Recitation 12

Alex Dong

CDS, NYU

Fall 2020

Gradient Descent

- We made it! Last recitation
- Conceptually very simple; in practice, can be pretty tricky
  - How to pick α?
  - Many, MANY variations. SGD, Adagrad, Adam
  - How to deal with nondifferentiability?
  - A lot more will be covered in Machine Learning
- Analyzing convergence depends on eigenvalues of the Hessian.

Questions: Linear Separator

Let $(x_1, y_1), ..., (x_n, y_n)$ be a centered dataset.
Each $x_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$.
Assume that our points are *linearly seperable*.

1. Intuitively, describe what it means to be linearly separable.
2. Mathematically, give a condition for determining whether the points are linearly seperable or not. (Hint: use inner products)
3. Let our loss function be $\ell(w_i; y_i, x_i)) = e^{-f(w; y_i, x_i)}$. (Note: $f(w, x_i)$ is defined in the answer to the previous question). Write the loss function for the entire dataset.
4. What is the gradient descent update step?
5. What's the computational cost of the update step in terms of n, d?

Solutions 1: Linear Separator

Let $(x_1, y_1), ..., (x_n, y_n)$ be a centered dataset.
Each $x_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$.
We want to determine whether the points are linearly separable.

Solution

1. *Intuitively, describe what it means to be linearly separable.* *A two class dataset is linearly seperable when we can draw a hyperplane in the space, where all the points of one class fall entirely on one side of the hyperplane, and all the points of the other class fall entirely on the other side of the hyperplane.*

2. *Mathematically, give a condition for determining whether the points are linearly seperable or not. (Hint: use inner products)* *The dataset is seperable if* $\exists w$*, such that*

   $f(w; y_i, x_i) = y_i \langle w, x_i \rangle \geq 0, \forall i \in \{1, ..., n\}$

Solutions 2: Linear Separator

Let $(x_1, y_1), ..., (x_n, y_n)$ be a centered dataset.
Each $x_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$.
We want to determine whether the points are linearly separable.

Solution

3. *Let our loss function be* $\ell(w; y_i, x_i)) = e^{-f(w;y_i,x_i)}$.
   Here, we use $f(w; y_i, x_i) = y_i\langle w, x_i\rangle$. *The loss function will be*
   $L(w; (x_1, y_1), ..., (x_n, y_n)) = \sum_{i=1}^{n} e^{-y_i w^T x_i}$
4. *What is the gradient descent update step?*
   The gradient is $\nabla_w(L) = \sum_{i=1}^{n} -y_i x_i e^{-y_i w^T x_i}$.
   The update step is $w_{t+1} = w_t + \alpha \sum_{i=1}^{n} y_i x_i e^{-y_i w^T x_i}$
5. *What's the computational cost of the update step in terms of* n, d*?*
   y_i *is a scalar,* x_i *and* w *are d-dimensional vectors*
   Calculating $-y_i x_i e^{-y_i w^T x_i}$ *is about* d *calculations*
   Calculating $\sum_{i=1}^{n} -y_i x_i e^{-y_i w^T x_i}$ *is about* nd *calculations*

Questions: Gradient Descent

True or False:

1. Gradient descent will always find a global minimum.
2. Gradient descent will always find a local minimum.
3. Standardizing your features will help with gradient descent.

Conceptual

4. The step size in gradient descent is not actual “step size” because the true length of the step in gradient descent will be $\alpha \|\nabla_w L(w; y, x)\|$. Suppose we scale the gradients to make them unit norm before we do gradient descent. What’s the problem with this?
5. What happens to the speed of gradient descent for linear regression when we first perform some dimensionality reduction to the features?
6. What are possible stopping criteria the gradient descent algorithm?

Solutions 1: Gradient Descent

Solution

1. *Gradient descent will always find a global minimum.*
   False, it can find a local minimum, (but it could also diverge)
2. *Gradient descent will always find a local minimum.*
   False, if your step size is too large, you could diverge
3. *Standardizing your features will help with gradient descent. True. Gradient descent converges best when all the features are on the same scale.*
   Another way to think about this is that you don't want large differences in the eigenvalues of the Hessian at any point. If there are large differences in the eigenvalues of the Hessian, then that will mean you are in a valley.

Solutions 2: Gradient Descent

Solution

4. When you get close to the minimum, you will step away from it/ be unable to converge to it.

5. Assuming you reduce irrelevant dimensions, dimensionality reduction will "remove" the smallest eigenvalues of the covariance matrix (which turns out to be also the Hessian of the cost function), hence improving the condition number and therefore the speed of convergence. Also, iterations are computationally less expensive.

6. 1) Iterate a fixed amount of steps.
2) Loss function doesn't change much: $\|L(w_{t+1}) - L(w_t)\| < \epsilon$
3) weight parameter doesnt change much $\|w_{t+1} - w_t\| < \epsilon$
We tend to prefer (2), because we are specifically interested in minimizing loss. Also weights are in high dimension, and we dont know how sensitive they are. (Small change in weight could lead to large change in loss)