CS4248
AY 2022/23 Semester 1
Tutorial 4

1. A perceptron F receives inputs $x_1, \ldots, x_n$ and outputs the following:

$$F(x_1, \ldots, x_n) \quad = \quad \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + \cdots + w_n x_n > 0 \\ 0 & \text{otherwise} \end{cases}$$

(a) Give 3 weights w_0, w_1, w_2 such that F implements the Boolean function $x_1 \vee x_2$.

(b) Give 3 weights w_0, w_1, w_2 such that F implements the Boolean function $\neg x_1 \wedge x_2$.

2. Consider a neural network defined as follows:

$$s_1 = \begin{bmatrix} i_1 & i_2 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

$$h_1 = \tanh(s_1) = \frac{e^{s_1} - e^{-s_1}}{e^{s_1} + e^{-s_1}}$$

$$\begin{bmatrix} o_1 & o_2 & o_3 \end{bmatrix} = \begin{bmatrix} h_1 & 1 \end{bmatrix} \begin{bmatrix} w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$$

$$L = (o_1 - t_1)^2 + (o_2 - t_2)^2 + (o_3 - t_3)^2$$

where $[i_1 \quad i_2]$ is the input vector, $[o_1 \quad o_2 \quad o_3]$ is the output vector, $[t_1 \quad t_2 \quad t_3]$ is the target output vector, L is the loss function, and $w_1, w_2, \ldots, w_9$ are the weight parameters to be learned.

The weights w_i are initialized as follows:

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
0.25	0.25	0.1	0.1	-0.1	0.2	0.3	-0.2	-0.3

The learning rate is 0.5. The neural network is given one training example as follows:
$i_1 = 0.2, i_2 = 0.8, t_1 = 1, t_2 = 0, t_3 = 0$.

(a) Derive an expression for $\frac{\partial L}{\partial w_3}$ in terms of (some of) $i_1, i_2, o_1, o_2, o_3, t_1, t_2, t_3, h_1, w_1, w_2, \ldots, w_9$. Show clearly the steps of your derivation and provide appropriate justification.

(b) For the given training example, carry out forward computation to compute the value of the loss function L. Show clearly all your intermediate calculations.

(c) Use backpropagation to compute the weight w_3 after one iteration of weight update. Show clearly the steps of your calculation.

3. Consider the use of dropout in neural network training when the non-linear activation function is tanh. Dropout applies a random masking vector $\boldsymbol{m}$ to a hidden layer vector $\boldsymbol{h}$, as follows:

$$\boldsymbol{h} = \tanh(\boldsymbol{xW} + \boldsymbol{b})$$
$$\widetilde{\boldsymbol{h_1}} = \boldsymbol{m} \odot \boldsymbol{h}$$

where $\odot$ is element-wise multiplication, and

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

John proposes a different formula for computation:

$$\widetilde{\boldsymbol{h_2}} = \tanh(\boldsymbol{m} \odot (\boldsymbol{xW} + \boldsymbol{b}))$$

(a) Is John's proposal a correct approach of implementing dropout? Give your justification. Be as precise as possible.

(b) Suppose the nonlinear activation function is changed from tanh to the sigmoid function σ:

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

Is John's proposal a correct approach of implementing dropout? Give your justification. Be as precise as possible.