Homework III

Due date: 10/6/2024 at 11:59pm

Submission Guidelines: 1. Put all the documents into one folder, name that folder as firstnamelastname_UIN and compress it into a .zip file.

2. For coding problems, your submission should include a code file (either .py or .ipynb), and also a pdf file (in one file together with other non-coding questions) to report the results that are required in the question.

Problem 1: Regularized Logistic Regression (35 points)

Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be the training examples, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$. The negative log-likelihood of the regularized logistic regression, denoted by $L(\mathbf{w})$ is written as

$$L(\mathbf{w}) = C \sum_{i=1}^{n} \ln(1 + \exp(-y_i \mathbf{x}_i^\top \mathbf{w})) + \frac{1}{2}\|\mathbf{w}\|_2^2 \tag{1}$$

where C is a parameter that controls the balance between the loss and the regularization. The optimal solution for $\mathbf{w} \in \mathbb{R}^d$ is obtained by minimizing $L(\mathbf{w})$.

- Prove the objective is convex. (Hint: first prove the convexity of the function $h(s) = \log(1 + \exp(s))$ and then use the rules that preserve convexity.)
- Show $w_k = w_l$ for the optimal solution $\mathbf{w}$ if two attributes k and l are identical, i.e., $x_{i,k} = x_{i,l}$ for any training example $\mathbf{x}_i$.

Problem 2: Lagrange Dual (25 points)

In this problem, we are going to derive the Lagrange dual of a distributionally robust optimization. Sometimes we add a temperature parameter τ in the softmax function to increase the flexibility of modeling (e.g., for knowledge distillation), e.g.

$$\Pr(y|\mathbf{x}) = \frac{\exp(\mathbf{w}_y^\top \mathbf{x}/\tau)}{\sum_{l=1}^{K} \exp(\mathbf{w}_l^\top \mathbf{x}/\tau)}$$

Then the cross-entropy loss becomes $\ell(\mathbf{w}, \mathbf{x}, y) = -\log P(y|\mathbf{x}) = \log \sum_{k=1}^{K} \exp(\mathbf{w}_k^\top \mathbf{x}/\tau - \mathbf{w}_y^\top \mathbf{x}/\tau)$. A question arises is how can we learn individual τ for each data. To address this question, let us

fix x and y, and define $s_k = \mathbf{w}_k^\top \mathbf{x} - \mathbf{w}_y^\top \mathbf{x}$. Let us consider the following problem:

$$L(s_1, \ldots, s_K) = \max_{\mathbf{p} \in \mathbb{R}^K} \sum_{k=1}^{K} p_k s_k$$
$$s.t., \sum_{k=1}^{K} p_k = 1, \sum_{k=1}^{K} p_k \log(p_k K) \leq \rho.$$

where $\mathbf{p} = (p_1, \ldots, p_K)^\top, \rho > 0$. Regarding the above problem, answer the following question:

- (5') Prove the above problem is a convex optimization problem.
- (5') Show the above problem is strictly feasible so the slaters' constraint qualification will be satisfied.
- (10') Derive the Lagrange Dual problem of the above problem, written in a form of minimization. Discuss how we can leverage the dual problem to learn individual temperature in cross-entropy loss.
- (5') Derive the form of $\mathbf{p}$ given the optimal Lagrangian dual variables.

Problem 3: Multi-class Logistic Regression(40 points)

In this problem we consider a 10-class digit recognition task with a linear model. Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be the training examples, where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the input image (flattened) and $y_i \in \{0, 1, \cdots, 9\}$. The negative log-likelihood of the Multi-class Logistic Regression(MLR) is written as

$$L(\mathbf{w}) = -\frac{1}{n} \sum_{i=1}^{n} \ln(Pr(y_i|\mathbf{x}_i))$$

where $Pr(y_i|\mathbf{x}_i)$ is computed by so-called softmax function given the score of each class for the sample:

$$Pr(y_i = k|\mathbf{x}_i) = \frac{e^{\mathbf{w}_k^T \mathbf{x}_i}}{\sum_{j=0}^{9} e^{\mathbf{w}_j^T \mathbf{x}_i}}$$

where $\mathbf{w}_k^T \mathbf{x}_i$ is the score of $\mathbf{x}_i$ for class k given by linear model.
In this problem you are asked to set up the whole pipeline of training a linear model to classify hand-written digits from the MNIST dataset.

(1) Data preprocessing(10 points)

(a) Split the dataset(70k samples) into train set(50k), validation set(10k) and test set(10k).

(b) normalize all the raw image as training samples, i.e. $\mathbf{x}_{i,processed}^{s,t} = \frac{\mathbf{x}_i^{s,t}}{256}$

(2) Minibatch training(15 points)

Minibatch is a very popular technique for stochastic optimization algorithm in deep learning, take SGD as an example. Assume that the loss function is of the following form:

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{w}; \mathbf{x}_i)$$

then minibatch SGD do the following update for model parameters:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t G_t$$

where $G_t = \frac{1}{B}\sum_{i\in B_t} \nabla f(\mathbf{w}_t, \mathbf{x}_i)$ is an unbiased stochastic gradient estimator, i.e. $\mathbb{E}_{B_t}[G_t] = \nabla f(\mathbf{w}_t)$, B_t is a randomly sampled minibatch and $B = |B_t|$ is the number of samples in a minibatch. Note that when $B = 1$ minibatch SGD is equivalent to standard SGD; when $B = n$, minibatch SGD is equivalent to (Full) Gradient Descent.

(a) For each batch size in [1, 10, 100, 1000], train the MLR model with minibatch SGD for 20 epochs on training data and compute the cross entropy(CE) loss on the training data of the obtained model every $\frac{5000}{batchsize}$ iterations, i.e. 10 times per epoch.
Note: the CE loss is actually the same as negative log-likelihood of MLR we define before.

(b) Plot the CE loss vs iteration curves and the CE loss vs epoch curves on the training data. [1]

(c) Plot the curve of (i)the number of epochs needed to reach the lowest loss on training data vs batch size and (ii)the lowest training loss achieved vs batch size.

(d) Discuss your observations of the curves in (b)(c) and analyse why this happens. Selected the best batchsize and explain your criteria.

(3) Weight Decay(15 points)

Weight decay is another commonly used technique in deep learning to avoid overfitting. Taking SGD as an example again. Compare to standard SGD update, SGD with weight decay[2] add a term proportional to the parameter itself to the stochastic gradient:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t(G_t + \lambda\mathbf{w})$$

Note: in this part you should use the batchsize selected in (2).

(a) For each value of λ in [0, 0.01, 0.1, 1] train the MLR model on training data and compute the CE loss on both the train set and the validation set.

(b) Plot the curves for CE loss on both the train set and the validation set vs different values of λ. Discuss your observations of the error curves, and report the best value of λ.

(c) Plot the curve of $||\mathbf{w}||^2$ vs different values of λ. Discuss your observations.

(d) Train the model on the whole training data(train set + validation set) using the selected λ and compute the CE loss on the test set. Plot the testing loss curve.

[1] This two curves are essentially the same, but it provides you different views.

[2] Note that this is equivalent to adding a L2-regularization term in loss function because $\nabla(f(\mathbf{w}) + \frac{\lambda}{2}|\mathbf{w}|^2) = \nabla f(\mathbf{w}) + \lambda\mathbf{w}$