A Theoretical Analysis on Feature Learning in Neural Networks: Emergence from Inputs and Advantage over Fixed Features

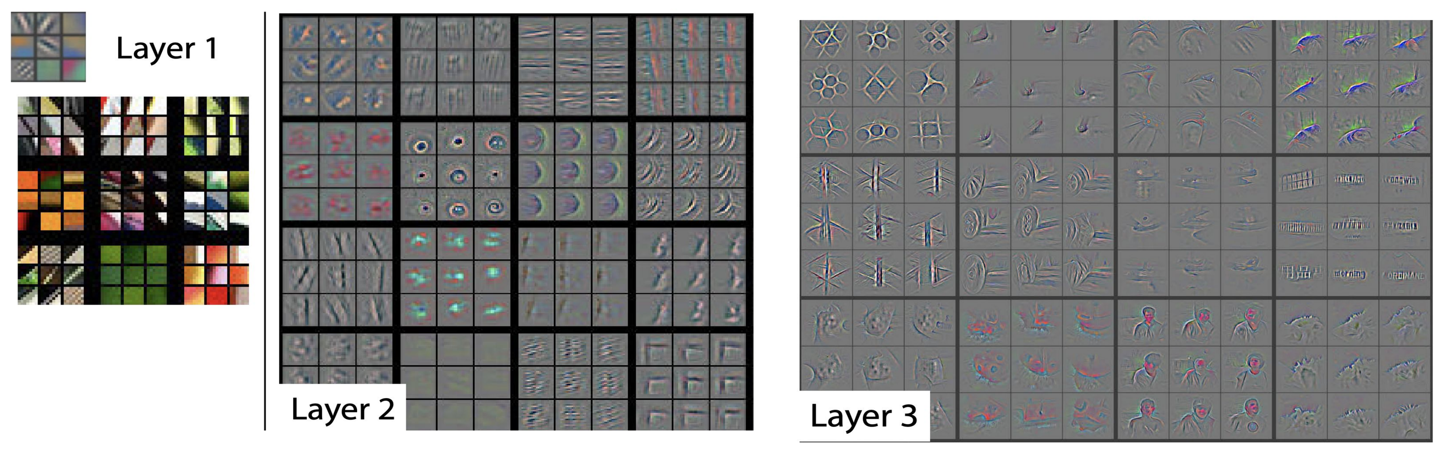
Zhenmei Shi*, Junyi Wei*, Yingyu Liang

University of Wisconsin-Madison

ICLR 2022

Motivation

- **Hidden Layers:** good representations of the inputs for prediction
- **Neurons:** correspond to interesting patterns in the inputs



Visualization of neurons in a convnet.

Figures from: Visualizing and Understanding Convolutional Networks, Zeiler and Fergus, ECCV'14.

Questions

- How features learned from inputs via gradient descent?
- Is learning features from inputs necessary for the superior performance?

Our results

- Propose a theoretical model of the data **with input structure**
- Prove network learning via gradient descent can succeed
- Prove **fixed feature** approaches fail
- Prove learning **without input structure** fails

Problem Setting

Dictionary Learning: input = sparse combination of base patterns

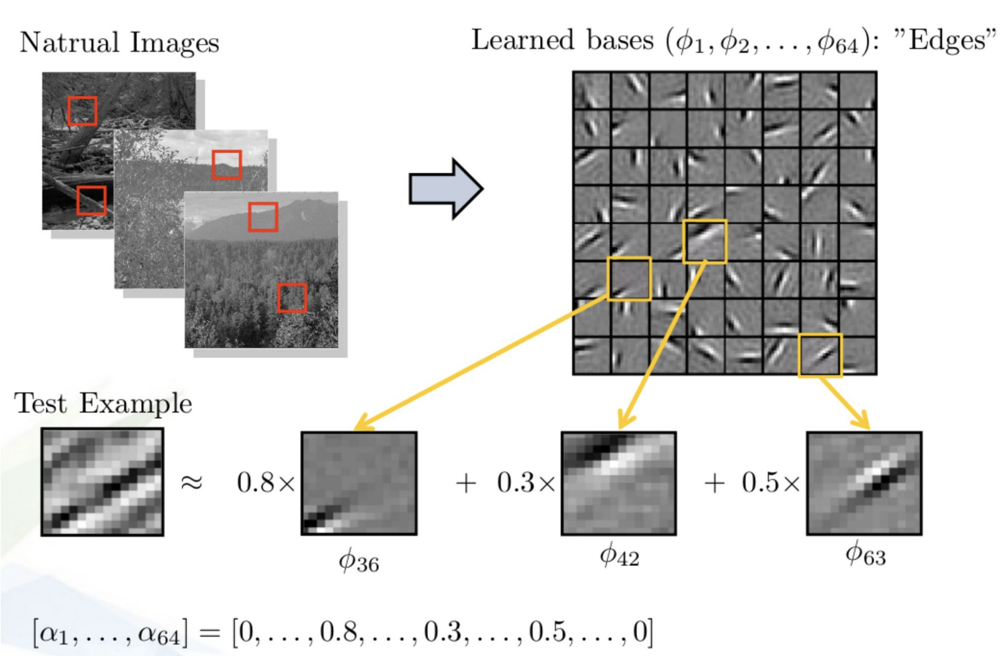


Figure from Brian Booth

- Input $x = M\phi$, with dictionary $M \in R^{d \times D}$, and pattern indicator $\phi \in \{0,1\}^D$
- Assume orthonormal M

Modeling the Labels: Relevant Pattern Counts

1. Sample ϕ from distribution D_ϕ
2. Generate input x using ϕ and the dictionary M
3. Generate label y using ϕ and A, P

Assumptions on D_ϕ

A. Balance classes: $\Pr[y=+1] = \Pr[y=-1] = \frac{1}{2}$

B. Relevant patterns: for any $i \in A$, $\gamma = \mathbf{E}[y\phi_i] - \mathbf{E}[y]E[\phi_i] > 0$

C. Irrelevant patterns: for any $i \notin A$, ϕ_i is i.i.d. with $p_0 = \Pr[\phi_i = 1]$

Network: 2-Layer, Hinge-loss, L2-regularizer, Gaussian init, Gradient descent

- Train a network: $g(x) = \sum_{i=1}^{2m} a_i \sigma(\langle w_i, x\rangle + b_i)$
- Activation: truncated ReLU $\sigma(z) = \min(1, \max(0, z))$

Network Learning Result

Theorem (informal)

For any $\epsilon, \delta \in (0,1)$, if

$$k = \Omega\left(\log^2 \frac{Dm}{\delta\gamma}\right), p_0 = \Omega\left(\frac{k^2}{D}\right), m \geq \max\left\{D,\ \Omega\left(\frac{k^{12}}{\epsilon^{1.5}}\right)\right\}$$

then with proper hyperparameters (e.g., step size), w.p. at least $1-\delta$ we can get a network with error at most ϵ.

1. **With** input structure, ***poly***-size 2-layer **neural networks** can **achieve small classification loss** with high probability.
2. Success comes from feature learning:
   - First learns and improves the neuron weights s.t. there is a good classifier on the neurons
   - Then learns a good classifier

Lower Bound for Fixed Feature Approach

- Fixed feature approach:
  - Let $\Psi(x) \in [-1,1]^N$ be any **data-independent** N-dim feature mapping
  - Linear models $h(x) = \langle \Psi(x), \theta \rangle$ with bounded weight $||\theta||_2 \leq B$

Theorem (informal)

There exist data distributions on which all such models h must have hinge-loss at least $p_0\left(1 - \frac{\sqrt{2NB}}{2^k}\right)$

There exist data distributions on which **all *poly*-size fixed feature** approaches **cannot achieve as small loss**.

Lower Bound for Without Input Structure

- Without input structure: sample ϕ uniformly from $\{0,1\}^D$
- Statistical Query (SQ) algorithms:
  - Asks statistical queries (Q, τ) about the data
  - Receives an estimation of $\Pr[Q(x,y) \text{ is true}]$ within error τ

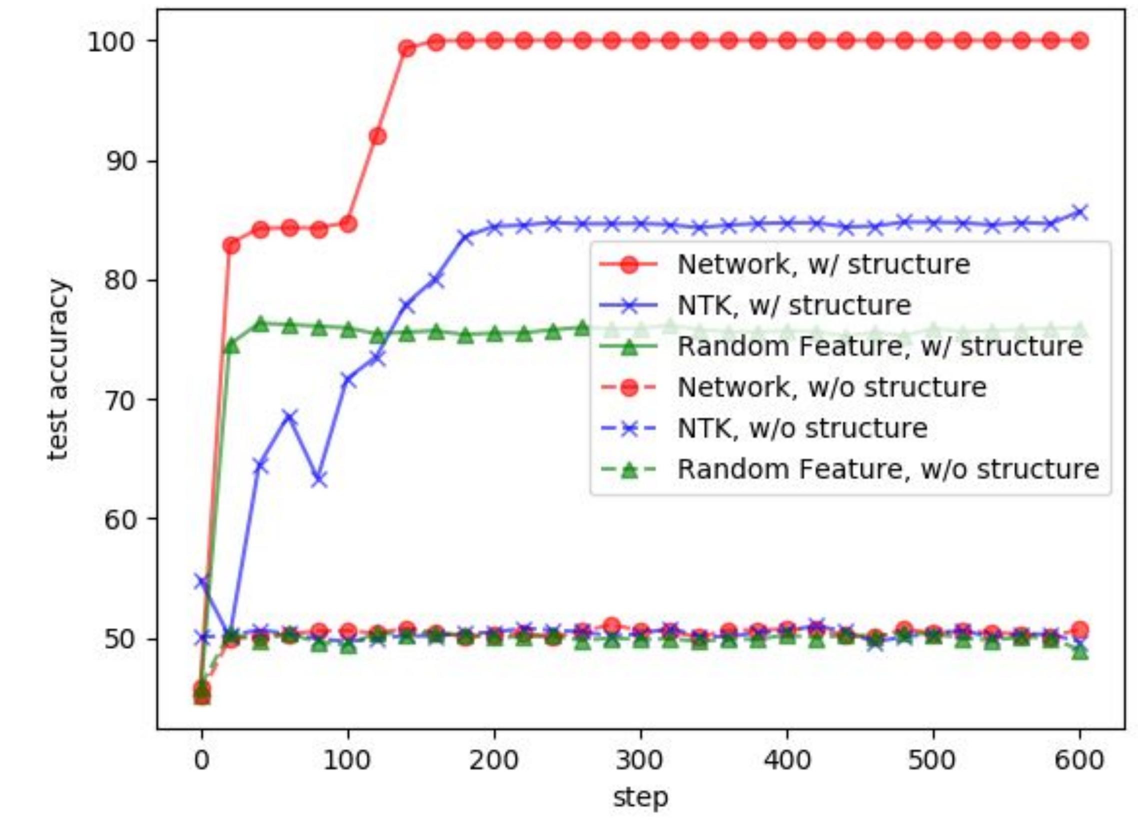
Theorem (informal)

For any SQ algorithm that can learn without the input structure to classification error less than $\frac{1}{2} - \frac{1}{\binom{D}{k}^3}$, either the number of queries or $\frac{1}{\tau}$ must be at least $\frac{1}{2}\binom{D}{k}^{1/3}$

Without input structure, **all *poly*** algo in the Statistical Query model (including networks and fixed features above) **cannot achieve as small loss**.

Experiments

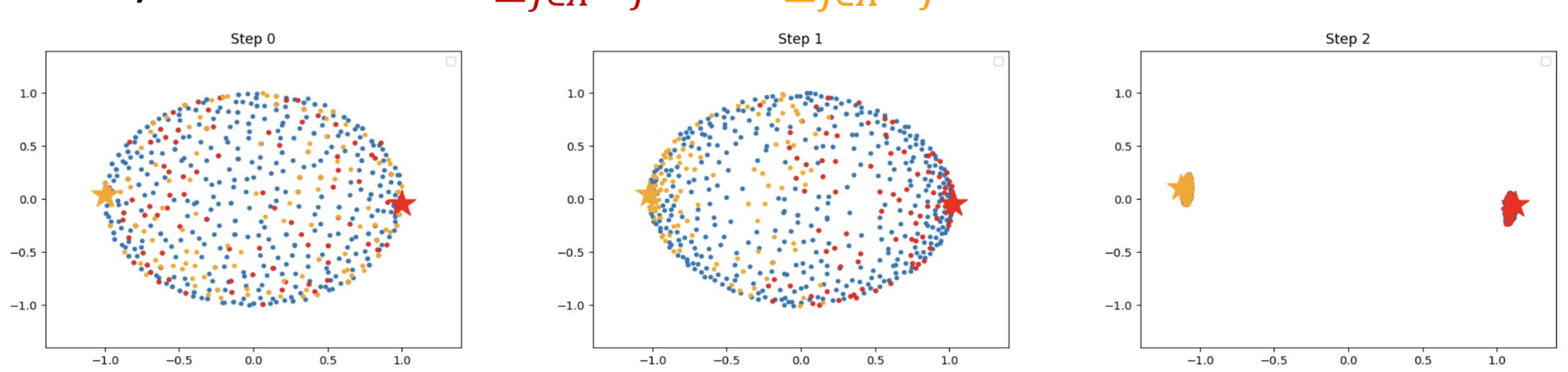
Verification of the Analysis



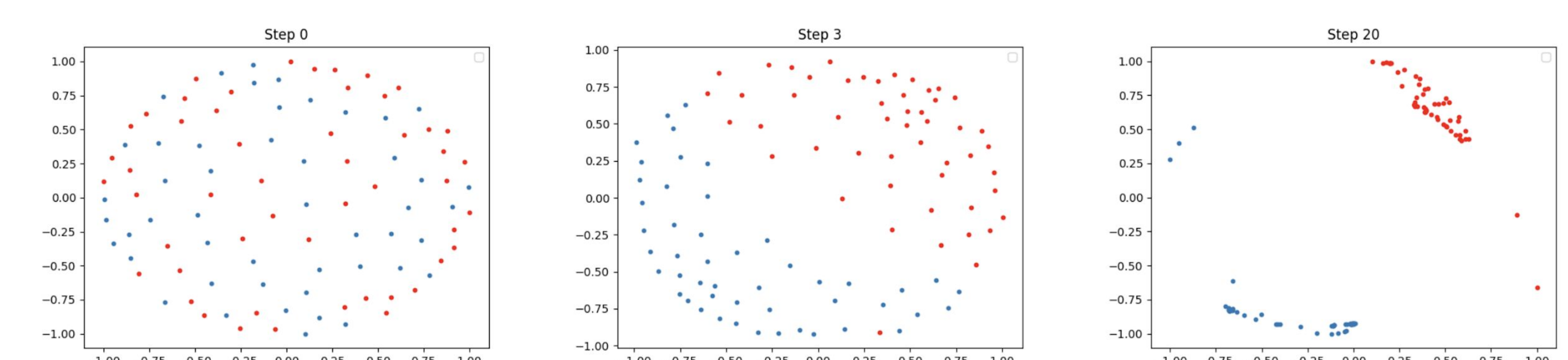
Network Learning Result	Network—
Lower Bound for Fixed Feature Approach	NTK— & Random Feature—
Lower Bound for Without Input Structure	— vs ---

Feature Learning on Synthetic Data

- Visualization of the neuron weights (normalized to unit length)
- They clustered around $\sum_{j \in A} M_j$ and $-\sum_{j \in A} M_j$



Feature Learning on MNIST(0/1)



- The neurons gradually form two clusters around ground-truth weights
- Show the emergence of the features in the neural networks
- However, in fixed feature approaches, there is no feature learning

Take Home Message

Input Structure ➔ Feature Learning

➔ Superior Performance