Lab 3. Training Perceptrons & PyTorch

LING-581-Natural Language Processing 1

Instructor: Hakyung Sung

September 11, 2025

Minimal guidelines

- This tutorial is designed to provide hands-on experience with the concepts covered in class, including:
    - Training a single perceptron
    - Training a multi-layered perceptron (MLP)
    - PyTorch tutorial
- Grading will focus on the clarity and completeness of your responses in the ### TODO lines, rather than on the specific code you write (10 points).
- Your task is to walk through the provided code, explain your understanding, and apply what you have learned.
- To submit, please download your completed *.ipynb* file from Google Colab by selecting:
    - *File → Download → Download .ipynb*
- Submit the file with your name included in the filename (e.g., *Lab3_HakyungSung.ipynb*).

Section 1. Perceptron and linear classification

We will implement a **simple perceptron classifier** and see how it learns a linear decision boundary.

Update rule:

$$w_{\text{new}} \; = \; w_{\text{current}} \; + \; \eta \, (y - \hat{y}) \, x$$

- y: actual label (0 or 1), $\hat{y}$: predicted label
- η: learning rate, x: input vector

Grading (3 pts):

- Step function condition (1-1)   [1 pt]
- *gen_training_data* function (1-2)   [1 pt]
- Training (error definition)   [1 pt]

Note: Uses *NumPy* (cf. my short tutorial).

Section 2. Multi-layer perceptron (MLP)

We extend the perceptron to a **multi-layer perceptron (MLP)** with:

- 2 inputs
- 1 hidden layer (2 hidden nodes)
- 1 output node

This architecture can solve the **XOR problem**, which a single perceptron cannot (If you're curious to know more the details about the XOR problem, see tutorial).

Grading (3 pts):

- Step1 blanks   [1 pt]
- Complete training   [1 pt]
- Text on XOR cases   [1 pt]

Section 3. PyTorch tutorial

- Intro to *PyTorch* with a toy task.
- For a short primer: PyTorch: A Minimal Introduction.

Grading (4 pts):

· Quiz	[1 pt]
· Exercise	[1 pt]
· Complete part 1	[1 pt]
· Complete part 2	[1 pt]