6.034 QUIZ 2

November 18, 2002

Name	DAN PORTS
E-mail	drkp@mit.edu

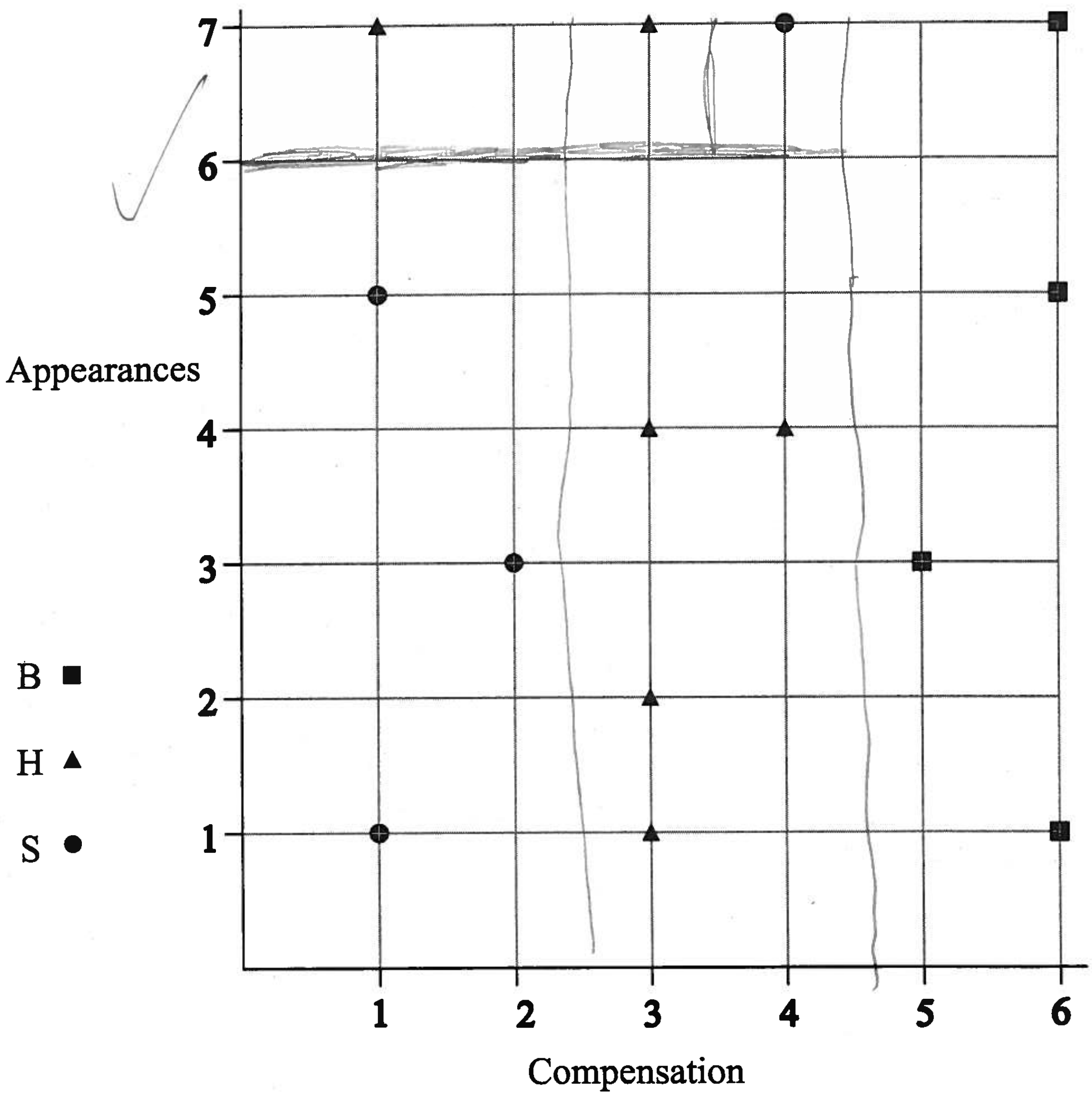
Indicate TA:

Jake Beal	
Stephen Larson	X
Hope Lestier	
Justin Schmidt	
Arian Shahdadi	
Ozlem Uzuner	

Problem Number	Maximum	Score
Problem 1	**35**	35 JB
Problem 2	**35**	34 AS
Problem 3	**16**	16 OU
Problem 4	**14**	14 JB
Total	**100**	99

Problem 1: Identification Trees (35 points)

You have been hired by a wealthy Sloan graduate to manage his stock portfolio. He informs you that he likes to measure stocks by the compensation (K) paid to the company CEO and the number of times the company name has appeared (A) on the front page of the Wall Street Journal during the past month. Then he hands you the following graph, on which he has provided labeled samples for three categories, Buy, Hold, and Sell (B, H, and S).

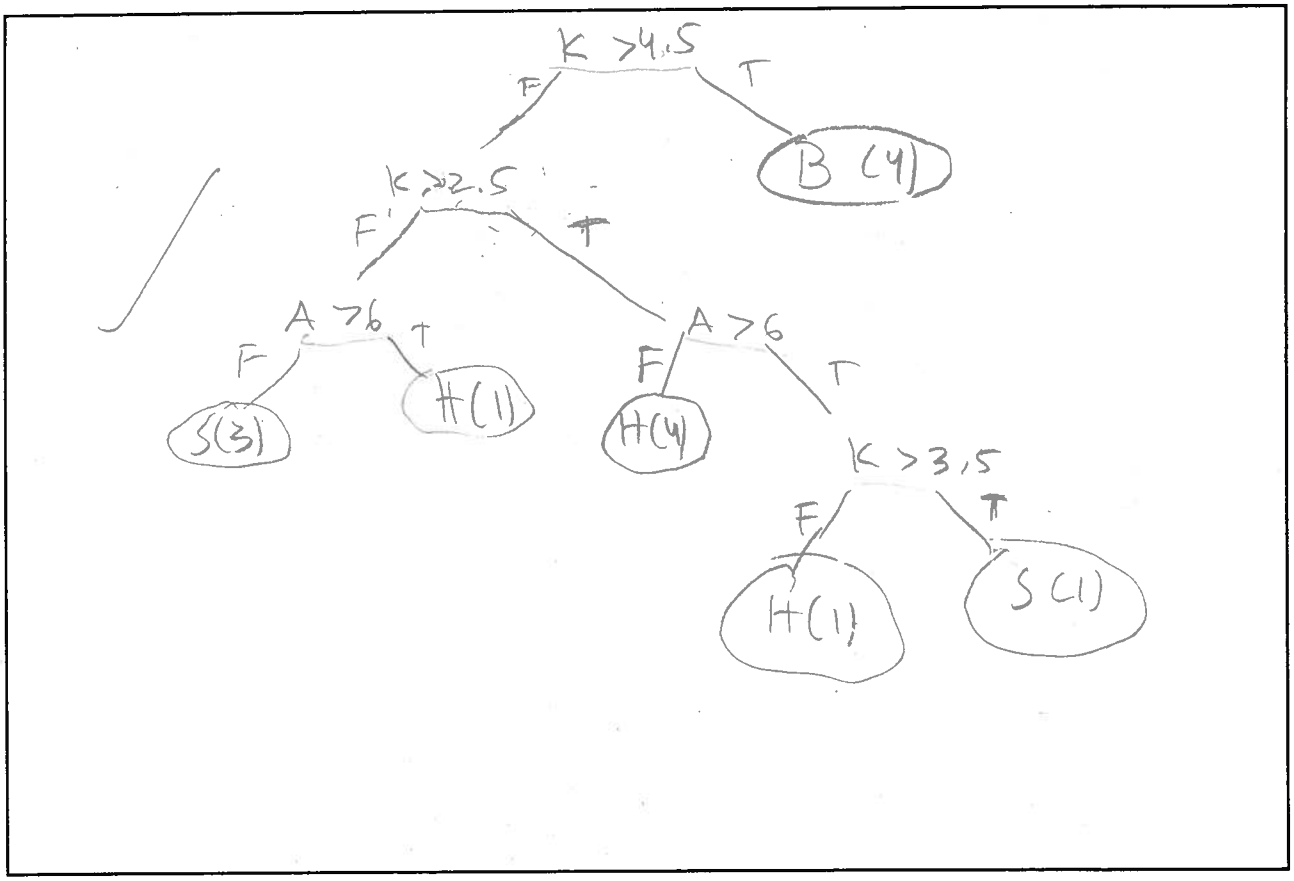


Part A: Partitioning Data (7 points)

You decide to create an identification tree using the training samples. Draw the partitions produced using the training samples on the graph. You should be able to do this without detailed computations. There may or may not be more than one correct answer.

Part B: Tree Construction (8 points)

Draw the identification tree. Include specifications for all tests and the number and type of the samples that end up at each leaf.



Part C: Disorder (7 points)

Calculate the average disorder for the best of the possible first tests on the training data. Express your answer either as a number or in terms of a variable-free algebraic expression.

Test is $k > 4.5$

$$\frac{4}{14}(0) + \frac{10}{14}\left[-\frac{4}{10}\log_2\left(\frac{4}{10}\right) - \frac{6}{10}\log_2(6/10)\right]$$

$$= \frac{10}{14} \cdot .97 \approx \boxed{.693}$$

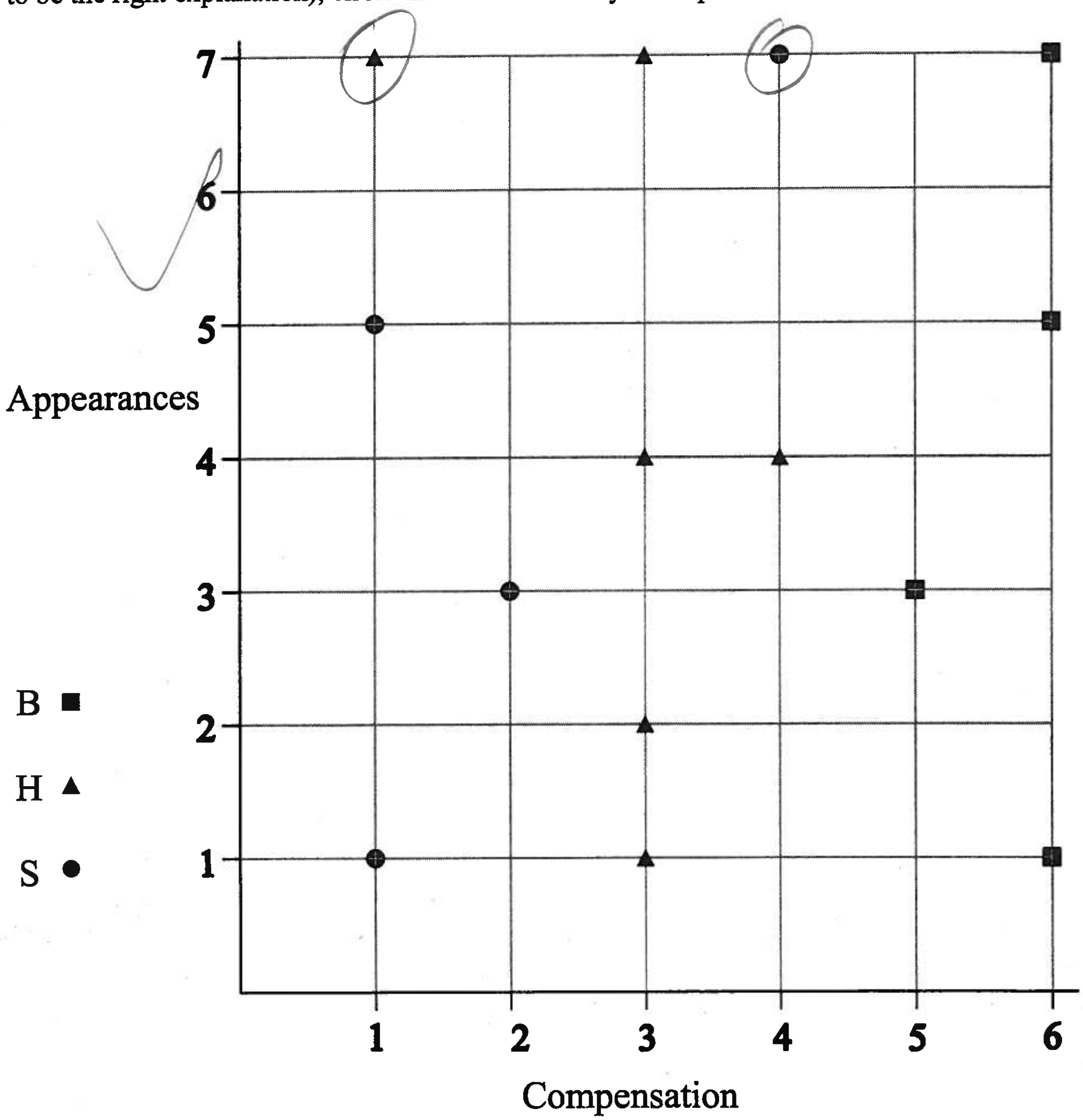
You can use the following table (also on the tear-off page), in case you forgot your calculator. If the number you need is not in the table, use the table to estimate the number you need.

	n	$\log_2 n$	$- n \log_2 n - (1-n)\log_2(1-n)$
	0.00	0.00	0.00
	0.05	-4.32	0.29
	0.10	-3.32	0.47
	0.15	-2.74	0.61
	0.20	-2.32	0.72
	0.25	-2.00	0.81
	0.30	-1.74	0.88
	0.35	-1.51	0.93
	0.40	-1.32	0.97
	0.45	-1.15	0.99
	0.50	-1.00	1.00

.65
.81 .714

Part D: Noise (6 points)

Carefully examine the decision boundaries you drew in Part A. Then, suppose you decide that two noise points have led to overfitting. Then, keeping Occam's razor in mind (the simplest explanation is most likely to be the right explanation), circle the two most likely noise points below:



Part E: Rules (7 Points)

Now, use the identification tree you would get ***with the noise points removed,*** to create three if-then rules for recognizing each type of stock:

R1: IF $K > 4.5$
THEN BUY

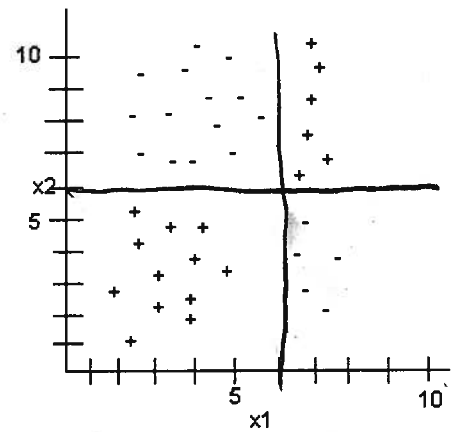
R2: IF $K > 2.5$ AND $K < 4.5$
THEN HOLD

R3: IF $K < 2.5$
THEN SELL

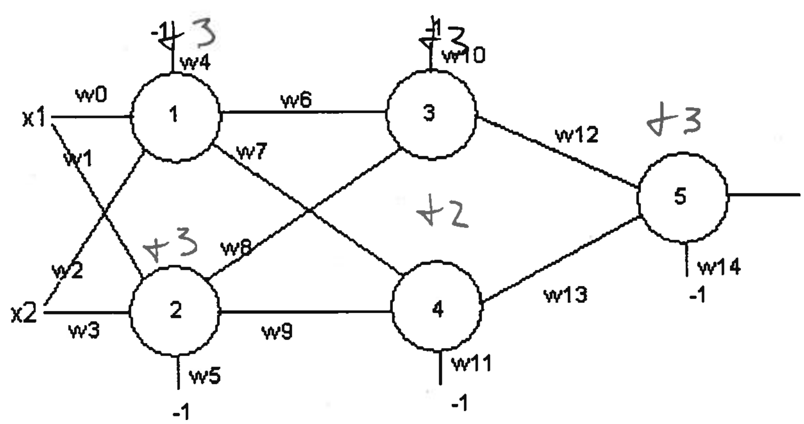
Problem 2: Neural Nets (35 points)

Part A: Decision Boundaries and Perceptrons (17 points) 16

A1. (2 points) You are given a graph of feature vectors (x1, x2). Draw decision boundaries provided by the perceptron network shown in part A2, assuming the net has been trained to properly classify all points.



A2. (15 points) You are given the following **PERCEPTRON** network. Each perceptron, in contrast with the standard version discussed in class, will return a –1 if the total of its weighted input is below threshold, and a +1 if the total of its weighted input is above threshold. Assign the weights w_0 through w_{14} to properly classify the data given in the graph above.

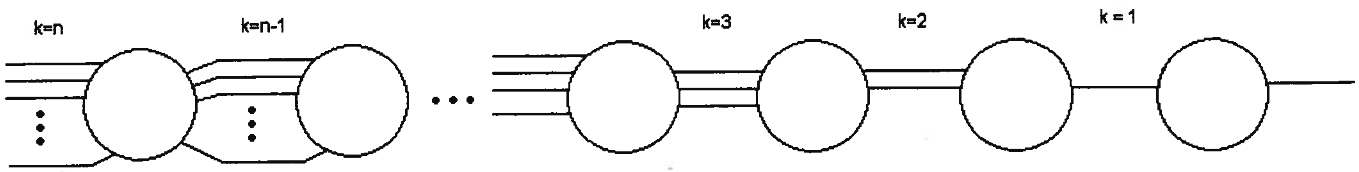


Weight	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	w_{13}	w_{14}
Value	1	0	0	1	6	6	-1	1	-1	1	1/2	1/2	1	1	-1

Part B: Backpropagation (18 points) 18

Graphs of the sigmoid function used by the units have been included, along with backpropagation notes, on the next page.

You are given the following neural network with sigmoid units. There are n units in the network, and each unit has k inputs, with a weight of 1/k on each input. k ranges from n to 1. The first unit on the left has n inputs. There are no threshold inputs. The inputs to the network, i.e. to the leftmost neuron, are all 1.



√ **B1. (6 points)** What is the output of the leftmost neuron?

$\frac{1}{1+e^{-1}} = .73$

√ **B2. (6 points)** For large n, what is output of the rightmost neuron to the nearest tenth?

0.659

√ **B3. (6 points)** Write an expression for the values of the input weights, w_k, for the k^{th} unit after backpropagation with a desired value y* of 0.5 and a learning rate of 1. You may write your expression in terms of k, δ_{k-1}, as well as y_{k-1} (the inputs to the k-1th unit) and y_k (the inputs to the k^{th} unit). Do not use δ_k in your expression.

$w_k = w_k - r\delta_k y_{k-1} = w_k - 1 \cdot \frac{ds(y_k)}{dy_k} \sum \delta_{k-1} w_{k \to k-1} \cdot y_{k-1}$

$= 1/k - 1 \cdot y_{k-1}(1-y_{k-1})\,\delta_{k-1}\, y_k$

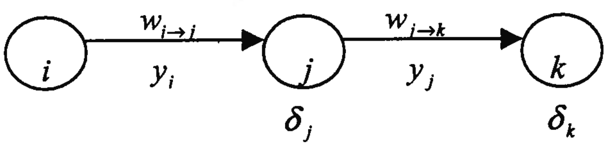
Backpropagation Notes

An efficient method of implementing gradient descent for neural networks

$w_{i\to j} = w_{i\to j} - r\delta_j y_i$ **Descent Rule**

$\delta_j = \dfrac{ds(z_j)}{dz_j}\sum_k \delta_k w_{j\to k}$ **Backprop rule**

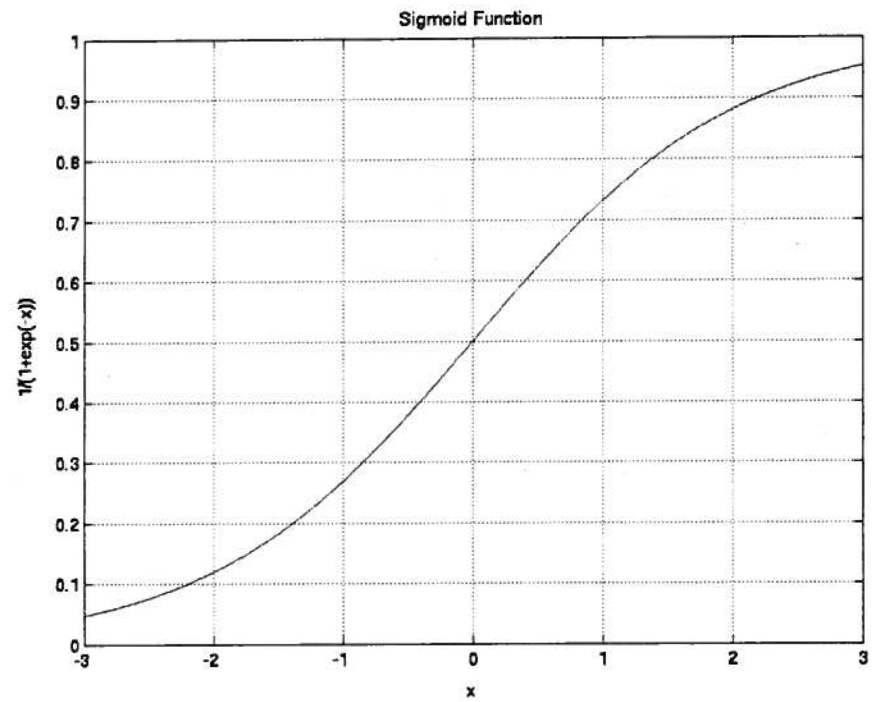
1. Initialize weights to small random values
2. Choose a random sample input feature vector
3. Compute total input (z_j) and output (y_j) for each unit (forward prop)
4. Compute δ_n for output layer $\delta_n = \dfrac{ds(z_n)}{dz_n}(y_n - y_n^*) = y_n(1-y_n)(y_n - y_n^*)$
5. Compute δ_j for preceding layer by backprop rule (repeat for all layers)
6. Compute weight change by descent rule (repeat for all weights)



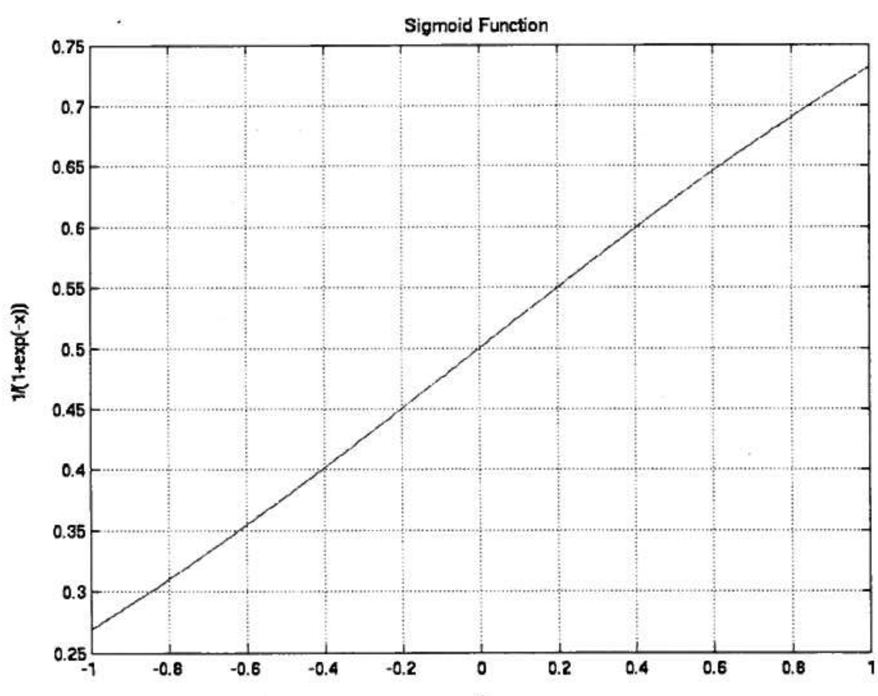
y_i is x_i for input layer

Graphs of sigmoid function are on next page.

Wide-angle view



Zoomed view



Problem 3: Genetic Algorithms (16 points)

Professor C. Ross Ovorr has become sick and tired of creating new final exam questions, so he has decided to try to use a genetic algorithm to generate exam questions from previous exam questions.

First, he decides on a question fitness function, f. Then, he figures out how he can encode questions in "genes" on which his program can perform crossover and mutation operations.

To refine his algorithm, he decides to devote a few weeks of recitation classes to trying out the questions generated by four algorithm variations. Fortunately, the students are half asleep, so they do not remember anything from week to week, so he does not have to worry about memory or any kind of learning mucking up his experiments.

Section R1 receives questions that are generated by crossing the best two questions given to R1 the week before. There is no mutation. The crossover rate is such that there is usually one crossover when two genes are combined.

Section R2 receives questions that are generated by mutating the best two questions given to R2 the week before. There is no crossover. The mutation rate is such that most of the parts of the mutated gene are different from its not-mutated source.

Section R3 receives questions that are generated by crossing and mutating selected questions given to R3 the previous week.

Candidate questions are selected for crossover and mutation using the following survival probabilities:

$P_i = f_i / \Sigma_i f_i$

where f_i is the fitness of the question.

Note that all candidates are considered for each selection, so the same candidate can be selected multiple times.

Section R4 receives questions that are generated in the same way as section R3, except that candidate questions are selected using the following survival probabilities:

$P_i = [f_i^2 + d_i^2]^{1/2} / \Sigma_i [f_i^2 + d_i^2]^{1/2}$

where f_i is the fitness of question *i* and d_i, the diversity of question *i*, is given by:

$d_i = 0$ if question *i* is the most fit question

$d_i = f_h - f_i$ otherwise, where h identifies the question with the ***next higher*** fitness

Thus, for four questions with fitnesses of 3, 0, 3, and 4, d_i would be 1, 3, 1, and 0. (d_1 is 1 because its fitness is 3 and the fitness of the next higher-fitness question is 4, d_2 is 3 because its fitness is 0 and the fitness of the next higher-fitness question is 3, d_3 is 1 by the same argument as for d_1, d_4 is 0 because it is most fit.)

Part A: Performance (4 points)

Which of the four section's methods is **least** likely to produce steady improvement? Circle the best answer:

R1 R2 R3 R4

Part B: Diversity (12 points)

3 0 3 4
1 3 1 0
$\sqrt{10}$ 3 $\sqrt{10}$ 4

B.1 (3 points)

In week 1, the fitness scores of the questions given to R4 were: 3, 0, 3, and 4. Calculate for each question the probability of being a parent for a question in R4 in week 2.

.237, .225, .237, and .300

B.2 (3 points)

Consider the ratio of the probability of the most-likely-to-survive question in a recitation section to the probability of the least likely to survive. Circle the best statement:

- The ratio will likely be greater in R4 than R3.
- The ratio will likely be greater in R3 than R4.
- They will be the same.

B.3 (3 points)

Assume that genetic diversity of the questions is the same in both R3 and R4 in week 1. From week 2 onwards, is the genetic diversity in R3 likely to be less or more than in R4? Circle the best answer:

- R3 is more diverse.
- R4 is more diverse.
- Both are equally likely to be more diverse.

B.4 (3 points)

After many weeks, is the most fit question in the general population of questions likely to be higher in R3 or R4? Circle the best answer:

- Question in R3 is most fit.
- Question in R4 is most fit.
- R3 and R4 are equally likely to produce the most fit question.

Problem 4: Miscellaneous (14 points)

Select the single **BEST** answer for the following questions. No points can be awarded for multiple selections.

1. Alan Turing evidently believed that the computations performed by the human brain are:
    1. Computable by a universal Turing machine.
    2. Computable by a universal Turing machine with 10 fundamental exceptions.
    3. Beyond the reach of all conceivable Turing machines.
    4. Fundamentally different from the computations performed by other primates.
    5. None of the above.

2. Alan Turing argued that:
    1. Computers can be intelligent because they will eventually pass the Turing test.
    2. Computers can be intelligent by refuting contrary arguments.
    3. Computers cannot be truly intelligent because they cannot have free will.
    4. Computers cannot be truly intelligent because they do not have bodies.
    5. None of the above.

3. According to Minsky, frames are suited for:
    1. Representing situations.
    2. Representing visual scenes.
    3. Dealing with multiple perspectives.
    4. All of the above.
    5. None of the above.

4. According to Minsky, continuity in visual experience is a consequence of:
    1. The use of default values to fill frame slots.
    2. Terminals shared by multiple viewpoints.
    3. Holographic frame representation.
    4. Inheritance via multiple parents.
    5. Multiple argument function dispatch.

5. Transition-spaces answer questions about:
    1. Path properties.
    2. Changes in state.
    3. Relational data bases.
    4. Sentence structure.
    5. All of the above.
    6. None of the above.

6. The Rete algorithm:
   1. Makes forward chaining inefficient.
   2. Decreases the efficiency of backward chaining.
   3. Relies only on relational SELECT operations.
   4. Gains speed by using incremental JOIN operations on new assertions.
   5. Places rule antecedents at Beta nodes.

7. Semantic Transition Trees:
   1. Have no nodes with two inputs.
   2. Have links labeled with domain-specific content.
   3. Have links labeled with specific words.
   4. Use recursion.
   5. All of the above.
   6. None of the above.