50.021 Artificial Intelligence
Homework 3

[Q1]. Consider the following CNN that has:

1. Input of 14×14, with 30 channels.
2. A convolutional layer C with 12 filters, each of size 4×4. The convolution zero-padding is 1 and the stride is 2, followed by a relu activation.
3. A max pooling layer P that is applied over each of the C's output feature maps, using 3×3 receptive fields and stride 2.

What is the total size of C's output feature map?
What is the total size of P's output feature map?

Now we want to compute the overhead of the above CNN in terms of floating point operation (FLOP). FLOP can be used to measure computer's performance. A decent processor nowadays can perform in Giga-FLOPS, that means billions of FLOP per second. Assume the inputs are all scalars (we have $14 \times 14 \times 30$ scalars as input), we have the computational cost of:

1. 1 FLOP for a single scalar multiplication $x_i \cdot x_j$
2. 1 FLOP for a single scalar addition $x_i + x_j$
3. $(n-1)$ FLOPs for a max operation over n items: $\max\{x_1, ..., x_n\}$

How many FLOPs layer C and P cost in total to do one forward pass?

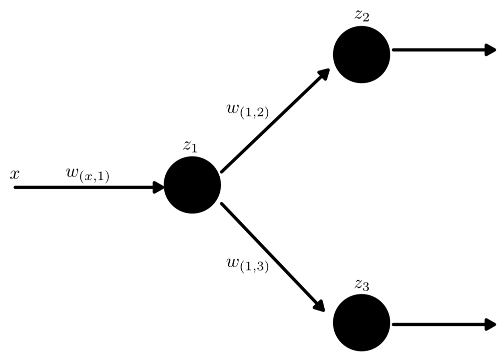

Figure 1: Mini Neural Network

[Q2]. Refer to the neural network at figure 1 with input $x \in \mathbb{R}^1$. The activation function for z_1, z_2, and z_3 is the sigmoid function: $\frac{1}{1+e^{-w \cdot x}}$,

$$h(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$z_1 = h(x \cdot w_{(x,1)}) \tag{2}$$

$$z_2 = h(z_1 \cdot w_{(1,2)}) \tag{3}$$

$$z_3 = h(z_1 \cdot w_{(1,3)}) \tag{4}$$

For the error E, instead of using the softmax function we learned in class, we use the quadratic error function for regression purpose,

$$E = \sum_{i \in data} ((z_2 - y_{2i})^2 + (z_3 - y_{3i})^2)$$

[6p] Write down an expression for the gradients of all three weights: $\frac{\partial E}{\partial w_{(x,1)}}, \frac{\partial E}{\partial w_{(1,2)}}, \frac{\partial E}{\partial w_{(1,3)}}$.