

HOME ASSIGNMENT 01

Text Mining

Due by **23rd May, 2025 at 23:59 CET**

Sandipan Sikdar, Jonas Wallat, Huyen Nguyen

Summer Semester 2025

Submission Details

- Please submit a single compressed folder (.zip) containing one PDF file with all text responses and one Jupyter Notebook for the coding tasks. Please keep all generated outputs in the Jupyter Notebook so we can grade your submission without rerunning your code.
- For each answer/code snippet, clearly state what task you are referencing. If you are not confident in an answer, try nonetheless, as it is possible to receive partial points.
- You are encouraged to work in groups of up to three students. Add all team members' names and matriculation numbers at the top of the PDF file **and** your notebook.
- You may use online resources for help, but copying online resources is prohibited. Cite or link any resources you use.
- If you have any questions, create a post in the forum.

1 Gradient Descent [10 points]

1. In your own words, explain why we need gradient/derivative calculation during neural network training. (5 points)
2. Explain the role of the loss function in gradient descent. What happens if we choose a bad loss function for our task? (5 points)

2 Neural Network Basics [50 points]

In this task, you will create a 2-layer neural network from scratch using NumPy. The network will calculate $y = W_2 * ReLU(W_1 * x + b_1) + b_2$, where x is the input, W_* are weight matrices and b_* are bias terms. Do not use PyTorch or any similar framework. The given Jupyter Notebook includes the general setup. Complete the TODOs in the notebook. You may change any existing parts as long as you do not change the objective of the task (all parts below need to be present in some form) and the code remains easy to understand and well documented. Please show the step-by-step computation of any derivations in your PDF file. The point distribution is as follows:

- Initialization (5 points)
- Loss calculation (7 points)
- Gradient derivation & implementation (15 points)
- Calculation of output y (10 points)
- Training loop (Loss calculation and update step) (7 points)

- Plotting (6 points)

3 Embeddings [40 points]

1. In your own words, explain why we use embeddings. (4 points)
2. Complete the following tasks in the given Jupyter Notebook (36 points):
 - Perform intrinsic evaluation of embeddings
 - Create a word embedding-based classifier

See the tasks in the Jupyter Notebook for more information and point distribution.