Scaling

Morgan Sinclaire

I am large, I contain multitudes.

– Walt Whitman

Outline

1. Architectures, Critically

2. Scaling, Bigly

3. Intelligence, Generally

Marr's 3 Levels of Analysis

When analyzing a computational system, there are 3 levels at which to analyze it:

1. **Computational level**: What problem(s) is the system solving?
2. **Algorithmic level**: What methods/processes is the system using to solve (1)?
3. **Physical/implementation level**: How, engineering-wise, is (2) physically realized?

This framework applies to *any* system (Marr's original example was a cash register).

Marr's 3 Levels of Analysis

E.g. the human brain (as we understand it):[1]

1. **Computational level**: Minimize prediction error in continuous sensory data to make quick decisions, while respecting constraints such as energy efficiency.
2. **Algorithmic level**: In a hierarchy of predictive systems, sense flows up, predictions flow down, and intermediate levels resolve the conflicts based on relative strength.
3. **Physical/implementation level**: Carbon-based biological neurons, which communicate via rate/timing of electrical signals over ion channels.

[1] According to predictive processing theory, which is lately the main paradigm in neuroscience.

Marr's 3 Levels of Analysis

A generative transformer:

1. **Computational level**: Next-token prediction on (its subsample of) internet data.
2. **Algorithmic level**: Attention algorithm: use KQ pairs to compute relevance, use this and V to compute cumulative meaning of entire text. Also, feedforward layers for nonlinguistic thinking/reflection. Gradient descent to correct errors in loss function.
3. **Physical/implementation level**: Uses matrices of floating-point numbers on GPUs, representing KQV scores as individual row/column vectors to be multiplied. LOTS of low-level black magic hacks to make the H100s sing in chorus.

Are transformers the unique secret to intelligence?

No. As far as we can tell, RNNs had similar returns to scale. Language models weren't worth scaling up until the late 2010s.

Alternate timeline: transformer never invented. GPT-2 comes out in the 2020s as a char-RNN. RNNs hailed as "the universal architecture of intelligence".

Another timeline: RNNs never invented. GPT-2 comes out in the 2030s as a feedforward NN trained on next-char prediction. Feedforward NNs hailed as [...]

The actual star of the story: training on a general-purpose task (e.g. next-token prediction) with trillions of data points.

Are transformers emulating the human brain?

Definitely no. There's a *very loose* analogy to be made:

CNNs : the visual cortex
RNNs : the rest of the brain

In particular, RNNs have recurrent (self-looping) connections, like most of the brain.[2]

In contrast, a transformer is completely acyclic (i.e. a DAG); RNNs → transformers was a step *away* from biological inspiration.

The DAG-ness is what allows more parallelization in training; essentially, it's an efficiency hack to more quickly do what an (attention-based) RNN already could (evolution wasn't clever enough to make this hack).

This should not be surprising! E.g. we didn't get flying machines by closely emulating bird wings; the results are over 100x faster.

[2] This was thought to be necessary for "thinking"; turns out a transformer's MLP layers suffice.

Outline

1 Architectures, Critically

2 Scaling, Bigly

3 Intelligence, Generally

Language Models: Large and In Charge

Model	Year	Params	Corpus Size	FLOP	Cost
GPT-1	2018	0.117B	1e9	1.8e19	?
GPT-2	2019	1.5B	8e9	1.9e21	$\sim$ $50K
GPT-3	2020	175B	3.7e11	3.1e23	$2.1M
GPT-4	2023	$\sim$ 1T	?	1-4e25	$\sim$ $50M

Reference: Epoch AI. Many figures are estimates.

Fun fact: OpenAI incorrectly calculated the model sizes for the GPT-2 subfamily, and their original paper is wrong (1, 2).

LLM Dimensionalities

Standard notation for dimensionality hyperparameters:

d_{model}	size of token embedding space
d_{head}	size of query/key/value vectors
d_{ff}	size of feedforward/MLP activation space ("neurons")
n_{layer}	# of attention layers
n_{heads}	# of parallel heads per layer
n_{ctx}	max context size
d_{vocab}	BPE vocab size

LLM Dimensionalities

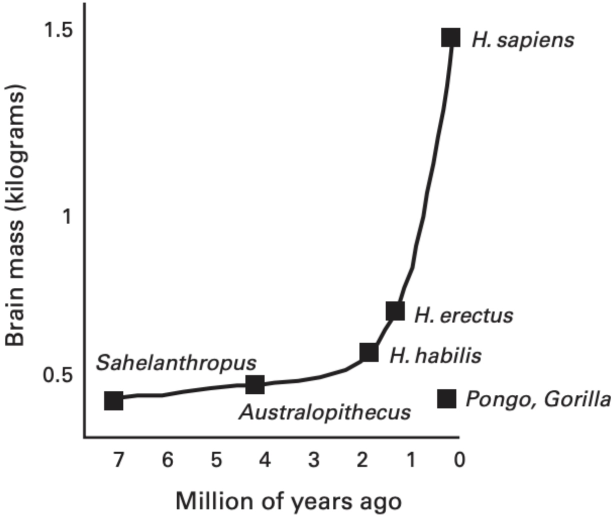
	d_{model}	d_{head}	d_{ff}	n_{heads}	n_{layer}	n_{ctx}	d_{vocab}
GPT-1	768	64	3072	12	12	512	40478
GPT-2 small	768	64	3072	12	12	1024	50257
GPT-2	1600	64	6400	25	48	1024	50257
GPT-3	12288	128	49152	96	96	2048	50257
DeepSeek-V3	7168	128	18432[3]	128	61	128k	128k

Common conventions:

1. $d_{key} = d_{value}$ (i.e. d_{head} is well-defined)
2. $d_{model} = n_{heads} \cdot d_{head}$ (orth. decompose the embedding space)
3. $d_{ff} = 4 \cdot d_{model}$

[3] Uses MoE with 1 shared expert and 8 routed experts (chosen from 256). Each has dimension 2048, so effectively $d_{ff} = (1 + 8) \cdot 2048 = 18432$.

Scaling in Natural Neural Networks (NNNs)



Source: Suzana Herculano-Houzel, *The Human Advantage: A New Understanding of How Our Brain Became Remarkable*

Outline

1. Architectures, Critically

2. Scaling, Bigly

3. Intelligence, Generally

What's the actual secret to intelligence? (Opinion)

The bitter/unsatisfying lesson:

A massive, efficient search over the space of Turing machines[4]

Scaling gives us that:

- Model size: need enough capacity to store most of the useful TMs.
- Dataset size: need a well-trained prior to conduct the search.
- Inference-time compute: need more "juice" to conduct the search.

[4] Each TM represents an "idea"/"action"/"procedure" that may be useful in some situations.

What's the actual secret to intelligence? (Opinion)

The bitter/unsatisfying lesson:

A massive, efficient search over the space of Turing machines[4]

Scaling gives us that:

- Model size: need enough capacity to store most of the useful TMs.
- Dataset size: need a well-trained prior to conduct the search.
- Inference-time compute: need more "juice" to conduct the search.

But what about creativity?

[4] Each TM represents an "idea"/"action"/"procedure" that may be useful in some situations.