Training convolutional neural networks for biomedical data analysis: a tensor based approach

Presented by: Amina Ghrissi
PhD student at I3S Laboratory

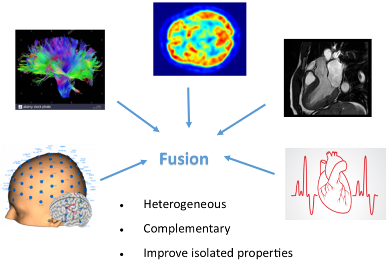
Supervised by: Prof. Vicente Zarzoso, SIS team
Prof. Johan Montagnat, SPARKS team

September 28, 2018

Outline

PhD
Multimodal analysis of Biomedical Data

- Funding: IDEX UCA$^{\text{JEDI}}$
- Project: Intégration et Analyse de Données Biomédicales (IADB)
- Mission: Extract biomarkers within workflow $\rightarrow$ D
- Multimodality:



From Multimodality to Deep Learning for Biomedical Data analysis

- Deep learning (DL) widely used in computer vision and biomedical data analysis
- Convolutional Neural Networks (CNN) success: ad hoc, empirical design
  - Deep Convolutional Neural Networks and Learning ECG Features for Screening Paroxysmal Atrial Fibrillation Patients [1]

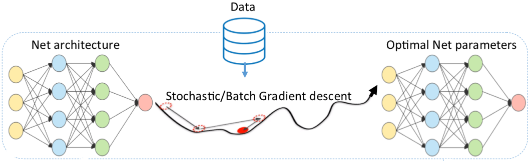
	hand-crafted	end-to-end CNN	conv+KNN	conv+SVM
CCR[2] (%)	82	85	91	87

[1] 2017, https://ieeexplore.ieee.org/document/7937819

[2] Correct classification rate

Classical training of NN

Ad hoc success of NNs: problems of training/convergence of numerical optimization algorithms $\rightarrow$ estimation $\theta = ?$

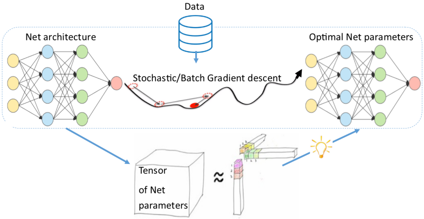


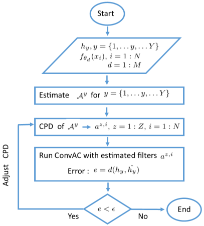
- $\exists$ Challenges in Neural Network Optimization Optimization [3]:
  - ill conditioning, local minima, long term dependencies ...

[3] [Chap 8, Deep learning book, Goodfellow, 2016]

Tensor based training of NNs

Estimate optimal neural network (NN) parameters: Tensor Decomposition (TD).





Equivalence between Convolutional Arithmetic Circuits (ConvAC) architecture and tensor factorization

💡 Estimate ConvAC parameters through tensor decomposition.

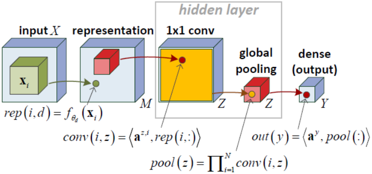
$\exists$ Equivalence between network architecture and tensor factorization:

Shallow networks (SN) $\Leftrightarrow$ CP (rank-1) decomposition

Deep network (DN) $\Leftrightarrow$ Hierarchical Tucker decomposition (HT)

🎯 Theoretical formulation of expressive power: efficiency & inductive bias [Cohen, 2015-2017]

ConvAC



- task: classifying an instance $X = (x_1, \dots, x_N)$ into categories $\mathcal{Y} := \{1, \dots Y\}$
  $x_1, \dots, x_N$ patches around pixels, $x_i \in \mathbb{R}^s$
- score function: $\{h_y\}_{y \in \mathcal{Y}}$
- representation functions: $f_{\theta_1} \dots f_{\theta_M} \colon \mathbb{R}^s \to \mathbb{R}$

? ConvAC Vs. SimNet

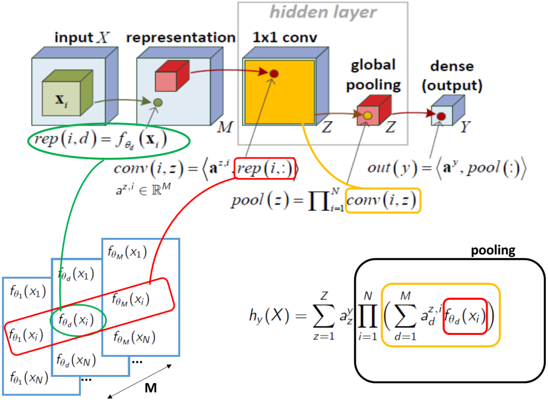
$$MEX_{\xi}\underset{i=1,\dots,n}{\{c_i\}} := \frac{1}{\xi}\log\left(\frac{1}{n}\sum_{i=1}^{n}\exp\{\xi \cdot c_i\}\right)$$

$$MEX_{\xi}\{c_i\}_{i=1}^{n} \underset{\xi \to +\infty}{\longrightarrow} \max\{c_i\}_{i=1}^{n}$$

$$MEX_{\xi}\{c_i\}_{i=1}^{n} \underset{\xi \to 0}{\longrightarrow} \text{mean}\{c_i\}_{i=1}^{n}$$

$$MEX_{\xi}\{c_i\}_{i=1}^{n} \underset{\xi \to -\infty}{\longrightarrow} \min\{c_i\}_{i=1}^{n}$$

Shallow ConvAC



[4]

[4] joint decomposition: same vectors $a^{z,i}$ shared across $\forall$ classes y.

Shallow ConvAC as a tensor

$$h_y(x_1,\dots,x_N) = \sum_{z=1}^{Z} a_z^y \prod_{i=1}^{N} \left(\sum_{d=1}^{M} a_d^{z,i} f_{\theta_d}(x_i) \right)$$

$$= \sum_{z=1}^{Z} a_z^y \left(\sum_{d_1=1}^{M} a_{d_1}^{z,1} f_{\theta_{d_1}}(x_1) \right)_1 \dots \left(\sum_{d_N=1}^{M} a_{d_N}^{z,N} f_{\theta_{d_N}}(x_N) \right)_N$$

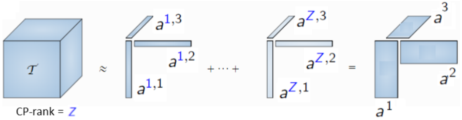
$$= \sum_{d_1,d_2\dots,d_N}^{M} \mathcal{A}^y_{d_1,\dots,d_N} \prod_{i=1}^{N} f_{\theta_{d_i}}(x_i)$$

$$\text{where } \mathcal{A}^y_{d_1,\dots,d_N} = \sum_{z=1}^{Z} a_z^y a_{d_1}^{z,1} \dots a_{d_N}^{z,N} \qquad (1)$$

Shallow ConvAC as CPD (1/2)

Canonical polyadic decomposition (CPD):
Tensor $\mathcal{A}$ sum of rank-1 tenors
$\mathcal{A}$ of dimensions $I_1 \times I_2 \times \cdots \times I_N$
A^n be matrices of size $I_n \times Z$
$a^{z,n}$ the z_{th} column of A^n

$$\mathcal{A} = \sum_{z=1}^{Z} \alpha_z a^{z,1} \otimes \cdots \otimes a^{z,N}, \qquad a^{z,i} \in \mathbb{R}^{M_i} \tag{2}$$

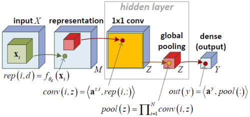


Shallow ConvAC as CPD (2/2)

Canonical polyadic decomposition (CPD):
Tensor $\mathcal{A}$ sum of rank-1 tenors[5]

$$\mathcal{A} = \sum_{z=1}^{Z} a_z a^{z,1} \otimes \cdots \otimes a^{z,N}, \qquad a^{z,i} \in \mathbb{R}^{M_i} \tag{3}$$

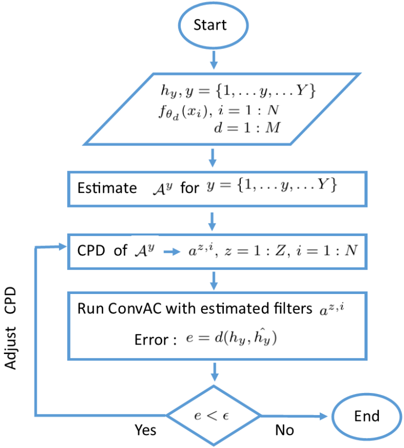
$$\mathcal{A}_{d_1,\ldots,d_N} = \sum_{z=1}^{Z} a_z a^{z,1}_{d_1} \ldots a^{z,N}_{d_N} \tag{4}$$



$$h_y(x_1,\ldots,x_N) = \sum_{(d_1,\ldots,d_N)}^{M} \mathcal{A}^y{}_{d_1,\ldots,d_N} \prod_{i=1}^{N} f_{\theta_{d_i}}(x_i), \qquad a_z := a_z{}^y \tag{5}$$

[5] CP-rank: min # of terms in CPD (min Z for with model holds)

Parameter estimation



Problem

1. Solve a system $h_y(X) =< \mathcal{A}^y, \mathcal{F}_\theta >$ for $y = 1, \ldots, Y$ $(*)$
2. Solve $(*)$ for a  of input signals

Approach

- $\mathcal{A}^y = \sum_{z=1}^{Z} a_z^y a^{z,1} \otimes \cdots \otimes a^{z,N} \rightarrow$ estimate $a^{z,i}$

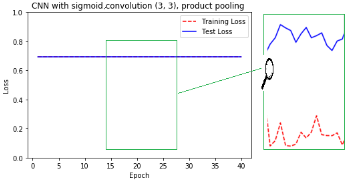
Estimation

- vectorize $(*)$ and solve
  - $\text{vec}(\mathcal{A}^y) = (\text{vec}(\mathcal{F}_\theta)\text{vec}(\mathcal{F}_\theta)')^{-1}\text{vec}(\mathcal{F})h_y$, through normal equation
  - $\text{vec}(\mathcal{A}^y)$ through multilinear regression
- coupled CPD($\hat{\mathcal{A}}^y$) for a system $y = 1, \ldots, Y$ of shared factors $a^{z,i}$'s
- standard convolution: $\forall i = 1 : N$, $a^{z,i} = a^z$ for $z = 1 : Z$

Implementation in progress

- **ConvAC**: Keras is a high-level neural networks API.
  We introduce product pooling layer & add softmax (decision layer in ConvAC)
- **CPD**: Tensorlab is a Matlab toolbox that provides various tools for tensor computations/decomposition.
- **SimNet** on GitHub (*Caffe, TensorFlow*)
  Problems if incompatibilities

Experimental results

- Synthetic data: 2 x random distributions of images 28x28 with μ_1, σ_1 & μ_2, σ_2 resp.
- vary $\mu 2$ to assess overlapping
-  Divergence "related" to pooling layer
  - 

- SimNet: **logarithmic** space[6]

[6] product instability: [Appendix E, Cohen 2016]

Recap: ConvAC as a tensor

ConvAC: ConvNet, choice of non-linearities:

- classify $X = (x_1, \ldots, x_N)$ into categories $\mathcal{Y} := \{1, \ldots Y\}$

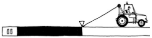
$$h_y(x_1, \ldots, x_N) = \sum_{(d_1, \ldots, d_N)}^{M} \mathcal{A}^y_{d_1, \ldots, d_N} \prod_{i=1}^{N} f_{\theta_{d_i}}(x_i) \tag{6}$$

- $\mathcal{A}^y$ CPD of convolution filters.

💡 Estimate CNN parameters from tensor characteristics: optimal CPD rank = # conv filters)

⚠ Computational challenge: exponential number of entries M^N

Perspectives 1



1. **Validation**: synthetic data & MNIST dataset
2. **Application**: biomedical data mainly electrocardiogram (ECG) for arrhythmia characterization, prediction of non recovery after ablation

Biomarkers within IADB workflow.

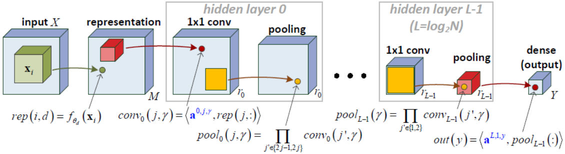
- Which biomedical topic, image?
- How to adapt NN design to the application, in practice?

Perspectives 2

Deep network ⇔ HT decomposition of $\mathcal{A}^y$

7

Deep network with size-2 pooling:



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

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

$$\cdots$$

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

$$\cdots$$

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

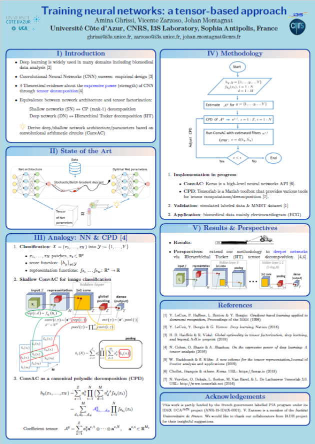
8

[7] The network, dubbed HT model, is universal: $\forall \{\mathcal{A}^y\}_y$ represented by **CP** model can be represented by **HT** model with only a polynomial penalty in terms of resources

[8] [N.Cohen, 2017]

Eurasip Summer School

Tensor-Based Signal Processing. August 27 - 31, 2018, Leuven, Belgium




It's just the beginning

References

[1] N. Cohen, O. Sharir, Y. Levine, R. Tamari, D. Yakira, and A. Shashua. Analysis and design of convolutional networks via hierarchical tensor decompositions. *arXiv preprint arXiv:1705.02302*, 2017.

[2] N. Cohen, O. Sharir, and A. Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on Learning Theory*, pages 698–728, 2016.

[3] W. Hackbusch and S. Kühn. A new scheme for the tensor representation. *Journal of Fourier analysis and applications*, 15(5):706–722, 2009.

[4] B. D. Haeffele and R. Vidal. Global optimality in tensor factorization, deep learning, and beyond. *arXiv preprint arXiv:1506.07540*, 2015.

[5] B. D. Haeffele and R. Vidal. Global optimality in neural network training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7331–7339, 2017.

[6] Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[7] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[8] Y. Levine, D. Yakira, N. Cohen, and A. Shashua. Deep learning and quantum entanglement: Fundamental connections with implications to network design. *CoRR, abs/1704.01552*, 2017.

[9] Rene Vidal, Joan Bruna, Raja Giryes, and Stefano Soatto. Mathematics of deep learning. *arXiv preprint arXiv:1712.04741*, 2017.

Shallow ConvAC as a tensor

$$h_y(x_1, \dots, x_N) = \sum_{(d_1,\dots,d_N)} \mathcal{A}^y_{d_1,\dots,d_N} \prod_{i=1}^{N} f_{\theta_{d_i}}(x_i) \quad (7)$$

f_θ **Gaussian / neurons (sigmoid or ReLU)**

$$h_y(x_1, \dots, x_N) = \sum_{(d_1,\dots,d_N)}^{M \text{ order of } 100} \mathcal{A}^y_{d_1,\dots,d_N} \prod_{i=1}^{N} f_{\theta_{d_i}}(x_i) \quad (8)$$

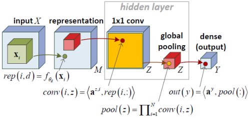
$\exists \mathcal{A}^y$ tensor: multi-dimensional array
N: order of $\mathcal{A}^y$
M: dimension, # of values and index can take in particular model i
$M.N \times \mathcal{A}^y_{d_1 d_2 \dots d_N} \in \mathbb{R}$ entries of $\mathcal{A}^y$, for $i \in [N]$ and $d \in [M_i]$

Shallow ConvAC as CPD

Canonical polyadic decomposition (CPD): Tensor $\mathcal{A}$ sum of rank-1 tenors[9]

$$\mathcal{A} = \sum_{z=1}^{Z} a_z a^{z,1} \otimes \cdots \otimes a^{z,N}, \qquad a^{z,i} \in \mathbb{R}^{M_i} \tag{9}$$

$$\mathcal{A}_{d_1,\ldots,d_N} = \sum_{z=1}^{Z} a_z a^{z,1}_{d_1} \ldots a^{z,N}_{d_N} \tag{10}$$



$$h_y(x_1,\ldots,x_N) = \sum_{(d_1,\ldots,d_N)}^{M} \mathcal{A}^y{}_{d_1,\ldots,d_N} \prod_{i=1}^{N} f_{\theta_{d_i}}(x_i), \qquad a_z := a_z{}^y \tag{11}$$

[9] CP-rank: min # of terms in CPD (min Z for with model holds)

Recap: ConvAC as a tensor

ConvAC: convNet, choice of non-linearities:

- classify $X = (x_1, \dots, x_N)$ into categories $\mathcal{Y} := \{1, \dots Y\}$

EXAMPLE X an image, $x_1, \dots, x_N$ patches around pixels

- score function: $\{h_y\}_{y \in \mathcal{Y}}$
- representation functions $f_{\theta_1} \dots f_{\theta_1}$: $\mathbb{R}^s \to \mathbb{R}$

$$h_y(x_1, \dots, x_N) = \sum_{(d_1, \dots, d_N)}^{M} \mathcal{A}^y_{d_1, \dots, d_N} \prod_{i=1}^{N} f_{\theta_{d_i}}(x_i) \tag{12}$$

$\mathcal{A}^y$ CPD of convolution operators / parameters.

Estimate NN parameters from tensor characteristics(rank, order, dimension ...)

Computational challenge: exponential number of entries M^N

SimNet & ConvAC

- In "Appendix E. Computation in Log-Space with SimNets" in paperhttps://arxiv.org/pdf/1509.05009.pdf : Authors suggest not to implementthe ConvAC in a standard way (the way we plan to do it) because ofinstability derived by product pooling. Instead, they suggest to use their"famous" simnet implementation
-