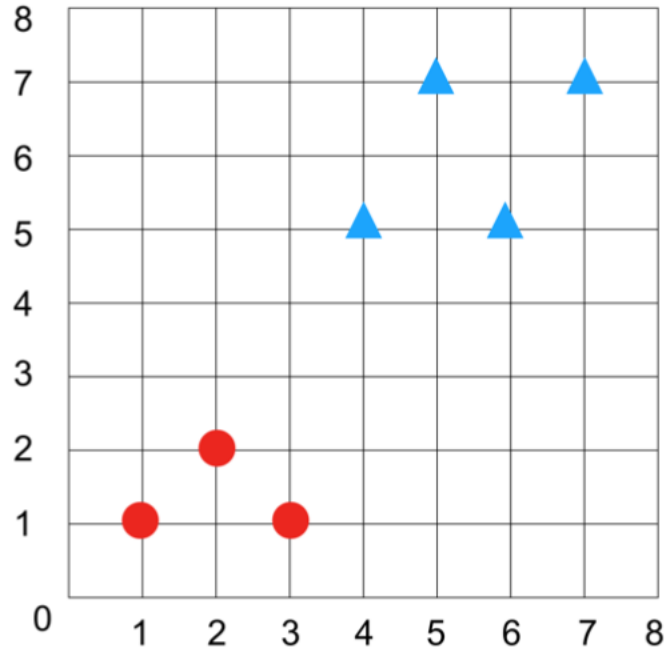
Supervised learning Q&A

Question 1. SVM

- A hard-margin support vector machine (SVM), takes n training points $X_1, X_2, \ldots, X_n \in \mathbb{R}^d$ with labels $y_1, y_2, \ldots, y_n \in \{+1, -1\}$, and finds parameters $w \in \mathbb{R}^d$ and $\alpha \in \mathbb{R}$ that satisfy a certain objective function subject to the constraints

$$y_i(X_i \cdot w + \alpha) \geq 1, \forall i \in \{1, \ldots, n\}.$$

For parts (a) and (b), consider the following training points. Circles are classified as positive examples with label +1 and triangles are classified as negative examples with label −1.



(a) Which points are the support vectors? Write it as $\begin{bmatrix}\text{horizontal} \\ \text{vertical}\end{bmatrix}$. E.g., the bottom right circle is $\begin{bmatrix}3 \\ 1\end{bmatrix}$.

(b) If we add the sample point x= $\begin{bmatrix}5 \\ 1\end{bmatrix}$ with label −1 (triangle) to the training set, which points are the support vectors?

For parts (c)–(f), forget about the figure above, but assume that there is at least one sample point in each class and that the sample points are linearly separable.

(c) Describe the geometric relationship between w and the decision boundary.

(d) Describe the relationship between w and the margin. (For the purposes of this question, the margin is just a number.)

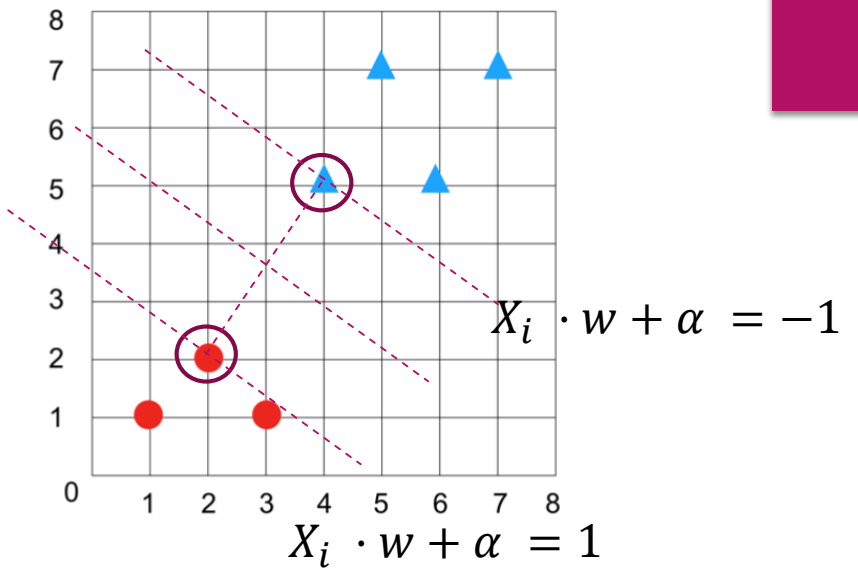
(e) Knowing what you know about the hard-margin SVM objective function, explain why for the optimal(w,α), there must be at least one sample point for which Xi ·w+α=1 and one sample point for which Xi ·w+α=−1.

(f) If we add new features to the sample points (while retaining all the original features), can the optimal $// w_{new} //$ in the enlarged SVM be greater than the optimal $// w_{old} //$ in the original SVM? Can it be smaller? Can it be the same? Explain why! (Most of the points will be for your explanation.)

Question 1.(a)(b)

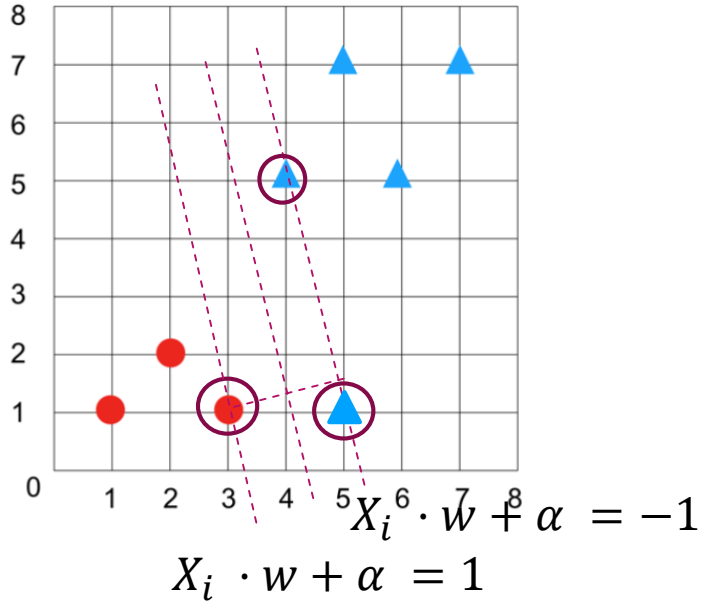
(a) Which points are the support vectors? Write it as $\begin{bmatrix}\text{horizontal}\\ \text{vertical}\end{bmatrix}$. E.g., the bottom right circle is $\begin{bmatrix}3\\1\end{bmatrix}$.

Answer: $\begin{bmatrix}2\\2\end{bmatrix}$, $\begin{bmatrix}4\\5\end{bmatrix}$



(b) If we add the sample point x= $\begin{bmatrix}5\\1\end{bmatrix}$ with label −1 (triangle) to the training set, which points are the support vectors?

Answer: $\begin{bmatrix}3\\1\end{bmatrix}$, $\begin{bmatrix}4\\5\end{bmatrix}$, $\begin{bmatrix}5\\1\end{bmatrix}$



Question 1.(c)(d)(e)

(c) Describe the geometric relationship between w and the decision boundary.

Answer: The weight vector w (the normal vector) is orthogonal to the decision boundary.

(d) Describe the relationship between w and the margin. (For this question, the margin is just a number.)

Answer: The margin is 2/‖w‖(the distance between the plus-plane and the minus-plane).

(e) Knowing what you know about the hard-margin SVM objective function, explain why for the optimal(w,α), there must be at least one sample point for which $X_i \cdot w + \alpha = 1$ and one sample point for which $X_i \cdot w + \alpha = -1$.

Answer: The objective is to minimize ‖w‖² (or equivalently, ‖w‖), subject to $y_i(X_i \cdot w + \alpha) \geq 1$.
If every sample point has $y_i(X_i \cdot w + \alpha) > 1$, we can simply scale w to make it smaller until there is a point such that $y_i(X_i \cdot w + \alpha) = 1$, thereby improving the "solution."
If we have a positive sample point for which $X_i \cdot w + \alpha = 1$ but every negative sample point has $X_i \cdot w + \alpha < -1$, we can make α a little greater so that every sample point has $y_i(X_i \cdot w + \alpha) > 1$.
Then we can shrink w some more.
So any such "solution" cannot be optimal. (The symmetric argument applies if a negative sample point touches the slab but not positive sample point does.)

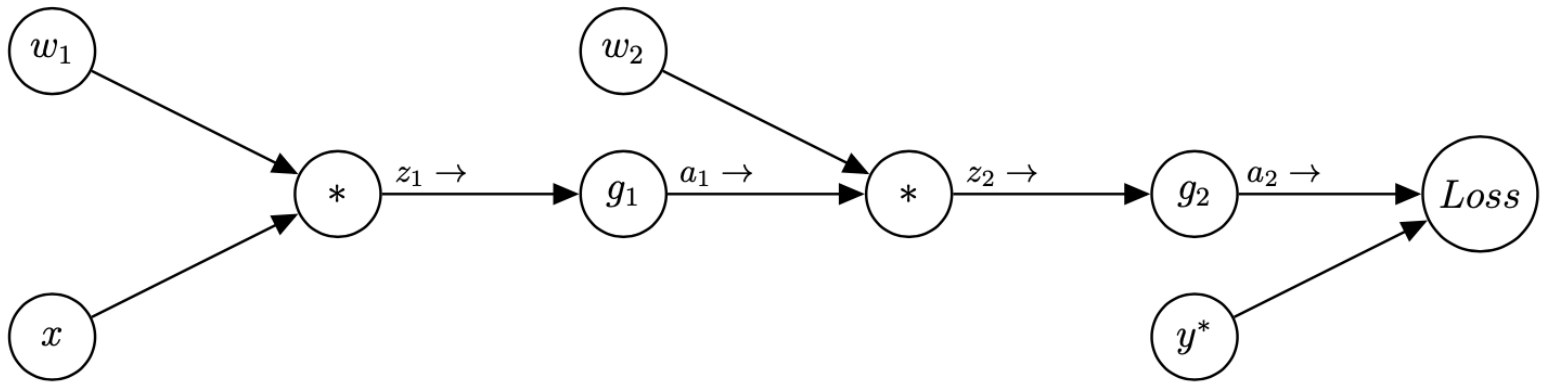
Question 1.(f)

(f) If we add new features to the sample points (while retaining all the original features), can the optimal $// w_{new} //$ in the enlarged SVM be greater than the optimal $// w_{old} //$ in the original SVM? Can it be smaller? Can it be the same? Explain why! (Most of the points will be for your explanation.)

Answer: It can be smaller, or it can be the same, but it cannot be greater.
If w_{old} and α are an optimal solution of the original SVM, when we add features we can create a w_{new} that has the same values as w_{old}, with zeros added for the new features. Then w_{new} and α satisfy all the constraints of the enlarged SVM. These might not be the optimal solution, but the optimal solution of the enlarged SVM cannot have $\|w_{new}\|$ greater than $\|w_{old}\|$.
$\|w_{new}\|$ can be smaller, because the new features can put an arbitrarily large amount of space between the classes, making the margin arbitrarily large.
$\|w_{new}\|$ will be the same as $\|w_{old}\|$ if the new features are all zeros in all the sample points.

Question 2: NN

- Consider the following computation graph for a simple neural network for binary classification. Here x is a single real-valued input feature with an associated class y⋆ (0 or 1). There are two weight parameters w_1 and w_2, and non-linearity functions g_1 and g_2 (to be defined later, below). The network will output a value a_2 between 0 and 1, representing the probability of being in class 1. We will be using a loss function Loss (to be defined later, below), to compare the prediction a_2 with the true class y⋆.



(a) Perform the forward pass on this network, writing the output values for each node z_1, a_1, z_2 *and* a_2 in terms of the node's input values.

(b) Compute the loss $Loss(a2, y\star)$ in terms of the input x, weights w_i, and activation functions g_i.

(c) Now we will work through parts of the backward pass, incrementally. Use the chain rule to derive $\frac{\partial Loss}{\partial w_2}$.

Write your expression as a product of partial derivatives at each node: i.e. the partial derivative of the node's output with respect to its inputs. (Hint: the series of expressions you wrote in part (a) will be helpful;you may use any of those variables.)

(d) Suppose the loss function is quadratic, $Loss(a_2, y^\star) = \frac{1}{2}(a2 - y^\star)^2$, and g_1 and g_2 are both sigmoid functions $g(z) = \frac{1}{1+e^{-z}}$ (note: it's typically better to use a different type of loss, cross-entropy, for classification problems, but we'll use this to make the math easier). Using the chain rule from Part (c), and the fact that $\frac{\partial g(z)}{\partial z} = g(z)(1 - g(z))$ for the sigmoid function, write $\frac{\partial Loss}{\partial w_2}$ in terms of the values from the forward pass, $y^\star, a_1, and\ a_2$.

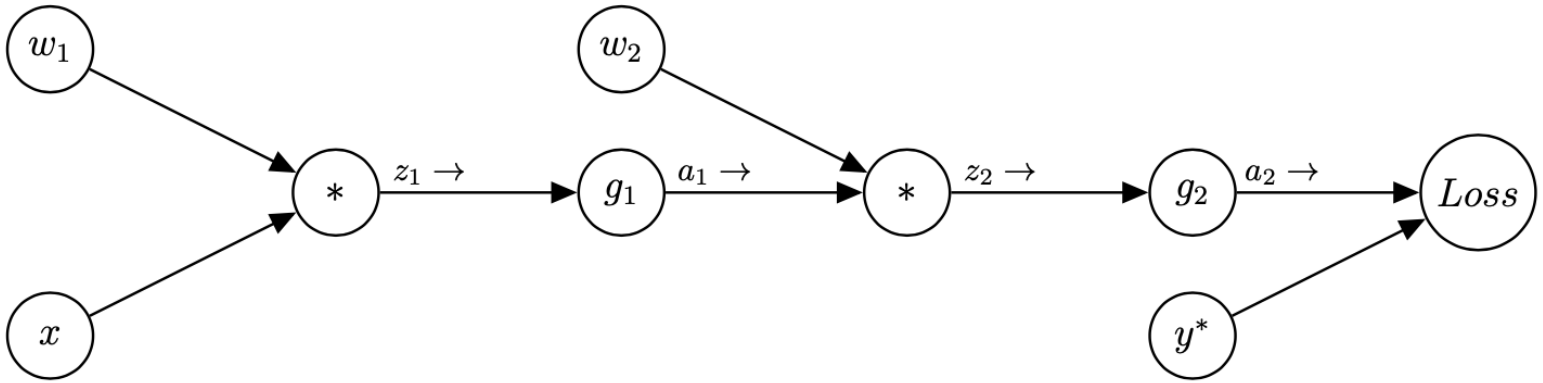
(e) Now use the chain rule to derive $\frac{\partial Loss}{\partial w_1}$ as a product of partial derivatives at each node used in the chain rule.

(f) Finally, write $\frac{\partial Loss}{\partial w_1}$ in terms of $x, y^\star, w_i, a_i, z_i$.

(g) What is the gradient descent update for w_1 with step-size α in terms of the values computed above?

Question 2.(a)

(a) Perform the forward pass on this network, writing the output values for each node z_1, a_1, z_2 *and* a_2 in terms of the node's input values.

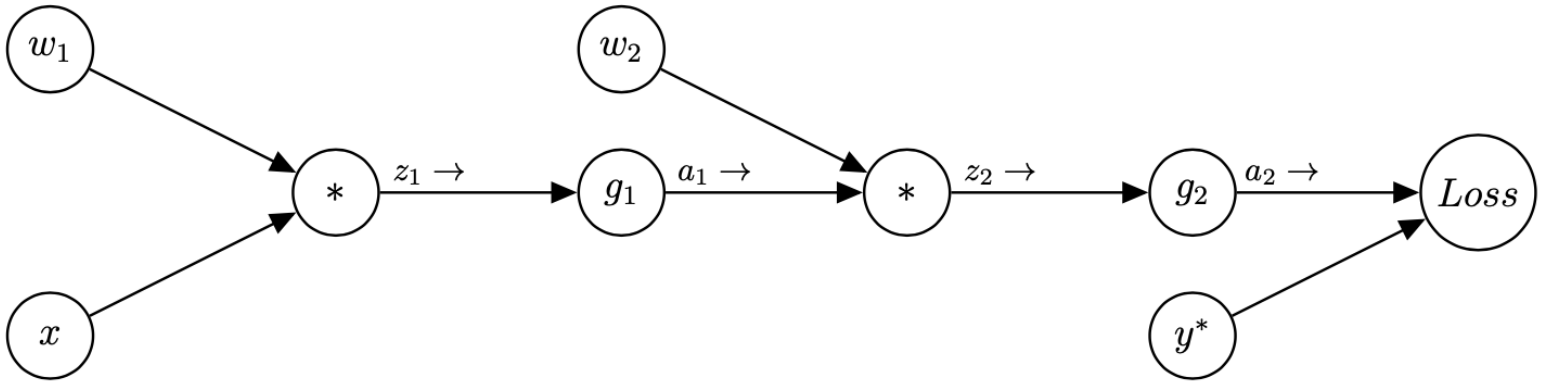


Answer:

$$z_1 = x * w_1$$
$$a_1 = g_1(z_1)$$
$$z_2 = a_1 * w_2$$
$$a_2 = g_2(z_2)$$

Question 2.(b)

(b) Compute the loss $Loss(a2, y\star)$ in terms of the input x, weights w_i, and activation functions g_i.



$$z_1 = x * w_1$$
$$a_1 = g_1(z_1)$$
$$z_2 = a_1 * w_2$$
$$a_2 = g_2(z_2)$$

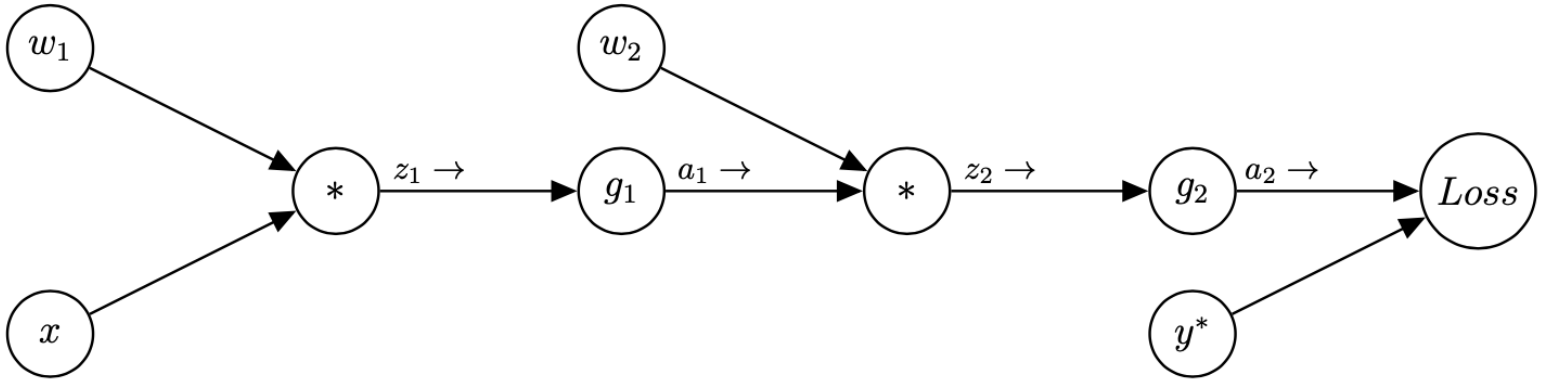
Answer:Recursively substituting the values computed in part (a), we have:

$$Loss(a_2, y^\star) = Loss(g_2(w_2 * g_1(w_1 * x)), y^\star)$$

Question 2.(c)

(c) Now we will work through parts of the backward pass, incrementally. Use the chain rule to derive $\frac{\partial Loss}{\partial w_2}$.

Write your expression as a product of partial derivatives at each node: i.e. the partial derivative of the node's output with respect to its inputs. (Hint: the series of expressions you wrote in part (a) will be helpful; you may use any of those variables.)



$$z_2 = a_1 * w_2$$
$$a_2 = g_2(z_2)$$

Answer:

$$\frac{\partial Loss}{\partial w_2} = \frac{\partial Loss}{\partial a_2}\frac{\partial a_2}{\partial z_2}\frac{\partial z_2}{\partial w_2}$$

Question 2.(d)

(d) Suppose the loss function is quadratic, $Loss(a_2, y^\star) = \frac{1}{2}(a_2 - y^\star)^2$, and g_1 and g_2 are both sigmoid functions $g(z) = \frac{1}{1+e^{-z}}$ (note: it's typically better to use a different type of loss, cross-entropy, for classification problems, but we'll use this to make the math easier). Using the chain rule from Part (c), and the fact that $\frac{\partial g(z)}{\partial z} = g(z)(1 - g(z))$ for the sigmoid function, write $\frac{\partial Loss}{\partial w_2}$ in terms of the values from the forward pass, $y^\star, a_1, and\ a_2$.

Answer:

First we'll compute the partial derivatives at each node:

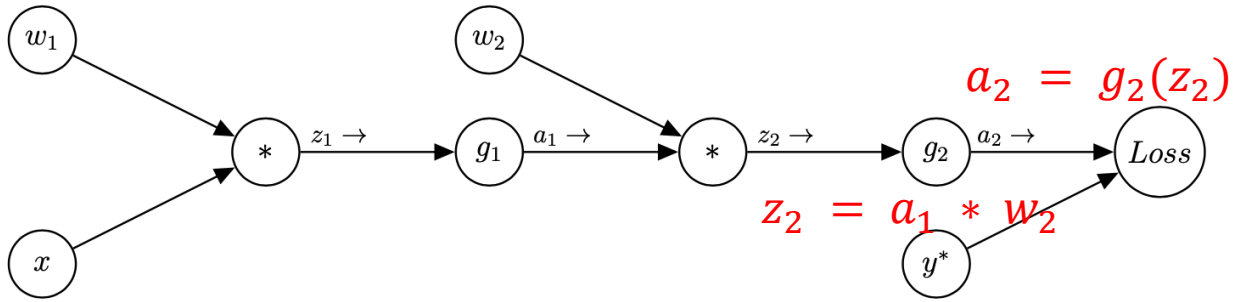
$$\frac{\partial Loss}{\partial a_2} = (a_2 - y^*)$$

$$\frac{\partial a_2}{\partial z_2} = \frac{\partial g_2(z_2)}{\partial z_2} = g_2(z_2)\big(1 - g_2(z_2)\big) = a_2(1 - a_2)$$

$$\frac{\partial z_2}{\partial w_2} = a_1$$

Now we can plug into the chain rule from part (c):

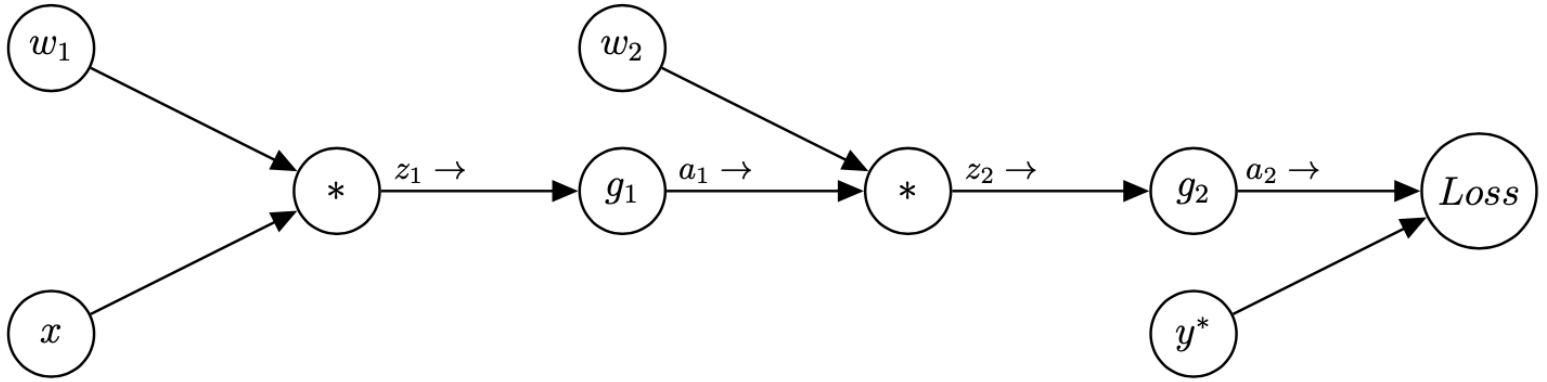
$$\frac{\partial Loss}{\partial w_2} = \frac{\partial Loss}{\partial a_2}\frac{\partial a_2}{\partial z_2}\frac{\partial z_2}{\partial w_2} = (a_2 - y^*) * a_2(1 - a_2) * a_1$$



$$Loss(a_2, y^\star) = \frac{1}{2}(a_2 - y^\star)^2$$

Question 2.(e)

(e) Now use the chain rule to derive $\frac{\partial Loss}{\partial w_1}$ as a product of partial derivatives at each node used in the chain rule.



$$z_1 = x * w_1$$
$$a_1 = g_1(z_1)$$
$$z_2 = a_1 * w_2$$
$$a_2 = g_2(z_2)$$

Answer:

$$\frac{\partial Loss}{\partial w_1} = \frac{\partial Loss}{\partial a_2}\frac{\partial a_2}{\partial z_2}\frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1}\frac{\partial z_1}{\partial w_1}$$

Question 2.(f)

(f) Finally, write $\frac{\partial Loss}{\partial w_1}$ in terms of $x, y^\star, w_i, a_i, z_i$.

part(e):

$$\frac{\partial Loss}{\partial a_2} = (a_2 - y^*)$$

$$\frac{\partial a_2}{\partial z_2} = \frac{\partial g_2(z_2)}{\partial z_2} = g_2(z_2)\big(1 - g_2(z_2)\big) = a_2(1 - a_2)$$

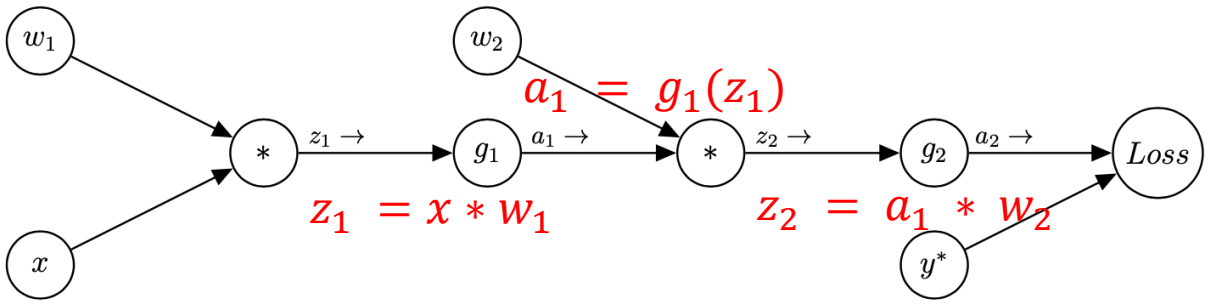
Answer:

The partial derivatives at each node (in addition to the ones we computed in Part (d)) are:

$$\frac{\partial z_2}{\partial a_1} = w_2$$

$$\frac{\partial a_1}{\partial z_1} = \frac{\partial g_1(z_1)}{\partial z_1} = g_1(z_1)\big(1 - g_1(z_1)\big) = a_1(1 - a_1)$$

$$\frac{\partial z_1}{\partial w_1} = x$$



Plugging into the chain rule from Part (e) gives:

$$\frac{\partial Loss}{\partial w_1} = \frac{\partial Loss}{\partial a_2}\frac{\partial a_2}{\partial z_2}\frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1}\frac{\partial z_1}{\partial w_1} = (a_2 - y^*) * a_2(1 - a_2) * w_2 * a_1(1 - a_1) * x$$

Question 2.(g)

(g) What is the gradient descent update for w_1 with step-size α in terms of the values computed above?

Answer:

$$w_1 \leftarrow w_1 - \alpha \frac{\partial Loss}{\partial w_1}$$

$$w_1 \leftarrow w_1 - \alpha(a_2 - y^*) * a_2(1 - a_2) * w_2 * a_1(1 - a_1) * x$$

Question 3. DT

- Here is a table which records some data about whether a student will go out to play. Use decision tree to analysis the following questions:

(a) Using entropy to analysis which attribute should be root node among outlook, temperature, humidity and windy? Write your analysis process in question .

(b) Draw the decision tree.

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

Review Entropy

- Entropy is a measure of the uncertainty of a random variable; the acquisition of information corresponds to a reduction in entropy.

$$H(V) = -\sum_{k} P(v_k) log_2 P(v_k)$$

- the entropy of a fair coin flip :

$$H(Fair) = -0.5 * log_2^{0.5} \quad - 0.5 * log_2^{0.5} = 1$$

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

$$H(Play) = -\frac{9}{14} * log_2^{\frac{9}{14}} \quad -\frac{5}{14} * log_2^{\frac{5}{14}} = 0.940$$

Information Gain

- $H(Y|X) = \sum_{x \in X} p(x) H(Y|X = x)$
- $\mathrm{IG(Y|X)} = H(Y) - H(Y|X)$

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

	YES	No	total
sunny	2	3	5
overcast	4	0	4
rain	3	2	5
total	9	5	14

outlook=sunny

outlook=overcast

outlook=rain

$$H(play|outlook) = \frac{5}{14} * \left(-\frac{2}{5} * log_2^{\frac{2}{5}} - \frac{3}{5} * log_2^{\frac{3}{5}}\right) + \frac{4}{14} * \left(-\frac{4}{4} * log_2^{\frac{4}{4}} - 0 * log_2^{0}\right) + \frac{5}{14} * \left(-\frac{3}{5} * log_2^{\frac{3}{5}} - \frac{2}{5} * log_2^{\frac{2}{5}}\right)$$

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

Outlook:

	YES	No	total
sunny	2	3	5
overcast	4	0	4
rain	3	2	5
total	9	5	14

Temperature:

	YES	No	total
hot	2	2	4
mid	4	2	6
cool	3	1	4
total	9	5	14

Humidity:

	YES	No	total
high	3	4	7
normal	6	1	7
total	9	5	14

Windy:

	YES	No	total
false	6	2	8
true	3	3	6
total	9	5	14

Question 3.(a)

(a) Using entropy to analysis which attribute should be root node among outlook, temperature, humidity and windy? Write your analysis process in question .

Answer:

$$H(play) = -\frac{9}{14} * log_2^{\frac{9}{14}} \quad -\frac{5}{14} * log_2^{\frac{5}{14}} = 0.940$$

$$H(play|outlook) = \frac{5}{14} * \left(-\frac{2}{5} * log_2^{\frac{2}{5}} - \frac{3}{5} * log_2^{\frac{3}{5}}\right) + \frac{4}{14} * \left(-\frac{4}{4} * log_2^{\frac{4}{4}} - 0 * log_2^{0}\right) + \frac{5}{14} * \left(-\frac{3}{5} * log_2^{\frac{3}{5}} - \frac{2}{5} * log_2^{\frac{2}{5}}\right)$$

$$= \frac{5}{14} * 0.971 + \frac{4}{14} * 0.0 + \frac{5}{14} * 0.971 = 0.693$$

$$H(play|temperature) = \frac{4}{14} * \left(-\frac{2}{4} * log_2^{\frac{2}{4}} - \frac{2}{4} * log_2^{\frac{2}{4}}\right) + \frac{6}{14} * \left(-\frac{4}{6} * log_2^{\frac{4}{6}} - \frac{2}{6} * log_2^{\frac{2}{6}}\right) + \frac{4}{14} * \left(-\frac{3}{4} * log_2^{\frac{3}{4}} - \frac{1}{4} * log_2^{\frac{1}{4}}\right) = 0.911$$

$$H(play|humidity) = \frac{7}{14} * \left(-\frac{3}{7} * log_2^{\frac{3}{7}} - \frac{4}{7} * log_2^{\frac{4}{7}}\right) + \frac{7}{14} * \left(-\frac{6}{7} * log_2^{\frac{6}{7}} - \frac{1}{7} * log_2^{\frac{1}{7}}\right) = 0.788$$

$$H(play|windy) = \frac{8}{14} * \left(-\frac{6}{8} * log_2^{\frac{6}{8}} - \frac{2}{8} * log_2^{\frac{2}{8}}\right) + \frac{6}{14} * \left(-\frac{3}{6} * log_2^{\frac{3}{6}} - \frac{3}{6} * log_2^{\frac{3}{6}}\right) = 0.892$$

IG(play|outlook) = 0.940-0.693 = 0.247 IG(play|temperature) = 0.029

IG(play|humidity) = 0.152 IG(play|windy)= 0.048

Choose outlook, because outlook provides maximum information gain.

Question 3.(b)

(b) Draw the decision tree.

How to construct a DT?

Training methods: Heuristic Search (Tree Search)

1. Start from the root, keep searching for a rule to branch a node.

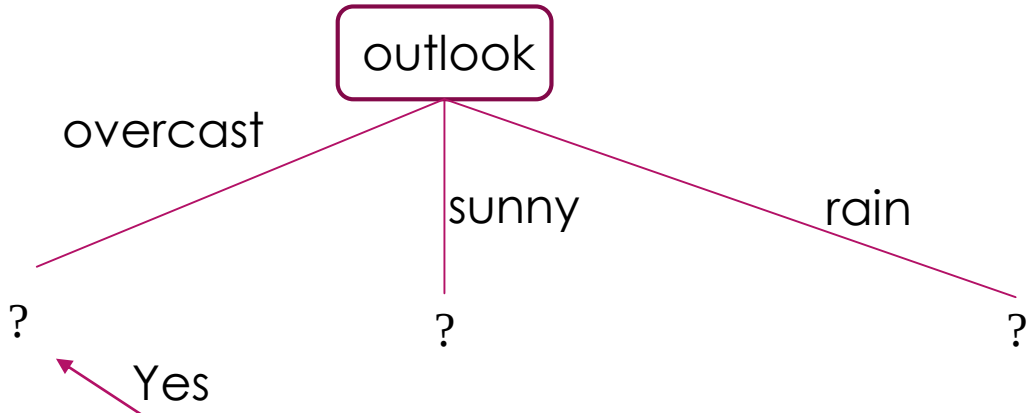
2. At each node, select the rule that leads to the most significant decrease in impurity (similar to gradient descent).

$$\Delta i(N) = i(N) - p_L i(N_L) - (1 - p_L) i(N_R)$$

3. When the process terminates, assign class label to the leaf nodes.
   - label a leaf node with the label of majority instances that fall into it.

Question 3.(b)

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No



Outlook:

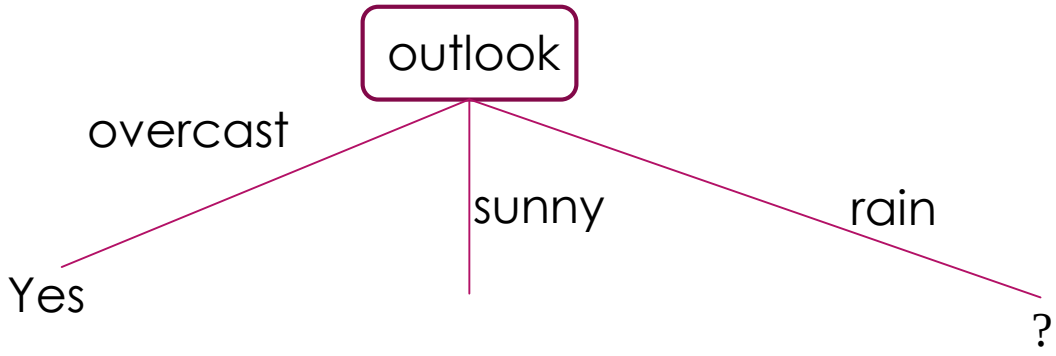
	Yes	No	total
sunny	2	3	5
overcast	4	0	4
rain	3	2	5
total	9	5	14

$$H(play|outlook=sunny) = -\frac{2}{5} * log_2^{\frac{2}{5}} - \frac{3}{5} * log_2^{\frac{3}{5}} = 0.971$$

$$H(play|outlook=overcast) = -\frac{4}{4} * log_2^{\frac{4}{4}} - 0 * log_2^{0} = 0$$

$$H(play|outlook=rain) = -\frac{3}{5} * log_2^{\frac{3}{5}} - \frac{2}{5} * log_2^{\frac{2}{5}} = 0.971$$

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No



Temperature: √

	YES	No	total
hot	0	3	3
mid	0	1	1
cool	1	0	1
total	1	4	5

H(play|temperature, outlook=sunny)=0

Outlook = sunny:

	Yes	No	total
sunny	2	3	5

$H(play|outlook=sunny) = -\frac{2}{5} * log_2^{\frac{2}{5}} - \frac{3}{5} * log_2^{\frac{3}{5}}$

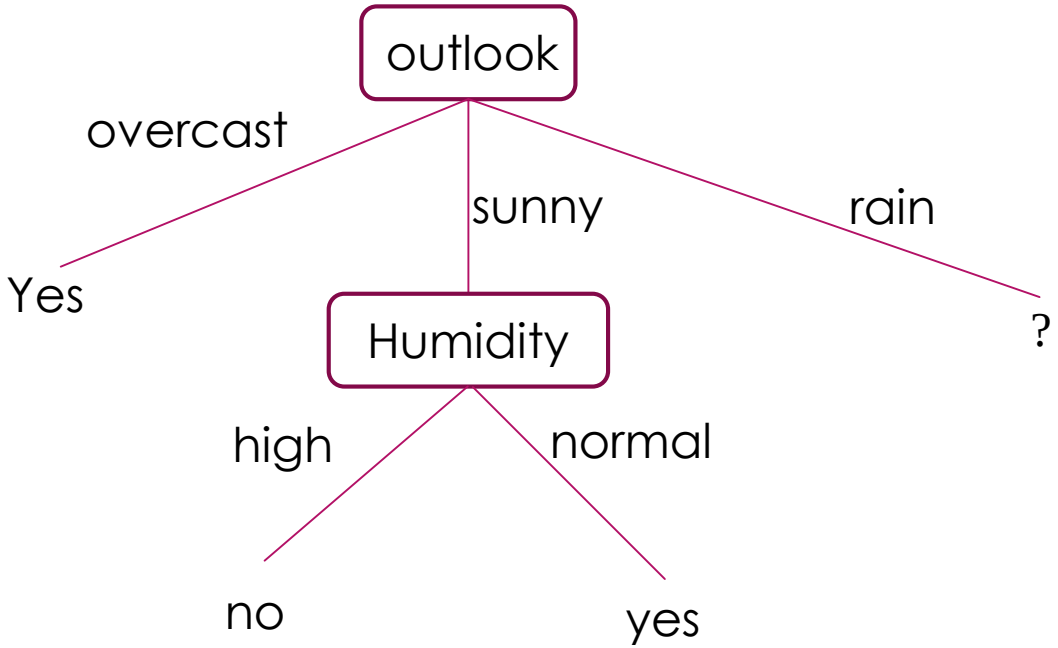
Windy:

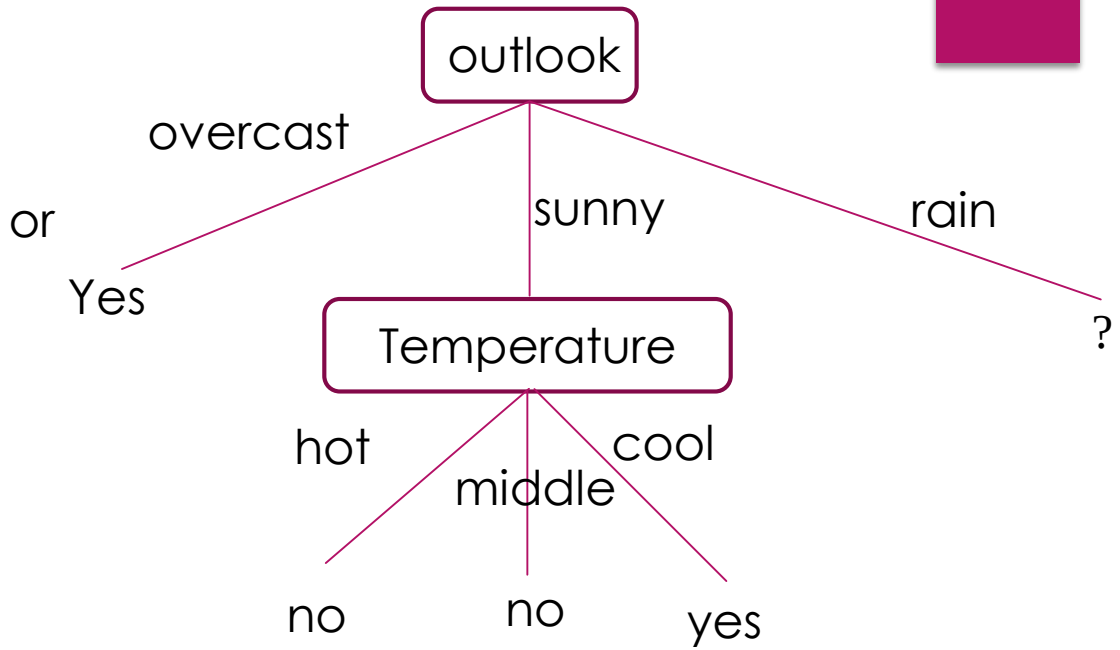
	YES	No	total
false	1	2	3
true	1	1	2
total	2	3	5

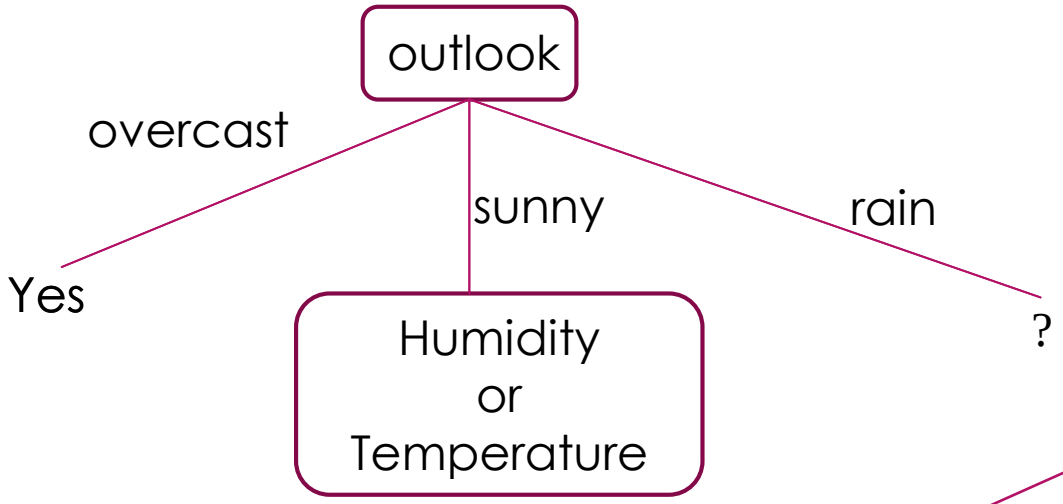
Humidity: √

	YES	No	total
high	0	3	3
normal	2	0	2
total	2	3	5

H(play| humidity, outlook=sunny)=0


outlook
overcast
sunny
rain
Yes
Humidity
?
high
normal
no
yes


outlook
overcast
or
sunny
rain
Yes
Temperature
?
hot
middle
cool
no
no
yes



Temperature:

	YES	No	total
hot	0	0	0
mid	2	1	3
cool	1	1	2
total	3	2	5

Outlook = rain:

	Yes	No	total
rain	3	2	5

$$H(play|outlook=rain) = -\frac{3}{5} * log_2^{\frac{3}{5}} - \frac{2}{5} * log_2^{\frac{2}{5}}$$

Humidity:

	YES	No	total
high	1	1	2
normal	2	1	3
total	2	3	5

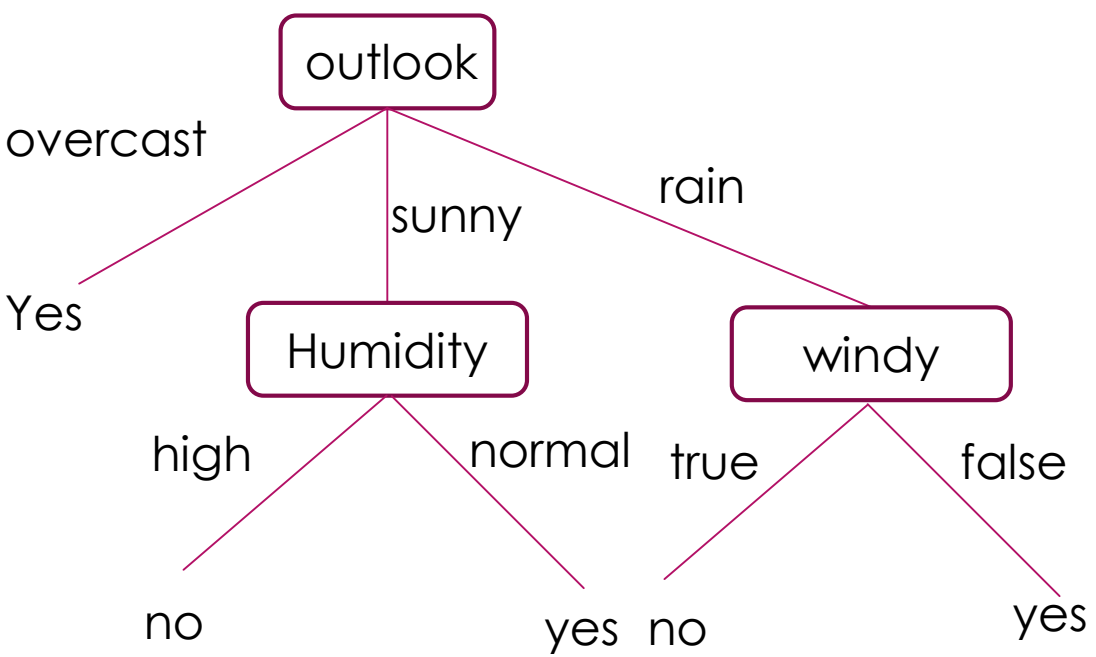
Windy:√

	YES	No	total
false	3	0	3
true	0	2	2
total	3	2	5

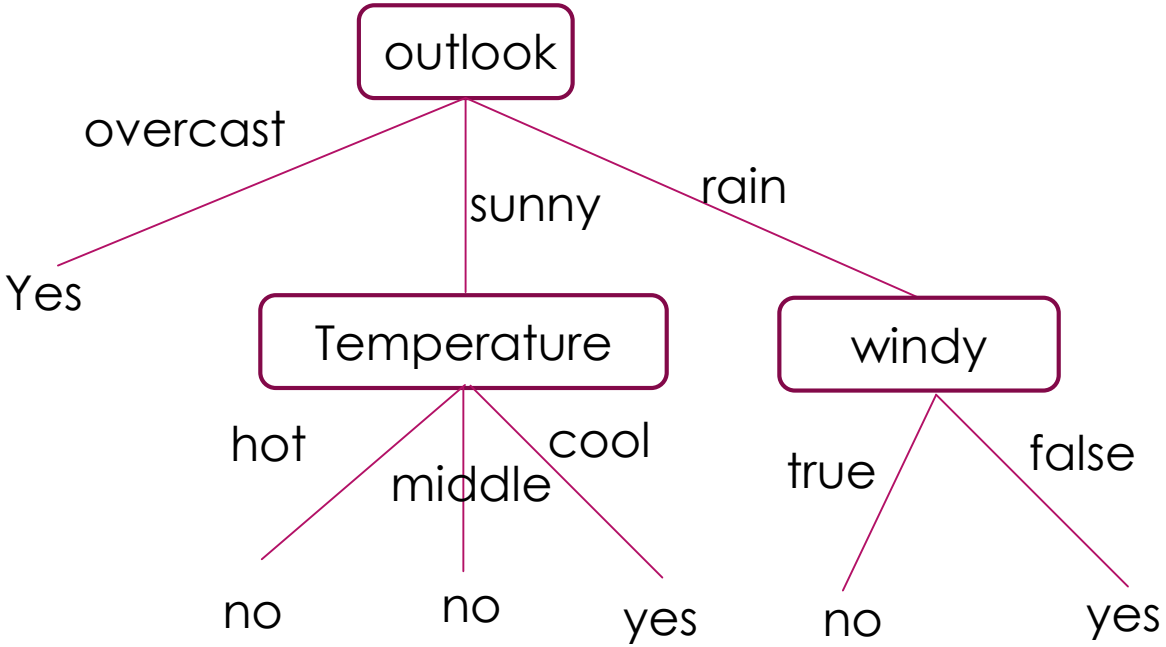
H(play| windy, outlook=rain)=0

Question 3.(b)

Answer:



or



Summary

- SVM
  - Support vectors
  - Margin maximization
  - decision boundary, plus-plane and minus-plane
- NN
  - Neural network forward computation
  - How to train NN?
  - Back Propagation
- DT
  - Problem formulation
  - How to measure the impurity?
  - How to construct a DT?