

1. **Reticular Theory (1871–1873)**
   * **Proposed by**: Joseph von Gerlach
   * **Concept**: Suggested the nervous system is a single, continuous network.
2. **Staining Technique (1871–1873)**
   * **Discovered by**: Camillo Golgi
   * **Contribution**: Developed a chemical staining method, aiding visualization of nervous tissue and supporting Reticular Theory.
3. **Neuron Doctrine (1888–1891)**
   * **Proposed by**: Santiago Ramón y Cajal
   * **Concept**: The nervous system is composed of distinct cells (neurons), contradicting the Reticular Theory.
4. **The Term Neuron (1891)**
   * **Coined by**: Heinrich Wilhelm von Waldeyer-Hartz
   * **Contribution**: Named individual nerve cells "neurons" and advocated for the Neuron Doctrine.
5. **Nobel Prize (1906)**
   * **Awarded to**: Golgi and Cajal
   * **Significance**: Recognized their contributions to understanding nervous system structure, despite opposing views.
6. **Confirmation of Neuron Doctrine (1950s)**
   * **Method**: Electron microscopy
   * **Outcome**: Provided definitive evidence that neurons are distinct cells connected by synapses.
7. **McCulloch–Pitts Neuron (1943)**
   * **Developed by**: Warren McCulloch and Walter Pitts
   * **Contribution**: Introduced a simplified model of neurons, foundational for neural network research.
8. **Perceptron (1957–1958)**
   * **Invented by**: Frank Rosenblatt
   * **Concept**: An artificial neuron model that could learn and make decisions.
9. **Limitations of Perceptrons (1969)**
   * **Outlined by**: Marvin Minsky and Seymour Papert
   * **Impact**: Highlighted the limitations of perceptrons, dampening enthusiasm for neural networks.
10. **AI Winter (1969–1986)**
    * **Context**: Decline in AI research funding and interest, following unrealized expectations for neural networks.
11. **Gradient Descent (1847)**
    * **Discovered by**: Augustin-Louis Cauchy
    * **Concept**: A technique for minimizing errors in models, essential for training neural networks.
12. **Convolutional Neural Networks (1989)**
    * **Developed by**: Yann LeCun et al.
    * **Contribution**: Applied CNNs to digit recognition, foundational for image processing.
13. **Universal Approximation Theorem (1989)**
    * **Concept**: Neural networks with a single hidden layer can approximate any continuous function, highlighting their versatility.
14. **Long Short Term Memory (LSTM, 1997)**
    * **Developed by**: Hochreiter and Schmidhuber
    * **Contribution**: A type of RNN capable of learning long-term dependencies, addressing limitations of standard RNNs.
15. **Backpropagation (1986)**
    * **Popularized by**: Rumelhart et al.
    * **Significance**: Enhanced neural network training, boosting learning efficiency and effectiveness.
16. **McCulloch-Pitts Neuron (1943)**
    * **Developed by**: Warren McCulloch and Walter Pitts
    * **Contribution**: Introduced a mathematical model of neuron function, pioneering neural network concepts.
17. **Perceptron (1957–1958)**
    * **Invented by**: Frank Rosenblatt
    * **Concept**: An artificial neuron that could learn and make simple decisions.
18. **Limitations of Perceptrons (1969)**
    * **Outlined by**: Marvin Minsky and Seymour Papert
    * **Impact**: Emphasized limitations of perceptrons, causing a reduction in neural network research interest.
19. **AI Winter (1969–1986)**
    * **Context**: Period of decreased AI funding and interest due to perceived limitations of early models.
20. **Unsupervised Pre-Training (2006)**
    * **Developed by**: Geoffrey Hinton and Ruslan Salakhutdinov
    * **Idea**: Weight initialization technique improving deep network learning.
21. **Success in Handwriting Recognition (2009)**
    * **Achieved by**: Graves et al.
    * **Outcome**: Outperformed competitors in handwriting recognition, validating deep learning's effectiveness.
22. **Success in Speech Recognition (2010)**
    * **Achieved by**: Dahl et al.
    * **Outcome**: Significant error reduction over prior systems, marking a breakthrough in speech recognition.
23. **New Record on MNIST (2010)**
    * **Achieved by**: Ciresan et al.
    * **Outcome**: Set a new digit recognition benchmark with deep learning on the MNIST dataset.
24. **Superhuman Performance in Visual Recognition (2011)**
    * **Achieved by**: D. C. Ciresan et al.
    * **Outcome**: Achieved 0.56% error in traffic sign recognition, surpassing human performance.
25. **Success on ImageNet (2012–2016)**
    * **Key Models**: AlexNet, ZFNet, VGGNet, GoogLeNet, MS ResNet
    * **Outcome**: Demonstrated deep learning's power in image classification tasks, establishing benchmark models.
26. **Deep Reinforcement Learning (2015)**
    * **Achievement**: DQNs reached human-level control in Atari games.
27. **AlphaGo Zero (2015)**
    * **Achievement**: Outperformed human Go players without prior data, showing advanced AI strategy.
28. **DeepStack (2016)**
    * **Achievement**: Beat professional poker players, demonstrating AI’s potential in complex decision-making.
29. **Dota 2 Bot (2017)**
    * **Achievement**: Defeated top players, advancing AI’s strategic capabilities.
30. **Language Modeling Advancements (2010–2015)**
    * **Impact**: Improved models led to significant progress in natural language understanding.
31. **Generative Models (2013–2017)**
    * **Key Techniques**: Development of VAEs and GANs, enabling realistic data generation.
32. **Explainability in AI (2017)**
    * **Focus**: Addressed transparency and reliability in AI, promoting interpretability in deep learning models.

**Convolutional Neural Networks (CNN)**

↪ Developed by: Yann LeCun et al.

↪ Contribution: CNNs are designed for processing structured grid data, such as images. They utilize convolutional layers to automatically learn spatial hierarchies of features, making them particularly effective for tasks like image classification and recognition. The architecture typically includes convolutional layers, pooling layers, and fully connected layers, allowing the network to learn increasingly abstract representations of the input data.

Better Optimization Methods

↪ 1983: Nesterov introduced methods that improved convergence speed and accuracy in optimization.

↪ 2011: Adagrad was developed, allowing for adaptive learning rates based on the parameters' historical gradients.

↪ 2012: RMSProp was introduced, addressing the diminishing learning rates problem in Adagrad.

↪ 2015: Adam was developed, combining the benefits of both Adagrad and RMSProp for efficient training of deep learning models.

↪ 2016: Eve was introduced as an optimization method that further improved training efficiency.

↪ 2018: Beyond Adam methods were explored, continuing the trend of enhancing optimization techniques in deep learning.

**The Curious Case of Sequences**

↪ Context: Sequences are prevalent in various domains such as time series, speech, music, text, and video. Each unit in a sequence interacts with other units, necessitating models that can capture these interactions effectively.

↪ Key Models:

↪ Hopfield Network (1982): A content↪addressable memory system for storing and retrieving patterns.

↪ Jordan Network (1986): A recurrent network where the output state of each time step is fed to the next time step, allowing for interactions between time steps.

↪ Elman Network (1990): Similar to the Jordan network but feeds the hidden state to the next time step.

↪ Long Short Term Memory (LSTMs, 1997): A type of RNN that can learn long↪term dependencies, addressing issues like exploding and vanishing gradients.

↪ Sequence To Sequence Learning (2014): Initial successes using RNNs/LSTMs for large↪scale sequence learning problems, introducing attention mechanisms that inspired further research.

**The Paradox of Deep Learning**

↪ Context: Despite its success, deep learning faces several challenges:

↪ High Capacity: Models are susceptible to overfitting due to their ability to memorize training data.

↪ Numerical Instability: Issues like vanishing and exploding gradients complicate training.

↪ Sharp Minima: The tendency to converge to sharp minima can lead to overfitting.

↪ Non↪Robustness: Deep learning models can be sensitive to small perturbations in input data.

↪ Current Focus: There is an increasing emphasis on explainability and theoretical justifications to understand why deep learning works so well, aiming to bring more clarity and sanity to the field.

**Advancements in Deep Learning from 2012 to 2016**

**1. Success on ImageNet (2012↪2016):**

↪ Deep learning models, particularly Convolutional Neural Networks (CNNs), achieved significant breakthroughs in image classification tasks, winning various visual recognition challenges.

↪ Notable architectures developed during this period include:

↪ AlexNet (2012): Achieved a top↪5 error rate of 16.0%.

↪ ZFNet (2013): Improved upon AlexNet with an error rate of 11.2%.

↪ VGGNet (2014): Further reduced the error rate to 7.3%.

↪ GoogLeNet (2014): Achieved an error rate of 6.7%.

↪ MS ResNet (2015): Set a new record with an error rate of 3.6% using a very deep architecture (152 layers).

**2. Advancements in Optimization Methods:**

↪ New optimization techniques were developed, including:

↪ Adam (2015): A widely used optimization algorithm that combines the benefits of Adagrad and RMSProp for efficient training.

↪ Eve (2016): Introduced as a further enhancement in optimization methods.

**3. Generative Models:**

↪ The introduction of Generative Adversarial Networks (GANs) in 2014 by Goodfellow et al. allowed for the generation of realistic data, significantly impacting areas like image synthesis.

**4. Natural Language Processing:**

↪ Deep learning techniques began to dominate tasks in natural language processing, including language modeling, machine translation, and conversational agents.

**Significance of the Neuron Doctrine**

↪ The Neuron Doctrine, proposed by Santiago Ramón y Cajal, established that the nervous system is composed of discrete individual cells (neurons) rather than a continuous network. This was a pivotal shift in understanding the structure and function of the nervous system.

**↪ Key Points:**

↪ It laid the foundation for modern neuroscience by emphasizing the role of individual neurons in transmitting signals.

↪ The doctrine helped clarify how neurons communicate through synapses, leading to a better understanding of neural networks in both biological and artificial contexts.

↪ The acceptance of the Neuron Doctrine marked the beginning of a more systematic study of neural circuits and their functions, influencing the development of artificial neural networks in computer science.

**Contribution of Backpropagation to Neural Networks**

↪ Backpropagation is a key algorithm used for training artificial neural networks, allowing them to learn from data by adjusting weights based on the error of predictions.

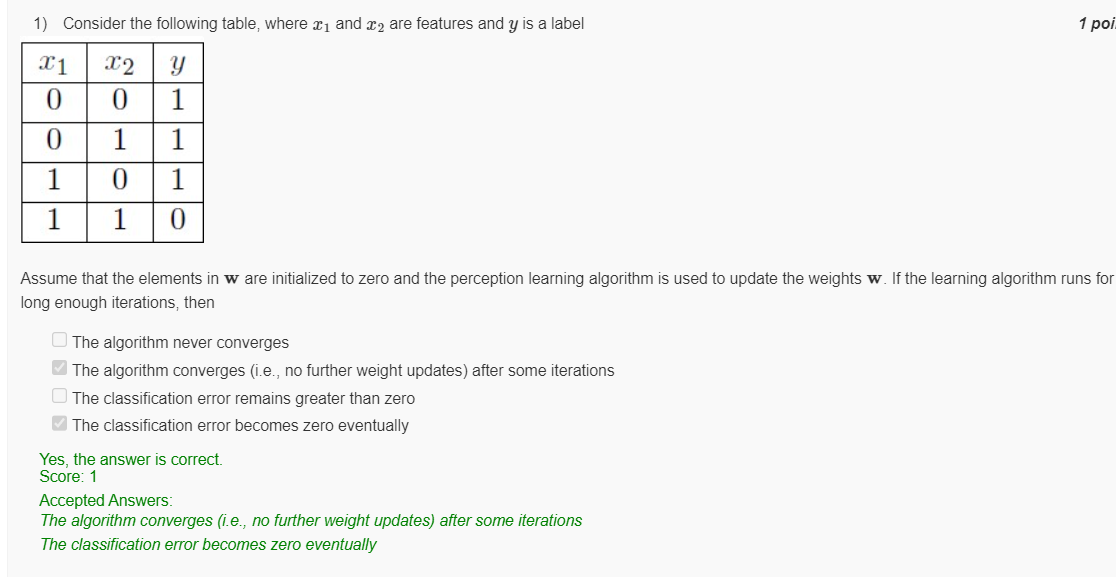
↪ Key Contributions:

↪ Error Minimization: Backpropagation computes the gradient of the loss function with respect to each weight by applying the chain rule, enabling the network to minimize the error in its predictions.

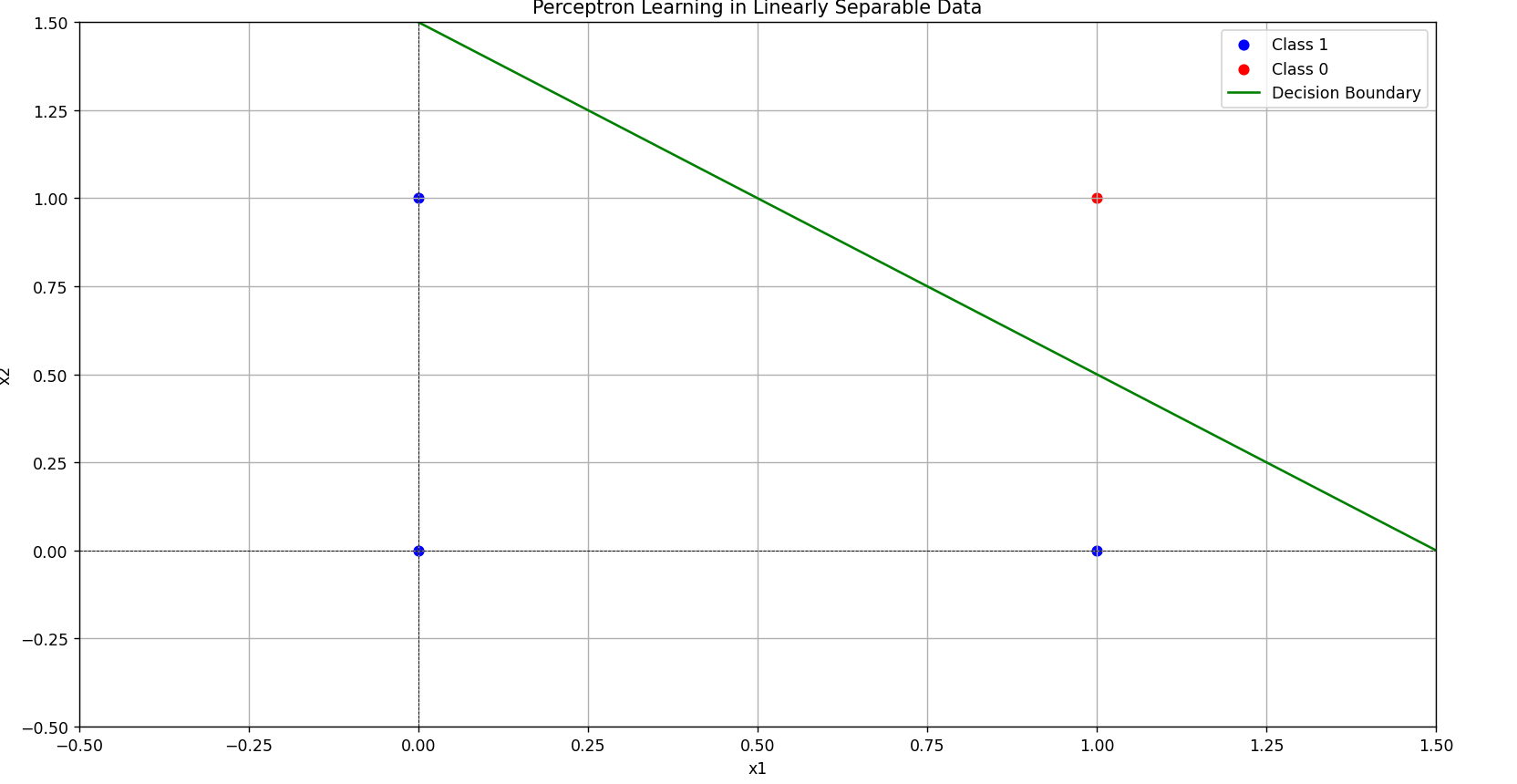
↪ Efficient Training: The algorithm allows for efficient computation of gradients, making it feasible to train deep networks with many layers.

↪ Foundation for Deep Learning: The rediscovery and popularization of backpropagation in the 1980s and its subsequent use in deep learning architectures have been crucial for the success of modern neural networks.

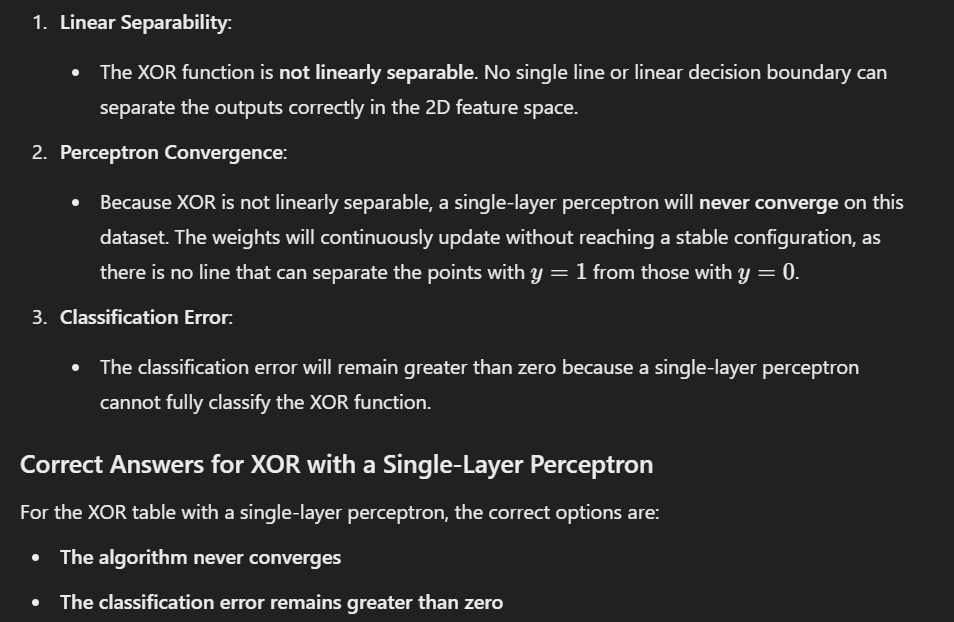
↪ Facilitated Complex Models: By enabling the training of multilayer networks, backpropagation has allowed for the development of complex models capable of capturing intricate patterns in data, leading to advancements in various fields such as computer vision, speech recognition, and natural language processing.



|  |  |  |
| --- | --- | --- |
| Assume that the elements in w are initialized to zero and the perceptron learning algorithm is used to update the weights w. If the learning algorithm runs for long enough iterations, then:   * The algorithm never converges * The algorithm converges (i.e., no further weight updates) after some iterations * The classification error remains greater than zero * The classification error becomes zero eventually |  |  |
| 1. **Linear Separability**: This particular table resembles the XOR pattern but differs in its label for the case (0,0)→1(0, 0) \rightarrow 1(0,0)→1. This specific label setup **is actually linearly separable**, unlike XOR, which has (0,0)→0(0, 0) \rightarrow 0(0,0)→0. 2. **Perceptron Convergence**:    * Since this configuration is linearly separable, the perceptron will converge after some iterations.    * The weights will eventually stabilize, reaching a point where no further updates are needed. 3. **Classification Error**:    * Once the perceptron converges, the classification error will become zero, as it can perfectly classify this linearly separable dataset.   **Conclusion:**  Given this table, the correct answers are:   * **The algorithm converges (i.e., no further weight updates) after some iterations** * **The classification error becomes zero eventually** |  |  |



For XOR

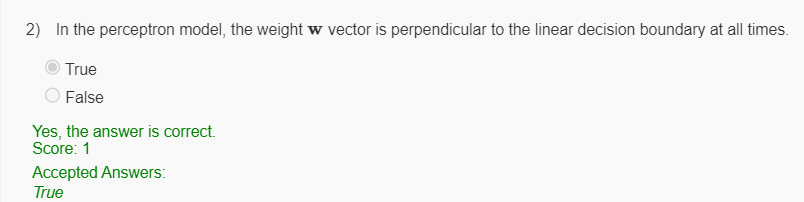


**Linearly Separable:**

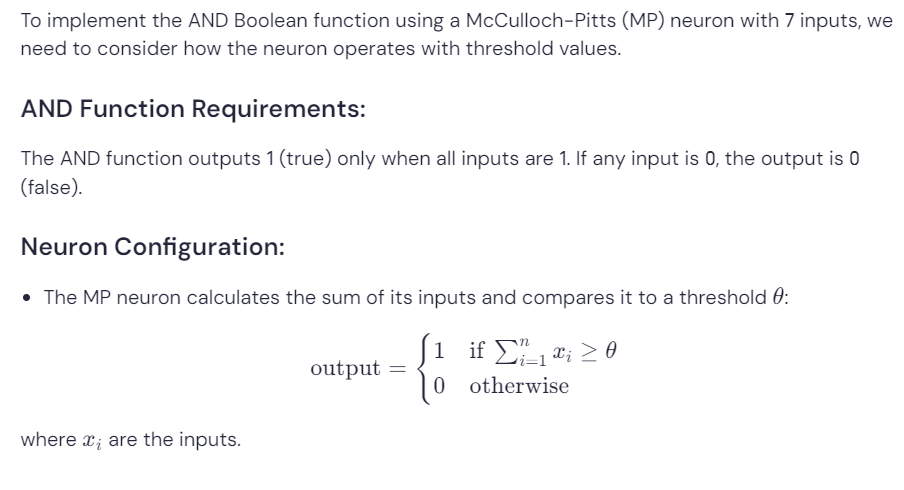
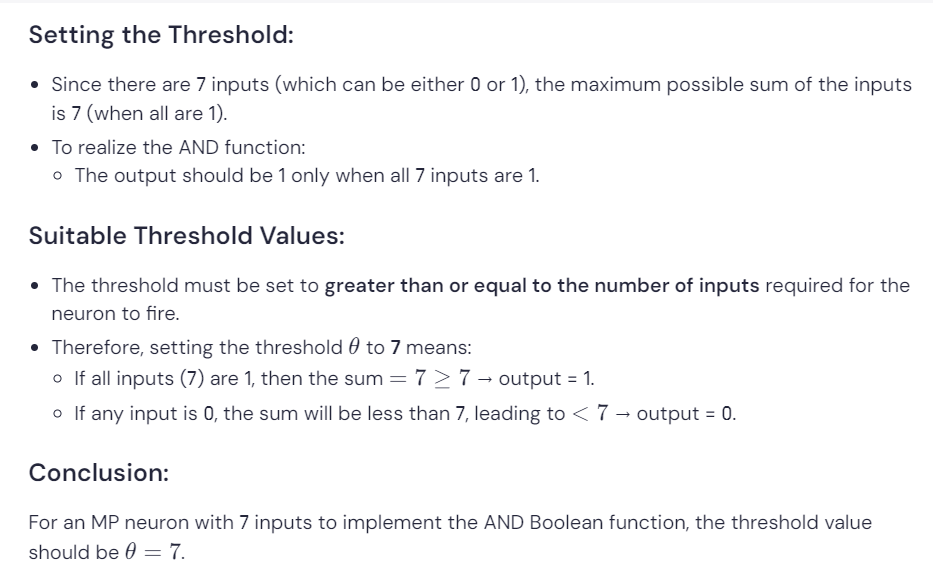
* **AND**
* **OR**
* **NAND**
* **NOR**

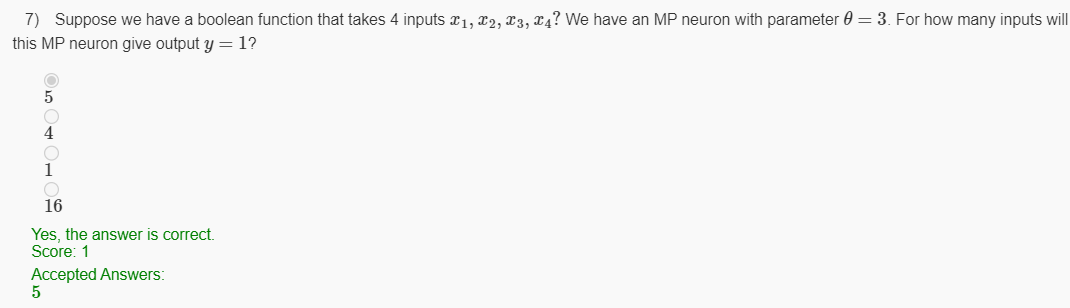
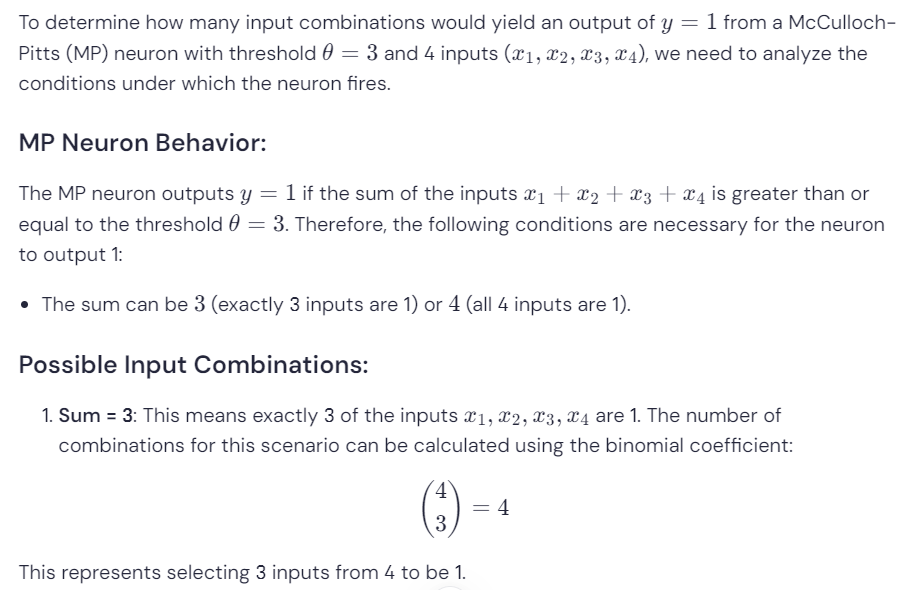
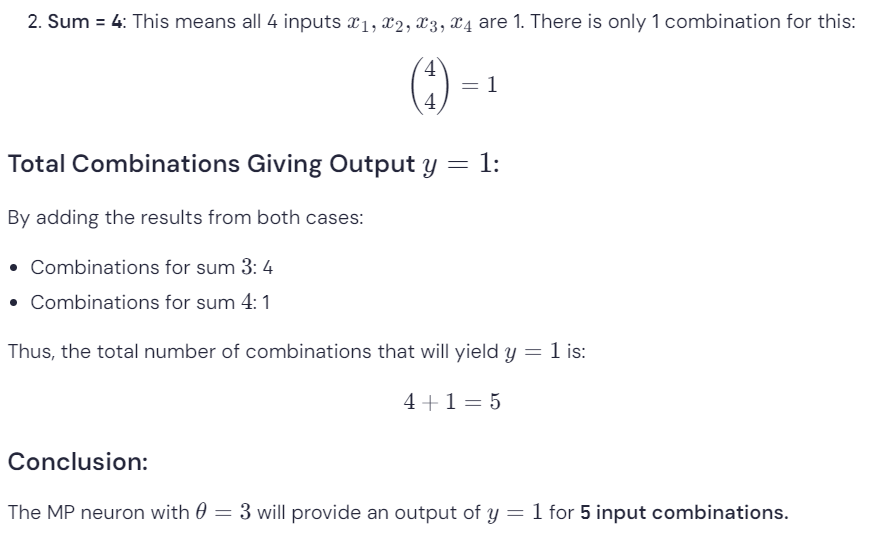
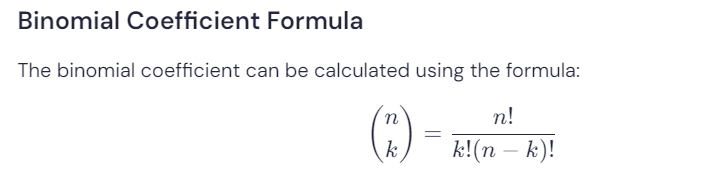
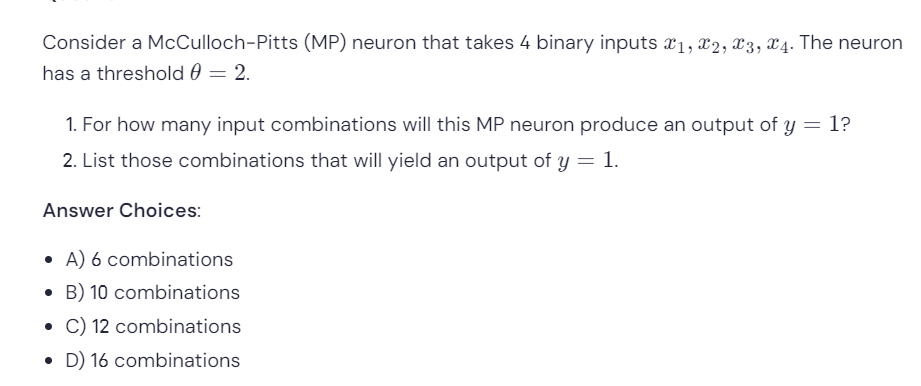
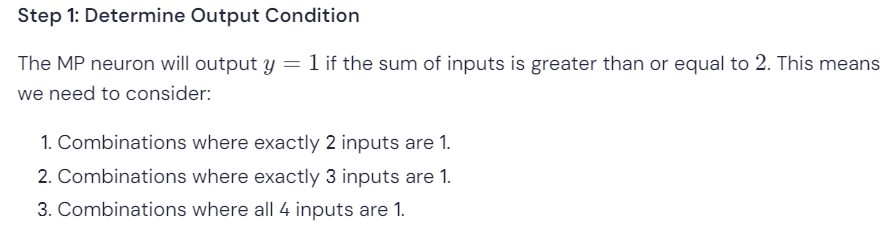
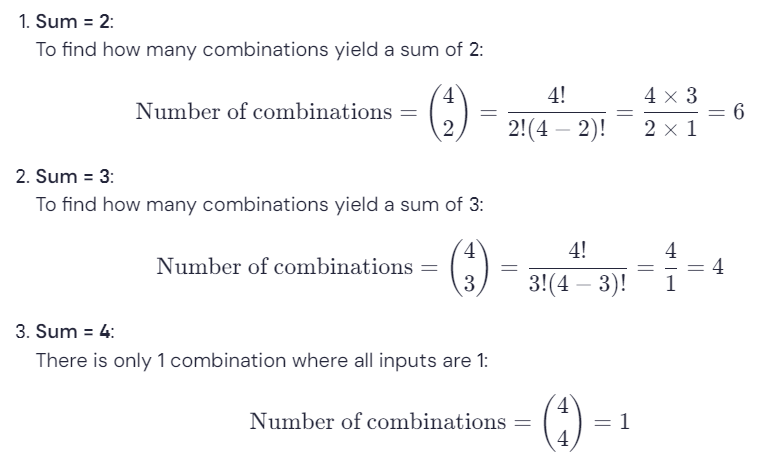
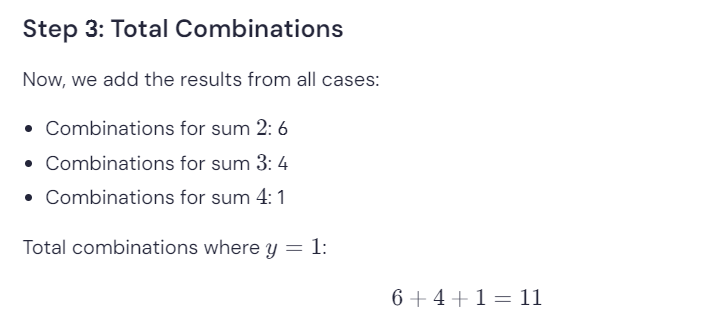
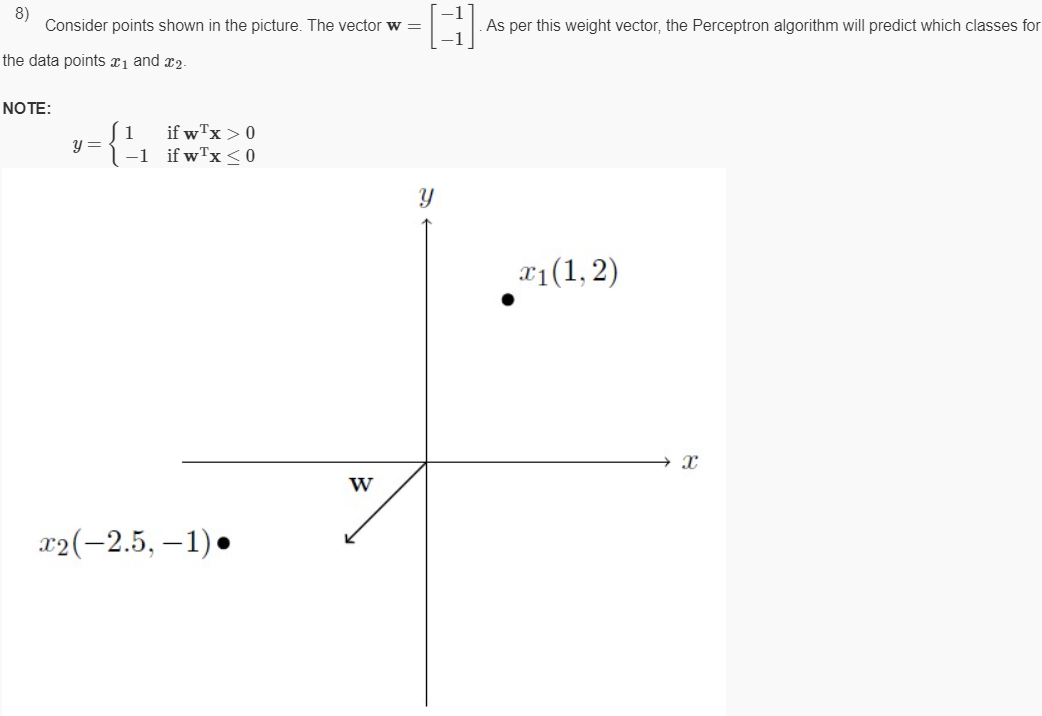
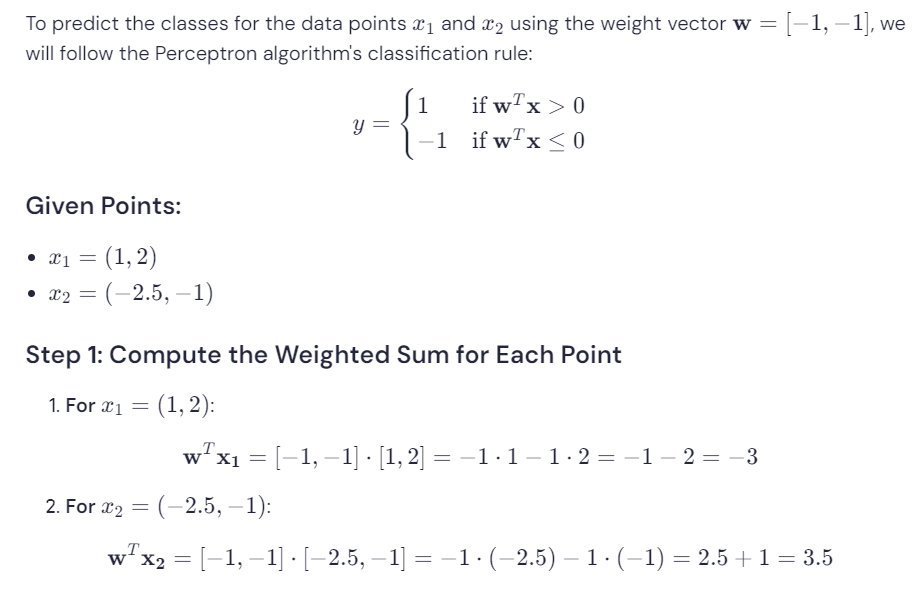
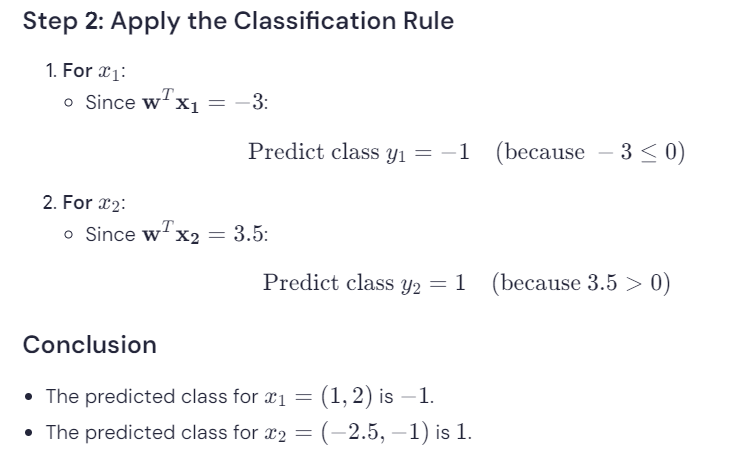
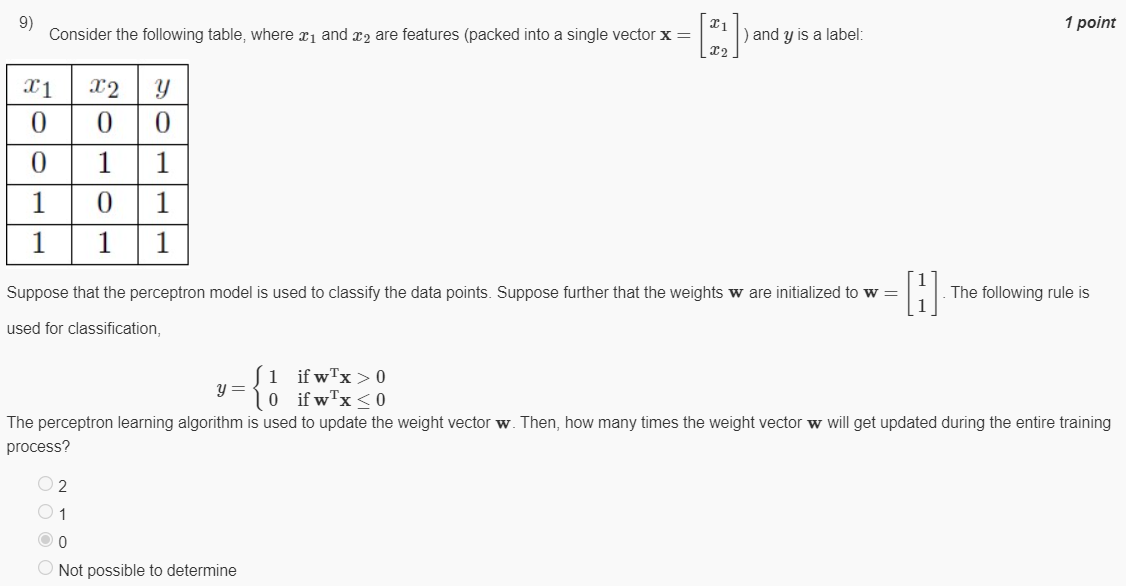
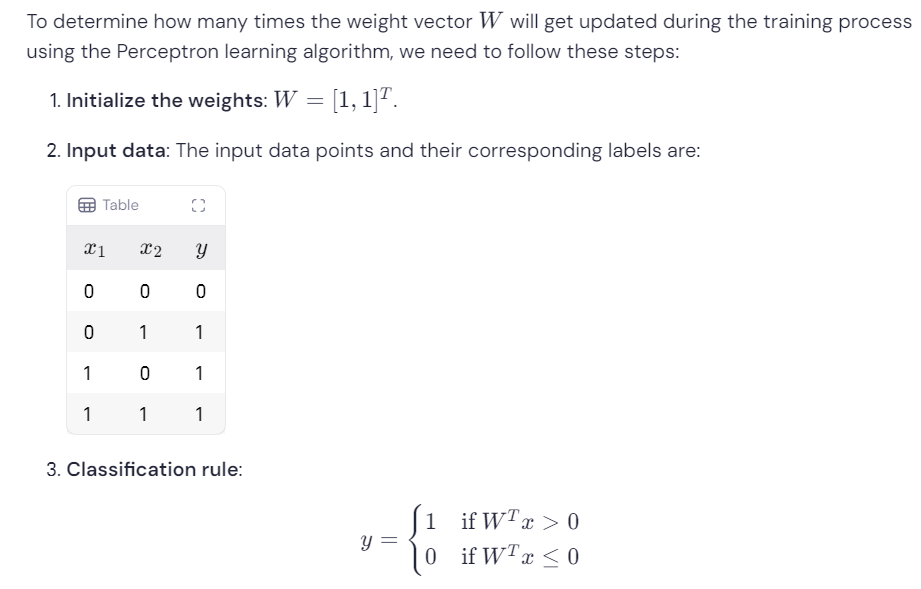
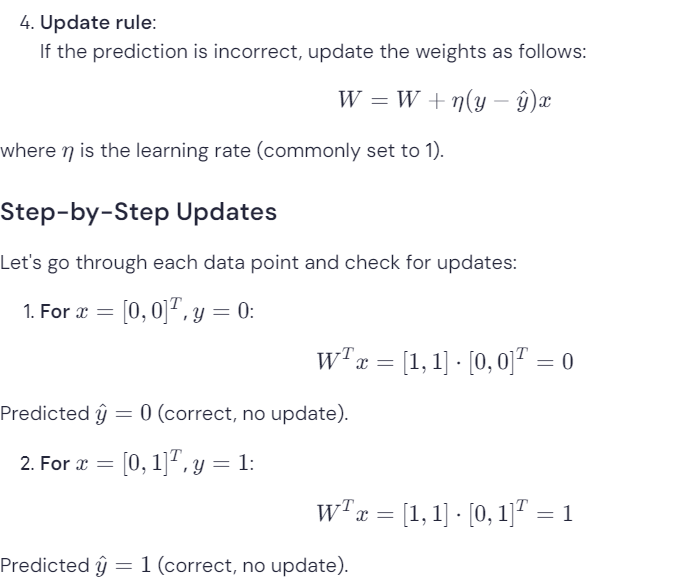
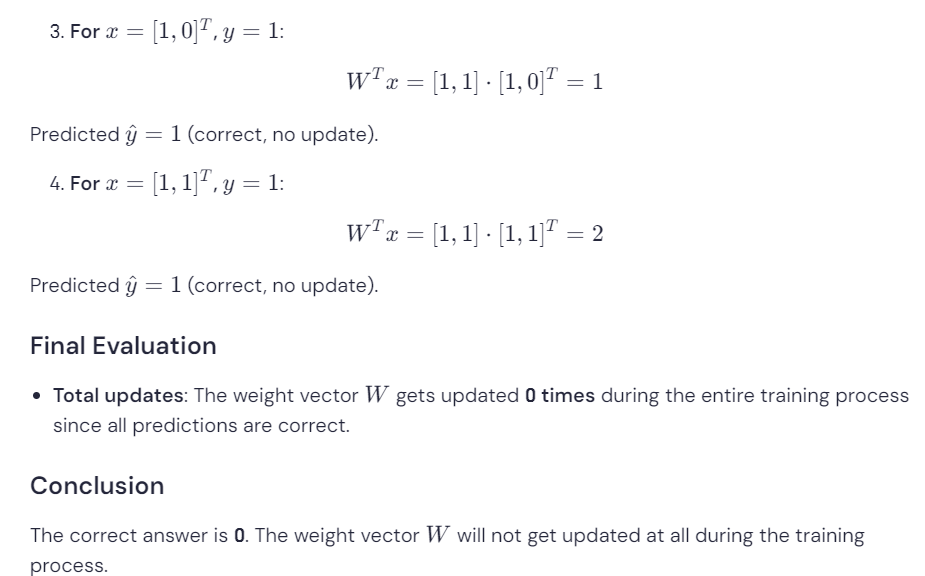
**Not Linearly Separable:**

* **XOR**
* **XNOR**



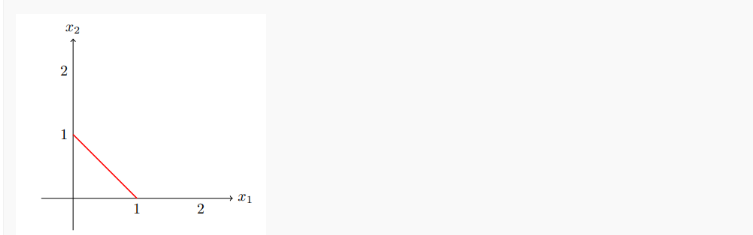
Which of the following threshold values of MP neuron implements AND Boolean function? Assume that the number of inputs to the neuron is 7 and the neuron does not have any inhibitory inputs.

***1 point***

Which Boolean function with two inputs x1 and x2 is represented by the following decision boundary? (Points on boundary or right of the decision boundary to be classified 1)



AND

 OR

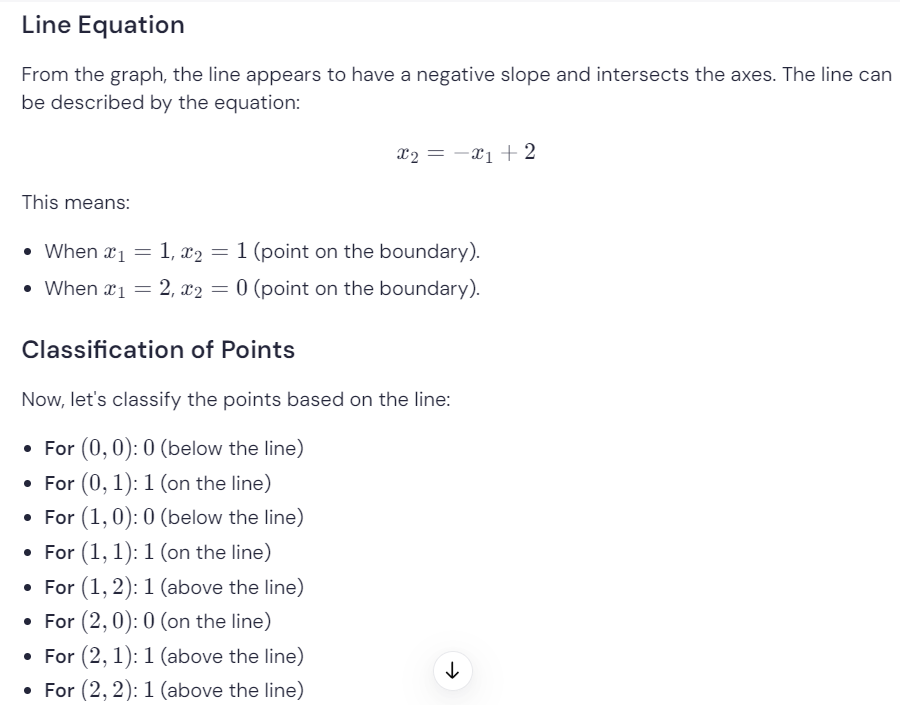
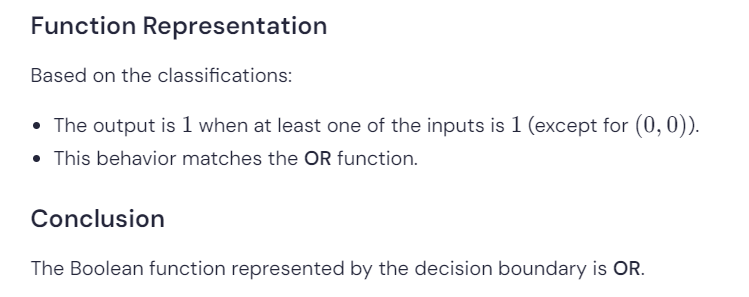
 XOR

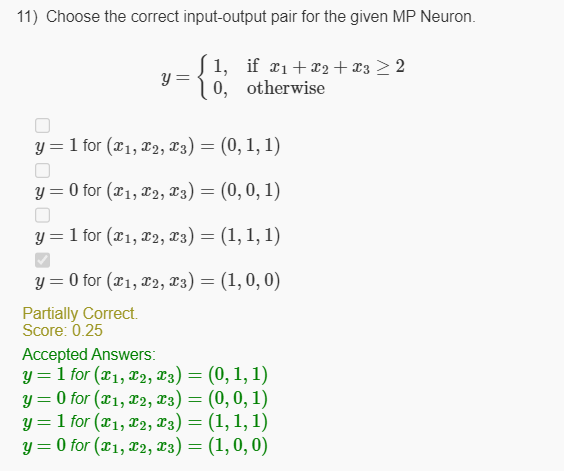
 NAND

Yes, the answer is correct.  
Score: 1

Accepted Answers:

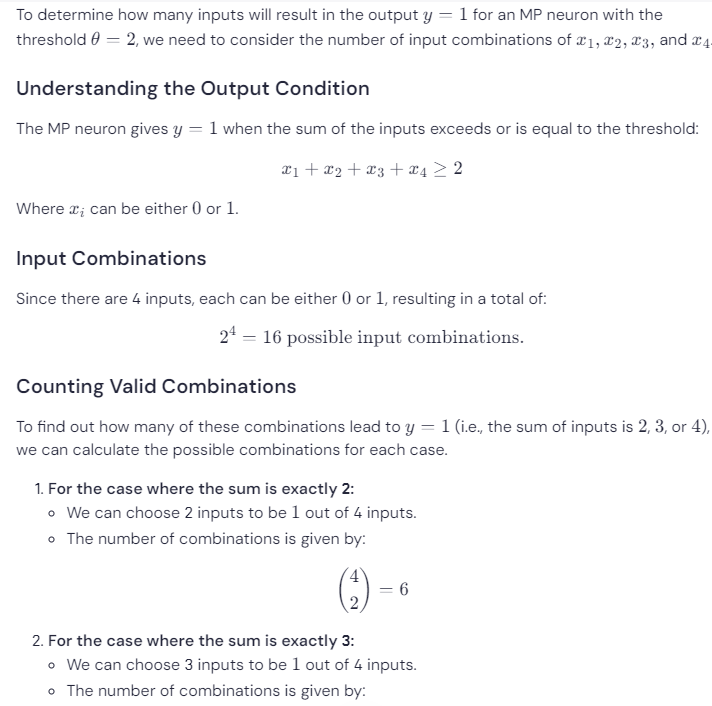
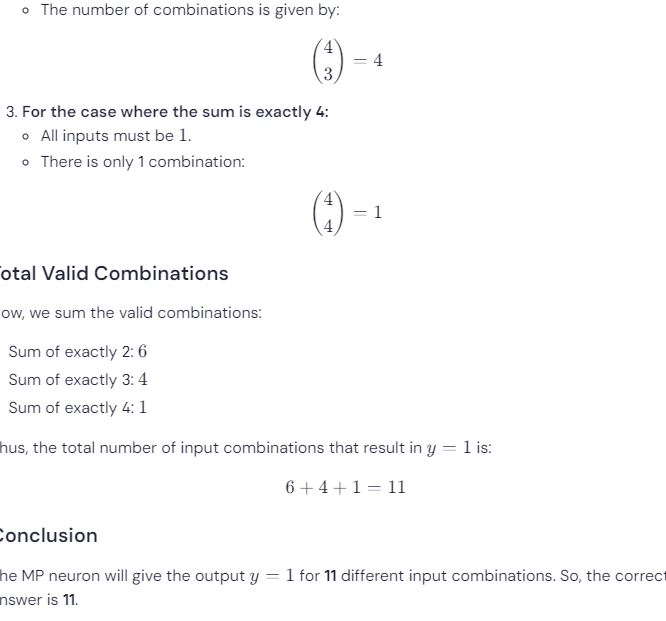
*OR*

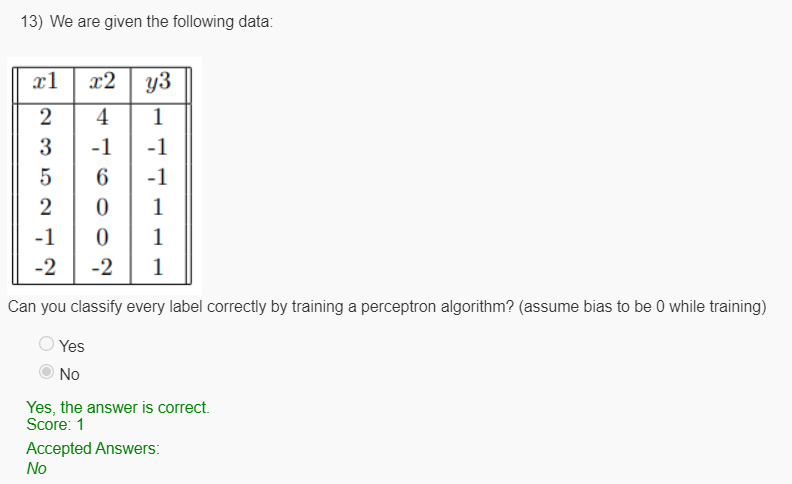
 

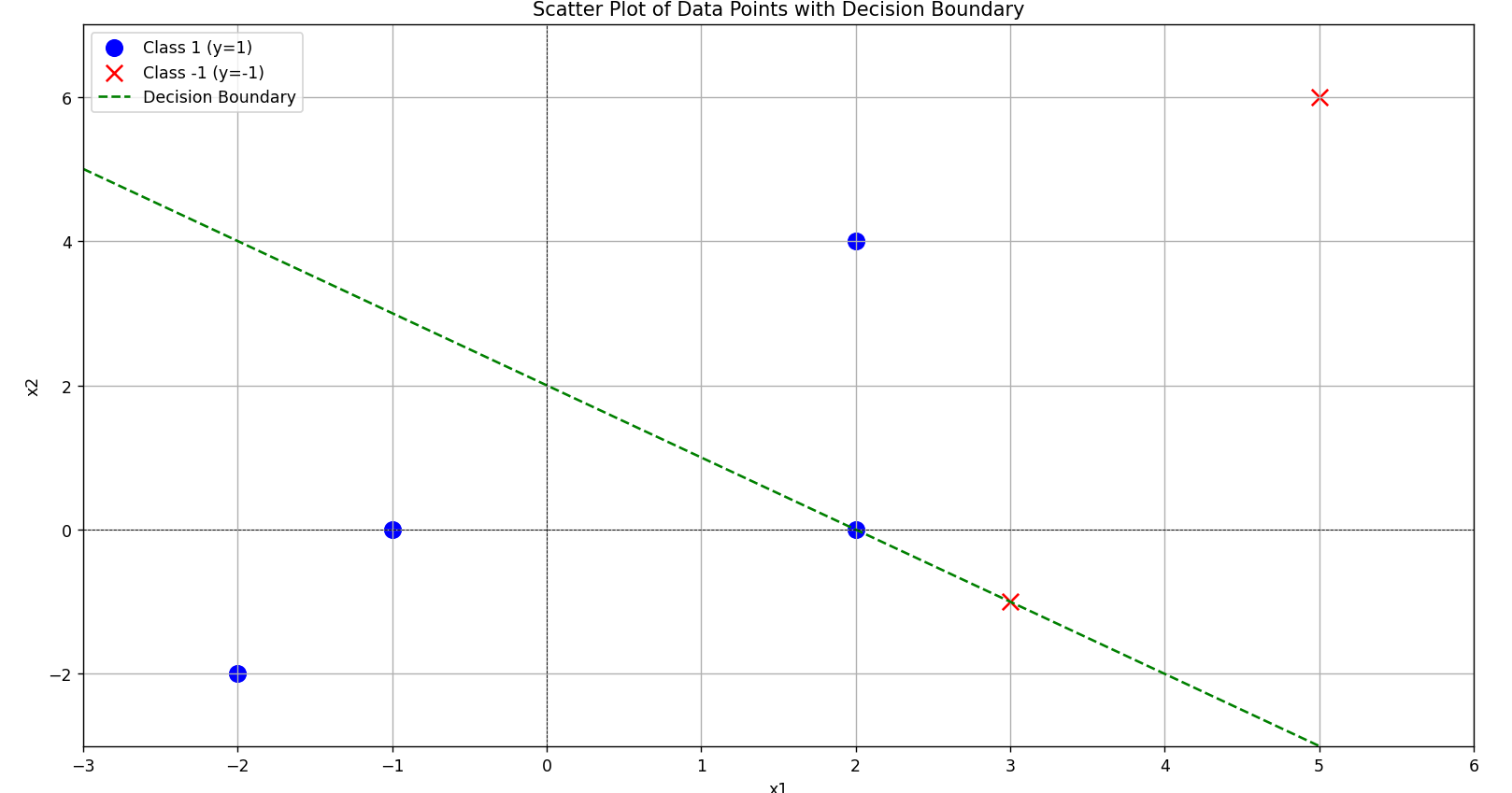
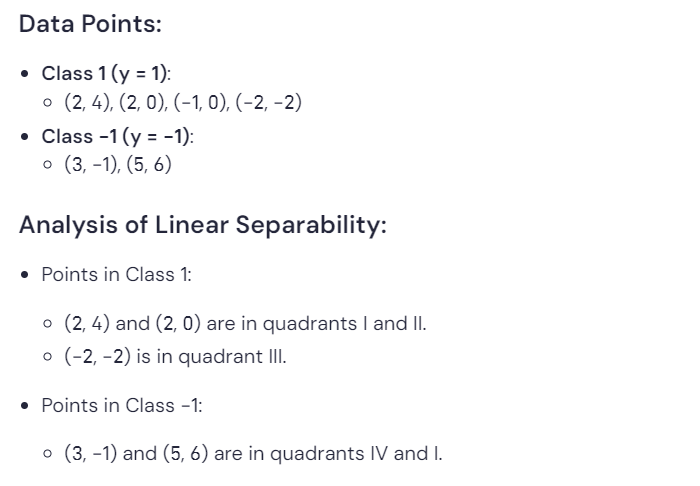
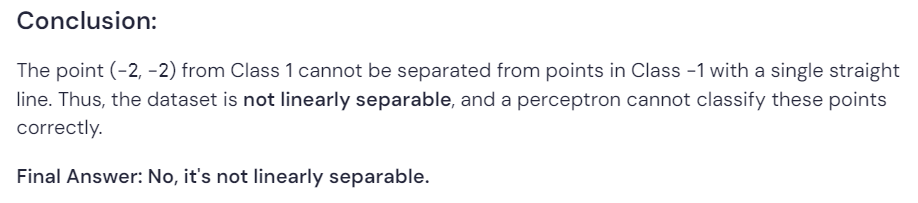


Suppose we have a boolean function that takes 4 inputs: x1, x2, x3, and x4. We have an MP neuron with a threshold parameter θ = 2. For how many input combinations will this MP neuron give an output y = 1?

* 11
* 21
* 15
* 8



We are given the following dataset with features as (x1, x2) and y as the label (-1, 1). If we apply the perceptron algorithm on the following dataset with w initialized as (0, 0), what will be the value of w when the algorithm converges? (Start the algorithm from (2, 2))

|  |  |  |
| --- | --- | --- |
| **x1** | **x2** | **y** |
| 2 | 2 | 1 |
| 2 | -2 | 1 |
| -2 | 1 | -1 |

Export to Sheets

* (2, 2)
* (2, 1)
* (2, -1)
* None of These

