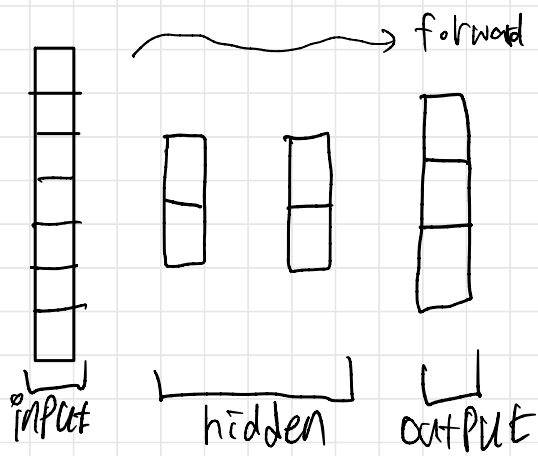
IIE4123 HW1
20191930l6 Kichang Lee
School of Integrated Technology, Yonsei Univ

1. Consider a neural network used to predict the course of an illness into one of three classes {severe, mild, none} based on 7 input variables. Use a network with 2 hidden layers and 2 hidden nodes in each hidden layer. (30 points)

   (a) Draw the appropriate network structure (you may omit constant nodes)



   (b) Clearly define output variables and provide the expression that the network minimizes as a function of the output and input variables.

$X \in \mathbb{R}^{1\times 7}$: input $W^{(1)} \in \mathbb{R}^{7\times 2}$: input → 1st hidden $b^{(1)} \in \mathbb{R}^{1\times 2}$: 1st bias

$W^{(2)} \in \mathbb{R}^{2\times 2}$: 1st → 2nd hidden layer $b^{(2)} \in \mathbb{R}^{1\times 2}$: 2nd bias $W^{(3)} \in \mathbb{R}^{2\times 3}$: 2nd → out $b^{(3)} \in \mathbb{R}^{1\times 3}$

$f_1(X), f_2(X), f_3(X)$: activation functions

$$\hat{Y} = f_3\left(W^{(3)} \cdot f_2\left(W^{(2)} \cdot f_1\left(XW^{(1)} + b^{(1)}\right) + b^{(2)}\right) + b^{(3)}\right) \quad \hat{Y} \in \mathbb{R}^{1\times 3}$$: output vector

$$L(Y, \hat{Y}) = \frac{1}{3}\sum (Y - \hat{Y})^2$$

$\frac{1}{3}$ → # of classes; $\sum$ → sum of all element

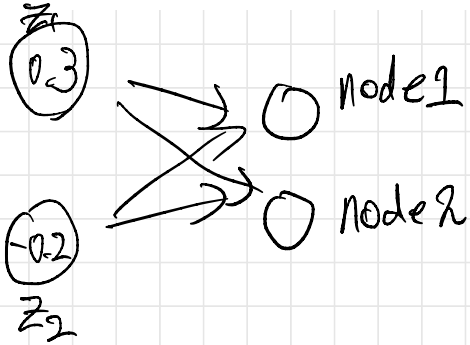
$$\rightsquigarrow \arg\min_{\theta} L(Y, f(X, \theta))$$

when θ is the parameters of the Network.

2. Consider a network with 2 hidden nodes and 2 output nodes that is used for a two-class classification problem (with classes yes and no). Output node 1 and 2 correspond to classes yes and no, respectively. Assume that for the current row of data the values at the hidden nodes are $z_1 = 0.3$, $z_2 = -0.2$ and the actual class is yes. Assume linear output functions at the output nodes with weights between the hidden nodes and the output nodes given in the following table. (40 points)

Hidden node weights	Output node1	Output node2
Constant	0.2	0.6
1	0.6	0.7
2	0.4	-0.2

(a) Calculate the output at each output node for the current row of data.



$$[0.3 \ -0.2]\begin{bmatrix} 0.6 & 0.7 \\ 0.4 & -0.2 \end{bmatrix} + [0.2 \ \ 0.6]$$

$$= [0.1 \ \ 0.25] + [0.2 \ \ 0.6] = [0.3 \ \ 0.85]$$

(b) The current row of data would be assigned to what class?

Class "NO" since $0.3 < 0.85$

(c) Calculate the value contributed to the usual loss function $L = \sum_{i=1}^{n} \sum_{k=1}^{2} (y_{ik} - f(x_i, \theta))^2$ from the current row of data

the actual class is "Yes" so, $Y = [1\ 0]$

the function $L = \sum_{i=1}^{n} \sum_{k=1}^{2} (y_{ik} - f(x_i, \theta))^2$

$\sum \|Y - \hat{Y}\|^2 = (1 - 0.3)^2 + (0 - 0.85)^2$

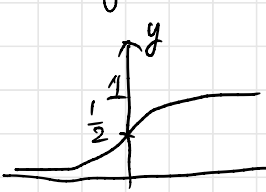
$= 1.2125$

3. Please answer following questions with one or two sentences. (30 points)

(a) What is the activation function? Please give the list of activation function.

Based on my understanding, the activation function gives a 'criteria' to the model whether the node is enough to 'fire' or not. The activation function is a tool to add some non-linearity in the model. Since the linear function's linearity and homogeneity, the layered linear functions can be re-writed into another one single linear function. Therefore we can not take advantage of deep and layered architecture The most famous activation functions are followed.

① Sigmoid

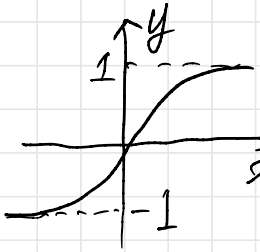


$$\sigma(x) = \frac{1}{1+e^{x}}$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

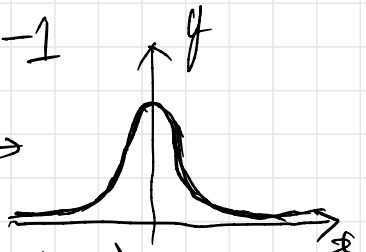
→ It has Vanishing gradient Problem

② tanh



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

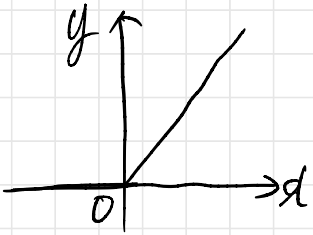
$$\tanh'(x) = 1 - \tanh^2(x) \rightarrow$$



$$= (1 + \tanh(x))(1 - \tanh(x))$$

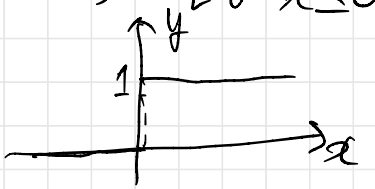
→ It also has Vanishing gradient Problem

③ ReLU



$$R(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

$$R'(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

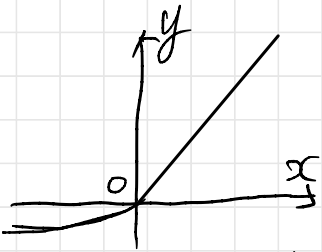


→ ReLU solved the Vanishing gradient Problem but it Cause a problem Called "dying ReLU"

④ Leaky ReLU & ELU

$$\text{Leaky ReLU}(x) = \begin{cases} x & x > 0 \\ \alpha x & x \leq 0 \end{cases}$$

(∵ α is a Coefficient closed to 0)



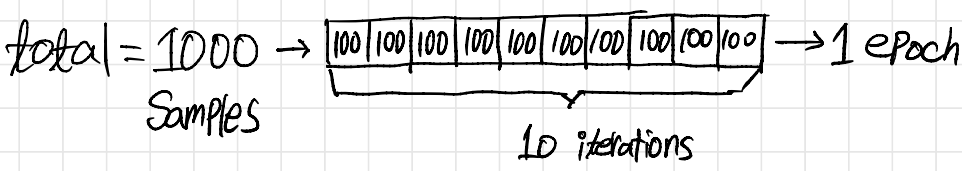
$$ELU(x) = \begin{cases} x & x > 0 \\ \alpha(e^{x} - 1) & x \leq 0 \end{cases}$$

These two functions tried to solve the Dying ReLU Problem

(b) What is the epoch?

A epoch means a generation of training the dataset To elaborate, if we say "I've trained the model for 10 epochs", the model trained trained the whole Dataset 10 times.

(c) Given 1000 samples, you have set the batch size as 100 and set the epoch as 10. How many times you will update weights? In order words, what it the total number of iterations in the training?



Since we have 10 epochs the total number of iteration is $10 \times 10 = 100$ (times). Which means there are 100 weights updates.

(d) What is the Backpropagation?

If we give the model a input data, the model will give you an output. Then, you can calculate the loss value between the output and the ground truth. If you know the quantity of loss, we can update the weights in the model by the calculated gradient by chain rule.