1 PCA

1.1

You have the following data:

data #	x	y
1	5.51	5.35
2	20.82	24.03
3	-0.77	-0.57
4	19.30	19.38
5	14.24	12.77
6	9.74	9.68
7	11.59	12.06
8	-6.08	-5.22

You want to reduce the data into a single dimension representation. You are given the first principal component (0.694, 0.720).

(1). What is the representation (projected coordinate) for data #1 (x=5.51, y=5.35) in the first principal space?

Answer: (-5.74 or -5.75)

(2). What are the xy coordinates in the original space reconstructed using this first principal representation for data #1 (x=5.51, y=5.35)?

Answer: (5.31, 5.55)

(3). What is the representation (projected coordinate) for data #1 (x=5.51, y=5.35) in the second principal space?

Answer: 0.28
(±0.28, ±0.25 are accepted.)

(4). What is the reconstruction error if you use two principal components to represent original data?

Answer: 0

1.2

Given 3 data points in 2-d space, $(1,1)$, $(2,2)$ and $(3,3)$,

(a) (1 pt) what is the first principle component?

Solutions: $pc = (1/\sqrt{2}, 1/\sqrt{2})' = (0.707, 0.707)'$, (the negation is also correct)

(b) (1 pt) If we want to project the original data points into 1-d space by principle component you choose, what is the variance of the projected data?

Solutions: $4/3 = 1.33$

(c) (1 pt) For the projected data in (b), now if we represent them in the original 2-d space, what is the reconstruction error?

Solutions: 0

2 SVM

2.1

(a) Kernels

i. [4 points] In class we learnt that SVM can be used to classify linearly inseparable data by transforming it to a higher dimensional space with a kernel $K(x, z) = \phi(x)^T\phi(z)$, where $\phi(x)$ is a feature mapping. Let K_1 and K_2 be $R^n \times R^n$ kernels, K_3 be a $R^d \times R^d$ kernel and $c \in R^+$ be a positive constant. $\phi_1 : R^n \to R^d$, $\phi_2 : R^n \to R^d$, and $\phi_3 : R^d \to R^d$ are feature mappings of K_1, K_2 and K_3 respectively. Explain how to use ϕ_1 and ϕ_2 to obtain the following kernels.

a. $K(x, z) = cK_1(x, z)$

b. $K(x, z) = K_1(x, z)K_2(x, z)$

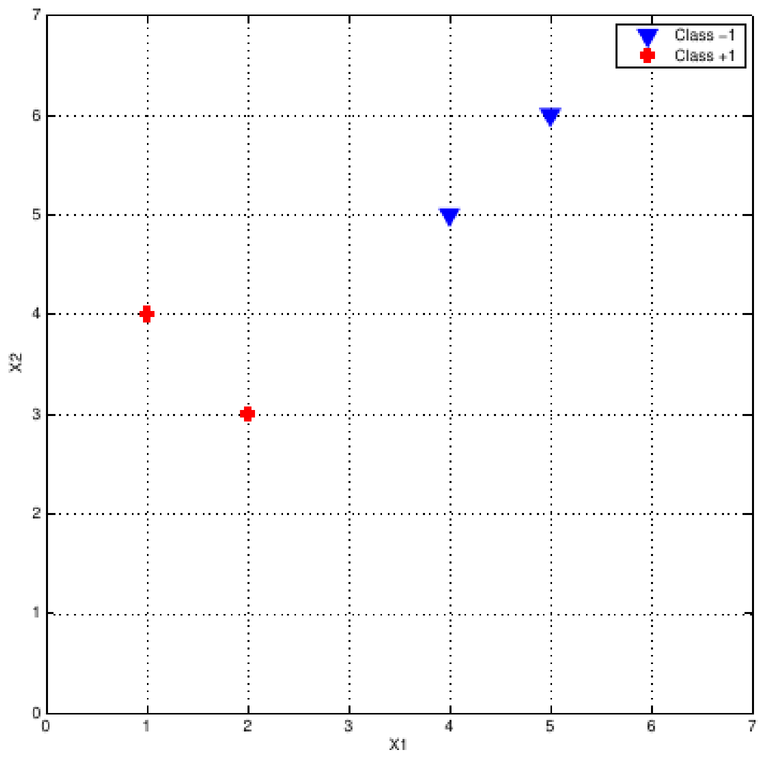
Solution:
a. $\phi(x) = \sqrt(c)\phi_1(x)$
b. $\phi(x) = \phi_1(x)\phi_2(x)$

ii. [2 points] One of the most commonly used kernels in SVM is the Gaussian RBF kernel: $k(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma}\right)$. Suppose we have three points, z_1, z_2, and x. z_1 is geometrically very close to x, and z_2 is geometrically far away from x. What is the value of $k(z_1, x)$ and $k(z_2, x)$?. Choose one of the following:

a. $k(z_1, x)$ will be close to 1 and $k(z_2, x)$ will be close to 0.
b. $k(z_1, x)$ will be close to 0 and $k(z_2, x)$ will be close to 1.
c. $k(z_1, x)$ will be close to c_1, $c_1 \gg 1$ and $k(z_2, x)$ will be close to c_2, $c_2 \ll 0$, where $c_1, c_2 \in R$
d. $k(z_1, x)$ will be close to c_1, $c_1 \ll 0$ and $k(z_2, x)$ will be close to c_2, $c_2 \gg 1$, where $c_1, c_2 \in R$

Solution:
Correct answer is a, RBF kernel generates a "bump" around the center x. For points z_1 close to the center of the bump, $K(z_1, x)$ will be close to 1, for points away from the center of the bump $K(z_2, x)$ will be close to 0.

(b) [3 points] Hard Margin SVM



Support vector machines learn a decision boundary leading to the largest margin from both classes. You are training SVM on a tiny dataset with 4 points shown in Figure 2. This dataset consists of two examples with class label -1 (denoted with plus), and two examples with class label +1 (denoted with triangles).

i. Find the weight vector **w** and bias b. What's the equation corresponding to the decision boundary?

Solution:
SVM tries to maximize the margin between two classes. Therefore, the optimal decision boundary is diagonal and it crosses the point (3,4). It is perpendicular to the line between support vectors (4,5) and (2,3), hence it is slope is m = -1. Thus the line equation is $(x_2 - 4) = -1(x_1 - 3) = x_1 + x_2 = 7$. From this equation, we can deduce that the weight vector has to be of the form (w_1, w_2), where $w_1 = w_2$. It also has to satisfy the following equations:
$2w_1 + 3w_2 + b = 1$ and
$4w_1 + 5w_2 + b = -1$

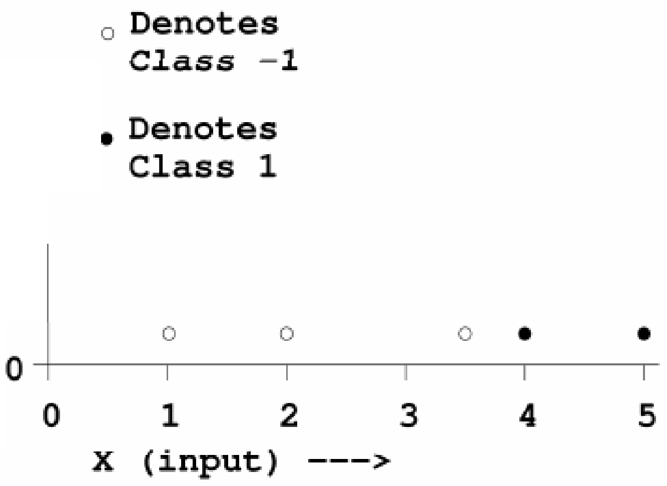
Hence $w_1 = w_2 = -1/2$ and $b = 7/2$

ii. Circle the support vectors and draw the decision boundary.

Solution:
See the solution above

2.2

Consider the following dataset. We are going to learn a linear SVM from it of the form $f(x) = \text{sign}(wx + b)$.



X	Y
1	-1
2	-1
3.5	-1
4	1
5	1

(a) What values for w and b will be learned by the linear SVM?

w = 4, b = -15

(b) What is the training set error of the above example? (expressed as the percentage of training points misclassified)

0

(c) What is the leave-one-out cross-validation error of the above example? (expressed as the percentage of left-out points misclassified)

2 wrong => 40%

2.3

(a) Kernel functions implicitly define some mapping function $\phi(\cdot)$ that transforms an input instance $\mathbf{x} \in \mathbb{R}^d$ to a high dimensional feature space Q by giving the form of dot product in Q: $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$.

Assume we use radial basis kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2)$. Thus we assume that there's some implicit unknown function $\phi(\mathbf{x})$ such that

$$\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

Prove that for any two input instances $\mathbf{x}_i$ and $\mathbf{x}_j$, the squared Euclidean distance of their corresponding points in the feature space Q is less than 2, i.e. prove that $\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 < 2$.

$$\begin{aligned}
&\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 \\
=&(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)) \cdot (\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)) \\
=&\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_i) + \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}_j) - 2 \cdot \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \\
=&2 - 2\exp(-\frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2) \\
<&2
\end{aligned}$$

(b) With the help of a kernel function, SVM attempts to construct a hyper-plane in the feature space Q that maximizes the margin between two classes. The classification decision of any $\mathbf{x}$ is made on the basis of the sign of

$$\hat{\mathbf{w}}^T \phi(\mathbf{x}) + \hat{w}_0 = \sum_{i \in SV} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + \hat{w}_0 = f(\mathbf{x}; \alpha, \hat{w}_0),$$

where $\hat{\mathbf{w}}$ and $\hat{w}_0$ are parameters for the classification hyper-plane in the feature space Q, SV is the set of support vectors, and α_i is the coefficient for the support vector.

Again we use the radial basis kernel function. Assume that the training instances are linearly separable in the feature space Q, and assume that the SVM finds a margin that perfectly separates the points.

([✓ True] or False) If we choose a test point $\mathbf{x}_{far}$ which is far away from any training instance $\mathbf{x}_i$ (distance here is measured in the original space $\mathbb{R}^d$), we will observe that $f(\mathbf{x}_{far}; \alpha, \hat{w}_0) \approx \hat{w}_0$.

$$\begin{aligned}
&\|\mathbf{x}_{far} - \mathbf{x}_i\| \gg 0, \ \forall i \in SV \\
\Longrightarrow &K(\mathbf{x}_{far}, \mathbf{x}_i) \approx 0, \ \forall i \in SV \\
\Longrightarrow &\sum\nolimits_{i \in SV} y_i \alpha_i K(\mathbf{x}_{far}, \mathbf{x}_i) \approx 0 \\
\Longrightarrow &f(\mathbf{x}_{far}; \alpha, \hat{w}_0) \approx \hat{w}_0
\end{aligned}$$

(c) **([✓ True] or False)** The SVM learning algorithm is guaranteed to find the globally optimal hypothesis with respect to its object function.

See Burges' tutorial.

(d) **(True or [✓ False])** The VC dimension of a Perceptron is smaller than the VC dimension of a simple linear SVM.

Both Perceptron and linear SVM are linear discriminators (i.e. a line in 2D space or a plane in 3D space ...), so they should have the same VC dimension.

(e) **([✓ True] or False)** After being mapped into feature space Q through a radial basis kernel function, a Perceptron may be able to achieve better classification performance than in its original space (though we can't guarantee this).

Sometimes it isn't sufficient for a given learning algorithm to work in the input space because the assumption behind the algorithm doesn't match the real pattern of the data. For example, SVM and Perceptron require the data are linearly separable. When the assumption isn't held, we may apply some kind of transformation to the data, mapping them to a new space where the learning algorithm can be used. Kernel function provides us a means to define the transformation. You may have read some papers that report improvements on classification performance using kernel function. However, the improvements are usually obtained from careful selection and tuning of parameters. Namely, we can't guarantee the improvements are always available.

(f) **(True or [✓ False])** After mapped into feature space Q through a radial basis kernel function, 1-NN using unweighted Euclidean distance may be able to achieve better classification performance than in original space (though we can't guarantee this).

Suppose $\mathbf{x}_i$ and $\mathbf{x}_j$ are two neighbors for the test instance $\mathbf{x}$ such that $\|\mathbf{x}-\mathbf{x}_i\| < \|\mathbf{x}-\mathbf{x}_j\|$. After mapped to feature space, $\|\phi(\mathbf{x}) - \phi(\mathbf{x}_i)\|^2 = 2 - 2\exp(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}_i\|^2) < 2 - 2\exp(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}_j\|^2) = \|\phi(\mathbf{x}) - \phi(\mathbf{x}_j)\|^2$. So, if $\mathbf{x}_i$ is the nearest neighbor of $\mathbf{x}$ in the original space, it will also be the nearest neighbor in the feature space. Therefore, 1-NN doesn't work better in the feature space. Please note that k-NN using non-Euclidean distance or weighted voting may work.

2.4

1. **Properties of Kernel**

   1.1. (2 pts) Prove that the kernel $K(x_1, x_2)$ is symmetric, where x_i and x_j are the feature vectors for i^{th} and j^{th} examples.

   hints: Your proof will not be longer than 2 or 3 lines.

   Solutions: Let $\Phi(x_1)$ and $\Phi(x_2)$ be the feature maps for x_i and x_j, respectively. Then, we have $K(x_1, x_2) = \Phi(x_1)'\Phi(x_2) = \Phi(x_2)'\Phi(x_1) = K(x_2, x_1)$

   1.2. (4 pts) Given n training examples $(x_i, x_j)(i, j = 1, ..., n)$, the kernel matrix $\mathbf{A}$ is an $n \times n$ square matrix, where $\mathbf{A}(i, j) = K(x_i, x_j)$. Prove that the kernel matrix $\mathbf{A}$ is semi-positive definite.

   hints: (1) Remember that an $n \times n$ matrix $\mathbf{A}$ is semi-positive definite iff. for any n dimensional vector $\mathbf{f}$, we have $\mathbf{f}'\mathbf{A}\mathbf{f} \geq 0$. (2) For simplicity, you can prove this statement just for the following particular kernel function: $K(x_i, x_j) = (1 + x_i x_j)^2$.

   Solutions: Let $\Phi(x_i)$ be the feature map for the i^{th} example and define the matrix $\mathbf{B} = [\mathbf{\Phi(x_1)}, ..., \mathbf{\Phi(x_n)}]$. It is easy to verify that $\mathbf{A} = \mathbf{B}'\mathbf{B}$. Then, we have $\mathbf{f'Af = (Bf)'Bf = \|Bf\|^2 \geq 0}$

3 Decision Tree

3.1

Master Yoda is concerned about the number of Jedi apprentices that have turned to the Dark Side, so he's decided to train a decision tree on some historical data to help identify problem cases in the future. The following table summarizes whether or not each of 12 initiates turned to the Dark Side based on their age when their Jedi training began, whether or not they completed their training, their general disposition, and their species.

Dark Side	Age Started Training	Completed Training	Disposition	Species
0	5	1	Happy	Human
0	9	1	Happy	Gungan
0	6	0	Happy	Wookiee
0	6	1	Sad	Mon Calamari
0	7	0	Sad	Human
0	8	1	Angry	Human
0	5	1	Angry	Ewok
1	9	0	Happy	Ewok
1	8	0	Sad	Human
1	8	0	Sad	Human
1	6	0	Angry	Wookiee
1	7	0	Angry	Mon Calamari

(a) *(3 points)* What is the initial entropy of *Dark Side*?

$-\frac{5}{12}\log_2\frac{5}{12} - \frac{7}{12}\log_2\frac{7}{12} = \mathbf{0.979868756651153}$

(b) *(3 points)* Which attribute would the decision-tree building algorithm choose to use for the root of the tree?

Completed Training

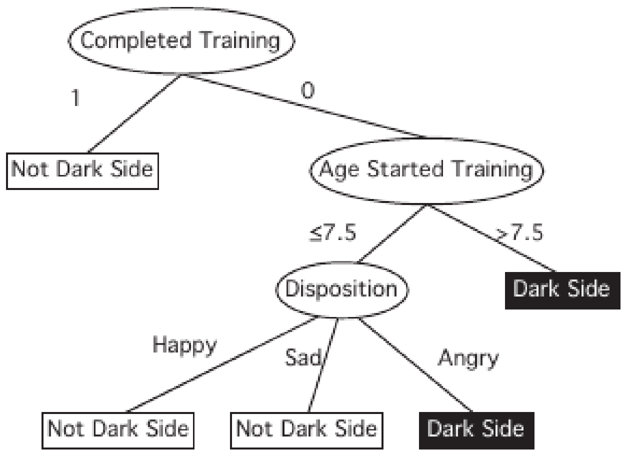
(c) *(3 points)* What is the information gain of the attribute you chose to split on in the previous question?

$\mathbf{a} - (\frac{5}{12}(-\frac{0}{5}\log_2\frac{0}{5} - \frac{5}{5}\log_2\frac{5}{5}) + \frac{7}{12}(-\frac{5}{7}\log_2\frac{5}{7} - \frac{2}{7}\log_2\frac{2}{7})) = \mathbf{0.476381758320618}$

where a is the answer to part (a)

(Note that $log0$ is $-\infty$, but we define $0log0 = 0$.)

(d) *(3 points)* Draw the full decision tree that would be learned for this data (with no pruning).



(e) *(2 points)* Consider the possibility that the input data above is noisy and not completely accurate, so that the decision tree you learned may not accurately reflect the function you want to learn. If you were to evaluate the three initiates represented by the data points below, on which one would you be most confident of your prediction, and why?

Name	Age Started Training	Completed Training	Disposition	Species
Ardath	5	0	Angry	Human
Barbar	8	0	Angry	Gungan
Caldar	8	0	Happy	Mon Calamari

Barbar. The rule we learned is that you turn to the Dark Side if you did not complete your training and you either were too old or angry. Barbar falls under both clauses of the OR part, so even if one half of the rule learned is wrong, he still goes to the Dark Side. A variety of answers were accepted provided they had suitable justification.

(f) *(3 points)* Assume we train a decision tree to predict Z from A, B, and C using the following data (with no pruning):

Z	A	B	C
0	0	0	0
0	0	0	1
0	0	0	1
0	0	1	0
0	0	1	1
1	0	1	1
0	1	0	0
1	1	0	1
1	1	1	0
1	1	1	0
0	1	1	1
1	1	1	1

What would be the training set error for this dataset? Express your answer as the number of records out of 12 that would be misclassified.

2. We have four pairs of records with duplicate input variables, but only two of these have contradictory output values. One item of each of these two pairs will always be misclassified.

(g) *(3 points)* Consider a decision tree built from an arbitrary set of data. If the output is discrete-valued and can take on k different possible values, what is the maximum training set error (expressed as a fraction) that any data set could possibly have?

$\frac{k-1}{k}$ **Consider a set of data points with identical inputs but with outputs evenly distributed among the k possible values. The tree will label all these points as a single class which will be wrong for the ones in the other $k-1$ classes. Increasing the relative amount of any one class will guarantee that that class will be chosen as the label for all the points, so the error fraction will decrease (as that class now represents more than $\frac{1}{k}$ of the points.**

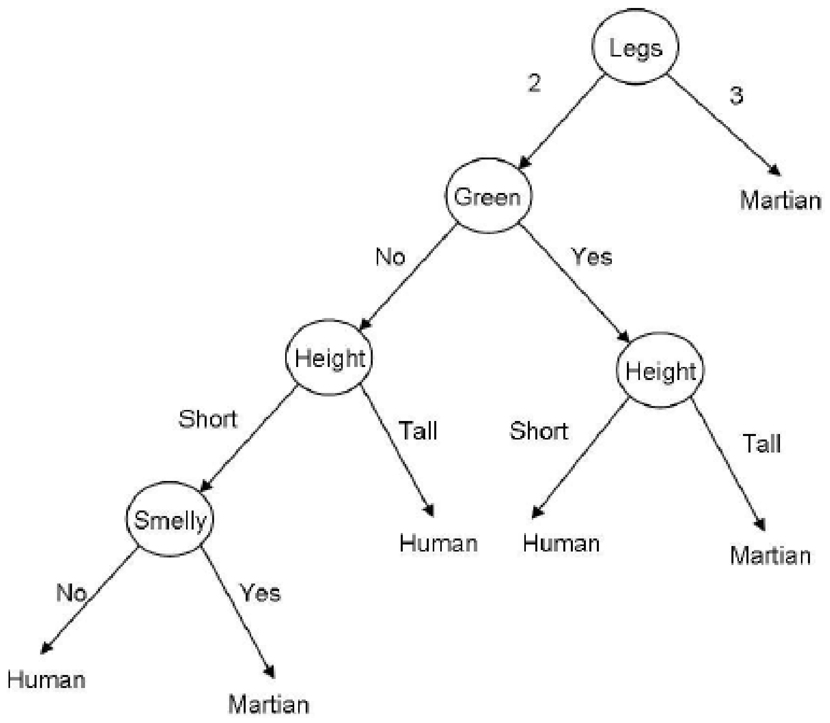
3.2

4 [16 points] Decision Trees

NASA wants to be able to discriminate between Martians (M) and Humans (H) based on the following characteristics: Green $\in \{N, Y\}$, Legs $\in \{2, 3\}$, Height $\in \{S, T\}$, Smelly $\in \{N, Y\}$. Our available training data is as follows:

	Species	Green	Legs	Height	Smelly
1)	M	N	3	S	Y
2)	M	Y	2	T	N
3)	M	Y	3	T	N
4)	M	N	2	S	Y
5)	M	Y	3	T	N
6)	H	N	2	T	Y
7)	H	N	2	S	N
8)	H	N	2	T	N
9)	H	Y	2	S	N
10)	H	N	2	T	Y

a)[8 points] Greedily learn a decision tree using the ID3 algorithm and draw the tree.
See the following figure for the ID3 decision tree:

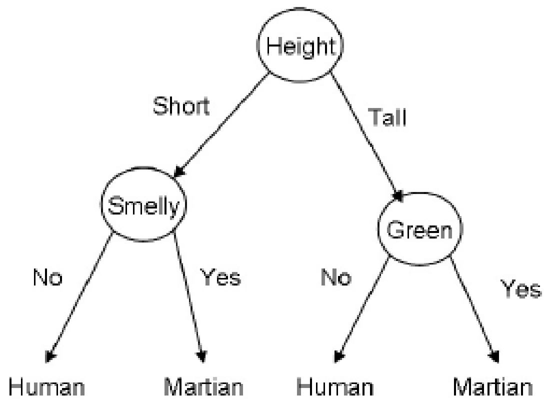


b) i) [3 points] Write the learned concept for Martian as a set of conjunctive rules (e.g., if (green=Y and legs=2 and height=T and smelly=N), then Martian; else if ... then Martian; ...; else Human).

Only the disjunction of conjunctions for Martians was required.
(Legs=3) $\vee$
(Legs=2 $\wedge$ *Green=Yes* $\wedge$ *Height=Tall)* $\vee$
(Legs=2 $\wedge$ *Green=No* $\wedge$ *Height=Short* $\wedge$ *Smelly=Yes)*

ii) [5 points] The solution of part b)i) above uses up to 4 attributes in each conjunction. Find a set of conjunctive rules using only 2 attributes per conjunction that still results in zero error in the training set. Can this simpler hypothesis be represented by a decision tree of depth 2? Justify.

We allowed a little variation on this one because the question could be interpreted as allowing conjunctions with up to two terms. In fact, only two two-term conjunctions are necessary:
(Green=Yes $\wedge$ *Height=Tall)* $\vee$ *(Smelly=Yes* $\wedge$ *Height=Short)*
These conjunctive rules share the height term, so a depth-2 tree is possible. See the figure below.



Notice how ID3 finds a tree that is much longer than the optimal tree. This is due to the greediness of the ID3 algorithm.

4 Neural Networks

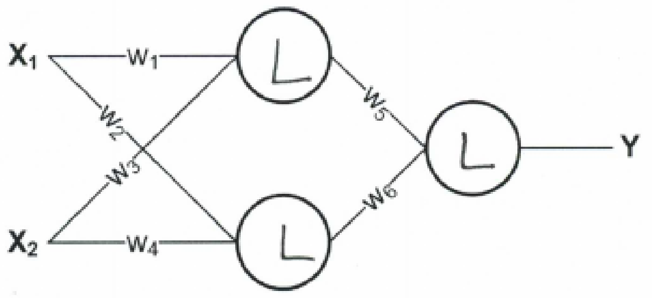
4.1

Consider a two-layer neural network to learn a function $f : X \to Y$ where $X = \langle X_1, X_2 \rangle$ consists of two attributes. The weights, $w_1, \cdots, w_6$, can be arbitrary. There are two possible choices for the function implemented by each unit in this network:

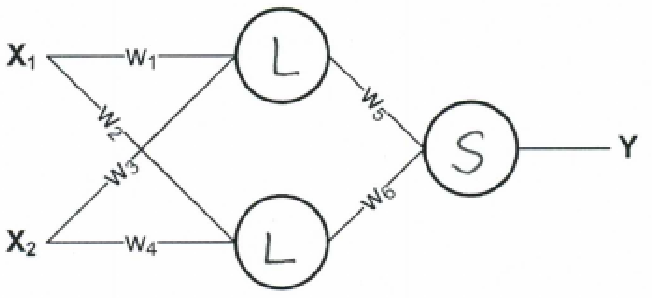
- **S**: signed sigmoid function $S(a) = sign[\sigma(a) - 0.5] = sign[\frac{1}{1+\exp(-a)} - 0.5]$
- **L**: linear function $L(a) = c\,a$

where in both cases $a = \sum_i w_i X_i$

1. (4 pts) Assign proper activation functions (**S** or **L**) to each unit in the following graph so this neural network simulates a linear regression: $Y = \beta_1 X_1 + \beta_2 X_2$.



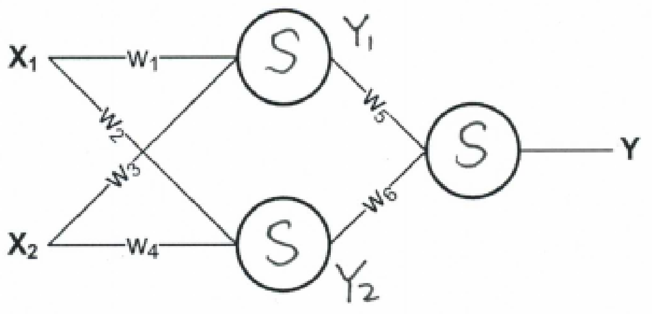
2. (4 pts) Assign proper activation functions (**S** or **L**) for each unit in the following graph so this neural network simulates a binary logistic regression classifier: $Y = \arg\max_y P(Y = y|X)$, where $P(Y = 1|X) = \frac{\exp(\beta_1 X_1 + \beta_2 X_2)}{1+\exp(\beta_1 X_1 + \beta_2 X_2)}$, $P(Y = -1|X) = \frac{1}{1+\exp(\beta_1 X_1 + \beta_2 X_2)}$.



3. (3 pts) Following problem 3.2, derive β_1 and β_2 in terms of $w_1, \cdots, w_6$.

$$\beta_1 = c\,(W_1 W_5 + W_2 W_6)$$
$$\beta_2 = c\,(W_3 W_5 + W_4 W_6)$$

4. (4 pts) Assign proper activation functions (**S** or **L**) for each unit in the following graph so this neural network simulates a boosting classifier which combines two logistic regression classifiers, $f_1 : X \to Y_1$ and $f_2 : X \to Y_2$, to produce its final prediction: $Y = sign[\alpha_1 Y_1 + \alpha_2 Y_2]$. Use the same definition in problem 3.2 for f_1 and f_2.



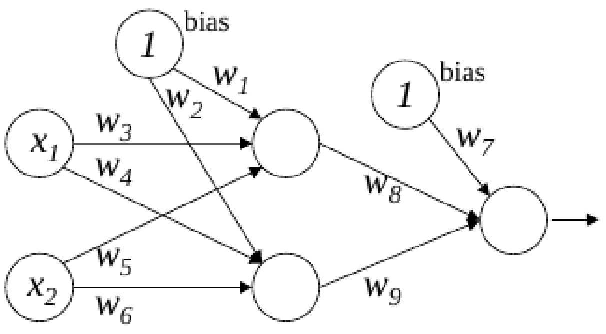
5. (3 pts) Following problem 3.4, derive α_1 and α_2 in terms of $w_1, \cdots, w_6$.

$\alpha_1 = W_5$ #

$\alpha_2 = W_6$

4.2

Consider a neural net for a binary classification which has one hidden layer as shown in the figure. We use a linear activation function $h(z) = cz$ at hidden units and a sigmoid activation function $g(z) = \frac{1}{1+e^{-z}}$ at the output unit to learn the function for $P(\mathbf{y} = 1|\mathbf{x}, \mathbf{w})$ where $\mathbf{x} = (x_1, x_2)$ and $\mathbf{w} = (w_1, w_2, \ldots, w_9)$.



1. (5%) What is the output $P(\mathbf{y} = 1 \mid \mathbf{x}, \mathbf{w})$ from the above neural net? Express it in terms of x_i, c and weights w_i. What is the final classification boundary?

   (sol.)

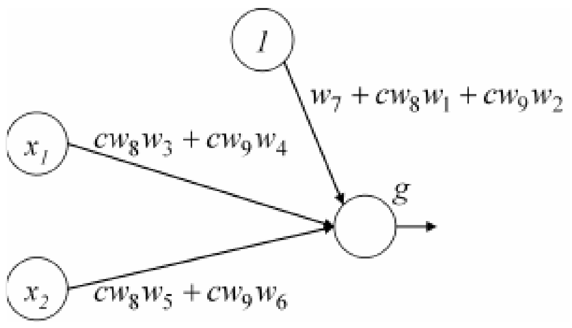
$$\begin{aligned} & g(w_7 + w_8 h(w_1 + w_3 x_1 + w_5 x_2) + w_9 h(w_2 + w_4 x_1 + w_6 x_2)) \\ = & \frac{1}{1 + \exp(-(w_7 + cw_8 w_1 + cw_9 w_2 + (cw_8 w_3 + cw_9 w_4)x_1 + (cw_8 w_5 + cw_9 w_6)x_2))} \end{aligned}$$

   The classification boundary is :

$$w_7 + cw_8 w_1 + cw_9 w_2 + (cw_8 w_3 + cw_9 w_4)x_1 + (cw_8 w_5 + cw_9 w_6)x_2 = 0$$

2. (5%) Draw a neural net with no hidden layer which is equivalent to the given neural net, and write weights $\tilde{\mathbf{w}}$ of this new neural net in terms of c and w_i.

   (sol.)



3. (5%) Is it true that any multi-layered neural net with linear activation functions at hidden layers can be represented as a neural net without any hidden layer? Briefly explain your answer.

   (sol.) Yes. If linear activation functions are used for all the hidden units, output from hidden units will be written as linear combination of input features. Since these intermediate output serves as input for the final output layer, we can always find an equivalent neural net which does not have any hidden layer as seen in the example above.