On Feature Learning in Neural Networks: Emergence from Inputs and Advantage over Fixed Features

Zhenmei Shi, Jenny Wei, Yingyu Liang

UW-Madison

Introduction

Deep Learning

- Remarkable success in applications.
- Advantage over traditional machine learning methods.

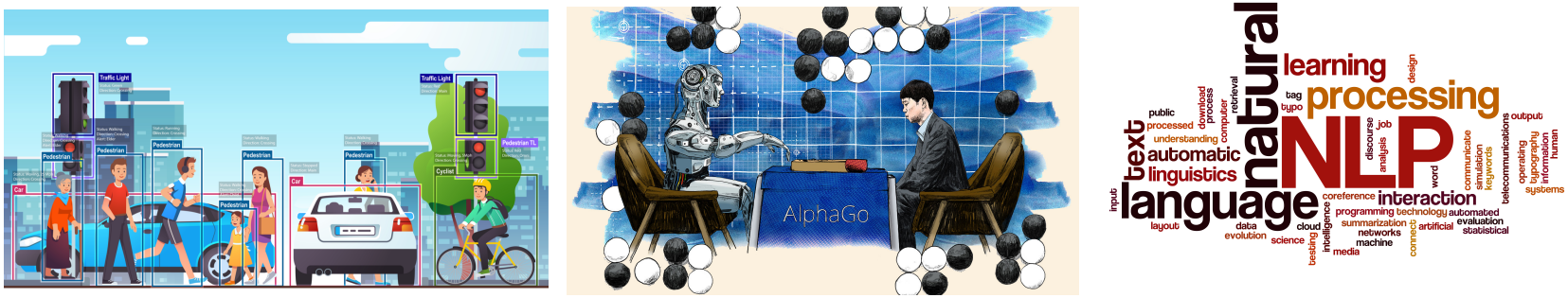
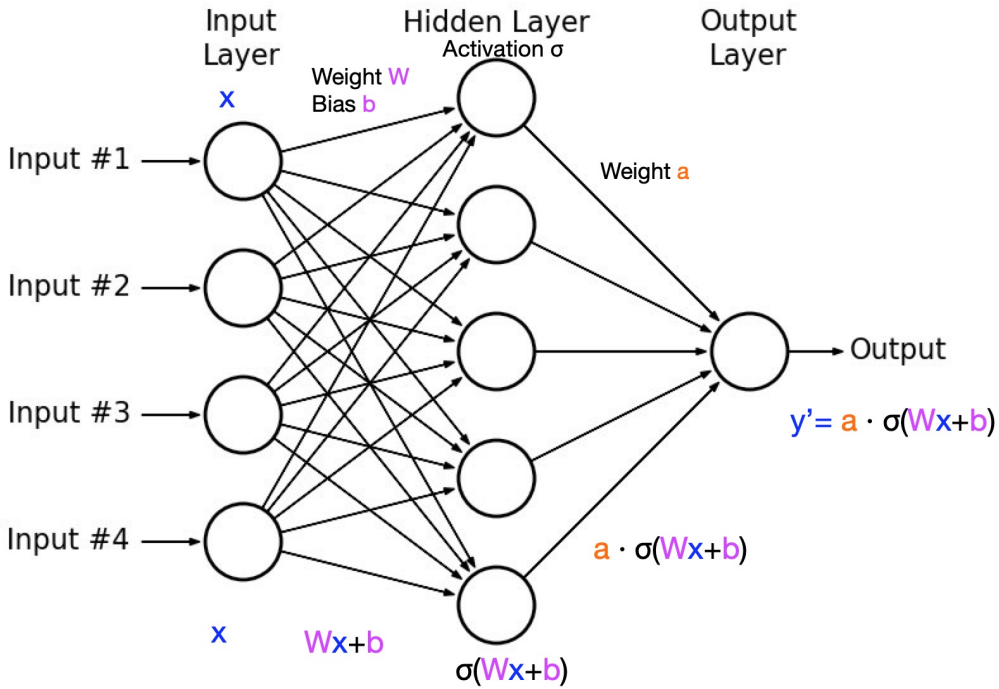


Figure 1: Computer Vision, Reinforcement Learning, Natural Language Processing

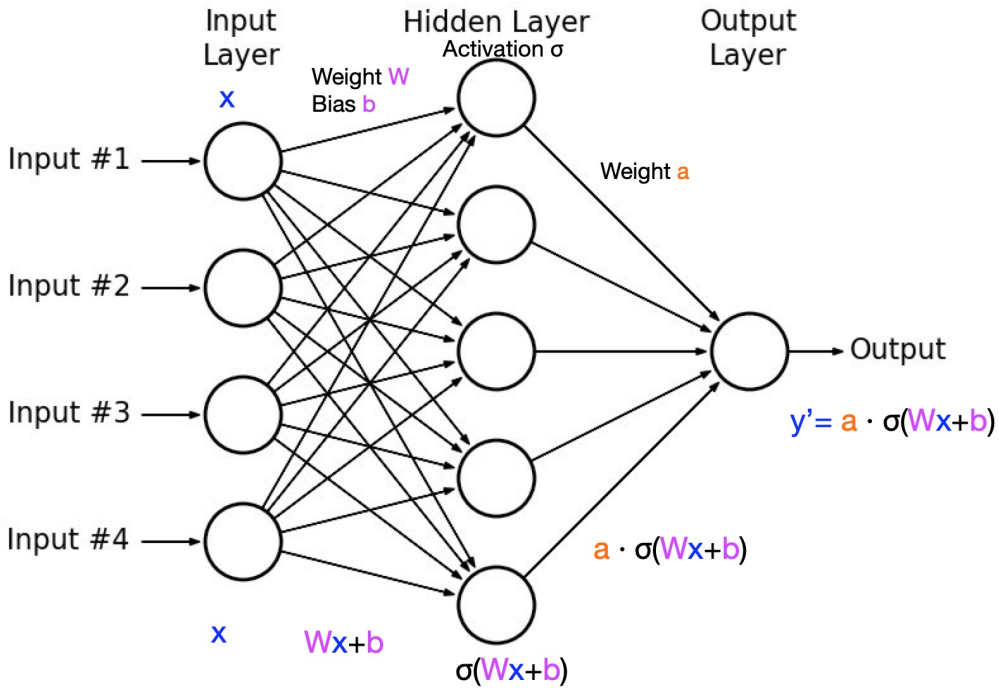
Neural Networks

Two-layer network: $y' = g(x) = a^T \sigma(Wx + b)$.



Neural Networks

Two-layer network: $y' = g(x) = a^T \sigma(Wx + b)$.



Train: gradient descent

$$\theta^{(t)} = \theta^{(t-1)} - \eta^{(t)} \nabla_\theta \left(L(g^{(t-1)}) \right)$$

θ denotes W, b and a.

Neural Networks

- Loss function $\ell(y, y')$: measure the cost incurred by taking a decision y' and y is true label.
- Evaluate the risk function: $L(g) = \mathbb{E}_{(x,y)}[\ell(y, g(x))]$.
- Regularization terms can be added.

Existing Works

Why neural networks success (overparameterized regime)?

Existing Works

Why neural networks success (overparameterized regime)?

Current theoretical understanding:

- Can be approximated by Neural Tangent Kernel (NTK regime or lazy learning regime).
- However, practical training not fit in the NTK regime. Also, NTK cannot explain the network advantage over traditional fixed feature methods (random features, kernel methods).

Existing Works

Why neural networks success (overparameterized regime)?

Current theoretical understanding:

- Can be approximated by Neural Tangent Kernel (NTK regime or lazy learning regime).
- However, practical training not fit in the NTK regime. Also, NTK cannot explain the network advantage over traditional fixed feature methods (random features, kernel methods).
- A recent line of work shows neural networks provably enjoy advantages over fixed feature methods including NTK.
- However, they have not investigated whether input structure is crucial for feature learning, or have not analyzed how gradient descent learns effective features, or rely on strong assumptions (e.g., special networks, Gaussian data, etc).

Empirical Observation

- Neural networks perform better on data with structures.
- Neural networks performs feature learning (feature learning regime).

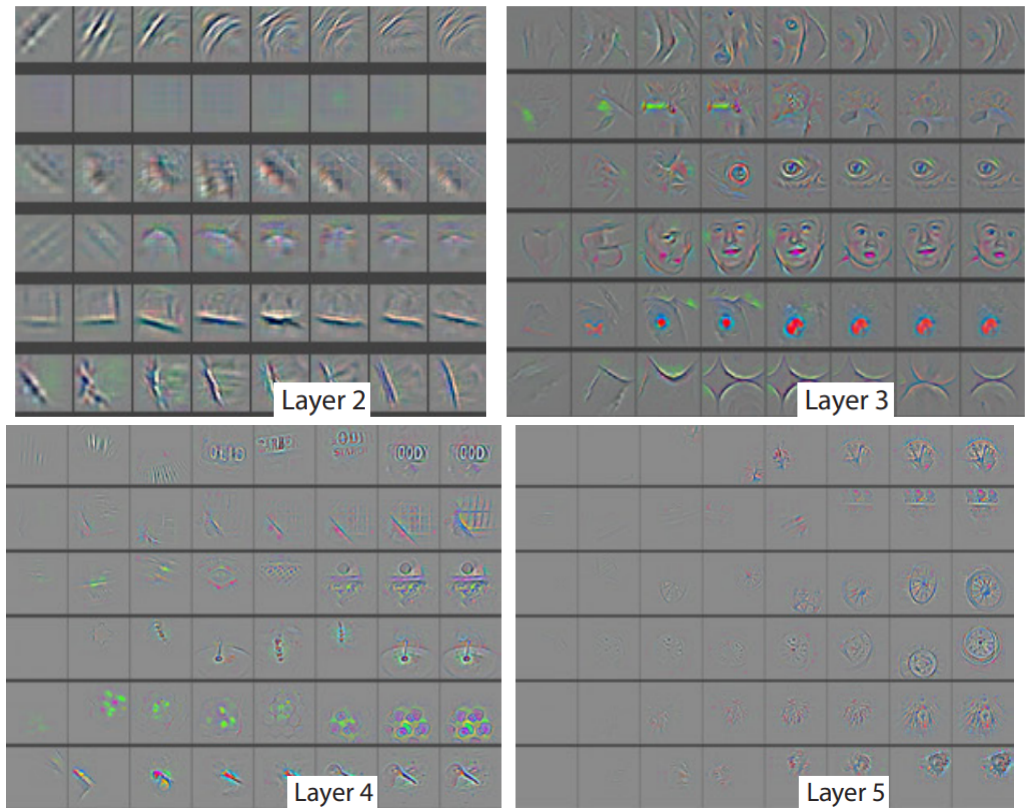


Figure 2: Networks can learn neurons that correspond to different semantic patterns in the inputs. [1]

[1] From paper "Visualizing and Understanding Convolutional Networks".

Question

Question 1:
How can effective features emerge from inputs in the training dynamics of gradient descent?

Question 2:
Is feature learning from inputs necessary for superior performance?

Roadmap

To answer the previous two questions:

*Step*1: Choose input data distributions with and without structures.

*Step*2: Show feature learning exist for input with structures. Analyze convergence of gradient descent with the aid of learned features.

*Step*3: Show fixed feature methods under the same condition (data with structures) cannot learn efficiently.

*Step*4: Show learning input data without structures is much harder for all methods.

Problem Setup

Pattern Counting Problem

Motivation:

- Dictionary learning and sparse coding.
- Images contain label relevant or label irrelevant patterns.
- If the image contains a sufficient number of label relevant patterns, the image may belong to a certain class.

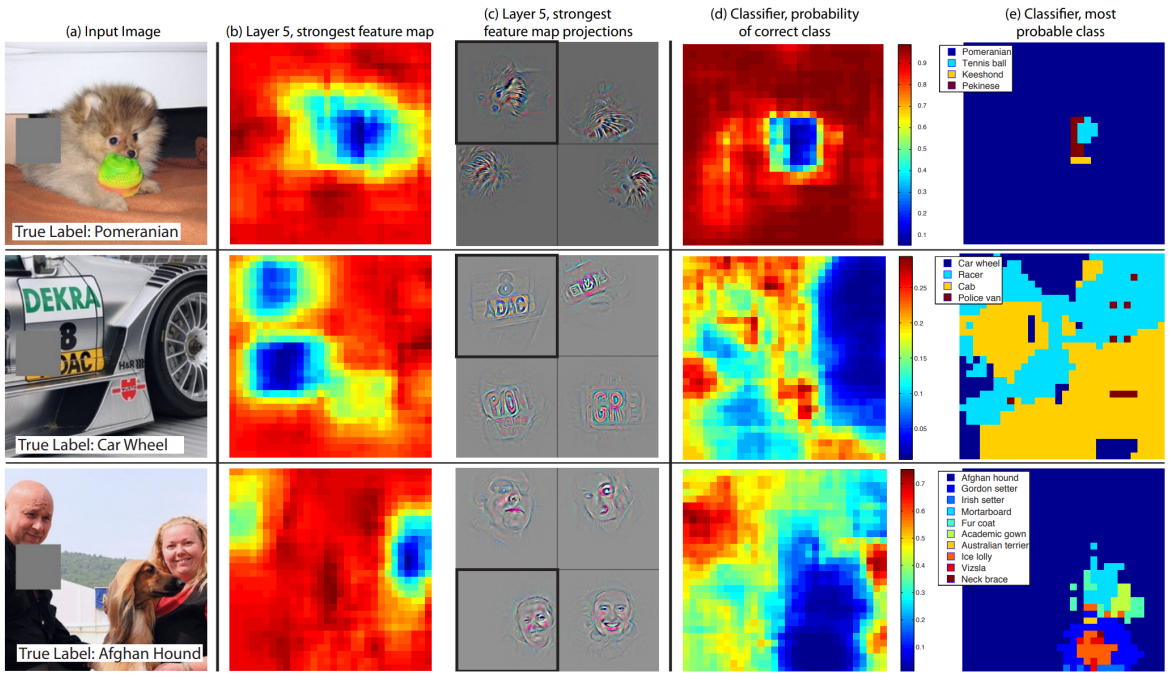


Figure 3: Systematically cover up different portions of the scene with a gray square and see how the top feature maps and classifier output changes. (b): for each position of the gray scale, record the activation in feature map (c): a visualization of this feature map projected down into the input image. (d): a map of correct class probability. (e): the most probable label as a function of occluder position.

Pattern Counting Problem

Hidden representation (pattern indicator vector):

- $\tilde{\phi} \in \{0,1\}^D$: Hidden vector indicates presence of each pattern.
- $\mathscr{D}_{\tilde{\phi}}$: A distribution of $\tilde{\phi}$.
- M: Unknown dictionary of patterns.

Pattern Counting Problem

Hidden representation (pattern indicator vector):

- $\tilde{\phi} \in \{0,1\}^D$: Hidden vector indicates presence of each pattern.
- $\mathscr{D}_{\tilde{\phi}}$: A distribution of $\tilde{\phi}$.
- M: Unknown dictionary of patterns.

Input:

- Given $\tilde{\phi}$, $\mathscr{D}_{\tilde{\phi}}$ and M, generate input $\tilde{x}$ by: $\tilde{x} = M\tilde{\phi}$.

Pattern Counting Problem

Hidden representation (pattern indicator vector):

- $\tilde{\phi} \in \{0,1\}^D$: Hidden vector indicates presence of each pattern.
- $\mathscr{D}_{\tilde{\phi}}$: A distribution of $\tilde{\phi}$.
- M: Unknown dictionary of patterns.

Input:

- Given $\tilde{\phi}$, $\mathscr{D}_{\tilde{\phi}}$ and M, generate input $\tilde{x}$ by: $\tilde{x} = M\tilde{\phi}$.

Label:

- $A \subseteq [D]$: subset of size k, corresponding to class relevant patterns.
- $P \subseteq [k]$, generate label y by:

$$y = \begin{cases} +1, & \text{if } \sum_{i \in A} \tilde{\phi}_i \in P, \\ -1, & \text{otherwise.} \end{cases}$$

- Intuition: y can be any binary function over the number of class relevant patterns.

Assumptions on Distribution over Hidden Representation

Input with structures (family of distributions $\mathscr{F}_{\Xi}$):

(A0) Equal class probability.

(A1) The patterns in A are correlated with the labels: for any $i \in A$, $\gamma = \mathbb{E}[y\tilde{\phi}_i] - \mathbb{E}[y]\mathbb{E}[\tilde{\phi}_i] > 0$.

(A2) Each pattern outside A is independent of all other patterns and identically distributed. Let $p_{\mathrm{o}} := \Pr[\tilde{\phi}_i = 1] \leq 1/2$ denote the probability they appear.

Assumptions on Distribution over Hidden Representation

Input with structures (family of distributions $\mathscr{F}_{\Xi}$):

(A0) Equal class probability.

(A1) The patterns in A are correlated with the labels: for any $i \in A$, $\gamma = \mathbb{E}[y\tilde{\phi}_i] - \mathbb{E}[y]\mathbb{E}[\tilde{\phi}_i] > 0$.

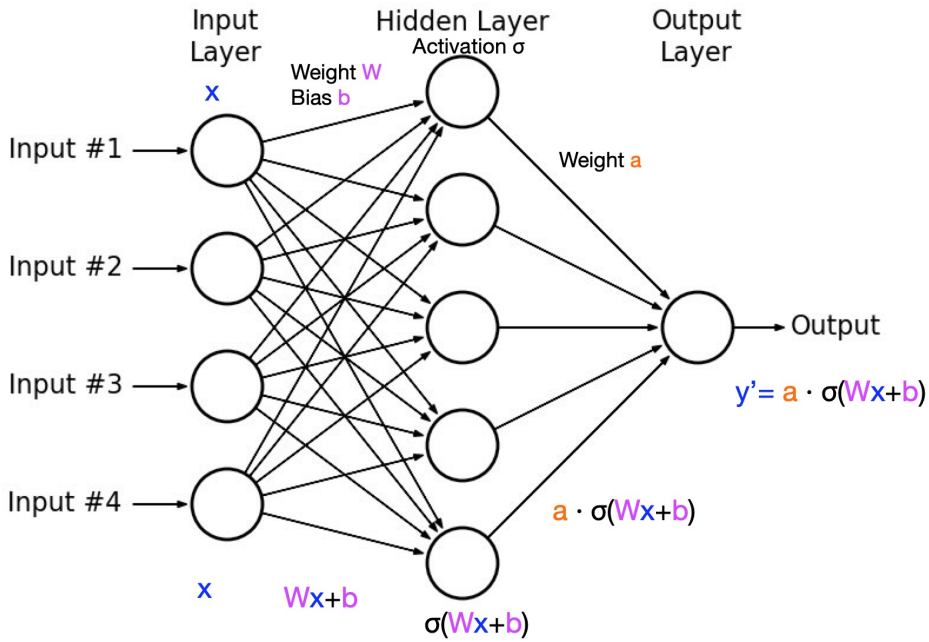
(A2) Each pattern outside A is independent of all other patterns and identically distributed. Let $p_{\mathrm{o}} := \Pr[\tilde{\phi}_i = 1] \leq 1/2$ denote the probability they appear.

Input without structures (family of distributions $\mathscr{F}_{\Xi_0}$):

(A1') The patterns are independent, and $\tilde{\phi}_i$ is uniform.

Network

- Two-layer network: $g(x) = \sum_{i=1}^{2m} a_i \sigma(\langle w_i, x \rangle + b_i)$.
- $\sigma(z) = \min(1, \max(z, 0))$: the truncated rectified linear unit (ReLU) activation function.



- Hinge loss and ℓ_2 regularization.
- Gaussian initialization and gradient descent.

Main Results

Provable Guarantee for Neural Networks

Theorem 1 (Informal)

For any small positive δ *and* ε, *if* $k = \Omega\left(\log^2\left(Dm/(\delta\gamma)\right)\right)$, $p_o = \Omega(k^2/D)$ *and* $m \geq \max\{\Omega(k^{12}/\varepsilon^{3/2}), D\}$, *then with proper hyper-parameters, for any* $\mathscr{D} \in \mathscr{F}_{\Xi}$, *with probability at least* $1-\delta$, *there exists* $t \in [T]$ *such that* $\Pr[\mathrm{sign}(g^{(t)}(x)) \neq y] \leq L_{\mathscr{D}}(g^{(t)}) \leq \varepsilon$.

Provable Guarantee for Neural Networks

Theorem 1 (Informal)

For any small positive δ *and* ε, *if* $k = \Omega\left(\log^2\left(Dm/(\delta\gamma)\right)\right)$, $p_o = \Omega(k^2/D)$ *and* $m \geq \max\{\Omega(k^{12}/\varepsilon^{3/2}), D\}$, *then with proper hyper-parameters, for any* $\mathscr{D} \in \mathscr{F}_{\Xi}$, *with probability at least* $1-\delta$, *there exists* $t \in [T]$ *such that* $\Pr[\mathrm{sign}(g^{(t)}(x)) \neq y] \leq L_{\mathscr{D}}(g^{(t)}) \leq \varepsilon$.

Message:

- For a wide range of the background pattern probability p_0 and the number of class relevant patterns k, given any input distribution with structure, neural network can achieve small population risk with polynomial number of neurons.
- The analysis shows the success comes from feature learning.

Lower Bound for Fixed Features Models

Fixed features model:

Suppose Ψ is a data-independent feature mapping of dimension N with bounded features, i.e., $\Psi : \mathscr{X} \to [-1,1]^N$. For $B > 0$, the family of linear models on Ψ with bounded norm B is $\mathscr{H}_B = \{h(\tilde{x}) : h(\tilde{x}) = \langle \Psi(\tilde{x}), w \rangle, \|w\|_2 \leq B\}$.

Lower Bound for Fixed Features Models

Fixed features model:

Suppose Ψ is a data-independent feature mapping of dimension N with bounded features, i.e., $\Psi : \mathscr{X} \to [-1,1]^N$. For $B > 0$, the family of linear models on Ψ with bounded norm B is $\mathscr{H}_B = \{h(\tilde{x}) : h(\tilde{x}) = \langle \Psi(\tilde{x}), w \rangle, \|w\|_2 \leq B\}$.

Theorem 2 (Informal)

With proper k, there exists $\mathscr{D} \in \mathscr{F}_{=}$ such that all $h \in \mathscr{H}_B$ have hinge-loss at least $p_o \left(1 - \frac{\sqrt{2N}B}{2^k}\right)$.

Lower Bound for Fixed Features Models

Fixed features model:

Suppose Ψ is a data-independent feature mapping of dimension N with bounded features, i.e., $\Psi : \mathscr{X} \to [-1,1]^N$. For $B > 0$, the family of linear models on Ψ with bounded norm B is $\mathscr{H}_B = \{h(\tilde{x}) : h(\tilde{x}) = \langle \Psi(\tilde{x}), w \rangle, \|w\|_2 \leq B\}$.

Theorem 2 (Informal)

With proper k, there exists $\mathscr{D} \in \mathscr{F}_{\equiv}$ such that all $h \in \mathscr{H}_B$ have hinge-loss at least $p_o \left(1 - \frac{\sqrt{2N}B}{2^k}\right)$.

Message:

There exists a input distribution with structure such that no fixed feature method with polynomial features can efficiently learn the task.

Lower Bound for Learning Without Input Structure

Statistical Query (SQ) model:

- Only receive information through statistical queries (Q, τ). Property predicate Q of labeled instances and tolerance $\tau \in [0, 1]$. Receive a response $\hat{P}_Q \in [P_Q - \tau, P_Q + \tau]$, where $P_Q = \Pr[Q(x, y) \text{ is true}]$.
- The SQ model captures almost all common learning algorithms including mini-batch SGD.

Lower Bound for Learning Without Input Structure

Statistical Query (SQ) model:

- Only receive information through statistical queries (Q, τ). Property predicate Q of labeled instances and tolerance $\tau \in [0, 1]$. Receive a response $\hat{P}_Q \in [P_Q - \tau, P_Q + \tau]$, where $P_Q = \Pr[Q(x, y) \text{ is true}]$.
- The SQ model captures almost all common learning algorithms including mini-batch SGD.

Theorem 3

For any algorithm in the Statistical Query model with query (Q, τ) that can learn over $\mathscr{F}_{\Xi_0}$ to classification error less than $\frac{1}{2} - \frac{1}{\binom{D}{k}^3}$, either the number of queries or $1/\tau$ must be at least $\frac{1}{2}\binom{D}{k}^{1/3}$.

Lower Bound for Learning Without Input Structure

Statistical Query (SQ) model:

- Only receive information through statistical queries (Q, τ). Property predicate Q of labeled instances and tolerance $\tau \in [0, 1]$. Receive a response $\hat{P}_Q \in [P_Q - \tau, P_Q + \tau]$, where $P_Q = \Pr[Q(x, y) \text{ is true}]$.
- The SQ model captures almost all common learning algorithms including mini-batch SGD.

Theorem 3

For any algorithm in the Statistical Query model with query (Q, τ) that can learn over $\mathscr{F}_{\Xi_0}$ to classification error less than $\frac{1}{2} - \frac{1}{\binom{D}{k}^3}$, either the number of queries or $1/\tau$ must be at least $\frac{1}{2}\binom{D}{k}^{1/3}$.

Message:

Without input structure, polynomial SQ model cannot non-trivially better than random guessing.

Proof Sketches

Existence of A Good Network

Intuition:

Find a "good" two-layer network that can represent the target labeling function, whose neurons are viewed as ground truth features. Then focus on analyzing how the network learns such neuron weights.

Existence of A Good Network

Intuition:

Find a "good" two-layer network that can represent the target labeling function, whose neurons are viewed as ground truth features. Then focus on analyzing how the network learns such neuron weights.

Lemma 4 (Informal)

For any $\mathscr{D} \in \mathscr{F}_{\Xi}$, here exists a two-layer network $g^*(x)$ with zero loss. Furthermore, the hidden neurons' weights in $g^*(x)$ are all proportional to $\sum_{j \in A} M_j$.

Feature Emergence in the First Gradient Step

Intuition:

After the first gradient step, the hidden neurons of the trained network become close to the ground truth features.

Feature Emergence in the First Gradient Step

Intuition:

After the first gradient step, the hidden neurons of the trained network become close to the ground truth features.

Lemma 5 (Informal)

$\frac{\partial}{\partial w_i} L_{\mathscr{D}}(g^{(0)}) = -a_i^{(0)} \sum_{j=1}^{D} M_j T_j$ *where* T_j *satisfies:*

- *if* $j \in A$, *then* $T_j \approx O(\gamma)$;
- *if* $j \notin A$, *then* $|T_j| \leq O(\varepsilon_{e1})$ *for a small* ε_{e1}.

Feature Emergence in the First Gradient Step

Intuition:

After the first gradient step, the hidden neurons of the trained network become close to the ground truth features.

Lemma 5 (Informal)

$\frac{\partial}{\partial w_i} L_{\mathscr{D}}(g^{(0)}) = -a_i^{(0)} \sum_{j=1}^{D} M_j T_j$ *where* T_j *satisfies:*

- *if* $j \in A$, *then* $T_j \approx O(\gamma)$;
- *if* $j \notin A$, *then* $|T_j| \leq O(\varepsilon_{e1})$ *for a small* ε_{e1}.

Message:

Gradients are updated uniformly in class-relevant pattern directions. Their updates in class-irrelevant pattern directions are relatively small.

Proof Ideas of Lemma 5

The gradient of w_i is: $\frac{\partial L_{\mathscr{D}}(g)}{\partial w_i} = -a_i \mathbb{E}_{(x,y)\sim\mathscr{D}} \{ y x \sigma'[\langle w_i, x\rangle + b_i]\}$.

Let $\phi = (\tilde{\phi} - \mathbb{E}[\tilde{\phi}])/\tilde{\sigma}$, then the component of the gradient on M_j is:

$$\langle M_j, \frac{\partial L_{\mathscr{D}}(g)}{\partial w_i} \rangle = -a_i \mathbb{E} \left\{ y \phi_j \sigma' \left[\sum_{\ell \in [D]} \phi_\ell q_\ell + b_i \right] \right\}. \tag{1}$$

Proof Ideas of Lemma 5

The gradient of w_i is: $\frac{\partial L_{\mathscr{D}}(g)}{\partial w_i} = -a_i \mathbb{E}_{(x,y)\sim\mathscr{D}}\{yx\sigma'[\langle w_i, x\rangle + b_i]\}$.

Let $\phi = (\tilde{\phi} - \mathbb{E}[\tilde{\phi}])/\tilde{\sigma}$, then the component of the gradient on M_j is:

$$\langle M_j, \frac{\partial L_{\mathscr{D}}(g)}{\partial w_i}\rangle = -a_i \mathbb{E}\left\{ y\phi_j \sigma'\left[\sum_{\ell\in[D]} \phi_\ell q_\ell + b_i\right]\right\}. \tag{1}$$

If the set of class relevant patterns A is relatively small, then

$$\mathbb{I}_{[D]} := \sigma'\left[\sum_{\ell\in[D]} \phi_\ell q_\ell + b_i\right] \approx \mathbb{I}_{-A} := \sigma'\left[\sum_{\ell\notin A} \phi_\ell q_\ell + b_i\right]. \tag{2}$$

Proof Ideas of Lemma 5

The gradient of w_i is: $\frac{\partial L_{\mathscr{D}}(g)}{\partial w_i} = -a_i \mathbb{E}_{(x,y)\sim\mathscr{D}}\{yx\sigma'[\langle w_i, x\rangle + b_i]\}$.

Let $\phi = (\tilde{\phi} - \mathbb{E}[\tilde{\phi}])/\tilde{\sigma}$, then the component of the gradient on M_j is:

$$\langle M_j, \frac{\partial L_{\mathscr{D}}(g)}{\partial w_i}\rangle = -a_i \mathbb{E}\left\{y\phi_j \sigma'\left[\sum_{\ell\in[D]} \phi_\ell q_\ell + b_i\right]\right\}. \quad (1)$$

If the set of class relevant patterns A is relatively small, then

$$\mathbb{I}_{[D]} := \sigma'\left[\sum_{\ell\in[D]} \phi_\ell q_\ell + b_i\right] \approx \mathbb{I}_{-A} := \sigma'\left[\sum_{\ell\notin A} \phi_\ell q_\ell + b_i\right]. \quad (2)$$

Thus, component of each class relevant pattern is nearly a constant:

$$\langle M_j, \frac{\partial L_{\mathscr{D}}(g)}{\partial w_i}\rangle \propto \mathbb{E}\{y\phi_j \mathbb{I}_{[D]}\} \approx \mathbb{E}\{y\phi_j \mathbb{I}_{-A}\} = \mathbb{E}\{y\phi_j\}\mathbb{E}[\mathbb{I}_{-A}]. \quad (3)$$

Similarly, for background patterns, the component is close to 0.

Feature Improvement in the Second Gradient Step

Intuition:

After the second gradient step, these neurons get improved to a sufficiently good level.

Feature Improvement in the Second Gradient Step

Intuition:

After the second gradient step, these neurons get improved to a sufficiently good level.

Lemma 6 (Informal)

$\frac{\partial}{\partial w_i} L_{\mathscr{D}}(g^{(1)}) = -a_i^{(1)} \sum_{j=1}^{D} M_j T_j$ *where* T_j *satisfies:*

- *if* $j \in A$, *then* $T_j \approx O(\gamma)$;
- *if* $j \notin A$, *then* $|T_j| \leq O(\varepsilon_{e2})$ *for a small* ε_{e2}, *where* ε_{e2} *much smaller than* ε_{e1}.

Feature Improvement in the Second Gradient Step

Intuition:

After the second gradient step, these neurons get improved to a sufficiently good level.

Lemma 6 (Informal)

$\frac{\partial}{\partial w_i} L_{\mathscr{D}}(g^{(1)}) = -a_i^{(1)} \sum_{j=1}^{D} M_j T_j$ *where* T_j *satisfies:*

- *if* $j \in A$, *then* $T_j \approx O(\gamma)$;
- *if* $j \notin A$, *then* $|T_j| \leq O(\varepsilon_{e2})$ *for a small* ε_{e2}, *where* ε_{e2} *much smaller than* ε_{e1}.

Message:

The signal-to-noise ratio improves in this step.

Experiments

Simulation: Test Accuracy VS Steps

Setting:

Generate simulated data with or without input structure and labels given by the parity function.

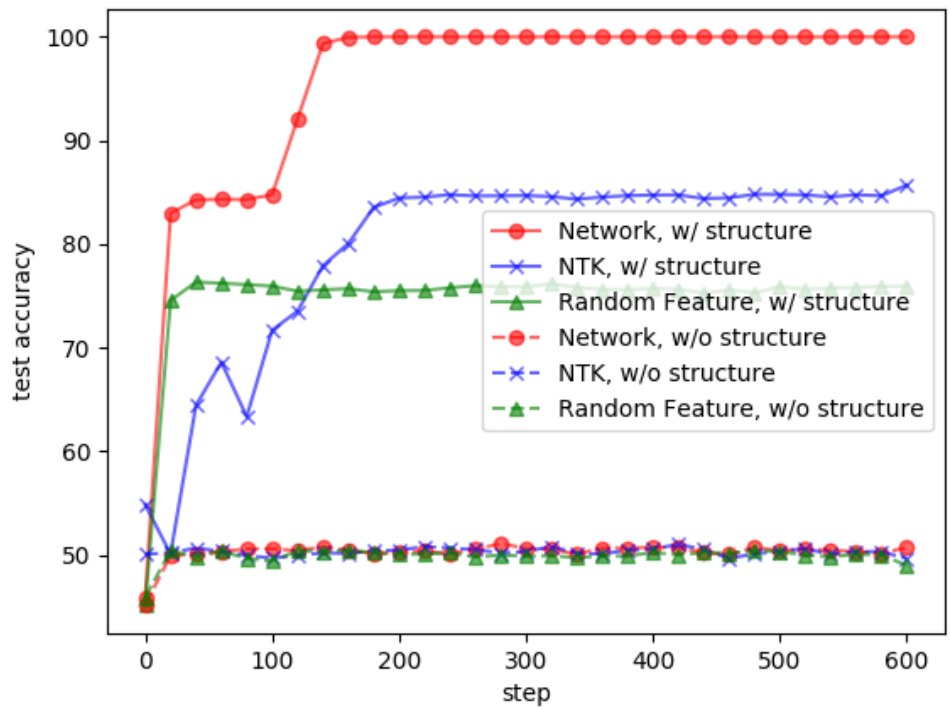
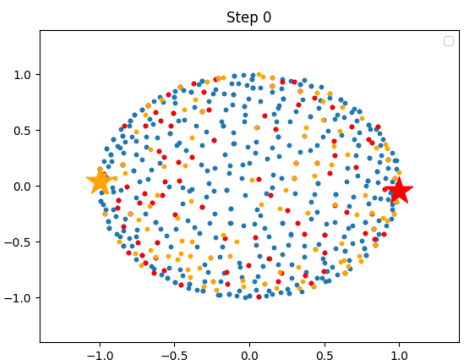


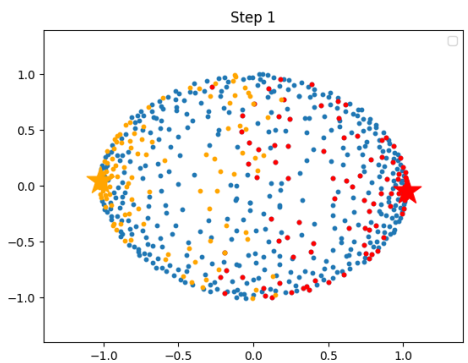
Figure 4: Test accuracy on simulated data with or without input structure.

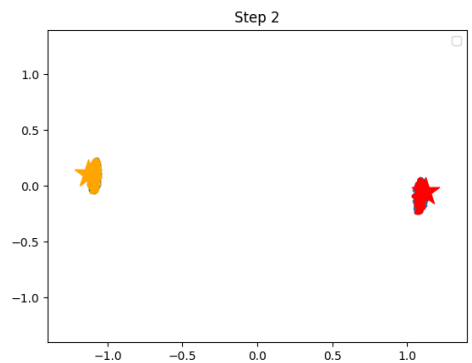
Synthetic Data: Feature Learning in Networks

Setting:

Compute the cosine similarities between the weights w_i's and visualize them by Multidimensional Scaling. Dots represent neurons and stars represent effective features $\pm\sum_{j\in A} M_j$.



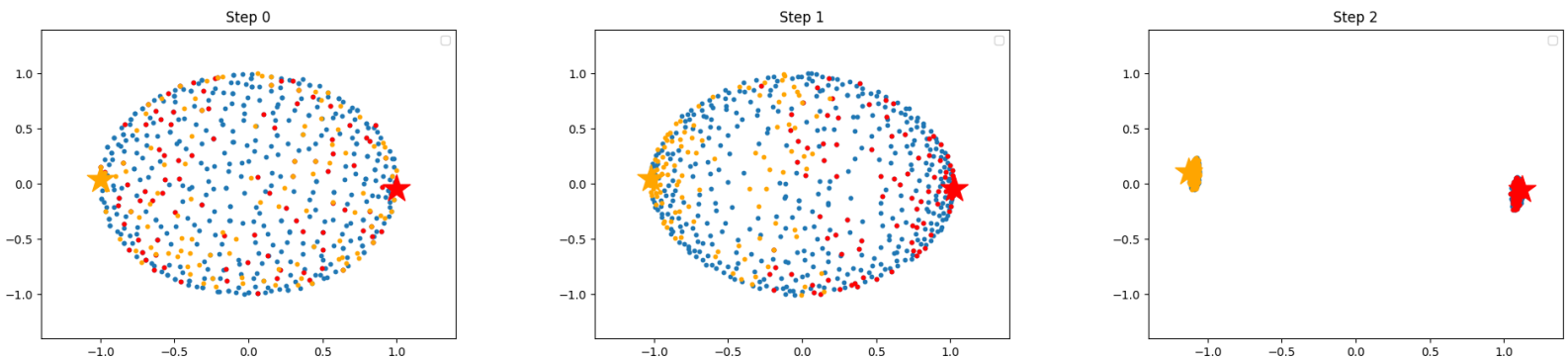




Synthetic Data: Feature Learning in Networks

Setting:

Compute the cosine similarities between the weights w_i's and visualize them by Multidimensional Scaling. Dots represent neurons and stars represent effective features $\pm\sum_{j\in A} M_j$.



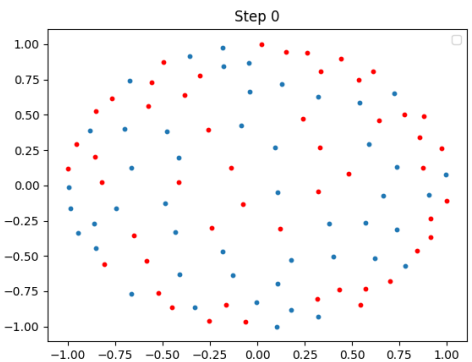
Message:

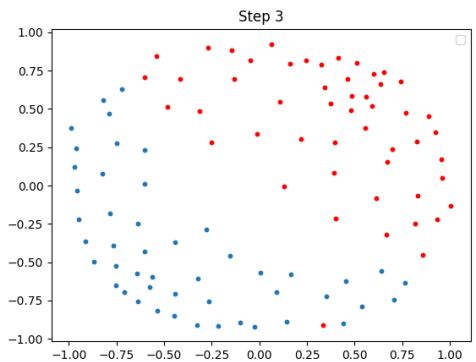
All neurons converge to the effective features after two steps.

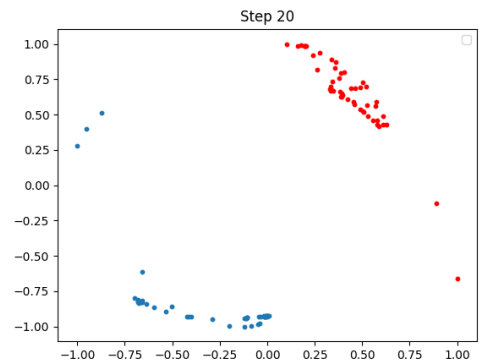
Real Data: Feature Learning in Networks

Setting:

Two-layer network trained on the subset of MNIST data with label 0/1.



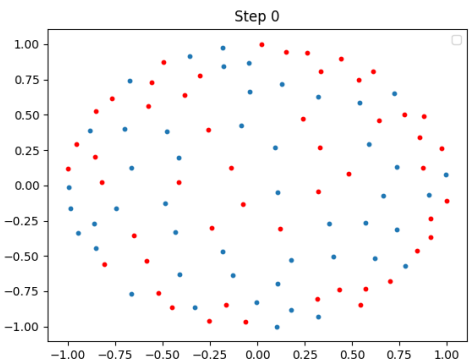


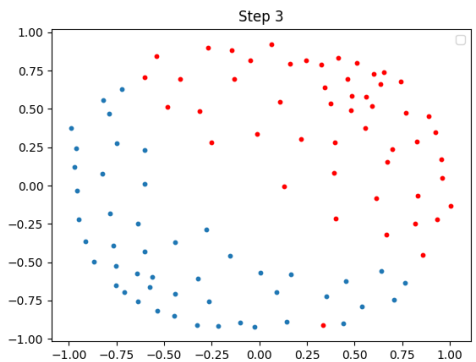


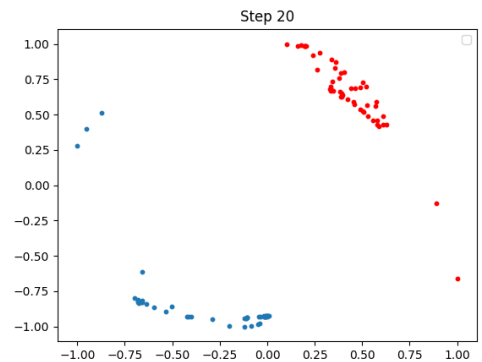
Real Data: Feature Learning in Networks

Setting:

Two-layer network trained on the subset of MNIST data with label 0/1.







Message:

Similar clustering effect.

Take Home Message

Question 1:
How can effective features emerge from inputs in the training dynamics of gradient descent?
Answer
Input structures provably influence effective feature learning.

Question 2:
Is feature learning from inputs necessary for superior performance?
Answer
Feature learning ability of neural networks provably leads to their success comparing to fixed feature methods.

Thank you!

Q&A