Name: _____Student ID:_____

# 11040A Neural Networks

**Assignment 4**

**Deadline: June 12, 2025 24:00**

1. Summarize the key idea of the BP algorithm briefly.

2. Consider an 1-M-1 multi-layer perceptron (MLP) whose input-output are denoted by $x(t), y(t)$ $t$=1,…,$N$ and the activation functions are $f_1(x) = \tanh(x)$ for hidden layer and $f_2(x) = x$ for output layer.

① Suppose there is teacher signal, $d(t), t = 1,..., N$, corresponding the input $x(t)$. The learning rate is μ. The loss function is defined as:

$$E = \frac{1}{2N}\sum_{t=1}^{N}(d(t) - y(t))^2$$

**(1)** Derive the backpropagation (BP) training algorithm and describe your derivation process. During your derivation, you should include the value of the parameters below:

$W_1$: the set of ω of hidden layer

$b_1$: the set of bias of hidden layer

$W_2$: the set of $\omega$ of output layer

$b_2$: the set of bias of output layer

$Z$: the set of outputs of the hidden layer (after passing through the activation function $f_1$)

$y(t)$: the output of the MLP of $t - th$ input $x(t)$

$\delta^1$: the set of δ of hidden layer

$\delta^2$: the set of $\delta$ of output layer

$\Delta W_1$: the set of $\mu \frac{\partial E}{\partial \omega}$ of hidden layer

$\Delta b_1$: the set of $\mu \frac{\partial E}{\partial b}$ of hidden layer

Name: _____Student ID:_____

$\Delta W_2$: the set of $\mu \frac{\partial E}{\partial \omega}$ of output layer

$\Delta b_2$: the set of $\mu \frac{\partial E}{\partial b}$ of output layer

**(2)** How many "δ"s are there in this MLP?

② Change the activation function for the hidden layer to $f_1(x) = ReLU(x)$. Train the MLP using the BP algorithm in the case of $M = 2, N = 1, x(1) = 0.7, d(1) = 0.68$, and learning rate $\mu = 0.2$. Refer to and fill out the following table.

*If you calculate this problem through programming, please attach your code.

| t | W1/b1 | W2/b2 | Z | y(1) | $\delta 1$ | $\delta 2$ | $\Delta W_1 / \Delta b_1$ | $\Delta W_2 / \Delta b_2$ |
|---|---|---|---|---|---|---|---|---|
| 0 | [[0.3 -0.3]] /[0.0 0.0] | [[-0.1] [0.1]] / [0.0] | [0.21 0] | [-0.021] | [-0.07 0.0] | [0.701] | [[-0.01 0.0]] / [-0.014 0.0] | [[0.029] [0.0]] / [0.14] |
| 1 | [[0.29 -0.3]] / [-0.014 0.0] | [[-0.071] [0.1]] / [0.14] | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| … | … | … | … | … | … | … | … | … |

**Hint:**
If you find this question is quite complex or are still confused about the BP algorithm, you may review what you have learned:
**Chapter 3: Neural Networks –Training; 3.2 BP Training Algorithm I**
We also provide the reference, which you can find in the references section of the course page:
**Hints for Backpropagation**

But please notice that:
(1) There are differences in some symbols between the course PPT and the reference.
(2) Each $\delta$ in the reference corresponds to the loss $C^t$ of one data point, while each $\delta$ in this question corresponds to the total loss $E$ of the entire batch of data. Therefore, if you refer to the reference, for each ω and $b$, you still need one more step to get the final result:

$$\nabla_{\omega,b} E = \nabla_{\omega,b} L(\theta) = \frac{1}{N} \sum_{t=1}^{N} \nabla_{\omega,b} C^t(\theta)$$

(3) In the course PPT, when representing the value of $\delta$, it excluded "−" from $\delta$ when calculating this part: $\frac{\partial E}{\partial y_k(t)}$ (The $\frac{\partial C^t}{\partial y_m^t}$ part of $\delta$ in reference). Therefore, the value of $\delta$ in the course PPT is opposite to the value of $\delta$ in the reference. For this question, both expressions will be considered correct. (Question (b) take the expression of the course PPT as the example.)

Name: _____Student ID:_____

# Programming part

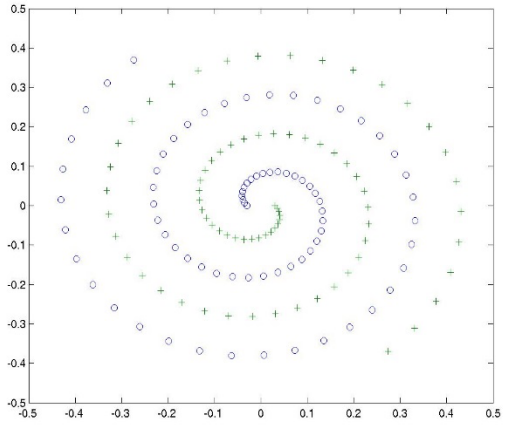
Please write a report on the following questions and send your report and source code to TA (Remember to pack them into a *.zip file). If you have any questions, feel free to contact the TA.

Note: to have a better understanding of BP, in this assignment, deep learning frameworks such as TensorFlow, Torch, Caffe, Theano etc. are **NOT** allowed to use. However, we suggest utilizing them in the next assignment.

We recommend that you use Python and only use the NumPy and Matplotlib libraries.

1. **SYSTEM: A Two-Nested-Spirals Problem**

   Two-Nest-Spirals problem is a well-known classification benchmark problem. It contains two nested spirals, 'o' and '+', as shown in figure. The task is to separate the two nested spirals.




2. **Purpose: Separating the Two Classes Using Neural Network Classifier**

   (1) Write a program to generate the data set.
   (2) Design a neural network classifier.
   (3) Define a loss function.
   (4) Implement BP algorithm and train the neural network. (Bonus: implement mini-batch, Momentum or other variants of BP)
      ① Discuss what difficulties you encountered and how they were resolved;
      ② You may show the error curve for training data.
   (5) After training the neural network, show in a figure (decision boundary) how well the trained neural network can separate the two spirals. (generalization ability)

Please write a report. The report should include your code & results and the explanation of the code.

(You can submit your programming files along with your report. Or you can organize your program into a markdown file or a Jupiter Notebook file as a report and submit it.)