Midterm Project 2: Neural Network Speed Up

Due Dates, Deliverables

Groups will be the same as in Midterm Project 1 (unless your instructor explicitly told you otherwise).

Due	Activity
check mycourses	Task+Method Category choice due (spreadsheet link in mycourses)
check mycourses	Code submission (ipynb), Presentation (pptx), Recording (spreadsheet) due

Overview

This is the second midterm project of the course. In this project, you will be required to use your creativity to speed up the training of a neural network. In your experiments, you need to work with one of the two pairs of neural network architectures from the following: Image Classification: {VGG (any) + RESNET (any)} OR Natural Language Modeling: { BERT-base + GPT-1 }. After half of the groups pick one choice, the remaining is required to pick the other, so we end up with roughly 50% of the groups for each pair.

Dataset choice

The choice of the pair of models and the associated task will determine the datasets that you will work with. You are required to work with both datasets. Each dataset will have 3 versions, more on that later in this writeup.

Image Datasets

- Caltech-256 (1.2 GB) - 256 classes
- Pascal Voc 12 (3.6GB) - 20 classes

Text Datasets

- Harvard Library (4GB)
- Yahoo Answers (4GB)

Method Category choice

The group will choose one of the following Method categories. In each section, up to 2 groups can choose the same approach. In this case groups (from all sections) need to make sure they do not implement the same approach.

- Optimization algorithm
- Weight initialization
- Data sampling
- Curriculum Learning
- Loss function manipulation
- Transfer learning / Knowledge distillation
- Neural architecture search
- Model implementation modification (e.g. library change)

Pre-project

Make sure you:

- Choose a method category by the due date
- Choose a task by the due date

Check mycourses for the due dates.

Implementation requirements

- Code should be in a single Jupyter Notebook file

For development purposes, any IDE or environment is allowed, however at time of submission the group need to make sure that the program is organized and submitted as a **single ipynb file**.

- Code should be runnable in the Google Colab environment

Before the submission, the group need to make sure that the code will be able to run in the Google Colab environment. Any instructions (e.g. the need to use a kernel with a GPU) need to be present as part of the ipynb file itself.

- Instructions and code to obtain and Pre-process the data should be available

Make sure the data used in this project is available publicly in some storage platform (e.g. Google Drive) without the need to logon. The ipynb should provide code to download and uncompress the data required for the training and inference of the model. If the license of the dataset does not allow you to publish the processed data, provide clear step-by-step instructions on how to obtain the data.

- You can only use functionality from ANY library or source.

Using any library such as pytorch, numpy, scipy is allowed. If you use code from others, you are required to cite the source.

- You should not make your project implementation public

This includes any code repositories. Not following this will be considered academic dishonest conduct, subject to university, department and instructor policy and sanctions.

- You should have at least one published work that describes your approach, or that is highly related to it, and you should list at least 3 published works that would fiti the same category you chose.

Make sure you research competing methods and provide references of at least 3 approaches that falls in the of the method category chosen.

Dataset Versions

You will work in your experiments with 3 versions of the dataset: 1MB, 10MB, 100MB. It is the group responsability to preprocess the data so the versions are created and used in the experiments.

You will need to provide code in your ipynb that perform all preprocessing. You are expected to maintain the original dataset's number of classes/categories and balancing.

Evalution and Visualizations

- Choose at least one commonly used metric for evaluation

The group should prefer metrics that are typically use in similar published papers that perform similar ML tasks. The group is encouraged to use more than one metric for evaluations. For instance, if you are performing classification, you might want to use accuracy or F1 score, but it is recommended that you also use precision and recall.

- Baseline evaluation

The main evaluation will be between the two architectures in the task that you chose (CV or NLP).

- Generate visualizations

Generate graphs showing the performance (using the chose metrics) of your approach on your baselines (the two neural network architectures)

- show the tuning (e.g. gridsearch) of the hyperparameters of your model (at least one hyperparameter for each neural network architecture)
- include convergence (hyperparameter x epochs to converge) plots for both neural network architectures
- include performance (metric after converged) plots for all datasets and splits (6 cases)

Presentation and Peer Feedback

- You will submit **a recording of your presentation** by the due date.
- Prepare and present the work completed in this project.
- You must use slides.
- It is fine to use additional resources such as showing visualizations, animations and demonstrations, but slides are mandatory.
- Every teammember needs to present
- Prefer to have yourself appear in the camera. You can do this using zoom. Turn your camera on and share your screen.
- The recorded presentation needs to be accessible without logins
- Use a streaming service (e.g. youtube, vimeo) that does not require login. You can have your video not be publicly listed as long as it is still acessible via the link. –>
- Paste the link of your recorded presentation on the project spreadsheet
- You presentation must be up to 12 minutes long.
- You must use the peer feedback form that will be available in mycourses to evaluate every other team. Each student needs to make one evaluation submission for each presentation watched. E.g. if there are 9 groups, you will submit the evaluation form 8 times.
- You will not evaluate your own group.
- The group that performs the best (based on peer feedback) will get an automatic 100 for the project presentation grade.
- The presentation needs to cover the following topics
    1. Dataset and task, including preprocessing choices
    2. Detailed explanation of the proposed approach
    3. Detailed explanation of the architectures of the chosen task
    4. Evaluation, comparisons with the baselines

Grading rubric

Code (50% of project grade)

- Code is in the right ipynb format in a single file
- Code runs on Google Colab without modifications
- Data was able to be downloaded just by running the ipynb cells or by following instructions
- Model chosen from list
- Dataset versions (1M, 10M, 100M, 1GB) were created and correctly used
- Evaluation metric was implemented
- Baseline (where the proposed approach is not used) present in the visualizations generated (e.g. Vanilla SGD with no momentum vs Nesterov)
- Visualization generated with at least two hyperparameter/variations showing impact in model convergence over epochs
- Visualization generated with at least two hyperparameter/variations showing impact in final model performance (impact on performance of the converged model)
- Different groups did not implement the same approach (50% grade penalty if this is not observed.)

Presentation (50% of project grade)

- Recording Link is posted in spreadsheet
- Recording is accessible without login
- Presentation length is/was less than 12 minutes
- All group members presented
- Peer feedback completed
- Peer feedback scores