CS-GY 6763: Lecture 8
Second order conditions

NYU, Prof. Ainesh Bakshi

Gradient Descent

Conditions:

- **Convexity:** f is a convex function, $\mathcal{S}$ is a convex set.
- **Bounded Initial Distance:**

$$\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2 \leq R$$

- **Bounded gradients (Lipschitz function)**:

$$\|\nabla f(\mathbf{x})\|_2 \leq G \text{ for all } \mathbf{x} \in \mathcal{S}.$$

Theorem (GD Convergence Bound)

(Projected) Gradient Descent outputs $\hat{\mathbf{x}}$ *with*

$$f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x}) + \epsilon, \quad \textit{after} \quad T = \frac{R^2 G^2}{\epsilon^2} \textit{ iterations.}$$

Beyond the Basic Bound

- The previous bounds are optimal for convex first order optimization in general.

Beyond the Basic Bound

- The previous bounds are optimal for convex first order optimization in general.
- But in practice, the dependence on $1/\epsilon^2$ is pessimistic: gradient descent typically requires far fewer steps to reach ϵ error.

Beyond the Basic Bound

- The previous bounds are optimal for convex first order optimization in general.
- But in practice, the dependence on $1/\epsilon^2$ is pessimistic: gradient descent typically requires far fewer steps to reach ϵ error.
- Previous bounds only make a very weak first order assumption:

$$\|\nabla f(x)\|_2 \leq G.$$

Beyond the Basic Bound

- The previous bounds are optimal for convex first order optimization in general.
- But in practice, the dependence on $1/\epsilon^2$ is pessimistic: gradient descent typically requires far fewer steps to reach ϵ error.
- Previous bounds only make a very weak first order assumption:

$$\|\nabla f(x)\|_2 \leq G.$$

- In practice, many functions satisfy stronger assumptions.

Second Order Conditions

- Often possible to place assumptions on the second derivative of f.

Second Order Conditions

- Often possible to place assumptions on the second derivative of f.
- In particular, we say that a scalar function f is α-strongly convex and β-smooth if for all x:

$$\alpha \leq f''(x) \leq \beta.$$

Second Order Conditions

- Often possible to place assumptions on the second derivative of f.
- In particular, we say that a scalar function f is α-strongly convex and β-smooth if for all x:

$$\alpha \leq f''(x) \leq \beta.$$

- We will give an appropriate generalization of these conditions to multi-dimensional functions shortly.

Second Order Conditions

- Often possible to place assumptions on the second derivative of f.
- In particular, we say that a scalar function f is α-strongly convex and β-smooth if for all x:

$$\alpha \leq f''(x) \leq \beta.$$

- We will give an appropriate generalization of these conditions to multi-dimensional functions shortly.
- **Takeaway:** Having either an upper or lower bound on the second derivative helps convergence. Having both helps a lot.

Improving Gradient Descent

Takeaway: Having either an upper and lower bound on the second derivative helps convergence. Having both helps a lot.

Number of iterations for ϵ error:

	G-Lipschitz	β-smooth
R bounded start	$O\left(\frac{G^2R^2}{\epsilon^2}\right)$	$O\left(\frac{\beta R^2}{\epsilon}\right)$
α-strong convex	$O\left(\frac{G^2}{\alpha\epsilon}\right)$	$O\left(\frac{\beta}{\alpha}\log(1/\epsilon)\right)$

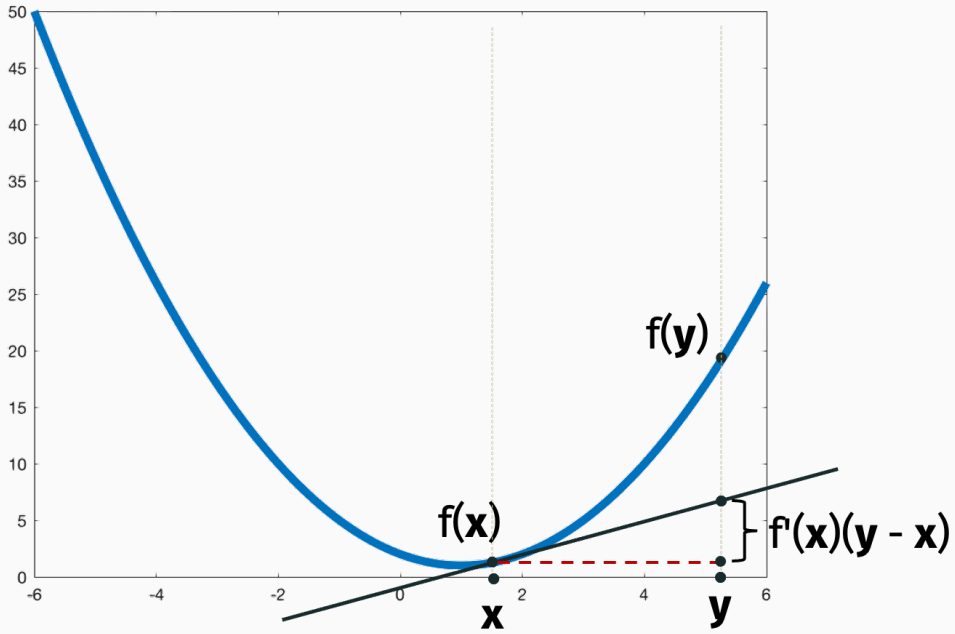
As we defined them so far, smoothness and strong convexity require f to be twice differentiable. On the other hand, gradient descent only requires first order differentiability.

Second Order Conditions

Equivalent conditions:

$$f''(x) \leq \beta \Rightarrow [f(y) - f(x)] - f'(x)(y-x) \leq \frac{\beta}{2}(y-x)^2$$

$$f''(x) \geq \alpha \Rightarrow [f(y) - f(x)] - f'(x)(y-x) \geq \frac{\alpha}{2}(y-x)^2$$



Recall: For all convex functions $[f(y) - f(x)] - f'(x)(y-x) \geq 0$.

Second Order Conditions

Proof that $f''(x) \leq \beta \Rightarrow [f(y) - f(x)] - f'(x)(y-x) \leq \frac{\beta}{2}(y-x)^2$:

Step 1: By FTC, $f(y) - f(x) = \int_x^y f'(t)\,dt$, so:

$$[f(y) - f(x)] - f'(x)(y-x) = \int_x^y [f'(t) - f'(x)]\,dt.$$

Second Order Conditions

Proof that $f''(x) \leq \beta \Rightarrow [f(y) - f(x)] - f'(x)(y-x) \leq \frac{\beta}{2}(y-x)^2$:

Step 1: By FTC, $f(y) - f(x) = \int_x^y f'(t)\,dt$, so:

$$[f(y) - f(x)] - f'(x)(y-x) = \int_x^y [f'(t) - f'(x)]\,dt.$$

Step 2: Apply FTC again: $f'(t) - f'(x) = \int_x^t f''(s)\,ds$, so:

$$\int_x^y [f'(t) - f'(x)]\,dt = \int_x^y \int_x^t f''(s)\,ds\,dt.$$

Second Order Conditions

Proof that $f''(x) \leq \beta \Rightarrow [f(y) - f(x)] - f'(x)(y-x) \leq \frac{\beta}{2}(y-x)^2$:

Step 1: By FTC, $f(y) - f(x) = \int_x^y f'(t)\,dt$, so:

$$[f(y) - f(x)] - f'(x)(y-x) = \int_x^y [f'(t) - f'(x)]\,dt.$$

Step 2: Apply FTC again: $f'(t) - f'(x) = \int_x^t f''(s)\,ds$, so:

$$\int_x^y [f'(t) - f'(x)]\,dt = \int_x^y \int_x^t f''(s)\,ds\,dt.$$

Step 3: Since $f''(s) \leq \beta$:

$$\int_x^y \int_x^t f''(s)\,ds\,dt \leq \beta \int_x^y (t-x)\,dt = \frac{\beta}{2}(y-x)^2. \quad \square$$

Second Order Conditions

Proof that $f''(x) \leq \beta \Rightarrow [f(y) - f(x)] - f'(x)(y-x) \leq \frac{\beta}{2}(y-x)^2$:

Step 1: By FTC, $f(y) - f(x) = \int_x^y f'(t)\,dt$, so:

$$[f(y) - f(x)] - f'(x)(y-x) = \int_x^y [f'(t) - f'(x)]\,dt.$$

Step 2: Apply FTC again: $f'(t) - f'(x) = \int_x^t f''(s)\,ds$, so:

$$\int_x^y [f'(t) - f'(x)]\,dt = \int_x^y \int_x^t f''(s)\,ds\,dt.$$

Step 3: Since $f''(s) \leq \beta$:

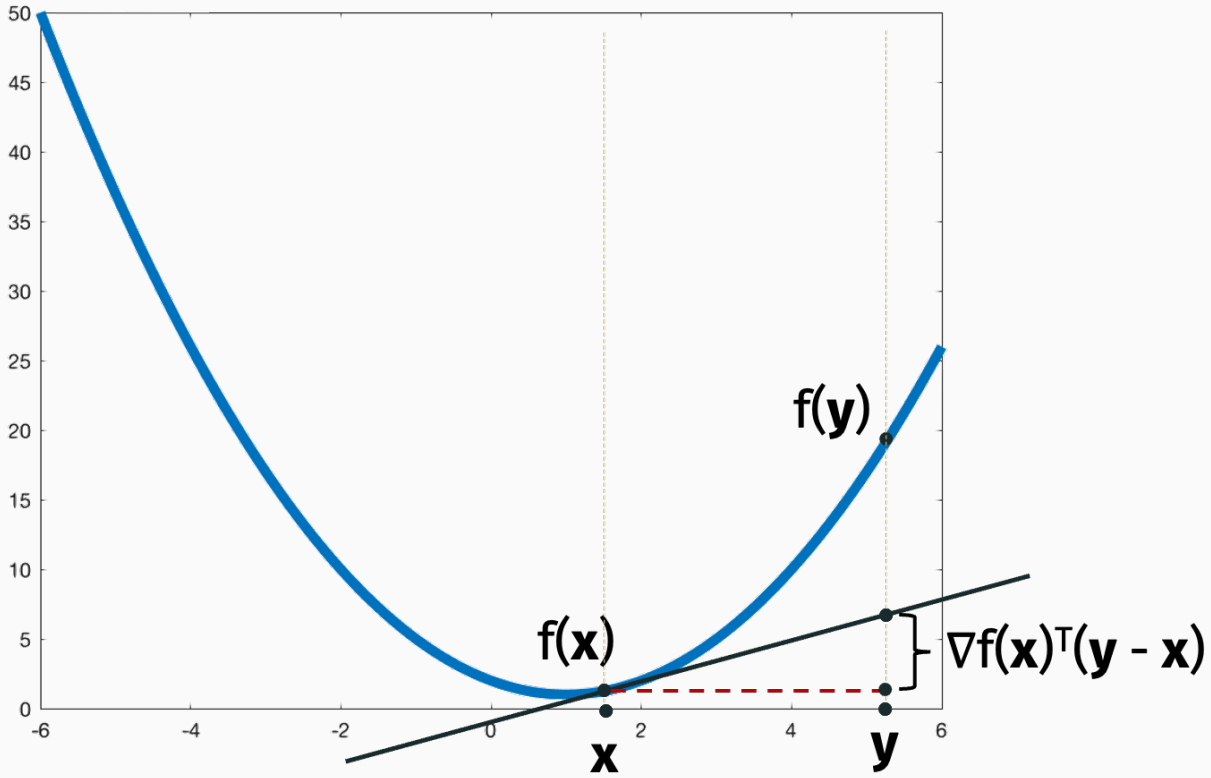
$$\int_x^y \int_x^t f''(s)\,ds\,dt \leq \beta \int_x^y (t-x)\,dt = \frac{\beta}{2}(y-x)^2. \quad \square$$

Proof for α-strongly convex is similar.

Multidimensional Generalization

A function is α-strongly convex and β-smooth if for all $\mathbf{x}$, $\mathbf{y}$:

$$\frac{\alpha}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \leq [f(\mathbf{y}) - f(\mathbf{x})] - \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) \leq \frac{\beta}{2}\|\mathbf{y}-\mathbf{x}\|_2^2$$



Convergence Guarantee

Theorem (GD Convergence for β-Smooth Functions)

Let f be a β-smooth convex function with $\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_2 \leq R$. Running GD for T steps gives:

$$f\left(\mathbf{x}^{(T)}\right) - f\left(\mathbf{x}^*\right) \leq \frac{2\beta R^2}{T}.$$

Corollary: If $T = O\left(\frac{\beta R^2}{\epsilon}\right)$ then $f\left(\mathbf{x}^{(T)}\right) - f\left(\mathbf{x}^*\right) \leq \epsilon$.

Compare to $T = O\left(\frac{G^2 R^2}{\epsilon^2}\right)$ without a smoothness assumption.

Guaranteed Progress

Gradient descent for β-smooth functions:

- Select starting point $\mathbf{x}^{(0)}$, $\eta = 1/\beta$.
- For $i = 0, \ldots, T$:
  - $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f\left(\mathbf{x}^{(i)}\right)$
- Return $\hat{\mathbf{x}} = \arg\min_i f\left(\mathbf{x}^{(i)}\right)$.

Why do you think gradient descent might be faster when a function is β-smooth?

Guaranteed Progress

Gradient descent for β-smooth functions:

- Select starting point $\mathbf{x}^{(0)}$, $\eta = 1/\beta$.
- For $i = 0, \ldots, T$:
  - $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f\left(\mathbf{x}^{(i)}\right)$
- Return $\hat{\mathbf{x}} = \arg\min_i f\left(\mathbf{x}^{(i)}\right)$.

Why do you think gradient descent might be faster when a function is β-smooth?

Intuitively, smoothness corresponds to the gradient being well-behaved everywhere.

Guaranteed Progress

With step size $\eta = \frac{1}{\beta}$, we have $\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)} = -\frac{1}{\beta}\nabla f\left(\mathbf{x}^{(t)}\right)$.

Claim: $f\left(\mathbf{x}^{(t)}\right) - f\left(\mathbf{x}^{(t+1)}\right) \geq \frac{1}{2\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2$.

Proof:

1. Apply β-smoothness at $\mathbf{x}^{(t)}, \mathbf{x}^{(t+1)}$:

$$f\left(\mathbf{x}^{(t+1)}\right) \leq f\left(\mathbf{x}^{(t)}\right) + \nabla f\left(\mathbf{x}^{(t)}\right)^T\left(\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\right) + \frac{\beta}{2}\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|_2^2.$$

Guaranteed Progress

With step size $\eta = \frac{1}{\beta}$, we have $\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)} = -\frac{1}{\beta}\nabla f\left(\mathbf{x}^{(t)}\right)$.

Claim: $f\left(\mathbf{x}^{(t)}\right) - f\left(\mathbf{x}^{(t+1)}\right) \geq \frac{1}{2\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2$.

Proof:

1. Apply β-smoothness at $\mathbf{x}^{(t)}, \mathbf{x}^{(t+1)}$:

$$f\left(\mathbf{x}^{(t+1)}\right) \leq f\left(\mathbf{x}^{(t)}\right) + \nabla f\left(\mathbf{x}^{(t)}\right)^T\left(\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\right) + \frac{\beta}{2}\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|_2^2.$$

2. Substitute $\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)} = -\frac{1}{\beta}\nabla f\left(\mathbf{x}^{(t)}\right)$:

$$f\left(\mathbf{x}^{(t+1)}\right) \leq f\left(\mathbf{x}^{(t)}\right) - \frac{1}{\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2 + \frac{\beta}{2}\|\frac{1}{\beta}\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2$$

Guaranteed Progress

With step size $\eta = \frac{1}{\beta}$, we have $\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)} = -\frac{1}{\beta}\nabla f\left(\mathbf{x}^{(t)}\right)$.

Claim: $f\left(\mathbf{x}^{(t)}\right) - f\left(\mathbf{x}^{(t+1)}\right) \geq \frac{1}{2\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2$.

Proof:

1. Apply β-smoothness at $\mathbf{x}^{(t)}, \mathbf{x}^{(t+1)}$:

$$f\left(\mathbf{x}^{(t+1)}\right) \leq f\left(\mathbf{x}^{(t)}\right) + \nabla f\left(\mathbf{x}^{(t)}\right)^T\left(\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\right) + \frac{\beta}{2}\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|_2^2.$$

2. Substitute $\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)} = -\frac{1}{\beta}\nabla f\left(\mathbf{x}^{(t)}\right)$:

$$\begin{aligned} f\left(\mathbf{x}^{(t+1)}\right) &\leq f\left(\mathbf{x}^{(t)}\right) - \frac{1}{\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2 + \frac{\beta}{2}\|\frac{1}{\beta}\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2 \\ &= f\left(\mathbf{x}^{(t)}\right) - \frac{1}{2\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2. \end{aligned}$$

Guaranteed Progress

With step size $\eta = \frac{1}{\beta}$, we have $\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)} = -\frac{1}{\beta}\nabla f\left(\mathbf{x}^{(t)}\right)$.

Claim: $f\left(\mathbf{x}^{(t)}\right) - f\left(\mathbf{x}^{(t+1)}\right) \geq \frac{1}{2\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2$.

Proof:

1. Apply β-smoothness at $\mathbf{x}^{(t)}, \mathbf{x}^{(t+1)}$:

$$f\left(\mathbf{x}^{(t+1)}\right) \leq f\left(\mathbf{x}^{(t)}\right) + \nabla f\left(\mathbf{x}^{(t)}\right)^T\left(\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\right) + \frac{\beta}{2}\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|_2^2.$$

2. Substitute $\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)} = -\frac{1}{\beta}\nabla f\left(\mathbf{x}^{(t)}\right)$:

$$\begin{aligned} f\left(\mathbf{x}^{(t+1)}\right) &\leq f\left(\mathbf{x}^{(t)}\right) - \frac{1}{\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2 + \frac{\beta}{2}\|\frac{1}{\beta}\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2 \\ &= f\left(\mathbf{x}^{(t)}\right) - \frac{1}{2\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2. \end{aligned}$$

Guaranteed Progress

With step size $\eta = \frac{1}{\beta}$, we have $\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)} = -\frac{1}{\beta}\nabla f\left(\mathbf{x}^{(t)}\right)$.

Claim: $f\left(\mathbf{x}^{(t)}\right) - f\left(\mathbf{x}^{(t+1)}\right) \geq \frac{1}{2\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2$.

Proof:

1. Apply β-smoothness at $\mathbf{x}^{(t)}, \mathbf{x}^{(t+1)}$:

$$f\left(\mathbf{x}^{(t+1)}\right) \leq f\left(\mathbf{x}^{(t)}\right) + \nabla f\left(\mathbf{x}^{(t)}\right)^T\left(\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\right) + \frac{\beta}{2}\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|_2^2.$$

2. Substitute $\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)} = -\frac{1}{\beta}\nabla f\left(\mathbf{x}^{(t)}\right)$:

$$\begin{aligned} f\left(\mathbf{x}^{(t+1)}\right) &\leq f\left(\mathbf{x}^{(t)}\right) - \frac{1}{\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2 + \frac{\beta}{2}\|\frac{1}{\beta}\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2 \\ &= f\left(\mathbf{x}^{(t)}\right) - \frac{1}{2\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2. \end{aligned}$$

3. Rearranging: $f\left(\mathbf{x}^{(t)}\right) - f\left(\mathbf{x}^{(t+1)}\right) \geq \frac{1}{2\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2$. □

Telescoping Sum Proof

We have that $\frac{1}{2\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2 \leq f\left(\mathbf{x}^{(t)}\right) - f\left(\mathbf{x}^{(t+1)}\right)$. So:

$$\sum_{t=0}^{T-1} \frac{1}{2\beta}\|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2 \leq f\left(\mathbf{x}^{(0)}\right) - f\left(\mathbf{x}^{(t)}\right)$$

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2 \leq \frac{2\beta}{T}\left(f\left(\mathbf{x}^{(0)}\right) - f\left(\mathbf{x}^{*}\right)\right)$$

$$\min_t \|\nabla f\left(\mathbf{x}^{(t)}\right)\|_2^2 \leq \frac{2\beta}{T}\left(f\left(\mathbf{x}^{(0)}\right) - f\left(\mathbf{x}^{*}\right)\right)$$

Completing the Proof

For convex functions, we want to further prove that:

$$f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right) \leq \frac{2\beta R^2}{T}$$

Convex functions only have one stationary point: the global minimum $\mathbf{x}^*$. Ideally, we would argue that a near-stationary point is a near-minimizer. However, this isn't always the case!

Convergence Guarantee

Nevertheless, not to hard to obtain a proof from the progress condition. A concise version can be found on Page 15 in Garrigos and Gower's notes.

Theorem (GD convergence for β-smooth functions.)

Let f be a β smooth convex function and assume we have $\|\mathbf{x}^* - \mathbf{x}^{(1)}\|_2 \leq R$. If we run GD for T steps with $\eta = \frac{1}{\beta}$ we have:

$$f\left(\mathbf{x}^{(T)}\right) - f\left(\mathbf{x}^*\right) \leq \frac{2\beta R^2}{T}$$

Convergence Guarantee

Nevertheless, not to hard to obtain a proof from the progress condition. A concise version can be found on Page 15 in Garrigos and Gower's notes.

Theorem (GD convergence for β-smooth functions.)

Let f be a β smooth convex function and assume we have $\|\mathbf{x}^* - \mathbf{x}^{(1)}\|_2 \leq R$. If we run GD for T steps with $\eta = \frac{1}{\beta}$ we have:

$$f\left(\mathbf{x}^{(T)}\right) - f\left(\mathbf{x}^*\right) \leq \frac{2\beta R^2}{T}$$

Corollary: If $T = O\left(\frac{\beta R^2}{\epsilon}\right)$ we have $f\left(\mathbf{x}^{(T)}\right) - f\left(\mathbf{x}^*\right) \leq \epsilon$.

Convergence Guarantee

Nevertheless, not to hard to obtain a proof from the progress condition. A concise version can be found on Page 15 in Garrigos and Gower's notes.

Theorem (GD convergence for β-smooth functions.)

Let f be a β smooth convex function and assume we have $\|\mathbf{x}^* - \mathbf{x}^{(1)}\|_2 \leq R$. If we run GD for T steps with $\eta = \frac{1}{\beta}$ we have:

$$f\left(\mathbf{x}^{(T)}\right) - f\left(\mathbf{x}^*\right) \leq \frac{2\beta R^2}{T}$$

Corollary: If $T = O\left(\frac{\beta R^2}{\epsilon}\right)$ we have $f\left(\mathbf{x}^{(T)}\right) - f\left(\mathbf{x}^*\right) \leq \epsilon$.

Note: This is not optimal! Can be improved to depend on $O(1/T^2)$ using a technique called acceleration.

Convergence Guarantee

What if f is both β-smooth and α-strongly convex?

Theorem (GD for β-smooth, α-strongly convex.)

Let f be a β-smooth and α-strongly convex function. If we run GD for T steps (with step size $\eta = \frac{1}{\beta}$) we have:

$$\|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2 \leq e^{-T\frac{\alpha}{\beta}} \|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2$$

$\kappa = \frac{\beta}{\alpha}$ is called the "condition number" of f.

Smooth and Strongly Convex

Converting to more familiar form:

- Recall β-smoothness gives:

$$f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \leq \frac{\beta}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$$

Smooth and Strongly Convex

Converting to more familiar form:

- Recall β-smoothness gives:

$$f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \leq \frac{\beta}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$$

- Plug in $\mathbf{x} = \mathbf{x}^*$, $\mathbf{y} = \mathbf{x}^{(T)}$. Since $\nabla f(\mathbf{x}^*) = \mathbf{0}$:

$$f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) \leq \frac{\beta}{2}\|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2$$

Smooth and Strongly Convex

Converting to more familiar form:

- Recall β-smoothness gives:
$$f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \leq \frac{\beta}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$$

- Plug in $\mathbf{x} = \mathbf{x}^*$, $\mathbf{y} = \mathbf{x}^{(T)}$. Since $\nabla f(\mathbf{x}^*) = \mathbf{0}$:
$$f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) \leq \frac{\beta}{2}\|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2$$

- Rearranging:
$$\frac{2}{\beta}\left[f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*)\right] \leq \|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2$$

Smooth and Strongly Convex

Converting to more familiar form:

- Recall β-smoothness gives:
$$f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \leq \frac{\beta}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$$

- Plug in $\mathbf{x} = \mathbf{x}^*$, $\mathbf{y} = \mathbf{x}^{(T)}$. Since $\nabla f(\mathbf{x}^*) = \mathbf{0}$:
$$f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) \leq \frac{\beta}{2}\|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2$$

- Rearranging:
$$\frac{2}{\beta}\left[f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*)\right] \leq \|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2$$

- We also assume $\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2 \leq R^2$. Combined with
$$\|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2 \leq e^{-T\alpha/\beta}\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2$$
we get
$$f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) \leq \frac{\beta}{2}e^{-T\frac{\alpha}{\beta}}R^2$$

Convergence Guarantee

Theorem (GD for β-smooth, α-strongly convex.)

Let f be a β-smooth and α-strongly convex function. If we run GD for T steps (with step size $\eta = \frac{1}{\beta}$) we have:

$$f\left(\mathbf{x}^{(T)}\right) - f\left(\mathbf{x}^{*}\right) \leq \frac{\beta}{2} e^{-T\frac{\alpha}{\beta}} \cdot R^2$$

Convergence Guarantee

Theorem (GD for β-smooth, α-strongly convex.)

Let f be a β-smooth and α-strongly convex function. If we run GD for T steps (with step size $\eta = \frac{1}{\beta}$) we have:

$$f\left(\mathbf{x}^{(T)}\right) - f\left(\mathbf{x}^*\right) \leq \frac{\beta}{2} e^{-T\frac{\alpha}{\beta}} \cdot R^2$$

Corollary: If $T = O\left(\frac{\beta}{\alpha}\log(R\beta/\epsilon)\right)$ we have:

$$f\left(\mathbf{x}^{(T)}\right) - f\left(\mathbf{x}^*\right) \leq \epsilon$$

Convergence Guarantee

Theorem (GD for β-smooth, α-strongly convex.)

Let f be a β-smooth and α-strongly convex function. If we run GD for T steps (with step size $\eta = \frac{1}{\beta}$) we have:

$$f\left(\mathbf{x}^{(T)}\right) - f\left(\mathbf{x}^*\right) \leq \frac{\beta}{2} e^{-T\frac{\alpha}{\beta}} \cdot R^2$$

Corollary: If $T = O\left(\frac{\beta}{\alpha}\log(R\beta/\epsilon)\right)$ we have:

$$f\left(\mathbf{x}^{(T)}\right) - f\left(\mathbf{x}^*\right) \leq \epsilon$$

Only depend on $\log(1/\epsilon)$ instead of on $1/\epsilon$ or $1/\epsilon^2$!

Can be further improved with acceleration to $O\left(\sqrt{\frac{\beta}{\alpha}}\log(\frac{R\beta}{\epsilon})\right)$!

Smooth, Strongly Convex Optimization

Later in class, we will prove the guarantee for the special case of:

$$f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

Goal: Get some of the key ideas across, introduces important concepts like the Hessian, and show the connection between conditioning and linear algebra.

Smooth, Strongly Convex Optimization

Later in class, we will prove the guarantee for the special case of:

$$f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

Goal: Get some of the key ideas across, introduces important concepts like the Hessian, and show the connection between conditioning and linear algebra.

Rest of Today: online gradient descent and stochastic gradient descent.

Online and Stochastic Gradient Descent

- Basics of Online Learning + Optimization.

Online and Stochastic Gradient Descent

- Basics of Online Learning + Optimization.
- Introduction to Regret Analysis.

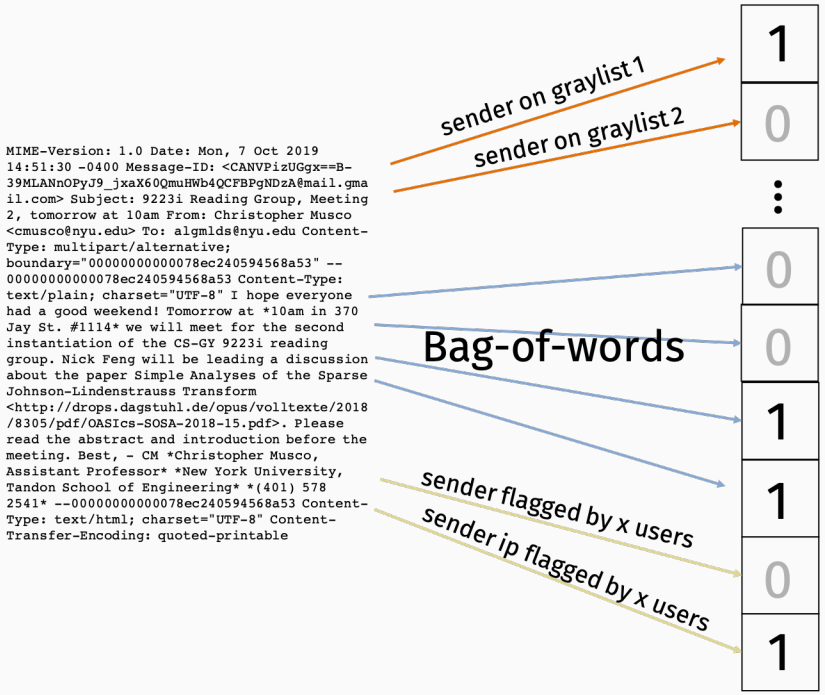
Online and Stochastic Gradient Descent

- Basics of Online Learning + Optimization.
- Introduction to Regret Analysis.
- Application to analyzing Stochastic Gradient Descent.

Original motivation for online learning: Often need to train machine learning models on constantly updating/changing data. Do not want to restart from scratch.

Example

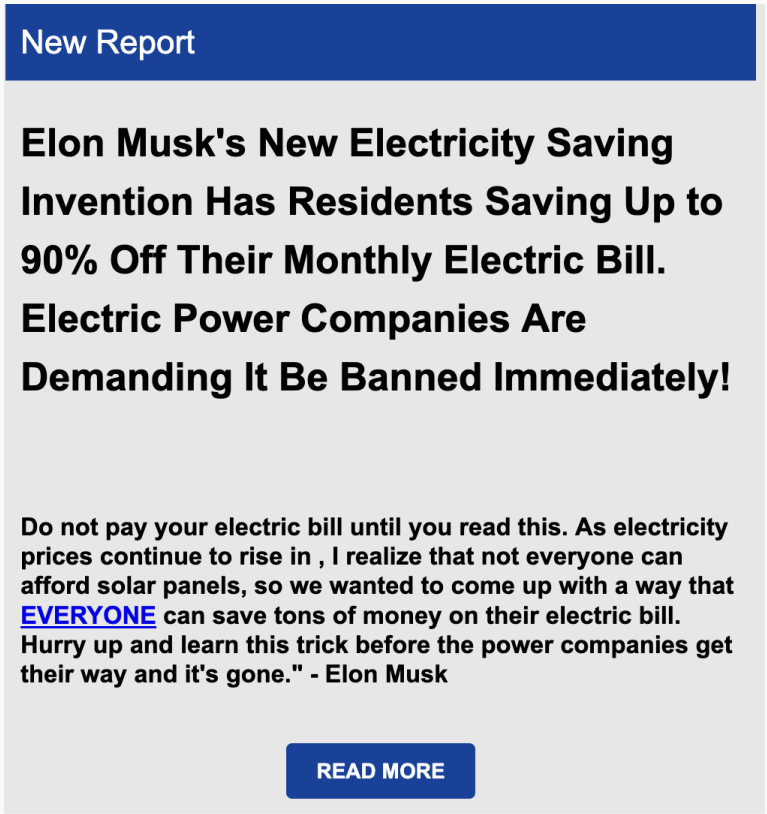
Machine learning based email spam filtering.



Markers for spam change overtime, so model might change.

Example

Machine learning based email spam filtering.



Markers for spam change overtime, so model might change.

Online Learning Framework

Choose some model $M_{\mathbf{x}}$ parameterized by parameters $\mathbf{x}$ and some loss function ℓ. At time steps $1, \ldots, T$, receive data vectors $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(T)}$.

- At each time step, we pick ("play") a parameter vector $\mathbf{x}^{(i)}$.

Online Learning Framework

Choose some model $M_{\mathbf{x}}$ parameterized by parameters $\mathbf{x}$ and some loss function ℓ. At time steps $1, \ldots, T$, receive data vectors $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(T)}$.

- At each time step, we pick ("play") a parameter vector $\mathbf{x}^{(i)}$.
- Make prediction $\tilde{y}^{(i)} = M_{\mathbf{x}^{(i)}}(\mathbf{a}_i)$.

Online Learning Framework

Choose some model $M_{\mathbf{x}}$ parameterized by parameters $\mathbf{x}$ and some loss function ℓ. At time steps $1, \ldots, T$, receive data vectors $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(T)}$.

- At each time step, we pick ("play") a parameter vector $\mathbf{x}^{(i)}$.
- Make prediction $\tilde{y}^{(i)} = M_{\mathbf{x}^{(i)}}(\mathbf{a}_i)$.
- Then told true value or label $y^{(i)}$. Possibly use this information to choose a new $\mathbf{x}^{(i+1)}$.

Online Learning Framework

Choose some model $M_{\mathbf{x}}$ parameterized by parameters $\mathbf{x}$ and some loss function ℓ. At time steps $1, \ldots, T$, receive data vectors $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(T)}$.

- At each time step, we pick ("play") a parameter vector $\mathbf{x}^{(i)}$.
- Make prediction $\tilde{y}^{(i)} = M_{\mathbf{x}^{(i)}}(\mathbf{a}_i)$.
- Then told true value or label $y^{(i)}$. Possibly use this information to choose a new $\mathbf{x}^{(i+1)}$.
- Goal is to minimize cumulative loss:

$$L = \sum_{i=1}^{T} \ell(\mathbf{x}^{(i)}, \mathbf{a}^{(i)}, y^{(i)})$$

Online Learning Framework

Choose some model $M_{\mathbf{x}}$ parameterized by parameters $\mathbf{x}$ and some loss function ℓ. At time steps $1, \ldots, T$, receive data vectors $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(T)}$.

- At each time step, we pick ("play") a parameter vector $\mathbf{x}^{(i)}$.
- Make prediction $\tilde{y}^{(i)} = M_{\mathbf{x}^{(i)}}(\mathbf{a}_i)$.
- Then told true value or label $y^{(i)}$. Possibly use this information to choose a new $\mathbf{x}^{(i+1)}$.
- Goal is to minimize cumulative loss:

$$L = \sum_{i=1}^{T} \ell(\mathbf{x}^{(i)}, \mathbf{a}^{(i)}, y^{(i)})$$

For example, for a regression problem we might use the ℓ_2 loss:

$$\ell(\mathbf{x}^{(i)}, \mathbf{a}^{(i)}, y^{(i)}) = \left(M_{\mathbf{x}^{(i)}}(\mathbf{a}_i) - y^{(i)}\right)^2.$$

For classification, we could use logistic/cross-entropy loss.

Online Optimization

Abstraction as optimization problem: Instead of a single objective function f, we have a unknown function $f_1, \ldots, f_T : \mathbb{R}^d \to \mathbb{R}$ for each time step.

- For time step $i \in 1, \ldots, T$, select vector $\mathbf{x}^{(i)}$.

Online Optimization

Abstraction as optimization problem: Instead of a single objective function f, we have a unknown function $f_1, \ldots, f_T : \mathbb{R}^d \to \mathbb{R}$ for each time step.

- For time step $i \in 1, \ldots, T$, select vector $\mathbf{x}^{(i)}$.
- Observe f_i and pay cost $f_i(\mathbf{x}^{(i)})$.

Online Optimization

Abstraction as optimization problem: Instead of a single objective function f, we have a unknown function $f_1, \ldots, f_T : \mathbb{R}^d \to \mathbb{R}$ for each time step.

- For time step $i \in 1, \ldots, T$, select vector $\mathbf{x}^{(i)}$.
- Observe f_i and pay cost $f_i(\mathbf{x}^{(i)})$.
- Goal is to minimize $\sum_{i=1}^{T} f_i(\mathbf{x}^{(i)})$.

Online Optimization

Abstraction as optimization problem: Instead of a single objective function f, we have a unknown function $f_1, \ldots, f_T : \mathbb{R}^d \to \mathbb{R}$ for each time step.

- For time step $i \in 1, \ldots, T$, select vector $\mathbf{x}^{(i)}$.
- Observe f_i and pay cost $f_i(\mathbf{x}^{(i)})$.
- Goal is to minimize $\sum_{i=1}^{T} f_i(\mathbf{x}^{(i)})$.

We make no assumptions that $f_1, \ldots, f_T$ are related to each other at all!

Regret Bound

In offline optimization, we wanted to find $\hat{\mathbf{x}}$ satisfying $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x}} f(\mathbf{x}) + \epsilon$. We ask for something similar here.

Objective: Choose $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}$ so that:

$$\sum_{i=1}^{T} f_i(\mathbf{x}^{(i)}) \leq \left[\min_{\mathbf{x}} \sum_{i=1}^{T} f_i(\mathbf{x}) \right] + \epsilon.$$

Regret Bound

In offline optimization, we wanted to find $\hat{\mathbf{x}}$ satisfying $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x}} f(\mathbf{x}) + \epsilon$. We ask for something similar here.

Objective: Choose $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}$ so that:

$$\sum_{i=1}^{T} f_i(\mathbf{x}^{(i)}) \leq \left[\min_{\mathbf{x}} \sum_{i=1}^{T} f_i(\mathbf{x}) \right] + \epsilon.$$

Here ϵ is called the **regret** of our solution sequence $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}$.

Regret Bound

In offline optimization, we wanted to find $\hat{\mathbf{x}}$ satisfying $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x}} f(\mathbf{x}) + \epsilon$. We ask for something similar here.

Objective: Choose $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}$ so that:

$$\sum_{i=1}^{T} f_i(\mathbf{x}^{(i)}) \leq \left[\min_{\mathbf{x}} \sum_{i=1}^{T} f_i(\mathbf{x}) \right] + \epsilon.$$

Here ϵ is called the **regret** of our solution sequence $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}$.

We typically want ϵ to grow sublinearly in T.

Regret Bound

Regret compares to the best fixed solution in hindsight.

$$\sum_{i=1}^{T} f_i\left(\mathbf{x}^{(i)}\right) \leq \left[\min_{\mathbf{x}} \sum_{i=1}^{T} f_i\left(\mathbf{x}\right)\right] + \epsilon.$$

Regret Bound

Regret compares to the best fixed solution in hindsight.

$$\sum_{i=1}^{T} f_i\left(\mathbf{x}^{(i)}\right) \leq \left[\min_{\mathbf{x}} \sum_{i=1}^{T} f_i\left(\mathbf{x}\right)\right] + \epsilon.$$

Note: it is very possible that no fixed $\mathbf{x}$ is good for all f_i. Could we hope for something stronger?

Regret Bound

Regret compares to the best fixed solution in hindsight.

$$\sum_{i=1}^{T} f_i\left(\mathbf{x}^{(i)}\right) \leq \left[\min_{\mathbf{x}} \sum_{i=1}^{T} f_i\left(\mathbf{x}\right)\right] + \epsilon.$$

Note: it is very possible that no fixed $\mathbf{x}$ is good for all f_i. Could we hope for something stronger?

Exercise: Argue that the following is impossible to achieve:

$$\sum_{i=1}^{T} f_i\left(\mathbf{x}^{(i)}\right) \leq \left[\sum_{i=1}^{T} \min_{\mathbf{x}} f_i\left(\mathbf{x}\right)\right] + \epsilon.$$

Hard Example for Online Optimization

Convex functions:

$$f_1(x) = |x - h_1|$$

$$\vdots$$

$$f_T(x) = |x - h_T|$$

where $h_1, \ldots, h_T$ are i.i.d. uniform $\{0, 1\}$.

$$\mathrm{RHS} = \sum_{i=1}^{T} \min_{\mathbf{x_i}} f_i(\mathbf{x_i}) = 0 \qquad \text{by setting } x_i = h_i\,.$$

Hard Example for Online Optimization

Convex functions:

$$f_1(x) = |x - h_1|$$

$$\vdots$$

$$f_T(x) = |x - h_T|$$

where $h_1, \ldots, h_T$ are i.i.d. uniform $\{0, 1\}$.

$$\text{RHS} = \sum_{i=1}^{T} \min_{\mathbf{x_i}} f_i(\mathbf{x_i}) = 0 \quad \text{by setting } x_i = h_i\,.$$

$$\text{LHS} = \mathbb{E}\left[\sum_{i=1}^{T} f_i(\mathbf{x}^{(i)})\right] = \sum_{i=1}^{T} \frac{1}{2}\|0 - \mathbf{x}^{(i)}\| + \frac{1}{2}\|1 - \mathbf{x}^{(i)}\| \geq \frac{T}{2}$$

Regret Bounds

$$\sum_{i=1}^{T} f_i\left(\mathbf{x}^{(i)}\right) \leq \left[\min_{\mathbf{x}} \sum_{i=1}^{T} f_i\left(\mathbf{x}\right)\right] + \epsilon.$$

Beautiful balance:

- Either $f_1, \ldots, f_T$ are similar or changing slowly, so we can learn/predict f_i from earlier functions.

Regret Bounds

$$\sum_{i=1}^{T} f_i\left(\mathbf{x}^{(i)}\right) \leq \left[\min_{\mathbf{x}} \sum_{i=1}^{T} f_i\left(\mathbf{x}\right)\right] + \epsilon.$$

Beautiful balance:

- Either $f_1, \ldots, f_T$ are similar or changing slowly, so we can learn/predict f_i from earlier functions.
- Or $f_1, \ldots, f_T$ are very different, in which case $\min_{\mathbf{x}} \sum_{i=1}^{T} f_i\left(\mathbf{x}\right)$ is large, so the regret bound is easy to achieve.

Regret Bounds

$$\sum_{i=1}^{T} f_i\left(\mathbf{x}^{(i)}\right) \leq \left[\min_{\mathbf{x}} \sum_{i=1}^{T} f_i\left(\mathbf{x}\right)\right] + \epsilon.$$

Beautiful balance:

- Either $f_1, \ldots, f_T$ are similar or changing slowly, so we can learn/predict f_i from earlier functions.
- Or $f_1, \ldots, f_T$ are very different, in which case $\min_{\mathbf{x}} \sum_{i=1}^{T} f_i\left(\mathbf{x}\right)$ is large, so the regret bound is easy to achieve.
- Or we live somewhere in the middle.

Online Gradient Descent (OGD)

Online Gradient Descent:

- Choose $\mathbf{x}^{(1)}$ and η.

Online Gradient Descent (OGD)

Online Gradient Descent:

- Choose $\mathbf{x}^{(1)}$ and η.
- For $i = 1, \ldots, T$:
  - Play $\mathbf{x}^{(i)}$.
  - Observe f_i and incur cost $f_i\left(\mathbf{x}^{(i)}\right)$.
  - $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f_i\left(\mathbf{x}^{(i)}\right)$

Online Gradient Descent (OGD)

Online Gradient Descent:

- Choose $\mathbf{x}^{(1)}$ and η.
- For $i = 1, \ldots, T$:
  - Play $\mathbf{x}^{(i)}$.
  - Observe f_i and incur cost $f_i\left(\mathbf{x}^{(i)}\right)$.
  - $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f_i\left(\mathbf{x}^{(i)}\right)$

If $f_1, \ldots, f_T = f$ are all the same, this reduces to regular gradient descent.

Online Gradient Descent (OGD)

Online Gradient Descent:

- Choose $\mathbf{x}^{(1)}$ and η.
- For $i = 1, \ldots, T$:
  - Play $\mathbf{x}^{(i)}$.
  - Observe f_i and incur cost $f_i\left(\mathbf{x}^{(i)}\right)$.
  - $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f_i\left(\mathbf{x}^{(i)}\right)$

If $f_1, \ldots, f_T = f$ are all the same, this reduces to regular gradient descent.

Only requires calculating gradient at one data point in each step

Online Gradient Descent (OGD)

$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{i=1}^{T} f_i(\mathbf{x})$ (the offline optimum)

Assume:

- $f_1, \ldots, f_T$ are all convex.
- Each is G-Lipschitz: for all $\mathbf{x}$, i, $\|\nabla f_i(\mathbf{x})\|_2 \leq G$.
- Starting radius: $\|\mathbf{x}^* - \mathbf{x}^{(1)}\|_2 \leq R$.

Online Gradient Descent (OGD)

$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{i=1}^{T} f_i(\mathbf{x})$ (the offline optimum)

Assume:

- $f_1, \ldots, f_T$ are all convex.
- Each is G-Lipschitz: for all $\mathbf{x}$, i, $\|\nabla f_i(\mathbf{x})\|_2 \leq G$.
- Starting radius: $\|\mathbf{x}^* - \mathbf{x}^{(1)}\|_2 \leq R$.

Online Gradient Descent:

- Choose $\mathbf{x}^{(1)}$ and $\eta = \frac{R}{G\sqrt{T}}$.

Online Gradient Descent (OGD)

$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{i=1}^{T} f_i(\mathbf{x})$ (the offline optimum)

Assume:

- $f_1, \ldots, f_T$ are all convex.
- Each is G-Lipschitz: for all $\mathbf{x}$, i, $\|\nabla f_i(\mathbf{x})\|_2 \leq G$.
- Starting radius: $\|\mathbf{x}^* - \mathbf{x}^{(1)}\|_2 \leq R$.

Online Gradient Descent:

- Choose $\mathbf{x}^{(1)}$ and $\eta = \frac{R}{G\sqrt{T}}$.
- For $i = 1, \ldots, T$: play $\mathbf{x}^{(i)}$, observe f_i, update

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f_i(\mathbf{x}^{(i)}).$$

Online Gradient Descent Analysis

Theorem (OGD Regret Bound)

If the conditions of the previous slide hold, then after T steps,

$$\epsilon = \left[\sum_{i=1}^{T} f_i\left(\mathbf{x}^{(i)}\right)\right] - \left[\sum_{i=1}^{T} f_i\left(\mathbf{x}^{*}\right)\right] \leq RG\sqrt{T}.$$

Online Gradient Descent Analysis

Theorem (OGD Regret Bound)

If the conditions of the previous slide hold, then after T steps,

$$\epsilon = \left[\sum_{i=1}^{T} f_i\left(\mathbf{x}^{(i)}\right)\right] - \left[\sum_{i=1}^{T} f_i\left(\mathbf{x}^{*}\right)\right] \leq RG\sqrt{T}.$$

- Average regret is bounded by $\frac{\epsilon}{T} \leq \frac{RG}{\sqrt{T}} \to 0$ as $T \to \infty$.

Online Gradient Descent Analysis

Theorem (OGD Regret Bound)

If the conditions of the previous slide hold, then after T steps,

$$\epsilon = \left[\sum_{i=1}^{T} f_i\left(\mathbf{x}^{(i)}\right)\right] - \left[\sum_{i=1}^{T} f_i\left(\mathbf{x}^{*}\right)\right] \leq RG\sqrt{T}.$$

- Average regret is bounded by $\frac{\epsilon}{T} \leq \frac{RG}{\sqrt{T}} \to 0$ as $T \to \infty$.
- All this with no assumptions on how $f_1, \ldots, f_T$ relate to each other — they could even be chosen **adversarially**.

Online Gradient Descent Analysis

Theorem (OGD Regret Bound)

If the conditions of the previous slide hold, then after T steps, $\epsilon = \left[\sum_{i=1}^{T} f_i(\mathbf{x}^{(i)})\right] - \left[\sum_{i=1}^{T} f_i(\mathbf{x}^*)\right] \leq RG\sqrt{T}$.

Claim 1: For all $i = 1, \ldots, T$,

$$f_i(\mathbf{x}^{(i)}) - f_i(\mathbf{x}^*) \leq \frac{\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(i+1)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$$

(We proved this by just plugging in $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f_i(\mathbf{x}^{(i)})$)

Online Gradient Descent Analysis

Theorem (OGD Regret Bound)

After T steps, $\epsilon = \left[\sum_{i=1}^{T} f_i(\mathbf{x}^{(i)})\right] - \left[\sum_{i=1}^{T} f_i(\mathbf{x}^*)\right] \leq RG\sqrt{T}$.

Claim 1: For all $i = 1, \ldots, T$,

$$f_i(\mathbf{x}^{(i)}) - f_i(\mathbf{x}^*) \leq \frac{\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(i+1)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$$

Telescoping Sum:

$$\sum_{i=1}^{T} \left[f_i(\mathbf{x}^{(i)}) - f_i(\mathbf{x}^*)\right] \leq \frac{\|\mathbf{x}^{(1)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{T\eta G^2}{2}$$

Online Gradient Descent Analysis

Theorem (OGD Regret Bound)

After T steps, $\epsilon = \left[\sum_{i=1}^{T} f_i(\mathbf{x}^{(i)})\right] - \left[\sum_{i=1}^{T} f_i(\mathbf{x}^*)\right] \leq RG\sqrt{T}$.

Claim 1: For all $i = 1, \ldots, T$,

$$f_i(\mathbf{x}^{(i)}) - f_i(\mathbf{x}^*) \leq \frac{\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(i+1)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$$

Telescoping Sum:

$$\begin{aligned}\sum_{i=1}^{T}\left[f_i(\mathbf{x}^{(i)}) - f_i(\mathbf{x}^*)\right] &\leq \frac{\|\mathbf{x}^{(1)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{T\eta G^2}{2} \\ &\leq \frac{R^2}{2\eta} + \frac{T\eta G^2}{2}\end{aligned}$$

Online Gradient Descent Analysis

Theorem (OGD Regret Bound)

After T steps, $\epsilon = \left[\sum_{i=1}^{T} f_i(\mathbf{x}^{(i)})\right] - \left[\sum_{i=1}^{T} f_i(\mathbf{x}^*)\right] \leq RG\sqrt{T}$.

Claim 1: For all $i = 1, \ldots, T$,

$$f_i(\mathbf{x}^{(i)}) - f_i(\mathbf{x}^*) \leq \frac{\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(i+1)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$$

Telescoping Sum:

$$\begin{aligned}
\sum_{i=1}^{T} \left[f_i(\mathbf{x}^{(i)}) - f_i(\mathbf{x}^*)\right] &\leq \frac{\|\mathbf{x}^{(1)} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^{(T)} - \mathbf{x}^*\|_2^2}{2\eta} + \frac{T\eta G^2}{2} \\
&\leq \frac{R^2}{2\eta} + \frac{T\eta G^2}{2} \\
&\leq RG\sqrt{T} \quad \left(\text{recall } \eta = \frac{R}{G\sqrt{T}}\right)
\end{aligned}$$

Stochastic Gradient Descent (SGD)

Efficient offline optimization method for functions f with finite sum structure:

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}).$$

Goal: find $\hat{\mathbf{x}}$ such that $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

Stochastic Gradient Descent (SGD)

Efficient offline optimization method for functions f with finite sum structure:

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}).$$

Goal: find $\hat{\mathbf{x}}$ such that $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

- The most widely used optimization algorithm in modern machine learning.

Stochastic Gradient Descent (SGD)

Efficient offline optimization method for functions f with finite sum structure:

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}).$$

Goal: find $\hat{\mathbf{x}}$ such that $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

- The most widely used optimization algorithm in modern machine learning.
- Easily analyzed as a special case of online gradient descent!

Stochastic Gradient Descent

Recall the machine learning setup. In empirical risk minimization, we can typically write:

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$$

where f_i is the loss function for a particular data example $(\mathbf{a}^{(i)}, y^{(i)})$

.

Stochastic Gradient Descent

Recall the machine learning setup. In empirical risk minimization, we can typically write:

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$$

where f_i is the loss function for a particular data example $(\mathbf{a}^{(i)}, y^{(i)})$.

Example: Least Squares Linear Regression.

$$f(\mathbf{x}) = \sum_{i=1}^{n} (\mathbf{x}^T \mathbf{a}^{(i)} - y^{(i)})^2$$

By linearity, $\nabla f(\mathbf{x}) = \sum_{i=1}^{n} \nabla f_i(\mathbf{x}) = \sum_{i=1}^{n} 2(\mathbf{x}^T \mathbf{a}^{(i)} - y^{(i)}) \cdot \mathbf{a}^{(i)}$.

Stochastic Gradient Descent

Recall the machine learning setup. In empirical risk minimization, we can typically write:

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$$

where f_i is the loss function for a particular data example $(\mathbf{a}^{(i)}, y^{(i)})$.

Example: Least Squares Linear Regression.

$$f(\mathbf{x}) = \sum_{i=1}^{n} (\mathbf{x}^T \mathbf{a}^{(i)} - y^{(i)})^2$$

By linearity, $\nabla f(\mathbf{x}) = \sum_{i=1}^{n} \nabla f_i(\mathbf{x}) = \sum_{i=1}^{n} 2(\mathbf{x}^T \mathbf{a}^{(i)} - y^{(i)}) \cdot \mathbf{a}^{(i)}$.

Key Idea: Compute only one term in the sum, chosen at random

Stochastic Gradient Descent

Key Idea: Use a random approximate gradient in place of the actual gradient

Pick random $j \in 1, \ldots, n$ and update $\mathbf{x}$ using $\nabla f_j(\mathbf{x})$:

$$\mathbb{E}\left[\nabla f_j(\mathbf{x})\right] = \frac{1}{n}\nabla f(\mathbf{x}).$$

Stochastic Gradient Descent

Key Idea: Use a random approximate gradient in place of the actual gradient

Pick random $j \in 1, \ldots, n$ and update $\mathbf{x}$ using $\nabla f_j(\mathbf{x})$:

$$\mathbb{E}\left[\nabla f_j(\mathbf{x})\right] = \frac{1}{n}\nabla f(\mathbf{x}).$$

- $n \cdot \nabla f_j(\mathbf{x})$ is an unbiased estimate for $\nabla f(\mathbf{x})$

Stochastic Gradient Descent

Key Idea: Use a random approximate gradient in place of the actual gradient

Pick random $j \in 1, \ldots, n$ and update $\mathbf{x}$ using $\nabla f_j(\mathbf{x})$:

$$\mathbb{E}\left[\nabla f_j(\mathbf{x})\right] = \frac{1}{n}\nabla f(\mathbf{x}).$$

- $n \cdot \nabla f_j(\mathbf{x})$ is an unbiased estimate for $\nabla f(\mathbf{x})$
- Computable in $O(d)$ time, factor of n faster than computing the entire gradient!

Stochastic Gradient Descent

Key Idea: Use a random approximate gradient in place of the actual gradient

Pick random $j \in 1, \ldots, n$ and update $\mathbf{x}$ using $\nabla f_j(\mathbf{x})$:

$$\mathbb{E}\left[\nabla f_j(\mathbf{x})\right] = \frac{1}{n}\nabla f(\mathbf{x}).$$

- $n \cdot \nabla f_j(\mathbf{x})$ is an unbiased estimate for $\nabla f(\mathbf{x})$
- Computable in $O(d)$ time, factor of n faster than computing the entire gradient!

Trade slower convergence for cheaper iterations.

Stochastic Gradient Descent

Stochastic first-order oracle for $f(\mathbf{x}) = \sum_{j=1}^{n} f_j(\mathbf{x})$:

- **Function Query:** For any chosen $j, \mathbf{x}$, return $f_j(\mathbf{x})$
- **Gradient Query:** For any chosen $j, \mathbf{x}$, return $\nabla f_j(\mathbf{x})$

Stochastic Gradient Descent

Stochastic first-order oracle for $f(\mathbf{x}) = \sum_{j=1}^{n} f_j(\mathbf{x})$:

- **Function Query:** For any chosen $j, \mathbf{x}$, return $f_j(\mathbf{x})$
- **Gradient Query:** For any chosen $j, \mathbf{x}$, return $\nabla f_j(\mathbf{x})$

Stochastic Gradient Descent:

- Choose starting vector $\mathbf{x}^{(1)}$, step size η.

Stochastic Gradient Descent

Stochastic first-order oracle for $f(\mathbf{x}) = \sum_{j=1}^{n} f_j(\mathbf{x})$:

- **Function Query:** For any chosen $j, \mathbf{x}$, return $f_j(\mathbf{x})$
- **Gradient Query:** For any chosen $j, \mathbf{x}$, return $\nabla f_j(\mathbf{x})$

Stochastic Gradient Descent:

- Choose starting vector $\mathbf{x}^{(1)}$, step size η.
- For $i = 1, \ldots, T$: pick random $j_i \in \{1, \ldots, n\}$, update

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f_{j_i}(\mathbf{x}^{(i)}).$$

Stochastic Gradient Descent

Stochastic first-order oracle for $f(\mathbf{x}) = \sum_{j=1}^{n} f_j(\mathbf{x})$:

- **Function Query:** For any chosen $j, \mathbf{x}$, return $f_j(\mathbf{x})$
- **Gradient Query:** For any chosen $j, \mathbf{x}$, return $\nabla f_j(\mathbf{x})$

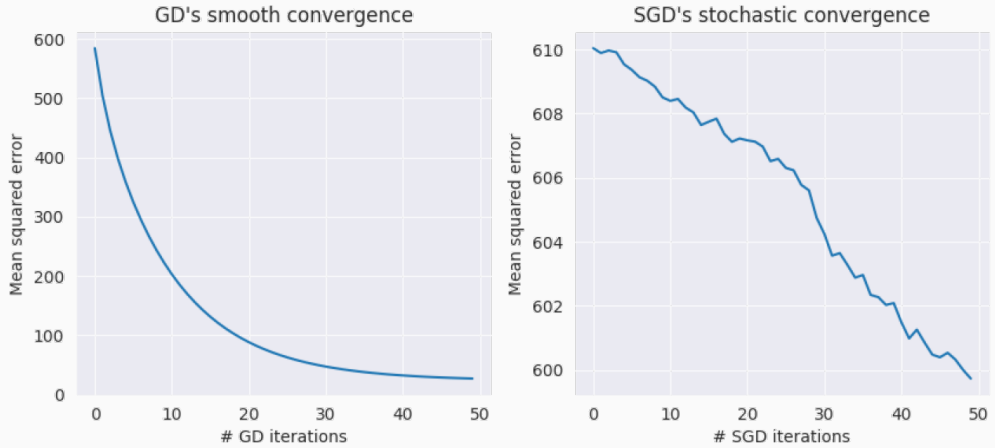
Stochastic Gradient Descent:

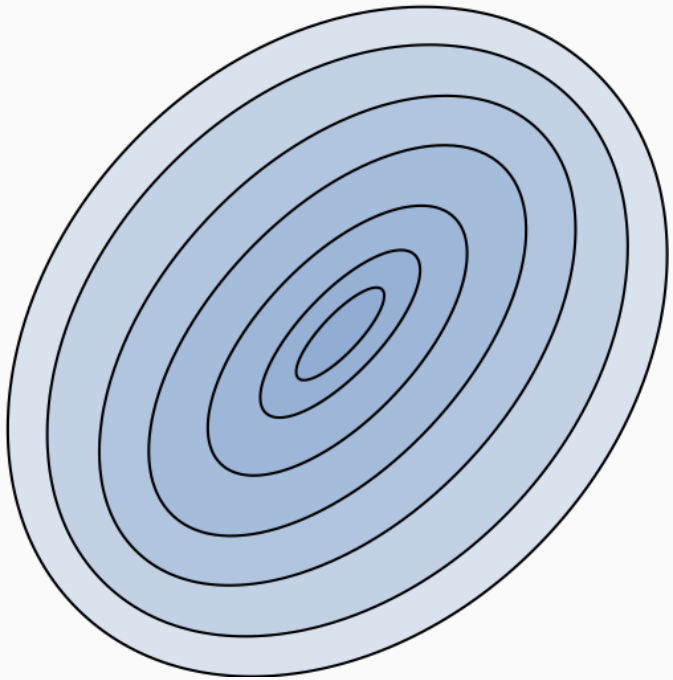
- Choose starting vector $\mathbf{x}^{(1)}$, step size η.
- For $i = 1, \ldots, T$: pick random $j_i \in \{1, \ldots, n\}$, update

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f_{j_i}(\mathbf{x}^{(i)}).$$

- Return $\hat{\mathbf{x}} = \frac{1}{T} \sum_{i=1}^{T} \mathbf{x}^{(i)}$.

visualizing SGD





Stochastic Gradient Descent

Assume:

- Finite sum structure: $f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$, with $f_1, \ldots, f_n$ all convex.

Stochastic Gradient Descent

Assume:

- Finite sum structure: $f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$, with $f_1, \ldots, f_n$ all convex.
- Lipschitz functions: for all $\mathbf{x}$, j, $\|\nabla f_j(\mathbf{x})\|_2 \leq \frac{G'}{n}$. (What does this imply about the Lipschitz constant of f?)

Stochastic Gradient Descent

Assume:

- Finite sum structure: $f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$, with $f_1, \ldots, f_n$ all convex.
- Lipschitz functions: for all $\mathbf{x}$, j, $\|\nabla f_j(\mathbf{x})\|_2 \leq \frac{G'}{n}$. (What does this imply about the Lipschitz constant of f?)
- Starting radius: $\|\mathbf{x}^* - \mathbf{x}^{(1)}\|_2 \leq R$.

Stochastic Gradient Descent:

- Choose $\mathbf{x}^{(1)}$, steps T, step size $\eta = \frac{R}{G'\sqrt{T}}$.

Stochastic Gradient Descent

Assume:

- Finite sum structure: $f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$, with $f_1, \ldots, f_n$ all convex.
- Lipschitz functions: for all $\mathbf{x}$, j, $\|\nabla f_j(\mathbf{x})\|_2 \leq \frac{G'}{n}$. (What does this imply about the Lipschitz constant of f?)
- Starting radius: $\|\mathbf{x}^* - \mathbf{x}^{(1)}\|_2 \leq R$.

Stochastic Gradient Descent:

- Choose $\mathbf{x}^{(1)}$, steps T, step size $\eta = \frac{R}{G'\sqrt{T}}$.
- For $i = 1, \ldots, T$: pick random $j_i \in \{1, \ldots, n\}$, update

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f_{j_i}(\mathbf{x}^{(i)}).$$

Stochastic Gradient Descent

Assume:

- Finite sum structure: $f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$, with $f_1, \ldots, f_n$ all convex.
- Lipschitz functions: for all $\mathbf{x}$, j, $\|\nabla f_j(\mathbf{x})\|_2 \leq \frac{G'}{n}$. (What does this imply about the Lipschitz constant of f?)
- Starting radius: $\|\mathbf{x}^* - \mathbf{x}^{(1)}\|_2 \leq R$.

Stochastic Gradient Descent:

- Choose $\mathbf{x}^{(1)}$, steps T, step size $\eta = \frac{R}{G'\sqrt{T}}$.
- For $i = 1, \ldots, T$: pick random $j_i \in \{1, \ldots, n\}$, update

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \eta \nabla f_{j_i}(\mathbf{x}^{(i)}).$$

- Return $\hat{\mathbf{x}} = \frac{1}{T}\sum_{i=1}^{T} \mathbf{x}^{(i)}$.

View as online gradient descent run on function sequence $f_{j_1}, \ldots, f_{j_T}$.

Stochastic Gradient Descent Bound

Claim (SGD Convergence)

After $T = \frac{R^2 G'^2}{\epsilon^2}$ *iterations:*

$$\mathbb{E}\left[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)\right] \leq \epsilon.$$

Stochastic Gradient Descent Bound

Claim (SGD Convergence)

After $T = \frac{R^2 G'^2}{\epsilon^2}$ *iterations:*

$$\mathbb{E}\left[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)\right] \leq \epsilon.$$

Will prove using:

1. Black-box result for online gradient descent (already proven).

Stochastic Gradient Descent Bound

Claim (SGD Convergence)

After $T = \frac{R^2 G'^2}{\epsilon^2}$ *iterations:*

$$\mathbb{E}\left[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)\right] \leq \epsilon.$$

Will prove using:

1. Black-box result for online gradient descent (already proven).
2. The fact that $n \cdot \mathbb{E}[f_{j_i}\left(\mathbf{x}^{(i)}\right)] = f\left(\mathbf{x}^{(i)}\right)$.

Stochastic Gradient Descent Bound

Claim (SGD Convergence)

After $T = \frac{R^2 G'^2}{\epsilon^2}$ *iterations:*

$$\mathbb{E}\left[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)\right] \leq \epsilon.$$

Will prove using:

1. Black-box result for online gradient descent (already proven).
2. The fact that $n \cdot \mathbb{E}[f_{j_i}\left(\mathbf{x}^{(i)}\right)] = f\left(\mathbf{x}^{(i)}\right)$.
3. Jensen's inequality.

Jensen's Inequality

For a convex function f and points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(t)}$:

$$f\left(\frac{1}{t}\sum_{k=1}^{t}\mathbf{x}^{(k)}\right) \leq \frac{1}{t}\sum_{k=1}^{t} f\left(\mathbf{x}^{(k)}\right)$$

Jensen's Inequality

For a convex function f and points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(t)}$:

$$f\left(\frac{1}{t}\sum_{k=1}^{t}\mathbf{x}^{(k)}\right) \leq \frac{1}{t}\sum_{k=1}^{t} f\left(\mathbf{x}^{(k)}\right)$$

In other words: the function of the average is at most the average of the function.

Jensen's Inequality

For a convex function f and points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(t)}$:

$$f\left(\frac{1}{t}\sum_{k=1}^{t}\mathbf{x}^{(k)}\right) \leq \frac{1}{t}\sum_{k=1}^{t} f\left(\mathbf{x}^{(k)}\right)$$

In other words: the function of the average is at most the average of the function.

This is a generalization of

$$f\left(\frac{1}{2}x + \frac{1}{2}y\right) \leq \frac{1}{2}f(x) + \frac{1}{2}f(y)$$

Stochastic Gradient Descent Analysis

Claim (SGD Convergence)

Recall, $\hat{\mathbf{x}} = \frac{1}{T}\sum_{i=1}^{T}\mathbf{x}^{(i)}$. *After* $T = \frac{R^2 G'^2}{\epsilon^2}$ *iterations:*

$$\mathbb{E}\left[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)\right] \leq \epsilon.$$

Claim 1: Cost of average iterate is bounded by average cost of the iterates.

$$f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right) \leq \frac{1}{T}\sum_{i=1}^{T}\left[f\left(\mathbf{x}^{(i)}\right) - f\left(\mathbf{x}^*\right)\right]$$

Prove using Jensen's Inequality:

$$f\left(\frac{1}{T}\sum_{i=1}^{T}\mathbf{x}^{(i)}\right) - \frac{1}{T}\sum_{i=1}^{T}f\left(\mathbf{x}^*\right) \leq \frac{1}{T}\sum_{i=1}^{T}f\left(\mathbf{x}^{(i)}\right) - \frac{1}{T}\sum_{i=1}^{T}f\left(\mathbf{x}^*\right)$$

Stochastic Gradient Descent Analysis

Claim (SGD Convergence)

Recall, $\hat{\mathbf{x}} = \frac{1}{T}\sum_{i=1}^{T}\mathbf{x}^{(i)}$. *After* $T = \frac{R^2 G'^2}{\epsilon^2}$ *iterations:*

$$\mathbb{E}\left[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)\right] \leq \epsilon.$$

$$\mathbb{E}[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)] \leq \frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\left[f\left(\mathbf{x}^{(i)}\right) - f\left(\mathbf{x}^*\right)\right]$$

Stochastic Gradient Descent Analysis

Claim (SGD Convergence)

Recall, $\hat{\mathbf{x}} = \frac{1}{T}\sum_{i=1}^{T} \mathbf{x}^{(i)}$. *After* $T = \frac{R^2 G'^2}{\epsilon^2}$ *iterations:*

$$\mathbb{E}\left[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)\right] \leq \epsilon.$$

$$\mathbb{E}[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)] \leq \frac{1}{T}\sum_{i=1}^{T} \mathbb{E}\left[f\left(\mathbf{x}^{(i)}\right) - f\left(\mathbf{x}^*\right)\right]$$

$$= \frac{1}{T}\sum_{i=1}^{T} n\mathbb{E}\left[f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^*)\right] \quad (\mathbb{E}[f\left(\mathbf{x}^{(i)}\right)] = n\mathbb{E}[f_{j_i}(\mathbf{x}^{(i)})])$$

Stochastic Gradient Descent Analysis

Claim (SGD Convergence)

Recall, $\hat{\mathbf{x}} = \frac{1}{T}\sum_{i=1}^{T} \mathbf{x}^{(i)}$. *After* $T = \frac{R^2 G'^2}{\epsilon^2}$ *iterations:*

$$\mathbb{E}\left[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)\right] \leq \epsilon.$$

$$\begin{aligned}
\mathbb{E}[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)] &\leq \frac{1}{T}\sum_{i=1}^{T} \mathbb{E}\left[f\left(\mathbf{x}^{(i)}\right) - f\left(\mathbf{x}^*\right)\right] \\
&= \frac{1}{T}\sum_{i=1}^{T} n\mathbb{E}\left[f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^*)\right] \quad (\mathbb{E}[f\left(\mathbf{x}^{(i)}\right)] = n\mathbb{E}[f_{j_i}(\mathbf{x}^{(i)})]) \\
&= \frac{n}{T}\,\mathbb{E}\left[\sum_{i=1}^{T} f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^*)\right]
\end{aligned}$$

Stochastic Gradient Descent Analysis

Claim (SGD Convergence)

Recall, $\hat{\mathbf{x}} = \frac{1}{T}\sum_{i=1}^{T}\mathbf{x}^{(i)}$. *After* $T = \frac{R^2 G'^2}{\epsilon^2}$ *iterations:*

$$\mathbb{E}\left[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)\right] \leq \epsilon.$$

$$\begin{aligned}
\mathbb{E}[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)] &\leq \frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\left[f\left(\mathbf{x}^{(i)}\right) - f\left(\mathbf{x}^*\right)\right] \\
&= \frac{1}{T}\sum_{i=1}^{T} n\mathbb{E}\left[f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^*)\right] \quad (\mathbb{E}[f\left(\mathbf{x}^{(i)}\right)] = n\mathbb{E}[f_{j_i}(\mathbf{x}^{(i)})]) \\
&= \frac{n}{T}\,\mathbb{E}\left[\sum_{i=1}^{T} f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^*)\right] \\
&\leq \frac{n}{T}\,\mathbb{E}\left[\sum_{i=1}^{T} f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^{offline})\right]
\end{aligned}$$

Stochastic Gradient Descent Analysis

Claim (SGD Convergence)

Recall, $\hat{\mathbf{x}} = \frac{1}{T}\sum_{i=1}^{T}\mathbf{x}^{(i)}$. *After* $T = \frac{R^2 G'^2}{\epsilon^2}$ *iterations:*

$$\mathbb{E}\left[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)\right] \leq \epsilon.$$

$$\begin{aligned}
\mathbb{E}[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)] &\leq \frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\left[f\left(\mathbf{x}^{(i)}\right) - f\left(\mathbf{x}^*\right)\right] \\
&= \frac{1}{T}\sum_{i=1}^{T} n\mathbb{E}\left[f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^*)\right] \quad (\mathbb{E}[f\left(\mathbf{x}^{(i)}\right)] = n\mathbb{E}[f_{j_i}(\mathbf{x}^{(i)})]) \\
&= \frac{n}{T}\,\mathbb{E}\left[\sum_{i=1}^{T} f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^*)\right] \\
&\leq \frac{n}{T}\,\mathbb{E}\left[\sum_{i=1}^{T} f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^{offline})\right]
\end{aligned}$$

Stochastic Gradient Descent Analysis

Claim (SGD Convergence)

Recall, $\hat{\mathbf{x}} = \frac{1}{T}\sum_{i=1}^{T} \mathbf{x}^{(i)}$. *After* $T = \frac{R^2 G'^2}{\epsilon^2}$ *iterations:*

$$\mathbb{E}\left[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)\right] \leq \epsilon.$$

$$\begin{aligned}
\mathbb{E}[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)] &\leq \frac{1}{T}\sum_{i=1}^{T} \mathbb{E}\left[f\left(\mathbf{x}^{(i)}\right) - f\left(\mathbf{x}^*\right)\right] \\
&= \frac{1}{T}\sum_{i=1}^{T} n\mathbb{E}\left[f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^*)\right] \quad (\mathbb{E}[f\left(\mathbf{x}^{(i)}\right)] = n\mathbb{E}[f_{j_i}(\mathbf{x}^{(i)})]) \\
&= \frac{n}{T}\,\mathbb{E}\left[\sum_{i=1}^{T} f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^*)\right] \\
&\leq \frac{n}{T}\,\mathbb{E}\left[\sum_{i=1}^{T} f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^{offline})\right]
\end{aligned}$$

where $\mathbf{x}^{offline} = \arg\min_x \sum_{i=1}^{T} f_{j_i}(\mathbf{x})$. Thus, $\sum_{i=1}^{T} f_{j_i}(\mathbf{x}^{offline}) \leq \sum_{i=1}^{T} f_{j_i}(\mathbf{x}^*)$

Stochastic Gradient Descent Analysis

Claim (SGD Convergence)

After $T = \frac{R^2 G'^2}{\epsilon^2}$ *iterations:*

$$\mathbb{E}\left[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)\right] \leq \epsilon.$$

$$\begin{aligned}
\mathbb{E}[f\left(\hat{\mathbf{x}}\right) - f\left(\mathbf{x}^*\right)] &\leq \frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\left[f\left(\mathbf{x}^{(i)}\right) - f\left(\mathbf{x}^*\right)\right] \\
&= \frac{1}{T}\sum_{i=1}^{T} n\mathbb{E}\left[f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^*)\right] \\
&\leq \frac{n}{T}\cdot\mathbb{E}\left[\sum_{i=1}^{T} f_{j_i}(\mathbf{x}^{(i)}) - f_{j_i}(\mathbf{x}^{offline})\right] \\
&\leq \frac{n}{T}\cdot\left(R\cdot\frac{G'}{n}\cdot\sqrt{T}\right) \qquad \text{(by OGD guarantee.)} \\
&\leq \frac{RG'}{\sqrt{T}} \leq \epsilon
\end{aligned}$$

Stochastic vs. Full Batch Gradient Descent

Number of iterations for error ϵ:

- **Gradient Descent**: $T = \frac{R^2 G^2}{\epsilon^2}$.
- **Stochastic Gradient Descent**: $T = \frac{R^2 G'^2}{\epsilon^2}$.

Stochastic vs. Full Batch Gradient Descent

Number of iterations for error ϵ:

- **Gradient Descent**: $T = \frac{R^2 G^2}{\epsilon^2}$.
- **Stochastic Gradient Descent**: $T = \frac{R^2 G'^2}{\epsilon^2}$.

Always have $G \leq G'$:

$$\max_{\mathbf{x}} \|\nabla f(\mathbf{x})\|_2 = \max_{\mathbf{x}} \|\nabla f_1(\mathbf{x}) + \nabla f_2(\mathbf{x}) + \ldots \nabla f_n(\mathbf{x})\|_2$$

Stochastic vs. Full Batch Gradient Descent

Number of iterations for error ϵ:

- **Gradient Descent**: $T = \frac{R^2 G^2}{\epsilon^2}$.
- **Stochastic Gradient Descent**: $T = \frac{R^2 G'^2}{\epsilon^2}$.

Always have $G \leq G'$:

$$\begin{aligned}\max_{\mathbf{x}} \|\nabla f(\mathbf{x})\|_2 &= \max_{\mathbf{x}} \|\nabla f_1(\mathbf{x}) + \nabla f_2(\mathbf{x}) + \ldots \nabla f_n(\mathbf{x})\|_2 \\ &\leq \max_{\mathbf{x}} (\|\nabla f_1(\mathbf{x})\|_2) + \ldots + \max_{\mathbf{x}} (\|\nabla f_n(\mathbf{x})\|_2)\end{aligned}$$

Stochastic vs. Full Batch Gradient Descent

Number of iterations for error ϵ:

- **Gradient Descent**: $T = \frac{R^2 G^2}{\epsilon^2}$.
- **Stochastic Gradient Descent**: $T = \frac{R^2 G'^2}{\epsilon^2}$.

Always have $G \leq G'$:

$$\begin{aligned}
\max_{\mathbf{x}} \|\nabla f(\mathbf{x})\|_2 &= \max_{\mathbf{x}} \|\nabla f_1(\mathbf{x}) + \nabla f_2(\mathbf{x}) + \ldots \nabla f_n(\mathbf{x})\|_2 \\
&\leq \max_{\mathbf{x}} (\|\nabla f_1(\mathbf{x})\|_2) + \ldots + \max_{\mathbf{x}} (\|\nabla f_n(\mathbf{x})\|_2) \\
&\leq n \cdot \frac{G'}{n} = G'.
\end{aligned}$$

Stochastic vs. Full Batch Gradient Descent

Number of iterations for error ϵ:

- **Gradient Descent**: $T = \frac{R^2 G^2}{\epsilon^2}$.
- **Stochastic Gradient Descent**: $T = \frac{R^2 G'^2}{\epsilon^2}$.

Always have $G \leq G'$:

$$\begin{aligned}
\max_{\mathbf{x}} \|\nabla f(\mathbf{x})\|_2 &= \max_{\mathbf{x}} \|\nabla f_1(\mathbf{x}) + \nabla f_2(\mathbf{x}) + \ldots \nabla f_n(\mathbf{x})\|_2 \\
&\leq \max_{\mathbf{x}} (\|\nabla f_1(\mathbf{x})\|_2) + \ldots + \max_{\mathbf{x}} (\|\nabla f_n(\mathbf{x})\|_2) \\
&\leq n \cdot \frac{G'}{n} = G'.
\end{aligned}$$

Stochastic vs. Full Batch Gradient Descent

Number of iterations for error ϵ:

- **Gradient Descent**: $T = \frac{R^2 G^2}{\epsilon^2}$.
- **Stochastic Gradient Descent**: $T = \frac{R^2 G'^2}{\epsilon^2}$.

Always have $G \leq G'$:

$$\begin{aligned}
\max_{\mathbf{x}} \|\nabla f(\mathbf{x})\|_2 &= \max_{\mathbf{x}} \|\nabla f_1(\mathbf{x}) + \nabla f_2(\mathbf{x}) + \ldots \nabla f_n(\mathbf{x})\|_2 \\
&\leq \max_{\mathbf{x}} (\|\nabla f_1(\mathbf{x})\|_2) + \ldots + \max_{\mathbf{x}} (\|\nabla f_n(\mathbf{x})\|_2) \\
&\leq n \cdot \frac{G'}{n} = G'.
\end{aligned}$$

So GD converges in fewer iterations than SGD.

Stochastic vs. Full Batch Gradient Descent

Number of iterations for error ϵ:

- **Gradient Descent**: $T = \frac{R^2 G^2}{\epsilon^2}$.
- **Stochastic Gradient Descent**: $T = \frac{R^2 G'^2}{\epsilon^2}$.

Always have $G \leq G'$:

So GD converges in fewer iterations than SGD.

Stochastic vs. Full Batch Gradient Descent

Number of iterations for error ϵ:

- **Gradient Descent**: $T = \frac{R^2 G^2}{\epsilon^2}$.
- **Stochastic Gradient Descent**: $T = \frac{R^2 G'^2}{\epsilon^2}$.

Always have $G \leq G'$:

So GD converges in fewer iterations than SGD.

But for a fair comparison:

- SGD cost = (# of iterations) $\cdot\, O(1)$
- GD cost = (# of iterations) $\cdot\, O(n)$

Stochastic vs. Full Batch Gradient Descent

We always have $G \leq G'$. When it is much smaller, GD will perform better. When it is close to this upper bound, SGD will perform better.

Stochastic vs. Full Batch Gradient Descent

We always have $G \leq G'$. When it is much smaller, GD will perform better. When it is close to this upper bound, SGD will perform better.

What is an extreme case where $G = G'$?

Stochastic vs. Full Batch Gradient Descent

We always have $G \leq G'$. When it is much smaller, GD will perform better. When it is close to this upper bound, SGD will perform better.

What is an extreme case where $G = G'$?

- When all the f_i's are the same, i.e. is all the data points are identical

Stochastic vs. Full Batch Gradient Descent

We always have $G \leq G'$. When it is much smaller, GD will perform better. When it is close to this upper bound, SGD will perform better.

What is an extreme case where $G = G'$?

- When all the f_i's are the same, i.e. is all the data points are identical
- SGD does better when there are lots of repetitions in the dataset

Stochastic vs. Full Batch Gradient Descent

What if each gradient $\nabla f_i(\mathbf{x})$ looks like a random vector in $\mathbb{R}^d$ with $\mathcal{N}(0,1)$ entries?

$$\mathbb{E}\left[\|\nabla f_i(\mathbf{x})\|_2^2\right] = d$$

Stochastic vs. Full Batch Gradient Descent

What if each gradient $\nabla f_i(\mathbf{x})$ looks like a random vector in $\mathbb{R}^d$ with $\mathcal{N}(0,1)$ entries?

$$\mathbb{E}\left[\|\nabla f_i(\mathbf{x})\|_2^2\right] = d$$

$$\mathbb{E}\left[\|\nabla f(\mathbf{x})\|_2^2\right] = \mathbb{E}\left[\left\|\sum_{i=1}^{n} \nabla f_i(\mathbf{x})\right\|_2^2\right] = nd$$

Stochastic vs. Full Batch Gradient Descent

What if each gradient $\nabla f_i(\mathbf{x})$ looks like a random vector in $\mathbb{R}^d$ with $\mathcal{N}(0,1)$ entries?

$$\mathbb{E}\left[\|\nabla f_i(\mathbf{x})\|_2^2\right] = d$$

$$\mathbb{E}\left[\|\nabla f(\mathbf{x})\|_2^2\right] = \mathbb{E}\left[\left\|\sum_{i=1}^{n} \nabla f_i(\mathbf{x})\right\|_2^2\right] = nd$$

If you compare cost of GD vs SGD on such an instance, there will be no gain.

Stochastic vs. Full Batch Gradient Descent

Takeaway: SGD performs better when there is more structure or repetition in the data set.