Dual Numbers and Differentiation

This is a brief introduction to dual numbers and how they can be used to obtain derivatives. It assumes that the reader has taken a introductory course in calculus. Probably a moot point as if one hasn't, knowing or caring about differentiation is unlikely!

Definition: Let

$$d = x + \epsilon x'$$

where x, x' are real numbers and $\epsilon^2 = 0$ for $\epsilon > 0$ be a *dual number*.

Now let us consider basic operations on dual numbers. Suppose that $d_1 = x + \epsilon x'$ and $d_2 = y + \epsilon y'$:

1) Addition:

$$d_1 + d_2 = x + \epsilon x' + y + \epsilon y' = (x + y) + \epsilon(x' + y')$$

2) Subtraction:

$$d_1 - d_2 = (x + \epsilon x') - (y + \epsilon y') = (x - y) + \epsilon(x' - y')$$

3) Multiplication:

$$\begin{aligned} d_1 d_2 &= (x + \epsilon x') \cdot (y + \epsilon y') = xy + \epsilon x y' + \epsilon x' y + \epsilon^2 x' y' \\ &= xy + \epsilon(x'y + xy') + \overset{0}{\cancel{\epsilon^2}} x'y' = xy + \epsilon(x'y + xy') \end{aligned}$$

4) Division: Dividing dual numbers is handled by multiplying top and bottom of the fraction with the conjugate of the denominator (just like complex numbers).

$$\begin{aligned} \frac{d_1}{d_2} &= \frac{x + \epsilon x'}{y + \epsilon y'} = \frac{x + \epsilon x'}{y + \epsilon y'} \cdot \frac{y - \epsilon y'}{y - \epsilon y'} = \frac{xy - \epsilon x y' + \epsilon x' y - \overset{0}{\cancel{\epsilon^2}} x'y'}{y^2 \cancelto{0}{- \epsilon y y' + \epsilon y' y} + \overset{0}{\cancel{\epsilon^2}} y'y'} \\ &= \frac{xy + \epsilon(x'y - xy')}{y^2} = \frac{x}{y} + \epsilon \frac{x'y - xy'}{y^2} \end{aligned}$$

Recalling our basic calculus this is looking promising, i.e. the first term of the dual number is the operation in question and the second term, the value associated with the ϵ, its derivative. To consider how dual numbers behave under function application recall the Taylor series expansion of a differentiable function $f(x)$ about a:

$$\begin{aligned} f(x) &= f(a) + f'(a)(x - a) + f''(a)\frac{(x-a)^2}{2!} + f'''(a)\frac{(x-a)^3}{3!} + \dots \\ &= \sum_{i=0}^{\infty} f^{(i)}(a) \frac{(x-a)^i}{i!} \end{aligned}$$

Now suppose we take a Taylor expansion of $f(x+\epsilon x')$ about x:

$$\begin{aligned} f(x+\epsilon x') &= f(x) + f'(x)\epsilon x' + f''(x)\frac{(\epsilon x')^2}{2!} f'''(x)\frac{(\epsilon x')^3}{3!} + \dots \\ &= f(x) + f'(x)\epsilon x' + f''(x)\frac{\cancelto{0}{\epsilon^2}(x')^2}{2!} f'''(x)\frac{\cancelto{0}{\epsilon^2}\epsilon(x')^3}{3!} + \cancelto{0}{\dots} \\ &= f(x) + \epsilon f'(x)x' \end{aligned}$$

So this tells us that if we apply function f to a dual number $x+\epsilon x'$ we get a dual number with the function and its derivative evaluated at x; the second term is nothing other than the good old chain rule, yay! If you are still not convinced suppose you have a dual number $x+\epsilon x'$ and x is one of the following:

1) A constant a:

$$x = a \;\Rightarrow\; x' = 0 \;\Rightarrow\; f(x+\epsilon x') = f(a) + \epsilon f'(x)0 = f(a) + \epsilon 0$$

2) Itself, so any real number (why you create a dual number with $x' = 1$):

$$x = x \;\Rightarrow\; x' = 1 \;\Rightarrow\; f(x+\epsilon x') = f(x+\epsilon) = f(x) + \epsilon f'(x)$$

3) A differentiable function g:

$$\begin{aligned} x = g(x) \;\Rightarrow\; x' = g'(x) \;\Rightarrow\; f(x+\epsilon x') &= f(g(x) + \epsilon g'(x)) \\ &= f(g(x)) + \epsilon f'(g(x))g'(x) \end{aligned}$$

Implementing dual numbers in your programming language of choice gives you what is referred to as forward-mode *automatic differentiation* (AD). The presentation above is for univariate functions but the idea can be extended to functions with multi-dimensional input and output. Unfortunately, forward-mode AD is only computationally efficient for functions with a small number of inputs relative to outputs. A lot of useful problems such as minimizing a loss function are of the many inputs to one output variety and so in such cases naive forward-mode AD isn't the best choice. One could use clever function call caching (i.e. memoization) strategies to mitigate this problem but the implementation would become much more complicated. There is a technique called reverse-mode AD that is efficient for many-to-one functions but to my knowledge doesn't have a nice mathematical representation like forward-mode AD via dual numbers and again implementation isn't straight forward (parsing expressions, optimizing the resulting graph, . . .).