UNIVERSITY OF THESSALY

NEURO-FUZZY COMPUTING

ECE447

1st Problem Set

Alexandra Gianni   Nikos Stylianou

ID: 3382   ID: 2917

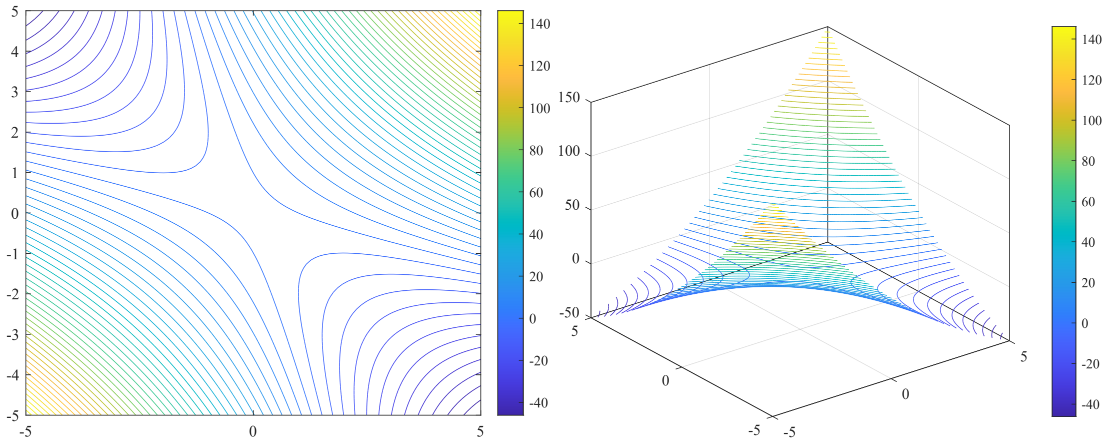
December 15, 2023

1 Problem 1

The contour lines of $f(x, y)$ are plotted with the following MATLAB code and are presented in figure 1.

```
function [Z] = plot_contour(start_num, end_num)

        x = linspace(start_num, end_num, 100);
        y = x;
        [X, Y] = meshgrid(x, y);
        Z = X.^2 + 4*X.*Y + Y.^2;
        contour(X, Y, Z, 40);
        xlabel('X');
        ylabel('Y');
end
```



(a) 2D Plot (b) 3D Plot

Figure 1: Contour lines of $f(x, y)$

A general formula of a quadratic equation is $f(x, y) = ax^2+2bxy+cy^2$. Writing our formula in the previous form, we find that $a = 1,\ b = 2,\ c = 1$. Calculation of the discriminant can help us calculate the location of the function's local minimum/maximum.

$$D = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix} = f_{xx}f_{yy} - f_{xy}^2 = 2 \times 2 - 4^2 = -12 < 0, \quad \text{where}$$
$$f_{xx} = \frac{\partial^2 f}{\partial x^2} = 2, \quad f_{yy} = \frac{\partial^2 f}{\partial y^2} = 2, \quad f_{xy} = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right) = 4D = 2 \times 2 - 4^2 = -12 < 0. \tag{1}$$

So, we only have to find the point at which $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ are equal to 0. This point will be a saddle point at which the gradients in each orthogonal direction are 0, but this point is neither a local minimum nor a maximum. More precisely:

$$\begin{cases} \dfrac{\partial f}{\partial x} = 2x + 4y = 0 \\ \dfrac{\partial f}{\partial y} = 4x + 2y = 0 \end{cases} \Rightarrow \begin{cases} x = 0 \\ y = 0 \end{cases} \tag{2}$$

Thus, the point $(x, y) = (0, 0)$ is the saddle point mentioned for the given function and this can be justified using the plotted contour lines.

2 Problem 4

In this problem, we will express the derivatives in respect to the matching activation function.

2.1 LogSig

This activation function is expressed as

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

Multiplying itself with $(1+e^{-x})$ gives

$$\left(1+e^{-x}\right)\ S(x) = 1 \Leftrightarrow e^{-x} = \frac{1}{S(x)} - 1$$

So, activation function's derivative will be

$$\begin{aligned}\frac{dS}{dx} &= \frac{d\left(\left(1+e^{-x}\right)^{-1}\right)}{dx} = \left(1+e^{-x}\right)^{-2}\ e^{-x} = S^2(x)\ \left(\frac{1}{S(x)} - 1\right)\\ &= S(x) - S^2(x) = S(x)\ \left(1 - S(x)\right)\end{aligned}$$

2.2 TanSig

Activation function is

$$S(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Its derivative is

$$\begin{aligned}\frac{dS}{dx} &= \frac{\left(e^x + e^{-x}\right)\left(e^x + e^{-x}\right) - \left(e^x - e^{-x}\right)\left(e^x - e^{-x}\right)}{\left(e^x + e^{-x}\right)^2} = \\ &= \frac{\left(e^x + e^{-x}\right)^2 - \left(e^x - e^{-x}\right)^2}{\left(e^x + e^{-x}\right)^2} = 1 - \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right)^2 = 1 - S^2(x)\end{aligned} \tag{3}$$

2.3 Swish

Activation function is

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

The derivative in respect to x is

$$\frac{dS}{dx} = \frac{1 + e^{-x} + xe^{-x}}{\left(1+e^{-x}\right)^2} = \frac{1+e^{-x}}{\left(1+e^{-x}\right)^2} + \frac{xe^{-x}}{\left(1+e^{-x}\right)^2} = \frac{1}{1+e^{-x}} + x\frac{e^{-x}}{\left(1+e^{-x}\right)^2}$$

Rewriting the function gives us

$$\frac{S(x)}{x} = \frac{1}{1+e^{-x}} \quad \text{and} \quad e^{-x} = \frac{x - S(x)}{S(x)}$$

So, continuing with the derivative:

$$\begin{aligned}\frac{dS}{dx} &= \frac{1}{1+e^{-x}} + x\frac{e^{-x}}{(1+e^{-x})^2} = \frac{S(x)}{x} + \frac{e^{-x}}{1+e^{-x}}\frac{x}{1+e^{-x}} = \frac{S(x)}{x} + S(x)\frac{x}{1+e^{-x}} = \\ &= \frac{S(x)}{x} + S(x)\frac{S(x)}{x}e^{-x} = \frac{S(x)}{x}\left(1+S(x)e^{-x}\right) = \frac{S(x)}{x}\left(1+S(x)\frac{x-S(x)}{S(x)}\right) = \\ &= \frac{S(x)}{x}\left(1+x-S(x)\right) = \frac{S(x)}{x} + S(x) - \frac{S^2(x)}{x}\end{aligned}$$

Let $\sigma = \dfrac{1}{1+e^{-x}}$, thus

$$\frac{dS}{dx} = \frac{S(x)}{x} + S(x) - \frac{S^2(x)}{x} = \sigma + S(x) - \sigma S(x) = S(x) + \sigma\left(1-S(x)\right) \tag{4}$$

2.4 Custom tanh

Activation function is

$$S(x) = x\ \tanh\left(\ln\left(1+e^x\right)\right) = x\ \tanh\left(g(x)\right),\ \text{where } g(x) = \ln\left(1+e^x\right).$$

By calculating the derivative of this function as is, we get

$$\begin{aligned}\frac{dS}{dx} &= \tanh(g(x)) + x\ \tanh^{'}(g(x))\ \frac{dg}{x} = \tanh(g(x)) + x\ \tanh^{'}(g(x))\ \frac{1}{1+e^{-x}} = \\ &= \tanh(g(x)) + \tanh^{'}(g(x))\ x\ \frac{\text{Swish}(x)}{x} = \frac{S(x)}{x} + \tanh^{'}(g(x))\ \text{Swish}(x) = \\ &= \frac{S(x)}{x} + (1-\tanh^2(g(x)))\ \text{Swish}(x) = \frac{S(x)}{x} + \text{Swish}(x) - \tanh^2(g(x))\ \text{Swish}(x)\end{aligned}$$

So, $\phi(x,S)$ is

$$\frac{S(x)}{x} + \text{Swish}(x) - \frac{S^2(x)}{x^2}\ \text{Swish}(x), \quad \text{where} \tag{5}$$
$S(x)$ is the activation function and $\text{Swish}(x)$ is the Swish activation function from before

3 Problem 5

The given neural network consists of two layers and of three neurons. On the first one, activation function is `logsig` or `swish` and on the second one is `purelin`. On the left side of figure 2 we see the sketches for all outputs when the activation function is `logsig` and on the right side all outputs with activation function being `swish`.

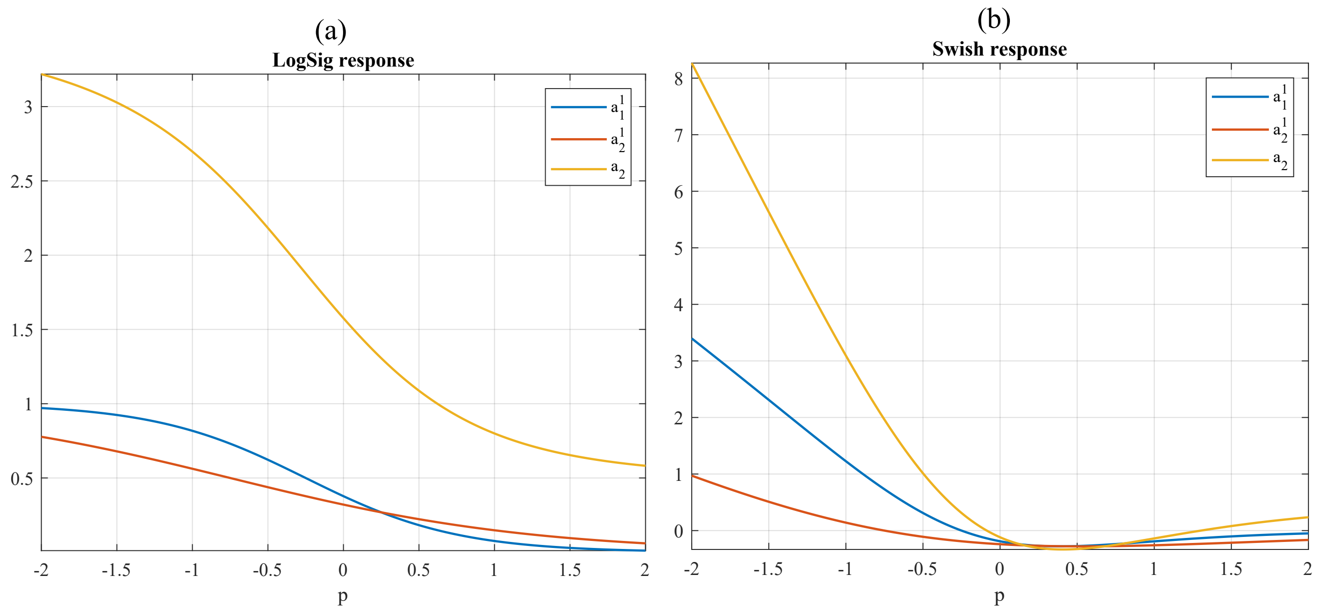


Figure 2: Responses of different outputs of the neural network

4 Problem 8

ADALINE is a single-layer artificial neural network that can learn and adapt to non-linear relationships between inputs and outputs.
It consists of a single neuron with a linear activation function. Each input of the neuron has a corresponding weight,which is adapted during training to minimize the error between the network's output and the desired output. To adjust the weights the algorithm uses the learning rule α.

Suppose that we have the following three reference patterns and their targets:

$$\left\{p_1 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, t_1 = [26]\right\} \quad \left\{p_2 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, t_2 = [26]\right\} \quad \left\{p_3 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, t_3 = [-26]\right\}$$

The probability of vector p1 is P1 = 0.20, the probability of vector p2 is P2 = 0.70, and the probability of vector p3 is P3 = 0.10.

4.1 Question a

The number of inputs to an ADALINE network for any neural network is determined by the dimensionality of our data,not the number of patterns that we have. In our case, each pattern is a 2-dimensional vector. Therefore, our ADALINE network has two inputs, one for each dimension.
In an ADALINE network, the number of weights is equal to the number of inputs. In our case, each pattern is a 2-dimensional vector. Thus, our neaural network will have two weights, one for each dimension of the input.

5 Problem 10

5.1 Question A

The patterns that we want to separate are plotted in figure 3.

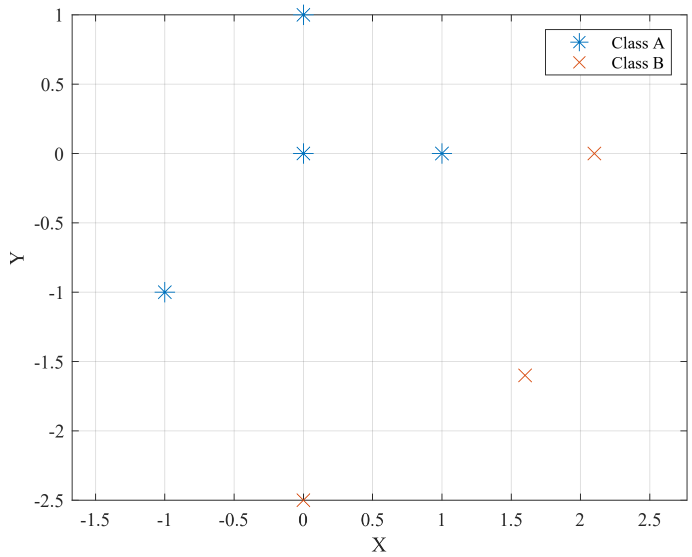


Figure 3: Plot of patterns

We can clearly see that there can be a straight line that can separate the two classes, thus an ADALINE neural network can work in classification for this system.

5.2 Question B

The designed ADALINE neural network will be of the following architecture

- Input Layer: Since the patterns are two-dimensional (each pattern has two values), the input layer will have two nodes.
- Output Layer: The output layer will have one node. This is because the task is a binary classification. The output node will use a linear activation function, as is standard in ADALINE networks.
- Weights and Bias: There will be two weights (one for each input node) and one bias. The weights and bias are parameters that the network will learn during the training process.
- Learning Rule: The network will use the LMS learning rule (*Least Mean Square*) to update the weights and bias. This rule minimizes the mean square error between the network's output and the target output.

This architecture described beforehand is shown in figure 4.

5.3 Question c

The ADALINE neural network mentioned above was coded in MATLAB. During training, we plotted its weights and bias in order to check their progression. Maximum iteration value was defined in code to be 10^4

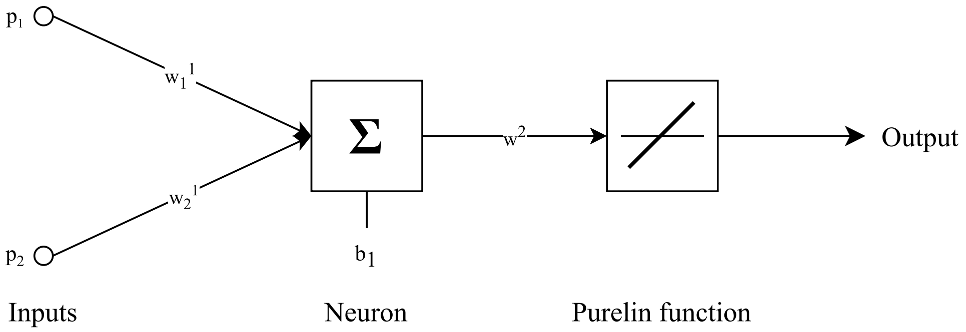


Figure 4: ADALINE neural network architecture

and minimum error to end train and consider the solutions converged is $epsilon$ of the machine, specifically $eps = 2.2204\ 10^{-16}$.
After converging, the final weights and bias are presented in table 1 below.

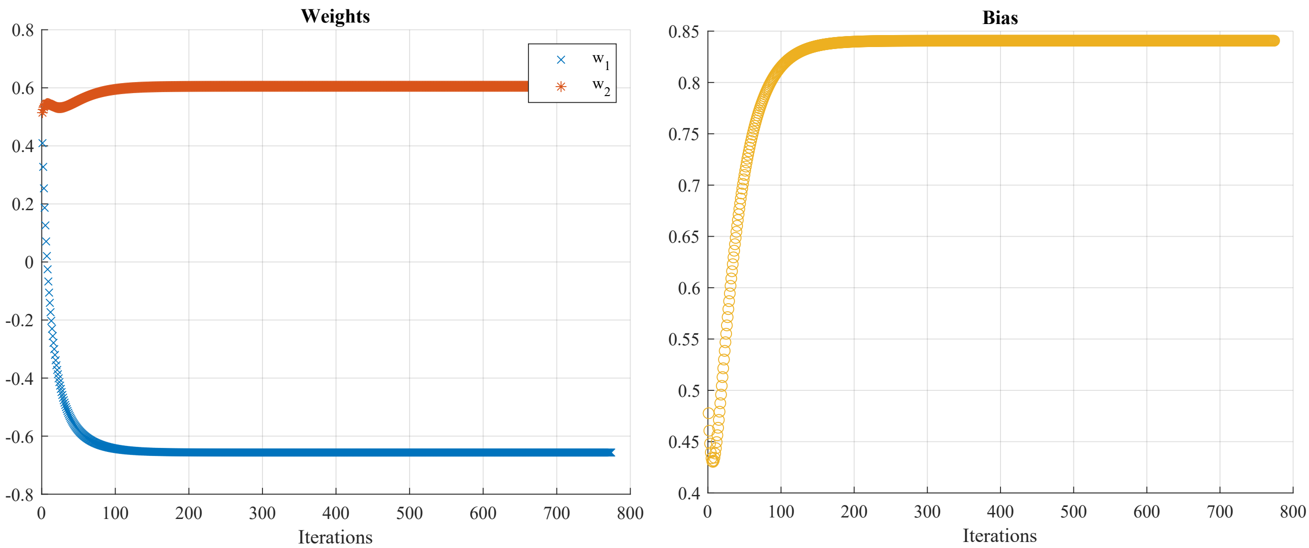


Figure 5: Plots of weights and biases during training

Weight 1	**Weight 2**	**Bias**
-0.6564	0.6052	0.8407

Table 1: Final table of weights and bias

6 Problem 11

Fuzzy logic is a type of logic that deals with vague, imprecise, or uncertain information. It is based on the concept of fuzzy sets, which are sets that can have any degree of membership between 0 and 1. The value zero is used to represent complete non-membership, the value one is used to represent complete membership, and values in between are used to represent intermediate degrees of membership.This means that an element can be a member of a fuzzy set to some degree, rather than all or nothing.

The uniqueness of fuzzy logic is that fuzzy logic can handle imprecise and uncertain information,which makes it a valuable tool for dealing with real-life problems that are inherently vague or fuzzy.
On this exercise, we are dealing with the linguistic variable *Truth* with a possible membership set:

T = Absolutely false, Very false, False, Fairly true, True, Very true, Absolutely true

Based on that set we may define the membership function of truth as:

True(u) = u   *False(u) = 1-u*

for each $u \in [0, 1]$.

7 Problem 12

In order to evaluate the expression "`not(A(x) OR B(x))`", we must first take a look at how fuzzy logic differs from binary logic at operation level. In binary logic we have three basic operations: `AND(x,y)`, `OR(x,y)` and `NOT(x)`. But, in fuzzy logic, where a function can have a value in the range of $[0...1]$, things are slightly different. The binary operation `AND(x,y)` is equivalent to `MIN(x,y)` from fuzzy logic, `OR(x,y)` to `MAX(x,y)` and `NOT(x)` to `1-x`.

We need to find the proper x for which the previous expression has the maximum value. First, we calculate the expression and then find the correct x. To achieve this, we need to divide our calculations into ranges.
Starting for $x \leq 2$, "$A(x)$ AND $B(x)$" is equal to "$max\,(A(x), B(x))$" $= 1$. Applying De Morgan's law, we therefore have $max\,(A(x), B(x)) \Rightarrow not\,(A(x)\ or\ B(x)) = 0$. Exactly the same result is obtained with $x \geq 7$.
Things are a bit different in $2 \leq x \leq 7$. The function $A(x)$ starts to fall while $B(x)$ starts to rise. The point at which the two functions cross is important for the definition of the required expression and can be obtained by solving the equation:

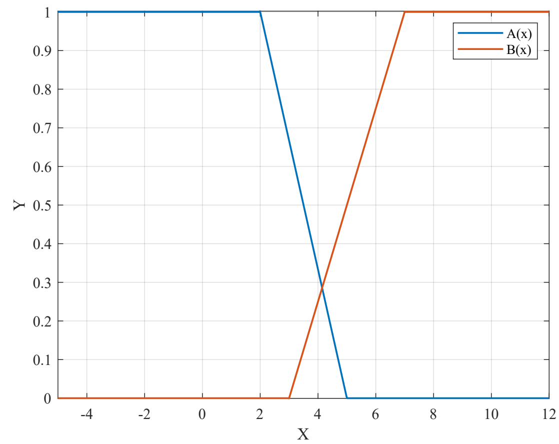


Figure 6: Plot of A(x), B(x)

$$A(x_{crit}) = B(x_{crit}) \Leftrightarrow 1 - \frac{x_{crit} - 2}{3} = \frac{x_{crit} - 3}{4} \Rightarrow x_{crit} = \frac{29}{7}$$

For $2 \leq x \leq \frac{29}{7}$, $max\,(A(x), B(x)) = 1 - \dfrac{x-2}{3} = A(x)$, because in this region $A(x)$ lies above $B(x)$.
Thus, $not\,(A(x)\ or\ B(x)) = \dfrac{x-2}{3}$.
Using the same logic, we find out that for $\frac{29}{7} \leq x \leq 7$, $not\,(A(x)\ or\ B(x)) = 1 - \dfrac{x-3}{4}$.
Therefore, the expression $f(x) = not\,(A(x)\ or\ B(x))$ is summarized below:

$$f(x) = \begin{cases} 0 & x \leq 2, \\ \dfrac{x-2}{3} & 2 \leq x \leq \frac{29}{7}, \\ 1 - \dfrac{x-3}{4} & \frac{29}{7} \leq x \leq 7, \\ 0 & x \geq 7 \end{cases} \tag{6}$$

By plotting this function in figure 7, we can clearly see that the maximum occurs at $x = x_{crit} = \dfrac{29}{7}$ and its value is $\dfrac{10}{14}$ or 0.715465.

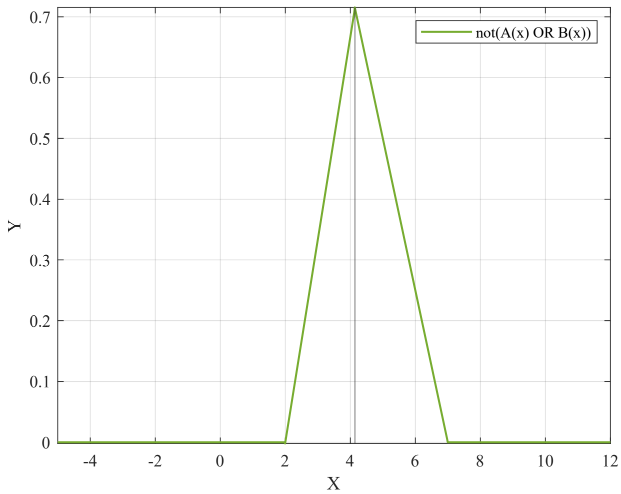


Figure 7: Expression's plot