6.034 Quiz 3
November 12, 2008

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
Sam Glidden
Mike Klein
Mark Seifter
Theopholis T. Yeh

Thu	
Time	Instructor
1-2	Bob Berwick
2-3	Bob Berwick
3-4	Bob Berwick

Fri	
Time	Instructor
1-2	Howard Shrobe
2-3	Howard Shrobe
3-4	Howard Shrobe

Problem number	Maximum	Score	Grader
1	50		
2	50		
Total	100		

There are 13 pages in this quiz, including this one. In addition, tear-off sheets are provided at the end with duplicate drawings, data, and neural net notes.

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

Problem 1: Nearest Neightbors and ID Trees (50 points)

Taking advantage shifts in MIT's requirements, Johnathan decides to take a geology subject instead of 8.02. He begins by learning how to classify minerals as sedimentary or metamorphic. He decides to take advantage techniques he learned in 6.034.

Johnathan has a data set of eight minerals classified as either sedimentary or metamorphic, with the following characteristics:

Sample #	Mineral	Hardness	Density	Grainsize
1	Sedimentary	3	4	huge
2	Sedimentary	2	3	normal
3	Sedimentary	4	3	huge
4	Sedimentary	4	2	huge
5	Metamorphic	2	5	tiny
6	Metamorphic	3	2	normal
7	Metamorphic	4	5	tiny
8	Metamorphic	5	3	tiny

He wants to classify these two minerals:

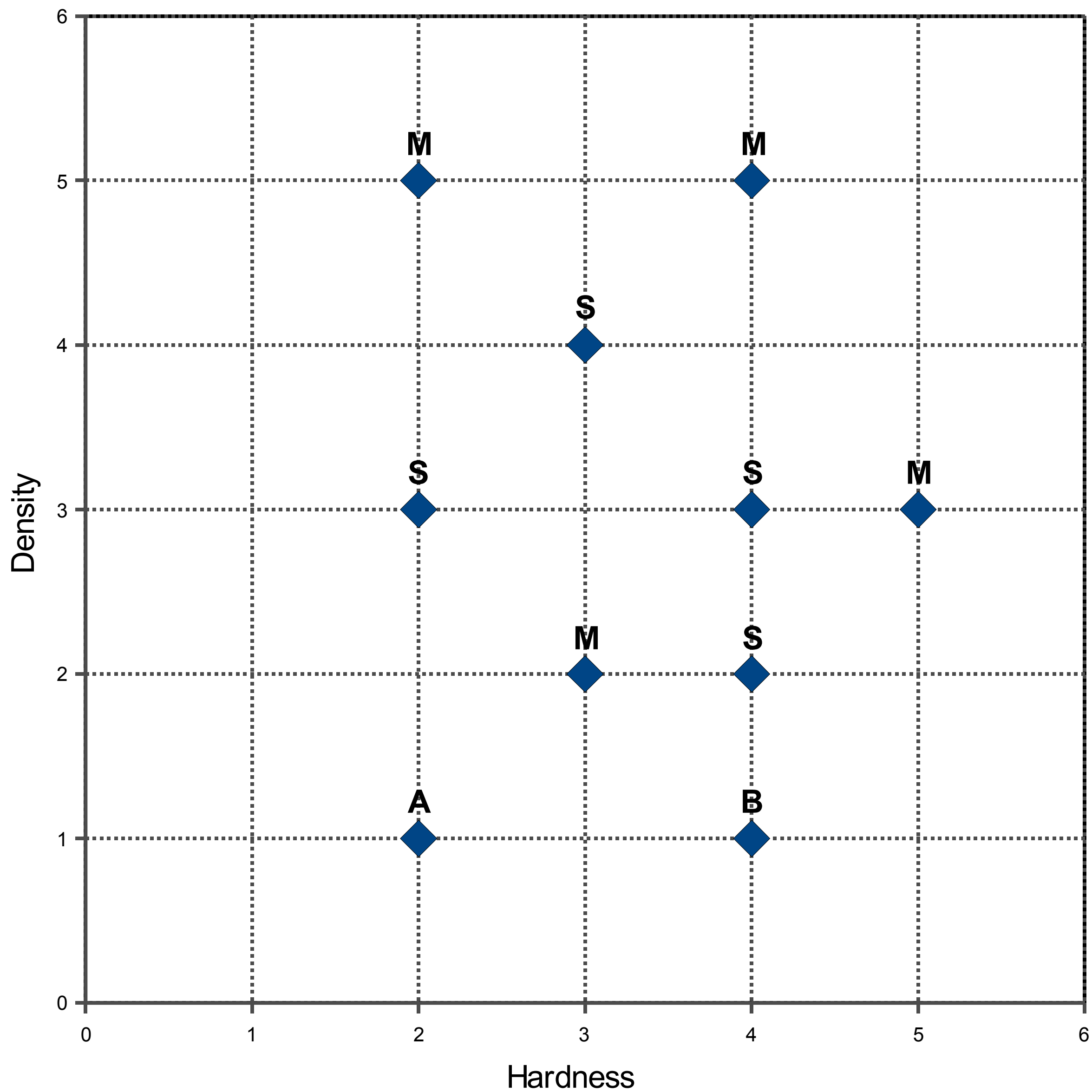
Sample	Mineral	Hardness	Density	Grainsize
A	?	2	1	huge
B	?	4	1	tiny

Part A: Nearest Neighbors (25 points)

Johnathan decides to try using Nearest Neighbors to classify minerals A and B. To make it easier, he only uses the first two characteristics: hardness and density.

Part A1 (17 points)

On the following graph, draw the decision boundaries produced by 1-Nearest Neighbor. Ignore Samples A and B.



Part A2 (8 points)

How is Sample A classified by 1-NN?

By 3-NN?

How is Sample B classified by 1-NN?

By 3-NN?

Part B: Identification Trees (25 points)

Nora is not pleased by the results of Johnathan's work with Nearest Neighbors and so decides to use Identification Trees to classify Samples A and B. She decides to consider all three characteristics; hardness, density, and grainsize.

Part B1 (12 points)

Nora picks tests to use so as to minimize disorder.

For the top of the tree, she considers the following three tests:

TEST 1	Hardness > 3.5
TEST 2	Density > 4.5
TEST 3	Grainsize is huge, normal, or tiny

Which test is best among the three tests listed?

What is the disorder of the test you have selected? You may leave your answer in terms of logarithms.

Part B2 (9 points)

Draw the complete ID-Tree, beginning with the test Nora has selected from the three tests she considers, that will correctly classify each Sample 1-8. You can use the grain-size test or any test of the form Density > a threshold or any test of the form Hardness > a threshold. Select tests that minimize disorder.

Part B3 (4 points)

How does you ID-Tree classify Sample A?

Sample B?

Problem 2: Neural Nets (50 points)

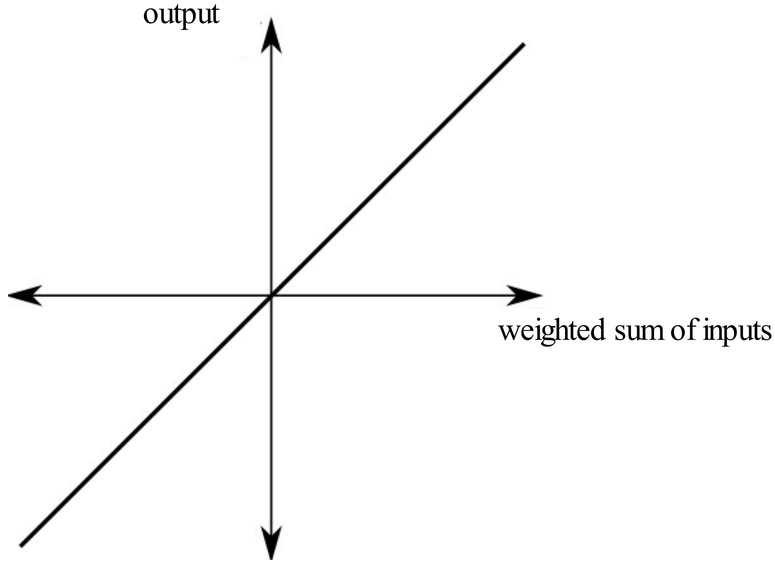
Part A (12 points)

Lyla wants to train a neural net, but she has forgotten what the sigmoid function is. “Oh well,” she says to herself, “it probably wasn't important anyway.”

She sets up a neural net where the nodes are **not sigmoid functions;** they are simply **adders**. Each node simply adds up its weighted inputs and outputs the sum, so

output = weighted sum of inputs

instead of *output*= σ(weighted sum of *inputs*). The derivative of the adder's output with respect to the weighted sum of the inputs is 1 everywhere.

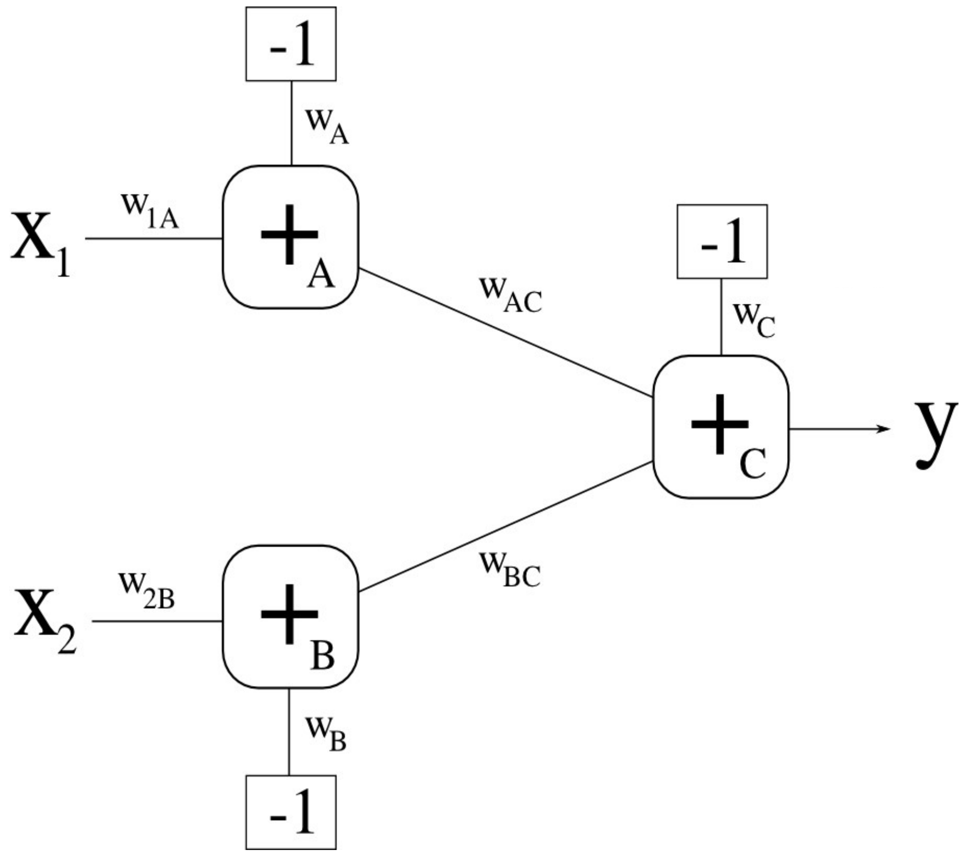


Lyla continues to use the standard performance function $P = -\frac{1}{2}\ (y^* - y)^2$ to train the neural net. The net she is training is shown on the next page and on the tear off sheet.

Part A1 (6 points)

What is the new equation for $\delta_i = \partial P/\partial z$ at an output node, where z is the weighted sum of the inputs to the output node. Your equation should be in terms of y and y^* and zero or more of the input values and weights that appear in the diagram on the next page.

The neural net that Lyla is training looks like this:



Part A2 (6 points)

What is the new equation for δ_A at internal node A? Your equation should be in terms of *y*, y*, and zero or more of the input values and weights that appear in the diagram above. **See the notes at the end of the quiz if you need help wth the definition of δ_A**

Part B (26 points)

Lyla runs **one step** of back-propagation on this same neural net (which is shown again on the next page and on the tear-off sheet) with the following parameters:

- All weights are initially 1, **except that** $w_C = -0.5$
- $x_1 = x_2 = 1$
- $y^* = 1$
- $r = 1$

Part B1 (5 points)

What is the output y of the neural net before back-propagation?

Part B2 (10 points)

Run **one step** of back-propagation (keeping in mind that the nodes are **adders** and not sigmoids). What are the new weights on the edges? For partial credit, you should show your work for each weight, unless that weight is unchanged.

$w_{AC}' =$

$w_{BC}' =$

$w_C' =$

$w_{1A}' =$

$w_{2B}' =$

$w_A' =$

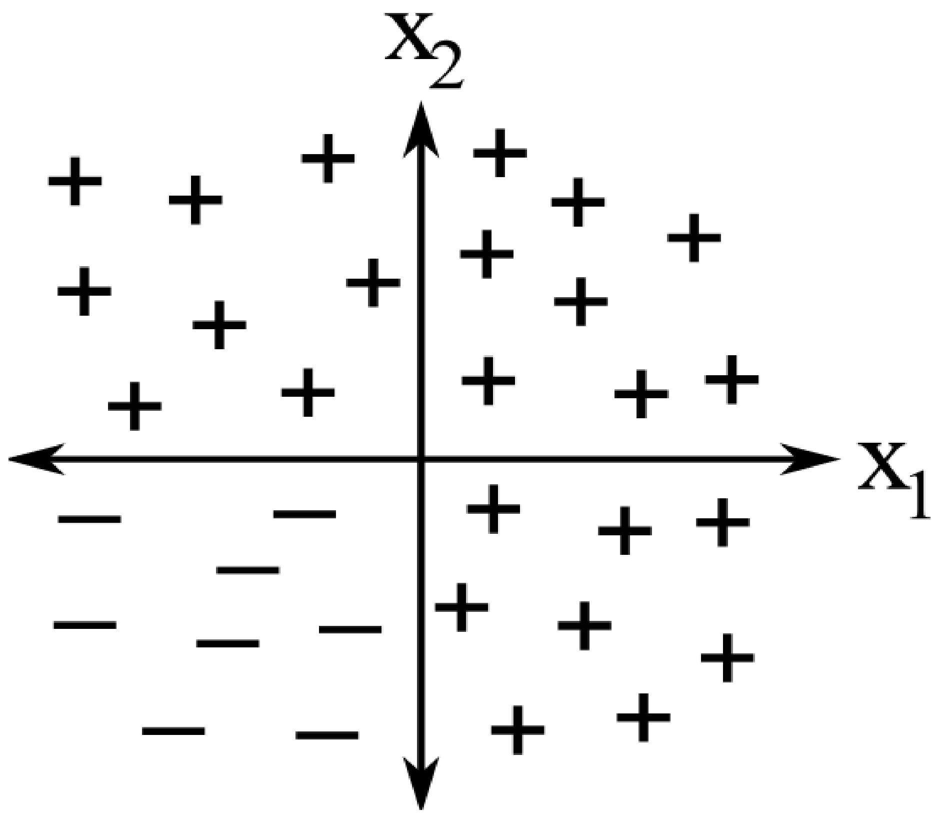
$w_B' =$

Part B3 (5 points)

What is the output *y* of the neural net after one step of back-propagation?

Part B4 (6 points)

Lyla trains her **adder** neural net on data in which all samples occupy a square and the negative samples seem to always occupy the lower left quadrant of the square. The following is representative:



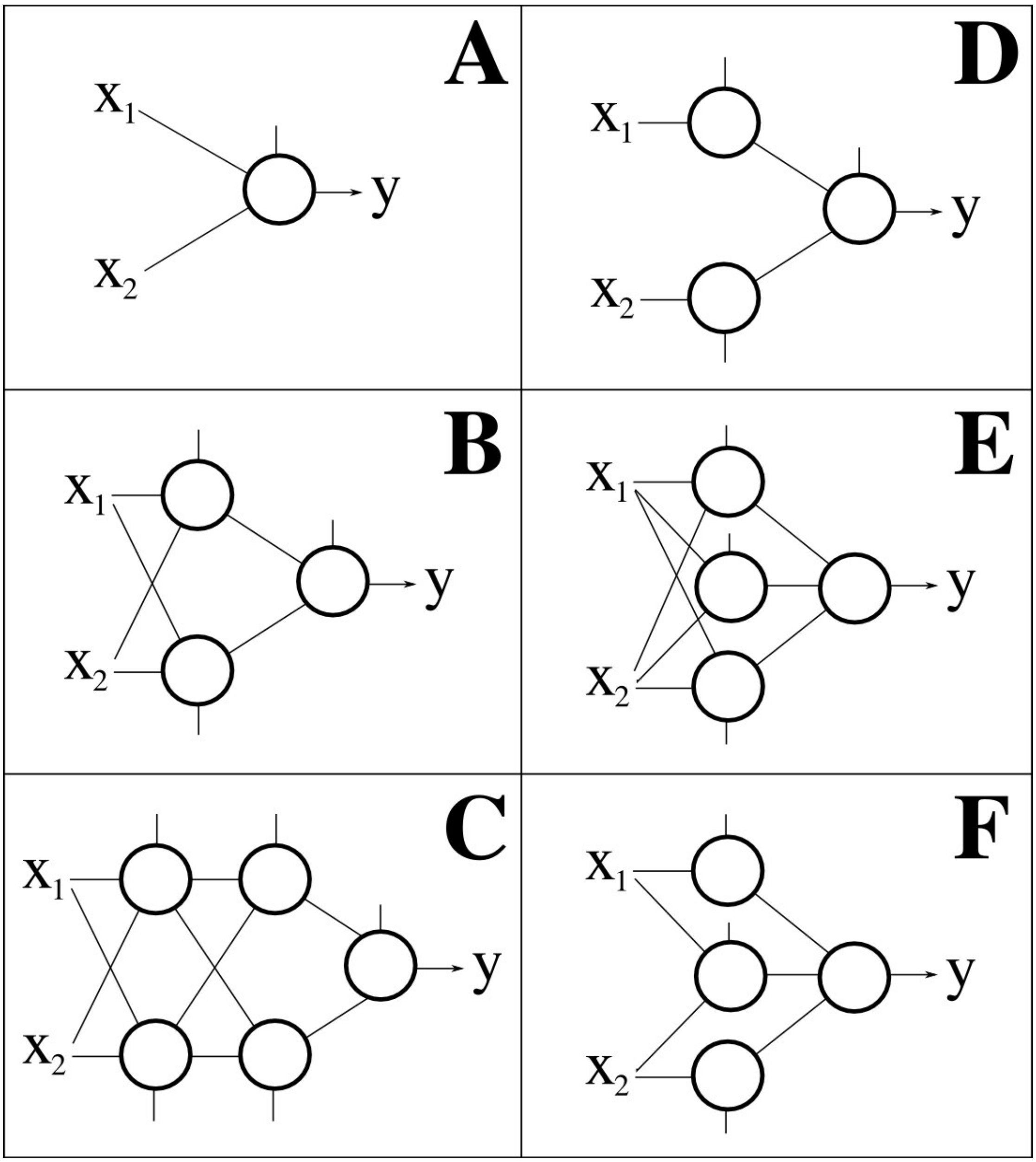
The neural net is supposed to output a positive value for the data points marked +, and a negative value for the data points marked -. Lyla finds that no matter what initial weights she uses, and how and how much she trains the neural net, its error rate on the training data never goes below about 1/8. Why?

Part C (12 points)

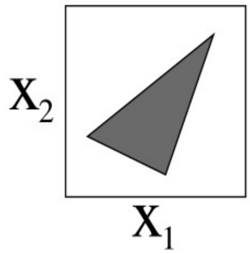
Stephanie has been training some **sigmoid** neural nets, but was so busy celebrating the election results last week that she forgot which one was which.

This page contains diagrams of six neural nets, labeled A through F. The next page shows six classifier functions. Determine which classifier came from which neural net.

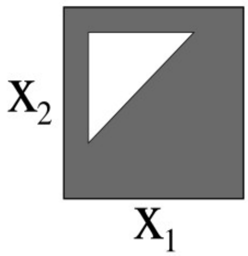
The six answers should all be unique, but you can give reasonable non-unique answers for partial credit.



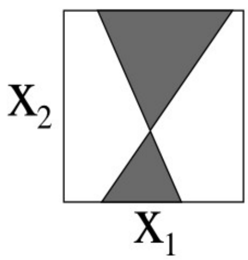
Classifier

1. 

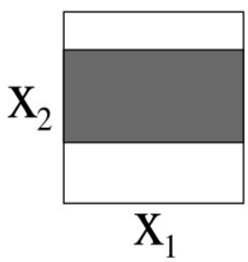
Which neural net? (A-F)

2. 

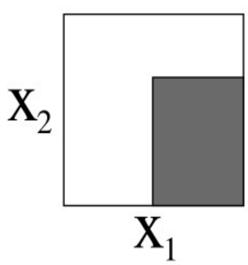
Which neural net? (A-F)

3. 

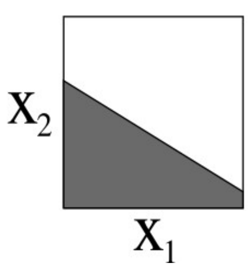
Which neural net? (A-F)

4. 

Which neural net? (A-F)

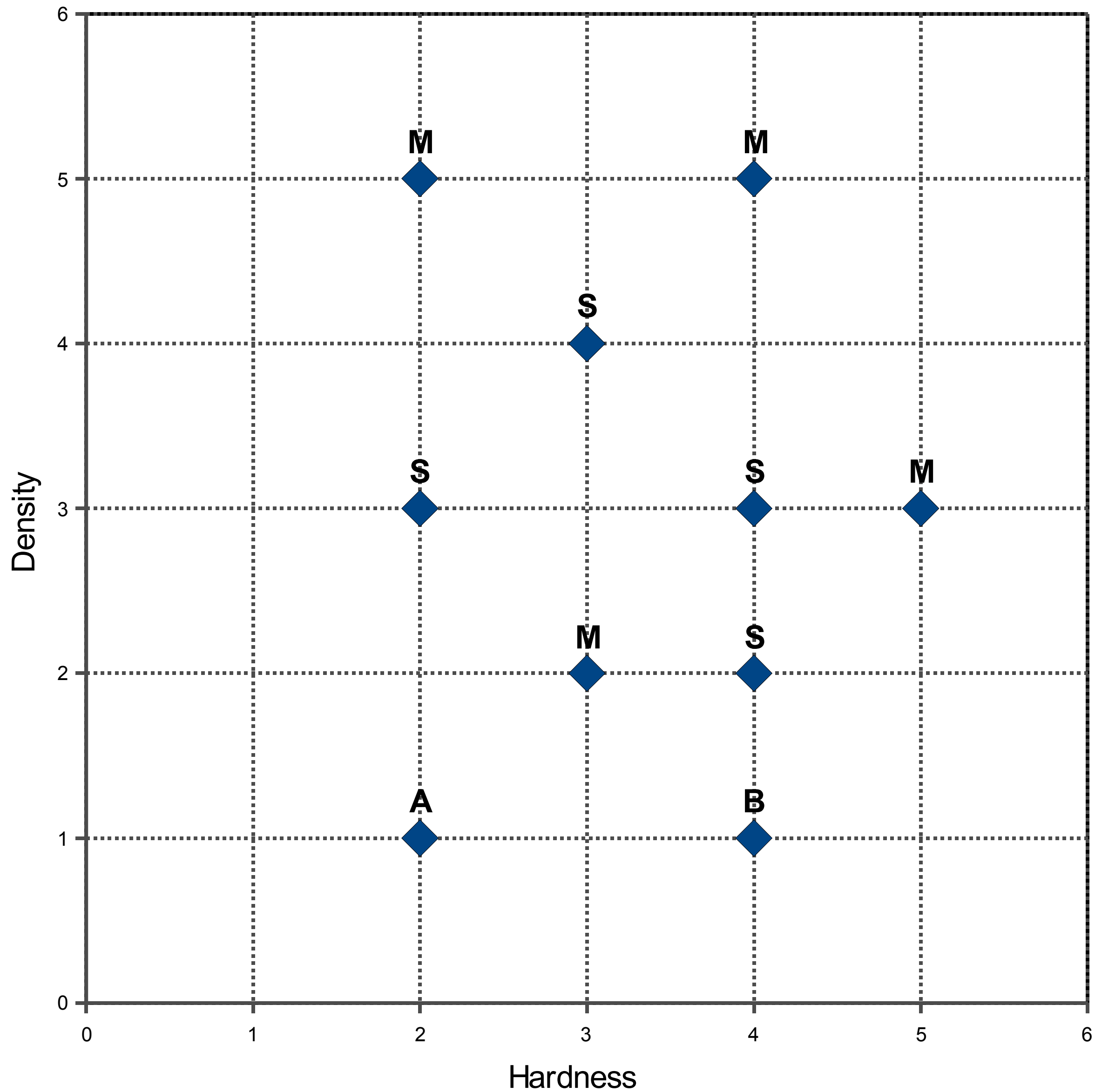
5. 

Which neural net? (A-F)

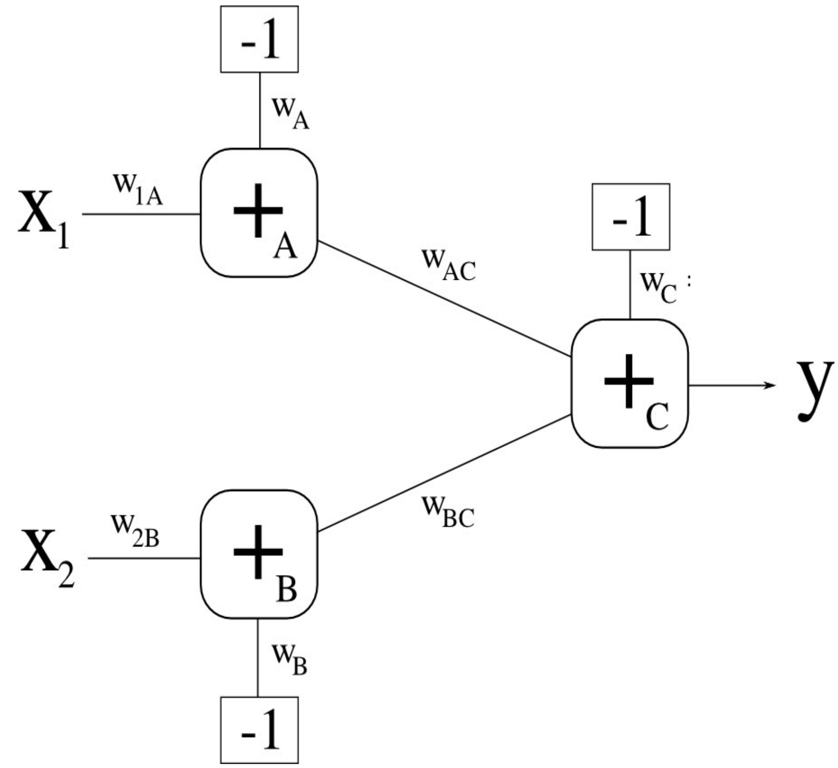
6. 

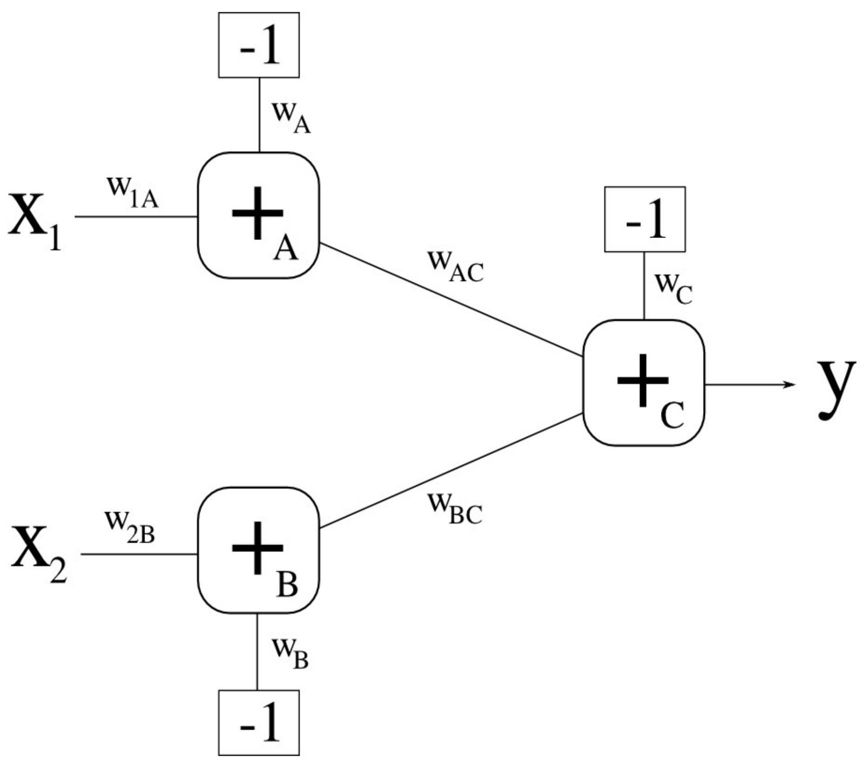
Which neural net? (A-F)

Tear off sheet. You need not hand this in.

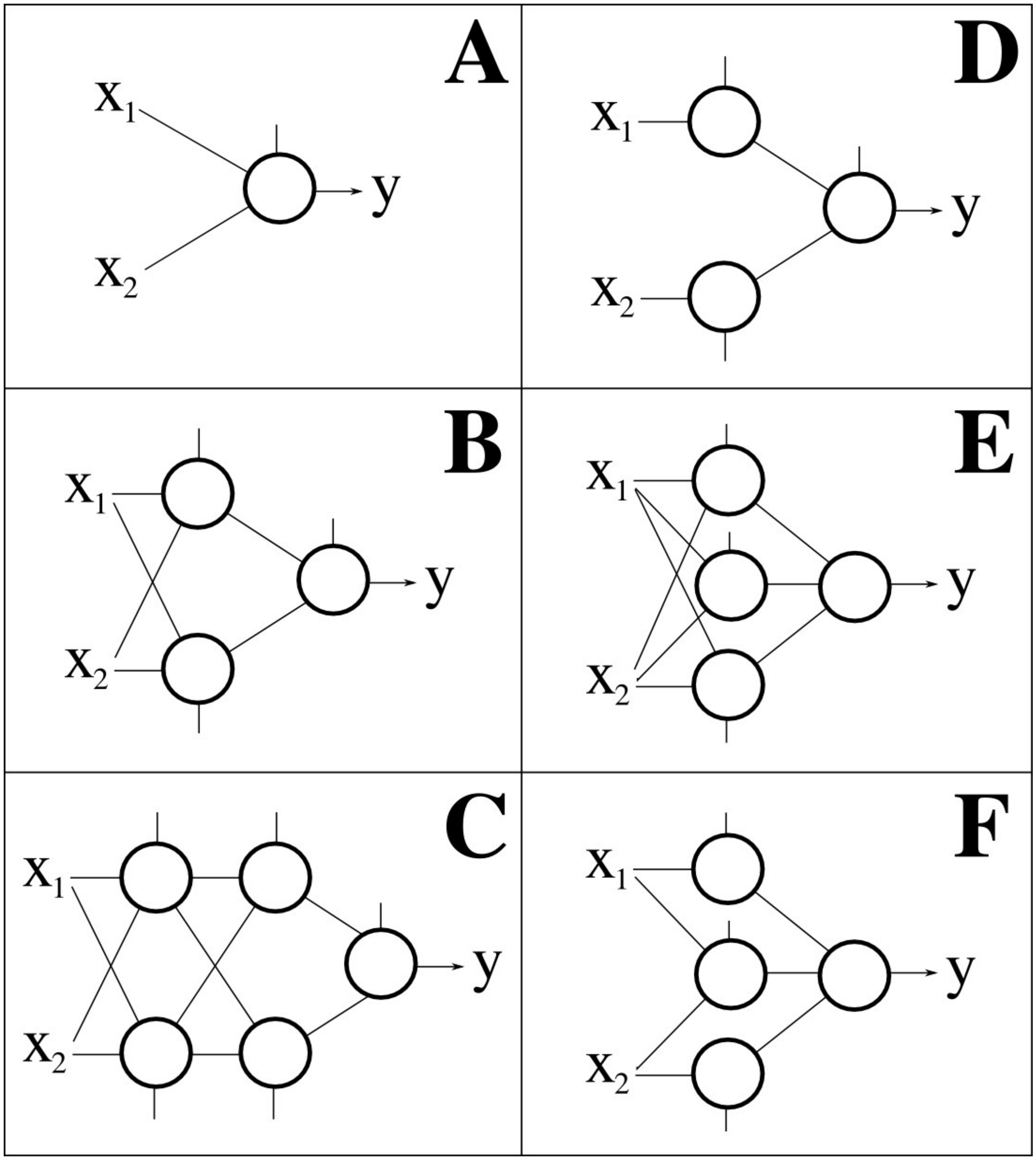


Tear off sheet. You need not hand this in.





Tear off sheet. You need not hand this in.

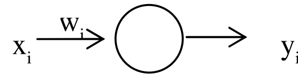


Backpropagation Notes

New weights depend on a learning rate and the derivative of a performance function with respect to weights:

(1) $w_i' = w_i + r\dfrac{\partial P}{\partial w_i}$

Using the chain rule, where y_i designates a neuron's output:



(2) $\dfrac{\partial P}{\partial w_i} = \dfrac{\partial P}{\partial y_i}\dfrac{\partial y_i}{\partial w_i}$

For the standard performance function, where y^* is the final desired output and y is the actual final output, $P = -\frac{1}{2}\sum (y^* - y)^2$:

(3) $\dfrac{\partial P}{\partial y_i} = \dfrac{\partial}{\partial y_i}(-\frac{1}{2}(y^* - y)^2) = (y^* - y)$

For a neural net, the total input to a neuron is $z = \sum w_i x_i$

(Note that x_i is sometimes written y_i to indicate that in a multilayer network, the input for one node is the output for a previous layer node)

For a sigmoid neural net, the output of a neuron, where z is the total input, is $y = s(z) = \dfrac{1}{1+e^{-z}}$.

Recall that the derivative $\dfrac{\partial y}{\partial z} = y(1-y)$.

So for the output layer of a sigmoid neural net:

(4) $\dfrac{\partial y_i}{\partial w_i} = \dfrac{\partial y}{\partial z}\dfrac{\partial z}{\partial w_i} = y(1-y)x_i$

Substituting (3) and (4) into (2):

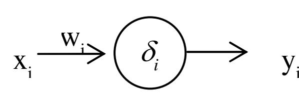
(5) $\dfrac{\partial P}{\partial w_i} = \dfrac{\partial P}{\partial y_i}\dfrac{\partial y_i}{\partial w_i} = (y^* - y)y(1-y)x_i$

Substituting (5) into the weight equation (1):

$w_i' = w_i + r(y^* - y)y(1-y)x_i$

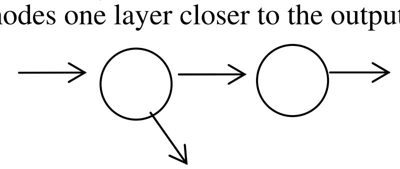
... which can be written in terms of δ, which is the derivative of P with respect to total input, $\dfrac{\partial P}{\partial z}$:

$w_i' = w_i + r\delta_i x_i$, where $\delta_i = \dfrac{\partial P}{\partial y}\dfrac{\partial y}{\partial z} = (y^* - y)y(1-y) = y(1-y)(y^* - y)$



(Note that to change P, one only needs to substitute the new $\dfrac{\partial P}{\partial y}$ into this equation.)

For an inner node, we use a similar equation that takes into account the output y_i for that node, and the δ and w values for the next layer, where "next layer" means the neighboring nodes one layer closer to the output layer:



$w_i' = w_i + r\delta_i x_i$, where $\delta_i = y_i(1-y_i)\sum \delta_j w_{ij}$

and w_{ij} is the weight between layers i and j.

Summary: For each layer, $w_i' = w_i + r\delta_i x_i$, where for outer layer $\delta_i = y(1-y)(y^* - y)$ inner layer $\delta_i = y_i(1-y_i)\sum \delta_j w_{ij}$