Emergence and scaling laws for SGD learning of shallow neural networks.

Yunwei Ren*[1], Eshaan Nichani*[1], Denny Wu[2,3], Jason D. Lee[1]

[1]Princeton University [2]New York University [3]Flatiron Institute

April 22, 2025

Task

Target function (two-layer orthogonal networks).

$$f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x}), \quad \boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I}_d),$$

where $a_p > 0$, $\{\boldsymbol{w}_p^*\}_p \subset \mathbb{S}^{d-1}$ are the unknown ground truth weights and σ is the activation/link function.

- (orthogonal weights) $\{\boldsymbol{w}_k^*\}_k$ is orthonormal.
- (large width) $1 \ll P \ll d^c$.
- (large condition number) $\kappa := \max_p a_p / \min_p a_p \gg 1$.

Task

Target function (two-layer orthogonal networks).

$$f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x}), \quad \boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I}_d),$$

where $a_p > 0$, $\{\boldsymbol{w}_p^*\}_p \subset \mathbb{S}^{d-1}$ are the unknown ground truth weights and σ is the activation/link function.

- (orthogonal weights) $\{\boldsymbol{w}_k^*\}_k$ is orthonormal.
- (large width) $1 \ll P \ll d^c$.
- (large condition number) $\kappa := \max_p a_p / \min_p a_p \gg 1$.

Q. Can we learn this function class using a two-layer network and vanilla online SGD?

- poly(d, P, κ) sample/iteration complexity;
- $\tilde{O}(P)$ learner neurons;
- No strange modifications to the algorithm.

Motivations from the empirical side

- **Neural scaling laws** [Kaplan et al. 20], [Hoffmann et al. 22]
  Observed in practice that increasing compute and data leads to smooth power-law decay in the loss.
- **Emergence** [Wei et al. 22], [Ganguli et al. 22]
  Learning of individual tasks/skills exhibits sharp transitions.
- **Q.** How to reconcile these two observations?

Motivations from the empirical side

Q. How to reconcile the emergent behavior in skill acquisition and the smooth power-law decay in the loss?

Hypothesis (Additive Model [Michaud et al. 24], [Nam et al. 24])

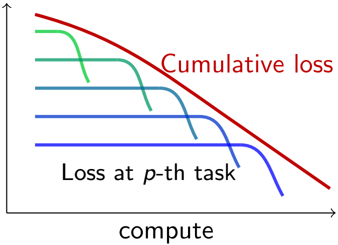
- Cumulative objective can be decomposed into a large number of distinct "skills", learning of each exhibits sharp transitions.
- Combination of numerous emergent learning curves at different time scales results in a power-law rate.

Motivations from the empirical side

Q. How to reconcile the emergent behavior in skill acquisition and the smooth power-law decay in the loss?

Hypothesis (Additive Model [Michaud et al. 24], [Nam et al. 24])

- Cumulative objective can be decomposed into a large number of distinct "skills", learning of each exhibits sharp transitions.
- Combination of numerous emergent learning curves at different time scales results in a power-law rate.

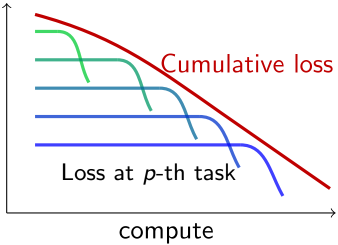


Motivations from the empirical side

Q. How to reconcile the emergent behavior in skill acquisition and the smooth power-law decay in the loss?

Hypothesis (Additive Model [Michaud et al. 24], [Nam et al. 24])

- Cumulative objective can be decomposed into a large number of distinct "skills", learning of each exhibits sharp transitions.
- Combination of numerous emergent learning curves at different time scales results in a power-law rate.

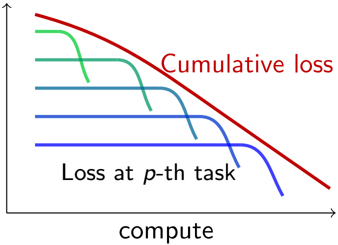


Motivations from the empirical side

Q. How to reconcile the emergent behavior in skill acquisition and the smooth power-law decay in the loss?

Hypothesis (Additive Model [Michaud et al. 24], [Nam et al. 24]**)**

- Cumulative objective can be decomposed into a large number of distinct "skills", learning of each exhibits sharp transitions.
- Combination of numerous emergent learning curves at different time scales results in a power-law rate.



This work: theoretical justification of the additive model hypothesis in SGD learning of the target function:

$$f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x}),$$

with $a_p \propto p^{-\beta}$.

Motivations from the theory side

Theorem ([Li, Ma, Zhang, 2020])

- *(Orthogonal, well-conditioned teacher)* $f_*(\boldsymbol{x}) = \sum_{p=1}^{d} a_p^* \, |\langle \boldsymbol{e}_p, \boldsymbol{x}\rangle|$ *with condition number* $\max_p a_p^* / \min_p a_p^* = \kappa$.
- *(extremely overparameterized, 2-homogeneous student)* $f(\boldsymbol{x}) = \sum_{k=1}^{m} \|\boldsymbol{w}_k\| \, \mathrm{ReLU}(\boldsymbol{w}_k \cdot \boldsymbol{x})$ *with width* $m = e^{\kappa} \operatorname{poly} d$.

$\Rightarrow$ *Online SGD can efficiently minimize* $\mathcal{L}$ *and recovery* $\{(a_p^*, \boldsymbol{e}_p)\}_p$.

Motivations from the theory side

Theorem ([Li, Ma, Zhang, 2020])

- *(Orthogonal, well-conditioned teacher)* $f_*(\boldsymbol{x}) = \sum_{p=1}^{d} a_p^* \left|\langle \boldsymbol{e}_p, \boldsymbol{x}\rangle\right|$ *with condition number* $\max_p a_p^* / \min_p a_p^* = \kappa$.
- *(extremely overparameterized, 2-homogeneous student)* $f(\boldsymbol{x}) = \sum_{k=1}^{m} \|\boldsymbol{w}_k\| \, \mathrm{ReLU}(\boldsymbol{w}_k \cdot \boldsymbol{x})$ *with width* $m = e^{\kappa} \operatorname{poly} d$.

$\Rightarrow$ *Online SGD can efficiently minimize* $\mathcal{L}$ *and recovery* $\{(a_p^*, \boldsymbol{e}_p)\}_p$.

(Motivation: separating kernel methods and neural networks.)

Proof strategy.

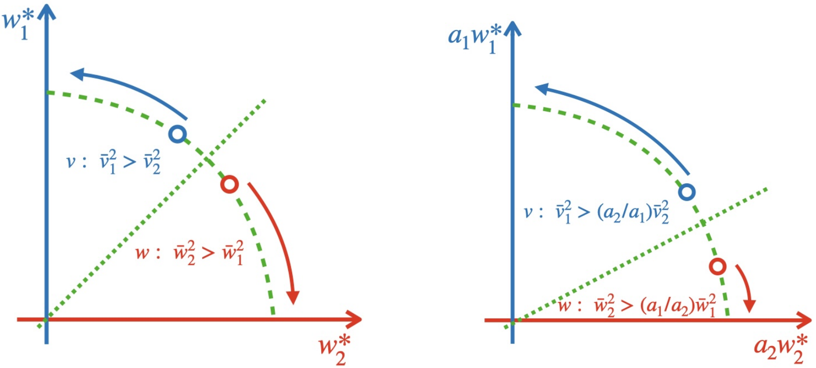
- Convert the task to orthogonal tensor decomposition using Hermite analysis.
- Gradient descent mimics the tensor power method.
- 4th order orthogonal tensor decomposition can be efficiently solved by the tensor power method (with deflation).

The e^κ factor

Why does [LMZ20] need $m = e^\kappa \operatorname{poly} d$ neurons?

The e^{κ} factor

Why does [LMZ20] need $m = e^{\kappa}\,\mathrm{poly}\,d$ neurons?



In tensor power method (4th order, without deflation):

- Need $a_p \bar{v}_p^2 > \max_{q \neq p} a_q \bar{v}_q^2$ for $\boldsymbol{v}$ to converge to $\boldsymbol{w}_p^*$.
- $\Rightarrow$ Need $e^{\kappa}\,\mathrm{poly}\,d$ neurons to cover all directions.

The e^κ factor

Why removing the e^κ factor (without using manual deflation or reinitialization) is meaningful?

The e^κ factor

Why removing the e^κ factor (without using manual deflation or reinitialization) is meaningful?

In practice:

- Can not expect the condition number to be small;
- Despite being overparameterized, the number of neurons in each layer is not extremely large;
- Practitioners only use variants of SGD with no manual deflation.

The e^κ factor

Why removing the e^κ factor (without using manual deflation or reinitialization) is meaningful?

In practice:

- Can not expect the condition number to be small;
- Despite being overparameterized, the number of neurons in each layer is not extremely large;
- Practitioners only use variants of SGD with no manual deflation.

The e^κ factor

Why removing the e^κ factor (without using manual deflation or reinitialization) is meaningful?

In practice:

- Can not expect the condition number to be small;
- Despite being overparameterized, the number of neurons in each layer is not extremely large;
- Practitioners only use variants of SGD with no manual deflation.

The e^κ factor

Why removing the e^κ factor (without using manual deflation or reinitialization) is meaningful?

In practice:

- Can not expect the condition number to be small;
- Despite being overparameterized, the number of neurons in each layer is not extremely large;
- Practitioners only use variants of SGD with no manual deflation.

The e^{κ} factor

Why removing the e^{κ} factor (without using manual deflation or reinitialization) is meaningful?

In practice:

- Can <u>not</u> expect the condition number to be small;
- Despite being overparameterized, the number of neurons in each layer is <u>not</u> extremely large;
- Practitioners only use variants of SGD with <u>no</u> manual deflation.

A conceptual question:
Can we efficiently learn all directions in parallel when the condition number is large?

A brief summary of our discussion so far

- **Task.** Learning orthogonal shallow networks

$$f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x}).$$

- **Algorithm.** Online SGD with no manual deflation/reinitialization.

A brief summary of our discussion so far

- **Task.** Learning orthogonal shallow networks

$$f_*(\boldsymbol{x}) = \textstyle\sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x}).$$

- **Algorithm.** Online SGD with no manual deflation/reinitialization.
- **Motivation (Additive model hypothesis)**
  - Does the learning of each direction $a_p \boldsymbol{w}_p^*$ has a sharp transition?
  - Can they lead to a non-trivial power law decay in the loss?

A brief summary of our discussion so far

- **Task.** Learning orthogonal shallow networks

$$f_*(\boldsymbol{x}) = \textstyle\sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x}).$$

- **Algorithm.** Online SGD with no manual deflation/reinitialization.
- **Motivation (Additive model hypothesis)**
  - Does the learning of each direction $a_p \boldsymbol{w}_p^*$ has a sharp transition?
  - Can they lead to a non-trivial power law decay in the loss?
- **Motivation (Learning when the condition number $\kappa \gg 1$)**
  - Is it necessary to have e^κ neurons?
  - How to avoid the large directions attracting all the neurons?

A brief summary of our discussion so far

- **Task.** Learning orthogonal shallow networks

$$f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x}).$$

- **Algorithm.** Online SGD with no manual deflation/reinitialization.
- **Motivation (Additive model hypothesis)**
  - Does the learning of each direction $a_p \boldsymbol{w}_p^*$ has a sharp transition?
    - Yes, when $\mathrm{IE}(\sigma) > 2$. [Ben Arous, Gheissari, Jagannath, 2021]
  - Can they lead to a non-trivial power law decay in the loss?
    - Yes. (This work)
- **Motivation (Learning when the condition number $\kappa \gg 1$)**
  - Is it necessary to have e^κ neurons?
    - No. $O(P \log P)$ neurons suffice. (This work)
  - How to avoid the large directions attracting all the neurons?
    - Rely on the sharp transitions/emergence. (This work)

Emergence in single-index models

Definition (Single-index models)

A single-index model is a two-layer neural network with one neuron:

$$f_*(\boldsymbol{x}) = \sigma(\boldsymbol{w}^* \cdot \boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathbb{R}^d,$$

where $\boldsymbol{w}^* \in \mathbb{S}^{d-1}$ is the ground truth direction, and $\sigma : \mathbb{R} \to \mathbb{R}$ the link function.

- A long history, dated at least to [Ichimura, 1993].
- Have different names: generalized linear models, learning a single neuron, phase retrieval...

Q. Sample complexity of learning a single-index model when $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I}_d)$?

Information exponent [Ben Arous, Gheissari, Jagannath, 2021]

Hermite expansion. $\sigma(z) = \sum_{i=0}^{\infty} \hat{\sigma}_i h_i$, where h_i is the i-th (normalized) Hermite polynomial and $\hat{\sigma}_i = \mathbb{E}_{z \sim \mathcal{N}(0,1)}[\sigma(z) h_i(z)]$.

- **Fact.** $\mathbb{E}_{\boldsymbol{x}}[h_i(\boldsymbol{v} \cdot \boldsymbol{x}) h_j(\boldsymbol{w} \cdot \boldsymbol{x})] = \mathbb{1}\{i = j\} \langle \boldsymbol{v}, \boldsymbol{w} \rangle^i$.

Definition (Information exponent)

Suppose $\sigma = \sum_{i=1}^{\infty} \hat{\sigma}_i h_i$. The information exponent of σ is

$$\mathrm{IE}(\sigma) := \min \{ i > 0 \, : \, \hat{\sigma}_i \neq 0 \} .$$

Information exponent [Ben Arous, Gheissari, Jagannath, 2021]

Hermite expansion. $\sigma(z) = \sum_{i=0}^{\infty} \hat{\sigma}_i h_i$, where h_i is the i-th (normalized) Hermite polynomial and $\hat{\sigma}_i = \mathbb{E}_{z \sim \mathcal{N}(0,1)}[\sigma(z) h_i(z)]$.

- **Fact.** $\mathbb{E}_{\boldsymbol{x}}[h_i(\boldsymbol{v} \cdot \boldsymbol{x}) h_j(\boldsymbol{w} \cdot \boldsymbol{x})] = \mathbb{1}\{i = j\} \langle \boldsymbol{v}, \boldsymbol{w} \rangle^i$.

Definition (Information exponent)

Suppose $\sigma = \sum_{i=1}^{\infty} \hat{\sigma}_i h_i$. The information exponent of σ is

$$\mathrm{IE}(\sigma) := \min \{ i > 0 \, : \, \hat{\sigma}_i \neq 0 \}.$$

$$\mathop{\mathbb{E}}_{\boldsymbol{x}} [\sigma(\boldsymbol{w}_* \cdot \boldsymbol{x}) \sigma(\boldsymbol{w} \cdot \boldsymbol{x})] = \sum_{i,j=\mathrm{IE}}^{\infty} \hat{\sigma}_i \hat{\sigma}_j \mathop{\mathbb{E}}_{\boldsymbol{x}} [h_i(\boldsymbol{w}^* \cdot \boldsymbol{x}) h_j(\boldsymbol{w} \cdot \boldsymbol{x})]$$

Information exponent [Ben Arous, Gheissari, Jagannath, 2021]

Hermite expansion. $\sigma(z) = \sum_{i=0}^{\infty} \hat{\sigma}_i h_i$, where h_i is the i-th (normalized) Hermite polynomial and $\hat{\sigma}_i = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[\sigma(z)h_i(z)]$.

- **Fact.** $\mathbb{E}_{\boldsymbol{x}}[h_i(\boldsymbol{v}\cdot\boldsymbol{x})h_j(\boldsymbol{w}\cdot\boldsymbol{x})] = \mathbb{1}\{i=j\}\left\langle \boldsymbol{v},\boldsymbol{w}\right\rangle^i$.

Definition (Information exponent)

Suppose $\sigma = \sum_{i=1}^{\infty} \hat{\sigma}_i h_i$. The information exponent of σ is

$$\mathrm{IE}(\sigma) := \min\left\{i > 0 \,:\, \hat{\sigma}_i \neq 0\right\}.$$

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[\sigma(\boldsymbol{w}_*\cdot\boldsymbol{x})\sigma(\boldsymbol{w}\cdot\boldsymbol{x})\right] &= \sum_{i,j=\mathrm{IE}}^{\infty} \hat{\sigma}_i\hat{\sigma}_j \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[h_i(\boldsymbol{w}^*\cdot\boldsymbol{x})h_j(\boldsymbol{w}\cdot\boldsymbol{x})\right] \\ &= \sum_{i=\mathrm{IE}}^{\infty} \hat{\sigma}_i^2 \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[h_i(\boldsymbol{w}^*\cdot\boldsymbol{x})h_i(\boldsymbol{w}\cdot\boldsymbol{x})\right] \\ &\quad + \sum_{i\neq j} \hat{\sigma}_i\hat{\sigma}_j \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[h_i(\boldsymbol{w}^*\cdot\boldsymbol{x})h_j(\boldsymbol{w}\cdot\boldsymbol{x})\right]\end{aligned}$$

Information exponent [Ben Arous, Gheissari, Jagannath, 2021]

Hermite expansion. $\sigma(z) = \sum_{i=0}^{\infty} \hat{\sigma}_i h_i$, where h_i is the i-th (normalized) Hermite polynomial and $\hat{\sigma}_i = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[\sigma(z)h_i(z)]$.

- **Fact.** $\mathbb{E}_{\boldsymbol{x}}[h_i(\boldsymbol{v}\cdot\boldsymbol{x})h_j(\boldsymbol{w}\cdot\boldsymbol{x})] = \mathbb{1}\{i=j\}\langle \boldsymbol{v}, \boldsymbol{w}\rangle^i$.

Definition (Information exponent)

Suppose $\sigma = \sum_{i=1}^{\infty} \hat{\sigma}_i h_i$. The information exponent of σ is

$$\mathrm{IE}(\sigma) := \min\{i > 0 : \hat{\sigma}_i \neq 0\}.$$

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}}[\sigma(\boldsymbol{w}_*\cdot\boldsymbol{x})\sigma(\boldsymbol{w}\cdot\boldsymbol{x})] &= \sum_{i,j=\mathrm{IE}}^{\infty} \hat{\sigma}_i\hat{\sigma}_j \mathbb{E}_{\boldsymbol{x}}[h_i(\boldsymbol{w}^*\cdot\boldsymbol{x})h_j(\boldsymbol{w}\cdot\boldsymbol{x})] \\
&= \underbrace{\hat{\sigma}_{\mathrm{IE}}^2 \langle \boldsymbol{w}^*, \boldsymbol{w}\rangle^{\mathrm{IE}}}_{\text{the dominating term}} + \sum_{i=\mathrm{IE}+1}^{\infty} \hat{\sigma}_i^2 \langle \boldsymbol{w}^*, \boldsymbol{w}\rangle^i \\
&\quad \cancel{+ \sum_{i\neq j} \hat{\sigma}_i\hat{\sigma}_j \mathbb{E}_{\boldsymbol{x}}[h_i(\boldsymbol{w}^*\cdot\boldsymbol{x})h_j(\boldsymbol{w}\cdot\boldsymbol{x})]}
\end{aligned}$$

Information exponent [Ben Arous, Gheissari, Jagannath, 2021]

Theorem ([BAGJ21])

Suppose $\mathrm{IE}(\sigma) = k$ *and our algorithm is online (spherical) SGD with step size* $\eta = \tilde{\Theta}(1/d^{k/2 \vee 1})$*. Then, we can recover* $\boldsymbol{w}^*$ *with*

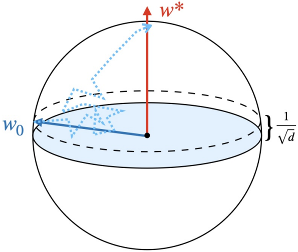
- $O(1/\eta) = \tilde{O}(d)$ *iterations/samples if* $k = 1$*;*
- $O(\log d/\eta) = \tilde{O}(d \log d)$ *iterations/samples if* $k = 2$*;*
- $O(d^{k/2-1}/\eta) = \tilde{O}(d^{k-1})$ *iterations/samples if* $k \geq 3$*.*

Information exponent [Ben Arous, Gheissari, Jagannath, 2021]

Theorem ([BAGJ21])

Suppose $\mathrm{IE}(\sigma) = k$ *and our algorithm is online (spherical) SGD with step size* $\eta = \tilde{\Theta}(1/d^{k/2 \vee 1})$*. Then, we can recover* $\boldsymbol{w}^*$ *with*

- $O(1/\eta) = \tilde{O}(d)$ *iterations/samples if* $k = 1$*;*
- $O(\log d/\eta) = \tilde{O}(d \log d)$ *iterations/samples if* $k = 2$*;*
- $O(d^{k/2-1}/\eta) = \tilde{O}(d^{k-1})$ *iterations/samples if* $k \geq 3$*.*



Emergent behavior:

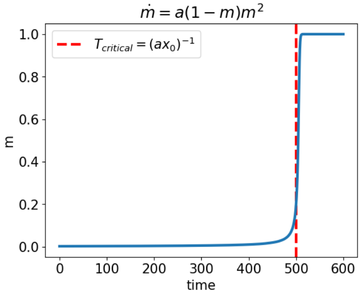
When $k = \mathrm{IE} \geq 3$,

- From $d^{-1/2}$ to $d^{-1/2+\delta}$: $\tilde{\Theta}(d^{k-1})$ steps;
- From $d^{-1/2+\delta}$ to $1 - \varepsilon$: $o(d^{k-1})$ steps.

[BAGJ21] Proof sketch

(Assume $\mathrm{IE} = 4$ for simplicity)
Dynamics of $m_t := \langle \boldsymbol{w}^*, \boldsymbol{w}_t \rangle^2$:

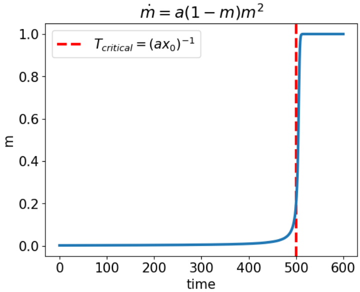
$$m_0 \approx 1/d,$$
$$m_{t+1} \approx m_t + \eta a(1 - m_t) m_t^2 + \eta^2 \zeta_{t+1}$$



[BAGJ21] Proof sketch

(Assume $\mathrm{IE} = 4$ for simplicity)
Dynamics of $m_t := \langle \boldsymbol{w}^*, \boldsymbol{w}_t \rangle^2$:

$$m_0 \approx 1/d,$$
$$m_{t+1} \approx m_t + \eta a(1 - m_t)m_t^2 + \eta^2 \zeta_{t+1}$$



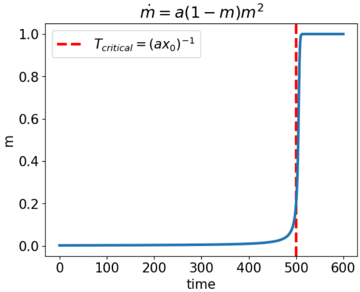
- (Need $\eta = \tilde{O}(1/d^2)$ to absorb the noise into the signal)

[BAGJ21] Proof sketch

(Assume $\mathrm{IE} = 4$ for simplicity)
Dynamics of $m_t := \langle \boldsymbol{w}^*, \boldsymbol{w}_t \rangle^2$:

$$m_0 \approx 1/d,$$

$$m_{t+1} \approx m_t + \eta a(1 - m_t) m_t^2 + \eta^2 \zeta_{t+1}$$



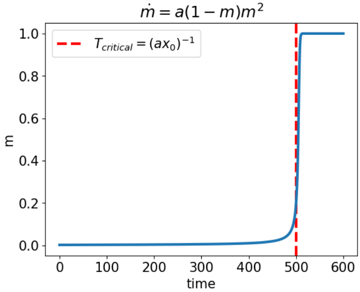
- (Need $\eta = \tilde{O}(1/d^2)$ to absorb the noise into the signal)
- Continuous-time counterpart:

$$\dot{m}_t = a(1 - m_t) m_t^2 \approx a m_t^2$$

[BAGJ21] Proof sketch

(Assume $\mathrm{IE} = 4$ for simplicity)
Dynamics of $m_t := \langle \boldsymbol{w}^*, \boldsymbol{w}_t \rangle^2$:

$$m_0 \approx 1/d,$$
$$m_{t+1} \approx m_t + \eta a(1 - m_t) m_t^2 + \eta^2 \zeta_{t+1}$$



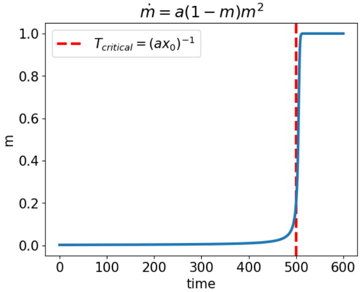
- (Need $\eta = \tilde{O}(1/d^2)$ to absorb the noise into the signal)
- Continuous-time counterpart:

$$\dot{m}_t = a(1 - m_t) m_t^2 \approx a m_t^2 \quad \Rightarrow \quad m_t \approx \frac{1}{1/m_0 - at}$$

[BAGJ21] Proof sketch

(Assume $\mathrm{IE} = 4$ for simplicity)
Dynamics of $m_t := \langle \boldsymbol{w}^*, \boldsymbol{w}_t \rangle^2$:

$$m_0 \approx 1/d,$$
$$m_{t+1} \approx m_t + \eta a(1 - m_t) m_t^2 + \eta^2 \zeta_{t+1}$$



- (Need $\eta = \tilde{O}(1/d^2)$ to absorb the noise into the signal)
- Continuous-time counterpart:

$$\dot{m}_t = a(1 - m_t) m_t^2 \approx a m_t^2 \quad \Rightarrow \quad m_t \approx \frac{1}{1/m_0 - at}$$

- $\Rightarrow$ sharp transition (faster than exponential) around time $1/(am_0) \approx d/a$.

The idealized dynamics

Our target function.

$$f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x}), \quad \boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I}_d),$$

(1) $P \ll d^c$; (2) $\{\boldsymbol{w}_p^*\}_p$ orthonormal; (3) σ even;
(4) For simplicity, assume $\mathrm{IE}(\sigma) = 4$.

The idealized dynamics

Our target function.

$$f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x}), \quad \boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I}_d),$$

(1) $P \ll d^c$; (2) $\{\boldsymbol{w}_p^*\}_p$ orthonormal; (3) σ even;
(4) For simplicity, assume $\mathrm{IE}(\sigma) = 4$.

- If we assume everything is decoupled ...
  - One $\boldsymbol{v}_p$ for one $a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x})$ and no interaction between them.

The idealized dynamics

Our target function.

$$f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x}), \quad \boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I}_d),$$

(1) $P \ll d^c$; (2) $\{\boldsymbol{w}_p^*\}_p$ orthonormal; (3) σ even;
(4) For simplicity, assume $\mathrm{IE}(\sigma) = 4$.

- If we assume everything is decoupled ...
  - One $\boldsymbol{v}_p$ for one $a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x})$ and no interaction between them.
- $\Rightarrow$ Direction $a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x})$ gets learned around time

$$T_p := \left(\eta a_p \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2 \right)^{-1}.$$

The idealized dynamics

Our target function.

$$f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x}), \quad \boldsymbol{x} \sim \mathcal{N}\left(0, \boldsymbol{I}_d\right),$$

(1) $P \ll d^c$; (2) $\{\boldsymbol{w}_p^*\}_p$ orthonormal; (3) σ even;
(4) For simplicity, assume $\mathrm{IE}(\sigma) = 4$.

- If we assume everything is decoupled ...
  - One $\boldsymbol{v}_p$ for one $a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x})$ and no interaction between them.
- $\Rightarrow$ Direction $a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x})$ gets learned around time

$$T_p := \left(\eta a_p \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2\right)^{-1}.$$

- $\Rightarrow$ Loss satisfies

$$\mathcal{L}(t) \approx \sum_{p=1}^{P} a_p^2 \mathbb{1}\left\{t < T_p\right\} = \sum_{p=1}^{P} a_p^2 \mathbb{1}\left\{t < \left(\eta a_p \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2\right)^{-1}\right\}.$$

From the idealized dynamics to the scaling law

$$\mathcal{L}(t) \approx \sum_{p=1}^{P} a_p^2 \mathbb{1}\left\{t < T_p\right\} = \sum_{p=1}^{P} a_p^2 \mathbb{1}\left\{t < \left(\eta a_p \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2\right)^{-1}\right\}.$$

Assumption (power law signal)

$a_p = p^{-\beta}$ for some constant $\beta > 1/2$.

From the idealized dynamics to the scaling law

$$\mathcal{L}(t) \approx \sum_{p=1}^{P} a_p^2 \mathbb{1}\{t < T_p\} = \sum_{p=1}^{P} a_p^2 \mathbb{1}\left\{t < \left(\eta a_p \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2\right)^{-1}\right\}.$$

Assumption (power law signal)

$a_p = p^{-\beta}$ for some constant $\beta > 1/2$.

$$\mathcal{L}(T_p) \approx \sum_{q=p}^{P} a_q^2$$

From the idealized dynamics to the scaling law

$$\mathcal{L}(t) \approx \sum_{p=1}^{P} a_p^2 \mathbb{1}\{t < T_p\} = \sum_{p=1}^{P} a_p^2 \mathbb{1}\left\{t < \left(\eta a_p \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2\right)^{-1}\right\}.$$

Assumption (power law signal)

$a_p = p^{-\beta}$ for some constant $\beta > 1/2$.

$$\mathcal{L}(T_p) \approx \sum_{q=p}^{P} a_q^2 = \sum_{q=p}^{P} q^{-2\beta}$$

From the idealized dynamics to the scaling law

$$\mathcal{L}(t) \approx \sum_{p=1}^{P} a_p^2 \mathbb{1}\{t < T_p\} = \sum_{p=1}^{P} a_p^2 \mathbb{1}\left\{t < \left(\eta a_p \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2\right)^{-1}\right\}.$$

Assumption (power law signal)

$a_p = p^{-\beta}$ for some constant $\beta > 1/2$.

$$\mathcal{L}(T_p) \approx \sum_{q=p}^{P} a_q^2 = \sum_{q=p}^{P} q^{-2\beta} \approx \sum_{q=p}^{\infty} q^{-2\beta}$$

From the idealized dynamics to the scaling law

$$\mathcal{L}(t) \approx \sum_{p=1}^{P} a_p^2 \mathbb{1}\left\{t < T_p\right\} = \sum_{p=1}^{P} a_p^2 \mathbb{1}\left\{t < \left(\eta a_p \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2\right)^{-1}\right\}.$$

Assumption (power law signal)

$a_p = p^{-\beta}$ for some constant $\beta > 1/2$.

$$\mathcal{L}(T_p) \approx \sum_{q=p}^{P} a_q^2 = \sum_{q=p}^{P} q^{-2\beta} \approx \sum_{q=p}^{\infty} q^{-2\beta} \approx \int_p^{\infty} s^{-2\beta}\,\mathrm{d}s = \frac{p^{1-2b}}{2b-1}.$$

From the idealized dynamics to the scaling law

$$\mathcal{L}(t) \approx \sum_{p=1}^{P} a_p^2 \mathbb{1}\{t < T_p\} = \sum_{p=1}^{P} a_p^2 \mathbb{1}\left\{t < \left(\eta a_p \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2\right)^{-1}\right\}.$$

Assumption (power law signal)

$a_p = p^{-\beta}$ for some constant $\beta > 1/2$.

$$\mathcal{L}(T_p) \approx \sum_{q=p}^{P} a_q^2 = \sum_{q=p}^{P} q^{-2\beta} \approx \sum_{q=p}^{\infty} q^{-2\beta} \approx \int_p^\infty s^{-2\beta}\,\mathrm{d}s = \frac{p^{1-2b}}{2b-1}.$$

Formal change-of-variables:

$$T_p = \left(\eta p^{-\beta} \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2\right)^{-1} = t \quad \Leftrightarrow \quad p = \left(\eta t \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2\right)^{1/\beta} \approx (\eta t/d)^{1/\beta}$$

From the idealized dynamics to the scaling law

$$\mathcal{L}(t) \approx \sum_{p=1}^{P} a_p^2 \mathbb{1}\{t < T_p\} = \sum_{p=1}^{P} a_p^2 \mathbb{1}\left\{t < \left(\eta a_p \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2\right)^{-1}\right\}.$$

Assumption (power law signal)

$a_p = p^{-\beta}$ for some constant $\beta > 1/2$.

$$\mathcal{L}(T_p) \approx \sum_{q=p}^{P} a_q^2 = \sum_{q=p}^{P} q^{-2\beta} \approx \sum_{q=p}^{\infty} q^{-2\beta} \approx \int_p^{\infty} s^{-2\beta}\, \mathrm{d}s = \frac{p^{1-2b}}{2b-1}.$$

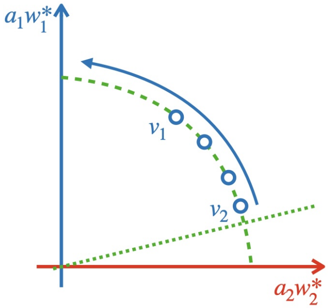
Formal change-of-variables:

$$T_p = \left(\eta p^{-\beta} \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2\right)^{-1} = t \quad \Leftrightarrow \quad p = \left(\eta t \left\langle \boldsymbol{w}_p^*, \bar{\boldsymbol{v}}_p \right\rangle^2\right)^{1/\beta} \approx (\eta t/d)^{1/\beta}$$

$$\Rightarrow \quad \mathcal{L}(t) \approx \frac{1}{2b-1} (\eta t/d)^{(1-2\beta)/\beta}$$

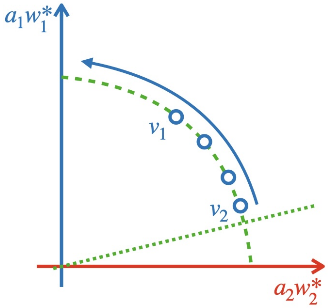
From the idealized to the actual dynamics

- **Issue of the existing analyses.**
  - Larger directions may attract too many neurons.
  - Need e^{κ} neurons to cover all directions.



From the idealized to the actual dynamics

- **Issue of the existing analyses.**
  - Larger directions may attract too many neurons.
  - Need e^{κ} neurons to cover all directions.

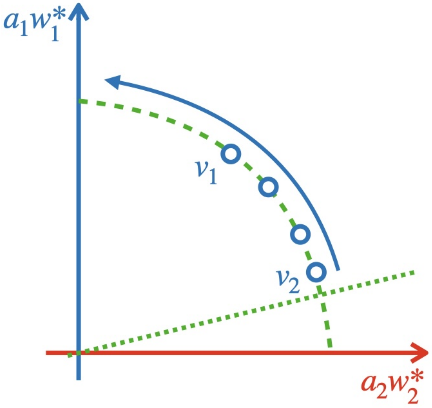


Claim 1. If all irrelevant coordinates $\bar{v}_{k,p}^2$ are $\tilde{O}(1/d)$, then the dynamics can be decoupled. (incoherence $\Rightarrow$ decoupled dynamics)

Claim 2. Sharp transitions $\Rightarrow$ small irrelevant coordinates.

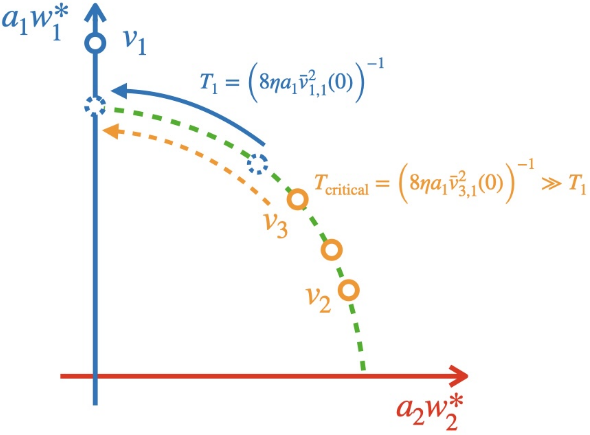
Proof intuition

- At first, most neurons get attracted by direction $a_1 \boldsymbol{w}_1^*$.
  - (Decoupled dynamics $\Rightarrow$ partial progress can be preserved.)



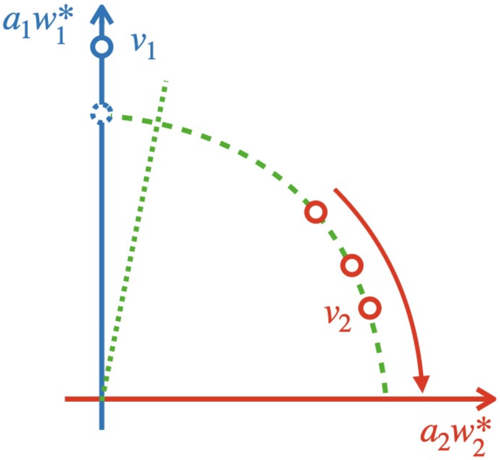
Proof intuition

- Sharp transitions $\Rightarrow \bar{v}_{3,1}^2 = \tilde{O}(1/d)$ until $t \approx T_{\mathrm{critical}}$.
- $\boldsymbol{v}_1$ fits $a_1 \boldsymbol{w}_1^*$ around time $T_1 < T_{\mathrm{critical}}$ and kills the signal.
- $\Rightarrow \bar{v}_{3,1}^2$ stays small throughout training.



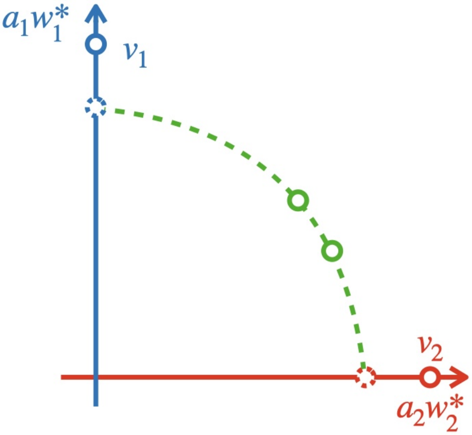
Proof intuition

- The remaining neurons get attracted by $a_2 \boldsymbol{w}_2^*$.



Proof intuition

- $\boldsymbol{v}_2$ fits $a_2\boldsymbol{w}_2^*$.
- The other neurons stay close to the initialization (and preserve the partial progress).



Main results

Theorem (Optimization)

- *Teacher network:* $f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x})$, *where* $P \ll d^c$, $\{\boldsymbol{w}_p^*\}_p$ *orthonormal,* σ *even and* $J := \mathrm{IE}(\sigma) \geq 4$.
- *Student network:* $f(\boldsymbol{x}) = \sum_{k=1}^{m} \|\boldsymbol{v}_k\|^2 \sigma(\bar{\boldsymbol{v}}_k \cdot \boldsymbol{x})$ *with* $m = O(P \log P)$.
- *Algorithm: online SGD with step size* $\eta = 1/(d^{J/2}\,\mathrm{poly}(P, \kappa))$.
- *Conclusion: there exists an injective* $\iota : [P] \to [m]$ *such that:*

  (a) **Unused neurons.** $\|\boldsymbol{v}_k\|$ *is small if* $k \notin \iota([P])$.

  (b) **Emergence.** $\forall p \in [P]$, $\boldsymbol{v}_{\iota(p)}$ *converges to and fits* $a_p \boldsymbol{w}_p^*$ *at time* $(1 \pm o(1))T_p$, *where* $T_p := 1/(8\eta a_p \langle \bar{\boldsymbol{v}}_{\iota(p)}, \boldsymbol{w}_p^* \rangle^{J-2})$.

Main results

Theorem (Optimization)

- *Teacher network:* $f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x})$, *where* $P \ll d^c$, $\{\boldsymbol{w}_p^*\}_p$ *orthonormal,* σ *even and* $J := \mathrm{IE}(\sigma) \geq 4$.
- *Student network:* $f(\boldsymbol{x}) = \sum_{k=1}^{m} \|\boldsymbol{v}_k\|^2 \sigma(\bar{\boldsymbol{v}}_k \cdot \boldsymbol{x})$ *with* $m = O(P \log P)$.
- *Algorithm: online SGD with step size* $\eta = 1/(d^{J/2} \operatorname{poly}(P, \kappa))$.
- *Conclusion: there exists an injective* $\iota : [P] \to [m]$ *such that:*

  (a) **Unused neurons.** $\|\boldsymbol{v}_k\|$ *is small if* $k \notin \iota([P])$.

  (b) **Emergence.** $\forall p \in [P]$, $\boldsymbol{v}_{\iota(p)}$ *converges to and fits* $a_p \boldsymbol{w}_p^*$ *at time* $(1 \pm o(1)) T_p$, *where* $T_p := 1/(8\eta a_p \langle \bar{\boldsymbol{v}}_{\iota(p)}, \boldsymbol{w}_p^* \rangle^{J-2})$.

Main results

Theorem (Optimization)

- *Teacher network:* $f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x})$, *where* $P \ll d^c$, $\{\boldsymbol{w}_p^*\}_p$ *orthonormal,* σ *even and* $J := \mathrm{IE}(\sigma) \geq 4$.
- *Student network:* $f(\boldsymbol{x}) = \sum_{k=1}^{m} \|\boldsymbol{v}_k\|^2 \sigma(\bar{\boldsymbol{v}}_k \cdot \boldsymbol{x})$ *with* $m = O(P \log P)$.
- *Algorithm: online SGD with step size* $\eta = 1/(d^{J/2} \operatorname{poly}(P, \kappa))$.
- *Conclusion: there exists an injective* $\iota : [P] \to [m]$ *such that:*

  (a) **Unused neurons.** $\|\boldsymbol{v}_k\|$ *is small if* $k \notin \iota([P])$.

  (b) **Emergence.** $\forall p \in [P]$, $\boldsymbol{v}_{\iota(p)}$ *converges to and fits* $a_p \boldsymbol{w}_p^*$ *at time* $(1 \pm o(1)) T_p$, *where* $T_p := 1/(8\eta a_p \langle \bar{\boldsymbol{v}}_{\iota(p)}, \boldsymbol{w}_p^* \rangle^{J-2})$.

Main results

Theorem (Optimization)

- *Teacher network:* $f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x})$, *where* $P \ll d^c$, $\{\boldsymbol{w}_p^*\}_p$ *orthonormal,* σ *even and* $J := \mathrm{IE}(\sigma) \geq 4$.
- *Student network:* $f(\boldsymbol{x}) = \sum_{k=1}^{m} \|\boldsymbol{v}_k\|^2 \sigma(\bar{\boldsymbol{v}}_k \cdot \boldsymbol{x})$ *with* $m = O(P \log P)$.
- *Algorithm: online SGD with step size* $\eta = 1/(d^{J/2} \operatorname{poly}(P, \kappa))$.
- *Conclusion: there exists an injective* $\iota : [P] \to [m]$ *such that:*

  (a) **Unused neurons.** $\|\boldsymbol{v}_k\|$ *is small if* $k \notin \iota([P])$.

  (b) **Emergence.** $\forall p \in [P]$, $\boldsymbol{v}_{\iota(p)}$ *converges to and fits* $a_p \boldsymbol{w}_p^*$ *at time* $(1 \pm o(1)) T_p$, *where* $T_p := 1/(8\eta a_p \langle \bar{\boldsymbol{v}}_{\iota(p)}, \boldsymbol{w}_p^* \rangle^{J-2})$.

Main results

Theorem (Optimization)

- *Teacher network:* $f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x})$, *where* $P \ll d^c$, $\{\boldsymbol{w}_p^*\}_p$ *orthonormal,* σ *even and* $J := \mathrm{IE}(\sigma) \geq 4$.
- *Student network:* $f(\boldsymbol{x}) = \sum_{k=1}^{m} \|\boldsymbol{v}_k\|^2 \sigma(\bar{\boldsymbol{v}}_k \cdot \boldsymbol{x})$ *with* $m = O(P \log P)$.
- *Algorithm: online SGD with step size* $\eta = 1/(d^{J/2}\,\mathrm{poly}(P, \kappa))$.
- *Conclusion: there exists an injective* $\iota : [P] \to [m]$ *such that:*

  (a) **Unused neurons.** $\|\boldsymbol{v}_k\|$ *is small if* $k \notin \iota([P])$.
  (b) **Emergence.** $\forall p \in [P]$, $\boldsymbol{v}_{\iota(p)}$ *converges to and fits* $a_p \boldsymbol{w}_p^*$ *at time* $(1 \pm o(1))\, T_p$, *where* $T_p := 1/(8\eta a_p \langle \bar{\boldsymbol{v}}_{\iota(p)}, \boldsymbol{w}_p^* \rangle^{J-2})$.

Main results

Theorem (Optimization)

- *Teacher network:* $f_*(\boldsymbol{x}) = \sum_{p=1}^{P} a_p \sigma(\boldsymbol{w}_p^* \cdot \boldsymbol{x})$, *where* $P \ll d^c$, $\{\boldsymbol{w}_p^*\}_p$ *orthonormal,* σ *even and* $J := \mathrm{IE}(\sigma) \geq 4$.
- *Student network:* $f(\boldsymbol{x}) = \sum_{k=1}^{m} \|\boldsymbol{v}_k\|^2 \sigma(\bar{\boldsymbol{v}}_k \cdot \boldsymbol{x})$ *with* $m = O(P \log P)$.
- *Algorithm: online SGD with step size* $\eta = 1/(d^{J/2}\,\mathrm{poly}(P, \kappa))$.
- *Conclusion: there exists an injective* $\iota : [P] \to [m]$ *such that:*
  (a) **Unused neurons.** $\|\boldsymbol{v}_k\|$ *is small if* $k \notin \iota([P])$.
  (b) **Emergence.** $\forall p \in [P]$, $\boldsymbol{v}_{\iota(p)}$ *converges to and fits* $a_p \boldsymbol{w}_p^*$ *at time* $(1 \pm o(1))T_p$, *where* $T_p := 1/(8\eta a_p \langle \bar{\boldsymbol{v}}_{\iota(p)}, \boldsymbol{w}_p^* \rangle^{J-2})$.

Corollary (Scaling laws)

$a_p \propto p^{-\beta}$ *for* $\beta > 1/2$. *Width-m learner (maybe under-parameterized). Online SGD with step size* η *and* t *iterations/samples.*

$$\mathcal{L}(m, t) \sim m^{1-2\beta} \vee \left(\eta t d^{1-J/2}\right)^{\frac{1-2\beta}{\beta}}$$

Conclusion and remarks

Takeaway

- The additive model hypothesis is true at least for orthogonal two-layer networks.
- Learning different directions/features with vastly different signal strength without deflation/reinitialization is possible.
- Sharp transitions help preserve the randomness from the initialization and prevent model collapse.

Remarks on sharp transitions

- Higher-order terms $\Rightarrow$ sharp transitions.
- Examples of sharp transitions.
  - $\mathcal{L}(w) = (w_* - w)^2$, $\mathcal{L}(w) = (w_* - w^2)^2$. ✗
  - $\mathcal{L}(\boldsymbol{w}) = (w_* - w_1 w_2 w_3)^2$, $\mathcal{L}(w) = (w_* - w^k)^2$, $k \geq 3$. ✓
- **Q.** Do deep architectures always lead to sharp transitions?
- **Q.** Do sharp transitions help training/feature learning in practice?

Conclusion and remarks

Takeaway

- The additive model hypothesis is true at least for orthogonal two-layer networks.
- Learning different directions/features with vastly different signal strength without deflation/reinitialization is possible.
- Sharp transitions help preserve the randomness from the initialization and prevent model collapse.

Remarks on sharp transitions

- Higher-order terms $\Rightarrow$ sharp transitions.
- Examples of sharp transitions.
  - $\mathcal{L}(w) = (w_* - w)^2$, $\mathcal{L}(w) = (w_* - w^2)^2$. ✗
  - $\mathcal{L}(\boldsymbol{w}) = (w_* - w_1 w_2 w_3)^2$, $\mathcal{L}(w) = (w_* - w^k)^2$, $k \geq 3$. ✓
- **Q.** Do deep architectures always lead to sharp transitions?
- **Q.** Do sharp transitions help training/feature learning in practice?

Conclusion and remarks

Takeaway

- The additive model hypothesis is true at least for orthogonal two-layer networks.
- Learning different directions/features with vastly different signal strength without deflation/reinitialization is possible.
- Sharp transitions help preserve the randomness from the initialization and prevent model collapse.

Remarks on sharp transitions

- Higher-order terms $\Rightarrow$ sharp transitions.
- Examples of sharp transitions.
  - $\mathcal{L}(w) = (w_* - w)^2$, $\mathcal{L}(w) = (w_* - w^2)^2$. ✗
  - $\mathcal{L}(\boldsymbol{w}) = (w_* - w_1 w_2 w_3)^2$, $\mathcal{L}(w) = (w_* - w^k)^2, k \geq 3$. ✓
- **Q.** Do deep architectures always lead to sharp transitions?
- **Q.** Do sharp transitions help training/feature learning in practice?

Conclusion and remarks

Takeaway

- The additive model hypothesis is true at least for orthogonal two-layer networks.
- Learning different directions/features with vastly different signal strength without deflation/reinitialization is possible.
- Sharp transitions help preserve the randomness from the initialization and prevent model collapse.

Remarks on sharp transitions

- Higher-order terms $\Rightarrow$ sharp transitions.
- Examples of sharp transitions.
  - $\mathcal{L}(w) = (w_* - w)^2$, $\mathcal{L}(w) = (w_* - w^2)^2$. X
  - $\mathcal{L}(\boldsymbol{w}) = (w_* - w_1 w_2 w_3)^2$, $\mathcal{L}(w) = (w_* - w^k)^2, k \geq 3$. ✓
- **Q.** Do deep architectures always lead to sharp transitions?
- **Q.** Do sharp transitions help training/feature learning in practice?

Conclusion and remarks

Takeaway

- The additive model hypothesis is true at least for orthogonal two-layer networks.
- Learning different directions/features with vastly different signal strength without deflation/reinitialization is possible.
- Sharp transitions help preserve the randomness from the initialization and prevent model collapse.

Remarks on sharp transitions

- Higher-order terms $\Rightarrow$ sharp transitions.
- Examples of sharp transitions.
  - $\mathcal{L}(w) = (w_* - w)^2$, $\mathcal{L}(w) = (w_* - w^2)^2$. X
  - $\mathcal{L}(\boldsymbol{w}) = (w_* - w_1 w_2 w_3)^2$, $\mathcal{L}(w) = (w_* - w^k)^2$, $k \geq 3$. ✓
- **Q.** Do deep architectures always lead to sharp transitions?
- **Q.** Do sharp transitions help training/feature learning in practice?

Conclusion and remarks

Takeaway

- The additive model hypothesis is true at least for orthogonal two-layer networks.
- Learning different directions/features with vastly different signal strength without deflation/reinitialization is possible.
- Sharp transitions help preserve the randomness from the initialization and prevent model collapse.

Remarks on sharp transitions

- Higher-order terms $\Rightarrow$ sharp transitions.
- Examples of sharp transitions.
  - $\mathcal{L}(w) = (w_* - w)^2$, $\mathcal{L}(w) = (w_* - w^2)^2$. X
  - $\mathcal{L}(\boldsymbol{w}) = (w_* - w_1 w_2 w_3)^2$, $\mathcal{L}(w) = (w_* - w^k)^2, k \geq 3$. ✓
- **Q.** Do deep architectures always lead to sharp transitions?
- **Q.** Do sharp transitions help training/feature learning in practice?