Generalized Tensor Models for RNNNs Valentin Khrulkov, Oleksii Hrinchuk, Ivan Oseledets (2019)

Tianyu Li, Bhairav Mehta and Koustuv Sinha

IFT 6760A

March 21, 2019

Overview

Motivation

- RNNs have been widely applied in many fields
- Theoretical side of RNNs is lacking
- Natural relationship between tensor decomposition and linear neural networks
- Work with tensor instead for analysis

Why Depth?

- Shown recently that **depth** allows neural networks to express rich functions *with relatively few* parameters.
- Theory not well understood, due to difficulty of incorporating *nonlinearities* during analysis.

Basics - Data representation

- Suppose we are given a dataset of sequential structure:

$$\mathbf{X} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(T)}), \ \mathbf{x}^t \in \mathbb{R}^N$$

- Transform the dataset in a feature tensor $\Phi(\mathbf{X})$ which is an outer product of the feature vectors.

$$f_\theta(\mathbf{x}) = \sigma(A\mathbf{x} + b)$$

$$\Phi(\mathbf{X}) = f_\theta(\mathbf{x}^{(1)}) \otimes f_\theta(\mathbf{x}^{(2)}) \cdots \otimes f_\theta(\mathbf{x}^{(T)})$$

Basics - Generalized Score function

- To get an estimate (such as MLE), we can use a tensor $\boldsymbol{\mathcal{W}}$ of the same order as our feature tensor $\Phi(\mathbf{X})$
- The estimate or *score function* can be expressed as:

$$\mathcal{L}(X) = \langle \boldsymbol{\mathcal{W}}, \Phi(\mathbf{X}) \rangle = (\mathrm{vec}(\boldsymbol{\mathcal{W}}))^\top \mathrm{vec}(\Phi(\mathbf{X}))$$

Representing the core tensor

- $\mathcal{W} \in \mathcal{R}^{m \times m \times \ldots m}$ is a trainable weight tensor.
- The inner product shown in last slide is just the total sum of the entry-wise product of $\Phi(\mathbf{X})$ and $\mathcal{W}$
- Storing the full tensor $\mathcal{W}$ requires exponential amount of memory.
- We therefore use tensor decompositions to efficiently represent this weight tensor.
- Rank of the decomposition determine the complexity of the architecture.

Tensor Decomposition

- CP Decomposition:

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

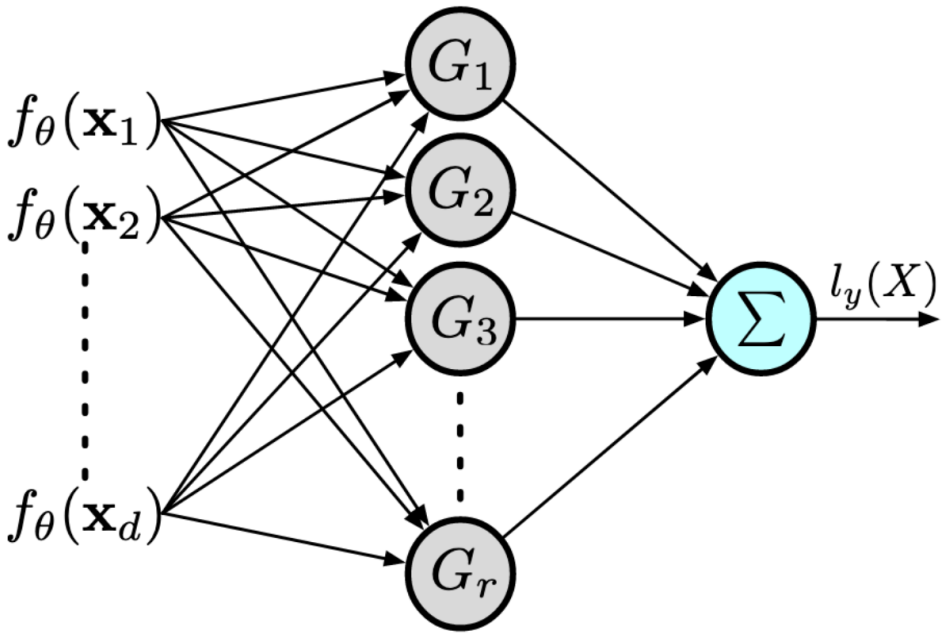
$$\mathcal{L}(X) = \sum_{r=1}^{R} \lambda_r \prod_{t=1}^{T} \langle f_\theta(\mathbf{x}^{(t)}), \mathbf{v}_r^{(t)} \rangle$$

- Tensor Train Decomposition:

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

$$\mathcal{L}(X) = \sum_{r_1=1}^{R_1} \cdots \sum_{r_{T-1}=1}^{R_{T-1}} \prod_{t=1}^{T} \langle f_\theta(\mathbf{x}^{(t)}), \mathbf{g}_{r_{T-1} r_T}^{(t)} \rangle$$

CP Decomposition and Shallow Networks

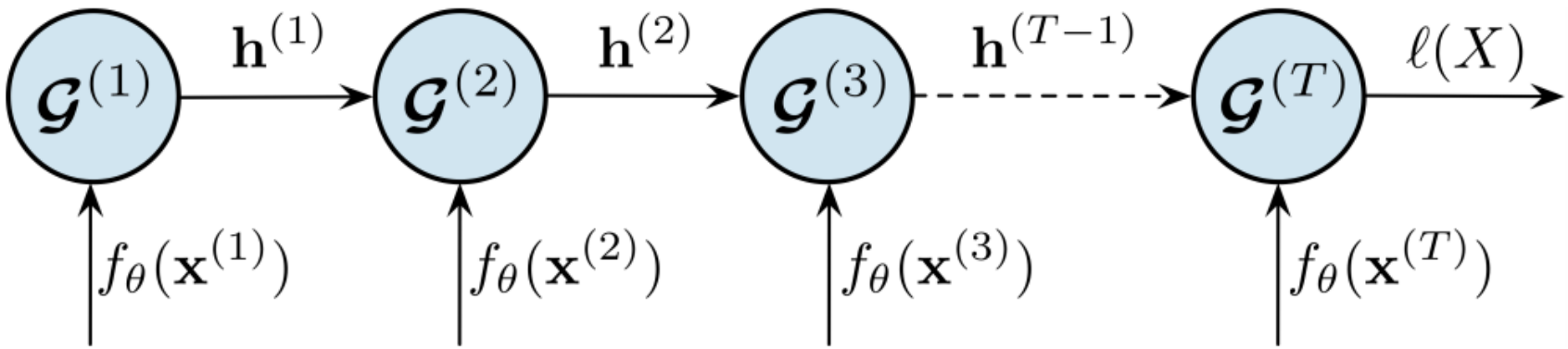


$$\mathcal{L}(X) = \sum_{r=1}^{R} \lambda_r \prod_{t=1}^{T} \langle f_\theta(\mathbf{x}^{(t)}, \mathbf{v}_r^{(t)}) \rangle$$

Tensor Trains and RNNs

- Idea: Show that TT exhibits particular recurrent structure as RNN.

$$\mathbf{h}_k^{(t)} = \sum_{i,j} \mathcal{G}_{ijk}^{(t)} f_\theta(\mathbf{x}^{(t)})_i \mathbf{h}_j^{(t-1)} = \sum_{i,j} \mathcal{G}_{ijk}^{(t)} [f_\theta(\mathbf{x}^{(t)}) \otimes \mathbf{h}^{(t-1)}]_{i,j}$$



- Combining the core tensors and weights to a single variable, we can rewrite the above equation in a general RNN formulation:

$$\mathbf{h}^{(t)} = g(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}; \Theta_{\mathcal{G}}^{(t)}), \ \mathbf{h}^{(t)} \in \mathbb{R}^{R_t}$$

Generalized Outer Product

- TTs $\rightarrow$ NNs of specific structure, *simpler* than the ones used in practice:

Only multiplicative nonlinearities allowed

- Idea: Change the nonlinearity

$$\otimes \rightarrow \otimes_\xi$$

- Generalized outer product, define ξ as an **associative** and **commutative** operator:

$$\mathcal{C} = \mathcal{A} \otimes_\xi \mathcal{B}$$

$$\mathcal{C}_{i_1 \cdots i_N j_1 \cdots j_M} = \xi(\mathcal{A}_{i_1 \cdots i_N}, \mathcal{B}_{j_1 \cdots j_M})$$

Generalized Outer Product

- Replace previous RNNs' outer product with new operator to get:

$$\xi(x, y) = \begin{cases} \max(x, y, 0) & \text{ReLU} \\ \ln(e^x + e^y) & \text{SoftPlus} \\ xy & \text{Multiplicative} \end{cases}$$

Generalized Shallow Network with ξ-nonlinearity

- Score function:

$$\mathcal{L}(\mathbf{X}) = \sum_{r=1}^{R} \lambda_r [\langle f_\theta(\mathbf{x}^{(1)}), \mathbf{v}_r^{(1)} \rangle \otimes_\xi \cdots \otimes_\xi \langle f_\theta(\mathbf{x}^{(T)}), \mathbf{v}_r^{(T)} \rangle]$$
$$= \sum_{r=1}^{R} \lambda_r \xi(\langle f_\theta(\mathbf{x}^{(1)}), \mathbf{v}_r^{(1)} \rangle, \cdots, \langle f_\theta(\mathbf{x}^{(T)}), \mathbf{v}_r^{(T)} \rangle)$$

- Parameters of the network:

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

- Can do same with RNNs to get a Generalized RNN

Great, and we are done?

- Switching $\otimes \rightarrow \otimes_\xi$ allows us to analyze more complex RNNs
- But, makes connection between RNNs and their TTs difficult to understand
- Weight tensor **no longer** exists for each and every generalized tensor network:

$$\mathcal{L}(\mathbf{X}) = \langle \mathcal{W}, \Phi(\mathbf{X}) \rangle$$

Grid Tensors

- Cohen and Shashua (2016) introduced **grid tensors**:
  M **fixed** vectors $\mathbb{X}$ (templates) $\rightarrow$ GT of order T and dimension M in each mode:

$$\Gamma^{\mathcal{L}}(\mathbb{X})_{i_1,i_2,\cdots,i_T} = \mathcal{L}(\mathbf{X}), \quad \mathbf{X} = (\mathbf{x}^{(i_1)}, \mathbf{x}^{(i_2)}, \cdots, \mathbf{x}^{(i_T)})$$

- Evaluate score function on every possible input combination of the template vectors, instead of **all** possible input sequences.

Grid Tensors

- Define a **feature matrix** $\mathbf{F} \in \mathbb{R}^{M \times M}$
- Run representation function $f_\theta : \mathbb{R}^N \to \mathbb{R}^M$ on each $\mathbf{x}^{(t)} \in \mathbb{X}$:

$$\mathbf{F} = [f_\theta(\mathbf{x}^{(1)}), f_\theta(\mathbf{x}^{(2)}), \cdots, f_\theta(\mathbf{x}^{(M)})]$$

- Each generalized tensor network has a corresponding grid tensor (shown: generalized shallow network)

$$\Gamma^{\mathcal{L}}(\mathbb{X}) = \sum_{r=1}^{R} \lambda_r (\mathbf{F}\mathbf{v}_r^{(1)}) \otimes_\xi (\mathbf{F}\mathbf{v}_r^{(2)}) \otimes_\xi \cdots \otimes_\xi (\mathbf{F}\mathbf{v}_r^{(T)})$$

Overview of the main results

Two problems need to be considered:

- Universality
  Can every tensor realizes a (generalized) shallow network/RNN ?
- Expressivity
  To represent the same function, which model uses less parameters?

Universality

- Regular case (linear outer product): Holds automatically

$$\mathcal{L}(\mathbf{X}) = \langle \mathcal{W}, \Phi(\mathbf{X}) \rangle$$

- Generalized case (Non-linear outer product): Can no longer work with $\mathcal{W}$. Instead, work with the grid tensor:

$$\Gamma^{\mathcal{L}}(\mathbb{X})_{i_1, i_2, \cdots, i_T} = \mathcal{L}(\mathbf{X}), \quad \mathbf{X} = (X^{(i_1)}, X^{(i_2)}, \ldots, X^{(i_T)})$$

Universality

Theorem 1

Given an arbitrary tensor $\mathcal{H} \in \mathbb{R}^{M \times M \times \cdots \times M}$ and a template $\mathbb{X}$, let the grid tensors for a:

- Generalized[a] shallow network $\tilde{\mathcal{S}}$ be: $\Gamma^{\mathcal{S}}(\mathbb{X})$
- Generalized[a] RNN $\tilde{\mathcal{G}}$ be: $\Gamma^{\mathcal{G}}(\mathbb{X})$

Then we can find $\tilde{\mathcal{S}}$ and $\tilde{\mathcal{G}}$ such that:

$$\mathcal{H} = \Gamma^{\mathcal{S}}(\mathbb{X}) = \Gamma^{\mathcal{G}}(\mathbb{X})$$

[a] All the results are based on rectifier nonlinearity

Expressivity

- Goal: compare models' representation ability in terms of their parameters
- Linear case: simply compare the rank of the tensor $\mathcal{W}$
- Generalized case: compare in terms of the grid tensor $\Gamma^{\mathcal{L}}(\mathbb{X})$

Expressivity

Theorem 2

Given a generalized RNN of rank at most R and its grid tensor $\Gamma^{\mathcal{G}}(\mathbb{X})$, its realization of generalized shallow network can be written as:

$$\Gamma^{\mathcal{G}}(\mathbb{X}) = \Gamma^{\mathcal{S}}(\mathbb{X}) = \sum_{r=1}^{\hat{R}} \lambda_r (\mathbf{F}\mathbf{v}_r^{(1)}) \otimes_\xi (\mathbf{F}\mathbf{v}_r^{(2)}) \otimes_\xi \cdots \otimes_\xi (\mathbf{F}\mathbf{v}_r^{(T)})$$

There exists $\tilde{\mathcal{G}}_1$, such that $\hat{R} \geq \frac{2}{MT} \min(M, R)^{T/2}$;

Expressivity

Theorem 3

Given a generalized RNN of rank R and its grid tensor $\Gamma^{\mathcal{G}}(\mathbb{X})$, its realization of generalized shallow network can be written as:

$$\Gamma^{\mathcal{G}}(\mathbb{X}) = \Gamma^{\mathcal{S}}(\mathbb{X}) = \sum_{r=1}^{\hat{R}} \lambda_r (\mathbf{F}\mathbf{v}_r^{(1)}) \otimes_\xi (\mathbf{F}\mathbf{v}_r^{(2)}) \otimes_\xi \cdots \otimes_\xi (\mathbf{F}\mathbf{v}_r^{(T)})$$

There exists $\tilde{\mathcal{G}}_2$, such that $\hat{R} = 1$

Experiment on IMDB sentiment analysis

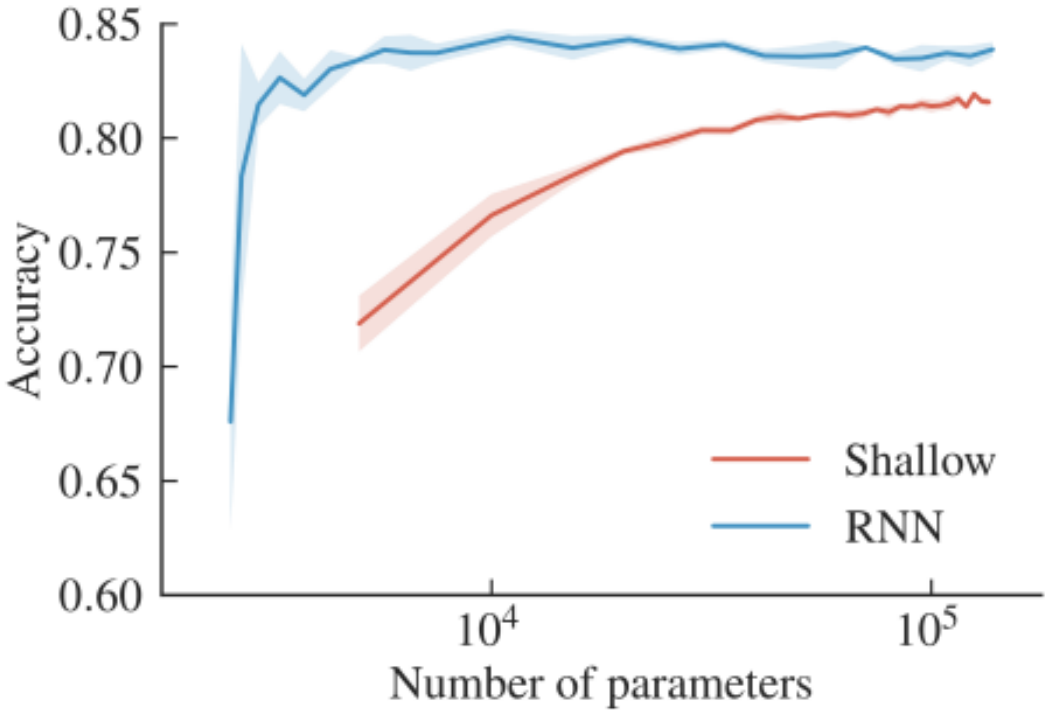


Figure 2: Test accuracy on IMDB dataset for generalized RNNs and generalized shallow networks with respect to the total number of parameters ($M = 50, T = 100, \xi(x, y) = \max(x, y, 0)$).

Experiment on Synthetic Data

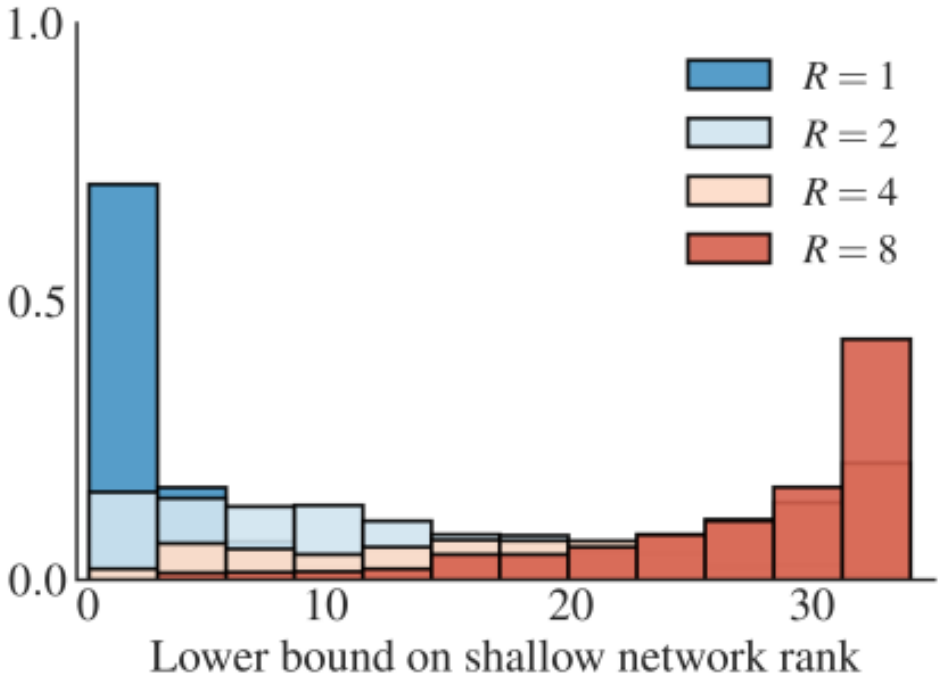


Figure 3: Distribution of lower bounds on the rank of generalized shallow networks equivalent to randomly generated generalized RNNs of ranks $1, 2, 4, 8$ ($M = 10, T = 6$).

Conclusion

- Draw links between RNNs and TT decomposition
- Introduce nontrivial nonlinearity into tensor framework
- Provide theoretical analysis on universality and expressivity under rectifier nonlinearity
- Extend this to LSTM and attention? Other nonlinearities?

Thank You