AI has a privacy problem, but these techniques could fix it

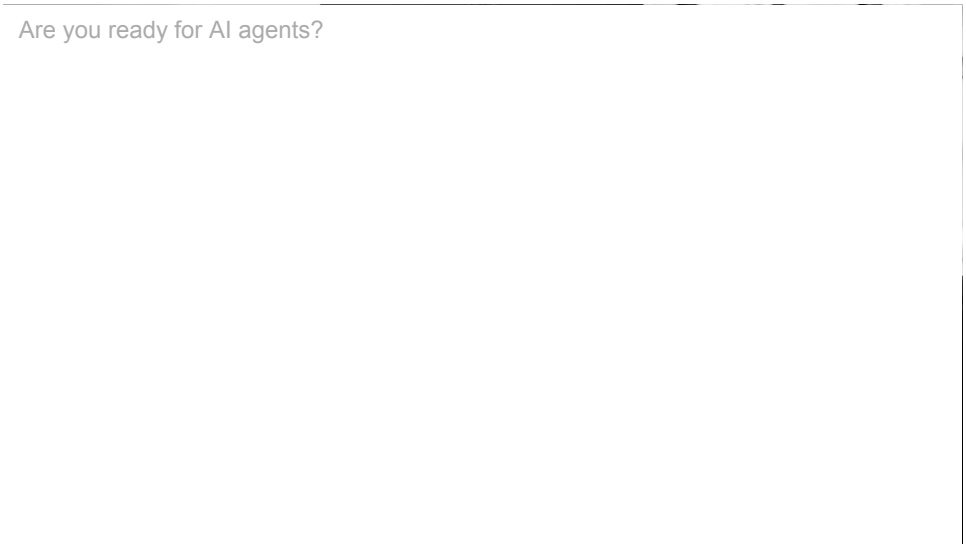
Kyle Wiggers

@Kyle_L_Wiggers

December 21, 2019 4:44 PM

Artificial intelligence promises to transform — and indeed, has already transformed — entire industries, from civic planning and health care to cybersecurity. But privacy remains an unsolved challenge in the industry, particularly where compliance and regulation are concerned.

Recent controversies put the problem into sharp relief. The Royal Free London NHS Foundation Trust, a division of the U.K.'s National Health Service based in London, provided Alphabet's DeepMind with data on 1.6 million patients without their consent. Google — whose health data-sharing partnership with Ascension became the subject of scrutiny in November — abandoned plans to publish scans of chest X-rays over concerns that they contained personally identifiable information. This past summer, Microsoft quietly removed a data set (MS Celeb) with more than 10 million images of people after it was revealed that some weren't aware they had been included.



Separately, tech giants including Apple and Google have been the subject of reports uncovering the potential misuse of recordings collected to improve assistants like Siri and Google Assistant. In April, Bloomberg revealed that Amazon employs contract workers to annotate thousands of hours of audio from Alexa-powered devices, prompting the company to roll out user-facing tools that quickly delete cloud-stored data.

Increasingly, privacy isn't merely a question of philosophy, but table stakes in the course of business. Laws at the state, local, and federal levels aim to make privacy a mandatory part of compliance management. Hundreds of bills that address privacy, cybersecurity, and data breaches are pending or have already been passed in 50 U.S. states, territories, and the District of Columbia. Arguably the most comprehensive of them all — the California Consumer Privacy Act — was signed into law roughly two years ago. That's not to mention the Health Insurance Portability and Accountability Act (HIPAA), which requires companies to seek authorization before disclosing individual health information. And international frameworks like the EU's General Privacy Data Protection Regulation (GDPR) aim to give consumers greater control over personal data collection and use.

AI technologies have not historically been developed with privacy in mind. But a subfield of machine learning — privacy-preserving machine learning — seeks to pioneer approaches that might prevent the compromise of personally identifiable data. Of the emerging techniques, federated learning, differential privacy, and homomorphic encryption are perhaps the most promising.

Neural networks and their vulnerabilities

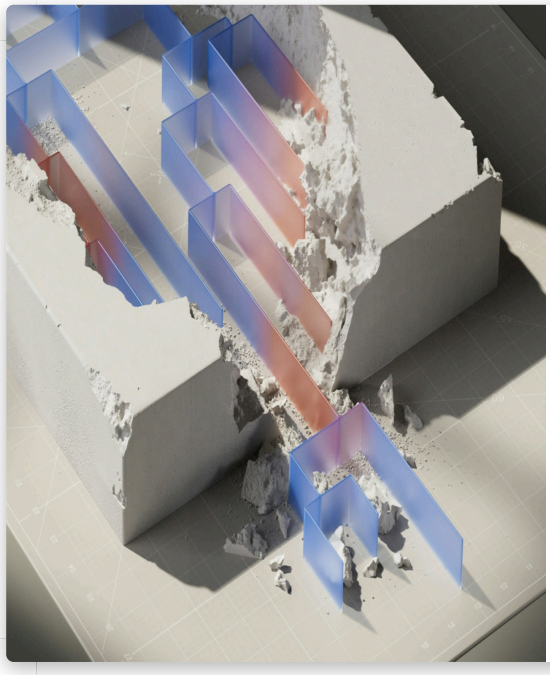
The so-called neural networks at the heart of most AI systems consist of functions (neurons) arranged in layers that transmit signals to other neurons. Those signals — the product of data, or inputs, fed into the network — travel from layer to layer and slowly "tune" the network, in effect adjusting the synaptic strength (weights) of each connection. Over time, the network extracts features from the data set and identifies cross-sample trends, eventually learning to make predictions.

Neural networks don't ingest raw images, videos, audio, or text. Rather, samples from training corpora are transformed algebraically into multidimensional arrays like scalars (single numbers), vectors (ordered arrays of scalars), and matrices (scalars arranged into one or more columns and one or more rows). A fourth entity type that encapsulates scalars, vectors, and matrices — tensors — adds in descriptions of valid linear transformations (or relations).

In spite of these transformations, it's often possible to discern potentially sensitive information from the outputs of the neural network. The data sets themselves are vulnerable, too, because they're not typically obfuscated, and because they're usually stored in centralized repositories that are vulnerable to data breaches.

By far the most common form of machine learning reverse engineering is called a membership inference attack, where an attacker — using a single data point or several data points — determines whether it belonged to the corpus on which a target model was trained. As it turns out, removing sensitive information from a data set doesn't mean it can't be re-inferred, because AI is exceptionally good at recreating samples. Barring the use of privacy-preserving techniques, trained models incorporate compromising information about whatever set they're fed.

In one study, researchers from the University of Wisconsin and the Marshfield Clinic Research Foundation were able to extract patients' genomic information from a machine learning model that was trained to predict medical dosage. In another, Carnegie Mellon and University of Wisconsin-Madison research scientists managed to reconstruct specific head shot images from a model trained to perform facial recognition.



AI Weekly

Your weekly look at how applied AI is changing the tech world

Email

Subscribe

We respect your privacy. Your email will only be used for sending our newsletter. You can unsubscribe at any time. Read our Privacy Policy.

which is intended to make it easier to experiment with deep learning and other computations on decentralized data.

Of course, no technique is without its flaws; federated learning requires frequent communication among nodes during the learning process. Tangibly, in order for the machine learning models to exchange parameters, they need significant amounts of processing power and memory. Other challenges include an inability to inspect training examples, as well as bias due in part to the fact that the AI models train only when power and a means of transmitting their parameters is available.

Differential privacy

Federated learning goes hand in hand with differential privacy, a system for publicly sharing information about a data set by describing patterns of groups within the corpus while withholding data about individuals. It usually entails injecting a small amount of noise into the raw data before it's fed into a local machine learning model, such that it becomes difficult for malicious actors to extract the original files from the trained model.

Intuitively, an algorithm can be considered differentially private if an observer seeing its output cannot tell if a particular individual's information was used in the computation. A differentially private federated learning process, then, enables nodes to jointly learn a model while hiding what data any node holds.

The open source TensorFlow library, TensorFlow Privacy, operates on the principle of differential privacy. Specifically, it fine-tunes models using a modified stochastic gradient descent that averages together multiple updates induced by training data examples, clips each of these updates, and adds noise to the final average. This prevents the memorization of rare details, and it offers some assurance that two machine learning models will be indistinguishable whether a person's data is used in their training or not.

Apple has been using some form of differential privacy since 2017 to identify popular emojis, media playback preferences in Safari, and more, and the company combined it with federated learning in its latest mobile operating system release (iOS 13). Both techniques help to improve the results delivered by Siri, as well as apps like Apple's QuickType keyboard and iOS' Found In Apps feature. The latter scans both calendar and mail apps for the names of contacts and callers whose numbers aren't stored locally.

For their part, researchers from Nvidia and King's College London recently employed federated learning to train a neural network for brain tumor segmentation, a milestone Nvidia claims is a first for medical image analysis. Their model uses a data set from the BraTS (Multimodal Brain Tumor Segmentation) Challenge of 285 patients with brain tumors, and as with the approaches taken by Google and Apple, it leverages differential privacy to add noise to that corpus.

"This way, [each participating node] stores the updates and limits the granularity of the information that we actually share among the institutions," Nicola Rieke, Nvidia senior researcher, told VentureBeat in a previous interview. "If you only see, let's say, 50% or 60% of the model updates, can we still combine the contributions in the way that the global model converges? And we found out 'Yes, we can.' It's actually quite impressive. So it's even possible to aggregate the model in a way if you only share 10% of the model."

Of course, differential privacy isn't perfect, either. Any noise injected into the underlying data, input, output, or parameters impacts the overall model's performance. In one study, after adding noise to a training data set, the authors noted a decline in predictive accuracy from 94.4% to 24.7%.

An alternative privacy-preserving machine learning technique — homomorphic encryption — suffers from none of those shortcomings, but it's far from an ace in the hole.

Homomorphic encryption

Homomorphic encryption isn't new — IBM researcher Craig Gentry developed the first scheme in 2009 — but it's gained traction in recent years, coinciding with advances in compute power and efficiency. It's basically a form of cryptography that enables computation on plaintext (file contents) encrypted using an algorithm (also known as ciphertexts), so that the generated encrypted result exactly matches the result of operations that would have been performed on unencrypted text. Using this technique, a "cryptonet" (e.g, any learned neural network that can be applied to encrypted data) can perform computation on data and return the encrypted result back to some client, which can then use the encryption key — which was never shared publicly — to decrypt the returned data and get the actual result.

"If I send my MRI images, I want my doctor to be able to see them immediately, but nobody else," Jonathan Ballon, vice president of Intel's IoT group, told VentureBeat in an interview earlier this year. "[Homomorphic] encryption delivers that, and in addition, the model itself is encrypted. So a company ... can put that model [on a public cloud], and that [cloud provider] has no idea what their model looks like."

In practice, homomorphic encryption libraries don't yet fully leverage modern hardware, and they're at least an order of magnitude slower than conventional models. But newer projects like cuHE, an accelerated encryption library, claim speedups of 12 to 50 times on various encrypted tasks over previous implementations. Moreover, libraries like PySyft and tf-encrypted — which are built on Facebook's PyTorch machine learning framework and TensorFlow, respectively — have made great strides in recent months. So, too, have abstraction layers like HE-Transformer, a backend for nGraph (Intel's neural network compiler) that delivers leading performance on some cryptonets.

In fact, just a few months ago, Intel researchers proposed nGraph-HE2, a successor to HE-Transformer that enables inference on standard, pretrained machine learning models using their native activation functions. They report in a paper that it was 3 times to 88 times faster at runtime in terms of scalar encoding (the encoding of a numeric value into an array of bits) with double the throughput, and that additional multiplication and addition optimizations yielded a further 2.6 times to 4.2 time runtime speedup.

IBM senior research scientist Flavio Bergamaschi has investigated the use of hardware at the edge to implement homomorphic encryption operations. In a recent study, he and colleagues deployed a local homomorphic database on a device equipped with an AI camera, enabling search to be performed directly on that camera. They report that performance was "homomorphically fast," with lookup taking only 1.28 seconds per database entry, which amounted to a 200-entry query in five minutes.

"We are at what I call inflection points in performance," he told VentureBeat in a recent phone interview. "Now, fully

homomorphic encryption is fast enough in terms of performance that it's perfectly adequate for certain use cases."

On the production side, Bergamaschi and team worked with a U.S.-based banking client to encrypt a machine learning process using homomorphic techniques. That machine learning process — a linear regression model with well over a dozen variables — analyzed 24 months of transaction data from current account holders to predict the financial health of those accounts, partly to recommend products like loans. Motivated by the client's privacy and compliance concerns, the IBM team encrypted the existing model and the transaction data in question, and they ran predictions using both the encrypted and unencrypted model to compare performance. While the former ran slower than the latter, the accuracy was the same.

"This is an important point. We showed that if we didn't have any model for [our] prediction, we could take transaction data and perform the training of a new model in production," Bergamaschi said.

Enthusiasm for homomorphic encryption has given rise to a cottage industry of startups aiming to bring it to production systems. Newark, New Jersey-based Duality Technologies, which recently attracted funding from one of Intel's venture capital arms, pitches its homomorphic encryption platform as a privacy-preserving solution for "numerous" enterprises, particularly those in regulated industries. Banks can conduct privacy-enhanced financial crime investigations across institutions, so goes the company's sales pitch, while scientists can tap it to collaborate on research involving patient records.

But like federated learning and differential privacy, homomorphic encryption offers no magic bullet. Even leading techniques can calculate only polynomial functions — a nonstarter for the many activation functions in machine learning that are non-polynomial. Plus, operations on encrypted data can involve only additions and multiplications of integers, which poses a challenge in cases where learning algorithms require floating point computations.

"In domains where you can take 10 seconds to turn around your inference, [homomorphic encryption] is fine, but If you need a three-millisecond turnaround time today, there's just no way to do it," Ballon said. "The amount of computation is too high, and this goes back to the domain of engineering."

Since 2014, Bergamaschi and colleagues have experimented with hardware approaches to accelerating homomorphic operations. Historically, bandwidth has been the biggest stumbling block — while accelerators yield strong benchmark performance individually, they don't yield strong systems performance overall. That's because the data required to perform the operations requires a lot of bandwidth between processors and the accelerator.

The solution might lie in techniques that make more efficient use of processors' on-chip memory. A paper published by researchers at the Korea Advanced Institute of Science and Technology advocates the use of a combined cache for all normal and security-supporting data, as well as memory scheduling and mapping schemes for secure processors and a type-aware cache insertion module. They say that together, the combined approaches could reduce encryption performance degradation from 25%-34% to less than 8%-14% in typical 8-core and 16-core secure processors, with minimal extra hardware costs.

A long way to go

New techniques might solve some of the privacy issues inherent in AI and machine learning, but they're in their infancy and not without their shortcomings.

Federated learning trains algorithms across decentralized edge devices without exchanging their data samples, but it's difficult to inspect and at the mercy of fluctuations in power, computation, and internet. Differential privacy, which exposes information about a data set while withholding information about the individuals, suffers dips in accuracy caused by injected noise. As for homomorphic encryption — a form of encryption that allows computation on encrypted data — it's somewhat slow and computationally demanding.

Nevertheless, folks like Ballon believe all three approaches are steps in the right direction. "This is very similar to going from HTTP to HTTPS," Ballon said. "We'll have the tools and capabilities to make [privacy in machine learning] seamless someday, but we're not quite there yet."