Research objective

Therefore, we study 2716 high-quality posts from Stack Overflow and 500 bug fix commits from Github about five popular deep learning libraries Caffe, Keras, Tensorflow, Theano, and Torch to understand the types of bugs, root causes of bugs, impacts of bugs, bug-prone stage of deep learning pipeline as well as whether there are some common antipatterns found in this buggy software.

RQ1: (Bug Type) What type of bugs are more frequent?

RQ2: (Root cause) What are the root causes of bugs?

RQ3: (Bug Impact) What are the frequent impacts of bugs?

RQ4: (Bug prone stages) Which deep learning pipeline stages are

more vulnerable to bugs?

RQ5: (Commonality) Do the bugs follow a common pattern?

RQ6: (Bug evolution) How did the bug pattern change over time?

Motivation

A class of machine learning algorithms known as deep learning has received much attention in both academia and industry. These algorithms use multiple layers of transformation functions to con- vert input to output, each layer learning successively higher-level of abstractions in the data. The availability of large datasets has made it feasible to train (adjust the weights of) these multiple lay- ers. While the jury is still out on the impact of deep learning on overall understanding of software’s behavior, a significant uptick in its usage and applications in wide ranging areas combine to warrant research on software engineering practices in the presence of deep learning. This work focuses on the characteristics of bugs in software that makes use of deep learning libraries.

Existing work focus on this

Previous work on this topic generally fall under two categories: those that have studied bugs in the implementation of machine learning libraries themselves, and those that have studied bugs in the usage of a specific deep learning library. A key work in the first category is Thung et al. [24] who studied bugs in the implementation of three machine learning systems Mahout, Lucene, and OpenNLP. In the second category, Zhanget al.[27] have studied bugs in software that make use of the Tensorflow library. While both categories of approaches have advanced our knowledge of ML systems, we do not yet have a comprehensive understanding of bugs encountered by the class of deep learning libraries.

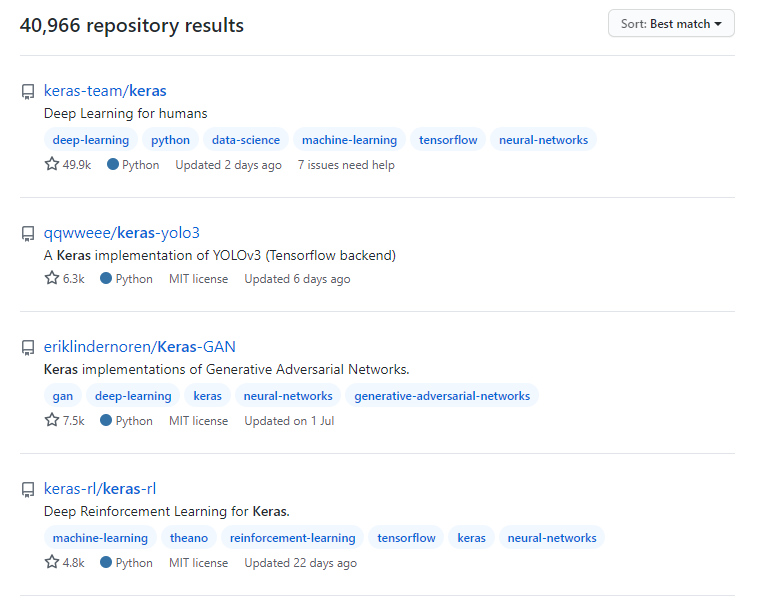
Method

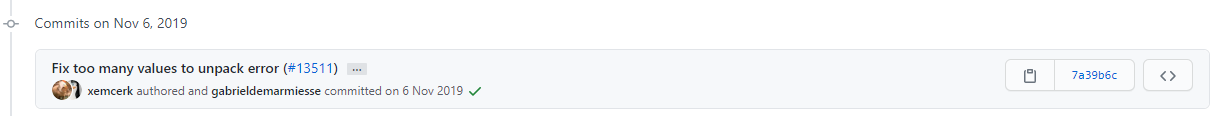
Data collection

1. Stackoverflow

* Searching for posts tagged with Caffe, Keras, Tensorflow, Theano, and Torch.
* Filter out posts that did not contain any source code because posts about bugs usually contain code snippets
* Grade more than 5
* Manually read: If the best-accepted answer was to fix the usages of the deep learning API(s) in the question, we considered that post as talking about deep learning bugs.

1. Github







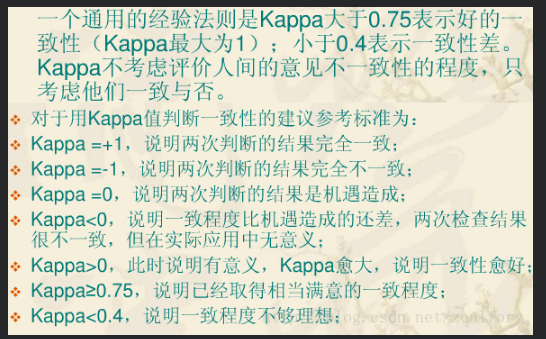


Classification

bug types, root causes and effects of bug

Labeling the bugs

Cohen’s Kappa coefficient



Types of Bugs

1. API Bug

The prime causes for triggering of deep learning API bugs can be because of the change of API definition with different versions, lack of inter-API compatibility and sometimes wrong or confusing documentation.

1. Coding Bug

programming mistakes.

1. Data Bug

an input to the deep learning software is not properly formatted or cleaned well

1. Structural Bug
   1. control and sequence bug.

wrong structure of control flow.

* 1. Dataflow bug

If a bug occurs due to the type or shape mismatch of input data after it has been fed to the deep learning model, we label it as Data Flow Bug

* 1. Initialization bug

Initialization Bug means the parameters or the functions are not initialized properly before they are used

* 1. Logic bug

an incorrect logical structure of the deep learning model

* 1. Processing bug

to make different

layers be compatible with each other, the data types of each layer

need to follow a contract between them. Processing Bugs happen

due to the violation of these contracts.

1. Non Model Structural Bug

this bug can happen in any deep learning stage except the modeling stage such as the training stage or the prediction stage.

root causes

1. Absence of Inter API Compatibility.
2. Absence of Type Checking.
3. API Change.
4. API Misuse
5. Confusion with Computation Model
6. Incorrect Model Parameter or Structure
7. Others.
8. Structure Inefficiency
9. Unaligned Tensor
10. Wrong Documentation.

Effects of Bugs

1. Bad Performance
2. Crash
3. Data Corruption
4. Hang
5. Incorrect Functionality.
6. Memory Out of Bound.