INT 3405: Machine Learning **First Semester 2023-2024**

Homework 3

- Feel free to talk to other students in the class when doing the homework. You should, however, write down your solution yourself. You also must indicate on each homework with whom you collaborated and cite any other sources you use including Internet sites.
- You will write your solution in LaTeX and submit the pdf file in zip files, including relevant materials, through courses.uet.vnu.edu.vn
- Deadline: 16/10/2023. Dont be late.

1 Learning XOR function - 10pts

Given a non-linear sepratable set, If our dataset is not linear separatable, for example:

	x^1	x^2	y
x_1	0	0	0
x_2	0	1	1
x_3	1	0	1
x_4	1	1	0

Table 1: Dataset for learning XOR function

If we fit the dataset with a logistic regression model, What is the results of the gradient descent training: Convergence or Non-Convergence?

1. Explain your choice
2. Could you verify your answer by some experiment results

2 Newton's method - 10pts

Newton's method optimization use the second derivative to calculate the optimization jump that comes from solving the equation

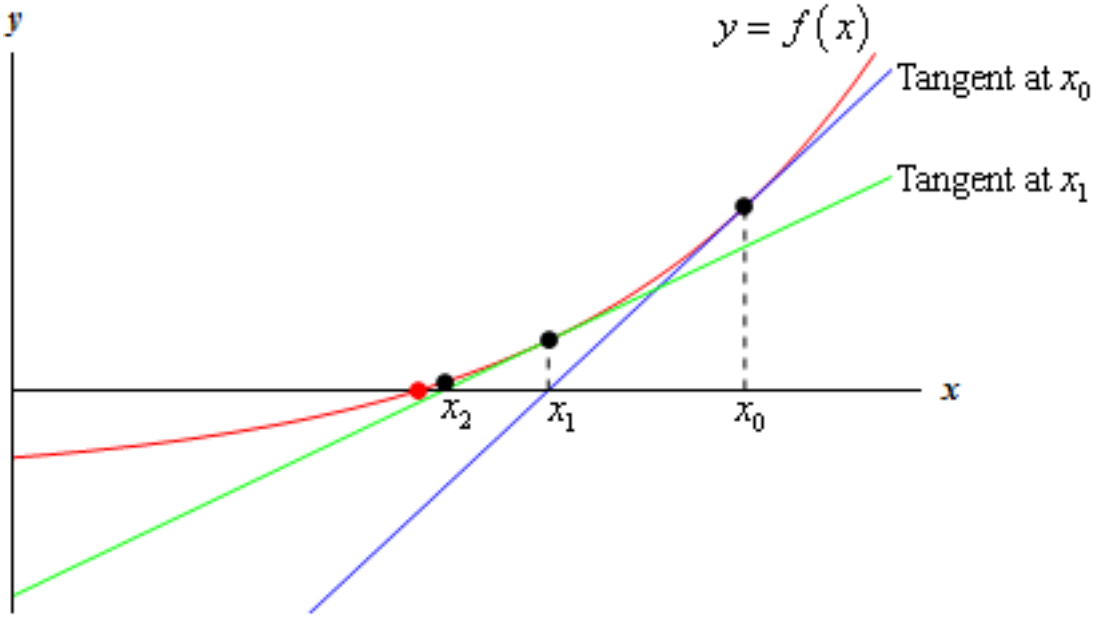
$$f(x) = 0$$

By first-order Taylor expansion of f around the starting point x_i:

$$f(x) \approx f(x_i) + f'(x_i)(x - x_i)$$

Set the right side to 0 and solve the next destination

$$x_{i+1} = x_i - f'(x_i)^{-1} f(x_i)$$



Apply Newton's method to solve equations of extremes (with zero derivatives)

$$f'(x) = 0$$

update as follows (replace f by $f^{'}$)

$$x_{i+1} = x_i - f''(x_i)^{-1} f'(x_i)$$

In case of $x \in \mathbb{R}^d$ (vector in d-dimension space) we use

$$x_{i+1} = x_i - H(x_i)^{-1} \nabla f(x_i)$$

where $\nabla f(x)$ is the derivative (gradient) and $H(x) = \left[\frac{\partial f^2}{\partial x_i \partial x_j}\right]$ is the Hessian matrix of f

1. Apply Newton's method to training the Logistic Regression Model (concretize updated formulas of parameter sets w, w_0.
2. Plot the learning path of Newton's method and the Gradient descent's method (like in slide) with the AND dataset and XOR dataset. You should test with several initial values of parameter sets w, w_0.

3 Multiclass Learning - 10pts

In the previous section, we use random variable Bernoulli for two-label classification. In order to describe the multi-label classification, we use the categorical distribution

Group distribution
Variable $X \sim Cat(x|\theta_1, \theta_2, \ldots, \theta_C)$ is the discrete random variable range from $1, 2, \ldots, C$ with probability

$$P(X = c) = \theta_c$$

with $\sum_{c=1}^{C} \theta_c$. Example: Balance toss have probability $P(X = c) = 1/6, c = 1, 2, \ldots, 6$
General formula

$$P(X = x) = \prod_{c=1}^{C} \theta_c^{\mathbb{I}(x=c)}$$

Exercise:

1. Calculate MLE of parameters $\theta_1, \theta_2, \ldots, \theta_C$ given by $D = \{x_1, x_2, \ldots, x_n\}$

2. **Multi-label Logistic Regression Model:** Use for multi-label classification: For each class c, we use a set of weights w_c, w_{0c}, calculate:

   $$f_c(x) = w_c^T x + w_{0c}$$

   then use softmax function to give the probability of each classifier (genelization of sigmoid function)

   $$P(Y = c|x) = \mu_c = \text{softmax}_c(f_1, f_2, \ldots, f_C) = \frac{e^{f_c}}{\sum_{k=1}^{C} e^{f_k}}$$

   Use gradient decent by derivative to train the model on the MLE principle given by data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ with $x_i \in \mathbb{R}^d, y_i \in \{1, 2, \ldots, C\}$.
   (**Hint**: Calculate the derivative of NLL for set of weights w_c, w_{0c} with $c = 1, 2, \ldots, C$