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 Purpose

Tensor-based training

Conclusion & Perspectives
PhD Purpose

Multimodal analysis of Biomedical Data

- Funding: IDEX UCA\textsuperscript{JEDI}
- Project: Intégration et Analyse de Données Biomédicales (IADB)
- Mission: Extract biomarkers within workflow
- Multimodality:
  - Heterogeneous
  - Complementary
  - Improve isolated properties
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

<table>
<thead>
<tr>
<th></th>
<th>hand-crafted</th>
<th>end-to-end CNN</th>
<th>conv+KNN</th>
<th>conv+SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCR (%)</td>
<td>82</td>
<td>85</td>
<td>91</td>
<td>87</td>
</tr>
</tbody>
</table>

2. Correct classification rate
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.

∃ Equivalence between network architecture and tensor factorization:
   Shallow networks (SN) ⇔ CP (rank-1) decomposition
   Deep network (DN) ⇔ Hierarchical Tucker decomposition (HT)

ConvAC

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

💡 ConvAC Vs. SimNet

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

- $MEX_{\xi}\{c_i\}_{i=1}^n \xrightarrow{\xi \rightarrow \infty} \max\{c_i\}_{i=1}^n$
- $MEX_{\xi}\{c_i\}_{i=1}^n \xrightarrow{\xi \rightarrow 0} \text{mean}\{c_i\}_{i=1}^n$
- $MEX_{\xi}\{c_i\}_{i=1}^n \xrightarrow{\xi \rightarrow -\infty} \min\{c_i\}_{i=1}^n$
Shallow ConvAC

4 joint decomposition: same vectors $a^{z,i}$ shared across $\forall$ classes $y$. 
Shallow ConvAC as a tensor

\[ h_y(x_1, \ldots, x_N) = \sum_{z=1}^{Z} a^y_z \prod_{i=1}^{N} \left( \sum_{d=1}^{M} a^z_i f_{\theta_d}(x_i) \right) \]

\[ = \sum_{z=1}^{Z} a^y_z \left( \sum_{d_1=1}^{M} a^{z,1}_{d_1} f_{\theta_{d_1}}(x_1) \right) \cdots \left( \sum_{d_N=1}^{M} a^{z,N}_{d_N} f_{\theta_{d_N}}(x_N) \right) \]

\[ = \sum_{d_1, d_2, \ldots, d_N} A^y_{d_1, \ldots, d_N} \prod_{i=1}^{N} f_{\theta_{d_i}}(x_i) \]

where \( A^y_{d_1, \ldots, d_N} = \sum_{z=1}^{Z} a^y_z a^{z,1}_{d_1} \cdots a^{z,N}_{d_N} \)
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 $A^n$

$$\mathcal{A} = \sum_{z=1}^{Z} \alpha_z a^{z,1} \otimes \cdots \otimes a^{z,N}, \quad a^{z,i} \in \mathbb{R}^{M_i} \quad (2)$$
Shallow ConvAC as CPD (2/2)

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

\[
\mathcal{A} = \sum_{z=1}^{Z} a_{z} a_{z,1}^{1} \otimes \cdots \otimes a_{z,N}^{N}, \quad a_{z,i} \in \mathbb{R}^{M_i} \quad (3)
\]

\[
\mathcal{A}_{d_1,...,d_N} = \sum_{z=1}^{Z} a_{z} a_{d_1}^{z,1} \cdots a_{d_N}^{z,N} \quad (4)
\]

\[
h_{y}(x_1,\ldots,x_N) = \sum_{(d_1,...,d_N)}^{M} \mathcal{A}^{y}_{d_1,...,d_N} \prod_{i=1}^{N} f_{\theta_{d_i}}(x_i), \quad a_{z} := a_{z}^{y} \quad (5)
\]

\(^5\)CP-rank: min # of terms in CPD (min $Z$ for with model holds)
Parameter estimation

Start

$h_y, y = \{1, \ldots, y, \ldots, Y\}$
$f_{\theta_d}(x_i), i = 1 : N$
$d = 1 : M$

Estimate $\mathcal{A}^y$ for $y = \{1, \ldots, y, \ldots, Y\}$

CPD of $\mathcal{A}^y$ $\rightarrow a^{z,i}, z = 1 : Z, i = 1 : N$

Run ConvAC with estimated filters $a^{z,i}$

Error: $e = d(h_y, \hat{h}_y)$

Adj CDP

Yes $e < \varepsilon$ No End
Problem

1. Solve a system \( h_y(X) = \langle A^y, F_\theta \rangle \) for \( y = 1, \ldots, Y \) (*)

2. Solve (*) for a \( \Box \) of input signals

Approach

- \( A^y = \sum_{z=1}^{Z} a^y_z a^{z,1} \otimes \cdots \otimes a^{z,N} \rightarrow \) estimate \( a^{z,i} \)

Estimation

- vectorize (*) and solve
  - \( \text{vec}(A^y) = (\text{vec}(F_\theta)\text{vec}(F_\theta)')^{-1}\text{vec}(F)h_y, \) through normal equation
  - \( \text{vec}(A^y) \) through multilinear regression
- coupled CPD( \( \hat{A}^y \)) 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 $\mu_1$, $\sigma_1$ & $\mu_2$, $\sigma_2$ 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 $Y := \{1, \ldots Y\}$

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

- $A^y$ CPD of convolution filters.
  - $\triangledown$ Estimate CNN parameters from tensor characteristics: optimal CPD rank = # conv filters)
  - $\Delta$ 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

ทำไมต้องการเรียกระดับที่不佳 within IADB workflow.

- Which biomedical topic, image?
- How to adapt NN design to the application, in practice?
Perspectives 2

Deep network $\Leftrightarrow$ HT decomposition of $A^y$

Deep network with size-2 pooling:

\[
\begin{align*}
\phi^{1,j,\gamma} &= \sum_{\alpha=1}^{n_0} a^{1,j,\gamma}_{\alpha} \cdot a^{0,2j-1,\alpha} \otimes a^{0,2j,\alpha} \\
\phi^{L,j,\gamma} &= \sum_{\alpha=1}^{n-1} a^{L,j,\gamma}_{\alpha} \cdot \phi^{l-1,2j-1,\alpha} \otimes \phi^{l-1,2j,\alpha} \\
A^y &= \sum_{\alpha=1}^{n-1} a^{L,1,\gamma}_{\alpha} \cdot \phi^{L-1,1,\alpha} \otimes \phi^{L-1,2,\alpha}
\end{align*}
\]

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

\footnote{[N.Cohen, 2017]}
Eurasip Summer School

It’s just the beginning
References


Shallow ConvAC as a tensor

\[ h_y(x_1, \ldots, x_N) = \sum_{(d_1, \ldots, d_N)} A_y^{d_1, \ldots, d_N} \prod_{i=1}^N f_{\theta_{d_i}}(x_i) \] (7)

\( f_\theta \) Gaussian / neurons (sigmoid or ReLU)

\[ h_y(x_1, \ldots, x_N) = \sum_{(d_1, \ldots, d_N)} A_y^{d_1, \ldots, d_N} \prod_{i=1}^N f_{\theta_{d_i}}(x_i) \] (8)

\( \exists A_y \) tensor: multi-dimensional array

\( N \): order of \( A_y \)

\( M \): dimension, \# of values and index can take in particular model \( i \)

\( M.N \times A_y^{d_1d_2 \ldots d_N} \in \mathbb{R} \) entries of \( A_y \), 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

$$\mathcal{A} = \sum_{z=1}^{Z} a^z_1 \otimes \ldots \otimes a^z_N, \quad a^z_i \in \mathbb{R}^{M_i}$$  \hspace{1cm} (9)

$$\mathcal{A}_{d_1, \ldots, d_N} = \sum_{z=1}^{Z} a^z_1 \otimes \ldots \otimes a^z_N$$  \hspace{1cm} (10)

$9^{\text{CP-rank: min \# of terms in CPD (min Z for with model holds)}}$

$$h_y(x_1, \ldots, x_N) = \sum_{(d_1, \ldots, d_N)} M \mathcal{A}^y_{d_1, \ldots, d_N} \prod_{i=1}^{N} f_{\theta_{d_i}}(x_i), \quad a_z := a^y_z$$  \hspace{1cm} (11)
Recap: ConvAC as a tensor

ConvAC: convNet, choice of non-linearities:

- classify $X = (x_1, \ldots, x_N)$ into categories $\mathcal{Y} := \{1, \ldots, \mathcal{Y}\}$
- score function: $\{h_y\}_{y \in \mathcal{Y}}$
- representation functions $f_{\theta_1} \ldots f_{\theta_1}: \mathbb{R}^s \rightarrow \mathbb{R}$

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

$\mathcal{A}^y$ 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 paper https://arxiv.org/pdf/1509.05009.pdf: Authors suggest not to implement the ConvAC in a standard way (the way we plan to do it) because of instability derived by product pooling. Instead, they suggest to use their "famous" simnet implementation.