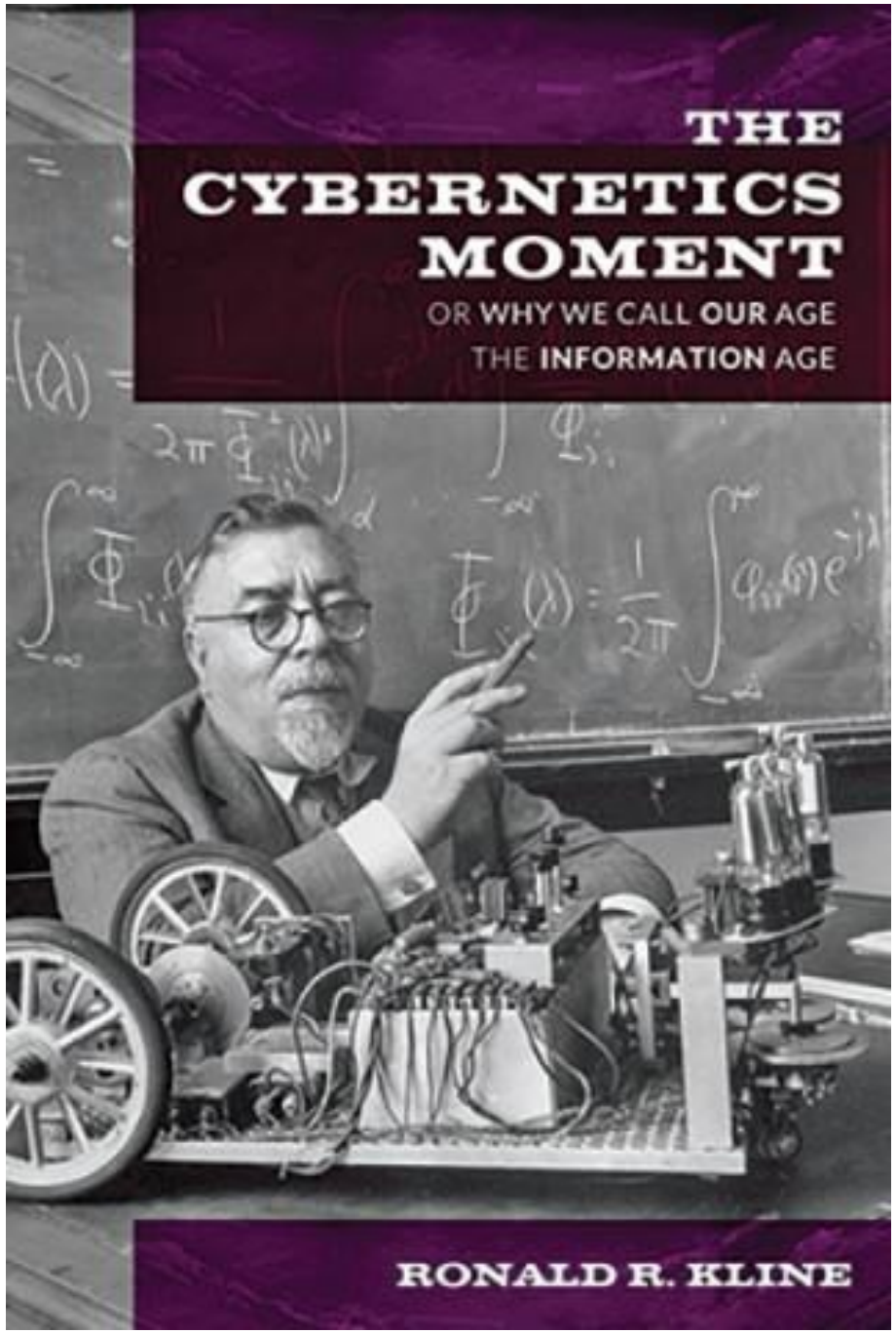
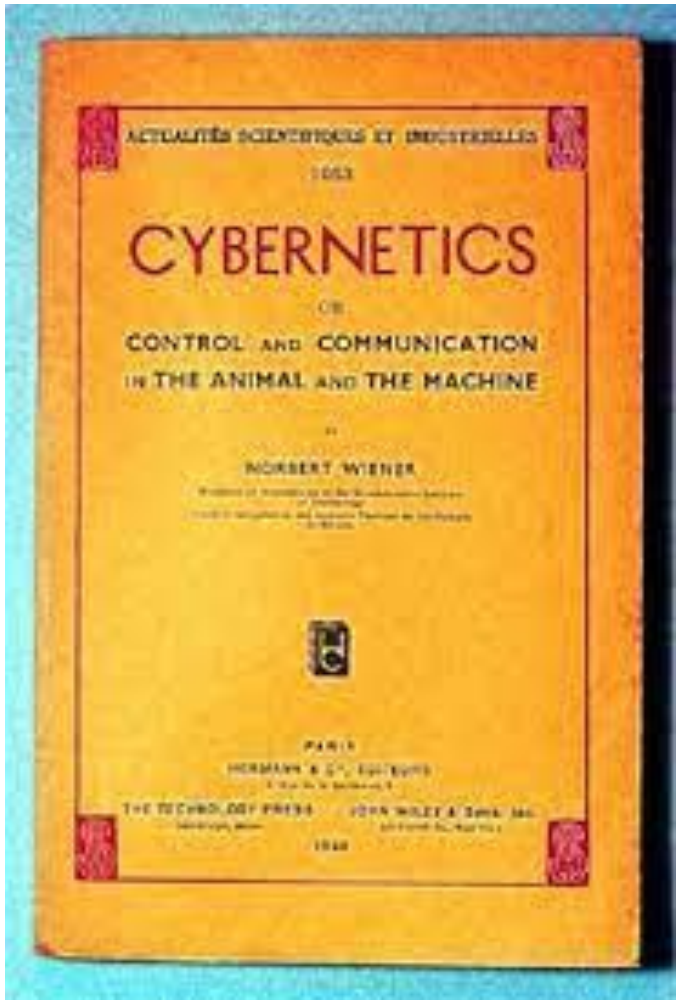
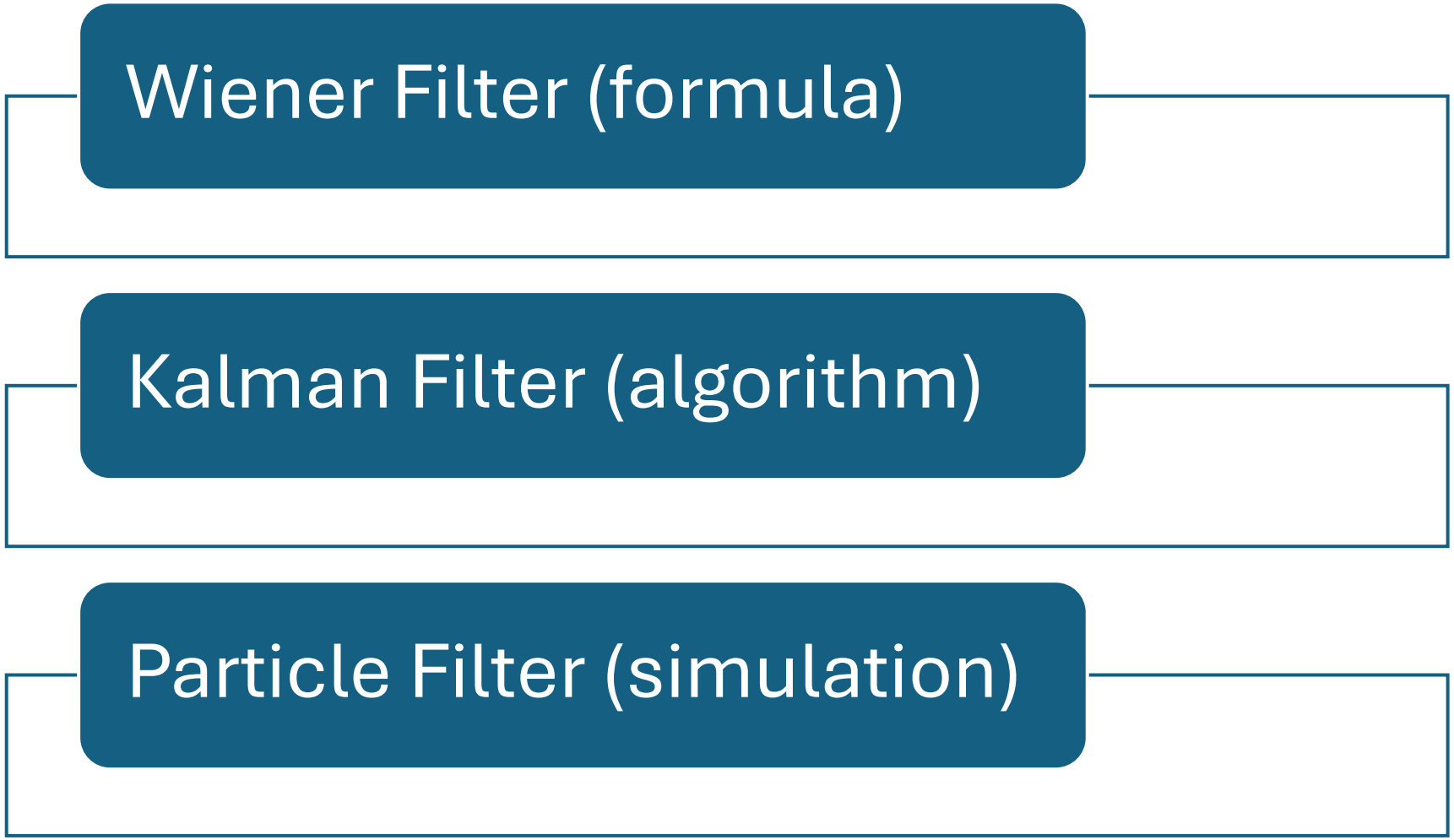
Representation of Information
ECE 598 LV – Lecture 23

Lav R. Varshney

11 April 2024


THE
CYBERNETICS
MOMENT
OR WHY WE CALL OUR AGE
THE INFORMATION AGE
RONALD R. KLINE


ACTUALITÉS SCIENTIFIQUES ET INDUSTRIELLES
CYBERNETICS
OR
CONTROL AND COMMUNICATION
IN THE ANIMAL AND THE MACHINE
NORBERT WIENER
PARIS
HERMANN & Cie, Editeurs
THE TECHNOLOGY PRESS
JOHN WILEY & SONS, Inc.
1948

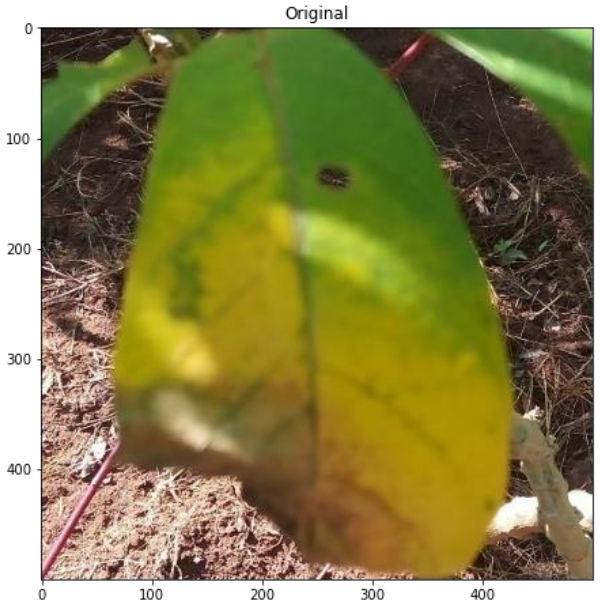

Wiener Filter (formula)
Kalman Filter (algorithm)
Particle Filter (simulation)

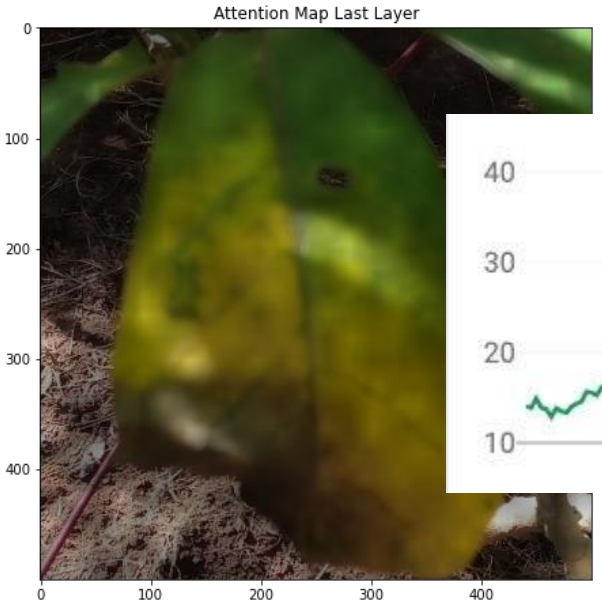
Mamba: Linear-Time Sequence Modeling with Selective State Spaces

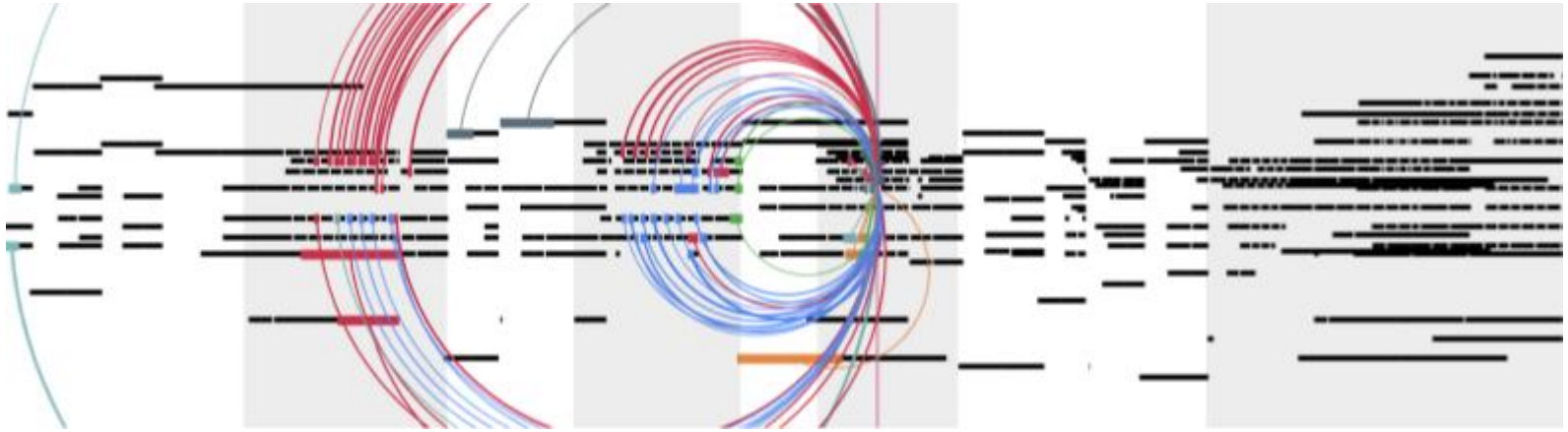
Albert Gu*[1] and Tri Dao*[2]

[1]Machine Learning Department, Carnegie Mellon University
[2]Department of Computer Science, Princeton University
agu@cs.cmu.edu, tri@tridao.me

Transformers as Universal over Domains







[https://magenta.tensorflow.org/music-transformer, https://www.kaggle.com/piantic/vision-transformer-vit-visualize-attention-map, https://neuravest.net/how-transformers-with-attention-networks-boost-time-series-forecasting/]

Transformers as Universal over Domains

https://www.youtube.com/watch?v=Elxn8rS88bI

Transformers as Universal Approximators

2 Transformer networks

A Transformer block is a sequence-to-sequence function mapping $\mathbb{R}^{d\times n}$ to $\mathbb{R}^{d\times n}$. It consists of two layers: a self-attention layer and a token-wise feed-forward layer, with both layers having a skip connection. More concretely, for an input $\boldsymbol{X} \in \mathbb{R}^{d\times n}$ consisting of d-dimensional embeddings of n tokens, a Transformer block with *multiplicative* or *dot-product* attention [Luong et al., 2015] consists of the following two layers[1]:

$$\mathrm{Attn}(\boldsymbol{X}) = \boldsymbol{X} + \sum_{i=1}^{h} \boldsymbol{W}_O^i \boldsymbol{W}_V^i \boldsymbol{X} \cdot \sigma[(\boldsymbol{W}_K^i \boldsymbol{X})^T \boldsymbol{W}_Q^i \boldsymbol{X}], \tag{1}$$

$$\mathrm{FF}(\boldsymbol{X}) = \mathrm{Attn}(\boldsymbol{X}) + \boldsymbol{W}_2 \cdot \mathrm{ReLU}(\boldsymbol{W}_1 \cdot \mathrm{Attn}(\boldsymbol{X}) + \boldsymbol{b}_1 \boldsymbol{1}_n^T) + \boldsymbol{b}_2 \boldsymbol{1}_n^T, \tag{2}$$

where $\boldsymbol{W}_O^i \in \mathbb{R}^{d\times m}, \boldsymbol{W}_V^i, \boldsymbol{W}_K^i, \boldsymbol{W}_Q^i \in \mathbb{R}^{m\times d}, \boldsymbol{W}_2 \in \mathbb{R}^{d\times r}, \boldsymbol{W}_1 \in \mathbb{R}^{r\times d}, \boldsymbol{b}_2 \in \mathbb{R}^d, \boldsymbol{b}_1 \in \mathbb{R}^r$, and $\mathrm{FF}(\boldsymbol{X})$ is the output of the Transformer block. The number of heads h and the head size m are two main parameters of the attention layer; and r denotes the hidden layer size of the feed-forward layer.

Notation. Given a matrix $\boldsymbol{A}$, we use $\|\boldsymbol{A}\|_p$ to denote the entry-wise ℓ^p norm of $\boldsymbol{A}$. Let $\sigma[\cdot]$ be the softmax operator, which takes a matrix as input and applies softmax operation to each column of the matrix, which results in a column stochastic matrix, i.e., a matrix that has non-negative entries with all columns summing to 1. We use $\boldsymbol{1}_n$ to denote a vector of length n whose entries are all 1. We use d and n to denote the embedding dimension and the sequence length, respectively. We assume throughout that $n \geq 2$, as the Transformers reduce to residual networks when $n = 1$.

Transformers as Universal Approximators

We define the Transformer networks as the composition of Transformer blocks. The family of the sequence-to-sequence functions corresponding to the Transformers can be defined as:

$$\mathcal{T}^{h,m,r} := \{g : \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n} \mid g \text{ is a composition of Transformer blocks } t^{h,m,r}\text{'s}\}. \quad (3)$$

where $t^{h,m,r} : \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n}$ denotes a Transformer block defined by an attention layer with h heads of size m each, and a feed-forward layer with r hidden nodes.

We say that a function $f : \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n}$ is *permutation equivariant* if for any permutation matrix $\boldsymbol{P}$, we have $f(\boldsymbol{X}\boldsymbol{P}) = f(\boldsymbol{X})\boldsymbol{P}$; i.e., if we permute the columns of $\boldsymbol{X}$, then the columns of $f(\boldsymbol{X})$ are permuted in the same way. A Transformer block is permutation equivariant, which we formally prove in Section B. This consequently establishes the permutation equivariance of the class $\mathcal{T}^{h,m,r}$.

Claim 1. *A Transformer block $t^{h,m,r}$ defines a permutation equivariant map from $\mathbb{R}^{d\times n}$ to $\mathbb{R}^{d\times n}$.*

Transformers as Universal Approximators

3 Transformers are universal approximators of seq-to-seq functions

In this section, we present our results showing that the Transformer networks are universal approximators of sequence-to-sequence functions. Let us start by defining the target function class $\mathcal{F}_{\mathrm{PE}}$, which consists of all continuous permutation equivariant functions with compact support that map $\mathbb{R}^{d\times n}$ to $\mathbb{R}^{d\times n}$. Here, continuity is defined with respect to any entry-wise ℓ^p norm, $1 \leq p < \infty$. Given two functions $f_1, f_2 : \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n}$, for $1 \leq p < \infty$, we define a distance between them as

$$\mathsf{d}_p(f_1, f_2) := \Big(\int \|f_1(\boldsymbol{X}) - f_2(\boldsymbol{X})\|_p^p \, d\boldsymbol{X} \Big)^{1/p}.$$

The following result shows that a Transformer network with a constant number of heads h, head size m, and hidden layer of size r can approximate any function in $\mathcal{F}_{\mathrm{PE}}$.

Theorem 2. *Let $1 \leq p < \infty$ and $\epsilon > 0$, then for any given $f \in \mathcal{F}_{\mathrm{PE}}$, there exists a Transformer network $g \in \mathcal{T}^{2,1,4}$, such that $\mathsf{d}_p(f, g) \leq \epsilon$.*

Transformers as Universal Approximators

3.1 Transformers with trainable positional encodings

In order to endow the Transformer networks with the ability to capture the information about the position of tokens in the input sequence, it is a common practice to add positional encodings $\boldsymbol{E} \in \mathbb{R}^{d\times n}$ to the input sequence before feeding it to the Transformer network [Vaswani et al., 2017, Devlin et al., 2018]. Consider the functions represented by Transformers with positional encodings:

$$\mathcal{T}_{\mathrm{P}}^{h,m,r} := \{g_{\mathrm{P}}(\boldsymbol{X}) = g(\boldsymbol{X}+\boldsymbol{E}) \mid g \in \mathcal{T}^{h,m,r} \text{ and } \boldsymbol{E} \in \mathbb{R}^{d\times n}\}. \tag{4}$$

Here we show that if $\boldsymbol{E}$ is trainable, these positional encodings are sufficient to remove the permutation equivariance restriction of the Transformers. Towards this, we define $\mathcal{F}_{\mathrm{CD}}$ to be the set of all continuous functions that map a compact domain in $\mathbb{R}^{d\times n}$ to $\mathbb{R}^{d\times n}$. Note that $\mathcal{F}_{\mathrm{CD}}$ does not have the restriction of permutation equivariance as in $\mathcal{F}_{\mathrm{PE}}$, but any $f \in \mathcal{F}_{\mathrm{CD}}$ is defined on a compact domain instead of the whole $\mathbb{R}^{d\times n}$. The following result states that, equipped with the trainable positional encodings, Transformers can approximate any sequence-to-sequence function in $\mathcal{F}_{\mathrm{CD}}$.

Theorem 3. *Let $1 \le p < \infty$ and $\epsilon > 0$, then for any given $f \in \mathcal{F}_{\mathrm{CD}}$, there exists a Transformer network $g \in \mathcal{T}_{\mathrm{P}}^{2,1,4}$ such that we have $\mathrm{d}_p(f,g) \le \epsilon$.*

Transformers as Universal Approximators

4 Conclusion

In this paper, we prove that Transformer networks are universal approximators of any continuous and permutation equivariant sequence-to-sequence functions, which shed light on the expressive power of Transformer networks. We also theoretically validate the use of additive positional encodings in Transformers, as they can remove the permutation equivaraince restriction and make Transformers universal approximators of arbitrary continuous sequence-to-sequence functions.

In the supplementary material, we present the proofs of our theorems, which reveal that self-attention layers in Transformer networks can compute *contextual mappings*; this is one of the crucial components that make Transformer networks universal. We also discuss and experiment with other simpler layers that can implement weaker forms of contextual mappings.

Transformers as Universal Approximators

C Proof of Theorem 2

Recall that we want to show that given a function $f \in \mathcal{F}_{\mathrm{PE}}$, we can find a Transformer network $g \in \mathcal{T}^{2,1,4}$ such that $\mathsf{d}_p(f, g) \leq \epsilon$. Without loss of generality, we can assume that the compact support of f is contained in $[0, 1]^{d \times n}$. We achieve our desired objective in three key steps:

Step 1. Approximate $\mathcal{F}_{\mathrm{PE}}$ with piece-wise constant functions. We first use (a variant of) the classical result that any continuous function can be approximated up to arbitrary accuracy by piece-wise constant functions. For $\delta > 0$, we define the following class of piece-wise constant functions.

$$\overline{\mathcal{F}}_{\mathrm{PE}}(\delta) := \left\{ f : \boldsymbol{X} \mapsto \sum\nolimits_{\boldsymbol{L} \in \mathbb{G}_\delta} \boldsymbol{A}_{\boldsymbol{L}} \mathbb{1}\left\{\boldsymbol{X} \in \mathbb{S}_{\boldsymbol{L}}\right\} \mid f \text{ is permutation equivariant, } \boldsymbol{A}_{\boldsymbol{L}} \in \mathbb{R}^{d \times n} \right\},$$

where $\mathbb{G}_\delta := \{0, \delta, \ldots, 1 - \delta\}^{d \times n}$ and, for a grid point $\boldsymbol{L} \in \mathbb{G}_\delta$, $\mathbb{S}_{\boldsymbol{L}} := \prod_{j=1}^{d} \prod_{k=1}^{n} [L_{j,k}, L_{j,k} + \delta) \subset [0, 1]^{d \times n}$ denotes the associated cube of width δ.

The following result states that the underlying function $f \in \mathcal{F}_{\mathrm{PE}}$ can be approximated using the function class $\overline{\mathcal{F}}_{\mathrm{PE}}(\delta)$.

Lemma 4. *For any given $f \in \mathcal{F}_{\mathrm{PE}}$ and $1 \leq p < \infty$, one can find a $\delta^* > 0$ such that $\exists\, \overline{f} \in \overline{\mathcal{F}}_{\mathrm{PE}}(\delta^*)$ which satisfies $\mathsf{d}_p(f, \overline{f}) \leq \epsilon/3$.*

Transformers as Universal Approximators

Step 2. Approximate $\overline{\mathcal{F}}_{\mathrm{PE}}(\delta)$ with *modified* Transformers. We then consider a slightly modified architecture for Transformer networks, where the softmax operator $\sigma[\cdot]$ and $\mathrm{ReLU}(\cdot)$ are replaced by the hardmax operator $\sigma_{\mathrm{H}}[\cdot]$ and an activation function $\phi \in \Phi$, respectively. Here, the set of allowed activations Φ consists of all piece-wise linear functions with at most three pieces, where at least one piece is constant. Let $\overline{\mathcal{T}}^{h,m,r}$ denote the function class corresponding to the sequence-to-sequence functions defined by the modified Transformer networks. The following result establishes that the modified Transformer networks in $\overline{\mathcal{T}}^{2,1,1}$ can closely approximate functions in $\overline{\mathcal{F}}_{\mathrm{PE}}(\delta)$.

Proposition 5. *For each* $\overline{f} \in \overline{\mathcal{F}}_{\mathrm{PE}}(\delta)$ *and* $1 \leq p < \infty$, $\exists\, \overline{g} \in \overline{\mathcal{T}}^{2,1,1}$ *such that* $\mathrm{d}_p(\overline{f}, \overline{g}) = O(\delta^{d/p})$.

Transformers as Universal Approximators

Step 3. Approximate modified Transformers with (original) Transformers. Finally, we show that $\overline{g} \in \overline{\mathcal{T}}^{2,1,1}$ can be approximated by $\mathcal{T}^{2,1,4}$. Let $g \in \mathcal{T}^{2,1,4}$ be such that $\mathsf{d}_p(\overline{g}, g) \leq \epsilon/3$.

The following result relies on the connection between the softmax operator and the hardmax operator; and the fact each activation $\phi \in \Phi$ can approximated by the sum of four ReLU's.

Lemma 6. *For each* $\overline{g} \in \overline{\mathcal{T}}^{2,1,1}$ *and* $1 \leq p < \infty$, $\exists\, g \in \mathcal{T}^{2,1,4}$ *such that* $\mathsf{d}_p(\overline{g}, g) \leq \epsilon/3$.

Theorem 2 now follows from these three steps, because we have

$$\mathsf{d}_p(f, g) \leq \mathsf{d}_p(f, \overline{f}) + \mathsf{d}_p(\overline{f}, \overline{g}) + \mathsf{d}_p(\overline{g}, g) \leq 2\epsilon/3 + O(\delta^{d/p}).$$

Choosing $\delta \leq \delta^*$ small enough ensures that $\mathsf{d}_p(f, g) \leq \epsilon$. □

Transformers as Universal Predictors?

1258 IEEE TRANSACTIONS ON INFORMATION THEORY VOL. 38, NO. 4, JULY 1992

Universal Prediction of Individual Sequences

Meir Feder, *Member, IEEE,* Neri Merhav, *Member, IEEE,* and
Michael Gutman, *Member, IEEE*

1506 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 50, NO. 7, JULY 2004

Finite-Memory Universal Prediction of Individual Sequences

Eado Meron and Meir Feder, *Fellow, IEEE*

Transformers as Universal Predictors?

Transformers are Universal Predictors

Sourya Basu [1 2] **Moulik Choraria** [1 2] **Lav R. Varshney** [1 2]

Batch Universal Prediction

Marco Bondaschi and Michael Gastpar
School of Computer and Communication Sciences
EPFL
Switzerland
Email: {marco.bondaschi, michael.gastpar}@epfl.ch

Allometric Scaling

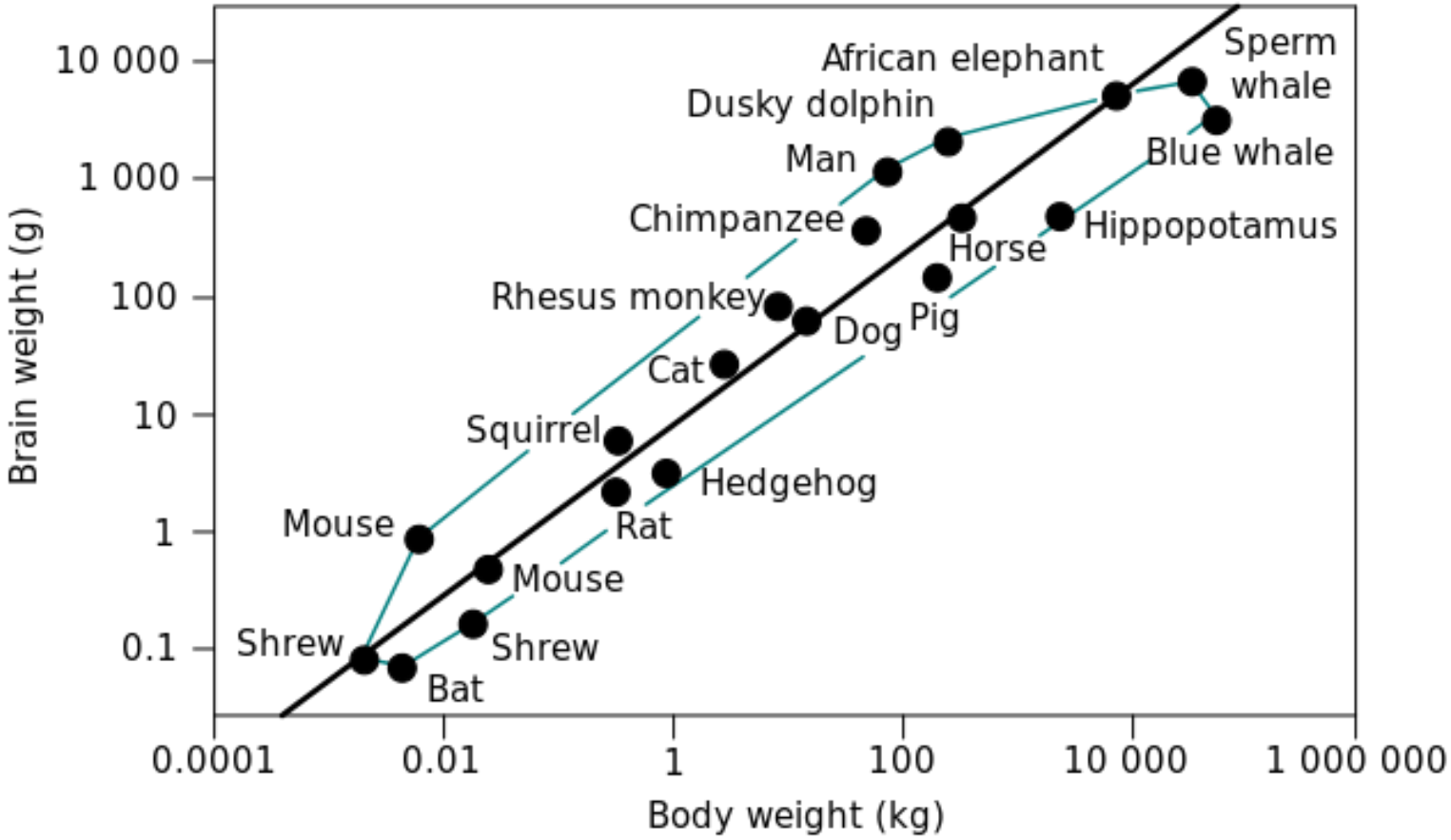
- Allometry studies the relationship between body size to shape. Goes back to D'Arcy Thompson's *On Growth and Form* (1917)
- In neurobiology, one can look at allometric scaling relationships:
  - across different species with similar brain architectures [evolution],
  - scaling relationships for different individuals of same species [growth],
  - properties of the brain within the same individual [structure]
- The relationship between the two measured quantities is usually expressed as a power law equation:

$$y = kx^{\alpha}$$

where α is the scaling exponent of the law.

- How should we interpret superlinear ($\alpha > 1$) or sublinear ($\alpha < 1$) scaling?

Allometric Scaling



Encephalization quotient

$E = CS^2$, where E and S are body and brain weights

[https://en.wikipedia.org/wiki/Brain-to-body_mass_ratio]

Allometric Scaling

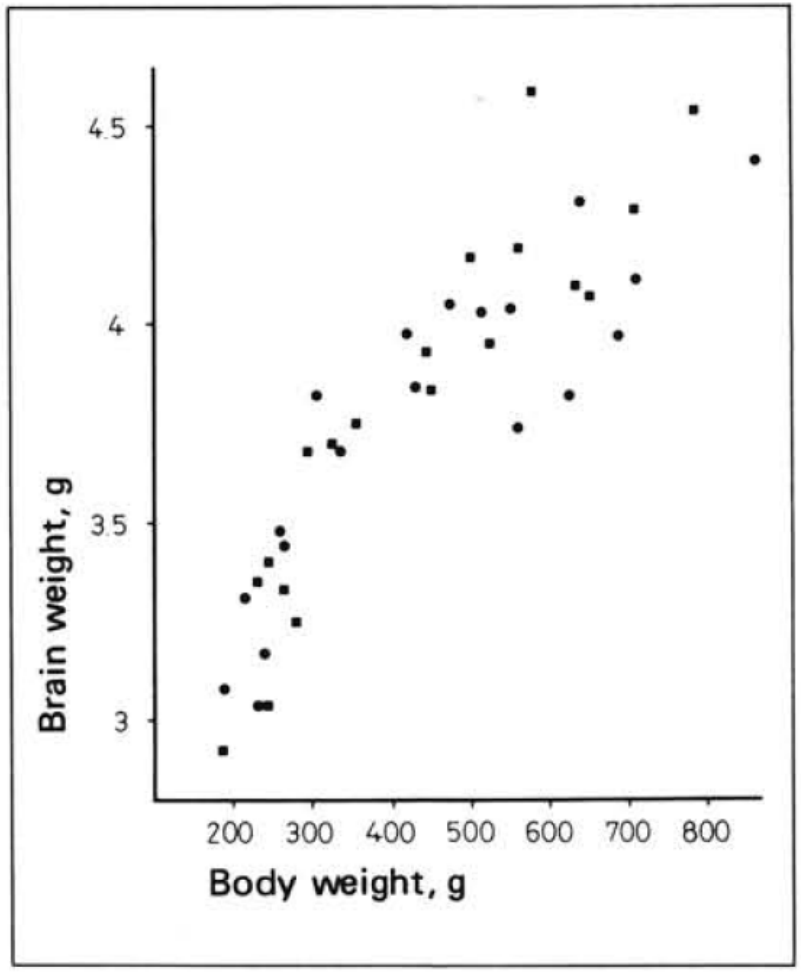
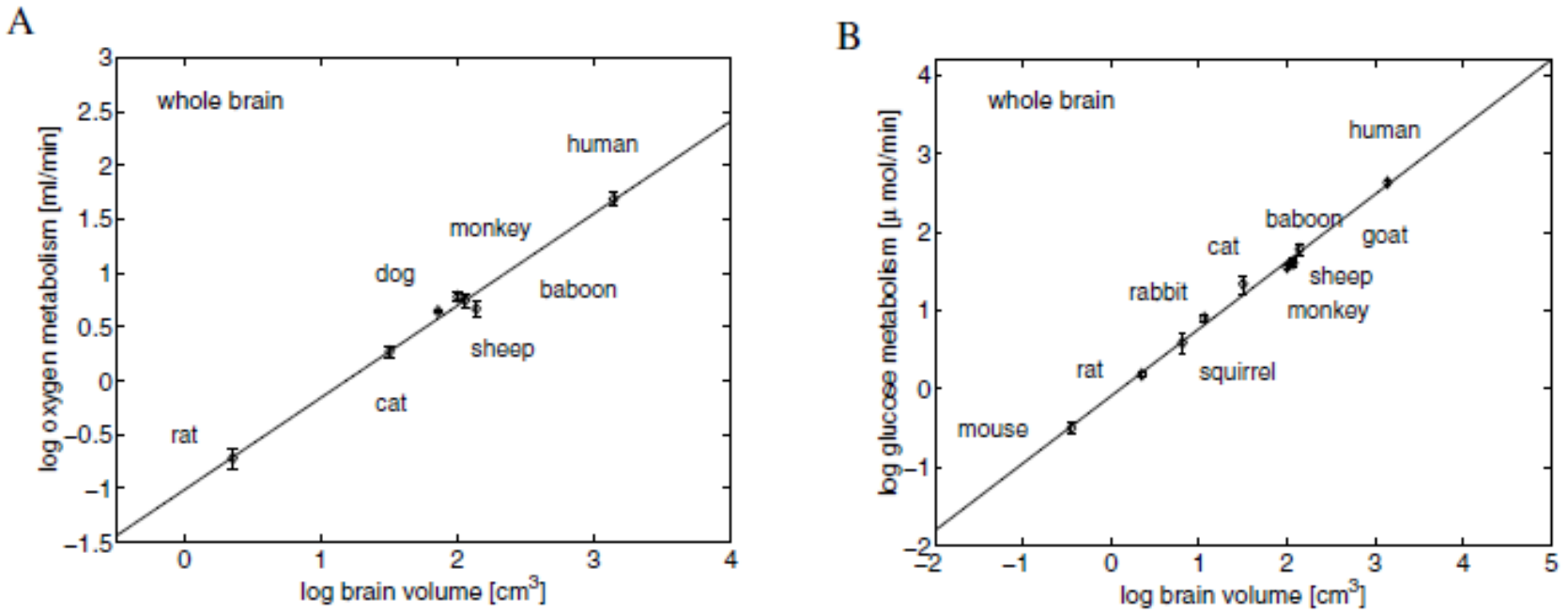


Fig. 1. Brain weights of guinea pigs *(Cavia cobaya)*

[Heinz Stephan, Heiko Frahm, and Georg Baron, "New and Revised Data on Volumes of Brain Structures in Insectovores and Primates," Folia Primatol., vol. 35, pp. 1-29, 1981.]

Allometric Scaling



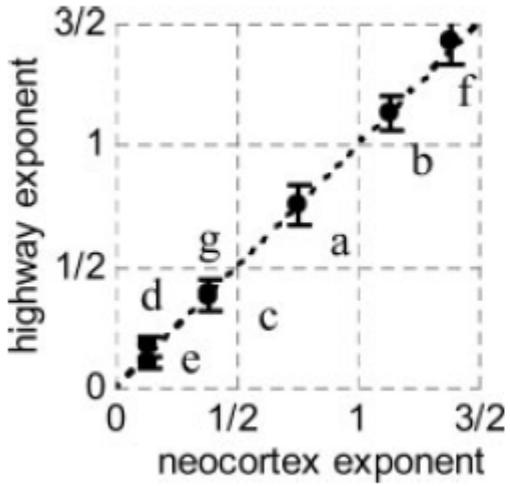
Scaling of the total basal cerebral metabolism with brain volume. The least-square fit line for the log – log plot yields the following. (**A**) For the total oxygen consumption
rate, the scaling exponent was 0.86 ± 0.04 ($y = 0.86x - 1.02$, $R^2 = 0.989$, $p < 10^{-4}$, $n = 7$), and its 95% confidence interval was 0.75 to 0.96. (**B**) For the total glucose utilization rate, an identical exponent 0.86 ± 0.03 was found ($y = 0.86x - 0.09$, $R^2 = 0.994$, $p < 10^{-4}$, $n = 10$) and its 95% confidence interval was 0.80 to 0.91.

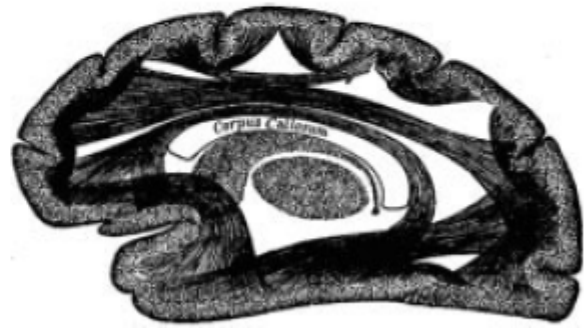
[Jan Karbowski, "Global and regional brain metabolic scaling and its functional consequences," *BMC Biology*, 2007, 5:18.]

Are there common allometric scalings among different kinds of networks?

Are there common allometric scalings among different kinds of networks?

Comparison of City Highway System and Neocortex Exponents for Quantities as a Function of Surface Area





Generic Name	Variable for City Highways	City Highway System Exponent	Variable for Neocortex	Neocortex Exponent
Surface area	Land area	1	Total convoluted surface area	1
(a) No. of conduits	No. of highways	0.759 (±0.083)	No. of pyramidal neurons	3/4 = 0.75
(b) Total no. of leaves	Total no. exits	1.138 (±0.072)	Total no. of synapses	9/8 = 1.125
(c) No. of leaves per conduit	No. of exits per highway	0.379 (±0.064)	No. of synapses per neuron	3/8 = 0.375
(d) Diameter of conduit	No. of highway lanes	0.174 (±0.038)	Diameter of white matter axon	1/8 = 0.125
(e) Propagation velocity	Velocity of cross-city travel	0.108 (±0.021)	Propagation velocity of white matter axon	1/8 = 0.125
(f) Total surface area of conduits	Total surface of highways	1.433 (±0.096)	Total surface area of white matter axons	11/8 = 1.375
	Population	1.462 (±0.141)		
			Total volume of white matter axons	3/2 = 1.5
(g) No. of compartments	No. of concentric ring regions	0.390 (±0.055)	No. of cortical areas	3/8 = 0.375

A General Model for the Origin of Allometric Scaling Laws in Biology

Geoffrey B. West, James H. Brown,* Brian J. Enquist

Allometric scaling relations, including the 3/4 power law for metabolic rates, are characteristic of all organisms and are here derived from a general model that describes how essential materials are transported through space-filling fractal networks of branching tubes. The model assumes that the energy dissipated is minimized and that the terminal tubes do not vary with body size. It provides a complete analysis of scaling relations for mammalian circulatory systems that are in agreement with data. More generally, the model predicts structural and functional properties of vertebrate cardiovascular and respiratory systems, plant vascular systems, insect tracheal tubes, and other distribution networks.

The Origins of Scaling in Cities

Luís M. A. Bettencourt

Despite the increasing importance of cities in human societies, our ability to understand them scientifically and manage them in practice has remained limited. The greatest difficulties to any scientific approach to cities have resulted from their many interdependent facets, as social, economic, infrastructural, and spatial complex systems that exist in similar but changing forms over a huge range of scales. Here, I show how all cities may evolve according to a small set of basic principles that operate locally. A theoretical framework was developed to predict the average social, spatial, and infrastructural properties of cities as a set of scaling relations that apply to all urban systems. Confirmation of these predictions was observed for thousands of cities worldwide, from many urban systems at different levels of development. Measures of urban efficiency, capturing the balance between socioeconomic outputs and infrastructural costs, were shown to be independent of city size and might be a useful means to evaluate urban planning strategies.

Scaling Laws for Neural Language Models

Jared Kaplan *
Johns Hopkins University, OpenAI
jaredk@jhu.edu

Sam McCandlish*
OpenAI
sam@openai.com

Tom Henighan
OpenAI
henighan@openai.com

Tom B. Brown
OpenAI
tom@openai.com

Benjamin Chess
OpenAI
bchess@openai.com

Rewon Child
OpenAI
rewon@openai.com

Scott Gray
OpenAI
scott@openai.com

Alec Radford
OpenAI
alec@openai.com

Jeffrey Wu
OpenAI
jeffwu@openai.com

Dario Amodei
OpenAI
damodei@openai.com

We study empirical scaling laws for language model performance on the cross-entropy loss. The loss scales as a power-law with model size, dataset size, and the amount of compute used for training, with some trends spanning more than seven orders of magnitude. Other architectural details such as network width or depth have minimal effects within a wide range.

Scaling Laws for Transfer

Danny Hernandez*

Jared Kaplan†‡ Tom Henighan† Sam McCandlish†

When we train increasingly large neural networks from-scratch on a fixed-size dataset, they eventually become data-limited and stop improving in performance (cross-entropy loss). When we do the same for models pre-trained on a large language dataset, the slope in performance gains is merely reduced rather than going to zero. We calculate the effective data “transferred” from pre-training by determining how much data a transformer of the same size would have required to achieve the same loss when training from scratch. In other words, we focus on units of data while holding everything else fixed. We find that the effective data transferred is described well in the low data regime by a power-law of parameter count and fine-tuning dataset size. We believe the exponents in these power-laws correspond to measures of the generality of a model and proximity of distributions (in a directed rather than symmetric sense). We find that pre-training effectively multiplies the fine-tuning dataset size.

Scaling Laws Under the Microscope: Predicting Transformer Performance from Small Scale Experiments

Maor Ivgi
Tel-Aviv University

Yair Carmon
Tel-Aviv University

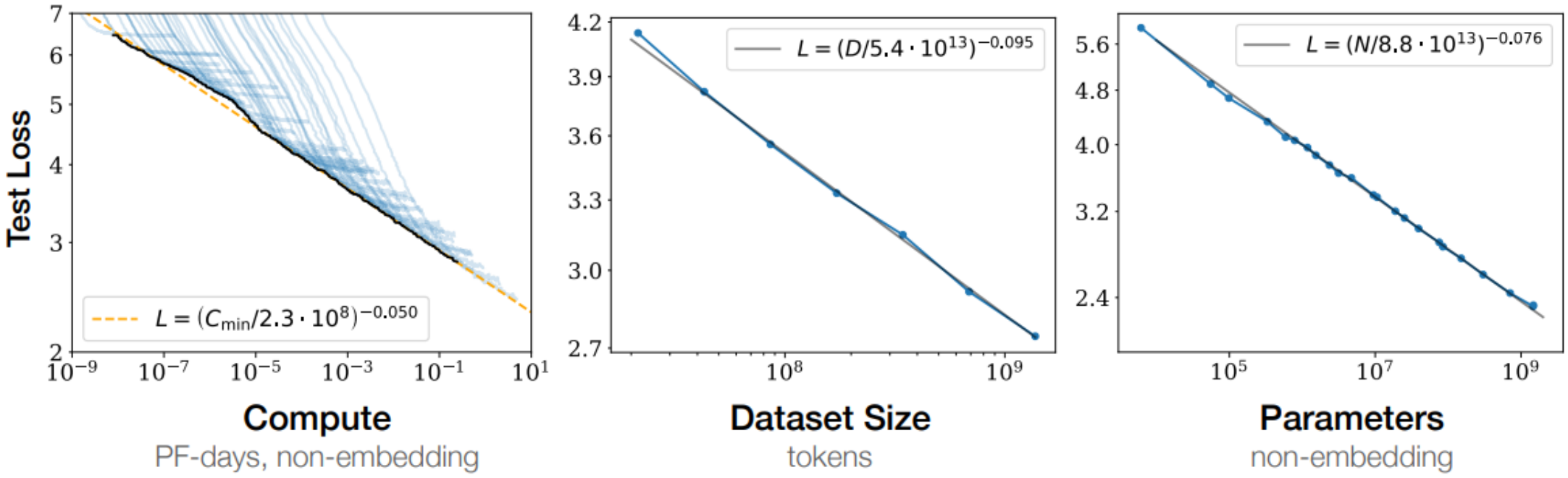
Jonathan Berant
Tel-Aviv University

Neural scaling laws define a predictable relationship between a model's parameter count and its performance after training in the form of a power law. However, most research to date has not explicitly investigated whether scaling laws can be used to accelerate model development. In this work, we perform such an empirical investigation across a wide range of language understanding tasks, starting from models with as few as 10K parameters, and evaluate downstream performance across 9 language understanding tasks. We find that scaling laws emerge at finetuning time in some NLP tasks, and that they can also be exploited for debugging convergence when training large models. Moreover, for tasks where scaling laws exist, they can be used to predict the performance of larger models, which enables effective model selection.

Scaling Laws for Transformers

Model performance depends most strongly on scale, which consists of three factors: the number of model parameters N (excluding embeddings), the size of the dataset D, and the amount of compute C used for training. Within reasonable limits, performance depends very weakly on other architectural hyperparameters such as depth vs. width.

Performance has a power-law relationship with each of the three scale factors N, D, C when not bottlenecked by the other two, with trends spanning more than six orders of magnitude



https://arxiv.org/pdf/2001.08361.pdf

Scaling Laws for Transformers

- **Universality of overfitting**: Performance improves predictably as long as we scale up N and D in tandem, but enters a regime of diminishing returns if either N or D is held fixed while the other increases. The performance penalty depends predictably on the ratio $N^{0.74}/D$, meaning that every time we increase the model size 8x, we only need to increase the data by roughly 5x to avoid a penalty.

[Biological ¾ law?]

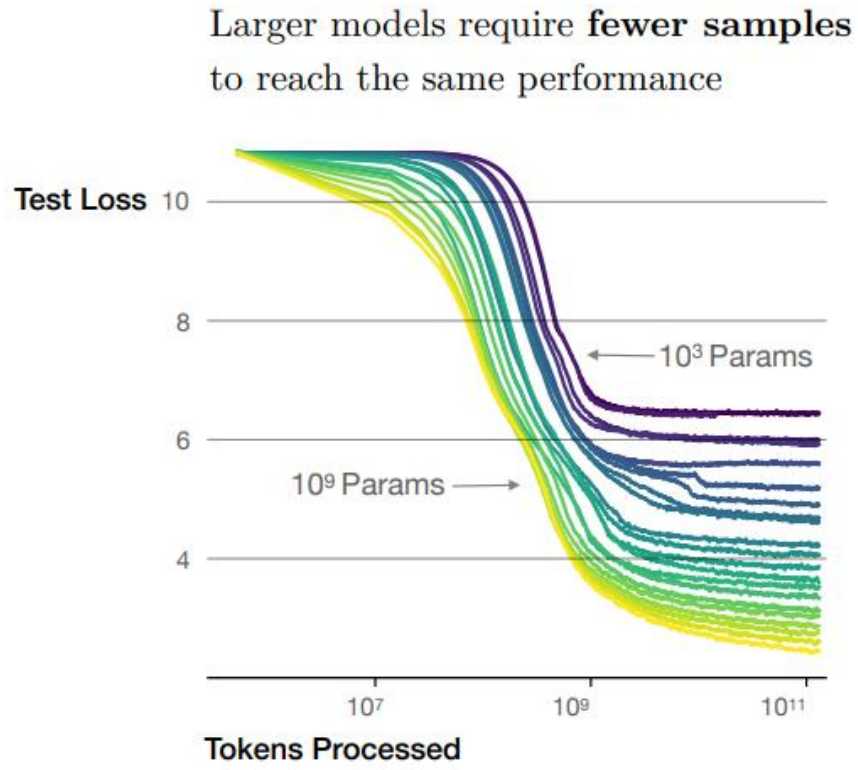
- **Universality of training**: Training curves follow predictable power-laws whose parameters are roughly independent of the model size. By extrapolating the early part of a training curve, we can roughly predict the loss that would be achieved if we trained for much longer.

- **Transfer improves with test performance**: When we evaluate models on text with a different distribution than they were trained on, the results are strongly correlated to those on the training validation set with a roughly constant offset in the loss – in other words, transfer to a different distribution incurs a constant penalty but otherwise improves roughly in line with performance on the training set.

https://arxiv.org/pdf/2001.08361.pdf

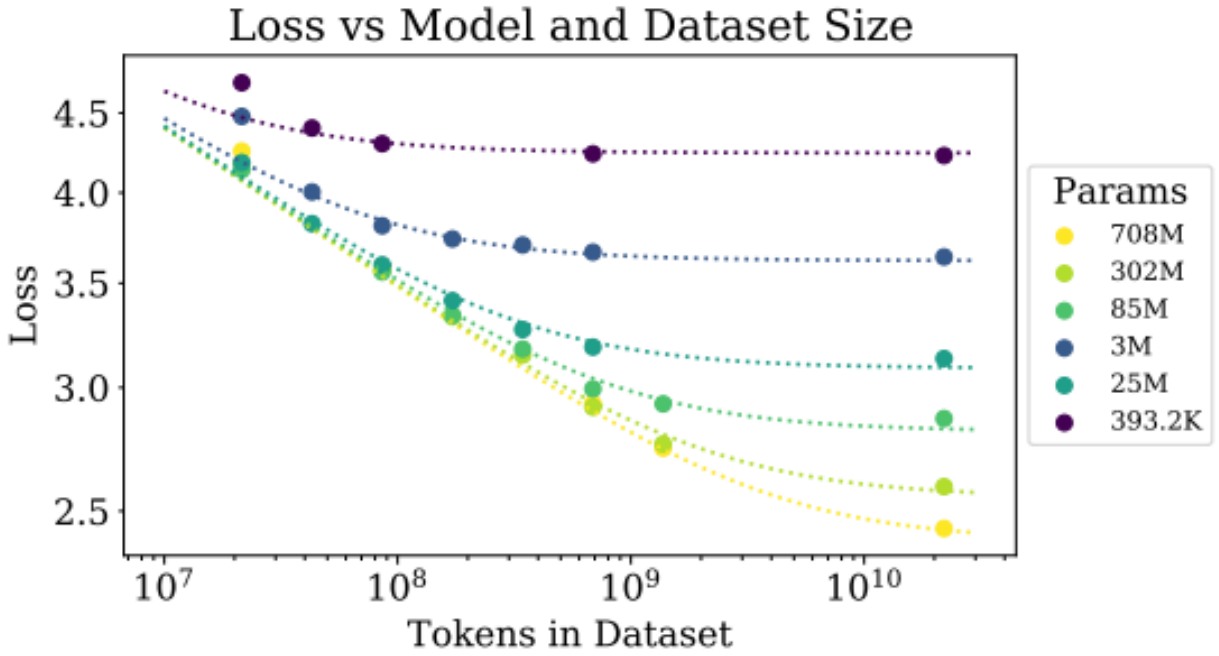
Scaling Laws for Transformers

- **Sample efficiency**: Large models are more sample-efficient than small models, reaching the same level of performance with fewer optimization steps and using fewer data points.



- **Convergence is inefficient**: When working within a fixed compute budget C but without any other restrictions on the model size N or available data D, we attain optimal performance by training very large models and stopping significantly short of convergence. Maximally compute-efficient training would therefore be far more sample efficient than one might expect based on training small models to convergence, with data requirements growing very slowly as $D \sim C^{0.27}$ with training compute.

https://arxiv.org/pdf/2001.08361.pdf



$$\alpha_N \sim 0.076,$$
$$\alpha_D \sim 0.095,$$

as we increase the model size, we should increase the dataset size sublinearly according to $D \propto N^{\{\alpha_N/\alpha_D\}} \sim N^{0.74}$

$$L(N, D) = \left[\left(\frac{N_c}{N}\right)^{\frac{\alpha_N}{\alpha_D}} + \frac{D_c}{D}\right]^{\alpha_D} \tag{1.5}$$

with fits pictured on the left in figure 4. We conjecture that this functional form may also parameterize the trained log-likelihood for other generative modeling tasks.