**INT410: Advanced Pattern Recognition** **Fall 2020/21**

# Assignment 1: Neural Network

*Lecturer: Kaizhu Huang* *Unit: INT Dept. of XJTLU*

**Disclaimer**:

*1. Lab reports deadlines are strict. University late submission policy will be applied.*

*2. Collusion and plagiarism are absolutely forbidden (University policy will be applied).*

## 1.1 Objectives

- Implement the MLP algorithm
- In this experiment, we will use the publicly dataset to verify our algorithm. Download the UCI Flare dataset: https://archive.ics.uci.edu/ml/datasets/Iris

## 1.2 Estimation of Classification Methods

- ( **5 marks** ) Read the Flare dataset into a list and shuffle it with the **random.shuffle** method. Hint: fix the random seed (e.g. **random.seed(17)** ) before calling **random.shuffle**
- ( **5 marks** ) Split the dataset into five parts to do cross-fold validation: Each of 5 subsets was used as test set and the remaining data was used for training. The 5 subsets were used for testing rotationally for evaluating the classification accuracy.

## 1.3 MLP Algorithm

- ( **10 marks** ) The input feature vector is augmented with the 1 since

$$w^T x + w_0 = \begin{bmatrix} w^T & w_0 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix}$$

- ( **10 marks** ) The label $l_n$ of the n-th example is converted to a $L$ dimensional vector $t_n$ as follows ($K$ is the number of the classes)

$$t_{nk} = \begin{cases} +1, & k = l \\ 0, & k \neq l. \end{cases}$$

- ( **10 marks** ) Initialize all weight $w_{ij}$ of MLP network such as $w_{ij} \in \left[-\sqrt{\frac{6}{D+1+L}}, \sqrt{\frac{6}{D+1+L}}\right]$ where $D$ and $L$ is the number of the input nodes an the output nodes, respectively.

- ( **20 marks** ) Choose randomly an input vector $x$ to network and forward propagate through the network

$$\begin{aligned} a_j &= \sum_{i=0}^{d} w_{ji}^{(1)} x_i \\ z_j &= \tanh(a_j) \\ y_k &= \sum_{j=0}^{K} w_{kj}^{(2)} z_j \end{aligned} \tag{1.1}$$

  The error rate is $\frac{1}{2}\sum_{l=1}^{L}(y_l - t_l)^2$ for the example $x$.

- ( **10 marks** ) Evaluate the $\delta_k$ for all output units

$$\delta_k = y_k - t_k$$

- ( **10 marks** ) Backpropagate the $\delta$'s to obtain $\delta_j$ for each hidden unit in the network

$$\begin{aligned} \delta_j &= \tanh(a_j)' \sum_k w_{kj}\delta_k \\ &= (1 - z_j^2)\sum_k w_{kj}\delta_k \end{aligned}$$

- ( **10 marks** ) The derivative with respect to the first-layer and the second-layer weights are given by

$$\frac{\partial E_n}{\partial w_{ji}^{(1)}} = \delta_j x_i, \quad \frac{\partial E_n}{\partial w_{kj}^{(2)}} = \delta_k z_j$$

- The framework of MLP algorithm is as follows, where $\eta = 0.001$ and $K = 20$ is the number of hidden nodes. Note that $K$, $\eta$, $T$ are the hyperparameters of the network.

**Algorithm 1** Stochastic Backpropagation Algorithm

Initialize $w$, $\eta$
**for** $t = 1$ to $T$ **do**
  Shuffle the training data set randomly.
  **for** $n = 1$ to $N$ **do**
    Choose the pattern $x_n$
    Forward the input $x_n$ through the network
    Backward the gradient from the output layer through network to obtain $\frac{\partial E_n}{\partial w_{ji}^{(1)}}$ and $\frac{\partial E_n}{\partial w_{kj}^{(2)}}$
    Update the weights of the network

$$w_j k = w_{jk} - \eta \frac{\partial E_n}{\partial w_{kj}^{(2)}}, \quad w_{ij} = w_{ij} - \eta \frac{\partial E_n}{\partial w_{ji}^{(1)}}$$

  **end for**
**end for**
**return** $w$

- The algorithm may be terminated by setting the total iteration $T$ except that setting the threshold $\theta$ of the gradient referred in the lecture slide.

- ( **10 marks** ) In the test stage, the test example $x$ is forwarded into the network to obtain the output $y_{L\times 1}$ and then assigned to the label with the maximum output value.

## 1.4 Lab Report

- Write a short report which should contain a concise description of your results and observations.
- **Please insert the clipped running image into your report for each step.**
- Submit the report and the python source code electronically into LearningMall.
- The report are strongly suggested to be written with the latex typesetting language.
- The report in pdf format and python source code of your implementation should be zipped into a single file. The naming of report is as follows:

  e.g. StudentID_LastName_FirstName_LabNumber.zip (123456789_Einstein_Albert_1.zip)

## 1.5 Hints

Please refer to the lecture slides and PRML book (Section 4.1.6) for more details.

- Latex IDE: texstudio
- Python IDE: pycharm
- Use the python numpy library flexibly.