Deep Learning breakthrough made by Rice University scientists

reader comments

38 with 28 posters participating, including story author

Share this story

In an earlier deep learning article, we talked about how inference workloads—the use of already-trained neural networks to analyze data—can run on fairly cheap hardware, but running the training workload that the neural network "learns" on is orders of magnitude more expensive.

In particular, the more potential inputs you have to an algorithm, the more out of control your scaling problem gets when analyzing its problem space. This is where MACH, a research project authored by Rice University's Tharun Medini and Anshumali Shrivastava, comes in. MACH is an acronym for Merged Average Classifiers via Hashing, and according to lead researcher Shrivastava, "[its] training times are about 7-10 times faster, and… memory footprints are 2-4 times smaller" than those of previous large-scale deep learning techniques.

In describing the scale of extreme classification problems, Medini refers to online shopping search queries, noting that "there are easily more than 100 million products online." This is, if anything, conservative—one data company claimed Amazon US alone sold 606 million separate products, with the entire company offering more than three billion products worldwide.

Another company reckons the US product count at 353 million. Medini continues, “a neural network that takes search input and predicts from 100 million outputs, or products, will typically end up with about 2,000 parameters per product. So you multiply those, and the final layer of the neural network is 200 billion parameters … [and] I’m talking about a very, very dead simple neural network model.”

At this scale, a supercomputer would likely need terabytes of working memory just to store the model. The memory problem gets even worse when you bring GPUs into the picture. GPUs can process neural network workloads orders of magnitude faster than general purpose CPUs can, but each GPU has a relatively small amount of RAM—even the most expensive Nvidia Tesla GPUs only have 32GB of RAM. Medini says, “training such a model is prohibitive due to massive inter-GPU communication.”

Instead of training on the entire 100 million outcomes—product purchases, in this example—Mach divides them into three “buckets,” each containing 33.3 million randomly selected outcomes. Now, MACH creates another “world,” and in that world, the 100 million outcomes are again randomly sorted into three buckets. Crucially, the random sorting is separate in World One and World Two—they each have the same 100 million outcomes, but their random distribution into buckets is different for each world.

With each world instantiated, a search is fed to both a “world one” classifier and a “world two” classifier, with only three possible outcomes apiece. “What is this person thinking about?” asks Shrivastava. “The most probable class is something that is common between these two buckets.”

At this point, there are nine possible outcomes—three buckets in World One times three buckets in World Two. But MACH only needed to create six classes—World One’s three buckets *plus* World Two’s three buckets—to model that nine-outcome search space. This advantage improves as more “worlds” are created; a

three-world approach produces 27 outcomes from only nine created classes, a four-world setup gives 81 outcomes from 12 classes, and so forth. “I am paying a cost linearly, and I am getting an exponential improvement,” Shrivastava says.

Better yet, MACH lends itself better to distributed computing on smaller individual instances. The worlds “don’t even have to talk to one another,” Medini says. “In principle, you could train each [world] on a single GPU, which is something you could never do with a non-independent approach.” In the real world, the researchers applied MACH to a 49 million product Amazon training database, randomly sorting it into 10,000 buckets in each of 32 separate worlds. That reduced the required parameters in the model more than an order of magnitude—and according to Medini, training the model required both less time and less memory than some of the best reported training times on models with comparable parameters.

Of course, this wouldn’t be an Ars article on deep learning if we didn’t close it out with a cynical reminder about unintended consequences. The unspoken reality is that the neural network isn’t actually learning to show shoppers what they asked for. Instead, it’s learning how to turn queries into *purchases.* The neural network doesn’t know or care what the human was actually searching for; it just has an idea what that human is most likely to buy—and without sufficient oversight, systems trained to increase outcome probabilities this way can end up suggesting baby products to women who’ve suffered miscarriages, or worse.

https://arstechnica.com/?p=1632445