1 Derivatives

Optimization is the most essential ingredient in the recipe of machine learning algorithms. It starts with defining some kind of loss function/cost function and ends with minimizing it using one or the other optimization routine. The optimization routines involve computation of derivatives at some point or the other. The types of derivatives that we will be working with in this class are:

- Derivative of a scalar with respect to a vector
- Derivative of a scalar with respect to a matrix
- Derivative of a vector with respect to a vector
- Derivative of a vector with respect to a matrix

In this discussion, we will focus on the first three types and in a future discussion we will cover the fourth type (Tensor derivatives).

1.1 Derivative of a scalar with respect to a vector

Let $y \in \mathbb{R}$ be a scalar and $\mathbf{x} \in \mathbb{R}^n$ be a vector, then the derivative of y with respect to $\mathbf{x}$ is defined as

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

Using the denominator layout, the dimension is given by

$$\nabla_{\mathbf{x}} y \in \mathbb{R}^{n \times 1}$$

1.2 Derivative of a scalar with respect to a matrix

Let $y \in \mathbb{R}$ be a scalar and $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a matrix, then the derivative of y with respect to $\mathbf{A}$ is defined as

$$\nabla_{\mathbf{A}} y = \begin{bmatrix} \frac{\partial y}{\partial a_{11}} & \frac{\partial y}{\partial a_{12}} & \cdots & \frac{\partial y}{\partial a_{1n}} \\ \vdots & & & \\ \frac{\partial y}{\partial a_{m1}} & \frac{\partial y}{\partial a_{m2}} & \cdots & \frac{\partial y}{\partial a_{mn}} \end{bmatrix}$$

Using the denominator layout, the dimension is given by

$$\nabla_{\mathbf{A}} y \in \mathbb{R}^{m \times n}$$

1.3 Derivative of a vector with respect to a vector

Let $\mathbf{y} \in \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{R}^m$ be vectors. Then using the derivative of scalar with respect to a vector, we know

$$\Delta y_i = \nabla_{\mathbf{x}} y_i \Delta \mathbf{x}$$

Hence,

$$\mathbf{J} = \begin{bmatrix} (\nabla_{\mathbf{x}} y_1)^T \\ \vdots \\ (\nabla_{\mathbf{x}} y_n)^T \end{bmatrix}$$

The above matrix is known as the Jacobian matrix ($\mathbf{J}$) and has dimensions $n \times m$ but in the denominator layout the dimension of the derivative ought to be $m \times n$. Therefore, we define the derivative as the transpose of the Jacobian matrix

$$\nabla_{\mathbf{x}}\mathbf{y} = \begin{bmatrix} \nabla_{\mathbf{x}} y_1 & \cdots & \nabla_{\mathbf{x}} y_n \end{bmatrix} = \mathbf{J}^T$$

1.4 Tricks for computing derivatives

In many cases using the definition might not be the easiest way to compute the derivatives. The two tricks that are often used to compute the derivatives bypassing the definition are:

- Properties of trace
- Chain rule of derivatives

1.4.1 Properties of trace

Trace of a square matrix, $\mathbf{A} \in \mathbb{R}^{n \times n}$, is defined as

$$Tr(A) = \sum_{i=1}^{n} a_{ii}$$

Some of the very useful properties of trace are:

1. $Tr(\mathbf{A}_1\mathbf{A}_2 \cdots \mathbf{A}_n) = Tr(\mathbf{A}_n\mathbf{A}_1 \cdots \mathbf{A}_{n-1})$
2. $Tr(\mathbf{A}) = Tr(\mathbf{A}^T)$
3. $Tr(\mathbf{A}^T\mathbf{B}) = \sum_i \sum_j a_{ij} b_{ij}$

1.4.2 Chain rule of derivatives

Let $\mathbf{x} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^p$. Further, let $\mathbf{y} = f(\mathbf{x})$ for $f : \mathbb{R}^m \to \mathbb{R}^n$ and $\mathbf{z} = g(\mathbf{y})$ for $g : \mathbb{R}^n \to \mathbb{R}^p$. Then

$$\nabla_{\mathbf{x}}\mathbf{z} = \nabla_{\mathbf{x}}\mathbf{y}\nabla_{\mathbf{y}}\mathbf{z}$$

From the above expression we can observe that the chain rule runs from right to left due to the denominator layout.

1.5 Practice problem on derivatives

Compute the following derivatives and specify the properties you are using.

1. $\nabla_{\mathbf{A}} Tr(\mathbf{AB})$
2. $\nabla_{\mathbf{A}} \mathbf{x}^T\mathbf{A}\mathbf{x}$.
3. $\nabla_{\mathbf{z}} (\mathbf{x} - \mathbf{z})^T \Sigma^{-1} (\mathbf{x} - \mathbf{z})$, where Σ^{-1} is a symmetric matrix.
4. Let $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times n}$ and $f = \|\mathbf{A} + \lambda\mathbf{B}\|_F^2$. What is $\nabla_{\mathbf{B}} f$ if $\mathbf{B}$ is an orthogonal matrix?

2 Regularized least squares

In lecture, we worked through the following least squares problem

$$\arg\min_{\theta} \ \frac{1}{2} \sum_{i=1}^{N} (y^{(i)} - \theta^T \hat{\mathbf{x}}^{(i)})^2$$

However, the least squares has a tendency to overfit the training data. One common technique used to address the overfitting problem is regularization. In this discussion, we work through one of the regularization techniques namely ridge regularization which is also known as the regularized least squares problem. In the regularized least squares we solve the following optimization problem

$$\arg\min_{\theta} \ \frac{1}{2} \sum_{i=1}^{N} (y^{(i)} - \theta^T \hat{\mathbf{x}}^{(i)})^2 + \frac{\lambda}{2} \|\theta\|_2^2$$

where λ is a tunable regularization parameter. From the above cost function it can be observed that we are seeking least squares solution with a smaller 2-norm.

Derive the solution to the regularized least squares problem, i.e Find θ^*.

3 Word translation with Linear Models

Machine translation (MT) is a sub-field of computational linguistics, where a trained model is used to translate text from one language to another. A key step in such a model is to build a mapping between the words. In this problem, we take a first stab at learning the mapping between words in English and Spanish. It is common in natural language processing (NLP) tasks to represents words in the form of vectors , so that they could be used to train Machine Learning models.

Assume that you are given a set of K words in English language whose vector representations are denoted by $\mathbf{a}^{(i)} \quad , \forall i = 1 \dots K$ where $\mathbf{a}^{(i)} \in R^n$. Also assume that you are given the vector representations of the Spanish equivalent of the same K words which denoted by $b^{(i)} \quad , \forall i = 1 \dots K$ where $b^{(i)} \in R^m$. You have learned about linear models in ECE C147/C247 and you decide to use it to learn the mapping between the vector representations of words in english to spanish (i.e from $\mathbb{R}^n$ space into $\mathbb{R}^m$ space). we learn the parameters of the linear model W by minimizing the following loss function:

$$L(W) = \frac{1}{2} \sum_{i=1}^{K} \|\mathbf{b}^{(i)} - W\mathbf{a}^{(i)}\|_2^2$$

1. Explain in words what the loss function is trying to achieve?
2. Compute $\nabla_W L$.