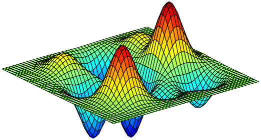
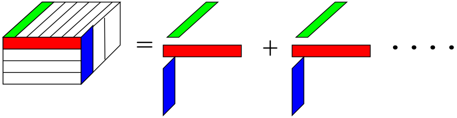
Tensor Methods for Guaranteed Machine Learning

Anima Anandkumar





U.C. Irvine

CVPR Bigvision 2016

Machine Learning - Modern Challenges

Massive datasets, growth in computation power, challenging tasks

Success of Supervised Learning


Image classification


Speech recognition


Text processing

Machine Learning - Modern Challenges

Massive datasets, growth in computation power, challenging tasks

Real AI requires Unsupervised Learning

Filter bank learning

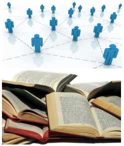
Feature extraction

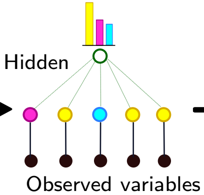
Embeddings, Topics

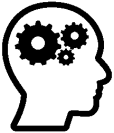
- Discover latent variables related to observations.
- Human vs. Machine Learning: Make discoveries automatically.

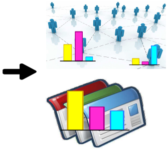
Unsupervised Learning via Probabilistic Models

Data $\rightarrow$ Model $\rightarrow$ Learning Algorithm $\rightarrow$ Predictions









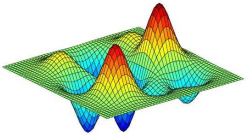
Challenges in High dimensional Learning

- Dimension of $x \gg$ dim. of latent variable h.
- Learning is like finding needle in a haystack.
- Computationally & statistically challenging.

Overview of Unsupervised Learning Methods

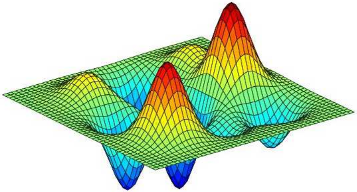
Goal: learn model parameters θ from observations x.

- Maximum likelihood: $\max_\theta p(x;\theta)$.
- Non-convex: stuck in local optima.
- Curse of dimensionality: Exponential no. of critical points.



- Heuristics: Expectation Maximization, Variational Inference ....
- Other mechanisms such as autoencoders, Generative Adversarial Nets also non-convex.

Guaranteed Learning through Tensor Methods



Replace the objective function

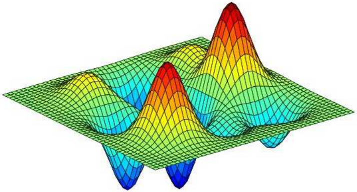
Max Likelihood vs. Best Tensor decomp.

Preserves Global Optimum (infinite samples)

$$\boxed{\arg\max_{\theta} p(x;\theta) = \arg\min_{\theta} \|\widehat{T}(x) - T(\theta)\|_{\mathbb{F}}^2}$$

$\widehat{T}(x)$: empirical tensor, $T(\theta)$: low rank tensor based on θ.

Guaranteed Learning through Tensor Methods



Replace the objective function

Max Likelihood vs. Best Tensor decomp.

Preserves Global Optimum (infinite samples)

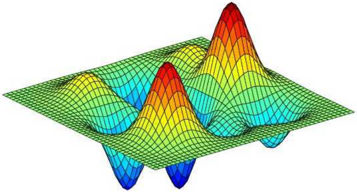
$$\boxed{\arg\max_{\theta} p(x;\theta) = \arg\min_{\theta} \|\widehat{T}(x) - T(\theta)\|_{\mathbb{F}}^2}$$

$\widehat{T}(x)$: empirical tensor, $T(\theta)$: low rank tensor based on θ.

Finding globally opt tensor decomposition

Simple algorithms succeed under mild and natural conditions for many learning problems.

Guaranteed Learning through Tensor Methods



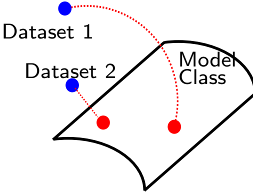
Replace the objective function

Max Likelihood vs. Best Tensor decomp.

Preserves Global Optimum (infinite samples)

$$\boxed{\arg\max_{\theta} p(x;\theta) = \arg\min_{\theta} \|\widehat{T}(x) - T(\theta)\|_{\mathbb{F}}^2}$$

$\widehat{T}(x)$: empirical tensor, $T(\theta)$: low rank tensor based on θ.

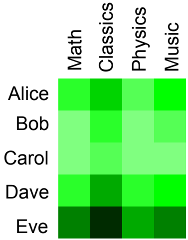


Finding globally opt tensor decomposition

Simple algorithms succeed under mild and natural conditions for many learning problems.

Outline

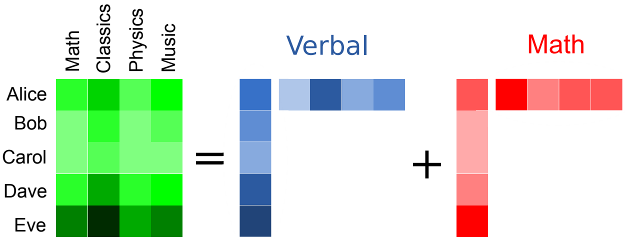
Matrix Decomposition: Discovering Latent Factors



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

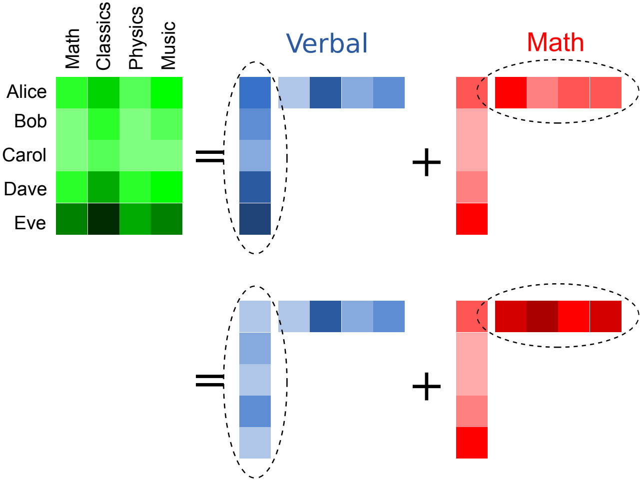
$$\begin{aligned}\text{Score (student,test)} &= \text{student}_{\text{verbal-intlg}} \times \text{test}_{\text{verbal}} \\ &+ \text{student}_{\text{math-intlg}} \times \text{test}_{\text{math}}\end{aligned}$$

Matrix Decomposition: Discovering Latent Factors



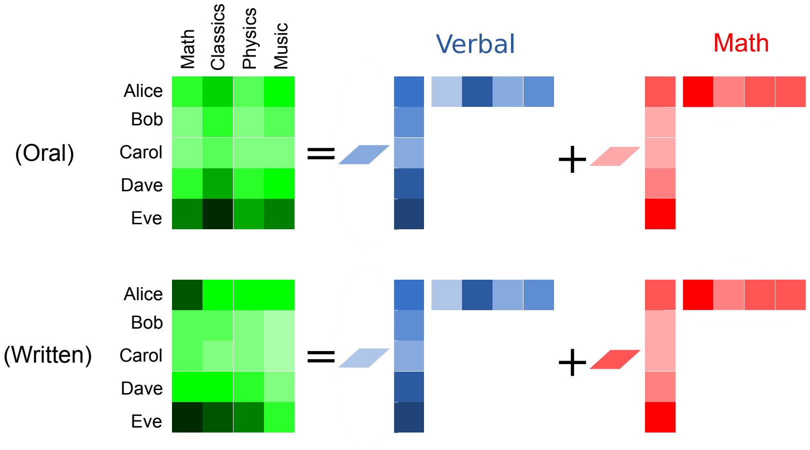
- Identifying hidden factors influencing the observations
- Characterized as matrix decomposition

Matrix Decomposition: Discovering Latent Factors



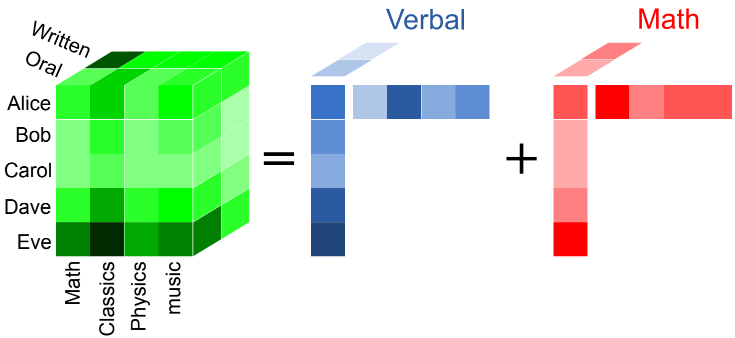
- Decomposition is not necessarily unique.
- Decomposition cannot be overcomplete.

Tensor: Shared Matrix Decomposition



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

Tensor Decomposition



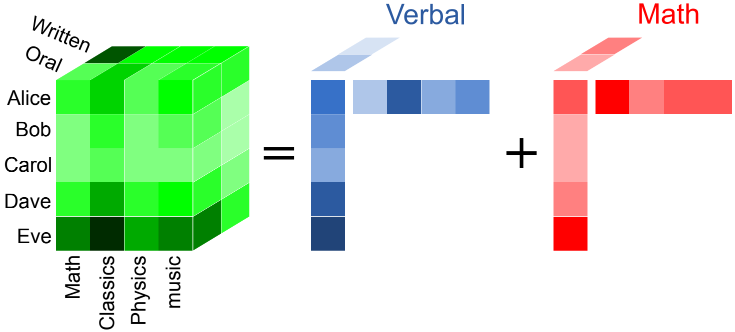
- Outer product notation:

$$T = u \otimes v \otimes w + \tilde{u} \otimes \tilde{v} \otimes \tilde{w}$$

$$\Updownarrow$$

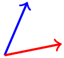
$$T_{i_1,i_2,i_3} = u_{i_1} \cdot v_{i_2} \cdot w_{i_3} + \tilde{u}_{i_1} \cdot \tilde{v}_{i_2} \cdot \tilde{w}_{i_3}$$

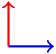
Tensor Decomposition



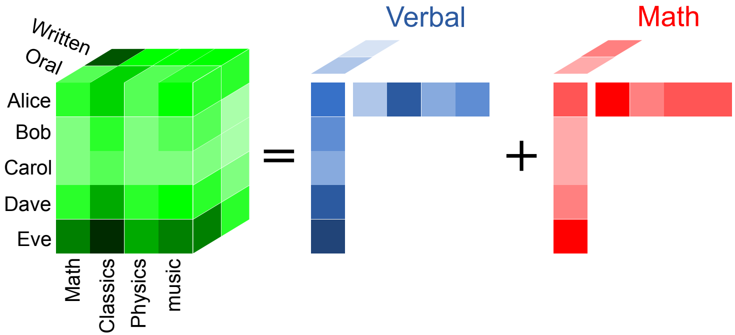
Uniqueness of Tensor Decomposition [J. Kruskal 1977]

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





Tensor Decomposition



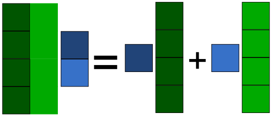
Finding Best Tensor Decomposition? Overcome Non-convexity?

Notion of Tensor Contraction

Extends the notion of matrix product

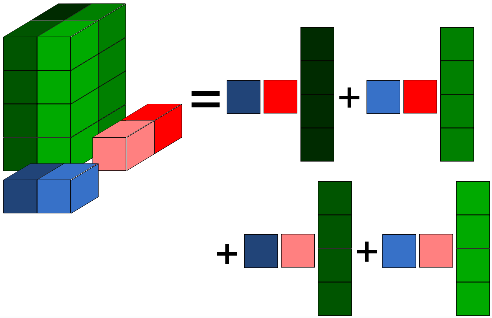
Matrix product

$$Mv = \sum_j v_j M_j$$

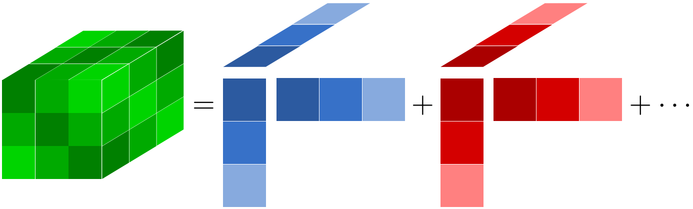


Tensor Contraction

$$T(u, v, \cdot) = \sum_{i,j} u_i v_j T_{i,j,:}$$



Symmetric Tensor Decomposition



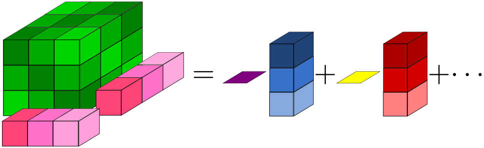
$$T = v_1^{\otimes 3} + v_2^{\otimes 3} + \cdots ,$$

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Symmetric Tensor Decomposition

Tensor Power Method

$$\boxed{v \mapsto \frac{T(v, v, \cdot)}{\|T(v, v, \cdot)\|}.}$$



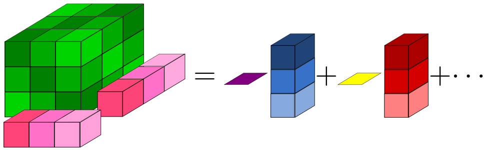
$$T(v, v, \cdot) = \langle v, v_1 \rangle^2 v_1 + \langle v, v_2 \rangle^2 v_2$$

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, “Tensor Decompositions for Learning Latent Variable Models,” JMLR 2014.

Symmetric Tensor Decomposition

Tensor Power Method

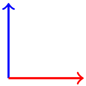
$$\boxed{v \mapsto \frac{T(v,v,\cdot)}{\|T(v,v,\cdot)\|}.}$$



$$T(v,v,\cdot) = \langle v, v_1\rangle^2 v_1 + \langle v, v_2\rangle^2 v_2$$

Orthogonal Tensors

- $\vec{v}_1 \perp \vec{v}_2$.
- $T(v_1, v_1, \cdot) = \lambda_1 v_1$.

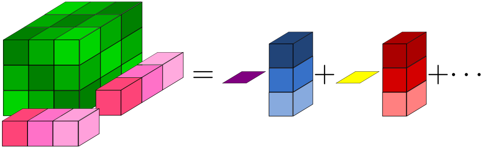


A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Symmetric Tensor Decomposition

Tensor Power Method

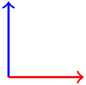
$$\boxed{v \mapsto \frac{T(v, v, \cdot)}{\|T(v, v, \cdot)\|}.}$$

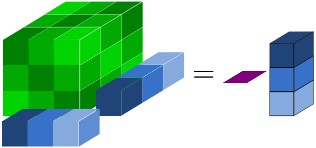


$$T(v, v, \cdot) = \langle v, v_1 \rangle^2 v_1 + \langle v, v_2 \rangle^2 v_2$$

Orthogonal Tensors

- $\vec{v}_1 \perp \vec{v}_2$.
- $T(v_1, v_1, \cdot) = \lambda_1 v_1$.



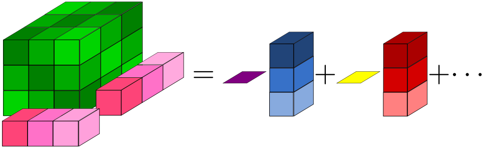


A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Symmetric Tensor Decomposition

Tensor Power Method

$$\boxed{v \mapsto \frac{T(v, v, \cdot)}{\|T(v, v, \cdot)\|}}.$$



$$T(v, v, \cdot) = \langle v, v_1 \rangle^2 v_1 + \langle v, v_2 \rangle^2 v_2$$

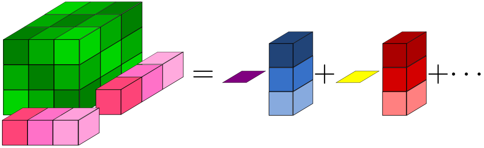
Exponential no. of stationary points for power method:
$T(v, v, \cdot) = \lambda v$

A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Symmetric Tensor Decomposition

Tensor Power Method

$$\boxed{v \mapsto \frac{T(v, v, \cdot)}{\|T(v, v, \cdot)\|}.}$$

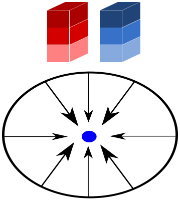


$$T(v, v, \cdot) = \langle v, v_1 \rangle^2 v_1 + \langle v, v_2 \rangle^2 v_2$$

Exponential no. of stationary points for power method:

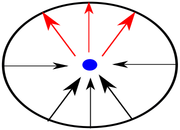
$T(v, v, \cdot) = \lambda v$

Stable



Unstable

Other statitionary points

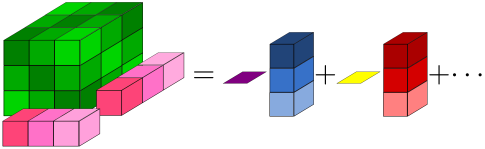


A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Symmetric Tensor Decomposition

Tensor Power Method

$$\boxed{v \mapsto \frac{T(v,v,\cdot)}{\|T(v,v,\cdot)\|}.}$$



$$T(v,v,\cdot) = \langle v, v_1\rangle^2 v_1 + \langle v, v_2\rangle^2 v_2$$

Exponential no. of stationary points for power method:
$T(v,v,\cdot) = \lambda v$

$\boxed{\text{For power method on orthogonal tensor, no spurious stable points}}$

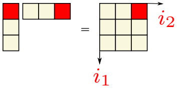
A., R. Ge, D. Hsu, S. Kakade, M. Telgarsky, "Tensor Decompositions for Learning Latent Variable Models," JMLR 2014.

Outline

Method of Moments

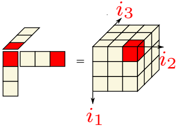
Matrix: Second Order Moments

- M_2: pair-wise relationship.
- $\mathbb{E}[x \otimes x]_{i_1,i_2} = \mathbb{E}[x_{i_1} x_{i_2}] \to [M_2]_{i_1,i_2}$



Tensor: Third Order Moments

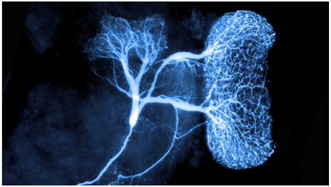
- M_3: triple-wise relationship.
- $\mathbb{E}[x \otimes x \otimes x]_{i_1,i_2,i_3} = \mathbb{E}[x_{i_1} x_{i_2} x_{i_3}] \to [M_3]_{i_1,i_2,i_3}$



Learning Representations

Sparse coding prevalent in neural signaling.

Neural sparse coding [Papadopoulou11]



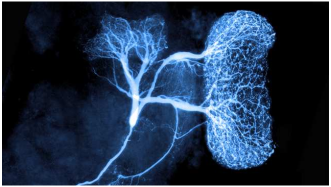
A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.

A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

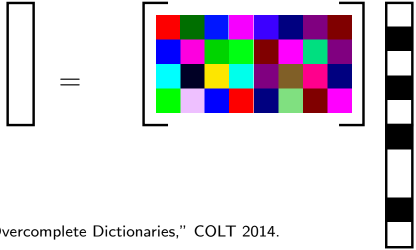
Learning Representations

Sparse coding prevalent in neural signaling.

Neural sparse coding [Papadopoulou11]



Linear Model with Overcomplete Dictionary



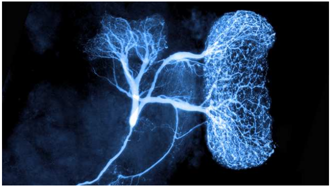
A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.

A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

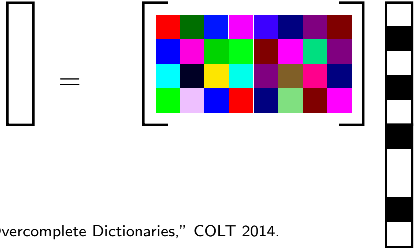
Learning Representations

$\boxed{\text{Contribution: learn overcomplete incoherent dictionaries}}$

Neural sparse coding [Papadopoulou11]



Linear Model with Overcomplete Dictionary



A. Agarwal, A, P. Jain, P. Netrapalli, "Learning Sparsely Used Overcomplete Dictionaries," COLT 2014.

A, M. Janzamin, R. Ge, "Overcomplete Tensor Decomposition, " COLT 2015.

Moment forms for Dictionary Models

$$x_i = Ah_i, \quad i \in [n].$$

Independent components analysis (ICA)

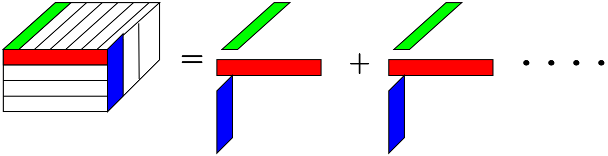
- h_i are independent, e.g. Bernoulli Gaussian

$$M_4 := \mathbb{E}[x \otimes x \otimes x \otimes x] - T, \text{ where}$$

$$T_{i_1,i_2,i_3,i_4} := \mathbb{E}[x_{i_1} x_{i_2}]\mathbb{E}[x_{i_3} x_{i_4}] + \mathbb{E}[x_{i_1} x_{i_3}]\mathbb{E}[x_{i_2} x_{i_4}] + \mathbb{E}[x_{i_1} x_{i_4}]\mathbb{E}[x_{i_2} x_{i_3}],$$

Let $\kappa_j := \mathbb{E}[h_j^4] - 3\mathbb{E}^2[h_j^2]$, $j \in [k]$. Then, we have

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



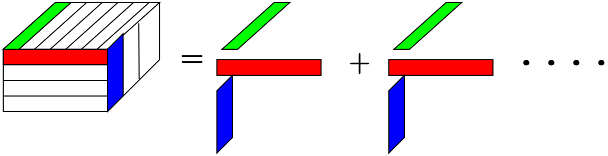
Moment forms for Dictionary Models

General (sparse) coefficients

$$x_i = Ah_i, \quad i \in [n], \quad \mathbb{E}[h_i] = s.$$

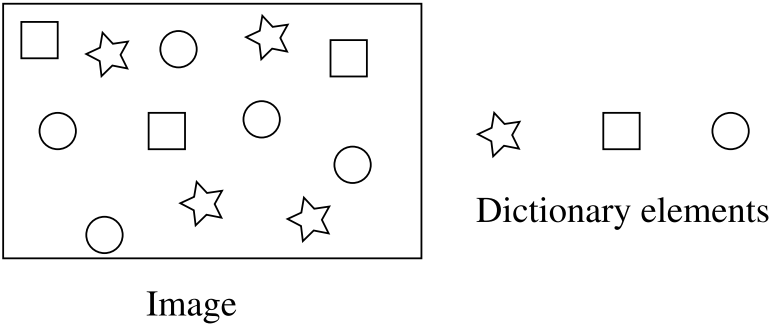
$$\begin{aligned} \mathbb{E}\left[h_i^4\right] = \mathbb{E}\left[h_i^2\right] &= \beta s/k, \\ \mathbb{E}\left[h_i^2 h_j^2\right] &\leq \tau, \quad i \neq j, \\ \mathbb{E}\left[h_i^3 h_j\right] &= 0, \quad i \neq j, \end{aligned}$$

$\mathbb{E}[x \otimes x \otimes x \otimes x] = \sum_{j \in [k]} \kappa_j a_j \otimes a_j \otimes a_j \otimes a_j + E$, where $\|E\| \leq \tau \|A\|^4$.

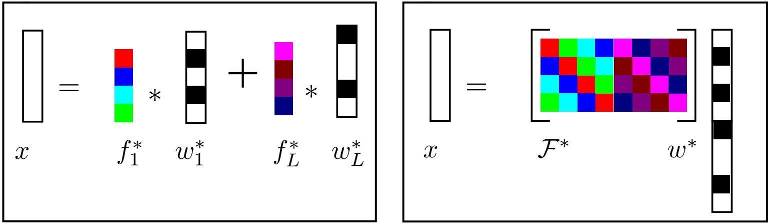


Convolutional Dictionary Model

- So far, invariances in dictionary are not incorporated.
- Convolutional models: incorporate invariances such as shift invariance.



Rewriting as a standard dictionary model



(a) Convolutional model **(b)** Reformulated model

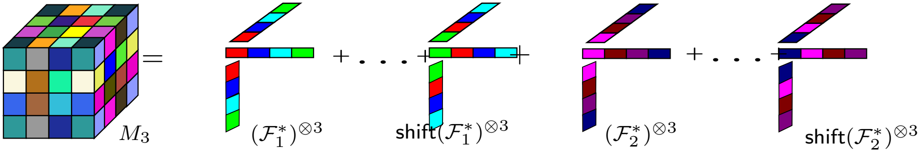
$$x = \sum_i f_i * w_i = \sum_i \mathsf{Cir}(f_i) w_i = \mathcal{F}^* w^*$$

- Circulant matrix has eigen decomposition
  $$\mathsf{Cir}(f) = U \mathsf{Diag}(\mathsf{DFT}_{\mathsf{1\text{–}d}}(f)) U^{\mathsf{H}} = U \mathsf{Diag}(\sqrt{n} U^{\mathsf{H}} \cdot f) U^{\mathsf{H}}$$
- U is the Discrete Fourier Transform Matrix.

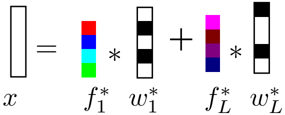
Moment forms and optimization

$$x = \sum_i f_i * w_i = \sum_i \mathsf{Cir}(f_i) w_i = \mathcal{F}^* w^*$$

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



Analysis



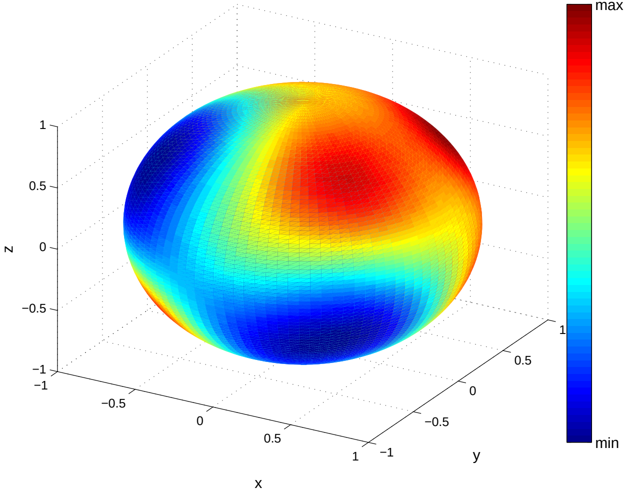
Comparison with Alternating Minimization(AM) method:

Methods	Running Time	Processors
Tensor Factorization	$O(\log(n)+\log(L))$	$O(L^2\ n^3)$
AM	$O(\max(\log(n)\log(L),\ \log(n)\log(N)))$	$O(\max(\frac{nNL}{logN}, \frac{nNL}{logL}))$

Table: Computation complexity (L is the number of filters, n is the dimension of filters. N is the number of samples)

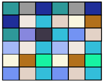
Analysis

- Non-convex optimization: guaranteed convergence to local optimum
- Local optima are shifted filters



Application: Paraphrase Detection

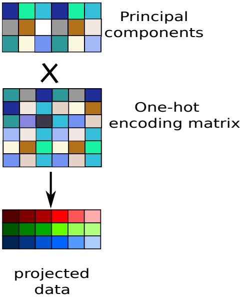
Microsoft paraphrase data: 5800 pairs of sentences



One-hot
encoding matrix

Application: Paraphrase Detection

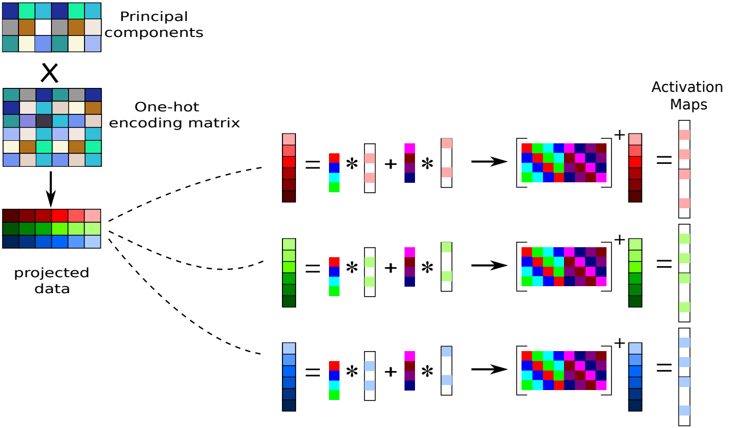
Microsoft paraphrase data: 5800 pairs of sentences



- PCA on One-hot Encoding Matrix $\rightarrow$ Subspace and Projected data

Application: Paraphrase Detection

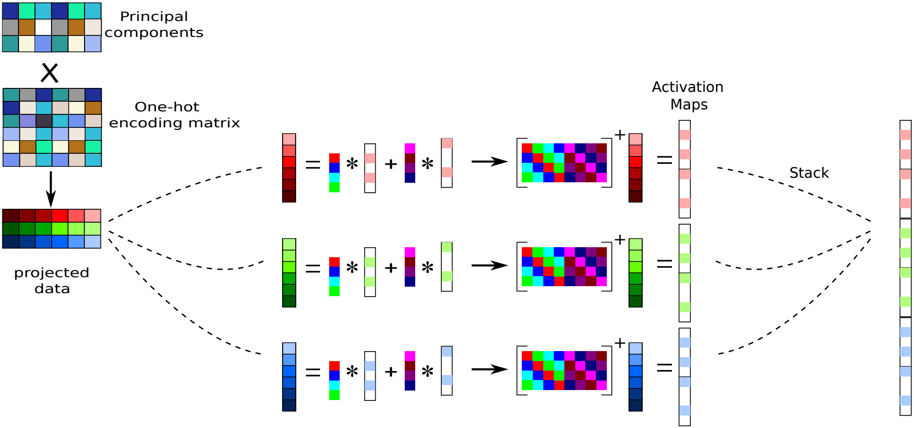
Microsoft paraphrase data: 5800 pairs of sentences



- CT on each coordinate $\rightarrow$ activation map for each coordinate

Application: Paraphrase Detection

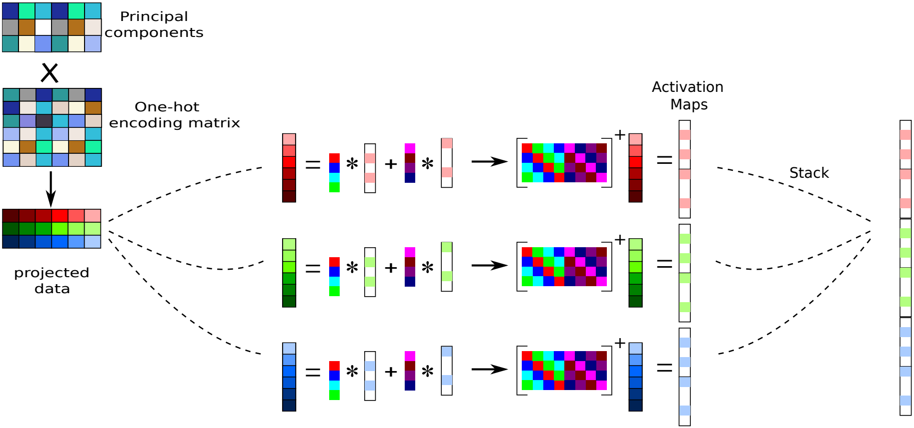
Microsoft paraphrase data: 5800 pairs of sentences



- Stack all activation maps → Sentence Embedding

Application: Paraphrase Detection

Microsoft paraphrase data: 5800 pairs of sentences



- Detects from scratch (unsupervised).
- Incorporates context.

Results using Sentence Embeddings

Sentiment Analysis

Method	MR	SUBJ
MNB	79.0	93.6
Paragraph-vector	74.8	90.5
Skip-thought	75.5	92.1
ConvDic+DeconvDec	**78.9**	**92.4**

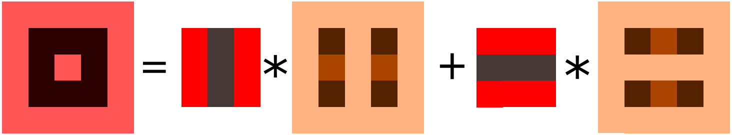
Paraphrase Detection

Method	Outside Information	F score
Vector Similarity	word similarity	75.3%
RMLMG	syntacticinfo	80.5%
ConvDic+DeconvDec	none	**80.7%**
Skip-thought	book corpus	81.9%

Image Pattern Learning through Tensor Factorization

- 2-D Convolutional Model

$$X = \sum_{j=1}^{L} V_j^* * W_j^*$$



Slides in this section prepared by Y. Shi

Image Pattern Learning through Tensor Factorization

Key points:

- Recall: 1-D circulant matrix eigen decomposition corresponds to 1-D Discrete Fourier Transform

Image Pattern Learning through Tensor Factorization

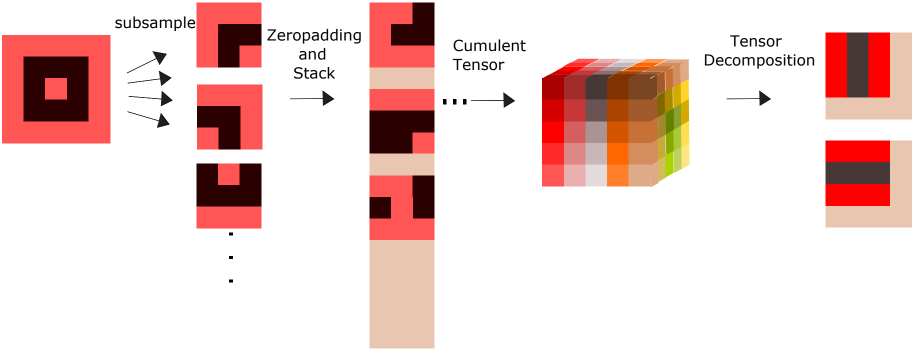
Key points:

- Recall: 1-D circulant matrix eigen decomposition corresponds to 1-D Discrete Fourier Transform
- 2-D circulant matrix eigen decomposition:
  $\mathsf{Cir}_{\mathsf{2-d}}(V) = (U \otimes U)\mathsf{Diag}(\mathsf{DFT}_{\mathsf{2-d}}(V))(U \otimes U)^{\mathsf{H}}$
- The eigenvector matrix for 2-D circulant matrix is $U \otimes U$.

Image Pattern Learning through Tensor Factorization

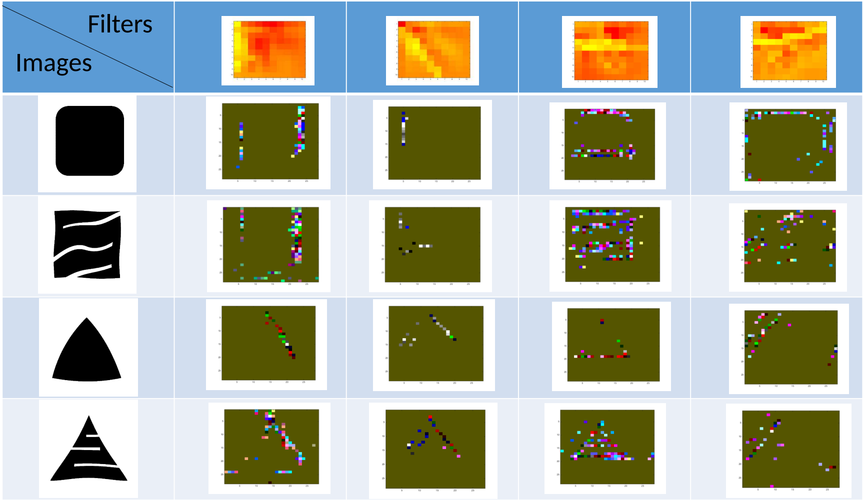
Main algorithm:

1. Subsample and batch
2. Form third order cumulant tensor
3. Tensor factorization



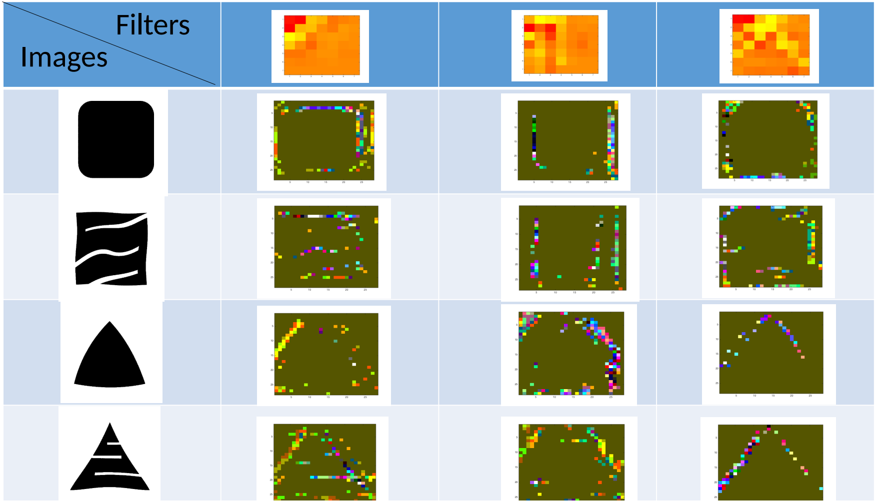
MPEG7 Dataset

- Image/activation map size: 28 X 28, filter size: 10 X 10
- First layer filters.



MPEG7 Dataset

- Image/activation map size: 28 X 28, filter size: 10 X 10
- Second layer filters after max-pooling.

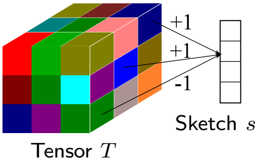


Outline

Tensor Sketches

Scaling up

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

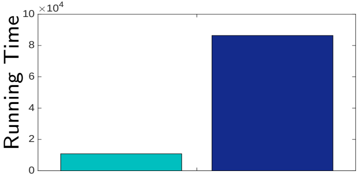


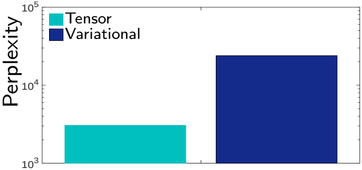
- Wang, Tung, Smola, A. " Guaranteed Tensor Decomposition via Sketching", NIPS'15.
- Neural Module Networks by J. Andreas, M. Rohrbach, T. Darrell, D. Klein, CVPR 2016.
- Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding by A. Fukui, D.H. Park, D. Yang, A. Rohrbach, T. Darrell, M. Rohrbach, CVPR 2016.

Tensors vs. Variational Inference

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

Learning Topics from PubMed on Spark, 8mil articles



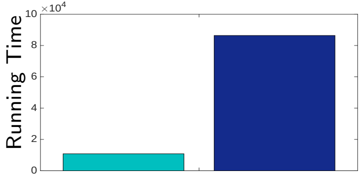


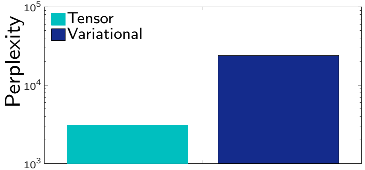
F. Huang, U.N. Niranjan, M. Hakeem, A, "Online tensor methods for training latent variable models," JMLR 2014.

Tensors vs. Variational Inference

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

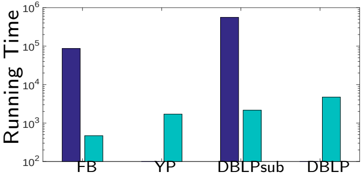
Learning Topics from PubMed on Spark, 8mil articles

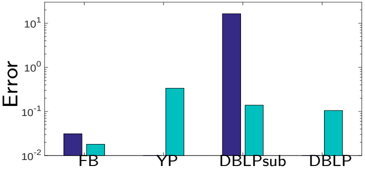




Learning network communities on single workstation

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



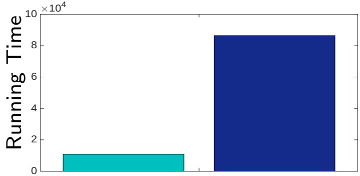


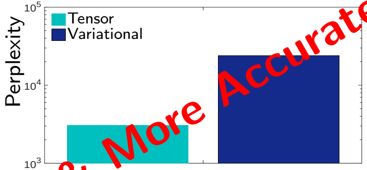
F. Huang, U.N. Niranjan, M. Hakeem, A, "Online tensor methods for training latent variable models," JMLR 2014.

Tensors vs. Variational Inference

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

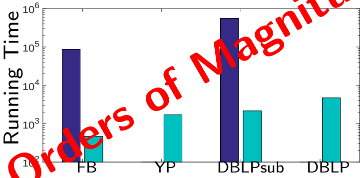
Learning Topics from PubMed on Spark, 8mil articles

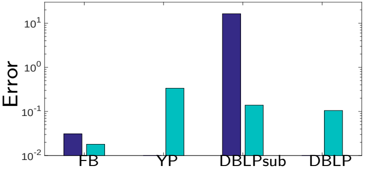




Learning network communities on single workstation

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



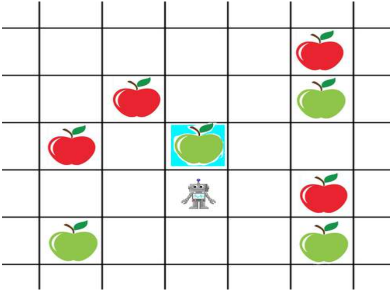


Orders of Magnitude Faster & More Accurate

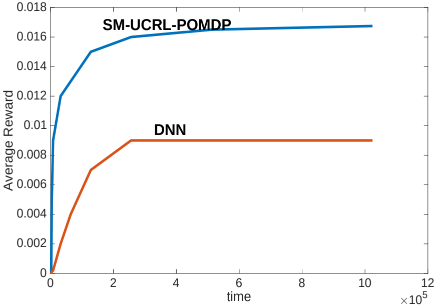
F. Huang, U.N. Niranjan, M. Hakeem, A, "Online tensor methods for training latent variable models," JMLR 2014.

Playing Atari Game

Observation Window



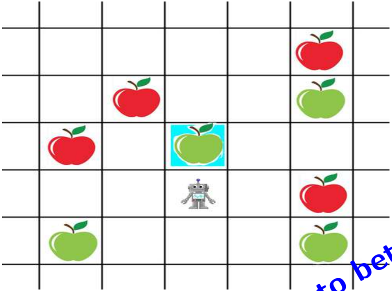
Average Reward vs. Time.



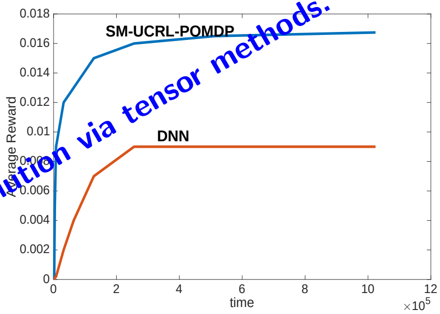
- POMDP model with 3 hidden states (trained using tensor methods) vs. NN with 3 hidden layers 10 neurons each (trained using RmsProp).

Playing Atari Game

Observation Window



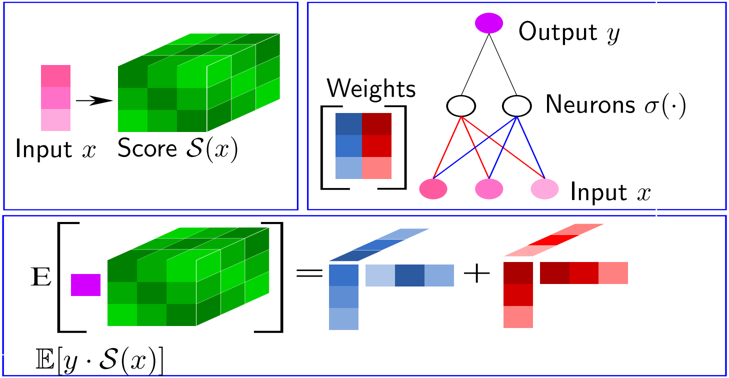
Average Reward vs. Time.



Faster convergence to better solution via tensor methods.

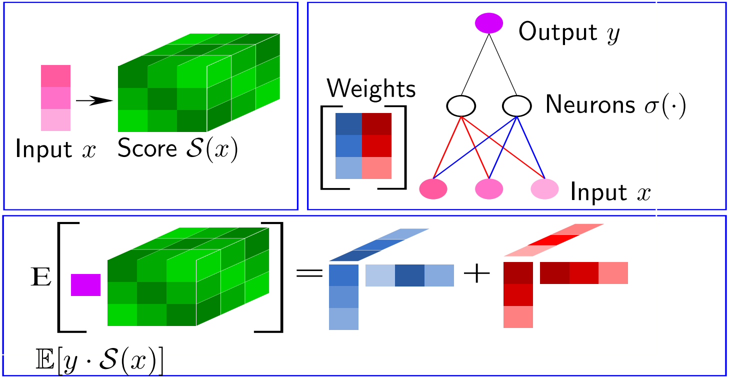
- POMDP model with 3 hidden states (trained using tensor methods) vs. NN with 3 hidden layers 10 neurons each (trained using RmsProp).

Training Neural Networks with Tensors

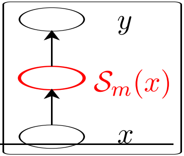


M. Janzamin, H. Sedghi, and A., "Beating the Perils of Non-Convexity: Guaranteed Training of Neural Networks using Tensor Methods," June. 2015.

Training Neural Networks with Tensors



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



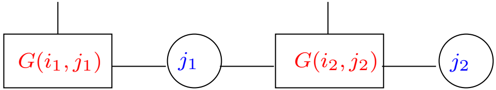
Gaussian $x \Rightarrow$ Hermite polynomials.

M. Janzamin, H. Sedghi, and A., "Beating the Perils of Non-Convexity: Guaranteed Training of Neural Networks using Tensor Methods," June. 2015.

Compression of Neural Networks using Tensors

- Multi-linear representation of dense layers of CNNs.
  - Tensor train format for low rank approximation of weight matrix.
- Compact representation: solves memory problem.

$$Y(i_1, i_2 \ldots) = \sum_{j_1, j_2 \ldots} G(i_1, j_1) G(i_2, j_2) \ldots X(j_1, j_2 \ldots)$$



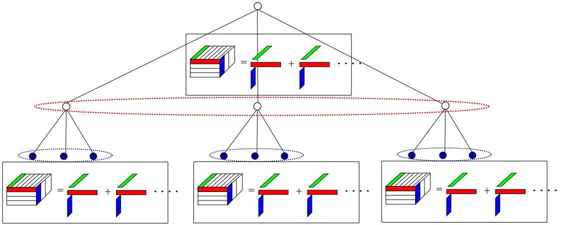
Results on ImageNet

- Compression rate $200,000!$
- Negligible performance loss.

A. Novikov, D. Podoprikhin, A. Osokin, D. Vetrov, "Tensorizing Neural Networks", NIPS 2015.

Tensors for Expressivity of Convnets

- Hierarchical tensors for representing arithmetic conv. nets.
- Employs locality, sharing and pooling.
- Exponentially more parameters in shallow net vs. deep net.

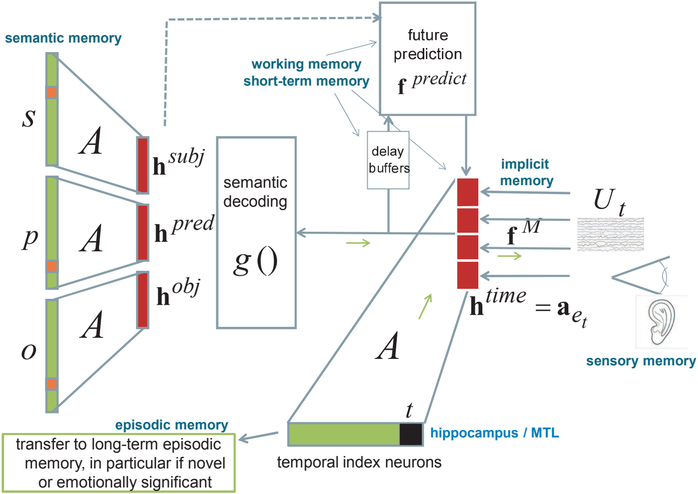


N. Cohen, O. Sharir, A. Shashua, "Deep SimNets" CVPR 2016.

N. Cohen, O. Sharir, A. Shashua, "On the Expressive Power of Deep Learning: A Tensor Analysis" COLT 2016.

Tensors in Memory Embeddings

Human Memory Model. Semantic decoding through Tensor Tucker.



V. Tresp, C. Esteban, Y. Yang, S. Baier and D. Krompab, "Learning with Memory Embeddingss ", 2015

Outline

Conclusion

Guaranteed Non-convex Optimization

- Non-convex optimization requires new theoretical frameworks.
- Matrix and tensor methods have desirable guarantees on reaching global optimum.
  - Applicable to unsupervised, supervised and reinforcement learning.
  - Polynomial computational and sample complexity.
  - Faster and better performance in practice.

Steps Forward

- Scaling up tensor methods: sketching algorithms, extended BLAS, ...
- Incorporating other invariance constraints into tensor methods

Resources and Research Connections

- http://www.offconvex.org/ blog.
- https://www.facebook.com/nonconvex group.
- http://newport.eecs.uci.edu/anandkumar/
- ICML and NIPS workshops.

Resources and Research Connections

- http://www.offconvex.org/ blog.
- https://www.facebook.com/nonconvex group.
- http://newport.eecs.uci.edu/anandkumar/
- ICML and NIPS workshops.

Collaborators

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