The way we train AI is fundamentally flawed

It's no secret that machine-learning models tuned and tweaked to near-perfect performance in the lab often fail in real settings. This is typically put down to a mismatch between the data the AI was trained and tested on and the data it encounters in the world, a problem known as data shift. For example, an AI trained to spot signs of disease in high-quality medical images will struggle with blurry or cropped images captured by a cheap camera in a busy clinic.

Now a group of 40 researchers across seven different teams at Google have identified another major cause for the common failure of machine-learning models. Called "underspecification," it could be an even bigger problem than data shift. "We are asking more of machine-learning models than we are able to guarantee with our current approach," says Alex D'Amour, who led the study.

Underspecification is a known issue in statistics, where observed effects can have many possible causes. D'Amour, who has a background in causal reasoning, wanted to know why his own machine-learning models often failed in practice. He wondered if underspecification might be the problem here too. D'Amour soon realized that many of his colleagues were noticing the same problem in their own models. "It's actually a phenomenon that happens all over the place," he says.

D'Amour's initial investigation snowballed and dozens of Google researchers ended up looking at a range of different AI applications, from image recognition to natural language processing (NLP) to disease prediction. They found that underspecification was to blame for poor performance in all of them. The problem lies in the way that machine-learning models are trained and tested, and there's no easy fix.

The paper is a "wrecking ball," says Brandon Rohrer, a machine-learning engineer at iRobot, who previously worked at Facebook and Microsoft and was not involved in the work.

Same but different

To understand exactly what's going on, we need to back up a bit. Roughly put, building a machine-learning model involves training it on a large number of examples and then testing it on a bunch of similar examples that it has not yet seen. When the model passes the test, you're done.

What the Google researchers point out is that this bar is too low. The training process can produce many different models that all pass the test but—and this is the crucial part—these models will differ in small, arbitrary ways, depending on things like the random values given to the nodes in a neural network before training starts, the way training data is selected or represented, the number of training runs, and so on. These small, often random, differences are typically overlooked if they don't affect how a model does on the test. But it turns out they can lead to huge variation in performance in the real world.

In other words, the process used to build most machine-learning models today cannot tell which

models will work in the real world and which ones won't.

This is not the same as data shift, where training fails to produce a good model because the training data does not match real-world examples. Underspecification means something different: even if a training process can produce a good model, it could still spit out a bad one because it won't know the difference. Neither would we.

The researchers looked at the impact of underspecification on a number of different applications. In each case they used the same training processes to produce multiple machine-learning models and then ran those models through stress tests designed to highlight specific differences in their performance.

For example, they trained 50 versions of an image recognition model on ImageNet, a dataset of images of everyday objects. The only difference between training runs were the random values assigned to the neural network at the start. Yet despite all 50 models scoring more or less the same in the training test—suggesting that they were equally accurate—their performance varied wildly in the stress test.

The stress test used ImageNet-C, a dataset of images from ImageNet that have been pixelated or had their brightness and contrast altered, and ObjectNet, a dataset of images of everyday objects in unusual poses, such as chairs on their backs, upside-down teapots, and T-shirts hanging from hooks. Some of the 50 models did well with pixelated images, some did well with the unusual poses; some did much better overall than others. But as far as the standard training process was concerned, they were all the same.

The researchers carried out similar experiments with two different NLP systems, and three medical AIs for predicting eye disease from retinal scans, cancer from skin lesions, and kidney failure from patient records. Every system had the same problem: models that should have been equally accurate performed differently when tested with real-world data, such as different retinal scans or skin types.

We might need to rethink how we evaluate neural networks, says Rohrer. "It pokes some significant holes in the fundamental assumptions we've been making."

D'Amour agrees. "The biggest, immediate takeaway is that we need to be doing a lot more testing," he says. That won't be easy, however. The stress tests were tailored specifically to each task, using data taken from the real world or data that mimicked the real world. This is not always available.

Some stress tests are also at odds with each other: models that were good at recognizing pixelated images were often bad at recognizing images with high contrast, for example. It might not always be possible to train a single model that passes all stress tests.

Multiple choice

One option is to design an additional stage to the training and testing process, in which many models are produced at once instead of just one. These competing models can then be tested again on specific real-world tasks to select the best one for the job.

That's a lot of work. But for a company like Google, which builds and deploys big models, it could be worth it, says Yannic Kilcher, a machine-learning researcher at ETH Zurich. Google

could offer 50 different versions of an NLP model and application developers could pick the one that worked best for them, he says.

D’Amour and his colleagues don’t yet have a fix but are exploring ways to improve the training process. “We need to get better at specifying exactly what our requirements are for our models,” he says. “Because often what ends up happening is that we discover these requirements only after the model has failed out in the world.”

Getting a fix is vital if AI is to have as much impact outside the lab as it is having inside. When AI underperforms in the real-world it makes people less willing to want to use it, says co-author Katherine Heller, who works at Google on AI for healthcare: “We've lost a lot of trust when it comes to the killer applications, that’s important trust that we want to regain.”