Introduction to gradient descent

- Derivation and intuitions
- Hessian

Introduction

- Our goal in machine learning is to optimize an objective function, $f(x)$. (Without loss of generality, we'll consider minimizing $f(x)$. This is equivalent to maximizing $-f(x)$.)
- From basic calculus, we recall that the derivative of a function, $\frac{df(x)}{dx}$ tells us the slope of $f(x)$ at point x.
  - For small enough ϵ, $f(x+\epsilon) \approx f(x) + \epsilon f'(x)$.
  - This tells us how to reduce (or increase) $f(\cdot)$ for small enough steps.
  - Recall that when $f'(x) = 0$, we are at a stationary point or critical point. This may be a local or global minimum, a local or global maximum, or a saddle point of the function.
- In this class we will consider cases where we would like to maximize f w.r.t. vectors and matrices, e.g., $f(\mathbf{x})$ and $f(\mathbf{X})$.
- Further, often $f(\cdot)$ contains a nonlinearity or non-differentiable function. In these cases, we can't simply set $f'(\cdot) = 0$, because this does not admit a closed-form solution.
- However, we can iteratively approach an critical point via gradient descent.

Terminology

- A **global minimum** is the point, $\mathbf{x}_g$, that achieves the absolute lowest value of $f(\mathbf{x})$. i.e., $f(\mathbf{x}) \geq f(\mathbf{x}_g)$ for all $\mathbf{x}$.
- A **local minimum** is a point, $\mathbf{x}_\ell$, that is a critical point of $f(\mathbf{x})$ and is lower than its neighboring points. However, $f(\mathbf{x}_\ell) > f(\mathbf{x}_g)$.
- Analogous definitions hold for the **global maximum** and **local maximum**.
- A **saddle point** are critical point of $f(\mathbf{x})$ that are not local maxima or minima. Concretely, neighboring points are both greater than and less than $f(\mathbf{x})$.

Gradient

Recall the gradient, $\nabla_{\mathbf{x}} f(\mathbf{x})$, is a vector whose ith element is the partial derivative of $f(\mathbf{x})$ w.r.t. x_i, the ith element of $\mathbf{x}$. Concretely, for $\mathbf{x} \in \mathbb{R}^n$,

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial (x_n)} \end{bmatrix}$$

- The gradient tells us how a small change in $\Delta\mathbf{x}$ affects $f(\mathbf{x})$ through

$$f(\mathbf{x} + \Delta\mathbf{x}) \approx f(\mathbf{x}) + \Delta\mathbf{x}^T \nabla_{\mathbf{x}} f(\mathbf{x})$$

- The directional derivative of $f(\mathbf{x})$ in the direction of the unit vector $\mathbf{u}$ is given by:

$$\mathbf{u}^T \nabla_{\mathbf{x}} f(\mathbf{x})$$

- The directional derivative tells us the slope of f in the direction $\mathbf{u}$.

Arriving at gradient descent

- To minimize $f(\mathbf{x})$, we want to find the direction in which $f(\mathbf{x})$ decreases the fastest. To do so, we find the direction $\mathbf{u}$ which minimizes the directional derivative.

$$\begin{aligned} \min_{\mathbf{u}, \|\mathbf{u}\|=1} \mathbf{u}^T \nabla_{\mathbf{x}} f(\mathbf{x}) &= \min_{\mathbf{u}, \|\mathbf{u}\|=1} \|\mathbf{u}\| \, \|\nabla_{\mathbf{x}} f(\mathbf{x})\| \cos\theta \\ &= \min_{\mathbf{u}} \|\nabla_{\mathbf{x}} f(\mathbf{x})\| \cos(\theta) \end{aligned}$$

  where θ is the angle between the vectors $\mathbf{u}$ and $\nabla_{\mathbf{x}} f(\mathbf{x})$.
- This quantity is minimized for $\mathbf{u}$ pointing in the opposite direction of the gradient, so that $\cos(\theta) = -1$.
- Hence, we arrive at gradient descent. To update $\mathbf{x}$ so as to minimize $f(\mathbf{x})$, we repeatedly calculate:

$$\mathbf{x} := \mathbf{x} - \epsilon \nabla_x f(\mathbf{x})$$

- ϵ is typically called the *learning rate*. It can change over iterations. Setting the value of ϵ appropriately is an important part of deep learning.

Hessian

Recall Newton's method determines the step size by considering the *curvature* of the function, which is the second derivative.

- The generalization of the second derivative of a scalar-valued function, $f(\mathbf{x})$, is the Hessian.
- The Hessian, $\mathbf{H}$, is a matrix whose (i, j)th element is the second derivative of $f(\mathbf{x})$ w.r.t. x_i and x_j. That is, the (i, j)th element of $\mathbf{H}$, denoted H_{ij}, is

$$H_{ij} = \frac{\partial^2}{\partial x_i \partial x_j} f(\mathbf{x})$$

The matrix, $\mathbf{H} \in \mathbb{R}^{n \times n}$ is:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_n} \end{bmatrix}$$

Hessian (cont.)

A few notes on the Hessian:

- Recall in the mathematical tools notes that we can denote the Hessian as

$$\nabla_x f^2(\mathbf{x})$$

- If the partial second derivatives are continuous, then ∂ is commutative, and the Hessian is thus symmetric.

Directional second derivative

- The directional second derivative along unit vector $\mathbf{u}$ is given by $\mathbf{u}^T \mathbf{H} \mathbf{u}$.
- As $\mathbf{H}$ in general is symmetric and thus has a set of real eigenvalues with an orthogonal basis of eigenvectors, if $\mathbf{u}$ points in the direction of an eigenvector of $\mathbf{H}$, the directional second derivative is the corresponding eigenvalue, λ.
- The maximum directional second derivative is λ_{max} and the minimum is λ_{min}.

The second derivative

- The 2nd derivative of $f(\mathbf{x})$ is a measure of the curvature of the function at $\mathbf{x}$.
- If $\mathbf{H}$ is positive definite at a critical point, then $\mathbf{x}$ is a local minimum. This is because the directional second derivative is positive, and so in every direction, the function curves upwards.
- If $\mathbf{H}$ is negative definite at a critical point, then $\mathbf{x}$ is a local maximum.
- If $\mathbf{H}$ has both positive and negative eigenvalues, it is a saddle point.
- Intuitively, if $f(\mathbf{x})$ is relatively flat around $\mathbf{x}$, we would like to take large steps along the gradient; if it is very curved, we would like to take small steps. However, incorporating the Hessian ought to enable taking better steps in each dimension.
- This is called **Newton's method**.

Newton's method

- We seek to find the step, $\Delta\mathbf{x}$ that minimizes the second order Taylor expansion of $f(\mathbf{x} + \Delta\mathbf{x})$.

$$f(\mathbf{x} + \Delta\mathbf{x}) \approx f(\mathbf{x}) + \nabla_{\mathbf{x}} f(\mathbf{x})^T \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T \mathbf{H} \Delta\mathbf{x}$$

- Taking the derivative w.r.t. $\Delta\mathbf{x}$ and setting it equal to zero, we arrive at:

$$\nabla_{\mathbf{x}} f(\mathbf{x}) + \mathbf{H} = 0$$

- Hence, we arrive at the optimal Newton step:

$$\Delta\mathbf{x} = -\mathbf{H}^{-1}\nabla_{\mathbf{x}} f(x)$$

- This method of optimization is called a **second-order** optimization algorithm because it uses the second derivative.

Limitations of Newton's method

- While Newton's method may converge far more quickly than gradient descent, it may also locate saddle points.
- Storing the Hessian can be expensive.
- Inverting the Hessian can be even more expensive.
- Typically deep learning does not use Newton's method, although there are second-order techniques that don't require computing and inverting the Hessian.
- We'll talk about this more in the optimization for neural networks slides.

Lipschitz continuity

- In deep learning, we typically deal with functions that are **Lipschitz continuous**.
- A function f is Lipschitz continuous if for a Lipschitz constant ℓ,

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq \ell \,\|\mathbf{x} - \mathbf{y}\|$$

  for all $\mathbf{x}$ and $\mathbf{y}$.
- This condition means that a small input made in gradient descent will have a small change on the output.