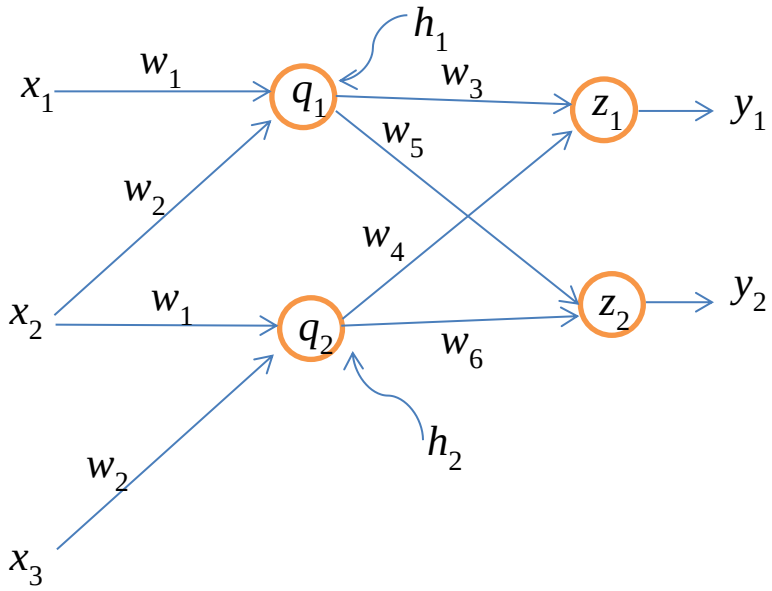
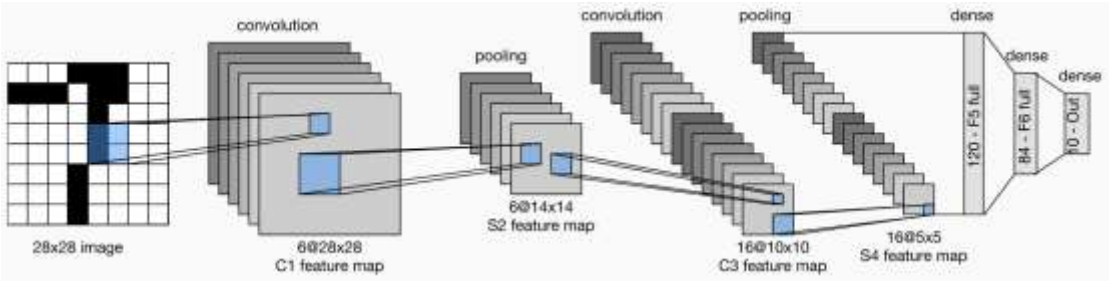
HW #7 Due: 6/7/2023

1. Find $\Delta w_1 = \eta \frac{\partial J}{\partial w_1}$ of the following CNN using backprop. The activation function from q_1 to h_1 and q_2 to h_2 is ReLU, the outputs y_1 and y_2 are softmax ouput, and the cost function is $J = -\log y_1$. Let w_1 to w_6 be 1.0, $x_1 = 0.5$, $x_2 = 1.0$, $x_3 = -0.5$, and $\eta = 0.1$.

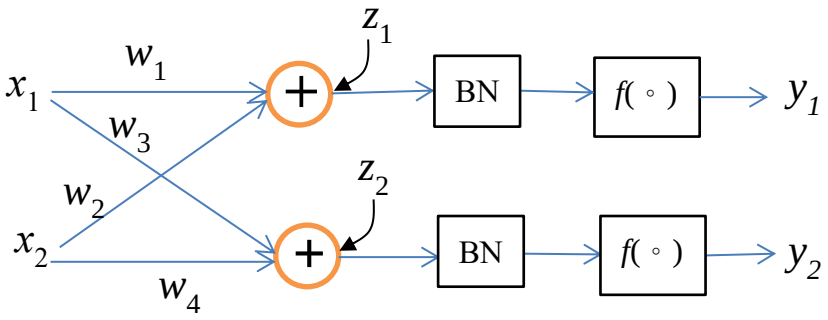


2. The following shows the LeNet-5 architecture. If we follow the modern design of the convolutional layers (taught in the lecture), compute the number of connections and trainable weights of the network. To compute the results, you need the following parameters for the convolutional layers: kernel size = 5×5, stride = 1, the first convolutional layer is padded, and the second convolutional layer is not padded. To simplify the computations, ignore the bias weights and let the output units be 10.



3. For the network shown below, derive the updating rule for β and γ in the batch normalization (pp. 16). Let the desired outputs be d_1 and d_2, and the mini-batch contains N samples. You may use notations of $x_1(i), y_1(i)$, etc. to indicate parameters associated with the i-th sample. To simplify the discussion,

let the activation function be sigmoid and the cost function is MSE.



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 the modified LeNet-5, shown on Problem 2, by using Keras to classify the MNIST dataset. Use maxpolling in the pooling layer, the ReLU in the middle layers, and the softmax in the output layer. The MNIST dataset is available in keras via tf.keras.datasets.mnist.load_data(). Use 20 epochs to train the CNN and report the average accuracy after 10 trials.