Calculus for ML

1 Gradients, Jacobians, and Hessians

Recall from lecture the following:

- The *gradient* of a scalar-valued function $f: \mathbb{R}^n \to \mathbb{R}$ is a column vector of length n, denoted as ∇f, containing the derivatives of components of f with respect to the input variables:

$$((\nabla f(x))_i = \frac{\partial f}{\partial x_i}(x), \quad i = 1, \ldots, n. \tag{1}$$

- The *Hessian* of a scalar-valued function $f: \mathbb{R}^n \to \mathbb{R}$ is an $n \times n$ matrix, denoted as $\nabla^2 f$, containing the second derivatives of components of f with respect to the input variables:

$$(\nabla^2 f(x))_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}, \quad i = 1, \ldots, n, \quad j = 1 \ldots, n. \tag{2}$$

- The *Jacobian* (or *Derivative*) of a vector-valued function $f: \mathbb{R}^n \to \mathbb{R}^m$ is an $m \times n$ matrix, denoted as $\frac{\partial f}{\partial x}$, containing the derivatives of components of f with respect to the input variables:

$$\left(\frac{\partial f}{\partial x}\right)_{ij} = \frac{\partial f_i}{\partial x_j}, \quad i = 1, \ldots, m, \quad j = 1 \ldots, n. \tag{3}$$

For the remainder of the class, we will repeatedly have to take gradients, Hessians, and derivatives of functions.

Furthermore, we outline two strategies one can use to answer the following parts. To illustrate this, we'll use the following example: $f(x) = w^\top x$

1. *Using computation via first principle.* In class we learned Taylor's Theorem which states: For a scalar function $f: \mathbb{R}^n \to \mathbb{R}$ around a point x_0, the first-order (linear) approximation is

$$\widehat{f_i}(x; x_0) = f(x_0) + \frac{\partial f}{\partial x}(x_0)(x - x_0). \tag{4}$$

   Let $x_0 = x + \Delta$, where Δ is tiny. Then, the first-order (linear) approximation of f around $x + \Delta$ is

$$f(x) = f(x + \Delta) + \frac{\partial f}{\partial x}(x + \Delta)(-\Delta). \tag{5}$$

   Rearranging gives us:

$$f(x + \Delta) = f(x) + \frac{\partial f}{\partial x}(x + \Delta)(\Delta). \tag{6}$$

   Via pattern matching, we see that

$$f(x + \Delta) = f(x) + (\textit{something})\Delta, \tag{7}$$

   where *something* is our derivative!

   Now, we use $f(x) = w^\top x$. Then we have

$$f(x + \Delta) = w^\top (x + \Delta) = w^\top x + w^\top \Delta = f(x) + w^\top \Delta. \tag{8}$$

   Comparing with equation (7), we conclude that

$$\frac{\partial f}{\partial x} = w^\top \quad \text{and, thus,} \quad \nabla f(x) = w. \tag{9}$$

2. *Using the formula.* The idea is to build up the gradient/derivative one component at a time. For $f(x) = w^\top x$, we have that $\sum_j w_j x_j$. Hence, we have

$$\frac{\partial f}{\partial x_i} = w_i. \tag{10}$$

Thus, we find that

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = w. \tag{11}$$

And $\frac{\partial f}{\partial x} = w^\top$.

For the following parts, before taking any gradients/Hessians/derivatives, identify what the gradient/Hessian/derivative looks like (is it a scalar, vector, or matrix?) and how we calculate each term in the gradient/Hessian/derivative. Then carefully solve for an arbitrary entry of the gradient/Hessian/derivative, then stack/arrange all of them to get a final result.

For the first two parts, suppose $X \in \mathbb{R}^{n \times n}$ is a square matrix whose entries are denoted X_{ij} and whose rows are denoted $X_1^\top, \ldots, X_n^\top$ and $b \in \mathbb{R}^n$ is a vector whose entries are denoted b_i.

(a) Compute the Jacobians for the following functions.

i. $f(w) = Xw$.

ii. $g(w) = f(w)w$ where $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable.

(b) Compute the gradient and Hessian for each of the following functions.

i. $h(w) = w^\top Xw$

ii. $l(w) = \|w\|_2^2$.

(c) Given $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $z = q(x)$, and $p(z) = z^\top z$, find the gradient $\nabla_z p(z)$, then find the gradient $\nabla_x p(z)$ in terms of $\frac{\partial z}{\partial x}$ and z.

2 Taylor's Theorem

In class, we learned that we can approximate any function $f(x)$ near a point x_0 using polynomials. This concept is incredibly useful in Deep Learning when we interpret how the parameters (input variables) of our model (function f) change after one step of gradient descent. Don't worry if you don't understand what this means–you will soon!

For the following problem, plot/hand draw the level sets of the function. Also, point out the gradient directions in the level-set diagram. Additionally, compute the first and second order Taylor series approximation around the point $x_0 = (1, 1)$ for the function and comment on how accurately they approximate the true function.

(a) $g(x_1, x_2) = \frac{x_1^2}{4} + \frac{x_2^2}{9}$

3 Least Squares

In our Linear Algebra for ML lecture, you learned about the following optimization problem:

$$w^* = \underset{w}{\operatorname{argmin}} \|y - Xw\|_2^2. \tag{12}$$

where $X \in \mathbb{R}^{n \times d}$ is a full-rank data matrix and $y \in \mathbb{R}^n$ is the target vector of measurement values.

In this problem, we will learn how to solve this using Calculus.

(a) Let $\mathcal{L}(w) = \|y - Xw\|_2^2$ be our objective function. Expand $\mathcal{L}$.

(b) Find the gradient of $\mathcal{L}$ *with respect to* w.

(c) Use the Main Theorem to find the optimal w^*. It is important to note that finding w^* in an arbitrary optimization problem by this method may not guarantee a minimum. However, the Least Squares problem has some nice properties that allows us to use this theorem directly.