HyperionDev

Exploring Multivariable Calculus

Task

Visit our website

Introduction

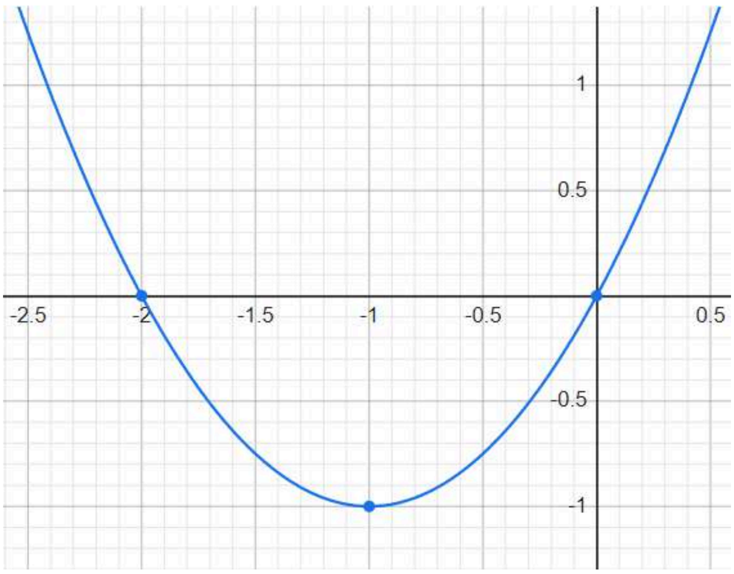
In our initial exploration of calculus, we covered fundamental concepts such as function notation, limits, the definition of a derivative, the power rule, and how turning points help identify the lowest point on error functions. Now, we are going to expand on this foundation and delve deeper into the mathematics necessary for understanding how more complex algorithms work.

Functions and multivariable calculus

Functions can vary greatly in complexity. Some functions are quite simple, consisting of a single term and one variable, such as $f(x) = x^2$. Others can be much more complex, involving multiple terms, different variables, and various operations. In this section, we will look at two kinds of functions: single-variable functions and multivariable functions.

Single-variable functions

We are quite familiar with these kinds of functions. We have a function, f, which takes in a single input, x, and returns an output. Let's look at the function $f(x) = x^2 + 2x$ which graphically, looks like this:



Let us discuss how the value of $f(x)$ changes as x varies. To the left of $x = -1$, the downward slope indicates that the function is decreasing. Imagine placing a ball at the top of the curve – it would roll down towards $x = -1$. Once we pass this point, the curve begins to rise, indicating the function is increasing, as shown by the upward slope.

By examining the curvature more closely, we observe that between $x = -2.5$ to $x = -2$ the slope is steeper, meaning the function decreases faster compared to the segment between $x = -2$ to $x = -1$. So, we can say that as x increases, the rate at which the function decreases slows down. Beyond $x = -1$, the slope gets steeper as x increases, showing that for values of x greater than -1, the function's rate of increase gets faster.

Multivariable functions

Functions can also depend on more than one variable. We've encountered such functions before, for instance, when dealing with linear systems involving multiple features. In these cases, we often fit a multiple linear regression model, which is essentially a function of several variables.

Mathematically, functions that depend on multiple variables are expressed as $f(x_1, x_2, , x_n)$. Let's look at an example of such a function: $f(x, y) = x^2 + y^2$. In this case, changes in **both** x and y will influence the outcome of our function.

Since we now have an additional variable, the two-dimensional plots we previously used are no longer sufficient to visualise this function. We would need to introduce another dimension to capture the effects of this second variable, resulting in a three-dimensional representation of our function $f(x, y) = x^2 + y^2$. While the technicalities of multidimensional plotting are beyond the scope of this task, the key takeaway is that when a function depends on multiple variables, the behaviour of the function – its slope or curvature at different points – depends on all the variables involved.

Derivatives and rates of change

Now that we have a grasp of both single-variable and multivariable functions and their behaviour, we can explore how they relate to derivatives. Recall that a derivative gives us the rate of change or the slope of a function at any specific point.

Single variable derivative

Given the function $f(x) = x^2 + 2x$, we can find its derivative by using the **sum rule**. This basically involves applying the power rule we looked at previously to each term individually, resulting in the derivative:

$$f'(x) = 2x + 2$$

We interpret this derivative in exactly the same way that we did in the first Calculus lesson. For example:

- At $x = 3$, the derivative is $f'(3) = 2(3) + 2 = 6 + 2 = 8$. This means the slope of the function at this point is 8.
- At $x = -1$, the derivative is $f'(-1) = 2(-1) + 2 = -2 + 2 = 0$. This means the slope of the function at this point is 0.

Partial derivatives

Now, what happens when we have multiple variables? We know that in multivariable functions, changes depend on all the variables involved. So, what we can do to find derivatives is focus on one variable at a time. For example, consider the function $f(x, y) = x^2 + y^2$. We can find the derivative in the direction of x while keeping y constant, and similarly, find the derivative in the direction of y while keeping x constant. This process is called finding **partial derivatives**.

The notation for the partial derivative of f with respect to x is $\frac{\partial f}{\partial x}$. To compute the partial derivative with respect to x, we treat y as a constant and differentiate only with respect to x. This gives us:

$$\frac{\partial f}{\partial x} = 2x^{2-1} + 0 = 2x$$

Similarly, the notation for the partial derivative of f with respect to y is $\frac{\partial f}{\partial y}$. We compute the partial derivative with respect to y by treating x as a constant and differentiating with respect y. This gives us:

$$\frac{\partial f}{\partial y} = 0 + 2y^{2-1} = 2y$$

Partial derivatives allow us to understand how the function changes as each individual variable changes, holding the others constant.

The gradient

We have explored how gradient descent helps minimise error in linear models by iteratively adjusting model parameters in the direction of steepest descent. Essentially, gradient descent helps us find the lowest point on an error function. The gradient vector builds on this concept, but in a broader sense. It represents the direction of steepest increase for any scalar function, not just for minimising error. While gradient descent uses this concept to identify optimal parameters by moving against the gradient, the gradient vector also helps us understand how a function changes across multiple dimensions.

The gradient is a vector that represents both the direction and rate of the fastest increase of a scalar function. In simpler terms, the gradient tells you how the function changes at a point and in which direction it changes the most rapidly. The gradient vector of a function is represented as ∇f. Each component of the gradient corresponds to the partial derivative with respect to a variable.

Consider again the function $f(x, y) = x^2 + y^2$. The first step to getting its gradient function would be to compute the partial derivatives with respect to each variable. We did this in the previous section where we found:

$$\frac{\partial f}{\partial x} = 2x \quad \text{and} \quad \frac{\partial f}{\partial y} = 2y$$

The gradient vector, ∇f,is formed by combining the partial derivatives of the function. Therefore, ∇f is expressed as:

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) = (2x, 2y)$$

We can use this gradient to figure out the fastest way to climb the surface defined by $f(x, y)$. Let's start at the origin, where $x = 0$ and $y = 0$. At this point, we have zero gradient:

$$\nabla f = (2(0), 2(0)) = (0, 0)$$

This is a **stationary point**, which is a location where the function neither increases nor decreases. In other words, the slope is zero, similar to the turning points we discussed previously. As we move away from the origin, the steeper the slope of that direction of the gradient. For example:

- At $x = 4$ and $y = 4$, we have $\nabla f = (2(4), 2(4)) = (8, 8)$.
- Moving further away from the origin, at $x = 10$ and $y = 23$, we have $\nabla f = (2(10), 2(23)) = (20, 46)$ indicating an even steeper gradient.

So, using the gradient vector, we can conclude that the fastest way to climb the surface defined by our function $f(x, y)$ would be to move farther away from the origin where the gradient becomes steeper.

The chain rule

Now that we understand how to compute the gradient of a multivariable function, we can explore how functions change when the variables themselves depend on other variables. In such cases, we need a way to account for these dependencies while differentiating. This is where the **chain rule** comes in. The chain rule allows us to compute the derivative of a **composition** of functions, providing a powerful tool for handling more complex relationships between variables.

The chain rule is a fundamental rule in calculus used to differentiate **composite functions**. When we say $f(x)$, the function f depends on x. Similarly, when we say $g(x)$, the function g depends on x. Now, if we combine these to form $f(g(x))$, this represents a composite function where f depends on $g(x)$, and g itself depends on x.

The chain rule allows us to find the rate of change of f with respect to x by accounting for the dependence of f on g, and g on x. Mathematically, for functions $f(g(x))$, the chain rule is written as:

$$\frac{d}{dx}f(g(x)) = \frac{df}{dg} \cdot \frac{dg}{dx}$$

In this expression, we first compute the derivative of f with respect to g (the outer function), and then multiply it by the derivative of g with respect to x (the inner function).

The chain rule and partial derivatives

When dealing with multivariable functions, the chain rule extends naturally to partial derivatives. In this case, if a function f depends on multiple variables, each of which depends on other variables, we need to apply the chain rule to account for all the paths through which f changes.

Let's say f is a function of x and y, but both x and y depend on another variable, say t. The chain rule for partial derivatives allows us to compute the total derivative of f with respect to t as follows:

$$\frac{df}{dt} = \frac{\partial f}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial f}{\partial y} \cdot \frac{dy}{dt}$$

This formula captures how the function f changes with respect to t by summing up the contributions from both x and y, weighted by how each of them changes with t.

The chain rule is incredibly useful in cases where we want to find derivatives, but the dependencies between functions and variables are a little more complex.

Optimisation and real-life applications

Tying it all together, let us talk about optimisation. Optimisation involves finding the maximum or minimum values of a function and is crucial in various fields such as economics, engineering, and data science. For example, an optimisation problem might focus on maximising revenue from the production of a product while minimising associated costs – leading to the question: how many units should be produced to achieve this? Another example could involve creating the largest possible rectangular enclosure for your pet while working with a limited length of fencing. You can find the solution to this calculus problem **here**.

Optimisation starts by identifying the function you want to optimise and any constraints involved. This often requires translating real-world scenarios into mathematical expressions. For this explanation, we will examine the familiar error function, which we aim to minimise by finding its lowest point. We can locate this point by searching for critical points (such as turning points) along the function. Calculus comes in very handy for this task:

- For single variable functions, we find the derivative and set it to zero, as we know the gradient is always zero at a turning point.
- For multivariable functions, we can use partial derivatives to find the critical points in the multidimensional space. Thereafter, the gradient vector can provide insight into the direction of steepest ascent (climbing up the hill for maximisation problems) or descent (rolling down the hill for minimisation problems).

Practical task 1

1. Discuss the concept of partial derivatives and explain how they differ from ordinary derivatives.
2. Submit your discussion in a file named **derivatives.txt**.

Practical task 2

Create a file named **multivariable.py**. Write a function that computes the gradient of a two-term multivariable function. For example:

User input:

```
x**2 + y**2
```

Output:

```
The gradient vector of the function f = x2 + y2 is ∇f = (2x, 2y).
```

Hint: The `sympy` library can be used to calculate derivatives. For example, if we have the function f = x^3 and we want the derivative with respect to x, we can do the following:

```
import sympy as sp
# Define symbol in your function
x = sp.symbols('x')
# Define function
f = x**3
# Calculate derivative
df_dx = sp.diff(f, x)
```

Hint: the nabla symbol ∇ can be printed out either using its symbol code '\u2207' or using the `sympy` library: `sp.symbols('nabla')`

Important: Be sure to upload all files required for the task submission inside your task folder and then click "Request review" on your dashboard.

Share your thoughts

Please take some time to complete this short feedback **form** to help us ensure we provide you with the best possible learning experience.