Q1 Score: **6.0** **Total Score: 92.5**

1. [10%] General AI Knowledge

For each of the statements below, fill in the bubble T if the statement is always and unconditionally true, or fill in the bubble F if it is always false, sometimes false, or just does not make sense.

1		
2	T	
3		F
4	T	
5	T	
6	T	
7	T	
8	T	
9	T	
10		F

1. It is clear and transparent to humans how learning happens in neural networks.
2. A single layer perceptron allows making XOR separable by one line.
3. A Hopfield network becomes unstable above a certain number of neurons per pattern.
4. Bayes' Theorem takes you from a prior belief to a posterior belief after you've collected some additional data.
5. Given the current state of the art as stated in class, it looks likely that a single neural network will be able to do everything the human brain can do.
6. The number of hidden layers needed in a neural network can be calculated from the input provided.
7. Ockham's razor means that we reject the simplest hypothesis that is consistent with the data.
8. The simplest neuron models use a continuous notion of time.
9. If we have an algorithm or formula that solves a problem, then learning is not necessary.
10. Genetic algorithms are well suited to determine the number of hidden units in neural networks.

2. [20 %] Planning

Consider the following STRIPS action schema for planning the end of semester:

BookFlight()
  Pre: HaveExamDate(Math), HaveExamDate(Physics)
  Post: Have(Ticket)

TakeExam(subject)
  Pre: At(School), HaveExamDate(subject), Prepared(subject), Enrolled(subject)
  Post: Passed(subject)

Drive (A, B)
  Pre: At(A)
  Post: At(B), ¬At(A)

Fly()
  Pre: At(Airport), Have(Ticket), Have(Luggage)
  Post: ¬At(Airport), ¬Have(Ticket), At(Home)

LookupExamDate(subject)
  Pre: [no preconditions]
  Post: HaveExamDate(subject)

Pack()
  Pre: At (School), Have(Gift)
  Post: Have(Luggage)

Shop()
  Pre: At(Store), Prepared(Math), Prepared(Physics)
  Post: Have(Gift)

Study(subject)
  Pre: HaveExamDate(subject)
  Post: Prepared(subject)

Given the following:

  Start: At(School), Enrolled(Math), Enrolled(Physics)
  Goal: At(Home), Have(Gift), Passed(Math), Passed(Physics)

Show the partial order plan for planning the end of semester. Your answer must be a **complete written plan listing a possible sequence of actions** and also show a **graph with the actions from the Start state to the Final state using the actions above**. Your graph must clearly show the **precedence relationships** between actions and causal links as applicable, allowing the plan executor to have as much choice in the sequencing of actions without violating any causal links. For each action, **show the required preconditions on the graph**. (Hint: You can assume that both exams are scheduled at different times that don't overlap.)

You should draw the partial order plan on the next page.

Q3 Score: **8.0**

2A. [8%] Written Plan

Ans: Plan is given below: * Start

(1) LookupExamDate(Math)
(2). LookupExamDate(Physics)
(3) Study(Math)
(4) Study(Physics).
(5) BookFlight()
(6). TakeExam(Math)
(7) TakeExam(Physics)
(8). Drive(School, Store)
(9). Shop()
(10) Drive(Store, School)
(11) Pack()
(12) Drive(School, Airport)
(13) Fly().

Goal: At(home). Passed(Math). Passed(Physics)

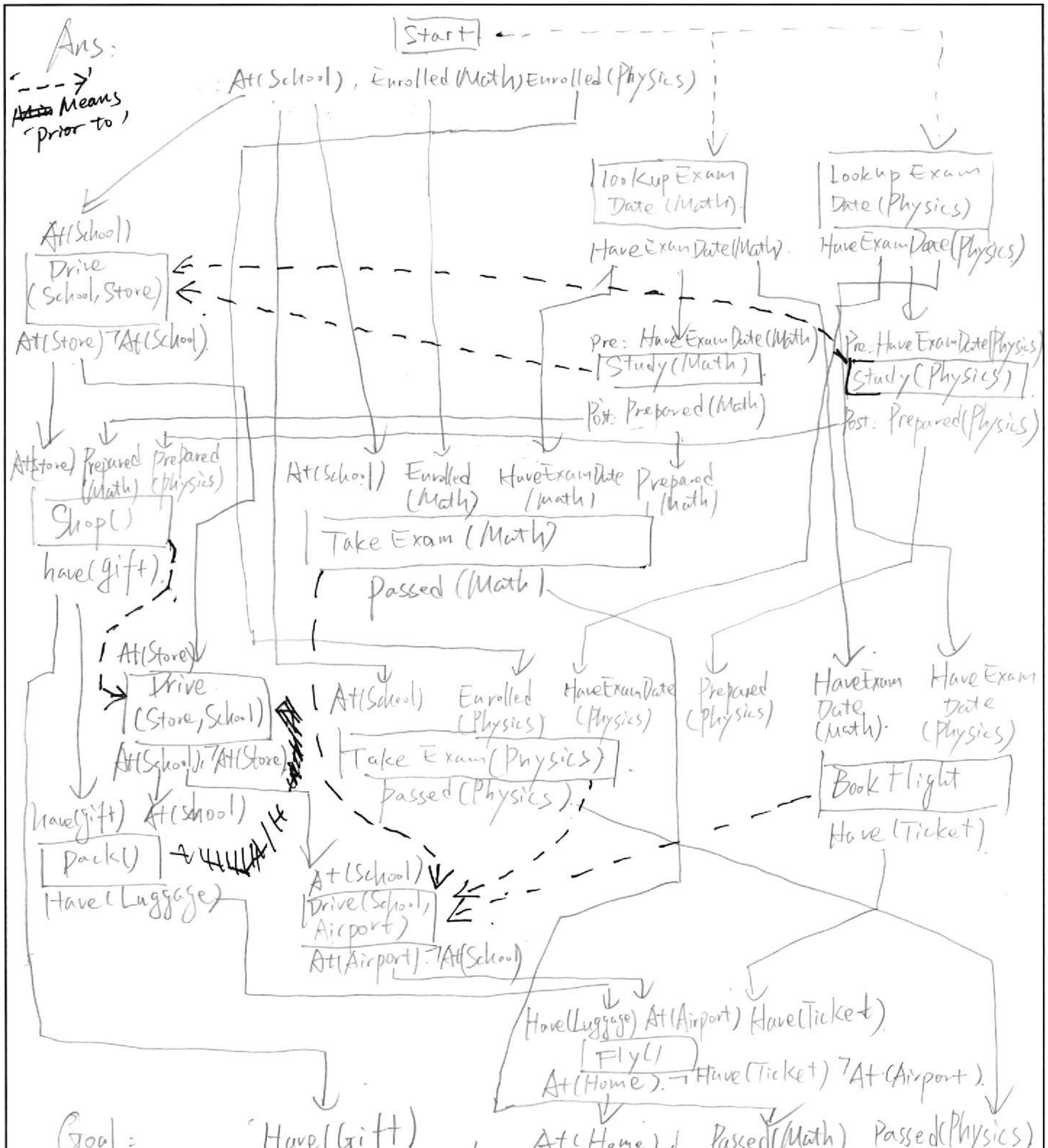
Q4 Score: **12.0**

2B. [12%] Graph (with Actions, and Preconditions and Precedence Relationships)

EE73C71A-9661-436C-87EA-D68AC3C1584B

561-F15-final

#49 5 of 10



Q5 Score: **20.0**

3. [20%] Probability

A [6%] You would like to know how useful it is for you to learn to take electric guitar lessons, so you would like to find out what the probability of getting hired by a heavy metal band is given that you took electric guitar lessons. What other probabilities would you need to know? (give a formula to show how these additional probabilities are used).

Ans:

P(getting hired ~~by a~~ | took electric guitar lessons) · P(took lessons)

= P (took lessons | getting hired) · P(getting hired)

according to the Bayesian Theorem.

So P(getting hired | took lessons) = $\dfrac{\text{P(took lessons | getting hired)} \cdot \text{P(getting hired)}}{\text{P(took lessons)}}$

B [14%] (1) We have made a machine to classify whether a room smells fresh or unpleasant based on parts per million (PPM) of methane gas molecules that are in the air. A robot has taken a sample from room which has 100 PPM. From prior sampling, we have derived the following probabilities:

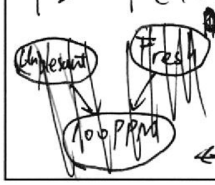
(a) P(100 PPM|Unpleasant) = 0.4
(b) P(100 PPM|Fresh) = 0.3
(c) P(Unpleasant) = 0.2
(d) P(Fresh) = 0.9
(e) P(100 PPM) = 0.2

Show using a Bayesian classifier whether the room's air should be classified as Fresh or Unpleasant. Be sure to show your work.

Ans: ~~P(100PPM)~~ ~~P(100PPM | Unpleasant, Fresh)~~

P_1 = P(Unpleasant | 100PPM) = $\dfrac{\text{P(100PPM | Unpleasant)} \cdot \text{P(Unpleasant)}}{\text{P(100PPM)}} = \dfrac{0.4 \times 0.2}{0.2} = 0.4$

P_2 = P(Fresh | 100PPM) = $\dfrac{\text{P(100PPM | Fresh)} \cdot \text{P(Fresh)}}{\text{P(100PPM)}} = \dfrac{0.3 \times 0.9}{0.2} = 1.35$



~~← Bayesian Network~~

$P_1 < P_2$ ∴ it is more likely to be fresh given the 100PPM in the air

4: [30%] Game Playing

Consider a normal version of the nim-game, a zero sum game:

The game consists of **N** piles of coins each containing ***m1, m2, ..., mN*** coins respectively. Two players take turns to pick up coins from the piles. During her turn, the player must pick 1 or more coins from exactly one pile (i.e. player cannot pick coins from different stacks in one turn). The player who picks up last wins the game.

For this question, assume that
N = 3; ***m1*** = 1; ***m2*** = 2; ***m3*** = 1

[A] [6%] Describe the (a) state representation to be used, (b) Initial State, and (c) Final/Terminal State

Ans: (a). State = { Turn of Player: [P_1 or P_2], coin remains in m_1, m_2 and m_3 } written as R_{m1}, R_{m2}, R_{m3}

(b). Initial State = { P_1, $R_{m1} = 1$, $R_{m2} = 2$, $R_{m3} = 1$ }.

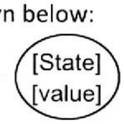
(c) Terminal State = { P_1/P_2, $R_{m1} = 0$, $R_{m2} = 0$, $R_{m3} > 0$
or $R_{m1} \cancel{>0} \geq 1$, $R_{m2} = 0$, $R_{m3} = 0$
or $R_{m1} = 0$, $R_{m2} > 0$, $R_{m3} = 0$ }.

[B] [4%] Describe the Utility function for the terminal states.

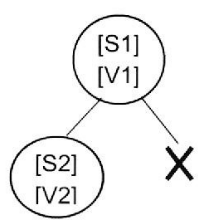
Ans: As is described in [A], the terminal states means, the player in who's turn at that state win the game, so the utility function can be written as: if P_2's turn, utility = −1 if P_1's turn, utility = 1.

[C] [20%] Draw the minimax game tree (on the next page) for the first player, and ***select the best move***. Apply the alpha-beta pruning algorithm while generating the minimax game tree. Use the following conventions:

1. Clearly mark the MAX and MIN levels in the tree.
2. For each node, describe the ***state*** using your representation described in **[A]** and the ***backed-up value*** according to the minimax algorithm, as shown below:



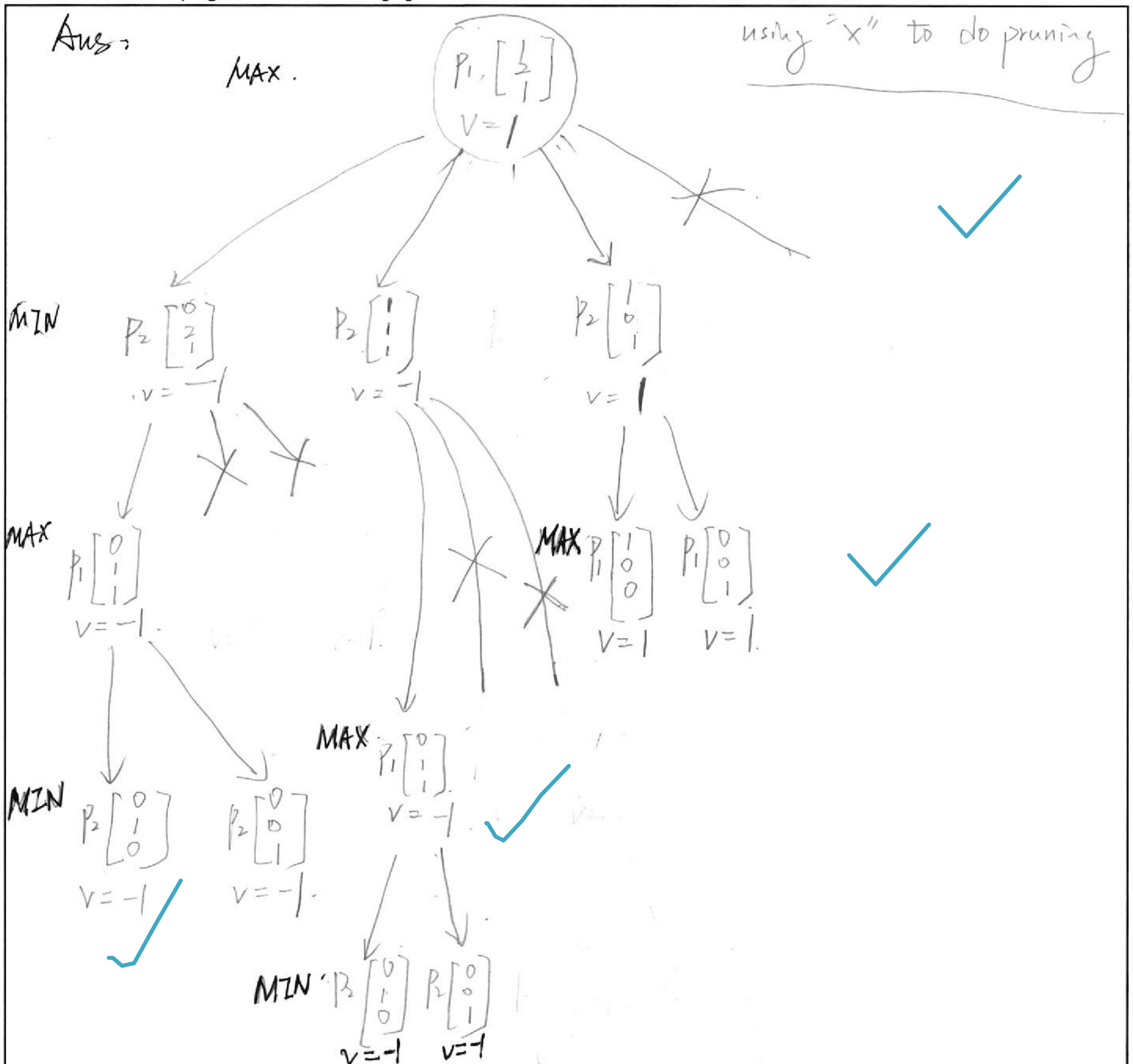
3. While applying the alpha-beta pruning algorithm, if some child nodes have been pruned, indicate that by a cross as follows:



Hint: You need not generate the entire tree, as you can stop as soon as the best move is found.

*NOTE: Use this page to answer **Q4 [C]**

Ans:



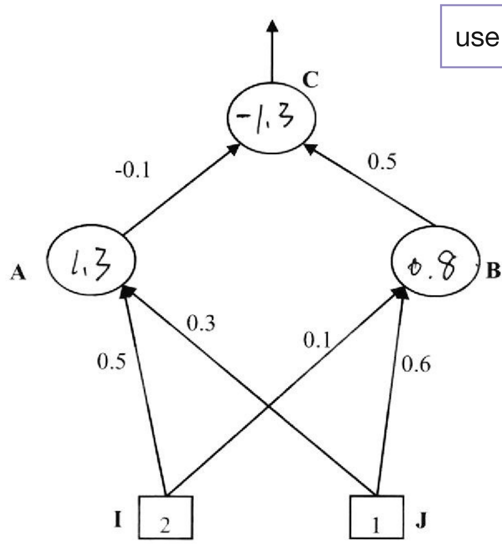
According to the MIN MAX Tree. P_1 pick all coins in pile 2 in first step will win the game.

Q8 Score: **4.5**

5 [20%] Artificial Neural Networks

5.1 [8%] Consider the neural network shown below, where I, J are inputs and A, B, C are perceptrons (neurons). For each perceptron assume its firing is based on the threshold rule discussed in class with Threshold = 1, i.e.,

$$f(x) = \begin{cases} 1, & if\ x \geq 1 \\ 0, & otherwise \end{cases}$$



use activation fn

−3

a) [3%] Write the ***output*** for each perceptron in the circles in the above figure.

b) [5%] Given the actual output to be 1, find the (a) error value, and the updated weights for (b) AC and (c) BC.

Error = | Expected - Observed | = | 1 - 0 |

Ans: (a). Error value = $y_0 - a_{out} = 2.3$. −0.5

According to the Backpropagation rule:

(b) $W_{AC} = -0.1 - 1 \cdot (-0.1) \cdot Error = -0.1 - (-0.23) = 0.13$ ✓

(c) $W_{BC} = 0.5 - 1(0.5) \cdot Error = 0.5 - 0.1 \cdot 15 = 0.5$ ✓

Q9 Score: **12.0**

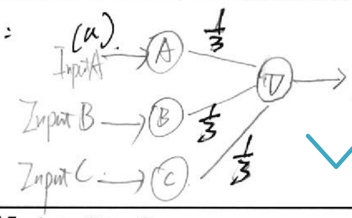
5.2 [12%] You are building a set of perceptrons to mimic logic circuits. For each perceptron assume its firing is based on the threshold rule discussed in class i.e.

$$f(x) = \begin{cases} 1, & if\ x \geq 1 \\ 0, & otherwise \end{cases}$$

Assume all inputs have value either 0 or 1. Show how perceptrons can be used to mimic the following logic sentences. For each expression, (a) draw the perceptrons, (b) show the weights (c) write the threshold.

a) **[2%]** $A \wedge B \wedge C$

Ans: (a)



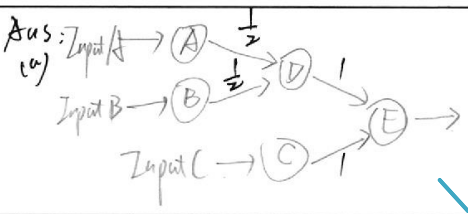
(b) ~~*~~ The ~~*~~ Output Node is D.

So. $W_{AD} = \frac{1}{3}$, $W_{BD} = \frac{1}{3}$, $W_{CD} = \frac{1}{3}$.

only A, B, C all assigned ~~to be~~ 1 can output be $\frac{1}{3}+\frac{1}{3}+\frac{1}{3} = 1 \geq 1$ otherwise it will output 0.

b) **[2%]** $A \wedge B \vee C$

Ans: (a)



(b). Output Node is E.

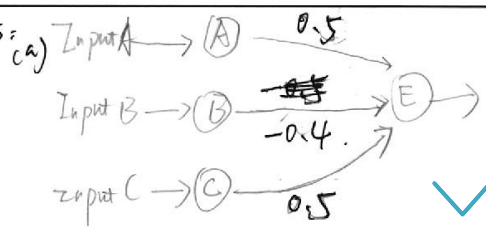
$W_{AD} = \frac{1}{2}$. $W_{BD} = \frac{1}{2}$.

$W_{CE} = 1$ $W_{DE} = 1$.

(c). Since Conjunction Prior to Disjunction A and B will calculate first and output 1 only when A = 1 and B = 1. ~~E~~ E will out put 1 either D = 1 or C = 1 or both C = D = 1.

c) **[3%]** $A \wedge \neg B \wedge C$

Ans: (a)



(b). Out put Node is E.

$W_{AE} = 0.5$ ~~$W_{AE}=1$~~.

$W_{CE} = 0.5$ ~~W_{CE}~~

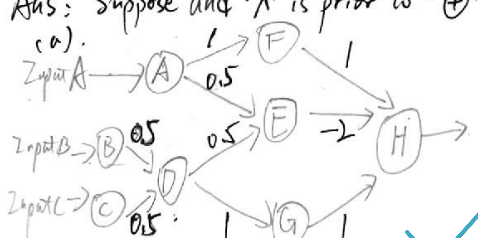
$W_{BE} = -0.4$.

(c). Output will be 1 Only when A = C = 1 and B = 0 other wise $\sum_{input} E < 1$.

d) **[5%]** $A \oplus B \wedge C$ (where $\oplus$ denotes the exclusive-OR (XOR) operator)

Ans: Suppose and 'Λ' is prior to '⊕'

(a).



(b) H is the output.

$W_{BD} = 0.5$ $W_{DG} = 1$

$W_{CD} = 0.5$. $W_{GH} = 1$

$W_{AF} = 1$ ~~W_{AH}~~

$W_{AE} = 0.5$ $W_{FH} = 1$

$W_{DE} = 0.5$ $W_{EH} = -2$

(c) D will be 1 only when B = C = 1.

$\begin{cases} A=1, D=1 \Rightarrow H=0. \\ A=1, D=0 \\ A=0, D=1 \Rightarrow H=1 \\ A=0, D=0 \Rightarrow H=0 \end{cases}$