Scaling laws, from Perceptrons to Deep networks

Francesco D'Amico

Dipartimento di Fisica

October 23, 2025

Outline of the talk

1. Review on neural scaling law
   - Empirical findings on neural scaling laws
   - Two models to predict power-laws exponents
   - Discussion (1^o part)

2. Our results (with Dario Bocchi and Matteo Negri)
   - Simple perceptron model
   - Experiments on deep networks
   - Discussion (2^o part)

Part IA: Empirical findings

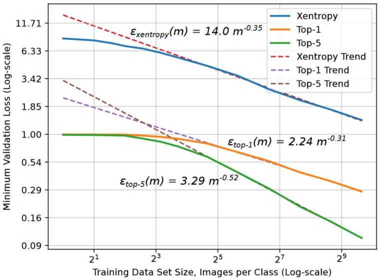
- Neural scaling laws phenomenology
- Why they motivated large scale LLMs like GPT-3/4
- How to use them to optimize compute cost

Glossary

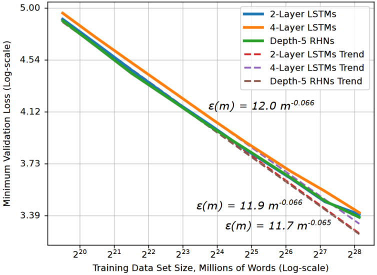
- P: number of training data
- N: total number of learnable parameters
- $\mathscr{L}$: generalization loss, i.e. cross-entropy in classification
- ε: generalization error

Hestness et al (2017):
Deep Learning Scaling is Predictable, Empirically

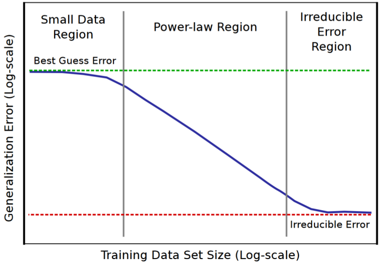
ResNet, image classification



LLM, next word prediction



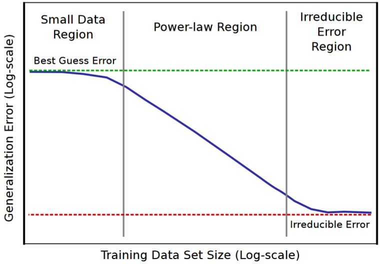
Hestness et al (2017): Deep Learning Scaling is Predictable, Empirically



Power law in intermediate regime:

$$\mathscr{L}(P) \sim cP^{-\gamma}$$

Hestness et al (2017):
Deep Learning Scaling is Predictable, Empirically



Power law in intermediate regime:

$$\mathscr{L}(P) \sim cP^{-\gamma}$$

Empirical properties of curves for model tested:

- Power laws in all domains tested
- Exponent γ depends on task/dataset
- Architectures change mainly constant c
- Same for optimizers (SGD, Adam ..)

Rosenfeld et al. (2020): A Constructive Prediction of the Generalization Error Across Scales

Two different scaling laws:

$$\varepsilon(N,P) \approx \begin{cases} aP^{-\alpha} + c_P(N) & \text{(data scaling at fixed model)} \\ bN^{-\beta} + c_N(P) & \text{(model scaling at fixed dataset)} \end{cases}$$

($P =$ #data, $N =$ #parameters)

Rosenfeld et al. (2020): A Constructive Prediction of the Generalization Error Across Scales

Two different scaling laws:

$$\varepsilon(N,P) \approx \begin{cases} aP^{-\alpha} + c_P(N) \\ bN^{-\beta} + c_N(P) \end{cases}$$

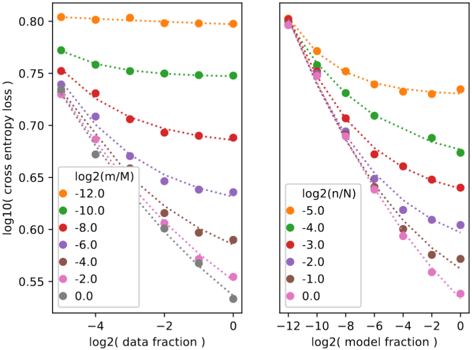
Saturating constant depends on the fixed parameter

Rosenfeld et al. (2020): A Constructive Prediction of the Generalization Error Across Scales

Two different scaling laws:

$$\varepsilon(N,P) \approx \begin{cases} aP^{-\alpha} + c_P(N) \\ bN^{-\beta} + c_N(P) \end{cases}$$

Saturating constant depends on the fixed parameter

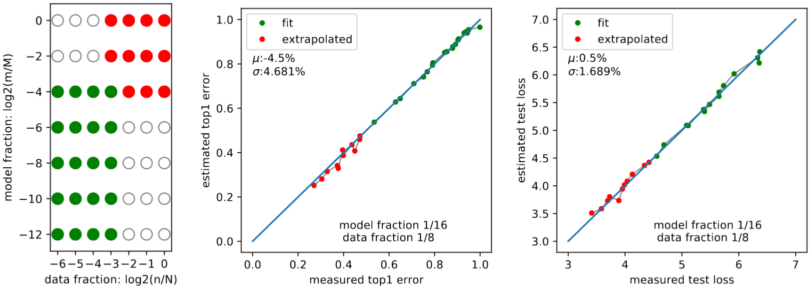


Rosenfeld et al. (2020): A Constructive Prediction of the Generalization Error Across Scales

Proposed scaling: $\varepsilon(N,P) = aP^{-\alpha} + bN^{-\beta} + c_\infty$

Rosenfeld et al. (2020): A Constructive Prediction of the Generalization Error Across Scales

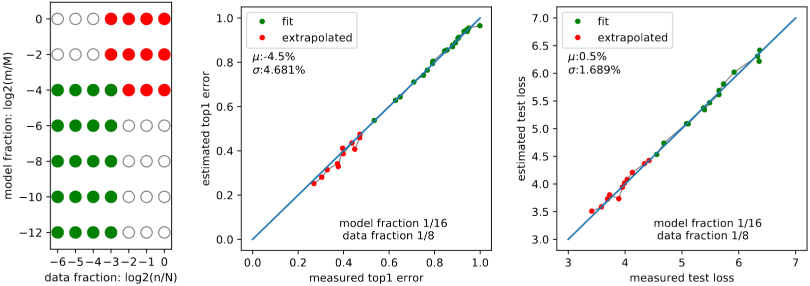
Proposed scaling: $\varepsilon(N,P) = aP^{-\alpha} + bN^{-\beta} + c_\infty$



(a) Illustration. (b) Extrapolation on ImageNet (c) Extrapolation on WikiText-103.

Rosenfeld et al. (2020): A Constructive Prediction of the Generalization Error Across Scales

Proposed scaling: $\varepsilon(N,P) = aP^{-\alpha} + bN^{-\beta} + c_\infty$

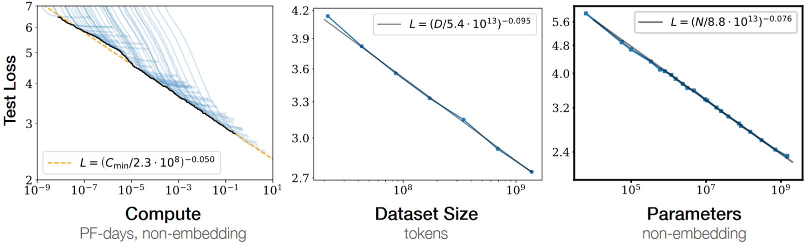


(a) Illustration. (b) Extrapolation on ImageNet (c) Extrapolation on WikiText-103.

$\Rightarrow$ small P,N models capable of predicting large P,N models

Kaplan et al (2020):
Scaling laws for neural language models

Almost perfect scaling laws in GPT models across many magnitudes



Kaplan et al (2020): Scaling laws for neural language models

Language modeling performance improves smoothly and predictably:

Kaplan et al (2020): Scaling laws for neural language models

Language modeling performance improves smoothly and predictably:

- Performance depends strongly on scale, weakly on model shape (i.e. width vs depth)

Kaplan et al (2020): Scaling laws for neural language models

Language modeling performance improves smoothly and predictably:

- Performance depends strongly on scale, weakly on model shape (i.e. width vs depth)
- Maximum exponent by scaling in tandem N, P

Kaplan et al (2020): Scaling laws for neural language models

Language modeling performance improves smoothly and predictably:

- Performance depends strongly on scale, weakly on model shape (i.e. width vs depth)
- Maximum exponent by scaling in tandem N, P
- Large models more sample-efficient than small models: same performance with fewer datapoints

Kaplan et al (2020): Scaling laws for neural language models

Language modeling performance improves smoothly and predictably:

- Performance depends strongly on scale, weakly on model shape (i.e. width vs depth)
- Maximum exponent by scaling in tandem N, P
- Large models more sample-efficient than small models: same performance with fewer datapoints
- Given a fixed compute budget C, best strategy $\Rightarrow$ very large model stopped very short of convergence

Kaplan et al (2020): Scaling laws for neural language models

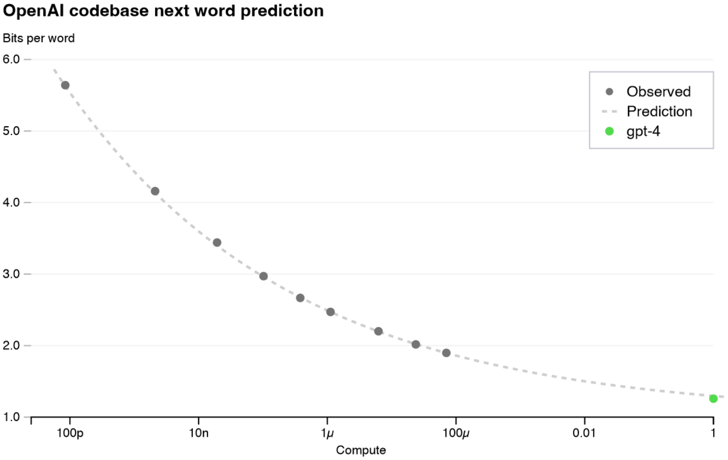
Language modeling performance improves smoothly and predictably:

- Performance depends strongly on scale, weakly on model shape (i.e. width vs depth)
- Maximum exponent by scaling in tandem N, P
- Large models more sample-efficient than small models: same performance with fewer datapoints
- Given a fixed compute budget C, best strategy $\Rightarrow$ very large model stopped very short of convergence

"Scaling is all you need"

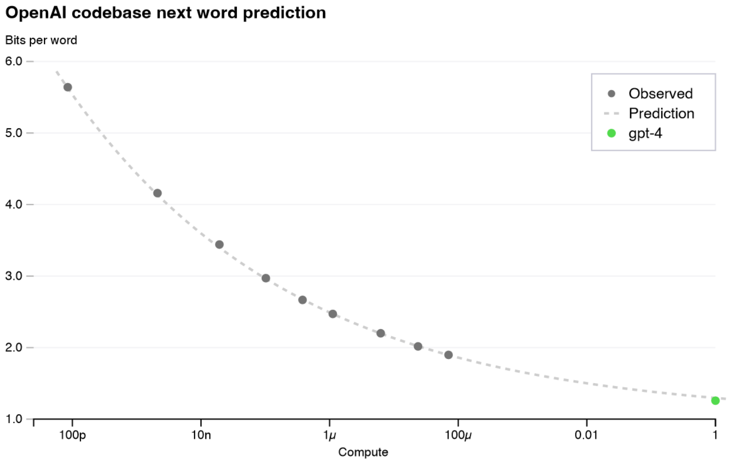
Kaplan et al (2020): Scaling laws for neural language models

All those results motivated extreme P, N scaling $\Rightarrow$ GPT-3/4 models



Kaplan et al (2020): Scaling laws for neural language models

All those results motivated extreme P, N scaling $\Rightarrow$ GPT-3/4 models



Smaller models fit predicted GPT-4 loss

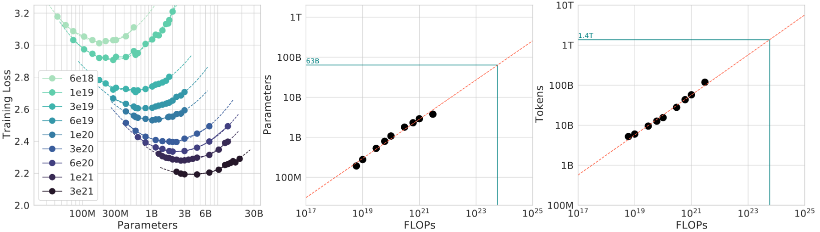
Hoffmann et al. (2022): Training Compute-Optimal Large Language Models

Given an available compute C, what is best choice of N, P?

Hoffmann et al. (2022): Training Compute-Optimal Large Language Models

Given an available compute C, what is best choice of N, P?

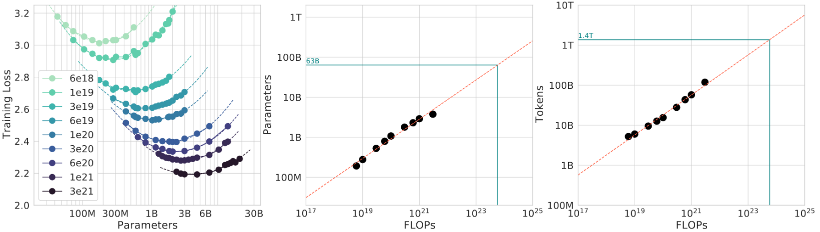
Isocurves at fixed C



Hoffmann et al. (2022): Training Compute-Optimal Large Language Models

Given an available compute C, what is best choice of N, P?

Isocurves at fixed C



$$\Rightarrow P_{\text{opt}}(C), N_{\text{opt}}(C) \text{ both } \sim C^{0.5}$$

Chinchilla scaling law

Summary of empirical results

1. Loss/error scales as $\varepsilon(N,P) = aP^{-\alpha} + bN^{-\beta} + c_\infty$
2. Exponents robusts wrt most of details of training and architectures
3. Exponents found $\in [0.05, 0.5]$
4. Best strategy given a compute C to scale $P, N \sim C^{0.5}$

Part IB. Two attempts to explain exponents: geometric bounds and DMFT models

Bahri et al. (2021): Explaining Neural Scaling Laws

Idea:

Bahri et al. (2021): Explaining Neural Scaling Laws

Idea:

Case $\mathscr{L}(P) - \mathscr{L}(\infty) = \Delta(P)$:

1. **Underparametrized** ($P \gg N \gg 1$): **variance** dominates $\Delta(P) \sim c_{\text{var}} P^{-1}$ (infinite limit + corrections)

Bahri et al. (2021): Explaining Neural Scaling Laws

Idea:

Case $\mathscr{L}(P) - \mathscr{L}(\infty) = \Delta(P)$:

1. **Underparametrized** ($P \gg N \gg 1$): **variance** dominates
   $\Delta(P) \sim c_{\text{var}} P^{-1}$ (infinite limit + corrections)
2. **Overparametrized** ($N \gg P \gg 1$): **bias** dominates
   $\Delta(P) \sim c_{\text{bias}} P^{-\alpha_{\text{bias}}}$

Bahri et al. (2021): Explaining Neural Scaling Laws

Idea:

Case $\mathscr{L}(P) - \mathscr{L}(\infty) = \Delta(P)$:

1. **Underparametrized** ($P \gg N \gg 1$): **variance** dominates
   $\Delta(P) \sim c_{\text{var}} P^{-1}$ (infinite limit + corrections)
2. **Overparametrized** ($N \gg P \gg 1$): **bias** dominates
   $\Delta(P) \sim c_{\text{bias}} P^{-\alpha_{\text{bias}}}$

Case $\mathscr{L}(N) - \mathscr{L}(\infty) = \Delta(N)$:

1. **Overparametrized** ($N \gg P \gg 1$): **variance** dominates
   $\Delta(N) \sim c_{\text{var}} N^{-1}$ ($N^{-1/2}$ deep case) (infinite limit + corrections)
2. **Underparametrized** ($P \gg N \gg 1$): **bias** dominates
   $\Delta(N) \sim c_{\text{bias}} N^{-\alpha_{\text{bias}}}$

Bahri et al. (2021): Explaining Neural Scaling Laws

Idea:

1. Exponents $\{-1, -1/2\}$ in variance-dominated regimes
2. Different exponents in bias-dominated regimes

Bahri et al. (2021): Explaining Neural Scaling Laws

Idea:

1. Exponents $\{-1, -1/2\}$ in variance-dominated regimes
2. Different exponents in bias-dominated regimes

Prediction for bias-dominated:

Bahri et al. (2021): Explaining Neural Scaling Laws

Idea:

1. Exponents $\{-1, -1/2\}$ in variance-dominated regimes
2. Different exponents in bias-dominated regimes

Prediction for bias-dominated:

$$\boxed{\Delta(P) \sim P^{-1/d} \;\; ; \;\; \Delta(N) \sim N^{-1/d}}$$

Bahri et al. (2021): Explaining Neural Scaling Laws

Idea:

1. Exponents $\{-1, -1/2\}$ in variance-dominated regimes
2. Different exponents in bias-dominated regimes

Prediction for bias-dominated:

$$\boxed{\Delta(P) \sim P^{-1/d} \ ; \ \Delta(N) \sim N^{-1/d}}$$

Assuming:

- Data lie on d-dimensional hidden manifold
- Teacher-student: $y = F(x)$ and $\hat{y} = f(x)$

Bahri et al. (2021): Explaining Neural Scaling Laws

Analytical model: linear random features

Analytical model: linear random features

- Teacher

$$F(x) = \sum_{M=1}^{S} \omega_M F_M(x)$$

Bahri et al. (2021): Explaining Neural Scaling Laws

Analytical model: linear random features

- Teacher

$$F(x) = \sum_{M=1}^{S} \omega_M F_M(x)$$

- Student

$$f(x) = \sum_{\mu=1}^{N} \theta_\mu f_\mu(x)$$

Bahri et al. (2021): Explaining Neural Scaling Laws

Analytical model: linear random features

- Teacher

$$F(x) = \sum_{M=1}^{S} \omega_M F_M(x)$$

- Student

$$f(x) = \sum_{\mu=1}^{N} \theta_\mu f_\mu(x)$$

- $\omega_M \sim \mathcal{N}(0, 1/S)$, θ_M learnable

Bahri et al. (2021): Explaining Neural Scaling Laws

Analytical model: linear random features

- Teacher

$$F(x) = \sum_{M=1}^{S} \omega_M F_M(x)$$

- Student

$$f(x) = \sum_{\mu=1}^{N} \theta_\mu f_\mu(x)$$

- $\omega_M \sim \mathcal{N}(0, 1/S)$, θ_M learnable
- Student features $f_\mu \in P$-dimensional subspace of teacher features

Analytical model: linear random features

Key ingredient: power-laws in features and data

Analytical model: linear random features

Key ingredient: power-laws in features and data

- Feature-feature second moment matrix:

$$\mathscr{C} = \mathbb{E}_x[F(x)F^T(x)]$$

- Data-data second moment matrix:

$$\mathscr{K}(x,x') = \frac{1}{S}F^T(x)F(x')$$

Bahri et al. (2021): Explaining Neural Scaling Laws

Analytical model: linear random features

Key ingredient: power-laws in features and data

- Feature-feature second moment matrix:

$$\mathscr{C} = \mathbb{E}_x[F(x)F^T(x)]$$

- Data-data second moment matrix:

$$\mathscr{K}(x, x') = \frac{1}{S} F^T(x) F(x')$$

- $\mathscr{C}, \mathscr{K}$ share non-zero eigenvalues λ_i

Bahri et al. (2021): Explaining Neural Scaling Laws

Analytical model: linear random features

Key ingredient: power-laws in features and data

- Feature-feature second moment matrix:

$$\mathscr{C} = \mathbb{E}_x[F(x)F^T(x)]$$

- Data-data second moment matrix:

$$\mathscr{K}(x,x') = \frac{1}{S}F^T(x)F(x')$$

- $\mathscr{C}, \mathscr{K}$ share non-zero eigenvalues λ_i

- **Key ingredient**: power-law spectrum $\lambda_i = \frac{1}{i^{1+\alpha_K}}$

Bahri et al. (2021): Explaining Neural Scaling Laws

Analytical model: linear random features

Key ingredient: power-laws in features and data

- Feature-feature second moment matrix:

$$\mathscr{C} = \mathbb{E}_x[F(x)F^T(x)]$$

- Data-data second moment matrix:

$$\mathscr{K}(x,x') = \frac{1}{S}F^T(x)F(x')$$

- $\mathscr{C}, \mathscr{K}$ share non-zero eigenvalues λ_i

- **Key ingredient**: power-law spectrum $\lambda_i = \frac{1}{i^{1+\alpha_K}}$

Results:

1. $\mathscr{L}(P) \sim P^{-\alpha_K}, \mathscr{L}(N) \sim N^{-\alpha_K}$

Bahri et al. (2021): Explaining Neural Scaling Laws

Analytical model: linear random features

Key ingredient: power-laws in features and data

- Feature-feature second moment matrix:

$$\mathscr{C} = \mathbb{E}_x[F(x)F^T(x)]$$

- Data-data second moment matrix:

$$\mathscr{K}(x,x') = \frac{1}{S}F^T(x)F(x')$$

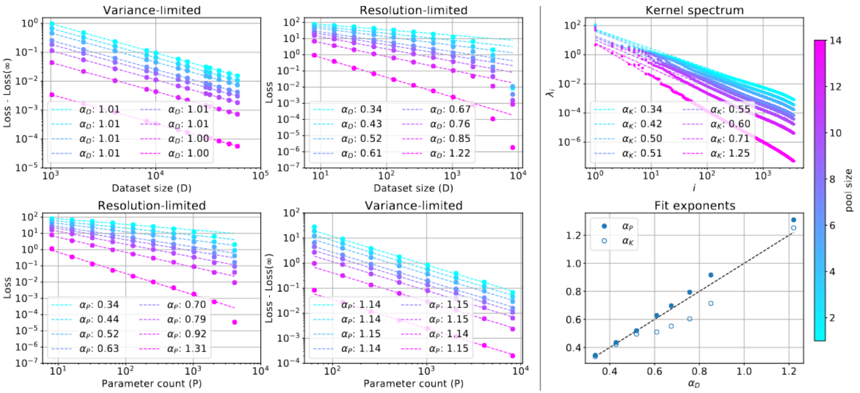
- $\mathscr{C}, \mathscr{K}$ share non-zero eigenvalues λ_i
- **Key ingredient**: power-law spectrum $\lambda_i = \frac{1}{i^{1+\alpha_K}}$

Results:

1. $\mathscr{L}(P) \sim P^{-\alpha_K}, \mathscr{L}(N) \sim N^{-\alpha_K}$
2. $\alpha_K \sim 1/d$

Bahri et al. (2021): Explaining Neural Scaling Laws

Result: linear random features



Bordelon et al. (2024):
A Dynamical Model of Neural Scaling Laws

Why studying dynamics?

They can address:

Bordelon et al. (2024): A Dynamical Model of Neural Scaling Laws

Why studying dynamics?

They can address:

- Scaling law in training time t

Bordelon et al. (2024): A Dynamical Model of Neural Scaling Laws

Why studying dynamics?

They can address:

- Scaling law in training time t
- Compute-optimal scalings

Bordelon et al. (2024): A Dynamical Model of Neural Scaling Laws

Why studying dynamics?

They can address:

- Scaling law in training time t
- Compute-optimal scalings
- All consistent at $t \to \infty$ with previous results

Bordelon et al. (2024):
A Dynamical Model of Neural Scaling Laws

Model: teacher-student random features

Bordelon et al. (2024): A Dynamical Model of Neural Scaling Laws

Model: teacher-student random features

- Data $\mathbf{x} \in \mathbb{R}^N$ drawn $\mathbf{x} \sim p(\mathbf{x})$

Bordelon et al. (2024): A Dynamical Model of Neural Scaling Laws

Model: teacher-student random features

- Data $\mathbf{x} \in \mathbb{R}^N$ drawn $\mathbf{x} \sim p(\mathbf{x})$
- Teacher from fixed features $\psi(\mathbf{x}) \in \mathbb{R}^M$ + noise:

$$y(\mathbf{x}) = \frac{1}{\sqrt{M}} \mathbf{w}^* \cdot \psi(\mathbf{x}) + \sigma \varepsilon(\mathbf{x})$$

Bordelon et al. (2024):
A Dynamical Model of Neural Scaling Laws

Model: teacher-student random features

- Data $\mathbf{x} \in \mathbb{R}^N$ drawn $\mathbf{x} \sim p(\mathbf{x})$
- Teacher from fixed features $\psi(\mathbf{x}) \in \mathbb{R}^M$ + noise:

$$y(\mathbf{x}) = \frac{1}{\sqrt{M}} \mathbf{w}^* \cdot \psi(\mathbf{x}) + \sigma \varepsilon(\mathbf{x})$$

- Student is a lower-dimensional projection of features $\mathbf{A}\psi(\mathbf{x})$ where $\mathbf{A} \in \mathbb{R}^{N \times M}$, A_{ij} i.i.d.

$$f(\mathbf{x}) = \frac{1}{\sqrt{N}} \mathbf{w} \cdot \mathbf{A}\psi(\mathbf{x})$$

Bordelon et al. (2024): A Dynamical Model of Neural Scaling Laws

Assumption: power-law features + data

1. Given $\langle \psi_k(\mathbf{x}) \psi_l(\mathbf{x}) \rangle_{\mathbf{x} \sim p(\mathbf{x})} = \delta_{kl} \lambda_k$ (fixed)

Bordelon et al. (2024): A Dynamical Model of Neural Scaling Laws

Assumption: power-law features + data

1. Given $\langle \psi_k(\mathbf{x}) \psi_l(\mathbf{x}) \rangle_{\mathbf{x} \sim p(\mathbf{x})} = \delta_{kl} \lambda_k$ (fixed)

$\Rightarrow$ assume $\lambda_k \sim k^{-b}$

Bordelon et al. (2024): A Dynamical Model of Neural Scaling Laws

Assumption: power-law features + data

1. Given $\langle \psi_k(\mathbf{x}) \psi_l(\mathbf{x}) \rangle_{\mathbf{x} \sim p(\mathbf{x})} = \delta_{kl} \lambda_k$ (fixed)

$$\Rightarrow \text{ assume } \lambda_k \sim k^{-b}$$

b inverse to data+kernel complexity

Bordelon et al. (2024): A Dynamical Model of Neural Scaling Laws

Assumption: power-law features + data

1. Given $\langle \psi_k(\mathbf{x}) \psi_l(\mathbf{x}) \rangle_{\mathbf{x} \sim p(\mathbf{x})} = \delta_{kl} \lambda_k$ (fixed)

   $$\Rightarrow \text{ assume } \lambda_k \sim k^{-b}$$

   b inverse to data+kernel complexity

2. Expand Teacher $f^*(\mathbf{x}) = \sum_k \omega_k^* \psi_k(\mathbf{x})$

   $$\Rightarrow \text{ assume } (\omega_k^*)^2 \lambda_k \sim k^{-a}$$

Bordelon et al. (2024): A Dynamical Model of Neural Scaling Laws

Assumption: power-law features + data

1. Given $\langle \psi_k(\mathbf{x}) \psi_l(\mathbf{x}) \rangle_{\mathbf{x} \sim p(\mathbf{x})} = \delta_{kl} \lambda_k$ (fixed)

   $\Rightarrow$ assume $\lambda_k \sim k^{-b}$

   b inverse to data+kernel complexity

2. Expand Teacher $f^*(\mathbf{x}) = \sum_k \omega_k^* \psi_k(\mathbf{x})$

   $\Rightarrow$ assume $(\omega_k^*)^2 \lambda_k \sim k^{-a}$

   - $(\omega_k^*)^2 \lambda_k$ controls generalization error per mode
   - Large a $\Rightarrow$ target error concentrated in first modes $\Rightarrow$ easy task

Bordelon et al. (2024):
A Dynamical Model of Neural Scaling Laws

DMFT results

(1) *Bottleneck scalings*

$$\mathscr{L}(t,P,N) \approx \begin{cases} t^{-\frac{a-1}{b}}, & P,N \to \infty \quad \text{(Time)}, \\ P^{-\min\{a-1,2b\}}, & t,N \to \infty \quad \text{(Data)}, \\ N^{-\min\{a-1,2b\}}, & t,P \to \infty \quad \text{(Model)}. \end{cases}$$

Bordelon et al. (2024): A Dynamical Model of Neural Scaling Laws

DMFT results

(1) *Bottleneck scalings*

$$\mathscr{L}(t,P,N) \approx \begin{cases} t^{-\frac{a-1}{b}}, & P,N \to \infty \quad \text{(Time)}, \\ P^{-\min\{a-1,2b\}}, & t,N \to \infty \quad \text{(Data)}, \\ N^{-\min\{a-1,2b\}}, & t,P \to \infty \quad \text{(Model)}. \end{cases}$$

(2) Compute optimal

- Compute optimal time-size: $\boxed{t \sim C^{\frac{b}{1+b}}\ ,\ N \sim C^{\frac{1}{1+b}}}$

  $\Rightarrow$ t has to be scaled more than N,P

Bordelon et al. (2024): A Dynamical Model of Neural Scaling Laws

DMFT results

(1) *Bottleneck scalings*

$$\mathscr{L}(t,P,N) \approx \begin{cases} t^{-\frac{a-1}{b}}, & P,N \to \infty \quad \text{(Time)}, \\ P^{-\min\{a-1,2b\}}, & t,N \to \infty \quad \text{(Data)}, \\ N^{-\min\{a-1,2b\}}, & t,P \to \infty \quad \text{(Model)}. \end{cases}$$

(2) Compute optimal

- Compute optimal time-size: $\boxed{t \sim C^{\frac{b}{1+b}}, \ N \sim C^{\frac{1}{1+b}}}$

  $\Rightarrow$ t has to be scaled more than N,P
- $\boxed{\mathscr{L}_{\text{opt}}(C) \sim C^{-\frac{a-1}{1+b}}}$

Limitations and new results

1. NTK/random features underestimate exponents

Limitations and new results

1. NTK/random features underestimate exponents

   Recent attempts with feature learning:

   - Bordelon et al. (ICLR 2025) How Feature Learning Can Improve Neural Scaling Laws
   - Defilippis et al. (Sept. 2025) Scaling Laws and Spectra of Shallow Neural Networks in the Feature Learning Regime

Limitations and new results

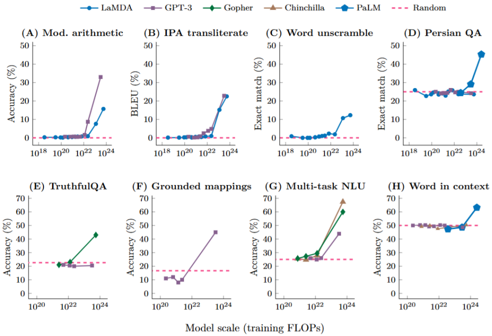
1. NTK/random features underestimate exponents

   Recent attempts with feature learning:

   - Bordelon et al. (ICLR 2025) How Feature Learning Can Improve Neural Scaling Laws
   - Defilippis et al. (Sept. 2025) Scaling Laws and Spectra of Shallow Neural Networks in the Feature Learning Regime

2. Different (complicated) tasks produce "phase-transitions"

   Wei et al., (2022): Emergent Abilities of Large Language Models



References

1. Hestness et al (2017): Deep Learning Scaling is Predictable, Empirically
2. Rosenfeld et al. (2020): A Constructive Prediction of the Generalization Error Across Scales
3. Kaplan et al (2020): Scaling laws for neural language models
4. Bahri et al. (2021): Explaining Neural Scaling Laws
5. Hoffmann et al. (2022): Training Compute-Optimal Large Language Models
6. Maloney et al. (2022): A Solvable Model of Neural Scaling Laws
7. Wei et al., (2022): Emergent Abilities of Large Language Models
8. Bordelon et al. (2024): A Dynamical Model of Neural Scaling Laws
9. Bordelon et al. (2025) How Feature Learning Can Improve Neural Scaling Laws
10. Defilippis et al. (2025) Scaling Laws and Spectra of Shallow Neural Networks in the Feature Learning Regime

Part II: Our work

Implicit bias produces neural scaling laws in learning curves, from perceptrons to deep networks

Francesco D'Amico[1,2]*, **Dario Bocchi**[1,2]*, **Matteo Negri**[1,2]

[1] Physics Department, University of Rome Sapienza, Piazzale Aldo Moro 5, Rome 00185

[2] 1CNR-Nanotec Rome unit, Piazzale Aldo Moro 5, Rome 00185

Part II: Our work

Implicit bias produces neural scaling laws in learning curves, from perceptrons to deep networks

Francesco D'Amico[1,2*], **Dario Bocchi**[1,2*], **Matteo Negri**[1,2]
[1] Physics Department, University of Rome Sapienza, Piazzale Aldo Moro 5, Rome 00185
[2] 1CNR-Nanotec Rome unit, Piazzale Aldo Moro 5, Rome 00185

Outline:

Part II: Our work

Implicit bias produces neural scaling laws in learning curves, from perceptrons to deep networks

Francesco D'Amico[1,2*], **Dario Bocchi**[1,2*], **Matteo Negri**[1,2]
[1] Physics Department, University of Rome Sapienza, Piazzale Aldo Moro 5, Rome 00185
[2] 1CNR-Nanotec Rome unit, Piazzale Aldo Moro 5, Rome 00185

Outline:

1. We show two new scalings laws in a simple Perceptron model

Part II: Our work

Implicit bias produces neural scaling laws in learning curves, from perceptrons to deep networks

Francesco D'Amico[1,2*], **Dario Bocchi**[1,2*], **Matteo Negri**[1,2]
[1] Physics Department, University of Rome Sapienza, Piazzale Aldo Moro 5, Rome 00185
[2] 1CNR-Nanotec Rome unit, Piazzale Aldo Moro 5, Rome 00185

Outline:

1. We show two new scalings laws in a simple Perceptron model
2. These new laws combined reproduce $\varepsilon \sim P^{-\gamma}$ scaling law

Part II: Our work

Implicit bias produces neural scaling laws in learning curves, from perceptrons to deep networks

Francesco D'Amico[1,2*], **Dario Bocchi**[1,2*], **Matteo Negri**[1,2]
[1] Physics Department, University of Rome Sapienza, Piazzale Aldo Moro 5, Rome 00185
[2] 1CNR-Nanotec Rome unit, Piazzale Aldo Moro 5, Rome 00185

Outline:

1. We show two new scalings laws in a simple Perceptron model
2. These new laws combined reproduce $\varepsilon \sim P^{-\gamma}$ scaling law
3. Valid empirically for Deep Nets in real image classification

Perceptron model

- *Student* perceptron $\boldsymbol{w} \in \mathbb{R}^N$, *Teacher* perceptron $\boldsymbol{w}^* \in \mathbb{R}^N$

Perceptron model

- *Student* perceptron $\boldsymbol{w} \in \mathbb{R}^N$, *Teacher* perceptron $\boldsymbol{w}^* \in \mathbb{R}^N$
- $P = \alpha N$ random binary examples $\boldsymbol{x}^\mu \in \{\pm 1\}^N$
- Labels $y^\mu = \text{sign}(\boldsymbol{x}^\mu \cdot \boldsymbol{w}^*)$

Perceptron model

- *Student* perceptron $\boldsymbol{w} \in \mathbb{R}^N$, *Teacher* perceptron $\boldsymbol{w}^* \in \mathbb{R}^N$
- $P = \alpha N$ random binary examples $\boldsymbol{x}^\mu \in \{\pm 1\}^N$
- Labels $y^\mu = \text{sign}(\boldsymbol{x}^\mu \cdot \boldsymbol{w}^*)$
- Spherical weights $\|\boldsymbol{w}^*\|^2 = \|\boldsymbol{w}\|^2 = \lambda N$

Perceptron model

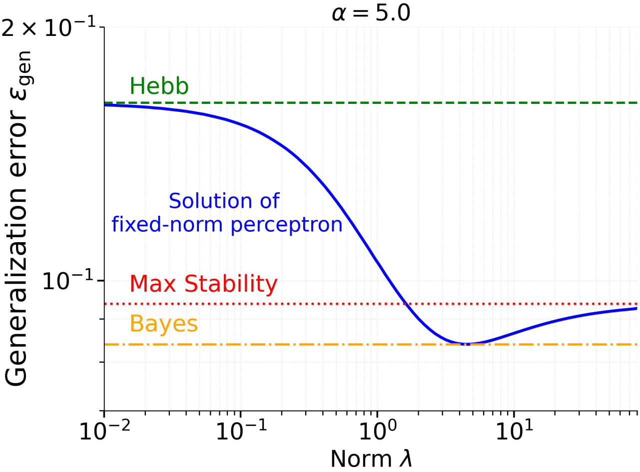
- *Student* perceptron $\boldsymbol{w} \in \mathbb{R}^N$, *Teacher* perceptron $\boldsymbol{w}^* \in \mathbb{R}^N$
- $P = \alpha N$ random binary examples $\boldsymbol{x}^\mu \in \{\pm 1\}^N$
- Labels $y^\mu = \text{sign}(\boldsymbol{x}^\mu \cdot \boldsymbol{w}^*)$
- Spherical weights $\|\boldsymbol{w}^*\|^2 = \|\boldsymbol{w}\|^2 = \lambda N$
- Cross-entropy (Pseudo-likelihood) Loss:

$$L(\boldsymbol{w};\lambda) = -\left[\sum_{\mu=1}^{P} \Delta^\mu - \log 2\cosh\left(\Delta^\mu\right)\right] = \sum_{\mu=1}^{P} V(\Delta^\mu)$$

where *margins*

$$\Delta^\mu \equiv y^\mu \left(\frac{\boldsymbol{w}\cdot\boldsymbol{x}^\mu}{\sqrt{\lambda N}}\right)$$

Solution at fixed α interpolates known learning rules



Unbounded norm perceptrons $\approx$ fixed-norm

- Norm $\lambda(t)$ increases monotonically for GD, Soudry et al., (2018)

Unbounded norm perceptrons $\approx$ fixed-norm

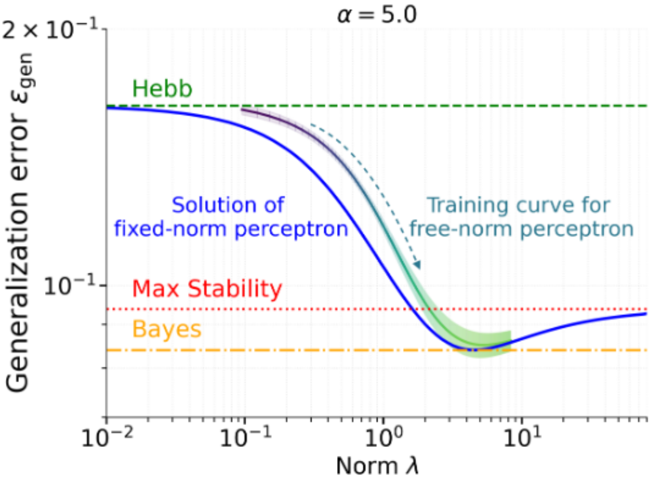
- Norm $\lambda(t)$ increases monotonically for GD, Soudry et al., (2018)
- $\varepsilon(\lambda)$ curves in fixed-norm case

Unbounded norm perceptrons $\approx$ fixed-norm

- Norm $\lambda(t)$ increases monotonically for GD, Soudry et al., (2018)
- $\varepsilon(\lambda)$ curves in fixed-norm case $\approx \varepsilon(\lambda(t))$ in unbounded case

Unbounded norm perceptrons ≈ fixed-norm

- Norm $\lambda(t)$ increases monotonically for GD, Soudry et al., (2018)
- $\varepsilon(\lambda)$ curves in fixed-norm case $\approx \varepsilon(\lambda(t))$ in unbounded case



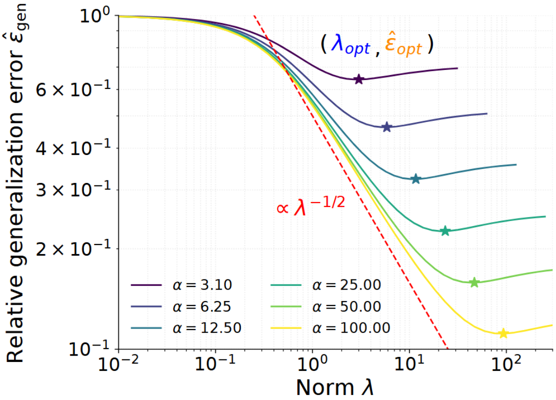
Result (1): Two new scaling laws in λ

Result (1): Two new scaling laws in λ

Relative error $\hat{\varepsilon}_{\text{gen}} \equiv \varepsilon_{\text{gen}}/\varepsilon_0$, where $\varepsilon_0 = \varepsilon(\lambda = 0)$

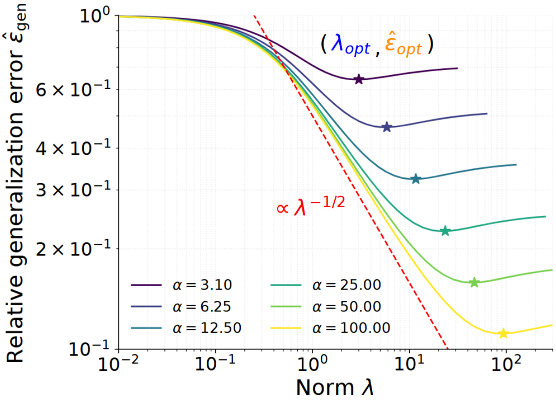
Result (1): Two new scaling laws in λ

Relative error $\hat{\varepsilon}_{\text{gen}} \equiv \varepsilon_{\text{gen}}/\varepsilon_0$, where $\varepsilon_0 = \varepsilon(\lambda = 0)$



Result (1): Two new scaling laws in λ

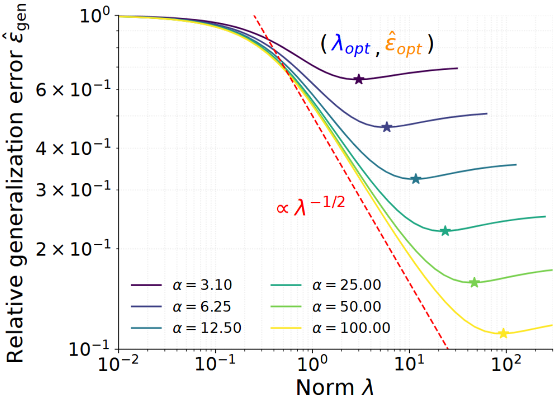
Relative error $\hat{\varepsilon}_{\text{gen}} \equiv \varepsilon_{\text{gen}}/\varepsilon_0$, where $\varepsilon_0 = \varepsilon(\lambda = 0)$



1. Early training ($\lambda < \lambda_{elbow}(\alpha)$) $\quad \rightarrow \quad \hat{\varepsilon}_{\text{gen}} \sim k_1 \lambda^{-\gamma_1}$

Result (1): Two new scaling laws in λ

Relative error $\hat{\varepsilon}_{\text{gen}} \equiv \varepsilon_{\text{gen}}/\varepsilon_0$, where $\varepsilon_0 = \varepsilon(\lambda = 0)$



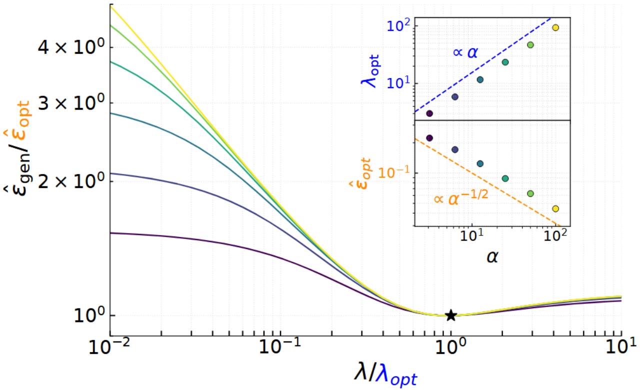
1. Early training ($\lambda < \lambda_{elbow}(\alpha)$) $\rightarrow$ $\hat{\varepsilon}_{\text{gen}} \sim k_1 \lambda^{-\gamma_1}$
2. Optima of curves ($\lambda > \lambda_{elbow}(\alpha)$) $\rightarrow$ $\lambda_{\text{opt}} \sim k_2 \alpha^{\gamma_2}$

Result (2): collapse on a master curve Φ

Define the rescaling $\hat{\varepsilon}_{\mathrm{gen}}/\hat{\varepsilon}_{\mathrm{opt}} = \Phi_\alpha(\lambda/\lambda_{\mathrm{opt}})$

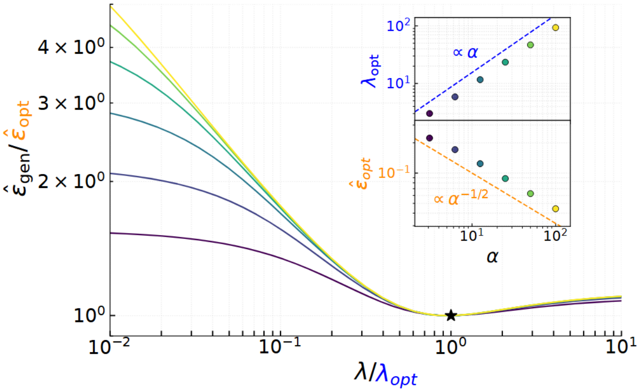
Result (2): collapse on a master curve Φ

Define the rescaling $\hat{\varepsilon}_{\text{gen}}/\hat{\varepsilon}_{\text{opt}} = \Phi_\alpha(\lambda/\lambda_{\text{opt}})$



Result (2): collapse on a master curve Φ

Define the rescaling $\hat{\varepsilon}_{\text{gen}}/\hat{\varepsilon}_{\text{opt}} = \Phi_\alpha(\lambda/\lambda_{\text{opt}})$



Curves converge to a master curve for $\alpha \gg 1$: $\Phi_\alpha \to \Phi$

Result (3): predict neural scaling law

1. $\hat{\varepsilon}_{\text{gen}} \sim k_1 \lambda^{-\gamma_1}$ for $\lambda < \lambda_{elbow}(\alpha)$
2. $\lambda_{\text{opt}} \sim k_2 \alpha^{\gamma_2}$ for $\lambda > \lambda_{elbow}(\alpha)$
3. $\hat{\varepsilon}_{\text{gen}} / \hat{\varepsilon}_{\text{opt}} = \Phi(\lambda / \lambda_{\text{opt}})$ for $\alpha \gg 1$

Result (3): predict neural scaling law

$$\left.\begin{array}{l} \text{1. } \hat{\varepsilon}_{\text{gen}} \sim k_1 \lambda^{-\gamma_1} \text{ for } \lambda < \lambda_{elbow}(\alpha) \\[2ex] \text{2. } \lambda_{\text{opt}} \sim k_2 \alpha^{\gamma_2} \text{ for } \lambda > \lambda_{elbow}(\alpha) \\[2ex] \text{3. } \hat{\varepsilon}_{\text{gen}}/\hat{\varepsilon}_{\text{opt}} = \Phi(\lambda/\lambda_{\text{opt}}) \text{ for } \alpha \gg 1 \end{array}\right\} \quad \hat{\varepsilon}_{\text{gen}} \sim k_1 k_2^{-\gamma_1} \alpha^{-\gamma_1 \gamma_2} \text{ for } \alpha \gg 1$$

Result (3): predict neural scaling law

1. $\hat{\varepsilon}_{\text{gen}} \sim k_1 \lambda^{-\gamma_1}$ for $\lambda < \lambda_{elbow}(\alpha)$
2. $\lambda_{\text{opt}} \sim k_2 \alpha^{\gamma_2}$ for $\lambda > \lambda_{elbow}(\alpha)$
3. $\hat{\varepsilon}_{\text{gen}} / \hat{\varepsilon}_{\text{opt}} = \Phi(\lambda / \lambda_{\text{opt}})$ for $\alpha \gg 1$

$\hat{\varepsilon}_{\text{gen}} \sim k_1 k_2^{-\gamma_1} \alpha^{-\gamma_1 \gamma_2}$ for $\alpha \gg 1$

$\hat{\varepsilon}_{\text{gen}} \sim \alpha^{-\gamma}$, with $\boxed{\gamma = \gamma_1 \gamma_2}$

Does the theory also apply to deep networks?

Architectures:

- Convolutional Neural Networks (CNN)
- Residual Neural Networks (ResNet)
- Vision Transformers (ViT)

Datasets:

- MNIST (greyscale digits, 10 classes)
- CIFAR10 (RGB images, 10 classes)
- CIFAR100 (RGB images, 100 classes)

Norm in deep networks: Bartlett et al. (2017) Spectrally-normalized margin bounds for neural networks

Spectral Complexity norm for a L-layer deep net with matrices A_i:

- ρ_i Lipschitz constant of layer i activation function
- $\|\cdot\|_\sigma$ biggest singular value (spectral norm)
- $\|\cdot\|_{2,1}$ sum of ℓ_2 norms of columns
- M_i reference matrix (can be $= \mathbf{0}$)

$$R_A = \left(\prod_{i=1}^{L} \rho_i \|A_i\|_\sigma \right) \left(\sum_{i=1}^{L} \frac{\|A_i^\top - M_i^\top\|_{2,1}^{2/3}}{\|A_i\|_\sigma^{2/3}} \right)^{3/2}$$

Norm in deep networks: Bartlett et al. (2017) Spectrally-normalized margin bounds for neural networks

Spectral Complexity norm for a L-layer deep net with matrices A_i:

- ρ_i Lipschitz constant of layer i activation function
- $\|\cdot\|_\sigma$ biggest singular value (spectral norm)
- $\|\cdot\|_{2,1}$ sum of ℓ_2 norms of columns
- M_i reference matrix (can be $= \mathbf{0}$)

$$R_A = \boxed{\left(\prod_{i=1}^{L} \rho_i \|A_i\|_\sigma\right)} \left(\sum_{i=1}^{L} \frac{\|A_i^\top - M_i^\top\|_{2,1}^{2/3}}{\|A_i\|_\sigma^{2/3}}\right)^{3/2}$$

Maximum expansion

Norm in deep networks: Bartlett et al. (2017) Spectrally-normalized margin bounds for neural networks

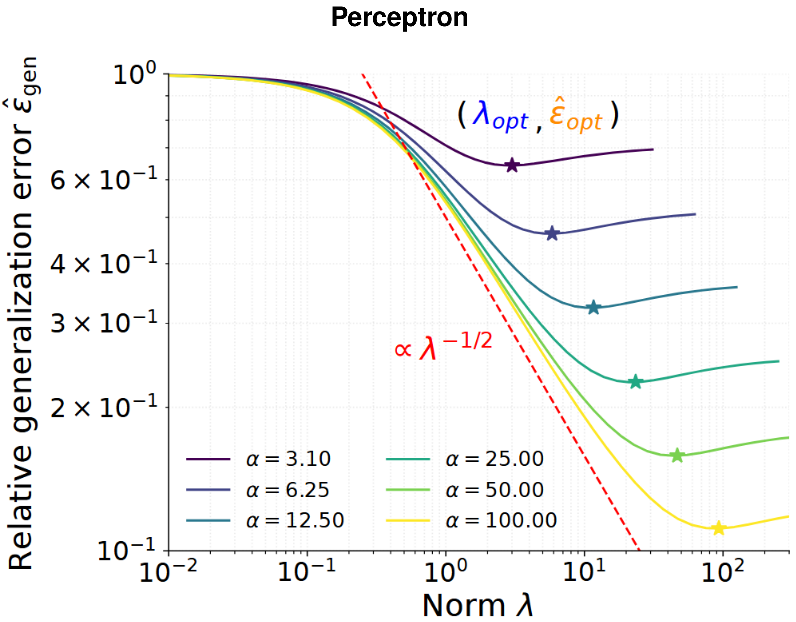
Spectral Complexity norm for a L-layer deep net with matrices A_i:

- ρ_i Lipschitz constant of layer i activation function
- $\|\cdot\|_\sigma$ biggest singular value (spectral norm)
- $\|\cdot\|_{2,1}$ sum of ℓ_2 norms of columns
- M_i reference matrix (can be $= \mathbf{0}$)

$$R_A = \boxed{\left(\prod_{i=1}^{L} \rho_i \|A_i\|_\sigma\right)} \boxed{\left(\sum_{i=1}^{L} \frac{\|A_i^\top - M_i^\top\|_{2,1}^{2/3}}{\|A_i\|_\sigma^{2/3}}\right)^{3/2}}$$

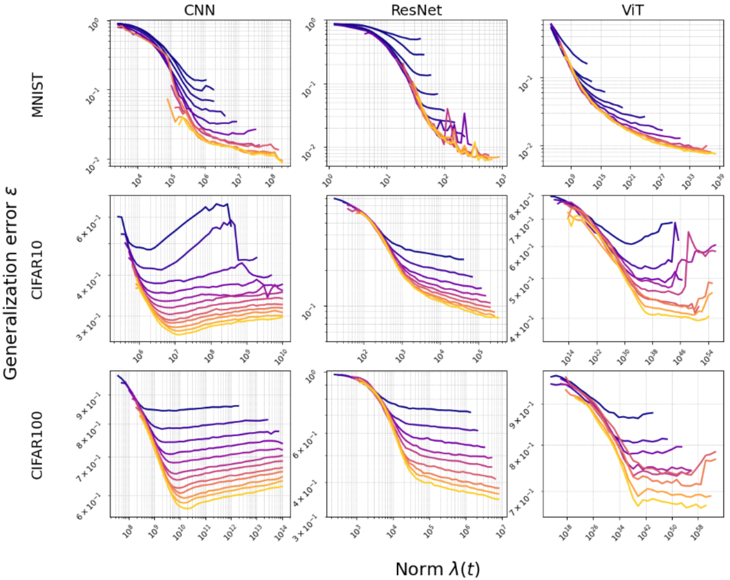
Maximum expansion **Effective rank**

Result (1): Two scaling laws



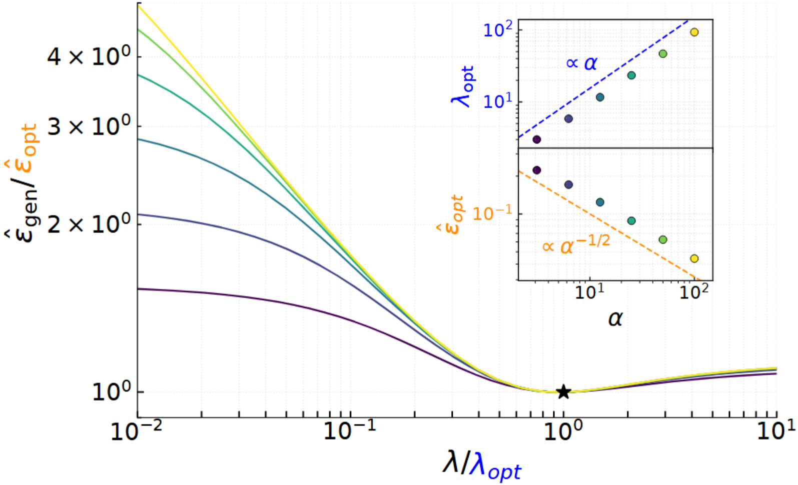
Result (1): Two scaling laws

Deep Networks



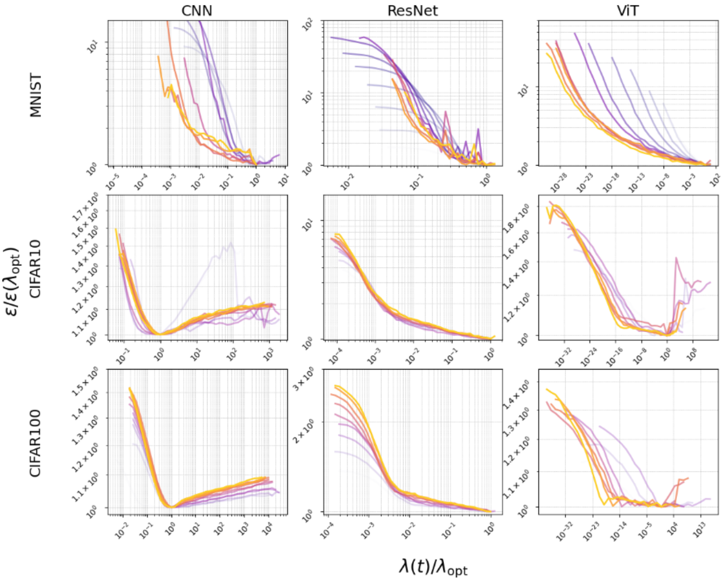
Result (2): Collapse on a master curve

Perceptron



Result (2): Collapse on a master curve

Deep Networks



Result (3): predict neural scaling law $\varepsilon_{gen} \sim P^{-\gamma}$

- Direct measure: γ_{meas}
- Measure γ_1, γ_2 and compute $\gamma_{pred} = \gamma_1 \gamma_2$

Result (3): predict neural scaling law $\varepsilon_{gen} \sim P^{-\gamma}$

- Direct measure: γ_{meas}
- Measure γ_1, γ_2 and compute $\gamma_{pred} = \gamma_1 \gamma_2$

In a realistic case:

$$\left.\begin{array}{ll} 1 & \varepsilon_{\text{gen}} = k_1 \lambda^{-\gamma_1} + q_1 \\ 2 & \lambda_{\text{opt}} = k_2 \alpha^{\gamma_2} + q_2 \end{array}\right\} \quad \varepsilon_{\text{gen}} = k_1 \left(k_2 P^{\gamma_2} + q_2\right)^{-\gamma_1} + q_1$$

Result (3): predict neural scaling law $\varepsilon_{gen} \sim P^{-\gamma}$

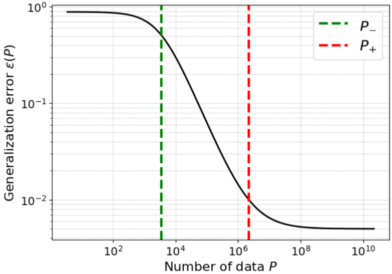
- Direct measure: γ_{meas}
- Measure γ_1, γ_2 and compute $\gamma_{pred} = \gamma_1 \gamma_2$

In a realistic case:

1. $\varepsilon_{\text{gen}} = k_1 \lambda^{-\gamma_1} + q_1$
2. $\lambda_{\text{opt}} = k_2 \alpha^{\gamma_2} + q_2$

$$\varepsilon_{\text{gen}} = k_1 \left(k_2 P^{\gamma_2} + q_2\right)^{-\gamma_1} + q_1$$

Intermediate regime $\varepsilon \sim P^{-\gamma_1 \gamma_2}$



Hestness et al (2017) empirical curve

Result (3): predict neural scaling law $\varepsilon_{gen} \sim P^{-\gamma}$

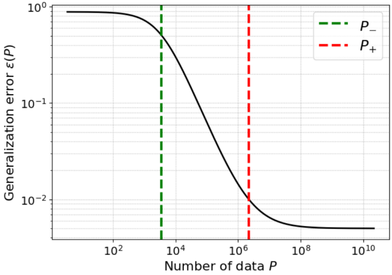
- Direct measure: γ_{meas}
- Measure γ_1, γ_2 and compute $\gamma_{pred} = \gamma_1 \gamma_2$

In a realistic case:

1. $\varepsilon_{\text{gen}} = k_1 \lambda^{-\gamma_1} + q_1$
2. $\lambda_{\text{opt}} = k_2 \alpha^{\gamma_2} + q_2$

$$\varepsilon_{\text{gen}} = k_1 \left(k_2 P^{\gamma_2} + q_2\right)^{-\gamma_1} + q_1$$

Intermediate regime $\varepsilon \sim P^{-\gamma_1 \gamma_2}$



Hestness et al (2017) empirical curve

Model	Dataset	γ_{pred}	γ_{meas}	σ
CNN	MNIST	0.60	0.55	0.09
CNN	CIFAR10	0.28	0.25	0.07
CNN	CIFAR100	0.16	0.16	0.03
ResNet	MNIST	0.57	0.69	0.08
ResNet	CIFAR10	0.54	0.56	0.04
ResNet	CIFAR100	0.31	0.37	0.03
ViT	MNIST	0.47	0.54	0.03
ViT	CIFAR10	0.23	0.21	0.03
ViT	CIFAR100	0.14	0.12	0.04

$\gamma_1 \gamma_2$ **compatible with** γ_{meas}

Robustness of phenomenology

Numerically we tested

1. A moderate weight decay

Robustness of phenomenology

Numerically we tested

1. A moderate weight decay
2. SGD instead of Adam

Robustness of phenomenology

Numerically we tested

1. A moderate weight decay
2. SGD instead of Adam
3. Other norm definitions: $\ell_1, \ell_2, G_{2,1}$, Spectral.

Robustness of phenomenology

Numerically we tested

1. A moderate weight decay
2. SGD instead of Adam
3. Other norm definitions: $\ell_1, \ell_2, G_{2,1}$, Spectral.

Results:

1. Qualitative picture is the same in all cases

Robustness of phenomenology

Numerically we tested

1. A moderate weight decay
2. SGD instead of Adam
3. Other norm definitions: $\ell_1, \ell_2, G_{2,1}$, Spectral.

Results:

1. Qualitative picture is the same in all cases
2. In (1) and (2) also $\gamma_1 \gamma_2$ compatible with γ (same γ as before)

Robustness of phenomenology

Numerically we tested

1. A moderate weight decay
2. SGD instead of Adam
3. Other norm definitions: $\ell_1, \ell_2, G_{2,1}$, Spectral.

Results:

1. Qualitative picture is the same in all cases
2. In (1) and (2) also $\gamma_1\gamma_2$ compatible with γ (same γ as before)
3. In (3) $\gamma_1\gamma_2 \neq \gamma \Rightarrow$ Spectral complexity is "special"

Limitations and possible extensions

- No hidden layer $\Rightarrow$ no scaling in N

  Extension: NTK or feature learning two-layers NN

Limitations and possible extensions

- No hidden layer $\Rightarrow$ no scaling in N

  Extension: NTK or feature learning two-layers NN

- Statical results to predict dynamics

  Extension: DMFT (i.e. Montanari and Urbani, (2025) Dynamical Decoupling of Generalization and Overfitting in Large Two-Layer Networks)

Limitations and possible extensions

- No hidden layer $\Rightarrow$ no scaling in N

  Extension: NTK or feature learning two-layers NN

- Statical results to predict dynamics

  Extension: DMFT (i.e. Montanari and Urbani, (2025) Dynamical Decoupling of Generalization and Overfitting in Large Two-Layer Networks)

- Only image classification

  Extension: LLMs (i.e. Maloney et al. (2022) A Solvable Model of Neural Scaling Laws)

Thank you for attention!

Francesco D'Amico

SAPIENZA
UNIVERSITÀ DI ROMA

Dipartimento di Fisica

October 23, 2025

Contact: francesco.damico@uniroma1.it

Bahri et al. (2021): Explaining Neural Scaling Laws

Idea:

Why $-1/d$ exponents? Arguments for *bounds*

1. Scaling in P (overparametrized):
   Distance of test points to closest training point $\mathcal{O}(P^{-1/d})$
2. Scaling in N (underparametrized):
   1. Take N *anchor* points $I = \{\mathbf{x}\}_{1,\ldots,N}$ from the huge dataset.
   2. f approximates F piecewise with N regions, centered on I points.
   3. Distance of test points to closest I: $\mathcal{O}(N^{-1/d})$