CS587: Assignment 1

Linear Regression

Issued: 13/03/2023
Deadline: 27/03/2023, 23:59

Description

The goal of this exercise is to serve as an introduction to the Stochastic Gradient Descent (SGD) algorithm, and, also to get you familiarized with the basics of training a simple parametric model.

More specifically, in this exercise you are called 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:

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

where Loss() represents the following version of a Hinge loss function

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

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.

The key elements (parameters) that define the linear classifier are the (a) *weight matrix,* W, and (b) *bias vector,* b. To estimate these parameters you will implement the mini-batch SGD algorithm.

In mini-batch SGD, the procedure starts with a random value choice for the model's parameters and performs the following steps at each iteration:

- Randomly samples a so-called mini-batch of size M, which consists of a set of M (input data, true label) points, $\{(x_i, y_i)\}_{i=1}^{M}$, that are randomly chosen from the original set of N training pairs, where $M \ll N$,
- Forms the following simplified objective of eq. (1) based on the sampled mini-batch:

$$L_{\mathrm{SGD}}(W, b) = \frac{1}{M} \sum_{i=1}^{M} \mathrm{Loss}(f(x_i; W, b), y_i) + \lambda R(W)$$

  and computes its partial derivatives with respect to each of the model parameters.
- Updates each model parameter, say θ, as $\theta_{new} = \theta_{old} - \gamma * \frac{\partial L_{\mathrm{SGD}}}{\partial \theta}$, where $\frac{\partial L_{\mathrm{SGD}}}{\partial \theta}$ denotes the partial derivative of L_{SGD} with respect to the scalar parameter θ.

In the above process, γ indicates the *learning rate*, which is a hyper-parameter that controls the change effect on the model.

Under this premise, using a mini-batch size M, the update step for the parameters of the linear classifier that we are building will have the following form:

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

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

where ∇ denotes the gradient operator (i.e., the vector of all partial derivatives).

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_classifier` 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_classifier`: this is the routine responsible for training the classifier using mini-batch GD. 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 classifier).

3. As a last step, you will use the validation set in order to choose proper values for some of the hyper-parameters 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 hyper-parameters (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[2].

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_assignment1/datasets/cifar-10-batches-py/.

[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 $\lambda R(\cdot)$.

[2] Click me

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 code+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.
- You will upload your submission in course's workplace at the e-learn platform as a SINGLE zip/rar/gz file. You will be requested to verify the submission! DO NOT FORGET TO PRESS SUBMIT, otherwise the assignment will not be uploaded.
- 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 kbacharidis@csd.uoc.gr or the mailing list.