HY587: Assignment 1

Mini-batch SGD

Issued: 31/03/2020
Deadline: 14/04/2020, 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 will need to correctly fill-in the function `compute_gradient_and_loss` 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]

   `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.

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.

Notes

Setup your Python environment

Python 2.7 is required to run the provided code and accomplish your assignment. Read the setup-README.txt file in the main folder of the assignment and the slides of the first tutorial to get info on how to setup your Python environment successfully. Link to slides of the 1st tutorial
`https://drive.google.com/open?id=1RanpIS8z72JlguTZAe1HsIfG7U4co61XrY_2nWoynOk`

[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 $\lambda R(\cdot)$.

Download the CIFAR-10 dataset

Before 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+report, remove the datasets folder from the working directory to be submitted.
- Use zip/rar/gz to compress your working folderand rename it to cs587_mylogin_assignment1.xxx in order to submit a single file.
- You will use the following link of Dropbox request to upload your submission as a SINGLE zip/rar/gz file. You will be requested fiil in your firstname/surname and an email address to upload. `https://www.dropbox.com/request/pHftrWe39HEccqxACKtK`
- You do not need to have a Dropbox account to upload your submission.
- You can upload your submission as many times as you need keeping the same filename.
- Uploading will not be available after the deadline date-time.

Subscribe to the mailing list

Subscribe to the mailing list (hy587-list@csd.uoc.gr) using the instructions provided in `http://www.csd.uoc.gr/CSD/index.jsp?custom=subscription&lang=en` to receive notifications and resources for the course.

Troubleshooting

In case you find any errors/bugs in the code please send an email to konarak@csd.uoc.gr or the mailing list.