Expressive Efficiency and Inductive Bias of Convolutional Networks: Analysis & Design via Hierarchical Tensor Decompositions

Nadav Cohen

The Hebrew University of Jerusalem

AAAI Spring Symposium Series 2017
Science of Intelligence: Computational Principles of Natural and Artificial Intelligence
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*
Collaborators

Prof. Amnon Shashua

Or Sharir
Ronen Tamari
Yoav Levine
David Yakira

Expressiveness of Convolutional Networks
Science of Intelligence, 2017
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
  - dilation factors
  - connectivity
  - and more...

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

Nadav Cohen (Hebrew U.)
Classic vs. State of the Art Deep Learning

**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...

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

Nadav Cohen (Hebrew U.)

Expressiveness of Convolutional Networks

Science of Intelligence, 2017
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

1. Expressiveness

2. Expressiveness of Convolutional Networks – Questions

3. Convolutional Arithmetic Circuits

4. Efficiency of Depth (Cohen+Sharir+Shashua@COLT'16, Cohen+Shashua@ICML'16)

5. Inductive Bias of Pooling Geometry (Cohen+Shashua@ICLR'17)

6. Efficiency of Overlapping Operations (Sharir+Shashua@arXiv'17)

7. Efficiency of Interconnectivity (Cohen+Tamari+Shashua@arXiv'17)
Expressiveness:

- Ability to compactly represent rich and effective classes of functions
- The driving force behind deep networks
Expressiveness:

- Ability to compactly represent rich and effective classes of function
- The driving force behind deep networks

Fundamental theoretical questions:

- What kind of function can different network architectures represent?
- Why are these functions suitable for real-world tasks?
- What is the representational benefit of depth?
- Can other architectural features deliver representational benefits?
Expressive efficiency compares network arch in terms of their ability to compactly represent func

Let: $H_A$ – space of func compactly representable by network arch $A$

$H_A$ is efficient w.r.t. $B$ if $H_B$ is a strict subset of $H_A$.

$H_A$ is completely efficient w.r.t. $B$ if $H_B$ has zero “volume” inside $H_A$. 
Expressive efficiency compares network arch in terms of their ability to compactly represent func

Let:

- $\mathcal{H}_A$ – space of func compactly representable by network arch $A$
- $\mathcal{H}_B$ – network arch $B$
Expressive efficiency compares network arch in terms of their ability to compactly represent func

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 $\mathcal{H}_A$
Expressive efficiency compares network arch in terms of their ability to compactly represent func

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 $\mathcal{H}_A$
Efficiency – Formal Definition

Network arch $A$ is **efficient** w.r.t. network arch $B$ if:

1. $\forall$ func realized by $B$ w/size $r_B$ can be realized by $A$ w/size $r_A \in \mathcal{O}(r_B)$
2. $\exists$ func realized by $A$ w/size $r_A$ requiring $B$ to have size $r_B \in \Omega(f(r_A))$, where $f(\cdot)$ is super-linear

$A$ is **completely efficient** w.r.t. $B$ if (2) holds for all of its func but a set of Lebesgue measure zero (in weight space)
Inductive Bias

Networks of reasonable size can only realize a fraction of all possible functions. Efficiency does not explain why this fraction is effective.

Why are these functions interesting?

\[ \mathcal{H}_A \subset \mathcal{H}_B \subset \text{all func} \]

Why are these functions interesting?
Inductive Bias

Networks of reasonable size can only realize a fraction of all possible functions.
Efficiency does not explain why this fraction is effective.

Why are these functions interesting?

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

1. Expressiveness

2. Expressiveness of Convolutional Networks – Questions

3. Convolutional Arithmetic Circuits

4. Efficiency of Depth (Cohen+Sharir+Shashua@COLT'16, Cohen+Shashua@ICML’16)

5. Inductive Bias of Pooling Geometry (Cohen+Shashua@ICLR’17)

6. Efficiency of Overlapping Operations (Sharir+Shashua@arXiv’17)

7. Efficiency of Interconnectivity (Cohen+Tamari+Shashua@arXiv’17)
Longstanding conjecture, proven for MLP:

depth networks are efficient w.r.t. shallow ones
Longstanding conjecture, proven for MLP:

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

Q: Can this be proven for ConvNets?
Efficiency of Depth

Longstanding conjecture, proven for MLP:

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

Q: Can this be proven for ConvNets?

Q: Is their efficiency of depth complete?
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
ConvNets typically employ square conv/pool windows

Recently, dilated windows have also become popular

**Q:** What is the inductive bias of conv/pool window geometry?
ConvNets typically employ square conv/pool windows

Recently, dilated windows have also become popular

**Q:** What is the inductive bias of conv/pool window geometry?

**Q:** Can the geometries be tailored for a given task?
Modern ConvNets employ both overlapping and non-overlapping conv/pool operations.
Modern ConvNets employ both overlapping and non-overlapping conv/pool operations.

Q: Do overlapping operations introduce efficiency?
Nearly all state of the art ConvNets employ elaborate connectivity schemes.
Nearly all state of the art ConvNets employ elaborate connectivity schemes.

Q: Can this be justified in terms of efficiency?
Outline

1. Expressiveness
2. Expressiveness of Convolutional Networks – Questions
3. Convolutional Arithmetic Circuits
4. Efficiency of Depth \((\text{Cohen+Sharir+Shashua@COLT’16, Cohen+Shashua@ICML’16})\)
5. Inductive Bias of Pooling Geometry \((\text{Cohen+Shashua@ICLR’17})\)
6. Efficiency of Overlapping Operations \((\text{Sharir+Shashua@arXiv’17})\)
7. Efficiency of Interconnectivity \((\text{Cohen+Tamari+Shashua@arXiv’17})\)
To address raised questions, we consider a surrogate (special case) of ConvNets – **Convolutional Arithmetic Circuits (ConvACs)**.
To address raised questions, we consider a surrogate (special case) of ConvNets – **Convolutional Arithmetic Circuits (ConvACs)**

ConvACs are equivalent to **hierarchical tensor decompositions**, allowing theoretical analysis w/mathematical tools from various fields, e.g.:

- Functional Analysis
- Measure Theory
- Matrix Algebra
- Graph Theory
To address raised questions, we consider a surrogate (special case) of ConvNets – **Convolutional Arithmetic Circuits (ConvACs)**

ConvACs are equivalent to **hierarchical tensor decompositions**, allowing theoretical analysis w/mathematical tools from various fields, e.g.:  
- Functional Analysis  
- Measure Theory  
- Matrix Algebra  
- Graph Theory

ConvACs are superior to ReLU ConvNets in terms of expressiveness$^1$; deliver promising results in practice:
- Excel in computationally constrained settings$^2$  
- Classify optimally under missing data$^3$

---

$^1$ *Convolutional Rectifier Networks as Generalized Tensor Decompositions, ICML'16*  
$^2$ *Deep SimNets, CVPR'16*  
$^3$ *Tractable Generative Convolutional Arithmetic Circuits, arXiv'17*
Baseline ConvAC architecture:

- **2D ConvNet**
- **Linear activation** ($\sigma(z) = z$), **product pooling** ($P\{c_j\} = \prod_j c_j$)
- **1 × 1 convolution windows**
- **Non-overlapping pooling windows**
Grid Tensors

ConvNets realize func over many local structures:

\[ f(x_1, x_2, \ldots, x_N) \]

\( x_i \) – image patches (2D network) / sequence samples (1D network)
ConvNets realize func over many local structures:

\[ f(x_1, x_2, \ldots, x_N) \]

\( x_i \) – image patches (2D network) / sequence samples (1D network)

\( f(\cdot) \) may be studied by discretizing each \( x_i \) into one of \( \{v^{(1)}, \ldots, v^{(M)}\} \):

\[ A_{d_1 \ldots d_N} = f(v^{(d_1)} \ldots v^{(d_N)}) \quad , d_1 \ldots d_N \in \{1, \ldots, M\} \]
ConvNets realize func over many local structures:

\[ f(x_1, x_2, \ldots, x_N) \]

\( x_i \) – image patches (2D network) / sequence samples (1D network)

\( f(\cdot) \) may be studied by discretizing each \( x_i \) into one of \( \{v^{(1)}, \ldots, v^{(M)}\} \):

\[ A_{d_1 \ldots d_N} = f(v^{(d_1)} \ldots v^{(d_N)}) \text{, } d_1 \ldots d_N \in \{1, \ldots, M\} \]

The lookup table \( A \) is:

\- an \( N \)-dim array (tensor) w/length \( M \) in each axis
\- referred to as the grid tensor of \( f(\cdot) \)
High-dim tensors (arrays) are exponentially large – cannot be used directly

May be represented and manipulated via **tensor decompositions**:

- Compact algebraic parameterizations
- Generalization of low-rank matrix decompositions
Hierarchical tensor decompositions represent high-dim tensors by incrementally generating intermediate tensors of increasing dimension.

Generation process can be described by a tree over tensor modes (axes).
Observation

Grid tensors of func realized by ConvACs are given by hierarchical tensor decompositions:

\[
\begin{align*}
\text{network structure} & \quad \leftrightarrow \quad \text{decomposition type} \\
& \quad \quad \text{(depth, width, pooling etc)} \quad \leftrightarrow \quad \text{(dim tree, internal ranks etc)} \\
\text{network weights} & \quad \leftrightarrow \quad \text{decomposition parameters}
\end{align*}
\]
Observation

Grid tensors of func realized by ConvACs are given by hierarchical tensor decompositions:

<table>
<thead>
<tr>
<th>network structure</th>
<th>decomposition type</th>
</tr>
</thead>
<tbody>
<tr>
<td>(depth, width, pooling etc)</td>
<td>(dim tree, internal ranks etc)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>network weights</th>
<th>decomposition parameters</th>
</tr>
</thead>
</table>

We can study networks through corresponding decompositions!
Example 1: Shallow Network $\leftrightarrow$ CP Decomposition

Shallow network (single hidden layer, global pooling):

\[ \text{rep}(i,d) = f_{\theta_d}(x_i) \]

\[ \text{conv}(j,\gamma) = \langle a^{0,j,\gamma}, \text{rep}(j,:)) \rangle \]

\[ \text{pool}(\gamma) = \prod_{j \text{ covers space}} \text{conv}(j,\gamma) \]

\[ \text{out}(y) = \langle a^{1,1,\gamma}, \text{pool}(:) \rangle \]

corresponds to classic **CP decomposition**:

\[ \mathcal{A}^y = \sum_{\gamma=1}^{r_0} a^{1,1,\gamma} \cdot a^{0,1,\gamma} \otimes a^{0,2,\gamma} \otimes \ldots \otimes a^{0,N,\gamma} \]

(\( \otimes \) – outer product)
Example 2: Deep Network $\leftrightarrow$ HT Decomposition

Deep network with size-2 pooling:

![Diagram of a deep network with size-2 pooling](image)

corresponds to **Hierarchical Tucker (HT) decomposition**: 

$$
\phi_{1,j,\gamma} = \sum_{\alpha=1}^{r_0} a_{1,j,\gamma}^{\alpha} \cdot a_{0,2j-1,\alpha}^{0,2j,\alpha} \\
\ldots
$$

$$
\phi_{l,j,\gamma} = \sum_{\alpha=1}^{r_{l-1}} a_{l,j,\gamma}^{\alpha} \cdot \phi_{l-1,2j-1,\alpha}^{l-1,2j,\alpha} \\
\ldots
$$

$$
A_{y} = \sum_{\alpha=1}^{r_{L-1}} a_{L,1,y}^{\alpha} \cdot \phi_{L-1,1,\alpha}^{L-1,2,\alpha} \\
\phi_{L,\gamma} = \prod_{j \in \{2^{L-1},2^L,\ldots,2^N\}} \phi_{l,j,\gamma}^{L-1,2j,\alpha} \\
\text{out}(y) = \langle a_{L,1,y}^{L,1,\alpha}, \phi_{L-1,1,\alpha}^{L-1,2,\alpha} \rangle
$$
Outline

1. Expressiveness

2. Expressiveness of Convolutional Networks – Questions

3. Convolutional Arithmetic Circuits

4. **Efficiency of Depth** *(Cohen+Sharir+Shashua@COLT'16, Cohen+Shashua@ICML'16)*

5. Inductive Bias of Pooling Geometry *(Cohen+Shashua@ICLR'17)*

6. Efficiency of Overlapping Operations *(Sharir+Shashua@arXiv'17)*

7. Efficiency of Interconnectivity *(Cohen+Tamari+Shashua@arXiv'17)*
Tensor Matricization

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

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

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

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

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

- Arrangement of $A$ as matrix
- Rows correspond to modes (axes) indexed by $I$
- Cols...
Exponential & Complete Efficiency of Depth

Claim

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

Tensors generated by CP decomposition w/r\textsubscript{0} terms, when matricized under any partition \((I, J)\), have rank \(r\textsubscript{0}\) or less

Theorem

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

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!
### Claim

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

### Theorem

*Consider the partition \(I_{\text{odd}} = \{1, 3, \ldots, N - 1\}\), \(J_{\text{even}} = \{2, 4, \ldots, N\}\). Besides a set of measure zero, all param settings of HT decomposition give tensors that when matricized w.r.t. \((I_{\text{odd}}, J_{\text{even}})\), 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 $r_0$ 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!
From Convolutional Arithmetic Circuits to Convolutional Rectifier Networks

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

- **linear activation** $\rightarrow$ **ReLU activation**: $\sigma(z) = \max\{z, 0\}$
- **product pooling** $\rightarrow$ **max/average pooling**: $P\{c_j\} = \max\{c_j\}/\text{mean}\{c_j\}$

Most successful deep learning architecture to date!
ConvACs correspond to tensor decompositions based on tensor product $\otimes$:

$$(\mathcal{A} \otimes \mathcal{B})_{d_1,...,d_{P+Q}} = \mathcal{A}_{d_1,...,d_P} \cdot \mathcal{B}_{d_{P+1},...,d_{P+Q}}$$
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 $g : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, 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

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

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

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

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

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

Generalized tensor decompositions are obtained by replacing $\otimes$ with $\otimes_g$
Convolutional Rectifier Networks $\iff$ Generalized Tensor Decompositions

Define the **activation-pooling operator**: 

$$
\rho_{\sigma/P}(a, b) := P\{\sigma(a), \sigma(b)\}
$$
Convolutional Rectifier Networks $\longleftrightarrow$ Generalized Tensor Decompositions

Define the **activation-pooling operator**:

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

Grid tensors of func realized by R-ConvNets are given by generalized tensor decompositions $w/g(\cdot) \equiv \rho_{\sigma/P}(\cdot)$:

- **Shallow R-ConvNet** $\longleftrightarrow$ **Generalized CP decomposition**
  
  \[ w/g(\cdot) \equiv \rho_{\sigma/P}(\cdot) \]

- **Deep R-ConvNet** $\longleftrightarrow$ **Generalized HT decomposition**
  
  \[ w/g(\cdot) \equiv \rho_{\sigma/P}(\cdot) \]
By analyzing matricization ranks of tensors realized by generalized CP and HT decompositions $w/g(\cdot) \equiv \frac{\rho_{\sigma}}{P(\cdot)}$, 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!
Exponential But Incomplete Efficiency of Depth

By analyzing matricization ranks of tensors realized by generalized CP and HT decompositions with $g(\cdot) \equiv \rho_\sigma / p(\cdot)$, 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!

Developing optimization methods for ConvACs may give rise to an arch that is provably superior but has so far been overlooked
Outline

1. Expressiveness
2. Expressiveness of Convolutional Networks – Questions
3. Convolutional Arithmetic Circuits
4. Efficiency of Depth  \((\text{Cohen+Sharir+Shashua@COLT'16, Cohen+Shashua@ICML'16})\)
5. Inductive Bias of Pooling Geometry  \((\text{Cohen+Shashua@ICLR'17})\)
6. Efficiency of Overlapping Operations  \((\text{Sharir+Shashua@arXiv'17})\)
7. Efficiency of Interconnectivity  \((\text{Cohen+Tamari+Shashua@arXiv'17})\)
Separation Rank – A Measure of Input Correlations

ConvNets realize func over many local structures:

\[ f(x_1, x_2, \ldots, x_N) \]

\( x_i \) – image patches (2D network) / sequence samples (1D network)
ConvNets realize func over many local structures:

\[ f(x_1, x_2, \ldots, x_N) \]

\( x_i \) – image patches (2D network) / sequence samples (1D network)

Important feature of \( f(\cdot) \) – **correlations** it models between the \( x_i \)'s
ConvNets realize func over many local structures:

\[ f(x_1, x_2, \ldots, x_N) \]

\( x_i \) – image patches (2D network) / sequence samples (1D network)

Important feature of \( f(\cdot) \) – **correlations** it models between the \( x_i \)’s

**Separation rank:**

Formal measure of these correlations

Sep rank of \( f(\cdot) \) w.r.t. input partition \((I, J)\) measures dist from separability

\[ \text{sep rank } \uparrow \implies \text{more correlation between } (x_i)_{i \in I} \text{ and } (x_j)_{j \in J} \]
Deep Networks Favor Some Correlations Over Others

Claim

\[ \text{W/ConvAC sep rank w.r.t } (I, J) \text{ is equal to rank of } [A^y]_{I,J} \text{ – grid tensor matricized w.r.t. } (I, J) \]
Deep Networks Favor Some Correlations Over Others

**Claim**

\[ W/\text{ConvAC} \text{ sep rank w.r.t } (I, J) \text{ is equal to rank of } \left[ A^Y \right]_{I,J} \text{ – grid tensor matricized w.r.t. } (I, J) \]

**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

\[ W/\text{ConvAC sep rank } w.r.t. (I, J) \text{ is equal to rank of } \left[ A^Y \right]_{I,J} - \text{grid tensor matricized } w.r.t. (I, J) \]

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

contiguous pooling

local correlations

alternative pooling

alternative correlations

Nadav Cohen (Hebrew U.) Expressiveness of Convolutional Networks Science of Intelligence, 2017 34 / 44
Outline

1. Expressiveness

2. Expressiveness of Convolutional Networks – Questions

3. Convolutional Arithmetic Circuits

4. Efficiency of Depth (Cohen+Sharir+Shashua@COLT'16, Cohen+Shashua@ICML'16)

5. Inductive Bias of Pooling Geometry (Cohen+Shashua@ICLR’17)

6. Efficiency of Overlapping Operations (Sharir+Shashua@arXiv’17)

7. Efficiency of Interconnectivity (Cohen+Tamari+Shashua@arXiv’17)
Baseline ConvAC arch has non-overlapping conv and pool windows:

\[
\text{rep}(i, d) = f_0^d(x_i)
\]

\[
\text{conv}_0(j, \gamma) = \langle a^{0, \gamma}, \text{rep}(j, \cdot) \rangle
\]

\[
\text{pool}_0(j, \gamma) = \prod_{j' \in \text{window } j} \text{conv}_0(j', \gamma)
\]

\[
\text{pool}_{L-1}(\gamma) = \prod_{j' \text{ covers space}} \text{conv}_{L-1}(j', \gamma)
\]

\[
\text{out}(y) = \langle a^{L, y}, \text{pool}_{L-1}(\cdot) \rangle
\]
Overlapping Operations

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

\[
\text{rep}(i,d) = f_{\theta_d}(x_i), \quad \text{conv}_0(j,\gamma) = \{a^{0,\gamma}, \text{rep}(j,:)\}, \quad \text{pool}_0(j,\gamma) = \prod_{j' \in \text{window } j} \text{conv}_0(j',\gamma)
\]

Replace those by (possibly) overlapping generalized convolution:

\[
\text{GCO}(x^{(j)}, w^{(c)}, b^{(c)}) = \prod_{k=1}^{R^2} \left( b^{(c)}_k + \sum_{d=1}^{D} w^{(c)}_{dk} x^{(j)}_{dk} \right)
\]
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 $2 \times 2$ "pool":

\[
\begin{align*}
\text{input } X & \quad \text{representation} & \quad \text{GC} & \quad \text{GC} \\
N & \quad N & \quad M & \quad M \\
\end{align*}
\]

\[
\begin{align*}
\text{Output} & \quad \text{GC} & \quad \text{GC} & \quad \text{GC} \\
K & \quad M & \quad M \\
\end{align*}
\]

\[
\begin{align*}
\text{input } X & \quad \text{representation} & \quad \text{BBB GC} & \quad 2x2 \text{ GC} & \quad \text{BBB GC} \\
N & \quad N & \quad M & \quad R_0 & \quad R_0 \\
\end{align*}
\]

\[
\begin{align*}
\text{Block } 0 & \quad \text{Block } L-1 & \quad \text{dense (output)} \\
\end{align*}
\]

W/ConvACs overlaps lead to exponential efficiency!
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 $2 \times 2$ “pool”:

W/ConvACs overlaps lead to exponential efficiency!
Outline

1. Expressiveness
2. Expressiveness of Convolutional Networks – Questions
3. Convolutional Arithmetic Circuits
4. Efficiency of Depth  \((\text{Cohen}+\text{Sharir}+\text{Shashua}@\text{COLT'16, Cohen}+\text{Shashua}@\text{ICML'16})\)
5. Inductive Bias of Pooling Geometry  \((\text{Cohen}+\text{Shashua}@\text{ICLR'17})\)
6. Efficiency of Overlapping Operations  \((\text{Sharir}+\text{Shashua}@\text{arXiv'17})\)
7. Efficiency of Interconnectivity  \((\text{Cohen}+\text{Tamari}+\text{Shashua}@\text{arXiv'17})\)
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!
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
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.
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!
**Conclusion**

- **Expressiveness** – the driving force behind deep networks
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
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
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

- Fundamental tool underlying all of our analyses:
  \[\text{ConvNets} \leftrightarrow \text{hierarchical tensor decompositions}\]
Thank You