1 A Few Notable Operators

Summation operator $\sum$ is a short notation for adding a large number of terms. Let $m$ be a natural number, and $a_i$ be a term for $i = 1, \ldots, m$.

$$\sum_{i=1}^{m} a_i := a_1 + a_2 + \ldots + a_{m-1} + a_m.$$  

(1)

Double summation is defined in a similar principle. Let $m$ and $n$ be two natural numbers, and each $a_{ij}$ is a term.

$$\sum_{i,j=1}^{n,m} a_{ij} := a_{11} + a_{12} + \ldots + a_{1,n-1} + a_{1n}$$

$$a_{21} + a_{22} + \ldots + a_{2,n-1} + a_{2n}$$

$$\vdots$$

$$a_{m-1,1} + a_{m-1,2} + \ldots + a_{m-1,n-1} + a_{m-1,n}$$

$$a_{m1} + a_{m2} + \ldots + a_{m,n-1} + a_{mn}$$

(2)

(3)

Distributive law works for summation.

$$b \cdot \sum_{i=1}^{m} a_i = \sum_{i=1}^{m} b \cdot a_i$$

(4)

The product operator $\prod$ is a short notation of multiplying a sequence of factors together.

$$\prod_{i=1}^{m} a_i = a_1 \cdot a_2 \cdot \ldots \cdot a_{m-1} \cdot a_m$$

(5)

$$\log \left( \prod_{i=1}^{m} a_i \right) = \sum_{i=1}^{m} \log(a_i), \text{ if } a_i > 0 \text{ for all } 1 \leq i \leq m$$

(6)

$$\exp \left( \sum_{i=1}^{m} a_i \right) = \prod_{i=1}^{m} \exp(a_i)$$

(7)

Maximization and minimization operators are super important in optimization. Here is one simple but important rule. Let $f(x)$ be a function of $x$, and $c$ is a constant with respect to $x$. If $c > 0$,

$$\max_{x \in S} c \cdot f(x) = c \left( \max_{x \in S} f(x) \right)$$

(8)

$$\arg \max_{x \in S} c \cdot f(x) = \arg \max_{x \in S} f(x)$$

(9)

Here $S$ is the set to optimize over ($f(x)$ is defined for every $x \in S$).
If \( c < 0 \),
\[
\max_{x \in S} c \cdot f(x) = c \left( \min_{x \in S} f(x) \right) 
\]
(10)
\[
\arg \max_{x \in S} c \cdot f(x) = \arg \min_{x \in S} f(x) 
\]
(11)

The rules for the min operator are similar. If \( c > 0 \),
\[
\min_{x \in S} c \cdot f(x) = c \left( \min_{x \in S} f(x) \right) 
\]
(12)
\[
\arg \min_{x \in S} c \cdot f(x) = \arg \min_{x \in S} f(x) 
\]
(13)

If \( c < 0 \),
\[
\min_{x \in S} c \cdot f(x) = c \left( \max_{x \in S} f(x) \right) 
\]
(14)
\[
\arg \min_{x \in S} c \cdot f(x) = \arg \max_{x \in S} f(x) 
\]
(15)

2 Differential Calculus

Differential calculus is about the relation between the changes of function inputs and the corresponding changes of function values. It is the foundation of numerical optimization, which is a cornerstone of machine learning.

The derivative of a function \( y = f(x) \) is
\[
f'(x) = \frac{dy}{dx} = \lim_{h \to 0} \frac{f(x + h) - f(x)}{h} 
\]
(16)

A function is non-differentiable at a location \( x \) if the limit above does not exist.

Examples:

<table>
<thead>
<tr>
<th>( f(x) )</th>
<th>( f'(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x^2 )</td>
<td>( 2x )</td>
</tr>
<tr>
<td>( x^3 )</td>
<td>( 3x^2 )</td>
</tr>
<tr>
<td>( \frac{1}{x} )</td>
<td>( -\frac{1}{x^2} )</td>
</tr>
</tbody>
</table>

Use the sum rule to calculate derivative of function summations. If \( f(x) = f_1(x) + f_2(x) \), then
\[
f'(x) = f'_1(x) + f'_2(x) 
\]
(17)

Use the product rule to calculate derivative of function multiplications. If \( f(x) = f_1(x) \cdot f_2(x) \), then
\[
f'(x) = f'_1(x) \cdot f_2(x) + f_1(x) \cdot f'_2(x) 
\]
(18)

Use the chain rule to calculate the derivative of the composition of functions. If \( f(x) = g(h(x)) \), then
\[
f'(x) = g'(h(x)) \cdot h'(x) 
\]
(19)

2.1 Multivariate Differential Calculus

Machine learning absolutely needs multivariate differential calculus as a model contains multiple variables. Suppose \( y = f(x_1, x_2, \ldots, x_m) \). Then the partial derivative \( \frac{\partial y}{\partial x_i} \) defines the derivative of the function with respect to one of its input variables.
\[
\frac{\partial y}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, x_2, \ldots, x_i + h, \ldots, x_m) - f(x_1, x_2, \ldots, x_i, \ldots, x_m)}{h} 
\]
(20)
The partial derivative of \( f(x_1, x_2, \ldots, x_m) \) with respect to \( x_i \) uses the same rule as derivatives and treats other variables as constants.

The **gradient** of a function \( f(x_1, \ldots, x_m) \) is

\[
\nabla f := \left( \frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2}, \ldots, \frac{\partial y}{\partial x_m} \right)
\]

(21)

If we write variables in vectors, then the calculation is much neater by using derivative rules with respect to vectors.

Denote \( \mathbf{x} = (x_1, x_2, \ldots, x_m) \), then the gradient \( \nabla f \) of the function \( f(\mathbf{x}) \) is also a vector with the same length as \( \mathbf{x} \).

The **summation** and **product** rules still work. The chain rule is much more complex than that of univariable differentiation, but we only need the following simple rule. If \( f(\mathbf{x}) = g(h(\mathbf{x})) \) with \( g(\cdot) \) as a univariate function, then

\[
\nabla f = g'(h(\mathbf{x})) \cdot \nabla h .
\]

(22)

The derivatives of several simple functions with respect to vectors are listed below.

<table>
<thead>
<tr>
<th>( f(\mathbf{x}) )</th>
<th>( \nabla f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f(\mathbf{x}) = \mathbf{a} \top \mathbf{x} )</td>
<td>( \nabla f = \mathbf{a} )</td>
</tr>
<tr>
<td>( f(\mathbf{x}) = \mathbf{x} \top \mathbf{x} )</td>
<td>( \nabla f = 2\mathbf{x} )</td>
</tr>
<tr>
<td>( f(\mathbf{x}) = (\mathbf{x} - \mathbf{a}) \top (\mathbf{x} - \mathbf{a}) )</td>
<td>( \nabla f = 2(\mathbf{x} - \mathbf{a}) )</td>
</tr>
</tbody>
</table>

See the [Wikipedia page](https://en.wikipedia.org/wiki/Gradient) for a more complete rules of vector derivatives.