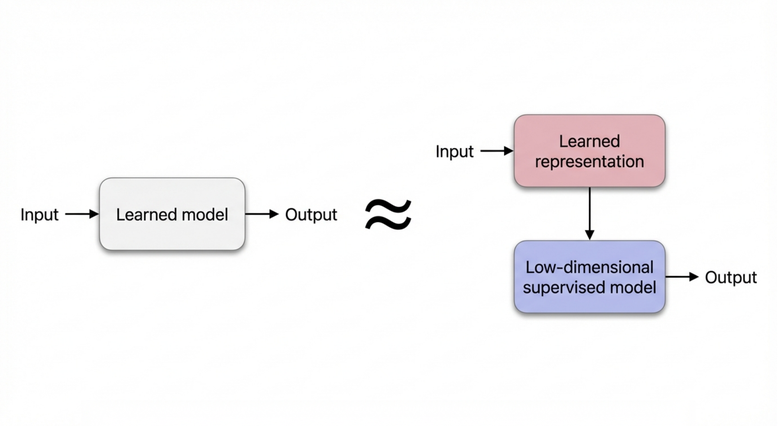
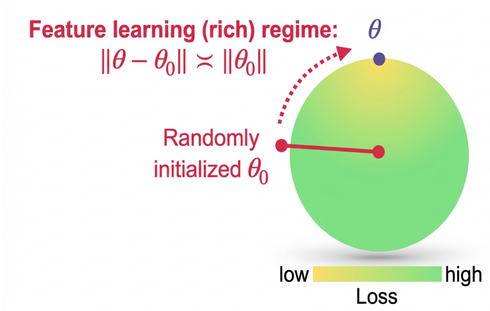
Phase Transitions for Feature Learning in Neural Networks

Andrea Montanari Zihao Wang
Stanford University

Feature learning





Neural networks learn by first identifying effective **low-dimensional features**, then fitting a model in that 'feature space'.

Multi-index models

Given n i.i.d. samples $(x_i, y_i)_{i\le n}$, $x_i \sim \mathsf{N}(0, I_d)$:

$$y_i = h\big(\Theta_*^\top x_i,\ \varepsilon_i\big)$$

- $\Theta_* \in \mathbb{R}^{d\times k}$ $(k = O(1))$: unknown latent directions
- $h : \mathbb{R}^{k+1} \to \mathbb{R}$: link function
- **Feature learning** = learning $\mathrm{span}(\Theta_*)$ from data

Neural networks

$$f_\Theta(x) = \frac{1}{m}\sum_{j=1}^{m} a_j \sigma(\theta_j^\top x + b_j), \qquad m = O(1)$$

Train first-layer weights Θ via **gradient descent**

$$\mathrm{Risk}(\Theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_\Theta(x_i))$$

$$\Theta(t+1) = \Theta(t) - \eta \nabla_\Theta \mathrm{Risk}(\Theta(t))$$

Proportional asymptotics: $n, d \to \infty$, $n/d \to \delta$

Key question

What is the threshold δ_{NN}
for feature learning via GD on neural networks?

- Do NNs achieve the algorithmic threshold δ_{alg}?
- How does δ_{NN} depend on activation, loss, learning rate?
- What is the mechanism that determines δ_{NN}?

Main contribution

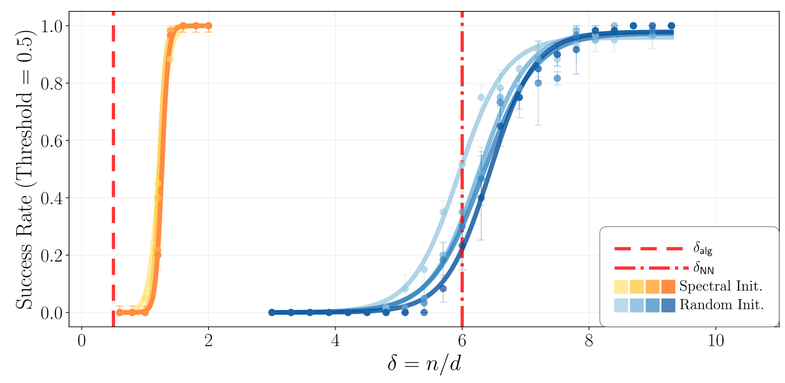
Given target h, loss ℓ, activation σ, learning rate η, we derive an **exact, explicit, easy-to-compute** threshold δ_{NN}:

- $\delta > \delta_{\mathsf{NN}}$: $\Rightarrow$ **feature learning***
- $\delta < \delta_{\mathsf{NN}}$: $\Rightarrow$ **no feature learning**

*Feature learning = weak recovery of the 'non-trivial' directions of Θ_*.

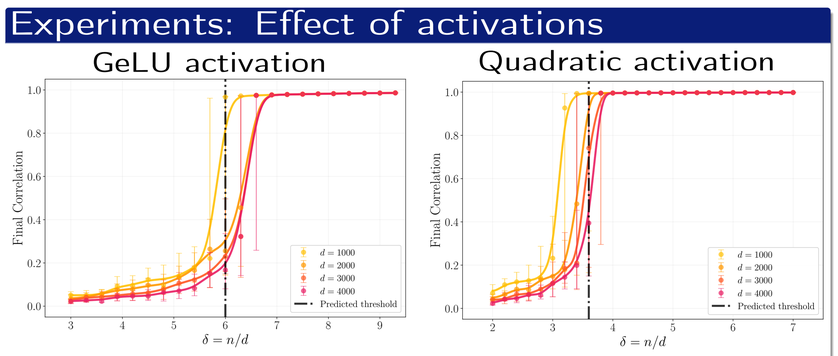
δ_{NN} is explicit enough to study how learning depends on target, loss, activation, width, and initialization.

Experiments: Phase transition



$h(z) = z^2$ noiseless phase retrieval. $\Rightarrow$ Factor-12 gap: $\delta_{\mathsf{alg}} = 0.5$ vs. $\delta_{\mathsf{NN}} \approx 6.0$.

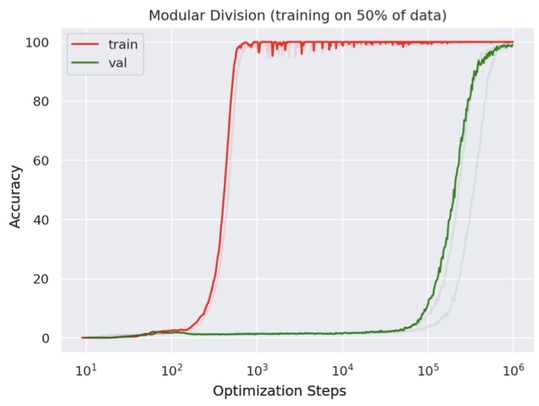
Experiments: Effect of activations



Different activations $\Rightarrow$ different δ_{NN}.

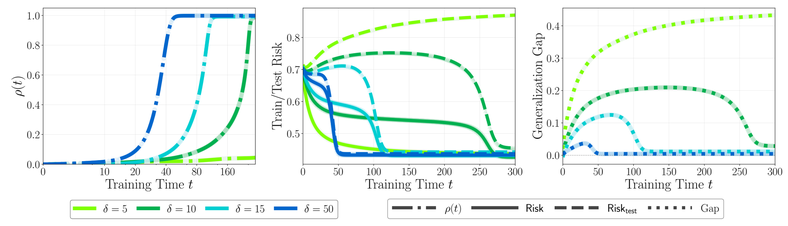
Grokking

Grokking = delayed generalization: training error drops quickly, but test error stays high for a long time before suddenly dropping.



Our explanation via δ_{NN}:

- In the first stage, the network **overfits** training data without learning features. Train loss $\ll$ test loss.
- Since $\delta > \delta_{\mathsf{NN}}$, the network eventually **learns features** in the second stage $\Rightarrow$ test loss drops $\Rightarrow$ **grokking!**



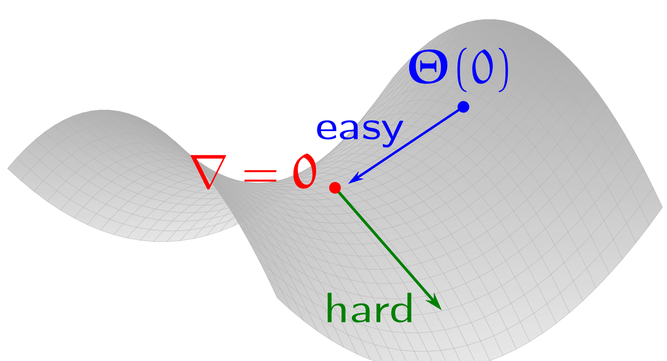
Grokking $\Leftrightarrow$ δ **moderately** above δ_{NN}

$\delta \gg \delta_{\mathsf{NN}}$: no overfitting in Stage 1 $\Rightarrow$ no grokking

Learning dynamics

$\mathrm{span}(\Theta_*)$ decomposes into **easy** directions and **hard** directions.

GD learns easy directions in $O(1)$ steps, then approaches a **saddle point**. Escaping along hard directions requires a **negative informative Hessian eigenvalue**:



Hessian structure

Study rescaled Hessian $H(t) := m\nabla^2 \mathrm{Risk}(\Theta(t))$ along the GD trajectory.

Case $m = 1$ (single neuron): $H(t) \in \mathbb{R}^{d\times d}$

$$H(t) = \frac{1}{n}\sum_{i=1}^{n} g(y_i, \theta(t)^\top x_i)\, x_i x_i^\top$$

Case $m \gg 1$ (wide network): $H(t) \in \mathbb{R}^{md\times md}$

- m^2 blocks of size $d \times d$; for convex loss: well-approximated by **block-diagonal $H_{\mathrm{diag}}(t)$**, up to $O(1/m)$

The j-th diagonal block is $a_j H_j(t)$:

$$H_j(t) = \frac{1}{n}\sum_{i=1}^{n} \ell'\big(y_i, f_\Theta(x_i)\big)\, \sigma''\big(\theta_j^\top x_i + b_j\big)\, x_i x_i^\top$$

$\Rightarrow$ Both cases: **spiked random matrix** with training-dependent weights.

Main theorem: Spectral phase transition at the saddle

Holds for $m = 1$ or $m \gg 1$ (m independent of n, d); stated for $m \gg 1$ for convenience.

After $O(1)$ GD steps, iterates approach a saddle. We derive **explicit** thresholds δ_j^* for each Hessian block:

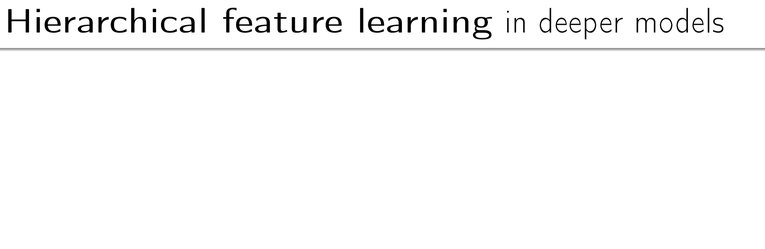
If $\delta > \delta_j^*$: $\exists$ outlier eigenvalue $\lambda_{n,d}$ with eigenvector $\xi_{n,d}$:

$$\lambda_{n,d} \xrightarrow{p} \lambda_j^* < \min\big(0,\ \inf(\mathrm{supp}(\mu_\infty^j))\big)$$

$$\left\|\Theta_{*H}^\top \xi_{n,d}\right\| / \|\xi_{n,d}\| \xrightarrow{p} c_j^* > 0$$

$\Rightarrow$ **escape saddle, learn features!**

If $\delta < \delta_j^*$: No eigenvector of H_j correlates with hard subspace $\Rightarrow$ no feature learning.



$\delta_{\mathsf{NN}} := \min_{j\in[m]} \delta_j^*$

Proof techniques

DMFT (GD dynamics) $\to$ **Gaussian conditioning** (identify the spiked structure of Hessian) $\to$ **RMT** (outlier localization via resolvent analysis)

Future directions

- Thresholds for **mini-batch SGD** and other algorithms
- Results for **finite width $m = O(1)$**
- **Training both layers** simultaneously
- Beyond $O(1)$ time: precise **long-time dynamics for learning hard directions**
- Thresholds for learning the **whole subspace** (2nd, 3rd, ... hard directions)
- **Non-isotropic** covariates
- **Hierarchical feature learning** in deeper models