

AI5031: Machine learning

1 1D derivatives

Compute the derivative $h'(x)$. Always give rule/decomposition along with the calculation!

- a) $h(x) = \sin(\sin(x))$
- b) $h(x) = 2 \sin(x) + 5 \cos(x)$
- c) $h(x) = \alpha \sin(x) + \beta \cos(x)$
- d) $h(x) = \ln(\sin(\cos(5.2)))$
- e) $h(x) = \exp(x) \ln(x)$
- f) $h(x) = x \cos(x)$
- g) $h(x) = e^{\cos(x)}$

2 Preparations (on paper)

Compute the following derivatives! No justification needed...

- a) $\frac{\partial}{\partial x_3} \sum_{i=1}^5 x_i$
- b) $\frac{\partial}{\partial x_6} \sum_{i=1}^5 x_i$
- c) $\frac{\partial}{\partial x_3} \sum_{i=1}^5 (2x_i - 5)^2$
- d) $\frac{\partial}{\partial x_3} \sum_{i=1}^5 (x_i^2 - 5x_i)^2$

For the remaining exercises, give rule, decomposition and calculation!

- e) $\frac{\partial}{\partial x_3} \cos(x_3)x_3$
- f) $\frac{\partial}{\partial x_3} \cos(x_3)x_4$
- g) $\frac{\partial}{\partial x_3} \cos(\exp(x_3))$

3 Approximation

- a) Derive the slope of the function $f(x) = x^3$ at an arbitrary point x using the limit treatment from class
- b) Give the approximating linear functions at $x_0 = 1$ and $x_0 = 2$

4 Gradient descent on paper

Consider the function $f(\vec{x}) = x_1^2 + 2x_2^2$. Assuming a starting point of $\vec{x}(0) = [1, 3]$ and a step size of $\epsilon = 0.1$: perform 2 steps of gradient descent, that is, compute the values of $\vec{x}(1)$, $\vec{x}(2)$ by iteration. Verify that the value of f is decreasing!

5 Gradient descent (on the computer)

Write a Python3 program that uses numpy to perform N steps of gradient descent for f , with starting point $\vec{x} = (1, 3)^T$ and steps size $\epsilon = 0.1$ defined by constants that can be changed easily. Implement the function $f(\vec{x})$ from the previous exercise, and its gradient $\text{grad}f(x)$, as Python3 functions that expect an *np.array* of length 2 representing \vec{x} and return the function value as a scalar, or the gradient as a *np.array*. After each iteration, print the current value of \vec{x} ! Play around with starting point and ϵ to see whether this changes the result! Bonus: You can use matplotlib to plot the trajectories!