CSE 547 - Assignment 2

Philip Pham

April 28, 2018

Problem 0

List of collaborators: I have not collaborated with anyone.

List of acknowledgements: None.

Certify that you have read the instructions: I have read and understood these policies.

Problem 1: Generalization, Streaming, and SGD

In class, we examined using Stochastic Gradient Descent (SGD) for empirical loss minimization, where we have an N sized training set $\mathcal{T}$. The empirical loss considered was:

$$F(w) = \frac{1}{N} \sum_{(x,y) \in \mathcal{T}} l\left(w, (x, y)\right). \tag{1}$$

Here, gradient descent for the function F is the algorithm:

1. Initialize at some point $w^{(0)}$.
2. Sample (x, y) uniformly at random from the set $\mathcal{T}$.
3. Update the parameters:
$$w^{(k+1)} = w^{(k)} - \eta_k \cdot \nabla l\left(w^{(k)}, (x, y)\right), \tag{2}$$
and go back to 2.

We provided guarantees assuming that F was smooth and the gradients in our training set were uniformly bounded, $\|\nabla l\left(w, (x, y)\right)\| \leq B$.

However, in practice, we care about generalization, that is, statements on how well we do on the underlying distribution. Define:

$$\mathcal{L}(w) = \mathbb{E}_{(x,y) \in \mathcal{D}} l\left(w, (x, y)\right), \tag{3}$$

where $\mathcal{D}$ is the underlying distribution.

Suppose we sought a point where $\|\nabla \mathcal{L}\|^2$ was small. Obtaining this quantity to be small even in expectation would be acceptable for this problem Assume that $\mathcal{L}$ is smooth and that the gradients are uniformly bounded, $\|\nabla l\left(w, (x, y)\right)\| \leq B$ for all parameters and all possible points (x, y) (under $\mathcal{D}$).

1. Assume we have sampling access to our underlying distribution $\mathcal{D}$. Explain how we can make $\|\mathcal{L}(w)\|^2$ small in expection. What can you guarantee if you obtain m samples and how would you do this?

Solution

ds

2. Suppose we contruct an N sized training set $\mathcal{T}$, where each point is sampled under $\mathcal{D}$; then we construct the empirical loss funciton $F(w)$; then we run SGD on F for K steps (suppose $K \geq N$). Is there an argument on this procedure that implies something non-trivial (and technically correct) about $\|\nabla \mathcal{L}(w)\|^2$, even in expectation?

Solution

Problem 4

We will now consider the multi-label classification problem. In the multi-label problem, there are multiple labels that could be "on" for each input x. You will use either the square loss or the binary logistic loss and consider training two models, namely (i) a linear model and (ii) a multi-layer perceptron (MLP) with a number of hidden nodes that you will tune.

You will try out three methods in each of the following: (1) SGD with a mini-batch size that you tune. You will use the same minibatch size for the other algorithms; (2) try out PolyakâĂŹs "heavy ball method" (aka momentum) or NesterovâĂŹs accelerated gradient descent (NAG); and (3) either Adagrad or Adam. You must tune all the parameters of these methods.

The dataset contains 18 total categories with a number of categories for each supercategory (vehicle or animal). In the dataset provided, each image contains objects of a single supercategory, say vehicle, and potentially multiple objects from the supercategory, such as car, boat, etc. In this exercise we shall build a classifier that learns to identify *all the categories of objects* present in each image, by optimizing either a square loss or a logistic loss objective. For the purposes of learning these classifiers, we shall use the dataset and features from the first homework. We shall also provide a larger version of this dataset since we need to train more parameters for this model.

The object function we choos to optimize is

$$L(w) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} l\left(y_{ij}, f_{ij}(w)\right), \tag{4}$$

where $f_{ij}(w) = w_j^\intercal x_i$ and $w_j \in \mathbb{R}^d$ is the jth column of $w \in \mathbb{R}^d \times \mathbb{R}^k$. Here, w is the linear model we wish to optimize over and $\lambda > 0$ is the strength of l_2 regularization. here l is the loss function:

- $l\left(y, \hat{y}\right) = \frac{1}{2}\left(y - \hat{y}\right)^2$ is the square error loss.
- $l\left(y, \hat{y}\right) = y \log\left(1 + \exp\left(-\hat{y}\right)\right) + (1 - y)\log\left(1 + \exp\left(\hat{y}\right)\right)$ is the logistic loss where the true label $y \in \{0, 1\}$.

Notice that we encode y_i as binary vector of length $k = 18$ (the number of categories) where a 1 indicates the presence of a category and 0 indicates the absence.

Determine which loss function works better for a linear classifier and use that loss throughout the question.

When using MLP, $f_{ij}(w) = \left\langle w_j^{(2)}, \text{relu}\left(w^{(1)} x_i\right)\right\rangle$, where $w^{(1)} \in \mathbb{R}^h \times \mathbb{R}^d$ are the weights in the first layer and h is the number of hidden nodes. Again $w_j^{(2)} \in \mathbb{R}^h$ is the jth column of $w^{(2)} \in \mathbb{R}^h \times \mathbb{R}^k$, the weights of the second layer.

SGD and Linear Regression

Now consider running stochastic gradient descent on $L(w)$.

1. What mini-batch size do you use? What stepsize did you use? What value of λ did you use? Specify your stepsize scheme if you chose to decay your stepsize. Which loss function did you find works better?

   Solution

Heavy Ball or Nesterov's method

Adagrad or Adam