AD and SciML

André Freitas

January 4, 2024

1 Forward Mode Automatic Differentiation

Automatic Differentiation (AD) is a computational technique for systematically and accurately evaluating derivatives of functions. Forward-mode AD is particularly valuable in machine learning applications where the number of input variables is relatively small compared to the number of outputs.

1.1 Dual Numbers

At the heart of forward-mode AD lies the concept of dual numbers. A dual number is an extension of real numbers that includes an infinitesimal part, often denoted by ε. A dual number can be expressed as $a + b\varepsilon$, where a and b are real numbers. The infinitesimal ε is chosen such that $\varepsilon^2 = 0$, allowing for the representation of both the function value and its derivative.

1.2 Computational Graph and Elementary Operations

To implement forward-mode AD, a computational graph is constructed to represent the sequence of elementary operations in a function. Each node in the graph corresponds to an elementary operation, and edges capture the dependencies between operations. Dual numbers are used to propagate both the function values and their derivatives through the graph.

1.3 Forward Pass and Dual Number Arithmetic

The forward pass in forward-mode AD involves evaluating the function at a particular input, using dual numbers to represent both the function value and its derivative. The arithmetic operations on dual numbers are extended to include the chain rule, allowing the simultaneous computation of function values and derivatives during the forward pass.

1.4 Mathematical Formulation

Consider a function $f(x)$, where x is a vector of input variables. The forward pass involves evaluating f at a point x_0 using dual numbers:

$$f(x_0 + \varepsilon h) = f(x_0) + h \cdot \nabla f(x_0)$$

Here, h is a real number representing the perturbation applied to each input variable, and $\nabla f(x_0)$ is the gradient of f at x_0.

1.5 Algorithmic Implementation

Implementing forward-mode AD involves augmenting each variable in the computation with its corresponding dual number representation. Pseudo-code for a simple forward-mode AD algorithm can be outlined as follows:

```
function forward_mode_AD(f, x)
    x_dual = create_dual(x)
    y_dual = f(x_dual)
    return get_derivative(y_dual)
end function
```

1.6 Advantages and Considerations

Forward-mode AD excels in scenarios where the dimensionality of the input space is low. However, its efficiency may diminish in high-dimensional spaces. Understanding the balance between forward and reverse modes is crucial for optimizing computational costs in the context of specific applications.

In conclusion, forward-mode AD, with its foundation in dual numbers, provides a powerful means of computing gradients in machine learning and optimization. The elegant combination of mathematical principles and algorithmic implementation makes it a valuable tool for practitioners seeking efficient and accurate differentiation in their applications.