SUBMISSION DEADLINE: 19.04.2023 @23.59

GIVEN

The assignment consists of 1 question. This file (‘assignment1.pdf’) describes the question, provided folder structure and submission folder structure.

The provided folder structure:

```
assignment1
├── assignment1.pdf
├── mlp.py
├── test1.py
├── test2.py
├── test3.py
└── test4.py
```

TO DO

- Complete the provided code in ‘mlp.py’
- Submit the completed code (‘mlp.py’) as a single zipped folder named with your student id (e.g. 1234567.zip).

Submission folder structure:

```
1234567
└── mlp.py
```

Backpropagation Implementation[1]

This assignment aims to help us better understand backpropagation by implementing it by ourselves. We will implement forward and backward pass for the one-hidden-layer MLP shown in Figure 1.

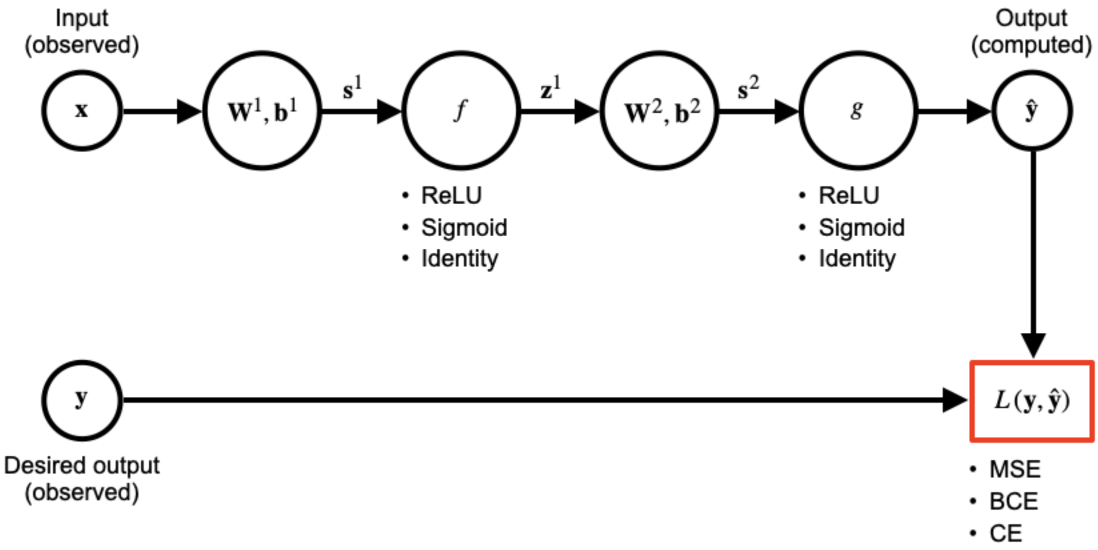

Figure 1: Multi-layer perceptron architecture.

The hidden and output layers, respectively, have point-wise non-linear activation functions f and g. Each of them can be one of the following, and we need to implement them.

- Identity
- ReLU
- Sigmoid

The network can be used for both regression and classification tasks. Therefore, we are required to implement the following loss functions.

- Mean Squared Error - MSE: $l_{mse}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{K}\sum_{k=1}^{K}(y_k - \hat{y}_k)^2$ where $\mathbf{y}, \hat{\mathbf{y}} \in \mathbb{R}^K$
- Binary Cross Entropy - BCE: $l_{bce}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{K}\sum_{k=1}^{K} -(y_k \log \hat{y}_k + (1 - y_k)\log(1 - \hat{y}_k))$ where $y_k, \hat{y}_k \in [0, 1]\ \forall_{k-1,\cdots,K}$
- Cross Entropy - CE: $l_{ce}(y^*, \hat{\mathbf{y}}) = -\log \frac{e^{\hat{y}_{y^*}}}{\sum_{k=1}^{K} e^{\hat{y}_k}}$ where $y^* \in \{0, 1, \cdots, K-1\}$ is the correct class index and $\hat{\mathbf{y}} \in \mathbb{R}^K$.

!!!BE CAREFULL!!!

- Submissions will be **auto-graded using an evaluation script**, which is similar to the given test files. Please make sure that your code successfully runs with the provided test#.py files. It is strongly recommended that you write your own test files as well.
- You are **NOT ALLOWED TO USE** PyTorch autograd!
- Be careful about the dimensionality! The shapes are provided inside the code!

[1] Adapted from Homework 1 of CSCI-GA 2572 Deep Learning by Professor Yann LeCun & Professor Alfredo Canziani, Spring 2021, NYU:https://drive.google.com/drive/folders/1g-uQNEi_NJyELGRMrJGXXxmARDabcXFd