Neural Networks (Derivatives)

Machine Learning, A.Y. 2022/23, Padova

Fabio Aiolli

October 26th, 2022

Derivative Rules

The derivative tells us the slope of a function at any point.
Given a point. Positive derivative means the function increases at that point. Negative derivative means the function decreases at that point. Null derivative means there is a stationary point (minimum, maximum, or saddle point)!

There are rules we can follow to find many derivatives.

- constant $y = c, c \in \mathbb{R} \Rightarrow y' = 0$
- line $y = c \cdot x, c \in \mathbb{R} \Rightarrow y' = c$
- power $y = x^n, n \in \mathbb{N} \Rightarrow y' = nx^{n-1}$
- exponential $y = e^x \Rightarrow y' = e^x$

Derivative Rules

- Multiplication by constant $D(k \cdot f(x)) = k \cdot f'(x)$
- Sum $D(\sum_i f_i(x)) = \sum_i f'(x)$
- Product $D(f(x) \cdot g(x)) = f'(x) \cdot g(x) + f(x) \cdot g'(x)$
- Quotient $D(\frac{f(x)}{g(x)}) = \frac{f'(x) \cdot g(x) - f(x) \cdot g'(x)}{g(x)^2}$
- Reciprocal $D(\frac{1}{g(x)}) = -\frac{g'(x)}{g(x)^2}$
- Composition/Chain $D(g(f(x)) = g'(f(x)) \cdot f'(x)$

Gradient and its properties

Definition of Gradient

In vector calculus, the gradient of a scalar-valued differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is a vector-valued function $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ whose value at point p is the vector whose components are the partial derivatives of f at p.

$$\nabla f(p) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(p) \\ \frac{\partial f}{\partial x_2}(p) \\ \vdots \\ \frac{\partial f}{\partial x_n}(p) \end{pmatrix}$$

- The gradient vector can be interpreted as the direction and rate of fastest increase;
- The gradient is the zero vector at a point if and only if the point is a stationary point (where the derivative vanishes);
- The gradient thus plays a fundamental role in optimization theory.

Minimization with gradient descent

Minimization problem

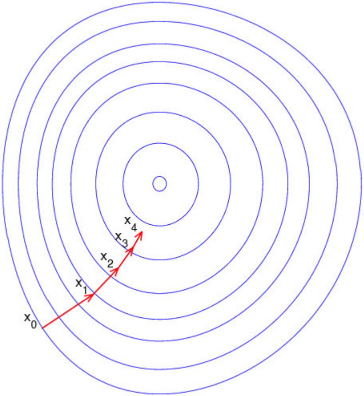
Let $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ be a vector-valued function, find the vector $\mathbf{x}^* \in \mathbb{R}^n$ that minimizes the function f, that is:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} f(\mathbf{x})$$

Gradient descent algorithm

- $k \leftarrow 0$, $\mathbf{x}_0 \in \mathbb{R}^n$
- **while** $\nabla f(\mathbf{x}_k) \neq 0$
  - Compute the descent direction $\mathbf{p}_k := -\nabla f(\mathbf{x}_k)$
  - Compute the step of descent η_k
  - Update $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \eta_k \mathbf{p}_k$
  - $k \leftarrow k + 1$

Minimization with gradient descent



Note: It is not guaranteed that it converges to a global minimum!