

# NNTI Assignment 1

November 2, 2023

**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, as the case might be for the assignments. 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 **CMS Forum**.

## Exercise 1

This exercise will help you get familiar with NumPy library in python. The goal of this exercise is for you to explore the documentation of different NumPy libraries and their implementation details. <sup>1</sup>

- Write a Feed Forward pass of a simple Neural Network:
  - Randomly generate a 4x16 input matrix X
  - Randomly generate three weight matrices W1, W2, W3 of sizes: 16x16, 32x16 2x32
  - Perform the forward pass:  $((X \cdot W_1^T) \cdot W_2^T) \cdot W_3^T$

**3 points**
- **(Bonus)** Generate a random 4x4 matrix and use NumPy library to compute eigenvalues and eigenvectors of the randomly generated matrix.

## Exercise 2

This exercise will help you get familiar with PyTorch Dataset and Dataloader classes. The goal of the exercise is to implement your own Dataset and Dataloader classes for different types of

---

<sup>1</sup>While dealing with random functions, always remember to fix your seeds in the code. It helps in reproducibility of your result.

data: tabular and sequential.

**Part 1:**

For the first dataset in the zip file (by the name of Dataset1), you are presented by a JSON file containing a set of already tokenized phrases extracted from the 'sms\_spam' database from the 'Huggingface' platform. You are only required to load the tokenized sentences from the JSON file, implement a Dataset and Dataloader for them by following these constraints:

- You will be required to split your data into 2 datasets, one for the training and one for the testing.
- Your custom dataset class should subclass the Dataset class from PyTorch
- Your dataset class should have the required function(s) to iterate over the requested dataset i.e. len(), iter, init etc.
- Your custom dataloader class should subclass the Dataloader class from PyTorch
- Your implementation of the dataloader class should work with the objects returned by your custom dataloader class
- Your dataloader should return a tuple of input/output

**4 points**

**Part 2:**

For the second Dataset in the zip file (By the name Dataset2) you are presented by a csv file which comprises of the information on the symptoms and prognosis related to 11 vector-borne diseases. The symptoms serve as the input for the machine learning model, while the prognosis represents the output. It's essential to perform one-hot encoding on the prognosis column to transform it into numerical values. You can learn more about on-hot encoding here: [link](#)  
The same guidelines concerning dataset and dataloader from part 1 apply here as well.

You are encouraged to lookup the source code of Dataloader and Dataset class in PyTorch. The main aim of this exercise is to help you understand the working of Dataset and Dataloader classes as we shall be using them in future assignments.

**3 points**