# First white-box testing model finds thousands of errors in self-driving cars

25 October 2017

Credit: CC0 Public Domain

How do you find errors in a system that exists in a black box? That is one of the challenges behind perfecting deep learning systems like self-driving cars. Deep learning systems are based on artificial neural networks that are modeled after the human brain, with neurons connected together in layers like a web. This web-like neural structure enables machines to process data with a non-linear approach—essentially teaching itself to analyze information through what is known as training data.

When an input is presented to the system after being "trained"—like an image of a typical two-lane highway presented to a self-driving car platform—the system recognizes it by running an analysis through its complex logic system. This process largely occurs in a black box and is not fully understood by anyone, including a system's creators.

Any errors also occur in a black box, making it difficult to identify them and fix them. This opacity presents a particular challenge to identifying corner case behaviors. A corner case is an incident that occurs outside normal operating parameters. A corner case example: a self-driving car system might be programmed to recognize the curve in a two-lane highway in most instances. However, if the lighting is lower or brighter than normal, the system may not recognize it and an error could occur. One recent example is the 2016 Tesla crash which was caused in part...

Shining a light into the black box of deep learning systems is what Yinzhi Cao of Lehigh University and Junfeng Yang and Suman Jana of Columbia University—along with the Columbia Ph.D. student Kexin Pei—have achieved with DeepXplore, the first automated white-box testing of such systems. Evaluating DeepXplore on real-world datasets, the researchers were able to expose thousands of unique incorrect corner-case behaviors. They will present their findings at the 2017 biennial ACM Symposium on Operating Systems Principles (SOSP) conference in Shanghai, China on October 29th in Session I: Bug Hunting.

"Our DeepXplore work proposes the first test coverage metric called 'neuron coverage' to empirically understand if a test input set has provided bad versus good coverage of the decision logic and behaviors of a deep neural network," says Cao, assistant professor of computer science and engineering.

In addition to introducing neuron coverage as a metric, the researchers demonstrate how a technique for detecting logic bugs in more traditional systems—called differential testing—can be applied to deep learning systems.

"DeepXplore solves another difficult challenge of requiring many manually labeled test inputs. It does so by cross-checking multiple DNNs and cleverly searching for inputs that lead to inconsistent results from the deep neural networks," says Yang, associate professor of computer science. "For instance, given an image captured by a self-driving

car camera, if two networks think that the car should turn left and the third thinks that the car should turn right, then a corner-case is likely in the third deep neural network. There is no need for manual labeling to detect this inconsistency."

The team evaluated DeepXplore on real-world datasets including Udacity self-driving car challenge data, image data from ImageNet and MNIST, Android malware data from Drebin, and PDF malware data from Contagio/VirusTotal, and production quality deep neural networks trained on these datasets, such as these ranked top in Udacity self-driving car challenge.

Their results show that DeepXplore found thousands of incorrect corner case behaviors (e.g., self-driving cars crashing into guard rails) in 15 state-of-the-art deep learning models with a total of 132, 057 neurons trained on five popular datasets containing around 162 GB of data.

The team has made their open-source software public for other researchers to use, and launched a website, DeepXplore, to let people upload their own data to see how the testing process works.

**More neuron coverage**

According to a paper to be published after the conference (see preliminary version here), DeepXplore is designed to generate inputs that maximize a deep learning (DL) system's neuron coverage.

The authors write: "At a high level, neuron coverage of DL systems is similar to code coverage of traditional systems, a standard metric for measuring the amount of code exercised by an input in a traditional software. However, code coverage itself is not a good metric for estimating coverage of DL systems as most rules in DL systems, unlike traditional software, are not written manually by a programmer but rather is learned from training data."

"We found that for most of the deep learning systems we tested, even a single randomly picked test input was able to achieve 100% code coverage—however, the neuron coverage was less than 10%," adds Jana, assistant professor of computer science.

The inputs generated by DeepXplore achieved 34.4% and 33.2% higher neuron coverage on average than the same number of randomly picked inputs and adversarial inputs (inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake) respectively.

**Differential testing applied to deep learning**

Cao and Yang show how multiple deep learning systems with similar functionality (e.g., self-driving cars by Google, Tesla, and Uber) can be used as cross-referencing oracles to identify erroneous corner-cases without manual checks. For example, if one self-driving car decides to turn left while others turn right for the same input, one of them is likely to be incorrect. Such differential testing techniques have been applied successfully in the past for detecting logic bugs without manual specifications in a wide variety of traditional software.

In their paper, they demonstrate how differential testing can be applied to deep learning systems.

Finally, the researchers' novel testing approach can be used to retrain systems to improve classification accuracy. During testing, they achieved up to 3% improvement in classification accuracy by retraining a deep learning model on inputs generated by DeepXplore compared to retraining on the same number of randomly picked or adversarial inputs.

"DeepXplore is able to generate numerous inputs that lead to deep neural network misclassifications automatically and efficiently," adds Yang. "These inputs can be fed back to the training process to improve accuracy."

Adds Cao: "Our ultimate goal is to be able to test a system, like self-driving cars, and tell the creators whether it is truly safe and under what conditions."

Provided by Lehigh University