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

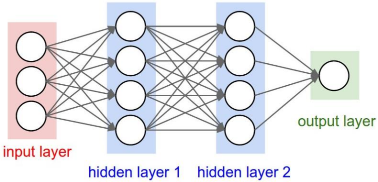
Ronen Tamari

Yoav Levine

David Yakira

Classic vs. State of the Art Deep Learning

Classic



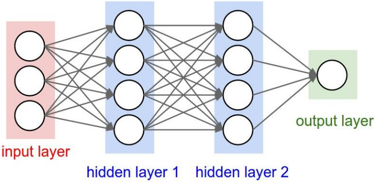
Multilayer Perceptron (MLP)

Architectural choices:

- depth
- layer widths
- activation types

Classic vs. State of the Art Deep Learning

Classic

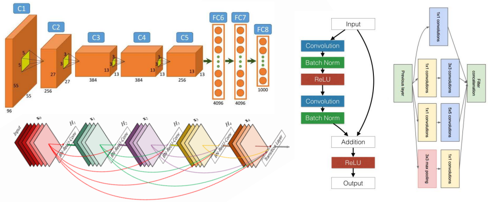


Multilayer Perceptron (MLP)

Architectural choices:

- depth
- layer widths
- activation types

State of the Art



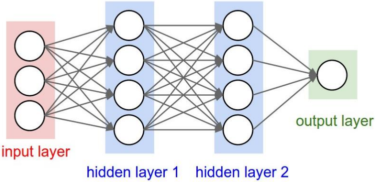
Convolutional Networks (ConvNets)

Architectural choices:

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

Classic vs. State of the Art Deep Learning

Classic

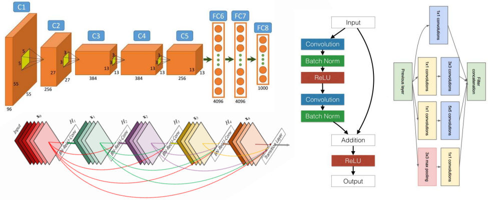


Multilayer Perceptron (MLP)

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

Expressiveness:

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

Expressiveness

Expressiveness:

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

Fundamental theoretical questions:

- What kind of func can different network arch represent?
- Why are these func suitable for real-world tasks?
- What is the representational benefit of depth?
- Can other arch features deliver representational benefits?

Efficiency

Expressive efficiency compares network arch in terms of their ability to compactly represent func

Efficiency

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

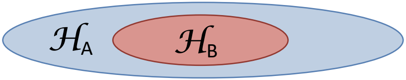
Efficiency

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 $\mathcal{H}_A$



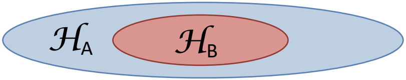
Efficiency

Expressive efficiency compares network arch in terms of their ability to compactly represent func

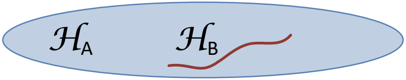
Let:

- $\mathcal{H}_A$ – 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 $r_A \in \mathcal{O}(r_B)$

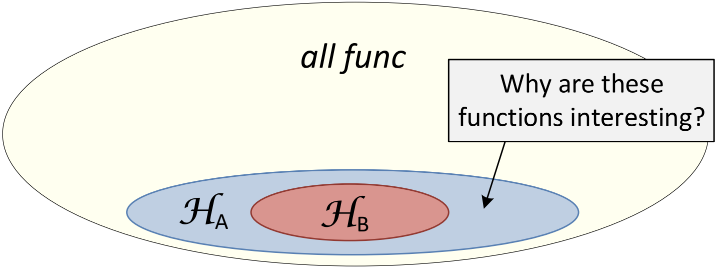
(2) $\exists$func realized by A w/size r_A requiring B to have size $r_B \in \Omega(f(r_A))$, where $f(\cdot)$ is super-linear

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 func

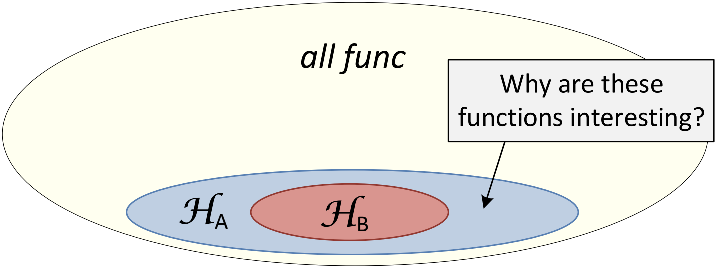
Efficiency does not explain why this fraction is effective



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**:

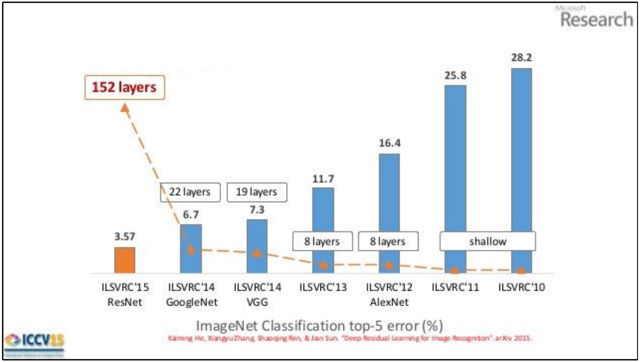
- Not all func are equally useful for a given task
- Network only needs to represent useful func

Outline

Efficiency of Depth

Longstanding conjecture, proven for MLP:

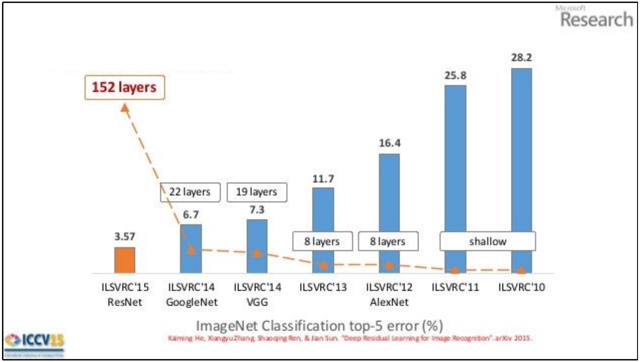
deep networks are efficient w.r.t. shallow ones



Efficiency of Depth

Longstanding conjecture, proven for MLP:

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

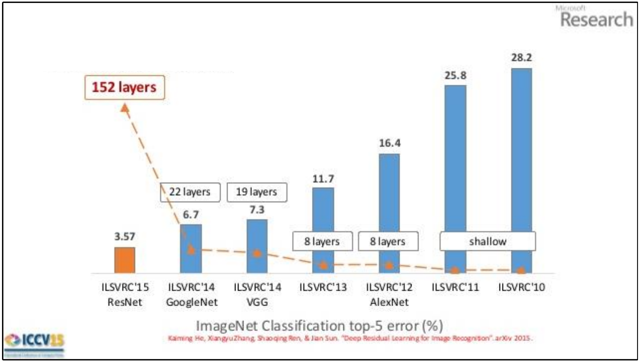


Q: *Can this be proven for ConvNets?*

Efficiency of Depth

Longstanding conjecture, proven for MLP:

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

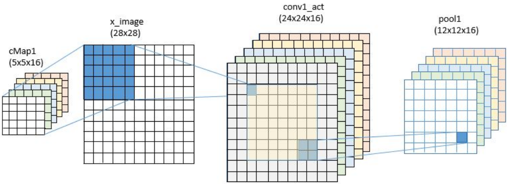


Q: *Can this be proven for ConvNets?*

Q: *Is their efficiency of depth complete?*

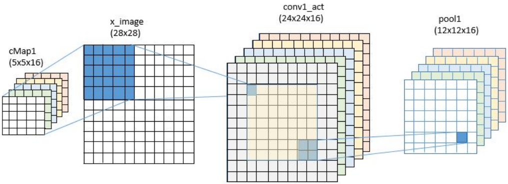
Inductive Bias of Convolution/Pooling Geometry

ConvNets typically employ square conv/pool windows

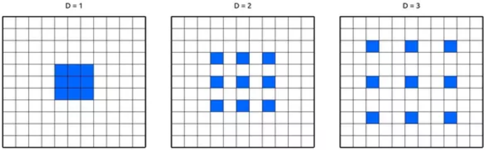


Inductive Bias of Convolution/Pooling Geometry

ConvNets typically employ square conv/pool windows

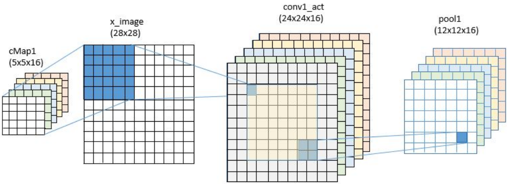


Recently, dilated windows have also become popular

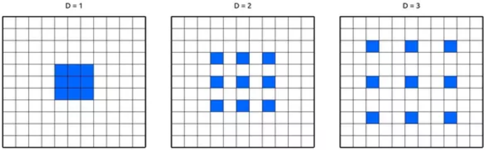


Inductive Bias of Convolution/Pooling Geometry

ConvNets typically employ square conv/pool windows



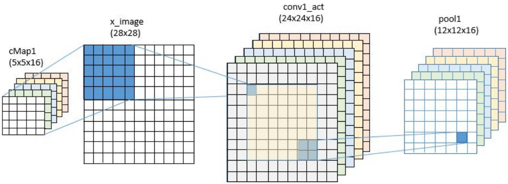
Recently, dilated windows have also become popular



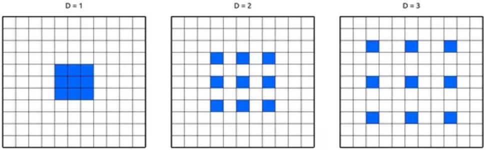
Q: *What is the inductive bias of conv/pool window geometry?*

Inductive Bias of Convolution/Pooling Geometry

ConvNets typically employ square conv/pool windows



Recently, dilated windows have also become popular

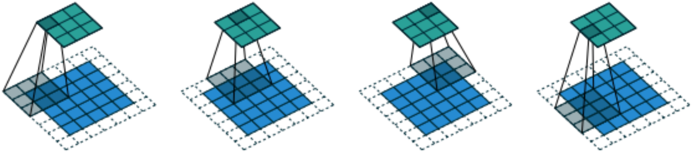


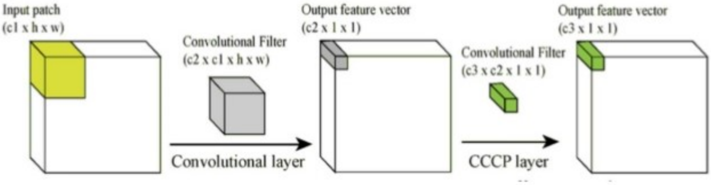
Q: *What is the inductive bias of conv/pool window geometry?*

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

Efficiency of Overlapping Operations

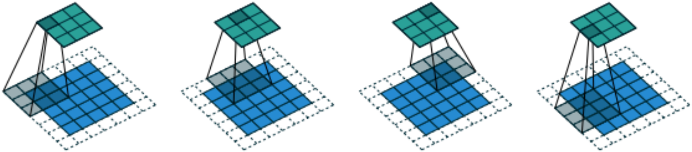
Modern ConvNets employ both overlapping and non-overlapping conv/pool operations

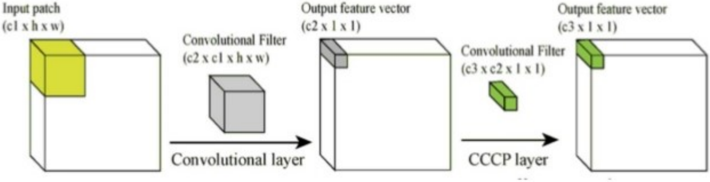




Efficiency of Overlapping Operations

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

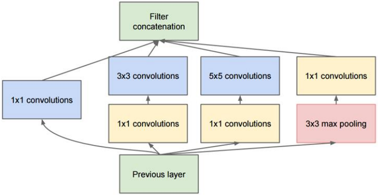




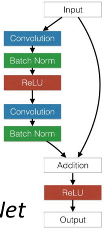
Q: *Do overlapping operations introduce efficiency?*

Efficiency of Connectivity Schemes

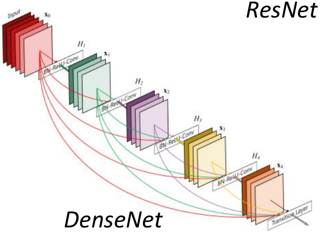
Nearly all state of the art ConvNets employ elaborate connectivity schemes



Inception (GoogLeNet)



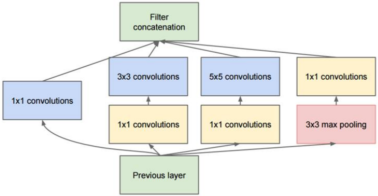
ResNet



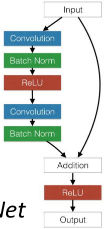
DenseNet

Efficiency of Connectivity Schemes

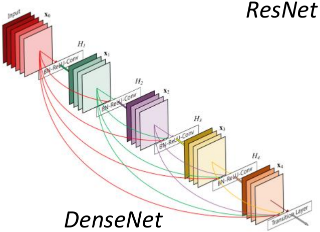
Nearly all state of the art ConvNets employ elaborate connectivity schemes



Inception (GoogLeNet)



ResNet



DenseNet

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

Outline

Convolutional Arithmetic Circuits

To address raised questions, we consider a surrogate (special case) of ConvNets – **Convolutional Arithmetic Circuits (ConvACs)**

Convolutional Arithmetic Circuits

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

Convolutional Arithmetic Circuits

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]

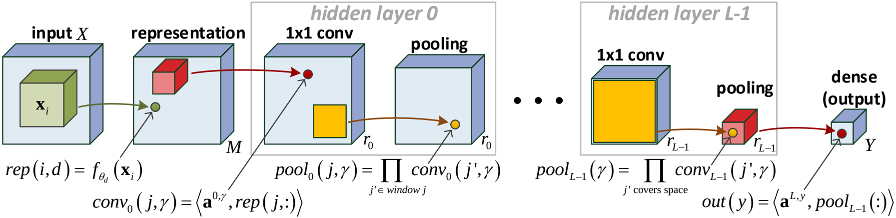
---

[1] *Convolutional Rectifier Networks as Generalized Tensor Decompositions, ICML'16*

[2] *Deep SimNets, CVPR'16*

[3] *Tractable Generative Convolutional Arithmetic Circuits, arXiv'17*

Baseline Architecture



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:

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

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

Grid Tensors

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)

$f(\cdot)$ may be studied by *discretizing* each $\mathbf{x}_i$ into one of $\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(M)}\}$:

$$\mathcal{A}_{d_1 \ldots d_N} = f(\mathbf{v}^{(d_1)} \ldots \mathbf{v}^{(d_N)}) \quad , d_1 \ldots d_N \in \{1, \ldots, M\}$$

Grid Tensors

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)

$f(\cdot)$ may be studied by *discretizing* each $\mathbf{x}_i$ into one of $\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(M)}\}$:

$$\mathcal{A}_{d_1 \ldots d_N} = f(\mathbf{v}^{(d_1)} \ldots \mathbf{v}^{(d_N)}) \quad , d_1 \ldots d_N \in \{1, \ldots, M\}$$

The lookup table $\mathcal{A}$ is:

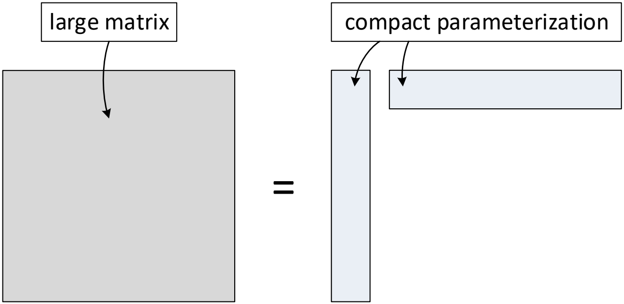
- an N-dim array (tensor) w/length M in each axis
- referred to as the **grid tensor** of $f(\cdot)$

Tensor Decompositions – Compact Parameterizations

High-dim tensors (arrays) are exponentially large – cannot be used directly

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

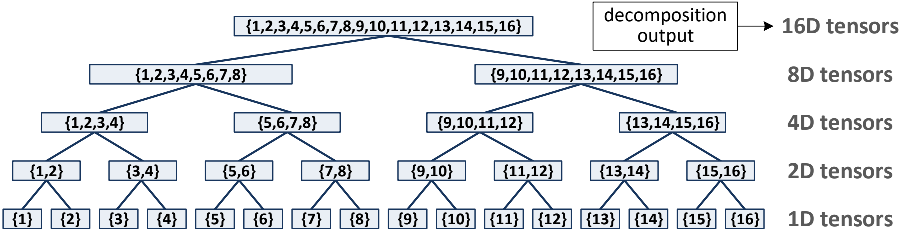
- Compact algebraic parameterizations
- Generalization of low-rank matrix decompositions



Hierarchical Tensor Decompositions

Hierarchical tensor decompositions represent high-dim tensors by incrementally generating intermediate tensors of increasing dim

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



Convolutional Arithmetic Circuits $\longleftrightarrow$ Hierarchical Tensor Decompositions

Observation

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

network structure (depth, width, pooling etc)	$\longleftrightarrow$	decomposition type (dim tree, internal ranks etc)
network weights	$\longleftrightarrow$	decomposition parameters

Convolutional Arithmetic Circuits $\longleftrightarrow$ Hierarchical Tensor Decompositions

Observation

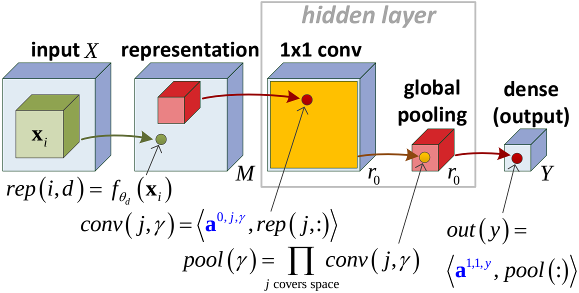
Grid tensors of func realized by ConvACs are given by hierarchical tensor decompositions:

network structure (depth, width, pooling etc)	$\longleftrightarrow$	decomposition type (dim tree, internal ranks etc)
network weights	$\longleftrightarrow$	decomposition parameters

We can study networks through corresponding decompositions!

Example 1: Shallow Network $\longleftrightarrow$ CP Decomposition

Shallow network (single hidden layer, global pooling):



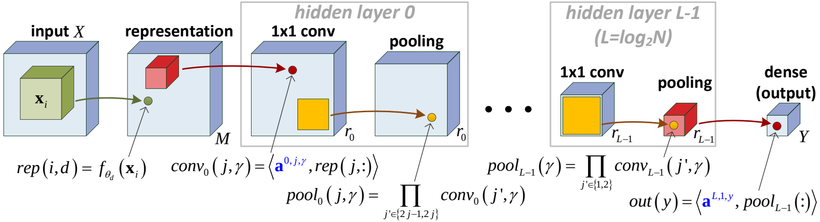
corresponds to classic **CP decomposition**:

$$\mathcal{A}^y = \sum_{\gamma=1}^{r_0} a_\gamma^{1,1,y} \cdot \mathbf{a}^{0,1,\gamma} \otimes \mathbf{a}^{0,2,\gamma} \otimes \cdots \otimes \mathbf{a}^{0,N,\gamma}$$

($\otimes$ – outer product)

Example 2: Deep Network $\longleftrightarrow$ HT Decomposition

Deep network with size-2 pooling:



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

$$\phi^{1,j,\gamma} = \sum_{\alpha=1}^{r_0} a_\alpha^{1,j,\gamma} \cdot \mathbf{a}^{0,2j-1,\alpha} \otimes \mathbf{a}^{0,2j,\alpha}$$

$$\cdots$$

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

$$\cdots$$

$$\mathcal{A}^y = \sum_{\alpha=1}^{r_{L-1}} a_\alpha^{L,1,y} \cdot \phi^{L-1,1,\alpha} \otimes \phi^{L-1,2,\alpha}$$

Outline

Tensor Matricization

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

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

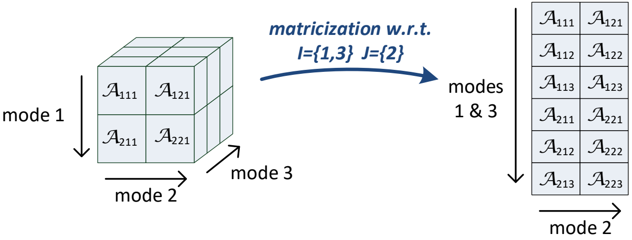
Tensor Matricization

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.

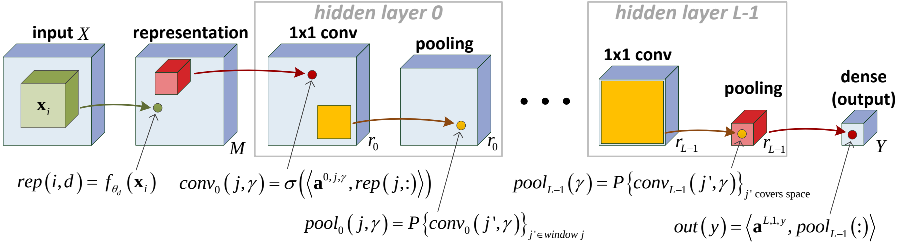
Since # of terms in CP decomposition corresponds to # of hidden channels in shallow ConvAC:

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!

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: $P\{c_j\} = max\{c_j\}/mean\{c_j\}$

Most successful deep learning architecture to date!

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

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**:

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

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/ $g(\cdot) \equiv \rho_{\sigma/P}(\cdot)$

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!

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!

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

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

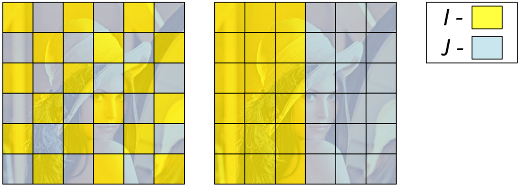
ConvNets realize func over many local structures:

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

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

Theorem

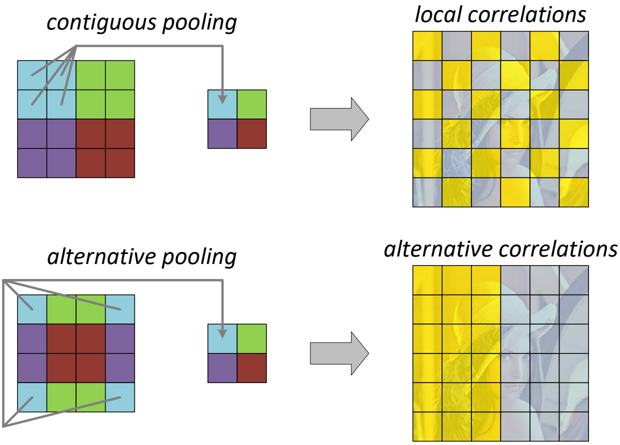
Maximal rank of tensor generated by HT decomposition, when matricized w.r.t. (I, J), is:

- *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 Controls the Preference

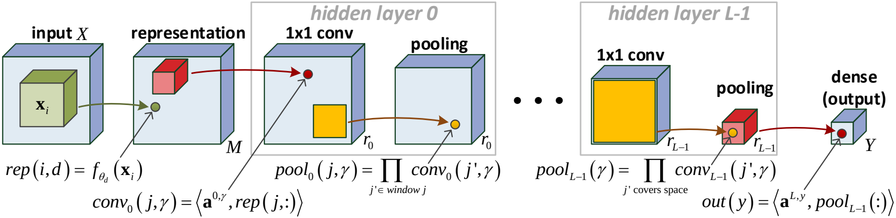


Pooling geometry of deep ConvAC determines which partitions are favored – controls the correlation profile (inductive bias)!

Outline

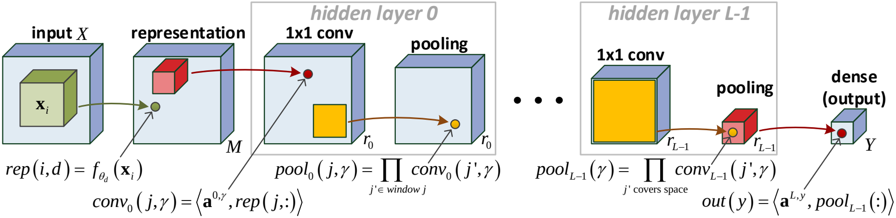
Overlapping Operations

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

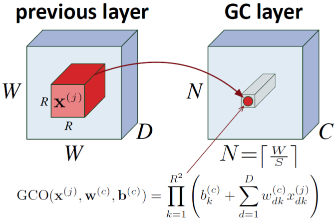


Overlapping Operations

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



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



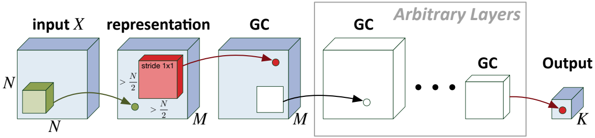
Exponential Efficiency

Theorem

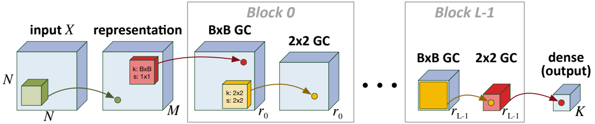
Various ConvACs w/overlapping GC layers realize func requiring ConvAC w/no overlaps to be exponentially large

Examples

- Network starts with large receptive field:



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



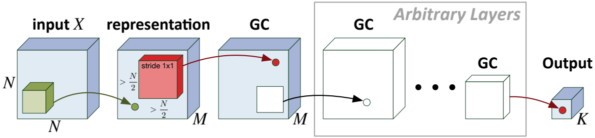
Exponential Efficiency

Theorem

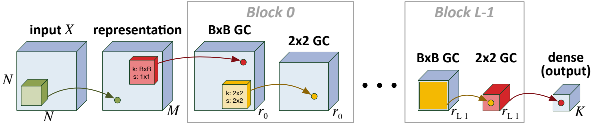
Various ConvACs w/overlapping GC layers realize func requiring ConvAC w/no overlaps to be exponentially large

Examples

- Network starts with large receptive field:



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



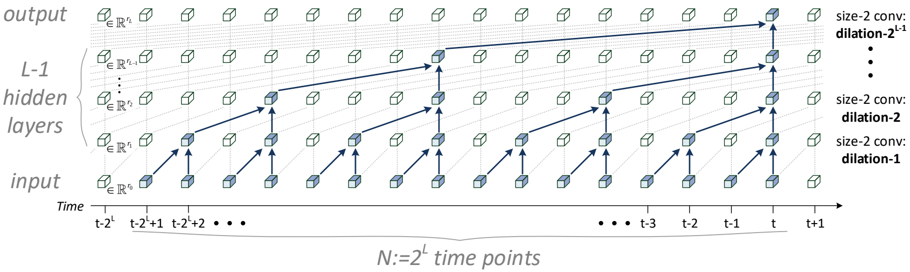
W/ConvACs overlaps lead to exponential efficiency!

Outline

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)*

Dilated Convolutional Networks

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

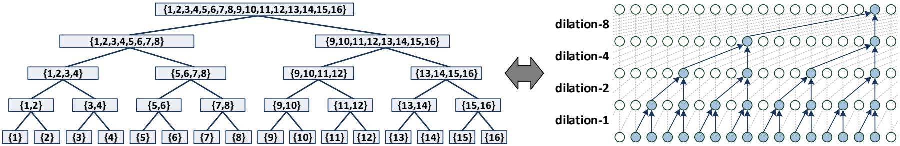


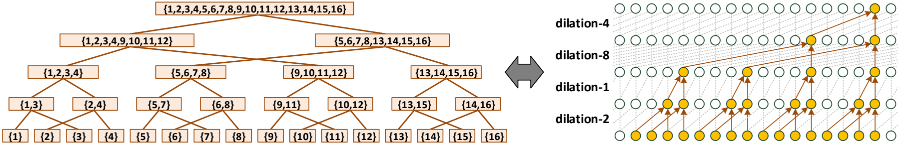
- 1D ConvNets (sequence data)
- Dilated (gapped) conv windows
- No pooling

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

Mixing Tensor Decompositions ⟶ 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

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:

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

Thank You