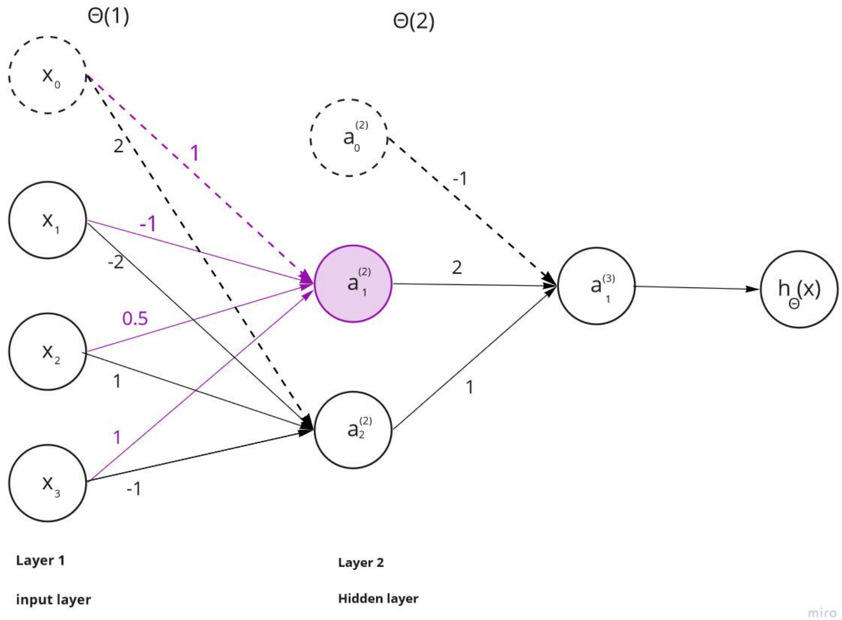
Neural networks Tut 1
Student: 2128753 - Jia Lin Lu

## Question 1

a)




b) $f(x_1, x_2, x_3) = \sigma(-1 + 2\sigma(1 - 1x_1 + 0.5x_2 + 1x_3) + 1\sigma(2 - 2x_1 + 1x_2 - 1x_3))$

c) i) $f(0, 0, 0) = \sigma(-1 + 2\sigma(1) + \sigma(2)) = \sigma(-1 + 1{,}462117 + 0{,}88079)$
$= 0{,}79297$

ii) $f(3,2,1) = \sigma(-1 + 2\sigma(0) + \sigma(-3)) = \sigma(-1 + 1 + 0{,}0474..)$
$= \sigma(0{,}047\ldots) = 0{,}51185$

iii) $f(-1,1,-1) = \sigma(-1 + 2\sigma(1.5) + \sigma(6)) = \sigma(-1 + 1.6351\ldots + 0{,}9975\ldots)$
$= \sigma(1{,}6326\ldots) = 0{,}8365$

d)

$$\Theta^{(1)} = \begin{bmatrix} 1 & -1 & 0.5 & 1 \\ 2 & -2 & 1 & -1 \end{bmatrix} \quad \Theta^{(2)} = \begin{bmatrix} -1 & 2 & 1 \end{bmatrix}$$

$$\text{i) } x = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{ii) } x = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \quad \text{iii) } x = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$$

$$a_1^{(2)} = \sum_{i=1}^{4} = \Theta^{(1)}_{1i} x_i$$

$$a_2^{(2)} = \sum_{i=1}^{4} \Theta^{(1)}_{2i} x_i$$

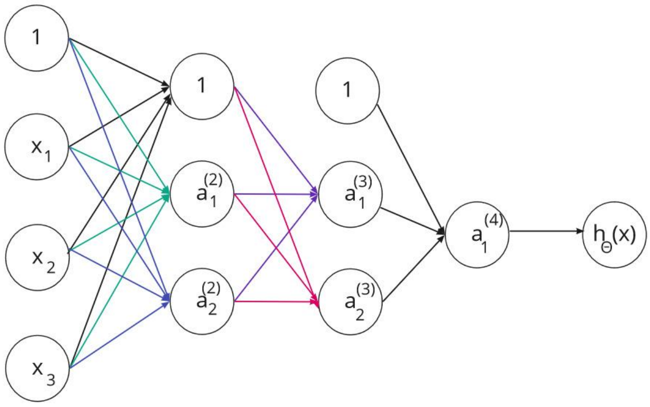
$$a_{1}^{(3)} = \sum_{i=1}^{3} \Theta^{(2)}_{1i} a_i$$

i) $a^{(2)}_1 = 1$
$a^{(2)}_2 = 2$
$h_\Theta(x) = 3$

ii) $a^{(2)}_1 = 0$
$a^{(2)}_2 = 0$
$h_\Theta(x) = 0$

iii) $a^{(2)}_1 = 1.5$
$a^{(2)}_2 = 6$
$h_\Theta(x) = 8$

# Question 2



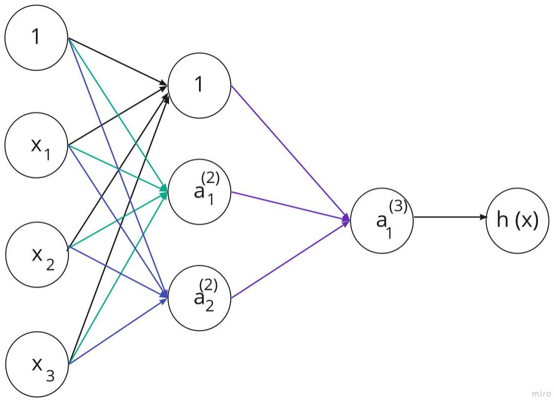

$$\Theta^{(1)} = \begin{bmatrix} 1 & 2 & 2 & 3 \\ 2 & 1 & 1 & 1 \end{bmatrix} \quad \Theta^{(2)} = \begin{bmatrix} 3 & 2 & 1 \\ 1 & 2 & 1 \end{bmatrix} \quad \Theta^{(3)} = \begin{bmatrix} 2 & 2 \end{bmatrix}$$

$$x = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$$

The results from this 4 layered neural network with linear activation functions is:

$$a^{(2)} = \begin{bmatrix} 15 \\ 8 \end{bmatrix} \quad a^{(3)} = \begin{bmatrix} 41 \\ 39 \end{bmatrix} \quad a^{(4)} = h_\Theta(x) = \begin{bmatrix} 160 \end{bmatrix}$$

Then I used the same inputs in a 3 layered neural network but with adjusted weights:




$$\Theta^{(1)} = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 2 & 3 & 2 & 2 \end{bmatrix} \quad \Theta^{(2)} = \begin{bmatrix} 4 & 5 & 1 \end{bmatrix}$$

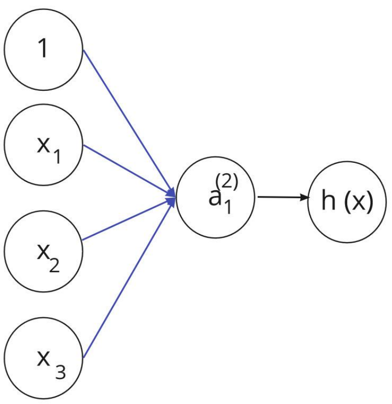
$$x = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$$

The results were as follows:

$$x = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} \quad a^{(2)} = \begin{bmatrix} 28 \\ 16 \end{bmatrix} \quad a^{(3)} = h_\Theta(x) = \begin{bmatrix} 160 \end{bmatrix}$$

It is clear that both have the same output. This is because the linear activation function outputs whatever was given as output - so the weight * input. This proves that a n layer linear neural network can compute the same function as a n-1 layer neural network.

Furthermore, this is why multilayered linear neural networks can be merged into one layer.

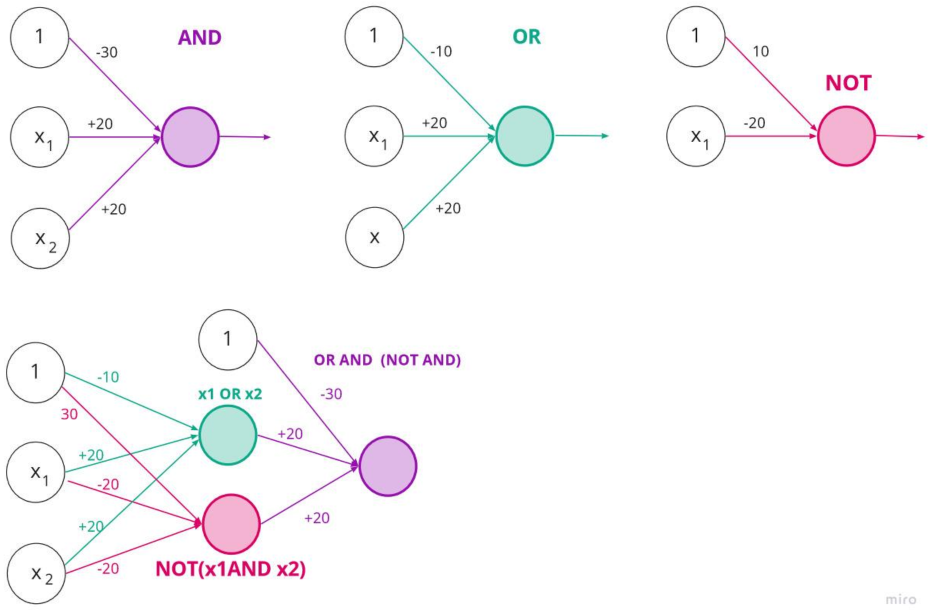



$$\Theta^{(1)} = \begin{bmatrix} 20 & 30 & 20 & 20 \end{bmatrix}$$

$$x = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$$

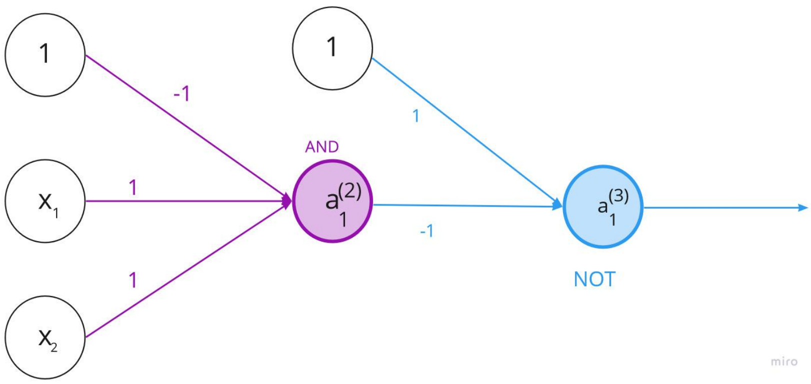
$$a^{(2)} = h_\Theta(x) = \begin{bmatrix} 160 \end{bmatrix}$$

## Question 3




## Question 4

Activation function for the inputs is a linear function.
Activation function for the hidden layer was a Binary threshold function with a threshold of 1.




$$w(1) = \begin{bmatrix} -1 & 1 & 1 \end{bmatrix} \quad w(2) = \begin{bmatrix} 1 & -1 \end{bmatrix}$$

$$x = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad a^{(2)} = \begin{bmatrix} -1 \end{bmatrix} \quad h_\Theta(x) = \begin{bmatrix} 1 \end{bmatrix} \qquad x = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad a^{(2)} = \begin{bmatrix} 0 \end{bmatrix} \quad h_\Theta(x) = \begin{bmatrix} 1 \end{bmatrix}$$

$$x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad a^{(2)} = \begin{bmatrix} 0 \end{bmatrix} \quad h_\Theta(x) = \begin{bmatrix} 1 \end{bmatrix} \qquad x = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad a^{(2)} = \begin{bmatrix} 1 \end{bmatrix} \quad h_\Theta(x) = \begin{bmatrix} 0 \end{bmatrix}$$

## Question 5

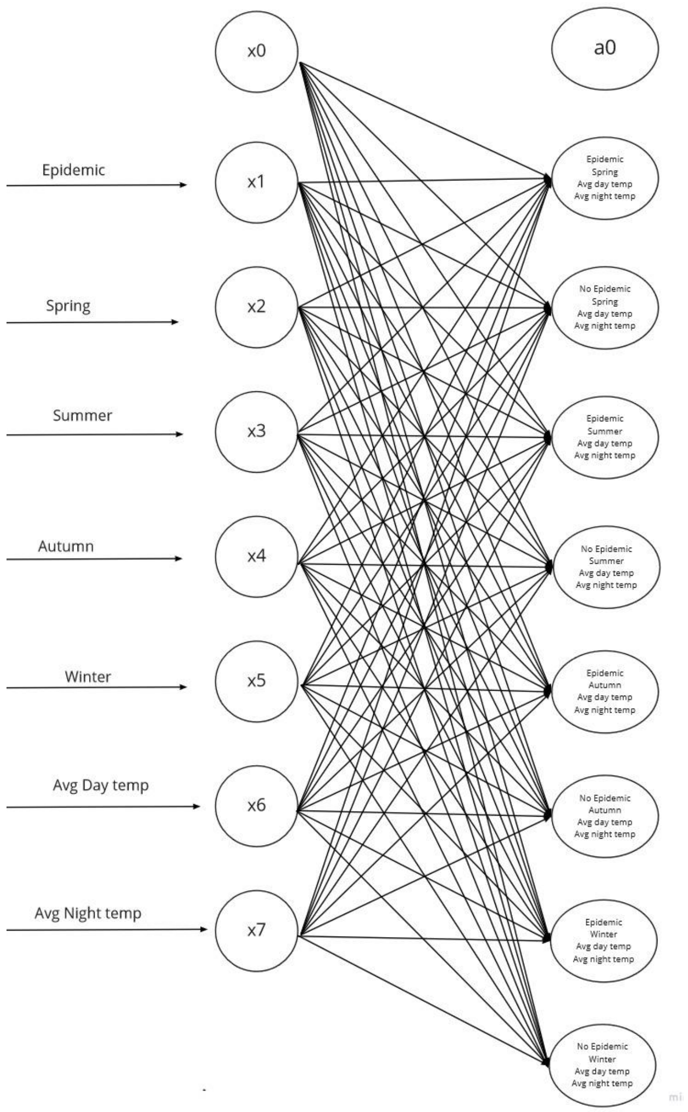
a) There are many ways to approach this question as there can be different ways to process the data before feeding it into the neural network and how many layers are needed. This can change depending on the approach one chooses.

I see this as a problem which can be solved with a forward propagation with ReLu's and a backward propagation to calculate the error of the weights.

The data that would be inputs to the neural network will be the average day temperature, average night temperature, season (spring, summer, autumn and winter) and if there was an epidemic or not.

The reasoning is that those are the factors of whether people are more likely to need to stay in hospital due to illness or not.

The reason why day and night temperatures need to be separated is because if it is warmer during the day and cold at night the likelihood of getting sick is higher than if it was cold the whole day. In winter even if people don't get sick the geriatric ward may be busy because in winter joint related problems may flare up and blood pressure may rise. In Spring and Autumn allergies and hayfever may be a concern. In summer heat related issues may flare up.




Epidemics are more common in colder weather but they certainly can happen during the warmer months of the year.

So with that the neural network will have 7 input nodes.
Epidemic and the seasons take values of 1 or 0 for the binary of yes and no.
Temperatures will be the real values of the temperature.

The first hidden layer will have 9 nodes.
Beyond the first hidden layer I am not sure how many more are needed.

This model will have a single output node which will be the number of beds needed.

Ultimately I have no way to assume if this model would work or not without programming this and testing the weights and activation functions used.

b) The modified network will have the same input values and hidden layers will mostly stay the same. The output layer will be what is different.

This network's output layer will have three nodes.

The possible outputs will be

$h_{\Theta}(x)$=[ 1, 0 , 0] - the hospital is under full

$h_{\Theta}(x)$=[ 0, 1 , 0] - the hospital is over full

$h_{\Theta}(x)$=[ 0, 0 , 1] - the hospital is at capacity.

Which means the last activation function will be a binary threshold function.

The reason why that is used instead of say a sigmoid function is because we want the output to tell us if the hospital is full or not and not the probability the hospital is full or not given the inputs (current conditions).

If there is uncertainty in the output there would simply be more than one 1 occurring in the output