Machine Learning

Lab 09: Multi-Layer Perceptron

Task 1.

In this task you will train a neural network on the MNIST handwritten digits database and compare the results for different hidden layer sizes.

A. Load the Digits dataset from the dataset repository of SciKit learn and split it into a training and a test set
B. Train an MLP classifier with different hidden layer sizes on the training set and evaluate the performance on both the training and the test set.
C. The coefficients of the first hidden layer correspond to the pixels of the 8x8 images of handwritten digits. Display the absolute values of these coefficients as 8x8 images to visualise the strength of each pixel in the activation of each hidden unit.

Task 2. (Polynomials)

A univariate polynomial of degree d in x is the function

$$p[x] = a_0 + a_1 x + a_2 x^2 + \cdots + a_d x^d$$

The $d + 1$ coefficients $a_0, \ldots, a_d$ define the polynomial, which then can be evaluated for every x. For example, the polynomial $p[x] = 3 - 2x + 4x^2$ has the coefficients $a = (3, -2, 4)$ and evaluates for $x = 2$ to $p[2] = 15$.

A bivariate polynomial of degree d in (x, y) is the function

$$\begin{aligned} p[x, y] &= a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 + \cdots + a_{0d}y^d \\ &= \sum_{n=0}^{d} \sum_{\substack{i,j \geq 0 \\ i+j=n}} a_{ij} x^i y^j \end{aligned}$$

Similar, a trivariate polynomial of degree d in (x, y, z) is the function

$$p[x, y, z] = \sum_{n=0}^{d} \sum_{\substack{i,j,k \geq 0 \\ i+j+k=n}} a_{ijk} x^i y^j z^k$$

A. Write a function that determines the number of coefficients for a bi-variate polynomial given the degree d.
B. Write a function that determines the number of coefficients for a tri-variate polynomial given the degree d.
C. Write a function that evaluates a univariate polynomial given the coefficient vector a and the value for x.
D. Write a function that evaluates a bivariate polynomial given the degree d, the correctly sized coefficient vector a, and the values of x and y.
E. Write a function that evaluates a trivariate polynomial given the degree d, the correctly sized coefficient vector a, and the values of x, y and z.

Task 3. (Derivatives of polynomials)

Given a polynomial

$$p[x] = a_0 + a_1 x + a_2 x^2 + \cdots + a_d x^d$$

the derivative is the polynomial

$$p'[x] = a_1 + 2a_2 x^1 + \cdots + d a_d x^{d-1}$$

While the derivative of polynomials can be computed like this, they could also be computed using the differential quotient

$$p'[x] = \lim_{\epsilon \to 0} \frac{p[x+\epsilon] - p[x]}{\epsilon}$$

Using this formula (without the lim) we can use small ϵ to numerically approximate the derivative of any function for which we have a means of calculating its values, by subtracting two values close to each other and dividing by the distance between the two.

A. Write a function that calculates the derivative numerically using the differential quotient.
B. Use the function developed in task 2 and compare the result to the analytical derivative for some example polynomial p.

Task 4. (Derivative of a quadratic form)

The derivative of a scalar product is

$$\frac{\partial}{\partial x} a^T x = a^T$$

A. Use the method of numerical differentiation (see task 3) to verify this by comparing this analytical derivative with the numerical derivative of the scalar product $a^T x$ for some example vectors a and x.

The derivative of a quadratic form is

$$\frac{\partial}{\partial x} x^T A x = x^T (A^T + A)$$

If the matrix is symmetric, i.e. $A = A^T$, then

$$\frac{\partial}{\partial x} x^T A x = 2x^T A^T$$

This is an important special case, because both inner and outer product of matrices, i.e. $A^T A$ and AA^T, are always symmetric.

B. Use the method of numerical differentiation (see task 4) to verify this by comparing this analytical derivative with the numerical derivative of the quadratic form $x^T A x$ for some example vector x and example matrix A.