Researchers report breakthrough in 'distributed deep learning'

9 December 2019, by Jade Boyd

Anshumali Shrivastava is an assistant professor of computer science at Rice University. Credit: Jeff Fitlow/Rice University

Online shoppers typically string together a few words to search for the product they want, but in a world with millions of products and shoppers, the task of matching those unspecific words to the right product is one of the biggest challenges in information retrieval.

Using a divide-and-conquer approach that leverages the power of compressed sensing, computer scientists from Rice University and Amazon have shown they can slash the amount of time and computational resources it takes to train computers for product search and similar "extreme classification problems" like speech translation and answering general questions.

The research will be presented this week at the 2019 Conference on Neural Information Processing Systems (NeurIPS 2019) in Vancouver. The results include tests performed in 2018 when lead researcher Anshumali Shrivastava and lead author Tharun Medini, both of Rice, were visiting Amazon Search in Palo Alto, California.

In tests on an Amazon search dataset that included some 70 million queries and more than 49 million products, Shrivastava, Medini and colleagues showed their approach of using "merged-average classifiers via hashing," (MACH) required a fraction of the training resources of some state-of-the-art commercial systems.

"Our training times are about 7-10 times faster, and our memory footprints are 2-4 times smaller than the best baseline performances of previously reported large-scale, distributed deep-learning systems," said Shrivastava, an assistant professor of computer science at Rice.

Medini, a Ph.D. student at Rice, said product search is challenging, in part, because of the sheer number of products. "There are about 1 million English words, for example, but there are easily more than 100 million products online."

Rice University computer science graduate students Beidi Chen and Tharun Medini collaborate during a group meeting. Credit: Jeff Fitlow/Rice University

There are also millions of people shopping for those products, each in their own way. Some type a

question. Others use keywords. And many aren't sure what they're looking for when they start. But because millions of online searches are performed every day, tech companies like Amazon, Google and Microsoft have a lot of data on successful and unsuccessful searches. And using this data for a type of machine learning called deep learning is one of the most effective ways to give better results to users.

Deep learning systems, or neural network models, are vast collections of mathematical equations that take a set of numbers called input vectors, and transform them into a different set of numbers called output vectors. The networks are composed of matrices with several parameters, and state-of-the-art distributed deep learning systems contain billions of parameters that are divided into multiple layers. During training, data is fed to the first layer, vectors are transformed, and the outputs are fed to the next layer and so on.

"Extreme classification problems" are ones with many possible outcomes, and thus, many parameters. Deep learning models for extreme classification are so large that they typically must be trained on what is effectively a supercomputer, a linked set of graphics processing units (GPU) where parameters are distributed and run in parallel, often for several days.

"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," Medini said. "So you multiply those, and the final layer of the neural network is now 200 billion parameters. And I have not done anything sophisticated. I'm talking about a very, very dead simple neural network model."

"It would take about 500 gigabytes of memory to store those 200 billion parameters," Medini said. "But if you look at current training algorithms, there's a famous one called Adam that takes two more parameters for every parameter in the model, because it needs statistics from those parameters to monitor the training process. So, now we are at 200 billion times three, and I will need 1.5 terabytes of working memory just to store the model. I haven't even gotten to the training data. The best GPUs out there have only 32 gigabytes of memory, so training such a model is prohibitive due to massive inter-GPU communication."

MACH takes a very different approach. Shrivastava describes it with a thought experiment randomly dividing the 100 million products into three classes, which take the form of buckets. "I'm mixing, let's say, iPhones with chargers and T-shirts all in the same bucket," he said. "It's a drastic reduction from 100 million to three."

In the thought experiment, the 100 million products are randomly sorted into three buckets in two different worlds, which means that products can wind up in different buckets in each world. A classifier is trained to assign searches to the buckets rather than the products inside them, meaning the classifier only needs to map a search to one of three classes of product.

"Now I feed a search to the classifier in world one, and it says bucket three, and I feed it to the classifier in world two, and it says bucket one," he said. "What is this person thinking about? The most probable class is something that is common between these two buckets. If you look at the possible intersection of the buckets there are three in world one times three in world two, or nine possibilities," he said. "So I have reduced my search space to one over nine, and I have only paid the cost of creating six classes."

Adding a third world, and three more buckets, increases the number of possible intersections by a factor of three. "There are now 27 possibilities for what this person is thinking," he said. "So I have reduced my search space by one over 27, but I've only paid the cost for nine classes. I am paying a cost linearly, and I am getting an exponential improvement."

In their experiments with Amazon's training database, Shrivastava, Medini and colleagues randomly divided the 49 million products into 10,000 classes, or buckets, and repeated the process 32 times. That reduced the number of parameters in the model from around 100 billion to 6.4 billion. And training the model took less time and less memory than some of the best reported

training times on models with comparable parameters, including Google's Sparsely-Gated Mixture-of-Experts (MoE) model, Medini said.

He said MACH's most significant feature is that it requires no communication between parallel processors. In the thought experiment, that is what's represented by the separate, independent worlds.

"They don't even have to talk to each other," Medini said. "In principle, you could train each of the 32 on one GPU, which is something you could never do with a nonindependent approach."

Shrivastava said, "In general, training has required communication across parameters, which means that all the processors that are running in parallel have to share information. Looking forward, communication is a huge issue in distributed deep learning. Google has expressed aspirations of training a 1 trillion parameter network, for example. MACH, currently, cannot be applied to use cases with small number of classes, but for extreme classification, it achieves the holy grail of zero communication."

Provided by Rice University