University of Texas San Antonio, AI Practicum, Fall 2023

Assignment 4: Challenges with popular AI models

Due Date: Oct 10th, 2023, 11:59PM

Objectives

This assignment helps you visualize the challenges associated with the choice of AI models for solving specific tasks. The source code should be uploaded to Assignment 4 repository on github classroom (**link**) and a report should be submitted on blackboard with the required deliverables. Following are the objectives of Assignment 4.

- Pick a problem which can be solved by a DNN(DeepNeuralNetwork)/RNN (Recurrent Neural Network)/RL (Reinforcement learning) network. Explain what the problem is and identify the unique constraints pertaining to the problem.
- Implement the network using tensorflow and identify the challenges associated with developing the solution.
- Suggest and implement optimizations to improve the performance of the model on the provided task.

1. Identification of the Problem (25 points)

The goal is to identify and dissect a problem, and understand how AI models solve it. Provide a one page description of what is the problem that is being addressed, what are the constraints associated with it and how are the existing approaches working? Also, briefly describe each of the constraints that you identify.

List of potential problems can include, but not limited to - object detection, anomaly detection, motion recognition, image classification, video activity recognition, prediction (stocks, sales, weather,etc), recommendation systems. Attached is a resource where you can find projects for solving different problems.

2. Implementation of solution in Tensorflow (35 points)

Implement at least one of the AI solutions for the problem identified in Q1. You can either build the solution from scratch or utilize resources from the git. If you are using a code from the git, provide the specific citation and reference of the codebase.

For the AI solution, describe how each of the steps in the ML training pipeline have been addressed along with any potential pitfalls->

- Data collection ->
- Data preprocessing ->
- Design and Training model ->
- Evaluation and optimization of the model ->
- Deployment and model inference ->

Report the performance of the baseline model on the selected task. Plot the train and test accuracy vs epoch, as applicable.

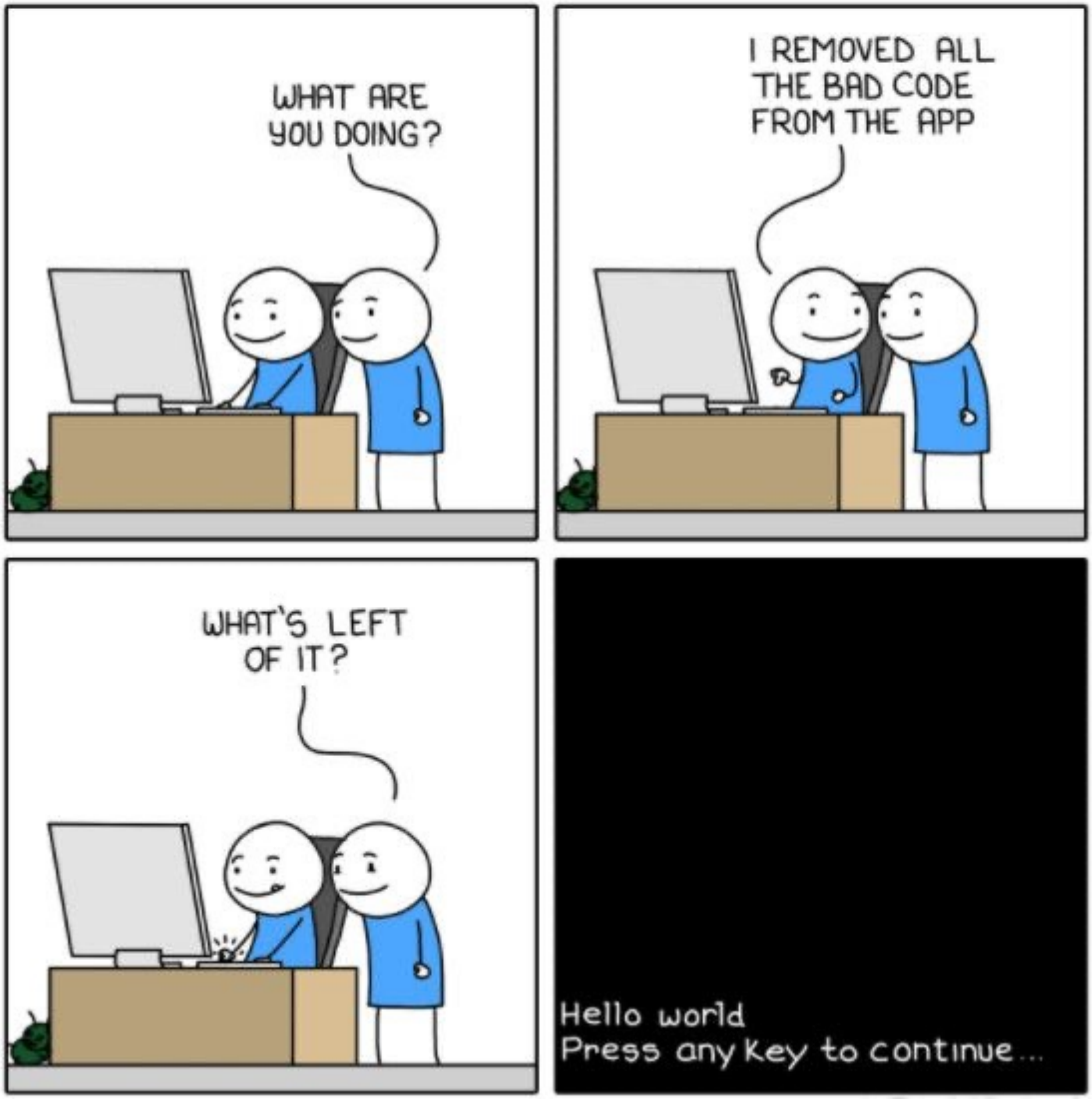
3. Optimization of the solution (40 points)

Optimize the implemented solution to improve either for accuracy or memory usage or training time or reliability. You can select to optimize at any stage of the pipeline and explain how that optimization helped in improving the performance. An example would be:

The baseline accuracy of the MobileNet network on keyword spotting was xx% for spotting the word "Hello". The optimization I selected was at the Design stage of the pipeline, wherein dropout was added to the output layer connections. This lead to xx+3% increase in accuracy. The reason for selecting dropout was to increase the separability of the features at the output layer by incorporating sparse representation using cross entropy loss. In addition to this, on further optimizing at the data preprocessing stage, a z% increase in performance was observed. The reason behind it being ... (description of the reason).

Upload the code on your git account and properly highlight the optimizations made using comments and elaboration in the README file. Plot the accuracy curves for the optimized model or any comparison metric for the baseline and the optimized model

(like training time for baseline vs training time for optimized model, etc. based on whichever is applicable for your scenario)


WHAT ARE YOU DOING?
I REMOVED ALL THE BAD CODE FROM THE APP
WHAT'S LEFT OF IT?
Hello world
Press any key to continue...