Convolutional Rectifier Networks as Generalized Tensor Decompositions

Nadav Cohen   Amnon Shashua

The Hebrew University of Jerusalem

International Conference on Machine Learning (ICML) 2016

Convolutional Rectifier Networks vs. Convolutional Arithmetic Circuits

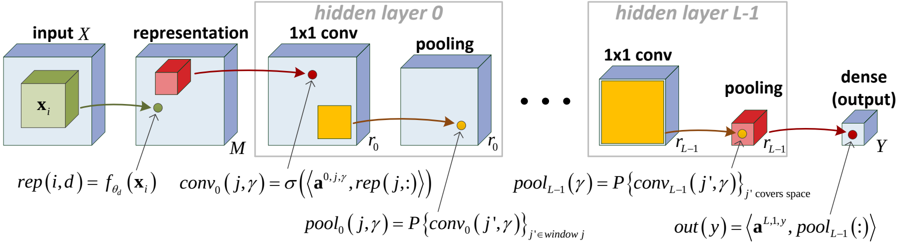
Convolutional rectifier networks	Convolutional arithmetic circuits
ConvNets with ReLU activation and max or average pooling	ConvNets with linear activation and product pooling
Most successful deep learning architecture to date	Equivalent to SimNets [1], but not as widely used
Expressive power yet to be analyzed	Expressive power analyzed through tensor decompositions [2]

We analyze the expressive power of convolutional rectifier networks by generalizing tensor decompositions

[1] *Deep SimNets, CVPR'16*

[2] *On the Expressive Power of Deep Learning: A Tensor Analysis, COLT'16*

Analyzed ConvNet Architecture



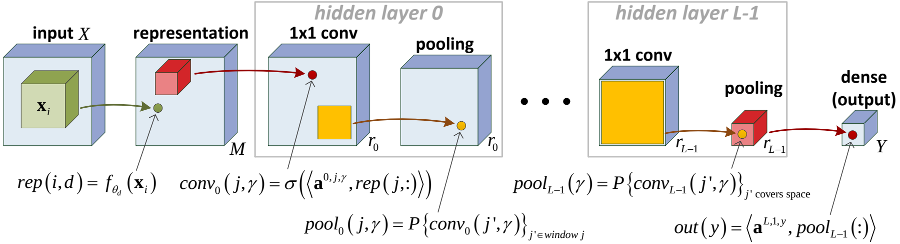
Convolutional network:

- locality
- sharing (optional)
- pooling

$\boxed{\sigma(\cdot) \text{ -- point-wise activation}}$

$\boxed{P\{\cdot\} \text{ -- pooling operator}}$

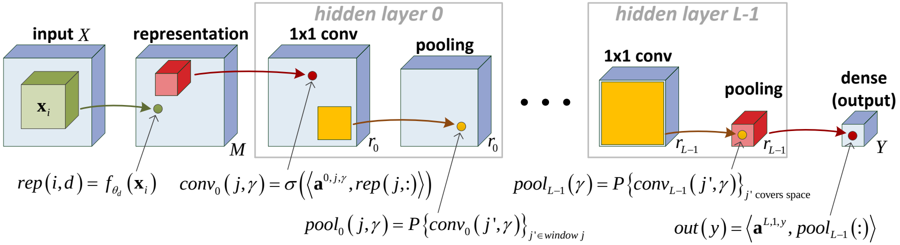
Activation and Pooling



Three configurations for activation $\sigma(\cdot)$ and pooling $P\{\cdot\}$:

Activation	Pooling	
linear $\sigma(z) = z$	**product** $P\{c_j\} = \prod c_j$	***convolutional arithmetic circuits***
ReLU $\sigma(z) = [z]_+$	**max** $P\{c_j\} = \max\{c_j\}$	***convolutional rectifier networks***
	average $P\{c_j\} = \text{mean}\{c_j\}$	

Activation | Pooling



$\boxed{\sigma(z) = z}$ $\boxed{P\{c_j\} = \prod c_j}$

Function realized by output y:

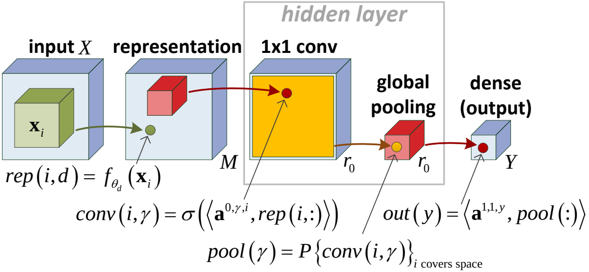
$$h_y(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \sum_{d_1 \ldots d_N = 1}^{M} \mathcal{A}^y_{d_1,\ldots,d_N} \prod_{i=1}^{N} f_{\theta_{d_i}}(\mathbf{x}_i)$$

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

[1] *On the Expressive Power of Deep Learning: A Tensor Analysis, COLT'16*

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

Shallow network (single hidden layer, global pooling):



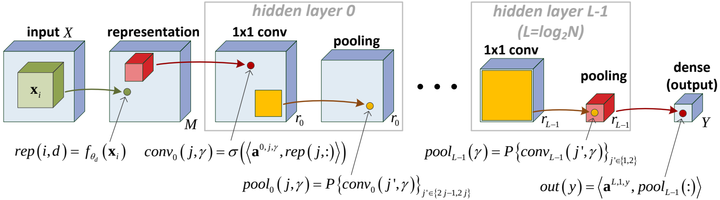
$\boxed{\sigma(z) = z}$ $\boxed{P\{c_j\} = \prod c_j}$

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

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

Deep Convolutional Arithmetic Circuit $\longleftrightarrow$ Hierarchical Tucker Decomposition

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



$\boxed{\sigma(z) = z}$ $\boxed{P\{c_j\} = \prod c_j}$

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

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

Convolutional Arithmetic Circuits: Expressive Power [1]

Universality:
When a network can realize any function given unlimited size

Depth efficiency:
When a function realized by polynomially sized deep network requires shallow networks to have super-polynomial size

Complete depth efficiency:
When all but a negligible (zero measure) set of the functions realizable by a deep network are depth efficient

Claim

Convolutional arithmetic circuits are universal

Theorem

Convolutional arithmetic circuits exhibit complete depth efficiency

[1] *On the Expressive Power of Deep Learning: A Tensor Analysis, COLT'16*

Generalized Tensor Decompositions

Convolutional arithmetic circuits 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 associative and commutative operator $g : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, the **generalized tensor product** $\otimes_g$ is defined by:

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

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

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

Generalized Tensor Decompositions $\longrightarrow$ Convolutional Rectifier Networks

Define the **activation-pooling operator**:

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

If $\rho_{\sigma/P}$ is associative and commutative:

Generalized CP decomposition with $\otimes_{\rho_{\sigma/P}}$	$\longleftrightarrow$	*Shallow ConvNet with activation* $\sigma(\cdot)$ *and pooling* $P\{\cdot\}$
Generalized Hierarchical Tucker decomposition with $\otimes_{\rho_{\sigma/P}}$	$\longleftrightarrow$	*Deep ConvNet with activation* $\sigma(\cdot)$ *and pooling* $P\{\cdot\}$

Example

Convolutional rectifier network with max pooling:

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

Meets the associativity and commutativity requirements

Convolutional Rectifier Networks: Expressive Power

Universality:

Claim

Convolutional rectifier networks are universal with max pooling, but not with average pooling

Depth efficiency:

Claim

Convolutional rectifier networks realize depth efficient functions

Claim

Convolutional rectifier networks do ***not*** *exhibit complete depth efficiency*

Conclusion

Generalized tensor decompositions relate convolutional rectifier networks to convolutional arithmetic circuits, opening door to various mathematical tools

We analyze the expressive power of convolutional rectifier networks:

- Universality holds with max pooling, but not with average pooling
- Depth efficiency exists, but is not complete

	convolutional rectifier networks	*convolutional arithmetic circuits*
expressive power	*incomplete depth efficiency*	*complete depth efficiency*
optimization methods	*well studied and developed*	*addressed only recently* [1]

Developing optimization methods for convolutional arithmetic circuits may give rise to an architecture that is provably superior but has so far been overlooked

[1] *Deep SimNets, CVPR'16*

Thank You