Introduction to Machine Learning (67577)

Recitation 02
Multivariate Calculus

Second Semester, 2021

Contents

2 Multivariate Calculus

Often when considering different machine learning techniques we will consider high dimensional functions $f : \mathbb{R}^d \to \mathbb{R}$. Usually, these functions will represent some distance measurement between our estimated prediction and the true values. As such, we would like to solve the optimization problem of minimizing this distance. Namely:

$$\underset{\mathbf{w}\in\mathbb{R}^d}{argmin}\, f(\mathbf{w})$$

2.1 Derivatives, Gradients and Jacobians

2.1.1 Derivatives

A common approach for finding such a minimizer is by computing the derivative of the function, equating to zero and solving for our parameters $\mathbf{w}$. When discussing multivariate functions we use gradients rather than scalar derivatives.

Definition 2.1 — Derivative. Let $f : \mathbb{R} \to \mathbb{R}$. The derivative of f at point $x \in \mathbb{R}$ is defined as

$$\frac{d}{dx} f(x) := \lim_{a\to 0} \frac{f(x+a) - f(x)}{a}$$

■ **Example 2.1 — ReLU.** Consider the **R**ectified **L**inear **U**nit function defined as the positive part of its argument: $f(x) = x^+ = \max(0,x)$. This function is in common use in the context of artificial neural networks. The derivative if this function is:

$$\frac{df(x)}{dx} = \begin{cases} 0 & x < 0 \\ 1 & x > 0 \end{cases}$$

■

Note that at $x = 0$ the derivative of ReLU is undefined. We will discuss ways to handle such cases in **??**. Next, let us expand the derivative definition to multivariate functions.

Definition 2.2 — Partial Derivative. Let $f : \mathbb{R}^d \to \mathbb{R}$. The partial derivative of f at point $\mathbf{x} \in \mathbb{R}^d$ with respect to x_i is defined as

$$\begin{aligned} \frac{\partial}{\partial x_i} f(\mathbf{x}) &:= \lim_{a\to 0} \frac{f(\mathbf{x} + a\mathbf{e}_i) - f(\mathbf{x})}{a} \\ &= \lim_{a\to 0} \frac{f(x_1,\ldots,x_i+a,\ldots,x_d) - f(x_1,\ldots,x_i,\ldots,x_d)}{a} \end{aligned}$$

where $\mathbf{e}_i$ is the i-th standard basis vector.

Namely, a partial derivative of a function is its derivative with respect to one of its variables, while all others are kept constant.

■ **Example 2.2 — Max.** For $f(\mathbf{x}) = max_i(x_1,\ldots,x_d)$ the partial derivatives of f at x_i are:

$$\frac{\partial}{\partial x_i} f(\mathbf{x}) = \begin{cases} 1 & i = argmax(x_1,\ldots,x_d) \\ 0 & i \neq argmax(x_1,\ldots,x_d) \end{cases}$$

■

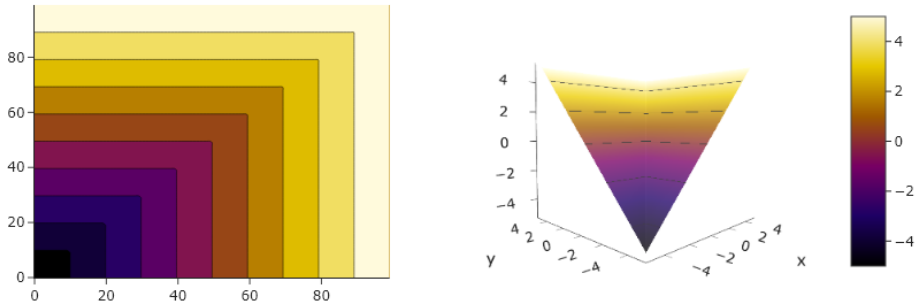

Figure 1: *Visualization of bi-variate max function in 3D and 2D*

■ **Example 2.3 — Polynomial.** For $f(x,y) = x^2 + xy + y^2$ the partial derivatives of f at (x_0, y_0) are

$$\frac{\partial}{\partial x} f(x_0, y_0) = 2x_0 + y_0, \quad \frac{\partial}{\partial y} f(x_0, y_0) = 2y_0 + x_0$$

■

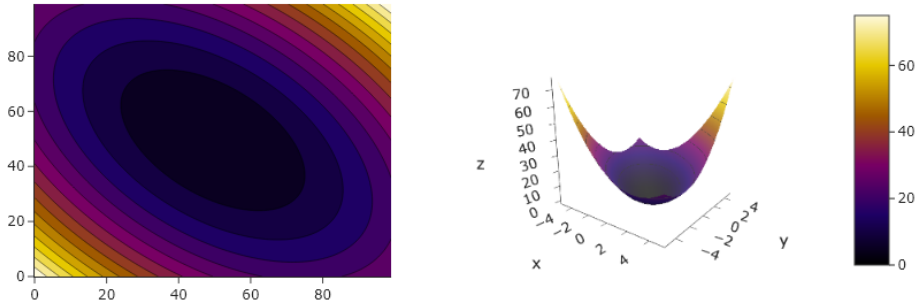

Figure 2: *Visualization of bi-variate polynomial of degree 2 in 3D and 2D*

2.1.2 Gradients

Definition 2.3 — Gradient. The gradient of f at $\mathbf{x}$ is the vector of partial derivatives:

$$\nabla f(\mathbf{x}) := \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_d} \right)^\top$$

■ **Example 2.4** By using the partial derivatives previously calculated of a second order bivariate polynomial $f((x,y) = x^2 + xy + y^2$, the gradient of f at $\mathbf{x}_0 = (x_0, y_0)$ is

$$\nabla f(\mathbf{x}_0) = (2x_0 + y_0, 2y_0 + x_0)^\top.$$

■

Exercise 2.1 — Linear Functional. Let $\mathbf{w} \in \mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}$ defined as $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$. Compute the gradient of f at point $\mathbf{x}$. ■

Proof. From linearity of the derivative:

$$\frac{\partial}{\partial x_j} f(\mathbf{x}) = \sum_i \frac{\partial}{\partial x_j} f(\mathbf{x})_i = \sum \frac{\partial}{\partial x_j} \mathbf{w}_i \mathbf{x}_i = \mathbf{w}_j$$

Therefore, in vector notation $\nabla f(\mathbf{x}) = \mathbf{w}$. ■

Exercise 2.2 — Norm. Let $f : \mathbb{R}^d \to \mathbb{R}$ be defined by $f(\mathbf{x}) = ||\mathbf{x}||^2$. Compute the gradient of f at point $\mathbf{x}$. ■

Proof. Similar to the previous exercise, using the linearity of the derivative then:

$$\frac{\partial}{\partial x_j} f(x) = \sum_i \frac{\partial}{\partial x_j} x_i^2 = 2x_j$$

which in vector notation can be written as: $\nabla f(\mathbf{x}) = 2\mathbf{x}$. ■

2.1.3 Jacobians

In different applications in machine learning, the function we are trying to minimize is a vector-valued function, namely $f : \mathbb{R}^m \to \mathbb{R}^d$.

For example, consider the following scenario. We have gathered historic information of season, time of day, latitude and longitude at different points in the world. Next, we have described some function that given these parameters states the temperature and air-pressure of that location and time. We want to find the the extrema points of this function. In this scenario, the described function gets fours parameters and returns two outputs $f : \mathbb{R}^4 \to \mathbb{R}^2$. In order to find the extrema points we need to generalize the gradient definition to vector-valued functions.

Definition 2.4 — Jacobian. Let $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^d$ where $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_d(\mathbf{x}))^\top$. The Jacobian of f is the $d \times m$ matrix of all partial derivatives:

$$J_{\mathbf{x}}(\mathbf{f}) := \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_m} \\ \vdots & & \vdots \\ \frac{\partial f_d(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_d(\mathbf{x})}{\partial x_m} \end{bmatrix}$$

■ **Example 2.5** Let us revisit 2.4 where $f : \mathbb{R}^2 \to \mathbb{R}$ defined as $f(\mathbf{x}) = x_1^2 + x_2^2$. The Jacobian of f is:

$$J_x(\mathbf{f}) = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} \end{bmatrix} = [2x_1, 2x_2] = \nabla f(\mathbf{x})^\top$$

■

Notice that for any function where $k = 1$ the Jacobian is infact the transposed gradient vector: $J_{\mathbf{x}}(f) = \nabla f(\mathbf{x})^\top$.

Exercise 2.3 Let $A \in \mathbb{R}^{m \times d}$ and let $f : \mathbb{R}^d \to \mathbb{R}^m$ defined as $f(\mathbf{x}) = A\mathbf{x}$. Find the Jacobian of f: $J_{\mathbf{x}}(f)$). ■

Proof. Let us define the set of functions computing each coordinate in the output vector $\forall i \in [d] \quad f_i(\mathbf{x}) = A_i^\top \mathbf{x}$. Then the jacobian of f is comprised of the gradients of $f_1, \ldots, f_d$ as rows. Notice that we have already computed the gradient of linear functionals in 2.1 so:

$$J_{\mathbf{x}}(f) = \begin{bmatrix} \nabla f_1(\mathbf{x})^\top \\ \vdots \\ \nabla f_d(\mathbf{x})^\top \end{bmatrix} = \begin{bmatrix} -A_1- \\ \vdots \\ -A_d- \end{bmatrix} = A$$

■

2.1.4 Chain Rules

Theorem 2.1 — Chain Rule - Univariate. Let $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ be two differential functions, then the derivative of the composite $f \circ g$ is:

$$(f \circ g)' := (f' \circ g) \cdot g'$$

Namely, for $h(x) = f(g(x))$ then $\forall x \in \mathbb{R}\ h'(x) = f'(g(x)) \cdot g'(x)$.

Theorem 2.2 — Chain Rule - Multivariate. Let $f : \mathbb{R}^d \to \mathbb{R}^m$ and $g : \mathbb{R}^k \to \mathbb{R}^d$. The Jacobian of the composition $(f \circ g) : \mathbb{R}^k \to \mathbb{R}^m$ at $\mathbf{x}$ is

$$J_{\mathbf{x}}(f \circ g) = J_{g(\mathbf{x})}(f) J_{\mathbf{x}}(g) := \begin{bmatrix} \frac{\partial f_1(g(\mathbf{x}))}{\partial g_1(\mathbf{x})} & \cdots & \frac{\partial f_1(g(\mathbf{x}))}{\partial g_d(\mathbf{x})} \\ \vdots & & \vdots \\ \frac{\partial f_m(g(\mathbf{x}))}{\partial g_1(\mathbf{x})} & \cdots & \frac{\partial f_m(g(\mathbf{x}))}{\partial g_d(\mathbf{x})} \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial g_1(\mathbf{x})}{\partial x_k} \\ \vdots & & \vdots \\ \frac{\partial g_d(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial g_d(\mathbf{x})}{\partial x_k} \end{bmatrix}$$

In element-wise notation:

$$J_{\mathbf{x}}(f \circ g)_{i,j} := \sum_l \frac{\partial f_i(g(\mathbf{x}))}{\partial g_l(\mathbf{x})} \frac{\partial g_l(\mathbf{x})}{\partial x_j}$$

Exercise 2.4 Let $f(\mathbf{x}) = ||\mathbf{x}||^2$ and $g(\mathbf{x}) = A\mathbf{x}$ for some matrix $A \in \mathbb{R}^{m \times d}$. Calculate the jacobian of $f \circ g$. ■

Proof. From the previous theorem we know that:

$$J_{\mathbf{x}}(f \circ g) = J_{g(\mathbf{x})}(f) J_{\mathbf{x}}(g)$$

As g a linear transformation we have see in 2.3 that $J_{\mathbf{x}}(g) = A$. Notice, that as $Im(f) \subseteq \mathbb{R}$, the jacobian of f equals to the transpose of its grad ient. As seen in 2.2, $J_{g(\mathbf{x})}(f) = (2g(\mathbf{x}))^\top$. Therefore:

$$J_{\mathbf{x}}(f \circ g) = 2\mathbf{x}^\top A^\top A$$

■

Next, let us calculate the gradient of the following function: $f(\mathbf{x}) = \frac{1}{2}||A\mathbf{x}-\mathbf{y}||^2$. Applying the chain rule then:

$$\begin{aligned}\nabla f(\mathbf{x}) &= \frac{\partial}{\partial \mathbf{x}}\frac{1}{2}||A\mathbf{x}-\mathbf{y}||^2 \\ &= \frac{1}{2}\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^\top A^\top A\mathbf{x} - 2\mathbf{y}^\top A\mathbf{x} + \mathbf{y}^\top y\right) \\ &= A^\top A\mathbf{x} - A^\top y \\ &= A^\top(A\mathbf{x}-\mathbf{y})\end{aligned}$$

This function is known as the Mean Square Error (MSE) and in **??** we will use the above derivation in order to find the values of $\mathbf{x}$ that minimize f as above.

■ **Example 2.6 — Soft-Max.** The softmax function defined over $S : \mathbb{R}^d \to [0,1]^d$ returns a vector that its coordinates sum up to 1. It is defined by

$$S(\mathbf{a})_j = \frac{e^{a_j}}{\sum_{k=1}^N e^{a_k}}$$

As the coefficients a_j are in the power of the exponent function, all outputted values are strictly positive. Moreover, since the numerator appears in the denominator summed up with some other positive numbers, $S_j < 1$. Therefore, it is in the range $(0,1)$. For example, the 3-element vector $(1,2,5)$ gets transformed into $(0.02, 0.05, 0.93)$. The order of elements by relative size is preserved, and they add up to 1.0. In addition, the third element is now farther away from the first two, Its softmax value is dominating the overall slice of size 1.0 in the output.

Intuitively, the $softmax$ function is a "soft" version of the $argmax$ function. Instead of just selecting one maximal element, softmax breaks the vector up into segments with the maximal input element getting a proportionally larger chunk, but the other elements getting some of it as well. Softmax is often used in neural networks, to map the non-normalized output to a probability distribution over predicted output classes.

Let us calculate the derivative of the softmax function. Denote $g_{i(\mathbf{a})} \coloneqq e^{a_i}$ and $h(\mathbf{a}) \coloneqq \sum_{k=1}^N g_i(\mathbf{a})$. So:

$$\frac{\partial S_i}{\partial a_j} = \frac{\partial}{\partial a_j}\frac{e^{a_i}}{\sum_{k=1}^N e^{a_k}} = \frac{\partial}{\partial a_j}\frac{g_i}{h}$$

Note that for any a_j the derivative of h is e^{a_j}. In the case of g_i, when deriving with respect to a_j we get that the derivative is e^{a_j} only if $i = j$. Otherwise, the derivative is 0. Therefore, the derivative of S_i in the case where $i = j$ is:

$$\frac{\partial}{\partial a_j}\frac{e^{a_i}}{\sum_{k=1}^N e^{a_k}} = \frac{e^{a_i}(\sum_{k=1}^N e^{a_k}) - e^{a_i}e^{a_j}}{(\sum_{k=1}^N e^{a_k})^2} = \frac{e^{a_i}}{(\sum_{k=1}^N e^{a_k})}\cdot\frac{(\sum_{k=1}^N e^{a_k}) - e^{a_j}}{(\sum_{k=1}^N e^{a_k})} = S_i(1-S_j)$$

It is left to show the derivative in the case where $i \neq j$. ■

2.2 First Order Function Approximation

As motivated in the begining of this section, we are often interested in finding the minimizers of some multivariate objective function. Many times, these functions are very hard, or even impossible, to solve analytically. In such cases we might consider approximating the true function with a simpler one that we are able to solve.

Consider a function $f:\mathbb{R}\to\mathbb{R}$ and recall the definition of the Taylor series:

$$T(x_0+x)=\sum_{n=0}^{\infty}\frac{f^{(n)}(x_0)}{n!}x^n=f(x_0)+f'(x_0)x+\frac{1}{2}f''(x_0)x^2+\dots$$

A linear approximation (or first order approximation) is an approximation of a general function using a linear function. For a twice continuously differentiable function $f:\mathbb{R}\to\mathbb{R}$, Taylor's theorem implies that for small x

$$f(x_0+x)\approx f(x_0)+f'(x_0)x$$

We can now extend this theorem to define linear approximations of multivariate functions.

Definition 2.5 — Linear Approximation. Let $f:\mathbb{R}^d\to\mathbb{R}$ and $\mathbf{p}_0\in\mathbb{R}^d$. The linear approximation of f for every $\mathbf{p}$ near $\mathbf{p}_0$ is defined as

$$f(\mathbf{p}_0)+\langle\nabla f(\mathbf{p}_0),\mathbf{p}-\mathbf{p}_0\rangle$$

Equivalently, if we treat $\mathbf{p}$ as the deviation from $\mathbf{p}_0$ then:

$$f(\mathbf{p}_0+\mathbf{p})\approx f(\mathbf{p}_0)+\langle\nabla f(\mathbf{p}_0),\mathbf{p}\rangle$$

So if for example, we consider a bivariate function $f:\mathbb{R}^2\to\mathbb{R}$, the linear approximation of f near the point (x_0,y_0) is:

$$f(x_0+x,y_0+y)\approx f(x_0,y_0)+x\cdot\frac{\partial f(x_0,y_0)}{\partial x}+y\cdot\frac{\partial f(x_0,y_0)}{\partial y}$$

Now, if f is a linear function, intuition dictates that the linear approximation of f would be the function itself. Let $\mathbf{b}\in\mathbb{R}^d$ and let $f:\mathbb{R}^d\to\mathbb{R}$ be defined by $f(\mathbf{x})=\mathbf{b}^\top\mathbf{x}$. Then, the linear approximation of f at $\mathbf{p}_0$ is

$$\begin{aligned}f(\mathbf{p}_0)+\langle\nabla f(\mathbf{p}_0),\mathbf{p}-\mathbf{p}_0\rangle &= \mathbf{b}^\top\mathbf{p}_0+\langle\mathbf{b},\mathbf{p}-\mathbf{p}_0\rangle\\ &= \mathbf{b}^\top(\mathbf{p}_0+\mathbf{p}-\mathbf{p}_0)=f(\mathbf{p})\end{aligned}$$

Exercise 2.5 Let $f:R^2\to\mathbb{R}$ be defined by $f(x,y)=\sqrt{x^2+y^2}$. Calculate the linear approximation of f near $(3,4)$. ■

Proof. We begin with expressing the gradient of f. So, the partial derivative of f with respect to first argument at point x is:

$$\frac{\partial}{\partial x}=f(x_0,y_0)=2x_0\cdots\frac{1}{2\sqrt{x_0^2+y_0^2}}$$

Therefore the gradient of f is $\nabla f(\mathbf{x})=\left(\frac{x_0}{\sqrt{x_0^2+y_0^2}},\frac{y_0}{\sqrt{x_0^2+y_0^2}}\right)^\top$. So for a point (x,y) in the vicinity of $(3,4)$ the linear approximation is:

$$f(3+x,4+y) \approx 5+\frac{3}{5}x+\frac{4}{5}y$$

If for example $x=0.1, y=0.2$ then $f(3+0.1,4+0.2)=5.2201.. \approx 5.22 = 5+3/5\cdot 0.1+4/5\cdot 0.2$ ■

Another use of first order approximations is as follows. Suppose we investigate some objective function $f:\mathbb{R}^d \to \mathbb{R}$ at point $\mathbf{p}_0 \in \mathbb{R}^d$. Now, we would like to "take a step" in $\mathbb{R}^d$ in the direction where f grows with the fastest rate. That is, we are searching for the direction in space, that if we move in, f grows the most. How can we find such direction?

Recall that $f(\mathbf{p}_0+\mathbf{p}) \approx f(\mathbf{p}_0)+\nabla f(\mathbf{p}_0)\cdot\mathbf{p}$. In addition, the angle between $\nabla f(\mathbf{p}_0)$ and $\mathbf{p}$ is $\nabla f(\mathbf{p}_0)\cdot\mathbf{p} = ||\nabla f(\mathbf{p}_0)||\cdot||\mathbf{p}||\cos\theta$. As we are only interested in the direction in which to step, and not the step size, let us assume that $||(\mathbf{p})|| = 1$. Since $\cos x \in [-1,1]$, the direction that maximizes f is $\mathbf{p}_{max} \coloneqq \nabla f(\mathbf{p}_0)/||\nabla(\mathbf{p}_0)||$. Similarly the direction that minimizes f is $\mathbf{p}_{min} \coloneqq -\nabla f(\mathbf{p}_0)/||\nabla(\mathbf{p}_0)||$. So, if we adapt a step-wise approach, for sufficiently small enough steps sizes, for maximizing/minimizing f, moving in the direction of the gradient or opposite to the gradient is a good approach. We will revisit this concept in **??**