Al-Mustaqbal University
College of Engineering
Department of Biomedical Engineering
Stage: 5th Subject : Neural Eng.

2025-2026
Lecture ( ):

4. Types of Problems:

There are many different problems that can be solved with a neural network. However, neural networks are commonly used to address particular types of problems. The following four types of problem are frequently solved with neural networks:

- Classification.
- Prediction.
- Pattern recognition.
- Optimization.

1. Classification:

Classification is the process of classifying input into groups. For example, an insurance company may want to classify insurance applications into different risk categories, or an online organization may want its email system to classify incoming mail into groups of spam and non-spam messages.

Often, the neural network is trained by presenting it with a sample group of data and instructions as to which group each data element belongs. This allows the neural network to learn the characteristics that may indicate group membership.

2. Prediction

Prediction is another common application for neural networks. Given a time-based series of input data, a neural network will predict future values. The accuracy of the guess will be dependent upon many factors, such as the quantity and relevancy of the input data. For example, neural networks are commonly applied to problems involving predicting movements in financial markets.

3. Pattern Recognition

Pattern recognition is one of the most common uses for neural networks. Pattern recognition is a form of classification. Pattern recognition is simply the ability to recognize a pattern. The pattern must be recognized even when it is distorted. Consider the following everyday use of pattern recognition.

Every person who holds a driver’s license should be able to accurately identify a traffic light. This is an extremely critical pattern recognition procedure carried out by countless drivers every day. However, not every traffic light looks the same, and the appearance of a particular traffic light can be altered depending on the time of day or the season. In addition, many variations of the traffic light exist. Still, recognizing a traffic light is not a hard task for a human driver.

How hard is it to write a computer program that accepts an image and tells you if it is a traffic light? Without the use of neural networks, this could be a very complex task. Most common programming algorithms are quickly exhausted when presented with a complex pattern recognition problem.

4.4 Optimization

Another common use for neural networks is optimization. Optimization can be applied to many different problems for which an optimal solution is sought. The neural network may not always find the optimal solution; rather, it seeks to find an acceptable solution. Optimization problems include circuit board assembly, resource allocation, and many others.

Perhaps one of the most well-known optimization problems is the traveling salesman problem (TSP). A salesman must visit a set number of cities. He would like to visit all cities and travel the fewest number of miles possible. With only a few cities, this is not a complex problem. However, with a large number of cities, brute force methods of calculation do not work nearly as well as a neural network approach.

5. Examples:

The following simple examples will illustrate classification problems.

5.1 Example: The AND Operation

We will now look at a neural network that acts as an AND gate. Table 1 shows the truth table for the AND logical operation.

Table 1: The AND Logical Operation (Binary)

p_1	p_2	p_1 AND p_2
0	0	0
0	1	0
1	0	0
1	1	1

A simple neural network can be create that recognizes the AND logical operation. This network will contain two inputs and one neuron (perceptron) with threshold as the transfer function. A neural network that recognizes the AND logical operation is shown in Figure 20.

There are two inputs to the network shown in Figure 20. Each input has a weight of one. The threshold is T = 1.5. Therefore, a neuron will only fire ($output = 1$) if both inputs are **true**. If either input is false, the sum of the two inputs will not exceed the threshold of T = 1.5 ($output = 0$).

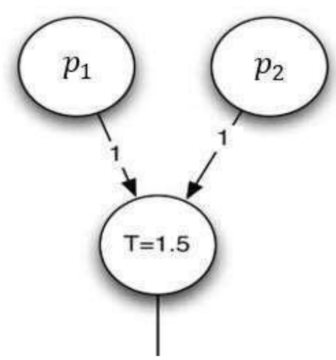

Figure 20: A neural network that recognizes the AND logical operation.

$$f(n) = \begin{cases} 0 & n < 1.5 \\ 1 & n \geq 1.5 \end{cases}$$

First method:

$$a = 1 \quad \text{if} \quad w_{1,1}\, p_1 + w_{1,2}\, p_2 > T$$

$a = 0$ if $w_{1,1}\, p_1 + w_{1,2}\, p_2 < T$

We make use of the truth Table 1 and the following inequalities are obtained.

$0 < T$

$w_{1,2} < T$

$w_{1,1} < T$

$w_{1,1} + w_{1,2} > T$

The task is to determine the values for $w_{1,1}$, $w_{1,2}$ & T so that the output satisfies the logic AND function.

We can choose $w_{1,1} = 0.5$, $w_{1,2} = 0.8$ & T = 1 or $w_{1,1} = 1$, $w_{1,2} = 1$ & T = 1.5 as the solution (satisfy the inequalities).

Second method:

Another way to determine the values for $w_{1,1}$, $w_{1,2}$ & T by finding a line separating the 3 "0" from the "1" as in Figure 21, which is called ***decision boundary***. Apparently, the 3 lines and many other lines can satisfy the requirement. We say that the problem is ***linear separable***.

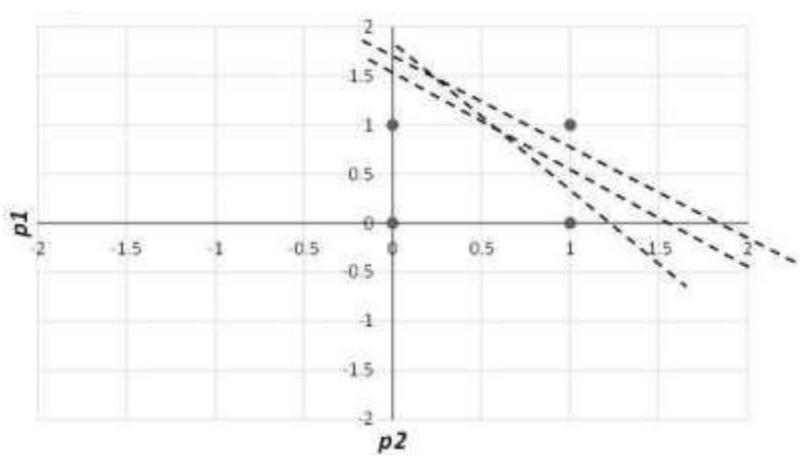

Figure 21: Truth Table (Binary)

To find the separating line, we need to find the slope and the interceptions of the line i.e. the points (0, 1.5) , (1.5,0):

$$w_{1,1}\,p_1 + w_{1,2}\,p_2 = \mathrm{T} \quad \Rightarrow \quad p_2 = \frac{\mathrm{T}}{w_{1,2}} - \frac{w_{1,1}\,p_1}{w_{1,2}}$$

$$m = \frac{1.5-0}{0-1.5} = -1$$

$$(y - y_1) = m(x - x_1) \quad \Rightarrow \quad y = m(x - x_1) + y_1 = -x + 1.5$$

$$\therefore \frac{w_{1,1}}{w_{1,2}} = 1 \quad \Rightarrow \quad w_{1,1} = w_{1,2}$$

$$\frac{\mathrm{T}}{w_{1,2}} = 1.5 \quad \Rightarrow \quad \mathrm{T} = 1.5\,w_{1,2}$$

$$\text{If } w_{1,1} = w_{1,2} = 1 \quad \Rightarrow \quad \mathrm{T} = 1.5$$

Now, Table 2 shows the truth table for the AND logical operation as bipolar representation.

Table 2: The AND Logical Operation (Bipolar)

p_1	p_2	p_1 AND p_2
−1	−1	−1
−1	1	−1
1	−1	−1
1	1	1

First method:

$a = 1$ if $w_{1,1}\,p_1 + w_{1,2}\,p_2 > \mathrm{T}$

$a = -1$ if $w_{1,1}\,p_1 + w_{1,2}\,p_2 < \mathrm{T}$

We make use of the truth Table 2 and the following inequalities are obtained.

$-w_{1,1} - w_{1,2} < \mathrm{T}$

$-w_{1,1} + w_{1,2} < \mathrm{T}$

$w_{1,1} - w_{1,2} < \mathrm{T}$

$w_{1,1} + w_{1,2} > \mathrm{T}$

The task is to determine the values for $w_{1,1}$, $w_{1,2}$ & T so that the output satisfies the logic AND function.

We can choose $w_{1,1} = 0.5$, $w_{1,2} = 0.8$ & T = 1 or $w_{1,1} = 1$, $w_{1,2} = 1$ & T = 1.5 as the solution (satisfy the inequalities).

Second method:

Another way to determine the values for $w_{1,1}$, $w_{1,2}$ & T by finding a line separating the 3 "−1" from the "1" , one possible ***decision boundary*** for this function is shown in Figure 22.

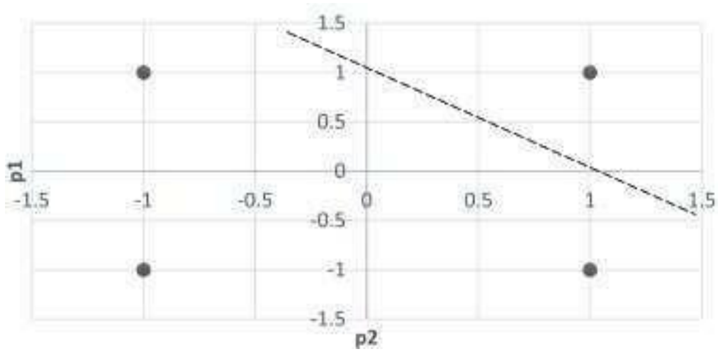

Figure 22: Truth Table (Bipolar)

So, the equation for the separating line in Figure 22 is:

$$m = \frac{1 - 0}{0 - 1} = -1$$

$$y = m(x - x_1) + y_1 = -x + 1$$

As we see previously $p_2 = \frac{T}{w_{1,2}} - \frac{w_{1,1}\, p_1}{w_{1,2}}$

$$\therefore \frac{w_{1,1}}{w_{1,2}} = 1 \quad \rightarrow \quad w_{1,1} = w_{1,2}$$

$$\frac{T}{w_{1,2}} = 1 \quad \rightarrow \quad T = w_{1,2}$$

If $w_{1,1} = w_{1,2} = 1 \quad \rightarrow \quad T = 1$

From above we see that the neuron (perceptron) with two inputs no bias can satisfy the truth table for AND logic by changing the boundary for threshold (hard limit) transfer function to the value T .

Third method:

If we don't want to change the boundary for the function

$$f(n) = \begin{cases} 0 & n < 0 \\ 1 & n \geq 0 \end{cases}$$

In this situation, we need to add a bias to the neuron as in Figure 23.

Then, the input Table 2 to the neuron is $w_{1,1}\,p_1 + w_{1,2}\,p_2 + b$

The decision line is

$$w_{1,1}\,p_1 + w_{1,2}\,p_2 + b = 0$$

$$\rightarrow \quad p_2 = -\frac{b}{w_{1,2}} - \frac{w_{1,1}\,p_1}{w_{1,2}}$$

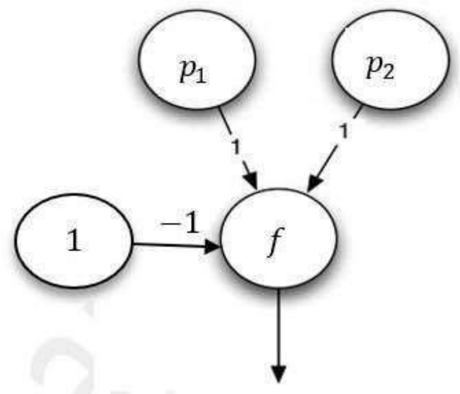

Figure 23: A neural network that recognizes the AND logical operation.

From above the separating line in Figure 22 is:

$$y = -x + 1$$

$$\therefore \frac{w_{1,1}}{w_{1,2}} = 1 \quad \rightarrow \quad w_{1,1} = w_{1,2}$$

$$-\frac{b}{w_{1,2}} = 1 \quad \rightarrow \quad b = -w_{1,2}$$

If $w_{1,1} = w_{1,2} = 1 \quad \rightarrow \quad b = -1$

5.2 Example: The OR Operation

Neural networks can be created to recognize other logical operations as well. Consider the **OR** logical operation. The truth table for the **OR** logical operation is shown in Table 3. The **OR** logical operation is true if either input is **true**.

Table 3: The OR Logical Operation (Binary)

p_1	p_2	p_1 **OR** p_2
0	0	0
0	1	1
1	0	1
1	1	1

The neural network that will recognize the OR operation is shown in Figure 24.

The **OR** neural network looks very similar to the **AND** neural network. The biggest difference is the threshold value. Because the threshold is lower, only one of the inputs needs to have a value of **true** for the output neuron to fire.

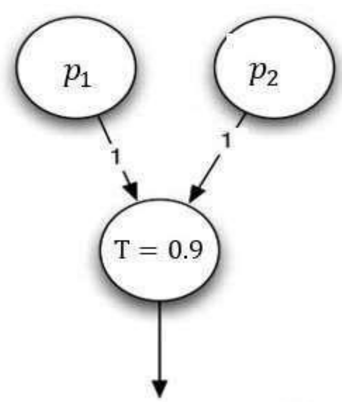

Figure 24: A neural network that recognizes the OR logical operation.

$a = 1 \quad \text{if} \quad w_{1,1}\, p_1 + w_{1,2}\, p_2 > T$

$a = 0 \quad \text{if} \quad w_{1,1}\, p_1 + w_{1,2}\, p_2 < T$

We make use of the truth Table 3 and the following inequalities are obtained.

$0 < T$

$w_{1,2} > T$

$w_{1,1} > T$

$w_{1,1} + w_{1,2} > T$

We can choose $w_{1,1} = 0.5$, $w_{1,2} = 0.8$ & $T = 0.4$ or $w_{1,1} = 1$, $w_{1,2} = 1$ & $T = 0.9$ as the solution (satisfy the inequalities).