CSC420: Assignment #2

Wan-Jin Kim
(1003611424)

"In solving the questions in this assignment, I worked alone. I confirm that I have written the solutions / code / report in my own words.

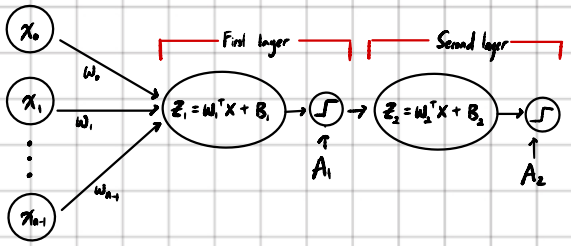
Part 1: Theoretical Problem

[Question 1] Activation Functions (10 marks)

Show that in a fully connected neural network with linear activation functions, the number of layers has effectively no impact on the network.

Hint: Express the output of a network as a function of its inputs and its weights of layers.

Q1:



: activation function
↳ $\sigma(z) = W_\sigma z + B_\sigma$

$$A_2 = W_\sigma(W_2^T(W_\sigma(W_1^T X + B_1) + B_\sigma) + B_2) + B_\sigma$$

$$= \underbrace{W_\sigma W_2^T W_\sigma W_1^T}_{W} X + \underbrace{W_\sigma W_2^T W_\sigma B_1 + W_\sigma W_2^T B_\sigma + W_\sigma B_2 + B_\sigma}_{B}$$

$$= WX + B$$

Even though we stacked two layers, it persist a form of $WX+B$ which could be considered as single layer perceptron. therefore, the number of layers has effectively no impact on the network.

[Question 2] Back Propagation (15 marks)

Consider the neural network architecture shown in the Figure below. Nodes denoted by x_i are input variables, and $\hat{y}$ is the output variable). The node Σ takes the sum of its inputs, and σ denotes the logistic function

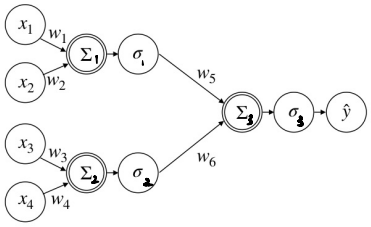
$$\sigma(x) = \frac{1}{1+e^{-x}}.$$

Suppose the loss function used for training this neural network is **L2** loss, i.e. $L(y, \hat{y}) = ||y - \hat{y}||_2^2$. Assume that the network has its weights set as:

$$(w_1, w_2, w_3, w_4, w_5, w_6) = (0.75, -0.63, 0.24, -1.7, 0.8, -0.2)$$

Given an input data point $(x_1, x_2, x_3, x_4) = (0.9, -1.1, -0.3, 0.8)$ with true label of 0.5, compute the partial derivative $\frac{\partial L}{w_3}$, by using the back-propagation algorithm. Round all your calculations to 4 decimal places.

Hint: The gradient of **L2** loss function $||y - \hat{y}||_2^2$ is $2||y - \hat{y}||$, and you do not need to write codes for this question! You can do it by hand.



Q2: $\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \Sigma_3} \frac{\partial \Sigma_3}{\partial \sigma_2} \frac{\partial \sigma_2}{\partial \Sigma_2} \frac{\partial \Sigma_2}{\partial w_3}$

Σ_1 : $x_1 w_1 + x_2 w_2 = (0.9)0.75 + (-1.1)(-0.63) = 1.368$

σ_1 : $\text{sigmoid}(\Sigma_1) = \frac{1}{1+e^{-\Sigma_1}} = 0.202943$

Σ_2 : $x_3 w_3 + x_4 w_4 = (-0.3)(0.24) + (0.8)(-1.7) = -1.432$

σ_2 : $\text{sigmoid}(\Sigma_2) = \frac{1}{1+e^{-\Sigma_2}} = 0.192787$

Σ_3 : $\sigma_1 w_5 + \sigma_2 w_6 = (0.202943)(0.8) + (0.192787)(-0.2)$
$= 0.123797$

σ_3 : $\text{sigmoid}(\Sigma_3) = \frac{1}{1+e^{-\Sigma_3}} = 0.530909$

$L(y, \hat{y}) = ||y - \hat{y}||_2^2 = ||0.5 - 0.530909||_2^2 = 0.000955$

$\frac{\partial L}{\partial \hat{y}} = 2||y - \hat{y}|| = 0.061818$

$\frac{\partial \hat{y}}{\partial \Sigma_3} = \hat{y}(1-\hat{y}) = 0.249044$

$\frac{\partial \Sigma_3}{\partial \sigma_2} = w_6 = -0.2$

$\frac{\partial \sigma_2}{\partial \Sigma_2} = \sigma_2 \times (1-\sigma_2) = 0.155620$

$\frac{\partial \Sigma_2}{\partial w_3} = x_3 = -0.3$

$\therefore \frac{\partial L}{\partial w_3} = (0.061818)(0.249044)(-0.2)(0.155620)(-0.3) = 0.00143$

$\frac{\partial L}{\partial w_3} = 0.0001$

[Question 3] Convolutional Neural Network (15 marks)

In this problem, our goal is to estimate the computation overhead of CNNs by counting the FLOPs (floating point operations). Consider a convolutional layer C followed by a max pooling layer P. The input of layer C has 50 channels, each of which is of size 12×12. Layer C has 20 filters, each of which is of size 4×4. The convolution padding is 1 and the stride is 2. Layer P performs max pooling over each of the C's output feature maps, with 3×3 local receptive fields, and stride 1.

Given scalar inputs $x_1, x_2, ..., x_n$, we assume:

- A scalar multiplication $x_i.x_j$ accounts for one FLOP.
- A scalar addition $x_i + x_j$ accounts for one FLOP.
- A max operation $\max(x_1, x_2, ..., x_n)$ accounts for $n - 1$ FLOPs.
- All other operations do not account for FLOPs.

How many FLOPs layer C and P conduct in total during one forward pass, with and without accounting for bias?

Q3:

C_{out}: $\frac{12+2(1)-4}{2} + 1 = 6$

∴ C_{out} has size $6 \times 6 \times 20$

# of FLOP(C) = $2C_i \times k^2 \times C_o \times W \times H$
w/ bias

$= (50 \times 4 \times 4 \times 2) \times (6 \times 6 \times 20)$

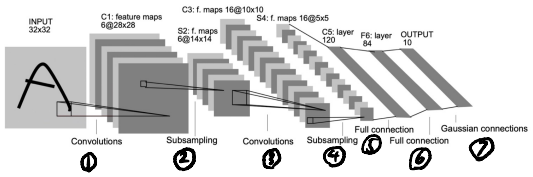
# of FLOP(P) = # of output × operation per output elem
$(4 \times 4 \times 20) \times (9-1)$

∴ with bias FLOP = $(50 \times 4 \times 4 \times 2) \times (6 \times 6 \times 20) + (8 \times 4 \times 4 \times 20)$
$= 1154560$

∴ without bias FLOP = $(50 \times 4 \times 4 \times 2 - 1) \times (6 \times 6 \times 20) + (8 \times 4 \times 4 \times 20)$
$= 1153840$

[Question 4] Trainable Parameters (15 marks)

The following CNN architecture is one of the most influential architectures that was presented in the 90s. Count the total number of trainable parameters in this network. Note that the Gaussian connections in the output layer can be treated as a fully connected layer similar to $F6$.



Q4: Convolution: padding = 0, stride = 1, filtersize = 5×5
Subsampling: padding = 0, stride = 2, filtersize = 2×2

# of param for Conv: (Width of filter × height of filter
× # of filters in prev layer + 1)
× # of filters.

# of param in pooling (subsampling): 0

" in Fully Connected layer: (Current layer neurons
× previous layer neurons)
+ Current layer neurons

TP_i = # of training parameters $i \in 0, ..., 7$

@ ① $TP_1 = (5 \times 5 \times 1 + 1) \times 6 = 156$

@ ② $TP_2 = 0$

@ ③ $TP_3 = (5 \times 5 \times 6 + 1) \times 16 = 2416$

@ ④ $TP_4 = 0$

@ ⑤ $TP_5 = (120)400 + 120 = 48120$

@ ⑥ $TP_6 = (84)120 + 84 = 10164$

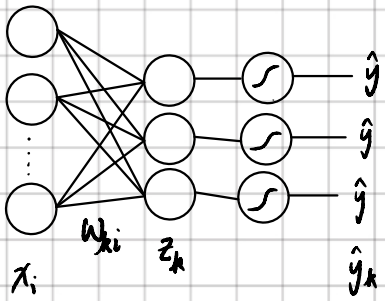
@ ⑦ $TP_7 = (10)(84) + 10 = 850$

$TP_{tot} = TP_1 + TP_2 + TP_3 + TP_4 + TP_5 + TP_6 + TP_7$
$= 156 + 2416 + 48120 + 10164 + 850$
$= 61706$

[Question 5] Logistic Activation Function (10 marks)

Show that for backpropagation in a neural network with logistic activation function, as long as we have the output of the neurons, there is no need for the inputs.

Hint: Find the derivative of a neuron's output with respect to its inputs.



$$\frac{\partial \mathcal{L}}{\partial w_{ki}} = \frac{\partial \mathcal{L}}{\partial \hat{y}_K} \frac{\partial \hat{y}_k}{\partial z_k} \frac{\partial z_k}{\partial w_{ki}}$$

assume Loss is MSE $\quad \frac{\partial \mathcal{L}}{\partial \hat{y}_k} = \hat{y}_K^{(n)} - t_K^{(n)}$ (t: target)

where $\hat{y}_k^{(n)} = \sigma(z_k^{(n)}) = \frac{1}{1+e^{-z_k^{(n)}}}$

$$\frac{\partial \hat{y}_k^{(n)}}{\partial z_k^{(n)}} = \hat{y}_k^{(n)}(1-\hat{y}_k^{(n)})$$

$$\frac{\partial \mathcal{L}}{\partial w_{ki}} = \sum_{n=1}^{N} \frac{\partial \mathcal{L}}{\partial \hat{y}_k^{(n)}} \frac{\partial \hat{y}_k^{(n)}}{\partial z_k^{(n)}} \frac{\partial z_k^{(n)}}{\partial w_{ki}}$$

$$= \sum_{n=1}^{N} (\hat{y}_k^{(n)} - t_k^{(n)}) \hat{y}_k^{(n)}(1-\hat{y}_k^{(n)}) x_i^{(n)}$$

thus Gradient descent update rule:

$$w_{ki} \leftarrow w_{ki} - \eta \frac{\partial \mathcal{L}}{\partial w_{ki}} = w_{ki} - \eta \sum_{n=1}^{N} (\hat{y}_k^{(n)} - t_k^{(n)}) \hat{y}_k^{(n)}(1-\hat{y}_k^{(n)}) x_i^{(n)}$$

■

[Question 6] Hyperbolic Tangent Activation Function (15 marks)

One alternative to the logistic activation function, is hyperbolic tangent function:

$$\tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}.$$

- (a) What is the output range for this function, and how it differs from the output range of logistic function?
- (b) Show that its gradient can be formulated as a function of logistic function; and
- (c) When do we want to use each of these activation functions?

Q6:

(a) logistic function's range : $(0, 1)$

tanh's range : $(-1, 1)$

(b) $\tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$

$$\tanh'(x) = \frac{d}{dx}\left(\frac{1-e^{-2x}}{1+e^{-2x}}\right)$$

$$= \frac{\frac{d}{dx}(1-e^{-2x})(1+e^{-2x}) - (1-e^{-2x})\frac{d}{dx}(1+e^{-2x})}{(1+e^{-2x})^2}$$

$$= \frac{2e^{-2x}(1+e^{-2x}) + 2e^{-2x}(1-e^{-2x})}{(1+e^{-2x})^2}$$

$$= \frac{2e^{-2x}+2e^{-2x}e^{-2x}+2e^{-2x}-2e^{-2x}e^{-2x}}{(1+e^{-2x})^2}$$

$$= \frac{4e^{-2x}}{(1+e^{-2x})^2} = 4\left(\frac{1+e^{-2x}-1}{(1+e^{-2x})^2}\right)$$

$$= 4\left(\frac{1+e^{-2x}}{(1+e^{-2x})^2} - \frac{1^2}{(1+e^{-2x})^2}\right)$$

$$= 4\left(\frac{1}{1+e^{-2x}} - \left(\frac{1}{(1+e^{-2x})}\right)^2\right)$$

$$= 4\,(\text{sigmoid}(2x) - \text{sigmoid}^2(2x))$$

where sigmoid is logistic function.

(c) tanh activation is rescaled version of logistic function. However the performance that is induced from each network is different. Logistic function has maximum gradient of $\frac{1}{4}$, which make error calculation difficult as layer increases, causing vanishing Gradient problem. tanh is slightly better improvement over logistic by moving the center of the graph to 0. however problem still persists.