

# **HOMEWORK 3: ADVANCED REGRESSION, LOGISTICS REGRESSION, AND NEURAL NETWORK CPE 490/590 ST**

**Instructor: Rahul Bhadani**

**Due: March 07, 2025, 11:59 PM  
170 points**

You are allowed to use a generative model-based AI tool for your assignment. However, you must submit an accompanying reflection report detailing how you used the AI tool, the specific query you made, and how it improved your understanding of the subject. You are also required to submit screenshots of your conversation with any large language model (LLM) or equivalent conversational AI, clearly showing the prompts and your login avatar. Some conversational AIs provide a way to share a conversation link, and such a link is desirable for authenticity. Failure to do so may result in actions taken in compliance with the plagiarism policy.

Additionally, you must include your thoughts on how you would approach the assignment if such a tool were not available. Failure to provide a reflection report for every assignment where an AI tool is used may result in a penalty, and subsequent actions will be taken in line with the plagiarism policy.

## **Submission instruction:**

Submission instruction for this homework supersedes one mentioned in the Syllabus.

This homework requires all answers recorded in a single `.ipynb` Python notebook. You may upload hand-written PDF for the theory portion. You can use a combination of text cell (i.e. markdown formatted cell) and code cell to provide your answer. To add equations you should be able to use Latex syntax in the text cells of your Python notebook. As a part of your submission, you must provide executed notebook with code, text, and outputs. Alternatively, you can also provide a url (whose permission you must change to 'anyone with link can view') of your Python notebook from Google Colab. The naming convention for your notebook should follow the format `{firstname_lastname}_CPE 490/590 ST_hw03.ipynb`. For example, if your name is Sam Wells, and you are enrolled in CPE 490 your file name should be `sam_wells.CPE490_hw03.ipynb`.

Please refer to [https://github.com/rahulbhadani/CPE490\\_590\\_Sp2025/blob/master/Code/](https://github.com/rahulbhadani/CPE490_590_Sp2025/blob/master/Code/)

for hands-on.

## Theory

### 1 Neural Network by Hand (20 points)

Let's consider features of two training samples  $\mathbf{x} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}] = [(x_1^{(1)}, x_2^{(1)}), (x_1^{(2)}, x_2^{(2)})] = [(2, 4), (4, 5)]$  and the response variable as  $\mathbf{y} = [y^{(1)}, y^{(2)}] = [-1, 1]$ . Note that when we don't write superscript, i.e.  $x_1$  or  $x_2$ , we are simply considering features of a training sample without specifying which training sample.

Consider a classification problem for which we will develop a neural network model. The loss function to be considered is

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \mathcal{L} = -\frac{1}{n} \sum_{i=1}^n \left( y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right)$$

Note that this is similar to the problem demonstrated in the class except we will calculate the loss by taking both training samples.

Here,

$$\frac{\partial \mathcal{L}}{\partial \hat{y}^{(1)}} = -\frac{y^{(1)}}{\hat{y}^{(1)}} + \frac{1 - y^{(1)}}{1 - \hat{y}^{(1)}}$$

and

$$\frac{\partial \mathcal{L}}{\partial \hat{y}^{(2)}} = -\frac{y^{(2)}}{\hat{y}^{(2)}} + \frac{1 - y^{(2)}}{1 - \hat{y}^{(2)}}$$

Individual training sample  $i$  will lead to the gradient with respect to a weight  $W$  as:

$$\nabla_W \mathcal{L}^{(i)} = \frac{\partial \mathcal{L}}{\partial W^{(i)}} = \frac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}} \times \frac{\hat{y}^{(i)}}{\partial z^{(i)}} \times \frac{\partial z^{(i)}}{\partial W^{(i)}} \quad (1)$$

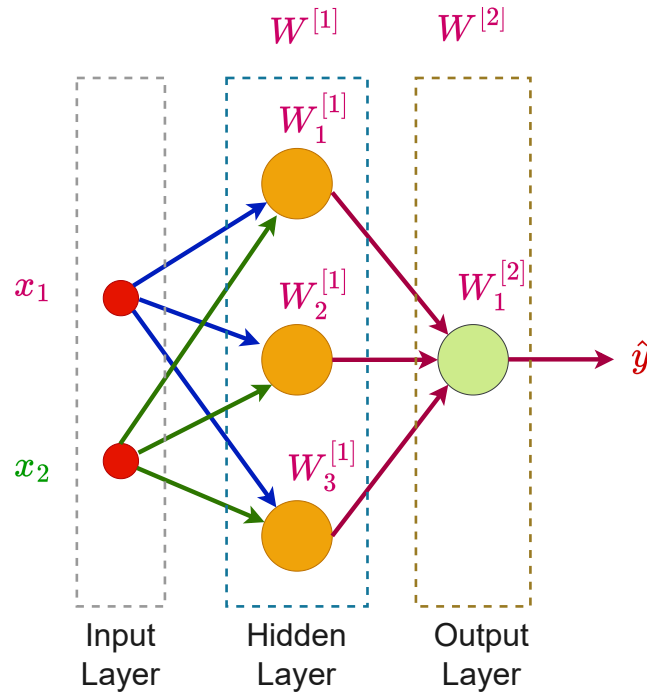


Figure 1: Homework 3: Advanced Regression, Logistics Regression, and Neural Network: Q1

In Equation (1), superscript ( $i$ ) denotes that gradient is calculated for individual samples. Then we calculate the average gradient as

$$\nabla_W \mathcal{L} = \frac{1}{n} \sum_i^n \frac{\partial \mathcal{L}}{\partial W^{(i)}} \quad (2)$$

so that the update rule for stochastic gradient descent becomes:

$$W = W - \eta \nabla_W \mathcal{L} \quad (3)$$

What it means is that you need to calculate the gradient for both training samples, take the average, and then update the weights.

Now, consider a neural network with one hidden layer containing three neurons so that the architecture looks as shown in Figure 1. In this problem, we won't consider bias, that is  $b = 0$  at all neurons of the given Neural Network.

Further, let's consider the following initial values of weights:

$$\begin{aligned}W_1^{[1]} &= [0.1, 0.3] \\W_2^{[1]} &= [0.2, 0.4] \\W_3^{[1]} &= [0.1, 0.5] \\W_1^{[2]} &= [0.1, 0.2, 0.1]\end{aligned}\tag{4}$$

Further, we will also consider the activation functions at all four neurons to be sigmoid, i.e.

$$a = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Note that the derivative of  $a$  with respect to  $z$  is given

$$\frac{\partial a}{\partial z} = \sigma(z) \left(1 - \sigma(z)\right)$$

You are required to complete the following task (only for one epoch) **(5 points each)**:

1. Perform the forward propagation on the first hidden layer and write down  $z_1^{[1]}, z_2^{[1]}, z_3^{[1]}, a_1^{[1]}, a_2^{[1]},$  and  $a_3^{[1]}$  for **both training samples**.
2. Perform the forward propagation on the output layer and write down  $z_1^{[2]}, a_1^{[2]}$  for **both training samples**.
3. Calculate the final average loss value taking into account both training samples.
4. Perform the backward propagation to write down the updated values of  $W_1^{[2]}$ . Note the vector  $W_1^{[2]}$  will contain three values as opposed to the two values you saw in the example done in the classroom. Don't forget to average the gradient using both training samples.

## 2 Dropout (5 points)

What is dropout in the context of a neural network and how is it beneficial in training a neural network?

### 3 Activation Functions (5 points)

What is the role of activation functions in neural networks, and how do they impact the network's ability to model complex functions?

### 4 Softmax Functions (5 points)

Explain the difference between a softmax function and a sigmoid function when used in the output layer of a fully connected neural network. In which scenarios would you choose one over the other?

### 5 Logistics Regression by Hand (20 Points)

The following dataset consists of 4 training examples, where  $x_k^{(i)}$  denotes the  $k$ -th dimension of the  $i$ -th training example  $\mathbf{x}^{(i)}$ , and  $y^{(i)}$  is the corresponding label ( $k \in \{1, 2, 3\}$  and  $i \in \{1, 2, 3, 4\}$ ).

$i$	$x_1$	$x_2$	$x_3$	$y$
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0

A binary logistic regression model is trained on this dataset, and the parameter vector  $\theta$  after training is

$$\theta = [1.5 \ 2 \ 1]^T.$$

*Note:* There is **no intercept term** used in this problem.

Use the data above to answer the following questions. For all numerical answers, please use one number rounded to the fourth decimal place; e.g., 0.1234. Showing your work in these questions is optional, but it is recommended to help us understand where any misconceptions may occur.

1. Calculate  $J(\theta)$ ,  $\frac{1}{N}$  times the negative log-likelihood over the given data and parameter  $\theta$ . (Note here we are using natural log, i.e., the base is  $e$ ).
2. Calculate the gradients  $\frac{\partial J(\theta)}{\partial \theta_j}$  with respect to  $\theta_j$  for all  $j \in \{1, 2, 3\}$ .
3. Update the parameters following the parameter update step  $\theta_j \leftarrow \theta_j - \eta \frac{\partial J(\theta)}{\partial \theta_j}$  and write the updated (numerical) value of the vector  $\theta$ . Use learning rate  $\eta = 1$ .

## 6 Locally-weighted Regression (10 Points)

Let the training set be denoted by

$\{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$ . Consider positive weights  $a^{(1)}, \dots, a^{(N)}$ . We define the weighted least squares problem as follows:

$$\hat{\mathbf{w}} = \arg \min \frac{1}{2} \sum_{i=1}^N a^{(i)} \left( y^{(i)} - \mathbf{w}^\top \mathbf{x}^{(i)} \right)^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2 \quad (5)$$

Show that the closed-form solution to the weighted least squares problem is given by the formula below.

$$\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{A} \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{A} \mathbf{y} \quad (6)$$

where  $\mathbf{X}$  is  $N \times D$  design matrix,  $\mathbf{A}$  is a diagonal matrix where  $\mathbf{A}_{ii} = a^{(i)}$ .

In this case, we define the loss function is defined to be the total loss (not the average loss) over all the training examples.

### Practice

## 7 Classification with Neural Network (25 Points)

For this task, you will be working with the diabetes dataset. Your goal is to implement a Neural Network model that can classify whether a given training sample indicates the presence of diabetes (Outcome = 1) or not (Outcome = 0).

1. **Data Standardization (2 points)** Standardize the dataset using the StandardScaler from the sklearn library.
2. **Neural Network Implementation (5 points)** Implement a Neural Network with the following specifications:
  - The network should have two hidden layers.
  - The first hidden layer should contain 4 neurons.
  - The second hidden layer should contain 3 neurons.
3. **Training the Model (5 points)** Train the neural network using the Adam optimizer with a learning rate of 0.01. The training should be carried out for 1000 epochs. Choose the appropriate loss function for the classification problem.
4. **Loss Plot (3 points)** After training the model, plot the loss against the number of epochs.
5. **Model Saving (5 points)** Save the trained model for future use or deployment in ONNX format.
6. **Model Prediction (5 points)** Load the trained model you saved in the previous step, predict the outcome of diabetes (either 0 or 1) for the following data:

Feature	Value
Pregnancies	7
Glucose	149
BloodPressure	73
SkinThickness	94
Insulin	94
BMI	32
DiabetesPedigreeFunction	0.672
Age	45

Table 1: Dataset

## 8 Logistics Regression to Predict Equipment Failure (35 Points)

In this problem, we will use Logistics Regression to predict equipment faults during the wafer fabrication process of the semiconductor industries. Dataset has been added to the course Github repo at [https://github.com/rahulbhadani/CPE490\\_590\\_Sp2025/tree/master/Data/secom](https://github.com/rahulbhadani/CPE490_590_Sp2025/tree/master/Data/secom)

First, read the description of the dataset from the original source : <https://archive.ics.uci.edu/dataset/179/secom>. The dataset contains features and labels in two separate files. Label -1 means failure and + 1 means success. Based on your observation, answer the following question or perform the given task:

1. For a given sample, some features may contain NaN (not a number). NaN represents a missing value. Remove all the features where more than 20% samples contain NaN. **(5 Points)**
2. For the remaining samples, replace the NaN with the mean of the given feature by taking mean from the entire sample that doesn't contain NaN. **(5 Points)**
3. What is the percentage of remaining sample attributed to failure? Create an appropriate visualization to convey your message. Is there class imbalance in the dataset? **(5 Points)**
4. Scale the dataset using StandardScaler. **(5 Points)**
5. Use Logistics Regression with maximum iteration of 250 to create a prediction model. One-third of the sample must be test dataset that you should divide randomly. **(5 Points)**
6. After training, compute the accuracy, F1-score, precision, and recall on the training and the test dataset. **(5 Points)**
7. Repeat the step 5 and 6 five times, and gather all sets of accuracy, F1-score, precision, and recall and plot a box plot showing the variation of these metrics. What conclusion do you make from the boxplot? **(5 Points)**

## 9 Multiple Regression with PyTorch (45 Points)

We will build a multiple linear regression model for predicting the net hourly electrical energy output (PE) in MegaWatt. Download the dataset from [https://github.com/rahulbhadani/CPE490\\_590\\_Sp2025/tree/master/Data/CombinedCyclePowerPlant](https://github.com/rahulbhadani/CPE490_590_Sp2025/tree/master/Data/CombinedCyclePowerPlant). The dataset contains following features:

- $x_1$ : Ambient Temperature (AT) in  $^{\circ}\text{C}$ ,
- $x_2$ : Exhaust Vacuum (V) in cm Hg  $x_2$
- $x_3$ : Ambient Pressure (AP) in milibar  $x_3$
- $x_4$ : Relative Humidity (RH) in percent  $x_4$

Perform the following task and answer the following questions:

1. Create a scatterplot matrix figure that shows correlation among features as well as features and the target variable. **(5 Points)**



2. We are interested in creating a multiple linear regression model

$$y = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$$

Perform multiple linear regression using above model using Statsmodel package. However, for this purpose use the Cauchy Loss function:

$$\ell(\hat{y}, y) = \frac{c^2}{2} \log \left( 1 + \left( \frac{y - \hat{y}}{c} \right)^2 \right) \quad (7)$$

where  $c$  is the Cauchy loss function constant. In this question, we will use  $c = 1$ .

**(10 Points)**

3. What are the coefficient values for the fitted model? **(5 Points)**
4. What's the final loss value? **(5 Points)**
5. What does the 95% confidence interval for the coefficients tell you? **(5 Points)**
6. Create the residual plot  $\hat{y} - y$  against  $x_1, x_2, x_3$ , and  $x_4$ . What do you conclude? **(5 Points)**
7. Provide the alternative implementation using PyTorch. Use appropriate values for learning rate and number of epochs. **(10 Points)**