Expressive Efficiency and Inductive Bias of Convolutional Networks:

Analysis & Design via Hierarchical Tensor Decompositions

Amnon Shashua

The Hebrew University of Jerusalem

Sources

Deep SimNets
N. Cohen, O. Sharir and A. Shashua
Computer Vision and Pattern Recognition (CVPR) 2016

On the Expressive Power of Deep Learning: A Tensor Analysis
N. Cohen, O. Sharir and A. Shashua
Conference on Learning Theory (COLT) 2016

Convolutional Rectifier Networks as Generalized Tensor Decompositions
N. Cohen and A. Shashua
International Conference on Machine Learning (ICML) 2016

Inductive Bias of Deep Convolutional Networks through Pooling Geometry
N. Cohen and A. Shashua
International Conference on Learning Representations (ICLR) 2017

Tractable Generative Convolutional Arithmetic Circuits
O. Sharir. R. Tamari, N. Cohen and A. Shashua
arXiv preprint 2017

On the Expressive Power of Overlapping Operations of Deep Networks
O. Sharir and A. Shashua
arXiv preprint 2017

Boosting Dilated Convolutional Networks with Mixed Tensor Decompositions
N. Cohen, R. Tamari and A. Shashua
arXiv preprint 2017

Deep Learning and Quantum Entanglement: Fundamental Connections with Implications to Network Design Y. Levine, D. Yakira, N. Cohen and A. Shashua
arXiv preprint 2017

Students


Nadav Cohen

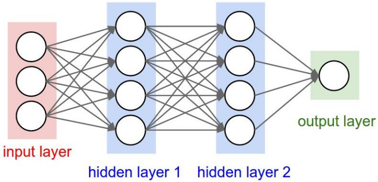

Or Sharir


Ronen Tamari


David Yakira


Yoav Levine

Classic



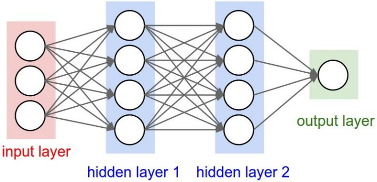
Multilayer Perceptron (MLP)

Architectural choices:

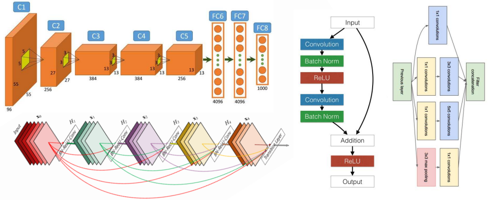
- depth
- layer widths
- activation types

Classic vs. State of the Art Deep Learning

Classic



State of the Art



Multilayer Perceptron (MLP)

Architectural choices:

- depth
- layer widths
- activation types

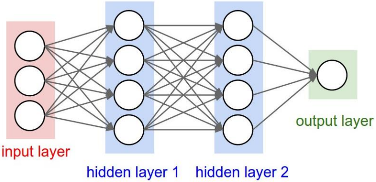
Convolutional Networks (ConvNets)

Architectural choices:

- depth
- layer widths
- activation types
- pooling types
- convolution/pooling windows
- convolution/pooling strides
- dilation factors
- connectivity
- and more...

Classic vs. State of the Art Deep Learning

Classic

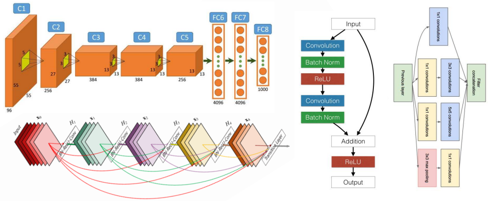


Multilayer Perceptron (MLP)

Architectural choices:

- depth
- layer widths
- activation types

State of the Art



Convolutional Networks (ConvNets)

Architectural choices:

- depth
- layer widths
- activation types
- pooling types
- convolution/pooling windows
- convolution/pooling strides
- and more...

Can the architectural choices of state of the art ConvNets be theoretically analyzed?

Outline

Expressiveness

Fundamental theoretical questions:

- What kind of functions can different network architectures represent?
- Why are these functions suitable for real-world tasks?
- What is the representational benefit of depth?
- Can other architecture features deliver representational benefits?
- What does it mean to have a "representational benefit"?

Efficiency

Expressive efficiency compares network architectures in terms of their ability to compactly represent functions

Efficiency

Expressive efficiency compares network architectures in terms of their ability to compactly represent functions

Let:

- $\mathcal{H}_A$ – space of func compactly representable by network arch A
- $\mathcal{H}_B$ – -"- network arch B

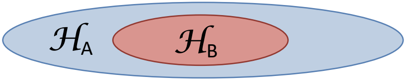
Efficiency

Expressive efficiency compares network architectures in terms of their ability to compactly represent functions

Let:

- $\mathcal{H}_A$ – space of func compactly representable by network arch A
- $\mathcal{H}_B$ – -"- network arch B

A is **efficient** w.r.t. B if $\mathcal{H}_B$ is a strict subset of $\mathcal{H}_A$



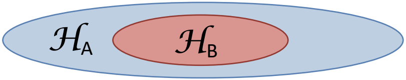
Efficiency

Expressive efficiency compares network architectures in terms of their ability to compactly represent functions

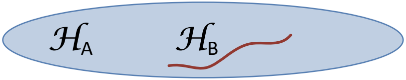
Let:

- $\mathcal{H}_A$ – space of func compactly representable by network arch A
- $\mathcal{H}_B$ – -"- network arch B

A is **efficient** w.r.t. B if $\mathcal{H}_B$ is a strict subset of $\mathcal{H}_A$



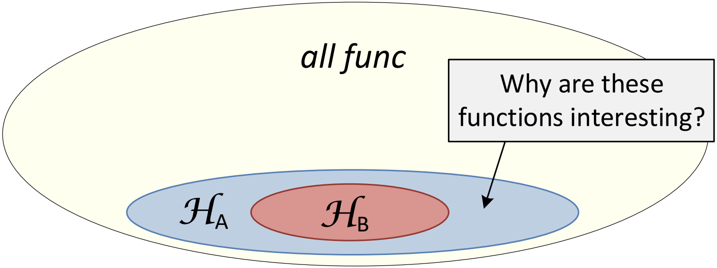
A is **completely efficient** w.r.t. B if $\mathcal{H}_B$ has zero "volume" inside $\mathcal{H}_A$



Inductive Bias

Networks of reasonable size can only realize a fraction of all possible func

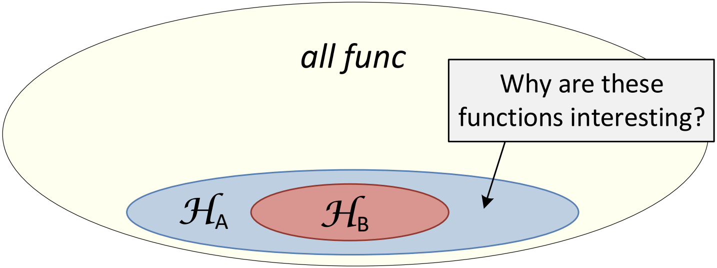
Efficiency does not explain why this fraction is effective



Inductive Bias

Networks of reasonable size can only realize a fraction of all possible func

Efficiency does not explain why this fraction is effective



To explain the effectiveness, one must consider the **inductive bias**:

- Not all functions are equally useful for a given task
- Network only needs to represent useful functions

Outline

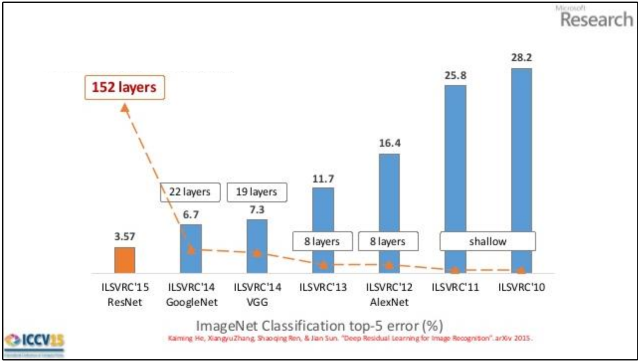
Questions about Efficiency and Inductive Bias

- Depth Efficiency: deep ConvNets are (exponentially) Efficient compared to shallow networks
- Pooling scheme affects inductive bias in an Efficient manner
- ConvNets with Overlapping convolution are Efficient compared to non-overlapping ones.
- Modern connectivity schemes (split/merge/skip) are Efficient compared to standard feed-forward (LeNet, AlexNet,..).
- Layer width distribution affects inductive bias in an Efficient manner.

Efficiency of Depth

Longstanding conjecture, proven for MLP:

deep networks are efficient w.r.t. shallow ones

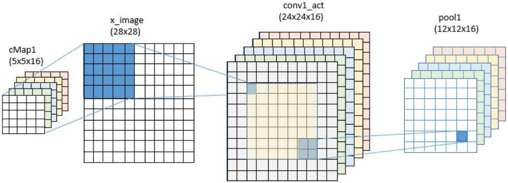


Q: *Can this be proven for ConvNets?*

Q: *Is their efficiency of depth complete?*

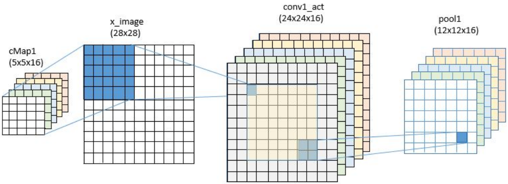
Inductive Bias of Convolution/Pooling Geometry

ConvNets typically employ square conv/pool windows

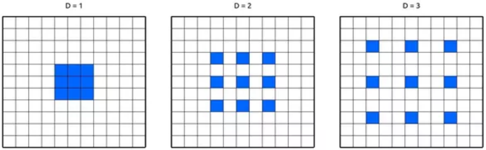


Inductive Bias of Convolution/Pooling Geometry

ConvNets typically employ square conv/pool windows



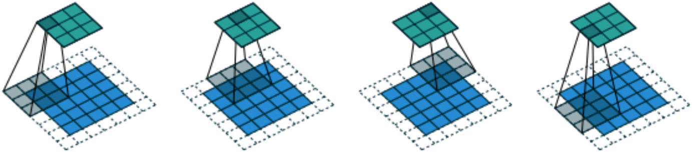
Recently, dilated windows have also become popular

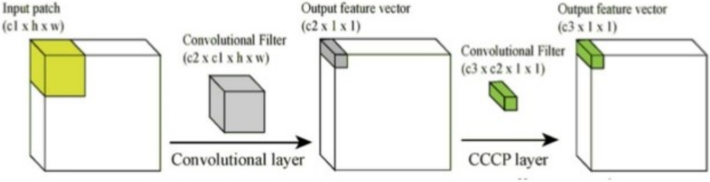


Q: *Conv/Pooling Scheme* $\leftrightarrow$ *Set of functions modeled per network size* $\leftrightarrow$ *Suitability per task*

Efficiency of Overlapping Operations

Modern ConvNets employ both overlapping and non-overlapping conv/pool operations

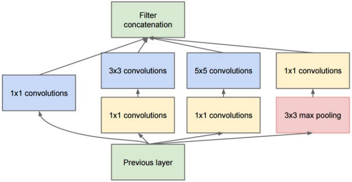




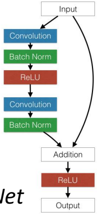
Q: *ConvNets with Overlapping conv are expressively Efficient w.r.t. those without (stride = kernel size)?*

Efficiency of Connectivity Schemes

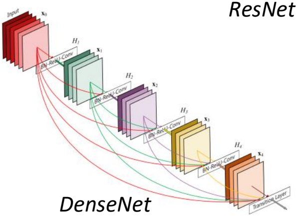
Nearly all state of the art ConvNets employ elaborate connectivity schemes: layers in parallel, split/merge/skip connections..



Inception (GoogLeNet)



ResNet

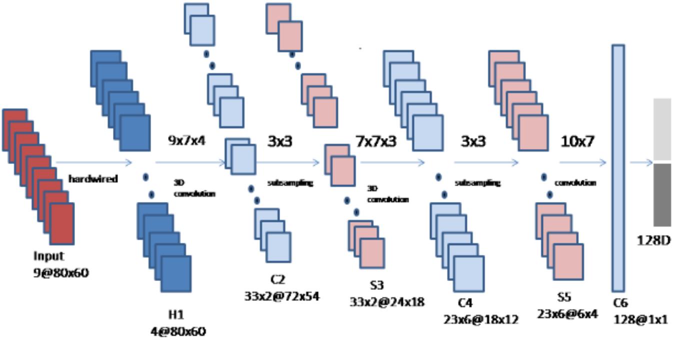


DenseNet

Q: *Connectivity schemes are Efficient compared to standard feed-forward (LeNet, Alexnet,..)?*

Inductive Bias of Layer Widths

No clear principle for setting widths (# of channels) of ConvNet layers

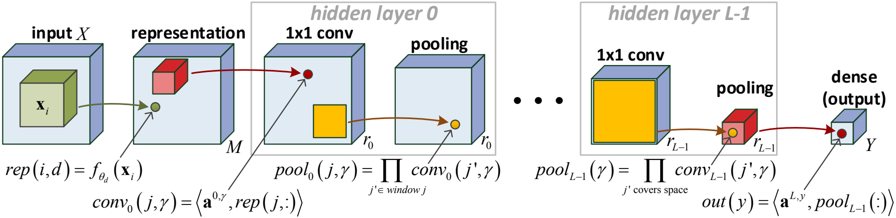


Q: *What is the inductive bias of one layer's width vs. another's?*

Q: *Can the widths be tailored for a given task?*

Outline

Convolutional Arithmetic Circuits: Baseline Architecture

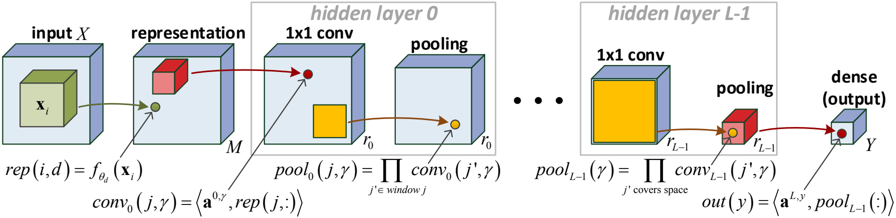


Baseline ConvAC architecture:

- *Linear activation* ($\sigma(z) = z$), *product pooling* ($P\{c_j\} = \prod_j c_j$)
- 1×1 convolution windows (non-overlapping convolution: stride = kernel size).

Intimate relationship to math machinery: tensor analysis, measure theory, functional analysis and graph theory.

Coefficient Tensor



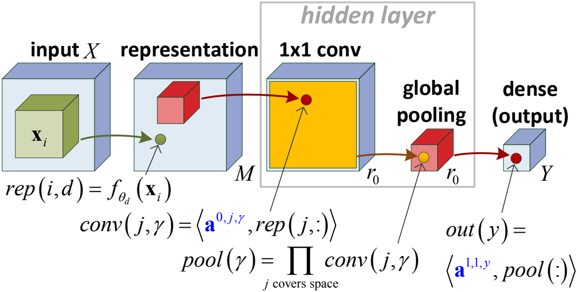
Function realized by output y:

$$h_y(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \sum_{d_1 \ldots d_N = 1}^{M} \mathcal{A}^y_{d_1,\ldots,d_N} \prod_{i=1}^{N} f_{\theta_{d_i}}(\mathbf{x}_i)$$

- $\mathbf{x}_1 \ldots \mathbf{x}_N$ – input patches
- $f_{\theta_1} \ldots f_{\theta_M}$ – representation layer functions
- $\mathcal{A}^y$ – **coefficient tensor** (M^N entries, polynomials in weights $\mathbf{a}^{l,j,\gamma}$)

Shallow Convolutional Arithmetic Circuit $\longleftrightarrow$ CP (CANDECOMP/PARAFAC) Decomposition

Shallow network (single hidden layer, global pooling):



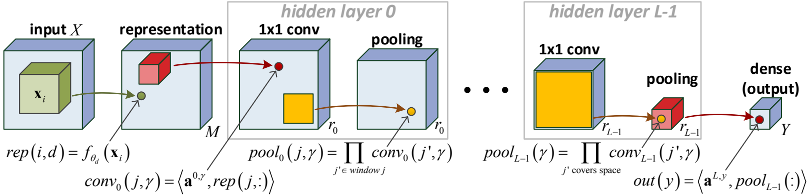
Coefficient tensor $\mathcal{A}^y$ given by classic **CP decomposition**:

$$\mathcal{A}^y = \sum_{\gamma=1}^{r_0} a_\gamma^{1,1,y} \cdot \underbrace{\mathbf{a}^{0,1,\gamma} \otimes \mathbf{a}^{0,2,\gamma} \otimes \cdots \otimes \mathbf{a}^{0,N,\gamma}}_{\text{rank-1 tensor}}$$

$$(rank(\mathcal{A}^y) \leq r_0)$$

Deep Convolutional Arithmetic Circuit $\longleftrightarrow$ Hierarchical Tucker Decomposition

Deep network ($L = \log_2 N$ hidden layers, size-2 pooling windows):



Coefficient tensor $\mathcal{A}^y$ given by **Hierarchical Tucker decomposition**:

$$
\begin{aligned}
\phi^{1,j,\gamma} &= \sum\nolimits_{\alpha=1}^{r_0} a_\alpha^{1,j,\gamma} \cdot \mathbf{a}^{0,2j-1,\alpha} \otimes \mathbf{a}^{0,2j,\alpha} \\
&\cdots \\
\phi^{l,j,\gamma} &= \sum\nolimits_{\alpha=1}^{r_{l-1}} a_\alpha^{l,j,\gamma} \cdot \phi^{l-1,2j-1,\alpha} \otimes \phi^{l-1,2j,\alpha} \\
&\cdots \\
\mathcal{A}^y &= \sum\nolimits_{\alpha=1}^{r_{L-1}} a_\alpha^{L,1,y} \cdot \phi^{L-1,1,\alpha} \otimes \phi^{L-1,2,\alpha}
\end{aligned}
$$

Universality

Fact:

CP decomposition can realize any tensor $\mathcal{A}^y$ given M^N terms

Implies:

Shallow network can realize any function given M^N hidden channels

Fact:

Hierarchical Tucker decomposition is a superset of CP decomposition if each level has matching number of terms

Implies:

Deep network can realize any function given M^N channels in each of its hidden layers

$$\boxed{\text{Convolutional arithmetic circuits are universal}}$$

Outline

Tensor Matricization

Let $\mathcal{A}$ be a tensor of order (dim) N

Let (I, J) be a partition of $[N]$, i.e. $I \dot\cup J = [N] := \{1, \ldots, N\}$

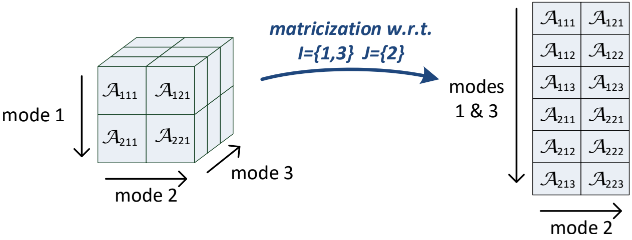
Tensor Matricization

Let $\mathcal{A}$ be a tensor of order (dim) N

Let (I, J) be a partition of $[N]$, i.e. $I \dot\cup J = [N] := \{1, \ldots, N\}$

$[\![\mathcal{A}]\!]_{I,J}$ – **matricization of $\mathcal{A}$ w.r.t. (I, J)**:

- Arrangement of $\mathcal{A}$ as matrix
- Rows correspond to modes (axes) indexed by I
- Cols -"- J



Exponential & Complete Efficiency of Depth

Claim

Tensors generated by CP decomposition w/r_0 terms, when matricized under any partition (I, J), have rank r_0 or less

Exponential & Complete Efficiency of Depth

Claim

Tensors generated by CP decomposition w/r_0 terms, when matricized under any partition (I, J), have rank r_0 or less

Theorem

Consider the partition $I_{odd} = \{1, 3, \ldots, N-1\}$, $J_{even} = \{2, 4, \ldots, N\}$. Besides a set of measure zero, all param settings of HT decomposition give tensors that when matricized w.r.t. (I_{odd}, J_{even}), have exponential ranks.

Exponential & Complete Efficiency of Depth

Claim

Tensors generated by CP decomposition w/r_0 terms, when matricized under any partition (I, J), have rank r_0 or less

Theorem

Consider the partition $I_{odd} = \{1, 3, \ldots, N-1\}$, $J_{even} = \{2, 4, \ldots, N\}$. Besides a set of measure zero, all param settings of HT decomposition give tensors that when matricized w.r.t. (I_{odd}, J_{even}), have exponential ranks.

Since # of terms in CP decomposition corresponds to # of hidden channels in shallow ConvAC:

Corollary

Almost all func realizable by deep ConvAC cannot be replicated by shallow ConvAC with less than exponentially many hidden channels

Exponential & Complete Efficiency of Depth

Claim

Tensors generated by CP decomposition w/r_0 terms, when matricized under any partition (I, J), have rank r_0 or less

Theorem

Consider the partition $I_{odd} = \{1, 3, \ldots, N-1\}$, $J_{even} = \{2, 4, \ldots, N\}$. Besides a set of measure zero, all param settings of HT decomposition give tensors that when matricized w.r.t. (I_{odd}, J_{even}), have exponential ranks.

Since # of terms in CP decomposition corresponds to # of hidden channels in shallow ConvAC:

Corollary

Almost all func realizable by deep ConvAC cannot be replicated by shallow ConvAC with less than exponentially many hidden channels

W/ConvACs efficiency of depth is exponential and complete!

Depth Efficiency Theorem – Proof Sketch

- $[\![\mathcal{A}]\!]$ – arrangement of tensor $\mathcal{A}$ as matrix (*matricization*)
- Relation between tensor and Kronecker products: $[\![\mathcal{A} \otimes \mathcal{B}]\!] = [\![\mathcal{A}]\!] \odot [\![\mathcal{B}]\!]$
- $\odot$ – Kronecker product for matrices. Holds: $rank(A \odot B) = rank(A) \cdot rank(B)$
- Implies: $\mathcal{A} = \sum_{z=1}^{Z} \lambda_z \mathbf{v}_1^{(z)} \otimes \cdots \otimes \mathbf{v}_{2^L}^{(z)} \implies rank[\![\mathcal{A}]\!] \leq Z$
- By induction over $l = 1 \ldots L$, almost everywhere w.r.t. $\{\mathbf{a}^{l,j,\gamma}\}_{l,j,\gamma}$:

$$\forall j \in [N/2^l], \gamma \in [r_l] : rank[\![\phi^{l,j,\gamma}]\!] \geq (\min\{r_0, M\})^{2^l/2}$$

  - Base: "SVD has maximal rank almost everywhere"
  - Step: $rank[\![\mathcal{A} \otimes \mathcal{B}]\!] = rank([\![\mathcal{A}]\!] \odot [\![\mathcal{B}]\!]) = rank[\![\mathcal{A}]\!] \cdot rank[\![\mathcal{B}]\!]$, and "linear combination preserves rank almost everywhere"

A Note about Measure Zero

- Depth Efficiency occurs with probability 1, i..e, besides a set of measure zero, all functions that can be implemented by a deep network of polynomial size, require exponential size in order to be realized (or even approximated) by a shallow network.

A Note about Measure Zero

- Depth Efficiency occurs with probability 1, i..e, besides a set of measure zero, all functions that can be implemented by a deep network of polynomial size, require exponential size in order to be realized (or even approximated) by a shallow network.
- The set is a zero set of a certain polynomial (based on determinants).
- The zero set of a polynomial is closed, i.e., cannot approximate anything that is not included in the set.
- In other words, the *closure* of the set is also of *measure zero*.

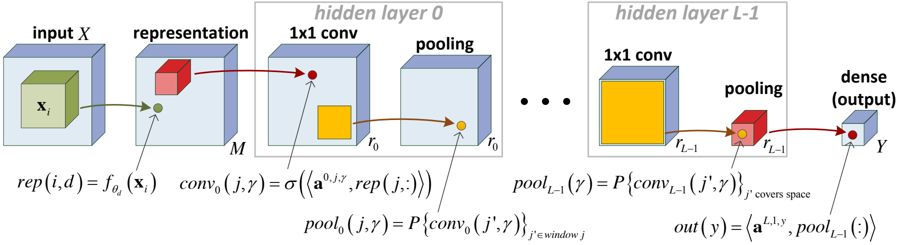
A Note about Measure Zero

- Depth Efficiency occurs with probability 1, i..e, besides a set of measure zero, all functions that can be implemented by a deep network of polynomial size, require exponential size in order to be realized (or even approximated) by a shallow network.
- The set is a zero set of a certain polynomial (based on determinants).
- The zero set of a polynomial is closed, i.e., cannot approximate anything that is not included in the set.
- In other words, the *closure* of the set is also of *measure zero*.
- For example, the set of Rational numbers is of measure zero, but the closure of the set is **not** of measure zero. It actually fills the entire space.

A Note about Measure Zero

- Depth Efficiency occurs with probability 1, i..e, besides a set of measure zero, all functions that can be implemented by a deep network of polynomial size, require exponential size in order to be realized (or even approximated) by a shallow network.
- The set is a zero set of a certain polynomial (based on determinants).
- The zero set of a polynomial is closed, i.e., cannot approximate anything that is not included in the set.
- In other words, the *closure* of the set is also of *measure zero*.
- For example, the set of Rational numbers is of measure zero, but the closure of the set is **not** of measure zero. It actually fills the entire space.
- Therefore, the set of functions that do not satisfy depth efficiency should be viewed as a low-dimensional manifold rather than a scattered set in space.

From Convolutional Arithmetic Circuits to Convolutional Rectifier Networks



Transform ConvACs into **convolutional rectifier networks** (R-ConvNets):

linear activation $\longrightarrow$ ReLU activation: $\sigma(z) = \max\{z, 0\}$

product pooling $\longrightarrow$ max/average pooling: $P\{c_j\} = max\{c_j\}/mean\{c_j\}$

Generalized Tensor Decompositions

ConvACs correspond to tensor decompositions based on tensor product $\otimes$:

$$(\mathcal{A} \otimes \mathcal{B})_{d_1,\ldots,d_{P+Q}} = \mathcal{A}_{d_1,\ldots,d_P} \cdot \mathcal{B}_{d_{P+1},\ldots,d_{P+Q}}$$

Generalized Tensor Decompositions

ConvACs correspond to tensor decompositions based on tensor product $\otimes$:

$$(\mathcal{A} \otimes \mathcal{B})_{d_1,\ldots,d_{P+Q}} = \mathcal{A}_{d_1,\ldots,d_P} \cdot \mathcal{B}_{d_{P+1},\ldots,d_{P+Q}}$$

For an operator $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, the **generalized tensor product** $\otimes_g$:

$$(\mathcal{A} \otimes_g \mathcal{B})_{d_1,\ldots,d_{P+Q}} := g(\mathcal{A}_{d_1,\ldots,d_P}, \mathcal{B}_{d_{P+1},\ldots,d_{P+Q}})$$

(same as $\otimes$ but with $g(\cdot)$ instead of multiplication)

Generalized tensor decompositions are obtained by replacing $\otimes$ with $\otimes_g$

Convolutional Rectifier Networks $\longleftrightarrow$ Generalized Tensor Decompositions

Define the **activation-pooling operator**:

$$\rho_{\sigma/P}(a,b) := P\{\sigma(a),\sigma(b)\}$$

[1] Sum and average pooling are equivalent in terms of expressiveness

Convolutional Rectifier Networks $\longleftrightarrow$ Generalized Tensor Decompositions

Define the **activation-pooling operator**:

$$\rho_{\sigma/P}(a,b) := P\{\sigma(a), \sigma(b)\}$$

- **ReLU activation**: $\sigma(z) = [z]_+ := \max\{z, 0\}$
- **max/average pooling**: $P\{c_j\} = max\{c_j\}/mean\{c_j\}$

[1] Sum and average pooling are equivalent in terms of expressiveness

Convolutional Rectifier Networks $\longleftrightarrow$ Generalized Tensor Decompositions

Define the **activation-pooling operator**:

$$\rho_{\sigma/P}(a,b) := P\{\sigma(a),\sigma(b)\}$$

- **ReLU activation**: $\sigma(z) = [z]_+ := \max\{z,0\}$
- **max/average pooling**: $P\{c_j\} = max\{c_j\}/mean\{c_j\}$

Corresponding activation-pooling operators associative and commutative:

- $\rho_{ReLU/max}(a,b) := \max\{[a]_+,[b]_+\} = \max\{a,b,0\}$
- $\rho_{ReLU/sum}(a,b) := [a]_+ + [b]_+$ [1]

[1] Sum and average pooling are equivalent in terms of expressiveness

Exponential But Incomplete Efficiency of Depth

By analyzing matricization ranks of tensors realized by generalized CP and HT decompositions w/$g(\cdot) \equiv \rho_{\sigma/P}(\cdot)$, we show:

Claim

There exist func realizable by deep R-ConvNet requiring shallow R-ConvNet to be exponentially large

Exponential But Incomplete Efficiency of Depth

By analyzing matricization ranks of tensors realized by generalized CP and HT decompositions w/$g(\cdot) \equiv \rho_{\sigma/P}(\cdot)$, we show:

Claim

There exist func realizable by deep R-ConvNet requiring shallow R-ConvNet to be exponentially large

On the other hand:

Claim

A non-negligible (positive measure) set of the func realizable by deep R-ConvNet can be replicated by shallow R-ConvNet w/few hidden channels

Exponential But Incomplete Efficiency of Depth

By analyzing matricization ranks of tensors realized by generalized CP and HT decompositions w/$g(\cdot) \equiv \rho_{\sigma/P}(\cdot)$, we show:

Claim

There exist func realizable by deep R-ConvNet requiring shallow R-ConvNet to be exponentially large

On the other hand:

Claim

A non-negligible (positive measure) set of the func realizable by deep R-ConvNet can be replicated by shallow R-ConvNet w/few hidden channels

W/R-ConvNets efficiency of depth is exponential but incomplete!

Outline

Separation Rank – A Measure of Input Correlations

ConvNets realize func over many local structures:

$$f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$$

$\mathbf{x}_i$ – image patches (2D network) / sequence samples (1D network)

Separation Rank – A Measure of Input Correlations

ConvNets realize func over many local structures:

$$f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$$

$\mathbf{x}_i$ – image patches (2D network) / sequence samples (1D network)

Important feature of $f(\cdot)$ – **correlations** it models between the $\mathbf{x}_i$'s

Separation Rank – A Measure of Input Correlations

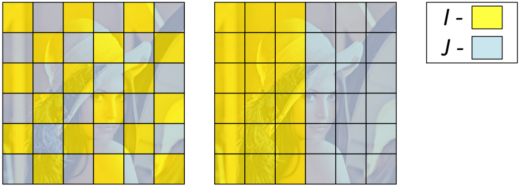
ConvNets realize func over many local structures:

$$f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$$

$\mathbf{x}_i$ – image patches (2D network) / sequence samples (1D network)

Important feature of $f(\cdot)$ – **correlations** it models between the $\mathbf{x}_i$'s

Separation rank:
Formal measure of these correlations



Partition A *Partition B*

Sep rank of $f(\cdot)$ w.r.t. input partition (I, J) measures dist from separability
(sep rank $\nearrow$ $\implies$ more correlation between $(\mathbf{x}_i)_{i \in I}$ and $(\mathbf{x}_j)_{j \in J}$)

Deep Networks Favor Some Correlations Over Others

Claim

W/ConvAC sep rank w.r.t (I, J) is equal to rank of $[\![\mathcal{A}^y]\!]_{I,J}$ – matricized w.r.t. (I, J)

Deep Networks Favor Some Correlations Over Others

Claim

W/ConvAC sep rank w.r.t (I, J) is equal to rank of $[\![\mathcal{A}^y]\!]_{I,J}$ – matricized w.r.t. (I, J)

Theorem

Maximal rank of tensor generated by HT decomposition, when matricized w.r.t. (I, J), is:

- *Exponential for "interleaved" partitions*
- *Polynomial for "coarse" partitions*

Deep Networks Favor Some Correlations Over Others

Claim

W/ConvAC sep rank w.r.t (I, J) is equal to rank of $[\![\mathcal{A}^y]\!]_{I,J}$ – matricized w.r.t. (I, J)

Theorem

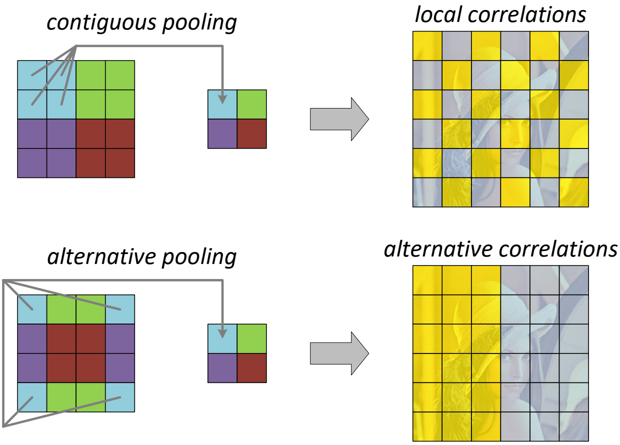
Maximal rank of tensor generated by HT decomposition, when matricized w.r.t. (I, J), is:

- *Exponential for "interleaved" partitions*
- *Polynomial for "coarse" partitions*

Corollary

Deep ConvAC can realize exponential sep ranks (correlations) for favored partitions, polynomial for others

Pooling Geometry Controls the Preference

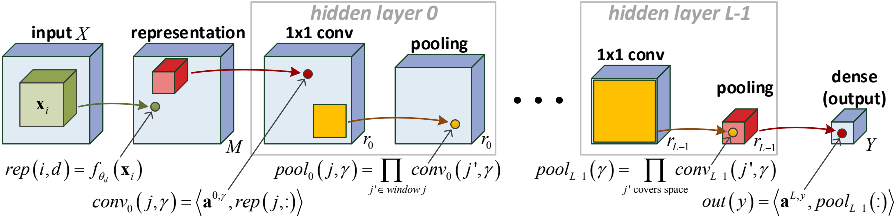


Pooling geometry of deep ConvAC determines which partitions are favored – controls the correlation profile (inductive bias)!

Outline

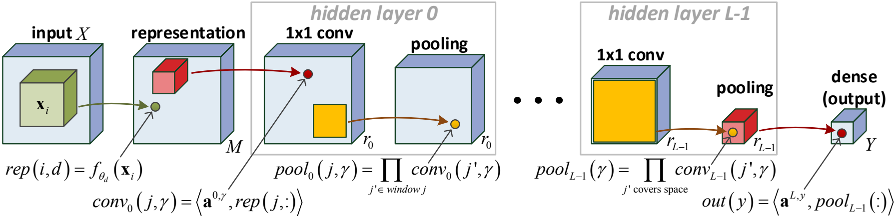
Overlapping Operations

Baseline ConvAC arch has non-overlapping conv and pool windows:

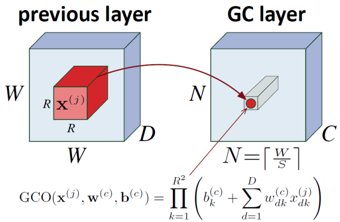


Overlapping Operations

Baseline ConvAC arch has non-overlapping conv and pool windows:



Replace those by (possibly) overlapping **generalized convolution**:



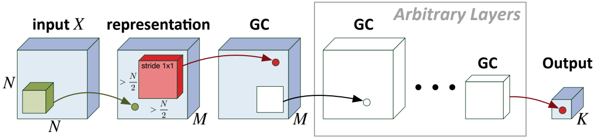
Exponential Efficiency

Theorem

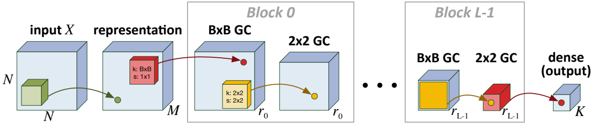
Various ConvACs w/overlapping GC layers realize func requiring ConvAC w/no overlaps to be exponentially large

Examples

- Network starts with large receptive field:



- Typical scheme of alternating $B \times B$ "conv" and 2×2 "pool":



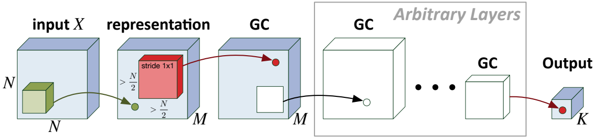
Exponential Efficiency

Theorem

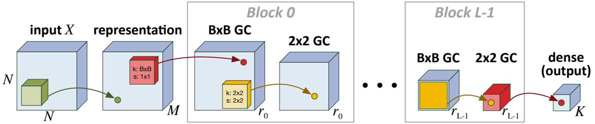
Various ConvACs w/overlapping GC layers realize func requiring ConvAC w/no overlaps to be exponentially large

Examples

- Network starts with large receptive field:



- Typical scheme of alternating $B \times B$ "conv" and 2×2 "pool":

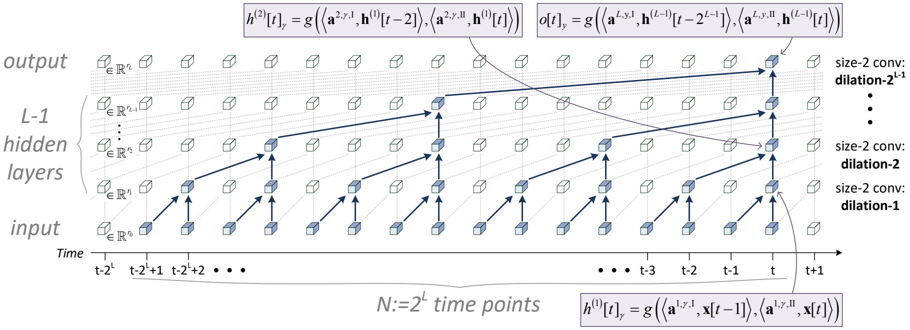


W/ConvACs overlaps lead to exponential efficiency!

Outline

Dilated Convolutional Networks

Study efficiency of interconnectivity w/**dilated convolutional networks**:

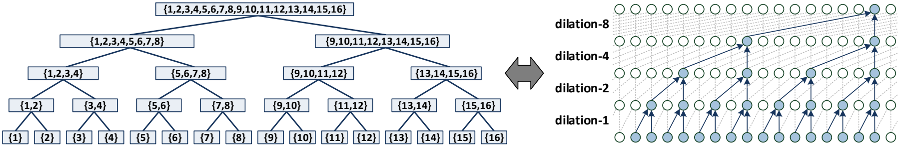


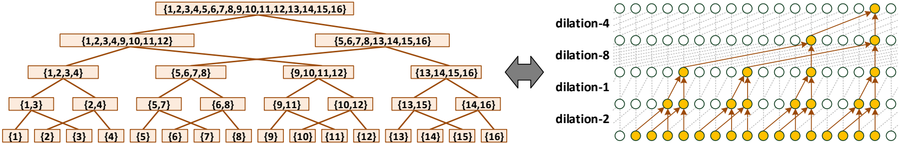
- 1D ConvNets (sequence data)
- Dilated (gapped) conv windows
- No pooling

Underlie Google's WaveNet & ByteNet – state of the art for audio & text!

Mixing Tensor Decompositions $\longrightarrow$ Interconnectivity

With dilated ConvNets, mode (axes) tree underlying corresponding tensor decomposition determines dilation scheme





Mixed tensor decomposition blending different mode (axes) trees corresponds to interconnected networks with different dilations

Efficiency of Interconnectivity

Theorem

Mixed tensor decomposition generates tensors that can only be realized by individual decompositions if these grow quadratically

Corollary

Interconnected dilated ConvNets realize func that cannot be realized by individual networks unless these are quadratically larger

Efficiency of Interconnectivity

Theorem

Mixed tensor decomposition generates tensors that can only be realized by individual decompositions if these grow quadratically

Corollary

Interconnected dilated ConvNets realize func that cannot be realized by individual networks unless these are quadratically larger

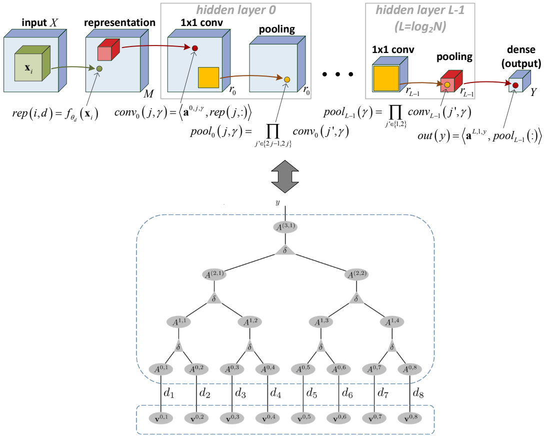
W/dilated ConvNets interconnectivity brings efficiency!

Outline

Convolutional Arithmetic Circuits ⟷ Contraction Graphs

Computation of ConvAC can be cast as a **contraction graph** G, where:

- Edge weights hold layer widths (# of channels)
- Degree-1 nodes correspond to input patches



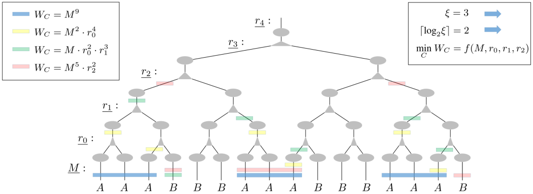
Correlations $\longleftrightarrow$ Min-Cut over Layer Widths

Theorem

For input partition (I, J), the rank of $\mathcal{A}^y$ matricized w.r.t. (I, J) is upper-bounded by the min-cut in G separating the degree-1 nodes of I from those of J.

Corollary

To model interactions between input regions represented by a specific bi-partition, it is required to set layer widths such that the min-cut is of high value. A low value represents "bottlenecks" in expressivity.



The Quantum Many-Body Wave Function

A state of a system (interchangeably its wave function) is denoted by:

$$|\psi\rangle \in \mathcal{H}$$

- $\mathcal{H}$ - the relevant Hilbert Space
- $|\psi\rangle$ - vector in the Hilbert Space ('ket' notation)

The Quantum Many-Body Wave Function

A state of a system (interchangeably its wave function) is denoted by:

$$|\psi\rangle \in \mathcal{H}$$

- $\mathcal{H}$ - the relevant Hilbert Space
- $|\psi\rangle$ - vector in the Hilbert Space ('ket' notation)

For a single particle with a WF in an M dimensional Hilbert space $\mathcal{H}_1$:

$$|\psi\rangle = \sum_{d=1}^{M} \underbrace{v_d}_{\substack{\textbf{coefficients}\\ \textbf{vector}}} |\psi_d\rangle$$

The Quantum Many-Body Wave Function

A state of a system (interchangeably its wave function) is denoted by:

$$|\psi\rangle \in \mathcal{H}$$

- $\mathcal{H}$ - the relevant Hilbert Space
- $|\psi\rangle$ - vector in the Hilbert Space ('ket' notation)

For a single particle with a WF in an M dimensional Hilbert space $\mathcal{H}_1$:

$$|\psi\rangle = \sum_{d=1}^{M} \underbrace{v_d}_{\substack{\textbf{coefficients}\\ \textbf{vector}}} |\psi_d\rangle$$

<u>The quantum many-body WF</u>: $\left(|\psi\rangle \in \mathcal{H} = \otimes_{j=1}^{N} \mathcal{H}_j\right)$

$$|\psi\rangle = \sum_{d_1 \ldots d_N=1}^{M} \underbrace{\mathcal{A}_{d_1 \ldots d_N}}_{\substack{\textbf{coefficients}\\ \textbf{tensor}}} |\psi_{d_1}\rangle \otimes \cdots \otimes |\psi_{d_N}\rangle$$

A Tailored Product State

Consider a single tensor product of local states $|\phi_j\rangle \in \mathcal{H}_j$:

$$|\psi^{\text{ps}}\rangle = |\phi_1\rangle \otimes \cdots \otimes |\phi_N\rangle$$

A Tailored Product State

Consider a single tensor product of local states $|\phi_j\rangle \in \mathcal{H}_j$:

$$|\psi^{\text{ps}}\rangle = |\phi_1\rangle \otimes \cdots \otimes |\phi_N\rangle$$

By expanding each local state in the respective basis,
$|\phi_j\rangle = \sum_{d_j=1}^{M} v_{d_j}^{(j)} \left|\psi_{d_j}\right\rangle$, the product state assumes the form:

$$|\psi^{\text{ps}}\rangle = \sum_{d_1 \ldots d_N=1}^{M} \mathcal{A}^{\text{ps}}_{d_1 \ldots d_N} |\psi_{d_1}\rangle \otimes \cdots \otimes |\psi_{d_N}\rangle$$

$\mathcal{A}^{\text{ps}}_{d_1 \ldots d_N} = \prod_{j=1}^{N} v_{d_j}^{(j)}$ is a rank-1 tensor

A Tailored Product State

Consider a single tensor product of local states $|\phi_j\rangle \in \mathcal{H}_j$:

$$|\psi^{\text{ ps}}\rangle = |\phi_1\rangle \otimes \cdots \otimes |\phi_N\rangle$$

By expanding each local state in the respective basis,

$|\phi_j\rangle = \sum_{d_j=1}^{M} v_{d_j}^{(j)} \left|\psi_{d_j}\right\rangle$, the product state assumes the form:

$$|\psi^{\text{ ps}}\rangle = \sum_{d_1 \ldots d_N=1}^{M} \mathcal{A}_{d_1 \ldots d_N}^{\text{ps}} |\psi_{d_1}\rangle \otimes \cdots \otimes |\psi_{d_N}\rangle$$

$\mathcal{A}_{d_1 \ldots d_N}^{\text{ps}} = \prod_{j=1}^{N} v_{d_j}^{(j)}$ is a rank-1 tensor

We compose each local state $|\phi_j\rangle$ s.t. its projection on the local basis vector equals $v_d^{(j)} = \langle\psi_d|\phi_j\rangle = f_{\theta_d}(\mathbf{x}_j)$

$\longrightarrow \quad \mathcal{A}_{d_1 \ldots d_N}^{\text{ps}} = \prod_{j=1}^{N} f_{\theta_{d_j}}(\mathbf{x}_j)$

Equivalence to a ConvAC

- Many-body WF:
  $|\psi\rangle = \sum_{d_1 \dots d_N = 1}^{M} \mathcal{A}_{d_1 \dots d_N} |\psi_{d_1}\rangle \otimes \cdots \otimes |\psi_{d_N}\rangle$
- Constructed product state:
  $|\psi^{\text{ ps}}\rangle = \sum_{d_1 \dots d_N = 1}^{M} \prod_{j=1}^{N} f_{\theta_{d_j}}(\mathbf{x}_j) |\psi_{d_1}\rangle \otimes \cdots \otimes |\psi_{d_N}\rangle$

Equivalence to a ConvAC

- Many-body WF:
  $|\psi\rangle = \sum_{d_1 \ldots d_N=1}^{M} \mathcal{A}_{d_1 \ldots d_N} |\psi_{d_1}\rangle \otimes \cdots \otimes |\psi_{d_N}\rangle$
- Constructed product state:
  $|\psi^{\text{ ps}}\rangle = \sum_{d_1 \ldots d_N=1}^{M} \prod_{j=1}^{N} f_{\theta_{d_j}}(\mathbf{x}_j) |\psi_{d_1}\rangle \otimes \cdots \otimes |\psi_{d_N}\rangle$

$$\longrightarrow \quad \langle \psi^{\text{ ps}} | \psi \rangle = \sum_{d_1 \ldots d_N=1}^{M} \mathcal{A}_{d_1 \ldots d_N} \prod_{j=1}^{N} f_{\theta_{d_j}}(\mathbf{x}_j) = \mathbf{h}_y(\mathbf{x}_1, \ldots, \mathbf{x}_N)$$

Exactly reproducing the form of the function realized by a ConvAC!

conv weights tensor $\longleftrightarrow$ coefficients tensor

rep. functions on the inputs $\longleftrightarrow$ constructed product state

Quantum Entanglement

Lend means of quantifying physical correlations:

"Quantum Entanglement"

Quantum Entanglement

Lend means of quantifying physical correlations:

"Quantum Entanglement"

I **J**

Many-body WF:

$$|\psi\rangle = \sum_{\alpha=1}^{\dim(\mathcal{H}^I)} \sum_{\beta=1}^{\dim(\mathcal{H}^J)} (\llbracket\mathcal{A}\rrbracket_{I,J})_{\alpha,\beta} \left|\psi_\alpha^I\right\rangle \otimes \left|\psi_\beta^J\right\rangle$$

$\llbracket\mathcal{A}\rrbracket_{I,J}$ - **matricization of** $\mathcal{A}$ **according to** (I,J)

Quantum Entanglement

Lend means of quantifying physical correlations:

"Quantum Entanglement"

I **J**

Many-body WF:

$$|\psi\rangle = \sum_{\alpha=1}^{\dim(\mathcal{H}^I)} \sum_{\beta=1}^{\dim(\mathcal{H}^J)} (\llbracket \mathcal{A} \rrbracket_{I,J})_{\alpha,\beta} \left|\psi_\alpha^I\right\rangle \otimes \left|\psi_\beta^J\right\rangle$$

$\llbracket \mathcal{A} \rrbracket_{I,J}$ - **matricization of $\mathcal{A}$ according to (I, J)**

Change of basis (SVD) $\longrightarrow |\psi\rangle = \sum_{\alpha=1}^{r} \lambda_\alpha \left|\phi_\alpha^I\right\rangle \otimes \left|\phi_\alpha^J\right\rangle$

λ_α - **singular values of $\llbracket \mathcal{A} \rrbracket_{I,J}$**

Measures of Entanglement

Using the singular values of $[\![\mathcal{A}]\!]_{I,J}$, we can define several **Measures of Entanglement** between I and J.

Measures of Entanglement

Using the singular values of $[\![\mathcal{A}]\!]_{I,J}$, we can define several **Measures of Entanglement** between I and J.

Examples:

- Entanglement Entropy: the entropy of singular values
  $-\sum_\alpha |\lambda_\alpha|^2 \ln |\lambda_\alpha|^2$
- Geometric Measure: the L^2 distance of $|\psi\rangle$ from the set of separable states
  $\min_{|\psi^{\mathrm{sp}(I,J)}\rangle} |\langle\psi^{\mathrm{sp}(I,J)}|\psi\rangle|^2$ (**shown to be related to separation rank**)
- Schmidt Number: the number of non-zero singular values
  $\mathrm{rank}([\![\mathcal{A}]\!]_{I,J})$

Measures of Entanglement

Using the singular values of $[\![\mathcal{A}]\!]_{I,J}$, we can define several **Measures of Entanglement** between I and J.

Examples:

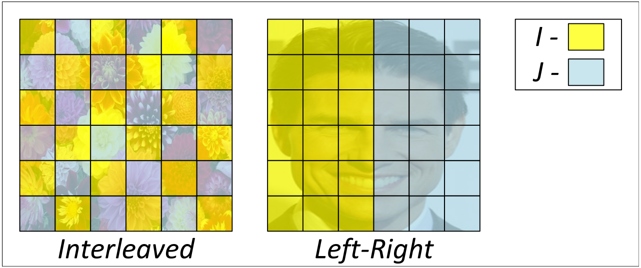
- Entanglement Entropy: the entropy of singular values
  $-\sum_\alpha |\lambda_\alpha|^2 \ln |\lambda_\alpha|^2$
- Geometric Measure: the L^2 distance of $|\psi\rangle$ from the set of separable states
  $\min_{|\psi^{\mathrm{sp}(I,J)}\rangle} |\langle\psi^{\mathrm{sp}(I,J)}|\psi\rangle|^2$ (**shown to be related to separation rank**)
- Schmidt Number: the number of non-zero singular values
  $\mathrm{rank}([\![\mathcal{A}]\!]_{I,J})$

All measures of entanglement:

- minimal for a separable state
- increase as the dependance between I and J becomes more complicated

Measures of Entanglement - Convolutional Network

Can now use entanglement measures to describe the correlations supported by a ConvAC:



The network should support high entanglement measures for the partitions which correspond to input correlations.

Tensor Networks

Physicists' approach for efficient representation of many-body WFs:

Tensor Networks (TNs)

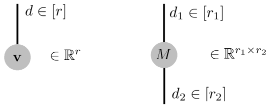
Tensor Networks

Physicists' approach for efficient representation of many-body WFs:

Tensor Networks (TNs)

The basic building blocks of a TN are tensors – nodes in the network:

vector: matrix:



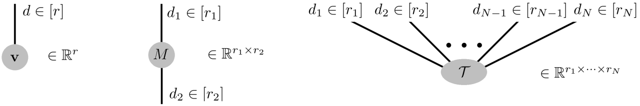
Tensor Networks

Physicists' approach for efficient representation of many-body WFs:

Tensor Networks (TNs)

The basic building blocks of a TN are tensors – nodes in the network:

vector: matrix: order N tensor:

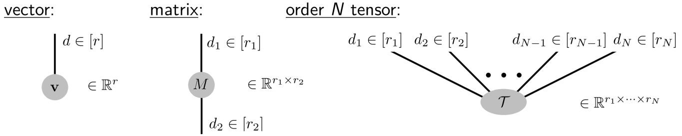


Tensor Networks

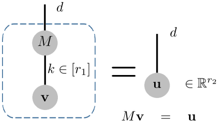
Physicists' approach for efficient representation of many-body WFs:

Tensor Networks (TNs)

The basic building blocks of a TN are tensors – nodes in the network:



- Internal indices are summed upon
- External indices belong to the resultant tensor



$$M\mathbf{v} = \mathbf{u}$$

$$\sum_{k=1}^{r_1} M_{dk} v_k = u_d$$

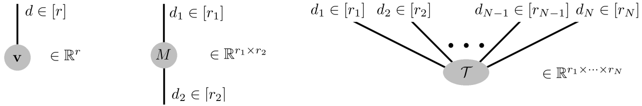
Tensor Networks

Physicists' approach for efficient representation of many-body WFs:

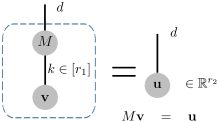
Tensor Networks (TNs)

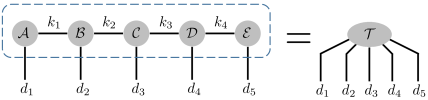
The basic building blocks of a TN are tensors – nodes in the network:

vector: matrix: order N tensor:



- Internal indices are summed upon
- External indices belong to the resultant tensor





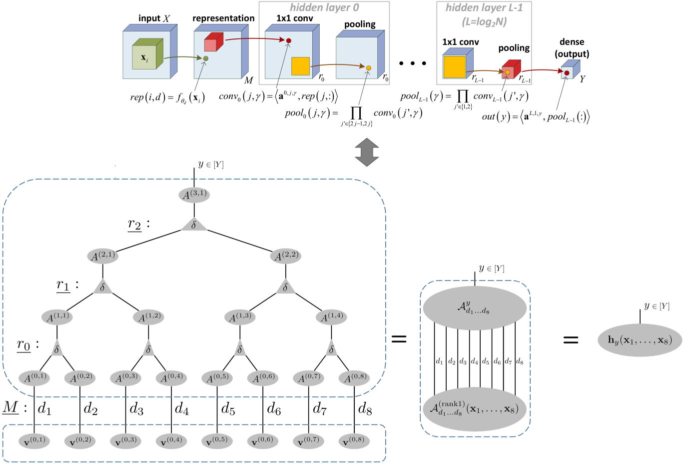
$$M\mathbf{v} = \mathbf{u}$$
$$\sum_{k=1}^{r_1} M_{dk} v_k = u_d$$

$$\mathcal{T}_{d_1 d_2 d_3 d_4 d_5} = \sum_{k_1,k_2,k_3,k_4=1}^{r} \mathcal{A}_{d_1 k_1} \mathcal{B}_{k_1 d_2 k_2} \mathcal{C}_{k_2 d_3 k_3} \mathcal{D}_{k_3 d_4 k_4} \mathcal{E}_{k_4 d_5} \quad \in \otimes_{i=1}^{5} \mathbb{R}^{r}$$

Convolutional Arithmetic Circuits $\longleftrightarrow$ Tensor Networks

Computation of ConvAC can be cast as a **Tensor Network**:

- Edge weights hold layer widths (# of channels)
- Degree-1 nodes correspond to input patches



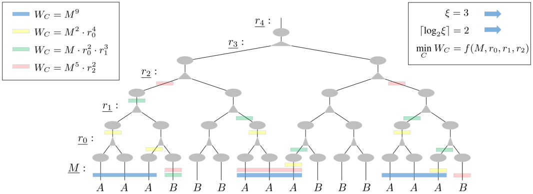
Correlations $\longleftrightarrow$ Min-Cut over Layer Widths

Theorem

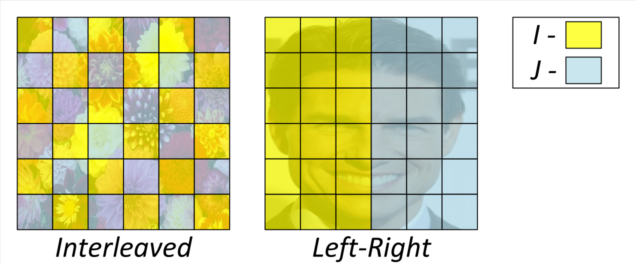
For input partition (I, J), the rank of $\mathcal{A}^y$ matricized w.r.t. (I, J) is upper-bounded by the min-cut in G separating the degree-1 nodes of I from those of J.

Corollary

To model interactions between input regions represented by a specific bi-partition, it is required to set layer widths such that the min-cut is of high value. A low value represents "bottlenecks" in expressivity.



Implications of the Quantum-min-cut on Layer Width



$$W_C^{\text{left-right}} = \min(r_{L-1}, r_{L-2}, ..., r_l^{2^{(L-2-l)}}, ..., r_0^{N/4}, M^{N/2}), \tag{1}$$

whereas the minimal weight of a cut w.r.t. the interleaved partition is guaranteed to be exponential in N and obeys:

$$W_C^{\text{interleaved}} = \min(r_0^{N/4}, M^{N/2}). \tag{2}$$

Outline

Conclusion

- **Expressiveness** – the driving force behind deep networks
- Formal concepts for treating expressiveness:
  - **Efficiency** – network arch realizes func requiring alternative arch to be much larger
  - **Inductive bias** – prioritization of some func over others given prior knowledge on task at hand
- We analyzed efficiency and inductive bias of ConvNet arch features:
  - depth
  - pooling geometry
  - overlapping operations
  - interconnectivity
  - layer widths
- Fundamental tool underlying all of our analyses:

ConvNets $\longleftrightarrow$ **hierarchical tensor decompositions**

Thank You