Emergence.

Tolga Birdal

I tell you four numbers:

1. The first number is 2.
2. The second number is 4.
3. The third number is 6.
4. The fourth number is 8.

What is the fifth number? Because the numbers are generated by:

$$n^4 - 10n^3 + 35n^2 - 48n + 24, \tag{1}$$

the fifth number is 34. Can this really be true? Let's look deeper. Suppose that I tossed a coin 40 times and obtained:

$$1010101010101010101010101010101010101010 \tag{2}$$

Do you believe me? Suppose instead that I tossed a coin 40 times and obtained:

$$11101010101001010100111001010010111100010 \tag{3}$$

Do you believe me now?

In classical probability two strings have the same probability:

$$p(1010101010101010101010101010101010101010) = 2^{-40} \tag{4}$$

$$p(11101010101001010100111001010010111100010) = 2^{-40} \tag{5}$$

In other words, classical probability theory does not capture intuitive notion of "random". Complex outputs are harder to generate by random sampling of inputs than simpler ones are. This is explained by **Occam's Razor**:

Pluralitas non est ponenda sine necessitate.
Plurality is not to be posited without necessity.

When faced with two or more possible explanations, the simplest is the one most likely to be true.

But, what is simplicity precisely? According to **Kolmogorov**: "Simplicity is **Algorithmic Complexity**":

> The epistemological value of probability theory is based on the fact that chance phenomena, considered collectively and on a grand scale, create non-random regularity.

Let us momentarily admit the following definitions:

Definition 1 (Program)**.** *A model of a dataset is a **program** that generates the dataset efficiently, i.e. succinctly.*

Definition 2 (Cognitive system)**.** *A cognitive system or agent is a model-building **semi-isolated** computational system controlling some of its couplings/information interfaces with **the rest of the universe and driven by an internal optimization function**.*

Definition 3 (Kolmogorov(–Chaitin) Complexity $K(X)$)**.** *Let X be a binary string and U be a **universal Turing computer/machine** (UTM). Denote by $\ell(X)$ the length of X. The Kolmogorov complexity of X with respect to U is defined as the length of the shortest input program P that generates output Y when it is fed into a computer:*

$$K_U(X) = \min_{\ell : U(P)=X} \ell(P). \tag{6}$$

For example, one can express the string in eq. (2) as:

$$1010101010101010101010101010101010101010 \equiv \text{Repeat “10” 20 times.} \tag{7}$$

How about writing a compact program for the other:

$$1110101010100101010011100101001011100010 \equiv \,? \tag{8}$$

As you notice by now:

Random sequences have **maximum complexity**: a random sequence can have no generating algorithm shorter than simply listing the sequence.

Unfortunately, **K(X) is uncomputable**.

Definition 4 ((In)compressibility)**.** *A string x is incompressible if:*

$$K(X) \geq \ell(x). \tag{9}$$

Definition 5 (Church's thesis)**.** *All non-trivial computational models (UTMs) are equivalent; thus, one can consider any universal computer (which can simulate other computers):*

$$K_{U_1}(X) \approx K_{U_2}(X) \tag{10}$$

for arbitrary UTMs, U_1 and U_2. In fact:

$$|K_{U_1}(X) - K_{U_2}(X)| \leq C. \tag{11}$$

for some C that is independent from X. This suggests that the Kolmogorov complexity is an intrinsic property of the string. a string.

All these beg for a re-definition of probability to take into account the notion of 'simplicity':

Definition 6 (Algorithmic probability)**.** *The probability of a given string being produced by a random program:*

$$p(X) = 2^{-K(X)}. \tag{12}$$

Definition 7 (The Wisdom of Crowds)**.** *The idea is that if you take a bunch of very different kinds of people and ask them (independently) for a solution to a difficult problem, then a suitable average of their solutions will very often be better than the best in the set.*

Kolmogorov was a very important mathematician. He was a man of nature and instrumental in building modern foundations of probability. Even modern diffusion theory stems from his work [1]. Vladimir Arnold once said: "Kolmogorov – Poincaré – Gauss – Euler – Newton, are only five lives separating us from the source of our science."

1. Statistical Learning Theory (STL): *Nothing is more practical than a good theory.*

Definition 8 (Data)**.** *Data space is denoted by $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, and $\mathcal{X}$ and $\mathcal{Y}$ respectively denote the features and the labels. We assume that the data is generated via an unknown data distribution $\mathcal{D}$ and we have access to a training set of n points, i.e., $S = \{z_1, \dots, z_n\}$, with the samples $\{z_i\}_{i=1}^n$ are independent and identically (i.i.d.) drawn from $\mathcal{D}$.*

Definition 9 (Hypothesis Class)**.** *Let $\mathcal{H}$ denote the space of model hypotheses,* i.e.*, set of predictors. We further parameterize the hypothesis class by $\mathcal{W} \subset \mathbb{R}^d$, that potentially depends on S.*

Definition 10 (Loss function)**.** *To measure the quality of a parameter vector $w \in \mathcal{W}$, we use a loss function $\ell : \mathbb{R}^d \times \mathcal{Z} \mapsto \mathbb{R}_+$, such that $\ell(w, z)$ denotes the loss corresponding to a single data point z. We then denote the population and empirical risks respectively by:*

$$\mathcal{R}(w) := \mathbb{E}_z[\ell(w, z)] \qquad \text{(Population Risk)}$$

$$\hat{\mathcal{R}}(w, S) := \frac{1}{n} \sum\nolimits_{i=1}^{n} \ell(w, z_i). \qquad \text{(Empirical Risk)}$$

Note that $\mathcal{R}(w)$ involves an uncomputable integral. Possible choices of empirical risk involve:

$$\ell(w, z) = \mathbb{1}[h(x) \neq y] \qquad \text{(0-1 loss)}$$
$$\ell(w, z) = (h(x) - y)^2 \qquad \text{(MSE loss)}$$
$$\ell(w, z) = \max(0, 1 - x \cdot y) \qquad \text{(Hinge loss)}$$
$$\ell(w, z) = -\log(h(x)) \qquad \text{(log-loss)}$$

Definition 11 (Training Algorithm)**.** *We assume that the model is trained via an algorithm $\mathcal{A} : \mathcal{Z} \to \mathcal{H}$, which is a function of two variables, the dataset S and a random variable U that encapsulates the algorithmic randomness,* i.e.*, stochasticity. The algorithm $\mathcal{A}(S)$ returns the entire (random) trajectory of the network weights in the time frame $[0, T]$, such that $[\mathcal{A}(S)]_t = w_t$ being the network weights returned by $\mathcal{A}$ at 'time' t, and t is a continuous iteration index.*

Definition 12 (Generalization Error)**.** *The generalization error is hence defined as*

$$sup_w|\hat{\mathcal{R}}(w, S) - \mathcal{R}(w)| \quad or \quad \mathbb{E}|\hat{\mathcal{R}}(w, S) - \mathcal{R}(w)|. \tag{13}$$

Definition 13 (Risk Decomposition)**.** *Introduce the best function $h^\star \in \mathcal{H}$ parameterized by $w^\star$, which attains a risk of*

$$R(w^\star) = \inf_{h \in \mathcal{H}} R(w). \tag{14}$$

Why, from the unimaginably large set of functions that give zero error on S, do DNNs converge on a minuscule subset of functions that generalize well?

Proposition 1. *With probability at least $1 - \delta$, the population is risk bounded by [2]:*

$$\mathcal{R}(w) \leq \hat{\mathcal{R}}(w, S) + \sqrt{\frac{K(\mathcal{H}) + \log(1/\delta)}{2n}}. \tag{15}$$

where $K(\mathcal{H})$ counts the number of bits needed to specify any hypothesis.

Solutions found by many machine learning models on real datasets are highly compressible, and this reflects their bias for simple low Kolmogorov complexity functions. An Occam's razor like inductive bias towards simple functions, combined with structured data seems to explain this.

Note that, while these types of bounds are well studied in the non-deep learning setting [4], the behaviour of modern DL defies the classical wisdom of STL, and hence begs for a new learning theory [4]. And Occam's razor becomes essential in establishing these new foundations [3].

References

[1] X. Guo, O. Hernández-Lerma, X. Guo, and O. Hernández-Lerma. *Continuous-time Markov decision processes*. Springer, 2009. 2

[2] S. Lotfi, M. Finzi, S. Kapoor, A. Potapczynski, M. Goldblum, and A. G. Wilson. Pac-bayes compression bounds so tight that they can explain generalization. *Advances in Neural Information Processing Systems*, 35:31459–31473, 2022. 3

[3] C. Mingard, H. Rees, G. Valle-Pérez, and A. A. Louis. Do deep neural networks have an inbuilt occam's razor? *arXiv preprint arXiv:2304.06670*, 2023. 3

[4] A. Xu and M. Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. *Advances in Neural Information Processing Systems*, 30, 2017. 3