Discussions with Dr. Paul Christiano, Fall 2019-Spring 2020

Participants

- Dr. Paul Christiano - Researcher, OpenAI
- Joseph Carlsmith - Research Analyst, Open Philanthropy

Note: These notes were compiled by Open Philanthropy and give an overview of the major points made by Dr. Christiano. Some of these points were made in conversation, and some via electronic communication, over the course of fall of 2019 and spring of 2020.

Summary

Open Philanthropy reached out to Dr. Paul Christiano of OpenAI as part of its investigation of what we can learn from the brain about the computational power ("compute") sufficient to match human-level task performance. The discussions focused on compute estimates based on communication in the brain, and on the applicability of Landauer's principle to the brain's information-processing.

The communication method

Across many different models of computation (e.g. Turing Machines, RAM machines, circuits, etc), computational resources tend to fall into a number of broad categories, including:

- memory (e.g., data the computer can store),
- communication (roughly, the amount of information the computer can send from one part to another),
- compute/number of operations.

The exact meaning of these concepts varies across models, but they are often useful to work with.

It may be easier to estimate the communication capacity of the brain than to estimate its compute capacity. The relationship between communication and computation might then serve as a basis for compute estimates. What's more, estimates of the brain's communication capacity are valuable in their own right (e.g., you won't be able to simulate the brain without that much communication).

Communication in the brain

You can roughly estimate the bandwidth of axon communication by dividing the firing rate by the temporal resolution of spiking. Thus, for example, if the temporal precision is 1 ms, and neurons are spiking at roughly 1 Hz, then each spike would communicate ~10 bits of information (e.g., $\log_2(1000)$). If you increase the temporal precision to every microsecond, that's only a factor of two difference (e.g., $\log_2(1{,}000{,}000) = {\sim}20$ bits).

There are other communication mechanisms in the brain (e.g., glia, neuromodulation, ephaptic effects), but Dr. Christiano expects that these will be lower-bandwidth than axon communication. What's more, the brain invests a sizeable portion of its energy and volume into communication via axons, which would be a strange investment if it had some other, superior communication mechanism available.

One can also distinguish between the bandwidth available at different distances. Axons vary in length, shorter-distance communication in neurons occurs via dendrites, and at sufficiently short distances, the distinction between communication and computation becomes blurry. For example, a multiply is in some sense mostly communication, and one can think of different processes taking place within neurons as communication as well. For longer-distance communication, though, axons seems like the brain's primary mechanism.

From communication to computation

In designing brains, evolution had to make trade-offs in allocating resources (e.g., energy consumption, space) to additional communication mechanisms, vs. additional mechanisms used for computation. Human engineers designing chips also have to make trade-offs in budgeting resources (energy, chip real-estate) to computation vs. communication.

Equipped with an estimate of the communication profile of the brain, then, we might be able to use our knowledge of how to balance communication and computation in human computers to estimate what it would take to match the compute power of the brain, or to match its overall performance.

For example, Dr. Christiano puts some weight on the following type of *a priori* argument: if you have two computers that are comparable on one dimension (e.g., communication), but you can't measure how they compare along any other dimensions, then *a priori* your median guess should be that they are comparable on these other dimensions as well (e.g., it would be strange to have a strong view about which is better).

One complication here is that the communication to computation ratio in human computers has changed over time. For example, traditional CPUs had less computation per unit communication than the current hardware used for AI applications, like GPUs (Dr. Christiano says that this is partly because it is easier to write software if you can operate on anything in memory rather than needing to worry about communication and parallelization). If we applied CPU-like ratios to the brain, we would get very low compute estimates. Current supercomputers, though, spend more comparable amounts of energy on communication (including within chips) and compute.

Dr. Christiano's approach requires some sort of production function relating the returns from investment in communication to investment in compute. Dr. Christiano's starting point would be something like logarithmic returns (though there aren't really two buckets, so a more accurate model would be much messier), and he thinks that when you have two complimentary quantities (say, X and Y), a 50/50 resource split between them is reasonable across a wide range of production functions. After all, a 50% allocation to X will likely give you at least 50% of the maximal value that X can provide, and halving your allocation to X will only allow you to increase your allocation to Y by 50%.

Such a production function would also allow you to estimate what it would take to match the overall performance of the brain, even without matching its compute capacity. Thus, for example, it's theoretically possible that biological systems have access to large amounts of very efficient computation. If we assume that the value of additional computation diminishes if communication is held fixed, though, then even if the brain has substantially more computation than human computers can mobilize, we might be able to match its overall performance regardless, by exceeding its communication capacity (and hence increasing the value of our marginal compute to overall performance).

Comparison with a V100

Roughly 1e8 axons cross the corpus callosum, and these account for a significant fraction of the length of all axons (AI Impacts has some estimates in this regard). Based on estimates Dr. Christiano has seen for the total length of all axons and dendrites, and the estimate that 1 spike/second = 10 bits/second across each, he thinks the following bounds are likely: 1e9 bytes/s of long-distance communication (across the brain), 1e11 bytes/s of short-distance communication (where each neuron could access about 10 million nearby neurons), and larger amounts of very-short distance communication.

A V100 GPU has about 1e12 bytes/s of memory bandwidth on the chip (~10x the brain's 1e11 bytes of short-distance communication, estimated above), and 3e11 bytes/s of off-chip bandwidth (~300x the brain's 1e9 bytes/s of long-distance communication, estimated above). Dr. Christiano thinks that these memory access numbers are comparable, based on matching up the memory of a V100 (respectively, cluster of V100s) to the amount of information stored in synapses accessible by the "short-distance" (respectively, "long-distance") connections described above.

If we knew nothing else about the brain, then, this might suggest that the brain's computational capacity will be less than, or at least comparable to, a V100's computational capacity (~1e14 FLOP/s) as well. And even if our compute estimates for the brain are higher, communication estimates are plausibly more robust, and they provide a different indication of how powerful the brain is relative to our computers.

Landauer's principle and the brain

Reversible computing

The algorithmic overhead involved in reversible computing (specifically, the overhead involved in un-computing what you have already computed) is not that bad. Most of the difficulty lies in designing such efficient hardware.

Partly for this reason, Dr. Christiano does not think that you can get an upper bound on the FLOP/s required to do what the brain does, purely by appealing to the energy required to erase bits. We believe that you can perform extremely complex computations with almost no bit erasures using good enough hardware.

What's more, Dr. Christiano does not think that logically irreversible operations are a more natural or default computational unit than reversible ones. And once we're engaging with models of brain computation that invoke computations performed by low-level, reversible elements, then we are assuming that the brain is able to make use of such elements, in which case it may well have evolved a reliance on them from the start.

For example, if it were possible to use proteins to directly perform large tunable matrix multiplications, Landauer's principle implies that those matrix multiplications would necessarily be invertible or even unitary. But unitary matrix multiplications are just as useful for deep learning as general matrix multiplications, so Landauer's principle per se doesn't tell us anything about the feasibility of the scenario. Instead the focus should be on other arguments (e.g. regarding consistency and flexibility).

Overall plausibility of more than 1 FLOP per bit-erasure

Dr. Christiano expects that experts in physics, chemistry, and computer engineering would generally think it extremely unlikely that the brain is erasing less than one bit per computationally useful FLOP it performs. If the brain were doing this, Dr. Christiano believes that this would mean that the brain is qualitatively much more impressive than any other other biological machinery we are aware of.

In irreversible computers, you do not need to keep track of and take into account what happens to each degree of freedom, because you are able to expend energy to reset the system to a state it needs to be in for your computation to proceed successfully. With reversible computers, however, you aren't able to expend such energy, so what happens to any degree of freedom that could influence your computation starts to matter a lot; you can't simply force the relevant physical variables into a particular state, so your computation needs to work for the particular state that those variables happen to be in. Given the reversibility of physics, this is a very difficult engineering challenge.

Dr. Christiano would be extremely surprised if the brain got more computational mileage out of a single ATP than human engineers can get out of a FLOP, and he would be very willing to bet that it takes at least 10 ATPs to get the equivalent of a FLOP. Mr. Carlsmith estimates that the brain can be using no more than ~1e20 ATPs/second. If this estimate is right, then Dr. Christiano is very confident that you do not need more than 1e20 FLOP/s to replicate the brain's task-performance.

Modeling biophysical mechanisms

Synaptic stochasticity

One way of modeling synaptic stochasticity is by assigning a fixed release probability to each synaptic vesicle, conditional on presynaptic activity. Dr. Christiano does not think that modeling spikes through synapses in this way would constitute a significant increase in required compute, relative to modeling each spike through synapse deterministically.

Sampling from a normal distribution is cheap unless you need a lot of precision, and even then, Dr. Christiano believes that the cost is just linear in the number of bits of precision that you want. At 8 bits of precision and 10 vesicles, he expects that it would be possible to perform the relevant sampling with about the same amount of energy as a FLOP.

Complexity of firing decisions

Neurons receive only a limited number of bits in, and they output only a limited number of bits. However, in principle, you can imagine computational elements receiving encodings of computationally intensive problems via their synaptic inputs (e.g., "is this boolean formula satisfiable?"), and then outputting one of a comparatively small set of difficult-to-arrive-at answers.

To Dr. Christiano, the complexity of a Hodgkin-Huxley model does not appear intuitively useful for performing a wide range of complex tasks, relative to simpler models. It is difficult to describe any function for which (a) the Hodgkin-Huxley model is a useful computational building block, and (b) its usefulness arises from some feature it possesses that simpler computational building blocks do not also possess.

A ReLU costs less than a FLOP. Indeed, it can be performed with many fewer transistors than a multiply of equivalent precision.

Dr. Christiano is very skeptical of the hypothesis that a single, biological cortical neuron could be used to classify handwritten digits.

Frequency of firing-decision computation

Dr. Christiano expects that in modeling a neuron's input-output function, one would not need to compute, every time-step, whether or not the neuron fires during that time-step. Rather, you could accumulate information about the inputs to a neuron over a longer period, and then compute the timing of its spikes over that period all at once.

This definitely holds in a purely feedforward context - e.g., for a given neuron, you could simply compute all of the times that the neuron fires, and then use this information to compute when all of the downstream neurons fire, and so on. The fact that the brain's architecture is highly recurrent complicates this picture, as the firing pattern of a particular neuron may be able to influence the inputs that that same neuron receives. However, the time it takes for an action potential to propagate would be a lower bound on how long it would be possible to wait in accumulating synaptic inputs (since the timescale of a neuron's influence on its own inputs is capped by the propagation time of its outgoing signals).

Learning

Based on his understanding of the brain's physiology, Dr. Christiano thinks it extremely implausible that the brain could be implementing second-order optimization methods.

Conceptual foundations

Brain-like-ness

Dr. Christiano expects that the distinction between the FLOP/s required to perform every task that the brain can perform, and the FLOP/s required to perform every task that the brain can perform *in a manner that exhibits a specified type of resemblance to the brain* (e.g., that satisfies some "brain-like-ness constraint") will not make a material difference to FLOP/s estimates.

If you include a sufficiently broad range of tasks that the human brain can perform, and require similarly useful task-performance across the full range of inputs to which the brain could be exposed, it is likely that for at least one of the tasks in the relevant profile, for some set of inputs, the brain's method will (a) be close to maximally algorithmically efficient (e.g., within an order of magnitude or two), and (b) use a substantial portion of the computational resources that the brain has available.

For example, if you take a computer from the 60s, and you look at all of the tasks it could perform, Dr. Christiano expects that many of the algorithms it was running (for example: sorting), were close to optimally efficient. As another example, there is a very inefficient algorithm for SAT solving, which takes 2^n time. For many inputs, we can improve on this algorithm by a huge amount, but we can't for every input: indeed, there is a rough consensus amongst computer scientists that the very inefficient algorithm is close to the best one can do.

Indeed, Dr. Christiano expects that for most algorithms, there will be some family of instances on which it does reasonably well. And given how large the space of possible tasks the brain performs is (we can imagine a very wide set of evaluation metrics and input regimes), the density of roughly-optimal-on-some-inputs algorithms doesn't need to be that high for them to appear in the brain.

Avoiding overcounting

In thinking about conceptual standards to use in generating estimates for the FLOP/s necessary to run a task-functional model of a computational system that exhibits some degree of similarity to that system, one constraint is that when you apply your standard to digital systems that actually perform FLOPs, it ought to yield an answer of one FLOP per FLOP (e.g., your estimate for a V100, which performs ~1e14 FLOP/s, should be 1e14 FLOP/s). That is, it shouldn't yield an estimate of the FLOPs necessary to e.g. model every transistor, or to model lower-level physical processes in transistors leading to e.g. specific patterns of mistaken bit-flips.

Multiple levels of abstraction

If we attempt to use some standard of the form "models such that there is some reasonable mapping from their internal functional structure to the internal structure of the computational system in question," then Dr. Christiano expects that there will be a large number of models that satisfy this criteria, which map onto descriptions of the system at different levels of abstraction.

The hierarchy of abstraction in a digital computer runs from wires and transistor components, to transistors, to gates like NAND gates, to the implementation of an ALU, to the ALU itself, to function calls, all the way up to very high-level descriptions of the system. If you require only that the models you care about map onto one of these levels, then you aren't imposing much of a constraint, relative to simply requiring that your model replicate overall task-performance.

In the case of the brain, for example, a high-level description might be something like "it divides the work between these two hemispheres in the following way." Thus, to meet the relevant standard, "brain-like" computational models will only need to replicate that hemispheric division. Beyond that, they can just employ the maximally efficient way of performing the task.

Attempting to use some standard like "the description of the system you would give if you really understood how the system worked" might well result in over-estimates, since it would plausibly result in descriptions at lower levels, like transistors or NAND gates.

All Open Philanthropy conversations are available at
http://www.openphilanthropy.org/research/conversations