Universität des Saarlandes
Prof. Dr. Dietrich Klakow
Lehrstuhl für Signalverarbeitung
NNTI Winter Term 2024/2025

Exercise Sheet 7

Backpropagation and Computational Graphs

Deadline: 11.12.2024 23:59

Guidelines: You are expected to work in a group of 2-3 students. While submitting the assignments, please make sure to include the following information for all our teammates in your PDF/python script:

Name:

Student ID (matriculation number):

Email:

Your submissions should be zipped as **Name1_id1_Name2_id2_Name3_id3.zip** when you have multiple files. For assignments where you are submitting a single file, use the **same naming convention** without creating a zip. For any clarification, please reach out to us on the **CMS Forum**.
Note that the above instructions are mandatory. If you are not following them, tutors can decide not to correct your exercise.

Exercise 7.1 - Backpropagation (0.5 + 0.5 + 1 = 2 points)

Consider a simple 2-layer neural network with the following architecture:

- **Input Layer:** Dimension = 2
- **Hidden Layer:** 3 neurons, activation function = ReLU
- **Output Layer:** Dimension = 2, activation function = Sigmoid

The network's forward propagation is defined as:

$$h = \text{ReLU}(W_1 x + b_1)$$

$$y = \text{Sigmoid}(W_2 h + b_2)$$

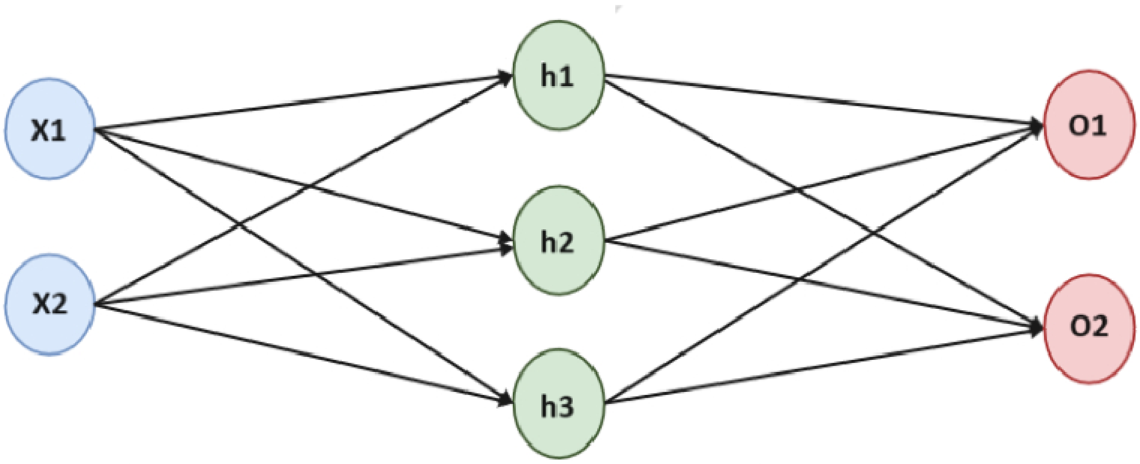
where:

- $x \in \mathbb{R}^2$: Input vector
- $W_1 \in \mathbb{R}^{3\times 2}$: Weights of the hidden layer
- $b_1 \in \mathbb{R}^3$: Biases of the hidden layer
- $W_2 \in \mathbb{R}^{2\times 3}$: Weights of the output layer
- $b_2 \in \mathbb{R}^2$: Biases of the output layer

The loss function is defined as:

$$L = -\frac{1}{2}\sum_{i=1}^{2}\left[t_i \log(y_i) + (1 - t_i)\log(1 - y_i)\right],$$

where $t \in \mathbb{R}^2$ is the true label vector, and $y \in \mathbb{R}^2$ is the predicted output.



Input and Output

Given:

$$x = \begin{bmatrix} 0.5 \\ 0.2 \end{bmatrix}, \quad t = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$W_1 = \begin{bmatrix} 0.1 & 0.4 \\ 0.2 & 0.5 \\ 0.3 & 0.6 \end{bmatrix}, \quad b_1 = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \end{bmatrix},$$

$$W_2 = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.5 & 0.3 & 0.4 \end{bmatrix}, \quad b_2 = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}.$$

Tasks

Perform the following:

a) Compute the forward pass to find the output y.

b) Compute the loss L.

c) Find the gradients of the loss L with respect to the following weights:

$$\frac{\partial L}{\partial W_1(1,1)}, \quad \frac{\partial L}{\partial b_1(2)}, \quad \frac{\partial L}{\partial W_2(1,1)}, \quad \frac{\partial L}{\partial b_2(1)}.$$

Exercise 7.2 - Computation Graph (0.5 + 0.5 = 1 points)

Consider the following mathematical expression:

$$z = (a + b) \cdot (c - d),$$

where a, b, c, and d are scalar parameters.

Tasks

a) Draw the computation graph for this expression, labeling all intermediate nodes.

b) Assume the loss function is defined as:

$$L = \frac{1}{2}z^2.$$

Compute the gradients of L with respect to all the parameters: a, b, c, and d as it would be done in backpropagation using the computation graph. Show every step.

Example Values

Use the following values for your computations:

$$a = 1, \quad b = 2, \quad c = 3, \quad d = 4.$$

Be sure to show the forward pass, the loss and the steps for backpropagation.

Exercise 7.3 - Backpropagation through sort (1 + 1 + 1 = 3 points)

Consider an input vector $x = (x_0, x_1, x_2, x_3, x_4)$, and the following functions:

- $F_1(x)$ is a sorting function that rearranges the components of x in the following order:
$$F_1(x) = (x_2, x_4, x_3, x_0, x_1).$$
- $F_2(x) = x_0 \cdot x$ (element-wise multiplication of x_0 with each element of x).
- $F_3(x) = x_0 \cdot F_1(x)$ (element-wise multiplication of x_0 with the output of $F_1(x)$).

Let the upstream gradient of the loss function with respect to F, denoted $\frac{\partial L}{\partial F}$, be:

$$\frac{\partial L}{\partial F} = (d_0, d_1, d_2, d_3, d_4).$$

Task

You need to compute the gradient of the loss function with respect to the input vector x, i.e., $\frac{\partial L}{\partial x}$, for each of the functions:

a) Compute $\frac{\partial L}{\partial x}$ for the function $F_1(x) = (x_2, x_4, x_3, x_0, x_1)$.

b) Compute $\frac{\partial L}{\partial x}$ for the function $F_2(x) = x_0 \cdot x$.

c) Compute $\frac{\partial L}{\partial x}$ for the function $F_3(x) = x_0 \cdot F_1(x)$.

For each case, show all your steps and thinking clearly.

Exercise 7.4 - Backpropagation Implementation (4 points)

See jupyter-notebook