Project Report – Team 1

Introduction to Machine Learning (Spring 18) – Prof. Yao Wang

Table of Contents

Team and Project Details

Team Number: New Proj Team 1
Team Members: Anurag Marwah (Net ID: am8482)

Project Title: Recursive Image Classifier using VGG16
Github Link: https://github.com/anuragmarwah/VGGNet-on-FPGA

Problem Statement

To design an Image Classifier that recursively breaks down the image into sub-images and classifies the overall image and its sub-images using a pre-trained VGG16 Neural Net.

Problem Formulation

- I have used a pre-trained VGG 16 Neural Network.
- The idea is to to be able to classify such images that have more than 1 class of data, like multiple animals in a single image.
    - For this purpose, a small yet specific data set is made in which all the images have a clear presence of more than one class of objects
- The approach is to break down the image into parts (discussed in detail below) and then test each part for image classification using the pre-trained VGG 16 Neural Network.
- At the end, the output is a table of results per image, instead of just a single result

Neural Network and Training

- A pre-trained VGG 16 Neural Network from Keras is used for this project.
- The weights are downloaded from ImageNet

Approach towards the Problem

- Each test image is broken down into sub-images in 3 ways:
    1. Vertically, into two halves
    2. Horizontally, into two halves
    3. Vertically and horizontally both, into four quarters
- In total, we have 9 images (1 - overall, 2 – vertically split, 2 – horizontally split, 4 – cross-split)
- Each image is individually used to predict the class of the object

Results

- The model is tested over 10 different types of images
- For some images, the results are very good
    - Example: In the Penguin and Arctic Bear image, the model is able to classify the arctic bear in the broken down image with 99% accuracy.
- For some other images, like the one with one cat and two dogs, the model is not able to detect the cat in any of the images

Previous Project Topic and Its Issues

- The original project topic was to use a pre-trained VGG 16 Neural Net and implement it on FPGA
- This involved using a Digilent PYNQ-Z1 SoC board [1]
- But, I had a few issues in the implementation (discussed below)
- Thus, I could not finish it by the deadline and had to inevitably migrate towards another topic.

Initial Issues with the PYNQ board

- The PYNQ board can be powered either through the USB port, or through a external power adaptor. So, in the initial days, once I powered the PYNQ board from one of the USB ports at Roger's Hall 2nd Floor sitting area, and did not realize that the current rating was 3.1 Amps
    - As a result, it blew up the USB port of my PYNQ board

Resolution of Initial Issues

- I purchased the power adaptor and was able to successfully power up the board through it
    - Though, since I blew up my USB port, I could not use it for diagnostics anymore
    - If I would connect the PYNQ board to my laptop using the USB cable, it would keep on getting restarted again and again

Intermediate Issues with the PYNQ Board

- There were 2 major issues:
    1. The board was unable to connect to the internet
    2. The board was unable to use the *pynq* python packages already installed

Resolution of Intermediate Issues

- The cause of the first issue was the difference of Wi-Fi Networks at NYU and at home.
    - NYU uses a WPA-2 Enterprise network, which does not allow internet sharing between a laptop and a connected device, like an FPGA Kit, as it will become a vulnerability
    - To solve this, I connected the board directly to my Wi-Fi Router at home, thus getting internet
- The cause of the second issue was the difference between use of Python2 and Python3.6

- When the programs were run on the PYNQ board using just the “python” command, they would not work because by default it would use Python2
- After some figuring out the problem, when I eventually used Python3.6 explicitly, the existing packages were running just fine.

Further Issues

- There were two more issues:
  1. The PYNQ board would shut down inexplicably when I tried the following activities:
     a. Tried to create/ open a Jupyter Notebook
     b. Tried to install “pip” package (or any other package) via command line on the PYNQ board
  2. Could not install the Tensorflow and Keras package

Attempted Resolutions

- For the unexpected shutting down of the board, I could not find a solid reason
  - If I were downloading a package, it would start downloading till 98% but will then shut down.
  - If I were to open a Jupyter Notebook, it would immediately shut down
    - As a solution to this, I switched towards using .PY files instead of the Notebook files
- For the Tensorflow and Keras packages, I realized very late from a forum post that PYNQ board does not support Tensorflow because Pynq board is a 32-bit board, and Tensorflow requires 64-bit. [2]
  - It was at this point that I realized that it will not be possible to implement my original problem statement on this PYNQ board

References

1. Digilent PYNQ Board - https://reference.digilentinc.com/reference/programmable-logic/pynq-z1/start
2. Tensorflow not supported on Pynq - https://github.com/tensorflow/tensorflow/issues/14013