{\rm H}} W^{\rm H}_{\tilde{\alpha}_{t-1}\alpha_{t-1}} \delta_{\alpha_t\tilde{\alpha}_{t-1}\tilde{s}_t} W^{\rm I}_{\tilde{s}_t s_t}$, where the triangle in (c) represents the tensor of rank 3 defined by δ_{jkl} which is equal to 1 if $j = k = l$ and is 0 otherwise. The structure of the building block in (c) arises from the Hadamard product imposed by RAC activation function in (a). Here we denote χ as the bond dimension of the MPS, which is equal to $d_{\rm H}$, the number of hidden units of RACs in (a). The vertical bond in (c) has the dimension $d_{\rm I}$, identical to that of the word vector embedding $\boldsymbol{\Phi}$.

Schmidt-decomposition, and compute the resulting von-Neumann entropy [35]. However, the MPS in (10) still has an open boundary. To make the boundary close and properly compute the EE, one needs to contract the indices α_0, and α_T by vectors of dimension $\chi = d_{\rm H}$. In the limit $T \gg 1$, this choice of vectors should not significantly affect the EE if the partition is made at half of the chain. The details on an appropriate choice of vectors for contraction to close the boundary in our numerical experiments will be discussed in the following section. Suppose now that the contraction has been properly made and the MPS with a close boundary is given by $\tilde{\Psi}^{s_T\ldots s_2 s_1}$, then the corresponding quantum state of the MPS is

$$|\tilde{\Psi}\rangle_{\rm MPS} = \sum_{s_1,\ldots,s_T=1}^{d_{\rm I}} \tilde{\Psi}^{s_T\ldots s_2 s_1} |s_T\rangle \otimes \cdots \otimes |s_2\rangle \otimes |s_1\rangle, \tag{13}$$

which has the Schmidt decomposition (singular value decomposition) for the bipartition at the $\lceil T/2 \rceil$th bond into the left and right sectors as

$$|\tilde{\Psi}\rangle_{\rm MPS} = \sum_{i=1}^{r} \lambda_i \, |\phi_i^{\rm L}\rangle \otimes |\phi_i^{\rm R}\rangle \,, \tag{14}$$

where the Schmidt coefficients λ_i's are the real, non-negative singular values satisfying $\sum_{i=1}^{r} \lambda_i^2 = 1$, and r is the Schmidt rank (Schmidt number). The Schmidt rank r is 1 only for a product state and is greater than 1 when a state has the two subsystems that are entangled.

The von-Neumann (entanglement) entropy is a well-defined measure of entanglement between the two subsystems and can be calculated as

$$S = -\sum_{i=1}^{r} \lambda_i^2 \log_2 \lambda_i^2. \tag{15}$$

Importantly, this EE, when translated into the RNN language, can quantify the amount of temporal correlation between the signal in the earlier times $\{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{\lceil T/2 \rceil}\}$ and the signal in the later times $\{\boldsymbol{h}_{\lceil T/2 \rceil+1}, \ldots, \boldsymbol{h}_T\}$, also known as *start–end separation rank* [14, 36]. If the EE is zero, the signals in the earlier and the later times are statistically independent. The prediction task from models with vanishing EE thus has a short-term memory, neglecting the knowledge in the past $t < \lceil T/2 \rceil$. One then would expect the

models with larger EE to be more desirable in encapsulating long-range sequence correlations. We shall then intuitively interpret the EE computed from (15) as *the proxy for information propagation* in the RACs networks. RACs that possess low EE might have a low expressiveness (high bias in statistical learning theory framework), and thus are unable to efficiently approximate data distribution with long-range statistical correlations.

It is well known that an MPS obeys the area law of EE, which constrains the upper bound on EE as $S = O(\log_2(\chi))$ [37]. In fact, the state with the maximum entropy in (15) is attained with the value $\log_2(\chi)$ when all the Schmidt coefficients are identically $1/\sqrt{\chi}$ with the Schmidt rank $r = \chi$.[9] Since the upper bound is independent of the system size T, temporal data with long-range statistical correlation might not be efficiently approximated by an MPS (or, equivalently, single-layer RACs) variational ansatz. This result seems to warrant a no-go statement for using MPS to model sequential data with long-range correlation. Alternative models that can incorporate long-range correlation, such as deep RACs, have been theoretically analyzed, though no experimental results on these network performance on realistic temporal data sets have been reported [14, 15, 36].

However, thus far, the analysis on the expressive power of single-layer RACs concerns only that of the recurrent units, not of the combined system that includes a representation Φ of the input embedding. In practice, even in simple RNNs, incorporating trainable word embedding function Φ into the model can tremendously increase the prediction accuracy. In the following section, we shall investigate, in realistic sequence modeling settings, whether low EE of models alone suffices to enforce a no-go theorem for such models. The answer is an affirmative no, and single-layer RACs are still useful in realistic sequence modeling tasks.

3. Sentiment analysis by single-layer RACs with an entanglement entropy below the area law: numerical experiments

In this section, we first provide the details of our numerical experiments to analyze the behaviors of single-layer RACs for sentiment analysis in realistic movie reviews data sets. Then, we discuss the importance of additive biases in RAC activation function, and elucidate how to convert RACs with additive biases into MPS for the purpose of EE analysis. We then report the behaviors of single-layer RACs together with their EE. First, we show that when a pre-trained word vector embedding $\mathbf{\Phi}$ is fixed, the prediction accuracies strongly correlate with the amount of information propagation within RACs as measured by the EE. Interestingly, the high prediction accuracies saturate when the EE saturates, enabling one to determine the minimal model (model with the smallest bond dimension χ^* that saturates the EE) that can best approximate the statistics of sequential data. This EE saturation is a reflection of the convergence of entanglement spectrum to the limiting entanglement spectrum that we numerically report. Second, when the embedding layer is trained along with RACs, there is an intriguing interplay between RACs and the embedding layer such that, even when the EE drops, the prediction accuracy is boosted. Contrary to a common belief that long-range information propagation in the network is the main source of RNN's expressiveness, we show that, when the bond dimension is large, RACs harness its high expressiveness from meaningful word embeddings.

3.1. Details of the numerical experiments

In the main text, we use the IMDb movies and critic reviews data set, which is one of the standard data sets for sentiment analysis using binary classification [38]. The training set and the test set contain $M = 40\,000$ and 10 000 different samples respectively. Both sets are approximately balanced: the ratio of positive to negative reviews in the training and the test set are given by, respectively, 20 027:19 973 and 4913:5027. The length of each review is set to $T = 50$ and the dictionary size is $N = 10\,000$. We also perform sentiment analysis on the RT data set using the same methodology which leads to similar conclusions as the ones presented in this section. The details and the results for RT data sets are shown in the appendix.

To train the model, we implement single-layer RACs using Keras [39] which is a high-level API of TensorFlow. Batch training is deployed with 200 epochs with the batch size of 128. An early stopping is applied to terminate the training process if the change in the cost function after four epochs is smaller than 0.001. The cost function is optimized using Adam optimizer. The optimization process is repeated 50 times, each with a random initialization of the variational parameters, and the averaged prediction accuracies for the training and the test data set are obtained for each number of hidden neurons d_{H}.

[9] The discrete distribution P_i that maximizes the Shannon's entropy $-\sum_{i=1}^{m} P_i \log_2 P_i$ is the uniform distribution $P_i = 1/m$.

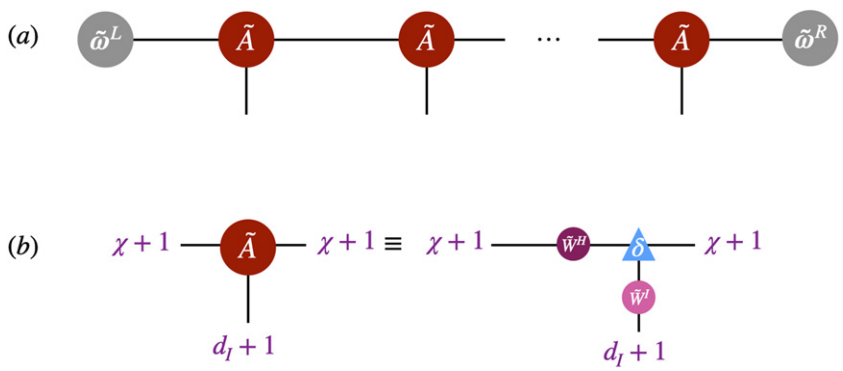

Figure 3. (a) The modified translational invariant MPS in a close boundary for calculating the EE of RACs with additive biases $\boldsymbol{b}_{\rm I}, \boldsymbol{b}_{\rm H}$. (b) The tensor $\tilde{A}^{s_t}_{\alpha_t\alpha_{t-1}}$ is defined according to (20), which consists of the contraction between δ_{jkl} and the modified weight matrices $\tilde{W}^{\rm H}, \tilde{W}^{\rm I}$ of (16).

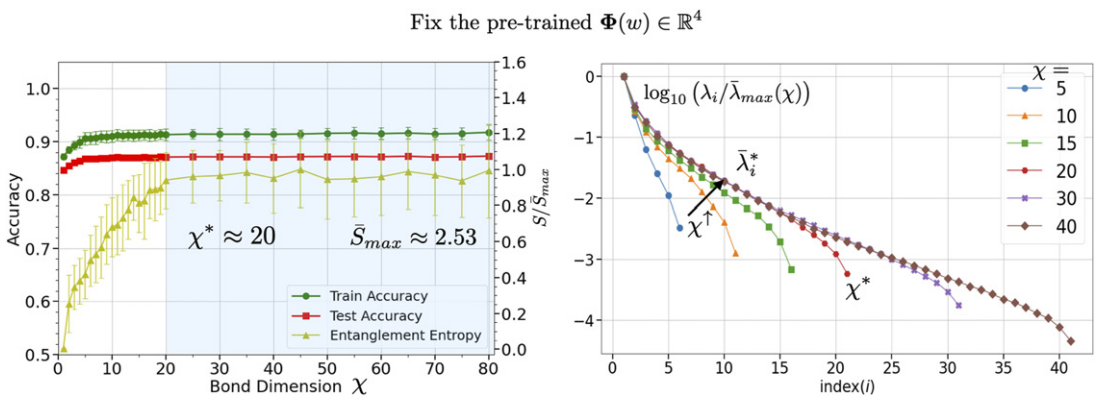

Figure 4. The behaviors of trained RACs with additive biases for sentiment analysis of the IMDb data set, when the fixed pre-trained word embedding has dimension $d_{\rm I} = 4$. (Left) The prediction accuracies saturate when the EE saturates. The critical bond dimension $\chi^* \approx 20$, at which the EE is maximal, enables one to infer a minimal single-layer RACs model for IMDb sentiment analysis. The average maximum EE $\bar{S}_{\rm max}$, which is below the upper bound from the area law, is attained when $\chi \gtrsim \chi^*$. (Right) Above the critical bond dimension, the Schmidt coefficients collapse onto the limiting entanglement spectrum $\lambda^*{}_i$ that sets the slowest exponential decay rate of the Schmidt coefficients. Here, the average maximum Schmidt coefficients for $\chi = 5, 10, 15, 20, 30, 40$ are $\bar{\lambda}_{\rm max}(\chi) \approx 0.75, 0.63, 0.58, 0.51, 0.51, 0.52$, respectively. The average are taken over 50 trained models; each begins with a random initialization of RACs with additive biases. The error bars for λ_i are not shown for the clarity of presentation.

3.2. Entanglement entropy of single-layer RACs with additive biases

It is important to note that for RACs not to suffer from the vanishing or exploding gradient problem during model training[10], we found that it is crucial to add trainable bias vectors $\boldsymbol{b}_{\rm H}, \boldsymbol{b}_{\rm I} \in \mathbb{R}^\chi$ to the aggregated inputs of the RAC activation function. In particular, to achieve model trainability in practice requires the time evolution of the form $\boldsymbol{h}_t \equiv f_{\rm RAC}(W^{\rm I}\mathbf{\Phi}(w_t) + \boldsymbol{b}_{\rm I}, W^{\rm H}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\rm H})$. Fortunately, recasting the recurrent computation with additive bias vectors as the MPS structure only requires a minor modification to the prescriptions in the previous section, which we now discuss (figure 3).

Define the new input and hidden weight matrices as

$$\tilde{W}^I \equiv \underbrace{\left[\begin{array}{c|c} W^I & \boldsymbol{b}_I \\ \hline 0\ldots0 & 1 \end{array}\right]}_{d_I+1} \Big\}\chi+1, \qquad \tilde{W}^H \equiv \underbrace{\left[\begin{array}{c|c} W^H & \boldsymbol{b}_H \\ \hline 0\ldots0 & 1 \end{array}\right]}_{\chi+1} \Big\}\chi+1. \tag{16}$$

[10] Since RACs iteratively multiply signals, backpropagation during gradient computation can lead to the iterated product of very small numbers or very large numbers for poorly initialized training parameters, leading to vanishing or exploding gradients problem respectively. Adding a trainable bias is a way to control the scale of multiplicative iteration and help mitigate the vanishing or exploding gradients problem.

Define also the new word vector embedding and the new hidden vector

$$\tilde{\boldsymbol{\Phi}}(w_t) \equiv \begin{bmatrix} \boldsymbol{\Phi}(w_t) \\ 1 \end{bmatrix}, \qquad \tilde{\boldsymbol{h}}_t \equiv \begin{bmatrix} \boldsymbol{h}_t \\ 1 \end{bmatrix}. \tag{17}$$

These definitions give

$$\tilde{W}^{\mathrm{I}}\tilde{\boldsymbol{\Phi}}(w_t) = \begin{bmatrix} W^{\mathrm{I}}\boldsymbol{\Phi}(w_t) + \boldsymbol{b}_{\mathrm{I}} \\ 1 \end{bmatrix}, \qquad \tilde{W}^{\mathrm{H}}\tilde{\boldsymbol{h}}_t = \begin{bmatrix} W^{\mathrm{H}}\boldsymbol{h}_t + \boldsymbol{b}_{\mathrm{H}} \\ 1 \end{bmatrix}. \tag{18}$$

Therefore,

$$\tilde{\boldsymbol{h}}_t = \left(\tilde{W}^{\mathrm{H}}\tilde{\boldsymbol{h}}_{t-1}\right) \odot \left(\tilde{W}^{\mathrm{I}}\tilde{\boldsymbol{\Phi}}(w_t)\right) = \begin{bmatrix} f_{\mathrm{RAC}}(W^{\mathrm{H}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\mathrm{H}}, W^{\mathrm{I}}\boldsymbol{\Phi}(w_t) + \boldsymbol{b}_{\mathrm{I}}) \\ 1 \end{bmatrix}. \tag{19}$$

The last equality states that the time evolution from RAC with the bias vectors of the original problem can be encoded into the time evolution from standard RAC (without additive biases) in one higher dimension, resulting in the translational invariant MPS with the following tensor as a building block

$$\tilde{A}^{s_t}_{\alpha_t\alpha_{t-1}} \equiv \sum_{\tilde{\alpha}_{t-1},\tilde{s}_t=1}^{\chi+1} \tilde{W}^{\mathrm{H}}_{\tilde{\alpha}_{t-1}\alpha_{t-1}} \delta_{\alpha_t\tilde{\alpha}_{t-1}\tilde{s}_t} \tilde{W}^{\mathrm{I}}_{\tilde{s}_t s_t}, \tag{20}$$

where the input index s_t now takes the value from $\{1, \ldots, \chi + 1\}$.

After we obtain all the variational parameters including $\boldsymbol{b}_{\mathrm{I}}, \boldsymbol{b}_{\mathrm{H}}$ at the end of a training procedure with Adam optimizer, $\tilde{A}^{s_t}_{\alpha_t\alpha_{t-1}}$ that defines the MPS/TT with an open boundary can be constructed. The EE is computed according to section 2.4, where we close the left and the right boundary by the contraction with the boundary vectors $\tilde{\boldsymbol{\omega}}^{\mathrm{L}}, \tilde{\boldsymbol{\omega}}^{\mathrm{R}} \in \mathbb{R}^{\chi+1}$ defined by

$$\tilde{\boldsymbol{\omega}}^{\mathrm{L}} \equiv \tilde{\boldsymbol{h}}_0 = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \qquad \tilde{\boldsymbol{\omega}}^{\mathrm{R}} \equiv \begin{bmatrix} \mathbf{1} \\ 1 \end{bmatrix}. \tag{21}$$

$\tilde{\boldsymbol{\omega}}^{\mathrm{L}}$ is chosen as the left boundary vector because the initial hidden vector $\boldsymbol{h}_0$ fed into vanilla RNNs is typically chosen to be a zero vector, whereas $\tilde{\boldsymbol{\omega}}^{\mathrm{R}}$ is chosen as the right boundary vector to ensure that all the output components are taken into account. But, in the large T limit, these choices should not significantly change the EE when the bipartition is taken at the bond $\lceil T/2 \rceil$, which are far away from the boundary.

3.3. RACs with a pre-trained embedding layer

To isolate the interaction between RACs and the embedding layer, we pre-train the word embedding Φ (recall section 2.2) independently from RACs. First, we train Φ on the IMDb training data with a flatten layer publicly available in Keras, while the output is still the sigmoid function discussed earlier. The flatten layer contains no trainable parameters and as a result the classification accuracy is optimized based solely on the trainable word embedding. After we train the embedding layer for 100 epochs with the early stopping criterion explained in section 3.1, we arrive at a pre-trained Φ that is not specifically optimized for RACs, thereby isolating the expressiveness that could arise from the interaction between RACs and the embedding layer. After we obtain this pre-trained embedding layer Φ, Φ is fixed and training optimizes only weights and biases of RACs. Figure 4 (left) shows the prediction accuracies and the EE as a function of the bond dimension χ for embedding dimension $d_{\mathrm{I}} = 4$. It can be seen that from bond dimension 1 to approximately 20, the training accuracy increases monotonically from 87.1% to 91.5% while the test accuracy increases from 84.7% to 86.3%. Both quantities saturate at $\chi \approx 20 \equiv \chi^*$. The EE also increases rapidly before the onset of the accuracy saturation, then for $\chi > \chi^*$ it saturates at the (average) maximum value of $\bar{S}_{\max} \approx 2.53$. The results suggest a critical model size χ^* such that RACs expressiveness is maximal. Above this critical size both the prediction accuracies and the EE saturate. For practical purposes, this critical size χ^* is valuable for identifying a minimal single-layer RACs model that can best estimate the statistics of IMDb training data set.

For the IMDb data set, the minimal model size for single-layer RACs with a fixed pre-trained embedding with $d_{\mathrm{I}} = 4$ is $\chi^* \approx 20$. We also observe similar behaviors on the saturation of prediction accuracies that correspond to the saturation of EE for larger pre-trained embeddings with $d_{\mathrm{I}} = 8, 16, 32$ with the average maximum EE of $\bar{S}_{\max} \approx 3.87, 4.86, 5.09$, respectively. For larger embedding dimensions, not only the maximum EE increases, the critical bond dimensions and the saturated prediction accuracies also increase (not shown here due to redundancy of the plots). These results are not specific to the IMDb

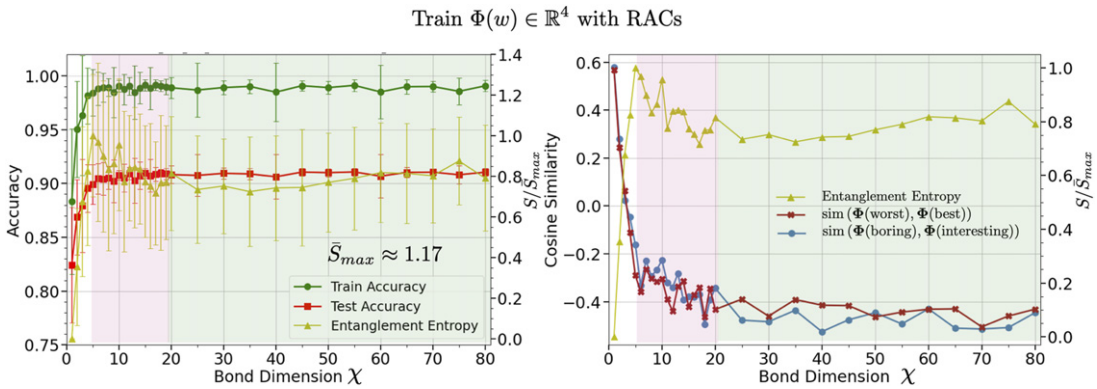

Figure 5. The behaviors of RACs with additive biases for sentiment analysis of the IMDb data set, when the word embedding of dimension $d_1 = 4$ is trained together with RACs. (Left) The prediction accuracies slowly increase when the EE drops down from the maximum value $\bar{S}_{\max} \approx 1.17$ at $\chi \approx 5$ to the saturated value $0.8\bar{S}_{\max}$ at $\chi \approx 20$, after which the accuracies also saturate. The saturated EE here is much smaller than that of the fixed embedding case, suggesting that the expressivity is harnessed from a more meaningful representation Φ. (Right) The cosine similarity between the two word embedding computed from (7) reveals that indeed the expressivity is boosted via meaningful word embeddings, which arise in larger single-layer RACs.

data set, as we observe similar trends in single-layer RACs with a fixed pre-trained embedding in a smaller RT movie review data set as well. The results for the RT data set is provided in the appendix.[11]

To understand how the EE becomes saturated above a critical bond dimension χ^*, we investigate the behaviors of the average Schmidt coefficients for the model size from $\chi = 5$ to $\chi = 40$. Interestingly, figure 4 (right) reveals that above the critical model size, the larger values of the entanglement spectrum (the function defined by the Schmidt coefficients indexed in a descending order) all collapse onto a *limiting entanglement spectrum* $\bar{\lambda}_i^*$, which exhibits the *slowest possible* exponential decay rate of the Schmidt coefficients. Thus this limiting entanglement spectrum $\bar{\lambda}_i^*$ defines the average maximum EE achievable by our MPS ansatz for this data set, whose value is given by

$$\bar{S}_{\max} = -\sum_{i=1}^{\chi^*} \bar{\lambda}^{*2}_i \log_2 \bar{\lambda}^{*2}_i. \tag{22}$$

This unique explainability of RACs allows us to infer a minimal RNNs-based model with the minimal number of hidden neurons $d_{\mathrm{H}}^* + 1 = \chi^*$ for a given task, which is not possible with standard RNNs. From statistical learning theory point of view, the limiting function $\bar{\lambda}^*{}_i$ determines the bias (in the bias-variance tradeoff sense) of single-layer RACs, which constrains the information propagation capacity as measured by the average maximum EE (22). It is interesting to note that the maximum EE is below the upper bound from the area law of $\log_2(\chi)$, as $\bar{S}_{\max} \approx 1.17 < 4.32 \approx \log_2(\chi = 20)$. Hence, a realistic sequence modeling task such as sentiment analysis can still achieve high prediction accuracies using easily trainable RACs, even when the maximum information propagation is bounded above. In fact, the embedding layer Φ plays a crucial role in attaining high expressive power, as we show next.

3.4. The interplay between RACs and the word embedding

To analyze the interplay between the recurrent units in RACs and the embedding layer Φ, we now train both components simultaneously. The prediction accuracy and the EE as a function of the bond dimension is depicted in figure 5 (left). It can be seen that the training and the test accuracy rapidly increases to 98.3% and 90%, respectively, at $\chi = 5$. The training accuracy then saturates and fluctuates mildly around 98.5% for $\chi > 5$, while the test accuracy slowly increases for $\chi \in [5, 20]$, after which it saturates at around 90.5% accuracy. Despite being simple, our model is ranked 21 (out of 35) in top-performing models (measured by test accuracy) for IMDb sentiment analysis [40]. The best performing model [41] achieving the test accuracy of 97.2% also uses simple neural network architecture but with the improved quality of the word embeddings. Interestingly, unlike in the fixed word embedding case where the maximum of EE is attained at its saturation, here the EE attains its maximum at $\chi \approx 5$ at the value of $\bar{S}_{\max} \approx 1.17$ before dropping down and saturating at $0.8\bar{S}_{\max}$ when $\chi \approx 20$, after which it fluctuates mildly around the saturated value. Although the EE drops after its peak value, the prediction accuracies counter-intuitively increase. Also,

[11] For the embedding dimension 4, the critical bond dimension for RT movie data set is $\chi^* \approx 40$, beyond which the EE very slowly increases and plateaus out at the maximum value of 1.

compared to the fixed embedding case, $\bar{S}_{\max}$ here is smaller and the prediction accuracies, especially the training accuracy, are higher. These behaviors also arise in larger word embedding size of $d_{\mathrm{I}} = 8, 16, 32$, though the maximum EE at the peak are larger and occurs at a larger bond dimension for a larger model (the plots are not shown here due to redundancy). The larger model also attains higher saturated prediction accuracies. These results suggest that information propagation or long-range temporal correlation in sequence modeling is not the main source of expressiveness in estimating the distribution $P(\sigma|w_{1:T})$ in sentiment analysis tasks. In fact, the drop in the EE as the accuracies increase suggests that single-layer RACs must have gained the expressivity through the word embedding Φ.

To test the hypothesis, we plot the cosine similarity (7) between embedding vectors of two opposite words that most frequently appear and tends to have a strong influence on the review sentiment, i.e. 'boring' and 'interesting', 'worst' and 'best', depicted in figure 5 (right). We see that the cosine similarity drops monotonically with the bond dimension and saturates at $\chi \approx 5$. This might suggest that for $\chi < 5$, the prediction accuracy stems mostly from the temporal correlation in RACs, while at $\chi \geqslant 5$, the word embedding layer better learns word semantics and start to contribute to higher prediction accuracy.

4. Discussion and outlook

We have recasted single-layer RACs with additive biases as the dual MPSs for the EE analysis of a real-world sequence modeling task, the sentiment analysis of large realistic movie review data sets. The results elucidate that, although the EE of the models is bounded above, single-layer RACs can harness their expressive power from trainable word embedding $\mathbf{\Phi}$, achieving considerably high prediction accuracies. Even for a fixed word embedding, single-layer RACs can already achieve high prediction accuracies that saturate when the EE saturates at its maximum value $\bar{S}_{\max}$. This $\bar{S}_{\max}$ allows one to identify the minimal bond dimension χ^* that RACs can best approximate the sentiment distribution of data sequence $P(\sigma|w_{1:T})$. This $\bar{S}_{\max}$ is also *below* the upper bound of the area law for EE of an MPS. Therefore, for sentiment analysis tasks, a low EE is not a warrant to disregard simple yet easily trainable models such as single-layer RACs. Importantly, the crucial interplay between information propagation in the recurrent networks (as reflected by the EE) and the meaningful word embedding $\mathbf{\Phi}$ enables single-layer RACs to very well estimate the sentiment distribution of word sequence $P(\sigma|w_{1:T})$. Our analysis also quantitatively reveals the nature of movie review sentiment analysis that NLP practitioners are intuitively aware of; reading only a few statements that contain meaningful keywords might be an efficient strategy to correctly classify the sentiment of a long review.

Despite the simplicity of our single-layer architecture with low-dimensional word embeddings, we still achieve the test accuracy of 90.5% for the sentiment analysis of the IMDb data set, placing our minimal model in the list of top-performing models [40]. Some top-performing models utilize powerful modern neural network architectures such as graph neural networks [42] or transformers [43, 44]. All of which still lack explainability. It is interesting to note that some simple models, such as classic LSTM architectures (with high quality word embeddings) [45, 46], are also in the top-performing list. Remarkably, the best performing model utilizes a very simple neural architecture with the emphasis on constructing highest quality word embeddings [41]. This observation agrees with our quantitative evidence that long-range information propagation is not the main source for RNNs' successes in sentiment analysis, but high model expressiveness can be attained from the subtle interplay between the information propagation and the quality of word vector embeddings.

It would be interesting to generalize the current analysis to deep (multi-layer) RACs models [36] to see the interplay between long-range information propagation in the recurrent networks and the meaningful word embedding in other realistic NLP tasks, such as sequence to sequence modeling. Perhaps one could also find a minimal deep RACs model that reproduces the power-law decay in the mutual information between characters, which is a feature of classical English texts [47, 48]. Recently, variants of standard many-body quantum states have been analyzed as highly expressive variational ansatz to estimate probability distribution [13, 19, 49]; it'd also be interesting to implement such models for realistic NLP tasks and investigate how word embedding could help boost models prediction accuracy. Lastly, regarding the limiting entanglement spectrum that sets the maximum EE of single-layer RACs, theoretical understanding of such entanglement spectrum may hint at the minimum bias (in the bias-variance tradeoff sense) attainable by RACs to estimate a data distribution, which could provide a guideline to systematically study the expressive power of RNNs from statistical learning theory viewpoints.

Acknowledgments

This research has received funding support from the National Science, Research and Innovation Fund (NSRF) via the Program Management Unit for Human Resources & Institutional Development, Research and Innovation (Grant No. B05F640051), and from Thailand Science Research and Innovation Fund Chulalongkorn University (CU_FRB65_ind (5)_110_23_40). J Tangpanitanon, and P Bhadola are supported by Blueqat Inc. We acknowledge the National Science and Technology Development Agency, National e-Science Infrastructure Consortium, Chulalongkorn University and the Chulalongkorn Academic Advancement into its 2nd Century Project (Thailand) for providing computing infrastructure that has contributed to the research results reported within this paper (URL: www.e-science.in.th.) We also thank V Ngampruetikorn for a useful discussion, A T Rutherford and C Polpanumas for providing helpful feedbacks on the manuscript, and K Phornsiricharoenphant for providing technical supports on computational hardware used in this work.

Data availability statement

The data generated and/or analyzed during the current study are not publicly available for legal/ethical reasons but are available from the corresponding author on reasonable request.

Appendix. Sentiment analysis results for the Rotten Tomatoes movie review data set

To show that our conclusions apply to other data set, here we report similar results for sentiment analysis of the RT movie review data set. RT movie review data set is a smaller standard data set for sentiment analysis using binary classification [50]. The training set and the test set contain $M = 8400$ and 2662 different samples respectively. Both sets are balanced such that each set contains an equal number of positive and negative reviews. The length of each review is set to $T = 20$ and the dictionary size is $N = 3000$. Similar training procedure to those in the main text is applied. The batch size, however, is set to 32 for this smaller data set.

Figure A1 (left) shows the prediction accuracy and the EE as a function of the bond dimension χ for embedding dimension $d_{\mathrm{I}} = 4$. It can be seen that from bond dimension 1 to approximately 40, the training accuracy increases monotonically from 82.2% to 99.3% while the test accuracy drops from 71.8% to 69.1%. Both quantities saturate at $\chi \approx 40 \equiv \chi^*$. The increase in the training accuracy and the decrease in the test accuracy as the number of model parameters increases suggests that the model is overfitting, which can perhaps be alleviated by adding dropout though it is not clear whether RACs with dropout can be mapped to MPS. On the other hand, the EE increases rapidly before the onset of the prediction accuracy saturation at χ^*, beyond which it almost plateaus out at large χ. The results suggest a critical model size χ^* such that RACs expressiveness is maximal. Above this critical size the prediction accuracies saturate, and the EE increases very slowly or plateaus out. Similar to the IMDb data set, this critical size χ^* is valuable for identifying a minimal model that can achieve highest training accuracies for this class of model architecture.

For RT data set, the minimal model size for single-layer RACs with a fixed pre-trained embedding with $d_{\mathrm{I}} = 4$ is $\chi^* \approx 40$. We also observe similar behaviors on the saturation of prediction accuracies that correspond to the saturation of EE for larger pre-trained embeddings with $d_{\mathrm{I}} = 8, 16, 32$ with the average maximum EE of $\bar{S}_{\max} \approx 3.90, 4.14, 4.38$, respectively. For larger embedding dimensions, not only the maximum EE increases, the critical bond dimensions and the saturated prediction accuracies are also larger (not shown here due to redundancy of the plots.)

Figure A1 (right) reveals that above the critical model size χ^*, the Schmidt coefficients (indexed in a descending order) are converging towards the limiting $\bar{\lambda}_i^*$, which, similar to the IMDb data set in the main text, constrains the *slowest possible* exponential decay rate of the Schmidt coefficients. This limiting entanglement spectrum $\bar{\lambda}_i^*$ should constrain the average maximum EE according to (22) and also defines the bias (in the bias-variance tradeoff sense) in the RACs architecture for sentiment analysis modeling. Similar to the IMDb data set, we also note that the maximum EE is below the upper bound from the area law of $\log_2(\chi)$, as $\bar{S}_{\max} \approx 1.20 < 5.32 \approx \log_2(\chi = 40)$.

To analyze the interplay between RACs and the embedding layer, we now train both components simultaneously. The prediction accuracy and the EE as a function of the bond dimension is depicted in figure A2 (left). It can be seen that the training accuracy increases monotonically to saturation with a 99.10% accuracy at $\chi \approx 5$, while the test accuracy rapidly increases to 67% at $\chi = 5$ then gradually increases to saturation with a 70% accuracy at $\chi \approx 20$. On the other hand, the EE displays a peak at $\chi \approx 5$ before dropping rather steadily to saturation at $\approx 0.8\bar{S}_{\max}$ at $\chi \approx 20$, after which it fluctuates mildly around

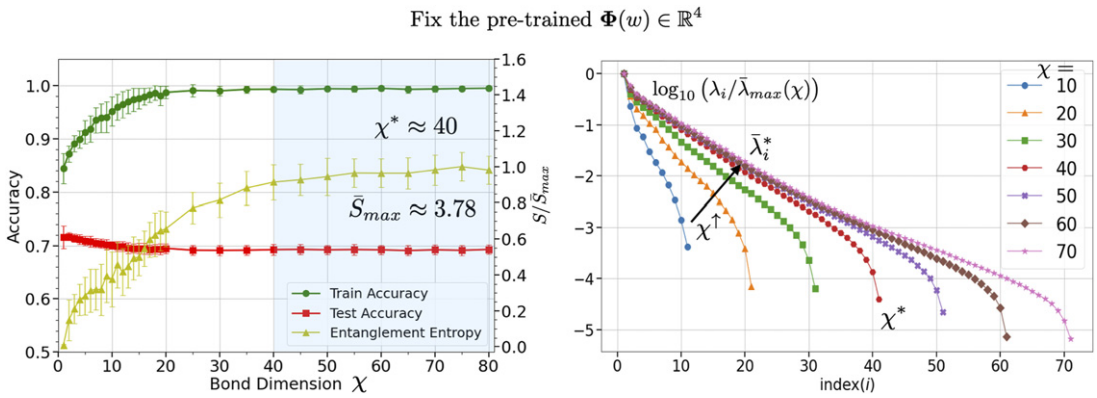

Figure A1. The behaviors of trained RACs with additive biases for sentiment analysis of the RT data set, when the fixed pre-trained word embedding has dimension $d_{\mathrm{I}} = 4$. (Left) The prediction accuracies saturate when the EE very slowly increases to saturation. The critical bond dimension $\chi^* \approx 40$, after which the EE very slowly increases and eventually saturates, enables one to infer a minimal single-layer RACs model for this data set. The average maximum EE $\bar{S}_{\max}$, which is below the upper bound from the area law, is attained when $\chi \gtrsim \chi*$. (Right) Above the critical bond dimension, the Schmidt coefficients almost collapse onto the limiting entanglement spectrum $\bar{\lambda}^*{}_i$ that sets the slowest exponential decay rate of the Schmidt coefficients. Here, the average maximum Schmidt coefficients for $\chi = 10, 20, 30, 40, 50, 60, 70$ are $\bar{\lambda}_{\max}(\chi) \approx 0.69, 0.48, 0.37, 0.30, 0.28, 0.27, 0.25$, respectively. The average are taken over 50 trained models; each begins with a random initialization of RACs with additive biases. The error bars for λ_i are not shown for the clarity of presentation.

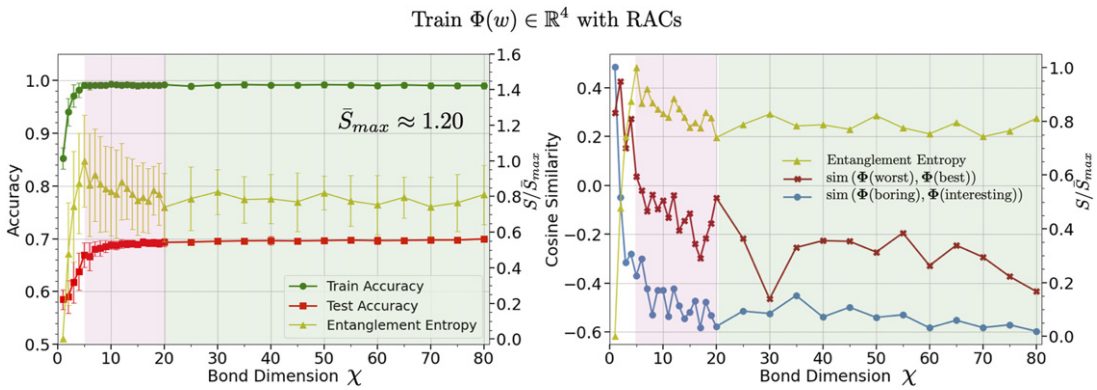

Figure A2. The behaviors of RACs with additive biases for sentiment analysis of the RT data set, when the word embedding of dimension $d_{\mathrm{I}} = 4$ is trained together with RACs. (Left) The prediction accuracies slowly increase when the EE drops down from the maximum value $\bar{S}_{\max} \approx 1.20$ at $\chi \approx 5$ to the saturated value $0.8\bar{S}_{\max}$ at $\chi \approx 20$, after which the accuracies also saturate. The saturated EE here is much smaller than that of the fixed embedding case, suggesting that the expressivity is harnessed from a more meaningful representation Φ. (Right) The cosine similarity between the two word embedding computed from (7) reveals that indeed the expressivity is boosted via meaningful word embeddings, which arise in larger single-layer RACs.

the saturated value. The test accuracy in our simple setting is comparable to the last entry in the state-of-the-art list for RT sentiment analysis [51], which achieves 76% test accuracy using graph convolutional neural network architecture [52].

Similar to IMDb data set in the main text, the EE drops after its peak value, while the prediction accuracies increase. These behaviors also arise in larger word embedding sizes, though the maximum EE at the peak are larger ($d_{\mathrm{I}} = 8, 16, 32$, $\bar{S}_{\max} \approx 1.68, 2.08, 2.51$) and occurs at a larger bond dimension for a larger model (plots are not shown here due to redundancy). The decay in the EE that corresponds to the increase in the prediction accuracies can be attributed to a more meaningful word embedding $\mathbf{\Phi}$, as shown in the cosine similarity plots between embedding vectors of the two opposite words, depicted in figure A2 (right).

ORCID iDs

Thiparat Chotibut https://orcid.org/0000-0002-8936-1413

References

[1] Verstraete F and Cirac J I 2006 *Phys. Rev.* B **73** 094423
[2] Verstraete F and Cirac I 2004 Renormalization algorithms for quantum-many body systems in two and higher dimensions (arXiv:cond-mat/0407066)
[3] Vidal G 2007 *Phys. Rev. Lett.* **99** 220405
[4] Orús R 2019 *Nat. Rev. Phys.* **1** 538–50
[5] Goodfellow I J, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)
[6] Melko R G, Carleo G, Carrasquilla J and Cirac J I 2019 *Nat. Phys.* **15** 887–92
[7] Torlai G and Melko R G 2020 *Annu. Rev. Condens. Matter Phys.* **11** 325–44
[8] Carleo G and Troyer M 2017 *Science* **355** 602–6
[9] Carrasquilla J and Melko R G 2017 *Nat. Phys.* **13** 431–4
[10] Stoudenmire E and Schwab D J 2016 Supervised learning with tensor networks *Advances in Neural Information Processing Systems* vol 29 ed D Lee, M Sugiyama, U Luxburg, I Guyon and R Garnett (Curran Associates, Inc.) https://proceedings.neurips.cc/paper/2016/file/5314b9674c86e3f9d1ba25ef9bb32895-Paper.pdf
[11] Stoudenmire E M 2018 *Quantum Sci. Technol.* **3** 034003
[12] Liu D, Ran S-J, Wittek P, Peng C, García R B, Su G and Lewenstein M 2019 *New J. Phys.* **21** 073059
[13] Glasser I, Pancotti N and Cirac J I 2020 *IEEE Access* **8** 68169–82
[14] Levine Y, Sharir O, Cohen N and Shashua A 2019 *Phys. Rev. Lett.* **112** 065301
[15] Levine Y, Yakira D, Cohen N and Shashua A 2018 Deep learning and quantum entanglement: fundamental connections with implications to network design *6th Int. Conf. on Learning Representations, ICLR 2018* (Vancouver, BC, Canada April 30–May 3, 2018) Conf. Track Proc. (OpenReview.net) https://openreview.net/forum?id=SywXXwJAb
[16] Wolf M M, Verstraete F, Hastings M B and Cirac J I 2008 *Phys. Rev. Lett.* **100** 070502
[17] Bradley T-D, Stoudenmire E M and Terilla J 2020 *Mach. Learn.: Sci. Technol.* **1** 035008
[18] Stokes J and Terilla J 2019 *Entropy* **21** 1236
[19] Miller J, Rabusseau G and Terilla J 2021 Tensor networks for probabilistic sequence modeling *Proc. of the 24th Int. Conf. on Artificial Intelligence and Statistics* (*Proc. of Machine Learning Research* vol 130) ed A Banerjee and K Fukumizu (PMLR) pp 3079–87 https://proceedings.mlr.press/v130/miller21a.html
[20] Guo C, Jie Z, Lu W and Poletti D 2018 *Phys. Rev.* E **98** 042114
[21] Zhang L, Zhang P, Ma X, Gu S, Su Z and Song D 2019 *Proc. of the AAAI Conf. on Artificial Intelligence* vol 33 pp 7450–8 https://ojs.aaai.org/index.php/AAAI/article/view/4735
[22] Jurafsky D and Martin J H 2009 *Speech and Language Processing* 2nd edn (Englewood Cliffs, NJ: Prentice-Hall)
[23] Simpson J A and Weiner E S C *The Oxford English Dictionary* 1989 2nd edn (Oxford: Oxford University Press)
[24] Mikolov T, Sutskever I, Chen K, Corrado G S and Dean J 2013 Distributed representations of words and phrases and their compositionality *Advances in Neural Information Processing Systems* vol 26 ed C J C Burges, L Bottou, M Welling, Z Ghahramani and K Q Weinberger (Curran Associates, Inc.) https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf
[25] Pennington J, Socher R and Manning C D 2014 Glove: global vectors for word representation *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing, EMNLP 2014* (*A Meeting of SIGDAT, a Special Interest Group of the ACL*) (Doha, Qatar 25–29 October 2014) ed A Moschitti, B Pang and W Daelemans (ACL) pp 1532–43
[26] Mehta P, Bukov M, Wang C-H, Day A G R, Richardson C, Fisher C K and Schwab D J 2019 *Phys. Rep.* **810** 1–124
[27] Murphy K P 2013 *Machine Learning: A Probabilistic Perspective* (Cambridge, MA: MIT Press)
[28] Bishop C M 2007 *Pattern Recognition and Machine Learning* (*Information Science and Statistics*) (New York: Springer)
[29] Peters M E, Neumann M, Iyyer M, Gardner M, Clark C, Lee K and Zettlemoyer L 2018 Deep contextualized word representations *Proc. of the 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, LA: Association for Computational Linguistics) pp 2227–37 https://aclanthology.org/N18-1202
[30] Wu Y, Zhang S, Zhang Y, Bengio Y and Salakhutdinov R R 2016 On multiplicative integration with recurrent neural networks *Advances in Neural Information Processing Systems* vol 29 ed D D Lee, M Sugiyama, U V Luxburg, I Guyon and R Garnett (Curran Associates, Inc.) pp 2856–64
[31] Sutskever I, Martens J and Hinton G E 2011 Generating text with recurrent neural networks *ICML* pp 1017–24 https://icml.cc/2011/papers/524_icmlpaper.pdf
[32] Poon H and Domingos P 2011 Sum-product networks: a new deep architecture *2011 IEEE Int. Conf. on Computer Vision Workshops (ICCV Workshops)* pp 689–90
[33] Delalleau O and Bengio Y 2011 Shallow vs deep sum-product networks *Advances in Neural Information Processing Systems* vol 24 ed J Shawe-Taylor, R Zemel, P Bartlett, F Pereira and K Q Weinberger (Curran Associates, Inc.) https://proceedings.neurips.cc/2011/file/8e6b42f1644ecb1327dc03ab345e618b-Paper.pdf
[34] Oseledets I V 2011 *SIAM J. Sci. Comput.* **33** 2295–317
[35] Ekert A and Knight P L 1995 *Am. J. Phys.* **63** 415–23
[36] Levine Y, Sharir O and Shashua A 2018 Benefits of depth for long-term memory of recurrent networks https://openreview.net/forum?id=HJ3d2Ax0-Paper.pdf
[37] Eisert J, Cramer M and Plenio M B 2010 *Rev. Mod. Phys.* **82** 277–306
[38] Maas A L, Daly R E, Pham P T, Huang D, Ng A Y and Potts C 2011 Learning word vectors for sentiment analysis *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Portland, Oregon, USA: Association for Computational Linguistics) pp 142–50 http://aclweb.org/anthology/P11-1015
[39] Chollet F *et al* 2015 *Keras* https://github.com/fchollet/keras
[40] Papers with code—IMDb benchmark (sentiment analysis) https://paperswithcode.com/sota/sentiment-analysis-on-imdb (accessed 22 March 2022)
[41] Thongtan T and Phienthrakul T 2019 Sentiment classification using document embeddings trained with cosine similarity *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop* (Florence, Italy: Association for Computational Linguistics) pp 407–14 https://aclanthology.org/P19-2057
[42] Haonan L, Huang S H, Ye T and Xiuyan G 2019 Graph star net for generalized multi-task learning (arXiv:1906.12330)

[43] Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R R and Le Q V 2019 XLNet: generalized autoregressive pretraining for language understanding *Advances in Neural Information Processing Systems* vol 32 ed H Wallach, H Larochelle, A Beygelzimer, F d Alché-Buc, E Fox and R Garnett (Curran Associates, Inc.) https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf
[44] Wang S, Fang H, Khabsa M, Mao H and Ma H 2021 Entailment as few-shot learner (arXiv:2104.14690)
[45] Sachan D S, Zaheer M and Salakhutdinov R 2019 Revisiting LSTM networks for semi-supervised text classification via mixed objective function *Proc. of the 33rd AAAI Conf. on Artificial Intelligence and 31st Innovative Applications of Artificial Intelligence Conf. and 9th AAAI Symp. on Educational Advances in Artificial Intelligence AAAI'19/IAAI'19/EAAI'19* (AAAI Press)
[46] Howard J and Ruder S 2018 Universal language model fine-tuning for text classification *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Melbourne, Australia: Association for Computational Linguistics) pp 328–39 https://aclanthology.org/P18-1031
[47] Lin H and Tegmark M 2017 *Entropy* **19** 299
[48] Lu S, Kanasz-Nagy M, Kukuljan I and Cirac I 2021 arXiv:2103.06872(quant-ph)
[49] Glasser I, Sweke R, Pancotti N, Eisert J and Cirac I 2019 Expressive power of tensor-network factorizations for probabilistic modeling *Advances in Neural Information Processing Systems* vol 32 (Curran Associates, Inc.) https://proceedings.neurips.cc/paper/2019/file/b86e8d03fe992d1b0e19656875ee557c-Paper.pdf
[50] Pang B and Lee L 2005 Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales *Proc. of the ACL*
[51] Papers with code—MR benchmark (sentiment analysis) https://paperswithcode.com/sota/sentiment-analysis-on-mr (accessed 22 March 2022)
[52] Wu F, Souza A, Zhang T, Fifty C, Yu T and Weinberger K 2019 Simplifying graph convolutional networks *Proc. of the 36th Int. Conf. on Machine Learning* (*Proc. of Machine Learning Research* vol 97) ed K Chaudhuri and R Salakhutdinov (PMLR) pp 6861–71 https://proceedings.mlr.press/v97/wu19e.html