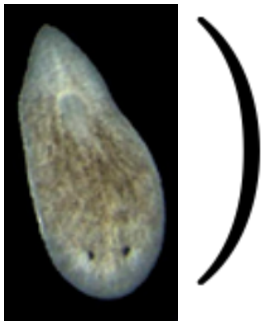
# Universal Approximation Theorem

Why neural networks can approximate any continuous function

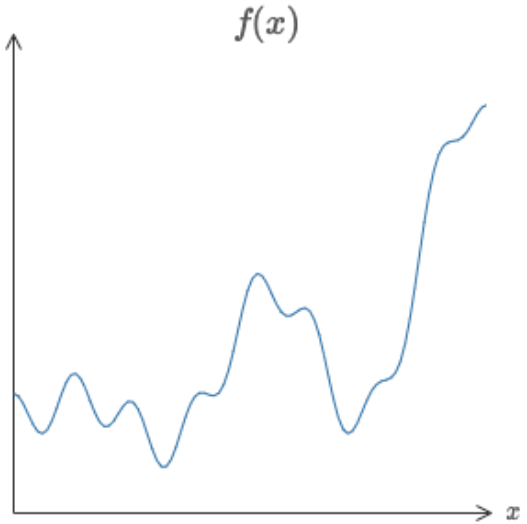
$f\left(\right.$  $\left.\right) =$ “渦虫”

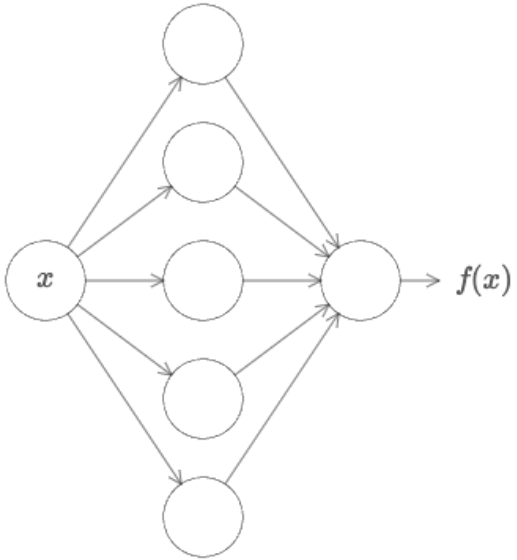
# Overview

- Can a network approximate any continuous function?
- Why UAT is mathematically true.
- If yes, why do we still build deep networks?
- Depth separation & Scaling laws.

# UAT is saying that

"For any continuous function, there exists a single-layer neural network that approximates the function to arbitrary precision."



# Universal Approximation Theorem (Cybenko, 1989)

For every continuous function $f : S \to \mathbb{R}$,

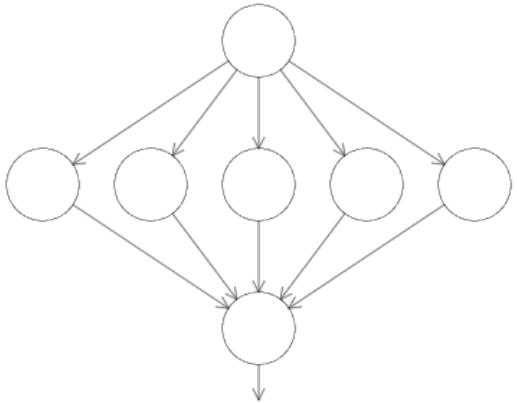
and every $\varepsilon > 0$,

there exists a single-layer neural network

$$G(x) = \sum_{i=1}^{N} a_i \, \sigma\left(w_i \cdot x + b_i\right)$$

such that $|G(x) - f(x)| < \varepsilon$

for every $x \in S$.



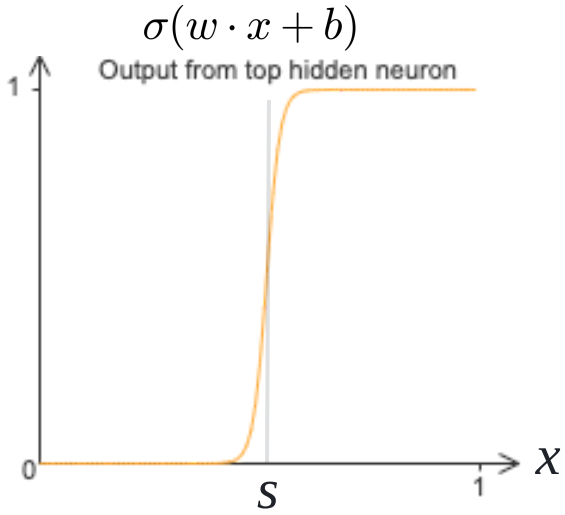
The visual constructive proof follows Michael Nielsen (2015)

Goal: To construct a step-function from sigmoid.

as $|w| \to \infty, |b| \to \infty$

$$s = -b/w$$

$$\sigma(w \cdot x + b) \approx \begin{cases} 1, & x > s, \\ 0, & x \leq s. \end{cases}$$

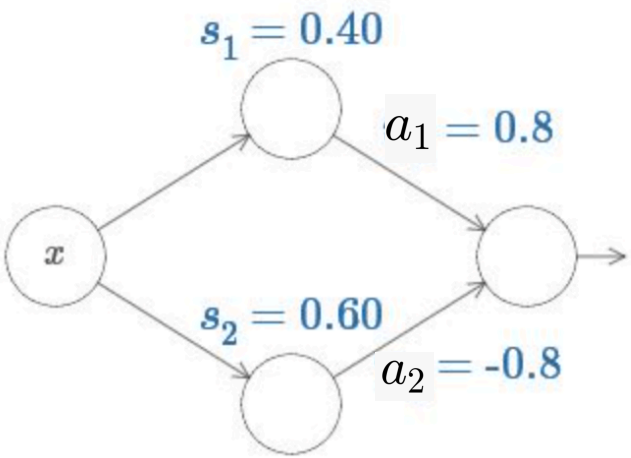


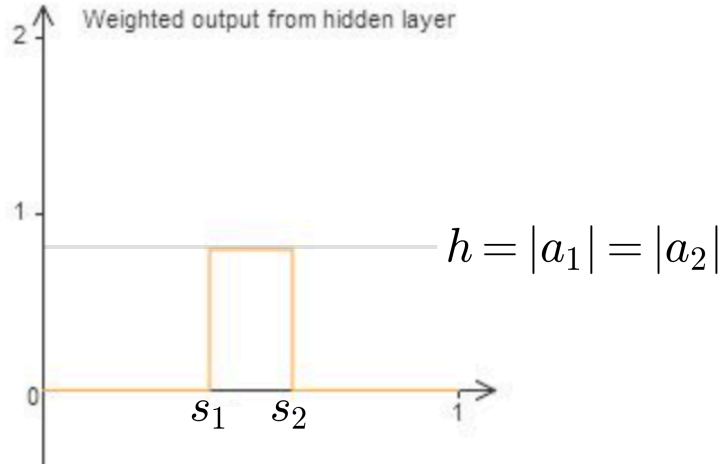

*Adapted from Nielsen (2019), Chapter 4.

The visual constructive proof follows Michael Nielsen (2015)

Goal: To construct a 'bump' from two step functions

$$h(\sigma(s_1) - \sigma(s_2)) = \text{Bump}(s1, s2, h)$$

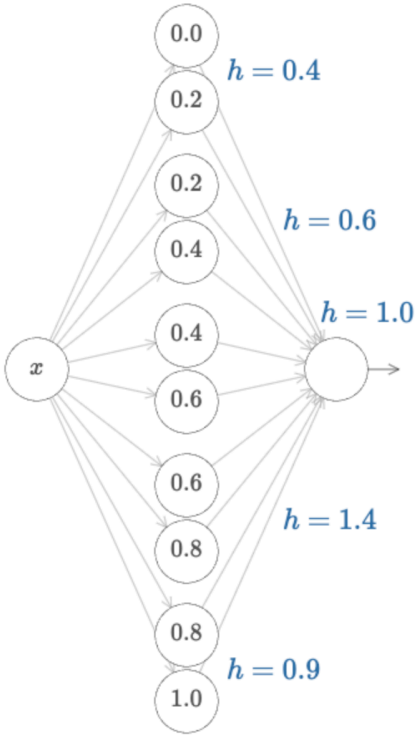


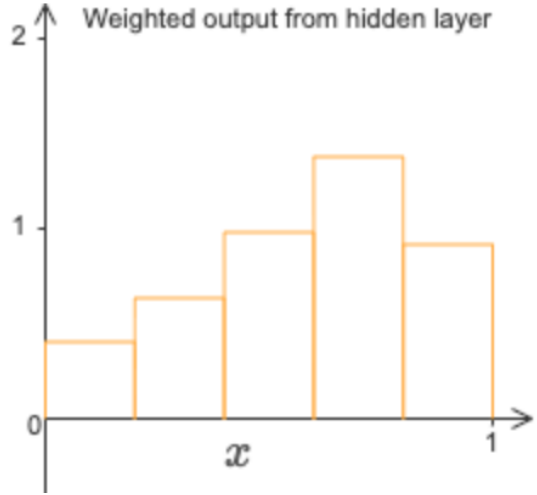




*Adapted from Nielsen (2019), Chapter 4.

The visual constructive proof follows Michael Nielsen (2015)

Goal: To approximate continuous function via Riemann sum

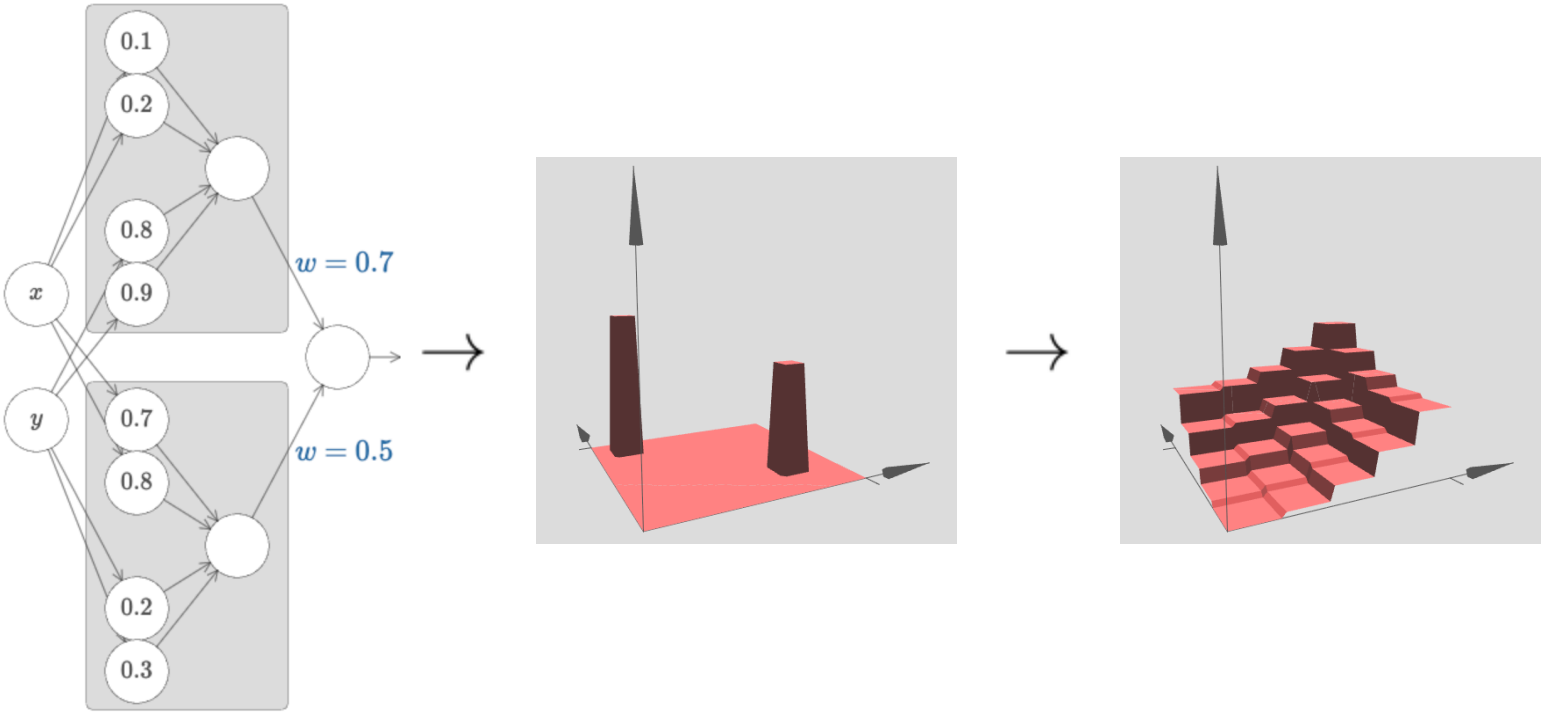



*Adapted from Nielsen (2019), Chapter 4.

The visual constructive proof follows Michael Nielsen (2015)

To higher dimensions...




*Adapted from Nielsen (2019), Chapter 4.

# Then why deep learning?

"If one hidden layer is enough,

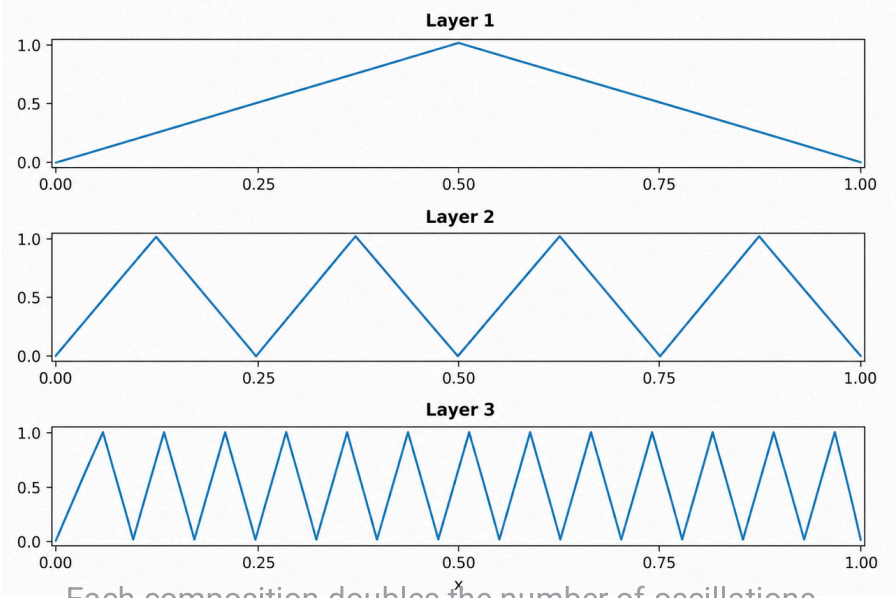
why do modern models have hundreds of layers?"

- Vanishing gradient
- Exploding gradient
- Highly non-convex optimisation
- ... :(

# Depth Separation

To represent certain hierarchically composed functions:

- shallow network needs exponentially many neurons, $N_{\text{shallow}} = \Omega(2^k)$
- whereas deep network needs only linearly many neurons, $N_{\text{deep}} = O(k)$




Each composition doubles the number of oscillations.

eg.

$$f(x) = \Delta^{(k)}(x)$$

$$\Delta(x) = \begin{cases} 2x, & 0 \le x \le \frac{1}{2}, \\ 2 - 2x, & \frac{1}{2} < x \le 1. \end{cases}$$

# Depth Separation



## Hierarchical Representation Hypothesis

**Bird (*Ramphastos toco*)**

↑

Parts

**Eye • Giant Beak • Feathers • Claws • Branch**

↑

Features

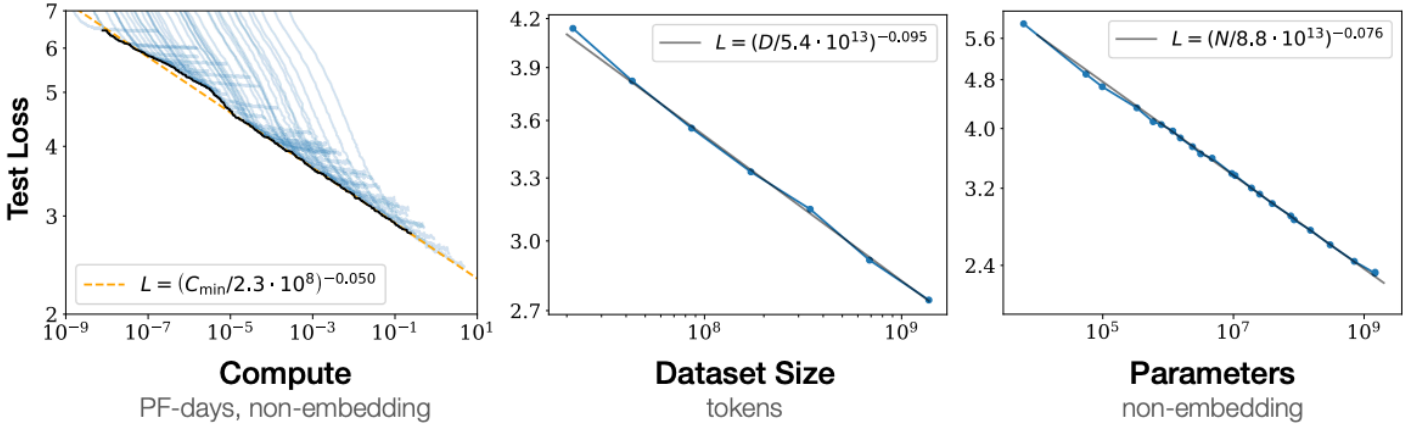
**Edges • Curves • Textures**

↑

**Pixels**

# Scaling Laws

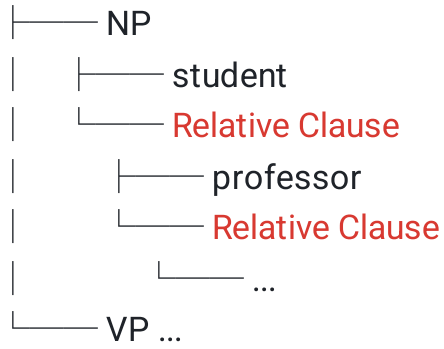
**Key findings**: LLM performance depends strongly on scale.



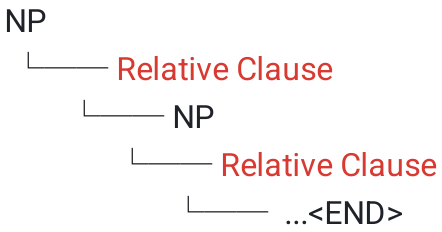

"Within reasonable limits, performance depends very weakly on other architectural hyperparameters **such as depth vs. width**."

# Natural languages exhibit hierarchical composition

eg. "The student (who the professor (that I met yesterday) recommended) finally graduated."




*This is the syntax tree from Linguistics




**Finding: Natural language is hierarchical, but not arbitrarily deep.**

# Reconciling Two Perspectives

*This is an interpretation rather than a proven theoretical result.

- **Depth Separation**
  Some functions require exponentially more neurons when depth is limited.

**Scaling Laws**
Modern Transformers show little sensitivity to depth–width trade-offs.

- **Possible interpretation**
  Natural language exhibits hierarchical composition, but probably not the extreme compositional complexity of worst-case theoretical constructions. $f(x) = \Delta^{(k)}(x)$

  Transformers may already operate beyond a **critical depth**, where additional depth provides diminishing returns.

*GPT-3 (175B) contains 96 layers of Transformer decoder blocks.

# Further thoughts...

- If Transformers already operate beyond a critical depth, why do increasingly large models continue to generalize well rather than simply overfitting?

- Scaling laws are empirical observations. Is there a deeper theoretical explanation for the emergence of power-law scaling?

# Reference

Cybenko, G. (1989). *Approximation by superpositions of a sigmoidal function*. Mathematics of Control, Signals and Systems, 2(4), 303–314.

Hornik, K., Stinchcombe, M., & White, H. (1989). *Multilayer feedforward networks are universal approximators*. Neural Networks, 2(5), 359–366.

Nielsen, M. (2019). *Neural Networks and Deep Learning*. Chapter 4 & Chapter 5. [http://neuralnetworksanddeeplearning.com/]

Telgarsky, M. (2016). *Benefits of Depth in Neural Networks*. Proceedings of the 29th Conference on Learning Theory (COLT).

Kaplan, J., et al. (2020). *Scaling Laws for Neural Language Models*. arXiv:2001.08361.

***NB**: Most figures illustrating the constructive proof are adapted from Michael Nielsen's *Neural Networks and Deep Learning* (Chapter 4).