Your grades for **CSCI561-Su16-Final**

DEN ISC

Total Score: **93.0**

Summary

Class scores distribution

Total (/score/1e7164cf-379d-4c65-8e10-1114686793a1)

Q1 (/score/1e7164cf-379d-4c65-8e10-1114686793a1/Q1)

Q2 (/score/1e7164cf-379d-4c65-8e10-1114686793a1/Q2)

Q3 (/score/1e7164cf-379d-4c65-8e10-1114686793a1/Q3)

Q4 (/score/1e7164cf-379d-4c65-8e10-1114686793a1/Q4)

Q5 (/score/1e7164cf-379d-4c65-8e10-1114686793a1/Q5)

Q6 (/score/1e7164cf-379d-4c65-8e10-1114686793a1/Q6)

Q7 (/score/1e7164cf-379d-4c65-8e10-1114686793a1/Q7)

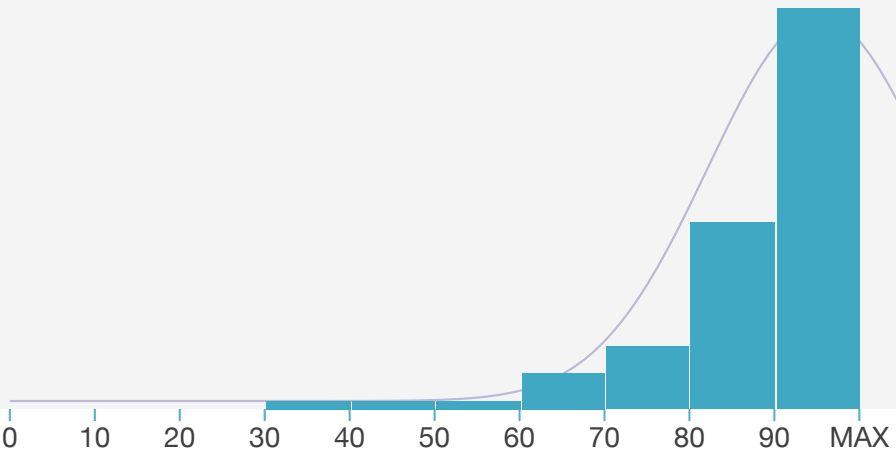
Q8 (/score/1e7164cf-379d-4c65-8e10-1114686793a1/Q8)

Q9 (/score/1e7164cf-379d-4c65-8e10-1114686793a1/Q9)

Q10 (/score/1e7164cf-379d-4c65-8e10-1114686793a1/Q10)

Q11 (/score/1e7164cf-379d-4c65-8e10-1114686793a1/Q11)

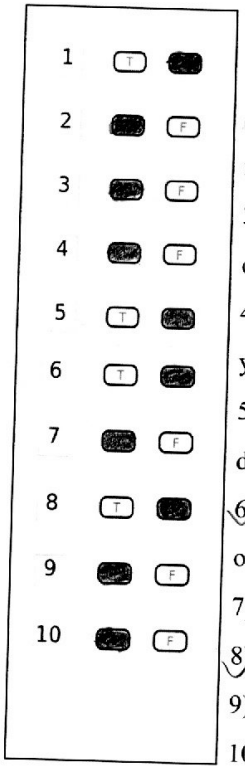
Students: 80 Median: 91 Mean: 86.6 Std. Dev: 11.904



Q1 Score: **10.0**

1. True/False Questions (10 points)

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



1) The exact evaluation function values affect minimax decision even if the ordering of these values is maintained.

2) Soundness of an inference algorithm means that the algorithm doesn't reach bogus conclusions.

3) The number of hidden layers needed in a neural network cannot be calculated from the input provided.

4) Bayes' Theorem takes you from a prior belief to a posterior belief after you've collected some additional data

5) If all cars would be self-driving, the world's roads would be a deterministic environment.

6) The entropy of a binary random variable decreases as data become less ordered.

7) Even a good knowledge base will not be able to answer any question.

8) Perceptrons, arranged with no hidden layer, cannot express logical NOT.

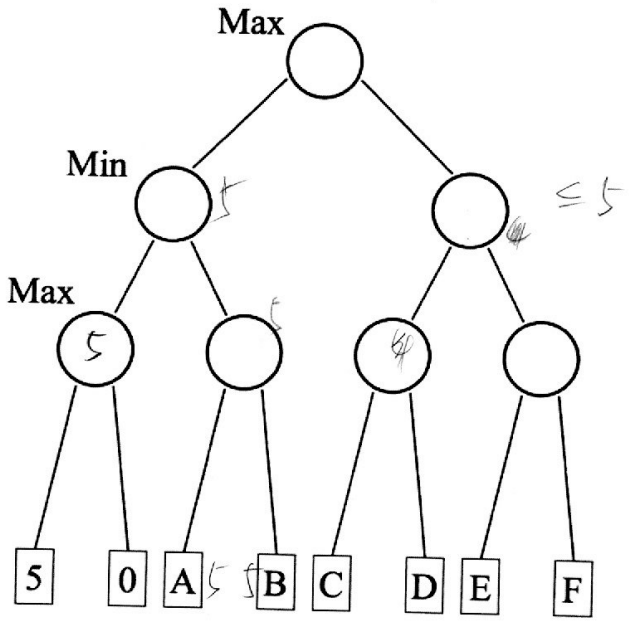
9) SVM can classify data that is not linearly separable.

10) The back-propagation learning algorithm is based on the gradient descent method.

Q2 Score: **8.0**

2. Game Trees (20 points)

Consider the game tree picture below where *A-F* represent some integer values.
Assume the nodes are explored from left to right and standard alpha beta pruning is used.



(a) [4 pts] Give a value of *A* such that *B* is pruned.

As long as $A \geq 5$, for example $A = 8$

(b) [4 pts] Give a value of *A* such that *B* is not pruned.

As long as $A < 5$, for example $A = 2$

Q3 Score: **12.0**

(c) [4 pts] **True of False:** There are SOME values of A and B such that the subtree containing C and D is pruned. Explain Why.

False, Because knowing A,B we know the value of the left node in Min layer, for the right subtree with Min node as root, we must know at least one value in the lower Max level to determine whether to purn. And C,D are in the first subtree, so we need to know their value and get the first Max value, So there is no way to purn C,D.

(d) [4 pts] Assuming that $B = 5$ and $A = 5$, give a value of C and D such that the subtree containing E and F is pruned.

As long as $\max\{C,D\} \leq 5$

for example $C=3$ $D=4$

(e) [4 pts] If you are allowed to assign A-F arbitrarily, what is the MAXIMUM number of leaves that can be pruned? Provide the name of the leaves as well.

The Maximum number of leaves can be pruned is: 3

we can prun: B, E, F

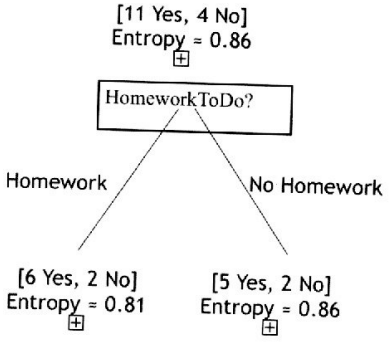
Q4 Score: **0.0**

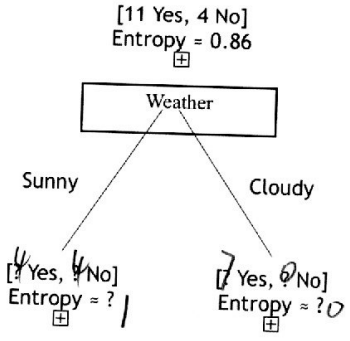
3. Decision Trees (20 points)

In trying to decide whether or not to go jogging, some of the factors I consider are the weather and whether or not I have homework to do. There are other factors as well of course. Some example decisions are summarized in the table below.

Weather	HomeworkToDo	Jogging
Sunny	No Homework	Yes
Sunny	Homework	No
Sunny	No Homework	Yes
Sunny	Homework	No
Sunny	Homework	Yes
Sunny	No Homework	No
Sunny	Homework	Yes
Sunny	No Homework	No
Cloudy	Homework	Yes
Cloudy	Homework	Yes
Cloudy	Homework	Yes
Cloudy	No Homework	Yes
Cloudy	Homework	Yes
Cloudy	No Homework	Yes
Cloudy	No Homework	Yes

With 11 positive examples of jogging and 4 negatives, the entropy or information of this decision in bits is 0.86 Consider the following decision trees, reflecting splitting on the attributes of **Weather** or **HomeworkToDo**.





Q5 Score: **16.0**

A. [8 pts] The HomeworkToDo tree has been filled out. Complete the values for the Weather tree, including entropy. (Show your work)

When Whether is Sunny, there are: 4 Yes, 4 No
its entropy is: $I(\frac{4}{8}, \frac{4}{8}) = -\frac{4}{8}\log_2\frac{4}{8} - \frac{4}{8}\log_2\frac{4}{8}$
$= -\frac{1}{2}(-1) - \frac{1}{2}(-1) = 1$

When whether is cloudy, there are: 7 Yes, 0 No
its entropy is: $I(1, 0) = -\frac{7}{7}\log_2\frac{7}{7} - \frac{0}{7}\log_2\frac{0}{7}$
$= 0$

8

B. [8 pts] Calculate the information gain IG from splitting on HomeworkToDo and Weather. Please show formulas used and steps clearly. You do not need to compute a final numerical result. Plug in values to the formulas so that the result could be calculated with a calculator.

$= I(\frac{11}{15}, \frac{4}{15}) - \text{Remainder(HomeworkTodo)}$

IG(HomeworkToDo) = $I(\frac{11}{15}, \frac{4}{15}) - [P(\text{Homework}) \times I(\frac{6}{8}, \frac{2}{8})$
$+ P(\text{No Homework}) \times I(\frac{5}{7}, \frac{2}{7})] = I(\frac{11}{15}, \frac{4}{15}) - [\frac{8}{15} \times I(\frac{6}{8}, \frac{2}{8}) + \frac{7}{15} \times I(\frac{5}{7}, \frac{2}{7})]$
$= 0.86 - \frac{8}{15} \times 0.81 - \frac{7}{15} \times 0.86 = 0.0267$

IG(Weather) = $I(\frac{11}{15}, \frac{4}{15}) - \text{Remainder(Weather)}$
$= I(\frac{11}{15}, \frac{4}{15}) - [P(\text{Sunny}) \times I(\frac{4}{8}, \frac{4}{8}) + P(\text{cloudy}) \times I(1, 0)]$
$= 0.86 - \frac{8}{15} \times 1 - \frac{8}{15} \times 0 = 0.327$

8

Q6 Score: **4.0**

C. [4 pts] Which of the two attributes is a better choice in constructing a decision tree? Why?

Since IG(weather) > IG(Homework ToDo)
and we should choose attributes with Highest IG to split
So we should choose weather to split first in decision tree

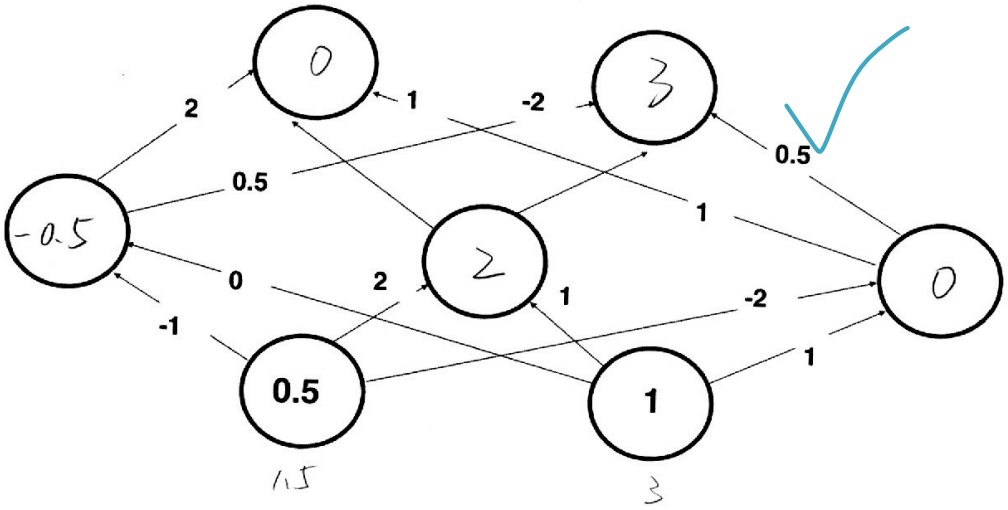
4

Q7 Score: **20.0**

4. Neural Networks (25 points)

The following is a network of **linear** neurons, that is, neurons whose output is identical to their net input. The numbers in the circles indicate the output of a neuron, and the numbers at connections indicate the value of the corresponding weight. (Check the directions of the arrows on the connections while answering the questions).

a) [10 pts] Compute the output of the hidden-layer and output-layer neurons for the given input and enter those values into the corresponding circles.



b) [10 pts] What is the output of the network for the input (1.5,3) i.e. the left input neuron having the value 1.5 and the right one having the value 3? Fill in values for all the circles.

For the hidden layer:
left circle = $-1\times1.5+0\times3=-1.5$, middle circle = $2\times1.5+1\times3=6$
right circle = $-2\times1.5+1\times3=0$ So the values for the three hidden variables from left to right are: -1.5, 6, 0
For the output:
left output = $2\times(-1.5)+0.5\times6+1\times0=0$
right output = $-2\times(-1.5)+1\times6+0.5\times0=9$
So output from left to right are: 0, 9

Q8 Score: **5.0**

c) [5 pts] To answer part (b), do you have to do all the network computations once again? Explain why you do or do not have to do this.

No, I don't need

Because in (a) the inputs are (0.5, 1)

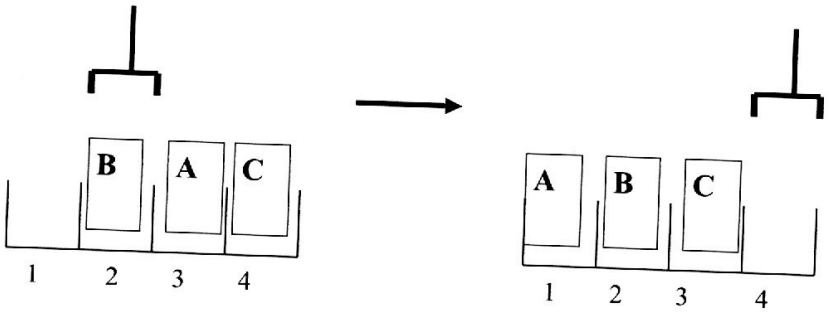
in (b) the inputs are (1.5, 3), both of the two values are just 3 times the three values in (a), and since neural network between two layers $y = w_0 + \sum_{i=1}^{n} w_i x_i$, a linear relationship about x_i, so if all the x_i multiply 3, the result y should multiply 3. So all the values in (b) is just 3 times the values in (a), we just need to $\times 3$, don't need to compute again

✓

Q9 Score: **5.0**

5. Planning (15 points)

Consider writing a planner that will solve the following block-moving problem:
6.1 [10pts] Consider the given domain, and formalize the following sentences:



There are three operators available:

- **pickup (obj1,pos1)**
  Pick up the object **obj1**, which is currently at position **pos1**. The gripper must also be at **pos1**.
- **putdown (obj2,pos2)**
  Put down the object **?obj2**, which is currently being held, at position **pos2**. Gripper must be at **pos2** as well.
- **move-gripper (pos1,pos2)**
  Move the gripper from position **pos1** to position **pos2**.

The gripper can only be holding one thing at a time, and there can be at most one block at a position at a time.

a) [5 pts] Write the operator definitions for **pickup**, **putdown**, and **move-gripper**.

pickup (obj1, pos1):
precondition: obj1 at pos1 ∧ obj1 on table ∧ Gripper at pos1 ∧ Gripper is Empty
effect: add: obj1 at pos1 ∧ obj1 being held ∧ Gripper at pos1 ∧ Gripper hold something
delete: obj1 on table ∧ Gripper is Empty
Putdown (obj2, pos2):
precondition: obj2 at pos2 ∧ obj2 being held ∧ Gripper at pos2 ∧ Gripper hold something
effect: add: obj2 at pos2 ∧ obj2 on table ∧ Gripper at pos2 ∧ Gripper is Empty
delete: obj2 being held ∧ Gripper holds something
Move-gripper (pos1, pos2):
precondition: Gripper at pos1 ∧ Gripper holds something
effect: add: Gripper at pos2 ∧ Gripper holds something delete: Gripper at pos1

Q10 Score: **5.0**

b) [5 pts] Demonstrate a regression search for a solution: show the goal and two regressions, one that fails, and one that succeeds.

+5

goal: At(A, pos1) ∧ At(B, pos2) ∧ At(C, pos3) ∧ Ontable(A)
∧ Ontable(B) ∧ Ontable(C) ∧ At(Gripper, pos4) ∧ Empty(Gripper)

Fail regression:

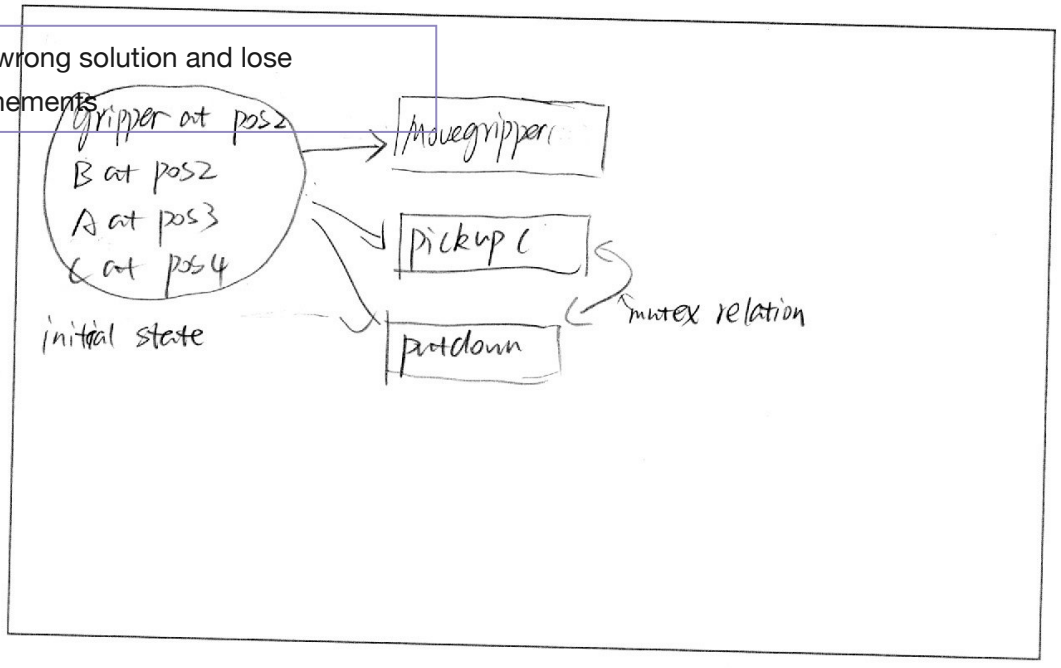
pickup(B,2), movegripper(2,1), putdown(B,1),
movegripper(1,3), pickup(A,3), movegripper(3,2), putdown(A,2)
movegripper(2,4), pickup(C,4), movegripper(4,3), putdown(C,3),
movegripper(3,4)

Succeed Regression:

Movegripper(2,3), pickup(A,3), movegripper(3,1),
putdown(A,1), Movegripper(1,4), pickup(C,4), movegripper(4,3),
putdown(C,3), Movegripper(3,4)

c) [5 pts] Demonstrate a plan-space search for a solution: show the initial plan and three refinements: one that links to the initial state, one that involves adding an operator, and one that involves resolving a threat.

-5, wrong solution and lose refinements



Q11 Score: **8.0**

6. Short Answers (10 points)

1. [2 pts] List 2 reasons why machine learning is needed?

Laziness
Want a simpler environment
Huge amount of Data

1

2. [2 pts] What is Ockham's razor? How is it used in decision tree learning?

The bias term
Help to simplify the calculation

1

3. [6 pts] Choose 2 research topics from the poster sessions and describe each topic, including the limitations/pitfalls and future directions. Compare and contrast how uncertainty was handled.

Microsoft Seeing AI: help visually impaired people to describe surroundings, read texts and identify emotions, answer questions.
its limitation includes still could not do very well in describing images human not just tell us what is in the picture. So future directions include combine machine learning to extract more precise information from image and train it to describe more like human
Facebook Deepface: Identify face from images and identify whether two images show the same people. Limitation includes it still have many requirements about the background of images to make sure the accuracy. Further direction include using deeplearning to improve the accuracy in complex background
The uncertainty handling: Both of them uses Deep Neural Network. Seeing AI uses residual Neural Network and Deepface use a 9-layer Neural Network, to handle the uncertainty in image recognition. They both try to gather more data and use machine learning ~~to~~ Seeing AI also combins image processing and NLP to provide more precise information.

6