Modelling Biological Systems

BIOS13 **Student:** Chandrashekar CR, ch1131ch-s@student.lu.se

Lecturer: Anders Brodin

> **Problem 1: ANN and GA (5p)**
>
> You should now be familiar with the simplest artificial neural network, the single-layer perceptron consisting of one node with two inputs and a bias.
>
> a) Explain why it cannot solve the logical operator XOR? (2p)

1a) The single-layer perceptron cannot solve the logical operator XOR due to the linear separability limitation. The XOR (exclusive OR) operator outputs 1 if exactly one if its two inputs is 1, and 0 otherwise. The truth table for XOR is given below:

Table 1: Truth Table for XOR Operation

Input 1 (x_1)	**Input 2 (x_2)**	**XOR Output (y)**
0	0	0
0	1	1
1	0	1
1	1	0

A perceptron applies a weighted sum of the inputs plus a bias component to determine whether the output is 0 or 1.

$$y = f(w_1x_1 + w_2x_2 + b) \quad (1)$$
$$\text{where f is an activation function}$$

A single layer perceptoron can classify inputs correctly only if the outputs can be separated by a straight line in the input space (2D). Logical AND and OR can be separated linearly in a 2D plane, but XOR cannot because the outputs form a pattern that is not linearly separable.

If we plot the inputs (x1,x2) and their corresponding XOR outputs in a 2D plane:

Points for XOR:

$$(0,0) \to 0$$
$$(0,1) \to 1$$
$$(1,0) \to 1$$
$$(1,1) \to 0$$

```r
# Clear environment
rm(list = ls())

# Load necessary library
```

```r
library(ggplot2)

# Create the XOR truth table as a data frame
xor_data = data.frame(
  Input1 = c(0, 0, 1, 1),
  Input2 = c(0, 1, 0, 1),
  Output = c(0, 1, 1, 0)
)

# Add a column for the class label
xor_data$Class = factor(xor_data$Output, labels = c("Class 0", "Class 1"))

# Plot the XOR data
xor_plot = ggplot(xor_data, aes(x = Input1, y = Input2, color = Class)) +
  geom_point(size = 5) +
  geom_abline(intercept = 0.5, slope = -1, linetype = "dashed", color = "green", linewidth =
      1) +
  geom_abline(intercept = 1.5, slope = -1, linetype = "dashed", color = "red", linewidth =
      1) +
  annotate("text", x = 0, y = 0, label = "(0, 0)", vjust = -1.2, size = 4) +
  annotate("text", x = 0, y = 1, label = "(0, 1)", vjust = -1.2, size = 4) +
  annotate("text", x = 1, y = 0, label = "(1, 0)", vjust = -1.2, size = 4) +
  annotate("text", x = 1, y = 1, label = "(1, 1)", vjust = -1.2, size = 4) +
  coord_cartesian(xlim = c(0.0, 1.5), ylim = c(0.0, 1.5)) +
  labs(
    title = "XOR Gate: Non-Linearly Separable Problem",
    x = "Input 1 (x1)",
    y = "Input 2 (x2)",
    color = "Class"
  ) +
  theme_classic() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    legend.position = "bottom"
  )

# Display the plot
xor_plot

ggsave("xor_plot.png",path = "./exam_plots", width = 7, height =5 , dpi=700)
```

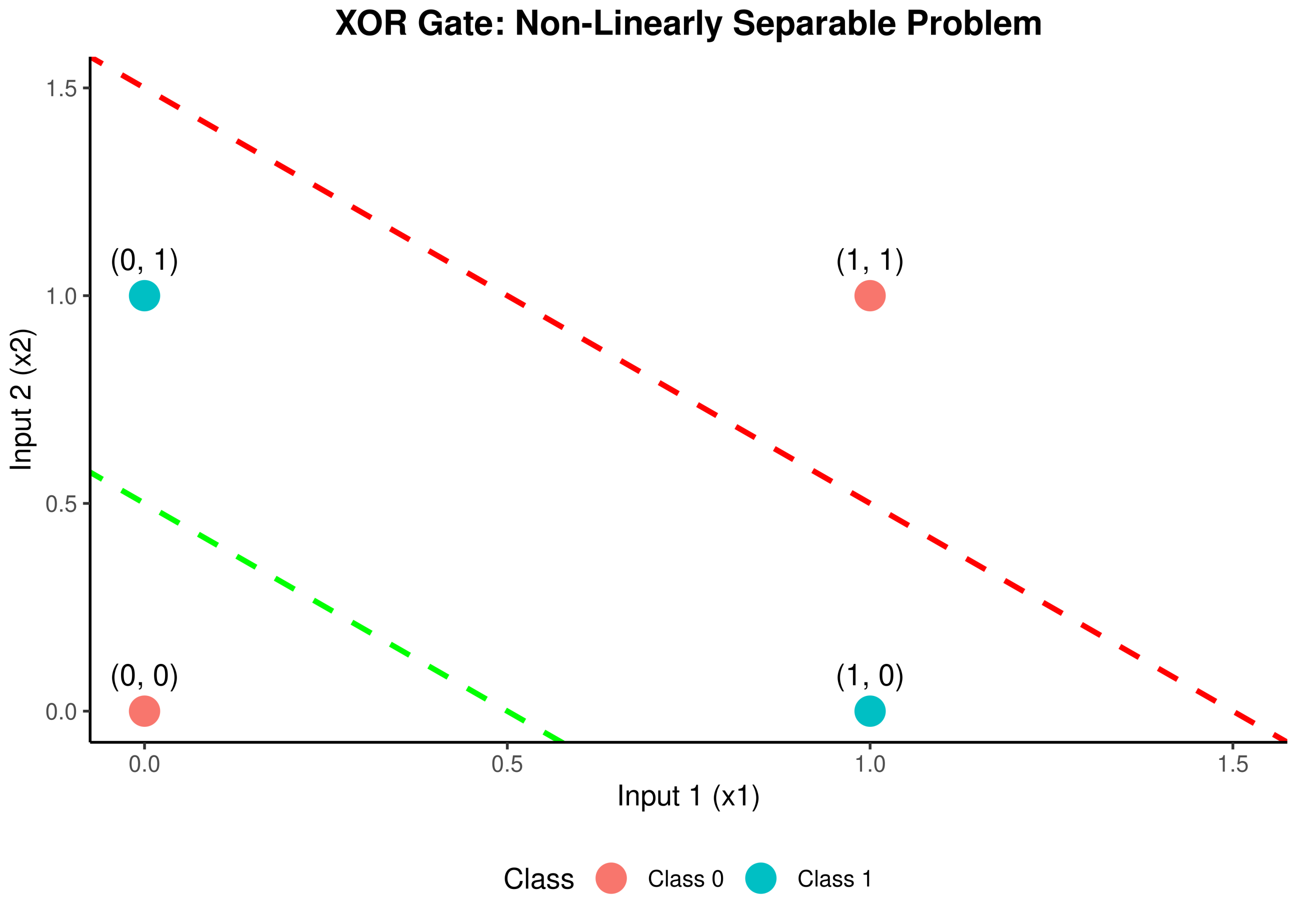

Figure 1: XOR Gate: Non-linearly separable problem

The plot shows that $(0, 0)$ and $(1, 1)$ belong to one class (output 0), $(0, 1)$ and $(1, 0)$ belong to another class (output 1). XOR requires a non-linear decision boundary such as a curved or segmented line, which a single-layer perceptron cannot produce. Therefore to solve the Logical XOR, a neural network must include atleast one hidden layer and non-linear activations functions.

> **Problem 1: ANN and GA (5p)**
>
> b)What will happen with the separation line if we remove the bias (or set it to zero)? (1p)

1b) If the bias is removed or set to zero in a single-layer perceptron, the separation line will always pass through the origin (0,0). This is because the equation of the decision boundary in a perceptron is given by:

$$\begin{gathered} w_1x_1 + w_2x_2 + b = 0 \\ \text{Where } w_1 \text{ and } w_2 \text{ are the weights for the inputs} x_1 \text{ and } x_2 \text{ and } b \text{ is the bias.} \\ \text{When } b \text{ is 0 (b=0),} \\ w_1x_1 + w_2x_2 = 0 \end{gathered} \tag{2}$$

This represents a line that always passes through the origin, as the constant term is eliminated. By removing the bias, the model's ability to shift the decision boundary away from the origin is lost. For the XOR, where a flexible separation is needed, removing the bias makes it even more impossible for the perceptron to learn th correct classification, as no line through the origin can separate the XOR points correctly.

> **Problem 1: ANN and GA (5p)**
>
> c) Give four differences between a binary (0 and 1) and a continuous (decimal numbers) genetic algorithm? (2p)

1c)

Table 2: Comparison of Binary and Continuous Genetic Algorithms

Sl No	Binary Genetic Algorithm	Continuous Genetic Algorithm
1	Solutions (individuals) are encoded as binary strings (e.g., 0 and 1).	Solutions are represented as real-valued (decimal) numbers.
2	Operates in a discrete search space.	Operates in a continuous search space.
3	Uses bitwise operations like flipping (mutation) and binary crossover.	Uses operations like Gaussian mutation and arithmetic crossover.
4	Limited precision due to binary encoding (depends on string length).	High precision as real numbers directly represent the solution.