**BME646 and ECE60146: Homework 6**

**Spring 2025**
**Due Date: Tuesday, Feb 25, 2025, 11:59pm**
**TA: Akshita Kamsali (akamsali@purdue.edu)**

Turn in typed solutions via Gradescope. Post questions to Piazza. Additional instructions can be found at the end. Late submissions will be accepted with penalty: -10 points per-late-day, up to 5 days.

# 1 Introduction

Based on the material in your instructor's Week 6 presentation, you now know the perils of creating deep networks by blindly stacking together convolutional and other layers without regard to the problems caused by vanishing gradients. The same lecture material has also taught you the importance of incorporating skip connections in deep networks.

In this homework, you will first gain a deeper understanding of your instructor's implementation of the `SkipBlock` building-block network by focusing on some of its key statements. Subsequently, you will experiment with a network of your choosing (along the lines of `BMEnet` in DLStudio) and a skip-block of your own design (along the lines of either the `SkipBlock`, which is an inner class of `BMEnet`) or along the lines of such blocks in ResNet. And you will report the classification performance improvements achieved with connection skipping.

# 2 Playing with Skip Connections

To get started do the following:

1. Open the main module file `DLStudio/DLStudio.py` in your text editor and search for the string `illustrating Skip Connections`.

2. Read the doc string associated with the class definition for the class `BMEnet`.

3. Go over the implementation of the inner class `SkipBlock`. This will serve as the building-block for the network you will be creating in the next steps.

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 uses the inner class `SkipBlock` as a building block.

   `playing_with_skip_connections.py`

   Make sure you have a thorough understanding of the relationship between the `BMEnet` and the `SkipBlock` class.

5. 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.

After you have familiarized with the `BMEnet` class and also understood the working of its inner class `SkipBlock`, **review the definition of the latter on Slides 16 and 17 of your instructor's Week 6 presentation.** In at least one of the results you show for this homework in which you compare with-and-without skip-connections results, you must also incorporate the following variations in your invocation of skip connections:

1. Focusing on Lines (I), (K), and (L) on Slide 16, you'll see that the convo operator is using a $1 \times 1$ kernel. Does it make any sense to be convolving with a 1 pixel kernel? Reflect on the difference that a $1 \times 1$ kernel is being used for in Line (I) vis-a-vis the use in Lines (J) and (K). **Write your observations in the report.**

2. The two operators being created in Lines (K) and (L) are for downsampling an image. But there are other ways of doing the same thing. For example, we could have used a `torch.nn.MaxPool2d(2,2)` layer for the downsampling needed. Or we could have used any ordinary convo layer with a stride of 2. Make a choice for an alternative approach and compare the results with and without your alternative.

## 2.1 Experimental Tasks and Analysis

1. **Training and Evaluation of Neural Networks:** Execute BMENet with Maxpool and the alternative of setting stride to 2 and report:

   (a) Training loss curve

(b) Confusion Matrix

2. **Overall Classification Accuracy** Report the overall classification accuracy for each model in a tabular format (Table 1), where each row corresponds to a network and its respective accuracy score. Try to show atleast 50% accuracy with modifying the parameters.

3. **Per class classification Accuracy** Construct a 10x2 table (Table 2) summarizing per-class classification accuracy. Each row represents a distinct class.

4. Examine the overall classification accuracy and provide a concise discussion highlighting key findings, such as trends in model performance, the impact of network depth, and any potential trade-offs observed in classification accuracy incomparison with HW5 networks and Maxpool vs Stride.

The `run_code_for_testing` function already computes the necessary values for tasks 2-4 (overall classification accuracy, per-class accuracy, and confusion matrix). You have the flexibility to either **post-process its output** to extract and format the required information or modify the testing code through inheritance to generate the desired outputs directly.

## 3 Skip Connections with MS-COCO

The completion of the above tasks provides you with the necessary skills to apply skip-connection-based CNN architectures to the MSCOCO dataset you curated. I recommend utilizing the following classes:

[ airplane, bus, cat, dog, pizza]

To prevent overfitting when using such a large model, ensure that each single instance class contains 1,500 training images and 500 test images. There are that many images in the dataset. This balanced dataset distribution will help the model generalize effectively while maintaining robust evaluation metrics. **Plot 5x3 images - 3 images from each class.**

1. Inherit the class `SkipBlock` or write a skip clock inspired from the class with appropriate citations to the source. Alternatively, you may directly copy over the `SkipBlock` from DLStudio 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

```
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`

Repeat section 2.1 deliverables for BMEnet outputs: Train loss curve, confusion matrix(5x5) and per class accuracy(5 rows x 1 column). Write your observations of this model's performance on COCO dataset. Yes, still try to show atleast 50% overall accuracy with modifying the parameters.

# 4 Submission Instructions

Include a typed report explaining how you solved the given programming tasks. You may refer to the homework solutions posted at the class website for the previous years for examples of how to structure your report

1. **Turn in a PDF file and mark all pages on gradescope**.
2. Submit your code files(s) as zip file.
3. **Code and Output Placement:** Include the output directly next to the corresponding code block in your submission. Avoid placing the code and output in separate sections as this can make it difficult to follow.
4. **Output Requirement:** Ensure that all your code produces outputs and that these outputs are included in the submitted PDF. Submissions without outputs may not receive full credit, even if the code appears correct.

5. For this homework, you are encouraged to use `.ipynb` for development and the report. If you use `.ipynb`, please convert code to `.py` and submit that as source code. **Do NOT submit .ipynb notebooks.**

6. 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.** Otherwise, we cannot guarantee that your latest submission will be pulled for grading and will not accept related regrade requests.

7. The sample solutions from previous years are for reference only. **Your code and final report must be your own work.**