Problem 3

Let $p(y|x;\theta) = (1-\delta)(h_\theta(x))^y(1-h_\theta(x))^{1-y}$ since we have $p(error) = \delta$
Then the likelihood function is:
$L(\theta) = \Pi_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta) = \Pi_{i=1}^{m}(1-\delta)(h_\theta(x^{(i)}))^{y^{(i)}}(1-h_\theta(x^{(i)}))^{1-y^{(i)}}$
Thus the log-likelihood is $l(\theta) = \sum_{i=1}^{m}\log(1-\delta) + y^{(i)}\log h_\theta(x^{(i)}) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))$

Problem 4

The loss function is
$L = \frac{1}{2}\sum_i(y-\hat{y})^2 = \frac{1}{2}(80(1-\hat{y})^2 + 20(0-\hat{y})^2)$
So the derivative of L is $-80(1-\hat{y}) + 20\hat{y} = 100\hat{y} - 80$
Let it be 0, then $\hat{y} = 0.8$
Therefore, there is 0.8 probability to output 1.

Problem 5&6

In the implementation part, I followed the psuedocode in R&N textbook. According to professor's word in piazza, I normalize the data by substracting the column mean, and if column standard deviations is not zero, I divide the data by them. And then to initialize the weight, I used the recommended Xavier Initialization. And then the whole algorithm follows the textbook by first doing forward propagation, and then back propagation.
But my code does not have code accuracy, I am not sure