Part II
Concept Learning 2

(12 points) Structured questions. Answer in the space provided on the script.

1. (12 points) Consider the hypothetical task of learning the target concept $MLGrade$ to understand the factors affecting the grades of students enrolled in an ML class and the hypothesis space H that is represented by a conjunction of constraints on input attributes, as previously described on page 7 of the "Concept Learning" lecture slides. Each constraint on an input attribute can be a specific value, don't care (denoted by '?'), and no value allowed (denoted by '$\emptyset$'), as previously described on page 5 of the "Concept Learning" lecture slides. Each input instance is represented by the following input attributes:

   - $AttendClass$ (with possible values $Always$, $Sometimes$, $Rarely$),
   - $FinalsGrade$ (with possible values $Good$, $Average$, $Poor$),
   - $ProjectGrade$ (with possible values $Good$, $Average$, $Poor$), and
   - $LoveML$ (with possible values Yes, No).

   For example, a typical hypothesis in H is

   $$\langle ?, Average, ?, Yes \rangle \ .$$

   Trace the CANDIDATE-ELIMINATION algorithm (reproduced below in Fig. 1) for the hypothesis space H given the sequence of positive ($MLGrade = Pass$) and negative ($MLGrade = Fail$) training examples from Table 1 below (i.e., show the sequence of S and G boundary sets).

1. $G \leftarrow$ maximally general hypotheses in H
2. $S \leftarrow$ maximally specific hypotheses in H
3. For each training example d
   - If d is a positive example
     - Remove from G any hypothesis inconsistent with d
     - For each $s \in S$ not consistent with d
       - Remove s from S
       - Add to S all minimal generalizations h of s s.t.
         h is consistent with d, and
         some member of G is more general than or equal to h
       - Remove from S any hypothesis that is more general than another hypothesis in S
   - If d is a negative example
     - Remove from S any hypothesis inconsistent with d
     - For each $g \in G$ not consistent with d
       - Remove g from G
       - Add to G all minimal specializations h of g s.t.
         h is consistent with d, and
         some member of S is more specific than or equal to h
       - Remove from G any hypothesis that is more specific than another hypothesis in G

Figure 1: CANDIDATE-ELIMINATION algorithm.

Example Student	Input Instances $AttendClass$	$FinalsGrade$	$ProjectGrade$	$LoveML$	Target Concept $MLGrade$
1. $Ryutaro$	$Sometimes$	$Good$	$Poor$	Yes	$Pass$
2. $Haibin$	$Sometimes$	$Good$	$Average$	Yes	$Pass$
3. $Jinho$	$Rarely$	$Average$	$Average$	No	$Fail$
4. $Jingfeng$	$Sometimes$	$Poor$	$Average$	No	$Fail$

Table 1: Positive ($MLGrade = Pass$) and negative ($MLGrade = Fail$) training examples for target concept $MLGrade$.

Solution:

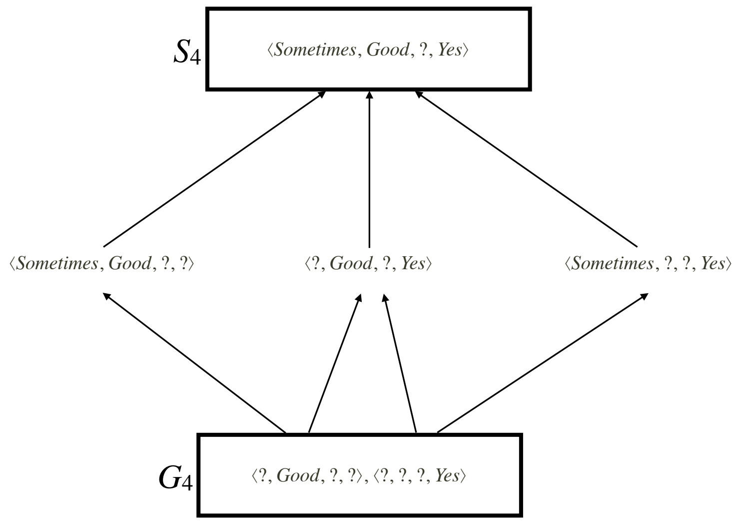
$$
\begin{array}{rcl}
G_0 &=& \{\langle ?,?,?,? \rangle\} \\
S_0 &=& \{\langle \emptyset,\emptyset,\emptyset,\emptyset \rangle\} \\
G_1 &=& G_0 \\
S_1 &=& \{\langle Sometimes, Good, Poor, Yes \rangle\} \\
G_2 &=& G_1 \\
S_2 &=& \{\langle Sometimes, Good, ?, Yes \rangle\} \\
S_3 &=& S_2 \\
G_3 &=& \{\langle Sometimes, ?,?,? \rangle, \langle ?, Good, ?, ? \rangle, \langle ?,?,?, Yes \rangle\} \\
 & & \textbf{Recall that } \forall g \in G_3 \ \exists s \in S_3 \ \ g \geqslant_g s\textbf{.} \\
 & & \textbf{That is, } \forall g \in G_3 \ \ g \geqslant_g \langle Sometimes, Good, ?, Yes \rangle\textbf{.} \\
S_4 &=& S_3 \\
G_4 &=& \{\langle ?, Good, ?, ? \rangle, \langle ?,?,?, Yes \rangle\}
\end{array}
$$

Suppose that the target concept c is in the hypothesis space H (i.e., $c \in H$) and an active learner has already observed the set D of 4 training examples in Table 1 above. State **every** possible input instance (i.e., assuming such a student exists) that the active learner can query next for the 5-th training example to reduce the version space $VS_{H,D}$ by at least half. Note that the active learner does not know the output label $c(x)$ of any input instance x that it has not yet observed.

Hint: Draw the version space $VS_{H,D}$.

Solution: The student only needs to provide the following 6 input instances as answers:
$\langle Always, Good, Good, Yes \rangle$, $\langle Always, Good, Average, Yes \rangle$, $\langle Always, Good, Poor, Yes \rangle$,
$\langle Rarely, Good, Good, Yes \rangle$, $\langle Rarely, Good, Average, Yes \rangle$, $\langle Rarely, Good, Poor, Yes \rangle$.

For example, if $c(\langle Always, Good, Good, Yes \rangle) = 0$, then VS reduces to $\{\langle Sometimes, Good, ?, ? \rangle$, $\langle Sometimes, ?, ?, Yes \rangle$, $\langle Sometimes, Good, ?, Yes \rangle\}$. If $c(\langle Always, Good, Good, Yes \rangle) = 1$, then VS reduces to $\{\langle ?, Good, ?, ? \rangle, \langle ?, ?, ?, Yes \rangle, \langle ?, Good, ?, Yes \rangle\}$. Every of the 6 input instances reduces the version space VS by exactly half.



Part V
Neural Networks

(20 points) Structured questions. Answer in the space provided on the script.

1. (4 points) Supposing the weights w_1 and w_2 of a perceptron (see page 6 of "Neural Networks" lecture slides) are both set to the value of **–1**, derive the largest possible range of the values of w_0 that can be set for the perceptron to represent the NAND gate (i.e., NAND(x_1, x_2)). Assume that the inputs x_1 and x_2 and output $o(x_1, x_2)$ of the perceptron are Boolean with the values of **1 or –1**. Show the steps of your derivation. **No marks will be awarded for not doing so.**

Solution:

- Case 1: Perceptron outputs false if both inputs are true (i.e., $o(x_1, x_2) = -1$ if $x_1 = 1 \wedge x_2 = 1$).

$$\begin{array}{rrcll} & x_1 + x_2 & \underset{x_1=1\wedge x_2=1}{=} & 1+1 & \\ \Rightarrow & w_0 - x_1 - x_2 & = & w_0 - 2 & \underset{o(x_1,x_2)=-1}{\leq} 0 \\ \Rightarrow & & & w_0 & \leq 2\,. \end{array}$$

- Case 2: Perceptron outputs true if some input is false (i.e., $o(x_1, x_2) = 1$ if $x_1 = -1 \vee x_2 = -1$).

$$\begin{array}{rrcll} & x_1 + x_2 & \underset{x_1=-1\vee x_2=-1}{\leq} & -1+1 & \\ \Rightarrow & w_0 - x_1 - x_2 & \geq & w_0 & \underset{o(x_1,x_2)=1}{>} 0\,. \end{array}$$

Therefore, $0 < w_0 \leq 2$.

2. (8 points) Supposing the weights $w_1, w_2, \ldots, w_n$ of a perceptron (see page 6 of "Neural Networks" lecture slides) are all set to the value of **1**, derive the largest possible range of the values of w_0 (in terms of n) that can be set for the perceptron to represent the OR function. That is, the perceptron outputs false if all n Boolean inputs to the perceptron are false, and true otherwise. Assume that the inputs $x_1, x_2, \ldots, x_n$ and output $o(x_1, x_2, \ldots, x_n)$ of the perceptron are Boolean with the values of **1 (i.e., true) or –1 (i.e., false)**. Show the steps of your derivation. **No marks will be awarded for not doing so.**

Solution:

- Case 1: Perceptron outputs false if all inputs are false (i.e., $o(x_1, \ldots, x_n) = -1$ if $\wedge_{i=1}^n (x_i = -1)$).

$$\begin{array}{rrclcl} & \sum_{i=1}^n x_i & \underset{\wedge_{i=1}^n (x_i=-1)}{=} & \sum_{i=1}^n -1 & = & -n \\ \Rightarrow & w_0 + \sum_{i=1}^n x_i & = & w_0 - n & \underset{o(x_1,\ldots,x_n)=-1}{\leq} & 0 \\ \Rightarrow & & & w_0 & \leq & n\,. \end{array}$$

- Case 2: Perceptron outputs true if at least one input is true (i.e., $o(x_1, \ldots, x_n) = 1$ if $\vee_{i=1}^n (x_i = 1)$).

$$\begin{array}{rrclcl} & \sum_{i=1}^n x_i & \underset{\vee_{i=1}^n (x_i=1)}{\geq} & 1 + (n-1)(-1) & = & 2-n \\ \Rightarrow & w_0 + \sum_{i=1}^n x_i & \geq & w_0 + 2 - n & \underset{o(x_1,\ldots,x_n)=1}{>} & 0 \\ \Rightarrow & & & w_0 & > & n-2\,. \end{array}$$

Therefore, $n - 2 < w_0 \leq n$.

3. (8 points) Construct and draw a network of perceptron units with **only one hidden layer (of four units)** that implements (x_1 XOR x_2) XOR x_3 based on the following rules:

   - There should be only one (Boolean) output unit and an input unit for every (Boolean) input.
   - A Boolean is **–1** if false, and **1** if true.
   - The activation function of every (non-input) unit is a -1 to 1 step function (refer to page 6 of the "Neural Networks" lecture slides), including that of the output unit.
   - **Your weights must take on one of the following values: –1, 0, 1, 3.**
   - You don't have to draw edges with weight 0.

   Hint: Observe the truth table of (x_1 XOR x_2) XOR x_3.

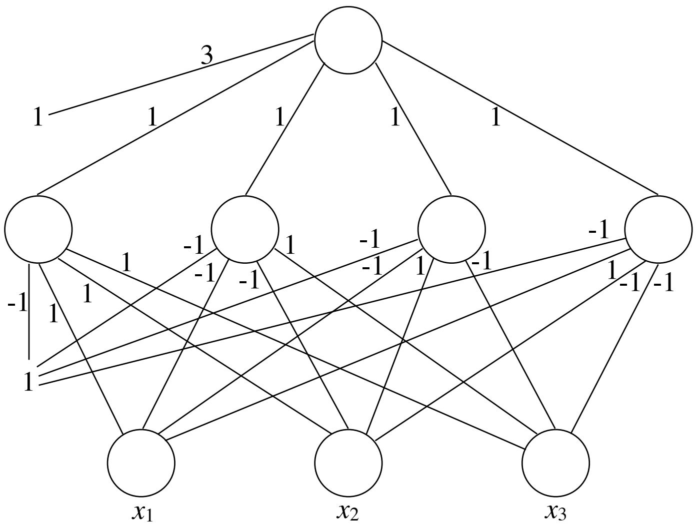
Solution: For the first hidden unit to fire, $x_1 = 1 \wedge x_2 = 1 \wedge x_3 = 1$, which corresponds to first row of truth table.

For the second hidden unit to fire, $x_1 = -1 \wedge x_2 = -1 \wedge x_3 = 1$, which corresponds to second row of truth table.

For the third hidden unit to fire, $x_1 = -1 \wedge x_2 = 1 \wedge x_3 = -1$, which corresponds to third row of truth table.

For the fourth hidden unit to fire, $x_1 = 1 \wedge x_2 = -1 \wedge x_3 = -1$, which corresponds to fourth row of truth table.

x_1	x_2	x_3	(x_1 XOR x_2) XOR x_3
1	1	1	1
-1	-1	1	1
-1	1	-1	1
1	-1	-1	1
-1	-1	-1	-1
1	1	-1	-1
1	-1	1	-1
-1	1	1	-1



END OF PAPER