Deep Learning Mini-Project

Fall 2022

Please upload your mini-project report before 5pm, November 22, 2022.

This mini-project will comprise 25% of your overall grade. Please perform this project in groups of (at most) 3. If you are looking for project team members, please broadcast your interest in the Class Campuswire in the `project-team-formation` chatroom.

Goal

In this mini-project you are tasked with coming up with a modified residual network (ResNet) architecture with the **highest test accuracy** on the CIFAR-10 image classification dataset, under the constraint that your model has **no more than 5 million parameters**.

Details

Recall that a residual network (ResNet) architecture is any convolutional network with *skipped* connections. Here is a picture of ResNet-18:

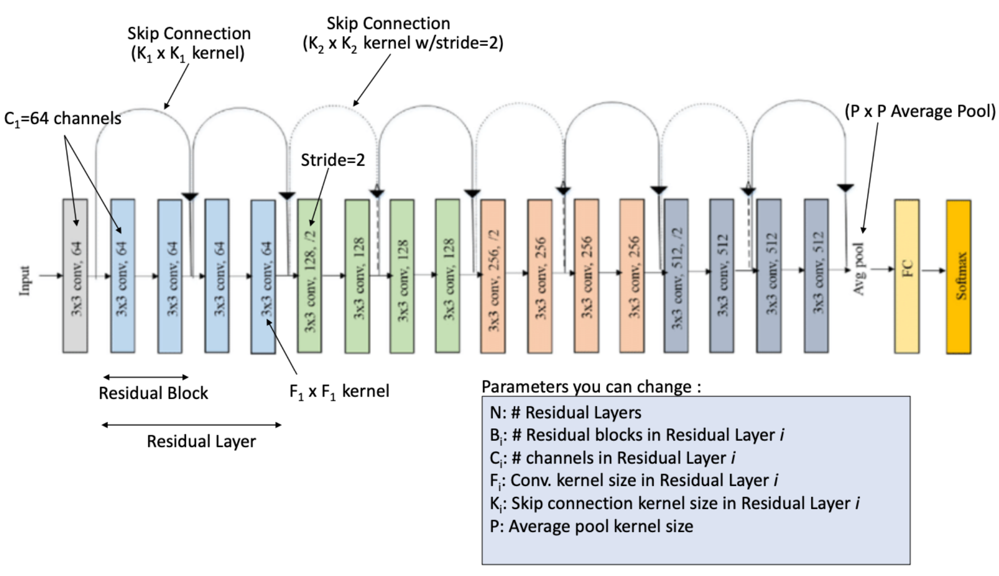

Figure 1: Block diagram of Resnet-18 model

The key component in ResNet models is a *residual block* that implements:

$$\text{ReLU}(S(x) + F(x))$$

where $S(x)$ refers to the skipped connection and $F(x)$ is a *block* that implements `conv -> BN -> relu -> conv -> BN`; here, "BN" stands for batch normalization. Chaining such blocks serially gives a deep ResNet.

Hyperparameters (design variables) in such architectures include:

- C_i, the number of channels in the ith layer.
- F_i, the filter size in the ith layer
- K_i, the kernel size in the ith skip connection
- P, the pool size in the average pool layer,

etc.

You are free to experiment around and adjust these parameters to gain boosts in accuracy, as long as the *total* number of trainable parameters does not exceed 5 million. (You can use the `torchsummary.summary` function to check the number of parameters in your model.)

You are also free to experiment with:

- any optimizer (SGD, ADAM, etc)
- any data augmentation strategy
- any regularizer
- any choice of learning rate, batch size, epochs, etc.

You are **not** allowed to:

- simply load pre-trained model weights from the web; you **have** to be able to train your model from scratch.
- use other/bigger datasets such as ImageNet.

Resources (optional)

This repository has excellent PyTorch code for training various ResNet models on CIFAR-10 from scratch. If you do use any parts of this repository, please include a clear citation. You are free to use any other online resources and/or techniques you like, as long as you include citations.

Unfortunately, NYU's HPC is not available for class use this semester. However, we have tested AWS's SageMaker Studio Lab, which allows free (automatic) signup to students and lets you reliably train ResNet models relevant to this project. Google Colab could be an option but be really wary of session timeouts, since you may need to train for upwards of 1 hour.

Deliverables

This project has two deliverables:

- A project report (15 points).
- A project codebase (10 points).

Projects will be graded on:

- clarity and quality of submitted project report;
- quality of final results. Aim for test accuracy of at least 80% as a minimum baseline; if your design is sensible, you should be able to achieve upwards of 90%;
- and clarity and quality of submitted codebase.

Project report

Your report has to be no more than 4 pages long including all figures, tables, and citations, **typeset in two-column AAAI 2023 camera-ready format**. Any report not in this format will not be graded. Here is a link to the format in LaTeX and MSWord formats; see the "Camera-ready" folder in this link for example documents.

Please upload a PDF of your report to Gradescope. **Only one team member has to upload** on Gradescope, as long as they tag all other team members.

In your report, please include:

- Names of all team members
- A short overview of your project, along with a summary of your findings
- A methodology section that explains how you went about designing and training your models, pros and cons of different architectural choices, what lessons you learned during the design process, etc.
- A results section that reports your final test accuracy, model architecture, and number of parameters.
- Any relevant citations.

Project codebase

In the first page of your report, please provide a link to a **publicly accessible** Github repository. Your repository should contain the code necessary to reproduce the results in your report. Please include well-documented code and/or Jupyter notebooks for easy visualization and verification of your work.