# CSE Research and Technology Forum 2022

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

# May my AI systems be better assured?

Shing-Chi Cheung

Supported by ITC under MHP-055-19

# Sound familiar?

# A Common Issue in AI Systems

Does my AI system perform reliably?

# Defects in AI systems can bring great loss

- Defects occurring at deployment can pose threats to lives and economy
- Defects occurring at training can waste weeks or months of valuable computational resources
  - Result in NaN, crashes and gradient vanishing after a long training period

2018/3/19: Uber car hit a pedestrian, causing death due to incorrect object detection

```
[[ nan  nan]
 [ nan  nan]
 [ nan  nan]
 [ nan  nan]
 [ nan  nan]
 [ nan  nan]
 [ nan  nan]
 [ nan  nan]
 [ nan  nan]
 [ nan  nan]]
```

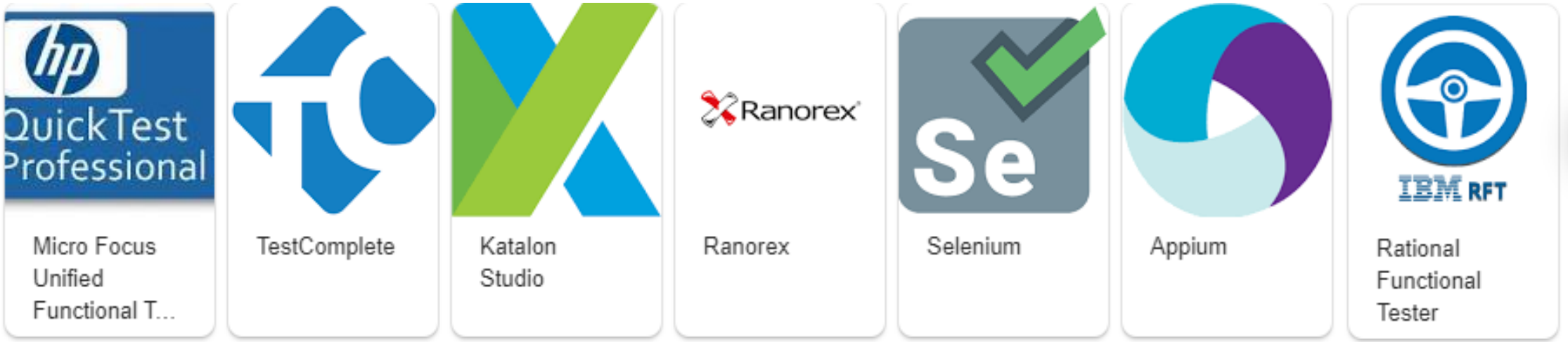
# How frequent are real-life AI projects updated?

| Deep Learning Project | Code Size | #Code Commits | #Commits in recent 6 months | #Issues reported in recent 6 month |
|---|---|---|---|---|
| Transformer | 29,409 | 1,971 | 1,549 | 891 |
| DeepSpeech | 145,167 | 2,108 | 594 | 252 |
| Real-Time-Voice-Clone | 5,126 | 237 | 148 | 148 |
| DeepCreamPy | 878 | 376 | 86 | 37 |

Source: Four most popular active TensorFlow projects on Github
Figures based on end of 2019

Can bugs in neural networks be detected before training?

# Absence of testing tools for DL systems




- There are commercial testing tools for conventional software
- BUT none are designed for AI systems or modules

# First empirical study on bugs in DL programs

| | StackOverflow QAs | Github Projects |
|---|---|---|
| Counts | 87 | 88 |

## StackOverflow QAs

- Searched Tensorflow related questions
- Manually reviewed QA pages
- Analyzed answers and discussions

## Github Projects

- Searched Tensorflow related projects
- Manually reviewed commits
- Analyzed commit/pull request messages and issue discussions

Yuhao Zhang, Yifan Chen, Shing-Chi Cheung, Yingfei Xiong, Lu Zhang. An Empirical Study on TensorFlow Program Bugs. In ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA2018).

# Selected findings from popular Github projects

- Failures often occur at the training stage and after many training cycles
- Testing one training instance alone is unlikely to catch such failures
- When failures occur, the error messages are often confusing
  - Error messages may not pinpoint which parts of the software go wrong
  - Discussions at StackOverflow suggest that fault determination is non-trivial even for a small AI system

# A common issue in many AI projects

- API misuses are common
  - API mostly designed for numeric computation
  - API documentation is either brief or difficult to follow

```
h_fc3 = tf.nn.relu(conv2d(h_fc1_drop, W_fc2) + b_fc2)
y_conv = tf.nn.softmax(tf.reshape(h_pool3, [-1, 10]))
cross_entropy = -tf.reduce_sum(y_*tf.log(y_conv))
train_step = tf.train.AdamOptimizer(1e-4).minimize(cross_entropy)
```

Stack Overflow #33699174

  - API often evolves to meet dynamic market demand and algorithm advancement

# DEBAR: Scalable AI Defect Analyzer

## We propose two abstraction techniques:

1. Tensor Partitioning

   Numeric computation can be abstracted using intervals
   Many tensor elements are subject to the same computation

2. Interval Abstraction with Affine Equality Relation

   Many computations are affine operations ($w_0 = \sum_i w_i x_i$)
   -> affine equality relations to enhance the precision of interval abstraction


Distinguished paper award
ESEC/FSE 2020

Tool implemented in Python
https://github.com/ForeverZyh/DEBAR

# Main results

## Framework

(Tensor Abstraction + Numerical Abstraction)

## Our technique

(Tensor Partitioning + Affine Equality Relation):
Accuracy: 93.0%, all in 3 minutes, 12.1s on average
100% accuracy on 9 buggy architectures

Tensor Partitioning + Sole Interval Abstraction
Accuracy: 80.6%, 12.1s on average

expand every element
$\sigma(A) = \begin{pmatrix} [0,1] & [-1,0] \\ [-1,0] & [1,1] \end{pmatrix}$

**Tensor Expansion** + Affine Equality Relation:
33/57 > 30mins; on rest 24, DEBAR doesn't lose accuracy

**Tensor Smashing** + Affine Equality Relation:
Accuracy: 87.1%, 12.2s on average

Smash a tensor into an element
$\sigma(A) = [-2,2]$

$$Accuracy = \frac{\#TP+\#TN}{\#TP+\#TN+\#FP+\#FN}$$

# CASTLE Group Members

Shing-Chi Cheung
Professor at HKUST

Chang Xu
Professor at NJU

Yepang Liu
Asistant Professor at SUSTech

Rongxin Wu
Associate Professor at Xiamen University

Ming Wen
Associate Professor at HUST

Valerio Terragni
Lecturer at The University of Auckland

Yanyan Jiang
Associate Researcher at NJU

Ying Wang
Associate Professor at NEU

Mijung Kim
Assistant Professor at UNIST

Lili Wei
Postdoc at HKUST

## Students

Jue Wang
PhD Student at NJU

Huaxun Huang
PhD Student at HKUST

Yongqiang Tian
PhD Student at HKUST&UWaterloo

Cong Li
PhD Student at NJU

Jiarong Wu
PhD Student at HKUST

Wuqi Zhang
PhD Student at HKUST

Lu Liu
PhD Student at HKUST

Yuqing Quan
MPhil Student at HKUST

Jialun Cao
PhD Student at HKUST

Meiziniu Li
PhD Student at HKUST

Hengcheng Zhu
MPhil Student at HKUST

David Mak
MPhil Student at HKUST

Haoyang Ma
PhD Student at HKUST

# CASTLE Group - Code AnalySe, Testing and LEarning

http://castle.cse.ust.hk/castle/people.html