Learning Sentence Embeddings through Tensor Methods

Anima Anandkumar

Joint work with Dr. Furong Huang

ACL Workshop 2016
Representations for Text Understanding

- **Word Embedding**
  - Incorporates short range relationships, Easy to train.

- **Word Sequence Embedding**
  - Incorporates long range relationships, hard to train.

Example:
- The weather is good.
  - Embedding for this sentence
- Her life spanned years of incredible change for women.
  - Embedding for this sentence
- Mary lived through an era of liberating reform for women.
  - Embedding for this sentence
Various Frameworks for Sentence Embeddings

Compositional Models (M. Iyyer et al ‘15, T. Kenter ‘16)

- Composition of word embedding vectors: usually simple averaging.
- Compositional operator (averaging weights) based on neural nets.
- Weakly supervised (only averaging weights based on labels) or strongly supervised (joint training).

Paragraph Vector (Q. V. Le & T. Mikolov ‘14)

- Augmented representation of paragraph + word embeddings.
- Supervised framework to train paragraph vector.

For both frameworks

- **Pros:** Simple and cheap to train. Can use existing word embeddings.
- **Cons:** Word order not incorporated. Supervised. Not universal.
Skip thought Vectors for Sentence Embeddings

- Learn sentence embedding based on joint probability of words, represented using RNN.
Skip thought Vectors for Sentence Embeddings

- Learn sentence embedding based on joint probability of words, represented using RNN.

- **Pros:** Incorporates word order, unsupervised, universal.
- **Cons:** Requires contiguous long text, lots of data, slow training time. Cannot use domain specific training.

Convolutional Models for Sentence Embeddings

(N. Kalchbrenner, E. Grefenstette, P. Blunsom ‘14)
Convolutional Models for Sentence Embeddings
(N. Kalchbrenner, E. Grefenstette, P. Blunsom ‘14)

- **Pros:** Incorporates word order. Detect polysemy.
- **Cons:** Supervised training. Not universal.
Convolutional Models for Sentence Embeddings

(F. Huang & A. ‘15)

A sample sentence

Word encoding

Word order

Activation Maps

max-k pooling

Label
Convolutional Models for Sentence Embeddings

(F. Huang & A. ‘15)

- **Pros:** Word order, polysemy, unsupervised, universal.
- **Cons:** Difficulty in training.
Shift invariance natural in images: image templates in different locations.

Image

Dictionary elements
Intuition behind Convolutional Model

- **Shift invariance** natural in images: *image templates* in different locations.

![Image](image.png)

**Dictionary elements**

- **Shift invariance** in language: *phrase templates* in different parts of the sentence.

*Image*
Learning Convolutional Dictionary Models

$$x = f_1 * w_1 + f_L * w_2$$

- Input $x$, phrase templates (filters) $f_1, f_2$, activations $w_1, w_2$
Learning Convolutional Dictionary Models

Input $x$, phrase templates (filters) $f_1, f_2$, activations $w_1, w_2$

Training objective: $\min_{f_i, w_i} \|x - \sum_i f_i \ast w_i\|_2^2$
Learning Convolutional Dictionary Models

\[
\begin{bmatrix}
\end{bmatrix} = \begin{bmatrix}
\end{bmatrix} \ast \begin{bmatrix}
\end{bmatrix} + \begin{bmatrix}
\end{bmatrix} \ast \begin{bmatrix}
\end{bmatrix}
\]

\[
x \quad f_1 \quad w_1 \quad f_L \quad w_2
\]

- Input \( x \), phrase templates (filters) \( f_1, f_2 \), activations \( w_1, w_2 \)

- **Training objective:** \( \min_{f_i, w_i} \| x - \sum_i f_i \ast w_i \|_2^2 \)

**Challenges**

- **Nonconvex optimization:** no guaranteed solution in general.
- **Alternating minimization:** Fix \( w_i \)'s to update \( f_i \)'s and vice versa.
- Not guaranteed to reach **global optimum** (or even a stationary point!)
- **Expensive in large sample regime:** needs updating of \( w_i \)'s.
Convex vs. Non-convex Optimization

Guarantees for mostly convex..

But non-convex is trending!

Images taken from https://www.facebook.com/nonconvex
Convex vs. Nonconvex Optimization

- Unique optimum: global/local.
- Multiple local optima

Guaranteed approaches for reaching global optima?
Non-convex Optimization in High Dimensions

Critical/statitionary points: $x : \nabla_x f(x) = 0$.

- Curse of dimensionality: exponential number of critical points.
- Saddle points slow down improvement.
- Lack of stopping criteria for local search methods.

Fast escape from saddle points in high dimensions?
Outline

1 Introduction

2 Why Tensors?

3 Tensor Decomposition Methods

4 Other Applications

5 Conclusion
Example: Discovering Latent Factors

- List of scores for students in different tests
- Learn **hidden factors** for **Verbal** and **Mathematical Intelligence** [C. Spearman 1904]

\[
\text{Score} \ (\text{student}, \text{test}) = \text{student}_{\text{verbal-intlg}} \times \text{test}_{\text{verbal}} \\
+ \text{student}_{\text{math-intlg}} \times \text{test}_{\text{math}}
\]
Matrix Decomposition: Discovering Latent Factors

- Identifying **hidden factors** influencing the observations
- Characterized as **matrix decomposition**
Matrix Decomposition: Discovering Latent Factors

- Decomposition is not necessarily unique.
- Decomposition cannot be overcomplete.
Tensor: Shared Matrix Decomposition

- **Shared** decomposition with different scaling factors
- **Combine** matrix slices as a **tensor**
Tensor Decomposition

Outer product notation:

\[ T = u \otimes v \otimes w + \tilde{u} \otimes \tilde{v} \otimes \tilde{w} \]

\[ T_{i_1, i_2, i_3} = u_{i_1} \cdot v_{i_2} \cdot w_{i_3} + \tilde{u}_{i_1} \cdot \tilde{v}_{i_2} \cdot \tilde{w}_{i_3} \]
Identifiability under Tensor Decomposition

\[ T = v_1 \otimes^3 + v_2 \otimes^3 + \cdots, \]

Uniqueness of Tensor Decomposition [J. Kruskal 1977]

- Above tensor decomposition: unique when rank one pairs are linearly independent
- Matrix case: when rank one pairs are orthogonal
Identifiability under Tensor Decomposition

\[ T = v_1 \otimes^3 + v_2 \otimes^3 + \cdots, \]

Uniqueness of Tensor Decomposition [J. Kruskal 1977]

- Above tensor decomposition: unique when rank one pairs are linearly independent
- Matrix case: when rank one pairs are orthogonal
Identifiability under Tensor Decomposition

\[ T = v_1 \otimes^3 + v_2 \otimes^3 + \cdots, \]

Uniqueness of Tensor Decomposition [J. Kruskal 1977]

- Above tensor decomposition: unique when rank one pairs are linearly independent
- Matrix case: when rank one pairs are orthogonal
Moment-based Estimation

Matrix: Pairwise Moments
- $\mathbb{E}[x \otimes x] \in \mathbb{R}^{d \times d}$ is a second order tensor.
- $\mathbb{E}[x \otimes x]_{i_1,i_2} = \mathbb{E}[x_{i_1} x_{i_2}]$.
- For matrices: $\mathbb{E}[x \otimes x] = \mathbb{E}[xx^\top]$.
- $M = uu^\top$ is rank-1 and $M_{i,j} = u_i u_j$.

Tensor: Higher order Moments
- $\mathbb{E}[x \otimes x \otimes x] \in \mathbb{R}^{d \times d \times d}$ is a third order tensor.
- $\mathbb{E}[x \otimes x \otimes x]_{i_1,i_2,i_3} = \mathbb{E}[x_{i_1} x_{i_2} x_{i_3}]$.
- $T = u \otimes u \otimes u$ is rank-1 and $T_{i,j,k} = u_i u_j u_k$. 
Moment forms for Linear Dictionary Models

\[
\begin{bmatrix}
\end{bmatrix}
= \begin{bmatrix}
\end{bmatrix}
\begin{bmatrix}
\end{bmatrix}
\begin{bmatrix}
\end{bmatrix}
\]
Moment forms for Linear Dictionary Models

Independent components analysis (ICA)

- Independent coefficients, e.g. Bernoulli Gaussian.
- Can be relaxed to sparse coefficients with limited dependency.

Fourth order cumulant: $M_4 = \sum_{j \in [k]} \kappa_j a_j \otimes a_j \otimes a_j \otimes a_j$. 

= \[ \begin{array}{c} \text{Diagram} \end{array} \] + \[ \begin{array}{c} \text{Diagram} \end{array} \] \ldots
Convolutional dictionary model

\[ x = f_1^* w_1^* + f_L^* w_L^* \]

(a) Convolutional model

\[ x = \sum_i f_i \ast w_i = \sum_i \text{Cir}(f_i) w_i = F^* w^* \]

(b) Reformulated model
Moment forms and optimization

\[ x = \sum_i f_i \ast w_i = \sum_i \text{Cir}(f_i)w_i = \mathcal{F}^*w^* \]

- Assume coefficients \( w_i \) are independent (convolutional ICA model)
- Cumulant tensor has decomposition with components \( \mathcal{F}_i^* \).

Learning Convolutional model through Tensor Decomposition
Outline

1 Introduction

2 Why Tensors?

3 Tensor Decomposition Methods

4 Other Applications

5 Conclusion
Notion of Tensor Contraction

Extends the notion of matrix product

Matrix product

\[ Mv = \sum_{j} v_j M_j \]

Tensor Contraction

\[ T(u, v, \cdot) = \sum_{i,j} u_i v_j T_{i,j,:} \]
Tensor Decomposition - ALS

Objective: \( \| T - \sum_i a_i \otimes b_i \otimes c_i \|^2_2 \)
Tensor Decomposition - ALS

- Objective: \[ \| T - \sum_{i} a_i \otimes b_i \otimes c_i \|_2^2 \]

- Key observation: If \( b_i, c_i \)’s are fixed, objective is linear in \( a_i \)’s.
Tensor Decomposition - ALS

- Objective: \( \|T - \sum_i a_i \otimes b_i \otimes c_i\|_2^2 \)

- Key observation: If \( b_i, c_i \)'s are fixed, objective is **linear** in \( a_i \)'s.

- Tensor unfolding

\[
\begin{array}{c}
\text{i3} \\
\text{i2} \\
\text{i1}
\end{array}
\quad =
\begin{array}{c}
\text{red} \\
\text{blue}
\end{array}
+ \begin{array}{c}
\text{red} \\
\text{blue}
\end{array}
\]
Tensor Decomposition - ALS

- Objective: \[ \|T - \sum_i a_i \otimes b_i \otimes c_i\|^2 \]

- Key observation: If \( b_i, c_i \)'s are fixed, objective is **linear** in \( a_i \)'s.

- Tensor unfolding

\[ i_1 \hspace{1cm} i_2 \]

\[ = \begin{array}{c}
\text{\includegraphics[width=2cm]{red}} \\
\text{\includegraphics[width=2cm]{blue}}
\end{array} \]

\[ + \]

\[ \begin{array}{c}
\text{\includegraphics[width=2cm]{red}} \\
\text{\includegraphics[width=2cm]{blue}}
\end{array} \]
Tensor Decomposition - ALS

- Objective: \[ \| T - \sum_i a_i \otimes b_i \otimes c_i \|_2^2 \]
- Key observation: If \( b_i, c_i \)'s are fixed, objective is \textit{linear} in \( a_i \)'s.
- Tensor unfolding

![Tensor unfolding diagram]
Objective: \[ \| T - \sum_i a_i \otimes b_i \otimes c_i \|_2^2 \]

Key observation: If \( b_i, c_i \)’s are fixed, objective is linear in \( a_i \)’s.

Tensor unfolding
Convolutional Tensor Decomposition

Objective: \[ \| T - \sum_i a_i \otimes a_i \otimes a_i \|_2^2 \]

Constraint: \( A := [a_1, a_2, \ldots] \) is concatenation of circulant matrices.
Convolutional Tensor Decomposition

- Objective: $\|T - \sum_i a_i \otimes a_i \otimes a_i\|_2^2$

- Constraint: $A := [a_1, a_2, \ldots]$ is concatenation of circulant matrices.

Modified Alternating Least Squares Method

- Project onto set of concatenated circulant matrices in each step.
Convolutional Tensor Decomposition

Objective: \( \|T - \sum_i a_i \otimes a_i \otimes a_i\|_2^2 \)

Constraint: \( A := [a_1, a_2, \ldots] \) is concatenation of circulant matrices.

Modified Alternating Least Squares Method

- Project onto set of concatenated circulant matrices in each step.
- **Our contribution:** Efficient computation through FFT and blocking.
Comparison with Alternating Minimization

\[
x = f_1^* w_1^* + f_L^* w_L^*
\]

- \(L\) is the number of filters.
- \(n\) is the dimension of filters.
- \(N\) is the number of samples.

**Computation complexity**

<table>
<thead>
<tr>
<th>Methods</th>
<th>Running Time</th>
<th>Processors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tensor Factorization</td>
<td>(O(\log(n) + \log(L)))</td>
<td>(O(L^2 n^3))</td>
</tr>
<tr>
<td>Alt. Min</td>
<td>(O(\max(\log(n)\log(L), \log(n)\log(N))))</td>
<td>(O(\max(NnL, NnL)))</td>
</tr>
</tbody>
</table>

Complexity for tensor method independent of sample size.
Analysis

- Non-convex optimization: guaranteed convergence to local optimum
- Local optima are shifted filters
# Experiments using Sentence Embeddings

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review</td>
<td>Movie Reviews</td>
<td>64720</td>
</tr>
<tr>
<td>SUBJ</td>
<td>Obj/Subj comments</td>
<td>1000</td>
</tr>
<tr>
<td>MSRpara</td>
<td>news sources</td>
<td>5801×2</td>
</tr>
<tr>
<td>STS-MSRpar</td>
<td>newswire</td>
<td>1500×2</td>
</tr>
<tr>
<td>STS-MSRvid</td>
<td>video caption</td>
<td>1500×2</td>
</tr>
<tr>
<td>STS-OnWN</td>
<td>glosses</td>
<td>750×2</td>
</tr>
<tr>
<td>STS-SMTeuroparl</td>
<td>machine translation</td>
<td>1193×2</td>
</tr>
<tr>
<td>STS-SMTnews</td>
<td>machine translation</td>
<td>399×2</td>
</tr>
</tbody>
</table>
Experiments using Sentence Embeddings

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review</td>
<td>Movie Reviews</td>
<td>64720</td>
</tr>
<tr>
<td>SUBJ</td>
<td>Obj/Subj comments</td>
<td>1000</td>
</tr>
<tr>
<td>MSRpara</td>
<td>news sources</td>
<td>5801×2</td>
</tr>
<tr>
<td>STS-MSRpar</td>
<td>newswire</td>
<td>1500×2</td>
</tr>
<tr>
<td>STS-MSRvid</td>
<td>video caption</td>
<td>1500×2</td>
</tr>
<tr>
<td>STS-OnWN</td>
<td>glosses</td>
<td>750×2</td>
</tr>
<tr>
<td>STS-SMTeuroparl</td>
<td>machine translation</td>
<td>1193×2</td>
</tr>
<tr>
<td>STS-SMTnews</td>
<td>machine translation</td>
<td>399×2</td>
</tr>
</tbody>
</table>

Sentiment Analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>MR</th>
<th>SUBJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paragraph-vector</td>
<td>74.8</td>
<td>90.5</td>
</tr>
<tr>
<td>Skip-thought</td>
<td>75.5</td>
<td>92.1</td>
</tr>
<tr>
<td><strong>ConvDic+DeconvDec</strong></td>
<td><strong>78.9</strong></td>
<td><strong>92.4</strong></td>
</tr>
</tbody>
</table>

- Paragraph vector weakly supervised.
- Skip thought and our method unsupervised
**Paraphrase Detection Results**

<table>
<thead>
<tr>
<th>Method</th>
<th>Outside Information</th>
<th>F score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector Similarity</td>
<td>word similarity</td>
<td>75.3%</td>
</tr>
<tr>
<td>RMLMG</td>
<td>syntacticinfo</td>
<td>80.5%</td>
</tr>
<tr>
<td>ConvDic+DeconvDec</td>
<td>none</td>
<td><strong>80.7%</strong></td>
</tr>
<tr>
<td>Skip-thought</td>
<td>book corpus</td>
<td>81.9%</td>
</tr>
</tbody>
</table>

- **Paraphrase detected:** (1) Amrozi accused his brother, whom he called the witness, of deliberately distorting his evidence. (2) Referring to him as only the witness, Amrozi accused his brother of deliberately distorting his evidence.

- **Non-paraphrase detected:** (1) I never organised a youth camp for the diocese of Bendigo. (2) I never attended a youth camp organised by that diocese.
## Semantic Textual Similarity Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Supervised</th>
<th>Unsupervised</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DAN</td>
<td>RNN</td>
</tr>
<tr>
<td>MSRpar</td>
<td>40.3</td>
<td>18.6</td>
</tr>
<tr>
<td>MSRvid</td>
<td>70.0</td>
<td>66.5</td>
</tr>
<tr>
<td>SMT-eur</td>
<td>43.8</td>
<td>40.9</td>
</tr>
<tr>
<td>OnWN</td>
<td>65.9</td>
<td>63.1</td>
</tr>
<tr>
<td>SMT-news</td>
<td>60.0</td>
<td>51.3</td>
</tr>
</tbody>
</table>
Outline

1. Introduction
2. Why Tensors?
3. Tensor Decomposition Methods
4. Other Applications
5. Conclusion
Tensor Sketches for Multilinear Representations

- Randomized dimensionality reduction through sketching.
  - Complexity independent of tensor order: exponential gain!

Tensor Sketches for Multilinear Representations

- Randomized dimensionality reduction through sketching.
  - Complexity independent of tensor order: exponential gain!

State of art results for visual Q & A

Tensor Methods for Topic Modeling

- Topic-word matrix $\mathbb{P}[\text{word} = i|\text{topic} = j]$
- Linearly independent columns

Moment Tensor: Co-occurrence of Word Triplets
**Tensors vs. Variational Inference**

Criterion: Perplexity $= \exp[-\text{likelihood}]$.

Learning Topics from PubMed on Spark, 8mil articles

---

Tensors vs. Variational Inference
Criterion: Perplexity = \( \exp[-\text{likelihood}] \).

Learning Topics from PubMed on Spark, 8mil articles

Learning network communities from social network data
Facebook \( n \sim 20k \), Yelp \( n \sim 40k \), DBLP-sub \( n \sim 1e5 \), DBLP \( n \sim 1e6 \).

Tensors vs. Variational Inference

Criterion: Perplexity = $\exp[-\text{likelihood}]$.

Learning Topics from PubMed on Spark, 8mil articles

Learning network communities from social network data

Facebook $n \sim 20k$, Yelp $n \sim 40k$, DBLP-sub $n \sim 1e5$, DBLP $n \sim 1e6$.

Reinforcement Learning

- Rewards from hidden state.
- Actions drive hidden state evolution.
Reinforcement Learning of POMDPs

Reinforcement Learning

- Rewards from hidden state.
- Actions drive hidden state evolution.

Partially Observable Markov Decision Process

Learning using tensor methods under memoryless policies
Reinforcement Learning of POMDPs

Reinforcement Learning

- Rewards from hidden state.
- Actions drive hidden state evolution.

Partially Observable Markov Decision Process
Learning using tensor methods under memoryless policies
POMDP model with 3 hidden states (trained using tensor methods) vs. NN with 3 hidden layers 10 neurons each (trained using RmsProp).

K. Azzizade, Lazaric, A, Reinforcement Learning of POMDPs using Spectral Methods, COLT16.

http://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html
Reinforcement Learning of POMDPs

- POMDP model with 8 hidden states (trained using tensor methods) vs. NN with 3 hidden layers 30 neurons each (trained using RmsProp).

K. Azzizade, Lazaric, A, Reinforcement Learning of POMDPs using Spectral Methods, COLT16.

http://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html
POMDP model with 8 hidden states (trained using tensor methods) vs. NN with 3 hidden layers 30 neurons each (trained using RmsProp).

K. Azzizade, Lazaric, A, Reinforcement Learning of POMDPs using Spectral Methods, COLT16.

http://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html
Local Optima in Backpropagation

“..few researchers dare to train their models from scratch.. small miscalibration of initial weights leads to vanishing or exploding gradients.. poor convergence..”

\[ y = 1 \]
\[ y = -1 \]

Local optimum  
Global optimum

Exponential (in dimensions) no. of local optima for backpropagation

\[ y \]
\[ \sigma(\cdot) \]
\[ x \]
\[ x_1 \]
\[ x_2 \]

Training Neural Networks with Tensors

Input $x$ \rightarrow \text{Score } S(x)

Weights

Neurons $\sigma(\cdot)$

Output $y$

Input $x$

$E[y \cdot S(x)]$

Training Neural Networks with Tensors

Given input pdf $p(\cdot)$, $S_m(x) := (-1)^m \frac{\nabla^m p(x)}{p(x)}$.

Gaussian $x \Rightarrow$ Hermite polynomials.

Outline

1. Introduction

2. Why Tensors?

3. Tensor Decomposition Methods

4. Other Applications

5. Conclusion
Conclusion

Unsupervised Convolutional Models for Sentence Embedding

- Desirable properties: incorporates word order, polysemy, universality.
- Efficient training through tensor methods.
- Faster and better performance in practice.
Conclusion

Unsupervised Convolutional Models for Sentence Embedding
- Desirable properties: incorporates word order, polysemy, universality.
- Efficient training through tensor methods.
- Faster and better performance in practice.

Steps Forward
- **Universal** embeddings using tensor methods on large corpus.
- More challenging setups: **multilingual, multimodal** (e.g. image and caption embeddings) etc.
- **Bias-free embeddings?** Can gender/race and other undesirable biases be avoided?
Research Connections and Resources

Collaborators
Rong Ge (Duke), Daniel Hsu (Columbia), Sham Kakade (UW), Jennifer Chayes, Christian Borgs, Alex Smola (CMU), Prateek Jain, Alekh Agarwal & Praneeth Netrapalli (MSR), Srinivas Turaga (Janelia), Allesandro Lazaric (Inria), Hossein Mobahi (Google).

- Podcast/lectures/papers/software available at http://newport.eecs.uci.edu/anandkumar/