Machine Learning

Homework 4

Karan Sunil Kumbhar
Id. - 12140860
BTech CSE
2025

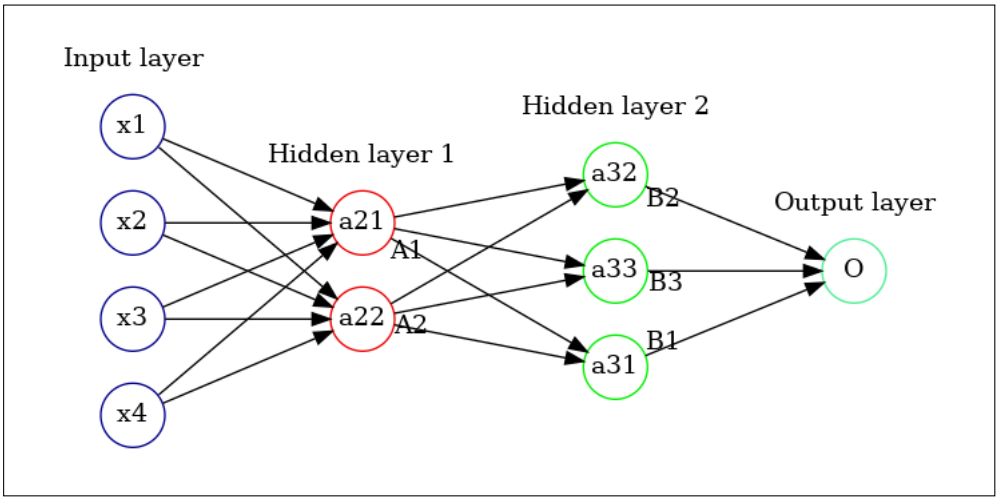
CS550 Machine Learning

October 1, 2023

Problem 1
Ch10_Q1

Solution.

(a) Neural network described with p = 4 input units, 2 units in the first hidden layer, 3 units in the second hidden layer, and a single output



(b) Assinging weights and biases to each nueron whcih is shown in below figure 1

Let's break down the calculations step by step. We'll assume ReLU activation functions and denote weights as w_i and biases as b_i for simplicity.

Step 1: Calculating the inputs for the first hidden layer:

$$\text{Input to } a_{21} = (x_1 \cdot w_{11}) + (x_2 \cdot w_{12}) + (x_3 \cdot w_{13}) + (x_4 \cdot w_{14}) + b_{21}$$
$$\text{Input to } a_{22} = (x_1 \cdot w_{21}) + (x_2 \cdot w_{22}) + (x_3 \cdot w_{23}) + (x_4 \cdot w_{24}) + b_{22}$$

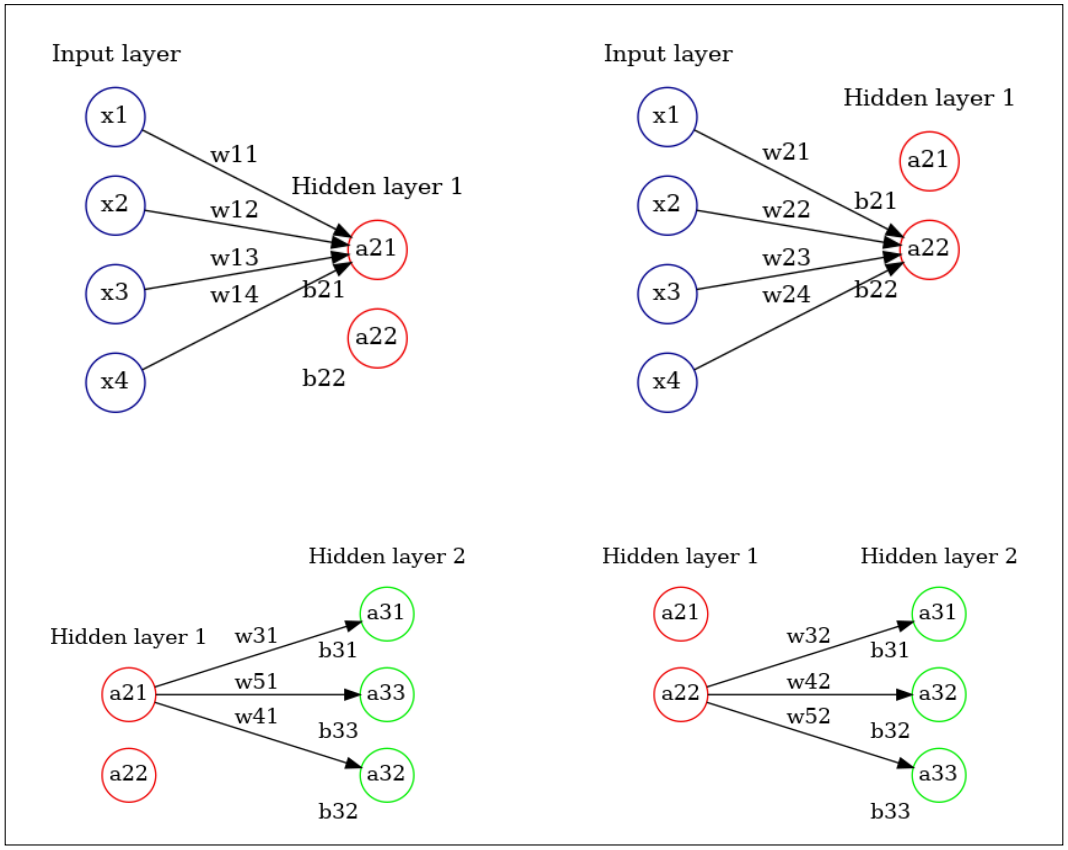
Step 2: Applying the ReLU activation function to the inputs:

$$A_1 = \text{ReLU}(\text{Input to } a_{21})$$
$$A_2 = \text{ReLU}(\text{Input to } a_{22})$$

Step 3: Calculating the inputs for the second hidden layer using the activations from the first hidden layer:

$$\text{Input to } a_{31} = (A_1 \cdot w_{31}) + (A_2 \cdot w_{32}) + b_{31}$$
$$\text{Input to } a_{32} = (A_1 \cdot w_{41}) + (A_2 \cdot w_{42}) + b_{32}$$
$$\text{Input to } a_{33} = (A_1 \cdot w_{51}) + (A_2 \cdot w_{52}) + b_{33}$$

Figure 1: Assinging weights and biases



Step 4: Applying the ReLU activation function to the inputs:

$$B_1 = \text{ReLU}(\text{Input to } a_{31})$$
$$B_2 = \text{ReLU}(\text{Input to } a_{32})$$
$$B_3 = \text{ReLU}(\text{Input to } a_{33})$$

Now, we have the outputs of the second hidden layer as B_1, B_2, and B_3 in terms of x_1, x_2, x_3, x_4, weights (w), and biases (b).

Step 5: Calculating the input to the output layer using the activations from the second hidden layer:

$$\text{Input to } O = (B_1 \cdot w_{61}) + (B_2 \cdot w_{62}) + (B_3 \cdot w_{63}) + b_4$$

Step 6: Applying the ReLU activation function to obtain the final output O:

$$f(x) = \text{ReLU}(\text{Input to } O)$$

So, the final output $f(x)$ is a function of the weights (w_i),(x_i) and biases (b_i) for all layers, as well as the ReLU activation functions applied to the intermediate inputs.

(c) Let's put in some values for the coefficients and calculating the value of $f(X)$. We'll assume the following coefficients:

For the first hidden layer:

$$\begin{aligned} &w_{11} = 0.5, \quad w_{12} = -0.3, \quad w_{13} = 0.2, \quad w_{14} = 0.1, \\ &w_{21} = -0.1, \quad w_{22} = 0.2, \quad w_{23} = 0.4, \quad w_{24} = -0.2, \\ &b_{21} = 0.2, \quad b_{22} = 0.1. \end{aligned}$$

For the second hidden layer:

$$\begin{aligned} &w_{31} = 0.3, \quad w_{32} = -0.1, \quad b_{31} = 0.1, \\ &w_{41} = -0.2, \quad w_{42} = 0.3, \quad b_{32} = -0.1, \\ &w_{51} = 0.2, \quad w_{52} = 0.1, \quad b_{33} = 0.2. \end{aligned}$$

For the output layer:

$$w_{61} = -0.3, \quad w_{62} = 0.2, \quad w_{63} = -0.1, \quad b_4 = 0.3.$$

Now, let's calculate the value of $f(X)$ by following the steps mentioned in part (b) with these coefficient values. We'll substitute the given coefficients into the equations for each layer to obtain the final output $f(X)$.

$$\begin{aligned} \text{Input to } a_{21} &= (x_1 \cdot 0.5) + (x_2 \cdot (-0.3)) + (x_3 \cdot 0.2) + (x_4 \cdot 0.1) + 0.2 \\ &= 0.5x_1 - 0.3x_2 + 0.2x_3 + 0.1x_4 + 0.2 \\ &= 0.5 \cdot (1) - 0.3 \cdot (1) + 0.2(1) + 0.1 \cdot (1) + 0.2 \\ &= 0.7 \\ \text{Input to } a_{22} &= (x_1 \cdot (-0.1)) + (x_2 \cdot 0.2) + (x_3 \cdot 0.4) + (x_4 \cdot (-0.2)) + 0.1 \\ &= -0.1x_1 + 0.2x_2 + 0.4x_3 - 0.2x_4 + 0.1 \\ &= -0.1 \cdot (1) + 0.2 \cdot (1) + 0.4 \cdot (1) - 0.2 \cdot (1) + 0.1 \\ &= 0.4 \end{aligned}$$

Now, Applying the ReLU activation function to the inputs:

$$\begin{aligned} A_1 &= \max(0, 0.7) \\ &= 0.7 \\ A_2 &= \max(0, 0.4) \\ &= 0.4 \end{aligned}$$

For the second hidden layer :

$$\begin{aligned} \text{Input to } a_{31} &= (A_1 \cdot 0.3) + (A_2 \cdot (-0.1)) + 0.1 \\ &= 0.3A_1 - 0.1A_2 + 0.1 \\ &= 0.3 \cdot (0.7) - 0.1 \cdot (0.4) + 0.1 \\ &= 0.21 - 0.04 + 0.1 \\ &= 0.27 \\ \text{Input to } a_{32} &= (A_1 \cdot (-0.2)) + (A_2 \cdot 0.3) - 0.1 \\ &= -0.2A_1 + 0.3A_2 - 0.1 \\ &= -0.2 \cdot (0.7) + 0.3 \cdot (0.4) - 0.1 \\ &= -0.14 + 0.12 - 0.1 \\ &= -0.12 \\ \text{Input to } a_{33} &= (A_1 \cdot 0.2) + (A_2 \cdot 0.1) + 0.2 \\ &= 0.2A_1 + 0.1A_2 + 0.2 \\ &= 0.2 \cdot (0.7) + 0.1 \cdot (0.4) + 0.2 \\ &= 0.14 + 0.04 + 0.2 \\ &= 0.38 \end{aligned}$$

Applying the ReLU activation function to the inputs:

$$\begin{aligned} B_1 &= \max(0, 0.27) \quad &= 0.27 \\ B_2 &= \max(0, -0.12) \quad &= 0 \\ B_3 &= \max(0, 0.38) \quad &= 0.38 \end{aligned}$$

Now, for the output layer, use the activations from the second hidden layer:

$$\begin{aligned} \text{Input to } O &= (B_1 \cdot (-0.3)) + (B_2 \cdot 0.2) + (B_3 \cdot (-0.1)) + 0.3 \\ &= -0.3B_1 + 0.2B_2 - 0.1B_3 + 0.3 \\ &= -0.3 \cdot (0.27) + 0.2 \cdot (0) - 0.1 \cdot (0.38) + 0.3 \\ &= 0.181 \end{aligned}$$

Finally, apply the ReLU activation function to obtain the final output O:

$$f(x) = \max(0, 0.181)$$
$$f(x) = 0.181$$

(d) Number of Parameters:

To calculate the number of parameters in the neural network, we need to count the weights and biases. Here's the breakdown:

- **First Hidden Layer:**
  - 2 neurons in the first hidden layer.
  - Each neuron has 4 weights ($w_{11}, w_{12}, w_{13}, w_{14}$) and 1 bias ($b_{21}$).
  - Total parameters for the first hidden layer: $2 \times (4 \text{ weights} + 1 \text{ bias}) = 10$ parameters.
- **Second Hidden Layer:**
  - 3 neurons in the second hidden layer.
  - Each neuron has 2 weights (w_{31}, w_{32}) and 1 bias (b_{31}).
  - Total parameters for the second hidden layer: $3 \times (2 \text{ weights} + 1 \text{ bias}) = 9$ parameters.
- **Output Layer:**
  - 1 neuron in the output layer.
  - This neuron has 3 weights (w_{61}, w_{62}, w_{63}) and 1 bias (b_4).
  - Total parameters for the output layer: $1 \times (3 \text{ weights} + 1 \text{ bias}) = 4$ parameters.

Now, sum up the parameters from all layers:

$$\begin{aligned}\text{Total Parameters} &= \text{Parameters in the first hidden layer} \\ &\quad + \text{Parameters in the second hidden layer} \\ &\quad + \text{Parameters in the output layer} \\ &= 10 + 9 + 4 \\ &= 23 \text{ parameters}\end{aligned}$$

So, there are a total of 23 parameters in this neural network.

Problem 2
Ch10_Q2

Solution.

Equation 4.13

$$\log\left(Pr(Y=k|X=x)\right) = \frac{e^{\beta_{k0}+\beta_{k1}x_1+\ldots+\beta_{kp}x_p}}{\sum_{l=1}^{K} e^{\beta_{l0}+\beta_{l1}x_1+\ldots+\beta_{lp}x_p}}$$

Equation 10.13

$$f_m(X) = Pr(Y=m|X) = \frac{e^{Z_m}}{\sum_{l=0}^{9} e^{Z_l}}$$

(a) In equation (10.13):

$$f_m(X) = Pr(Y=m|X) = \frac{e^{Z_m}}{\sum_{l=0}^{9} e^{Z_l}}$$

If we add a constant c to each of the z_l values, the probability remains unchanged.
Proof :

Let $Z'_l = Z_l + c$ for $l = 0, 1, \ldots, 9$, where c is a constant.

Now, we'll compute the new probability $f'_m(X)$ with Z'_l:

$$f'_m(X) = Pr(Y=m|X) = \frac{e^{Z'_m}}{\sum_{l=0}^{9} e^{Z'_l}}$$

Substitute $Z'_l = Z_l + c$ into the equation:

$$f'_m(X) = \frac{e^{Z_m+c}}{\sum_{l=0}^{9} e^{Z_l+c}}$$

Now, we can factor out e^c from the numerator and denominator:

$$f'_m(X) = \frac{e^c \cdot e^{Z_m}}{e^c \cdot \sum_{l=0}^{9} e^{Z_l}}$$

we can remove e^c from the numerator and denominator:

$$f'_m(X) = \frac{e^{Z_m}}{\sum_{l=0}^{9} e^{Z_l}}$$

This is exactly the same as the original probability $f_m(X)$. Therefore, adding a constant c to each of the z_l values in equation (10.13) does not change the probability.

(b) Starting with Equation (4.13) for class k:

$$\log\left(Pr(Y=k|X=x)\right) = \frac{e^{\beta_{k0}+\beta_{k1}x_1+\ldots+\beta_{kp}x_p}}{\sum_{l=1}^{K} e^{\beta_{l0}+\beta_{l1}x_1+\ldots+\beta_{lp}x_p}}$$

Add constants c_j to coefficients for each class and feature:

$$\log\left(Pr(Y=k|X=x)\right) = \frac{e^{(\beta_{k0}+c_0)+(\beta_{k1}+c_1)x_1+\ldots+(\beta_{kp}+c_p)x_p}}{\sum_{l=1}^{K} e^{(\beta_{l0}+c_0)+(\beta_{l1}+c_1)x_1+\ldots+(\beta_{lp}+c_p)x_p}}$$

Now, simplify the terms in the numerator and denominator:

Numerator:

$$e^{(\beta_{k0}+c_0)} \cdot e^{(\beta_{k1}+c_1)x_1} \cdot \ldots = e^{(c_0)} \cdot e^{(c_1)x_1} \cdot \ldots \times e^{\beta_{k0}} \cdot e^{\beta_{k1}\times x_1} \cdot \ldots$$

Denominator:

$$\sum_{l=1}^{K} e^{(\beta_{l0}+c_0)} \cdot e^{(\beta_{l1}+c_1)x_1} \cdot \ldots = \sum_{l=1}^{K} e^{c_0} \cdot e^{c_1 \cdot x_1} \cdot \ldots \times e^{(\beta_{l0})} \cdot e^{(\beta_{l1})x_1} \cdot \ldots$$

The added constants c_j cancel out in both the numerator and denominator:

$$\begin{aligned}
\log\left(Pr(Y=k|X=x)\right) &= \frac{e^{(\beta_{k0}+c_0)} \cdot e^{(\beta_{k1}+c_1)x_1} \cdot \ldots \cdot e^{(\beta_{kp}+c_p)x_p}}{\sum_{l=1}^{K} e^{(\beta_{l0}+c_0)} \cdot e^{(\beta_{l1}+c_1)x_1} \cdot \ldots \cdot e^{(\beta_{lp}+c_p)x_p}} \\
&= \frac{e^{(c_0)} \cdot e^{(c_1)x_1} \cdot \ldots \times e^{\beta_{k0}} \cdot e^{\beta_{k1}\times x_1} \cdot \ldots}{\sum_{l=1}^{K} e^{c_0} \cdot e^{c_1 \cdot x_1} \cdot \ldots \times e^{(\beta_{l0})} \cdot e^{(\beta_{l1})x_1} \cdot \ldots} \\
&= \frac{(e^{(c_0)} \cdot e^{(c_1)x_1} \cdot \ldots) \times (e^{\beta_{k0}} \cdot e^{\beta_{k1}\times x_1} \cdot \ldots)}{(e^{c_0} \cdot e^{c_1 \cdot x_1} \cdot \ldots) \times (\sum_{l=1}^{K} \times e^{(\beta_{l0})} \cdot e^{(\beta_{l1})x_1} \cdot \ldots)} \\
&= \frac{e^{\beta_{k0}+\beta_{k1}x_1+\ldots+\beta_{kp}x_p}}{\sum_{l=1}^{K} e^{\beta_{l0}+\beta_{l1}x_1+\ldots+\beta_{lp}x_p}} \\
&= \log\left(Pr(Y=k|X=x)\right)
\end{aligned}$$

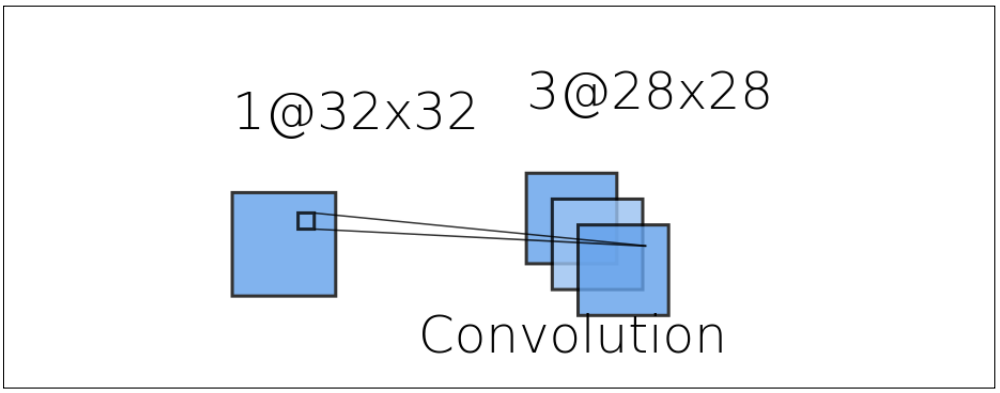
This shows that adding constants c_j to coefficients does not change the predictions at any new point x, and the probabilities remain the same.

Problem 3
Ch10_Q4

Solution.

(a) CNN having 32 × 32 grayscale images and has a single convolution layer with three 5 × 5 convolution filters.

When we apply three 5 × 5 filter to 32 × 32 grayscale image(single channel), We will get three 28 × 28 image.



(b) In given CNN,

- 32x32 grayscale input images
- Single convolutional layer
- Three 5x5 convolution filters

To calculate the number of parameters, we have to consider the following:

(a) **Parameters in Each Filter**: Each 5x5 filter has 5x5 = 25 weight parameters and 1 bias parameter. So, each filter contributes 25 (weights) + 1 (bias) = 26 parameters.

(b) **Number of Filters**: three filters in convolutional layer.

Now, calculationg the total number of parameters:

$$\begin{aligned}\text{Total Parameters} &= (\text{Parameters in Each Filter}) \times (\text{Number of Filters}) \\ &= 26 \times 3 \\ &= 78\end{aligned}$$

So, there are a total of 78 parameters in this CNN model. These parameters include the weights and biases of the three convolution filters.

(c) To convert given CNN mondel to ordinary feed-forward neural network we need to keep constraints on weights

Feed-forward nueral network will contain 32×32 nodes in input layer and and first hidden layer will conatin $28 \times 28 \times 3$ nodes

weights constraints :

- Pixel in hidden layer image of CNN model(1 pixel) will contain corresponding 5×5 pixel matrix(25 pixel) in input layer image in CNN model.

  Now in feed-forward neural network we have corresponding nodes for pixel in hidden layer(1 node) and input layer(25).

  So for node in hidden layer which is joined to corresponding input layer node other than mentioned 25 nodes having weights as zero

- So above mentioned 25 pixel(25 nodes) from input layer which is joined to hidden layer node.

  let's consider nodes as $n_1, n_2, n_3, \ldots, n_{25}$. One node in hidden layer will have corresponding 25 nodes in input layer for every node in hidden layer

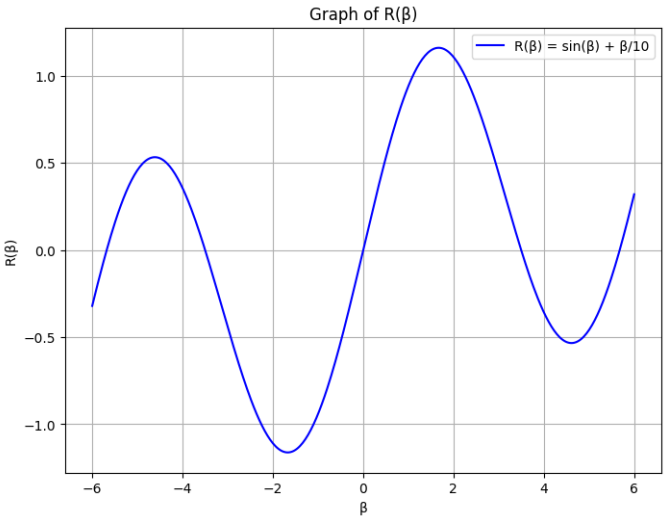
  So weight of n_k of node corresponding node j in hidden layer will have same weight as node n_k of corresponding node i in hidden layer from same channel in hidden layer image

(d) If there were no constraints, then $32 \times 32 \times 28 \times 28 \times 3$ weights would there be in the ordinary feed-forward neural network.

Problem 4
Ch10_Q6

Solution.

(a) Graph of the function $R(\beta) = sin(\beta) + \frac{\beta}{10}$ over the range $\beta\epsilon[6, 6]$.



(b) The derivative of the function $R(\beta) = \sin(\beta) + \frac{\beta}{10}$ with respect to β

- The derivative of $\sin(\beta)$ with respect to β is $\cos(\beta)$.
- The derivative of $\frac{\beta}{10}$ with respect to β is $\frac{1}{10}$.

Now, combining these derivatives, we get the derivative of $R(\beta)$:

$$R'(\beta) = \cos(\beta) + \frac{1}{10}$$

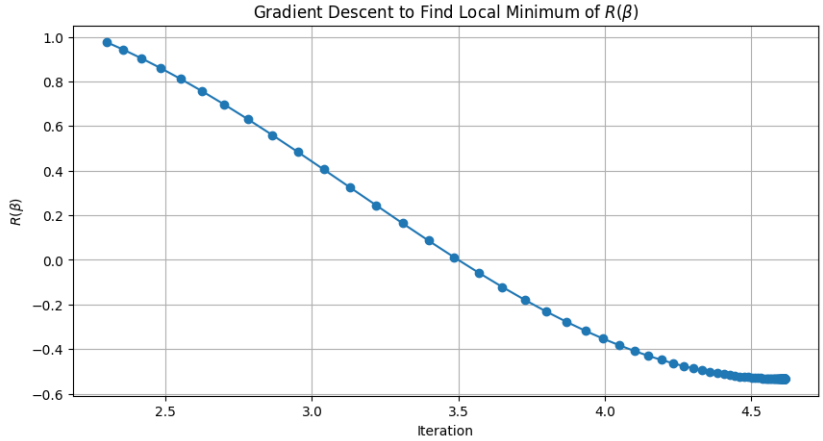
So, the derivative of the function $R(\beta) = \sin(\beta) + \frac{\beta}{10}$ is $R'(\beta) = \cos(\beta) + \frac{1}{10}$.

(c) Running a gradient descent to find a local minimum of $R(\beta) = \sin(\beta) + \frac{\beta}{10}$ with $\beta_0 = 2.3$ and a learning rate $\rho = 0.1$ involves iteratively updating β using the following formula:

$$\beta_{i+1} = \beta_i - \rho \cdot \frac{dR}{d\beta}$$

where $\frac{dR}{d\beta}$ is the derivative of $R(\beta)$.

We'll start with $\beta_0 = 2.3$ and update it iteratively until convergence.



Final value of β : 4.612220565617592
Final value of $R(\beta)$: -0.5337652811838157

So the local minima $R(\beta) = -0.53$ occurs at $\beta = 4.61$ approximately

(d) for $B^o = 1.4$

Final value of β : -1.6709610375631647
Final value of $R(\beta)$: -1.162083811898611
So the local minima $R(\beta) = -1.162$ occurs at $\beta = -1.67$ approximately