IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

NEURAL NETWORK

[1] Dr.Balaji K [2] Vinil Monteiro, [3] Tarun G, [4] Tarun Kulakarni,[5] Santosh Maddi, [6] Syed Kaif

[1]Professor, Department of MCA, Cambridge Institute of Technology CITech, Bengaluru, India, [2,3,4,5,6] Student, Department of MCA, CITech, Bengaluru, India.

Abstract:

Neural networks, also known as artificial neural networks (ANNs) or simply neural networks, are a subset of machine learning that form the foundation of deep learning techniques. Which was inspired by the structure and function of the human brain, mimicking the way real neurons communicate. ANNs are composed of many interconnected basic processors, functioning in parallel. This paper explores artificial neural networks and their basic types, outlining the fundamental neuron and the artificial computer model. It discusses network structures, learning methods, and some of the most commonly used ANNs..

1.Introduction

The roots of neural network development can be traced back to the early 1940s, with a significant resurgence in the late 1980s due to the discovery of new techniques and advancements in computer hardware.

Neural networks are essentially parallel computing devices designed to mimic the brain's functionality. Their primary goal is to outperform traditional systems in various computational tasks, such as classification, prediction, association, and data clustering, by executing these tasks more swiftly.

1.1 Tasks Neural Networks Perform

Neural networks are highly valuable because they excel at tasks that involve making sense of data while preserving their inherent attributes. They perform several critical tasks, including:

- **Classification**: Organizing patterns or datasets into predefined classes.
- **Prediction**: Generating expected output from given input.
- **Clustering**: Identifying unique features within data and classifying them without prior knowledge of the data.
- **Associating**: Training neural networks to "remember" patterns, allowing them to associate unfamiliar versions of patterns with the most similar version in their memory.

2.Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) is a computational system inspired by biological neural networks. Also known as "artificial neural systems" or "parallel distributed processing systems," ANNs consist of interconnected units, or nodes, that communicate with each other. These nodes, akin to neurons in the brain, are simple processors that work in parallel.

- In our brain, billions of neurons process information in the form of electric signals.
- External information is received by the dendrites of the neuron, processed in the neuron cell body, converted to an output, and passed through the axon to the next neuron.
- The next neuron can either accept or reject this signal based on its strength.

- Most neural networks have a "training" rule, meaning they learn from examples (similar to how children learn to recognize dogs from examples of dogs) and can generalize beyond the training data.

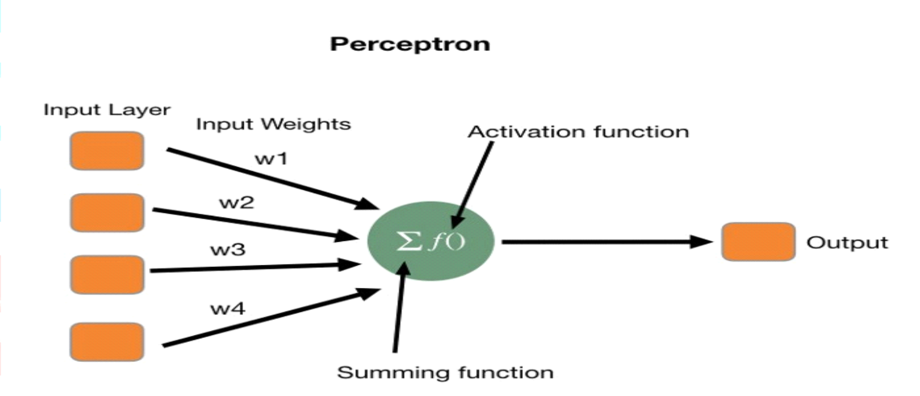
The learning algorithm of a neural network processes many labelled examples (data with "answers") during training. By using this answer key, the neural network learns the characteristics of the input needed to produce the correct output. Once a sufficient number of examples have been processed, the neural network can effectively process new, unseen inputs and provide accurate results. The accuracy typically improves with more examples and a variety of inputs, as the program learns from experience.

Every artificial neuron has the following main functions:

- Takes inputs from the input layer
- The neuron weighs each input separately and then sums them up.
- Pass this sum through a nonlinear function to produce output.

The perceptron(neuron) consists of 4 parts:

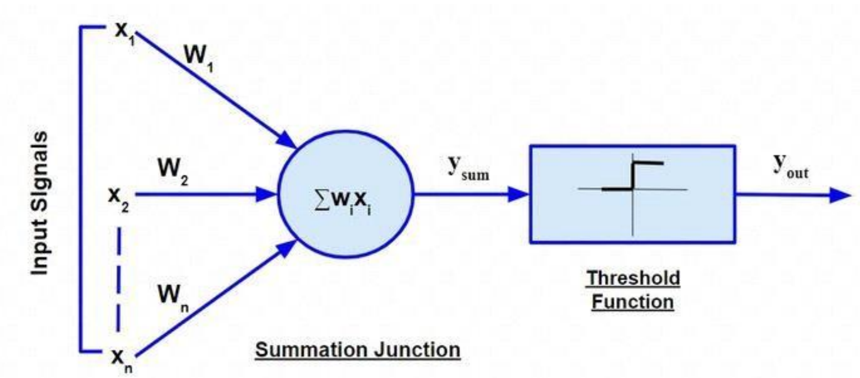
- Input values or One input layer
- Weights and Bias
- Activation Function
- Output Layer



2.2 McCulloch-Pitts Model of Neuron

The earliest neural model, the McCulloch-Pitts model, features two types of inputs: Excitatory and Inhibitory. Excitatory inputs have positive weights, while inhibitory inputs have negative weights. Inputs to the McCulloch-Pitts neuron are binary, taking values of either 0 or 1. It uses a threshold function as its activation function, where the output signal, y_out, is 1 if the sum of the inputs, y_sum, is greater than or equal to a specified threshold value; otherwise, it is 0. Here is a diagrammatic representation of the model:

[Diagram depicting McCulloch-Pitts neuron with excitatory and inhibitory inputs, threshold function, and output signals]



Simple McCulloch-Pitts neurons are capable of representing logical operations. To achieve this, the connection weights and the threshold function must be appropriately chosen.

Mathematical Definition:

McCulloch and Pitts introduced a mathematical formulation called a linear threshold gate, which models the behaviour of a single neuron with two states: firing or not firing. In its basic form, the mathematical formulation is as follows:

$$Sum = \sum_{i=1}^{N} I_i W_i$$

$$y(Sum) = \begin{cases} 1, & \text{if } Sum \geq T \\ 0, & \text{otherwise} \end{cases}$$

Where I1, I2, ..., IN are binary input values (0 or 1), W1, W2, ..., WN are weights associated with each input (-1 or 1), Sum is the weighted sum of inputs, and T is a predefined threshold value for the neuron activation (firing). Figure 3 illustrates a graphical representation of the McCulloch-Pitts artificial neuron.

2.3 Taxonomy of ANN – Connectivity

The basic principles of artificial neural networks (ANNs) remain consistent, but various models have emerged over its evolution. Here are a few examples:

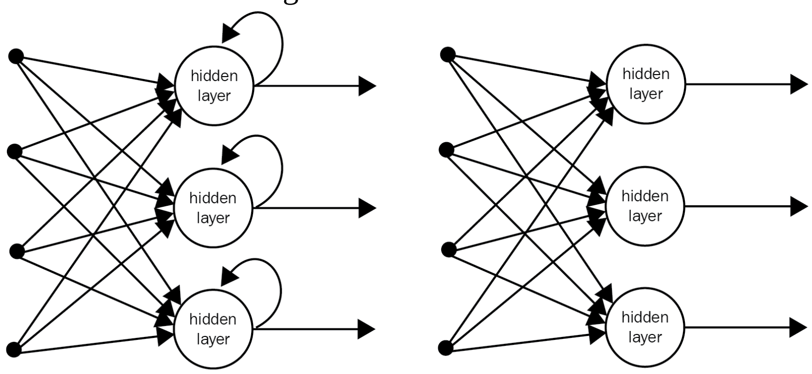
- **Adaptive Linear Element (ADALINE):** A simple perceptron capable of solving linear problems. Each neuron computes the weighted linear sum of inputs and passes it through a bipolar function, yielding either +1 or -1 based on the sum. If the net sum is >= 0, the output is +1; otherwise, it is -1.

- **Multiple ADALINEs (MADALINE):** A multilayer network composed of ADALINE units.

- **Perceptron:** Single-layer neural networks (single neuron or unit) where inputs are multidimensional vectors and the output is a function of the weighted sum of inputs.

- **Feed-forward Networks:** The simplest form of neural networks, processing data across layers without loops or cycles.

- **Recurrent Neural Networks (RNNs):** Contrary to feed-forward networks, recurrent neural networks (RNNs) which transmit information both forward, from earlier to later processing stages, and backward, from later to earlier stages.



(a) Recurrent neural network (b) Forward neural network

2.4 Kolmogorov theorem

The Kolmogorov extension theorem, also known as the Kolmogorov existence theorem, Kolmogorov consistency theorem, or Daniell-Kolmogorov theorem in some contexts, is a mathematical theorem. It ensures that a properly "consistent" set of finite-dimensional distributions can define a stochastic process. This theorem is attributed to the English mathematician Percy John Daniell and the Russian mathematician Andrey Nikolaevich Kolmogorov.

Statement of the theorem:

Let T denote some interval (thought of as "time"), and let $n \in \mathbb{N}$. For each $k \in \mathbb{N}$ and finite sequence of distinct times $t_1, \ldots, t_k \in T$, let $\nu_{t_1 \ldots t_k}$ be a probability measure on $(\mathbb{R}^n)^k$. Suppose that these measures satisfy two consistency conditions:

1. for all permutations π of $\{1, \ldots, k\}$ and measurable sets $F_i \subseteq \mathbb{R}^n$,

$$\nu_{t_{\pi(1)} \cdots t_{\pi(k)}} \left(F_{\pi(1)} \times \cdots \times F_{\pi(k)} \right) = \nu_{t_1 \ldots t_k} \left(F_1 \times \cdots \times F_k \right);$$

2. for all measurable sets $F_i \subseteq \mathbb{R}^n, m \in \mathbb{N}$

$$\nu_{t_1 \ldots t_k} \left(F_1 \times \cdots \times F_k \right) = \nu_{t_1 \ldots t_k, t_{k+1}, \ldots, t_{k+m}} \left(F_1 \times \cdots \times F_k \times \underbrace{\mathbb{R}^n \times \cdots \times \mathbb{R}^n}_{m} \right).$$

Then there exists a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and a stochastic process $X : T \times \Omega \to \mathbb{R}^n$ such that

$$\nu_{t_1 \ldots t_k} \left(F_1 \times \cdots \times F_k \right) = \mathbb{P} \left(X_{t_1} \in F_1, \ldots, X_{t_k} \in F_k \right)$$

for all $t_i \in T$, $k \in \mathbb{N}$ and measurable sets $F_i \subseteq \mathbb{R}^n$, i.e. X has $\nu_{t_1 \ldots t_k}$ as its finite-dimensional distributions relative to times $t_1 \ldots t_k$.

In fact, it is always possible to take as the underlying probability space $\Omega = (\mathbb{R}^n)^T$ and to take for X the canonical process $X\colon (t, Y) \mapsto Y_t$. Therefore, an alternative way of stating Kolmogorov's extension theorem is that, provided that the above consistency conditions hold, there exists a (unique) measure ν on $(\mathbb{R}^n)^T$ with marginals $\nu_{t_1 \ldots t_k}$ for any finite collection of times $t_1 \ldots t_k$. Kolmogorov's extension theorem applies when T is uncountable, but the price to pay for this level of generality is that the measure ν is only defined on the product σ-algebra of $(\mathbb{R}^n)^T$, which is not very rich.

3.Conclusion

Neural networks encompass a vast field, with many data scientists specializing solely in its techniques. While we focused on introductory concepts in this session, there are deeper aspects to explore. Advanced techniques beyond backpropagation exist, and neural networks perform exceptionally well in specific problem domains such as image recognition. However, implementing neural network algorithms can be computationally demanding, requiring efficient computing hardware. Managing large datasets in R can result in lengthy processing times, prompting the need to explore various options and packages. The field of neural networks is dynamic, with ongoing exciting research, making it an area ripe for further exploration. As you expand your understanding from this foundational knowledge, consider exploring topics like reinforced learning, deep learning, and other advanced concepts.

References

1. Dong, J., & Hu, S. (1997). Neural network research: Progress and prospects. Information and Control, 26(5), 360–368.

2. Jenkins, B. K., & Tanguay, A. R. (1995). Handbook of Neural Computing and Neural Networks. Boston: MIT Press.

3. Bnlsabi, A. (1993). Analysis solutions to the general problem for feedforward neural networks. Neural Networks, 6, 991–996.

4. Balcazar, J. (1997). Characterizing the computational power of neural networks in relation to Kolmogorov complexity. IEEE Transactions on Information Theory, 43(4), 1175–1183.

5. Setiono, R., & Leow, W. K. (2000). FERNN: An algorithm for fast extraction of rules from neural networks. Applied Intelligence, 12(1–2), 15–25.