

CZ4041/CE4041: Machine Learning

Week 7: Artificial Neural Networks

Question 1

- In Lecture 7, we showed how to use backpropagation to update the parameters of ANN with one initialization setting for w .
- Suppose now we initialize w with another set of values: $w_{13} = -1$, $w_{14} = -1$, $w_{23} = -1$, $w_{24} = -1$, $w_{35} = -1$, and $w_{45} = -1$.
- Run one epoch (i.e., run through the whole training dataset once), to show how the parameters are updated at each iteration.

Review: Updating Weights for MLN

- Initialize the weights in each layer ($\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(k)}, \dots, \mathbf{w}^{(m)}$)
- Adjust the weights such that the output of ANN is consistent with class labels of training examples

- Loss function for each training instance:

$$\mathcal{L} = \frac{1}{2} (y_i - \hat{y}_i)^2$$

- For each layer k , update the weights, $\mathbf{w}^{(k)}$, by gradient descent at each iteration t :

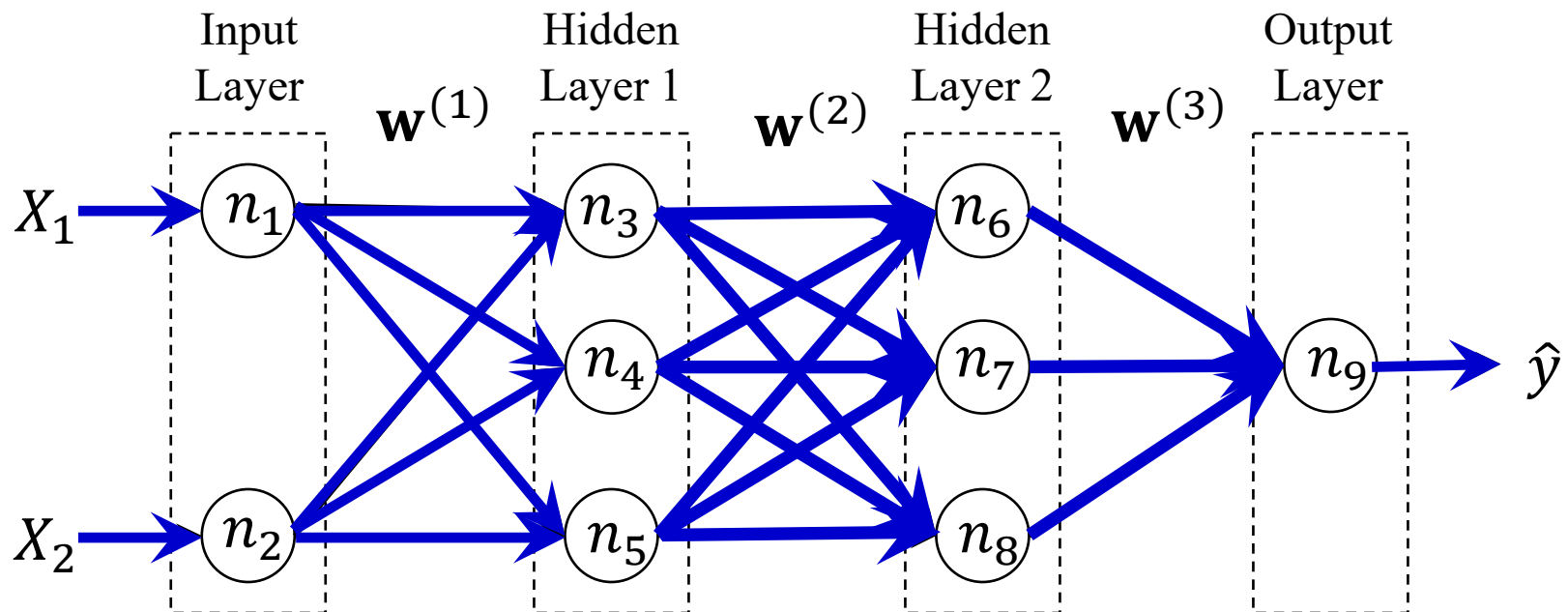
$$\mathbf{w}_{t+1}^{(k)} = \mathbf{w}_t^{(k)} - \lambda \frac{\partial \mathcal{L}}{\partial \mathbf{w}^{(k)}}$$

- Computing the gradient w.r.t. weights in each layer is computationally expensive!

The Backpropagation Algorithm

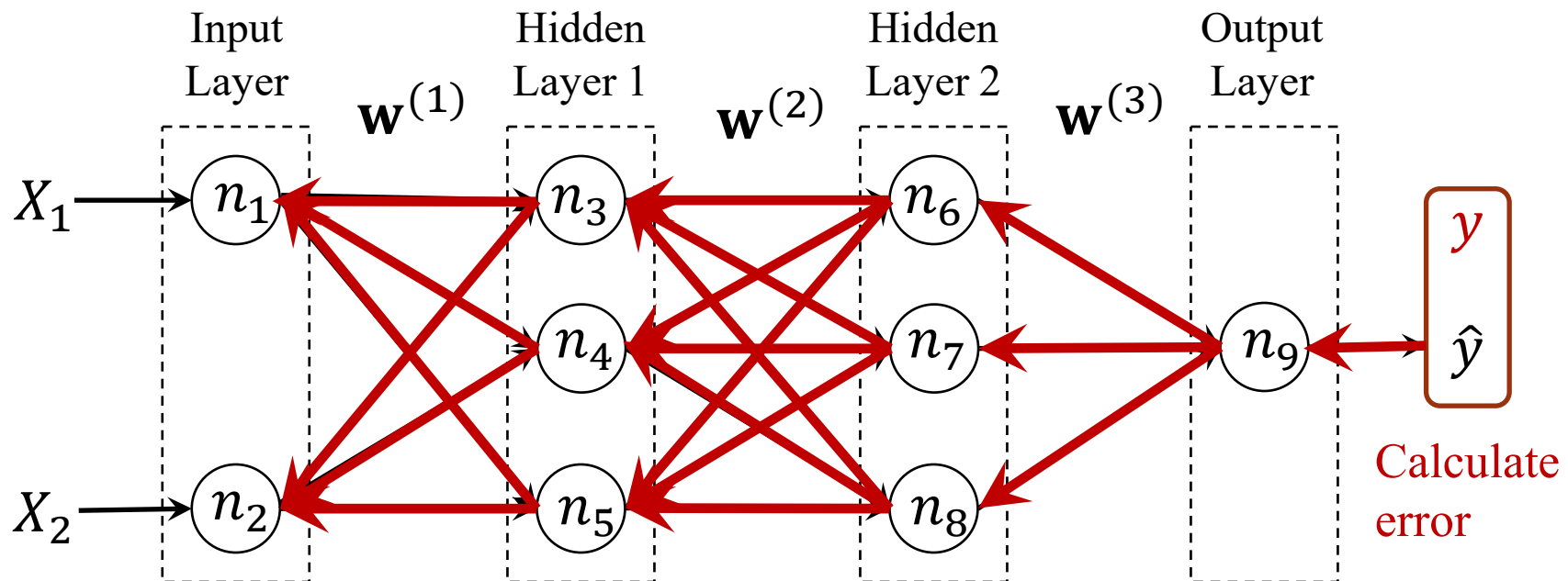
Review: Backpropagation

- Initialize the weights ($\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(3)}$)
- **Forward pass:** each training examples(\mathbf{x}_i, y_i) is used to compute outputs of each hidden layer and generate the final output \hat{y}_i based on the ANN



Review: Backpropagation

- **Backpropagation**: Starting with the output layer, to propagate error back to the previous layer in order to update the weights between the two layers, until the earliest hidden layer is reached



Review: Backpropagation

- Gradient of \mathcal{L} w.r.t. $w^{(3)}$: $\frac{\partial \mathcal{L}}{\partial w^{(3)}} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial w^{(3)}}$

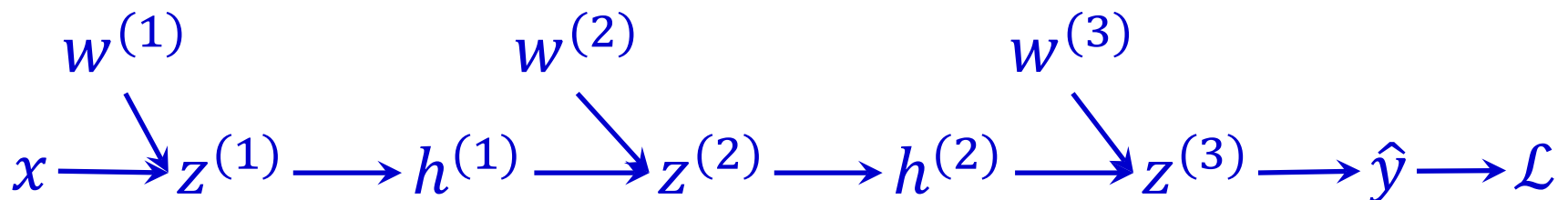
- Gradient of \mathcal{L} w.r.t. $w^{(2)}$:

$$\frac{\partial \mathcal{L}}{\partial w^{(2)}} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial h^{(2)}} \frac{\partial h^{(2)}}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial w^{(2)}}$$

- Gradient of \mathcal{L} w.r.t. $w^{(1)}$:

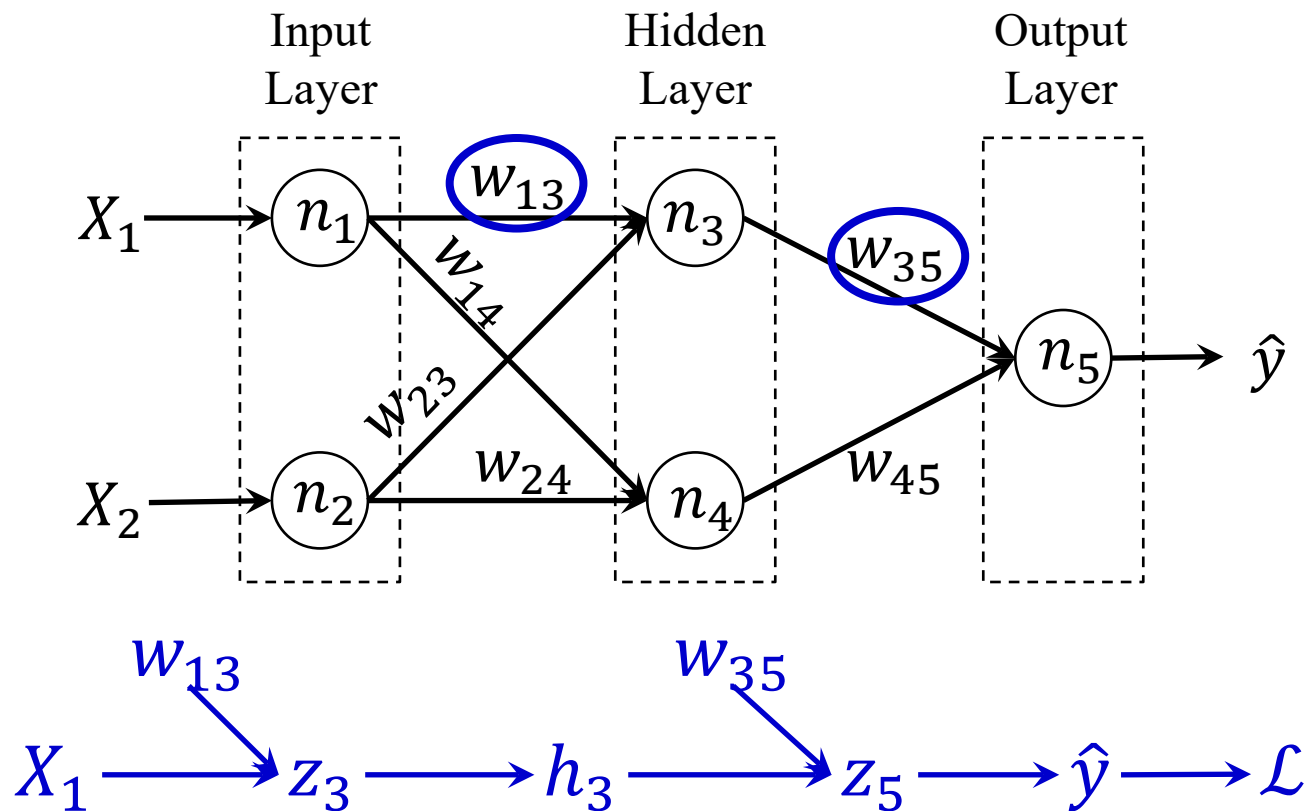
$$\frac{\partial \mathcal{L}}{\partial w^{(1)}} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial h^{(2)}} \frac{\partial h^{(2)}}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial h^{(1)}} \frac{\partial h^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial w^{(1)}}$$

Consider each layer contains a single unit



BP Algorithm Review

- Consider an ANN of 1 hidden layer as follows. Suppose the sign function and the weighted sum function are used for both hidden and output nodes

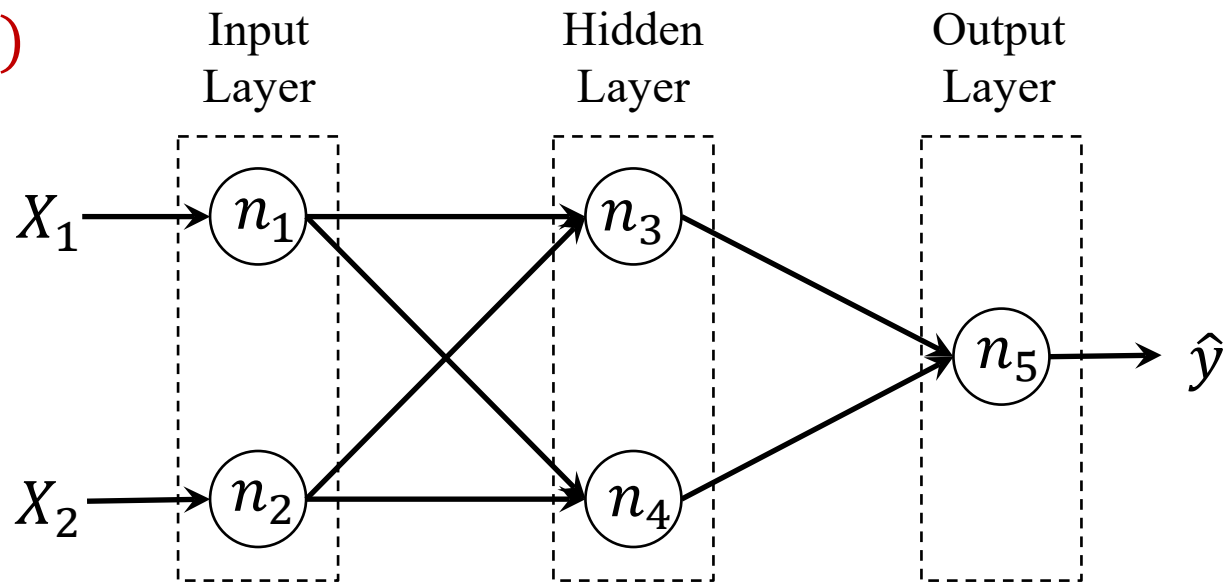


Question 1

$\lambda = 0.4, \theta = 0$

Sign function as $a(\cdot)$

| X_1 | X_2 | y |
|-------|-------|-----|
| 0 | 0 | -1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |



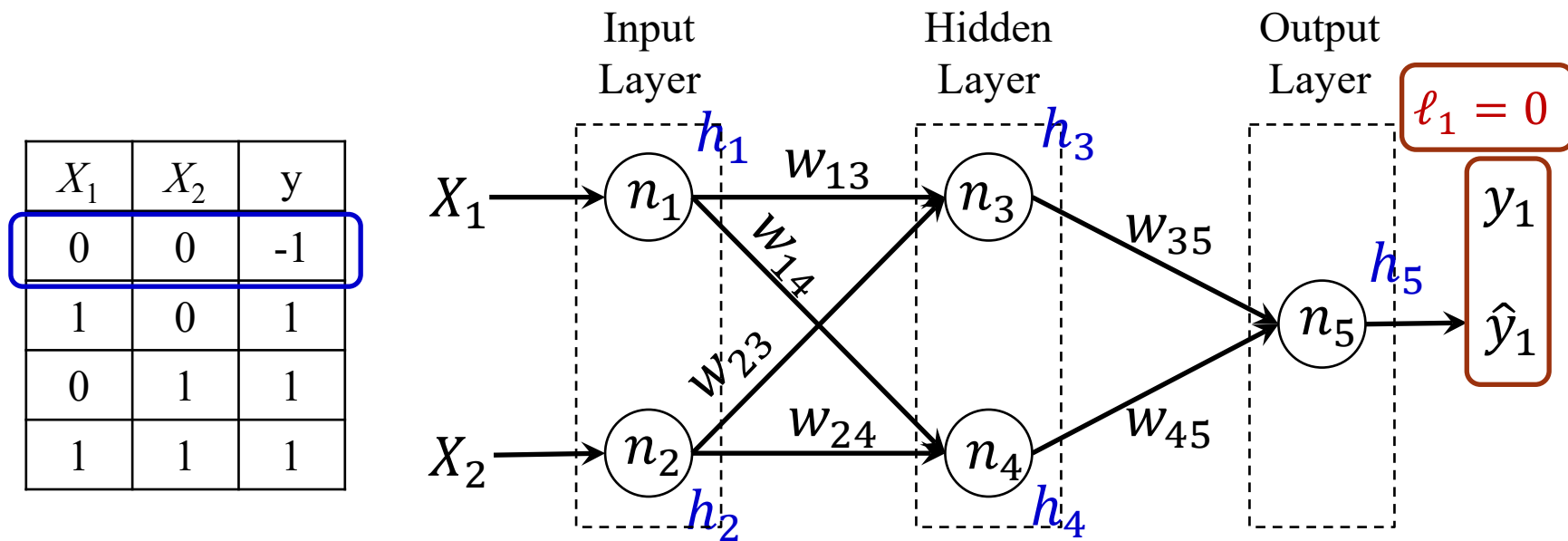
- Initialization 1 (lecture notes):

$$(w_{13} = 1, w_{14} = 1, w_{23} = 1, w_{24} = 1, w_{35} = 1, w_{45} = 1)$$

- Initialization 2:

$$(w_{13} = -1, w_{14} = -1, w_{23} = -1, w_{24} = -1, w_{35} = -1, w_{45} = -1)$$

BP Algorithm: Example (cont.)



Forward pass:

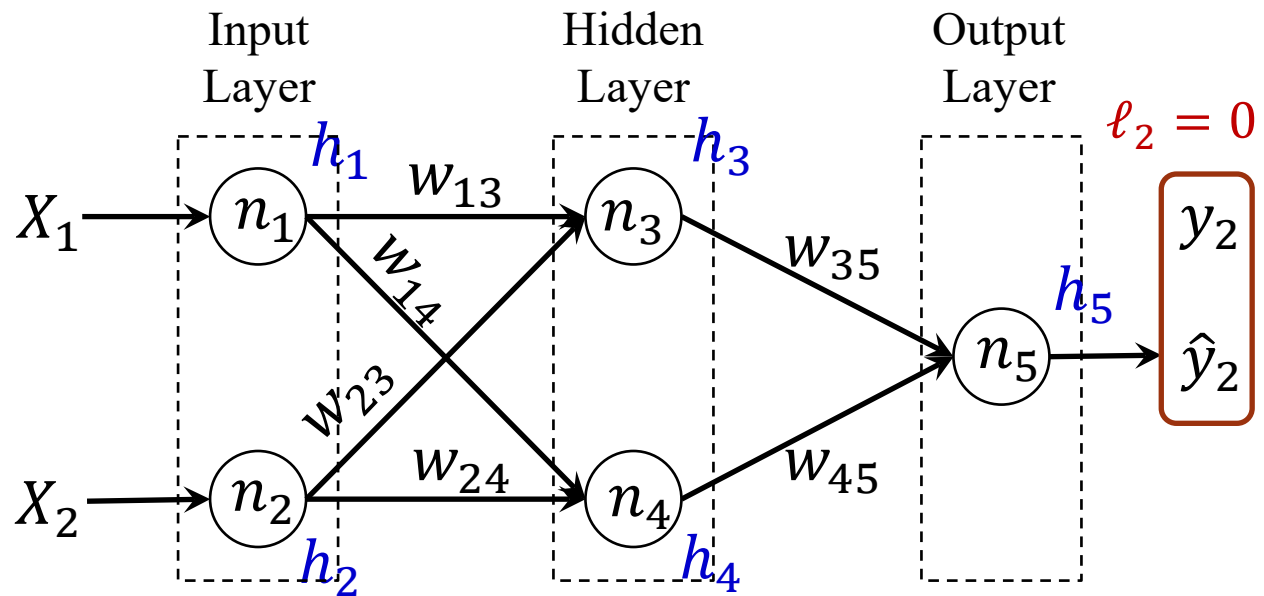
For the 1st example: $h_1 = 0$ and $h_2 = 0$

$h_3 = \text{sign}(0 \times (-1) + 0 \times (-1)) = 1$ and $h_4 = \text{sign}(0 \times (-1) + 0 \times (-1)) = 1$

Then $\hat{y}_1 = h_5 = \text{sign}(1 \times (-1) + 1 \times (-1)) = -1$

BP Algorithm: Example (cont.)

| X_1 | X_2 | y |
|-------|-------|-----|
| 0 | 0 | -1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |

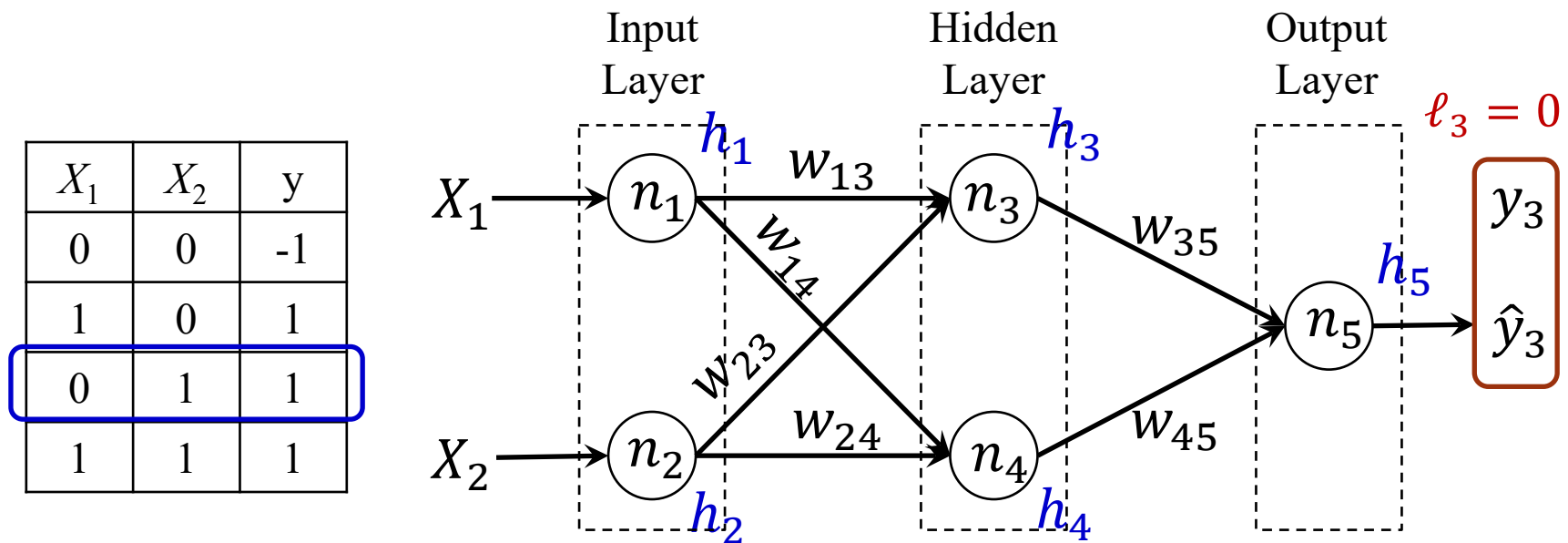


For the 2nd example: $h_1 = 1$ and $h_2 = 0$

$h_3 = \text{sign}(1 \times (-1) + 0 \times (-1)) = -1$ and $h_4 = \text{sign}(1 \times (-1) + 0 \times (-1)) = -1$

Then $\hat{y}_2 = h_5 = \text{sign}((-1) \times (-1) + (-1) \times (-1)) = 1$

BP Algorithm: Example (cont.)

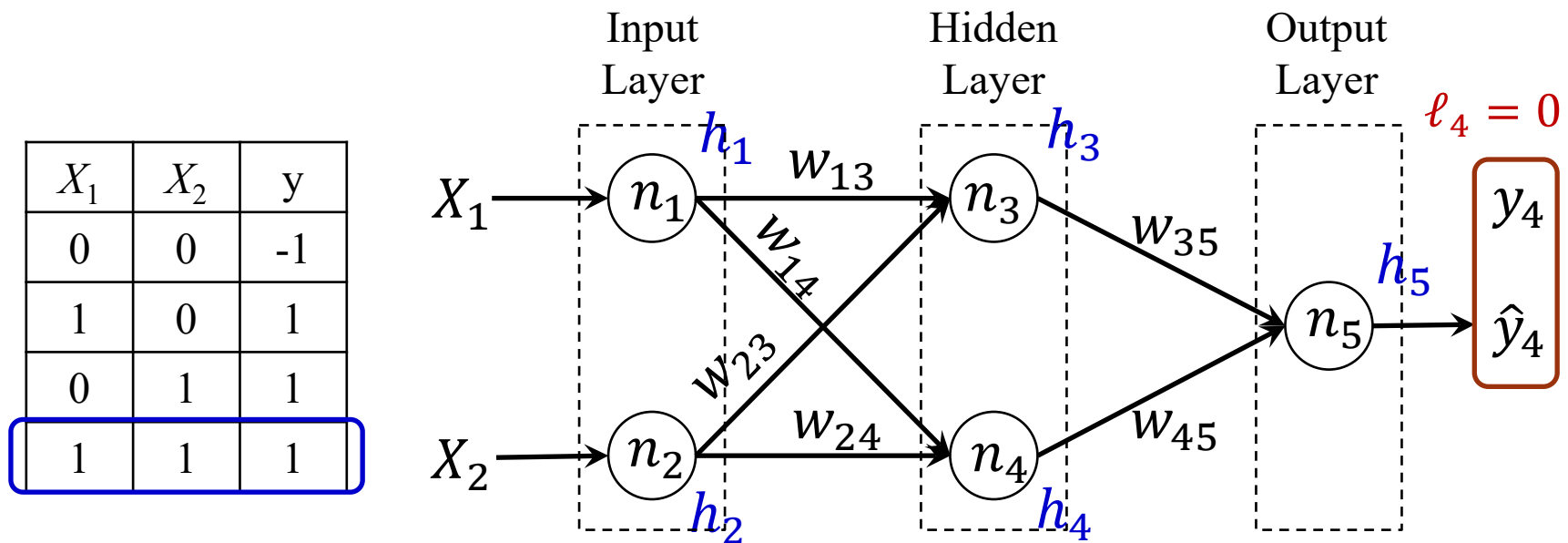


For the 3rd example: $h_1 = 0$ and $h_2 = 1$

$h_3 = \text{sign}(0 \times (-1) + 1 \times (-1)) = -1$ and $h_4 = \text{sign}(0 \times (-1) + 1 \times (-1)) = -1$

Then $\hat{y}_3 = h_5 = \text{sign}((-1) \times (-1) + (-1) \times (-1)) = 1$

BP Algorithm: Example (cont.)



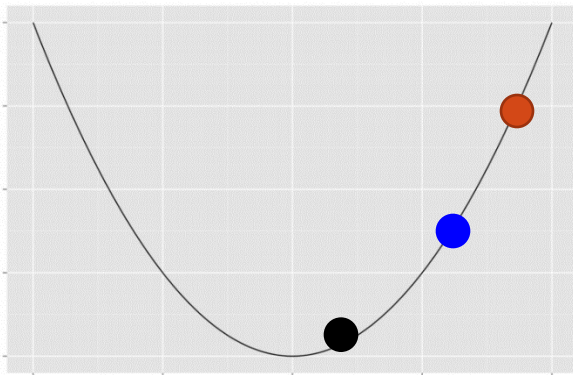
For the 4th example: $h_1 = 1$ and $h_2 = 1$

$h_3 = \text{sign}(1 \times (-1) + 1 \times (-1)) = -1$ and $h_4 = \text{sign}(1 \times (-1) + 1 \times (-1)) = -1$

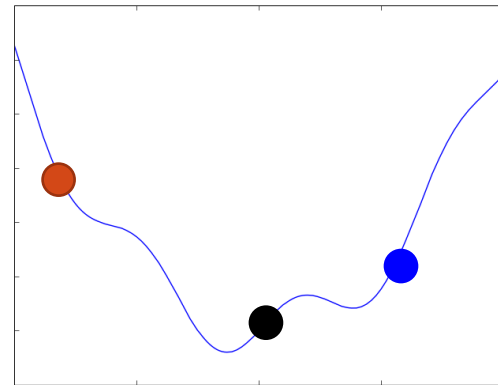
Then $\hat{y}_4 = h_5 = \text{sign}((-1) \times (-1) + (-1) \times (-1)) = 1$

Question 1: Conclusion

- Different initializations may lead to different convergence iterations
- Different initializations may lead to different local optima



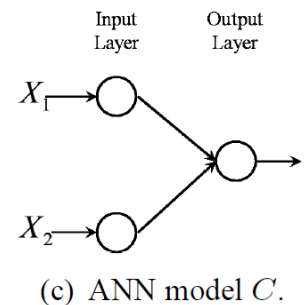
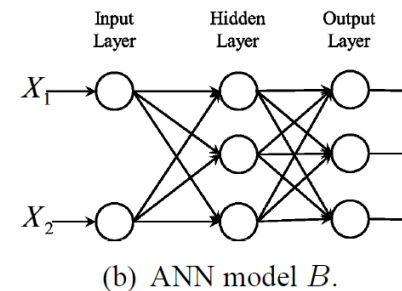
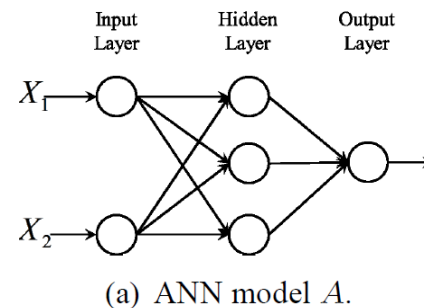
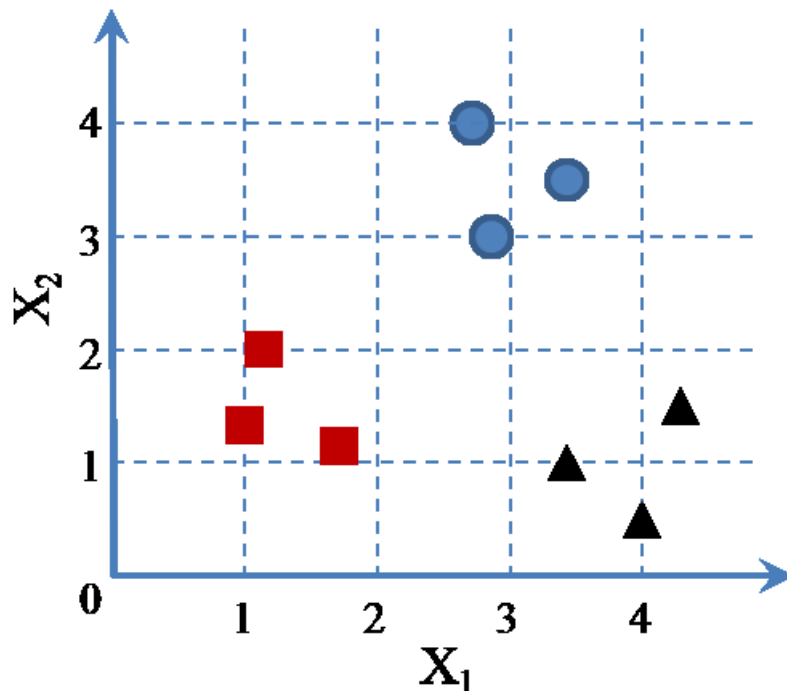
Convex



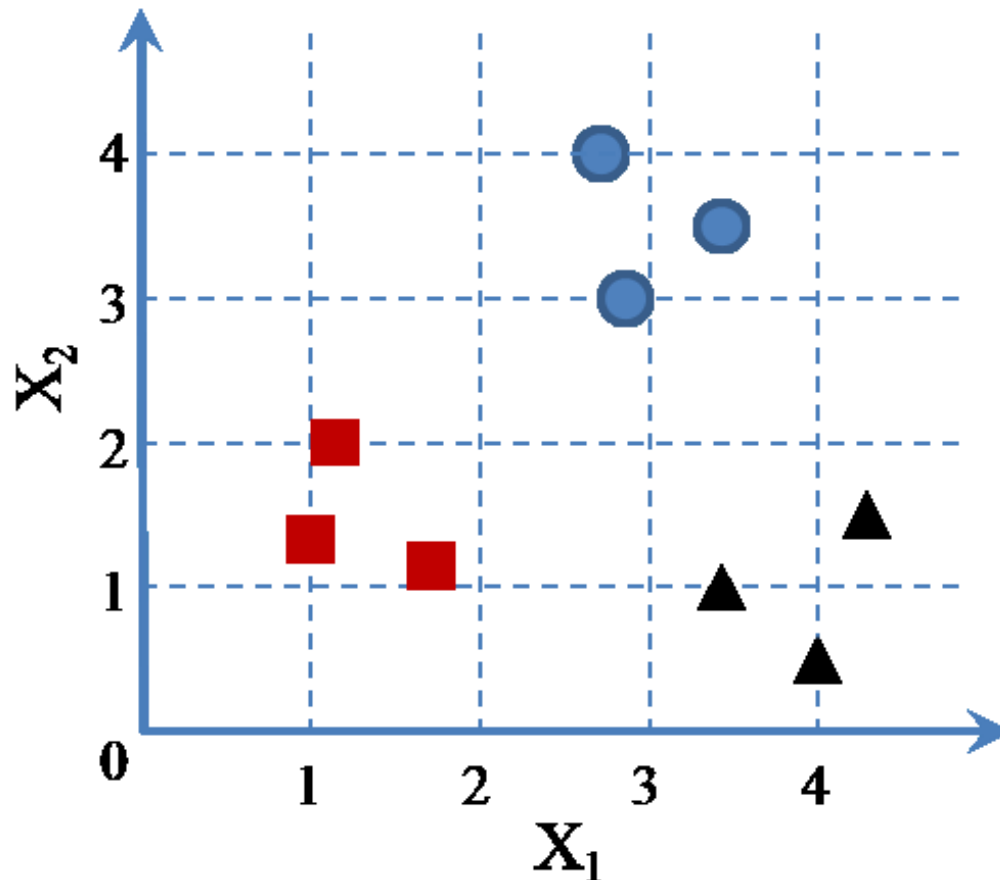
Non-convex

Question 2

- Consider a 2-dimensional dataset for three-class classification by ANN, as shown in Figure 1. Which ANN model as shown in Figure 2 is proper to solve the classification problem? Why?



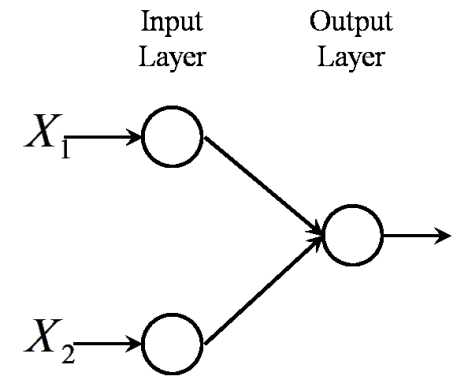
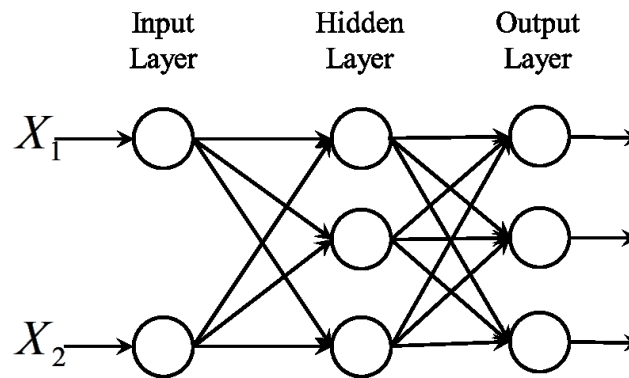
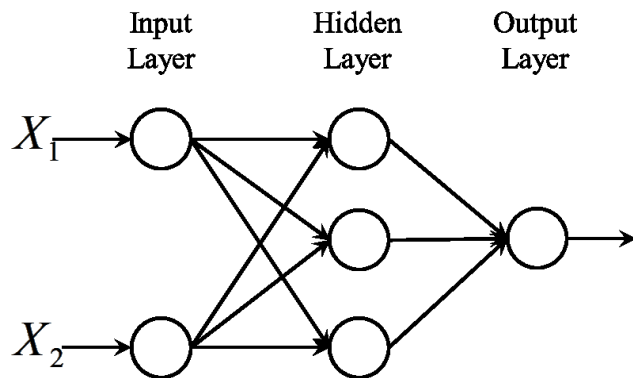
Question 2



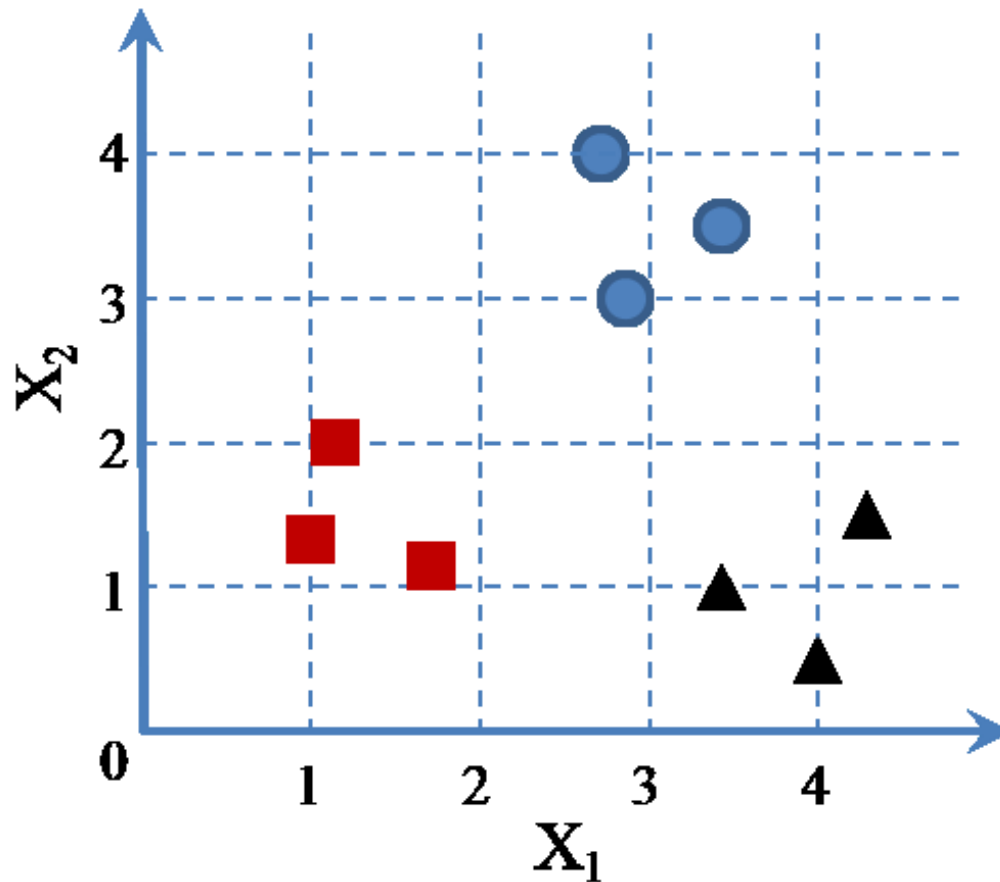
The instances of different classes are not able to be linearly separated

Therefore, the Perceptron model does not work

Question 2 (cont.)



Question 2 (cont.)



There are three
different classes

Design Issues for ANN

- The number of nodes in the input layer
 - Assign an input node to each numerical or binary input variable
- The number of nodes in the output layer
 - Binary class problem \rightarrow single node
 - C -class problem $\rightarrow C$ output nodes
- We output a one-hot encoding of the class
 - $[1, 0, 0]$ for class 1
 - $[0, 1, 0]$ for class 2

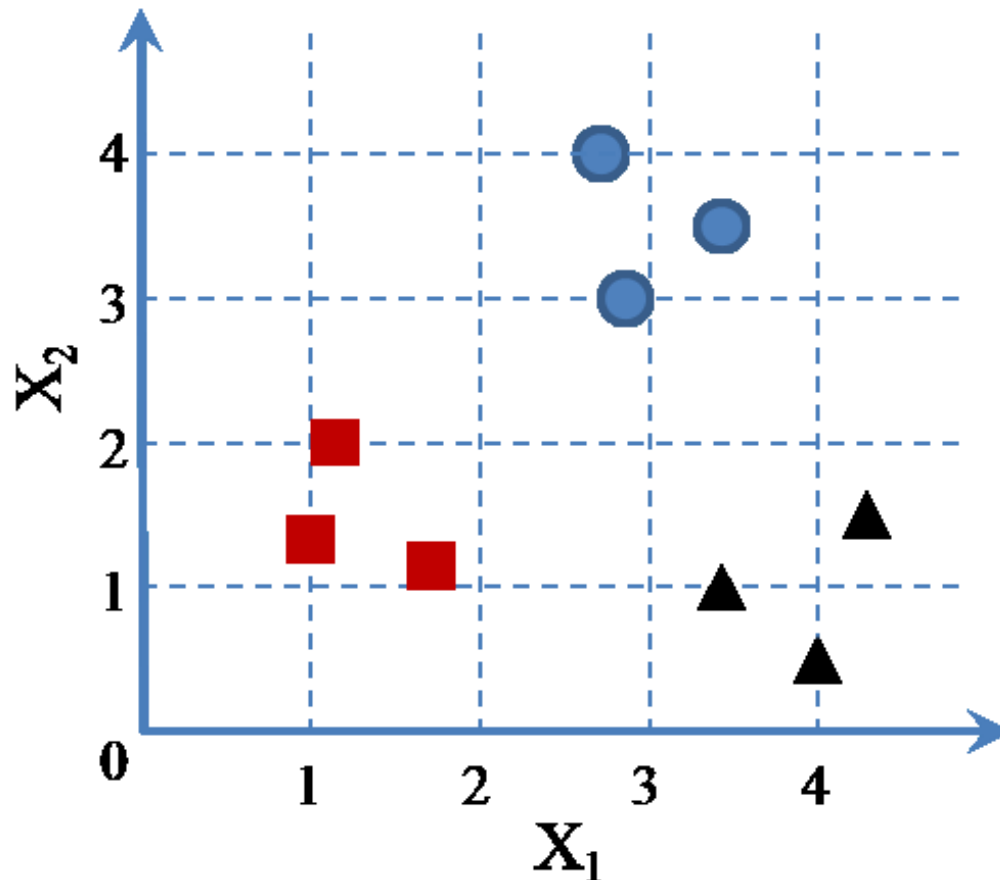
[Optional] Multi-class activation

- In a multi-class classification problem with C classes, we should output a one-hot encoding of the class
 - $[1, 0, 0]$ for class 1
 - $[0, 1, 0]$ for class 2
- We take the pre-activation outputs, $z_{L,1}, z_{L,2}$, and $z_{L,3}$, and feed them to a softmax function.

$$\hat{y}_i = \frac{\exp(z_{L,i})}{\sum_j \exp(z_{L,j})}$$

- This results in $1 > \hat{y}_i > 0$ and $\sum_i \hat{y}_i = 1$

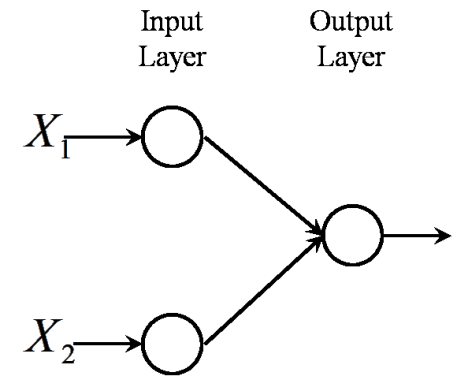
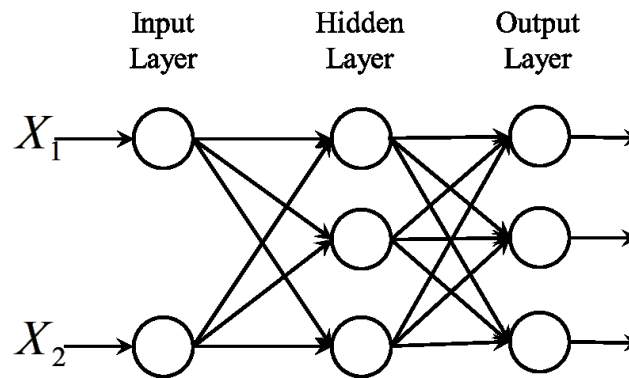
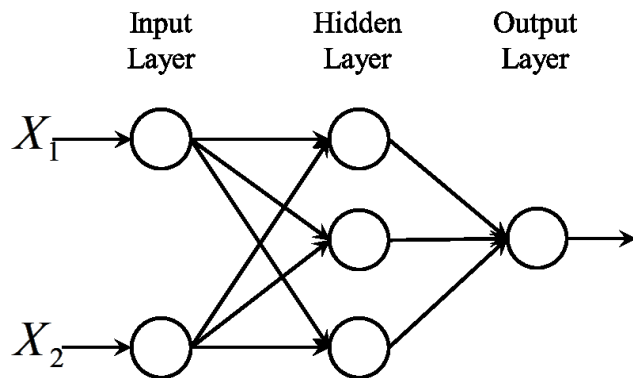
Question 2 (cont.)



There are three
different classes

Therefore, there
should be three
output nodes in the
output layer

Question 2 (cont.)



Question 3

- Compute the derivative of the sigmoid function w.r.t. z :


$$f(z) = \frac{1}{1 + e^{-z}}$$

- Denote $y = 1 + e^{-z}$, i.e., y is a function of z
- And then f is a function of y : $f = \frac{1}{y}$
- Based on the chain rule:

$$\frac{\partial f(z)}{\partial z} = \frac{\partial f}{\partial y} \frac{\partial y}{\partial z} = \left(\frac{-1}{y^2} \right) \left(\overset{0}{\cancel{\frac{\partial 1}{\partial z}}} + \overset{-e^{-z}}{\boxed{\frac{\partial e^{-z}}{\partial z}}} \right) = \frac{e^{-z}}{y^2}$$

Question 3 (cont.)

$$\frac{\partial f(z)}{\partial z} = \frac{\partial f}{\partial y} \frac{\partial y}{\partial z} = \frac{e^{-z}}{y^2}$$

$$y = 1 + e^{-z}$$


$$\frac{\partial f(z)}{\partial z} = \frac{e^{-z}}{(1 + e^{-z})^2} = \frac{1}{1 + e^{-z}} \left(\frac{e^{-z}}{1 + e^{-z}} \right)$$

$$\boxed{f(z) = \frac{1}{1 + e^{-z}}} = f(z) \left(1 - \frac{1}{1 + e^{-z}} \right)$$

$$= f(z)(1 - f(z))$$