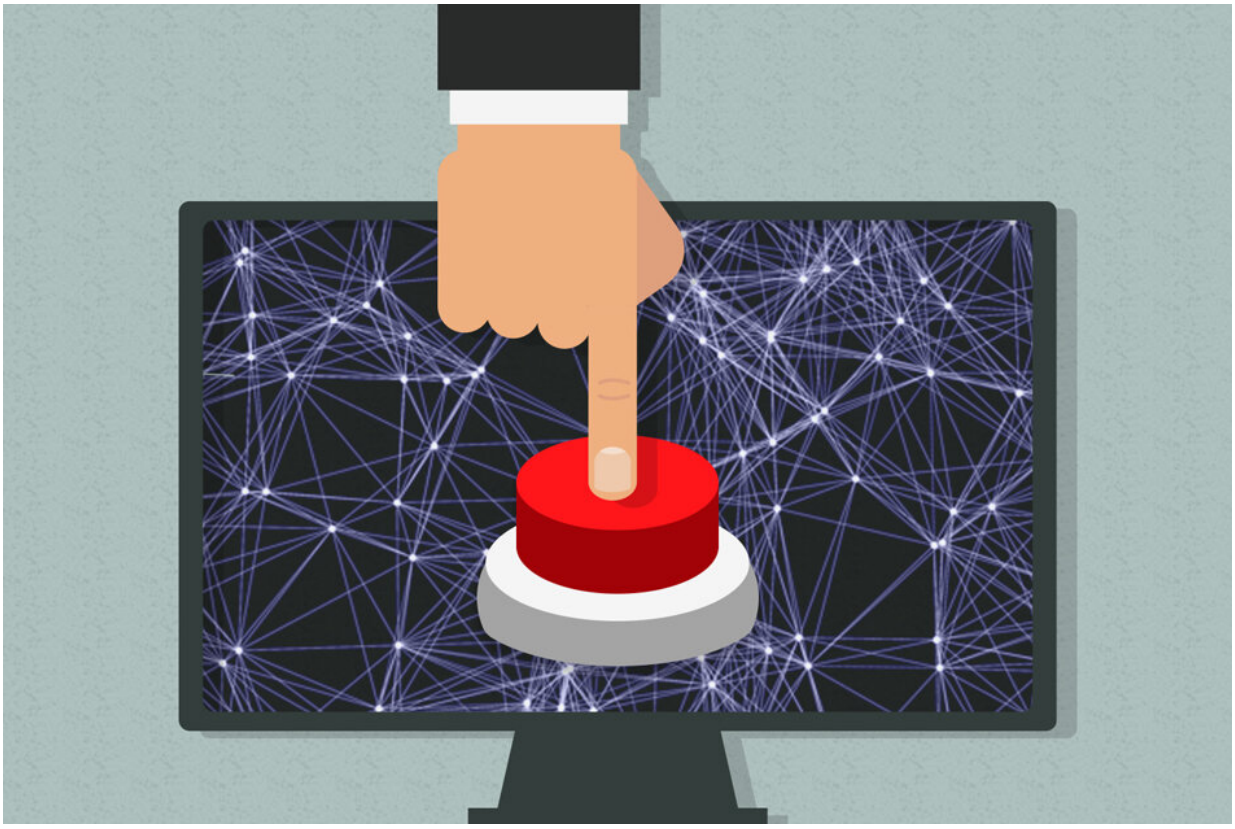
# Algorithm designs optimized machine-learning models up to 200 times faster than traditional methods

March 21 2019, by Rob Matheson



MIT researchers have developed an efficient algorithm that could provide a "push-button" solution for automatically designing fast-running neural networks on specific hardware. Credit: Chelsea Turner, MIT

A new area in artificial intelligence involves using algorithms to automatically design machine-learning systems known as neural networks, which are more accurate and efficient than those developed by human engineers. But this so-called neural architecture search (NAS) technique is computationally expensive.

One of the state-of-the-art NAS algorithms recently developed by Google took 48,000 hours of work by a squad of graphical processing units (GPUs) to produce a single convolutional neural network, used for image classification and identification tasks. Google has the wherewithal to run hundreds of GPUs and other specialized circuits in parallel, but that's out of reach for many others.

In a paper being presented at the International Conference on Learning Representations in May, MIT researchers describe an NAS algorithm that can directly learn specialized convolutional neural networks (CNNs) for target hardware platforms—when run on a massive image dataset—in only 200 GPU hours, which could enable far broader use of these types of algorithms.

Resource-strapped researchers and companies could benefit from the time- and cost-saving algorithm, the researchers say. The broad goal is "to democratize AI," says co-author Song Han, an assistant professor of electrical engineering and computer science and a researcher in the Microsystems Technology Laboratories at MIT. "We want to enable both AI experts and nonexperts to efficiently design neural network architectures with a push-button solution that runs fast on a specific hardware."

Han adds that such NAS algorithms will never replace human engineers. "The aim is to offload the repetitive and tedious work that comes with designing and refining neural network architectures," says Han, who is joined on the paper by two researchers in his group, Han Cai and Ligeng

Zhu.

## "Path-level" binarization and pruning

In their work, the researchers developed ways to delete unnecessary neural network design components, to cut computing times and use only a fraction of hardware memory to run a NAS algorithm. An additional innovation ensures each outputted CNN runs more efficiently on specific hardware platforms—CPUs, GPUs, and mobile devices—than those designed by traditional approaches. In tests, the researchers' CNNs were 1.8 times faster measured on a mobile phone than traditional gold-standard models with similar accuracy.

A CNN's architecture consists of layers of computation with adjustable parameters, called "filters," and the possible connections between those filters. Filters process image pixels in grids of squares—such as 3x3, 5x5, or 7x7—with each filter covering one square. The filters essentially move across the image and combine all the colors of their covered grid of pixels into a single pixel. Different layers may have different-sized filters, and connect to share data in different ways. The output is a condensed image—from the combined information from all the filters—that can be more easily analyzed by a computer.

Because the number of possible architectures to choose from—called the "search space"—is so large, applying NAS to create a neural network on massive image datasets is computationally prohibitive. Engineers typically run NAS on smaller proxy datasets and transfer their learned CNN architectures to the target task. This generalization method reduces the model's accuracy, however. Moreover, the same outputted architecture also is applied to all hardware platforms, which leads to efficiency issues.

The researchers trained and tested their new NAS algorithm on an image

classification task in the ImageNet dataset, which contains millions of images in a thousand classes. They first created a search space that contains all possible candidate CNN "paths"—meaning how the layers and filters connect to process the data. This gives the NAS algorithm free reign to find an optimal architecture.

This would typically mean all possible paths must be stored in memory, which would exceed GPU memory limits. To address this, the researchers leverage a technique called "path-level binarization," which stores only one sampled path at a time and saves an order of magnitude in memory consumption. They combine this binarization with "path-level pruning," a technique that traditionally learns which "neurons" in a neural network can be deleted without affecting the output. Instead of discarding neurons, however, the researchers' NAS algorithm prunes entire paths, which completely changes the neural network's architecture.

In training, all paths are initially given the same probability for selection. The algorithm then traces the paths—storing only one at a time—to note the accuracy and loss (a numerical penalty assigned for incorrect predictions) of their outputs. It then adjusts the probabilities of the paths to optimize both accuracy and efficiency. In the end, the algorithm prunes away all the low-probability paths and keeps only the path with the highest probability—which is the final CNN architecture.

## Hardware-aware

Another key innovation was making the NAS algorithm "hardware aware," Han says, meaning it uses the latency on each hardware platform as a feedback signal to optimize the architecture. To measure this latency on mobile devices, for instance, big companies such as Google will employ a "farm" of mobile devices, which is very expensive. The researchers instead built a model that predicts the latency using only a single mobile phone.

For each chosen layer of the network, the algorithm samples the architecture on that latency-prediction model. It then uses that information to design an architecture that runs as quickly as possible, while achieving high accuracy. In experiments, the researchers' CNN ran nearly twice as fast as a gold-standard model on mobile devices.

One interesting result, Han says, was that their NAS algorithm designed CNN architectures that were long dismissed as being too inefficient—but, in the researchers' tests, they were actually optimized for certain hardware. For instance, engineers have essentially stopped using 7x7 filters, because they're computationally more expensive than multiple, smaller filters. Yet, the researchers' NAS algorithm found architectures with some layers of 7x7 filters ran optimally on GPUs. That's because GPUs have high parallelization—meaning they compute many calculations simultaneously—so can process a single large filter at once more efficiently than processing multiple small filters one at a time.

"This goes against previous human thinking," Han says. "The larger the search space, the more unknown things you can find. You don't know if something will be better than the past human experience. Let the AI figure it out."

**More information:** ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware, arXiv:1812.00332 [cs.LG] arxiv.org/abs/1812.00332

Provided by Massachusetts Institute of Technology

https://techxplore.com/news/2019-03-algorithm-optimized-machine-learning-faster-traditional.html