Principles & Applications of Data Science Homework 4

Instructor: Joonseok Lee

Deadline: 2023/12/06 Wed, 11:00am

- Submit a single zipped file containing all files on ETL.
- No unapproved extension of deadline is allowed. Late submission will result in 0 credit.
- Optimize your code as much as you can. We do not guarantee to run unreasonably inefficient codes for grading. Remember, vectorization is important for efficient computation!
- Explicitly mention your collaborators or reference (*e.g.*, website) if any.

1 Hand-written Digits Classification using k-nearest Neighbors [50pts]

In this exercise, we are going to implement a k-nearest neighbor classifier directly working with pixel values. Download the MNIST dataset and skeleton code `PADS_hw4_knn.ipynb` provided on eTL.

(You may suffer from lack of memory space if you use the entire MNIST dataset for training and inference. To ease this problem, the provided code randomly selects 1000 training and 200 test examples by default. Please feel free to adjust these numbers depending on your computing environment.)

(a) Implement the `train(self, X, y)` method. [5 pts]
(Recall that for k-nearest neighbors, this is just memorizing the training data.)

(b) Complete the `compute_distance(self, X_test, dist_metric)` method. Specifically, compute the distance between each test example in `X_test` and each training example in `X_train` using k-nearest neighbors based on dot product, cosine similarity, L1, L2 (Euclidean). Details on each distance are given in the skeleton code. [15 pts]

(c) Complete the `predict_labels(self, X_test, dists, k)` method. It should return the class prediction for the test image using a majority vote from k closest points. [10 pts]

(d) Experiment with multiple values of k for various types of distances in computing k-nearest neighbors and plot the graph showing the results. [10 pts]

(e) Interpret the result from the plotted graph in (d). What distance metric would you use for MNIST classification, and why? If you select the best model based on the results and observe a slight performance drop on unseen datasets, what could be the potential reasons contributing to this outcome? Write your discussions in markdown in `PADS_hw4_knn.ipynb`. [10 pts]

2 Two-layer Neural Networks [50 pts]

In this question, we are going to implement a simple two-layer fully-connected neural network to classify the MNIST dataset. Specifically, the network we will build is given by

$$\hat{Y} = \text{softmax}(\texttt{Leaky\_ReLU}(W_1\mathbf{X} + b_1) \cdot W_2 + b_2).$$

where $\mathbf{X} \in \mathbb{R}^{N \times M}$ is the input matrix containing N images with M pixels, $\mathbf{Y} \in \mathbb{R}^{N \times C}$ is one-hot encoded ground truth with C classes.

For this assignment, do NOT use PyTorch, TensorFlow, or any other neural network packages. This will be the only opportunity for you to learn what happens behind the scene of the forward pass and backpropagation. Similarly to the previous question, work on the provided skeleton code `PADS_hw4_2nn.ipynb`.

(a) Implement `leaky_relu(z)` and `softmax(z)`.[5 pts]
(do NOT use python library function which directly calculate the results, such as `numpy.softmax, scipy.special.softmax`)

(b) Implement the `initialize_parameters(self, input_dim, num_hiddens, num_classes)`. [5 pts]

(c) Implement the `forward(self, X)`. [10 pts]

(d) Implement the `backward(self, X_batch, Y_batch, batch_size, lr)`, where `ff_dict` is the output from the forward step. [10 pts]

(e) Implement the `train_step(self, Y, Y_hat)` method, which takes a batch of train set and updates the parameters. [5 pts]

(f) Implement the `evaluate(self, Y, Y_hat)` method, which takes a test set and returns the classification accuracy. [5 pts]

(g) Set aside some training examples as validation set, and tune hyperparameters (e.g., learning rate, hidden dimensions, number of epochs, batch size) to optimize the validation accuracy. Report the best combination of hyperparameters you found along with your final test accuracy. [10 pts]

What to Submit

Please upload a single zip file named with your name and student ID (e.g., `홍길동_2023-20000.zip`) on eTL, containing

- Your complete `PADS_hw4_knn.ipynb` and `PADS_hw4_2nn.ipynb` files.