AI5031: Machine learning, exercise 5

1 Derivatives

Compute the following derivatives!

a) $\frac{\partial}{\partial x_3} \sum_{i=1}^{2} x_i$
b) $\frac{\partial}{\partial x_6} \sum_{i=1}^{1} 5x_i^3$
c) $\frac{\partial}{\partial x_3} \sum_{i=1}^{5} (2x_i - 5)^2$
d) $\frac{\partial}{\partial x_3} \sum_{i=1}^{5} (x_i^2 - 5x_i)^2$
e) $\frac{\partial}{\partial x_3} \log(x_3)x_3$
f) $\frac{\partial}{\partial x_3} \exp(x_3)x_4$
g) $\frac{\partial}{\partial x_2} \cos(\exp(x_3))$

2 Gradient descent, 5P

Perform three steps of gradient descent, for the function $f(\vec{x}) = \exp(x_1 + x_2)$, using initial values of $\vec{x}(0) = [5, 6]$ and $\epsilon = 0.1$. Use a calculator and round all values to 2 decimal places!

3 Implementing softmax

Write a python function S(x) which takes an 1D numpy array and returns the softmax, also as a numpy array! Print out results for $\vec{x} = [-1, -1, 5]$ and $\vec{x} = [1, 1, 2]$!

4 Implementing a single-sample version of cross-entropy

Write a python function CE(x,t) which takes an 1D numpy array and returns the its cross-entropy as a scalar! Print out results for $\vec{t} = [0, 0, 1]$ in the three cases of $\vec{x} = [0.1, 0.1, 0.8]$ and $\vec{x} = [0.3, 0.3, 0.4]$ and $\vec{x} = [0.8, 0.1, 0.1]$!

5 Multi-sample versions!!

Repeat the previous two exercises, but now the functions should work row-wise on three-row arrays in which the vectors $\vec{x}$ and $\vec{t}$ are duplicated three times! Batch (or multi-sample) softmax should return an array, whereas batch cross-entropy still returns a scalar using the formula given in the lecture (using mean).