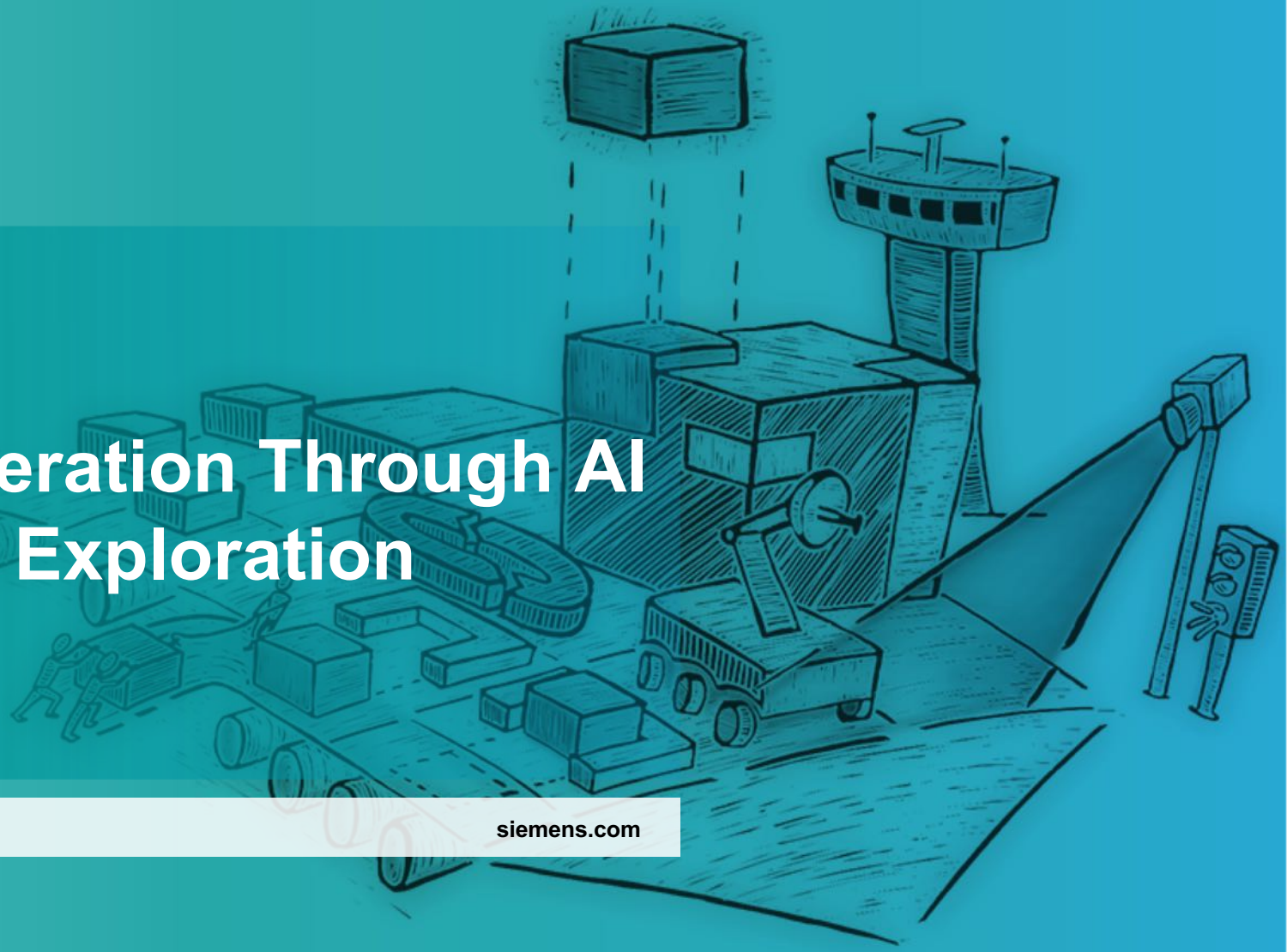
SIEMENS
Ingenuity for life

Guided Test Case Generation Through AI Enabled Output Space Exploration



SIEMENS CT RDA SSI AVI-US

Unrestricted © Siemens AG 2018

SIEMENS
Ingenuity for life

Problem Statement

Overview

SIEMENS
Ingenuity for life

Challenge

- Test data plays a crucial role in the overall software testing process
  - describes the initial conditions for a test
  - influence the behavior of the system under test
- Most test designs are based on test data generated either at random, or through systematic input space exploration, i.e. guided through fault ontology or some coverage metric

→ translates to highly non-deterministic probability and time frame for discovering relevant faults.

→ coverage of input space does not correlate to coverage of the output space.

Objective

- Systematically explore the SUT's output space to ensure an adequate coverage and to find new input/output pairs of interest, i.e.:
  - values, not covered by the existing test-suite, or
  - values where small changes to the input result in un-proportionally large changes of the output)

→ better chance of exposing system faults

Unrestricted © Siemens AG 2018

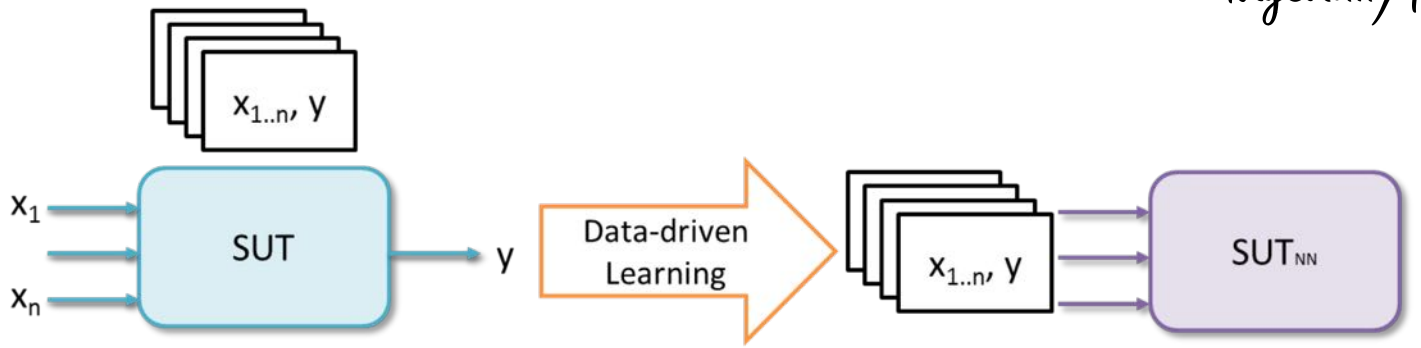
SIEMENS
Ingenuity for life

Guided Test Case Generation Through AI Enabled Output Space Exploration

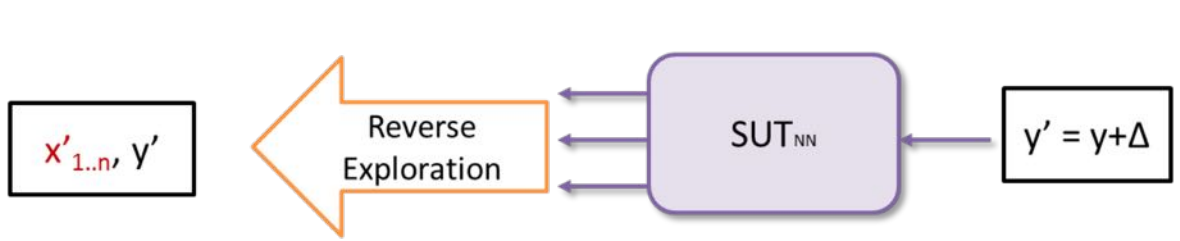
AI-enabled output-space exploration

SIEMENS
Ingenuity for life

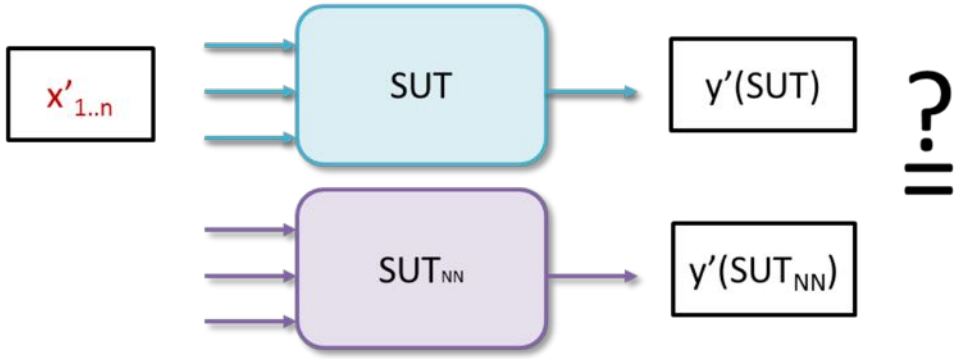
NN as SUT approximation



Output space exploration →
define of new outputs
Reverse exploration of NN →
obtain respective inputs



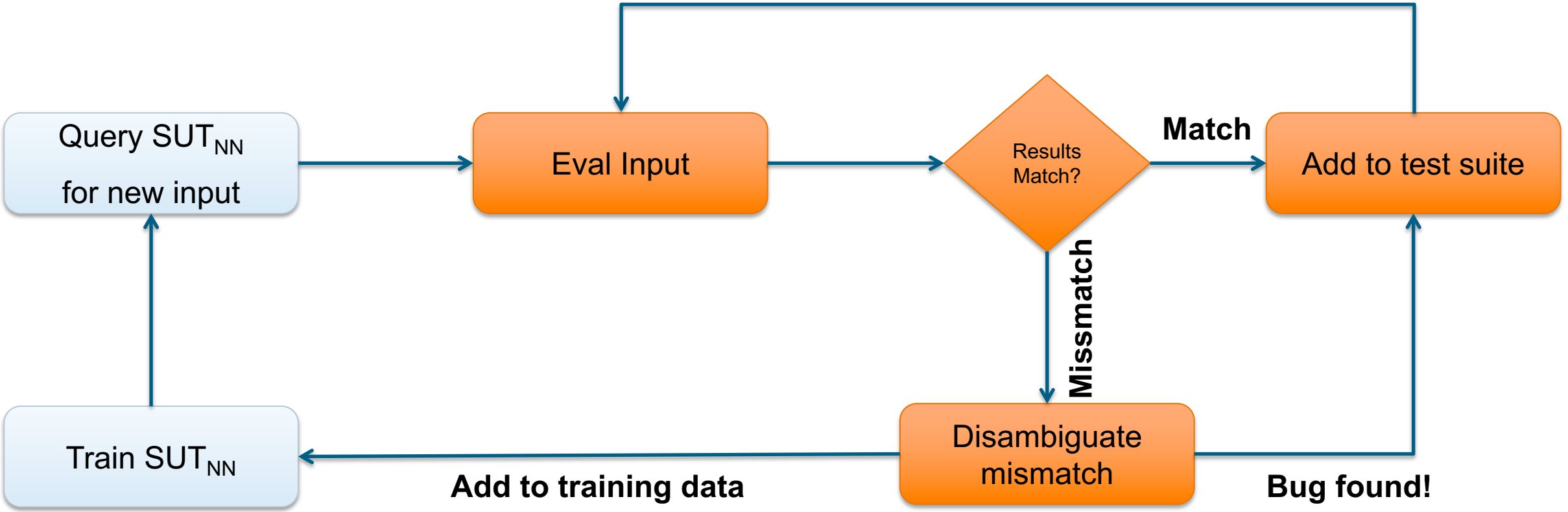
Use new inputs for testing



Unrestricted © Siemens AG 2017

The Evaluation Phase

SIEMENS
Ingenuity for life



Unrestricted © Siemens AG 2017

SIEMENS
Ingenuity for life

Pilot Context

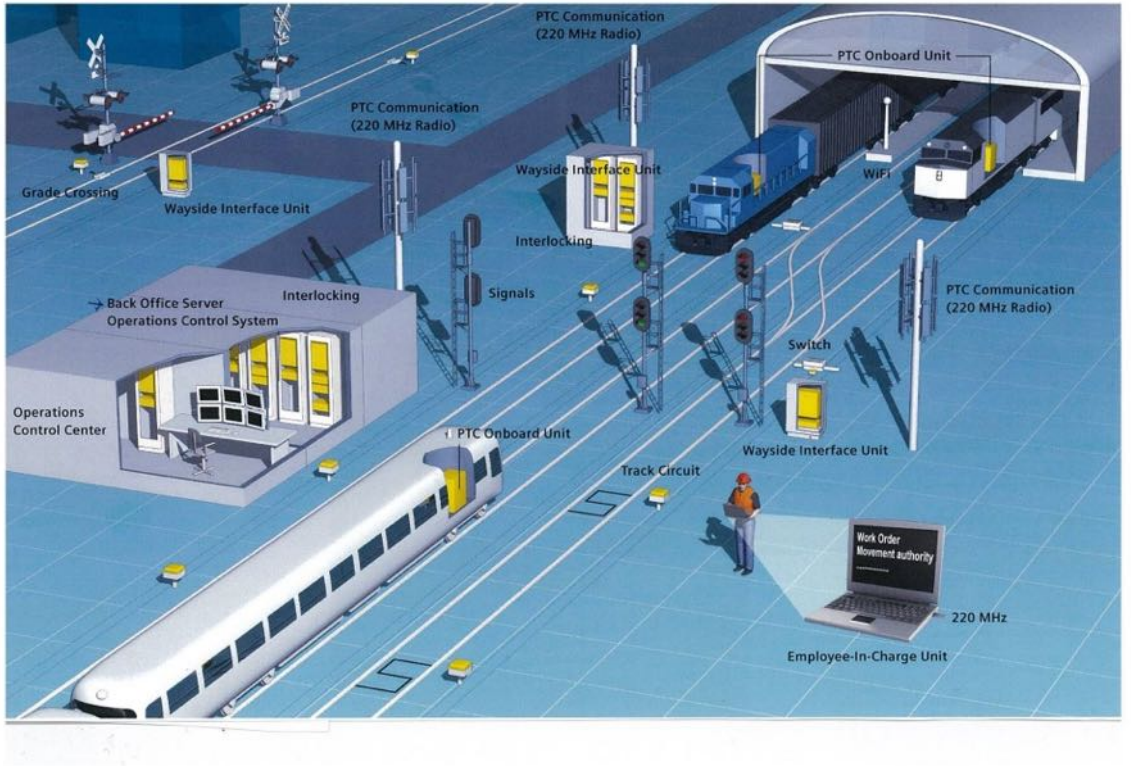
Research Context

Train Control System

an advanced train protection system designed to monitor and control train movements with the goal to:

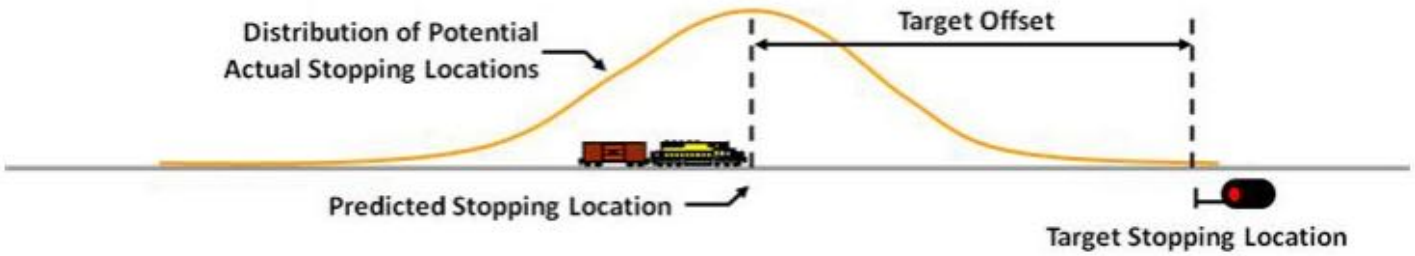
1. increase safety
2. reduce infrastructure utilization
3. increase operational efficiency

by automatically stopping the train in cases where the train engineer fails to act according to protocol.



Challenges

- high level of complexity, heterogeneity, and sensitivity to the uncertainties of sensor data (e.g. positioning information)
- safety-critical



Unrestricted © Siemens AG 2018

Pilot approach

SIEMENS
Ingenuity for life

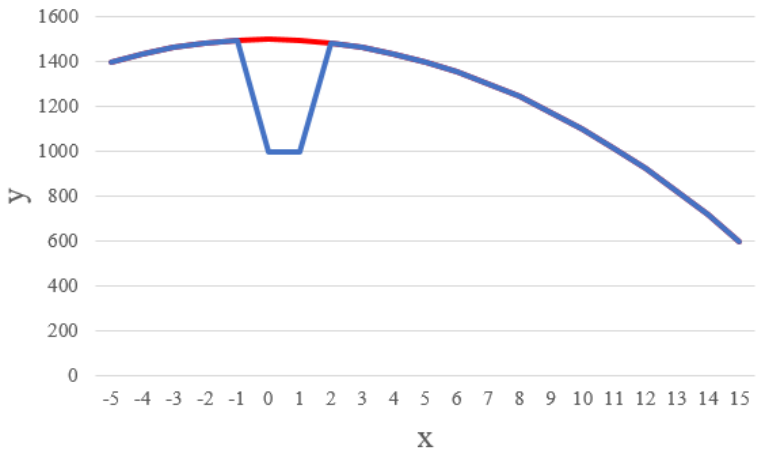
Step 1: Train

- Obtain Test Data
  - 2000 Observation with 80/20 split
- Neural Network Architecture & Training
  - 2-layer NN (1 fully-connected layer with 5 neurons) in TensorFlow
    - Inference: ReLU non-linearity applied to hidden layer, identity function to output layer
    - Loss Measurement: Mean Square Error
    - Optimizer: ADAM

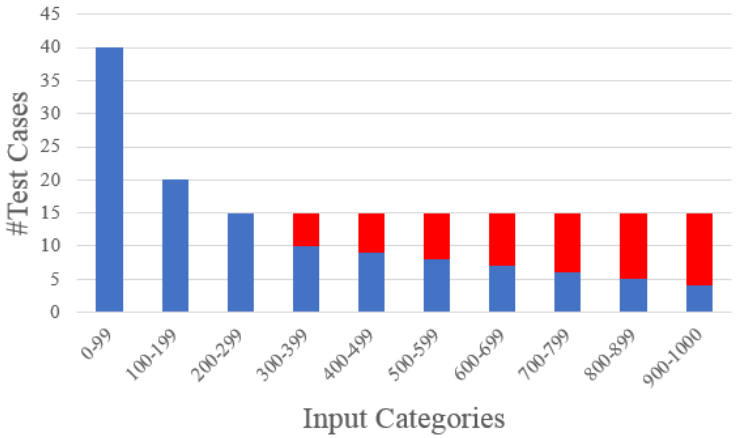
Step 2: Query

- Exploration Strategies
- Generation of Adversaries

Step 3: Evaluate



Functional Continuity Strategy



Output Categorization Strategy

Unrestricted © Siemens AG 2017

Query NN

SIEMENS
Ingenuity for life

- **Synthesizing an adversarial example:**
  - Use of FGSM (fast gradient sign method)
    - constrained minimization within a predefined max. change budget - ε.
  - Given a new breaking distance y', i.e. $target\_output = ([208.564966],)$
  - try to find an adversarial input x'
    - constrained by some distance metric used to quantify similarity (ε), so that $\|x - x'\| \leq \varepsilon$.
  - At the same time → fix the model
    - already trained w & b won't change
    - the only thing left to modify is the input vector x.

Published as a conference paper at ICLR 2015

EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES

Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy
Google Inc., Mountain View, CA
{goodfellow,shlens,szegedy}@google.com

arXiv:1412.6572v3 [stat.ML] 20 Mar 2015

ABSTRACT

Several machine learning models, including neural networks, consistently misclassify *adversarial examples*—inputs formed by applying small but intentionally worst-case perturbations to examples from the dataset, such that the perturbed input results in the model outputting an incorrect answer with high confidence. Early attempts at explaining this phenomenon focused on nonlinearity and overfitting. We argue instead that the primary cause of neural networks' vulnerability to adversarial perturbation is their linear nature. This explanation is supported by new quantitative results while giving the first explanation of the most intriguing fact about them: their generalization across architectures and training sets. Moreover, this view yields a simple and fast method of generating adversarial examples. Using this approach to provide examples for adversarial training, we reduce the test set error of a maxout network on the MNIST dataset.

1 INTRODUCTION

Szegedy et al. (2014b) made an intriguing discovery: several machine learning models, including state-of-the-art neural networks, are vulnerable to *adversarial examples*. That is, these machine learning models misclassify examples that are only slightly different from correctly classified examples drawn from the data distribution. In many cases, a wide variety of models with different architectures trained on different subsets of the training data misclassify the same adversarial example. This suggests that adversarial examples expose fundamental blind spots in our training algorithms.

The cause of these adversarial examples was a mystery, and speculative explanations have suggested it is due to extreme nonlinearity of deep neural networks, perhaps combined with insufficient model averaging and insufficient regularization of the purely supervised learning problem. We show that these speculative hypotheses are unnecessary. Linear behavior in high-dimensional spaces is sufficient to cause adversarial examples. This view enables us to design a fast method of generating adversarial examples that makes adversarial training practical. We show that adversarial training can provide an additional regularization benefit beyond that provided by using dropout (Srivastava et al., 2014) alone. Generic regularization strategies such as dropout, pretraining, and model averaging do not confer a significant reduction in a model's vulnerability to adversarial examples, but changing to nonlinear model families such as RBF networks can do so.

Our explanation suggests a fundamental tension between designing models that are easy to train due to their linearity and designing models that use nonlinear effects to resist adversarial perturbation. In the long run, it may be possible to escape this tradeoff by designing more powerful optimization methods that can successfully train more nonlinear models.

2 RELATED WORK

Szegedy et al. (2014b) demonstrated a variety of intriguing properties of neural networks and related models. Those most relevant to this paper include:

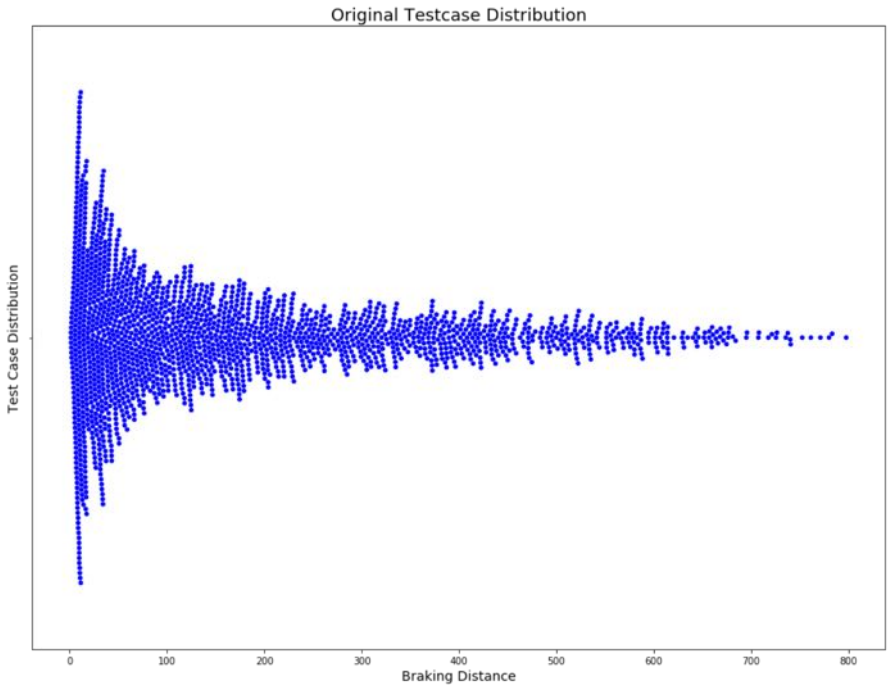
- Box-constrained L-BFGS can reliably find adversarial examples.
- On some datasets, such as ImageNet (Deng et al., 2009), the adversarial examples were so close to the original examples that the differences were indistinguishable to the human eye.
- The same adversarial example is often misclassified by a variety of classifiers with different architectures or trained on different subsets of the training data.

Explaining and Harnessing Adversarial Examples.pdf

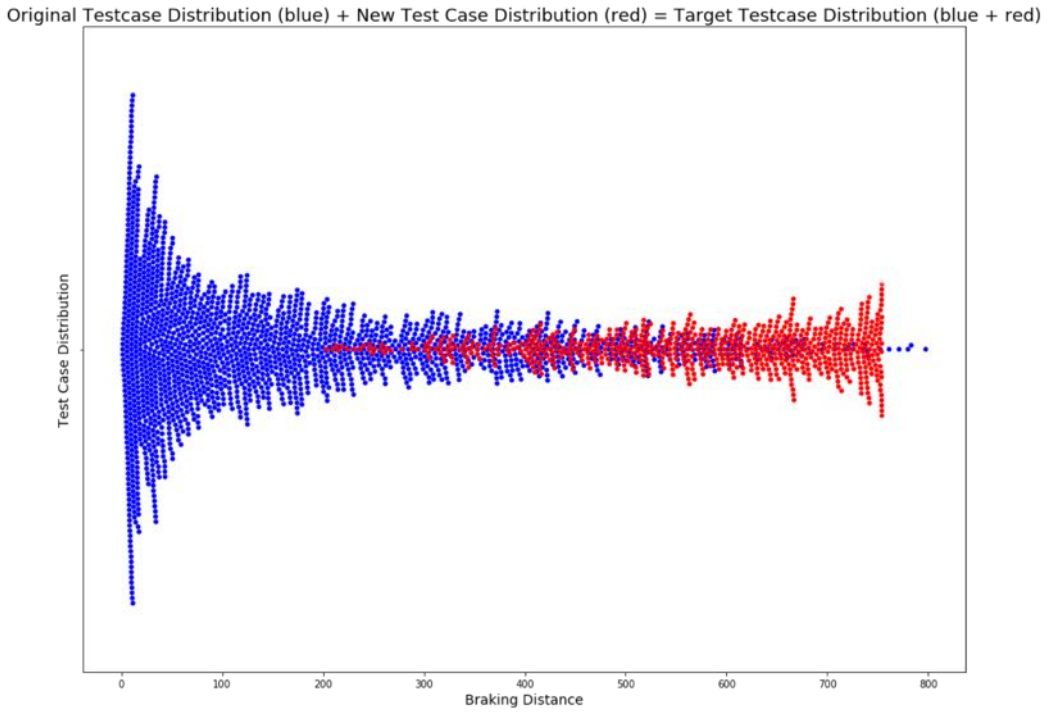
Unrestricted © Siemens AG 2018

Pilot results

- Original Coverage obtained from 2000 Test Cases generated at random
  - Non-uniform distribution in output space → Coverage in the category (700 – 800ft.) - <2%
- Coverage increase to at least 250 data points per category after application of our method



Original Coverage



New Test Case Distribution

Unrestricted © Siemens AG 2017

SIEMENS
Ingenuity for life

Summary

Summary, Next steps & Transfer to the BUs

SIEMENS
Ingenuity for life

Summary

- Ensured significant increase of output-space coverage
- Initial findings of deviations between results from SUT and NN on newly generated datasets

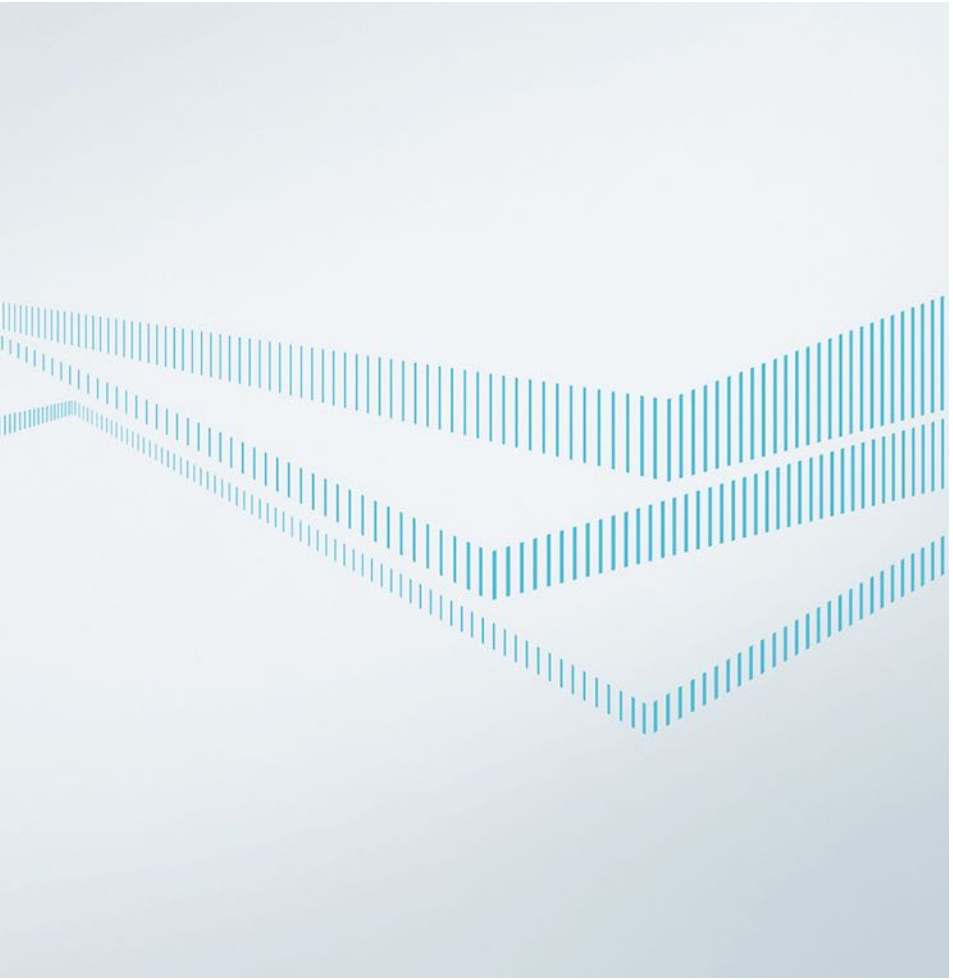
Next Steps

- Addressing some open questions:
  - Stateful vs stateless problems
  - Mapping real data to NN input layer
- Automated test oracle & test data generation
- Efficiency (vs customer's Monte-Carlo Simulation approach) & Effectiveness benchmark (Mutation Test)

Unrestricted © Siemens AG 2017

Contact

SIEMENS
Ingenuity for life



Georgi Markov | **Marco Gario** | **Christof Budnik** | **Zhu Wang**

CT RDA SSI AVI-US
Test Architect
Architecture and Verification of Intelligent Systems

755 College Rd E
Princeton, NJ 08540
USA
Phone: +1 (609) 216 6403

E-mail: georgi.markov@siemens.com

Unrestricted © Siemens AG 2018