

**University of Wollongong**  
**School of Computing and Information Technology**

**CSCI464/964**

**Computational Intelligence**

**Assignment 3**

**(Due: 11 PM**

**5 May)**

**10 marks**

**--- Part 1 (Self Organizing Map, 5 marks) ---**

**Aim:** This assignment is intended to provide basic experience in implementing self-organizing map (SOM). After having completed this assignment you should know how to realize an SOM network, understand its training process, and interpret the learned weights.

**Assignment Specification:**

1. A subset of the MNIST data set (<http://yann.lecun.com/exdb/mnist/>) is provided with this assignment in "SOM\_MNIST\_data.txt". It consists of 5,000 examples, each of which corresponds to one column of this text file. Each example has been reshaped from a 28 by 28 gray-level image into a 784-dimensional feature vector. You will be able to view the original images by reshaping each feature vector back and display the 28 by 28 matrix with appropriate image-processing software.
2. Read the lecture notes and other resources (for example, Chapter 9 of [1]) to review SOM. Basically, given a dataset, SOM aims to learn a set of prototypes of the data and spatially arrange the prototypes in a way that is indicative of the data distribution in the original input space. Implement an SOM neural network and train its weights with the provided dataset. The default size of the 2D lattice is 10 by 10. You can use a reasonably larger or smaller size according to the computational resource available to you. An example code written in Matlab is provided for your reference. Note that you are required to implement SOM in C++ by yourself and are **NOT** allowed to use this Matlab code for this assignment.  
[1] Neural Networks and Learning Machines (3<sup>rd</sup> Edition), Simon Haykin, Pearson, November 2008.
3. Write a report on this part. It shall include
  - 1) A brief introduction on the MNIST data set (read the above link) and the examples provided in this assignment;
  - 2) An introduction of the steps of training an SOM neural network. In particular, describe the two phases (ordering and convergence) of the training process and how to set the learning parameters in the two phases;
  - 3) The change of the weights between two consecutive epochs is indicative of the convergence of the training process. To characterize this change, for each weight vector compute the Euclidean distance between its values in the  $t$ -th and  $(t+1)$ -th iterations, and then use the sum of all the Euclidean distances as a criterion. Plot the value of this criterion with respect to the number of epochs and describe its evolution;
  - 4) Plot the learned weight vectors of the 2D lattices corresponding to the following three stages. The first one is at the initialization stage and the third one is at the convergence (or stable) stage, while the second one is in between. Each weight vector shall be plotted as a 28 by 28 image. Example figures of the first and last stages are provided in next page for your reference. Note that your figures are not necessarily same as the examples.
  - 5) You are encouraged to investigate various settings to train this SOM neural network, including the number of training examples, the size of 2D lattice, the learning rate, the size of neighborhood, and the number of epochs, etc. Provide detailed discussion and analysis of what you have observed and experienced.

## --- Part 2 (Deep Learning, 5 marks) ---

**Aim:** This assignment is intended to provide basic experience in using the recently released open source software library **TensorFlow** developed by Google Brain Team to implement and design simple neural networks for image classification. After having completed this assignment you should know how to realize a linear and convolutional neural network with TensorFlow, understand its training process, and interpret the classification result.

### Assignment Specification:

1. Install TensorFlow by following the instruction at <https://www.tensorflow.org/install/>. After that, verify it by following [https://www.tensorflow.org/install/install\\_mac#ValidateYourInstallation](https://www.tensorflow.org/install/install_mac#ValidateYourInstallation).
2. **(2 marks)** Run the simple multinomial logistic regression (a three-layer neural network without a hidden layer) by reading and following [https://www.tensorflow.org/get\\_started/mnist/beginners](https://www.tensorflow.org/get_started/mnist/beginners). Ensure that you can obtain the classification accuracy around 92%. Describe what this network does and the key steps (such as defining the networks, training, and test) with your own words. Vary the parameters like the number of training examples, the learning rate, and the batch size to observe the classification accuracy. Describe your observation.
3. **(3 marks)** Run the basic convolutional neural networks (a multi-layer neural network with hidden layers) by reading and following [https://www.tensorflow.org/get\\_started/mnist/pros](https://www.tensorflow.org/get_started/mnist/pros). Ensure that you can obtain the classification accuracy around 99%. Describe what this network does and the key steps (such as defining the networks, training, and test) with your own words. Vary the parameters like the patch size and the number of features (defined in [5, 5, 1, 32] and [5, 5, 32, 64]) to observe the classification accuracy. Describe your observation.
4. **(5 marks)** The following images are from the international competition on cell image classification hosted by International Conference on Pattern Recognition in 2014. The images have been pre-partitioned into training set (8701 images), validation set (2175 images), and test set (2720 images), which are provided with this assignment. In addition, a .csv file is enclosed. It contains the category of each image. This file consists of two columns: the first column is the image IDs of all the 13,596 images, and the IDs are consistent with the names of the images in the three sets; and the second column is the category of the cell image. **This is an open task.** You are expected to use the knowledge learned from this subject to achieve the classification accuracy on the test set as high as possible. Note that **you are free to use any classification techniques, toolboxes, software packages, and programming languages to accomplish this task.** For your reference, our research shows that the classification accuracy of 96% is achievable. In addition, a journal paper is enclosed in this assignment for your reference.
5. Write a report on the above three tasks of part 2. It shall include
  - 1) A brief introduction on the MNIST data set used in part 2;
  - 2) As indicated in the **point 2 above**, describe what this network does and the key steps (such as defining the networks, training, and test) with your own words. Vary the parameters like the number of training examples, the learning rate, and the batch size to observe the classification accuracy. Describe your observation;
  - 3) As indicated in the **point 3 above**, describe what this network does and the key steps (such as defining the networks, training, and test) with your own words. Vary the parameters like the patch size and the number of features (defined in [5, 5, 1, 32] and [5, 5, 32, 64]) to observe the classification accuracy. Describe your observation;
  - 4) For the open task in **point 4 above**:
    - a. Describe the classification technique, the toolboxes and the language you use;
    - b. Describe and plot the structure of the classification system with a diagram;
    - c. Describe how you process and prepare the image data when applicable;
    - d. Describe how you chose the parameters of this classification system;
    - e. Describe the training and validation accuracy and plot them when applicable;
    - f. Report the final classification accuracy on the test set;
    - g. Elaborate your observation during the course and provide detailed analysis.

## Submit:

Submit your program on UNIX via the submit command **before the deadline** and hand in your report with a cover page in the lecture.

**For part 1:** C/C++ code that is compilable and runnable on the UNIX platform; Before submitting your code check the format to ensure the format and newlines appear correct on UNIX. (Marks will be deducted for untidy or incorrectly formatted work.) To avoid formatting problems avoid using tabs and use 4 spaces instead of tab to indent you code. Make sure your file is named: **som.cpp**.

**For part 2:** Provide all the source code of your programs for this part (including points 2, 3 and 4). Since you may use programming languages other than C/C++ and other libraries or packages, **it is not required** that your code be runnable on the UNIX platform. **However, you may be asked to demonstrate** your program on your laptop or a computer that has all the needed environments.

**Put both part 1 and part 2 into a single report. Do not write two separated reports.**

Submit using the submit facility on UNIX ie:

**\$ submit -u login -c CSCI964 -a 3 assignment3.zip**

**where 'login' is your UNIX login ID.**

We will attempt to run your program of part 1 on banshee. If problems are encountered running your program, you may be required to demonstrate your program to the coordinator at a prearranged time. If a request for a demonstration is made and no demonstration is done, **a penalty of 2 marks (minimum)** will be applied. **Marks will be awarded for a comprehensive report, correct program design, implementation, style and performance.** Any request for an extension of the submission deadline must be made by applying for academic consideration before the submission deadline. Supporting documentation must accompany the request for any extension. Late assignment submissions without granted extension will be marked but the mark awarded will be reduced by 1 mark for each day late. Assignments will not be accepted if more than one week late.