Optimization for Machine Learning Review HW

name

All parts of each question are equally weighted. When solving one question/part, you may assume the results of all previous questions/parts. You may also assume all previous homework results and results from class or lecture notes, but please explain which result you are using when you use it.

1. Suppose that you have a dataset of $N = 10^8$ examples $z_1, \ldots, z_N$ and a loss function $\ell(w, z)$ relating model parameters w to examples z. Suppose that ℓ is G-Lipschitz and H-smooth and you are interested in minimizing the training error $\hat{L}(w) = \frac{1}{N}\sum_{i=1}^{N} \ell(w, z_i)$. Assume that it takes 1 millisecond to compute a gradient $\nabla\ell(w, z_i)$. For this problem we will ignore all other computing costs. Let $w_\star = \operatorname{argmin} \hat{\mathcal{L}}(w)$ and let w_1 be some given point. assume that $\|w_1 - w_\star\| \leq R$ and $\mathcal{L}(w_1) - \mathcal{L}(w_\star) \leq \Delta$ for some given R and Δ.

   (a) Suppose ℓ is also convex. If you run stochastic gradient descent (sampling a new z_t uniformly at random from $z_1, \ldots, z_N$ for each iteration), provide an upper bound on how long it will take (just counting gradient computation time) to find a point $\hat{w}$ satisfying $\mathbb{E}[\hat{L}(\hat{w}) - \hat{L}(w_\star)] \leq \epsilon$ as a function of ϵ, G, H, Δ, R and N (you may not need all variables in your expression). Describe how you will will find the point $\hat{w}$, and justify your expression for ϵ. You may provide your answer in any units you like.

   Solution:

   (b) Without supposing that ℓ is convex, if you still run stochastic gradient descent, provide an upper bound on how long it will take (again just counting gradient computation time) to find a point $\hat{w}$ satisfying $\mathbb{E}[\|\nabla\hat{\mathcal{L}}(\hat{w})\|] \leq \epsilon$ as a function of ϵ, G, H, Δ, R and N (you may not need all variables in your expression). Describe how you will will find the point $\hat{w}$, and justify your expression for ϵ.

   Solution:

   (c) Suppose that ℓ is non-convex and you have one day to train your model to obtain a point with the smallest possible value of $\mathbb{E}[\|\nabla\hat{\mathcal{L}}(\hat{w})\|] \leq \epsilon$. Is it reasonable to use variance reduction? Why or why not?

   Solution:

   (d) Suppose that ℓ is convex and you have one day to train your model to obtain a point with the smallest possible value of $\mathbb{E}[\hat{L}(\hat{w}) - \hat{L}(w_\star)]$. Is it reasonable to use variance reduction? Why or why not? (hint: this question can be subtle. For the purposes of this question, you may interpret "use variance reduction" to mean "use the algorithm for convex variance reduction discussed in class in exactly the way discussed". However, you may also propose alternative ways to use variance reduction, in which case your answer may vary. Regardless, you must justify your answer).

Solution:

2. A friend tells you they have developed a new first-order deterministic optimization algorithm. So long as the function is convex, Lipschitz, smooth and second-order smooth, their algorithm finds a point $\hat{w}$ such that $\mathcal{L}(\hat{w}) - \mathcal{L}(w_\star) \leq O(1/T^3)$ after T gradient evaluations. They tell you it avoids the lower-bound of $1/T^2$ because their algorithm requires $\mathcal{L}$ to be second-order smooth. Is their result plausible? Why or why not?

Solution:

3. The *support vector machine* (SVM) is a classical model in machine learning. They can be used for binary classification: given a input x, wish to predict $y \in \{-1, 1\}$. The SVM solves this problem by choosing a fixed *feature map* $\phi(x)$ that produces outputs in some vector space, and then predicting with $\hat{y} = \text{sign}(\langle \phi(x), \mathbf{w} \rangle)$ for some parameter $\mathbf{w}$. Technically, the vector space in which $\phi(x)$ and $\mathbf{w}$ live may be infinite dimensional, but for the purposes of this problem, you may assume $\phi(x) \in \mathbb{R}^d$ for some finite d if you wish. $\mathbf{w}$ is trained by minimizing the loss:

$$\mathcal{L}(\mathbf{w}) = \mathbb{E}[\ell(\mathbf{w}, (x, y))] = \mathbb{E}[\max(0, 1 - y\langle \phi(x), \mathbf{w} \rangle)]$$

Suppose that $\|\mathbf{w}_\star\| \leq R$ and $\|\phi(x)\| \leq B$ for all x. Show that with appropriate algorithm, given an i.i.d. dataset $(x_1, y_1), \ldots, (x_T, y_T)$, you can fine a point $\hat{\mathbf{w}}$ that ensures:

$$\mathbb{E}[\mathcal{L}(\hat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}_\star)] \leq O\left(\frac{RB}{\sqrt{T}}\right)$$

Solution: