IFT 6756 - Lecture 4
Optimization Background

January 29, 2021

Scribes: Arnaud L'Heureux, David Dobre and Ivan Puhachov **Instructor:** Gauthier Gidel

Summary

In this lecture we prove the convergence of gradient descent for strongly-convex and convex functions, show the convergence rate, and consequent optimization methods.
Keywords: smooth function, convex function, strictly convex function, condition number, Polyak-Lojasiewicz inequality, steepest descent, projected gradient descent.

1 Overview

A common goal of machine learning tasks is to learn parameters $\theta \in \Theta$ of a function f such that an objective is optimized. Objectives typically come in the form of a **loss function** l, which map one or more values of an event to a "cost" associated to that event. In the supervised learning problem, the inputs to the loss function are often $f(x_i)$ and y_i where x_i is an input sample and y_i is the corresponding label. By averaging the loss $l(f_\theta(x_i), y_i)$ over a finite dataset, we can get the **empirical risk** with respect to that dataset

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(f_\theta(x_i), y_i)\,.$$

The empirical risk is used to provide an estimate of the expected risk, which is the true generalization risk of a predictor. A common way to learn better parameters θ is via **empirical risk minimization**, which involves finding the parameters which minimize the empirical risk:

$$\begin{aligned}\theta^* &= \arg\min_{\theta\in\Theta} L(\theta) \\ &= \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} l(f_\theta(x_i), y_i)\,.\end{aligned}$$

We can consider this problem from two perspectives:

- **deterministic**: if the sum is over the entire dataset, we can treat it as a deterministic function $g(\theta)$ and evaluate

$$\arg\min_{\theta\in\Theta} g(\theta)$$

- **stochastic**: if the sum is over a subset of the dataset, we can treat as an expectation over the dataset and evaluate

$$\arg\min_{\theta} \mathbb{E}_{(x,y)\in\mathcal{D}_n}[l(f_\theta(x), y)]$$

A common optimization algorithm to efficiently solve this problem is **gradient descent**.
For the purposes of this course, we will focus on the deterministic perspective and study theoretical guarantees of batch gradient descent:

$$\theta_{t+1} = \theta_t - \eta \nabla g(\theta_t)\,.$$

If we do not use the full dataset to compute the gradients, we can instead estimate them by using a subset of the data (mini-batch); this is called **stochastic gradient descent**. The reader can consult [Nes03] or [Bub15] for more details on gradient descent.

We will prove the convergence of gradient descent for some classes of objective functions: strongly convex functions and convex functions. For non convex functions, consult referenced sources.

2 Assumptions

We restrict our analysis to a certain class of functions. Specifically, we will look at convex and strongly convex functions, as well those which satisfy a smoothness constraint. In this section, we will formally define all of the conditions and properties required to prove the convergence results in Section 3.

2.1 Convexity

We will discuss three conditions for convex functions[1]. These can be viewed as 0^{th}, 1^{st}, and 2^{nd} order convexity conditions which hold (and are equivalent) if the function is differentiable up to the appropriate order. For example, if a function is convex and twice differentiable, all three conditions hold and are equivalent.

Definition 1 (Convex function - 0^{th} order condition)**.** *A function g is* convex *iff* $\forall x, y, \forall \alpha \in [0,1]$,

$$g(\alpha x + (1-\alpha)y) \leq \alpha g(x) + (1-\alpha) g(y)$$

The term on the right of the inequality is called a **chord**. In two dimensions, a chord defines a straight line connecting two points of a function. Definition 1 can therefore be intuitively interpreted as "between any two points x and y in the domain of g, the function lies below the chord defined by $g(x)$ and $g(y)$".

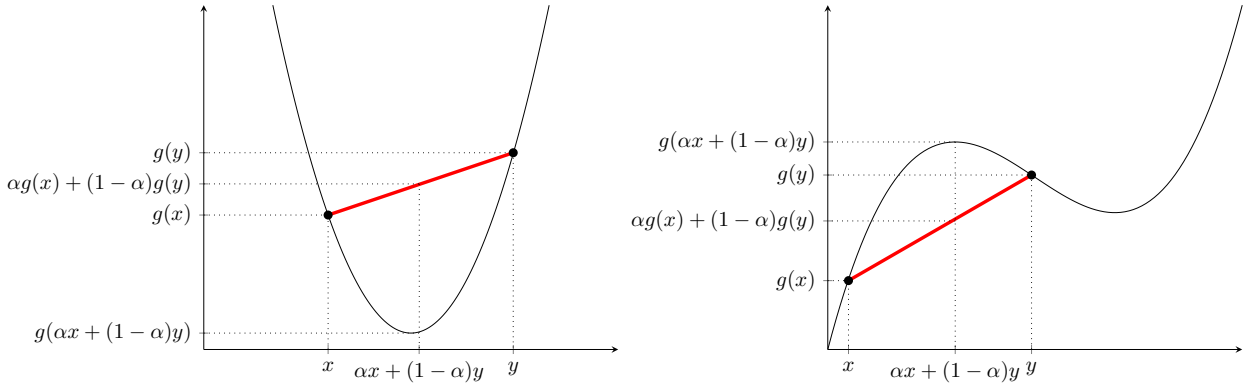

Figure 1: **Left:** Visual representation of a convex function. $\forall x, y \in \text{dom}(g)$, the chord $\alpha g(x) + (1-\alpha) g(y)$ lies above $g(\alpha x + (1-\alpha)y)$, $\alpha \in [0,1]$. **Right:** Visual representation of a non-convex function. It is shown that there exists points x and y such that $g(\alpha x + (1-\alpha)y) \geq \alpha g(x) + (1-\alpha) g(y)$.

Definition 2 (Convex function - 1^{st} order condition)**.** *A differentiable function g is convex iff* $\forall x, y$,

$$g(y) + g'(y)^\top (x - y) \leq g(x)$$

where g' is the first order gradient of g with respect to its parameters.

[1] Note that for each of these conditions, an additional requirement is that the domain of a convex function is a convex set but this is outside the scope of this course. See [BV11] for details.

Proof. Suppose g is convex. We can derive the first order convexity condition from Definition 1.

$$\begin{aligned} g\left(\alpha x + (1-\alpha)y\right) &\leq \alpha g(x) + (1-\alpha)g(y) \\ g\left(y + \alpha(x-y)\right) &\leq \alpha\left(g(x) - g(y)\right) + g(y) \\ \frac{g\left(y + \alpha(x-y)\right) - g(y)}{\alpha} &\leq g(x) - g(y) \end{aligned}$$

We can the limit of this inequality as $\alpha \to 0$, which yields the limit definition of a derivative of g with respect to y:

$$\begin{aligned} \lim_{\alpha \to \infty} \frac{g\left(y + \alpha(x-y)\right) - g(y)}{\alpha} &\leq g(x) - g(y) \\ g'(y)(x-y) &\leq g(x) - g(y) \\ g(y) + g'(y)(x-y) &\leq g(x) \end{aligned}$$

where one can see how $(x-y)$ comes out by applying l'Hopital's rule to evaluate the limit.
Now in case if we deal with vector-valued functions, $\lim_{\alpha\to\infty} \frac{g(y+\alpha(x-y))-g(y)}{\alpha}$ is a directional derivative, which is a dot product of gradient and the direction. This gives us:

$$\nabla g(y)^\top (x-y) \leq g(x) - g(y)$$

□

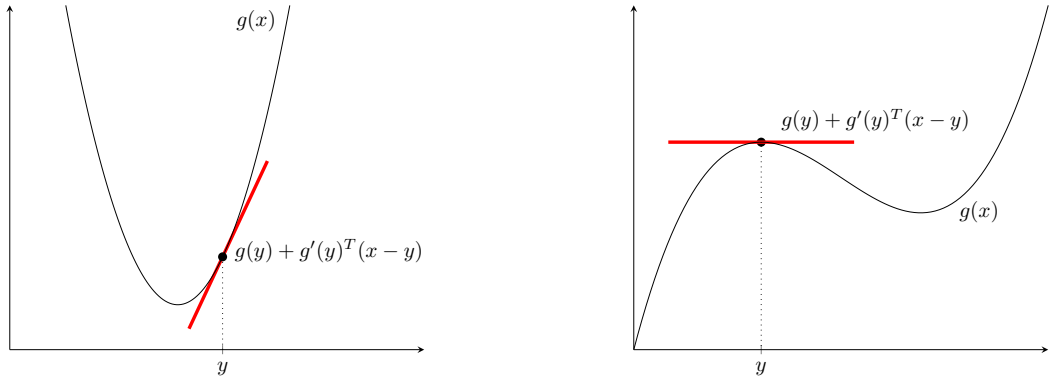

Figure 2: The first order convergence condition states that at every point on g, the tangent line at that point must lie completely below g.

Definition 3 (Convex function - 2nd order condition)**.** *A twice-differentiable function g is convex iff $\forall x$,*

$$\nabla^2 g(x) \succeq 0$$

The second order convexity condition states that the Hessian of g is positive semi-definite In other-words, all eigenvalues of g must be ≥ 0. The proof here is omitted; please refer to either Chapters 2 and 3 in [BV11], or the IFT 6085 scribe notes[2].

Theorem 4 (Local and global minima for convex functions)**.** *Let $g(x)$ be a convex function. If x^* is a local minimum of g, then x^* is also the global minimum of g.*

The proof for this theorem is also omitted but it can be fairly easily proven via contradiction, using the definition of a convex set (not covered in these notes) and definition 1. You can follow these notes for more details.

[2]Notes for all relevant proofs can be found in the scribe notes for "Basics of convex analysis and gradient descent", found on Ioannis Mitliagkas' site: `https://mitliagkas.github.io/ift6085-dl-theory-class/`

2.2 Strong convexity

Definition 5. *A function is **strongly convex** if:*

$$\forall a, b : g(b) + g'(b)(a-b) + \frac{\mu}{2}\|a-b\|^2 \leq g(a)$$

We will also call it μ-strongly convex for the values of μ that this condition holds.

Property 6. *If the function is twice differentiable we have:*

$$\forall \theta : \nabla^2 g(\theta) \succeq \mu \cdot I_d$$

Property 6 states that the Hessian of $g(\theta)$ is positive definite, and the smallest eignevalue is μ.

Example: Linear Regression

Consider the linear regression cost function:

$$g(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top \phi(x_i) - y_i)^2\,,$$

where $\phi(x)$ are features (fixed, so we optimize only weights). In this case the Hessian is

$$\nabla^2 g(\theta) = \frac{2}{n}\sum_{i=1}^{n}\phi(x_i)\phi(x_i)^\top\,.$$

Note that this Hessian only depends on the data. This means that with the right assumption on the data we automatically have the required assumptions on the function we want to optimize. For example, if data is bounded, we can bound the eigenvalues of hessian matrix. Having other assumptions on the data we can have even stronger assumptions on the function we optimize.

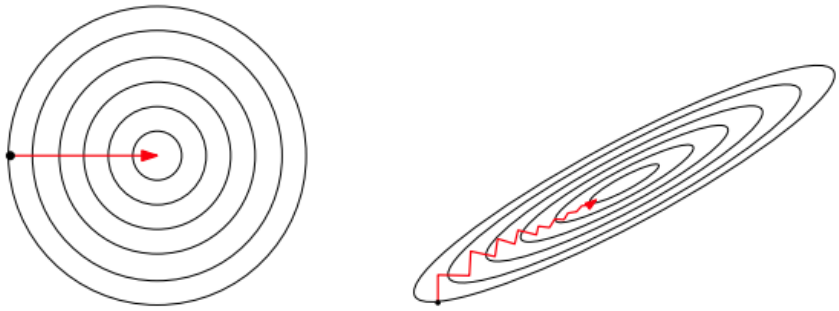

Figure 3: For a linear regression problem $\min_{x\in\mathbb{R}^n}\|Ax-b\|$ we are looking on eigenvalues of $AA^\top$. Having $\eta \preceq AA^\top \preceq L$ when $\frac{\eta}{L} \approx 1$ leads to good optimization, while $\frac{\eta}{L} \ll 1$ is called badly-conditioned, since optimization is way slower. Intuitively, we see that gradients steps on the right (badly-conditioned case) are pointed outside of true optimum direction.

2.3 Smoothness

A condition required for strongly convex functions is **smoothness** (specifically L-smoothness). In order to derive the convergence bounds for strongly convex functions, we will use some of the definitions defined in this section.

Definition 7 (Smooth function)**.** *A **smooth** function is a differentiable function with Lipschitz gradients:*

$$\forall \theta_1, \theta_2 \ : \ \|\nabla g(\theta_1) - \nabla g(\theta_2)\|_2 \leq L\|\theta_1 - \theta_2\|_2$$

We will call it L-smooth function.

Intuitively, this can be interpreted as a bound on how much the gradient can increase. See figure 4.

Property 8. *If the function is twice differentiable we have:*

$$\forall \theta \ : \ \nabla^2 g(\theta) \preceq L \cdot I_d \,,$$

where I_d is the d-dimensional identity matrix.

Property 8 states that the eigenvalues must be upper-bounded by a constant L.
Remark: the definition of $\preceq$ is: $A \preceq B \iff x^\top Ax \leq x^\top Bx \,, \quad \forall x.$

Lemma 9. *(Smoothness) For any L-smooth and twice differentiable function g:*

$$\forall \theta, \theta' : \ g(\theta') \leq g(\theta) + \nabla g(\theta)^\top (\theta' - \theta) + \frac{L}{2}\|\theta' - \theta\|_2^2$$

Proof. Apply Taylor expansion:

$$g(\theta') = g(\theta) + \nabla g(\theta)^\top (\theta' - \theta) + \frac{1}{2}(\theta' - \theta)^\top \nabla^2 g(\hat{\theta})(\theta' - \theta)$$

Now, having a smooth and twice differentiable g apply property 8:

$$\forall x \ : \ x^\top \cdot \nabla^2 g(\theta) \cdot x \leq L\|x\|_2^2$$

Which is true because L is the largest eigenvalue. Pass $x = (\theta' - \theta)$ to obtain:

$$g(\theta') \leq g(\theta) + \nabla g(\theta)^\top (\theta' - \theta) + \frac{L}{2}\|\theta' - \theta\|_2^2$$

□

Intuitively, this can be interpreted as the value of $g(\theta')$ lying between two quadratic functions.

2.4 Descent Property

One additional lemma that we will require is the descent lemma, which states that for appropriate step-sizes, an L-smooth convex function can never become larger as steps of gradient descent are taken.

Lemma 10. *Gradient Descent method satisfies "descent" property for a step-size $\eta \leq \frac{1}{L}$ when g is a twice differentiable and L-smooth function.*

$$\forall t \geq 0, \forall \eta \leq \frac{1}{L} : \ g(\theta_{t+1}) \leq g(\theta_t) - \frac{\eta}{2}\|\nabla g(\theta_t)\|_2^2$$

Note that since both η and $\|x\|_2^2$ are non-negative, this lemma shows that each step of gradient descent pulls our objective function closer to the minimum.

Proof. We begin with lemma 9 for $\theta = \theta_t$ and $\theta' = \theta_{t+1}$

$$g(\theta_{t+1}) \leq g(\theta_t) + \nabla g(\theta_t)^\top (\theta_{t+1} - \theta_t) + \frac{L}{2}\|\theta_{t+1} - \theta_t\|_2^2$$

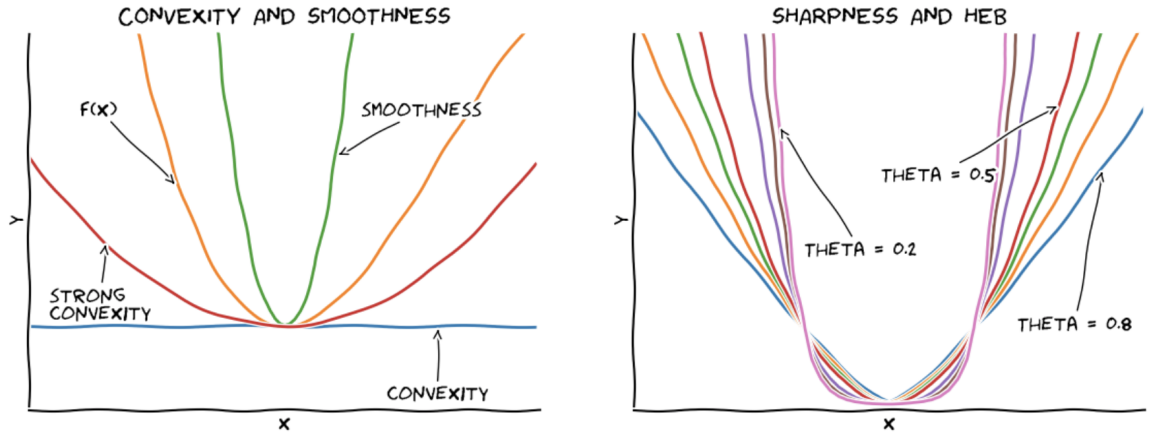

Figure 4: Visual sketch of definitions we introduced. On the right: convex, strongly convex, and smooth functions. On the left - Hölder Error Bounds, which are not considered in this lecture, yet interesting to explore. See the source [Pok] for more details.

Recall that gradient descent is defined as $\theta_{t+1} = \theta_t - \eta \nabla g(\theta_t)$ thus $\theta_{t+1} - \theta_t = -\eta \nabla g(\theta_t)$:

$$g(\theta_{t+1}) \leq g(\theta_t) - \eta \|\nabla g(\theta_t)\|_2^2 + \frac{L}{2}\eta^2 \|\nabla g(\theta_t)\|_2^2 = g(\theta_t) + (-\eta + \frac{L}{2}\eta^2) \cdot \|\nabla g(\theta_t)\|_2^2$$

We want this upper bound on $g(\theta_{t+1})$ to be as small as possible, and the only variable we control is the stepsize η. The minimizer of $-\eta + \frac{L}{2}\eta^2$ is $\eta^* = \frac{1}{L}$ (parabola peak), and is the best stepsize.
Finally, we must show that $(-\eta + \frac{L}{2}\eta^2) \leq -\frac{\eta}{2}$ for $0 < \eta \leq \frac{1}{L}$, $L > 0$. The simplest way is:

$$\eta \leq \frac{1}{L} \iff L\eta \leq 1 \iff \frac{L\eta^2}{2} \leq \frac{\eta}{2} \iff \frac{L}{2}\eta^2 - \eta \leq \frac{\eta}{2} - \eta = -\frac{\eta}{2}$$

Which concludes the proof. □

Simple example: having $g(\theta) = \frac{\theta^2}{2}$, minimum value is simply $g^* = 0$ achieved at $\arg\min g(\theta) = 0 = \theta^*$, and Lipshitz constant here is $L = 1$. Gradient descent is then $\theta_{t+1} = \theta_t - \eta\theta_t = (1 - \eta)\theta_t$.

- Optimal stepsize $\eta = \frac{1}{L} = 1$ will move us directly to the optimal point $\theta_1 = 0$
- Any stepsize $\eta \leq \frac{1}{L}$: $\eta \in (0, 1]$ will move the point closer to optimum
- Stepsize in a range $\eta \in (\frac{1}{L}, \frac{2}{L})$ (from peak of parabola to another intersection point) will make SGD oscillate between positive and negative values (this theorem does not cover this case)
- Stepsize $\eta = \frac{2}{L} = 2$ makes SGD to switch between $+\theta_0$ and $-\theta_0$, no progress here, the method is stuck
- Stepsize too big $\eta > \frac{2}{L}$ and SGD diverges.

Remark 1: We haven't used strong convexity yet, only smoothness.
Remark 2: The descent property ties together stepsize and Lipschitz constant (property of a function)

3 Proofs of convergence

With the majority of the background out of the way, we will now derive convergence bounds for both strongly convex, and convex functions. These bounds will give a convergence rate in terms of the step size.

3.1 Convergence for Strongly Convex Functions

Lemma 11 (Polyak-Lojasiewicz inequality)**.** *For a μ-strongly convex function g:*

$$\forall \theta : \ \|\nabla g(\theta)\|_2^2 \geq 2\mu \cdot (g(\theta) - g^*)$$

where $g^* = \min_\theta g(\theta)$.

Proof. From the definition of strong convexity (definition 5) having $b = \theta$ and $a = \phi$:

$$g(\theta) + \nabla g(\theta)^\top (\phi - \theta) + \frac{\mu}{2}\|\theta - \phi\|_2^2 \leq g(\phi)$$

Minimize both sides with respect to ϕ:

$$\min_\phi \left[g(\theta) + \nabla g(\theta)^\top (\phi - \theta) + \frac{\mu}{2}\|\theta - \phi\|_2^2 \right] \leq \min_\phi g(\phi) = g^*$$

To minimize left-hand side, compute the gradient with respect to ϕ and set it to zero. This gives us:

$$\nabla g(\theta) + \frac{\mu}{2}(2\phi - 2\theta) = 0$$

So $\phi = \theta - \frac{1}{\mu}\nabla g(\theta)$ is $\arg\min_\phi$ of the left-hand side. If we substitute it we have:

$$g(\theta) + \nabla g(\theta)^\top \left(-\frac{1}{\mu}\nabla g(\theta) \right) + \frac{\mu}{2} \cdot \frac{1}{\mu^2}\|\nabla g(\theta)\|_2^2 \leq g^*$$

$$g(\theta) + (-\frac{1}{\mu} + \frac{1}{2\mu}) \cdot \|\nabla g(\theta)\|_2^2 \leq g^*$$

And after re-arranging we have:

$$g(\theta) - g^* \leq g(\theta) - g(\theta')$$

□

Note that this also applies for the general convex case, but is useless since $\mu = 0$

Theorem 12 (Convergence result)**.** *For a L-smooth and μ-strictly convex function g we have an upper bound on gradient descent minimization process:*

$$g(\theta_t) - g^* \leq (1 - \frac{\mu}{L})^t (g(\theta_0) - g^*)$$

Proof. First apply descent lemma 10:

$$g(\theta_{t+1}) \leq g(\theta_t) - \frac{\eta}{2}\|\nabla g(\theta_t)\|_2^2$$

Now add distance lemma 11:

$$g(\theta_{t+1}) \leq g(\theta_t) - \eta\mu \cdot (g(\theta_t) - g^*)$$

Subtract g^* from both sides:

$$g(\theta_{t+1}) - g^* \leq (1 - \eta\mu)\,(g(\theta_t) - g^*)$$

Finally, as we obtain a recurrent relation, we can write

$$g(\theta_{t+1}) - g^* \leq (1 - \eta\mu)\,(g(\theta_t) - g^*) \leq (1 - \eta\mu)^2\,(g(\theta_{t-1}) - g^*) \leq ... \leq (1 - \eta\mu)^{t+1}(g(\theta_0) - g^*)$$

Passing the optimal stepsize from descent lemma 10: $\eta = \frac{1}{L}$ concludes the proof. □

3.1.1 Convergence rate

This convergence rate is called **linear**. We now want to consider how many steps is required to get some desired precision ϵ.

$$\begin{aligned} g(\theta_t) - g^* &\leq \epsilon \\ (1 - \frac{\mu}{L})^t (g(\theta_0) - g^*) &\leq \epsilon \\ t &\geq \frac{L}{\mu} \log(\frac{g(\theta_0) - g^*}{\epsilon}) \end{aligned}$$

The convergence result of Theorem 12 guarantees that $\frac{L}{\mu} \log(\frac{g(\theta_0)-g^*}{\epsilon})$ steps is enough to reach desired precision ϵ.

3.2 Convergence for Convex Functions

To derive the convergence bounds for convex functions, we will require two lemmas.

Lemma 13. *For a convex function g, step-size η, parameters at a given time step θ_i, and optimal parameters $\theta^* = \arg\min_{\theta \in \Theta} g(\theta)$,*

$$g(\theta_{t+1}) - g(\theta^*) \leq \nabla g(\theta_t)^\top (\theta_t - \theta^*) - \frac{\eta}{2} \|\nabla(\theta_t)\|_2^2$$

Proof. We will combine the first-order convexity condition (Lemma 2) and the descent lemma (Lemma 10). From the first-order convexity condition, we rearrange the terms to express the inequality in terms of the difference between g evaluated at θ_t and θ^*:

$$\begin{aligned} g(\theta^*) &\geq g(\theta_t) + \nabla g(\theta_t)^\top (\theta^* - \theta_t) && \text{(Lemma 2)} \\ g(\theta_t) - g(\theta^*) &\leq \nabla g(\theta_t)^\top (\theta_t - \theta^*) \end{aligned}$$

We can substitute the descent lemma, expressed as $g(\theta_t) = g(\theta_{t+1}) + \eta/2 \|\nabla g(\theta_t)\|_2^2$, into the above expression. Note that this is a valid operation because $g(\theta_t)$ is greater than or equal to the expression we will be substituting in, which ensures the inequality will hold.

$$\begin{aligned} \left(g(\theta_{t+1}) + \frac{\eta}{2} \|\nabla g(\theta_t)\|_2^2\right) - g(\theta^*) &\leq \nabla g(\theta_t)^\top (\theta_t - \theta^*) && \text{(Using Lemma 10)} \\ g(\theta_{t+1}) - g(\theta^*) &\leq \nabla g(\theta_t)^\top (\theta_t - \theta^*) - \frac{\eta}{2} \|\nabla g(\theta_t)\|_2^2 \end{aligned}$$

□

Lemma 14. *For a convex function g, step-size η, parameters at a given time step θ_i, and optimal parameters $\theta^* = \arg\min_{\theta \in \Theta} g(\theta)$,*

$$g(\theta_{t+1}) - g(\theta^*) \leq \frac{1}{2\eta} \left(\|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*)\|_2^2 \right)$$

Proof. We will find an equivalent expression for $\|\theta_{t+1} - \theta^*\|_2^2$ by using the descent lemma, and then apply Lemma 13 to convert this into an inequality.

$$\begin{aligned} \|\theta_{t+1} - \theta^*\|_2^2 &= \|(\theta_t - \theta^*) - \eta \nabla g(\theta_t)\|_2^2 && \text{(Using Lemma 10)} \\ \|\theta_{t+1} - \theta^*\|_2^2 &= \|\theta_t - \theta^*\|_2^2 - 2\eta\, g(\theta_t)^\top (\theta_t - \theta^*) + \eta^2 \|\nabla g(\theta_t)\|_2^2 \\ 2\eta \nabla g(\theta_t)^\top (\theta_t - \theta^*) - \eta^2 \|\nabla g(\theta_t)\|_2^2 &= \|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*\|_2^2 && \text{(Rearranging)} \\ \nabla g(\theta_t)^\top (\theta_t - \theta^*) - \frac{\eta}{2} \|\nabla g(\theta_t)\|_2^2 &= \frac{1}{2\eta} \left(\|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*\|_2^2 \right) \\ g(\theta_{t+1}) - g(\theta^*) &\leq \frac{1}{2\eta} \left(\|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*\|_2^2 \right) && \text{(Using Lemma 13)} \end{aligned}$$

□

Armed with Lemma 14, we can now derive convergence guarantees for convex functions.

Theorem 15 (Convergence for Convex Functions)**.** *For a convex function g, step-size $\eta \leq 1/L$, parameters at a given time step θ_i, and optimal parameters $\theta^* = \arg\min_{\theta \in \Theta} g(\theta)$,*

$$g(\theta_T) - g(\theta^*) \leq \frac{\|\theta_0 - \theta^*\|_2^2}{2\eta(T-1)}$$

Proof. We sum Lemma 14 over steps 0 to $T-1$ which yields a telescoping sum of the $\|\theta_i - \theta^*\|_2^2$ terms, allowing most of them to be cancelled. We then use the fact that g is convex to simplify the sum of $g(\theta_{t+1})$ terms and obtain the final expression.

$$\begin{aligned}
\sum_{t=0}^{T-1} (g(\theta_{t+1}) - g(\theta^*)) &\leq \frac{1}{2\eta} \sum_{t=0}^{T-1} \left(\|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*\|_2^2 \right) \\
\sum_{t=0}^{T-1} g(\theta_{t+1}) - T g(\theta^*) &\leq \frac{1}{2\eta} \sum_{t=0}^{T-1} (D_t - D_{t+1}) && \text{(Substituting } D_t = \|\theta_t - \theta^*\|_2^2\text{)} \\
\sum_{t=0}^{T-1} g(\theta_{t+1}) - T g(\theta^*) &\leq \frac{1}{2\eta} \left((D_0 - D_1) + (D_1 - D_2) + \cdots + (D_{T-1} - D_T) \right) \\
\sum_{t=0}^{T-1} g(\theta_{t+1}) - T g(\theta^*) &\leq \frac{1}{2\eta} (D_0 - D_T) \\
\sum_{t=0}^{T-1} g(\theta_{t+1}) - T g(\theta^*) &\leq \frac{1}{2\eta} D_0 && \text{(Using } D_0 - D_T \leq D_0\text{)} \\
T\, g(\theta_T) - T g(\theta^*) &\leq \frac{1}{2\eta} D_0 && \text{(Using } T\, g(\theta_T) \leq \textstyle\sum_{t=0}^{T-1} g(\theta_{t+1})\text{)} \\
g(\theta_T) - g(\theta^*) &\leq \frac{\|\theta_0 - \theta^*\|_2^2}{2\eta(T-1)}
\end{aligned}$$

The final substitution for $T\, g(\theta_T)$ is valid because g is convex. □

3.3 Why care about rate?

The total error of a machine learning model is the sum of three different error terms [BB08]:

- **Approximation error (bias)**: how well we can possibly approximate target with our class of estimators
- **Estimation error (variance)**: how well we can approximate the predictor given our subset of training data
- **Optimization error**: how well we fit the given data

There is no need to reduce the **optimization error** below the **estimation error**. Recall that the objective is to minimize the expected risk. We are already making errors due to the bias (minimizing with respect to θ instead of f), the variance (minimizing the empirical risk via a finite sum rather than the true expectation), and because we cannot fully solve this optimization problem. Minimizing the optimization error will not improve the bias and variance error terms, so it is sufficient for it to be of the same order of magnitude as the first two terms.

In summary, guarantees on convergence rates give us an idea of how good our optimization algorithms are. By the above arguments, we desire a fast optimization algorithm, but don't need to over-optimize beyond the bias and variance error terms. Knowing the convergence rates for different optimization algorithms allows us to know when an optimization algorithm is good enough to satisfy our problem requirements, saving both research and compute time.

4 Other Algorithms

4.1 Steepest Descent

Steepest descent is another (larger) class of optimization algorithms beyond gradient descent. The idea is to follow the steepest descent direction with respect to a particular geometry. This direction is defined as the minimizer in a given bowl(?) of the inner product with the gradient Steepest descent has an application for adversarial examples. Consider

$$\theta_{t+1} = \theta_t + \eta \cdot d$$
$$d := \arg \min_{\|d\| \leq 1} \nabla g(\theta_t)^\top d$$

Here, the motivation is that d is the direction that correlates the loss. For example, if we take L_2 norm ($\|d\|_2 \leq 1$) we obtain $d = -\nabla g$ and the overall method is (normalized) gradient descent.

Remark: Proof of convergence of these methods are not trivial.
Remark: to create adversarial examples, the method inside is exactly steepest **ascent** with L_∞ norm

Exercise: What if we take L_∞ norm? Answer: $-\text{sgn}(\nabla g)$

4.1.1 Penalized steepest descent

A more natural version of steepest descent is **penalized steepest descent**. In this algorithm, we add a penalty for large steps (given by the $\|\theta - \theta_t\|_n^2$ term).

$$\theta_{t+1} := \arg \min_{\theta \in \mathbb{R}^d} \nabla g(\theta_t)^\top (\theta - \theta_t) + \frac{1}{2\eta} \|\theta - \theta_t\|_n^2$$

If the L_2 norm is used in $\|\theta - \theta_t\|_n^2$, we recover gradient descent; other norms will give us new methods.

For further reading on steepest descent and regularized steepest descent methods, refer to [BV11] and Theorem 1 in [Kel+13].

4.2 Projected Gradient Descent

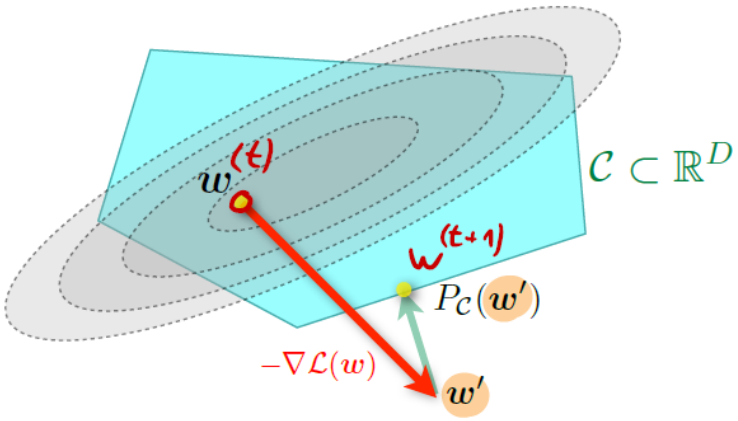

Figure 5: Visualization of projected gradient descent. $\mathcal{C}$ is the space of allowed parameters ω, $P_{\mathcal{C}}$ is the projection operator. Source [Kja]

If we have constraints in our space of hypothesis, you can use **projected gradient descent**. In this case, you still follow the gradients, but you project it on the constrained set. The notion of optimally in this case is a bit different;

instead of optimizing to a point in which the gradient is zero, the optimal point is such that the gradient is orthogonal to the front(?) of your constrained set. The update step for projected gradient descent is

$$\theta_{t+1} = P_\Theta[\theta_t - \eta \nabla g(\theta_t)]\,,$$

where $P_\theta[\cdot]$ is a projection.

Lemma 16. *If Θ is a convex set, then the projection on Θ is contractive:*

$$\|P_\Theta[x] - P_\Theta[y]\| \leq \|x - y\|$$

By using this property we can show that Lemma 10, Lemma 13 and Lemma 14 are still valid.

References

[BB08] Léon Bottou and Olivier Bousquet. "The Tradeoffs of Large Scale Learning". In: *Advances in Neural Information Processing Systems*. Ed. by J. Platt et al. Vol. 20. Curran Associates, Inc., 2008, pp. 161–168 (cit. on p. 9).

[Bub15] Sébastien Bubeck. "Convex optimization: Algorithms and complexity". In: *Foundations and Trends in Machine Learning* 8 (2015), pp. 231–358. DOI: 10.1561/2200000050 (cit. on p. 2).

[BV11] Stephen P. Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge Univ. Pr., 2011. URL: https://web.stanford.edu/~boyd/cvxbook/ (cit. on pp. 2, 3, 10).

[Kel+13] Jonathan A. Kelner et al. *An Almost-Linear-Time Algorithm for Approximate Max Flow in Undirected Graphs, and its Multicommodity Generalizations*. 2013. arXiv: 1304.2338 [cs.DS] (cit. on p. 10).

[Kja] Maxime Kjaer. *CS-443 Machine Learning Notes*. https://kjaer.io/ml/#projected-gradient-descent. Accessed: 2021-02-15 (cit. on p. 10).

[Nes03] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science Business Media, 2003 (cit. on p. 2).

[Pok] Sebastian Pokutta. *Cheat Sheet: Smooth Convex Optimization*. http://www.pokutta.com/blog/research/2018/12/07/cheatsheet-smooth-idealized.html. Accessed: 2021-02-15 (cit. on p. 6).