THE RANDOMNESS OF PRIME ENCODINGS AND THE COMPUTATIONAL COMPLEXITY OF INTEGER FACTORISATION

Aidan Rocke
aidanrocke@gmail.com

June 25, 2021

So far the hypothesis that prime encodings are random has been carefully evaluated using a variation of Yao's next-bit test as well as a primality test for all prime numbers under 10^9. To be precise, given a finite number of integers $X := [1, N] \subset \mathbb{N}$ with binary representation mapped to $Y := X_N \in \{0, 1\}^*$, a function is approximated using a feedforward neural network(f_θ):

$$f_\theta : X \to Y \tag{1}$$

whose ability to correctly evaluate the primality of integers is tested on integers $[N + 1, 2N]$. In agreement with the hypothesis, the true positive rate of such a function does not exceed 50%. This experimental verification is significant because I am implicitly using the universal approximation property of deep neural networks [6].

Now, it is worth pointing out that feedforward neural networks can only simulate computable functions that are polynomial time with respect to their input size. Thus, the hypothesis ought to be refined as follows:

$$\mathbb{E}[K^{\text{poly}}(X_N)] \sim \pi(N) \cdot \ln(N) \sim N \tag{2}$$

where $K^{\text{poly}}(\cdot)$ denotes a universal compressor that can only simulate Turing Machines with polynomial time complexity.

In order to understand the importance of integer factorisation, it is worth making two observations. First, a reasonable candidate for one-way functions is $f(P, Q) = P \cdot Q$ for randomly chosen primes P and Q, and the existence of one-way functions would imply that P != NP. Second, for a neural network to accurately represent a relation between the integers and the prime numbers with strong generalisation ability, it needs to discover an algorithmic formulation of the unique factorisation theorem. That is to say, an algorithmic version of Euclid's proposition that any integer is either prime or measured by some prime number.

If such a relation is approximated by a feedforward neural network, the underlying algorithm would be nothing less than a method for integer factorisation with polynomial time complexity.

References

[1] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions. In Proceedings of the 23rd IEEE Symposium on Foundations of Computer Science, 1982.

[2] Leonid Levin (2003). "The Tale of One-Way Functions". ACM. arXiv:cs.CR/0012023

[3] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron (2016). Deep Learning. MIT Press

[4] Lenstra, Arjen K (1988). "Fast and rigorous factorization under the generalized Riemann hypothesis". Indagationes Mathematicae. 50 (4): 443–454.

[5] Yanyi Liu, Rafael Pass. On One-way Functions and Kolmogorov Complexity. Arxiv. 2020.

[6] Hornik, Kurt (1991). "Approximation capabilities of multilayer feedforward networks". Neural Networks.