

2.

In an automatic air conditioner controller, there is a sensor to sense room temperature. If temperature increases then the controller should increase the speed of the rotor of the compressor. If the speed of the compressor increases then it increases the speed of the fan. The above controller is designed with a fuzzy logic controller with the following three fuzzy sets.

T: Temp is High

$$\left\{ \frac{0.3}{20}, \frac{0.7}{30}, \frac{0.8}{40}, \frac{1.0}{50} \right\}$$

R: Rotor is High

$$\left\{ \frac{0.2}{20}, \frac{0.4}{40}, \frac{0.6}{60}, \frac{0.8}{80}, \frac{0.9}{100}, \frac{1.0}{120} \right\}$$

S: Fan speed is High

$$\left\{ \frac{0.33}{500}, \frac{0.67}{1000}, \frac{0.95}{1500}, \frac{1.0}{2000} \right\}$$

a) Given the above, compute the following rules

R1: If temperature is High then rotation is High

R2: If rotation is High then fan speed is High

b) Compute $R1 \circ R2$, the composition of the relation R1 and R2.

c) Give the crisp value of the fuzzy value of fan speed when temperature is 40.

$$[5 \times 1 + 2.5 + 2.5 = 10]$$

Detail solution:

a) T: Temp is High

$$\left\{ \frac{0.3}{20}, \frac{0.7}{30}, \frac{0.8}{40}, \frac{1.0}{50} \right\}$$

R: Rotor is High

$$\left\{ \frac{0.2}{20}, \frac{0.4}{40}, \frac{0.6}{60}, \frac{0.8}{80}, \frac{0.9}{100}, \frac{1.0}{120} \right\}$$

R1: If temperature is High then rotation is High

	20	40	60	80	100	120
20	0.2	0.3	0.3	0.3	0.3	0.3
30	0.2	0.4	0.6	0.7	0.7	0.7
40	0.2	0.4	0.6	0.8	0.8	0.8
50	0.2	0.4	0.6	0.8	0.9	1.0

R: Rotor is High

$$\left\{ \frac{0.2}{20}, \frac{0.4}{40}, \frac{0.6}{60}, \frac{0.8}{80}, \frac{0.9}{100}, \frac{1.0}{120} \right\}$$

S: Fan speed is High

$$\left\{ \frac{0.33}{500}, \frac{0.67}{1000}, \frac{0.95}{1500}, \frac{1.0}{2000} \right\}$$

R2: If rotation is High then fan speed is High

	500	1000	1500	2000
20	0.2	0.2	0.2	0.2
40	0.33	0.4	0.4	0.4
60	0.33	0.6	0.6	0.6
80	0.33	0.67	0.8	0.8
100	0.33	0.67	0.9	0.9
120	0.33	0.67	0.95	1.0

b) Compute $R1 \circ R2$, the composition of the relation R1 and R2.

R1

R2

	500	1000	1500	2000
20	0.2	0.2	0.2	0.2
40	0.33	0.4	0.4	0.4
60	0.33	0.6	0.6	0.6
80	0.33	0.67	0.8	0.8
100	0.33	0.67	0.9	0.9
120	0.33	0.67	0.95	1.0

	20	40	60	80	100	120
20	0.2	0.3	0.3	0.3	0.3	0.3
30	0.2	0.4	0.6	0.7	0.7	0.7
40	0.2	0.4	0.6	0.8	0.8	0.8
50	0.2	0.4	0.6	0.8	0.9	1.0

Max-min composition is applied here.

$$\begin{aligned} \mu_T(20, 500) &= \max\{\min[\mu_R(20, 20), \mu_S(20, 500)], \min[\mu_R(20, 40), \mu_S(40, 500)], \min[\mu_R(20, 60), \\ &\mu_S(60, 500)], \min[\mu_R(20, 80), \mu_S(80, 500)], \min[\mu_R(20, 100), \mu_S(100, 500)], \min[\mu_R(20, 120), \\ &\mu_S(120, 500)]\} \\ &= \max[0.2, 0.3, 0.3, 0.3, 0.3, 0.3] = 0.3 \end{aligned}$$

$$\begin{aligned} \mu_T(20, 1000) &= \max\{\min[\mu_R(20, 20), \mu_S(20, 1000)], \min[\mu_R(20, 40), \mu_S(40, 1000)], \min[\mu_R(20, 60), \\ &\mu_S(60, 1000)], \min[\mu_R(20, 80), \mu_S(80, 1000)], \min[\mu_R(20, 100), \mu_S(100, 1000)], \min[\mu_R(20, 120), \\ &\mu_S(120, 1000)]\} \\ &= \max[0.2, 0.3, 0.3, 0.3, 0.3, 0.3] = 0.3 \end{aligned}$$

	500	1000	1500	2000
--	-----	------	------	------

20	0.3	0.3	0.3	0.3
30	0.33	0.67	0.7	0.7
40	0.33	0.67	0.8	0.8
50	0.33	0.67	0.95	1.0

Same way calculating other values we get

$R1 \circ R2$

c) Give the crisp value of the fuzzy value of fan speed when temperature is 40.

Defuzzified or crisp value of fan speed is

$$\begin{aligned}
 C &= \frac{\sum_{i=1}^n x_i \mu(x_i)}{\sum_{i=1}^n \mu(x_i)} \\
 &= \frac{(500 \cdot 0.33 + 1000 \cdot 0.67 + 1500 \cdot 0.8 + 2000 \cdot 0.8)}{0.33 + 0.67 + 0.8 + 0.8} \\
 &= 1398
 \end{aligned}$$

3. A smart washing machine is to be designed which would decide spin (the speed of the rotor in rotation per minute) and time (in minutes) of washing, and detergent (the amount in cc) to be dispensed. The machine will decide the above based on the following three input (will be decided with appropriate sensors attached within the machine)

- Weight in the range 0.5kg to 6.0 kg
- Cloth type: There are three types of clothes like cotton, woollen and synthetic and each type is measurable in 1, 2, ..., 10 scale.
- Pressure: water pressure in the range 1...6 pascal.

Given the above, answer the following questions.

- What are the linguistic hedges you should consider to fuzzily define each input and output. Draw the membership function for each fuzzy input and output set.

- b) How a fuzzy rule would look like. Give an example and illustrate your answer.
- c) How many total rules will be there in the rule base.
- d) Using the Takagi and Sugeno approach, how can a rule be inferred? And how can an output be computed?

[5+2.5+2.5+5 = 15]

Detailed solution:

a) Linguistic hedges are words or phrases that modify the meaning of fuzzy sets, such as very, more or less, slightly, etc. They can be used to adjust the shape of the membership functions and make them more flexible. For example, if we have a fuzzy set for weight called light, we can use the linguistic hedge very to create a new fuzzy set called very light, which has a narrower and higher membership function than light.

To fuzzily define each input and output, we need to choose appropriate fuzzy sets and membership functions for them. For example, we can use the following fuzzy sets and membership functions for the input weight:

- Very light: triangular membership function with parameters (0.5, 0.5, 1.5)
- Light: triangular membership function with parameters (0.5, 1.5, 2.5)
- Medium: triangular membership function with parameters (1.5, 2.5, 3.5)
- Heavy: triangular membership function with parameters (2.5, 3.5, 4.5)
- Very heavy: triangular membership function with parameters (3.5, 4.5, 6.0)

Similarly, we can use the following fuzzy sets and membership functions for the input cloth type:

- Cotton: trapezoidal membership function with parameters (1, 1, 3, 4)
- Woolen: trapezoidal membership function with parameters (3, 4, 6, 7)
- Synthetic: trapezoidal membership function with parameters (6, 7, 9, 10)

And for the input pressure:

- Low: triangular membership function with parameters (1, 1, 2.5)
- Medium: triangular membership function with parameters (1, 2.5, 4)
- High: triangular membership function with parameters (2.5, 4, 6)

For the output spin, we can use the following fuzzy sets and membership functions:

- Slow: triangular membership function with parameters (0, 0, 500)
- Medium: triangular membership function with parameters (0, 500, 1000)
- Fast: triangular membership function with parameters (500, 1000, 1500)

For the output time, we can use the following fuzzy sets and membership functions:

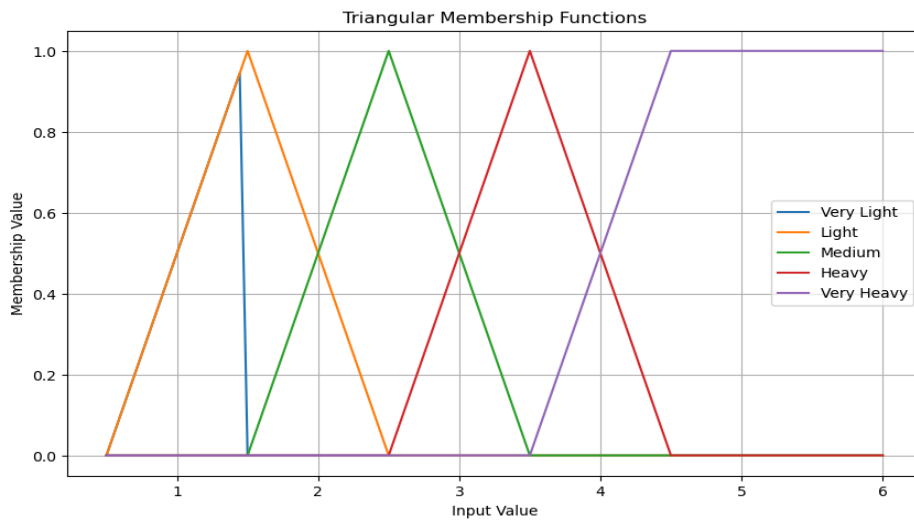
- Short: triangular membership function with parameters (0, 0, 15)

- Medium: triangular membership function with parameters (0, 15, 30)
- Long: triangular membership function with parameters (15, 30, 45)

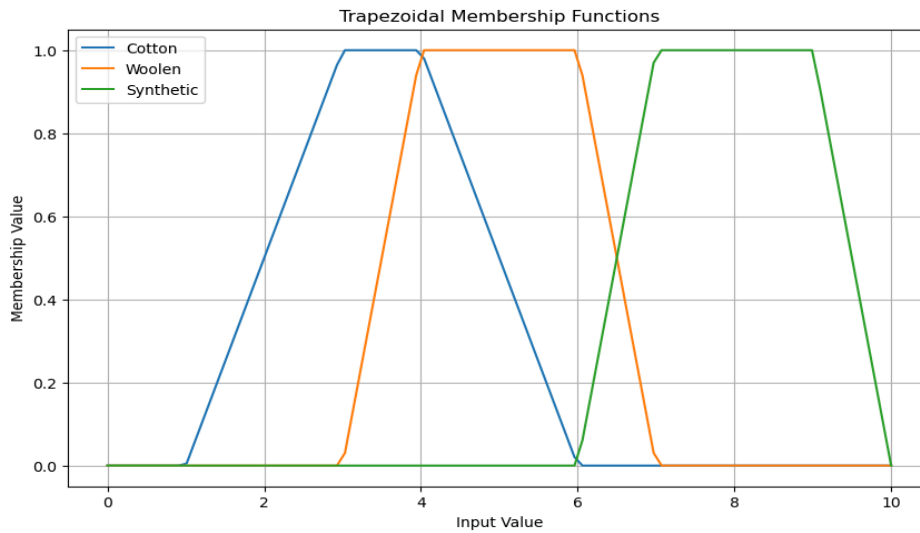
For the output detergent, we can use the following fuzzy sets and membership functions:

- Low: triangular membership function with parameters (0, 0, 50)
- Medium: triangular membership function with parameters (0, 50, 100)
- High: triangular membership function with parameters (50, 100, 150)

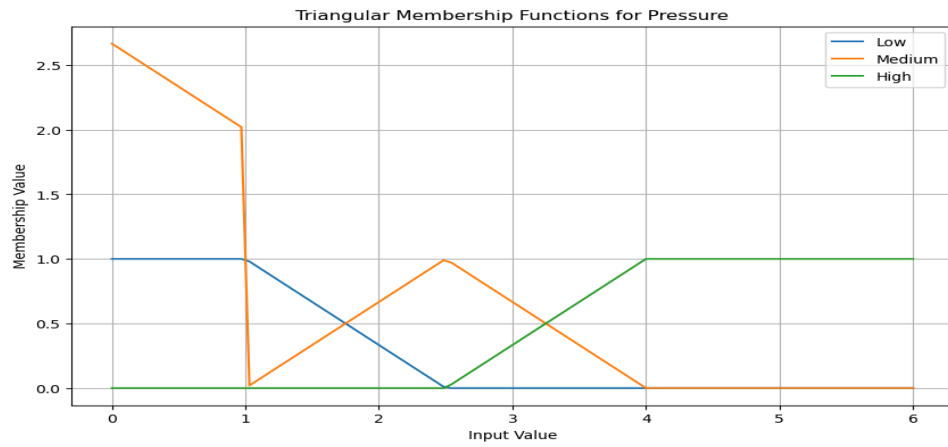
Weight:



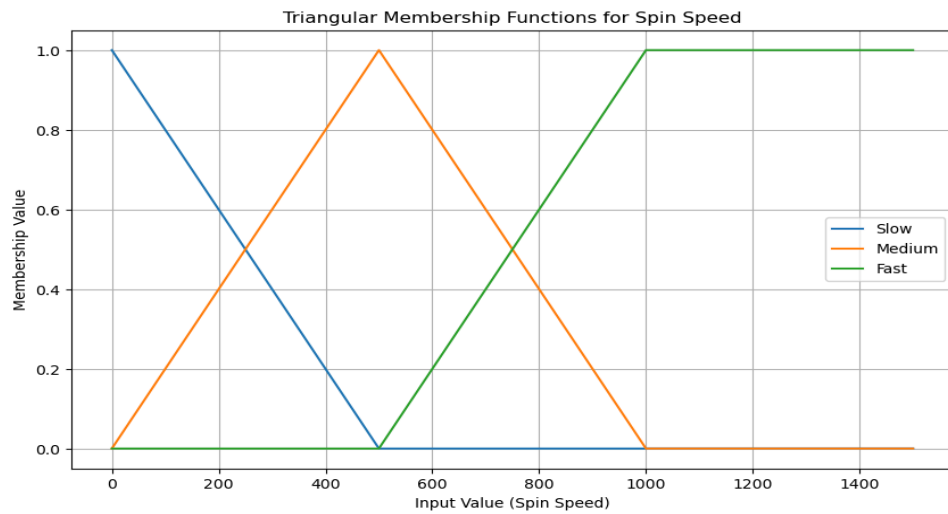
Cloth type:



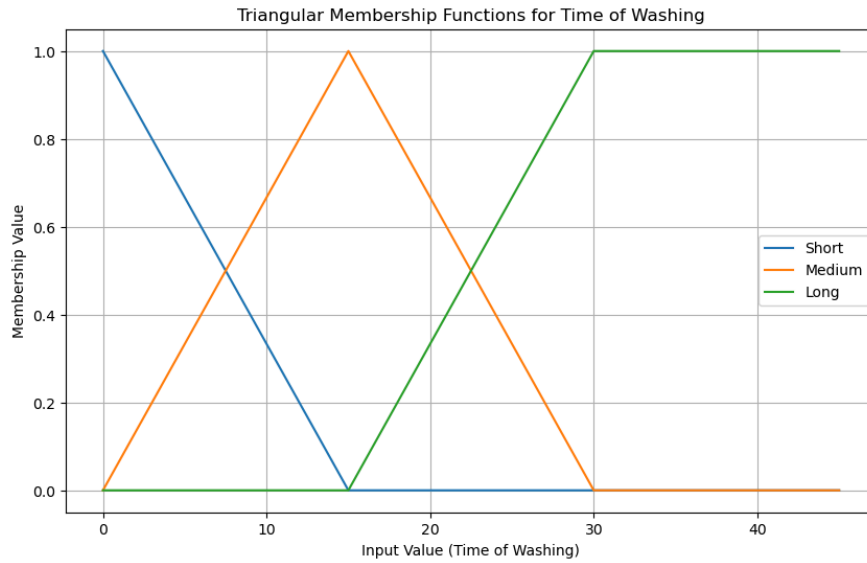
Pressure:



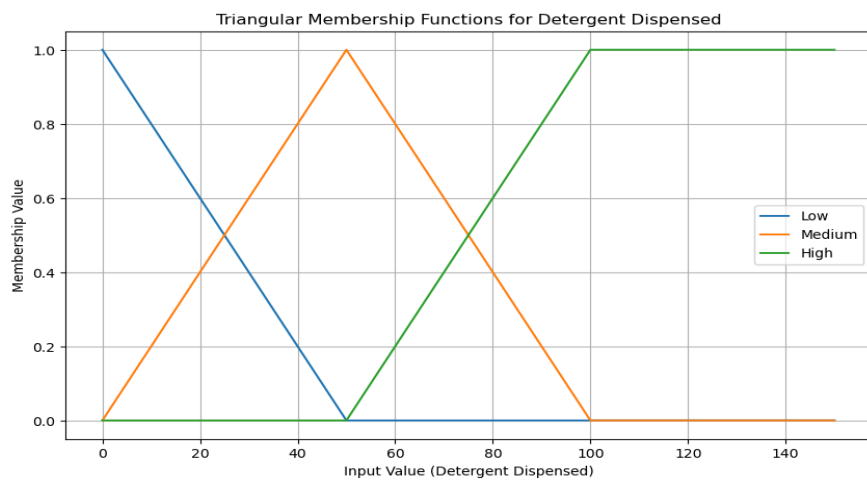
Spin:



Time:



Detergent:



b. A fuzzy rule could look like this:

Rule: If Weight is Very Light and Cloth Type is Cotton and Pressure is Low, then Spin Speed is Slow, Time of Washing is Short, and Detergent Dispensed is Little.

This rule illustrates how the linguistic terms for the input variables (Weight, Cloth Type, and Pressure) are combined to determine the linguistic terms for the output variables (Spin Speed, Time of Washing, and Detergent Dispensed) using fuzzy logic.

c. Total Rules in the Rule Base:

To determine the total number of rules, you multiply the number of linguistic terms for each input variable. Given:

$$\text{Total rules} = 5 * 3 * 3 = 45$$

d. Takagi and Sugeno Approach

Using the Takagi and Sugeno approach, a rule can be inferred by applying the fuzzy inference process to the antecedent part of the rule, which consists of fuzzy sets and logical operators.

- The antecedent part determines the degree of fulfilment of the rule for a given input.
- The output of the rule is computed by using the consequent part of the rule, which is a linear function of the input variables. The output of the rule is weighted by the degree of fulfilment of the rule.
- The final output of the system is obtained by aggregating the outputs of all the rules using a weighted average method.

For example, suppose we have the following rule for a washing machine:

IF weight is light AND cloth type is cotton THEN spin = 500 + 50 * weight

If the input is weight = 1.5 kg and cloth type = 2 (cotton), then we can infer the rule as follows:

- First, we calculate the membership values of the input variables in the fuzzy sets of the antecedent part using the triangular membership functions defined earlier.
 - weight is light: $\text{triangular_mf}(1.5, 0.5, 1.5, 2.5) = 1.0$
 - cloth type is cotton: $\text{trapezoidal_mf}(2, 1, 1, 3, 4) = 0.5$
- Second, we apply the logical operator AND to the membership values using the minimum function.
 - weight is light AND cloth type is cotton: $\min(1.0, 0.5) = 0.5$
- Third, we use the consequent part of the rule to compute the output value for the input variables using the linear function.
 - $\text{spin} = 500 + 50 * \text{weight} = 500 + 50 * 1.5 = 575$
- Fourth, we weight the output value by the degree of fulfilment of the rule, which is the result of the antecedent part.
 - $\text{weighted spin} = 0.5 * 575 = 287.5$

The final output of the system is the weighted average of the outputs of all the rules that are applicable for the input. For example, if there are two rules with outputs 287.5 and 312.5, and degrees of fulfilment 0.5 and 0.4, respectively, then the final output is:

$$(0.5 * 287.5 + 0.4 * 312.5) / (0.5 + 0.4) = 298.33$$

1a: Decide a minimally configured vanilla network that you should consider to build a prediction model with the given training data set. Justify the configure of your proposed network.

To build a prediction model with the given training dataset, we can use a minimally configured vanilla network with the following architecture:

- Input layer: 2 neurons (corresponding to Input (I1) and Input (I2))
- Hidden layer: 2 neurons
- Output layer: 1 neuron (corresponding to Output (O))

Justification:

- The input layer should have 2 neurons to accommodate the two input features (I1 and I2).
- The hidden layer with 2 neurons provides sufficient capacity for learning the underlying patterns in the data.
- The output layer with 1 neuron is suitable for regression tasks where the goal is to predict a continuous output (as in this case).

1b: For the network, you have given compute the following
i) Output of the input layer ii) Input to the hidden layer
ii) Output from the hidden layer iV) input to the output layer V) Output from the output layer

Let's assume some parameters to proceed further like activation function is sigmoid and learning rate is 0.5 and gradient descent is used.

- Weights from input layer to hidden layer: W_1
- Bias of hidden layer: b_1
- Weights from hidden layer to output layer: W_2
- Bias of output layer: b_2

Let's assume the weights and biases are initialized as follows for simplicity:

- $W_1 = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$
- $b_1 = \begin{bmatrix} 0.01 \\ 0.02 \end{bmatrix}$

- $W_2 = \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix}$
- $b_2 = 0.03$

The learning rate α is given as 0.5.

Forward Pass:

1. output of the input layer:

Same as input values

2. Input to the hidden layer:

$$Z_1 = XW_1 + b_1 = \begin{bmatrix} 0.4 & -0.7 \end{bmatrix} \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} + \begin{bmatrix} 0.01 \\ 0.02 \end{bmatrix} = \begin{bmatrix} -0.17 \\ -0.18 \end{bmatrix}$$

3. Output from the hidden layer (after applying sigmoid activation function):

$$A_1 = \sigma(Z_1) = \frac{1}{1+e^{-Z_1}} = \begin{bmatrix} \frac{1}{1+e^{0.17}} \\ \frac{1}{1+e^{0.18}} \end{bmatrix} = \begin{bmatrix} 0.457 \\ 0.455 \end{bmatrix}$$

4. Input to the output layer:

$$Z_2 = A_1W_2 + b_2 = \begin{bmatrix} 0.457 & 0.455 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix} + 0.03 = 0.548$$

5. Output from the output layer (after applying sigmoid activation function):

$$A_2 = \sigma(Z_2) = \frac{1}{1+e^{-Z_2}} = \frac{1}{1+e^{-0.548}} = 0.634$$

1C. Show the model parameters when the network got updated when it trained with any one training data.

We'll use the Mean Squared Error (MSE) loss function, which is defined as $L = \frac{1}{2}(A_2 - Y)^2$, where Y is the true output. The factor of 1/2 is included for convenience when taking the derivative.

1. Compute the derivative of the loss function with respect to A_2 :

$$\frac{\partial L}{\partial A_2} = A_2 - Y = 0.634 - 0.1 = 0.534.$$

2. Compute the derivative of A_2 with respect to Z_2 :

$$\frac{\partial A_2}{\partial Z_2} = A_2 * (1 - A_2) = 0.634 * (1 - 0.634) = 0.232.$$

3. Compute the derivative of the loss function with respect to Z_2 using the chain rule:

$$\frac{\partial L}{\partial Z_2} = \frac{\partial L}{\partial A_2} * \frac{\partial A_2}{\partial Z_2} = 0.534 * 0.232 = 0.124.$$

4. Compute the derivative of Z_2 with respect to W_2 , b_2 , and A_1 :

$$\frac{\partial Z_2}{\partial W_2} = A_1 = \begin{bmatrix} 0.457 \\ 0.455 \end{bmatrix}, \frac{\partial Z_2}{\partial b_2} = 1, \text{ and } \frac{\partial Z_2}{\partial A_1} = W_2 = \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix}.$$

5. Compute the derivative of the loss function with respect to W_2 and b_2 using the chain rule:

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial Z_2} * \frac{\partial Z_2}{\partial W_2} = 0.124 * \begin{bmatrix} 0.457 \\ 0.455 \end{bmatrix} = \begin{bmatrix} 0.057 \\ 0.056 \end{bmatrix} \text{ and } \frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial Z_2} * \frac{\partial Z_2}{\partial b_2} = 0.124 * 1 = 0.124.$$

6. Compute the derivative of the loss function with respect to A_1 using the chain rule:

$$\frac{\partial L}{\partial A_1} = \frac{\partial L}{\partial Z_2} * \frac{\partial Z_2}{\partial A_1} = 0.124 * \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix} = \begin{bmatrix} 0.062 \\ 0.074 \end{bmatrix}.$$

7. Compute the derivative of A_1 with respect to Z_1 :

$$\frac{\partial A_1}{\partial Z_1} = A_1 * (1 - A_1) = \begin{bmatrix} 0.457 * (1 - 0.457) \\ 0.455 * (1 - 0.455) \end{bmatrix} = \begin{bmatrix} 0.248 \\ 0.248 \end{bmatrix}.$$

8. Compute the derivative of the loss function with respect to Z_1 using the chain rule:

$$\frac{\partial L}{\partial Z_1} = \frac{\partial L}{\partial A_1} * \frac{\partial A_1}{\partial Z_1} = \begin{bmatrix} 0.062 \\ 0.074 \end{bmatrix} * \begin{bmatrix} 0.248 \\ 0.248 \end{bmatrix} = \begin{bmatrix} 0.015 \\ 0.018 \end{bmatrix}.$$

9. Compute the derivative of Z_1 with respect to W_1 and b_1 :

$$\frac{\partial Z_1}{\partial W_1} = X = \begin{bmatrix} 0.4 & -0.7 \end{bmatrix} \text{ and } \frac{\partial Z_1}{\partial b_1} = 1.$$

10. Compute the derivative of the loss function with respect to W_1 and b_1 using the chain rule:

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial Z_1} * \frac{\partial Z_1}{\partial W_1} = \begin{bmatrix} 0.015 \\ 0.018 \end{bmatrix} * \begin{bmatrix} 0.4 & -0.7 \end{bmatrix} = \begin{bmatrix} 0.006 & -0.011 \\ 0.007 & -0.013 \end{bmatrix} \text{ and } \frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial Z_1} * \frac{\partial Z_1}{\partial b_1} = \begin{bmatrix} 0.015 \\ 0.018 \end{bmatrix} * 1 = \begin{bmatrix} 0.015 \\ 0.018 \end{bmatrix}.$$

Update Weights and Biases:

- $W_1 = W_1 - \alpha \frac{\partial L}{\partial W_1} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} - 0.5 * \begin{bmatrix} 0.006 & -0.011 \\ 0.007 & -0.013 \end{bmatrix} = \begin{bmatrix} 0.097 & 0.206 \\ 0.297 & 0.407 \end{bmatrix}$
- $b_1 = b_1 - \alpha \frac{\partial L}{\partial b_1} = \begin{bmatrix} 0.01 \\ 0.02 \end{bmatrix} - 0.5 * \begin{bmatrix} 0.015 \\ 0.018 \end{bmatrix} = \begin{bmatrix} 0.003 \\ 0.011 \end{bmatrix}$
- $W_2 = W_2 - \alpha \frac{\partial L}{\partial W_2} = \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix} - 0.5 * 0.244 = \begin{bmatrix} 0.378 \\ 0.478 \end{bmatrix}$
- $b_2 = b_2 - \alpha \frac{\partial L}{\partial b_2} = 0.03 - 0.5 * 0.124 = -0.032$

4. Activation Functions

1. Log-Sigmoid Function

- Expression: $f(x) = \frac{1}{1+e^{-x}}$
- Parameters: The input x is the only parameter.
- Graph: The graph starts at 0, crosses 0.5 at $x = 0$, and ends at 1. It's an S-shaped curve. The steepness of the curve is controlled by the magnitude of x .

2. Tanh-Sigmoid Function

- Expression: $f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- Parameters: The input x is the only parameter.
- Graph: The graph starts at -1, crosses 0 at $x = 0$, and ends at 1. It's also an S-shaped curve, but it ranges from -1 to 1.

3. Radial Bias Function

- Expression: $f(x) = e^{-(x-c)^2/(2\sigma^2)}$
- Parameters: The center c and the spread σ .
- Graph: The graph is a bell-shaped curve centered at c . The width of the bell is controlled by σ .

4. Rectified Linear Unit Function (ReLU)

- Expression: $f(x) = \max(0, x)$
- Parameters: The input x is the only parameter.
- Graph: The graph is zero for negative x and linear with slope 1 for positive x .

5. Piecewise Linear Function

- Expression:

$$f(x) = \begin{cases} m_1x + c_1 & \text{if } x < x_1 \\ m_2x + c_2 & \text{if } x_1 \leq x < x_2 \\ \vdots & \\ m_nx + c_n & \text{if } x \geq x_{n-1} \end{cases}$$

- Parameters: The slopes m_i , the y-intercepts c_i , and the points x_i where the function changes.
- Graph: The graph is flat for $x < x_1$ and $x > x_n$, and has a slope of m_i between x_{i-1} and x_i for $i = 2, \dots, n$.