Question 1 - 15 points

1a (5 points). Function P_1 is a function defined on a set of samples S = {A, B, C, D}. P_1 is defined as shown below. Is there a value for y that makes P_1 a valid **probability function**? If yes, what is that value? Justify your answer.

$P_1(A) = 10*y$
$P_1(B) = 20*y$
$P_1(C) = 30*y$
$P_1(D) = 20*y$

y = 0.0125, so that the sum of all probabilities is 1.

1b (5 points). Function P_2 is a function defined on the set of real numbers. P_2 is defined as shown below. Is there a value for y that makes P_2 a valid **probability density function**? If yes, what is that value? Justify your answer.

$P_2(x) = 0$	if x < 100
$P_2(x) = 6*y$	if 100 <= x <= 110
$P_2(x) = 4*y$	if 110 <= x <= 130
$P_2(x) = 0$	if x > 130.

y = 1/140 = 0.00714, so that the integral of P_2 from –infinity to infinity is 1.

1c (5 points). Function P_3 is a function defined on the set of real numbers. P_3 is defined as shown below. Is there a value for y that makes P_3 a valid **probability density function**? If yes, what is that value? Justify your answer.

$P_3(x) = 0$	if x < 0
$P_3(x) = 7*y$	if 0 <= x <= 10
$P_3(x) = 3*y$	if x > 10

No. If y <= 0, then the integral of P^3 from –infinity to infinity is <= 0, which is illegal (the integral should be 1). If y > 0, then the integral of P^3 from –infinity to infinity is infinity, which is again illegal.

Question 2 – 10 points

Age of owner	Car	Minivan	SUV
under 30	0.15	0.05	0.1
between 30 and 50	0.1	0.15	0.1
over 50	0.15	0.15	0.05

The above joint distribution table shows the probability of combinations of vehicle types and vehicle owners. For example, the probability that a vehicle is an SUV and is owned by a person over 50 years old is 0.05. Using that table:

2a. (5 points) Determine the probability that a vehicle owner is under 30 years old.

P(vehicle owner age under 30) = 0.15 + 0.05 + 0.1 = 0.3

2b. (5 points) Determine P(vehicle type = Minivan | age of owner is under 30)

P(Minivan | age under 30) = P(Minivan AND age under 30) / P(age under 30)

P(Minivan AND age under 30) = 0.05

P(age under 30) = 0.3 (from Question **2a**).

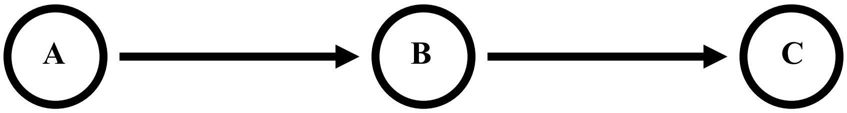
So, (Minivan AND age under 30) / P(age under 30) = 0.05 / 0.3 = 0.1667

Question 3 - 10 points

P(A)
0.6

A	P(B|A)
T	0.8
F	0.3

B	P(C|B)
T	0.2
F	0.9



3a. (5 points) Given the above Bayesian network, compute P((B=true) AND (C=false)). You do not have to carry out numerical calculations, but you have to write an expression that fully specifies the answer numerically.

P(B=t AND C=f) = P(A=t AND B=t AND C=f) + P(A=f AND B=t AND C=f)

= P(A=t) * P(B=t | A=t) * P(C=f | B=t) + P(A=f) * P(B=t | A=f) * P(C=f | B=t)

= 0.6 * 0.8 * (1 - 0.2) + (1-0.6) * 0.3 * (1 – 0.2) = 0.384 + 0.096 = 0.48

3b. (5 points) In the above Bayesian network, compute P(B). You do not have to carry out numerical calculations, but you have to write an expression that fully specifies the answer numerically.

P(B) = P(B | A) * P(A) + P(B | not A) * P(not A)

= 0.8 * 0.6 + 0.3 * 0.4 = 0.6

Question 4 = 15 points

4a (5 points). You have a Bayesian network of N nodes. Each node corresponds to a Boolean random variable. Each node has a maximum of 3 parents. How many numbers would you need to specify at most, in order to fully specify the probability distribution modeled by this Bayesian network? In other words, what is the maximum number of values you need to store (for the entire network) in order to full specify the probability table for each node? Justify your answer.

For each node, we must specify the probability of the variable of that node being true for each of at most 8 possible combinations of values for the parent nodes. So, we need to store at most 8*N values.

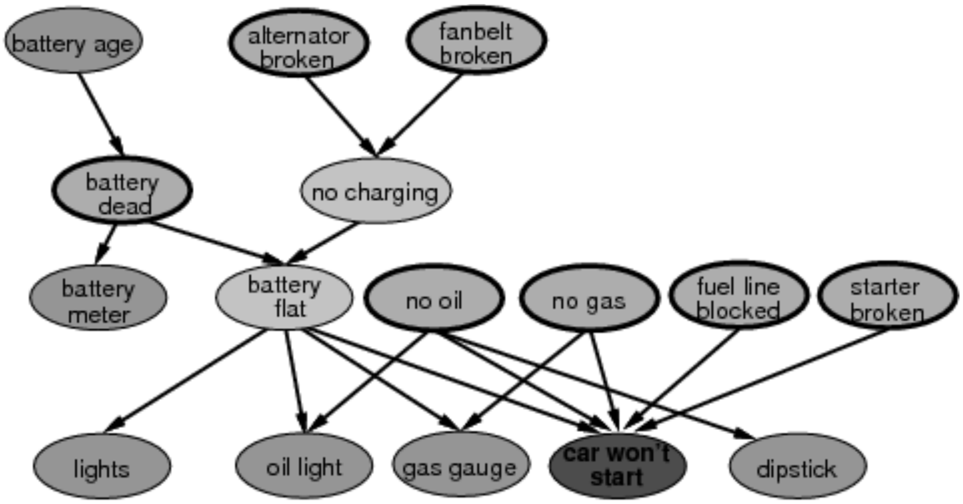
4b (5 points). What is the time complexity of doing inference in the Bayesian network of question 4a?

In the worst case, we must enumerate 2^N possible combinations of values for the network nodes. Applying the product rule for each combination of values takes O(N) time, so in total we need, in the worst case, $O(N2^N)$ time.

4c (5 points). Suppose that you want to model the same probability distribution as in question **4a**, but using a joint distribution table. How many values do you need to specify in that case?

We need to specify a probability value for each combination of values for the N variables, so we must specify 2^N values.

Question 5 – 10 points



5a (5 points). In the above Bayesian network, is the battery-flat event conditionally independent of the fanbelt-broken event given a value for the no-charging event? Justify your answer.

Yes. The fanbelt broken event influences the battery-flat event only through the no charging event. If we know the value for no-charging, then the value of fanbelt-broken makes no difference anymore in calculating the probability of battery flat.

5b (5 points). In the above Bayesian network, is the battery-flat event conditionally independent of the no-oil event given a value for the oil-light event? Justify your answer.

No. Given a value for the oil-light event, battery-flat and no-oil are competing causes, so they are conditionally dependent on each other.

Question 6 - 15 points

6a (5 points) Suppose that a decision tree is trained on 1000 training examples, and achieves 90% accuracy on the training examples. What is the smallest and largest accuracy that this decision tree can possibly achieve on a test set of 1000 examples? Justify your answer. You can assume there are only two classes.

The accuracy on the training set provides no guarantee whatsoever about accuracy on the test set, so the accuracy on the test set could be anywhere from 0% to 100%.

6b (5 points). Suppose that a decision tree is trained on 1000 training examples, and achieves 80% accuracy on the training examples. What is the smallest and largest possible value for the entropy at a leaf node of this decision tree? Remember that entropy is measured on the training set. You can assume there are only two classes.

Since some training examples are classified correctly, it is possible that some leaf node of the decision tree only receives training examples from a single class, in which case the entropy of that leaf node would be 0.

Since some training examples are classified incorrectly, it is possible that some leaf node of the decision tree receives equal numbers of training examples from both classes, in which case the entropy of that leaf node would be 1.

So, the entropy at a leaf node can be as small as 0 and as large as 1.

6c (5 points). Suppose that a decision tree is trained on 1000 training examples, and achieves 100% accuracy on the training examples. What is the smallest and largest possible value for the entropy at a leaf node of this decision tree? Again, remember that entropy is measured on the training set. You can assume there are only two classes.

Since no training examples are classified incorrectly, it is impossible that some leaf node of the decision tree receives training examples from both classes. Each leaf node receives training examples from a single class, and has entropy 0.

So, the entropy at a leaf node can be as small as 0 and as large as 0.

Question 7 - 5 points

We want to build a decision tree that determines whether a new laptop is going to break down or not during its first week. This decision tree is trained on 200 training examples (i.e., 200 cases of new laptops). The only thing that we know about each training example is the operating system that the laptop was running. In particular:

100 laptops in the training set crashed within their first week.
70 of those laptops were running operating system AA.
20 of those laptops were running operating system BB.
10 of those laptops were running operating system CC.

100 laptops in the training set did not crash within their first week.
10 of those laptops were running operating system AA.
20 of those laptops were running operating system BB.
70 of those laptops were running operating system CC.

Determine the entropy gain of choosing, at the root node, the predicate (Operating System = CC) as the test to apply at that node. You do not have to carry out numerical calculations, but you have to write an expression that fully specifies the answer numerically.

Let C1 be the child receiving examples where the operating system = CC.
Let C2 be the child receiving examples where the operating system != CC.

Entropy Gain = Entropy(parent) – 80/200 * Entropy(C1) – 120/200 * Entropy(C2)

Entropy(parent) = -0.5 * log2 (0.5) – 0.5 * log2 (0.5) = 1.

Entropy(C1) = -10/80 * log2 (10/80) – 70/80 * log2 (70/80) = 0.5436

Entropy(C2) = -90/120 * log2 (90/120) - 30/120 * log2 (30/120) = 0.8113

Entropy(parent) – 80/200 * Entropy(C1) – 120/200 * Entropy(C2)
= 1 – 80/200 * 0.5436 – 120/200 * 0.8113 = 0.2958.

Consequently, the entropy gain is 0.2958.

Question 8 - 10 points

8a (5 points). Design a perceptron that takes (in addition to the bias input) three inputs x, y, z, and outputs: 1 if x-2y-5 >= z, and 0 otherwise. Fully specify bias input and all weights. Your activation function g should return 1 if the weighted sum of inputs (including the bias input) is greater than or equal to 0, and g should return 0 otherwise.

x-2y-5 >= z => x-2y-z-5 >= 0.

Bias input has weight 5.
x has weight 1
y has weight -2
z has weight -1

8b (5 points). Design a neural network that takes X and Y as inputs, outputs 0 if X/Y > 5, and outputs 1 if X/Y <= 5. Fully specify bias input and all weights. Your activation function g should return 1 if the weighted sum of inputs (including the bias input) is greater than or equal to 0, and g should return 0 otherwise. Just for this question, you SHOULD ASSUME THAT X AND Y ARE GREATER THAN 0.

Since Y > 0, then X/Y <= 5 => X <= 5Y => X – 5Y <= 0 => 5Y - X >= 0. This can be recognized with a neuron P1 with weights:
0 for the bias input
-1 for X
5 for Y.

Question 9 - 10 points

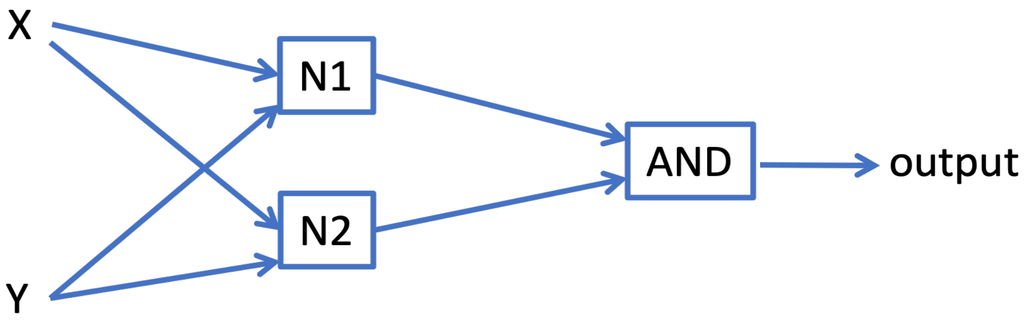
Design a neural network that takes X and Y as inputs, and outputs 1 if X - 3 = 2Y, and 0 otherwise. Fully specify bias input and all weights. Your activation function g should return 1 if the weighted sum of inputs (including the bias input) is greater than or equal to 0, and g should return 0 otherwise. If you are using a perceptron whose weights are fully specified in the textbook, just give the name of the perceptron, you do not have to specify those weights.

X – 3 = 2Y <=> (X – 3 >= 2Y) AND (X – 3 <= 2Y)

X – 3 >= 2Y => X – 2Y – 3 >= 0. We can recognize this case with a neuron N1 with the following weights:
3 for the bias input
1 for X
-2 for Y.

X – 3 <= 2Y => X – 2Y – 3 <= 0 => -X + 2Y + 3 >= 0. We can recognize this case with a neuron N1 with the following weights:
-3 for the bias input
-1 for X
2 for Y.

Using the neurons N1 and N2 defined above, and the AND neuron defined in the textbook, the final neural network is this:


X
Y
N1
N2
AND
output