More efficient security for cloud-based machine learning

August 17 2018, by Rob Matheson

A novel encryption method devised by MIT researchers secures data used in online neural networks, without dramatically slowing their runtimes, which holds promise for medical-image analysis using cloud-based neural networks and other applications. Credit: Chelsea Turner

A novel encryption method devised by MIT researchers secures data

used in online neural networks, without dramatically slowing their runtimes. This approach holds promise for using cloud-based neural networks for medical-image analysis and other applications that use sensitive data.

Outsourcing machine learning is a rising trend in industry. Major tech firms have launched cloud platforms that conduct computation-heavy tasks, such as, say, running data through a convolutional neural network (CNN) for image classification. Resource-strapped small businesses and other users can upload data to those services for a fee and get back results in several hours.

But what if there are leaks of private data? In recent years, researchers have explored various secure-computation techniques to protect such sensitive data. But those methods have performance drawbacks that make neural network evaluation (testing and validating) sluggish—sometimes as much as million times slower—limiting their wider adoption.

In a paper presented at this week's USENIX Security Conference, MIT researchers describe a system that blends two conventional techniques—homomorphic encryption and garbled circuits—in a way that helps the networks run orders of magnitude faster than they do with conventional approaches.

The researchers tested the system, called GAZELLE, on two-party image-classification tasks. A user sends encrypted image data to an online server evaluating a CNN running on GAZELLE. After this, both parties share encrypted information back and forth in order to classify the user's image. Throughout the process, the system ensures that the server never learns any uploaded data, while the user never learns anything about the network parameters. Compared to traditional systems, however, GAZELLE ran 20 to 30 times faster than state-of-the-

art models, while reducing the required network bandwidth by an order of magnitude.

One promising application for the system is training CNNs to diagnose diseases. Hospitals could, for instance, train a CNN to learn characteristics of certain medical conditions from magnetic resonance images (MRI) and identify those characteristics in uploaded MRIs. The hospital could make the model available in the cloud for other hospitals. But the model is trained on, and further relies on, private patient data. Because there are no efficient encryption models, this application isn't quite ready for prime time.

"In this work, we show how to efficiently do this kind of secure two-party communication by combining these two techniques in a clever way," says first author Chiraag Juvekar, a Ph.D. student in the Department of Electrical Engineering and Computer Science (EECS). "The next step is to take real medical data and show that, even when we scale it for applications real users care about, it still provides acceptable performance."

Co-authors on the paper are Vinod Vaikuntanathan, an associate professor in EECS and a member of the Computer Science and Artificial Intelligence Laboratory, and Anantha Chandrakasan, dean of the School of Engineering and the Vannevar Bush Professor of Electrical Engineering and Computer Science.

Maximizing performance

CNNs process image data through multiple linear and nonlinear layers of computation. Linear layers do the complex math, called linear algebra, and assign some values to the data. At a certain threshold, the data is outputted to nonlinear layers that do some simpler computation, make decisions (such as identifying image features), and send the data to the

next linear layer. The end result is an image with an assigned class, such as vehicle, animal, person, or anatomical feature.

Recent approaches to securing CNNs have involved applying homomorphic encryption or garbled circuits to process data throughout an entire network. These techniques are effective at securing data. "On paper, this looks like it solves the problem," Juvekar says. But they render complex neural networks inefficient, "so you wouldn't use them for any real-world application."

Homomorphic encryption, used in cloud computing, receives and executes computation all in encrypted data, called ciphertext, and generates an encrypted result that can then be decrypted by a user. When applied to neural networks, this technique is particularly fast and efficient at computing linear algebra. However, it must introduce a little noise into the data at each layer. Over multiple layers, noise accumulates, and the computation needed to filter that noise grows increasingly complex, slowing computation speeds.

Garbled circuits are a form of secure two-party computation. The technique takes an input from both parties, does some computation, and sends two separate inputs to each party. In that way, the parties send data to one another, but they never see the other party's data, only the relevant output on their side. The bandwidth needed to communicate data between parties, however, scales with computation complexity, not with the size of the input. In an online neural network, this technique works well in the nonlinear layers, where computation is minimal, but the bandwidth becomes unwieldy in math-heavy linear layers.

The MIT researchers, instead, combined the two techniques in a way that gets around their inefficiencies.

In their system, a user will upload ciphertext to a cloud-based CNN. The

user must have garbled circuits technique running on their own computer. The CNN does all the computation in the linear layer, then sends the data to the nonlinear layer. At that point, the CNN and user share the data. The user does some computation on garbled circuits, and sends the data back to the CNN. By splitting and sharing the workload, the system restricts the homomorphic encryption to doing complex math one layer at a time, so data doesn't become too noisy. It also limits the communication of the garbled circuits to just the nonlinear layers, where it performs optimally.

"We're only using the techniques for where they're most efficient," Juvekar says.

Secret sharing

The final step was ensuring both homomorphic and garbled circuit layers maintained a common randomization scheme, called "secret sharing." In this scheme, data is divided into separate parts that are given to separate parties. All parties synch their parts to reconstruct the full data.

In GAZELLE, when a user sends encrypted data to the cloud-based service, it's split between both parties. Added to each share is a secret key (random numbers) that only the owning party knows. Throughout computation, each party will always have some portion of the data, plus random numbers, so it appears fully random. At the end of computation, the two parties synch their data. Only then does the user ask the cloud-based service for its secret key. The user can then subtract the secret key from all the data to get the result.

"At the end of the computation, we want the first party to get the classification results and the second party to get absolutely nothing," Juvekar says. Additionally, "the first party learns nothing about the parameters of the model."

More information: GAZELLE: A Low Latency Framework for Secure Neural Network Inference: arxiv.org/pdf/1801.05507.pdf

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology