SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
An Autonomous Institution
MANGALURU

**Department of Computer Science and Engineering**

**(Artificial Intelligence and Machine Learning)**

**NEURAL NETWORKS AND DEEP LEARNING (21AI72)**

# Case study Report on

Biological to Artificial neuron

Prepared by


| | |
|---|---|
| Dishan D Karkera | 4SF21AD016 |
| Harthik | 4SF21AD019 |
| Prasanna | 4SF21AD038 |
| Preetham Pinto | 4SF21AD039 |


Under the Guidance of


**Dr. Gurusiddayya Hiremath**

Assistant Professor

Department of Computer Science and Engineering

(Artificial Intelligence and Machine Learning)

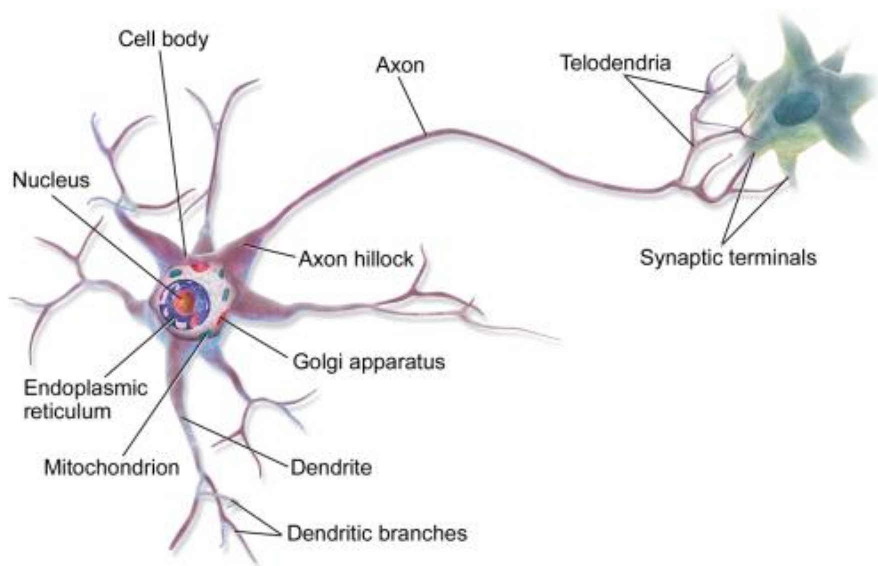
SCEM, Mangaluru


Academic Year: 2024-25

# Biological to Artificial Neurons

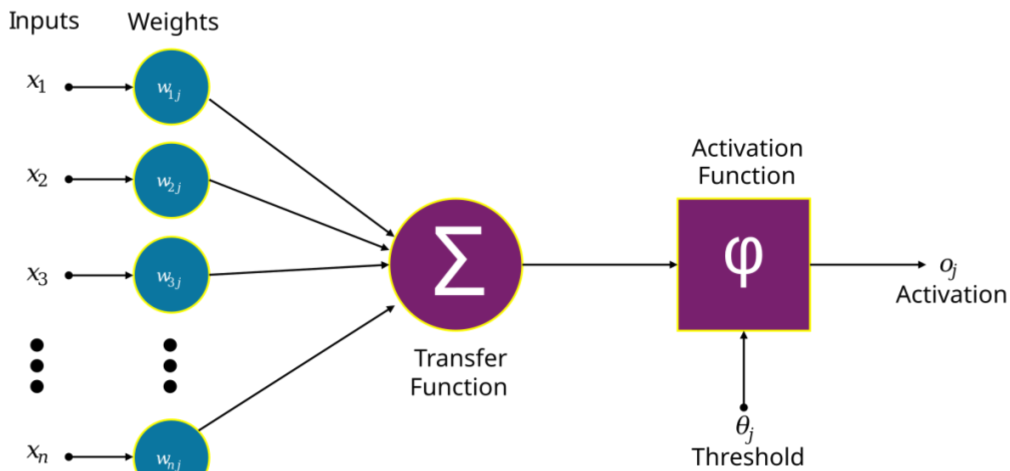
**Biological Neurons:**

Before we discuss artificial neurons, let's take a quick look at a biological neuron It is an unusual-looking cell mostly found in animal brains. It's composed of a cell body containing the nucleus and most of the cell's complex components, many branching extensions called dendrites, plus one very long extension called the axon. The axon's length may be just a few times longer than the cell body, or up to tens of thousands of times longer. Near its extremity the axon splits off into many branches called telodendria , and at the tip of these branches are minuscule structures called synaptic terminals (or simply synapses), which are connected to the dendrites or cell bodies of other neurons. Biological neurons produce short electrical impulses called action potentials (APs, or just signals) which travel along the axons and make the synapses release chemical signals called neurotransmitters. When a neuron receives a sufficient amount of these neurotransmitters within a few milliseconds, it fires its own electrical impulses (actually, it depends on the neurotransmitters, as some of them inhibit the neuron from firing). Thus, individual biological neurons seem to behave in a rather simple way, but they are organized in a vast network of billions, with each neuron typically connected to thousands of other neurons. Highly complex computations can be performed by a network of fairly simple neurons, much like a complex anthill can emerge from the combined efforts of simple ants. The architecture of biological neural networks (BNNs) is still the subject of active research, but some parts of the brain have been mapped, and it seems that neurons are often organized in consecutive layers, especially in the cerebral cortex (i.e., the outer layer of your brain).



**Artificial Neurons:**

McCulloch and Pitts proposed a very simple model of the biological neuron, which later became known as an artificial neuron: it has one or more binary (on/off) inputs and one binary output. The artificial neuron activates its output when more than a certain number of its inputs are active. In their paper, they showed that even with such a simplified model it is possible to build a network of artificial neurons that computes any logical proposition you want. To see how such a network works, let's build a few ANNs that perform various logical computations, assuming that a neuron is activated when at least two of its inputs are active.




**The Perceptron:**

The Perceptron is one of the simplest ANN architectures, invented in 1957 by Frank Rosenblatt. It is based on a slightly different artificial neuron (see Figure 10-4) called a threshold logic unit (TLU), or sometimes a linear threshold unit (LTU). The inputs and output are numbers (instead of binary on/off values), and each input connection is associated with a weight. The TLU computes a weighted sum of its inputs (z = w1 x1 + w2 x2 + ⋯ + wn xn = x ⊺ w), then applies a step function to that sum and outputs the result: hw(x) = step(z), where z = x ⊺ w.

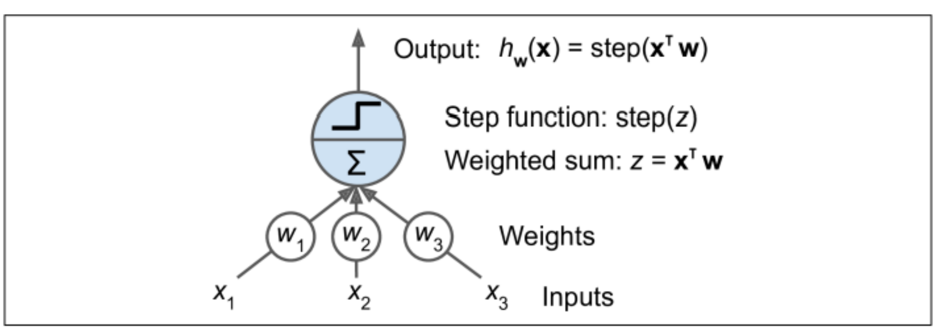



*Figure 10-4. Threshold logic unit: an artificial neuron which computes a weighted sum of its inputs then applies a step function*

$$\text{heaviside}(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases} \qquad \text{sgn}(z) = \begin{cases} -1 & \text{if } z < 0 \\ 0 & \text{if } z = 0 \\ +1 & \text{if } z > 0 \end{cases}$$

A single TLU can be used for simple linear binary classification. It computes a linear combination of the inputs, and if the result exceeds a threshold, it outputs the positive class. Otherwise it outputs the negative class (just like a Logistic Regression or linear SVM classifier). You could, for example, use a single TLU to classify iris flowers based on petal length and width (also adding an extra bias feature x0 = 1, just like we did in previous chapters). Training a TLU in this case means finding the right values for w0 , w1 , and w2 (the training algorithm is discussed shortly).

A Perceptron with two inputs and three outputs is represented in Figure. This Perceptron can classify instances simultaneously into three different binary classes, which makes it a multioutput classifier.

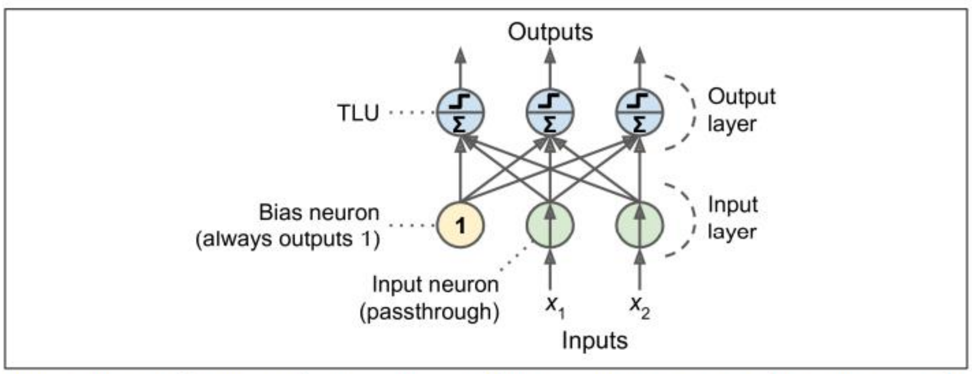



*Figure 10-5. Architecture of a Perceptron with two input neurons, one bias neuron, and three output neurons*

**Multilayer Perceptron(MLP's):**

An MLP is composed of one (passthrough) input layer, one or more layers of TLUs, called hidden layers, and one final layer of TLUs called the output layer (see Figure 10-7). The layers close to the input layer are usually called the lower layers, and the ones close to the outputs are usually called the upper layers. Every layer except the output layer includes a bias neuron and is fully connected to the next layer.

When an ANN contains a deep stack of hidden layers,9 it is called a deep neural net- work (DNN). The field of Deep Learning studies DNNs, and more generally models containing deep stacks of

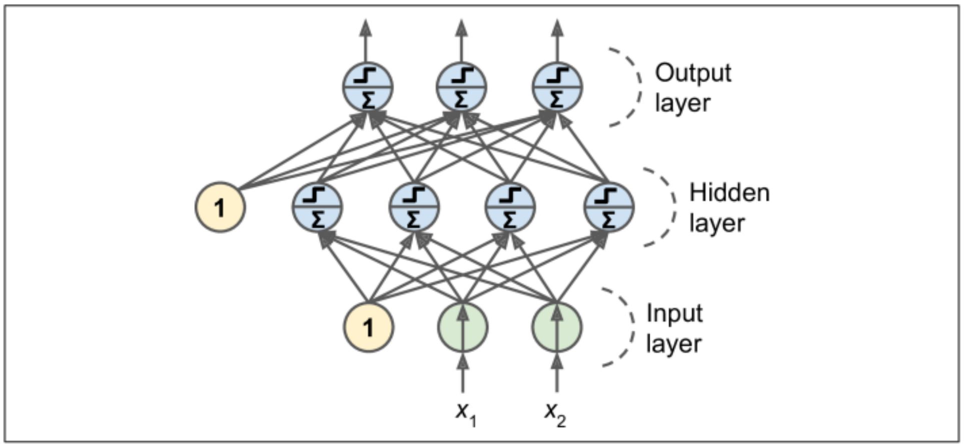



*Figure 10-7. Architecture of a Multilayer Perceptron with two inputs, one hidden layer of four neurons, and three output neurons (the bias neurons are shown here, but usually they are implicit)*

computations. Even so, many people talk about Deep Learning whenever neural networks are involved (even shallow ones).

**Regression MLP's:**

First, MLPs can be used for regression tasks. If you want to predict a single value (e.g., the price of a house, given many of its features), then you just need a single output neuron: its output is the predicted value. For multivariate regression (i.e., to predict multiple values at once), you need one output neuron per output dimension. For example, to locate the center of an object in an image, you need to predict 2D coordi- nates, so you need two output neurons. If you also want to place a bounding box around the object, then you need two more numbers: the width and the height of the object. So, you end up with four output neurons. In general, when building an MLP for regression, you do not want to use any activa- tion function for the output neurons, so they are free to output any range of values. If you want to guarantee that the output will always be positive, then you can use the ReLU activation function in the output layer. Alternatively, you can use the softplus activation function, which is a smooth variant of ReLU: softplus(z) = log(1 + exp(z)). It is close to 0 when z is negative, and close to z when z is positive. Finally, if you want to guarantee that the predictions will fall within a given range of values, then you can use the logistic function or the hyperbolic tangent, and then scale the labels to the appropriate range: 0 to 1 for the logistic function and –1 to 1 for the hyperbolic tangent.

**Classification MLP's:**

MLPs can also be used for classification tasks. For a binary classification problem, you just need a single output neuron using the logistic activation function: the output will be a number between

0 and 1, which you can interpret as the estimated probabil- ity of the positive class. The estimated probability of the negative class is equal to one minus that number.

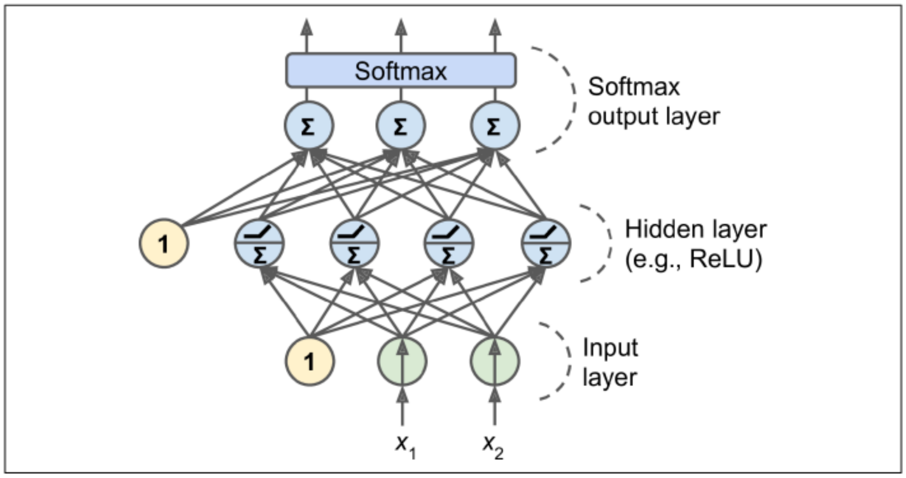



*Figure 10-9. A modern MLP (including ReLU and softmax) for classification*

MLPs can also easily handle multilabel binary classification tasks (see Chapter 3). For example, you could have an email classification system that predicts whether each incoming email is ham or spam, and simultaneously predicts whether it is an urgent or nonurgent email. In this case, you would need two output neurons, both using the logistic activation function: the first would output the probability that the email is spam, and the second would output the probability that it is urgent. More generally, you would dedicate one output neuron for each positive class. Note that the output probabilities do not necessarily add up to 1. This lets the model output any combina- tion of labels: you can have nonurgent ham, urgent ham, nonurgent spam, and per- haps even urgent spam (although that would probably be an error).

**Presentation Slides:**

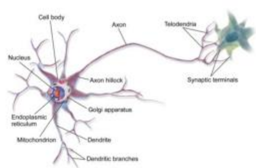
# BIOLOGICAL TO ARTIFICIAL NEURONS

## HISTORICAL BACKGROUND

- In 1943 Warren McCulloch and Walter Pitts introduced the first artificial neural network architecture.
- In 1957 Frank Rosenblatt invented the Perceptron.
- By 1980s Revival of interest with new architectures and training techniques.
- During 1990s ANNs overshadowed by Support Vector Machines.
- And in 2000s it Renewed interest due to availability of data, improved algorithms, and increased computational power.

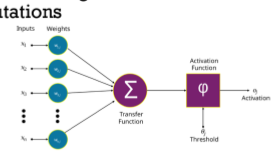
## BIOLOGICAL NEURONS

- Found in animal brains.
- Organized in vast networks, performing complex computations.




## ARTIFICIAL NEURONS

- Simplified computational model.
- Binary or numeric inputs and outputs.
- Can perform logical and mathematical computations




## THE PERCEPTRON

- Based on Threshold Logic Units (TLUs).
- Performs simple linear binary classification.
- Comprised of:
  - Input neurons (including a bias neuron).
  - A single layer of TLUs (fully connected).
  - Outputs for binary or multi-output classification.

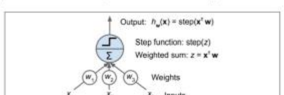



*Figure 10-4. Threshold logic unit: an artificial neuron which computes a weighted sum of its inputs then applies a step function*

- A Perceptron with two input neurons, one bias neuron, and three output neurons

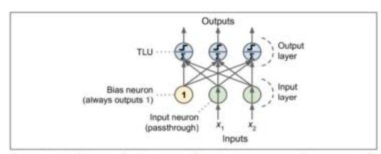



*Figure 10-5. Architecture of a Perceptron with two input neurons, one bias neuron, and three output neurons*

## MULTILAYER PERCEPTRONS (MLPS)

- An MLP is composed of one (passthrough) input layer.
- One or more layers of TLUs, called hidden layers
- One final layer of TLUs called the output layer

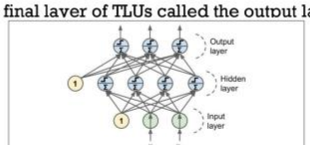



*Figure 10-7. Architecture of a Multilayer Perceptron with two inputs, one hidden layer of four neurons, and three output neurons (the bias neurons are shown here, but usually they are implicit)*

## CLASSIFICATION MLP

- MLPs can also be used for classification tasks.
- They can also easily handle multilabel binary classification tasks.

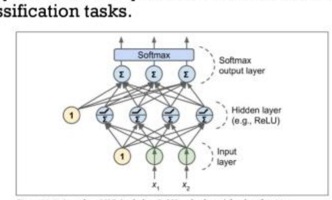



*Figure 10-9. A modern MLP (including ReLU and softmax) for classification*