CS 189 Introduction to Machine Learning
Spring 2018 Note 13

1 Optimization

As we move into the realm of neural networks and beyond, we will be solving arbitrary problems of the form

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

over an arbitrary objective function $f : \mathbb{R}^d \to \mathbb{R}$ and arbitrary domain $\mathcal{X}$. Solving such problems is the focus of **optimization**, an extensive field that has applications in control theory, finance, and machine learning. Most optimization problems do not necessarily have a closed form solution, and therefore an *iterative algorithm* is needed to solve them. As we will see, there is no one universal algorithm that is suited for solving all problems — rather, certain algorithms are more suitable over others depending on the specific underlying assumptions about the problem, such as convexity and smoothness.

1.1 Gradients

Gradients form the basis for many of the optimization algorithms that we will study. Given that f is continuously differentiable, the gradient is defined as the vector of partial derivatives of f, denoted by

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_d} \end{bmatrix}$$

Why do we care about gradients? Among many reasons, the gradient being zero is a necessary condition for local minima. Let's understand why this is true. We define **critical points** as points where the gradient is zero. These points can be classified into three categories:

(i) **local/global minimum**: a point $\mathbf{x} \in \mathcal{X}$ such that there exists a neighborhood around $\mathbf{x}$ where $f(\mathbf{x})$ attains the minimum value

(ii) **local/global minima**: a point $\mathbf{x} \in \mathcal{X}$ such that there exists a neighborhood around $\mathbf{x}$ where $f(\mathbf{x})$ attains the maximum value

(iii) **saddle point**: a point $\mathbf{x} \in \mathcal{X}$ such that for all neighborhoods around $\mathbf{x}$ there exists $\mathbf{y}, \mathbf{z}$ such that $f(\mathbf{y}) \leq f(\mathbf{x}) \leq f(\mathbf{z})$

In optimization we are interested in finding the **global minimum** of a function, but in many circumstances we may settle for **local minima** instead.

Note 13, ©UCB CS 189, Spring 2018. All Rights Reserved. This may not be publicly shared without explicit permission.

Proposition 1. *If $\mathbf{x}^*$ is a local minimum of f and f is continuously differentiable in a neighborhood of $\mathbf{x}^*$, then $\nabla f(\mathbf{x}^*) = \mathbf{0}$.*

Proof. See math4ml. □

Note however, that while setting the gradient to zero is a necessary condition for local minima, it is not a sufficient condition. In many circumstances, the function that we are optimizing may not have a local minima, and generally setting the gradient to zero could yield local maxima or saddle points.

1.2 Hessian

Given that f is twice continuously differentiable, we define the **Hessian** as the matrix of second partial derivatives of f, denoted by

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_d \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_d^2} \end{bmatrix}$$

The Hessian being PSD is another necessary condition for local minima.

Proposition 2. *If $\mathbf{x}^*$ is a local minimum of f and f is twice continuously differentiable in a neighborhood of $\mathbf{x}^*$, then $\nabla^2 f(\mathbf{x}^*)$ is positive semi-definite.*

Proof. See math4ml. □

Unfortunately, the gradient being zero and the Hessian being PSD together are still not sufficient conditions local minima (consider the function $f(x) = x^3$). However, gradient being zero and the Hessian being PSD in a *neighborhood* are sufficient conditions. Slightly stronger conditions are needed to establish that a point is a **strict local minimum**.

Proposition 3. *Suppose f is twice continuously differentiable with $\nabla^2 f$ positive semi-definite in a neighborhood of $\mathbf{x}^*$, and that $\nabla f(\mathbf{x}^*) = \mathbf{0}$. Then $\mathbf{x}^*$ is a local minimum of f. Furthermore if $\nabla^2 f(\mathbf{x}^*)$ is positive definite, then $\mathbf{x}^*$ is a strict local minimum.*

Proof. See math4ml. □

1.3 Convexity

Convex optimization is a subfield of optimization which deals with convex problems — problems in which the objective function is convex and the domain is a convex set. Convex functions are convenient due to their "bowl shape," which induces many useful properties.

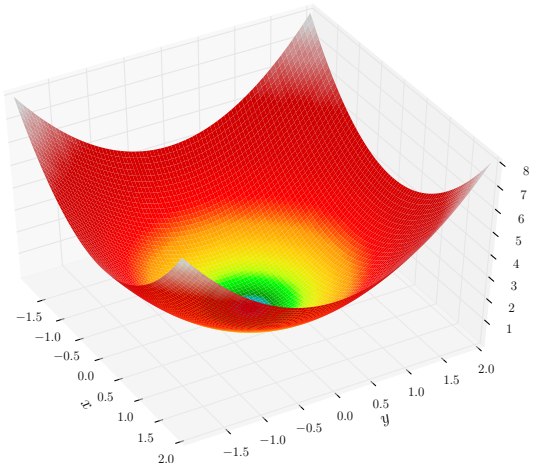

Figure 1: Source: Wikipedia

Given that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable, the following are equivalent conditions of convexity:

(i) $f(t\mathbf{x} + (1-t)\mathbf{y}) \leq tf(\mathbf{x}) + (1-t)f(\mathbf{y}), \quad \forall \mathbf{x}, \mathbf{y}, t \in [0,1]$

(ii) $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}), \quad \forall \mathbf{x}, \mathbf{y}$

(iii) $(\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}))^\top (y - x) \geq 0, \quad \forall \mathbf{x}, \mathbf{y}$

(iv) $\nabla^2 f(\mathbf{x}) \succeq \mathbf{0}, \quad \forall \mathbf{x}$

Let's understand the equivalent conditions of convexity. The first condition states that for any two points $\mathbf{x}, \mathbf{y}$, the function lies below the line segment connecting $\mathbf{x}$ and $\mathbf{y}$.

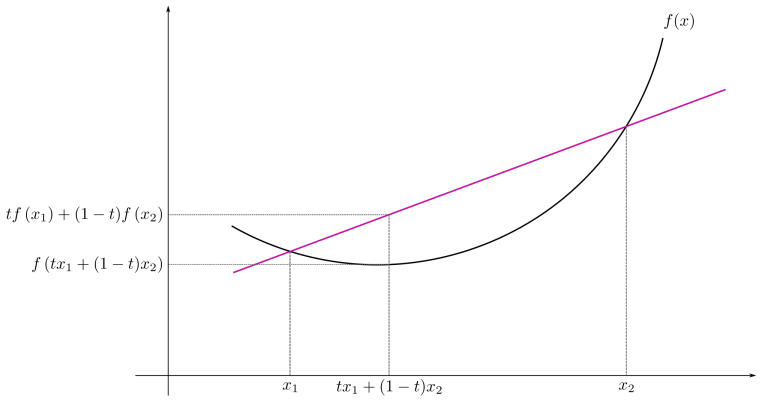

Figure 2: Source: Wikipedia

The second condition, also known as the first-order condition, states that any tangent line to f must lie below the entire function. The third condition intuitively states that if $\mathbf{y}$ is greater than $\mathbf{x}$, then the derivative of $\mathbf{y}$ is also greater than the derivative of $\mathbf{x}$. Finally, the last condition states that the second derivative of f is always non-negative.

Why are we interested in convex problems? One reason is that convex functions do not have saddle points or local maxima, beacause the Hessian is PSD for all points in the domain. As a result, any critical point must be a local minimum. In fact, any local minimum is also a global minimum, so any point that has a zero gradient must be the global minimum.

Proposition 4. *Let $\mathcal{X}$ be a convex set. If f is convex, then any local minimum of f in $\mathcal{X}$ is also a global minimum.*

Proof. See math4ml. □

Consequently we can find any point for which the gradient is zero and guarantee that it is the global minimum (this is exactly the case in OLS and Ridge Regression since the objective function is PSD and therefore convex). Note however, that this does not imply that the global minimum is unique — there could be several different points which achieve the global minimum.

1.3.1 Strong Convexity

For a strictly positive $m \in \mathbb{R}$, a function is m**-strongly convex** if the following equivalent conditions hold:

(i) $f(t\mathbf{x} + (1-t)\mathbf{y}) \leq tf(\mathbf{x}) + (1-t)f(\mathbf{y}) - \frac{t(1-t)m}{2}\|\mathbf{y}-\mathbf{x}\|^2, \quad \forall \mathbf{x}, \mathbf{y}, t \in [0,1]$

(ii) $g(\mathbf{x}) = f(\mathbf{x}) - \frac{m}{2}\|x\|^2$ is convex

(iii) $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{y}-\mathbf{x}) + \frac{m}{2}\|\mathbf{y}-\mathbf{x}\|^2, \quad \forall \mathbf{x}, \mathbf{y}$

(iv) $(\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}))^\top(y-x) \geq m\|\mathbf{y}-\mathbf{x}\|^2, \quad \forall \mathbf{x}, \mathbf{y}$

(v) $\nabla^2 f(\mathbf{x}) \succeq m\mathbf{I}, \quad \forall \mathbf{x}$

The conditions for strong convexity are identical to those for convex functions, with the exception of an additional term involving m. Strongly convex functions provide several advantages over general convex functions. Intuitively, strongly convex functions can be lower bounded by a quadratic function, which establishes the uniqueness of a global minimum.

Proposition 5. *Let $\mathcal{X}$ be a convex set. If f is strictly convex, then there exists at most one local minimum of f in $\mathcal{X}$. Consequently, if it exists it is the unique global minimum of f in $\mathcal{X}$.*

Proof. See math4ml. □

If the Hessian of $\nabla^2 f$ has eigenvalues that are all strictly positive at all points, then f is m-strongly convex with m equal to the the smallest eigenvalue of $\nabla^2 f$ (over all points $\mathbf{x}$). Recall from our discussion of OLS vs. Ridge Regression that while OLS may have several solutions, Ridge Regression has a unique solution. This is because the Ridge Regression formulation is positive definite and thus strongly convex, while OLS is positive semi-definite and not necessarily strongly convex.

1.3.2 Smoothness

While strongly convex functions are *lower bounded* by a quadratic function, smooth functions are *upper bounded* by a quadratic function.

An M**-smooth** or more formally **Lipschitz continuous gradient** function is one for which there exists a strictly positive $M \in \mathbb{R}$ such that

$$\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x})\| \leq M\|\mathbf{y} - \mathbf{x}\|, \quad \forall \mathbf{x}, \mathbf{y} \tag{1}$$

This definition does not assume that f is convex. 1 implies all of the following equivalent conditions:

(i) $f(t\mathbf{x} + (1-t)\mathbf{y}) \geq tf(\mathbf{x}) + (1-t)f(\mathbf{y}) - \frac{t(1-t)M}{2}\|\mathbf{y} - \mathbf{x}\|^2, \quad \forall \mathbf{x}, \mathbf{y}, t \in [0,1]$

(ii) $f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{y} - \mathbf{x}) + \frac{M}{2}\|\mathbf{y} - \mathbf{x}\|^2, \quad \forall \mathbf{x}, \mathbf{y}$

(iii) $(\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}))^\top(\mathbf{y} - \mathbf{x}) \leq M\|\mathbf{y} - \mathbf{x}\|^2, \quad \forall \mathbf{x}, \mathbf{y}$

(iv) $\nabla^2 f(\mathbf{x}) \preceq M\mathbf{I}, \quad \forall \mathbf{x}$

When f is convex, then the above conditions also imply 1, establishing equivalence among all of the conditions. Roughly speaking, smoothness is the same as strong convexity, except with the inequality signs flipped. If the Hessian of $\nabla^2 f$ has eigenvalues that are bounded from above, f is M-smooth with M equal to the the maximum eigenvalue of $\nabla^2 f$ (over all points $\mathbf{x}$). As we will see, strong convexity and smoothness in conjunction will provide lower and upper bounds for f, allowing us to achieve a significantly faster convergence rate for many optimization algorithms.

1.4 Gradient Descent Methods

Gradient Descent is an algorithm that iteratively takes small steps in a descent direction using gradient information. For the purposes of our discussion, let's assume our optimization problem is unconstrained over $\mathbf{w} \in \mathbb{R}^d$:

$$\min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w})$$

In its simplest form, gradient descent takes steps in the direction of *steepest descent*. Suppose that we are currently at a point $\mathbf{w}^{(t)}$ in the domain of the function. The direction of steepest descent at $\mathbf{w}^{(t)}$ is defined as the negative of the gradient at that point, $-\nabla f(\mathbf{w}^{(t)})$. To see why, recall that the directional derivative in a unit direction $\mathbf{u}$ at $\mathbf{w}^{(t)}$ is defined as the inner product of the gradient and the direction:

$$D_{\mathbf{u}} f(\mathbf{w}^{(t)}) = \langle \nabla f(\mathbf{w}^{(t)}), \mathbf{u} \rangle = \|\nabla f(\mathbf{w}^{(t)})\| \cdot \|\mathbf{u}\| \cdot \cos(\theta)$$

where θ is the angle between $\nabla f(\mathbf{w}^{(t)})$ and $\mathbf{u}$. We can minimize the directional derivative by setting $\theta = -\pi$, which will mean that the direction $\mathbf{u}$ and $\nabla f(\mathbf{w}^{(t)})$ are opposite to each other, and thus the direction of steepest descent $\mathbf{u}^*$ is in the direction opposite that of the gradient. Hence, the

direction of steepest descent is $-\nabla f(\mathbf{w}^{(t)})$. The gradient descent algorithm will take an arbitrary step in this direction, scaling the gradient by a scalar α_t.

Algorithm 1: Gradient Descent

Initialize $\mathbf{w}^{(0)}$ to a random point
while $f(\mathbf{w}^{(t)})$ *not converged* **do**
  $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \alpha_t \nabla f(\mathbf{w}^{(t)})$

Determining this scaling α_t is dependent on the attributes of the function f. Sometimes we can set the scaling to a constant value and converge to the optimum value, whereas in other instances we need to determine an *adaptive stepsize*. A scaling that is too high may cause the algorithm to diverge from the optimal solution, whereas a scaling that is too low may cause the algorithm to converge too slowly. For certain classes of functions, there are theoretical guarantees that establish convergence. Assuming that the distance from the initial point $\mathbf{w}^{(0)}$ and the optimal point $\mathbf{w}^*$ is R, we have the following:

properties of f	stepsize α_t	convergence rate to $f(\mathbf{w}^*)$
convex, L-Lipschitz	$\frac{R}{L\sqrt{t}}$	$O(\frac{1}{\sqrt{t}})$
m-strongly convex, L-Lipschitz	$\frac{2}{m(t+1)}$	$O(\frac{1}{t})$
convex, M-smooth	$\frac{1}{M}$	$O(\frac{1}{t})$
m-strongly convex, M-smooth	$\frac{1}{M}$	$O(\exp(-t\frac{m}{M}))$

For detailed proofs of rates above, refer to the EE 227C lecture notes. Individually, strong convexity and smoothness will allow us to accelerate the rate of convergence from $O(\frac{1}{\sqrt{t}})$ to $O(\frac{1}{t})$. Put together, they allow us to achieve an exponential convergence rate — a significant acceleration! The quantity $\kappa = \frac{M}{m}$ is known as the **condition number** — the ratio of the largest over smallest singular value of the Hessian of f. Recall from our discussion of OLS vs. Ridge Regression that Ridge Regression adds a small penalty term $\lambda\|\mathbf{w}\|^2$ to the objective, effectively making the problem strongly convex. Since the OLS is already smooth as well, then gradient descent can achieve an exponential rate of convergence to the optimal value. The higher the value of λ, the lower the value of the condition number κ, which leads to an even faster convergence rate.

1.4.1 Gradient Descent with Momentum

Algorithm 2: Gradient Descent with Momentum

Initialize $\mathbf{w}^{(0)}$ to a random point
while $f(\mathbf{w}^{(t)})$ *not converged* **do**
  $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \alpha_t \nabla f(\mathbf{w}^{(t)}) + \beta_t(f(\mathbf{w}^{(t)}) - f(\mathbf{w}_{t-1}))$

1.4.2 Stochastic Gradient Descent

Algorithm 3: Stochastic Gradient Descent

Initialize $\mathbf{w}^{(0)}$ to a random point
while $f(\mathbf{w}^{(t)})$ *not converged* **do**
  Sample a random index i_t
  $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \alpha_t \nabla f_{i_t}(\mathbf{w}^{(t)})$

1.4.3 Coordinate Descent

Algorithm 4: Coordinate Descent

Initialize $\mathbf{w}^{(0)}$ to a random point
while $f(\mathbf{w}^{(t)})$ *not converged* **do**
  Sample a random feature i_t
  $w_{i_t}^{(t+1)} \leftarrow w_{i_t}^{(t)} - \alpha_t \frac{\partial}{\partial w_{i_t}} f(\mathbf{w}^{(t)})$

1.5 Newton's Method

Up until this point, we have only considered first order methods to optimize functions. Now, we will present **Newton's method**, an iterative method that utilizes second order information to achieve a faster rate of convergence than existing first-order methods. Given an arbitrary twice continuously differentiable objective function f, Newton's Method minimizes f by iteratively setting the first-order Taylor expansion of the gradient to zero, or equivalently minimizing the second-order Taylor expansion of the objective function.

Newton's method is an example of iterative *root-finding algorithms* — methods that estimate the root of a function by iteratively approximating the function and finding the root of the approximation. To illustrate the point, consider a single variable function $\varphi\colon \mathbb{R} \to \mathbb{R}$. Our goal is to find a root of the non-linear equation $\varphi(w) = 0$. Suppose we have a current estimate of the root $\varphi(w^{(t)})$. From Taylor's theorem, we can express the first-order form of $\varphi(w)$ with respect to $\varphi(w^{(t)})$ as

$$\varphi(w) = \varphi(w^{(t)}) + \varphi'(w) \cdot (w - w^{(t)}) + o(\left|w - w^{(t)}\right|)$$

given $\delta = w - w^{(t)}$ we equivalently have that

$$\varphi(w^{(t)} + \delta) = \varphi(w^{(t)}) + \varphi'(w) \cdot \delta + o(|\delta|)$$

Disregarding the $o(|\delta|)$ term, we solve (over δ) the following objective:

$$\varphi(w^{(t)}) + \varphi'(w^{(t)})\delta = 0$$

Then, $\delta = -\frac{\varphi(w^{(t)})}{\varphi'(w^{(t)})}$, leading to the iteration $w^{(t+1)} = w^{(t)} - \frac{\varphi(w^{(t)})}{\varphi'(w^{(t)})}$. We can similarly make an argument for a multi variable function $F : \mathbb{R}^d \to \mathbb{R}^k$. Our goal is to solve $F(\mathbf{w}) = \mathbf{0}$. Again, from Taylor's theorem we have that

$$F(\mathbf{w} + \Delta) = F(\mathbf{w}) + J_F(\mathbf{w})\Delta + o(\|\Delta\|)$$

where J_F is the Jacobian. This gives us $\Delta = -J_F^{-1}(\mathbf{w})F(\mathbf{w})$, and the iteration

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - J_F^{-1}(\mathbf{w}^{(t)})F(\mathbf{w}^{(t)})$$

Newton's method is just a special application of this root-finding method, applied to the gradient function. That is, given that we are minimizing $f\colon \mathbb{R}^d \to \mathbb{R}$, Newton's method finds the roots of the gradient function $\nabla f\colon \mathbb{R}^d \to \mathbb{R}^d$. It uses the update rule

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \nabla^2 f(\mathbf{w}^{(t)})^{-1}\nabla f(\mathbf{w}^{(t)})$$

as the Hessian $\nabla^2 f(\mathbf{w}^{(t)})$ of the objective function corresponds to the Jacobian $J_F^{-1}(\mathbf{w})$ of the gradient. The algorithm is as follows:

Algorithm 5: Newton's Method

Initialize $\mathbf{w}^{(0)}$ to a random point
while $f(\mathbf{w}^{(t)})$ *not converged* **do**
  $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \nabla^2 f(\mathbf{w}^{(t)})^{-1}\nabla f(\mathbf{w}^{(t)})$

The motivation for Newton's method is as follows: our goal is to find local minima for f, points for which it is necessarily true that $\nabla f(\mathbf{w}) = \mathbf{0}$. Consequently, we wish to find points for which $\nabla f(\mathbf{w}) = \mathbf{0}$. The gradient $\nabla f(\mathbf{w})$ can be difficult or even intractable to work with, so instead we work with a first-order Taylor approximation of the gradient with respect to our current iterate $\mathbf{w}^{(t)}$. We solve for the roots of the first-order gradient, update our iterate, and repeat the process. Note that while solving $\nabla f(\mathbf{w}) = \mathbf{0}$ may yield local maxima or even saddle points, we are finding the roots of the linearized gradient, which is convex — therefore any point for which the first-order approximation of the gradient is zero yields a global minimum for the approximation.

A Newton step equivalently minimizes the second order Taylor approximation w.r.t. $\mathbf{w}^{(t)}$:

$$f(\mathbf{w}) \approx \bar{f}(\mathbf{w}) = f(\mathbf{w}^{(t)}) + \nabla f(\mathbf{w}^{(t)})^\top(\mathbf{w} - \mathbf{w}^{(t)}) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^{(t)})^\top\nabla^2 f(\mathbf{w}^{(t)})(\mathbf{w} - \mathbf{w}^{(t)})$$