Problem 2:

a)

$$\nabla l(\theta) = \sum_{i=i}^{n} \{ \frac{exp(-\theta x^i) x^i}{1 + exp(-\theta x^i)} + x^i (y^i - 1) \}$$

Pseudo-Code

Algorithm 1 GradientDescent1

Initialize parameter θ^0
while $t <$ number of iteration **do**
    Calculate $\nabla\theta$ using (2)
    Update θ using $\theta^t = \theta^{t-1} + \eta\nabla\theta$
end while

b) Stochastic Gradient Descent

$$\nabla\theta = \sum_{i \in S_k} \{ \frac{exp(-\theta x^i) x^i}{1 + exp(-\theta x^i)} + x^i (y^i - 1) \}$$

Algorithm 3 StochasticGradientDescent

Initialize parameter θ^0 and the difference ϵ
while $\epsilon <$ threshold **do**
    Randomly sample 10% of the data without replacement
    Calculate $\nabla\theta$ using (3)
    Update θ using $\theta^t = \theta^{t-1} + \eta\nabla\theta$
    Calculate the old objective value using θ^{t-1} and new objective value using θ^t, and do a difference to update ϵ
end while

Hessian Matrix

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

Second Derivative of Log-likelihood function

-

$$\begin{aligned} \frac{\partial^2 l(\theta)}{\partial \theta_p \partial \theta_q} &= \sum_{i=1}^{n} - \frac{x_p^{(i)} x_q^{(i)} \exp(\theta X^{(i)})}{(1 + \exp(\theta X^{(i)}))^2} \\ &= \sum_{i=1}^{n} -x_p^{(i)} x_q^{(i)} S(-\theta X) S(\theta X) \\ &= -X X^T S(-\theta X) S(\theta X) \end{aligned}$$

Where

$$S = \frac{1}{1+\exp(-x)} = \frac{\exp(x)}{\exp(x)+1}$$

As S is the sigmoid function, S is always positive, as is XX^T. Therefore, the training problem is concave. Thus, it does not has local minima, and with sufficiently small training rate, the gradient descent will converges to global minima.