CS 412 — Introduction to Machine Learning (UIC) February 23, 2025

Lecture 10

Instructor: Aadirupa Saha *Scribe(s): Aniket Anil Wagde*

Overview

In the last lecture, we covered the following main topics:

1. The intuition behind the convergence of gradient descent
2. Newton's method of optimizing convex functions

This lecture focuses on:

1. Convergence proof of Gradient Descent using:
    (a) Lipschitz property
    (b) Convexity of functions
    (c) Projection function
2. Stochastic Gradient Descent
3. Batched Gradient Descent

1 Gradient Descent Algorithm

We want to find the point x in a bounded set $x \in \mathcal{X}$, such that a function $f : \mathcal{X} \to \mathbb{R}$ produces the minimum value at x.

$$f : \mathcal{X} \to \mathbb{R} \tag{1}$$
$$x^* = \arg\min_{x \in \mathcal{X}} f(x) \tag{2}$$

1.1 Algorithm

This is an iterative algorithm. Where at each time step we update the value of x, until it is within a certain error margin of x^* (the optimal point of the algorithm. The algorithm of gradient descent is as follows:

Algorithm 1.1: Gradient Descent Algorithm

```
Input: A bounded set $\mathcal{X} \in \mathbb{R}^d$, an L-Lipschitz function $f$
Output: A point $x \in \mathcal{X}$ such that $|f(x) - f(x^*)| \leq \varepsilon$, where $\varepsilon$ is an error margin
Initialize $x_0 \in \mathcal{X}$
for t = 1,2,3...T (timesteps) do
    $\tilde{x}_t \leftarrow x_{t-1} - \eta_t \nabla f(x_{t-1})$
    $x_t \leftarrow P_{\mathcal{X}}(\tilde{x}_t)$
end for
return $x_T$
```

Where η_t in equation 7 is the learning rate at time t. Typically a fixed parameter, but can change is some variations of the gradient descent algorithm. $P_{\mathcal{X}}$ from equation 8 is the projection function described in section 6.4.
Algorithm 1.1 leads to the following performance results under the assumptions:

Assumption 1. *$f : \mathcal{X} \mapsto \mathbb{R}$ is a convex function. Refer to section 6.2.*

Assumption 2. *$\mathcal{X}$ is a bounded set in $\mathbb{R}^d$. Refer to section 6.4.*

Assumption 3. *f is L-Lipschitz. Refer to section 6.1.*

Assumption 2 and Assumption 3 lead to the following performance bound:

Theorem 1.1: CHANGOEEEEEEE!

For β being the diameter of the bounded set $\mathcal{X}$, $||x_1 - x_2||_2^2 \leq \beta$, where $x_1, x_2 \in \mathcal{X}$. After Gradient Descent is run over T timesteps. We can show that

$$f(\bar{x}_T) - f(x^*) \leq \frac{2\sqrt{DT}}{\sqrt{T}}$$

Where $\bar{x}_t$ is the average of the values of x over T timesteps.

1.2 Convergence proof of projected-Gradient Descent for convex functions

As f is a convex function, for any value of the iteration number s out of a maximum of t iterations. it follows from the property 25 that:

$$\begin{aligned}
f(y) &\geq f(x) + \nabla f(x)(y - x) \qquad \forall s \in 1, 2, ...t \\
&\text{let} \quad x = x^* \quad \text{(Optimal Point)} \\
&\text{let} \quad y = x_s \quad \text{(Value of } x \text{ at iteration } s\text{)} \\
f(x^*) &\geq f(x_s) + \nabla f(x_s)(x_s - x^*) \qquad \forall s \in 1, 2, ...t \\
f(x^*) - f(x_s) &\geq \nabla f(x_s)(x_s - x^*) \qquad \forall s \in 1, 2, ...t \\
f(x_s) - f(x^*) &\leq \nabla f(x_s)(x_s - x^*) \qquad \forall s \in 1, 2, ...t \qquad (3)
\end{aligned}$$

Let the learning rate η be time invariant, that is $\eta_1 = \eta_2 = \eta_3 \cdots = \eta_n = \eta$. Also $\tilde{x}_t$ represents the value of x updated at step 5 from 1.1 before the projection function is used. From the gradient update rule in 7 from Algorithm 1.1, we have:

$$\begin{aligned}
&\tilde{x}_t = x_{t-1} - \eta \nabla f(x_{t-1}) \\
&\text{Replacing } t \text{ with } s \\
&\tilde{x}_s = x_{s-1} - \eta \nabla f(x_{s-1}) \\
&\tilde{x}_s - x_{s-1} = -\eta \nabla f(x_{s-1}) \\
&x_{s-1} - \tilde{x}_s = \eta \nabla f(x_{s-1}) \\
&\frac{x_{s-1} - \tilde{x}_s}{\eta} = \nabla f(x_{s-1}) \\
&\frac{x_s - \tilde{x}_{s+1}}{\eta} = \nabla f(x_s) \qquad (4)
\end{aligned}$$

Using the gradient descent update value from (4) in (3), we get:

$$f(x_s) - f(x^*) \leq \frac{(x_s - \tilde{x}_{s+1})}{\eta}(x_s - x^*) \qquad \forall s \in 1, 2, ...t$$

After applying the projection step. $\tilde{x}_t \leftarrow P_{\mathcal{X}}(x_t)$ This translates into:

$$f(x_s) - f(x^*) \leq \frac{(x_s - \tilde{x}_{s+1})}{\eta}(x_s - x^*) \qquad \forall s \in 1, 2, ...t$$

Using the identity $[||a - b||_2^2 = a^T a - 2a^T b + b^t b]$, this can be shown to be equivalent to:

$$f(x_s) - f(x^*) \leq \frac{1}{2\eta}(||x_s - x^*||_2^2 + ||x_s - \tilde{x}_{s+1}||_2^2 - ||\tilde{x}_{s+1} - x^*||_2^2) \qquad \forall s \in 1, 2, ...t \qquad (5)$$

From (27) and the fact that $x^* \in \mathcal{X}$ and $x_{s+1} = P_{\mathcal{X}}(\tilde{x}_{s+1})$ we know that:

$$||x_{s+1} - x^*||_2^2 \leq ||\tilde{x}_{s+1} - x_t||_2^2 \qquad (6)$$

Applying (6) to (7) we get the bound after applying the projection function:

$$f(x_s) - f(x^*) \leq \frac{1}{2\eta}(||x_s - x^*||_2^2 + ||x_s - \tilde{x}_{s+1}||_2^2 - ||x_{s+1} - x^*||_2^2) \qquad \forall s \in 1, 2, ...t \qquad (7)$$

From the gradient descent update rule in 1.1 and 1 we know that:

$$||x_s - \tilde{x}_{s+1}||_2^2 = ||\eta \nabla f(x_s)||_2^2 \leq \eta^2 L^2 \qquad (8)$$

Lemma 1 (L-Lipschitz l-2 norm lemma)**.** *If $f : \mathcal{X} \mapsto \mathbb{R}$ is a convex function such that f is L-Lipschitz in the l-2 norm ($||.||_2$) then $\forall x \in \mathcal{X} \quad ; ||\nabla f(x)||_2 \leq L$.*
For proof refer to section 5.3.

Now applying (8) to (7) we get

$$f(x_s) - f(x^*) \leq \frac{1}{2\eta}(||x_s - x^*||_2^2 - ||x_{s+1} - x^*||_2^2) + \frac{\eta L^2}{2} \qquad \forall s \in 1, 2, ...t \tag{9}$$

We can now sum (9) over all the T steps to give us the average reward.

$$\sum_{s=1}^{T} [f(x_s) - f(x^*)] \leq \frac{1}{2\eta} \sum_{s=1}^{T} \left[||x_s - x^*||_2^2 - ||x_{s+1} - x^*||_2^2\right] + \frac{T\eta L^2}{2} \tag{10}$$

On expanding the summation and canceling out terms with opposite signs (10) can be simplified to:

$$\sum_{s=1}^{T} [f(x_s) - f(x^*)] \leq \frac{1}{2\eta} \left(||x_1 - x^*||_2^2 - ||x_{t+1} - x^*||_2^2\right) + \frac{T\eta L^2}{2} \tag{11}$$

As $-||x_{t+1} - x^*||_2^2$ is guaranteed to be negative, we can remove it, and the inequality will still hold true:

$$\sum_{s=1}^{T} [f(x_s) - f(x^*)] \leq \frac{1}{2\eta} \left(||x_1 - x^*||_2^2\right) + \frac{T\eta L^2}{2} \tag{12}$$

Now since we are operating in the bounded set $\mathcal{X}$ with diameter β, we know for a fact that $||x_1 - x^*||_2^2 \leq \beta^2$. Hence (12) can be re-written as:

$$\sum_{s=1}^{T} [f(x_s) - f(x^*)] \leq \frac{\beta^2}{2\eta} + \frac{T\eta L^2}{2} \tag{13}$$

The optimal value of η can be shown to be $\eta = \frac{L\sqrt{T}}{\beta}$, refer to 5.2. Using this value of η in (13) and simplifying we get:

$$\sum_{s=1}^{T} [f(x_s) - f(x^*)] \leq L\beta\sqrt{T}$$

Dividing both sides by T:

$$\frac{1}{T} \sum_{s=1}^{T} [f(x_s) - f(x^*)] \leq \frac{L\beta}{\sqrt{T}} \tag{14}$$

Since f is a convex function, it can be shown from the properties of convexity that:

$$\frac{1}{T} \sum_{s=1}^{T} f(x_s) \geq f\left(\sum_{s=1}^{T} \frac{x_s}{T}\right) \tag{15}$$

Applying (15) to (14), we get the final convergence bound of

$$f\left(\sum_{s=1}^{T} \frac{x_s}{T}\right) - f(x^*) \leq \frac{L\beta}{\sqrt{T}} \tag{16}$$

□

This concludes the convergence proof of Gradient Descent for convex, L-Lipschitz functions for $f : \mathcal{X} \mapsto \mathbb{R}$ for $\eta = \frac{L\sqrt{T}}{\beta}$.

2 Drawbacks of Gradient Descent

Let us attempt to use Gradient Descent for logistic regression objective. Assume that we have a dataset where n is the number of data points.

$$D = \{(x_i, y_i)\}_{i=1}^{n}$$

The objective function (over the empirical logistic loss) over n data points is defined as:

$$argmin_{W \in \mathbb{R}^d} \left[\frac{1}{n} \sum_{i=1}^{n} y_i log(\frac{1}{1+e^{-W^T x_i}}) + (1-y_i) log(\frac{1}{1+e^{W^T x_i}}) \right]$$

Now we isolate the loss and attempt to use gradient descent to optimize the loss with respect to W.
We denote the loss as

$$f(W) = \frac{1}{n} \sum_{i=1}^{n} y_i log(\frac{1}{1+e^{-W^T x_i}}) + (1-y_i) log(\frac{1}{1+e^{W^T x_i}}) \tag{17}$$

Applying Gradient Descent: First we initialize

$$W_0 \in \mathbb{R}^d$$

. Next we iterate the gradient update as:

$$W_{n+1} \leftarrow W_n - \eta \nabla f(W)$$

.
The computational complexity of calculating $\nabla f(W)$ is:

$$n \nabla f(W) = \sum_{i=1}^{n} \left[y_i - \frac{1}{1+e^{-W^T x_i}} \right]$$

Where $x_i \in \mathbb{R}^d$

This calculation needs to be performed over the entire dataset Therefore the computational complexity for a single step of Gradient Descent is $O(n \times d)$. Considering that the algorithm will require many iteration to converge close to the global minima, assuming T steps. The total computational complexity will be $O(n \times d \times T)$, which is infeasible to compute. A popular solution to this problem of infeasability is a technique called Stochastic Gradient Descent. Which is described in section 3.

3 Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is a small change to the gradient descent algorithm. Instead of calculating the gradient over the entire dataset and then updating the weights, SGD calculates the gradient at a single data point and updates the weights from that data point. It iterates this step multiple times until it converges.

3.1 Algorithm

Algorithm 3.1: Stochastic Gradient Descent Algorithm

```
**Input:** A bounded set $\mathcal{X} \in \mathbb{R}^d$, an L-Lipschitz function $f$
**Output:** A point $x \in \mathcal{X}$ such that $|f(x) - f(x^*)| \leq \varepsilon$, where $\varepsilon$ is an error margin
Initialize $x_0 \in \mathcal{X}$
**for** t = 1,2,3...T (timesteps) **do**
    Sample $i_t \sim Uniform[n]$
    Compute $\nabla f_{i_t}(x_{i_t})$
    $\tilde{x}_t \leftarrow x_{t-1} - \eta_t \nabla f_{i_t}(x_{i_t})$
    $x_t \leftarrow P_{\mathcal{X}}(\tilde{x}_t)$
**end for**
**return** $x_T$
```

3.2 Convergence of SGD

Taking the same assumptions 1, 2, 3 as for gradient descent. Setting the value of $\eta = \frac{\beta}{L\sqrt{T}}$, we have:

$$E\left[f(\bar{x}_t) - f(x^*)\right] \leq \frac{\beta}{L\sqrt{T}}$$

$$\text{Where:} \quad \bar{x}_T = \frac{\left(\sum_{s=1}^{T} x_s\right)}{T}$$

Here E represents the expectation over the entire algorithm. Since the weights are updated with a single datapoint each time, it is far more random. This bound works over the expectation, accounting for the randomness of the learning.

Note that if f is α-Strongly Convex and L-Lipschits then there is a significantly tighter bound on convergence. Setting $\eta_s = \frac{1}{\alpha s}$.

$$E\left[f(\bar{x}_t) - f(x^*)\right] \leq \frac{L^2}{2\alpha T}(1 + log(T+1))$$

$$\text{Where:} \quad \bar{x}_T = \frac{\left(\sum_{s=1}^{T} x_s\right)}{T}$$

4 m-Batched Stochastic Gradient Descent

This algorithm was built to act as a middle ground between Gradient Descent and Stochastic Gradient Descent. It combines the advantages of Gradient Descent with the advantages of Stochastic Gradient Descent. It uses a batch of data points to smoothen the learning process, while giving the flexibility to the user to choose a batch size that is computationally tractable based on the computational capacity of the system training the model.

Method	**Advantages**	**Disadvantages**
Classical Gradient Descent	1. Comes with stronger and relatively faster convergence guarantee 2. Suitable for small number of data points	1. Slower updates 2. Computationally heavier 3. Gets sutck in local minimas if f is non-convex 4. Time Consuming
Stochastic Gradient Descent	1. Fast 2. Simple update 3. Suitable for large number of datapoints.	2. Poorer convergence guarantee 2. Easily gets stuck in local minima or saddle points

Table 1: Comparison of Gradient Descent and Stochastic Gradient Descent

Algorithm 4.1: Stochastic Gradient Descent Algorithm

```
Input: A bounded set $\mathcal{X} \in \mathbb{R}^d$, an L-Lipschitz function $f$
Output: A point $x \in \mathcal{X}$ such that $|f(x) - f(x^*)| \leq \varepsilon$, where $\varepsilon$ is an error margin
Initialize $x_0 \in \mathcal{X}$
for t = 1,2,3...T (timesteps) do
    Sample $i_t^1, i_t^2, \ldots i_t^m, \sim Uniform[n]$
    Compute $g_t(x_t) = \frac{1}{m} \sum_{j=1}^m \nabla f_{i_t} j(x_{i_t})$
    $\tilde{x}_{t+1} \leftarrow x_t - \eta_t g_t(x_t)$
    $x_{t+1} \leftarrow P_{\mathcal{X}}(\tilde{x}_{t+1})$
end for
return $x_T$
```

5 Appendix

5.1 Further Reading

Here are some other algorithms that are interesting to refer to:

- Nostrov's Accelerated GD
- Momentum Based GD
- ADAGRAD (Adaptive Gradient Descent)
- RMSprop (Root Mean-Square Propagation)
- ADAM (Combining Momentum Based GD and RMSprop)

5.2 Setting optimal value of η

Our goal is to set the value of η such that it minimizes the following expression:

$$\frac{\beta^2}{2\eta} + \frac{\eta L^2}{2}$$

To do this we can simply differentiate the expression and equate it to zero. We get the following expresison:

$$\frac{-\beta^2}{2\eta^2} + \frac{L^2}{2} = 0$$

On simplifying

$$\eta = \frac{L\sqrt{T}}{\beta}$$

This value of eta is then used in the Gradient Descent algorithm.

5.3 Proving Lipschitz lemma for l-2 norm

To show: If $f : \mathcal{X} \mapsto \mathbb{R}$ is a convex function such that f is L-Lipschitz in the l-2 norm ($||.||_2$) then $\forall x \in \mathcal{X} \quad ; ||\nabla f(x)||_2 \leq L$. Proof: Let $y = x + \nabla f(x)$ for any $x \in \mathcal{X}$.
Then by convexity:

$$f(y) - f(x) \geq \nabla f(x)(y - x) = ||\nabla f(x)||_2^2 \tag{18}$$

Also, since f is L-Lipschitz (section 6.1), we know that:

$$|f(y) - f(x)| \leq L||x - y|| = L||\nabla f(x)||_2 \tag{19}$$

Now combining (18) and (19) we get:

$$\begin{aligned} &||\nabla f(x)||_2^2 \leq f(y) - f(x) \leq |f(y) - f(x)| \leq L||\nabla f(x)||_2 \\ \implies &||\nabla f(x)||_2 \leq L \end{aligned} \tag{20}$$

□

A noteworthy extension of this is: $\forall x, y \in \mathcal{X}$ by convexity of f we have:

$$f(y) \geq f(x) + \nabla f^T(x)(y - x)$$

$$f(x) - f(y) \leq \nabla f(x)(x - y) \leq ||\nabla f(x)||_2 \; ||x - y||_2 \tag{21}$$

$$|f(x) - f(y)| \leq ||\nabla f(x)||_2 \; ||x - y||_2 \leq M||x - y||_2 \tag{22}$$

$$\tag{23}$$

So f is always M-Lipschitz when $M = \max_x ||\nabla f(x)||_2$

6 Background

6.1 L-Lipschitz

Definition 1 (L-Lipschitz)**.** *A function $f : \mathcal{X} \mapsto \mathbb{R}; \mathcal{X} \subseteq \mathbb{R}^d$ is called L-Lipschitz if $|f(x) - f(y)| \leq L||x - y||_n$.*

Then f is L-lipschitz in $||.||_n$ (in this lecture we only refer to L-Lipschitz in the 2-norm).
An intuitive way to think about it is that the function is allowed only to increase or decrease at a maximum rate defined by L.

6.2 Convexity

Definition 2 (Convexity)**.** *A function $f : \mathcal{X} \mapsto \mathbb{R}; \mathcal{X} \subseteq \mathbb{R}^d$ is called convex if it has any of the three equivalent following properties for $\lambda : [0, 1]$:*

$$f(\lambda(x) + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda) f(y) \qquad \forall x, y \in \mathcal{X} \tag{24}$$

$$f(y) \geq f(x) + \nabla f^T(x)(y - x) \qquad \forall x, y \in \mathcal{X} \tag{25}$$

$$\nabla^2 f(x) \geq 0 \qquad \forall x \in \mathcal{X} \tag{26}$$

Some additional properties of convex functions are:

1. If f and g are both convex functions, then for $h(x) = f(x) + g(x)$, h is also convex.
2. For $h(x) = max\{f_1(x), f_1(x)...f_m(x)\}$, if f_i is convex $\forall i \in \{1, 2, ...m\}$, then $h(x)$ is also convex.

6.3 α-Strongly Convex functions

A function is said to be α-strongly convex [1] if there exists a value of α such that the following inequality is true:

$$f(y) \geq f(x) + \nabla f(x)(y - x) + \frac{\alpha}{2}||y - x||_2^2$$

6.4 Bounded Sets and Projection Functions

Definition 3 (Bounded Set)**.** *A set $\mathcal{X}$ is a bounded set in $\mathbb{R}^d$ if and only if, $\forall x_1, x_2 \in \mathcal{X}, ||x_1 - x_2||_2 \leq \beta$, where β in any constant value.*

The purpose of a projection function is to take as input any point that lies outside a bounded set, and translate it inside the bounded set. Here is an example of a projection function $P_{\mathcal{X}}$, that projects onto bounded set $\mathcal{X}$.

$$x_t \leftarrow P_{\mathcal{X}}(\tilde{x}_t)$$

$$P_{\mathcal{X}}(x) = \min_{y \in \mathcal{X}} ||x - y||_2^2$$

This can be thought of as the projection that translates a point x to another point in the bounded set $\mathcal{X}$ with the minimum l-2 distance from the original point. Also, if $x_{t+1} \in \mathcal{X}$ then $P_{\mathcal{X}}(x_{t+1}) = x_{t+1}$.

Note that not all projection functions need to minimize the distance between the point in the bounded set and the input point, this is just a useful projection. Here is a useful property of this projection function is:

$$||\tilde{x}_{t+1} - x_t||_2^2 \geq ||x_{t+1} - x_t||_2^2 \tag{27}$$

While it is infeasible for the projection function to test every point in the bounded set to find the point with the minimum distance from the input point. The distance function (where x is fixed): $D(x, y) = ||x - y||_2^2$ is convex in y, and thus can be solved by simply taking the derivative with respect to y and equating to 0. Hence, we can quickly find the output of the projection function.

Next Lecture

The next lecture will cover the following topics:

1. Max-Margin Formulation
2. SVM objective
3. KKT conditions

References:

1. https://www.stat.cmu.edu/ siva/teaching/725/lec2.pdf