Upper and Lower Bounds on the VC-Dimension of Tensor Network Models

Behnoush Khavari and *Guillaume Rabusseau*

Tensors: Quantum Information, Complexity and Combinatorics
CRM seminar series

Outline

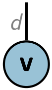
1. Tensor Networks and Tensor Decompositions
2. Learning with Tensor Networks
3. VC-dimension of TN-based Classifiers

Joint work with Behnoush Khavari

Tensor Networks and Tensor Decompositions

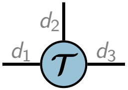
Tensor Networks

$$\boxed{\text{Degree of a node} \equiv \text{order of tensor}}$$



$\mathbf{v} \in \mathbb{R}^{d}$

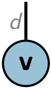
$\mathbf{M} \in \mathbb{R}^{m \times n}$



$\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$

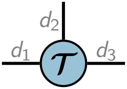
Tensor Networks

$\boxed{\text{Degree of a node} \equiv \text{order of tensor}}$



$\mathbf{v} \in \mathbb{R}^d$

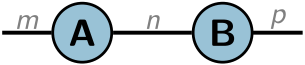
$\mathbf{M} \in \mathbb{R}^{m \times n}$



$\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$

$\boxed{\text{Edge} \equiv \text{contraction}}$

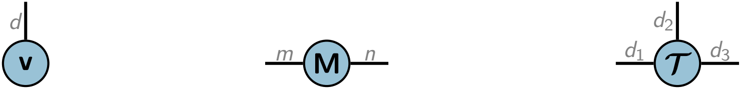
Matrix product:



$$(\mathbf{AB})_{i_1,i_2} = \sum_{k=1}^{n} \mathbf{A}_{i_1 k} \mathbf{B}_{k i_2}$$

Tensor Networks

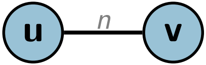
$\boxed{\text{Degree of a node} \equiv \text{order of tensor}}$



$\mathbf{v} \in \mathbb{R}^d$ $\mathbf{M} \in \mathbb{R}^{m \times n}$ $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$

$\boxed{\text{Edge} \equiv \text{contraction}}$

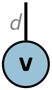
Inner product:



$$\mathbf{u}^\top \mathbf{v} = \sum_{k=1}^{n} \mathbf{u}_k \mathbf{v}_k$$

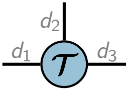
Tensor Networks

$\boxed{\text{Degree of a node} \equiv \text{order of tensor}}$



$\mathbf{v} \in \mathbb{R}^d$

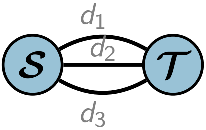
$\mathbf{M} \in \mathbb{R}^{m \times n}$



$\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$

$\boxed{\text{Edge} \equiv \text{contraction}}$

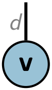
Inner product between tensors:



$$\langle \mathcal{S}, \mathcal{V} \rangle = \sum_{i_1=1}^{d_1} \sum_{i_2=1}^{d_2} \sum_{i_3=1}^{d_3} \mathcal{S}_{i_1 i_2 i_3} \mathcal{T}_{i_1 i_2 i_3}$$

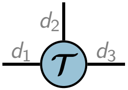
Tensor Networks

$\boxed{\text{Degree of a node} \equiv \text{order of tensor}}$



$\mathbf{v} \in \mathbb{R}^d$

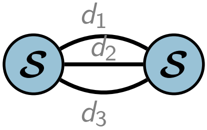
$\mathbf{M} \in \mathbb{R}^{m \times n}$



$\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$

$\boxed{\text{Edge} \equiv \text{contraction}}$

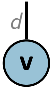
Frobenius norm of a tensor:



$$\|\mathcal{S}\|_F^2 = \sum_{i_1=1}^{d_1} \sum_{i_2=1}^{d_2} \sum_{i_3=1}^{d_3} (\mathcal{S}_{i_1 i_2 i_3})^2$$

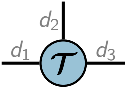
Tensor Networks

$\boxed{\text{Degree of a node} \equiv \text{order of tensor}}$



$\mathbf{v} \in \mathbb{R}^d$

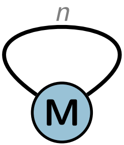
$\mathbf{M} \in \mathbb{R}^{m \times n}$



$\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$

$\boxed{\text{Edge} \equiv \text{contraction}}$

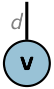
Trace of an $n \times n$ matrix:



$\text{Tr}(\mathbf{M}) = \sum_{i=1}^{n} \mathbf{M}_{ii}$

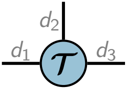
Tensor Networks

$\boxed{\text{Degree of a node} \equiv \text{order of tensor}}$



$\mathbf{v} \in \mathbb{R}^d$

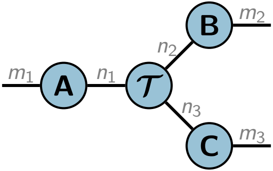
$\mathbf{M} \in \mathbb{R}^{m \times n}$



$\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$

$\boxed{\text{Edge} \equiv \text{contraction}}$

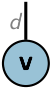
Tensor times matrices:



$$(\mathcal{T} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C})_{i_1,i_2,i_3} = \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \sum_{k_3=1}^{n_3} \mathcal{T}_{k_1 k_2 k_3} \mathbf{A}_{i_1 k_1} \mathbf{B}_{i_2 k_2} \mathbf{C}_{i_3 k_3}$$

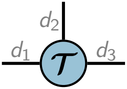
Tensor Networks

$\boxed{\text{Degree of a node} \equiv \text{order of tensor}}$



$\mathbf{v} \in \mathbb{R}^d$

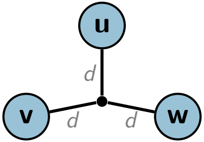
$\mathbf{M} \in \mathbb{R}^{m \times n}$



$\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$

$\boxed{\text{Edge} \equiv \text{contraction}}$

Hyperedge $\equiv$ contraction between more than 2 indices:

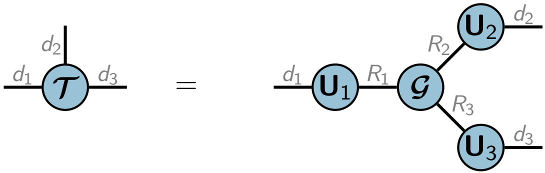


$$= \sum_{i=1}^{d} \mathbf{u}_i \mathbf{v}_i \mathbf{w}_i$$

$= \sum_{i=1}^{d} \mathbf{e}_i \otimes \mathbf{e}_i \otimes \mathbf{e}_i$ is the identity tensor (a.k.a. copy/spider tensor).

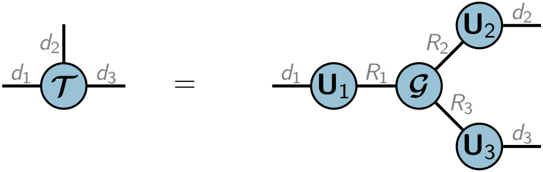
Tensor Decompositions

- Tucker decomposition [Tucker, 1966]:



Tensor Decompositions

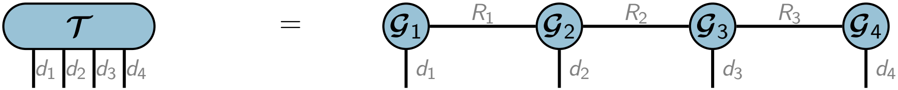
- Tucker decomposition [Tucker, 1966]:



$\hookrightarrow$ $R_1R_2R_3 + d_1R_1 + d_2R_2 + d_2R_2$ parameters instead of $d_1d_2d_3$.

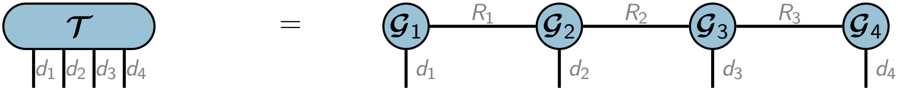
Tensor Decompositions

- Tensor Train decomposition [Oseledets, 2011]:



Tensor Decompositions

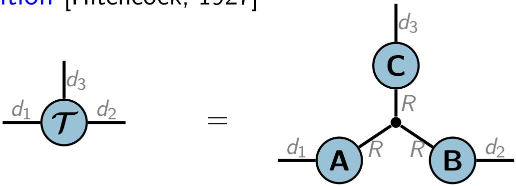
- Tensor Train decomposition [Oseledets, 2011]:



$\hookrightarrow$ $d_1 R_1 + R_1 d_2 R_2 + R_2 d_2 R_3 + R_3 d_4$ parameters instead of $d_1 d_2 d_3 d_4$.

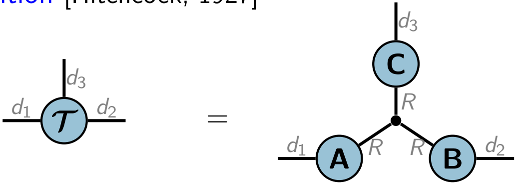
Tensor Decompositions

- CP decomposition [Hitchcock, 1927]



Tensor Decompositions

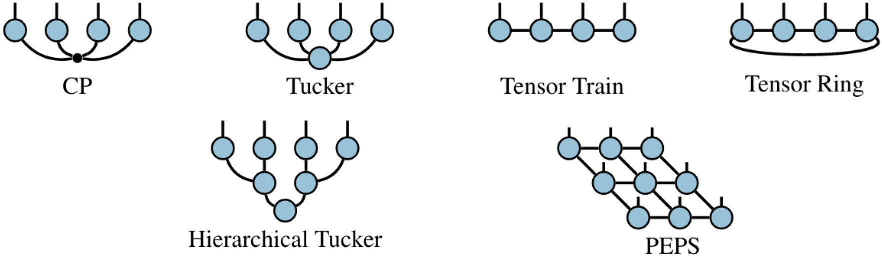
- CP decomposition [Hitchcock, 1927]



$\hookrightarrow$ $R(d_1 + d_2 + d_3)$ parameters instead of $d_1 d_2 d_3$.

Tensor Decompositions

- Lots of ways to decompose a tensor



Learning with Tensor Networks

Machine Learning Problems

Binary classification

Learn $f : \mathcal{X} \rightarrow \{-, +\}$ from a sample $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$

Machine Learning Problems

Binary classification

Learn $f : \mathcal{X} \to \{-,+\}$ from a sample $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$

Regression

Learn $f : \mathcal{X} \to \mathbb{R}$ from a sample $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$

Machine Learning Problems

Binary classification

Learn $f : \mathcal{X} \to \{-, +\}$ from a sample $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$

Regression

Learn $f : \mathcal{X} \to \mathbb{R}$ from a sample $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$

Completion

Learn a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_p}$ from a set of observed entries $D = \left\{\left((i_1, \cdots, i_p), \boldsymbol{\mathcal{X}}_{i_1, \cdots, i_p}\right) \mid (i_1, \cdots, i_p) \in \Omega\right\}$ where $\Omega \subset [d_1] \times \cdots \times [d_p]$

Machine Learning Problems

Binary classification

Learn $f : \mathcal{X} \to \{-, +\}$ from a sample $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$

Regression

Learn $f : \mathcal{X} \to \mathbb{R}$ from a sample $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$

Completion

Learn a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_p}$ from a set of observed entries $D = \{((i_1, \cdots, i_p), \boldsymbol{\mathcal{X}}_{i_1, \cdots, i_p}) \mid (i_1, \cdots, i_p) \in \Omega\}$ where $\Omega \subset [d_1] \times \cdots \times [d_p]$

$\hookrightarrow$ Completion $\simeq$ Regression: Learn $f : \mathcal{X} \to \mathbb{R}$ from a sample $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ where each $x_i \in [d_1] \times \cdots \times [d_p]$ and each $y_i \in \mathbb{R}$

Statistical Framework of Learning

- Supervised learning:
  - sample $S = \{(x_1, y_1), \cdots, (x_n, y_n)\} \overset{i.i.d.}{\sim} D$
  - hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$
  - loss $\ell : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$,

Statistical Framework of Learning

- Supervised learning:
  - sample $S = \{(x_1, y_1), \cdots, (x_n, y_n)\} \overset{i.i.d.}{\sim} D$
  - hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$
  - loss $\ell : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$,

Goal: find $h \in \mathcal{H}$ minimizing the risk

$$R(h) = \mathop{\mathbb{E}}_{(x,y)\sim D} \ell(h(x), y)$$

Statistical Framework of Learning

- Supervised learning:
  - sample $S = \{(x_1, y_1), \cdots, (x_n, y_n)\} \overset{i.i.d.}{\sim} D$
  - hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$
  - loss $\ell : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$,

  Goal: find $h \in \mathcal{H}$ minimizing the risk

$$R(h) = \mathop{\mathbb{E}}_{(x,y)\sim D} \ell(h(x), y)$$

- Unknown $D \to$ Empirical Risk Minimization:

$$\hat{R}_S(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i)$$

Statistical Framework of Learning

- Supervised learning:
  - sample $S = \{(x_1, y_1), \cdots, (x_n, y_n)\} \overset{i.i.d.}{\sim} D$
  - hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$
  - loss $\ell : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$,

  Goal: find $h \in \mathcal{H}$ minimizing the risk

$$R(h) = \underset{(x,y)\sim D}{\mathbb{E}} \ell(h(x), y)$$

- Unknown $D \to$ Empirical Risk Minimization:

$$\hat{R}_S(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i)$$

- Generalization gap $|R(h) - \hat{R}_S(h)|$ depends on:
  - sample size n (size of S)
  - complexity of $\mathcal{H}$.

Statistical Framework of Learning

- Generalization gap $|R(h) - \hat{R}_S(h)|$ depends on:
  - sample size n (size of S)
  - complexity of $\mathcal{H}$.

Finite $\mathcal{H}$ With probability at least $1 - \delta$, for all $h \in \mathcal{H}$

$$R(h) \leq \hat{R}_S(h) + \sqrt{\frac{\log |2\mathcal{H}| - \log \delta}{2n}}$$

Statistical Framework of Learning

- Generalization gap $|R(h) - \hat{R}_S(h)|$ depends on:
  - sample size n (size of S)
  - complexity of $\mathcal{H}$.

Finite $\mathcal{H}$ With probability at least $1-\delta$, for all $h \in \mathcal{H}$

$$R(h) \leq \hat{R}_S(h) + \sqrt{\frac{\log|2\mathcal{H}| - \log\delta}{2n}}$$

Infinite $\mathcal{H}$ With probability at least $1-\delta$, for all $h \in \mathcal{H}$

$$R(h) < \hat{R}_S(h) + 2\sqrt{\frac{2}{n}\left(d_{VC}\log\frac{2ne}{d_{VC}} + \log\frac{4}{\delta}\right)}$$

where d_{VC} is the VC dimension of $\mathcal{H}$.

VC dimension

VC dimension: maximum number of points that hypotheses in $\mathcal{H}$ can separate in all 2^n ways. Formally:

Definition

Let $\mathcal{H} \subset \{-1, +1\}^{\mathcal{X}}$ be a hypothesis class. The *VC-dimension* of $\mathcal{H}$, $d_{VC}(\mathcal{H})$, is the largest number of points $x_1, \cdots, x_n$ shattered by $\mathcal{H}$, i.e., for which $|\{(h(x_1), \ldots, h(x_n)) \mid h \in \mathcal{H}\}| = 2^n$.

VC dimension

VC dimension: maximum number of points that hypotheses in $\mathcal{H}$ can separate in all 2^n ways. Formally:

Definition

Let $\mathcal{H} \subset \{-1,+1\}^{\mathcal{X}}$ be a hypothesis class. The *VC-dimension* of $\mathcal{H}$, $d_{VC}(\mathcal{H})$, is the largest number of points $x_1, \cdots, x_n$ shattered by $\mathcal{H}$, i.e., for which $|\{(h(x_1), \ldots, h(x_n)) \mid h \in \mathcal{H}\}| = 2^n$.

Example:

- The VC dimension of linear classifiers in d dimensions is $d+1$:

$$d_{VC}\left(\{\mathbf{x} \mapsto \operatorname{sign} \mathbf{w}^\top \mathbf{x} + b \mid \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}\}\right) = d+1$$

VC dimension

VC dimension: maximum number of points that hypotheses in $\mathcal{H}$ can separate in all 2^n ways. Formally:

Definition

Let $\mathcal{H} \subset \{-1,+1\}^{\mathcal{X}}$ be a hypothesis class. The *VC-dimension* of $\mathcal{H}$, $d_{VC}(\mathcal{H})$, is the largest number of points $x_1, \cdots, x_n$ shattered by $\mathcal{H}$, i.e., for which $|\{(h(x_1), \ldots, h(x_n)) \mid h \in \mathcal{H}\}| = 2^n$.

For regression (and completion) tasks, this capacity measure can be extended to the one of pseudo-dimension:

For any $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$, $\quad \mathrm{Pdim}(\mathcal{H}) = d_{VC}\left(\{(x,t) \mapsto \mathrm{sign}(h(x) - t) \mid h \in \mathcal{H}\}\right)$

Tensor Networks in Machine Learning

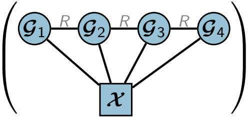
Idea: Tensor network parameterization of linear models in high-dimensional spaces.

Tensor Networks in Machine Learning

Idea: Tensor network parameterization of linear models in high-dimensional spaces.

- Classification with TT weight [Stoudenmire & Schwab, 2016]:

$$f(\boldsymbol{\mathcal{X}}) = \mathrm{sign}$$

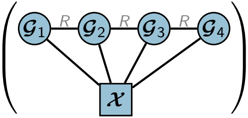


Tensor Networks in Machine Learning

Idea: Tensor network parameterization of linear models in high-dimensional spaces.

- Classification with TT weight [Stoudenmire & Schwab, 2016]:

$$f(\boldsymbol{\mathcal{X}}) = \mathrm{sign}$$



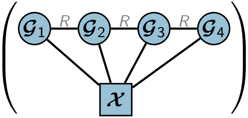
Goal: Study the VC dimension of tensor network ML models.

Tensor Networks in Machine Learning

Idea: Tensor network parameterization of linear models in high-dimensional spaces.

- Classification with TT weight [Stoudenmire & Schwab, 2016]:

$$f(\mathcal{X}) = \mathrm{sign}$$



Goal: Study the VC dimension of tensor network ML models.

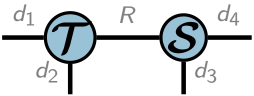
Mathematical open problems in Projected Entangled Pair States

J. Ignacio Cirac · José Garre-Rubio · David Pérez-García

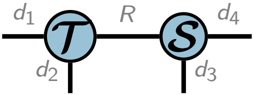
Machine Learning. MPS (and other Tensor Networks such as MERA) have been successfully used numerically in the context of Supervised Machine Learning (ML) [47]. They lack however an in-depth theoretical analysis. A concrete (relevant) question is the following:

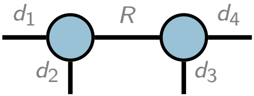
Question 13 Can one write the Rademacher complexity or the Vapnik-Chervonenkis (VC)-dimension for such ML algorithms as a function of the bond dimension?

TN Structure



TN Structure



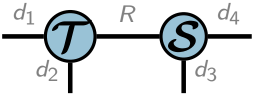


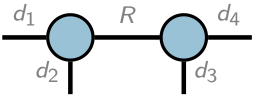
Graph structure $G = (V, E, \text{dim})$

$\mathcal{S} \in \mathbb{R}^{R \times d_3 \times d_4}$
$\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times R}$

Tensor parameters

TN Structure





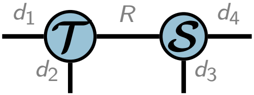
$\mathcal{S} \in \mathbb{R}^{R \times d_3 \times d_4}$
$\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times R}$

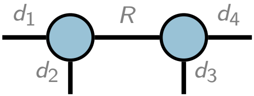
Graph structure $G = (V, E, \mathrm{dim})$

- V: vertices
- E: edges
- $\mathrm{dim} : E \to \mathbb{N}$

Tensor parameters

TN Structure



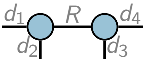


Graph structure $G = (V, E, \text{dim})$

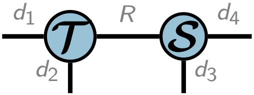
- V: vertices
- E: edges
- $\text{dim} : E \to \mathbb{N}$

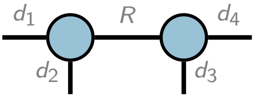
$$\boldsymbol{\mathcal{S}} \in \mathbb{R}^{R \times d_3 \times d_4}$$
$$\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times d_2 \times R}$$

Tensor parameters

$TN\Big($  $, \{\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}}\}\Big) \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times d_4}$

TN Structure





$\mathcal{S} \in \mathbb{R}^{R \times d_3 \times d_4}$
$\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times R}$

Graph structure $G = (V, E, \dim)$

Tensor parameters

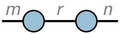
- V: vertices
- E: edges
- $\dim : E \to \mathbb{N}$

$$\mathcal{T}(G) = \left\{ TN\left(G, \{\mathcal{T}^v\}_{v \in V}\right) \ : \mathcal{T}^v \in \bigotimes_{e \in E_v} \mathbb{R}^{dim(e)}, v \in V \right\}$$

Tensor Structure Examples

- Low-rank matrices

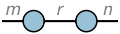
$$\{\mathbf{M} \in \mathbb{R}^{m \times n} \mid \operatorname{rank}(\mathbf{M}) \leq r\} = \mathcal{T}\left(\quad \right)$$



Tensor Structure Examples

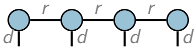
- Low-rank matrices

$$\{\mathbf{M} \in \mathbb{R}^{m \times n} \mid \operatorname{rank}(\mathbf{M}) \leq r\} = \boldsymbol{\mathcal{T}}\left(\quad \right)$$



- Tensor train tensors

$$\left\{\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d \times d \times d \times d} \mid \operatorname{rank}_{TT}(\boldsymbol{\mathcal{T}}) \leq (r, r, r)\right\} = \boldsymbol{\mathcal{T}}\left(\quad \right)$$



TN Hypothesis classes

For a fixed graph structure G, we can define hypotheses classes corresponding to linear models with weights paremeterized by tensor networks with structure G:

$$\mathcal{H}_G^{\mathsf{classif}} = \{h : \boldsymbol{\mathcal{X}} \mapsto \mathrm{sign}(\langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}} \rangle) \ \mid \boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{T}}(G)\}$$

TN Hypothesis classes

For a fixed graph structure G, we can define hypotheses classes corresponding to linear models with weights paremeterized by tensor networks with structure G:

$$\mathcal{H}_G^{\text{classif}} = \{h : \boldsymbol{\mathcal{X}} \mapsto \operatorname{sign}(\langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}} \rangle) \ \mid \boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{T}}(G)\}$$

$$\mathcal{H}_G^{\text{regression}} = \{h : \boldsymbol{\mathcal{X}} \mapsto \langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}} \rangle \mid \boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{T}}(G)\}$$

TN Hypothesis classes

For a fixed graph structure G, we can define hypotheses classes corresponding to linear models with weights paremeterized by tensor networks with structure G:

$$\mathcal{H}_G^{\text{classif}} = \{h : \boldsymbol{\mathcal{X}} \mapsto \operatorname{sign}(\langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}} \rangle) \;\mid \boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{T}}(G)\}$$

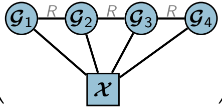
$$\mathcal{H}_G^{\text{regression}} = \{h : \boldsymbol{\mathcal{X}} \mapsto \langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}} \rangle \mid \boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{T}}(G)\}$$

$$\mathcal{H}_G^{\text{completion}} = \{h : (i_1, \cdots, i_p) \mapsto \boldsymbol{\mathcal{W}}_{i_1,\cdots,i_p} \mid \boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{T}}(G)\}$$

Tensor Networks in Machine Learning

- Classification with TT weight [Stoudenmire & Schwab, 2016]:

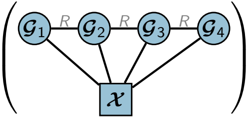
$$f(\boldsymbol{\mathcal{X}}) = \mathrm{sign}\left(\langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}} \rangle\right) = \mathrm{sign}\left(\right)$$



Tensor Networks in Machine Learning

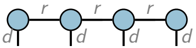
- Classification with TT weight [Stoudenmire & Schwab, 2016]:

$$f(\boldsymbol{\mathcal{X}}) = \mathrm{sign}\left(\langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}} \rangle\right) = \mathrm{sign}$$



- Class of hypotheses:

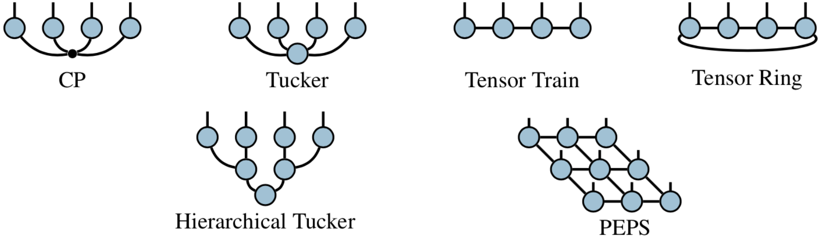
$$\mathcal{H}_G = \left\{ h : \boldsymbol{\mathcal{X}} \mapsto \mathrm{sign}(\langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}} \rangle) \mid \boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{T}}(\quad) \right\}$$



VC-dimension of TN-based Classifiers

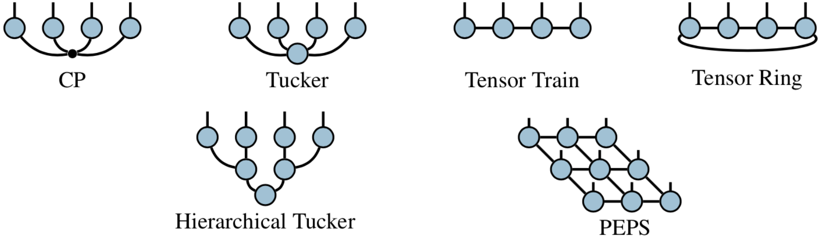
TN-based Classification Problem

- Classifier with TN-weight



TN-based Classification Problem

- Classifier with TN-weight



- Goal: For any arbitrary graph structure G, derive lower and upper bounds on $d_{VC}(\mathcal{H}_G)$, where

$$\mathcal{H}_G = \{h : \boldsymbol{\mathcal{X}} \mapsto \mathrm{sign}(\langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}} \rangle) \ \mid \boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{T}}(G)\}$$

Upper-bound on the VC-dimension

Theorem

Let $G = (V, E, \dim)$ be a tensor network structure. Then, for the corresponding hypothesis class $\mathcal{H}_G$:

$$d_{VC}(\mathcal{H}_G) \leq 2N_G \log(12|V|)$$

- N_G: # parameters of TN

Upper-bound on the VC-dimension

Theorem

Let $G = (V, E, \dim)$ be a tensor network structure. Then, for the corresponding hypothesis class $\mathcal{H}_G$:

$$d_{VC}(\mathcal{H}_G) \leq 2N_G \log(12|V|)$$

- N_G: # parameters of TN

$\hookrightarrow$ For any $\delta > 0$, with probability at least $1 - \delta$, $\forall h \in \mathcal{H}_G$

$$R(h) < \hat{R}_S(h) + 2\sqrt{\frac{2}{n}\left(N_G \log \frac{8en|V|}{N_G} + \log \frac{4}{\delta}\right)}.$$

Upper-bound on the VC-dimension

Theorem

Let $G = (V, E, \dim)$ be a tensor network structure. Then,

$$d_{VC}(\mathcal{H}_G) \leq 2N_G \log(12|V|)$$

Proof. Warren's theorem (1968):

Theorem

The number of sign patterns of n real polynomials, each of degree at most v, over N variables is at most $\left(\frac{4evn}{N}\right)^N$ for all $n > N > 2$ (where e is Euler's number).

$d_{VC}(\mathcal{H}_G)$ is the largest n for which $|\{(h(\boldsymbol{\mathcal{X}}_1), \ldots, h(\boldsymbol{\mathcal{X}}_n)) \mid h \in \mathcal{H}_G\}| = 2^n$

Upper-bound on the VC-dimension

Theorem

Let $G = (V, E, \dim)$ be a tensor network structure. Then,

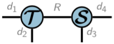
$$d_{VC}(\mathcal{H}_G) \leq 2N_G \log(12|V|)$$

Proof. Warren's theorem (1968):

Theorem

The number of sign patterns of n real polynomials, each of degree at most v, over N variables is at most $\left(\frac{4evn}{N}\right)^N$ for all $n > N > 2$ (where e is Euler's number).

$d_{VC}(\mathcal{H}_G)$ is the largest n for which $|\{(h(\boldsymbol{\mathcal{X}}_1), \ldots, h(\boldsymbol{\mathcal{X}}_n)) \mid h \in \mathcal{H}_G\}| = 2^n$

$h(\boldsymbol{\mathcal{X}}) = \operatorname{sign}(\langle$ 

$, \boldsymbol{\mathcal{X}}\rangle)$: polyn. in $N_G = (d_1 d_2 + d_3 d_4)R$ var. of deg. $|V| = 2$...

Upper-bound on the VC-dimension

Theorem

Let $G = (V, E, \dim)$ be a tensor network structure. Then, for the corresponding hypothesis class $\mathcal{H}_G$:

$$d_{VC}(\mathcal{H}_G) \leq 2N_G \log(12|V|)$$

Proof: Warren's theorem (1968);

Similar technique used for:

- $\hookrightarrow$ Matrix completion [Srebro et al., 2005]
- $\hookrightarrow$ Tucker completion / regression [Nickel et al., 2013 / Rabusseau and Kadri, 2016]

Special cases

- $d_1 \times d_2$ matrices of rank $\leq r$:

$$d_{VC}(\mathcal{H}_{G_{\mathrm{mat}}}(r)) \leq 10r(d_1 + d_2)$$

  - Previous bounds:

    $\hookrightarrow$ $r(d_1 + d_2)\log(r(d_1 + d_2))$ classification [Wolf et al., 2007]

    $\hookrightarrow$ $r(d_1 + d_2)\log\frac{16ed_1}{r}$ completion [Srebro et al., 2005]

Special cases

- $d_1 \times d_2$ matrices of rank $\leq r$:

$$d_{VC}(\mathcal{H}_{G_{\mathrm{mat}}}(r)) \leq 10r(d_1 + d_2)$$

  - Previous bounds:
    - $\hookrightarrow$ $r(d_1 + d_2)\log(r(d_1 + d_2))$ classification [Wolf et al., 2007]
    - $\hookrightarrow$ $r(d_1 + d_2)\log\frac{16ed_1}{r}$ completion [Srebro et al., 2005]

- Tensors of TT-rank $\leq r$:

$$d_{VC}(\mathcal{H}_{G_{\mathrm{TT}}}(r)) \leq dpr^2\log(p)$$

Special cases

- $d_1 \times d_2$ matrices of rank $\leq r$:

$$d_{VC}(\mathcal{H}_{G_{\mathrm{mat}}}(r)) \leq 10r(d_1 + d_2)$$

  - Previous bounds:
    $\hookrightarrow$ $r(d_1 + d_2)\log(r(d_1 + d_2))$ classification [Wolf et al., 2007]
    $\hookrightarrow$ $r(d_1 + d_2)\log\frac{16ed_1}{r}$ completion [Srebro et al., 2005]

- Tensors of TT-rank $\leq r$:

$$d_{VC}(\mathcal{H}_{G_{\mathrm{TT}}}(r)) \leq dpr^2\log(p)$$

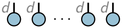
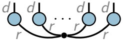
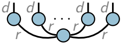
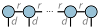
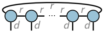
$\hookrightarrow$ applies to TT model introduced in [Stoudenmire et al., 2016]

$\hookrightarrow$ answers the open problem listed in [Cirac et al., 2019]

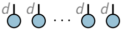
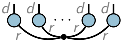
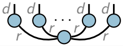
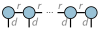
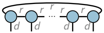
Lower-bounds

Theorem

The following lower bounds hold for the VC/pseudo-dimension of hypothesis classes for classification, completion and regression.

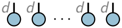
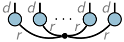
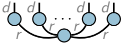
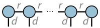
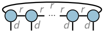
	rank one	CP	Tucker	TT / TR
Decomposition				 / 
Lower Bound (condition)	$(d-1)p$	$rd \quad (r \leq d^{p-1})$	$r^p \quad (r \leq d)$	$r^2 d \quad (r \leq d^{\lfloor \frac{p-1}{2} \rfloor}, p \geq 3)$ $\frac{p(r^2 d - 1)}{3} \quad (r = d, p/3 \in \mathbb{N})$
Upper bound	$2dp\log(12p)$	$2prd\log(12p)$	$2(r^p + prd)\log(24p)$	$2pr^2 d\log(12p)$

Lower-bounds

Decomposition	rank one 	CP 	Tucker 	TT / TR  / 
Lower Bound (condition)	$(d-1)p$	$rd \quad (r \leq d^{p-1})$	$r^p \quad (r \leq d)$	$r^2 d \quad (r \leq d^{\lfloor \frac{p-1}{2} \rfloor}, p \geq 3)$ $\frac{p(r^2 d - 1)}{3} \quad (r = d, p/3 \in \mathbb{N})$
Upper bound	$2dp \log(12p)$	$2prd \log(12p)$	$2(r^p + prd) \log(24p)$	$2pr^2 d \log(12p)$

- $d \times d$ matrices of rank at most r:
  $\hookrightarrow$ $rd \leq d_{VC}(\mathcal{H}_{G_{\mathrm{mat}}}(r)) \leq 20rd$
- TT tensor of rank at most r
  $\hookrightarrow$ $p(r^2 d - 1)/3 \leq d_{VC}(\mathcal{H}_{G_{\mathrm{TT}}}(r)) \leq pr^2 d \cdot 2 \log(12p)$

Lower-bounds

Decomposition	rank one 	CP 	Tucker 	TT / TR  / 
Lower Bound (condition)	$(d-1)p$	rd $(r \leq d^{p-1})$	r^p $(r \leq d)$	$r^2 d$ $(r \leq d^{\lfloor \frac{p-1}{2} \rfloor}, p \geq 3)$ $\frac{p(r^2d-1)}{3}$ $(r = d, p/3 \in \mathbb{N})$
Upper bound	$2dp\log(12p)$	$2prd\log(12p)$	$2(r^p + prd)\log(24p)$	$2pr^2d\log(12p)$

The proofs rely on the following useful lemma:

Lemma 10. *Let $V \subset \mathbb{R}^d$ and define the hypothesis classes*

$$\mathcal{H}^{completion} = \{h : i \mapsto \mathbf{w}_i \mid \mathbf{w} \in V\}$$

$$\mathcal{H}^{regression} = \{h : \mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle \mid \mathbf{w} \in V\}$$

$$\mathcal{H}^{classif} = \{h : \mathbf{x} \mapsto \operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle) \mid \mathbf{w} \in V\}.$$

If there exist k indices $i_1, \cdots, i_k \in [d]$ that are shattered by V, i.e., such that

$$|\{(\operatorname{sign}(\mathbf{w}_{i_1}), \operatorname{sign}(\mathbf{w}_{i_2}), \cdots, \operatorname{sign}(\mathbf{w}_{i_k})) \mid \mathbf{w} \in V\}| = 2^k,$$

then $d_{\mathrm{VC}}(\mathcal{H}^{classif})$, $\mathrm{Pdim}(\mathcal{H}^{regression})$ and $\mathrm{Pdim}(\mathcal{H}^{completion})$ are all lower-bounded by k.

Conclusion & Future Directions

- VC-dimension of TN-based Models
  - $\hookrightarrow$ General Upper Bound
  - $\hookrightarrow$ Lower Bounds for some common TNs (showing the upper bound is tight)
- Future work:
  - $\hookrightarrow$ Improve upper bound (remove $\log(12|V|)$ term
  - $\hookrightarrow$ Improve lower bounds for specific TNs (e.g. TT)
    - Leverage connections between TN and neural networks (see e.g., work of Nadav Cohen and co-authors) to connect our results to expressiveness and learnability of neural networks.

Thank you