

Stat 212b: Topics in Deep Learning

Lecture 24

Joan Bruna
UC Berkeley



Objective

- Tensor Decompositions and Deep Learning
 - Optimality certificates
 - Learning with high-order score function.
 - Hierarchical Tensor Decompositions
- Spin Glasses and Deep Learning
- Richard Zhang: “Colorful Image Colorization”
- Hoang Duong: “Learning Polynomial Factorization”

Tensor Methods in Deep Learning

- Optimizing the training error with a generic deep network is a non-convex problem.

$$\min_{\Theta} \frac{1}{n} \sum_{i \leq n} \ell(y_i, \Phi(x_i; \Theta)) + \mathcal{R}(\Theta) .$$

- Consider a network of depth d with ReLU nonlinearities. Seen as a function of its parameters Θ , $\Phi(x; \Theta)$ resembles a homogeneous “piece-wise” polynomial:

$$\Phi(x; \Theta) = \sum_p \pi(x; \Theta) x_{p(1)} \prod_{j=1}^d \Theta_{p(j)}^j , \quad \pi(x; \Theta) = \{0, 1\} .$$
$$\Theta = \{\Theta^1, \dots, \Theta^d\} .$$

Tensor Methods in Deep Learning

- Optimizing the training error with a generic deep network is a non-convex problem.

$$\min_{\Theta} \frac{1}{n} \sum_{i \leq n} \ell(y_i, \Phi(x_i; \Theta)) + \mathcal{R}(\Theta) .$$

- Consider a network of depth d with ReLU nonlinearities. Seen as a function of its parameters Θ , $\Phi(x; \Theta)$ resembles a homogeneous “piece-wise” polynomial:

$$\Phi(x; \Theta) = \sum_p \pi(x; \Theta) x_{p(1)} \prod_{j=1}^d \Theta_{p(j)}^j , \quad \pi(x; \Theta) = \{0, 1\} .$$
$$\Theta = \{\Theta^1, \dots, \Theta^d\} .$$

- The dependencies on Θ are partly captured by the d -order tensor $\Theta^1 \otimes \Theta^2 \dots \otimes \Theta^d$.

Tensor Methods

$$\min_{\Theta^1, \dots, \Theta^d} F(Y, \Psi_X(\Theta^1, \dots, \Theta^d)) + \mathcal{R}(\Theta^1, \dots, \Theta^d) .$$

- Tensor factorizations are a broad class of non-convex optimization problems.

Tensor Methods

$$\min_{\Theta^1, \dots, \Theta^d} F(Y, \Psi_X(\Theta^1, \dots, \Theta^d)) + \mathcal{R}(\Theta^1, \dots, \Theta^d) .$$

- Tensor factorizations are a broad class of non-convex optimization problems.
- A particularly famous instance is the matrix factorization problem:

$$\min_{U, V} \ell(Y, UV^T) + \mathcal{R}(U, V) , \quad Y \in \mathbb{R}^{n \times m}, U \in \mathbb{R}^{n \times d}, V \in \mathbb{R}^{m \times d} .$$

- Low-rank factorizations (e.g. PCA)
- Sparse factorizations (Dictionary Learning, NMF)

Motivation: Matrix factorization

- Example: low-rank factorization.

$$\min_{U,V} \ell(Y, UV^T) , \text{ s.t. } \text{rank}(UV^T) \leq r .$$

- When $\ell(Y, X) = \|Y - X\|_{op}$, $\ell(Y, X) = \|Y - X\|_F$ OK
- We can *lift* the problem and relax the constraint:

$$\min_X \ell(Y, X) + \lambda \|X\|_* , \quad \|X\|_* = \text{Nuclear norm of } X .$$

- Factorized and relaxed formulations are connected via a variational principle:

$$\|X\|_* = \min_{UV^T=X} \frac{1}{2} (\|U\|_F^2 + \|V\|_F^2) .$$

- Q: General case?

Tensor Norms [Bach, Haeffele&Vidal]

- A first generalization is the tensor norm

$$\|X\|_{u,v} = \inf_r \min_{UV^T=X} \frac{1}{2} \left(\sum_i \|U_i\|_u^2 + \|V_i\|_v^2 \right).$$

Theorem [H-V]: A local minimizer of the factorized problem
 $\min_{U,V} \ell(Y, UV^T) + \lambda \sum_{i \leq r} \|U_i\|_u \|V_i\|_v$
such that for some i $U_i = V_i = 0$ is a global minimizer of the convex problem $\min_X \ell(Y, X) + \lambda \|X\|_{u,v}$ as well as the factorized problem.

- This produces an *optimality certificate*: we use a surrogate convex problem to obtain a guarantee that a non-convex problem is solved optimally.

From Tensor Factorizations to Deep Nets

- We start by generalizing a multilinear mapping (tensor) to homogeneous maps $\phi(\Theta^1, \dots, \Theta^d)$:

$$\forall \Theta, \forall \alpha \geq 0, \phi(\alpha\Theta^1, \dots, \alpha\Theta^d) = \alpha^s \phi(\Theta^1, \dots, \Theta^d) .$$

s: degree of homogeneity.

Ex: ReLU $\rho(x) = \max(0, x)$ is homogeneous of degree 1.

- We construct models by adding r copies of homogeneous maps:

$$\Phi_r(\Theta^1, \dots, \Theta^d) = \sum_{i \leq r} \phi(\Theta_i^1, \dots, \Theta_i^d) .$$

- We consider

$$\min_{\Theta^1, \dots, \Theta^d} \ell(Y, \Phi_r(\Theta^1, \dots, \Theta^d)) + \lambda \mathcal{R}(\Theta^1, \dots, \Theta^d) ,$$

Key assumption: \mathcal{R} is positively homogeneous of the same degree as Φ .

From Tensor Factorizations to Deep Nets

$$\Phi_r(\Theta^1, \dots, \Theta^d) = \sum_{i=1}^r \phi(\Theta^1, \dots, \Theta^d) .$$

Matrices:

$$\Phi(U, V) = UV^T = \sum_{i=1}^r U_i V_i^T \quad (\phi(U_i, V_i) = U_i V_i^T) .$$

Higher-order Tensors:

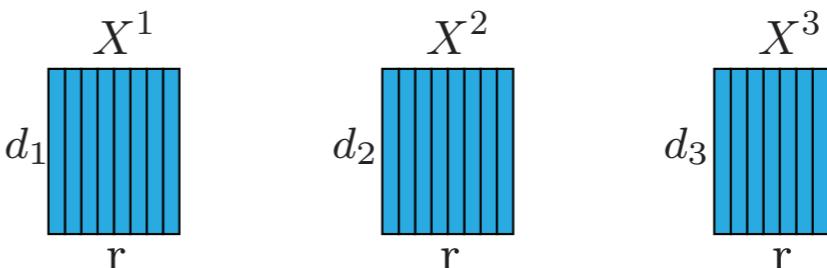


figure credit:
R. Vidal

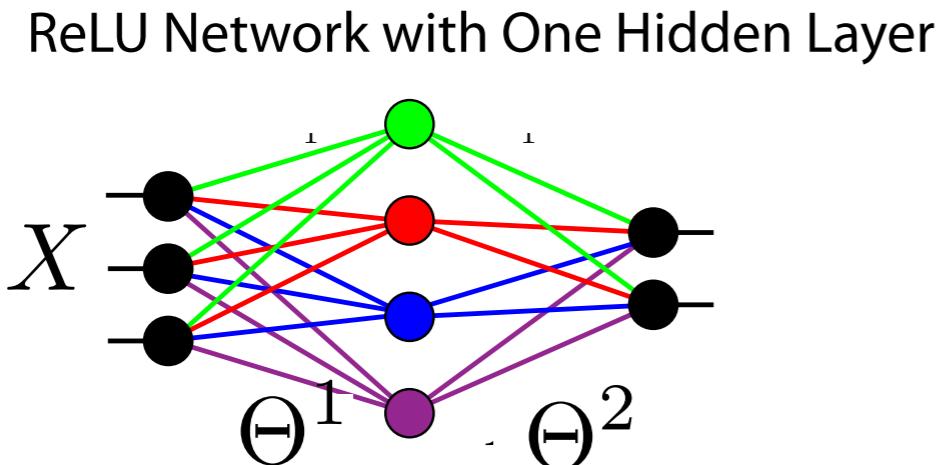
$$\phi(\Theta_i^1, \dots, \Theta_i^d) = \Theta_i^1 \otimes \dots \otimes \Theta_i^d .$$

$\Phi_r(X^1, X^2, X^3)$

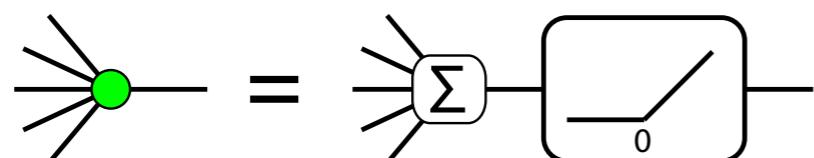
Candecomp/Parafac (CP) Tensor decomposition.

Adaptation to Deep Models

- ReLU Network:



Rectified Linear Unit (ReLU)



$$\Phi(\Theta^1, \dots, \Theta^d) = \sum_i \phi(\Theta_i^1, \dots, \Theta_i^d) .$$

figure credit:
R. Vidal

Multilayer ReLU
Parallel Network

Adaptation to Deep Models

- In the matrix case, the variational principle was

$$\|X\|_{u,v} = \min_{UV^T=X} \sum_{i \leq r} \|U_i\|_u \|V_i\|_v .$$

- This is generalized to

$$\mathcal{R}(\Theta) = \min_{\Theta^1, \dots, \Theta^d} \sum_{i \leq r} g(\Theta_i^1, \dots, \Theta_i^d) , \text{ s.t. } \Phi_r(\Theta^1, \dots, \Theta^d) = \Theta .$$

- **Proposition [H-V]:** \mathcal{R} is convex.
Also, if g is positively homogeneous of degree s , so is \mathcal{R} .

Adaptation to Deep Models

Theorem [H-V]: A local minimizer of the factorized problem

$$\min_{\Theta^k} \ell(Y, \sum_{i \leq r} \phi_r(\Theta_i^k)) + \lambda \sum_{i \leq r} g(\Theta_i^k)$$

such that for some i and all k $\Theta_i^k = 0$ is a global minimizer for both factorized problem and the convex formulation

$$\min_{\Theta} \ell(Y, \Theta) + \lambda \mathcal{R}(\Theta).$$

- Global optimality certificate for a broad class of non-convex optimization problems, including some form of deep learning architectures.
- Q: How to use this certificate in practice?

Adaptation to Deep Models

- Pros
 - Global optimality certificate, easy to check
 - Includes nonlinear models as long as they are homogeneous.
 - Provides a possible meta-algorithm: increase the lifting value r progressively if local optimum does not verify condition.
- Cons
 - How much do we need to increase r in practice?
 - How stringent is the homogenous regularization condition?

Tensor Decompositions and Neural Nets

- Suppose a label generating model of the form

$$\mathbb{E}(y|x) = f_0(x) = \langle a_2, \sigma(A_1 x + b_1) \rangle + b_2 ,$$

$\sigma(\cdot)$: point-wise nonlinearity
 $A_1 \in \mathbb{R}^{d \times k}$.

- Q: Given training samples $\{(x_i, y_i); y_i = f_0(x_i)\}_{i \leq n}$, can we estimate the parameters a_2, A_1, b_1, b_2 with provable risk?
- Q: Using a computationally efficient algorithm?

Breaking the Perils of (...)

[Janzamin, Sedghi, Anandkumar]

- If one assumes knowledge of the input distribution $p(x)$, then one can exploit the relationship between score functions and conditional expectations:

Def: The m -th order score function $S_m(x)$ is the m -th order tensor

$$S_m(x) = (-1)^m \frac{\nabla^m p(x)}{p(x)} .$$

Proposition: If $f(x) = \mathbb{E}(y|x)$, then

$$\mathbb{E}(y \cdot S_3(x)) = \mathbb{E}(\nabla^3 f(x)) .$$

Breaking the Perils of (...)

[Janzamin, Sedghi, Anandkumar]

- If one assumes knowledge of the input distribution $p(x)$, then one can exploit the relationship between score functions and conditional expectations.
- It results that when $\mathbb{E}(y|x) = f_0(x)$, we have

$$\mathbb{E}(y \cdot S_3(x)) = \sum_{j \leq k} \lambda_j (A_1)_j \otimes (A_1)_j \otimes (A_1)_j \in \mathbb{R}^{d \times d \times d}, \quad \lambda_j \in \mathbb{R}.$$

Breaking the Perils of (...)

[Janzamin, Sedghi, Anandkumar]

- Learning generalization bound in the “realizable” setting:

Theorem: The tensor algorithm *NN-Lift* learns the target function $\mathbb{E}(y|x) = f_0(x)$ up to error ϵ when the number of samples is of the order of

$$n \geq O\left(\frac{kd^3}{\epsilon^2} \frac{\lambda_{max}(A_1)^2}{\lambda_{min}(A_1)^6}\right).$$

(k : size of hidden layer)
(d : input dimension)

- Comments:
 - Polynomial sample complexity.
 - Algorithm has polynomial complexity as well.
 - Extension to non-realizable setting (see paper for details).

Breaking the Perils of (...)

[Janzamin, Sedghi, Anandkumar]

- Pros

- Statistical Guarantees that also incorporate computational feasibility.
- Learning is essentially reduced to finding low-rank tensor factorizations.

- Cons

- very strong hypothesis: knowledge of $p(x)$.
- only a particular Neural network architecture (one hidden layer so far).
- restrictive class of nonlinearities? : the proof needs

$$\mathbb{E}(\sigma'''(z)) , \mathbb{E}(\sigma''(z))$$

Deep Nets and Hierarchical Tensor Decompositions

[Cohen, Sharir, Shashua'15]

- Consider an input image x and its features extracted on localized patches:

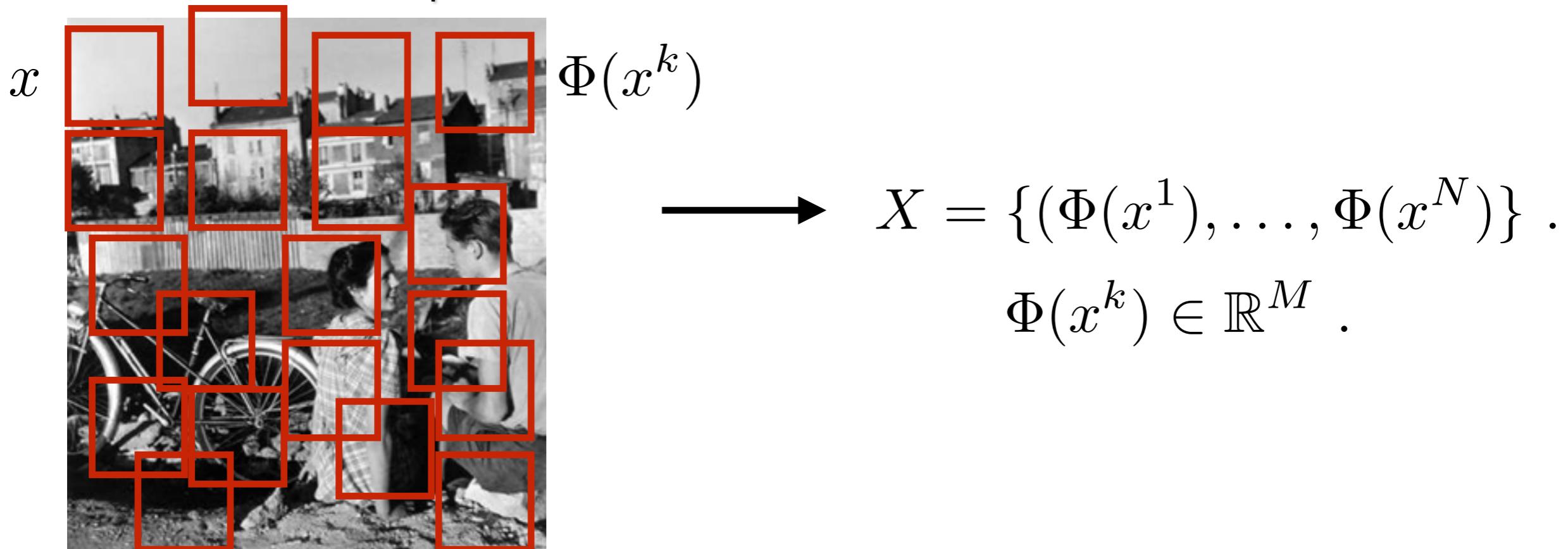
x



Deep Nets and Hierarchical Tensor Decompositions

[Cohen, Sharir, Shashua'15]

- Consider an input image x and its features extracted on dense, localized patches:



- Aggregate features by combining high-order information:

$$p(y|x) = \sum_{d_1, \dots, d_N=1}^M A_{d_1, \dots, d_N}^y \prod_{i=1}^N \Phi_{d_i}(x^i) ,$$

A^y : N -th order tensor
of dimensions $M_k = M$.

Deep Nets and Hierarchical Tensor Decompositions

[Cohen, Sharir, Shashua'15]

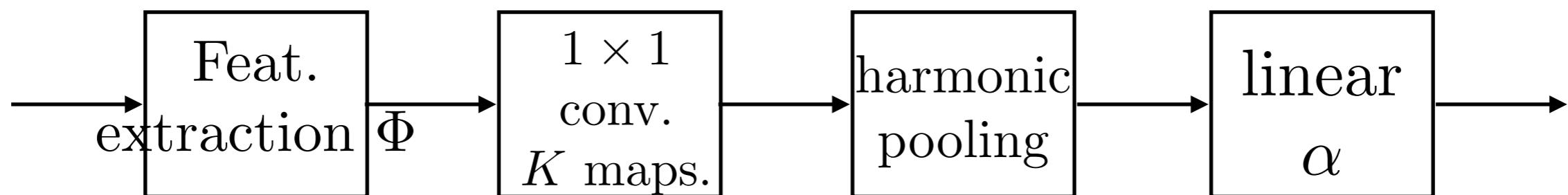
- Q: How to parametrize/factorize the tensors A^y ?
- CP (Candecomp/Parafac) decomposition:

$$A = \sum_{k=1}^K \alpha_k a_1^k \otimes a_2^k \otimes \dots \otimes a_N^k, \quad a_i^k \in \mathbb{R}^M.$$

sum of K rank-1 N -th order tensors of size M .

- The resulting model is a shallow network:

$$p_A(y|x) = \sum_{k=1}^K \alpha_k^y \prod_{i=1}^N \left(\sum_{m=1}^M a_i^k(m) \Phi_m(x^i) \right).$$



Deep Nets and Hierarchical Tensor Decompositions

[Cohen, Sharir, Shashua'15]

- Q: How to parametrize/factorize the tensors A^y ?
- Hierarchical-Tucker (HT) decompositions:

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

...

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

...

$$A^y = \sum_{\alpha=1}^{r_{L-1}} a_{\alpha}^{L,j,\gamma} \phi^{L-1,2j-1,\alpha} \otimes \phi^{L-1,2j,\alpha}, \text{ order } 2^L = N.$$

– Corresponds to a deep representation with $L = \log N$ layers.

Deep Nets and Hierarchical Tensor Decompositions

[Cohen, Sharir, Shashua'15]

- In both decompositions, given enough terms, any tensor can be approximated arbitrarily well.
- Depth efficiency question: for tensors that require a polynomial size in the HT decomposition, how many parameters in the CP representation do we need? and viceversa?

Deep Nets and Hierarchical Tensor Decompositions

[Cohen, Sharir, Shashua'15]

Theorem: Let A be a tensor of order N and dimension M in each slice, generated by the HT formula using ranks $r_l = r = O(M)$.

Then A will have CP-rank at least $r^{N/2}$ almost everywhere.

- The HT space with rank r blocks has $O(r^2N)$ parameters.
- Besides a negligible set, all functions that can be realized by a polynomially sized HT model require exponential size in order to be approximated by a CP model.
- The converse is not true: a CP model of size $O(NMK)$ can be represented in HT with $O(NK \max(K, M)) \simeq O(NK^2)$

Deep Nets and Hierarchical Tensor Decompositions

[Cohen, Sharir, Shashua'15]

- Pros

- Framework that explains that depth efficiency is universal: *all* hierarchical decompositions require exponentially more effort to parametrize using non-hierarchical factorizations.
- Role of Convolution: weight sharing in a CP decomposition reduces to symmetric tensors. Not the case in the HT decomposition.

- Cons

- Nonlinearities are multiplicative in this model: numerically and statistically unstable. Logarithms do not fully resolve instability.
- Approximation error results. Interplay with estimation and optimization error?

Deep Networks and Spin Glasses

[Choromaska, Henaff, Mathieu, LeCun, Ben Arous, '14]

- Suppose we have a linear deep network:

$$\Phi(x; \Theta_1, \dots, \Theta_K) = \Theta_K \Theta_{K-1} \dots \Theta_1 x .$$

- And suppose we train using least squares regression:

$$E(\Theta) = \frac{1}{n} \sum_{i \leq n} \|y_i - \Phi(x_i; \Theta)\|^2 .$$

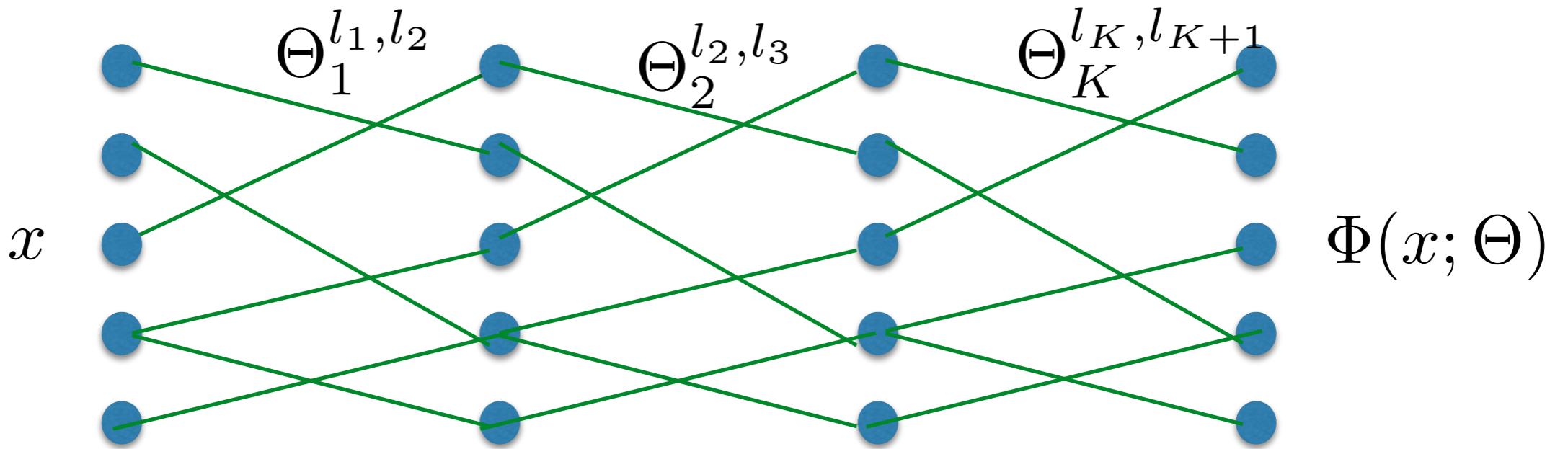
- In coordinates, $(\Theta_1 x)^j = \sum_l \Theta_1^{j,l} x^l ,$

$$(\Theta_2 \Theta_1 x)^j = \sum_{l_1, l_2} \Theta_2^{j, l_2} \Theta_1^{l_2, l_1} x^{l_1} ,$$

$$(\Theta_K \dots \Theta_2 \Theta_1 x)^j = \sum_{l_1, \dots, l_K} x^{l_1} \Theta_K^{j, l_K} \prod_{k=2}^{K-1} \Theta_k^{l_k, l_{k-1}} .$$

Deep Networks and Spin Glasses

[Choromaska, Henaff, Mathieu, LeCun, Ben Arous, '14]

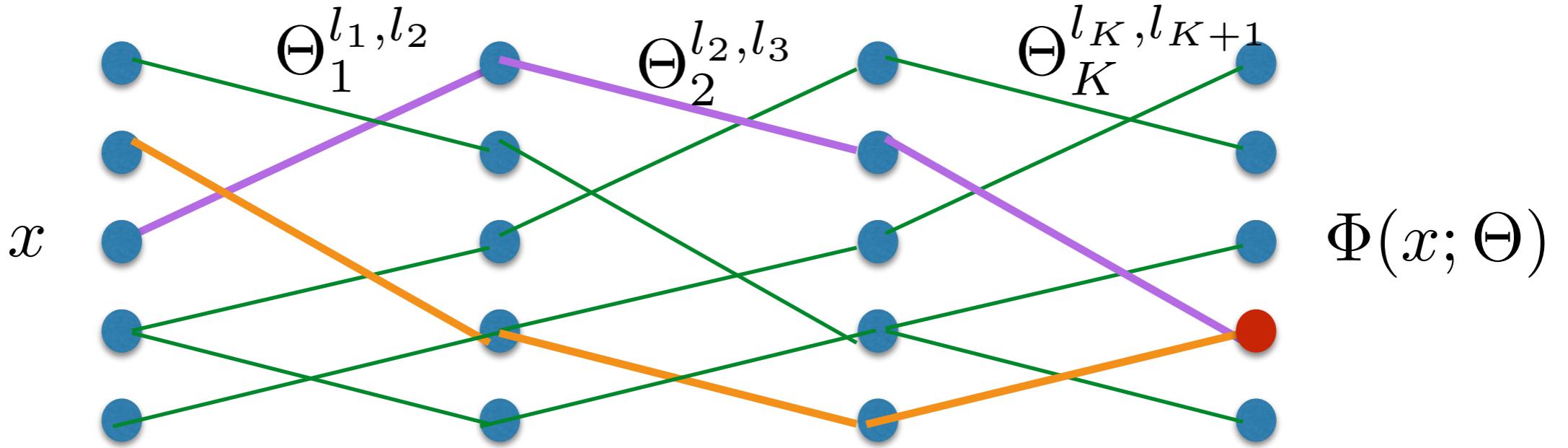


- Equivalently, we can define paths $p = (l_0, l_1, \dots, l_{K+1})$

$$\mathcal{P} = \{p = (l_0, \dots, l_{K+1}); 1 \leq l_k \leq M_k\}$$

Deep Networks and Spin Glasses

[Choromaska, Henaff, Mathieu, LeCun, Ben Arous, '14]



- Equivalently, we can define paths $p = (l_0, l_1, \dots, l_{K+1})$

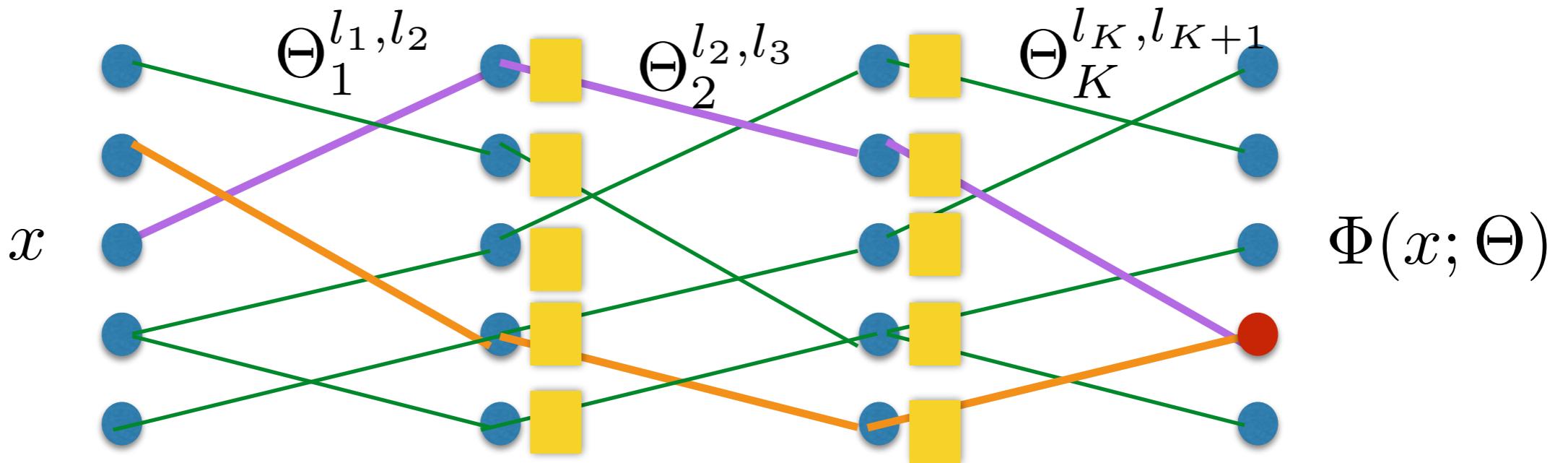
$$\mathcal{P} = \{p = (l_0, \dots, l_{K+1}); 1 \leq l_k \leq M_k\}$$

$$\Phi(x; \Theta)^j = \sum_{p \in \mathcal{P}; p(K+1)=j} x^{p(1)} \prod_{k \leq K} \Theta_k^{p(k), p(k+1)}.$$

- Homogeneous polynomial on Θ .
- Q: What about a ReLU network instead?

Deep Networks and Spin Glasses

[Choromaska, Henaff, Mathieu, LeCun, Ben Arous, '14]



- Now some paths will be stopped:

$$\text{Yellow square} : \rho(z) = \max(0, z).$$

$$\Phi(x; \Theta)^j = \sum_{p \in \mathcal{P}; p(K+1)=j} \pi(p, x, \Theta) \cdot x^{p(1)} \prod_{k \leq K} \Theta_k^{p(k), p(k+1)}, \quad \pi(p, x, \Theta) = \{0, 1\}$$

- $p = (l_0, \dots, l_K)$, $\tilde{p} = (l_0, \dots, l_{K-1})$

$$\pi(p, x, \Theta) = \pi(\tilde{p}, x, \Theta) \cdot \left(\sum_{p' \in \tilde{\mathcal{P}}; p'(K)=p(K)} \pi(p', x, \Theta) \prod_{k < K} \Theta_k^{p'(k), p'(k+1)} \right) > 0$$

- Biases produce low-order terms (we ignore them for now)

Deep Networks and Spin Glasses

[Choromaska, Henaff, Mathieu, LeCun, Ben Arous, '14]

- Loss becomes

$$E(\Theta) = \frac{1}{n} \sum_{i \leq n} \|y_i - \Phi(x_i; \Theta)\|^2$$

$$= \frac{1}{n} \sum_{i \leq n} \sum_{j=1}^{M_K} \left(y_i^j - \sum_{p \in \mathcal{P}; p(K+1)=j} \pi(p, x_i, \Theta) \cdot x_i^{p(1)} \prod_{k \leq K} \Theta_k^{p(k), p(k+1)} \right)^2$$

$$\begin{aligned} & \xrightarrow{n \rightarrow \infty} C + \sum_{p \in \mathcal{P}} q(X, Y, \Theta, p) \prod_{k \leq K} \Theta_k^{p(k), p(k+1)} \\ & + \sum_{p, p' \in \mathcal{P}} Q(X, \Theta, p, p') \prod_{k \leq K} \Theta_k^{p(k), p(k+1)} \Theta_k^{p'(k), p'(k+1)}, \text{ with} \end{aligned}$$

$$q(X, Y, \Theta, p) = \mathbb{E}_{X, Y} \left(\pi(p, X, \Theta) Y^{p(K)} X^{p(1)} \right),$$

$$Q(X, \Theta, p, p') = \mathbb{E}_X \left(\pi(p, X, \Theta) \pi(p', X, \Theta) X^{p(1)} X^{p'(1)} \right).$$

Deep Networks and Spin Glasses

[Choromaska, Henaff, Mathieu, LeCun, Ben Arous, '14]

- The loss “looks” like a polynomial in Θ provided we **break the dependency** of $\pi(p, x, \Theta)$ with respect to Θ .
 - It means that thresholding is independent of Θ .
- For large enough n (and assuming iid samples), it results that

$$q(X, Y, p) \sim \mathcal{N}(\mu_p, \sigma_p^2) ,$$
$$Q(X, p, p') \sim \mathcal{N}(\mu_{p,p'}, \sigma_{p,p'}^2) ,$$

Deep Networks and Spin Glasses

[Choromaska, Henaff, Mathieu, LeCun, Ben Arous, '14]

- Furthermore, if one also assumes *redundancy* (weights shared across layers), *uniformity* (same weights are not used too often along surviving paths) and *normalized weights*, authors arrive at

$$E(\Theta) \simeq \mathcal{L}_{\Lambda, K}(\Theta) = \frac{1}{\Lambda^{(K-1)/2}} \sum_{l_1, \dots, l_K=1}^{\Lambda} Z_{l_1, \dots, l_K} \Theta_{l_1} \dots \Theta_{l_K} ,$$

with $\|\Theta\|^2 = \Lambda$.

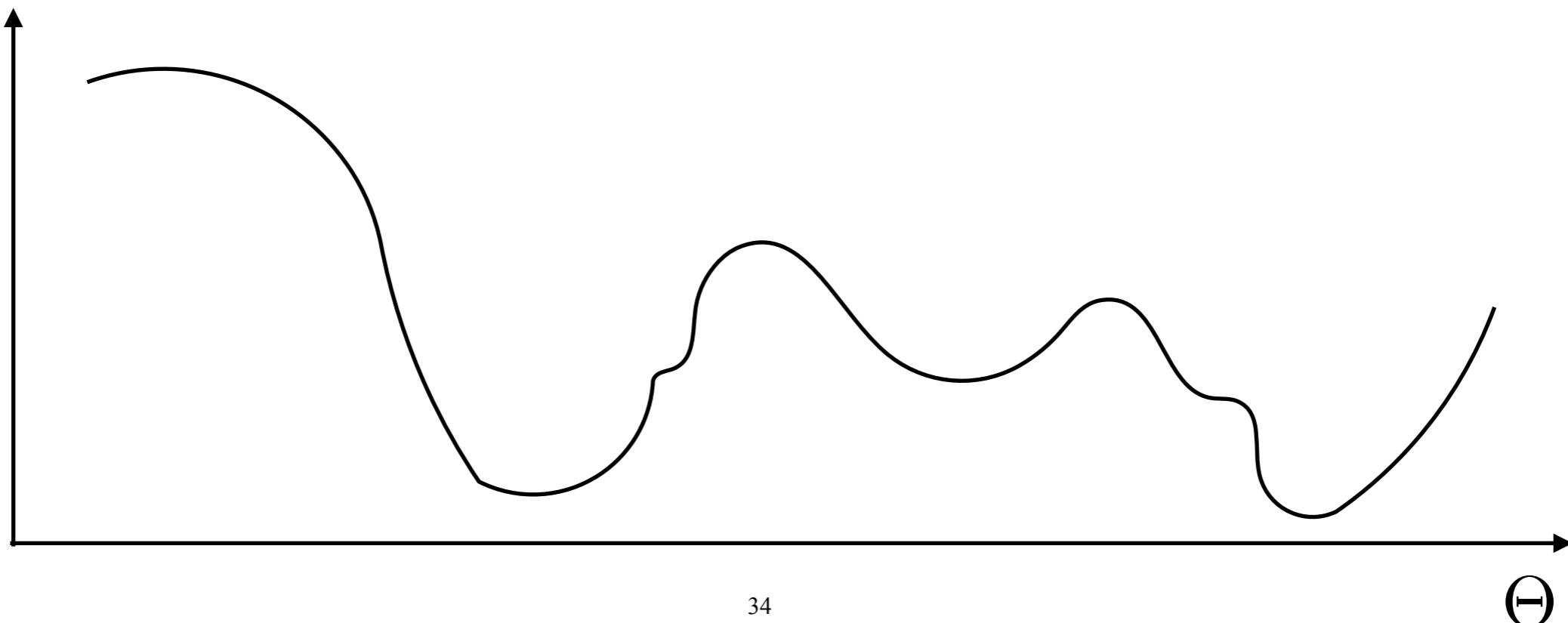
$\mathcal{L}_H(\Theta)$: Hamiltonian of the H -spin spherical spin glass model.

Deep Networks and Spin Glasses

[Choromaska, Henaff, Mathieu, LeCun, Ben Arous, '14]

- [Auffinger et al '10] [Auffinger, Ben Arous'13], obtained a complete description of the behavior of critical points of spherical spin glasses.

In particular, critical points (ratio of negative to positive eigenvalues of the Hessian) occur at different energy bands:

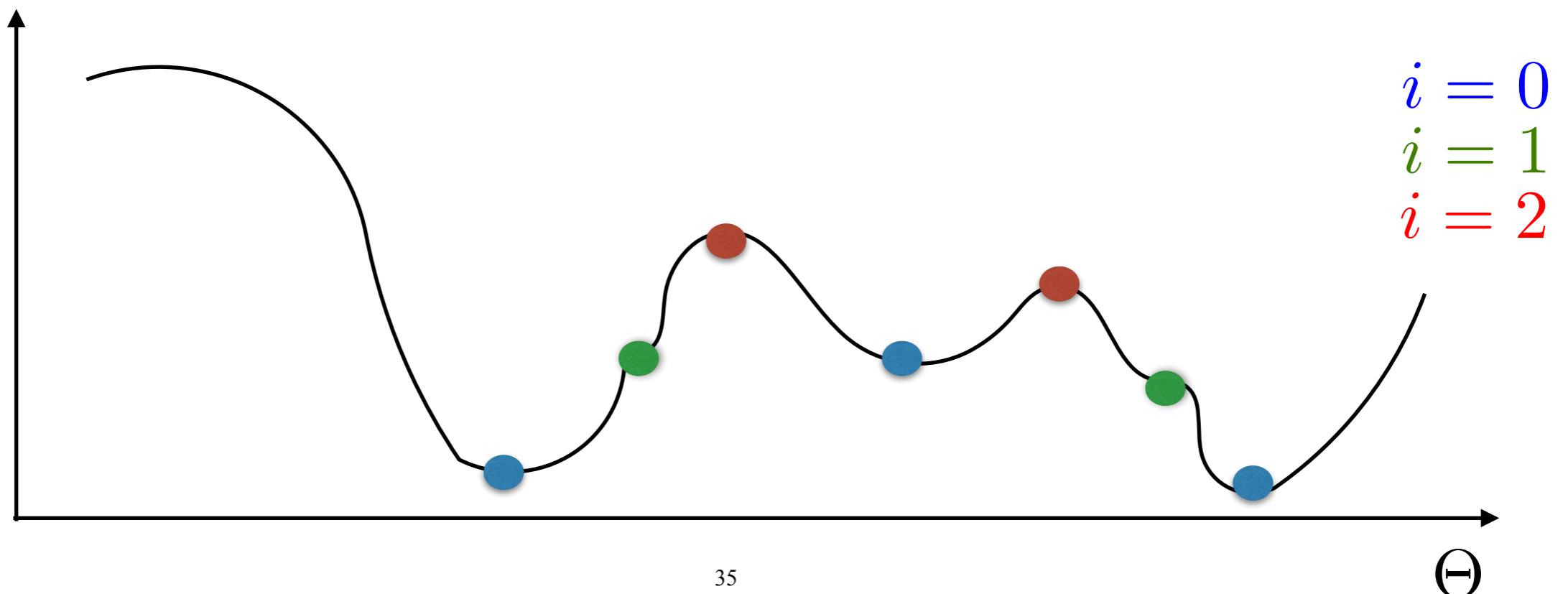


Deep Networks and Spin Glasses

[Choromaska, Henaff, Mathieu, LeCun, Ben Arous, '14]

- [Auffinger et al '10] [Auffinger, Ben Arous'13], obtained a complete description of the behavior of critical points of spherical spin glasses.

In particular, critical points (ratio of negative to positive eigenvalues of the Hessian) occur at different energy bands:

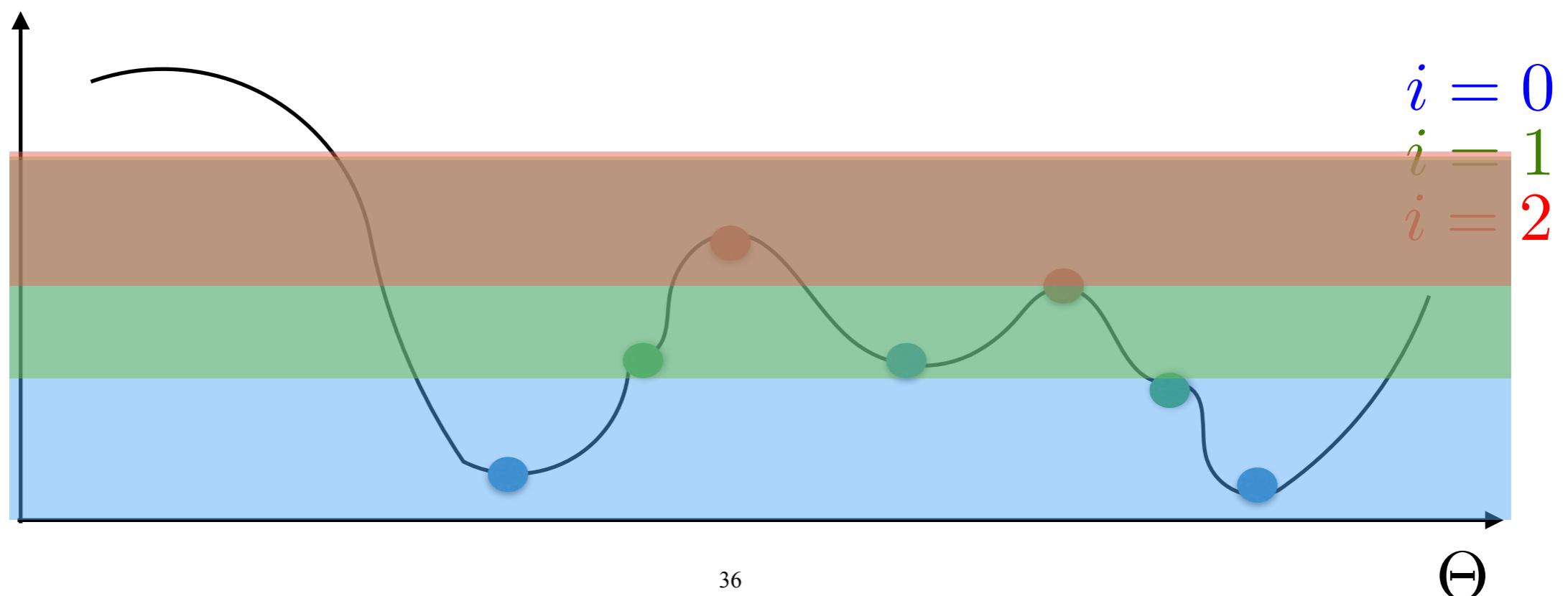


Deep Networks and Spin Glasses

[Choromaska, Henaff, Mathieu, LeCun, Ben Arous, '14]

- As $\Lambda \rightarrow \infty$, the distributions concentrate along different bands: each index concentrates in different bands.

As $\Lambda \rightarrow \infty$, the number of local minima dominate the rest of the indices.



Deep Networks and Spin Glasses

[Choromaska, Henaff, Mathieu, LeCun, Ben Arous, '14]

- Pros

- Macroscopic picture that explains some of the behavior of stochastic gradient descent on deep neural networks.
- Analysis tools from Random Matrix theory that explain non-local behavior and might complement invariance/symmetry arguments.

- Cons

- The simplifications on the model are very very strong.
- Does not inform about the role of convolutions in the energy landscape
- Does not really inform about the role of depth in the optimization.