CS587: Assignment 1

Mini-batch SGD

Issued: Tuesday 05/03/2024
Deadline: Tuesday 19/03/2024, 23:59

Description

In this exercise, you will need to implement a simple version of a mini-batch SGD algorithm in order to train a linear classifier for the CIFAR10 image classification problem. Your training set will consist of N pairs (x_i, y_i), where x_i represents a vectorized version of the i-th training image and $y_i \in \{0, 1, 2, \ldots, 9\}$ denotes the index of its ground truth class (there are 10 classes in this case). Your classifier will have the following form:

$$f(x; W, b) = Wx + b \ ,$$

where W is a $10 \times D$ matrix, b is a 10×1 bias vector and x is a vectorized image of size $D \times 1$.
To train the classifier, you will minimize the following regularized empirical loss function:

$$\mathcal{L}(W, b) = \lambda R(W) + \sum_{i=1}^{N} loss(f(x_i; W, b), y_i) \ , \tag{1}$$

where

$$loss(s, y) = \max(0, 1 + \max_{j \neq y} s_j - s_y) \ ,$$

the regularizer $R(W)$ can be either an l_2 or an l_1 regularization norm, i.e.,

$$R(W) = \sum_{i,j} W_{i,j}^2 \qquad (\text{l}_2 \text{ regularizer})$$

$$R(W) = \sum_{i,j} |W_{i,j}| \qquad (\text{l}_1 \text{ regularizer})$$

and λ is a scalar hyperparameter representing the regularization strength.

For the mini-batch SGD algorithm you will use a constant learning rate γ and a mini-batch size M, in which case the update step (for a mini-batch $\{(x_i, y_i)\}_{i=1}^{M}$) will have the following form:

$$W \leftarrow W - \gamma \left(\lambda \nabla_W R(W) + \frac{\sum_{i=1}^{M} \nabla_W loss(f(x_i; W, b), y_i)}{M} \right) \tag{2}$$

$$b \leftarrow b - \gamma \left(\frac{\sum_{i=1}^{M} \nabla_b loss(f(x_i; W, b), y_i)}{M} \right) \tag{3}$$

1. As a first step, you need to correctly fill-in the `compute_gradient_and_loss` function that takes as input a set of training samples and computes the loss (1) and the gradient of the loss (for the given training samples). You will call this function inside the `train_linear_classifer` routine in order to compute the gradient of each mini-batch, and for collecting the sequence of all mini-batch losses during training as well as the sequence of all validation losses during training.

2. To implement your linear classifier, you will need to fill-in the following two functions:

   `train_linear_classifer`: this is the routine responsible for training the classifier using mini-batch SGD. It should return the parameters of the trained classifier and the sequence of all mini-batch losses during training as well as the sequence of all validation losses during training.[1]

[1] The mini batch loss will include the losses $loss(\cdot)$ for the M samples in the mini-batch as well as the regularization loss $\lambda R(\cdot)$. The validation loss will include the losses $loss(\cdot)$ for all the samples in the validation dataset but will not include the regularization loss $\lambda R(\cdot)$.

`predict_image_class`: this routine takes as input an image and uses a trained classifier to predict its class (recall that the predicted class should be the one that is assigned the maximum score by the trained classifer).

3. As a last step, you will use the validation set in order to choose proper values for some of the hyperparameters of the problem (these include the regularization strength λ, the mini-batch size M and the type of regularization l_1 or l_2). To that end, you will train linear classifiers for a different number of combinations of these hyperparameters (see file `main_script`) and you will choose as your final classifier the one that achieves the highest accuracy in the validation set. Include your findings and explain your choices in your report.

4. For the final classifier, you should draw (in the same plot) the sequence of mini-batch losses and validation losses collected during training, as well as visualize (as images) the weights W (one image per row of W). Furthermore, you should evaluate the classifier on the test set and report the achieved test accuracy.

Comment on your results in a short report. Insufficient comments can lead to a significant reduction in the final grade of the assignment, even if the implementation is correct.

Notes

Setup your Python environment

Python 3.10 is required to run the provided code and accomplish your assignment. Read the file `setup-README.txt` in the main folder of the assignment and the files of the second tutorial to get information on how to successfully setup your Python environment.

Download the CIFAR-10 dataset

Before you start working on your assignment, you need to download the CIFAR-10 dataset:

- go to `http://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz`
- save the *tar.gz* file locally
- extract it in your /cs587_asssignment1/datasets/ folder.
- Check that the 8 files of the dataset folder are placed under /cs587_asssignment1/datasets/cifar-10-batches-py/.

Submission info

- Create a .pdf or .doc file to report the resulting scores, images/figures and any other comments or description of your work you may need to submit. Do not forget to include your name, login, ID in the report. Save this file in your working folder.
- After you have finalized your coding and report, remove the datasets folder from the working directory to be submitted.
- Use zip/rar/gz to compress your working folder and rename it to cs587_mylogin_assignment1.xxx in order to submit a single file.
- Upload your submission as a SINGLE zip/rar/gz file on e-learn.
- You can upload your submission as many times as you need, keeping the same filename.
- Late submissions will be accepted **within two days** of the original deadline. However, **a penalty of 25% per day** will be applied to the final grade for each late day. Assignments submitted more than two days late will not be accepted.

Subscribe to the mailing list

Subscribe to the mailing list (hy587-list@csd.uoc.gr) using the instructions provided here to receive notifications and resources for the course.

Academic integrity

The assignment is individual. You may discuss with each other in general terms, but the code and the report should be written individually. If you use existing material, be sure to cite the sources in your report. The use of AI language models to produce the report or your implementation is strictly prohibited, and submissions will be verified with an AI detection tool. You may be asked to take an additional short oral examination.

Troubleshooting & Contact

If you encounter any errors or bugs in the provided code, you can report them by sending an email to either kaziales@csd.uoc.gr or the course mailing list hy578-list@csd.uoc.gr. Please note that the course mailing list is the preferred channel for general questions about the assignment, as it allows everyone in the class to benefit from the answer.

For any questions of a more personal nature that are not relevant to the entire class, you can reach out to the teaching assistant directly at his email address kaziales@csd.uoc.gr.