Learning Curve Theory

Marcus Hutter

DeepMind, London, UK
http://www.hutter1.net/

Abstract

Recently a number of empirical "universal" scaling law papers have been published, most notably by OpenAI. 'Scaling laws' refers to power-law decreases of training or test error w.r.t. more data, larger neural networks, and/or more compute. In this work we focus on scaling w.r.t. data size n. Theoretical understanding of this phenomenon is limited, except in finite-dimensional models for which error typically decreases with $n^{-1/2}$ or n^{-1}, where n is the sample size. We develop and theoretically analyse the simplest possible (toy) model that can exhibit $n^{-\beta}$ learning curves for arbitrary power $\beta > 0$, and determine to which extent power laws are universal or depend on the data distribution or loss function: Roughly, learning curves exhibit a power law with $\beta = \frac{\alpha}{1+\alpha}$ for Zipf-distributed data with exponent $1 + \alpha$, independent of the choice of loss. Furthermore, noise rapidly deteriorates/improves in instantaneous/time-averaged learning curves for increasing n, suggesting that model selection should be based on cumulative (AUC) or time-averaged error, not final test error.

Table of Contents

Table of Contents

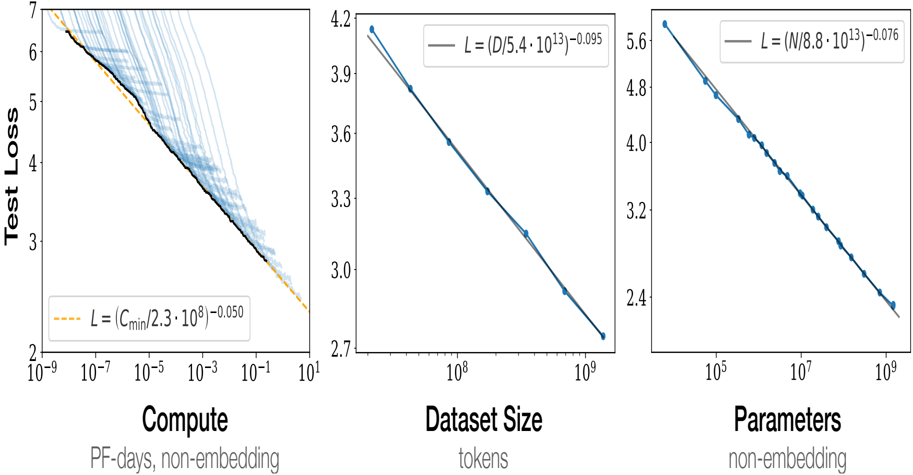
Power Laws in Large-Scale Machine Learning

- '*Mantra*' of modern machine learning: '*bigger is better*'.
- The larger and deeper *Neural Networks (NNs)* are,
  the more data they are fed, the longer they are trained,
  the better they perform.
- *Quantification:* Test error decreases as a *power law*,
  with the *data size*, with the *model size* (number of NN parameters),
  as well as with the *compute budget* used for training ...
- assuming one factor is not "*bottlenecked*" by the other two factors,
  -or- all three factors are increased appropriately in tandem.
- *Note: Subtract irreducible error* due to intrinsic noise in the data
  and/or non-vanishing model mis-specification.

Power Laws in Deep Learning

DeepLearning Scaling [KMH+20] – Log-log Plots

Test loss of a Transformer trained to autoregressively model language



Ubiquity/Universality of Power Laws

Power laws have been observed for many

- *problem types* (supervised, unsupervised, transfer learning)
- *data types* (images, video, text, even math)
- *many NN architectures* (Transformers, ConvNets, ...)
- *different loss functions* (cross-entropy, log, logistic, 0-1)

[HNA+17, RRBS19, HGLS20, HKK+20, KMH+20]

- This has *led some to the belief that power laws might be universal*: Whatever the problem, data, model, learning algorithm, or loss, learning curves follow power laws.
- To which extent this conjecture is true, we do not know, since *theoretical understanding* of this phenomenon is *limited*.

This Talk

- *Scaling with data size* n.
- *Problem*: Classical learning theory leads to scaling laws $n^{-\beta}$ with $\beta = \frac{1}{2}$ or $\beta = 1$, not the observed $\beta \approx 0.05...0.35 < \frac{1}{2}$.
- *Conjecture:* Any theoretical explanation of $\beta < \frac{1}{2}$ *requires real-world data and models of unbounded complexity*.

Possible suitable model choices:

(a) *scaling up the model* (e.g. NN) with data, as done in the experiments [intertwines scaling with data and scaling with model size]

(b) *non-parametric models* (e.g. kNN [SB14], Kernel regression [BCP20]) [more sophisticated analysis, manifold explanation]

(c) a *model with (countably-)infinitely-many parameters (this talk)* [Hut21] [more accurate analysis. Zipf explanation]

General Findings within our Toy Model

- For domains of unbounded complexity, a *variety of learning curves* are possible, not only power-laws.

- Real *data* is often *Zipf distributed* (e.g. the frequency of words in text), which is itself a power law. This *implies power law learning curves with "interesting"* $\beta < \frac{1}{2}$,

- Though many (even *non-Zipf*) distributions *also* lead to *power laws but with "***un***interesting"* $\beta = 1$.

It is plausible that these findings remain true for most infinite models.

Key Findings within our Toy Model

In general, learning curves consist of 3 terms

1. a *data-***in***dependent loss-dependent* power law (usually $n^{-1/2}$ or n^{-1}),
2. a *data-dependent loss-***in***dependent* power law $n^{-\beta}$ for $0 < \beta \leq 1$, with (typically small) $\beta = \frac{\alpha}{1+\alpha}$ for $(\alpha + 1)$-Zip-distributed data,
3. an *irreducible term* due to noise and/or model approximation error.

The signal-to-noise ratio

- rapidly *deteriorates* with n in instantaneous learning curves.
- rapidly *improves* with n in time-averaged learning curves.
- Consistent with arguments by [Hut06] for log-loss, *Model selection should be based on cumulative (AUC) or time-averaged error*, rather than final test error.

Table of Contents

Scaling with Model Size

- Consider a *function* $f : [0;1]^d \to \mathbb{R}$ which we wish *to approximate*.
- A naive approximation is to discretize the hyper-cube to an ε*-grid*. This constitutes a model with $m = (1/\varepsilon)^d$ *parameters*.
- If f is 1-Lipschitz, it can approximate f to *accuracy* $\varepsilon = m^{-1/d}$, i.e. the (absolute) error scales with model size m as a power law with exponent $-1/d$.
- More generally, if first k *derivatives* of f are bounded, m parameters suffice and are necessary for $\Theta(m^{-k/d})$ *approximation accuracy* [Mha96, DHM89]
- *Adapted to NNs* by [Pin99] and *empirically verified and extended* by [SK20] to using the dimension of the data distribution in the penultimate layer of the NN.

Data Size↔Iterations↔Compute

(i) Usually in deep learning, *compute is proportional to the number of learning iterations*, since/provided batch and model size are kept fixed.

(ii) in *online learning*, every data item is used only once,
hence the size of data used up to iteration n is proportional to n.

(iii) This is also true for *stochastic learning* algorithms for some recent networks, such as GPT-3, trained on massive data sets, where every data item is used at most once (with high probability).

(iv) When generating *artificial data*, it is natural to generate a new data item for each iteration.

Hence in these 4 settings, the *learning curves, error-with-data-size, error-with-iterations, and error-with-compute, are scaled versions of each other*. For this reason, *scaling of error with iterations also tells us how error scales with data size and even with compute*,
but scaling with model size is different.

Scaling with Data Size

- This is the traditional domain of *Statistical Learning Theory (SLT)* [SB14], *online learning* [GPS18], and *online convex optimization* [Haz16].
- The *fundamental (PAC) theorem of SLT* states that the empirical error converges to the generalization error at a rate of $n^{-1/2}$ for models of finite VC-dimension, and n i.i.d. samples.
- Applies to *many models* (SVMs, regression, NNs, finite decision trees, ...), *many algorithms* (Empirical Risk Minimization (ERM), (stochastic) gradient descent approximations, ...) *many losses* (convex-Lipschitz-bounded, convex-smooth-bounded, ...).

Scaling with Data Size (ctd)

- $n^{-1/2}$ scaling also trivially follows *from the central limit theorem* for virtually any finitely-parameterized model in the under-parameterized regime of more-data-than-parameters:
  Parameters can be estimated to accuracy $n^{-1/2}$ hence absolute (locally quadratic loss) decays with $n^{-1/2}$ (n^{-1}).

- We could easily create power laws with any β by choosing *exotic loss* $|\hat{y} - y_t|^{\beta/2}$, but this would *not explain* the observed β for the used standard losses.

- The average *regret* considered *in online learning* theory and online convex optimization has similar requirements on the model (e.g. finite-dimensional) and *exhibits the same rates* $n^{-1/2}$ or n^{-1} (or $\frac{1}{n}\ln n$ due to the time-average), *under similar conditions*.

Interesting Scaling with Data Beyond $n^{-1/2}$

- An example of a *non-parametric model* whose sample complexity has been analysed with "interesting" rate, is *k-nearest neighbors (kNN)*.
- For d-dimensional Lipschitz functions, the *error of kNN* is bounded by $n^{-1/(d+1)}$ [SB14, Thm.19.3&19.5].
- Power $-1/(d+1) \approx -1/d$ is *due to density of data points* being $n^{-1/d}$ similar to discretization discussed before in terms of model size.
- Learning curves $n^{-\alpha/d}$ for *kernel regression* [BCP20, SGW20, BDK⁺21].
- Also hold for *infinitely wide NNs*, since equivalent to kernel regression with a Neural Tangent Kernel (NTK)
- α depends on target smoothness and choice of loss function.
- The underlying mechanism of ε-covering a d-dimensional *data manifold* with $n \overset{\times}{\approx} (1/\varepsilon)^{d/\alpha}$ data points is the same.
- *The origin of the power law in our toy model is very different*.

Table of Contents

The Goal of this Work

Identify and study the simplest model that is able to exhibit power-law learning curves as empirically observed in Deep Learning.

- *Toy model:* i.i.d. classification problems with countable feature space.
- A natural practical *example* application would be *classifying words* w.r.t. some criterion.
- *Slides: deterministic labels* and *0-1 loss*
- *Toy algorithm* predicts/recalls the *class* for a new *feature* from a previously observed (*feature*,*class*) pair,
  or acts randomly on a novel *feature*.
- *Paper:* Extension to *noisy labels* and *general loss*.

The Toy Model

- *Classification:* $h \in \mathcal{H} := \mathcal{X} \to \mathcal{Y}$, e.g. $\mathcal{Y} = \{0, 1\}$ for binary.
- *Classifier* h learnt from data $\mathcal{D}_n := \{(i_1, y_1), ..., (i_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$.
- We need *infinite* $\mathcal{X}$ for interesting learning curves.
- *Smallest suitable* $\mathcal{X} \simeq \mathbb{N}$, which we henceforth assume.
- *Model class* $\mathcal{H} := \mathbb{N} \to \mathcal{Y}$ is uncountable and has ∞ VC-dim., hence is not PAC learnable, but still can be learnt consistently.
- *Features* $i_t \in \mathbb{N}$ are drawn i.i.d. with $\mathbb{P}[i_t = i] =: \theta_i \geq 0$ $(\sum_{i=1}^{\infty} \theta_i = 1)$.
- ∞ vector $\boldsymbol{\theta} \equiv (\theta_1, \theta_2, ...)$ characterizes the *feature distribution*.
- *Noise-free:* Label $y_t = h_0(i_t)$, where $h_0 \in \mathcal{H}$ is unknown true deterministic labelling function.
- *Results change little for noisy labels.*

The Toy Algorithm

Toy Algorithm $A : \mathbb{N} \times (\mathbb{N} \times \mathcal{Y})^* \to \mathcal{Y}$

- *memorizes* all past labelled features $\mathcal{D}_n$.
- on next feature $i_{n+1} = i$ recalls y_t if $i_t = i$ for some $i \leq n$,
- *or* outputs *undefined* if $i \notin i_{1:n}$ i.e. if i is new.

Formally:

$$A(i, \mathcal{D}_n) \;:=\; \begin{cases} y_t & \text{if} \quad i = i_t \text{ for some } t \leq n \\ \perp & \text{else} \quad \text{i.e. if } i \notin i_{1:n} \end{cases}$$

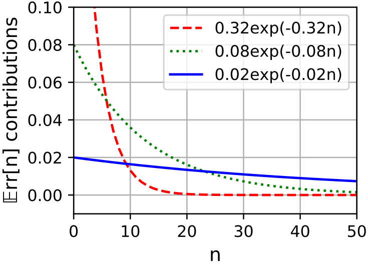
Error

- Algorithm A only makes an *error* predicting label y_{n+1} if $i_{1:n} \notin i_{1:n}$.
- Formally, the *(instantaneous) error* E_n of A when predicting label y_{n+1} for feature i_{n+1} from $\mathcal{D}_n$ is $\mathrm{E}_n \; := \; [\![i_{n+1} \notin i_{1:n}]\!]$.
- *Expected (instantaneous) error* (w.r.t. $\mathcal{D}_n$ and i_{n+1}): $\mathbb{E}_n \; := \; \mathbb{E}[\mathrm{E}_n] \; = \; \mathbb{P}[i_{n+1} \notin i_{1:n}] \; = \; \sum_{i=1}^{\infty} \theta_i (1-\theta_i)^n$
- *Intuition:* If feature i has not been observed so far (happens with prob. $(1-\theta_i)^n$), then feature i is observed (happens with prob. θ_i), the algorithm makes an error.
- $\mathbb{E}_n$ as a function of n constitutes an *(expected) learning curve*.
- Cf. *probability of discovering a new species from data* [Cha81], but usage&analyses of model & resulting expressions are totally different.
- *Results change little for most other loss functions*.

Table of Contents

Exponential Decay

- *Very simple case:* m of the θ_i are equal, the rest are 0.
- *Error* $\mathbb{E}_n = (1-\frac{1}{m})^n \leq e^{-n/m}$ decays exponentially with n.



This case is *not too interesting* to us, since

(a) this case corresponds to a *finite model*

(b) exponential decay is an "*artifact*" of the deterministic label and discontinuous 0-1 error.

(c) *becomes a power law* $1/n$ after time-averaging (see later).

(d) *does not explain the Deep Learning power law learning curves*.

Superposition of Exponentials

- Expected Error $\mathbb{E}_n$ is invariant under *bijective renumbering* of features $i \in \mathbb{N}$

- Hence we can *w.l.g. assume* $\theta_1 \geq \theta_2 \geq \theta_3 \geq ....$

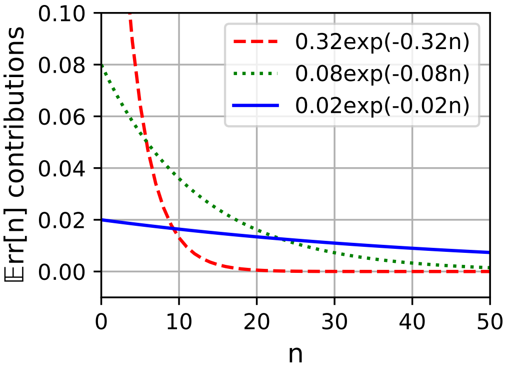
- Some θs may be equal.
  Group equal θs together into $\bar{\bar{\theta}}_j$ with multiplicity $m_j > 0$

- $\mathbb{E}_n \;=\; \sum_{j=1}^{M} m_j \bar{\bar{\theta}}_j e^{-n\bar{\bar{\vartheta}}_j}$, where $\bar{\bar{\vartheta}}_j := -\ln(1-\bar{\bar{\theta}}_j) \approx \bar{\bar{\theta}}_j$

- $M \in \mathbb{N} \cup \{\infty\}$ is the number of *different* $\theta_i > 0$.

Superposition of Exponentials

- $\mathbb{E}_n \;=\; \sum_{j=1}^M m_j\bar{\bar{\theta}}_j e^{-n\bar{\bar{\vartheta}}_j}$ is a *superposition of exponentials* in n with different decay rates $\bar{\bar{\vartheta}}_j$
- Sum will be *dominated* by different terms at different "times" n.
- Different *phases* of exponential decay



- For $M<\infty$, *eventually exponential decay* $e^{-n\bar{\bar{\vartheta}}_M}$ will dominate $\mathbb{E}_n$.
- The *same "caveats"* (a)-(d) apply as for $M=1$ two slides ago.

Approximation

- Let $f:\mathbb{R}\to\mathbb{R}$ be a smooth and monotone decreasing *interpolation* of $\boldsymbol{\theta}:\mathbb{N}\to\mathbb{R}$, i.e. $f(i):=\theta_i$ and $f'(x)<0$:

$$\mathbb{E}_n \;=\; \sum_{i=1}^{\infty} f(i)(1-f(i))^n \;\approx\; \int_1^{\infty} f(x)e^{-nf(x)}dx$$

$$\stackrel{(a)}{=}\; \int_0^{\theta_1} \frac{ue^{-nu}du}{|f'(f^{-1}(u))|} \stackrel{\times}{\approx} \frac{1}{n^2|f'(f^{-1}(\frac{1}{n}))|} \;=\; \frac{d}{dn}f^{-1}(\tfrac{1}{n})$$

(a) *Reparametrization* $u=f(x)$ and $f(1)=\theta_1$ and $f(\infty)=0$ and $dx=du/f'(x)$ and $f'<0$.

($\times$) Numerator ue^{-nu} *concentrated* around $u=1/n$, hence can replace u by $1/n$ in denominator.

- *Intuition:* $\mathbb{E}_n$ is dominated by samples i_0 for which $\theta_{i_0}\approx\frac{1}{n}$.

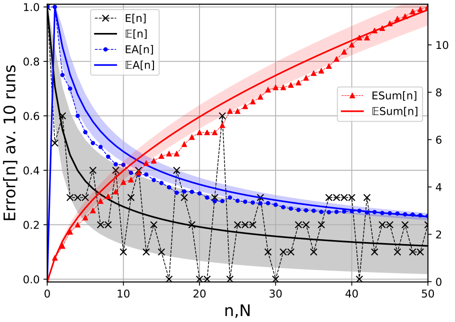
- *Accuracy* of the integral representation is $1/en+o(1/n)$.

Zipf-distributed data

- Empirically many *data* follow a power-law distribution called *Zipf distr.* in this context:

- The *frequency* of the ith most frequent item is approximately $\theta_i \propto i^{-(\alpha+1)}$ for some $\alpha > 0$.

- $\mathbb{E}_n \overset{\times}{=} n^{-\beta}$ where $\beta := \frac{\alpha}{1+\alpha}$



- That is, Zipf-distributed data (with power $\alpha + 1$) lead to a *power-law learning curve* (with power $\beta = \frac{\alpha}{1+\alpha} < 1$).

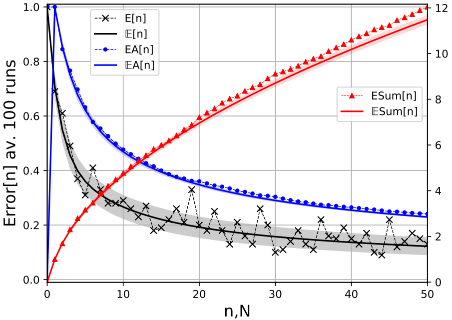
(Super)Exponentially-Distributed Data

- *Exponential data distr.* $\theta_i \propto e^{-\gamma i}$ is more skewed than any power law.
- Still $\mathbb{E}_n \approx 1/\gamma n$, i.e. *still leads to a power law* learning curve.
- But exponent $\beta = 1$ is "uninteresting" (much larger than observed)
- *Surprise:* Any *super-exponential* data (e.g. $\theta_i \propto e^{-\gamma i^2}$, but quite unrealistic) *always* leads to a (sort of) power law as long as $\theta_i > 0$ for infinitely many i, unlike finite model which gives exponential decay:
- $\mathbb{E}_n \overset{\times}{\leq} n^{-1}$ for all n and $\mathbb{E}_n \overset{\times}{\geq} n^{-1}$ for infinitely many n.

Table of Contents

Instantaneous Variance

- *Variance* $\mathbb{V}_n$ of $\text{E}_n := [\![i_{n+1} \notin i_{1:n}]\!]$ as a function of n is important.
- Useful learning curve requires *Standard Error (STE)* $\sqrt{\mathbb{V}_n/k} \; < \; \mathbb{E}[\text{E}_n] \equiv \mathbb{E}_n =: \mu_n$ when averaging over k runs.
- $\text{E}_n \in \{0,1\}$ hence $\text{E}_n^2 = \text{E}_n$ hence $\mathbb{V}[\text{E}_n] = \mathbb{E}[\text{E}_n^2] - \mathbb{E}[\text{E}_n]^2 = \mu_n(1-\mu_n)$
- Since $\mu_n \to 0$ for $n \to \infty$, the *Standard Deviation (STD)* $\sigma_n := \sqrt{\mathbb{V}[\text{E}_n]} = \sqrt{\mu_n(1-\mu_n)} \approx \sqrt{\mu_n} \quad \gg \quad \mu_n = \mathbb{E}_n$
- For good *signal-to-noise ratio* we need $k \gg \mu_n^{-1/2}$ runs (increasing with n!)



Time-Averaged Mean and Variance

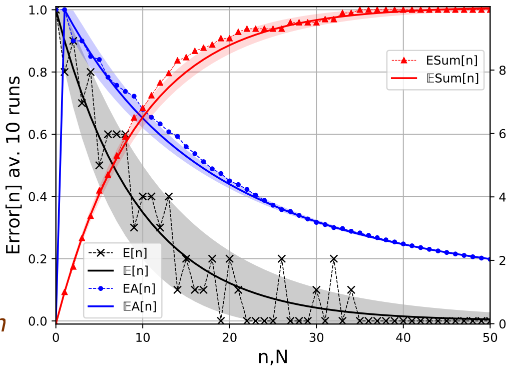
- *Alternative:* Report the *time-averaged error* $\bar{\mathrm{E}} := \frac{1}{N}\sum_{n=0}^{N-1}\mathrm{E}_n$, rather than the instantaneous error E_n.

- *Expectation:* $\mathbb{E}[\bar{\mathrm{E}}_N] = \frac{1}{N}\sum_{i=1}^{\infty}[1-(1-\theta_i)^N]$

- *Variance:* $$\mathbb{V}[\bar{\mathrm{E}}_N] = \frac{1}{N^2}\sum_{i=1}^{\infty}(1-\theta_i)^N[1-(1-\theta_i)^N] - \frac{1}{N^2}\sum_{i\neq j}[(1-\theta_i)^N(1-\theta_j)^N-(1-\theta_i-\theta_j)^N]$$

Uniform Case $\theta_i = \frac{1}{m}[\![i \leq m]\!]$

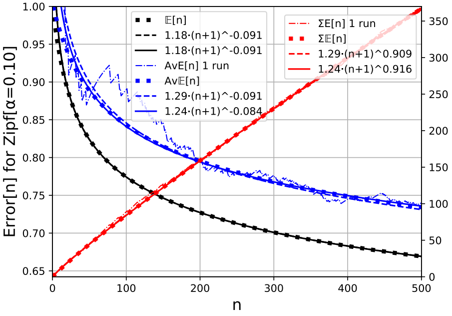
- $\mathbb{E}_n = (1-\frac{1}{m})^n \approx e^{-n/m}$ decays exponentially, but
- $\mathbb{E}[\bar{E}_N] = \frac{m}{N}[1-(1-\frac{1}{m})^N] \longrightarrow \frac{m}{N}$ for $N \to \infty$
- $\sigma[\bar{E}_N] \approx \frac{\sqrt{m}}{N} e^{-N/2m} \ll \frac{m}{N} \approx \mathbb{E}[\bar{E}_N]$ for $N \gg m$



- I.e. Standard Deviation is (much) smaller than the mean for $N \gg m$, so the *time-averaged learning curves have a much better signal-to-noise ratio*.

Zipf Case $\theta_i \propto i^{-(\alpha+1)}$

- Recall *expected error*: $\mathbb{E}_n \approx c_\alpha n^{-\beta}$, where $0 < \beta = \frac{\alpha}{1+\alpha} < 1$.
- *Time-averaged expected error*: $\mathbb{E}[\bar{\mathrm{E}}_N] \approx \frac{c_\alpha}{N}\int_0^N n^{-\beta}dn = \frac{c_\alpha}{1-\beta}N^{-\beta}$
- Same power law with the *same exponent* β (generic property)
- *STD* $\sigma[\bar{\mathrm{E}}_N] \overset{\times}{\approx} N^{-\frac{1/2+\alpha}{1+\alpha}} \quad \ll \quad N^{-\frac{\alpha}{1+\alpha}} \overset{\times}{\approx} \mathbb{E}[\bar{\mathrm{E}}_N]$
- *Signal-to-noise ratio* is $\sigma[\bar{\mathrm{E}}_N]/\mathbb{E}[\bar{\mathrm{E}}_N] \overset{\times}{\approx} N^{-1/(2+2\alpha)}$. *STD much smaller than Mean*.
- *Single run suffices* to get a good (and excellent for $n \gtrsim 500$) signal-to-noise ratio for ave. and cum. error



General θ Case

- *Signal-to-noise ratio:* $\frac{\sigma[\bar{E}_N]}{\mathbb{E}[\bar{E}_N]} \leq \frac{\sqrt{\frac{1}{N}\mathbb{E}_N}}{\mathbb{E}[\bar{E}_N]} = \frac{\sqrt{N\mathbb{E}_N}}{\sum_{n=0}^{N-1}\mathbb{E}_n} \stackrel{N\to\infty}{\longrightarrow} 0$

- *Proof* requires to distinguish two cases:

1) $\sum_{n=0}^{\infty}\mathbb{E}_n \leq c$ (e.g. exponential error decay in finite models),

2) $\sum_{n=0}^{N-1}\mathbb{E}_n \to \infty$ (most ∞ models, e.g. Zipf, even exponential θ_i)

Instantaneous vs. Time-Averaged Error

- *Trivial observation:* For $\theta_0 = 1$, we have $i_n = 1\ \forall n$, hence $\mathrm{E}_0 = 1$ and $\mathrm{E}_n = 0\ \forall n \geq 1$ and $\mathbb{V}[\mathrm{E}_n] = 0\ \forall n$.

- This is the fastest any error can decay, 0 after 1 observation, hence *always* $\overline{\mathrm{E}}_n = \Omega(1/n)$. *Fazit:*

  If $\mathbb{E}_n = o(1/n)$, *report* E_n, since $\ll \overline{\mathrm{E}}_n$.

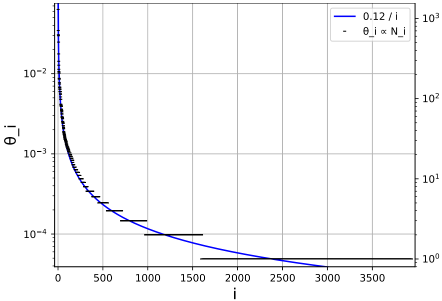
  If $\mathbb{E}_n = \tilde{\Omega}(1/n)$, *report* $\overline{\mathrm{E}}_n$, since $\stackrel{\times}{\approx} \mathrm{E}_n$ but variance is smaller.

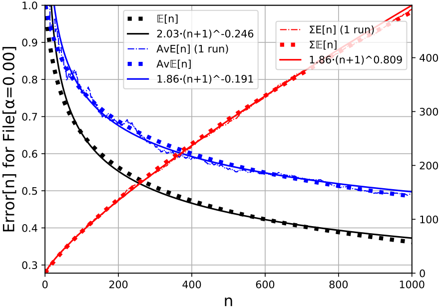
- Esp. in Deep Learning with small β, we have $\overline{\mathrm{E}}_n \ \approx\ \mathrm{E}_n$.

- Low variance does not follow directly from law of large numbers, since $\mathrm{E}_1, \mathrm{E}_2, \mathrm{E}_3, ...$ are not independent.

Zipf-Distributed Words in Typical Texts

first 20469 words in file 'book1' of the Calgary Corpus





Relative (left scale) and absolute (right scale) *word frequency, and fitted Zipf law*.

Power law fit to learning curve for this data set for a *word classification task*.

- The *power-law fit is good* if n is not too large.
- For large n, the error decays exponentially as $\exp(-\theta_{min} n)$, since word frequency is quantized ($\in \mathbb{N}$).

Table of Contents

Noisy Labels or Targets – Implications

(a) Need "smarter" "learning" algorithm, e.g. predicting the average.

(b) Subtract irreducible error due to label noise before studying scaling.

(c) Extra $n^{-1/2}$ (n^{-1}) additive error term for absolute (square loss) due to parameter estimation error, hence

(d) Inst. loss will not decay expon. anymore even if model is finite.

(e) Otherwise the *scaling laws for Zipf data are unchanged*.

In summary, *conceptually error/loss is a sum of 3 terms*:

(1) The *parameter learning rate* $n^{-1/2}$ (squared for locally quadratic loss)

(2) the *same power law* $n^{-\beta}$ as in the deterministic case,

(3) the inherent "*entropy*" in the data.

- *Remarkably:* Instantaneous square $\text{Loss}_n^{noisy}(A) \stackrel{\times}{=} \mathbb{E}_n^{det.} + \mathbb{E}[\bar{\text{E}}_n^{det.}]$.
- This "magically" ensures (c,d,e), since $\mathbb{E}[\bar{\text{E}}_n] \stackrel{\times}{\approx} \max\{\mathbb{E}_n, \frac{1}{n}\}$.
- For instance, for a finite model, $\text{Loss}_n(A) \stackrel{\times}{\approx} \mathbb{E}[\bar{\text{E}}_n] \stackrel{\times}{\approx} \frac{1}{n}$.

Other Loss Functions

- *Deterministic toy model*: $\mathbb{E}[\text{Loss}_n] \overset{\times}{=} \mathbb{E}_n$ for most loss functions

- *Noisy labels*: Same, but extra n^{-1} or $n^{-1/2}$ term
  (now fastest possible decay)

- *Universality* at least within toy model: For large models,
  scaling laws are indep. of loss function and not affected by noise.

Continuous Features

- Feature spaces are most often *vector spaces* $\mathbb{R}^d$.
- *No feature ever repeats exactly* ($x_n \neq x_m$ for $n \neq m$).
- *Simple processes*: Dirichlet = Chinese Restaurant = Stick-Breaking.
- Leads to *power law learning curves* n^{-1}, but $\beta = 1$ is *uninteresting*.
- Generalized 2-parameter *Poison Dirichlet Process* [BH10] also only leads to $\beta = 1$.
- *Open problem:* Finding analytically tractable models with continuous features that exhibit interesting learning curves.

Generalizing Algorithms

- Proper models/algorithms for continuous features *need to generalize* from observed inputs to similar future not-yet-observed inputs.
- *Simple model: Partition domain into countably many cells*
- If done a-priori and independent $\mathcal{D}_n$ reduces back to toy model
- More realistically, if partitioning, e.g. clustering of data, is data (size) dependent, it will affect the scaling.
- 'perfect prediction for exact repetition' *abstracts*
  'classify features in the same cell alike' *abstracts*
  'classify similar observations alike or similarly'.
- So maybe some of our findings or analysis tools approximately *transfer*.

Deep learning

- (Deep) neural networks are a particularly powerful class of models/algorithms that *can generalize*,
- But they are notoriously *difficult to theoretically analyse*.
- It may be a *long way from our toy model* to a similar analysis of NNs.
- Furthermore we have not at all considered the equally interesting questions of *scaling with model size*.

Table of Contents

Summary

- We introduced and analyzed the simplest model that can exhibit *power laws (decrease of error with data size)* consistent with recent findings in deep learning.
- Many but not all *data distributions* lead to power laws.
- *Zipf data* with exponent $\alpha + 1$ lead to power law with exponent $\beta = \alpha/(1+\alpha)$. Artifact of the model or wider validity?
- The *signal-to-noise ratio* for the time-averaged error tends to zero, which implies that a single experimental run suffices for stable results.
- *Model selection* should be based on cumulative (AUC) error, rather than final test error [Hut06].

Limitations

- The *toy model is totally unrealistic* as a Deep Learning model,
- but we believe it captures the (or at least a) true reason for the observed scaling laws w.r.t. data.
- Hopefully can be generalized to NNs
- We have not addressed *scaling laws w.r.t. model size*.

Applications

- May help making *better or more principled choices* for network architecture (depth, with, and beyond), hyper-parameters, fine-tuning, data augmentation, pre-training, etc. [CJS+93, HGLS20].
- Being able to extrapolate the consequences of such choices from *cheap training on a small subset of the data* to the whole corpus by simply fitting power laws can save significant compute.
- The *cost of training recent models has reached millions of dollars* and can exhaust and exceed even FAANGs computational resources.

List of Notation

Symbol	Explanation
$[\![\text{Bool}]\!]$	1 if Bool=True, 0 if Bool=False
$\mathbb{E}, \mathbb{V}$	Expectation, Variance
$\stackrel{\times}{=}$	Equal within a multiplicative constant
θ_i	probability of feature i
$\mathcal{D}_n$	Data consisting of n (feature i,label y) pairs
E_n	Instantaneous Error of A on i_{n+1} predicting y_{n+1} from $\mathcal{D}_n$
$\mathbb{E}_n$	Expectation of Instantaneous Error E_n w.r.t. $\mathcal{D}_{n+1}$
$\overline{\text{E}}_N$	Time-Averaged Error E_n from $n = 0, ..., N-1$
$\alpha + 1$	Exponent of Zipf distributed data frequency $i^{-(\alpha+1)}$
β	Exponent of power law $n^{-\beta}$ for error as a function of data size n
γ	Decay rate for exponential data distribution $e^{-\gamma i}$

References

[BDK+21] Y. Bahri, E. Dyer, J. Kaplan, J. Lee, and U. Sharma.
Explaining Neural Scaling Laws.
arXiv:2102.06701 [cond-mat, stat], February 2021.

[HKK+20] T. Henighan et al.
Scaling Laws for Autoregressive Generative Modeling.
arXiv:2010.14701 [cs], November 2020.

[HGLS20] D. Hoiem, T. Gupta, Z. Li, and M. Shlapentokh-Rothman.
Learning Curves for Analysis of Deep Networks.
arXiv:2010.11029 [cs, stat], October 2020.

[Hut06] M. Hutter. Human knowledge compression prize.
open ended, http://prize.hutter1.net/, 2006.

[Hut21] M. Hutter. Learning Curve Theory.
Technical Report, DeepMind, London, 2021.
http://www.hutter1.net/official/bib.htm#scaling