Project Instructions

Kaveh Fathian, Email: fathian@ariarobotics.com

Handout: 2024-11-22
Due: 2024-11-20, at 3:00 PM on Canvas

Project Logistics

- The project is for **bonus** points, and is completely **optional**
- You will work on your project as a **team** (2 to 3 people), not individually
- You will submit your report and code as a **team**. Reports/code that are **identical** across different **teams** will receive a grade of **zero**
- The submission of report/code will be on **Canvas**, and is due on **2024-12-04, 3:00pm**
- **Late submission** is **not accepted** under any circumstances

Project Deliverables

- You will submit a **single PDF file** as the report, a **single Python script** (**.py** format, **NOT** a Jupyter notebook) and **data files** (e.g., .npz or .keras formats) that contain your model weights.
- Your report should explain how you solved the project (e.g., how you designed or trained the neural net), answer all the questions mentioned below, and contain requested figures/plots
- Your NN should run **out of the box** in the **Conda environment we used in the course** – we will not debug the code for you, nor train your NN.

Grading

- In your report, you will grade your teammates based on their contribution and give each member grade in the **range of 0 to 1**
- The instructor will evaluate your project and assign a grade from 0-10. Project grade = report+code (6pts) + competition (4pts). Factors that can affect your grade include
  - If the requested tasks were accomplished
  - If the report is detailed, clear, and well-prepared
  - If the code is organized, commented, and runs out of the box (in our Conda environment)
- The evaluation is based on the discretion of the instructor and is non-negotiable!
- Each team member will receive a grade that is computed as the project grade assigned by the instructor times your teammates grade. For example, if the instructor gives the project a grade of 8 and your teammates give you a 0.5, your project grade will be 8x0.5=4pts.
- The project will add a maximum of 10pts to your final grade. For example, if your final grade (based on quizzes, homework, and attendance) is 89, and if you get 10 points for this project, your final grade will change to 89+10=99.
- See the course syllabus for the alphabetic grading rubric. For example, a final grade of 99 results in an A.

What characterizes a good project?

Using an existing model with pre-trained weights does not have any learning value. Similarly, (re)training an existing model by simply running the Adam optimizer in Google Colab is not interesting. However, if you make a tweak to a model, use different training strategies, or in general explore something new/different, you can expect to learn more (and earn more points).

Note that we will run all project reports/code in Turnitin for plagiarism check. If you use existing work/resources such as papers, code, blogs, or generative AI, please cite them in your report (and highlight what about your approach is different).

Project Instructions

Kaveh Fathian, Email: fathian@ariarobotics.com

Handout: 2024-11-22
Due: 2024-11-20, at 3:00 PM on Canvas

Project description

In this project, you will solve Problem 1 in Homework 5 by proposing **your own** neural network architecture. That is, you will design or adopt a network that can classify the images in the CIFAR-10 dataset. You are free to adopt any architecture, however, you **MUST train the network yourself**. That is, you cannot use any pre-trained models or weights. Instructions for training your NN using (free) Google Colab GPUs are provided below.

Your NN model must achieve a minimum accuracy score of 0.85 (or 85%) on the CIFAR-10 test set, otherwise your project will NOT be evaluated and will receive a grade of zero. For your reference, here is a benchmark for the latest results on the CIFAR-10 dataset: https://paperswithcode.com/sota/image-classification-on-cifar-10

Report

In your PDF report, describe **in detail** which model you used and how you trained it. More specifically, your report must contain:

- A picture of the network architecture (you can create your own diagram, or use many existing online tools such as https://alexlenail.me/NN-SVG/LeNet.html)
- Description of the loss function used for training
- The optimization algorithm used for training
- Plots of loss function vs. training epoch AND accuracy vs. epoch (accuracy is measured on the validation/test set)
- Final accuracy of trained model on the CIFAR-10 test set

Remember to include any references/resources that you used for your project, including papers, code, blogs, or generative AI.

Competition

We will run your model on our (undisclosed) test data to see how it performs. Our test images are NOT from the CIFAR-10 dataset; they are collected from other resources. We have 100 images in the competition dataset. Each image has one of the 10 classes of CIFAR-10, and has the same dimension in pixels. We will release the competition dataset after the due date so that you can evaluate your model yourself.

Your code must take in the competition images and labels, evaluate your model on them, and produce an accuracy score. To do this, use `project_template.py` that is provided to you in the project folder, and replace your code/model in the designated section. Note that your code must run out-of-the-box in the `cv` Conda environment we set up for the course.

Competition raking and scores

The groups will be ranked based on the accuracy score on the competition dataset. The teams will receive a score of 4 to 1 based on their quantile rank. For example, if 20 teams participate in the competition, the top 5 teams will review 4 points, teams 6-10 will receive 3 points, teams 11-15 will receive 2 points, and teams 16-20 will get 1 points.

Project Instructions

Kaveh Fathian, Email: fathian@ariarobotics.com

Handout: 2024-11-22
Due: 2024-11-20, at 3:00 PM on Canvas

Hints and Resources

Before committing to a model and start training, explore and see what are the state-of-the-art architectures that have resulted in good performance on the CIFAR-10 benchmark. Do not chose a very large model that is too complex to train. You are not constrained to use the CIFAR-10 dataset, and, to maximize your chances for the competition, you can use any other dataset for training your model. You can use data augmentation to improve the generalizability. For example, add some noise to the images and use these noisy images for training.

How to use remote GPU resources

You can use [Google Colaboratory]() (colloquially Colab) to remotely access graphics processing unit (GPU) hardware to speed up model training. However, if you know what you are doing, you are free to use any system to complete the project, including your own GPUs.

Colab is a free but limited resource: its file system is **temporary**. If you close the browser tab, your execution will **stop** and your files will **disappear**. If you sleep, turn off your computer, or remain idle (without code blocks executing) for **90 minutes**, your runtime session will stop and your files will disappear. Your execution will stop after **12 hours** and your files will disappear. To avoid setbacks, please be sure to download your results files (or push them to Github, depending on your workflow) before your session expires. Colab is not ideal but is a useful tool; alternatives like Google Cloud Platform are also not ideal and come with their own benefits and drawbacks.

Typical Colab workflow:

- Edit your stencil code locally
- Run your code locally on CPU to check for errors (does it train?)
- Upload your file to Google Drive and open it in Colab (just double click on it in Google Drive)
- Run your code on Colab GPUs for fast training
- Download results files from Colab (take your checkpoints, logs, etc.)

Recommended steps for training your model