Final Exam for Machine Learning, 6/25/2021

Carefully read the following items:

- ☐ I am on Teams so you can ask me questions
- ☐ Use A4 or white papers to write your answers (or you may directly use a text editing program) and write down your name and student ID on your answer sheets
- ☐ Take photos of (or scan) your answer sheets or send me your doc or docx file
- ☐ Make sure your photos are clear (sharp) enough for me to grade
- ☐ Combine all answer sheets into a single file
- ☐ Please name your file as <student_ID>_firstname_lastname
- ☐ E-mail your answer to both scyou@ntut.edu.tw & scyou@mail.ntut.edu.tw by 12:15PM
- ☐ Use your student ID as the subject of the e-mail
- ☐ Make sure you include the correct file
- ☐ You will receive an automatic reply after e-mail sent

Please show your work. That is, you need to provide the intermediate steps toward the answers. I cannot accept a reason like "the answers are directly obtained from a computer program, so I don't have the intermediate steps."

1. Consider the following training samples to be trained by Adaboost: (15%)
   - P1 = (1, 2), C = +1
   - P2 = (2, 1.5), C = +1
   - P3 = (1.5, 1) ,C = +1
   - P4 = (1, 1), C = -1
   - P5 = (2, 1), C = -1

   Find the weak classifier (based on a decision stump) in the 1st iteration and plot the decision tree. Hint: In the 1st iteration, all error weights are equal.

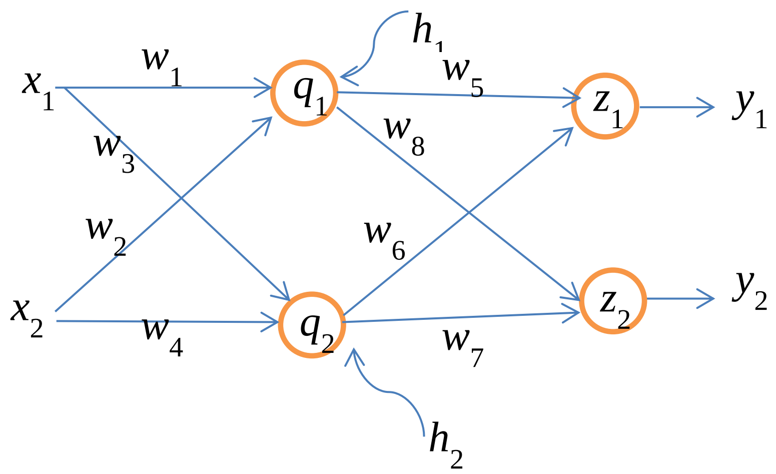
2. If we have 2-D data samples distributed uniformly inside a circle, what can be inferred on the computed principal components? Answer this question with the ratios of eigenvalues and directions of eigenvectors. (15%)

3. For the neural network given below, let $w_1 = 2.0$, w_2 to w_8 be 1.0, $d_1 = 1.0$, $d_2 = 0.0, \eta = 0.1, x_1 = 1.0$, and $x_2 = -1.0$. The activation function from q_1 to h_1 and q_2 to h_2 is ReLU, the activation function at the output nodes is linear (i.e., $y = z$), and the cost function is

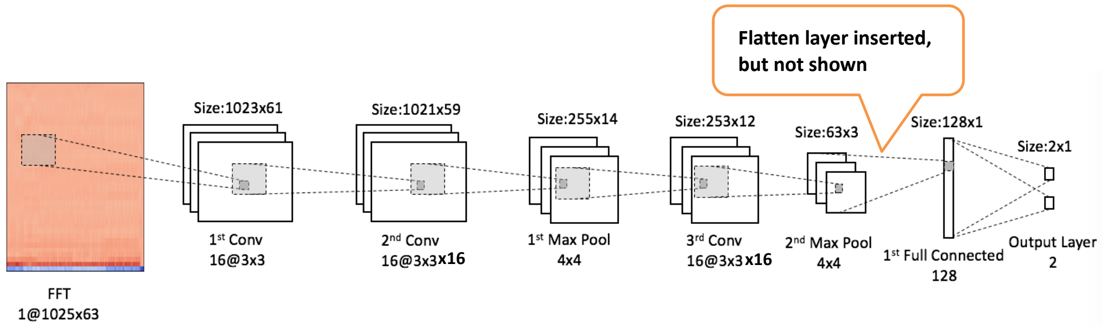
$$J = \frac{1}{2}\sum_{i=1}^{2}(y_i - d_i)^2.$$

(i) Find y_1 and y_2 (forward computation). (5%)

(ii) Find the value of $\Delta w_1 = \eta \frac{\partial J}{\partial w_1}$ by using the BP algorithm. (10%)



4. Compute the total number of non-unity *connections* (i.e., connections with weights not fixed to one) inside the following CNN. You need to figure out the number of feature maps in each layer. This number is NOT equal to 3. To simplify the problem, ignore the bias terms. From the size of each feature map you should have no problem to figure out if zero-padding is used or not. Note: 16@3x3 means 16 kernels with size of 3x3. (15%)



5. The CNN given in problem 4 serves as a binary classifier, but now the activation function in the output layer is changed to the softmax. Because the dataset already contains all of the available labeled samples (several thousands), it is not possible to increase the size of the dataset. Though the network has low training errors, unfortunately it has high validation errors. The followings are some possibilities considered to cope with this problem: (15%)

(i) Change the network structure. If the hyper-parameters of only one layer are allowed to change, which layer should be considered first and how to change the hyper-parameters?

(ii) Add/remove dropout layer. Will you add or remove dropout scheme during training? Why?

(iii) Change the objective (loss) function. Suppose originally the objective function is cross entropy. Do you expect to obtain large accuracy improvement if the objective function is switched to MSE? Why?

6. Suppose that a credit card company hires you to perform fraud detection. The dataset has 10,000 normal samples and 100 fraud samples, all from merchandise transactions. You decide to use autoencoder(s) for the detection problem. (15%)
   (i) How many models should be trained?
   (ii) What are the training samples for each model?
   (iii) How to partition the training, evaluation, and test sets?
   (iv) How to set the threshold to accept or reject an incoming transaction?

7. The following is a special case of a binary classification to detect received symbols. The probability density functions (pdf) of codes P and N with noise are given below. If the incoming signal (random value) *X* is greater than a threshold, the code is P; otherwise, the code is N. Based on the figure below, compute the following table for the classifier and then sketch the ROC curve. Note: Code P has a triangular pdf from -1 to 3, and code N has a uniform pdf from -3 to 1. Hint: To compute the probability (i.e., TPR or TNR), you need to perform integration, which is equal to calculate the area under a curve. (15%)

Threshold	-3	-2	-1	0	1	2	3
TPR							
FRR