Developer Study on Reproducibility of Deep Learning Bugs

Hello there!

Welcome to this survey! We are a group of researchers from Dalhousie University, Canada. Recently, we conducted an empirical study involving 85 reproducible bugs from Stack Overflow posts. Our aim was to understand two main aspects: (1) the edit actions that can be employed to complete code snippets for bug reproduction and (2) the information that enhances the reproducibility of bug reports. Our investigation has yielded several interesting findings, and we are seeking your feedback on them. We will share our findings with you once the study is completed.

* Indicates required question

We reproduced 85 bugs and found out that they could be reproduced using 10 edit actions. To enhance their reproducibility, there are 5 main information categories that need to be present. The edit actions and information categories are described below.

Edit Actions

1. Input Data Generation: Generating input data which simulates the data used for training the model.
2. Neural Network Construction: Reconstructing or modifying the neural network based on the information provided
3. Hyperparameter Initialization: Initializing the hyperparameters for training, such as batch size and number of epochs
4. Import Addition and Dependency Resolution: Determining the dependencies in the code snippet and adding the missing imports.
5. Logging: Adding appropriate logging statements to capture relevant information during reproduction
6. Obsolete Parameter Removal: Removing outdated parameters or functions to match the parameters of the latest library versions
7. Compiler Error Resolution: Debugging and resolving compiler errors that arise due to the errors in the provided code snippet.
8. Dataset Procurement: Acquiring the datasets and using them to train the model
9. Downloading Models & Tokenizers: Fetching pre-trained models and tokenizers from external sources.
10. Version Migration: Updating the code to adapt the changes introduced in newer library or framework versions.

Information Categories:

Data: Shape of the input data, type of data, data distribution.

Model: Neural network architecture, number of layers, neurons, activation function for layers.

Hyperparameters: Batch size, epochs, optimizers, loss function.

Code Snippet: Training code snippet, evaluation script, data preprocessing, and transformation operations.

Logs: Compiler error logs, training error logs

Demographics

As part of this developer study, we will require certain demographic information from you to facilitate further analysis.

1. Q1: What is your relevant work experience with deep learning? *

Mark only one oval.

○ <1 Year

○ 1-5 Years

○ 5-10 Years

○ >10 Years

2. Q2: What is your relevant experience with deep learning bug fixing?

Mark only one oval.

○ <1 Year

○ 1-5 Years

○ 5-10 Years

○ >10 Years

3. Q3: What is your current occupation?

Mark only one oval.

○ Software Practitioner (Software Engineer, Deep Learning Engineer, Machine Learning Engineer etc.)

○ Researcher (Masters/Doctoral Student, PostDoc, Faculty)

4. Q4: What are the deep learning frameworks you have worked with?

Tick all that apply.

□ Tensorflow

□ PyTorch

□ Keras

□ Other:

5. Q5: What challenges are associated with reproducing deep learning bugs in your day-to-day activities?

Bug #1

Given the issue description, and the code snippet. Please reproduce the bug, and select the most appropriate edit operations and critical information needed to reproduce this bug.

To help the reproduction process, we have provided the sample edit operations here.

Original Issue Report: https://stackoverflow.com/questions/59278771/super-low-accuracy-for-neural-network-model

Description:
I followed a tutorial on neural network model evaluation using cross-validation with code (given below). The accuracy was supposed to be around 95.33% (4.27%) but I got ~Accuracy: 34.00% (13.15%) on a few attempts. The model code seems exactly the same. I downloaded the data from here as instructed. What could go wrong? Thanks

Code Snippet: You can use this Colab notebook as the base notebook to start the reproduction process: https://colab.research.google.com/drive/1CH0EKq3Wc2ctcw1kWvAzxe3O7i-FA05f?usp=sharing

6. Q1: What are the edit operations that could be used to reproduce this bug? *

Tick all that apply.

□ Input Data Generation

□ Neural Network Construction

□ Hyperparameter Initialization

□ Import Addition and Dependency Resolution

□ Logging

□ Obsolete Parameter Removal

□ Compiler Error Resolution

□ Dataset Procurement

□ Downloading Models and Tokenizers

□ Version Migration

□ Other:

7. Why do you think these edit operations could prove useful in reproducing the bug? *

8. Q2: What are the critical information components that could help the reproducibility of this bug? *

Tick all that apply.

□ Data
□ Hyperparameters
□ Model
□ Code Snippet
□ Logs
□ Other: ____________________

9. How do you think the selected critical information could be useful in reproducing the bug? *

10. Q3: Did you implement any additional operations or actions beyond those suggested by us? Please let us know your thoughts.

Skip to question 11

Bug #2

Given the issue description, and the code snippet. Please reproduce the bug, select the most appropriate edit operations and critical information needed to reproduce this bug.

To help the reproduction process, we have provided the sample edit operations here.

Original Issue Report: https://stackoverflow.com/questions/39525358/neural-network-accuracy-optimization

Description:
I have constructed an ANN in keras which has 1 input layer(3 inputs), one output layer (1 output) and two hidden layers with with 12 and 3 nodes respectively, as shown below in the code.

The dataset has 4 columns: 3 columns with values in the range [60, 70] and the target variable is binary (0/1 output)

so after 150 epochs i get: **loss: 0.6932 - acc: 0.5000 - val_loss: 0.6970 - val_acc: 0.1429**

My question is: how could i modify my NN in order to achieve higher accuracy?

Code Snippet: You can use this Colab notebook as the base notebook to start the reproduction process: https://colab.research.google.com/drive/1O8y5vYDP7ODPcvi1cGNraMOP8iXxlLhM?usp=sharing

11. Q1: What are the edit operations that could be used to reproduce this bug? *

Tick all that apply.

□ Input Data Generation
□ Neural Network Construction
□ Hyperparameter Initialization
□ Import Addition and Dependency Resolution
□ Logging
□ Obsolete Parameter Removal
□ Compiler Error Resolution
□ Dataset Procurement
□ Downloading Models and Tokenizers
□ Version Migration
□ Other: ______________________

12. Why do you think these edit operations could prove useful in reproducing the bug? *

13. Q2: What are the critical information components that could help the reproducibility of this bug? *

Tick all that apply.

□ Data
□ Hyperparameters
□ Model
□ Code Snippet
□ Logs
□ Other: ______________________

14. How do you think the selected critical information could be useful in reproducing the bug? *

15. Q3: Did you implement any additional operations or actions beyond those suggested by us? Please let us know your thoughts.