

BME646 and ECE60146: Homework 5

Spring 2024

Due: 11:59pm, 15 Feb, 2023

TA: Akshita Kamsali (akamsali@purdue.edu)

Turn in typed solutions via BrightSpace. Additional instructions can be found at BrightSpace. **Late submissions will be accepted with penalty: -10 points per-late-day, up to 5 days.** Please try to get an early start on this homework. The more you accomplish before next Tuesday, the more you will get out of that class.

1 Introduction

As you surely realized in HW4, the classification accuracy did not improve significantly when you increased the number of convolutional layers in HW4's class Net3.

An important reason for that is the problem of vanishing gradients in deep networks.

The goal of this homework is for you to acquire some preliminary experience with using a new network building-block element known as a SkipBlock or a ResBlock for mitigating the problem of vanishing gradients. Your doing so would make it a bit more efficient for your instructor to provide you with a thorough presentation of the topic next Tuesday. If you wish to also see some instructional material on SkipBlocks before the class on Tuesday, feel free to look over your instructor's Week 6 slides at the course website.

2 Getting Ready for This Homework

Go through the following steps in order to get ready for this homework:

1. Open the main module file `DLStudio/DLStudio.py` in your text editor and search for the string `Class SkipConnections`.
2. Read the doc string associated with the class definition for the class `SkipConnections`.
3. Go over the implementation of the inner class `SkipBlock`. **This will serve as the building-block for the network you will be creating for this homework.**

4. Now go back to the installation directory for DLStudio and open the main Examples subdirectory. There you will find the following script that you will find helpful for his homework. This script uses **BMEnet** that is defined right after **SkipBlock** in the main DLStudio module file:

`playing_with_skip_connections.py`

The code you will be creating for this homework will be along the lines of what you see in the class **BMEnet** in DLStudio. If you so wish, you can simply create a larger version of **BMEnet** for your homework solution.

3 Programming Tasks

3.1 Building Your Deep Convolutional Neural Network

1. Copy over the **SkipBlock** mentioned above into your code and use it as a building block for creating your network. What that means is that you will NOT be directly using the `torch.nn` components like `nn.Conv2d` in your neural network. Instead your network will consist of layers of **SkipBlock** in the same manner as the **BMEnet** is built from the **SkipBlock** components.
2. Make sure that your network has at least 40 learnable layers. Since each **SkipBlock** uses two instances of `nn.Conv2d`, your network will need to contain at least 20 instances of **SkipBlock**. You can check the number of layers in your network by

```
1 num_layers = len(list(net.parameters()))
```

Make sure you have properly commented your code and cited the sources of the code fragments you have borrowed. The report must mention the total number of learnable layers in your network. As previously mentioned, you may also directly use **BMEnet** as a starter for your classification network. **BMEnet Class Path:** DLStudio → SkipConnections → **BMEnet**

3.2 Training and Evaluating Your Trained Network

- Use the same training and evaluation pipelines that you created for HW4.

- You may also adapt the training and evaluation routines provided in DLStudio for the classification task.
- Collect training loss for at least two different learning rates and include the plots in your report.
- Report the accuracy numbers and the confusion matrix for different learning rates on the test dataset.
- State your observations regarding the classification performance of HW5Net in comparison with what you achieved previously with Net3 in HW4. Also, attach your confusion matrix of Net3 from HW4.
- Optional: You may also attach the confusion matrix generated for CIFAR dataset. State your observations in comparison to COCO dataset performance.

4 Submission Instructions

Include a typed report explaining how did you solve the given programming tasks.

1. Your pdf must include a description of
 - The figures and descriptions as mentioned in Sec. 3.
 - Your source code. Make sure that your source code files are adequately commented and cleaned up.
2. Turn in a pdf file a typed self-contained pdf report with source code and results. Rename your .pdf file as hw5_<First Name><Last Name>.pdf
3. Turn in a zipped file, it should include all source code files (only .py files are accepted). Rename your .zip file as hw5_<First Name><Last Name>.zip .
4. **Do NOT submit your network weights.**
5. **Do NOT submit your dataset.**
6. For all homeworks, you are encouraged to use .ipynb for development and the report. If you use .ipynb, please convert it to .py and submit that as source code.

7. You can resubmit a homework assignment as many times as you want up to the deadline. Each submission will overwrite any previous submission. **If you are submitting late, do it only once on BrightSpace.** Otherwise, we cannot guarantee that your latest submission will be pulled for grading and will not accept related regrade requests.
8. The sample solutions from previous years are for reference only. **Your code and final report must be your own work.**
9. To help better provide feedback to you, make sure to **number your figures and tables.**