Mathematics for Machine Learning

Multivariate Calculus

Formula sheet

Dr Samuel J. Cooper
Prof. David Dye
Dr A. Freddie Page

Definition of a derivative

$$f'(x) = \frac{\mathrm{d}f(x)}{\mathrm{d}x} = \lim_{\Delta x \to 0} \left(\frac{f(x + \Delta x) - f(x)}{\Delta x} \right)$$

Time saving rules

- *Sum Rule*:

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(f(x) + g(x)\right) = \frac{\mathrm{d}}{\mathrm{d}x}(f(x)) + \frac{\mathrm{d}}{\mathrm{d}x}(g(x))$$

- *Power Rule*:

Given $f(x) = ax^b$,
then $f'(x) = abx^{(b-1)}$

- *Product Rule*:

Given $A(x) = f(x)g(x)$,
then $A'(x) = f'(x)g(x) + f(x)g'(x)$

- *Chain Rule*:

Given $h = h(p)$ and $p = p(m)$,

$$\text{then} \quad \frac{\mathrm{d}h}{\mathrm{d}m} = \frac{\mathrm{d}h}{\mathrm{d}p} \times \frac{\mathrm{d}p}{\mathrm{d}m}$$

- *Total derivative*:

For the function $f(x, y, z, ...)$, where each variable is a function of parameter t, the total derivative is

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial f}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial f}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\partial f}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}t} + \ldots$$

Derivatives of named functions

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{x}\right) &= -\frac{1}{x^2} \\
\frac{\mathrm{d}}{\mathrm{d}x}(\sin(x)) &= \cos(x) \\
\frac{\mathrm{d}}{\mathrm{d}x}(\cos(x)) &= -\sin(x) \\
\frac{\mathrm{d}}{\mathrm{d}x}(\exp(x)) &= \exp(x)
\end{aligned}$$

Derivative structures

Given $f = f(x, y, z)$

- *Jacobian*:

$$\mathbf{J}_f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right]$$

- *Hessian*:

$$\mathbf{H}_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2} \end{bmatrix}$$

Neural networks

- *Activation function*:

$$\sigma(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$
$$\frac{\mathrm{d}}{\mathrm{d}x}(\sigma(x)) = \frac{1}{\cosh^2(x)} = \frac{4}{(e^x + e^{-x})^2}$$

Taylor Series

- *Univariate*:

$$f(x) = f(c) + f'(c)(x-c) + \frac{1}{2}f''(c)(x-c)^2 + ...$$
$$= \sum_{n=0}^{\infty} \frac{f^{(n)}(c)}{n!}(x-c)^n$$

- *Multivariate*:

$$f(\mathbf{x}) = f(\mathbf{c}) + \mathbf{J}_f(\mathbf{c})(\mathbf{x}-\mathbf{c}) + \frac{1}{2}(\mathbf{x}-\mathbf{c})^t\mathbf{H}_f(\mathbf{c})(\mathbf{x}-\mathbf{c}) + ...$$

Optimization and Vector Calculus

- *Newton-Raphson*:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

- *Grad*:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix}$$

- *Directional Gradient*:

$$\nabla f.\hat{r}$$

- *Gradient Descent*:

$$s_{n+1} = s_n - \gamma \nabla f$$

- *Lagrange Multipliers* λ:

$$\nabla f = \lambda \nabla g$$

- *Least Squares -* χ^2 *minimization*:

$$\chi^2 = \sum_i^n \frac{(y_i - y(x_i; a_k))^2}{\sigma_i}$$

$$\text{criterion:} \quad \nabla \chi^2 = 0$$

$$a_{\text{next}} = a_{\text{cur}} - \gamma \nabla \chi^2$$
$$= a_{\text{cur}} + \gamma \sum_i^n \frac{(y_i - y(x_i; a_k))}{\sigma_i} \frac{\partial y}{\partial a_k}$$