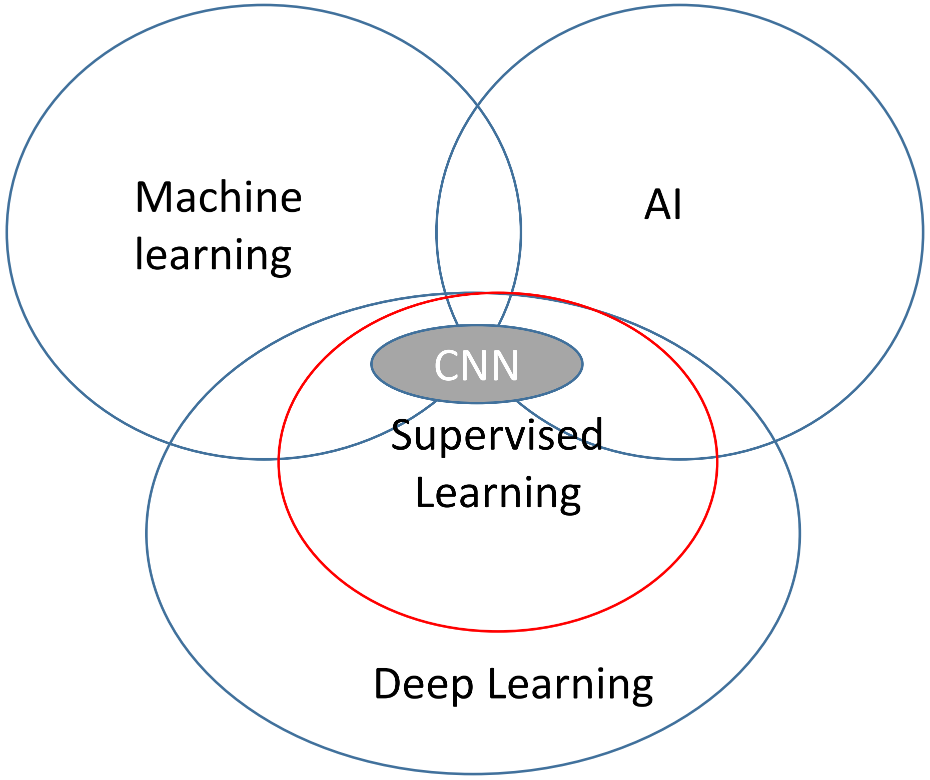
On the Expressive Power of Deep Learning: A Tensor Analysis

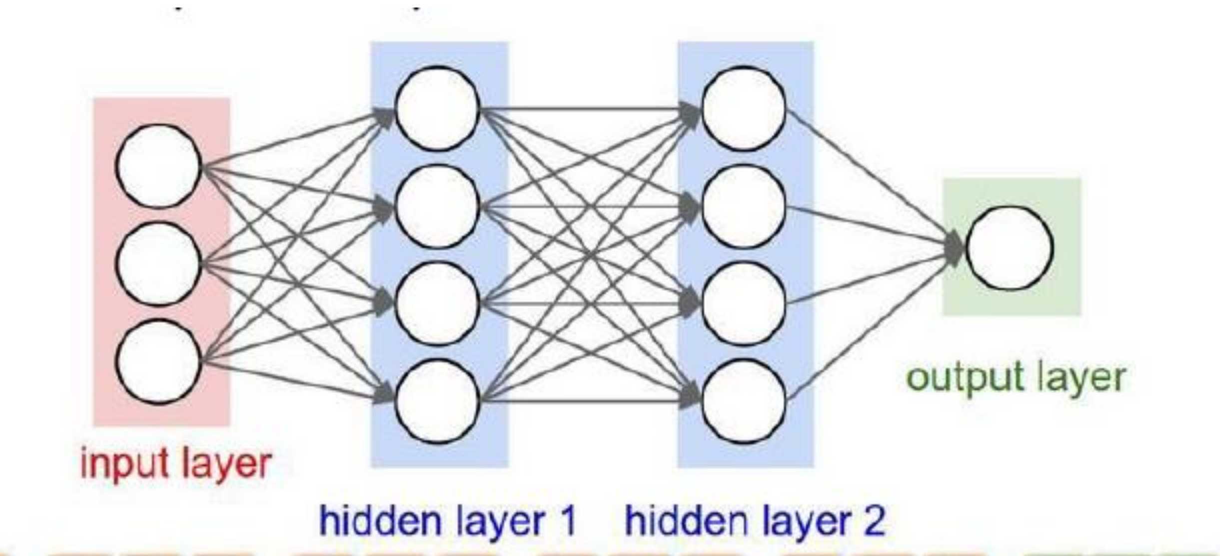
Benyou wang

Cohen N, Sharir O, Shashua A. On the expressive power of deep learning: A tensor analysis[C]//Conference on Learning Theory. 2016: 698-728.

Deep Learning

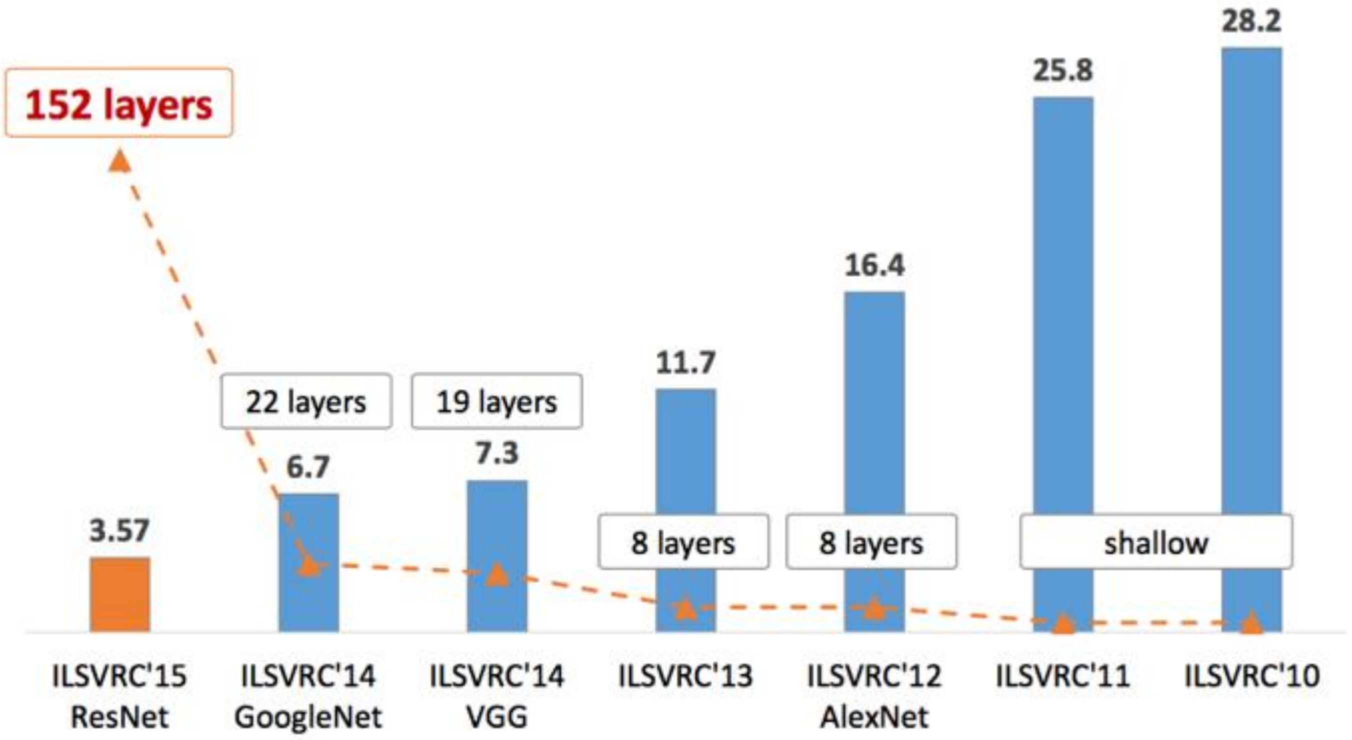


Deep and shadow Learning



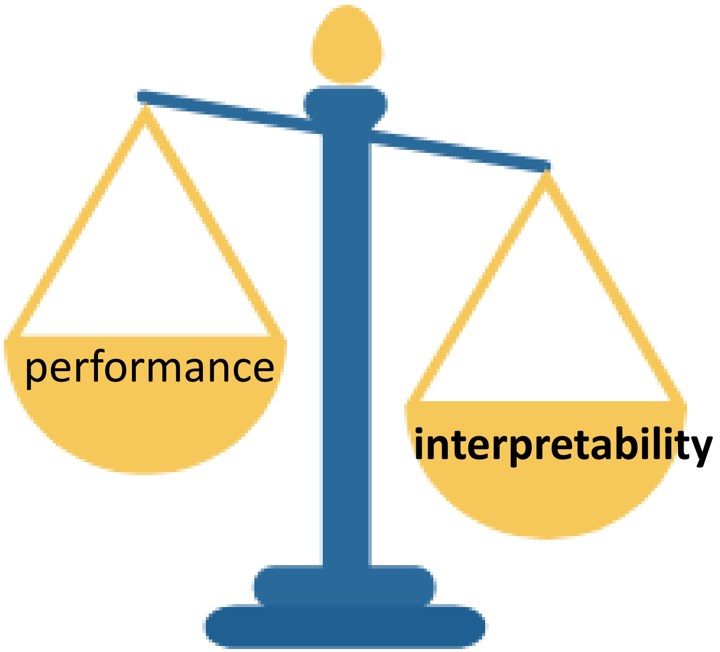
CNN, one typical NN of DL

Expressive power of depth – the driving force behind Deep Learning



Empirically, the deeper, the better?

Not only performance, but also **interpretability**



The ImageNet challenge ended in 2017. http://image-net.org/challenges/LSVRC/2017/

Questions

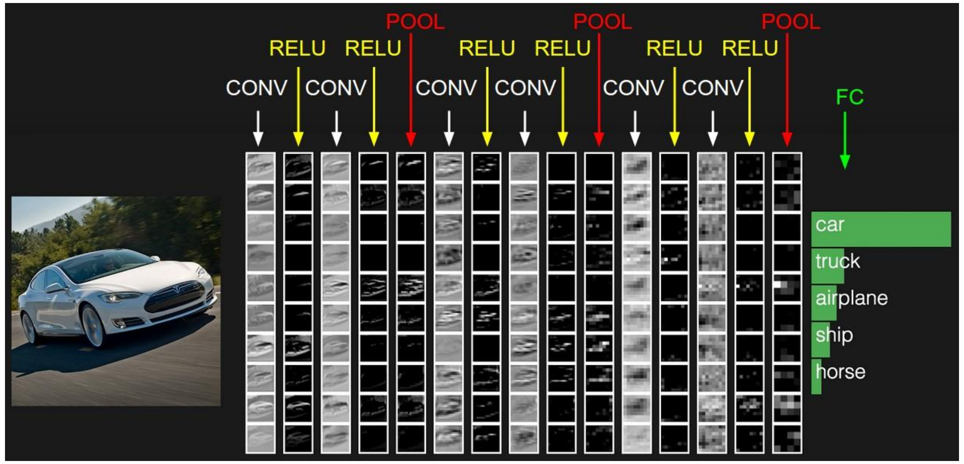
Why the deeper,
the better?

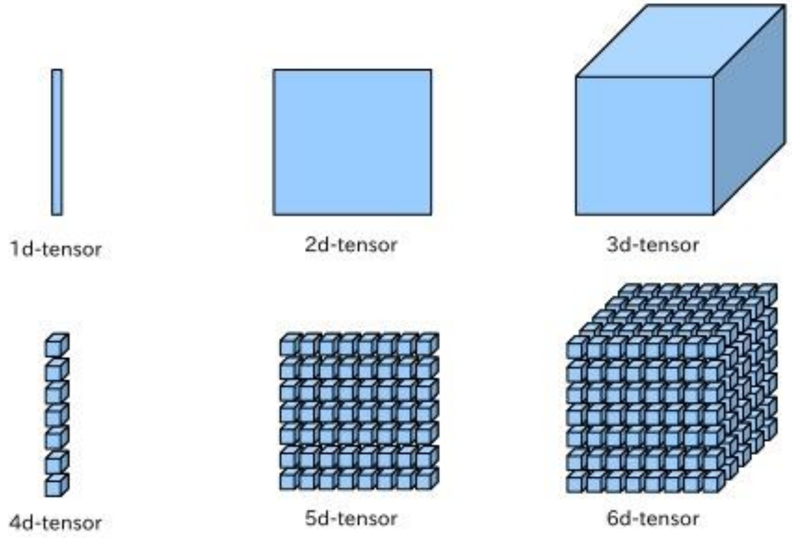
Expressive power of DL with a tensor analysis

- Link CNN to Tensor Decomposition
    - Shadow CNN
    - Deep CNN

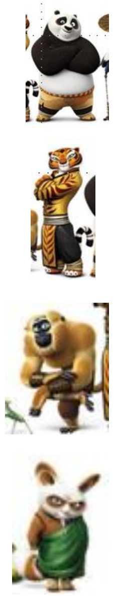
- Theorem of Network Capacity

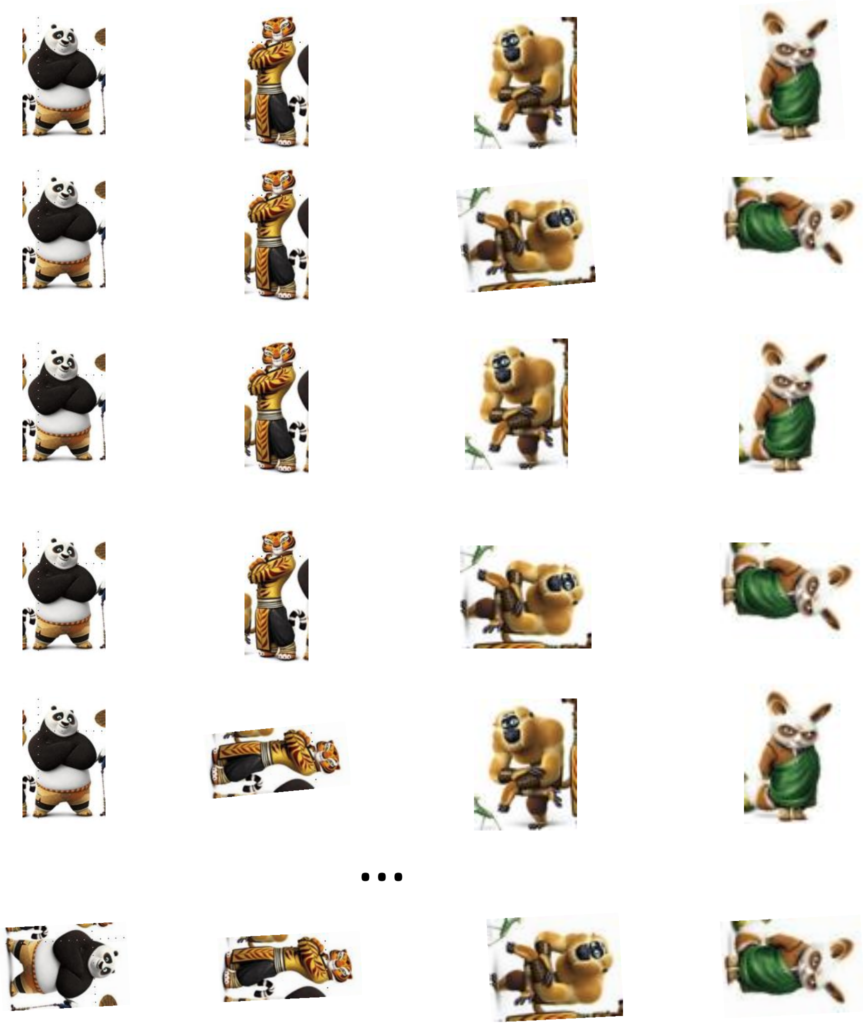
CNN and Tensor Decomposition





New hypotheses





$\lambda_{0000} f_0(x_0) f_0(x_1) f_0(x_2) f_0(x_3)$

$\lambda_{0001} f_0(x_1) f_0(x_1) f_0(x_2) f_1(x_3)$

$\lambda_{0010} f_0(x_0) f_0(x_1) f_1(x_2) f_0(x_3)$

$\lambda_{0011} f_0(x_0) f_0(x_1) f_1(x_2) f_1(x_3)$

$\lambda_{0100} f_0(x_0) f_1(x_1) f_0(x_2) f_0(x_3)$

...

$\lambda_{1111} f_1(x_0) f_1(x_1) f_1(x_2) f_1(x_3)$

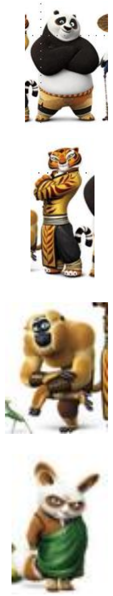
Examples of two representation functions, $f_1, f_2 : R^s \rightarrow R$
Natural choices for this family may be radial basis function(Gaussians)

$$h_y(X) = h_y(x_1, x_2, \ldots, x_N)$$
$$= \sum \boldsymbol{\lambda_{d_1 d_2,\ldots,d_n}} \prod_{i=1}^{N} f_{\theta_{di}}(x_i)$$

Representation layer

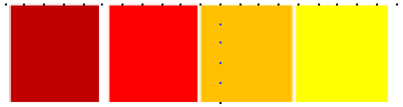
A **tensor** with M^N elements

- $h_y(X) = h_y(x_1, x_2, \dots, x_N) = \sum \lambda_{d_1 d_2, \dots, d_n} \prod_{i=1}^{N} f_{\theta_{di}}(x_i)$

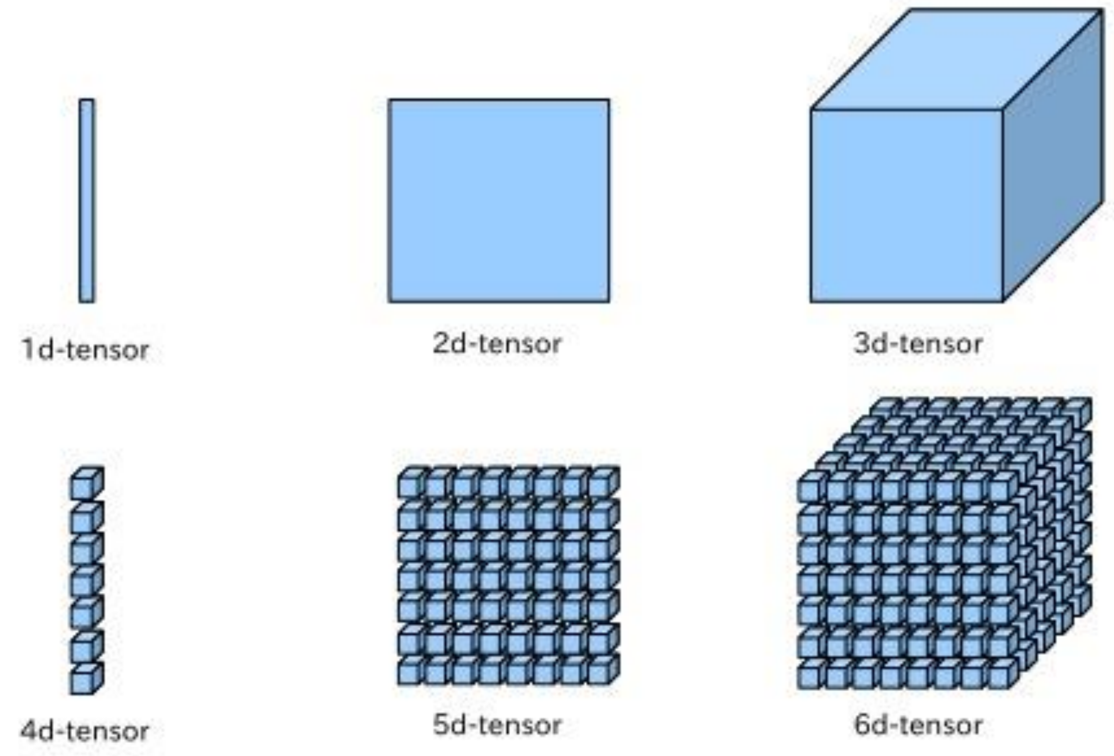


	f_1	f_2	…	f_m
x_1				
x_2				
…				
x_N				

A **tensor** with M^N *coefficients of* $\prod_{i=1}^{N} f_{\theta_{di}}(x_i)$



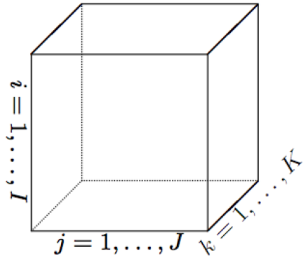
Tensor



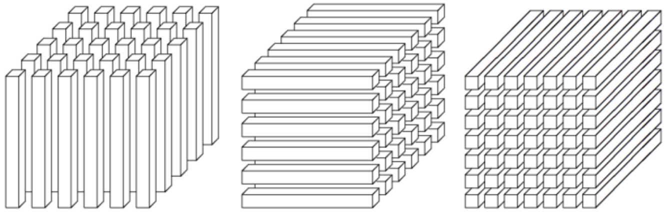
Tensor

Tensor is high-dimensional array $A \in R^{M_1 \times M_2 \times \ldots M_N}$

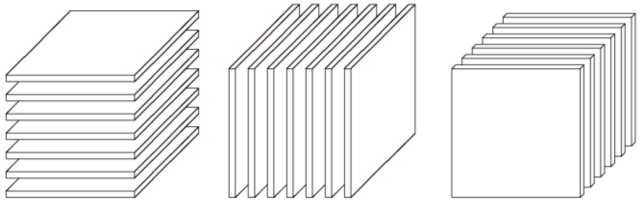
With a index/location $\{d_1, d_2, \ldots d_N\} \in I$ we can get an element $\lambda_{d_1 d_2 \ldots d_n}$ where $d_1 \in [M_1], d_N \in [M\_n]$



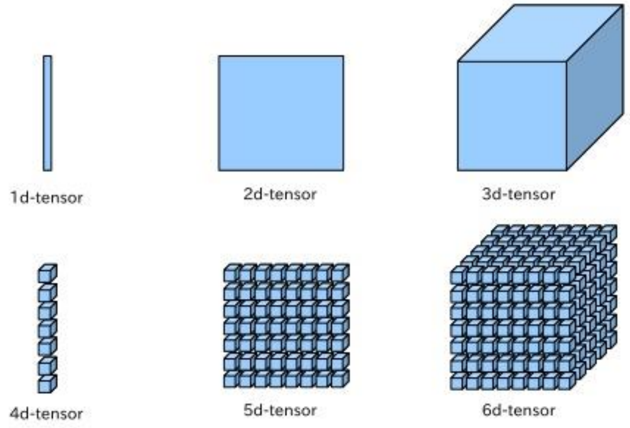
Fiber is high-dimensional analogue of column/row in matrix, for a 3-dimentional tensor, they are $A_{:,i_2,i_3}$, $A_{i_1,:,i_3}$ and $A_{i_1,i_2,:,}$



Slice is high-dimensional sections of a tensor, for a 3-dimentional tensor, they are $A_{i_1,:,:}$, $A_{:,i_2,:}$, and$A_{:,:,i_3}$



Tensor

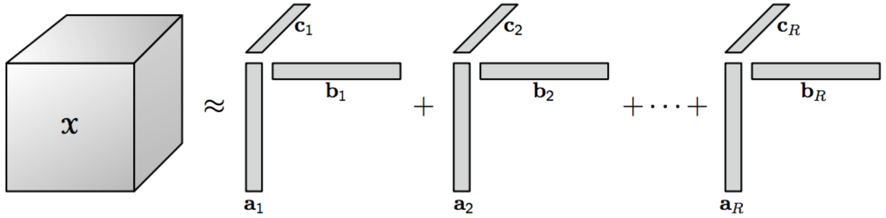


Matricization of A w.r.t the partition (I, J), i.e. I and J are disjoint subsets of [N] whose union is [N], where $I = \{i_1, i_2, \ldots, i_{|I|}\}, i_1 < i_2 < \cdots < i_{|I|}$ and similarly J $= \{j_1, j_2, \ldots, j_{|I|}\}, j_1 < j_2 < \cdots < j_{|J|}$ is denoted as $[[A]]_{I,J}$, which is a $\prod_{t=1}^{|I|} M_{i_t} - by - \prod_{t=1}^{|J|} M_{j_t}$ matrix holding the entries of A such that $\lambda_{d_1 d_2 \ldots d_n}$ is placed in row index $1 + \sum_{t-1}^{|I|}(d_{i_t} - 1) \prod_{t'=t+1}^{|I|} M_{i'_t}$ and column index $1 + \sum_{t-1}^{|J|}(d_{j_t} - 1) \prod_{t'=t+1}^{|J|} M_{j'_t}$

Tensor product (also named Kronecker product for matrix), denoted by $\otimes$**,** for example**,** $A \in R^{M_1 \times \ldots M_P}$ and $B \in R^{M_{P+1} \times \cdots \times M_{P+Q}}$**,** Order P and Q resp**.**
The tensor product between **A** and **B** is $\mathrm{A} \otimes B \in R^{M_1 \times \ldots M_{P+Q}}$,

Tensor Decomposition

CP Decomposition



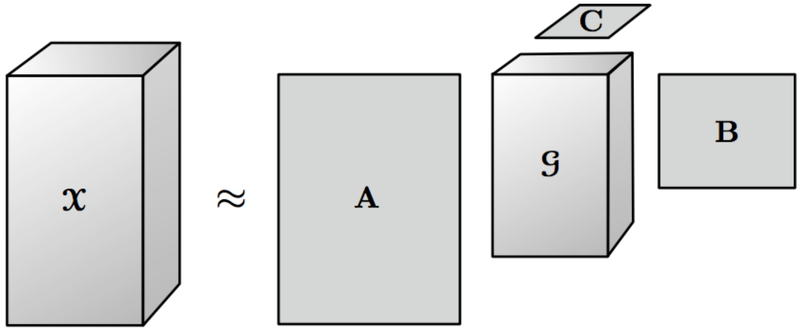
order

$$A = \sum_{z=1}^{Z} \boxed{v_z^{(1)} \otimes \cdots \otimes v_z^{(N)}}$$

The rank-one tensor is pure or elementary

Any tensor can be expressed as a sum of rank-1 tensors

Tucker Decomposition



Hierarchical Tucker Decomposition

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

$$\cdots$$

$$\phi^{l,j,\gamma} = \sum_{\alpha=1}^{r_{l-1}} a_\alpha^{l,j,\gamma} \underbrace{\phi^{l-1,2j-1,\alpha}}_{\text{order } 2^{l-1}} \otimes \underbrace{\phi^{l-1,2j,\alpha}}_{\text{order } 2^{l-1}}$$

$$\cdots$$

$$\phi^{L-1,j,\gamma} = \sum_{\alpha=1}^{r_{L-2}} a_\alpha^{L-1,j,\gamma} \underbrace{\phi^{L-2,2j-1,\alpha}}_{\text{order } \frac{N}{4}} \otimes \underbrace{\phi^{L-2,2j,\alpha}}_{\text{order } \frac{N}{4}}$$

$$\mathcal{A}^y = \sum_{\alpha=1}^{r_{L-1}} a_\alpha^{L,y} \underbrace{\phi^{L-1,1,\alpha}}_{\text{order } \frac{N}{2}} \otimes \underbrace{\phi^{L-1,2,\alpha}}_{\text{order } \frac{N}{2}}$$

The Tensor in the hypotheses

- $h_y(X) = h_y(x_1, x_2, \dots, x_N) = \sum \boldsymbol{\lambda_{d_1 d_2, \dots, d_n}} \prod_{i=1}^{N} f_{\theta_{di}}(x_i)$

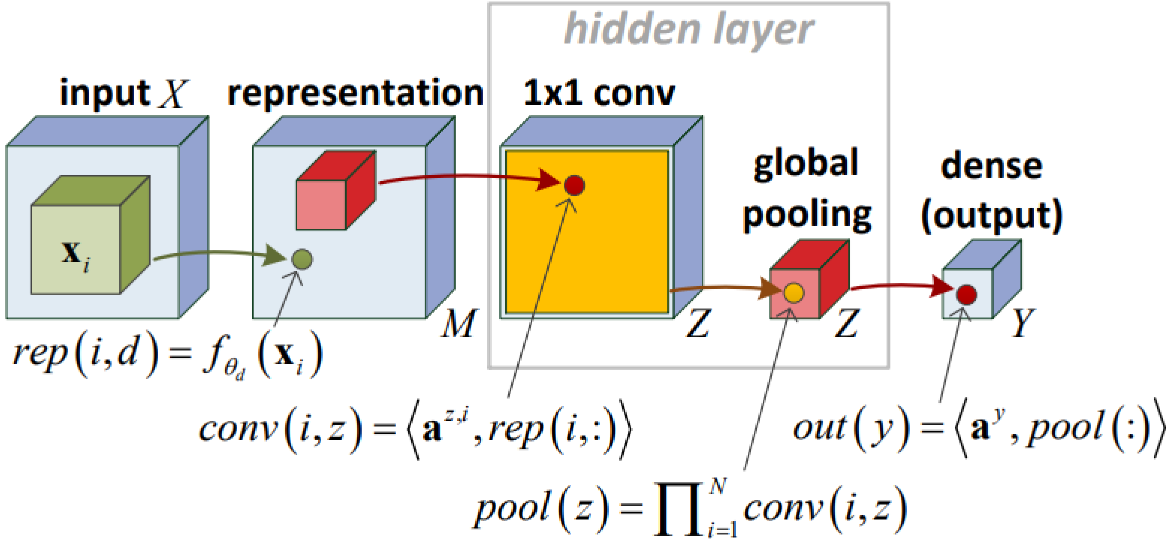
- A = $\{\boldsymbol{\lambda_{d_1 d_2, \dots, d_n}}\}_{d_1, d_2, \dots, d_n = 1}^{M} \in R^{M \times M \times \cdots M}, i.e.\, R^{M^N}$.

- Such exponential tensor is not easy to be learned or computed
- Thus we need to **decompose** the tensor.

Shallow CNN vs. CP Decomposition

With CP decomposition

$$A = \sum_{z=1}^{Z} \lambda_y^z \boldsymbol{a}^{z,1} \otimes \cdots \otimes \boldsymbol{a}^{z,N}$$



- $h_y(X) = \sum \lambda_{d_1 d_2,\ldots,d_n} \prod_{i=1}^{N} f_{\theta\, di}(x_i) = \sum_{z=1}^{Z} \lambda_y^z \prod_{i=1}^{N} \left(\sum_{d=1}^{M} a_d^{z,i}\, f_{\theta_d}(x_i) \right)$

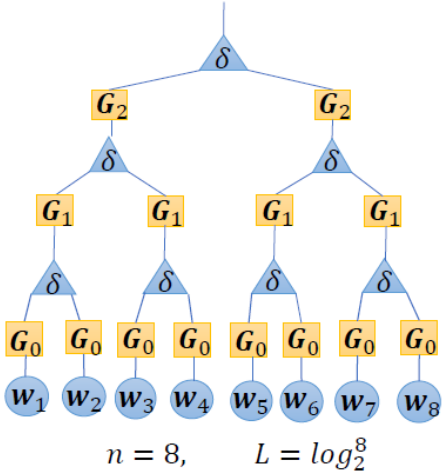
Convolution

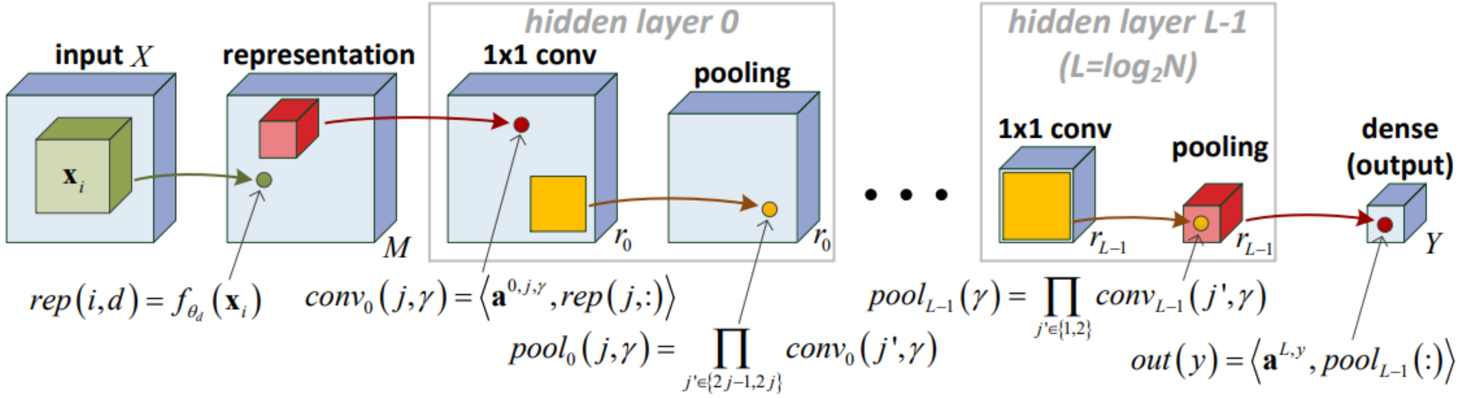
Pooling (product pooling)

Multiple channels

Deep CNN vs. HT Decomposition

L = log2 N hidden layers, non-overlap convolution, size-2 pooling windows





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

$$\cdots$$

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

$$\cdots$$

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

In the case of **Shared** weights

For CP model, coefficient sharing amounts to setting $\mathbf{a}^z := \mathbf{a}^{z,1} = \cdots = \mathbf{a}^{z,N}$ in the CP decomposition (eq. 3), transforming the latter to a symmetric CP decomposition:

$$\mathcal{A}^y = \sum_{z=1}^{Z} a_z^y \cdot \underbrace{\mathbf{a}^z \otimes \cdots \otimes \mathbf{a}^z}_{N \text{ times}} \quad , \mathbf{a}^z \in \mathbb{R}^M, \mathbf{a}^y \in \mathbb{R}^Z$$

CP model with sharing is not universal (not all tensors $\mathcal{A}^y$ are representable, no matter how large Z is allowed to be) – it can only represent symmetric tensors.

Core Theory

Besides a negligible (zero measure) set, all functions that can be realized by a **deep** network of **polynomial** size, require **exponential** size in order to be realized, or even approximated, by a **shallow** network

Proof Sketch

- $[\![\mathcal{A}]\!]$ – arrangement of tensor $\mathcal{A}$ as matrix (*matricization*)
- $\odot$ – Kronecker product for matrices. Holds: $rank(A \odot B) = rank(A) \cdot rank(B)$
- Relation between tensor and Kronecker products: $[\![\mathcal{A} \otimes \mathcal{B}]\!] = [\![\mathcal{A}]\!] \odot [\![\mathcal{B}]\!]$
- 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"

Shadow CNN & CP compostion

$$A = \sum_{z=1}^{Z} \boxed{v_z^{(1)} \otimes \cdots \otimes v_z^{(N)}}$$

$Rank\ [v_z^{(1)} \otimes \cdots \otimes v_z^{(N)}]$ = 1

Matricization is a linear operation

$$rank\left[\sum_{z=1}^{Z} \lambda_z \mathbf{v}_1^{(z)} \otimes \cdots \otimes \mathbf{v}_{2L}^{(z)}\right] = rank \sum_{z=1}^{Z} \lambda_z \left[\mathbf{v}_1^{(z)} \otimes \cdots \otimes \mathbf{v}_{2L}^{(z)}\right]$$

$$\leq \sum_{z=1}^{Z} rank\left[\mathbf{v}_1^{(z)} \otimes \cdots \otimes \mathbf{v}_{2L}^{(z)}\right] = Z$$

Deep CNN &

$\phi^{1,j,\gamma} \in R^M$, *thus* $\mathrm{rank}([\phi^{1,j,\gamma}]) = \min(r_0, M)$, almost everywhere

(if $\left\{a^{0,2j-1,\alpha} \otimes a^{0,2j,\alpha}\right\}_{j=1}^{2^{l-1}}$ are linearly independent)

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

$$\cdots$$

$$\phi^{l,j,\gamma} = \sum_{\alpha=1}^{r_{l-1}} a_\alpha^{l,j,\gamma} \boxed{\underbrace{\phi^{l-1,2j-1,\alpha}}_{\text{order } 2^{l-1}} \otimes \underbrace{\phi^{l-1,2j,\alpha}}_{\text{order } 2^{l-1}}}$$

$$\cdots$$

$$\phi^{L-1,j,\gamma} = \sum_{\alpha=1}^{r_{L-2}} a_\alpha^{L-1,j,\gamma} \underbrace{\phi^{L-2,2j-1,\alpha}}_{\text{order } \frac{N}{4}} \otimes \underbrace{\phi^{L-2,2j,\alpha}}_{\text{order } \frac{N}{4}}$$

$$\boxed{\mathcal{A}^y} = \sum_{\alpha=1}^{r_{L-1}} a_\alpha^{L,y} \underbrace{\phi^{L-1,1,\alpha}}_{\text{order } \frac{N}{2}} \otimes \underbrace{\phi^{L-1,2,\alpha}}_{\text{order } \frac{N}{2}}$$

Rank ($\phi^{0,2j-1,\alpha} \otimes \phi^{0,2j,\alpha}$) >= $\min(r_0, M)^{\,2}$, almost everywhere

Rank (A^y) >= $\min(r_0, M)^{\,N/2}$, almost everywhere

Publications

- Sentiment-Specific Embedding in Complex-valued Space, in process.
- Qiuchi Li*, **Benyou Wang***, Massimo Melucci. A Complex-valued Network for Matching. **NAACL 2019**
- **Benyou Wang***, Qiuchi Li*, Massimo Melucci, Dawei Song. Semantic Hilbert Space for Text Representation Learning. **WWW 2019**
- Wei Zhao*, **Benyou Wang***, Min Yang, Jianbo Ye, Zhou Zhao, Xiaojun Chen, Ying Shen.. Leveraging Long and Short-term Information in Content-aware Movie Recommendation via Adversarial Training. **IEEE Transactions on Cybernetics (TOC), 2019**