A Solvable Model of Neural Scaling Laws

Dan Roberts

MIT & Salesforce

August 26, 2022

Based on upcoming work w/ Alex Maloney and Jamie Sully.

LLMs are . . .

LLMs are . . . *Exciting*

LLMs are . . . *Exciting*

Large language models are *exciting*: they are really really good at language generation.

LLMs are . . . *Exciting*

Large language models are *exciting*: they are really really good at language generation.

- This is where I tell you that the text of this talk was actually secretly written by GPT-3 . . .

LLMs are . . . *Exciting*

Large language models are *exciting*: they are really really good at language generation.

- This is where I tell you that the text of this talk was actually secretly written by GPT-3 . . .
- . . . and while that's not actually true in this case the fact that you at least had to consider the possibility underscores the point that I want to make here.

LLMs are . . .

LLMs are . . . *Large*

Large language models are *large*. . .

LLMs are . . . *Large*

Large language models are *large*. . .

- . . . as in *size*: the Megatron-Turing NLG tops out at 530 billion parameters.

[Smith/Patwary/Microsoft/NVIDIA]

LLMs are . . . *Large*

Large language models are *large*. . .

- . . . as in *size*: the Megatron-Turing NLG tops out at 530 billion parameters.

[Smith/Patwary/Microsoft/NVIDIA]

- . . . as in (big) *data*: Chinchilla was trained on 1.4 trillion tokens.

[Hoffmann/Borgeaud/Mensch/DeepMind/Sifre]

LLMs are . . . *Large*

Large language models are *large*. . .

- . . . as in *size*: the Megatron-Turing NLG tops out at 530 billion parameters.

[Smith/Patwary/Microsoft/NVIDIA]

- . . . as in (big) *data*: Chinchilla was trained on 1.4 trillion tokens.

[Hoffmann/Borgeaud/Mensch/DeepMind/Sifre]

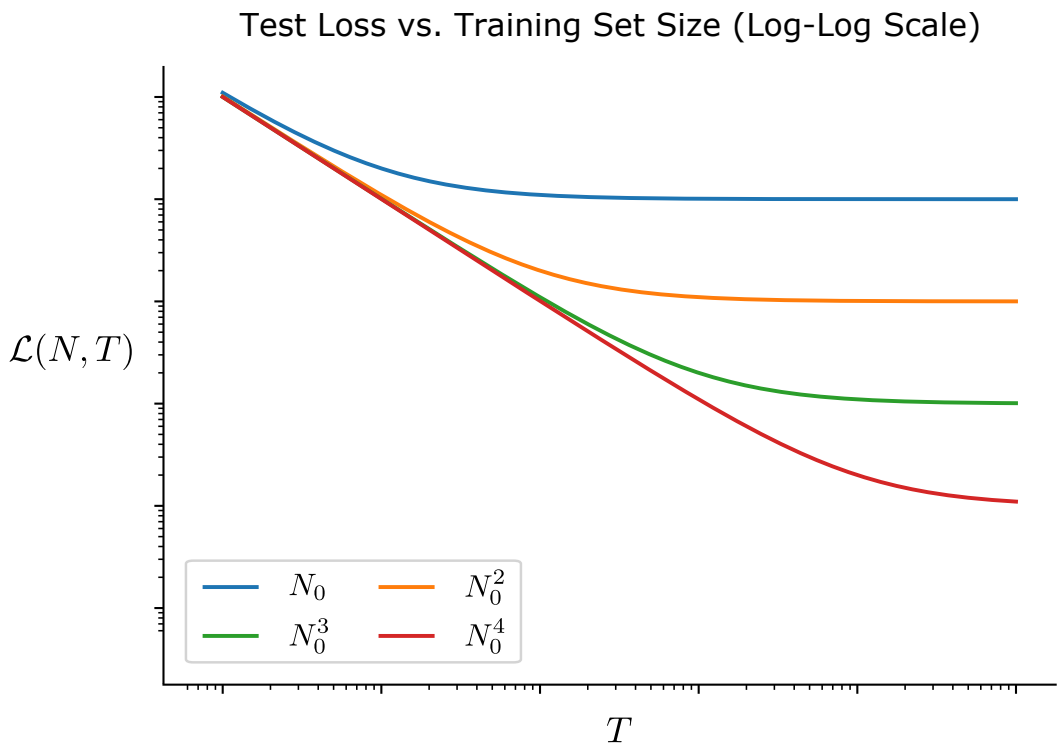
This is not a regime that's typically thought to be useful. . .

LLMs are . . . *Predictable*

Kaplan/McCandlish/OpenAI found empirical scaling laws in the test loss of (autoregressive transformer) LLMs trained *with early stopping* across a large variety of model and dataset sizes.

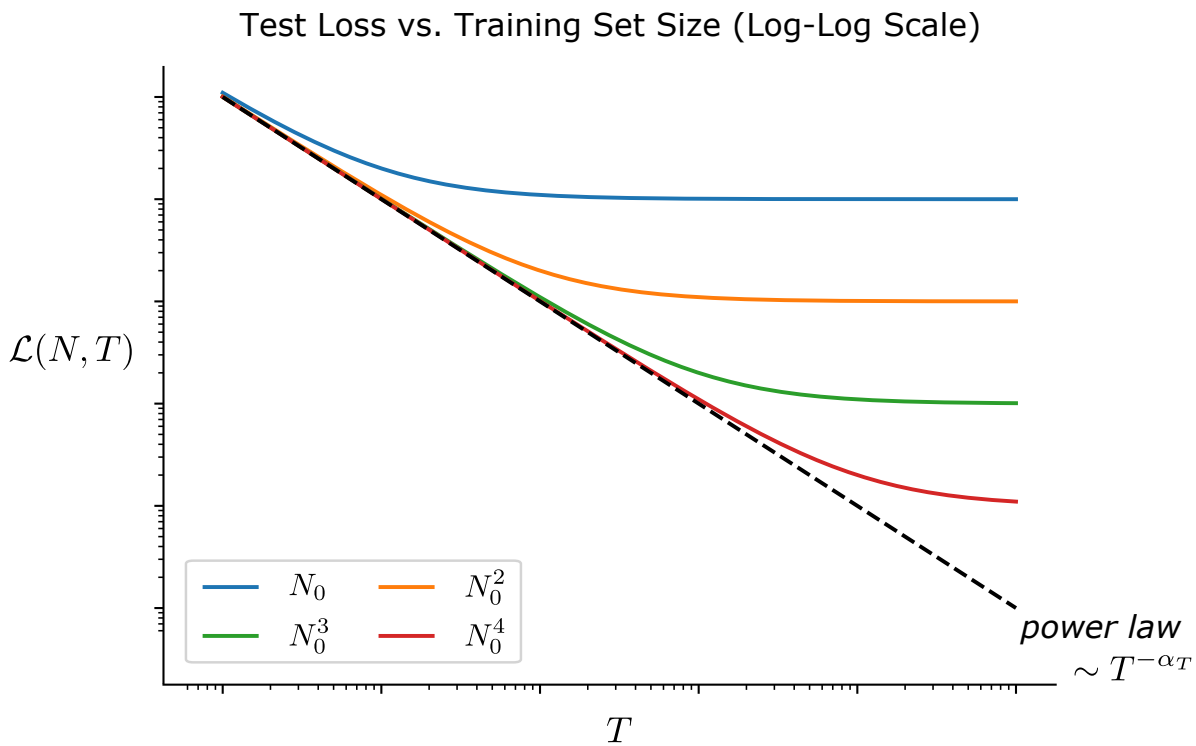
LLMs are . . . *Predictable*

Kaplan/McCandlish/OpenAI found empirical scaling laws in the test loss of (autoregressive transformer) LLMs trained *with early stopping* across a large variety of model and dataset sizes.



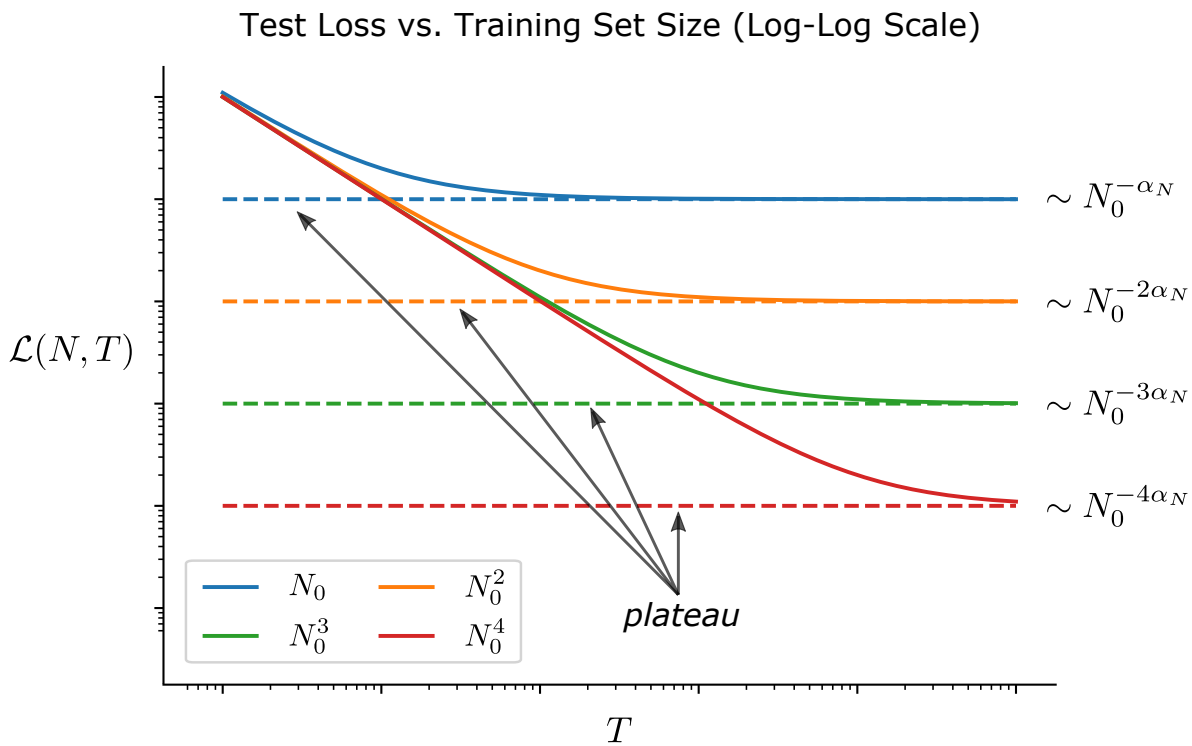
LLMs are . . . *Predictable*

Kaplan/McCandlish/OpenAI found empirical scaling laws in the test loss of (autoregressive transformer) LLMs trained *with early stopping* across a large variety of model and dataset sizes.



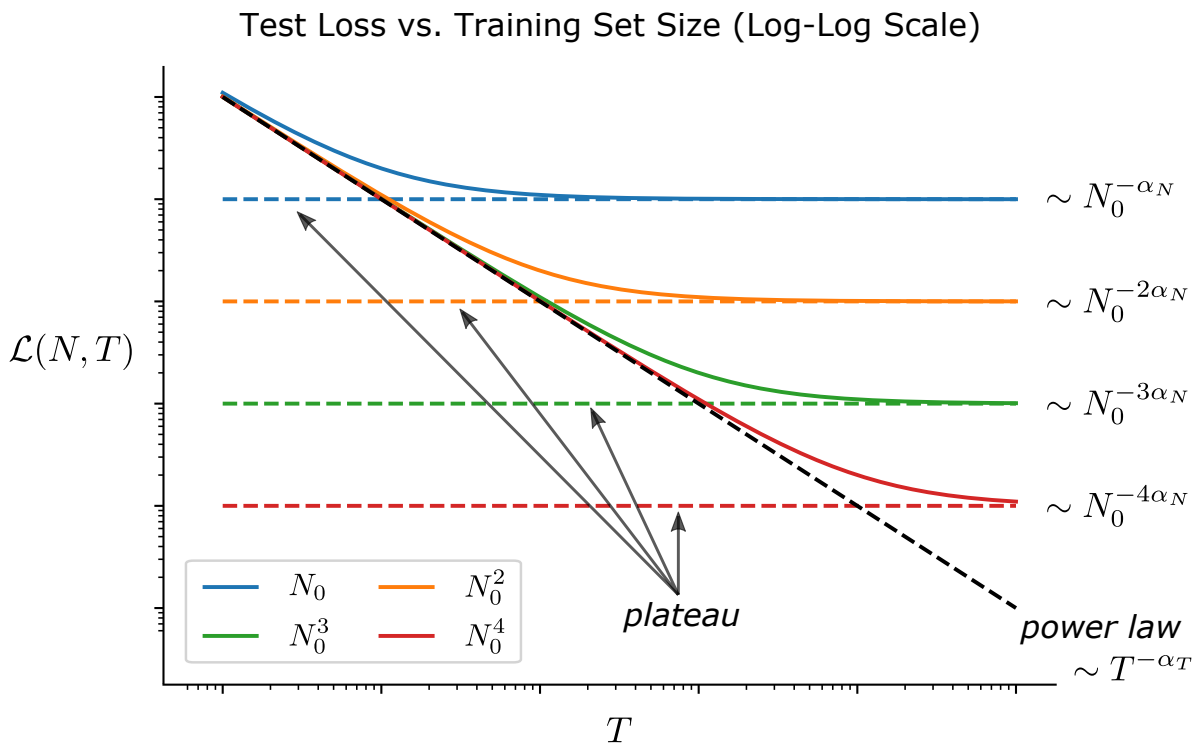
LLMs are . . . *Predictable*

Kaplan/McCandlish/OpenAI found empirical scaling laws in the test loss of (autoregressive transformer) LLMs trained *with early stopping* across a large variety of model and dataset sizes.



LLMs are . . . *Predictable*

Kaplan/McCandlish/OpenAI found empirical scaling laws in the test loss of (autoregressive transformer) LLMs trained *with early stopping* across a large variety of model and dataset sizes.



LLMs are . . . *Predictable*

Kaplan/McCandlish/OpenAI found empirical scaling laws in the test loss of (autoregressive transformer) LLMs trained *with early stopping* across a large variety of model and dataset sizes.

$$\mathcal{L}(N, T) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_T}} + \frac{T_c}{T} \right]^{\alpha_T}$$

LLMs are . . . *Predictable*

Kaplan/McCandlish/OpenAI found empirical scaling laws in the test loss of (autoregressive transformer) LLMs trained *with early stopping* across a large variety of model and dataset sizes.

$$N(T) = N_c \left(\frac{T}{T_c} \right)^{\frac{\alpha_T}{\alpha_N}}$$

LLMs are . . . *Questionable*

This empirical behavior implies many theoretical questions:

LLMs are . . . *Questionable*

This empirical behavior implies many theoretical questions:

- What are the properties of datasets that lead to scaling laws?

LLMs are . . . *Questionable*

This empirical behavior implies many theoretical questions:

- What are the properties of datasets that lead to scaling laws?
- Which DNNs have scaling laws when trained on that data?

LLMs are . . . *Questionable*

This empirical behavior implies many theoretical questions:

- What are the properties of datasets that lead to scaling laws?
- Which DNNs have scaling laws when trained on that data?
- How do they arise; what is the mechanism?

LLMs are . . . *Questionable*

This empirical behavior implies many theoretical questions:

- What are the properties of datasets that lead to scaling laws?
- Which DNNs have scaling laws when trained on that data?
- How do they arise; what is the mechanism?
- Do they break down; what is the behavior when they do?

LLMs are . . . *Here*

Large language models are powerful tools that can be used to accomplish a wide range of tasks. For example, BERT \cite{devlin-etal-2019-bert} was pre-trained on a large corpus and fine-tuned for a wide variety of tasks, including question answering and natural language inference, and achieved state-of-the-art performance. However, large language models usually require a lot of computational resources and training data, which limits their use in many real-world applications.

[GPT-3]

Goals

We want to understand this neural scaling phenomenology:

$$\mathcal{L}(N, T) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_T}} + \frac{T_c}{T} \right]^{\alpha_T} .$$

Goals

We want to understand this neural scaling phenomenology:

$$\mathcal{L}(N, T) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_T}} + \frac{T_c}{T} \right]^{\alpha_T} .$$

(i) Discover the joint properties of datasets and feature maps that lead to this behavior.

Goals

We want to understand this neural scaling phenomenology:

$$\mathcal{L}(N, T) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_T}} + \frac{T_c}{T} \right]^{\alpha_T} .$$

(i) Discover the joint properties of datasets and feature maps that lead to this behavior.

(ii) Find and solve a joint *generative data model* and *random feature model* that has same behavior.

Goals

We want to understand this neural scaling phenomenology:

$$\mathcal{L}(N, T) = \left[\frac{N_c}{N} + \frac{T_c}{T} \right]^{\alpha} .$$

$$(\alpha \equiv \alpha_N = \alpha_T)$$

(i) Discover the joint properties of datasets and feature maps that lead to this behavior.

(ii) Find and solve a joint *generative data model* and *random feature model* that has same behavior.

Goals

We want to understand this neural scaling phenomenology:

$$\mathcal{L}(N, T) = \left[\frac{N_c}{N} + \frac{T_c}{T} \right]^{\alpha} .$$

$$(\alpha \equiv \alpha_N = \alpha_T)$$

(i) Discover the joint properties of datasets and feature maps that lead to this behavior.

(ii) Find and solve a joint *generative data model* and *random feature model* that has same behavior.

(ii) Use the model to study mechanism and breakdown.

Data Properties

AI tasks in different domains use very different underlying data:

Data Properties

AI tasks in different domains use very different underlying data:

- token features of textual data are used LLMs for NLP

Data Properties

AI tasks in different domains use very different underlying data:

- token features of textual data are used LLMs for NLP
- pixel features of image data are used for CV

Data Properties

AI tasks in different domains use very different underlying data:

- token features of textual data are used LLMs for NLP
- pixel features of image data are used for CV

Both domains can exhibit the neural scaling law phenomenology – *Gaussian noise does not(!)* – so we should try to understand the structure in common between these natural datasets.

Data Properties: *notation*

Consider a dataset of T samples with components

$$x_{i;\alpha}\,, \quad \text{with} \quad i = 1, \ldots, N_{\text{in}}\,,$$

where the i indexes the N_{in} different **input features**, which may be a particular pixel or token, and α indexes into the T different **samples** in the dataset.

Data Properties: *notation*

Consider a dataset of T samples with components

$$x_{i;\alpha}\,, \quad \text{with} \quad i = 1, \ldots, N_{\text{in}}\,,$$

where the i indexes the N_{in} different **input features**, which may be a particular pixel or token, and α indexes into the T different **samples** in the dataset.

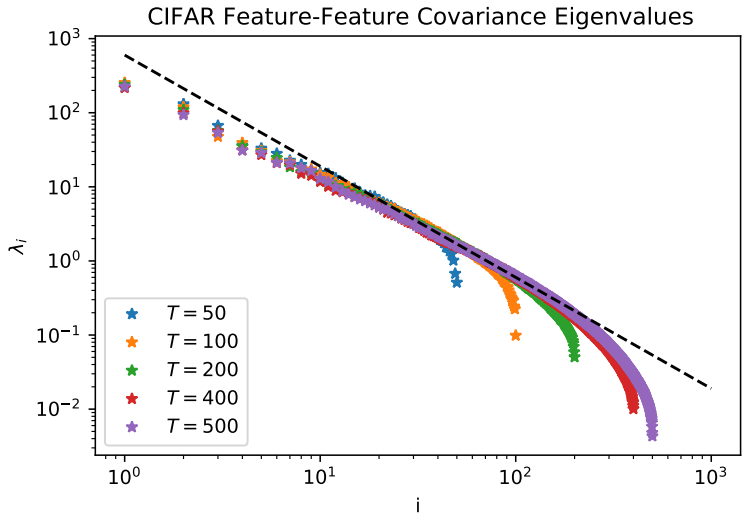
The correlation between input features in the dataset is characterized by the **feature-feature covariance matrix**:

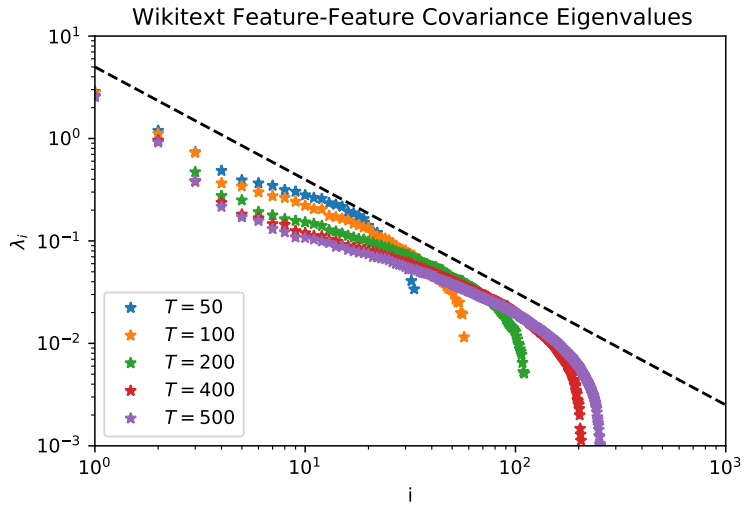
$$\Lambda_{ij} = \frac{1}{T} \sum_{\alpha=1}^{T} x_{i;\alpha} x_{j;\alpha}\,.$$

The **spectrum** of the dataset is the eigenvalues of λ_i of Λ_{ij}.

Data Properties: *spectrum*

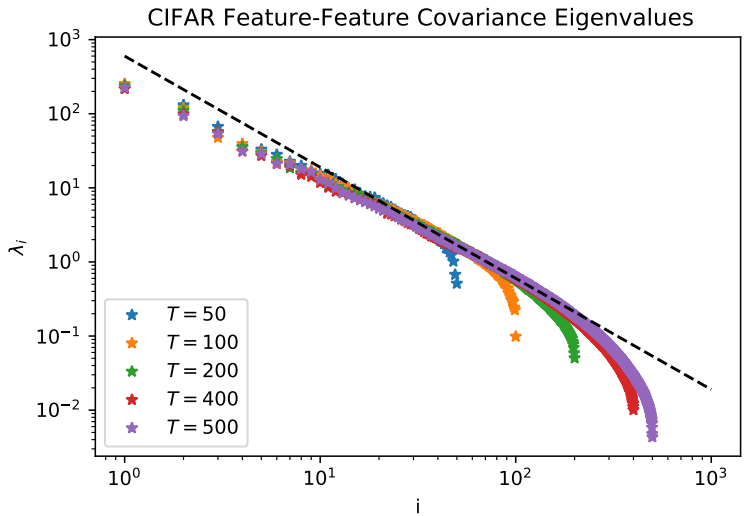
Let's look at the **spectrum**, λ_i, of some real natural datasets.

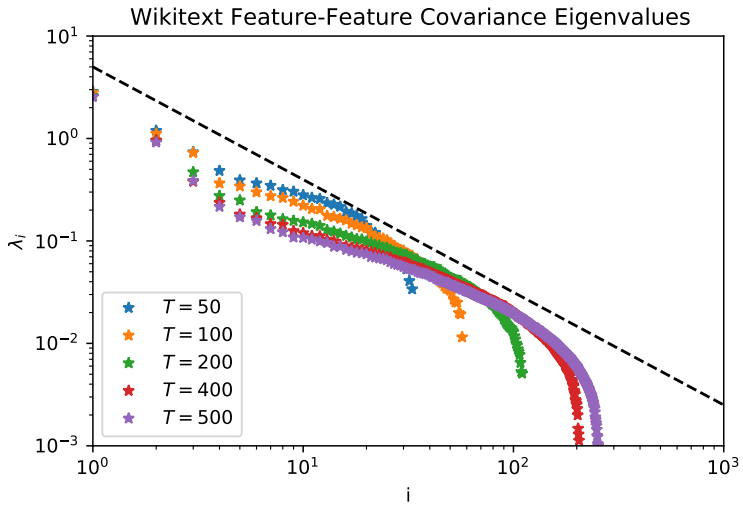




Data Properties: *spectrum*

Let's look at the **spectrum**, λ_i, of some real natural datasets.





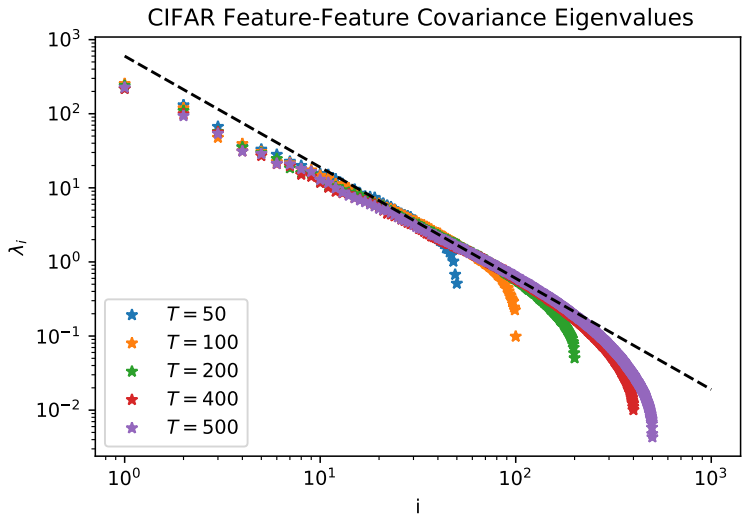
(1) λ_i are well fit by a *power law*:

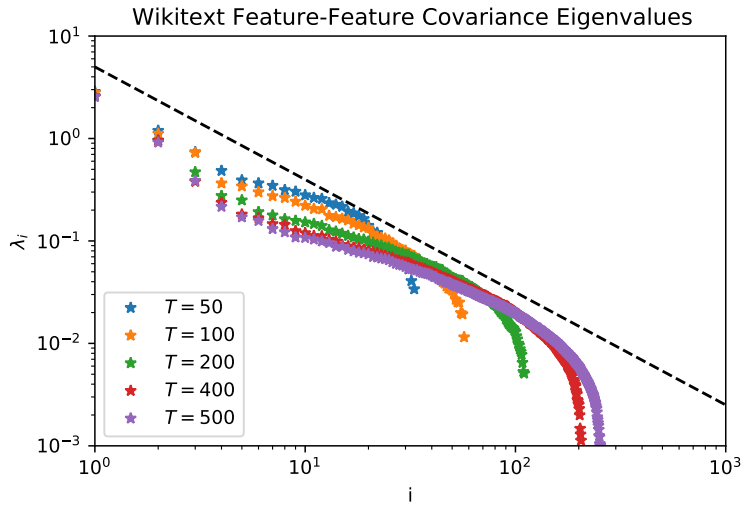
$$\lambda_i \sim \frac{1}{i^{1+\alpha}} \, .$$

[Bahri/Dyer/Kaplan/Lee/Sharma]

Data Properties: *spectrum*

Let's look at the **spectrum**, λ_i, of some real natural datasets.





(1) λ_i are well fit by a *power law*:

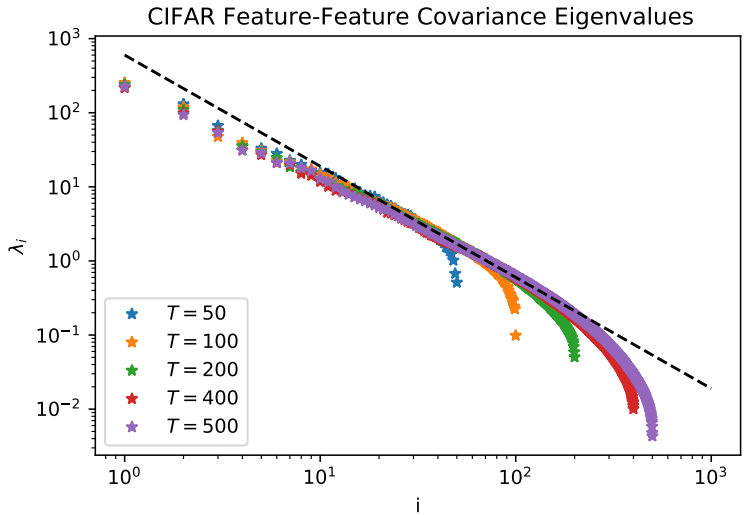
$$\lambda_i \sim \frac{1}{i^{1+\alpha}} .$$

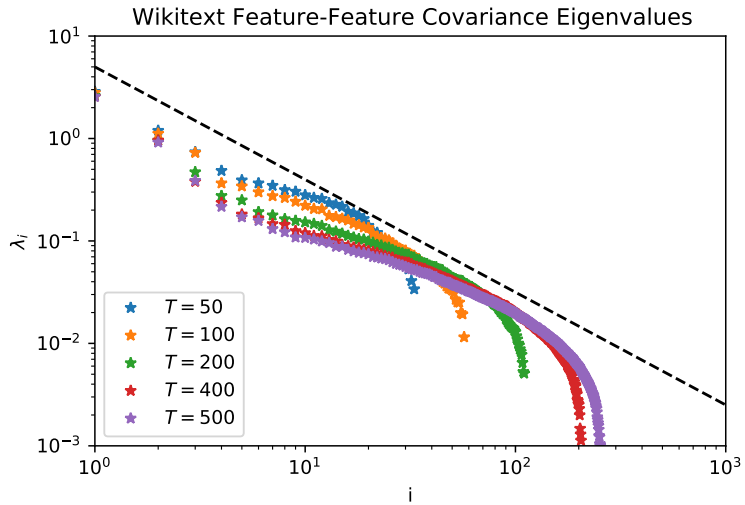
[Bahri/Dyer/Kaplan/Lee/Sharma]

(2) For each T, λ_i terminates in a very rapid decline.

Data Properties: *spectrum*

Let's look at the **spectrum**, λ_i, of some real natural datasets.





(1) λ_i are well fit by a *power law*:

$$\lambda_i \sim \frac{1}{i^{1+\alpha}} .$$

[Bahri/Dyer/Kaplan/Lee/Sharma]

(2) For each T, λ_i terminates in a very rapid decline.

(3) Varying T, we also very the extent of the power law.

Aside: PCA

In PCA, Λ_{ij} is diagonalized to find the linear combinations of the x_i that account for the majority of the variance of the data:

Aside: PCA

In PCA, Λ_{ij} is diagonalized to find the linear combinations of the x_i that account for the majority of the variance of the data:

- If the λ_i has a **gap** at some eigenvalue λ_M, such that M large eigenvalues account for the majority of the total variance, then the other $\lambda_{M+1}, \ldots, \lambda_{N_{\text{in}}}$ eigenvalues are unimportant.

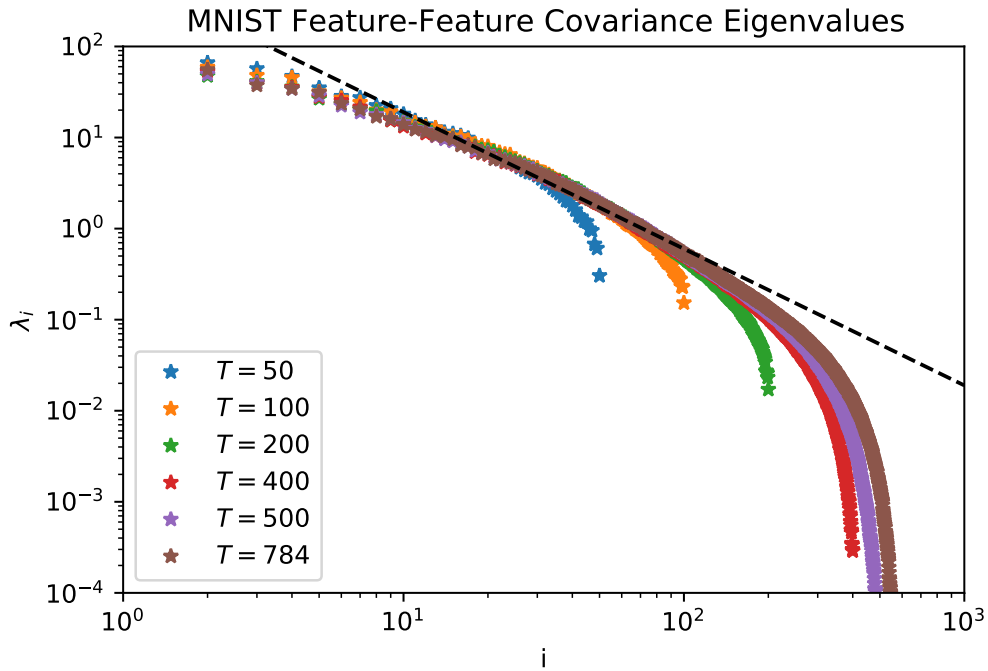
Aside: PCA

In PCA, Λ_{ij} is diagonalized to find the linear combinations of the x_i that account for the majority of the variance of the data:

- If the λ_i has a **gap** at some eigenvalue λ_M, such that M large eigenvalues account for the majority of the total variance, then the other $\lambda_{M+1}, \ldots, \lambda_{N_{\text{in}}}$ eigenvalues are unimportant.
- Contrast with our natural datasets of images and embedded text, which have *continuous* spectra (power law). Perhaps all the eigenvalues are relatively important?

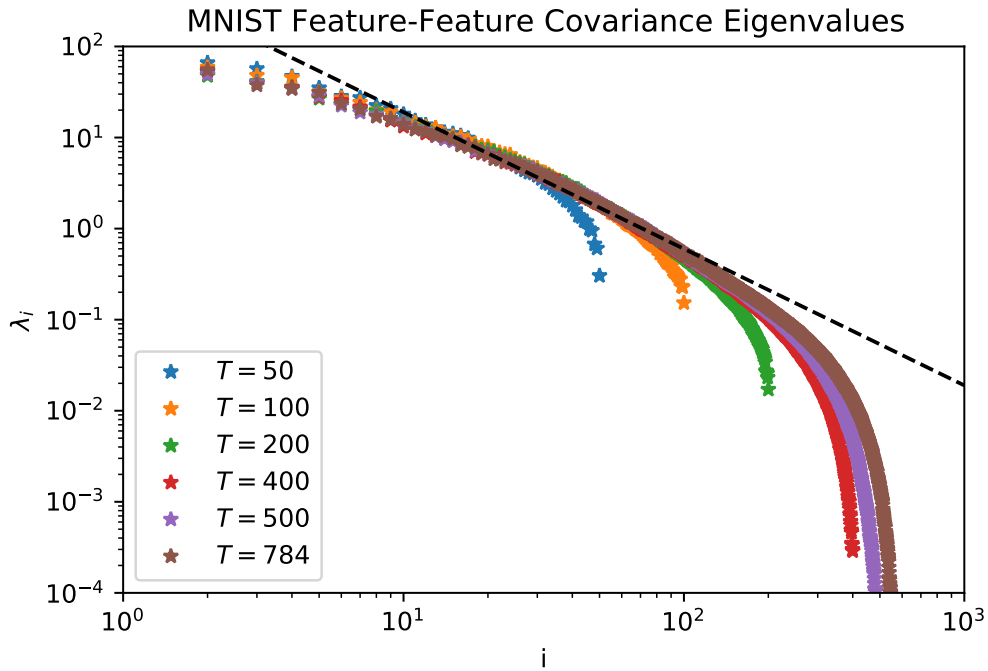
Data Properties: *spectrum*

The extent of a power law in λ_i is bounded by N_{in}.



Data Properties: *spectrum*

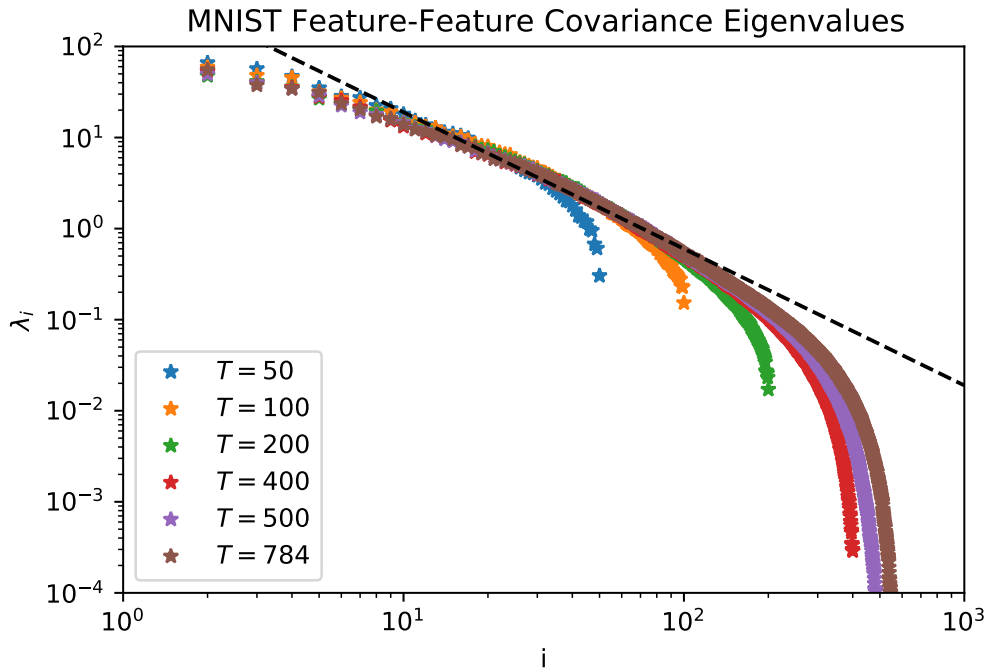
The extent of a power law in λ_i is bounded by N_{in}.



- If the data was generated by $p(x|N_{\text{in}})$, we would expect more information as we increased N_{in}.

Data Properties: *spectrum*

The extent of a power law in λ_i is bounded by N_{in}.



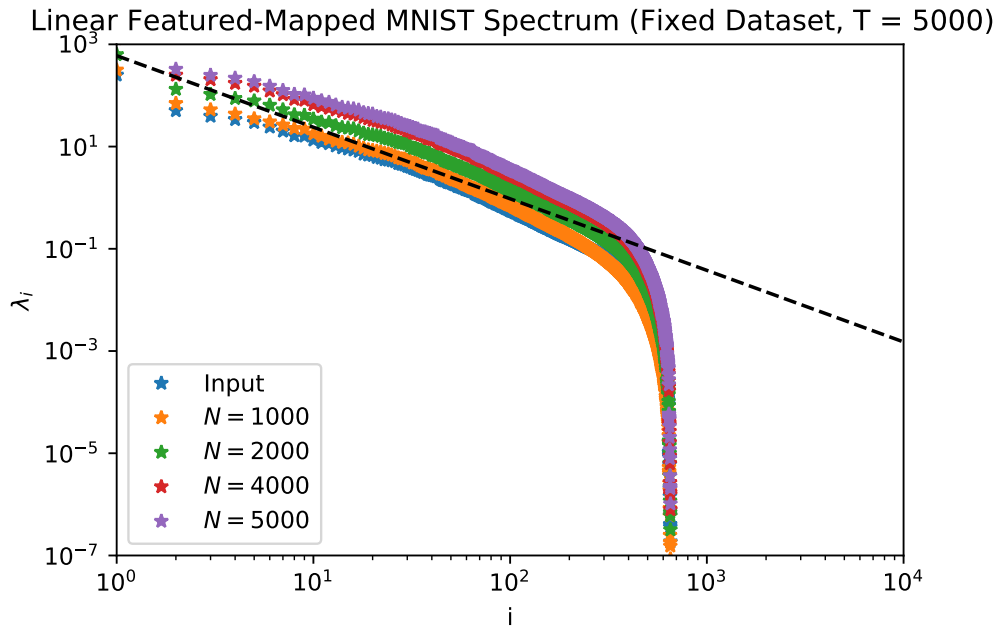
- If the data was generated by $p(x|N_{\text{in}})$, we would expect more information as we increased N_{in}.
- For fixed N_{in}, if increased T, is there additional information in those extra samples?

Feature Maps

What if we try to map to a space N that's *larger* than N_{in}?

Feature Maps

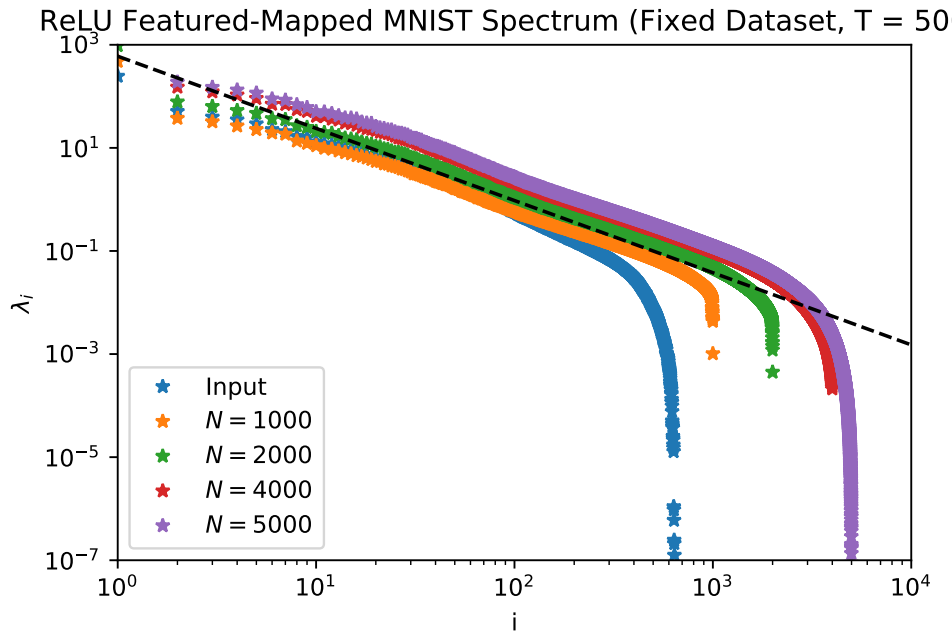
What if we try to map to a space N that's *larger* than N_{in}?



$$\varphi_j \equiv \sum_{k=1}^{N_{\text{in}}} W_{jk} x_k \,, \quad \text{with} \quad j = 1, \ldots, N \,.$$

Feature Maps

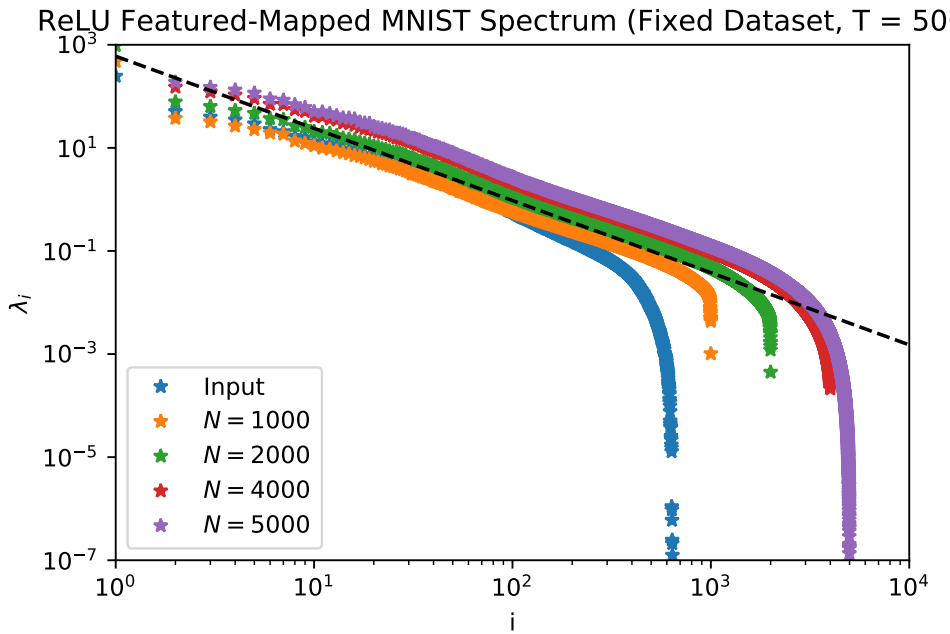
What if we try to map to a space N that's *larger* than N_{in}?



$$\varphi_j \equiv \sigma\left(\sum_{k=1}^{N_{\text{in}}} W_{jk} x_k\right) \,, \quad \text{with} \quad j = 1, \ldots, N \,.$$

Feature Maps

What if we try to map to a space N that's *larger* than N_{in}?



$$\varphi_j \equiv \sigma\left(\sum_{k=1}^{N_{\text{in}}} W_{jk} x_k\right) \,, \quad \text{with} \quad j = 1, \ldots, N \,.$$

DNN *extends* power law, samples for $T > N_{\text{in}}$ are useful!

A Statistical Model

Want a joint *generative data model* and *random feature model* that captures the broad empirical properties of these real datasets and the effect of the ReLU layer (our stand-in for more general DNNs).

A Statistical Model: *Data*

Rather than generating data in the raw input space, we will generate data in a *latent space*:

$$x_J, \quad \text{with} \quad J = 1, \ldots, M\,,$$

where J indexes the latent space features.

A Statistical Model: *Data*

Rather than generating data in the raw input space, we will generate data in a *latent space*:

$$x_J, \quad \text{with} \quad J = 1, \ldots, M\,,$$

where J indexes the latent space features.

Latent data are drawn from a zero-mean Gaussian distribution with latent features having a power-law covariance:

$$\langle x_J x_K \rangle = \delta_{JK} \lambda_J\,, \quad \lambda_J \equiv \lambda_+ \left(\frac{1}{J}\right)^{1+\alpha}\,.$$

A Statistical Model: *Data*

Rather than generating data in the raw input space, we will generate data in a *latent space*:

$$x_J, \quad \text{with} \quad J = 1, \ldots, M\,,$$

where J indexes the latent space features.

Latent data are drawn from a zero-mean Gaussian distribution with latent features having a power-law covariance:

$$\langle x_J x_K \rangle = \delta_{JK} \lambda_J\,, \quad \lambda_J \equiv \lambda_+ \left(\frac{1}{J}\right)^{1+\alpha}\,.$$

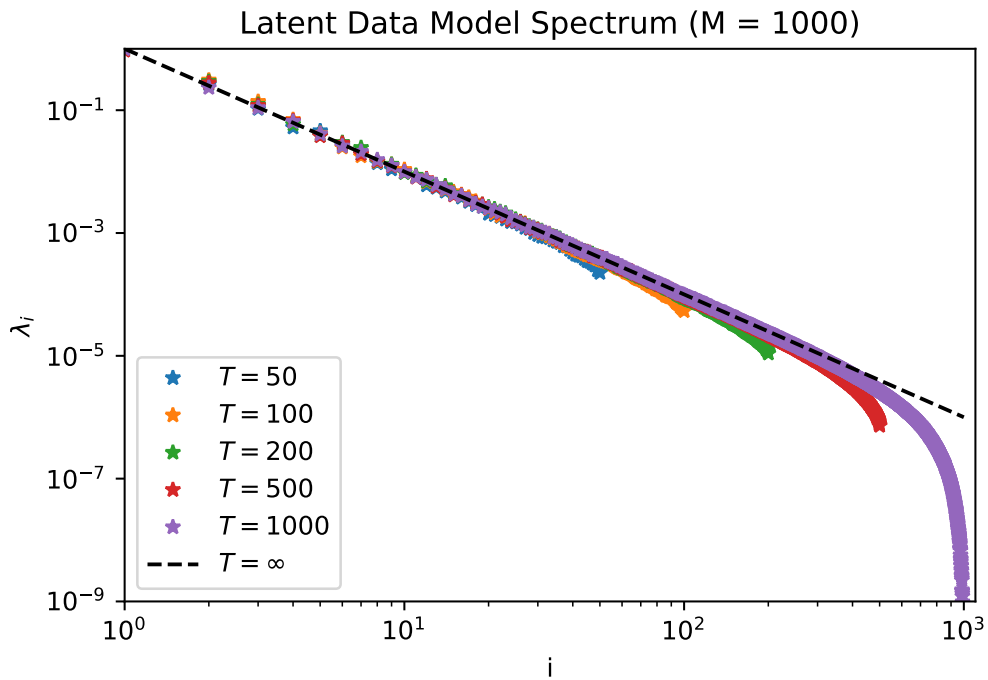
For every latent sample x_J, we will generate a teacher label:

$$y = \sum_{J=1}^{M} w_J x_J + \epsilon,$$

with w sampled from a zero-mean Gaussian and ϵ per sample noise.

A Statistical Model: *Data*

For a finite dataset of size T, the spectrum of latent data will be similar to what we observed empirically for natural data:

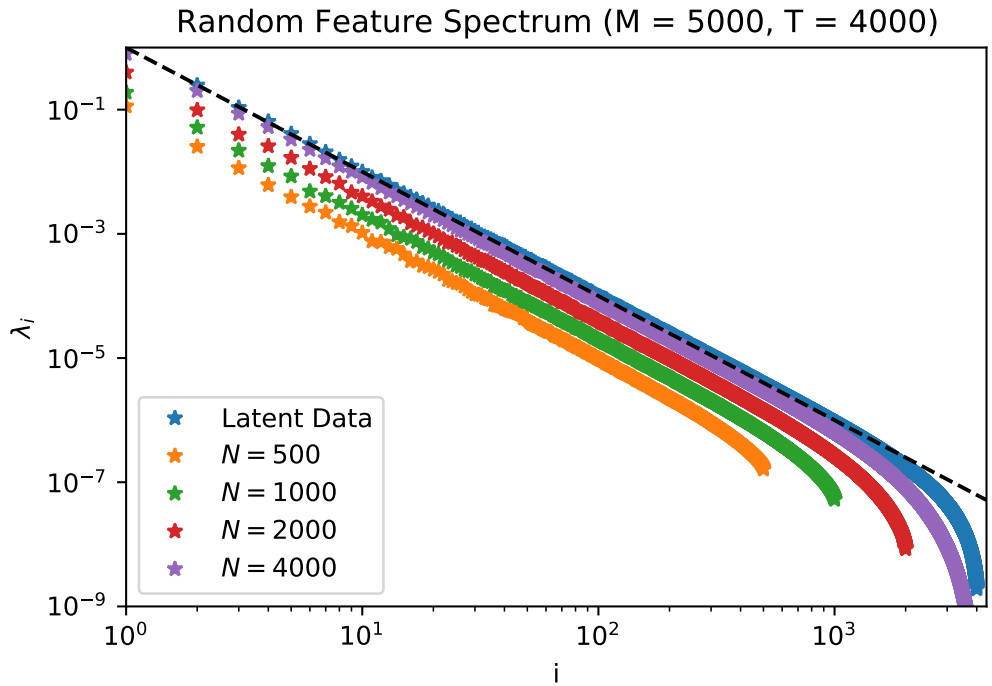


A Statistical Model: *Features*

What if we map T samples to a space N that's *smaller* than M?

A Statistical Model: *Features*

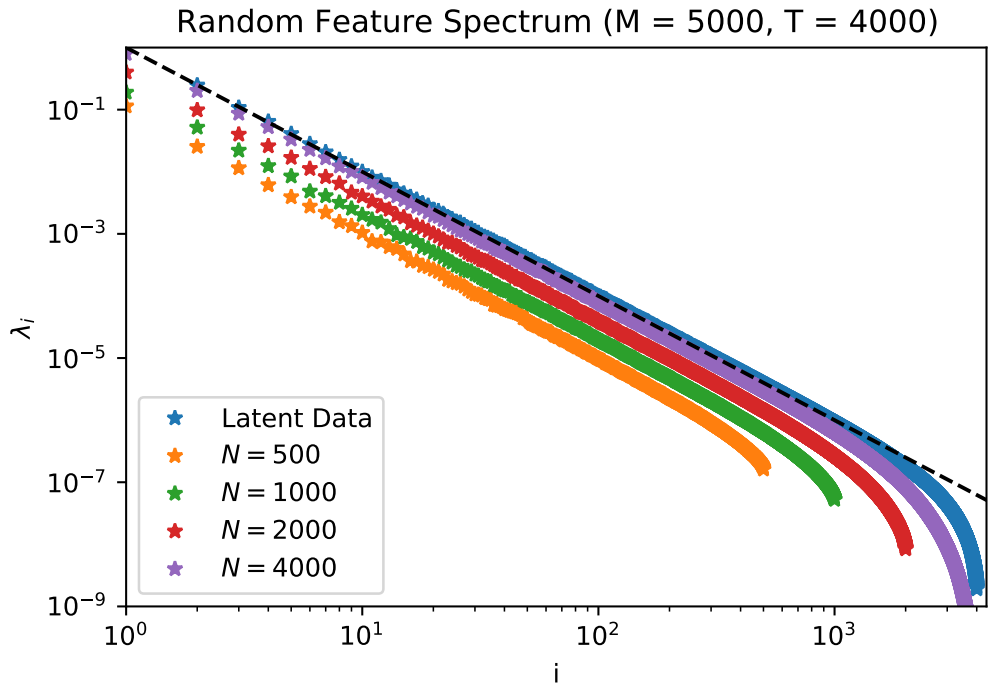
What if we map T samples to a space N that's *smaller* than M?



$$\varphi_i(x_J) \equiv \sum_{J=1}^{M} u_{iJ} x_J \,, \quad \text{with} \quad J = 1, \ldots, M \,.$$

A Statistical Model: *Features*

What if we map T samples to a space N that's *smaller* than M?

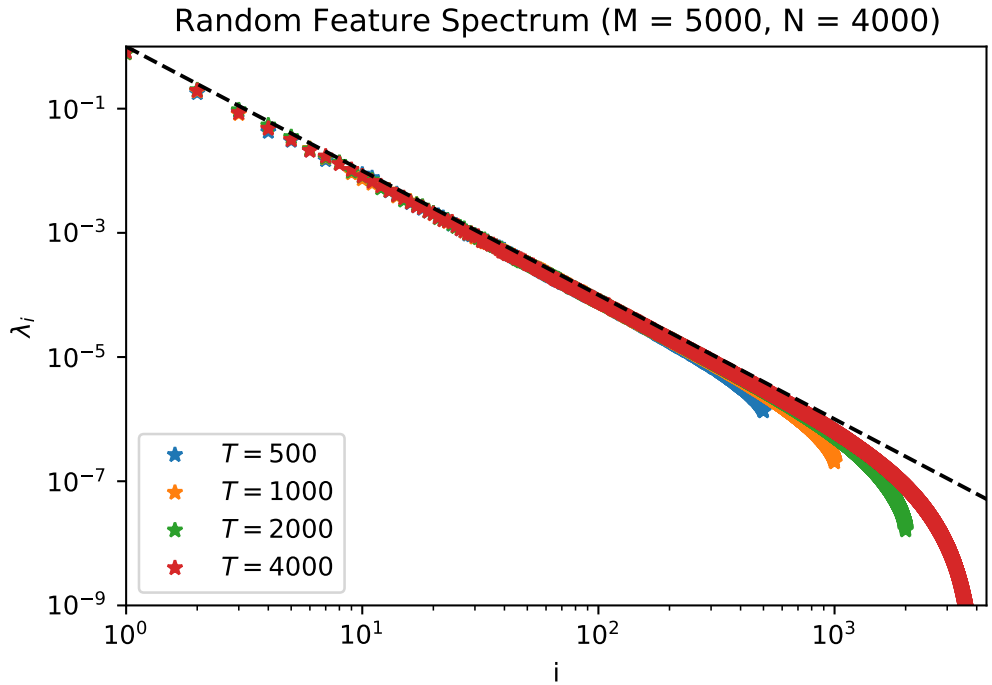


$$\varphi_i(x_J) \equiv \sum_{J=1}^{M} u_{iJ} x_J \,, \quad \text{with} \quad J = 1, \ldots, M \,.$$

Linear map controls extent of power law, giving N random features.

A Statistical Model: *Features*

What if we map T samples to a space N that's *smaller* than M?



$$\varphi_i(x_J) \equiv \sum_{J=1}^{M} u_{iJ} x_J \,, \quad \text{with} \quad J = 1, \ldots, M \,.$$

By varying either of N and T, we can control the extent.

A Statistical Model: *(Generalized) Linear Model*

We "train" a generalized linear model to reproduce the teacher labels (generated from the underlying latent features) using a linear transformation of only the random features (see also [Bahri/Dyer/Kaplan/Lee/Sharma]):

$$z = \sum_{j=1}^{N} \theta_i \varphi_i(x_J) \,.$$

A Statistical Model: *(Generalized) Linear Model*

We "train" a generalized linear model to reproduce the teacher labels (generated from the underlying latent features) using a linear transformation of only the random features (see also [Bahri/Dyer/Kaplan/Lee/Sharma]):

$$z = \sum_{j=1}^{N} \theta_i \varphi_i(x_J) \, .$$

We minimize a standard MSE loss with a ridge parameter:

$$\mathcal{L}_{\mathcal{A}}(\theta) = \frac{1}{2} \left\| \theta\varphi - y + \epsilon \right\|^2 + \frac{\gamma}{2} \left\| \theta \right\|^2 \, .$$

A Statistical Model: *(Generalized) Linear Model*

We "train" a generalized linear model to reproduce the teacher labels (generated from the underlying latent features) using a linear transformation of only the random features (see also [Bahri/Dyer/Kaplan/Lee/Sharma]):

$$z = \sum_{j=1}^{N} \theta_i \varphi_i(x_J) \,.$$

We minimize a standard MSE loss with a ridge parameter:

$$\mathcal{L}_{\mathcal{A}}(\theta) = \frac{1}{2} \left\| \theta\varphi - y + \epsilon \right\|^2 + \frac{\gamma}{2} \left\| \theta \right\|^2 \,.$$

This has a well known solution:

$$\theta^\star \equiv (y + \epsilon)\varphi^T q \,, \qquad q \equiv q(\gamma) = \frac{1}{\varphi\varphi^T + \gamma I_N} \,.$$

A Statistical Model: *Test Loss*

Sample a test set of $\widehat{T}$ samples, denoted by matrices $\{\widehat{x}, \widehat{y}\}$. The test loss is evaluated on our regression solution, $\widehat{z}^{\star} \equiv \theta^{\star} \cdot \varphi(\widehat{x})$:

$$\begin{aligned}\mathcal{L}_{\mathcal{B}}(\theta^{\star}) &= \frac{1}{2\widehat{\widehat{T}}} \left\| \widehat{z}^{\star} - \widehat{y} \right\|^2 \\ &= \frac{1}{2\widehat{\widehat{T}}} \left\| (y + \epsilon) \varphi^{T} q \widehat{\varphi} - \widehat{y} \right\|^2 .\end{aligned}$$

A Statistical Model: *Test Loss*

Sample a test set of $\widehat{T}$ samples, denoted by matrices $\{\widehat{x}, \widehat{y}\}$. The test loss is evaluated on our regression solution, $\widehat{z}^\star \equiv \theta^\star \cdot \varphi(\widehat{x})$:

$$\begin{aligned}\mathcal{L}_{\mathcal{B}}(\theta^\star) &= \frac{1}{2\widehat{T}} \left\|\widehat{z}^\star - \widehat{y}\right\|^2 \\ &= \frac{1}{2\widehat{T}} \left\|(y+\epsilon)\varphi^T q\widehat{\varphi} - \widehat{y}\right\|^2 .\end{aligned}$$

The goal of analysis is to compute the average:

$$\left\langle \mathcal{L}_{\mathcal{B}}(\theta^\star)\right\rangle_{\epsilon,w,\varphi(x),x} .$$

A Statistical Model: *Test Loss*

Sample a test set of $\widehat{T}$ samples, denoted by matrices $\{\widehat{x}, \widehat{y}\}$. The test loss is evaluated on our regression solution, $\widehat{z}^\star \equiv \theta^\star \cdot \varphi(\widehat{x})$:

$$\begin{aligned}\mathcal{L}_{\mathcal{B}}(\theta^\star) &= \frac{1}{2\widehat{\widehat{T}}} \left\|\widehat{z}^\star - \widehat{y}\right\|^2 \\ &= \frac{1}{2\widehat{\widehat{T}}} \left\|(y+\epsilon)\varphi^T q\widehat{\varphi} - \widehat{y}\right\|^2 .\end{aligned}$$

The goal of analysis is to compute the average:

$$\left\langle \mathcal{L}_{\mathcal{B}}(\theta^\star)\right\rangle_{\epsilon,w,\varphi(x),x} .$$

Some of which are easy:

$$\left\langle \mathcal{L}_{\mathcal{B}}(\theta^\star)\right\rangle_{\epsilon,w} = \frac{\sigma_w^2}{2\widehat{T}M}\left\|x\varphi^T q\widehat{\varphi} - \widehat{x}\right\|^2 + \frac{\sigma_\epsilon^2}{2\widehat{\widehat{T}}}\left\|\varphi^T q\widehat{\varphi}\right\|^2 .$$

A Statistical Model: *Test Loss*

Sample a test set of $\widehat{T}$ samples, denoted by matrices $\{\widehat{x}, \widehat{y}\}$. The test loss is evaluated on our regression solution, $\widehat{z}^\star \equiv \theta^\star \cdot \varphi(\widehat{x})$:

$$\begin{aligned}\mathcal{L}_{\mathcal{B}}(\theta^\star) &= \frac{1}{2\widehat{T}} \left\|\widehat{z}^\star - \widehat{y}\right\|^2 \\ &= \frac{1}{2\widehat{T}} \left\|(y+\epsilon)\varphi^T q\widehat{\varphi} - \widehat{y}\right\|^2 .\end{aligned}$$

The goal of analysis is to compute the average:

$$\left\langle \mathcal{L}_{\mathcal{B}}(\theta^\star)\right\rangle_{\epsilon,w,\varphi(x),x} .$$

Some of which are easy:

$$\left\langle \mathcal{L}_{\mathcal{B}}(\theta^\star)\right\rangle_{\epsilon,w} = \frac{\sigma_w^2}{2\widehat{T}M} \left\|x\varphi^T q\widehat{\varphi} - \widehat{x}\right\|^2 + \frac{\sigma_\epsilon^2}{2\widehat{T}} \left\|\varphi^T q\widehat{\varphi}\right\|^2 .$$

And the remaining ones are not...

[Louart/Liao/Couillet]

A Statistical Model: *Empirics*

One advantage of a joint model of data *and* features is that we can just simulate to see what happens:

A Statistical Model: *Empirics*

One advantage of a joint model of data *and* features is that we can just simulate to see what happens:

(We optimize over the ridge parameter $\gamma^\star$.)

A Statistical Model: *Empirics*

One advantage of a joint model of data *and* features is that we can just simulate to see what happens:

Can be fit with $\mathcal{L}(N, T) = k\left(\frac{1}{N} + \frac{1}{T}\right)^{\alpha}$.

A Statistical Model: *Empirics*

One advantage of a joint model of data *and* features is that we can just simulate to see what happens:

The $\gamma \to 0$ limit is what we are able to compute analytically.

A Statistical Model: *Empirics*

One advantage of a joint model of data *and* features is that we can just simulate to see what happens:

$$\mathcal{L}(N,T) \sim \begin{cases} \frac{1}{1-N/T}\left(\frac{1}{N}-\frac{1}{M}\right)^{\alpha}, & N < T \\ \frac{1}{1-T/N}\left(\frac{1}{T}-\frac{1}{M}\right)^{\alpha}, & N > T\,. \end{cases}$$

Breakdown of Scaling Laws

Breakdown of Scaling Laws

- Does this phenomenological model stop being predictive?

$$\mathcal{L}(N, T) = k\left(\frac{1}{N} + \frac{1}{T}\right)^{\alpha}$$

Breakdown of Scaling Laws

- Does this phenomenological model stop being predictive?

$$\mathcal{L}(N, T) = k\left(\frac{1}{N} + \frac{1}{T}\right)^{\alpha}$$

- What is behavior in the new regime?

Breakdown of Scaling Laws

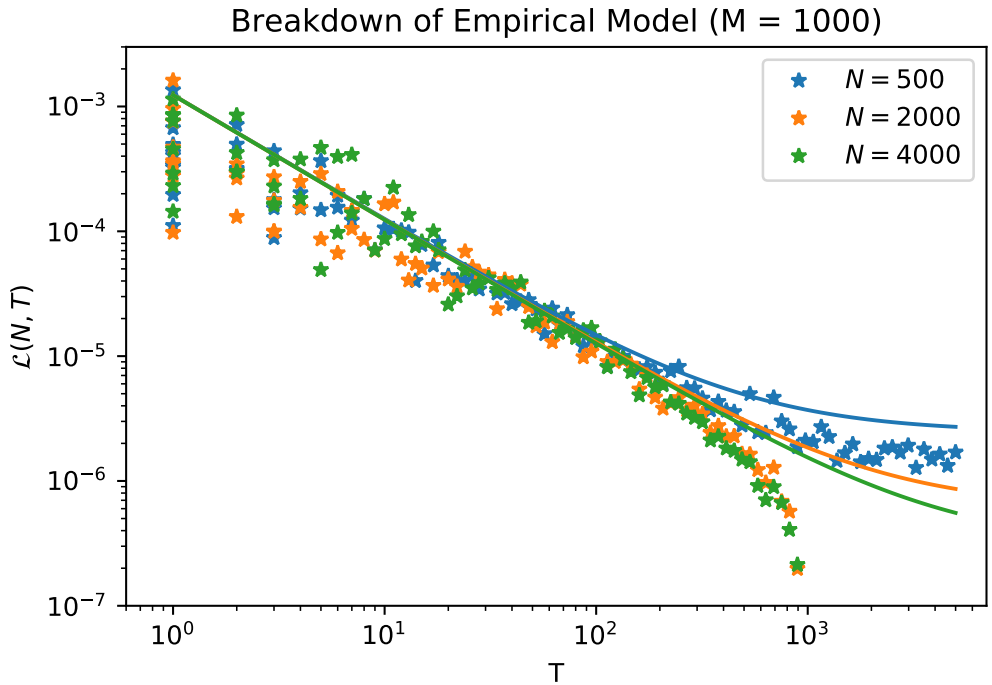
- Does this phenomenological model stop being predictive?

$$\mathcal{L}(N, T) = k\left(\frac{1}{N} + \frac{1}{T}\right)^{\alpha}$$

- What is behavior in the new regime?

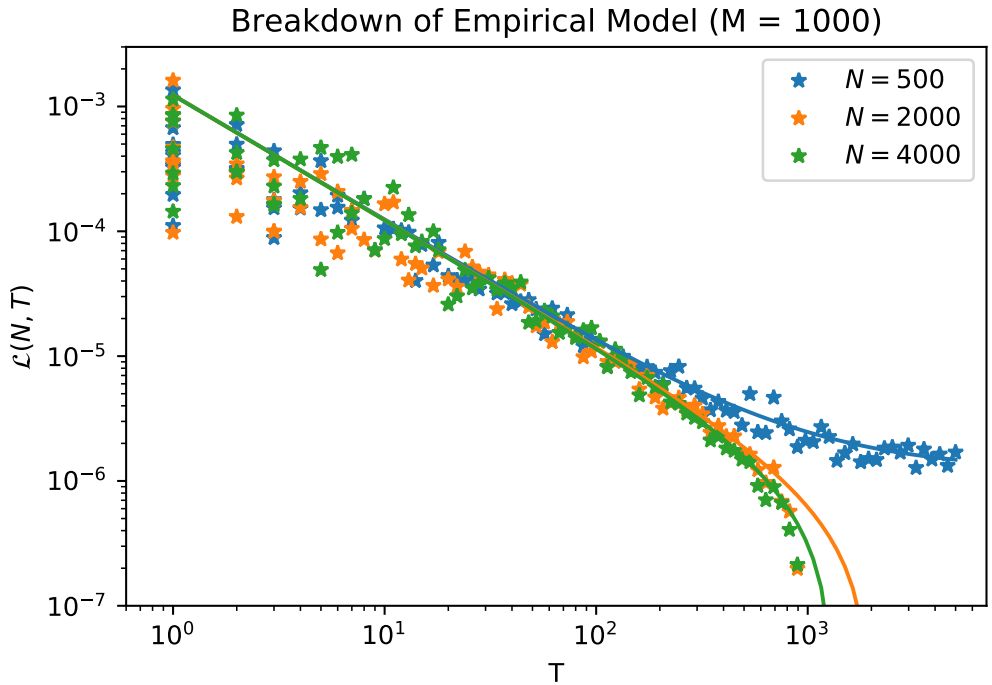
Hint: it only depends on 2 of the 3 scales in the problem...

Breakdown of Scaling Laws: $M \lesssim N$



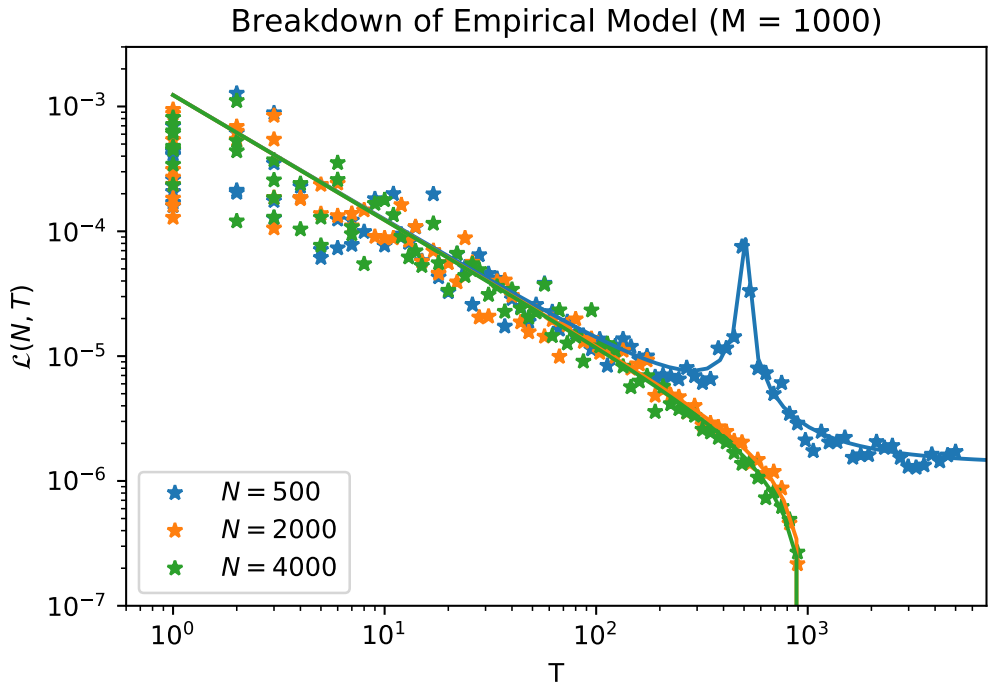
$\mathcal{L}(N, T) = k\left(\frac{1}{N} + \frac{1}{T}\right)^{\alpha}$ breaks down when $M \lesssim N$.

Breakdown of Scaling Laws: $M \lesssim N$



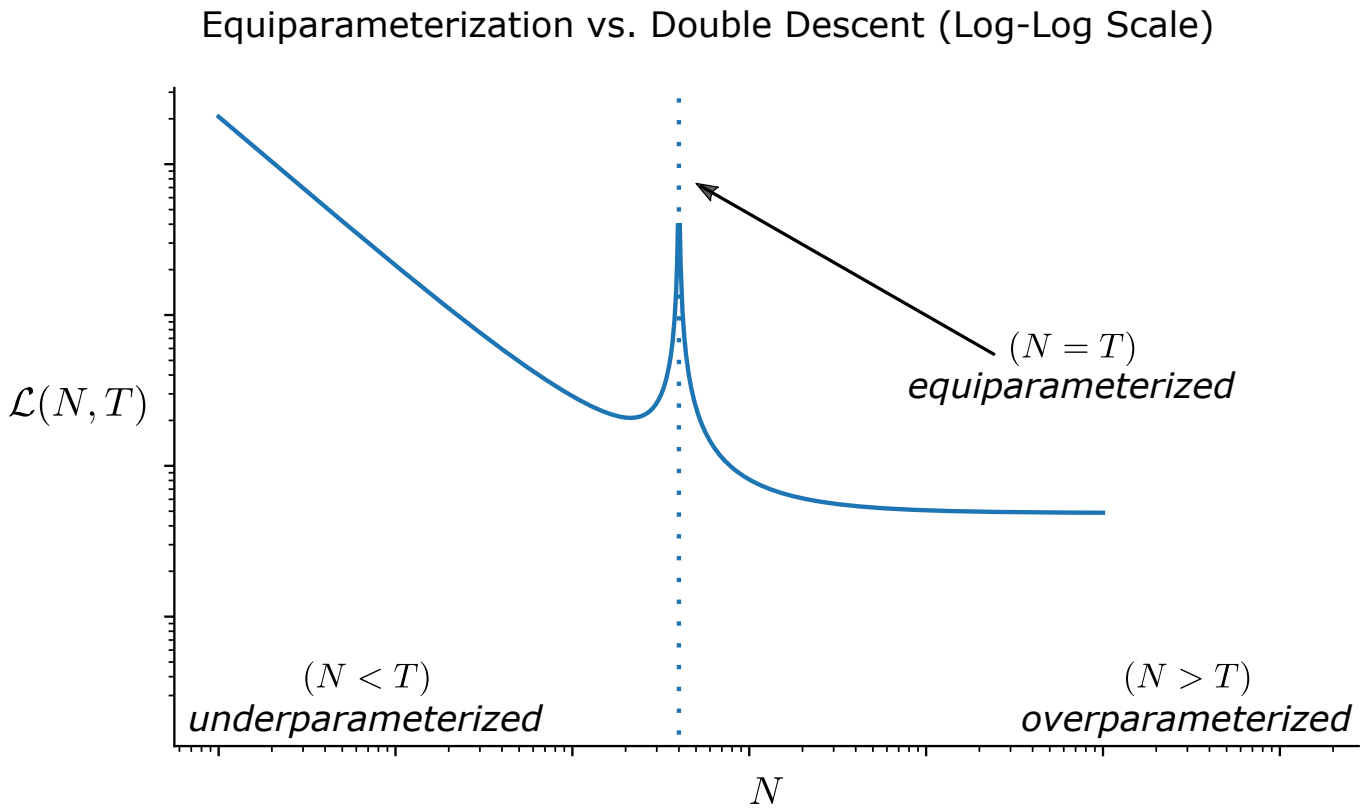
$\mathcal{L}(N, T) = k\left[\left(\frac{1}{N} + \frac{1}{T}\right)^{\alpha} - \left(\frac{1}{M}\right)^{\alpha}\right]$ fits well.

Breakdown of Scaling Laws: $M \lesssim N$



$$\mathcal{L}(N,T) \sim \begin{cases} \frac{1}{1-T/N}\left(\frac{1}{T}-\frac{1}{M}\right)^{\alpha}, & T<N, T<M \\ \frac{1}{1-N/T}\left(\frac{1}{N}-\frac{1}{M}\right)^{\alpha}, & T>N, N<M. \end{cases}$$

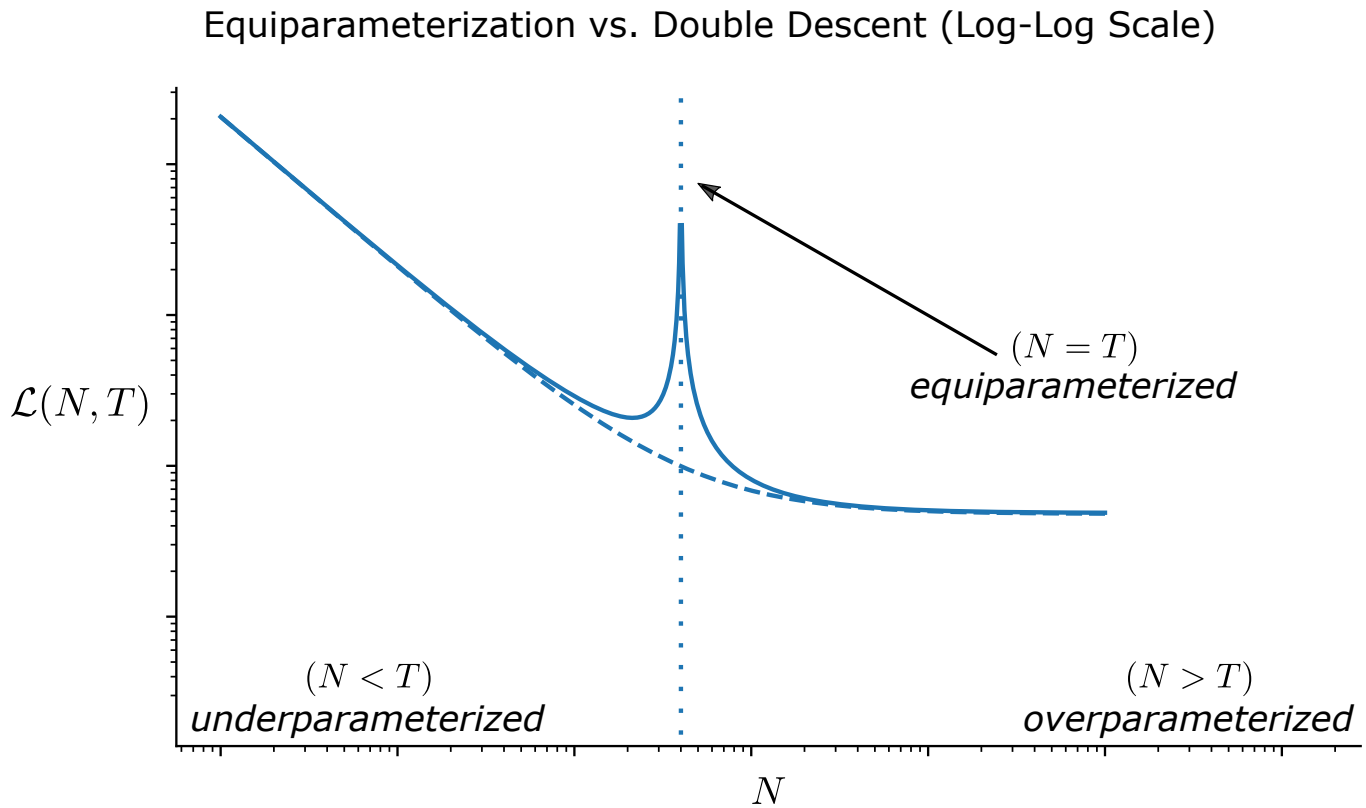
The Equiparameterization Regime



The *unregularized* test loss curve shows the **double descent** phenomenon in the overparameterized regime.

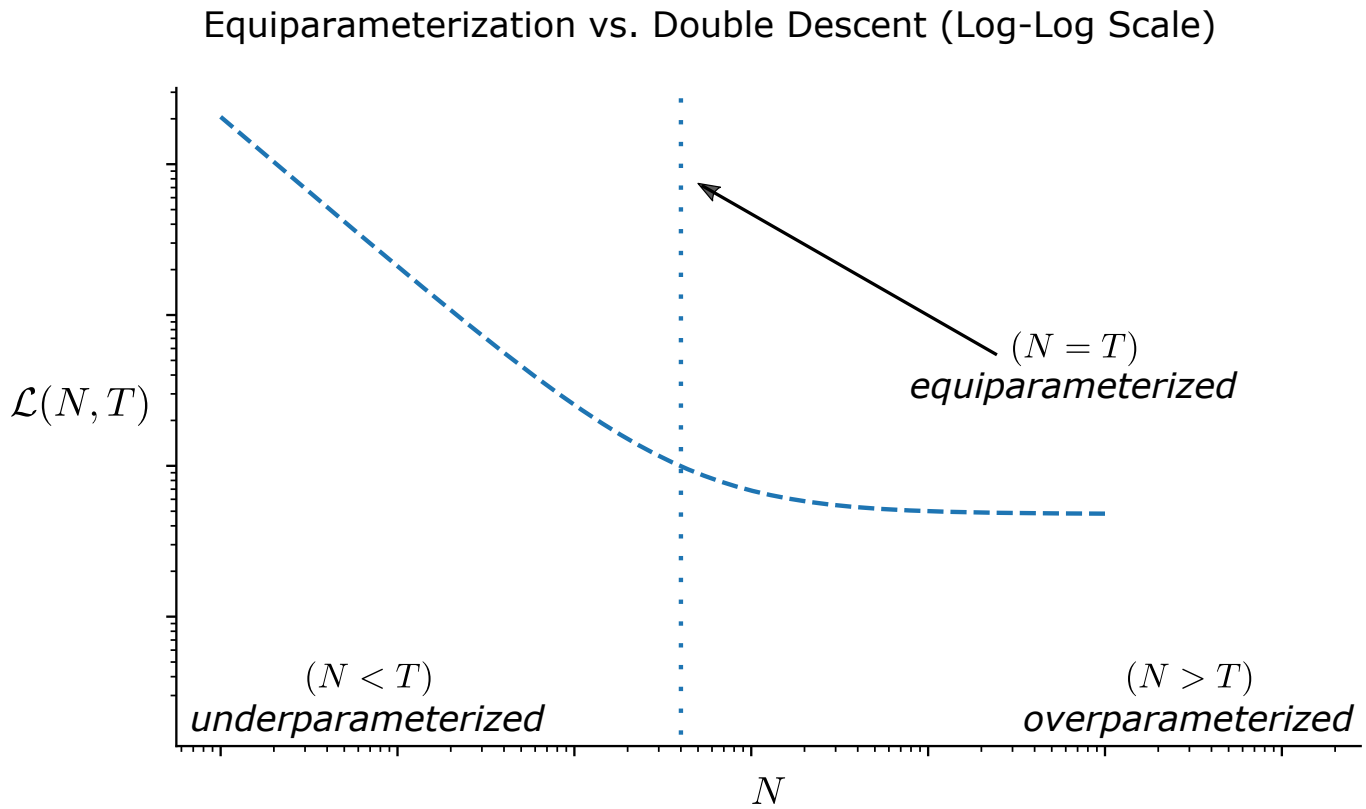
[Belkin/Hsu/Ma/Mandal]

The Equiparameterization Regime



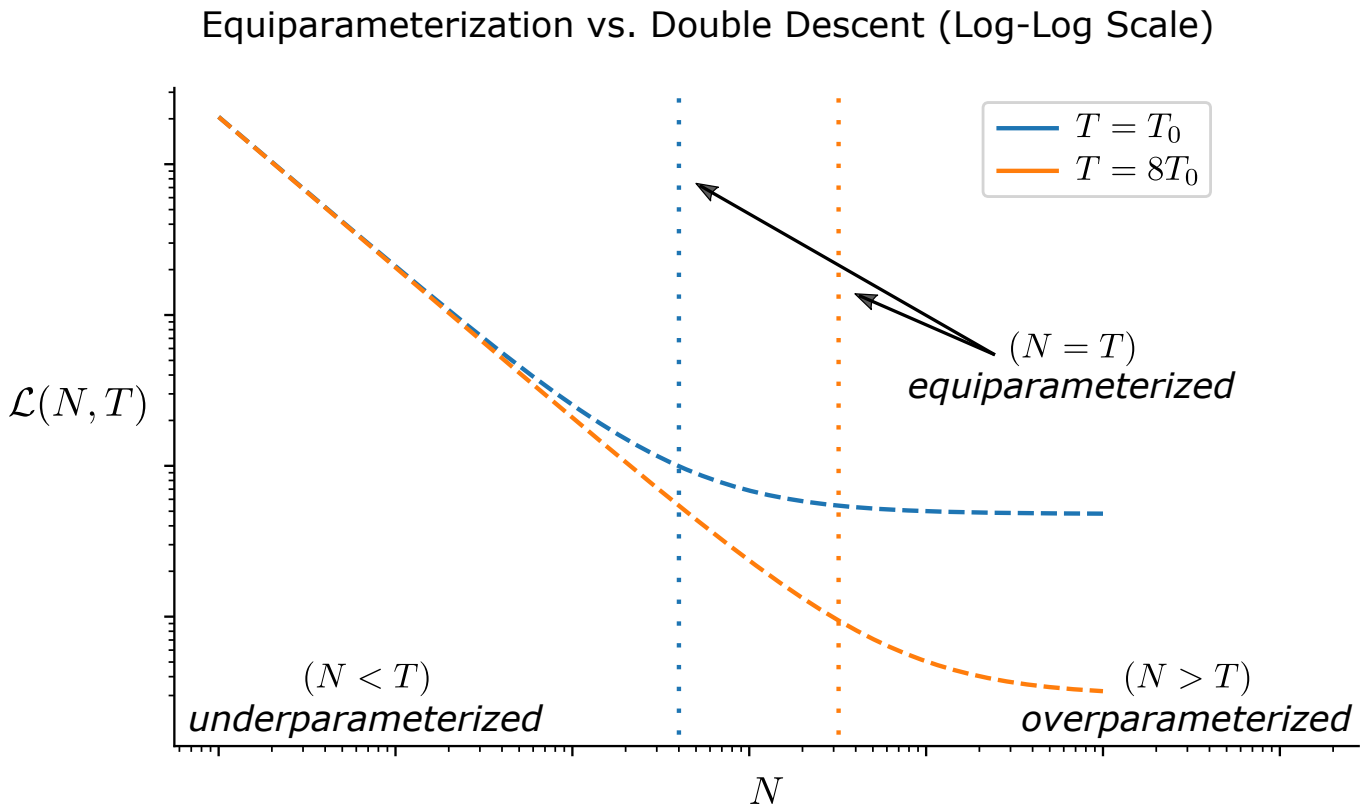
The non-analytic peak is an artifact and can be eliminated by regularization, e.g. early-stopping or a ridge parameter $\gamma^\star$.

The Equiparameterization Regime



Performance increases with increase the number of parameters. . .

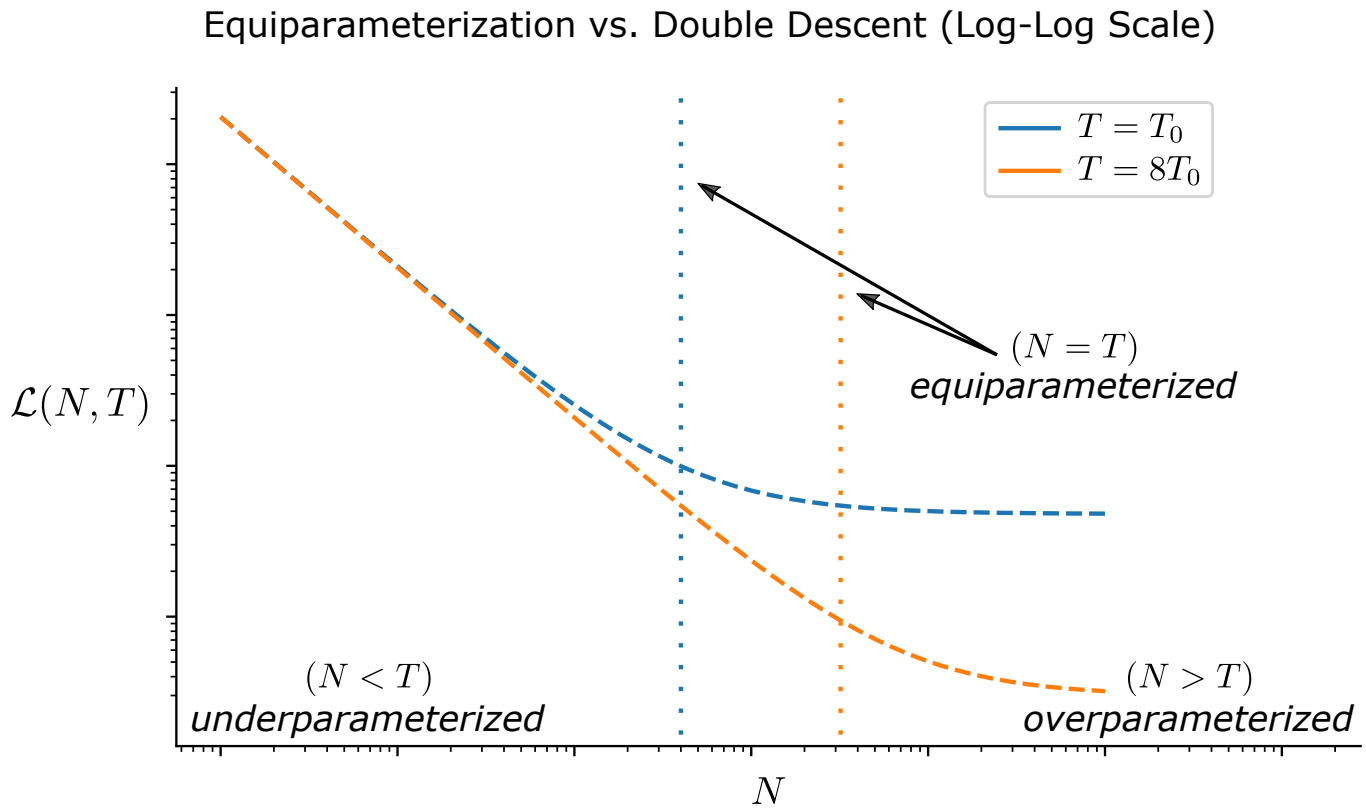
The Equiparameterization Regime



. . . but increases much more by scaling the parameters and training set size together: $N \sim T$.

[Kaplan/McCandlish/OpenAI, Hoffmann/Borgeaud/Mensch/DeepMind/Sifre]

The Equiparameterization Regime



. . . but increases much more by scaling the parameters and training set size together: $N \sim T$.

[Kaplan/McCandlish/OpenAI, Hoffmann/Borgeaud/Mensch/DeepMind/Sifre]

- Because of the power-law structure with $N, T < M$.

Latent Dimensions: *A Puzzle*

A Puzzle:

- We usually expect that the latent dataset is low dimensional encoded representation of the input.

$$M < N_{\text{in}}, N, T$$

Latent Dimensions: *A Puzzle*

A Puzzle:

- We usually expect that the latent dataset is low dimensional encoded representation of the input.

$$M < N_{\text{in}}, N, T$$

- Our scaling-law model requires that the size of the latent space is the largest scale in the problem.

$$M > N_{\text{in}}, N, T$$

Latent Dimensions: *A Possible Resolution*

There are multiple ways to think about the intrinsic dimension:

Latent Dimensions: *A Possible Resolution*

There are multiple ways to think about the intrinsic dimension:

1. M

Latent Dimensions: *A Possible Resolution*

There are multiple ways to think about the intrinsic dimension:

1. M
2. A nice method considers the typical (Euclidean) distance, $\langle \delta \rangle$, between neighboring points:

$$\langle \delta \rangle \sim T^{-1/d_{\text{intrinsic}}} .$$

[Levina/Bickel, Facco/et al.]

Latent Dimensions: *A Possible Resolution*

There are multiple ways to think about the intrinsic dimension:

1. M
2. A nice method considers the typical (Euclidean) distance, $\langle \delta \rangle$, between neighboring points:

$$\langle \delta \rangle \sim T^{-1/d_{\text{intrinsic}}} .$$

[Levina/Bickel, Facco/et al.]

Using this, Sharma/Kaplan argued that the test loss should be inversely proportional to typical linear size, $\langle \delta \rangle$, of a subregion occupied by each data point:

$$d_{\text{intrinsic}} = \frac{\#}{\alpha} .$$

Latent Dimensions: *A Possible Resolution*

There are multiple ways to think about the intrinsic dimension:

1. M
2. A nice method considers the typical (Euclidean) distance, $\langle \delta \rangle$, between neighboring points:

$$\langle \delta \rangle \sim T^{-1/d_{\text{intrinsic}}} .$$

[Levina/Bickel, Facco/et al.]

Using this, Sharma/Kaplan argued that the test loss should be inversely proportional to typical linear size, $\langle \delta \rangle$, of a subregion occupied by each data point:

$$d_{\text{intrinsic}} = \frac{\#}{\alpha} .$$

Conclusion: While the latent space is M-dimensional, it has a rigid power-law structure that leads to different notions of dimension.

Latent Dimensions: *A Possible Resolution*

There are multiple ways to think about the intrinsic dimension:

1. M
2. A nice method considers the typical (Euclidean) distance, $\langle \delta \rangle$, between neighboring points:

$$\langle \delta \rangle \sim T^{-1/d_{\text{intrinsic}}} .$$

[Levina/Bickel, Facco/et al.]

Using this, Sharma/Kaplan argued that the test loss should be inversely proportional to typical linear size, $\langle \delta \rangle$, of a subregion occupied by each data point:

$$d_{\text{intrinsic}} = \frac{\#}{\alpha} .$$

But also: Regardless, the analysis implies that an AI systems will still need to scale its resources as $T, N \lesssim M$.

Conclusion

- We presented explored the properties of datasets and feature maps that occur in natural datasets and DNNs and used that to build a joint generative data model and random feature model that captures the phenomenology of neural scaling laws.

Conclusion

- We presented explored the properties of datasets and feature maps that occur in natural datasets and DNNs and used that to build a joint generative data model and random feature model that captures the phenomenology of neural scaling laws.
- (We also solved the model, but we didn't explain how.)

Conclusion

- We presented explored the properties of datasets and feature maps that occur in natural datasets and DNNs and used that to build a joint generative data model and random feature model that captures the phenomenology of neural scaling laws.
- (We also solved the model, but we didn't explain how.)
- This let us explore how *power laws* and *plateaus* arise, the *breakdown* of the empirical LLM behavior, as well as understand why *equiparameterization* is important.

Future Directions

- Where do the power laws in natural datasets come from?
- Can we improve our theoretical analysis to optimize over the ridge parameter γ?
- Can we extend our scaling-law analysis to nonlinear models with feature learning such as **quadratic models**?

[DR/Yaida/Hanin]

- Can we learn the latent dimension M of real data since it shows up in our solution?
- Can we use our knowledge of why scaling laws arise to predict exponents in more complicated systems of practical relevance?

Thank You!