6.034 Quiz 3
November 9, 2009

Name	
EMail	

Circle your TA and recitation time, if any, so that we can more easily enter your score in our records and return your quiz to you promptly.

TAs
Erica Cooper
Matthew Peairs
Charles Watts
Mark Seifter
Yuan Shen
Jeremy Smith
Olga Wichrowska

Thu	
Time	Instructor
11-12	Gregory Marton
12-1	Gregory Marton
1-2	Bob Berwick
2-3	Bob Berwick
3-4	Bob Berwick

Fri	
Time	Instructor
1-2	Randall Davis
2-3	Randall Davis
3-4	Randall Davis

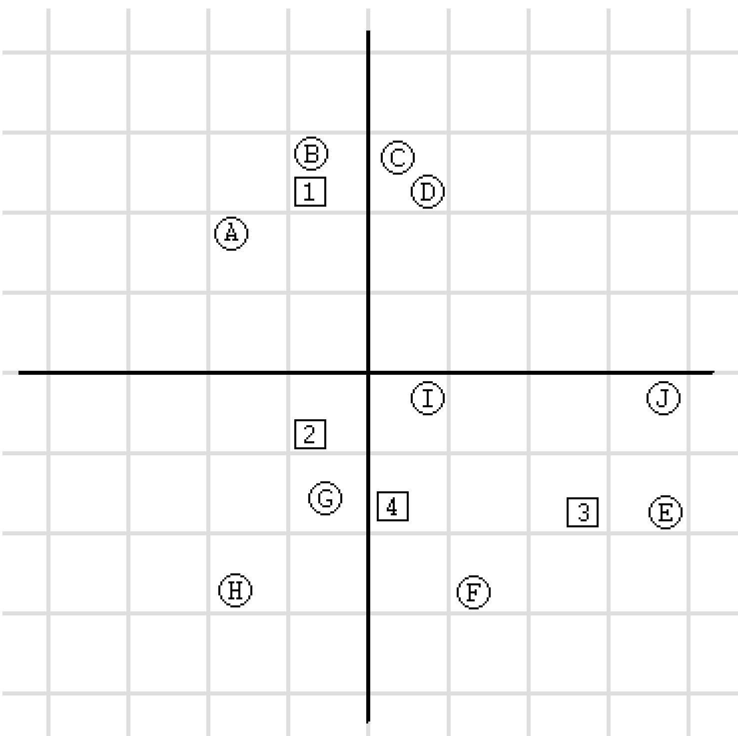
Problem number	Maximum	Score	Grader
1	50		
2	50		
Total	100		

There are ?? pages in this quiz, including this one. In addition, tear-off sheets are provided at the end with duplicate drawings and data.

As always, open book, open notes, open just about everything.

Problem 1:KNN & ID Trees (50 points)

Part A: K Nearest Neighbors, backwards (15 pts)
Shaun has been hired by the Joint Intelligence Committee to investigate the recent zombie infection in his hometown. The first thing Shaun needs to do is to make sense of the incomplete data the JIC has provided him. In the graph below, the circles correspond to observed people, but their labels, "zombie" or "healthy", were lost during the initial investigation. The square points represent people who still need to be classified (they are not themselves used to classify any other points).



Shaun is also given the table below, showing how the square points would have been classified using 1- and 3-nearest neighbors before the labels were lost. Given the map and the table below, Shaun needs to recover the original labels.

Square point	Using 1-nearest-neighbors	Using 3-nearest-neighbors
1	zombie	zombie
2	healthy	zombie
3	healthy	healthy
4	?	?

A1: Write down whether the following specimens are zombies (Z), healthy (H), or if it's unknown (U).
Circle A: ________

Circle B: ________

Circle C: ________

Circle D: ________

Circle E: ________

Circle F: ________

Circle G: ________

Circle H: ________

Circle I: ________

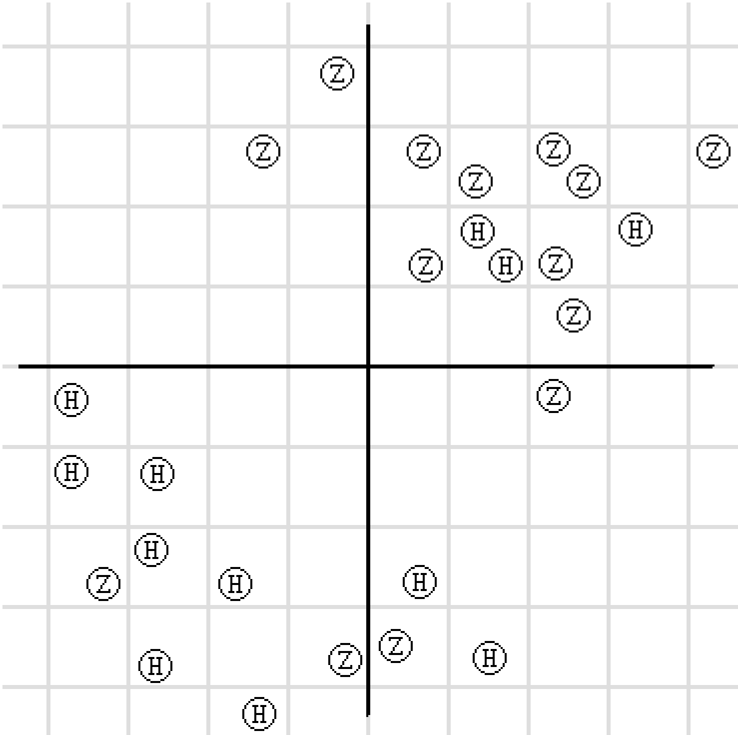
Circle J: ________

A2: How would point 4 be classified? (Again, choose Z, H, or U)

Using 1-nearest neighbor: ________

Using 3-nearest neighbors: ________

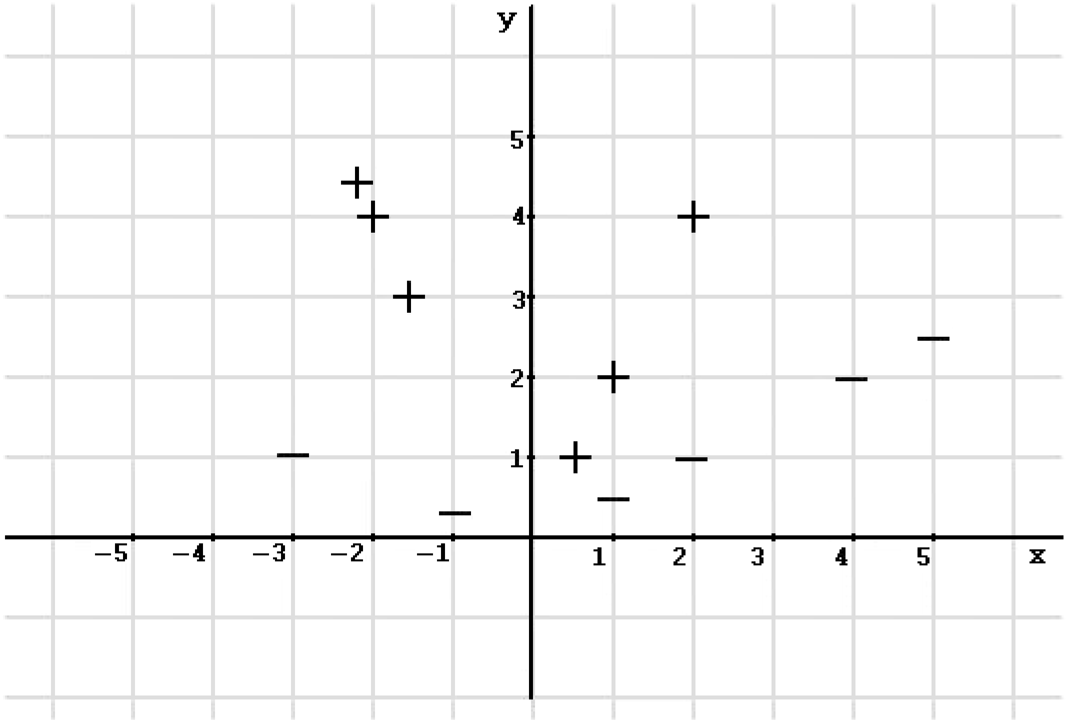
A3: Shaun is wondering whether this k-nearest-neighbor algorithm is really reliable. He decides to check it on some labeled zombie data from a neighboring town. In the graph below, zombies are labeled Z, and healthy people are labeled H.



Describe, in a sentence or two, what happens to the accuracy of k-nearest neighbors as k increases from 1 to 26 (the total number of samples).

Part B: ID Trees (35 pts)

Shaun quickly realizes that he will not be able to recover all the zombie infection information from the data he is given. Fortunately, Shaun's best friend Ed, who was in the middle of a reconnaissance mission in the town, managed to send in a bit more data before he was bitten. Shaun overlays the locations of the known zombies (marked with a +) and known healthy people (marked with a -) on a grid representing the town. (The zombies are currently not biting anyone, so you can trust the points not to change over time.) The JIC has tasked Shaun with figuring out where to build a series of walls separating the healthy people from the zombies. The walls will be built along the decision boundaries created by the identification tree algorithm.



B1. (13 pts)

Shaun's girlfriend Liz suggests building a wall at **y=1.5**. Compute the disorder at this decision boundary. Leave your answer only in terms of integers, fractions, arithmetic operations, and logarithms.

Shaun's flatmate Pete loudly insists that, instead, a wall should be built at **x=-4**. Compute the disorder at this boundary.

Whose idea is better, according to the heuristic described in class? (circle one)

Liz's Pete's

B2. (12 pts)
On the diagram above, draw the decision boundaries Shaun would produce using the identification tree algorithm. In case two decision boundaries are equally good, use the horizontal one. If there is still a tie, use the one with the lower-valued threshold. You will not need a calculator to solve this problem.

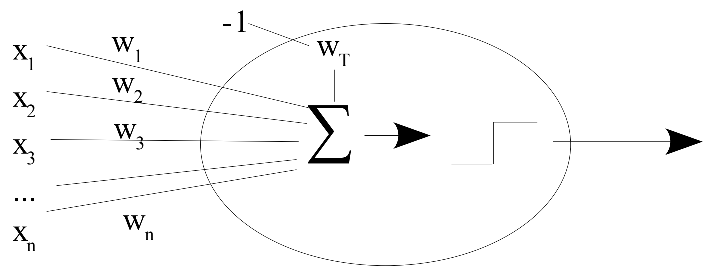
B3. (10 pts)
Shaun realizes that building all these walls is going to take a long time. In order to find out whether he can build a smaller number of walls instead, he decides to convert his data into polar coordinates. **Sketch** the 12 points from the previous graph on the polar graph below (making sure they still show the + and - labels). Show the decision boundaries produced by the identification tree algorithm on the same graph.

Describe in one sentence (or function) how the decision boundaries translate to walls on the original, x-y-plane graph.

Question 2: Neural Networks and Genetic Algorithms (50 points)

Part A (20 points)

Perceptrons are the basic units of neural networks as we have seen them. They take a list of inputs **x**, multiply them by a list of corresponding weights **w**, compute the sum of those products, and pass the result through a decision function. We use a "fake" input T, usually -1, times an associated weight w_T, as part of the sum. When using a single perceptron for classification, one usually uses a threshold decision function: if the sum z > 0, the output is 1, otherwise 0.



To explore what perceptrons can and cannot do, we will ask you to make up weights, rather than training them. Consider the boolean function A→B, and note that it is the same as ¬A∨B.

1. Can a single perceptron with inputs A and B output 1 iff A→B?

If yes give weights: w_A=_______ w_B=_______ w_T=_______

If not, why not?

2. Can a perceptron capture inequalities: given two real-valued inputs A and B, can the perceptron output 1 if A<B and 0 otherwise?

If yes give weights: w_A=_______ w_B=_______ w_T=_______

If not, why not?

3. You wonder about the real-valued function A=B within epsilon E, that is, with three inputs, A, B, and and E, can a perceptron capture whether |A-B| < E?

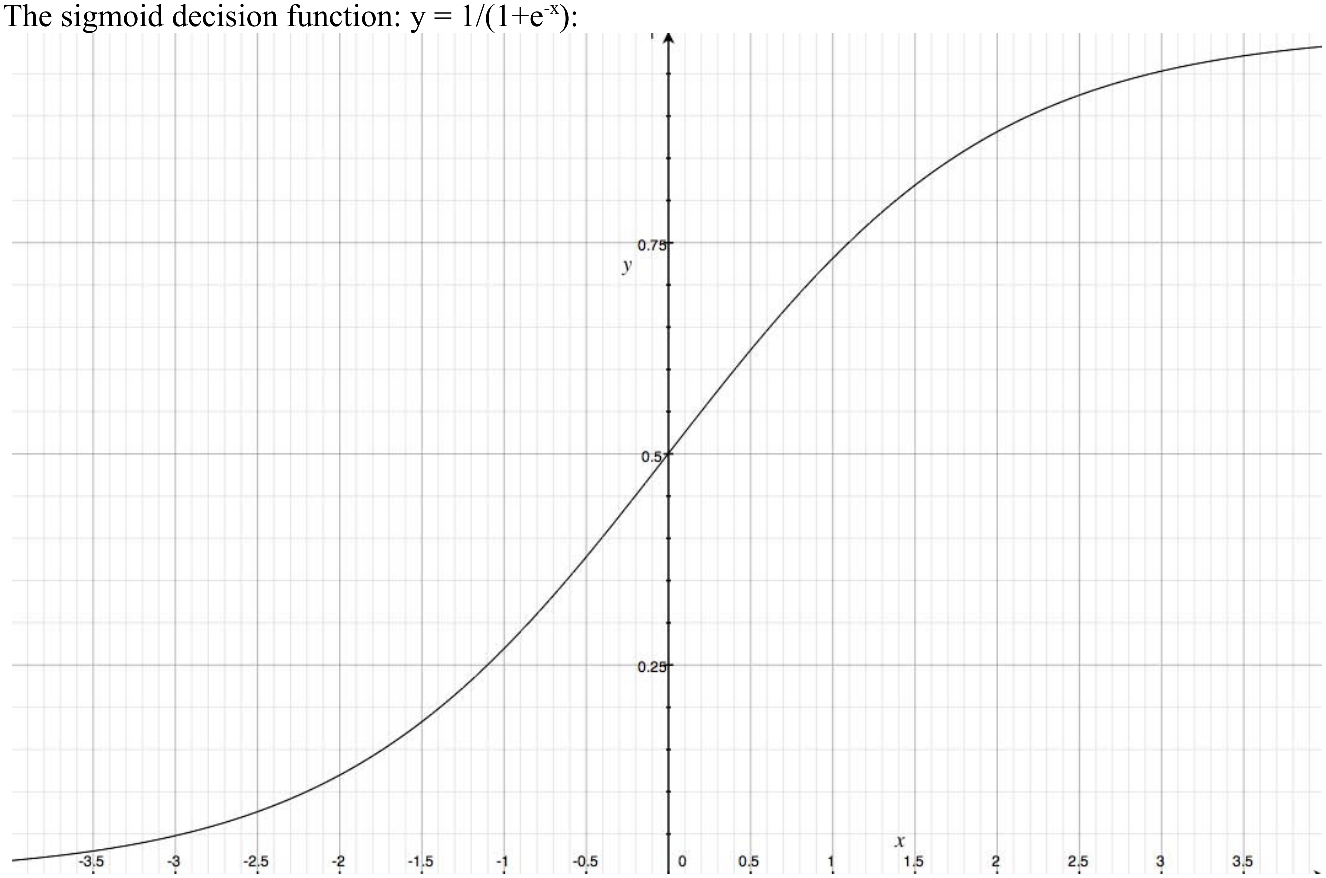
If yes give weights: w_A=_______ w_B=_______ w_E=_______ w_T=_______

If not, why not?

4. The questions in part A have asked for solutions using a threshold decision function. Do the first round of training for A→B with a **sigmoid** decision function. Let the learning rate=1.

A	B	T	w_A	w_B	w_T	Σ	y	y*
0	0	-1	0	0	1			1
0	1	-1						1
1	0	-1						0
1	1	-1						1

The sigmoid decision function: $y = 1/(1+e^{-x})$:



5. Which of the following is true about a perceptron when we use a sigmoid instead of a threshold?

A. The perceptron can learn XOR

B. The perceptron can no longer learn all linear classification boundaries

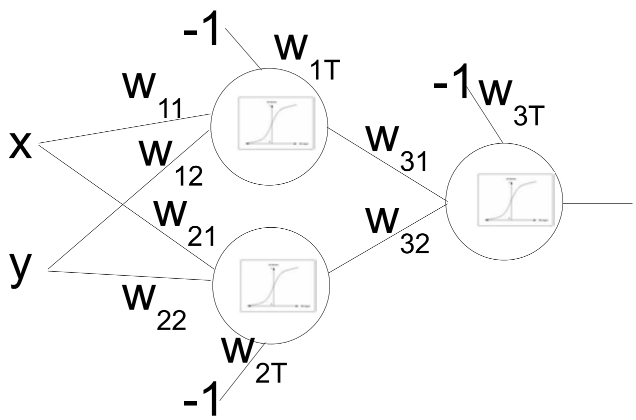
C. The perceptron will learn A→B in fewer training steps

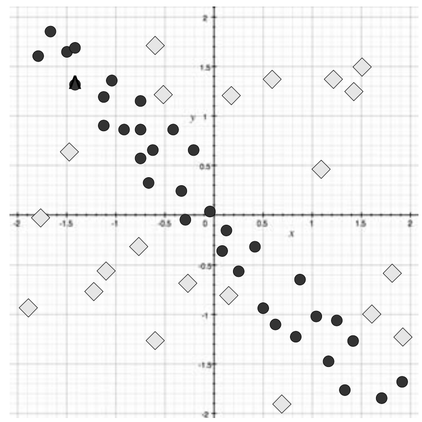
D. The perceptron will learn A→B in more training steps

E. None of these is true

Part B (20 points)

Given this three-node neural network, and the training data on the right





1. Indicate which of the following sets of weights will separate the dots from the diamonds by **circling** their letters. NOTE: more than one may work!

	w11	w12	w1T	w21	w22	w2T	w31	w32	w3T
A	3	2	1	4	5	-2	-100	-100	-150
B	-2	-2	-1	3	3	-1.5	128	128	173
C	1	1	0.5	1	1	-0.5	97	97	128
D	4	4	2	6	6	-3	96	-95	-52
E	2	-2	1	2	-2	1	100	100	50
F	4	4	2	6	6	-3	-101	102	148

2. For (at least) one of the sets of weights you chose above, write the simplest mathematical representation for it using + - * /, inequalities, and/or boolean operations.

Which line?	mathematical expression:

3. When training the three-node neural network at the top of the page using back-propagation, if the current weights are the ones given in choice A, then, for the training example x=0 y=0, if y*=1, what is δ_1? See the tear-off sheet for notes on back-propagation. You can leave your solution expressed as a product, and it may help us assign partial credit.

Part C (10 points)

Your friend is building a device to unlock a door when it hears a secret knock pattern, and realizes that a neural network could do it, if only it had a sense of timing. You suggest feeding the output of one neuron into the input of one of its ancestors, and thereby get a dependence on timing.

1. You think about training the network by standard back-propagation, but decide that you can't. Why?

The solution is clear: Genetic Algorithms! You'll set up a population of identical neural networks with random weights, you discretize your input every 100 milliseconds into a sequence $k_1 \ldots k_n$ of 0 if silence and 1 if a knock was heard, ensuring that k_1 is always 1, and timing out eventually. You'll choose the fittest few neural networks at each step. Your friend jots down a few ideas for fitness functions:

A. Whether the full knock pattern was correctly classified
B. The length of the subsequence $k_1 \ldots k_t$ that is correctly classified
C. The length of the longest subsequence $k_i \ldots k_j$ that is correctly classified
D. The number of k_i correctly classified
E. The number of knock subsequences $k_i \ldots k_j$ that are correctly classified

2. Select all of these fitness functions that one cannot evaluate using the neural net as a black box:

3. Select all of the fitness functions that will immediately trap the genetic algorithm in a fitness plateau:

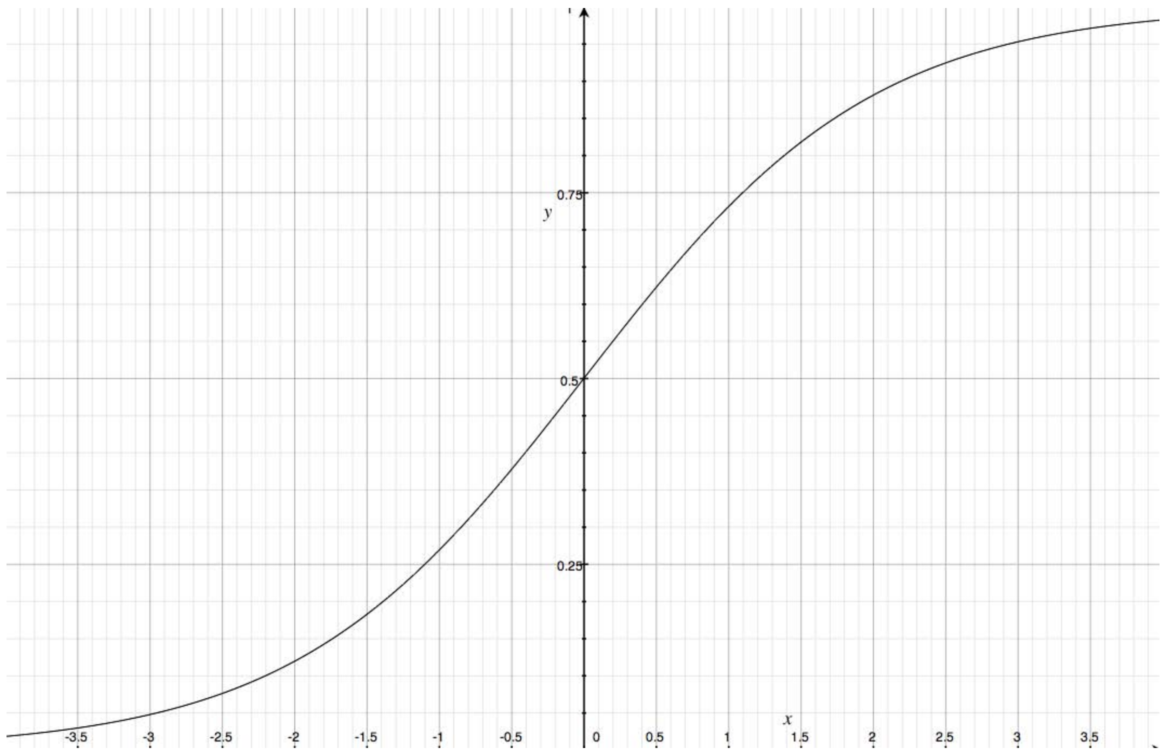
4. Select all of the fitness functions that do not correlate with the actual fitness:

5. Having selected a fitness function, you decide to mutate weights randomly, and choose about half of the weights from each parent for crossover. Your friend uses the GA to train an NN on the example knock sequence, and it consistently says true for that knock sequence. Excited, he installs it, goes outside, waits for it to lock, someone runs by, and the door opens. What was missing from his training data?

6. Having added that, he retrains the system on all the training data, and it's classifying things perfectly, and he goes outside and waits for it to lock, knocks the secret pattern, trying again and again, but you

eventually have to let him back in. What happened?

Tear-off sheet



A	B	T	w_A	w_B	w_T	∑	y	y*
0	0	-1	0	0	1			1
0	1	-1						1
1	0	-1						0
1	1	-1						1

Perceptron update:
if $|y^*-y| > 0$:
    for each w_i:
        $w'_i = w_i + r(y^*-y)x_i$

$$E = \frac{1}{2}\sum_k (o_k - d_k)^2$$

$$w_{i \to j} = w_{i \to j} - \Delta w_{i \to j}$$

$$\Delta w_{i \to j} = R \times o_{l_i} \times \delta_{r_j}$$

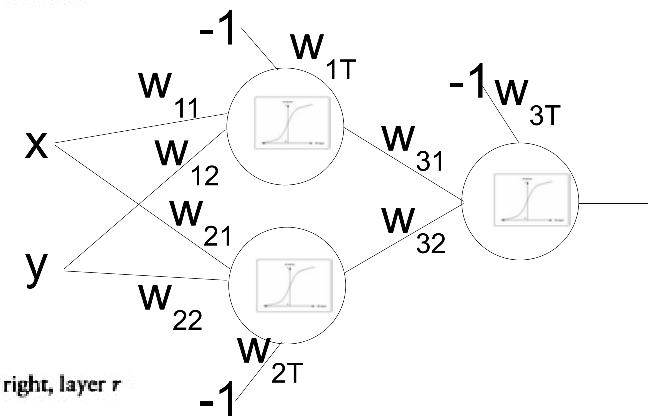
where R is a rate constant and the δs are computed with the following formulas:

$$\delta_k = o_k(1 - o_k) \times (o_k - d_k)$$

$$\delta_{l_i} = o_{l_i}(1 - o_{l_i}) \times \sum_j w_{i \to j} \times \delta_{r_j}$$

where

- o_k is output k of the output layer
- d_k is the desired output k of the output layer
- δ_k is a delta associated with the output layer
- o_{l_i} is output i of left layer in a left-right pair
- δ_{l_i} is a delta associated with the layer l
- δ_{r_j} is a delta associated with the adjacent layer to the right, layer r



For your entertainment after the quiz: http://www.youtube.com/watch?v=zE5PGeh2K9k