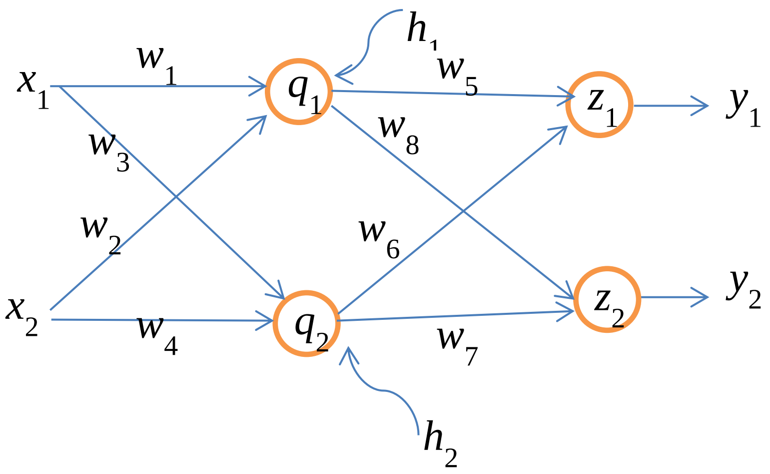
HW #7 Due: 6/7/2022

1. 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

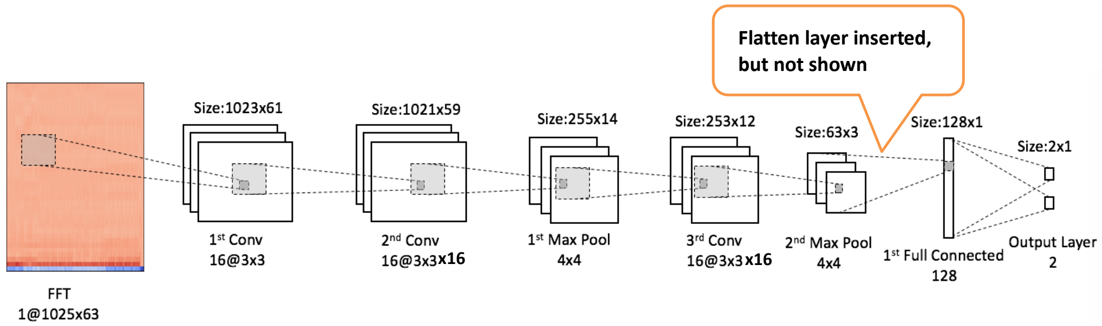
$$J = \frac{1}{2}\sum_{i=1}^{2}(y_i - d_i)^2.$$

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

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



2. Compute the total number of *trainable* weights in the following CNN. To do so, 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.



3. The CNN given in problem 2 is a binary classifier, and the activation function in the output layer is 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:

(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?

4. Build a 3-layer neural network by busing Keras to classify the Iris dataset. Vary the hidden units from 10 to 100 in the increment of 10 to observe the change of accuracy along with the number of hidden units. As usual, repeat the experiments 10 times to obtain the average accuracy. Use 10 epochs to train the network.
5. Build a modified LeNet-5, with original version shown below, by using Keras to classify the MNIST dataset. Note that the size of the images in the MNIST dataset is 28 x 28. The MNIST dataset is available in keras via tf.keras.datasets.mnist.load_data().Use 20 epochs to train the CNN.