A. Artifact Appendix

A.1 Abstract

This artifact description contains information about the functional workflow required to run our proposed Bio-FO with the PyTorch library and just CPU support for artifact evaluation out of the box. We provide the MNIST dataset to validate the results from our paper to make sure that our artifact is functional.

A.2 Artifact check-list (meta-information)

- **Algorithm:** A biologically-plausible forward-only algorithm (Bio-FO) for efficient on-device machine learning.
- **Program:** PyTorch; Version: torch-1.12.0.post2, torchvision-0.16.0; Python: 3.9.13.
- **Compilation:** NA.
- **Transformations:** NA.
- **Binary:** NA.
- **Data set:** The MNIST dataset.
- **Run-time environment:** not OS-specific.
- **Hardware:** CPU support.
- **Run-time state:** NA.
- **Execution:** NA.
- **Metrics:** Test accuracy.
- **Output:** Accuracy of the network on the 10000 test images: 98.43%.
- **Experiments:** README.md; variation of empirical results: 98.38±0.08 (%).
- **How much disk space required (approximately)?:** 200 MB.
- **How much time is needed to prepare workflow (approximately)?:** 10 mins.
- **How much time is needed to complete experiments (approximately)?:** 1 hour.
- **Publicly available?:** zip file or GitHub.
- **Code licenses (if publicly available)?:** MIT License.
- **Data licenses (if publicly available)?:** NA.
- **Workflow framework used?:** NA.
- **Archived (provide DOI)?:** I will provide it at the end of the evaluation for my final Artifact Appendix in case of any advice suggested from AE.

A.3 Description

A.3.1 How delivered

The artifact is attached in the zip file or GitHub..

A.3.2 Hardware dependencies

We provide Bio-FO with just CPU support for artifact evaluation out of the box.

A.3.3 Software dependencies

There are no OS-specific dependencies. The artifact uses Python (3.9.13) and PyTorch, where the requirements are torch-1.12.0.post2 and torchvision-0.16.0.

A.3.4 Data sets

We load the the MNIST dataset from torchvision. The example is shown below:

```
from torchvision.datasets import MNIST
train_loader = DataLoader(
    MNIST('./data/', train=True,
          download=True,
          transform=transform),
          batch_size=train_batch_size, shuffle=False)
```

A.4 Installation

In terminal on MacOS, the command is shown below:

```
pip3 install torch==1.12.0.post2 torchvision==0.16.0
```

A.5 Experiment workflow

The artifact only needs to run the "python BioFO.py", where the dataset will be downloaded automatically and loaded.

A.6 Evaluation and expected result

The output will be:

```
Accuracy of the network in 3 layer
on the 10000 test images: 98.43\%
```

The variation of empirical results: 98.38±0.08 (%), as presented in Table 1 of our paper (1.62±0.08 (%) in terms of error).

A.7 Experiment customization

The full repository (including benchmarks on all datasets) is mentioned at README.md from zip file or GitHub.

A.8 Notes

I just provide the MNIST dataset for artifact evaluation (Available and Functional) out of the box. I am open to providing more benchmarks and more datasets with NVIDIA Jetson Nano for artifact evaluation (Reproduced) if needed.

A.9 Methodology

Submission, reviewing and badging methodology:

- `http://cTuning.org/ae/submission-20190109.html`
- `http://cTuning.org/ae/reviewing-20190109.html`
- `https://www.acm.org/publications/policies/artifact-review-ba`