OPTIMIZING DEEP LEARNING INFERENCE: *REPLICATION OF ADAPTIVE MODEL SELECTION ON IMAGE CLASSIFICATION* [1]

Anel Mengdigali & Vladislav Yarovenko

anel.mengdigali@nu.edu.kz

vladislav.yarovenko@nu.edu.kz

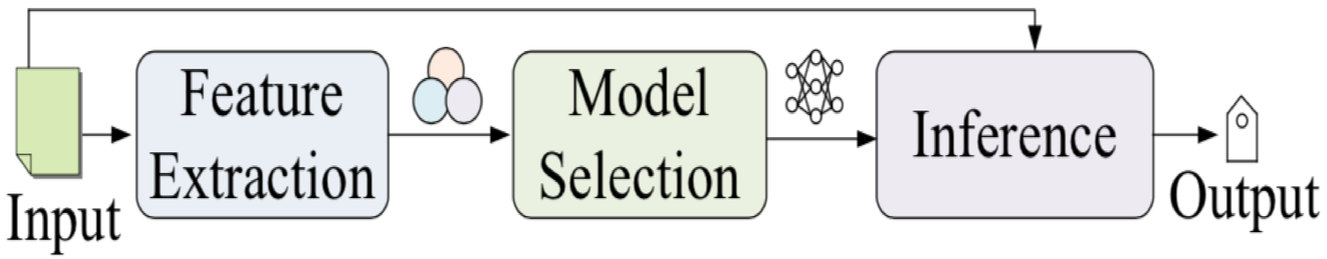
Nazarbayev University, Computer Science Department

Outline

- Introduction and Motivation
- Strategies and Methodologies
- Approaches and Reasons of Replication Failure
- Discussion and Scenario Plan
- Summary

Introduction

Design structure:



(Taylor et al., 2019 [1])

Introduction

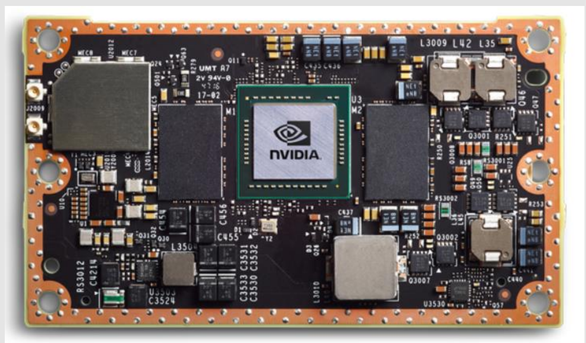
The whole algorithm of the original paper can be described in several steps:

1. Three pretrained **Convolutional Neural Networks** with different characteristics (low, medium, high accuracy and inference time).
2. **Feature extraction algorithm** that analyzes image's edges, hues and brightness, and produces result of each value in the range (0, 1).
3. **The premodel algorithm**, which chooses the suiting CNN based on the results of the feature extractor. The more complex the image, the more complex CNN should be chosen.
4. The chosen CNN performs the **image classification** and gives result.

Motivation

- The algorithm is a solution for optimal usage of CPU, memory and power without affecting any accuracy.

- While authors of the original paper established remote connection between the Desktop Computer and **NVIDIA Jetson TX2** module, our goal was to perform the inference using less complicated setup, which is only **one laptop**. Additionally, the performance of the code could be improved by using newer versions on CNNs and libraries. Two introduced devices:



Strategy & Methodology

- Goal for this work is **a reproduction of the experiment** of research paperwork on our personal laptop devices.
- Main parts of the experiment to be examined:

  1. DNN inference: examining used DNN models such as MobileNet, Inception and Resnet. Choosing the most efficient networks and using them in the premodel.

  2. Premodel algorithm: model selection using multiple KNN models for identifying the most optimal DNN model for the received input based on their inference time.

- **Replication process** of the two above parts constituted from two approaches: **top-down and bottom-up**.

Approach#1: Top-Down Approach

Initial attempt was to:

1. Obtain all required theoretical knowledge:

a) Study the original paper;

b) Analyze the provided source code;

c) Study used in the code libraries(TensorFlow, Scikit-learn, Pyro4, etc.).

2. Change the code to fit our setup:

a) Remove everything related to the remote connection;

b) Fix all errors related to libraries and their versions;

c) Fix all remaining logical and syntactical errors.

3. Run the resulting code and observe the results:

a) Observe the inference time;

b) Observe the accuracy of the premodel;

c) Compare results with the original paper.

Reasons of Replication Failure

Why the approach was not successful:

1. **Complexity of the progress**: rewriting project with more than 20 python files turned out to be complicated and time-consuming task.

2. Our inexperience: the source code contained many libraries that we have not worked with before, such as TensorFlow, Scikit-learn, Pyro4, Plotly, etc.

3. The absence of prior experience: the fact that our team has never been working with used libraries and utility software has affected our performance significantly.

Approach#2: Bottom-Up Approach

After the first attempt has failed, we decided to try the **second approach**:

1. **Creating a simple code:**
   a) Implement CNNs, that were used in original paper (MobileNet, ResNet, Inception);
   b) Train and test them using particularly chosen dataset.
2. **Adding main features of the original paper:**
   a) Feature extraction algorithm;
   b) Classifiers (K-Nearest Neighbor, Decision Tree, Support Vector Machine);
   c) Premodel algorithm.

Reasons of Replication Failure

Why the approach was not successful:

1. **Lack of time:** we spent too much time trying to make first approach work, which left only two weeks for this one.
2. **Insufficient knowledge:** building the algorithm from bottom required far more practical knowledge in machine learning, which we had not experienced.

Discussion

- The **work has not been finished** in a time of the internship.
- The usage of an ineffective initial plan, absence of the prior experience and knowledge about the subject, and lack of time has **led to this result**.
- **The future work** on this project is possible, and there is a room for improvement.
- We plan to continue working on this project in the future independently, so the updates on the results are probable.

Scenario Plan

Assuming we have additional two months or another internship, in order to **complete our reproduction work of the experiment** and **increase our success rate** we plan to continue with the **bottom-up** approach. Specific steps will include:

1. **Improve CNN inference** implementation to achieve high accuracy rate. This could be done by rewriting the code and choosing newer versions of used CNNs;
2. **Add several classifiers**, such as K-Nearest Neighbor and Decision Tree in order to choose the most efficient one;
3. **Implement the feature extraction** algorithm with the same features as in the original paper (brightness, hues and edge length);
4. **Develop the premodel** algorithm that chooses the CNN based on the results of feature extractor.
5. **(optional)** Use newer versions of the software, such as TensorFlow, MobileNet, ResNet and Inception.

Takeaways from the Project

- **Theoretical knowledge** in different spheres, techniques and algorithms of machine learning such as model selection, training model, feature selection, correlation-based feature selection, k-fold cross validation, DNN inference, K-nearest neighbors (KNN), and many machine learning libraries and APIs.
- **Practical experience** with many machine learning methods (k-fold cross validation, DNN inference, K-nearest neighbors) and libraries (Tensorflow, Scikit-learn).
- **Additional user experience with utility software**, such as Jupyter Notebook, Google Collaboratory, production of Comma-separated values (CSV) files, etc.

Summary

- **Reproduction work** was not fully completed: only DNN inference part of the experiment was implemented.
- All needed implementations for completion of the experiment is identified for possible **further work**.
- Skills gained throughout this summer research studies will possibly contribute to our successful completion of the experiment.

Reference List

[1] Marco, V., Taylor, B., Wang, Z., Elkhatib, Y., Optimizing Deep Learning Inference on Embedded Systems Through Adaptive Model Selection, 2019. ACM Transactions on Embedded Computing Systems.
https://doi.org/10.1145/3371154

[2] Jetson TX2 - embedded AI computing device.
https://developer.nvidia.com/embedded/jetson-tx2