Homework1: Perceptrons and Boolean Functions

(Course: GNR638) (Name: Deep Learning for Vision)

Due date: 22-01-2026

Learning Goals. By the end of this homework you should be able to: (i) write Boolean functions precisely (formula + truth table), (ii) test if a function is representable by a single perceptron (linear separability), (iii) design perceptron weights when possible, (iv) reason why some functions *cannot* be done by a single perceptron.

Background (Read This First)

A (binary) perceptron takes an input vector $\mathbf{x} \in \{0,1\}^n$ and produces an output

$$\hat{y} \;=\; \mathbb{I}\big[w^\top \mathbf{x} + b \geq 0\big],$$

where $w \in \mathbb{R}^n$ are weights, $b \in \mathbb{R}$ is the bias, and $\mathbb{I}[\cdot]$ is the indicator (equal to 1 if the condition is true, else 0).
Important fact: A Boolean function can be implemented by a single perceptron *if and only if* it is **linearly separable** in the input space.

Instructions

- You must answer all questions. Show your work.
- For each function, you must provide:
  - (a) A clear definition (formula).
  - (b) A complete truth table.
  - (c) A statement: *single perceptron possible?* (Yes/No) with justification.
  - (d) If **Yes**: provide (w, b) and verify it on all inputs.
  - (e) If **No**: give a short linear-separability argument (geometric or contradiction or separability test).

Problem 1: 3-bit Majority

Let $\mathbf{x} = (x_1, x_2, x_3) \in \{0,1\}^3$. Define the majority function:

$$\mathrm{MAJ}_3(x_1, x_2, x_3) \triangleq \mathbb{I}[x_1 + x_2 + x_3 \geq 2].$$

1.1 Write the full truth table for MAJ_3.

1.2 Show that MAJ_3 is linearly separable by explicitly giving perceptron parameters (w, b).

1.3 Verify your (w, b) on all 8 input patterns.

Problem 2: Exactly-1-of-3

Define:

$$\mathrm{EXACT1}_3(x_1, x_2, x_3) \triangleq \mathbb{I}[x_1 + x_2 + x_3 = 1].$$

2.1 Write the full truth table for $\mathrm{EXACT1}_3$.

2.2 Argue carefully why $\mathrm{EXACT1}_3$ is **not** linearly separable (hence not implementable by a single perceptron).

2.3 Propose a 2-layer network (one hidden layer) that can implement $\mathrm{EXACT1}_3$.

Problem 3: 3-bit Parity

Define 3-bit parity (odd parity):

$$\mathrm{PARITY}_3(x_1, x_2, x_3) \triangleq x_1 \oplus x_2 \oplus x_3,$$

where $\oplus$ is XOR.

3.1 Write the full truth table for PARITY_3.

3.2 Show that PARITY_3 is **not** linearly separable.

3.3 Give a small multilayer perceptron construction that computes parity using XOR compositions.

Problem 4: 4-to-1 Multiplexer

A 4-to-1 multiplexer has two *select* bits (s_1, s_0) and four *data* bits (d_0, d_1, d_2, d_3). It outputs the selected data bit:

$$\mathrm{MUX}_4(s_1, s_0, d_0, d_1, d_2, d_3) \triangleq \begin{cases} d_0, & (s_1 s_0) = 00, \\ d_1, & (s_1 s_0) = 01, \\ d_2, & (s_1 s_0) = 10, \\ d_3, & (s_1 s_0) = 11. \end{cases}$$

4.1 Write MUX_4 as a Boolean formula using AND ($\wedge$), OR ($\vee$), and NOT ($\neg$).

4.2 Create the truth table *in a structured way*: you do **not** need to write all $2^6 = 64$ rows explicitly. Instead, provide a compact table with 4 blocks (one block per select value) and within each block list the mapping from (d_0, d_1, d_2, d_3) to output.

4.3 Explain why MUX_4 is **not** linearly separable (hence not a single perceptron).

Problem 5: Parity of Two 3-bit Majorities

Let

$$m_A \triangleq \mathrm{MAJ3}(x_1, x_2, x_3), \qquad m_B \triangleq \mathrm{MAJ3}(x_4, x_5, x_6).$$

Define the final function as XOR of these two majority outputs:

$$f(x_1, \ldots, x_6) \triangleq m_A \oplus m_B.$$

5.1 Write f explicitly in words: when does it output 1?

5.2 Construct a *compact* truth description: You do **not** need to list all $2^6 = 64$ rows. Instead, treat (x_1, x_2, x_3) as one group and (x_4, x_5, x_6) as another, and organize the table in terms of $(m_A, m_B) \in \{0, 1\}^2$.

5.3 Argue why f is **not** representable by a single perceptron.

Submission Checklist

- All formulas are clearly written.
- Truth tables are complete (or compactly but correctly represented where allowed).
- Each "Yes/No" decision about single perceptron is justified.
- For Problem 1, you gave (w, b) and verified it.
- For Problems 2–5, you provided a clear non-separability argument.
- For good practice, please try to submit latex-written document.

Optional For each non-linearly separable function, design a small 2-layer network (with step activations) that computes it, and count the number of hidden units you used.