Final Project in Advanced Lectures in Learning Theory

Roi Livni

January 11, 2021

Do 2 of the following parts. Please look at the last recorded lecture for guidance and explanations.

1 Part I

- In the center of gravity method we do not choose the last iteration. Show an example of a convex function where the sequence of values in the centers $f(c_1), \dots, f(c_t)$ is not monotonically decreasing: In particular, the last iteration need not be optimal.

- Let f be a non-negative β–smooth, not necessarily convex, function. Show that gradient descent with update step $\eta = \frac{1}{\beta}$ converges to a stationary point. More accurately, after finitely many steps the algorithm will reach a point x_t such that $\|\nabla f(x_t)\| < \epsilon$.

- Let $\{\mathbf{x}_i, y_i\}_{i=1}^m$ be a sequence of examples, when is the following loss function strongly convex (compute the parameter α as a function of the sequence of examples)

$$L(\mathbf{w}) = \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2$$

- Let ℓ be a non-negative 1-Lipschitz convex function. Assume also that for any y $\ell(0, y) \le 1$. Show that the solution, $\hat{w}$, to the following regularized objective

$$\text{minimize,}\ \frac{\lambda}{2}\|w\|^2 + \frac{1}{m}\sum_{i=1} \ell(w \cdot x_i, y_i)$$

yields (w.p $1-\delta$)

$$F(\hat{w}) \le F(w^\star) + \frac{\lambda}{2}B^2 + O\left(\frac{\log 1/\delta}{\sqrt{\lambda m}}\right).$$

Conclude that for choice $\lambda = O(1/B\sqrt[4]{m})$

$$F(\hat{w}) \leq \min_{\|w^\star\| \leq B} F(w^\star) + O\left(\frac{B \log 1/\delta}{\sqrt[4]{m}}\right).$$

- Show that the solution to the following equation,

  $$\text{minimize,} \quad \frac{1}{m}\sum_{i=1}^{m}(w \cdot x_i - y_i)^2,$$

  which we denote by $\hat{w}$, lies in the span span$(\{x_i\})$ is the minimal norm solution. Namely, for any other solution $\bar{w}$, we have that $\|\bar{w}\| \geq \|\hat{w}\|$.

- 1. Show that if a sequence of functions $\{\ell_i(\mathbf{w})\}_{i=1}^m$ is β–smooth then $L(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^m \ell_i(\mathbf{w})$ is β-smooth
  2. Show that the loss function $L(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^m \ell(\mathbf{w}, (x_i, y_i)$, where ℓ is the log loss, namely: $\ell(\mathbf{w}, (x_i, y_i) = -\log(1 + \exp(-\mathbf{w} \cdot \mathbf{x}_i y_i)$ is smooth for any choice of $\{x_i, y_i\}$ and compute the parameter β (may depend on the choice of sequence.

- Prove the following Lemma:

  Lemma. *Let K be a convex set, $x \in K$ and $y \in \mathbb{R}^n$ then,*

  $$(\Pi_K(y) - x)^T (\Pi_K(y) - y) \leq 0$$

2 Part II

In this part we will revisit and try to experimentally study the role of regularization, implicit bias, and optimization in learning. For this exercise you will need to download MNIST data set `http://yann.lecun.com/exdb/mnist/` on which we will run our experiments.

Remark. *Feel free to choose any other data set as long as you properly describe it as well as any preprocessing procedure you perform over it.*

- The original data-set is divided to train and test, do not use this partition, just construct one big data set (which you will later divide to train-test by yourself.

  Many of the results we depicted in this lecture assume the data is bounded in a ball of diameter D

  - You are allowed, and should, at this point perform any preprocessing that you believe will improve the results in the next section (scaling, shifting etc..). But make sure you describe every type of preprocessing you propose and did on the dataset.

  - It is also okay to reparameterize the original features and use any other representation of the data (for example, consider the output of the last hidden layer of some architecture you want). As long as you explain what you did, how you choose the features and how your choice effects the experiments.

- Next, we will want (for simplicity) to turn the above problem to binary. Generate a binary problem from the MNIST dataset (for example, by choosing two figures against another, or by clustering the figures to two sets. Create 3 different classification problem this way to experiment with. In each of these experiments we will want to learn a classifier by minimizing the loss

  $$\mathcal{L}(w) = \mathop{\mathbb{E}}_{(x,y)\sim D}[\ell(\mathbf{w}\cdot\mathbf{x}, y)].$$

  Where D is a distribution that uniformly pick a point from the dataset and ℓ is some **convex** loss function for your choice (e.g. square loss, hinge loss, log-loss etc...).

- We will study the performance of the following optimization algorithms: GD, Constrained GD, Regularized GD and SGD. Recall that for a loss function F:

  1. For GD we consider the update rule:

     $$w_t - \eta\nabla F(w_t).$$

  2. For constrained GD we consider the update rule[1]:

     $$w_{t+1/2} = w_{t+1} - \eta\nabla F(w_t) \quad w_{t+1} = \Pi_K(w_{t+1/2}).$$

  3. For regularized GD we perform GD over the following regularized objective

     $$\lambda\|w\|^2 + F(w).$$

  4. For SGD we, at each iteration, draw a fresh example $(x, y) \sim D_S$ from the data set, without repetitions, and perform an update step:

     $$w_{t+1} = w_t - \eta\nabla\ell(w\cdot x, y).$$

     (here we assume $F(w) = \frac{1}{t}\sum_{i=1}^{t}\ell(w\cdot x_i, y_i)$)

  We will want to compare the performance of each of these algorithms in terms of generalization and optimization.

[1] You might want to choose $K = \{w : \|w\| \le B\}$ for some set B but you can choose any other constraint. The parameter B depends on your preprocessing and discuss your choice (you can experiment with different choices)

Experiment A: Optimization First, we will want to compare the running time of each of these algorithms.

1. First, using the lecture notes and what we studied in class, choose "theoretically justified" learning rates and other parameters (such as λ in regularized GD). State your choice and provide the justification.
2. Compare the running time of each algorithm with the choice of parameters: Namely provide a curve that shows the optimization error vs. number of iterations with the give choice of parameters.
3. Experiment with other possible learning rates and regularization terms with the aim to improve each optimization method.
4. Discuss the results and conclude, are all algorithms equal or do some perform better?

Remark. *Remember that all algorithm should be tested on their performance over the loss $\mathcal{L}$ – note that at this point we are not distinguishing between empirical loss and true error.*

Experiment B: Generalization Here, we will divide our dataset into two: test set and train set. We will denote the train set by S and we will consider the empirical loss

$$\mathcal{L}_S(w) = \frac{1}{t}\sum_{i=1}^{t} \ell(\mathbf{w}\cdot\mathbf{x}, y).$$

you may want to perform the experiments several times by drawing different train and test sets.

- Repeat the last experiments with both the theroetically justified parameters as well as those you found to be empirically optimal for optimization: only this time measure the performance of each model in terms of generalization.
- Specifically, show the test error as your curve and not the optimziation error over the sample set.
- Show also the binary 0-1 loss.

3 Part III

In this part we will review a generalization bound for GD that was recently obtained in `https://arxiv.org/pdf/2006.06914.pdf`.

In this paper the authors show that, if $F = \mathbb{E}_{z\sim D}[f(w,z)]$ is a stochastic convex function, and we choose w_S by running GD for T steps with step size η over the empirical risk (i.e. $\frac{1}{|S|}\sum_{i=1}^{m} f(w,z_i)$ then for any step size, running GD leads to the following generalization bound (see section 3.4 therein):

$$\mathop{\mathbb{E}}_{S\sim D} F(w_S) \leq 4L^2\sqrt{T}\eta + \frac{4L^2T\eta}{n} + \frac{R^2}{2\eta T} + \frac{\eta L^2}{2}.$$

(Please look at the paper for the exact definition of R and L)

- Choose sample complexity, n a step size, η, no. of Oracle calls, T, that will lead to meaningfull guarantees. In other words, how large should n be and how should we choose η and T as functions of ϵ if we want to obtain a generalization gap of at most ϵ.
- Discuss the variant of GD that this result induces, compare it to the complexity of SGD in terms of sample complexity and running time.
- Compare GD and SGD under different choices of η, and T: In particular analyse the performance of both algorithms when we choose $\eta = O(1/\sqrt{T})$ as well as the η and T you obtained from the last questions (If you've done part II you can don't have to repeat any experiment you already done.