Homework 5 – Learning

TA Jonghwan Hyeon (hyeon0145@kaist.ac.kr) Date assigned: Mon. 9 November 2015
TA Zae Myung Kim (zaemyung@kaist.ac.kr) Date due: Wed. 18 November 2015

Submit one zip file containing your report and program codes via KLMS (naming format of the zip file: <student#>_<name>.zip, for example "20150724_john.zip"). Late submissions will not be accepted. (Write your name and student ID in your report.) This homework should be done individually and written in English.

1. Bayesian networks (30 points)

(Source: Pattern Recognition and Machine Learning)

a	b	c	$p(a, b, c)$
0	0	0	0.192
0	0	1	0.144
0	1	0	0.048
0	1	1	0.216
1	0	0	0.192
1	0	1	0.064
1	1	0	0.048
1	1	1	0.096

A. Consider three binary variables $a, b, c \in \{0, 1\}$ having the joint distribution given in the above table. Show by direct evaluation that this distribution has the property that a and b are marginally dependent, so that $p(a, b) \neq p(a)p(b)$, but that they become independent when conditioned on c, so that $p(a, b \mid c) = p(a \mid c)p(b \mid c)$ for both $c = 0$ and $c = 1$. (15 points)

B. Evaluate distributions $p(a)$, $p(b \mid c)$, and $p(c \mid a)$ corresponding to the joint distribution given in the above table. Hence show by direct evaluation that $p(a, b, c) = p(a)p(c \mid a)p(b \mid c)$. Draw the corresponding directed graph. (15 points)

2. Decision tree (15 points)

A. Consider the following set of examples, each with six inputs and one target output. Run the decision tree learning rule, and show the resulting decision tree. (15 points)

$\mathbf{I}_1$	1	1	1	1	1	1	1	0	0	0	0	0	0	0
$\mathbf{I}_2$	0	0	0	1	1	0	0	1	1	0	1	0	1	1
$\mathbf{I}_3$	1	1	1	0	1	0	0	1	1	0	0	0	1	1
$\mathbf{I}_4$	0	1	0	0	1	0	0	1	0	1	1	1	0	1
$\mathbf{I}_5$	0	0	1	1	0	1	1	0	1	1	0	0	1	0
$\mathbf{I}_6$	0	0	0	1	0	1	0	1	1	0	1	1	1	0
T	1	1	1	1	1	1	0	1	0	0	0	0	0	0

3. Support vector machine (15 points)

A. Construct a support vector machine that computes XOR function. Use values of +1 and -1 (instead of 1 and 0) for both inputs and outputs, so that an example looks like ([-1, 1], 1) or ([-1, -1], -1). Map the input $[x_1, x_2]$ into a space consisting of x_1 and x_1x_2. Draw the four input points in this space, and the maximal margin separator. What is the margin? Now draw the separating line back in the original Euclidean input space. (15 points)

4. Artificial neural networks (40 points)

In this problem, you will implement a multi-layer feed forward neural network and the back propagation algorithm for classifying handwritten digit images. The image files are given in the homework zip file as "training.pickle", "validation.pickle" and "test.pickle". The data files are based on the original MNIST dataset. They have 10,000 training samples, 1,000 validation samples and 1,000 test samples, which all consist of pairs of a digit image and corresponding digit. For your convenience, we already implemented the part of reading the data from files.

For this problem, you need to implement the function indicated by "`FILL YOUR CODE HERE`". After you implement and execute the codes, the network will be evaluated and the program will output the error of the network.

To help us building the network, we will utilize the *numpy* module. NumPy is one of the most popular python module that provides common mathematical and numerical routines in pre-compiled and efficient manner.

For Windows environment, you can use *numpy* module by using WinPython

- http://winpython.github.io/

For Linux environment, you can install the *numpy* module by using the package manager such as *apt* and *yum*.

- Ubuntu and Debian: `apt-get install python-numpy python-scipy`
- Fedora: `yum install numpy scipy`

For Mac OSX environment, you can install the *numpy* module by using pip

- `pip install numpy scipy`

For more detailed information about installing *numpy*, please read the following instruction: http://www.scipy.org/install.html

Before implementing, you are encouraged to be familiar with *numpy* module by reading this document: http://www.engr.ucsb.edu/~shell/che210d/numpy.pdf. It can also be helpful to check the documentation: http://docs.scipy.org/doc/numpy/reference/

A. Implement sigmoid() function. (5 points)
B. Implement propagate() function. (10 points)
C. Implement backpropagate() function. (15 points)
D. Report the error of your network in the following format (5 points)

Iteration	Error
40000	
…	
49000	
Validation data	
Test data	

E. Tune the hyper-parameters (a size of the hidden layer, a learning rate) **to minimize the error on the validation data (not on the test data)** and report the best hyper-parameters with the error on both validation and test data (5 points)