



SNS COLLEGE OF TECHNOLOGY

An Autonomous Institution

Coimbatore-35



Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECT303-ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

III YEAR/ V SEMESTER

1

UNIT 4 – NEURONS AND NEURAL NETWORKS

4.2 Neural Networks

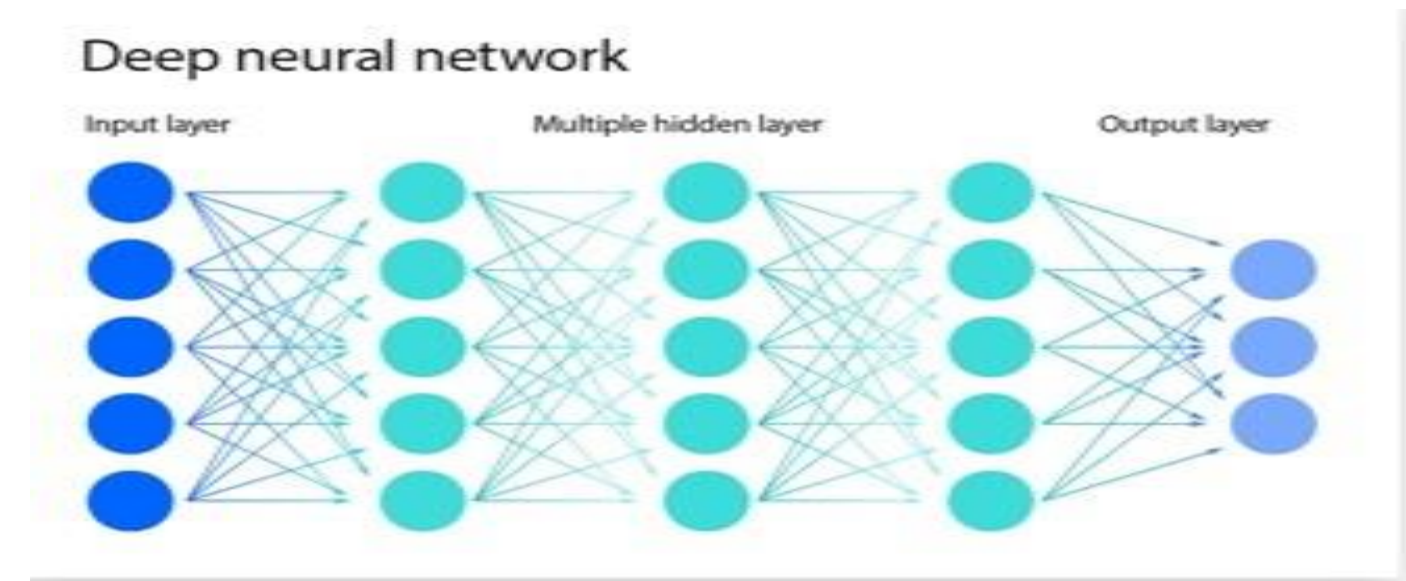


Neural Networks

- **Artificial Neural Networks (ANNs) or Simulated Neural Networks (SNNs) are a subset of Machine Learning and are at the heart of Deep Learning algorithms.**
- **Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.**

Working methods

- **ANNs are comprised of a**
 - **Node layers,**
 - **An input layer**
 - **One or more hidden layers**
 - **An output layer.**



- **Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.**



Neural Networks



Neural networks rely on training data to learn and improve their accuracy over time.

- However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity.
- Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts. One of the most well-known neural networks is Google's search algorithm.

How do neural networks work?

Think of each individual node as its own linear regression model, composed of input data, weights, a bias (or threshold), and an output.

$$\begin{aligned} \sum w_i x_i + \text{bias} &= w_1 x_1 + w_2 x_2 + w_3 x_3 + \text{bias} \\ \text{Output} &= f(x) = \begin{cases} 1 & \text{if } \sum w_1 x_1 + b \geq 0; \\ 0 & \text{if } \sum w_1 x_1 + b < 0 \end{cases} \end{aligned}$$



Neural Networks



- Once an input layer is determined, weights are assigned.
- These weights help determine the importance of any given variable, with larger ones contributing more significantly to the output compared to other inputs.
- All inputs are then multiplied by their respective weights and then summed. Afterward, the output is passed through an activation function, which determines the output.
- If that output exceeds a given threshold, it “fires” (or activates) the node, passing data to the next layer in the network.
- This results in the output of one node becoming in the input of the next node.
- This process of passing data from one layer to the next layer defines this neural network as a **feed forward network**.



Neural Networks



Let's break down what one single node might look like using binary values.

- **For surfing (Yes: 1, No: 0).**
- **The decision to go or not to go is our predicted outcome, or \hat{y} .**

Let's assume that there are three factors influencing your decision-making:

- **Are the waves good? (Yes: 1, No: 0)**
- **Is the line-up empty? (Yes: 1, No: 0)**
- **Has there been a recent shark attack? (Yes: 0, No: 1)**

Then, let's assume the following inputs:

5

- **$X_1 = 1$, since the waves are pumping**
- **$X_2 = 0$, since the crowds are out**
- **$X_3 = 1$, since there hasn't been a recent shark attack**



Neural Networks



- Now, we need to assign some weights to determine importance.
- Larger weights signify that particular variables are of greater importance to the decision or outcome.
- **W1 = 5, since large swells don't come around often**
- **W2 = 2, since you're used to the crowds**
- **W3 = 4, since you have a fear of sharks**
- Finally, we'll also assume a threshold value of 3, which would translate to a bias value of -3. With all the various inputs, we can start to plug in values into the formula to get the desired output.

$$\hat{Y} = (1*5) + (0*2) + (1*4) - 3 = 6$$

If we use the activation function from the beginning of this section, we can determine that the output of this node would be 1, since 6 is greater than 0. In this instance, you would go surfing;



Neural Networks



- But if we adjust the weights or the threshold, we can achieve different outcomes from the model. When we observe one decision, like in the above example, we can see how a neural network could make increasingly complex decisions depending on the output of previous decisions or layers.
- **In the example above**, we used perceptrons to illustrate some of the mathematics at play here, but neural networks leverage sigmoid neurons, which are distinguished by having values between 0 and 1.
- Since neural networks behave similarly to decision trees, cascading data from one node to another, having x values between 0 and 1 will reduce the impact of any given change of a single variable on the output of any given node, and subsequently, the output of the neural network.
- As we start to think about more practical use cases⁷ for neural networks, like image recognition or classification, we'll leverage supervised learning, or labeled datasets, to train the algorithm.
- **As we train the model**, we'll want to evaluate its accuracy using a cost (or loss) function. This is also commonly referred to as the mean squared error (MSE).



Neural Networks



- **i** represents the index of the sample,
- **\hat{y}** is the predicted outcome,
- **y** is the actual value, and
- **m** is the number of samples.

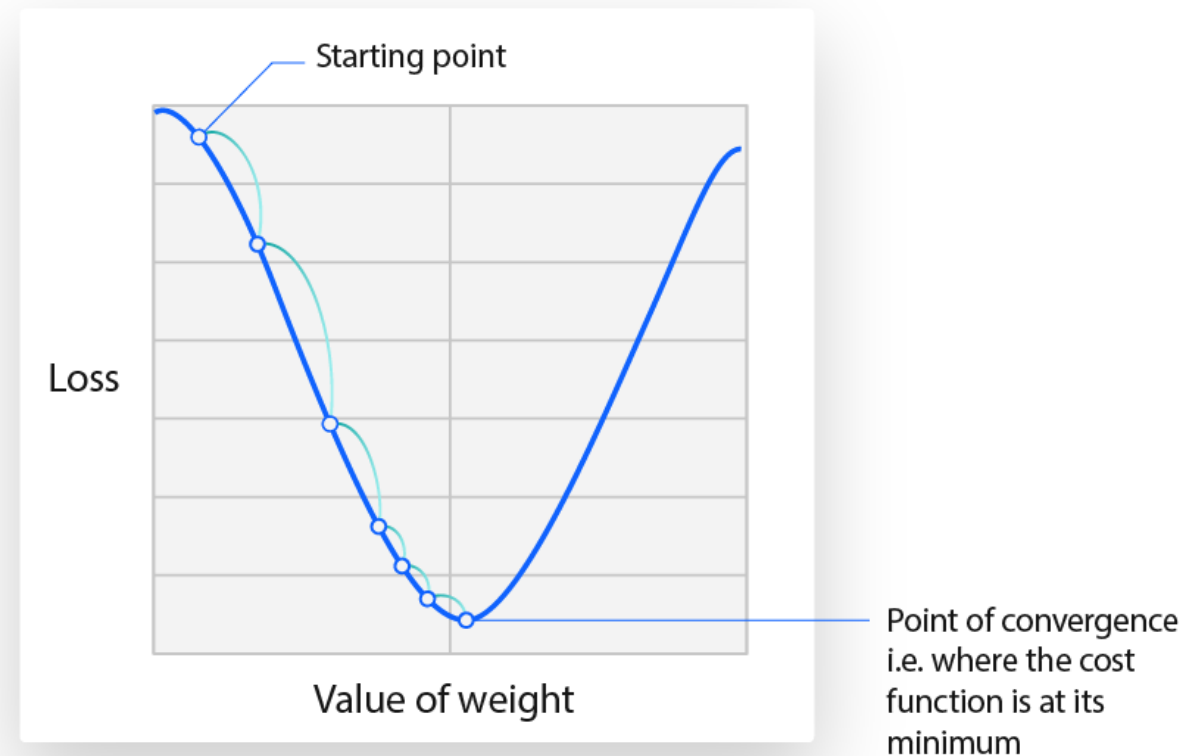
$$\text{Cost Function} = \text{MSE} = \frac{1}{2m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2$$

- Ultimately, the goal is to minimize our cost function to ensure correctness of fit for any given observation.
- As the model adjusts its weights and bias, it uses the cost function and reinforcement learning to reach the point of convergence, or the local minimum.
- The process in which the algorithm adjusts its weights is through gradient descent, allowing the model to determine the direction to take to reduce errors (or minimize the cost function). With each training example, the parameters of the model adjust to gradually converge at the minimum.
- Mostly ANN's are feed forward (flow in one direction only-from input to output.)
- May be trained into back propagation (from output to input) to calculate and attribute the error associated with each neuron, allowing us to adjust and fit the parameters of the model(s) appropriately.



Neural Networks

$$\text{Cost Function} = \text{MSE} = \frac{1}{2m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2$$





Neural Networks-Types

The **perceptron** is the oldest neural network, created by Frank Rosenblatt in 1958.

Feed forward neural networks, or multi-layer perceptrons (MLPs),

- It is comprised of an input layer, a hidden layer or layers, and an output layer.
- comprised of sigmoid neurons, not perceptrons, as most real-world problems are nonlinear.
- Data usually is fed into these models to train them, and they are the **foundation for computer vision, natural language processing, and other neural networks.**

Convolutional neural networks (CNNs) are similar to feed forward networks,

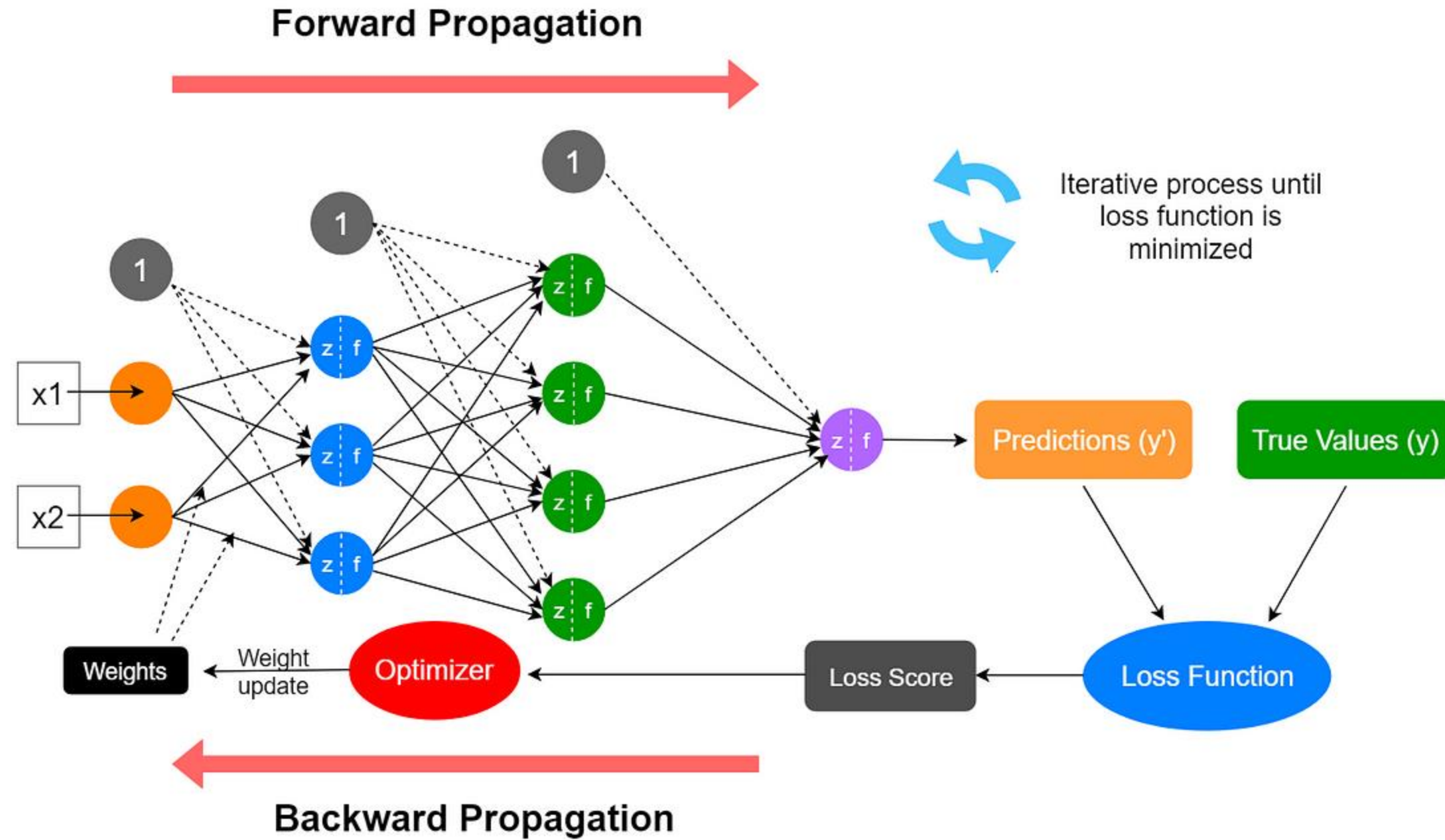
- utilized for image recognition, pattern recognition, and/or computer vision.
- principles from linear algebra, particularly matrix multiplication, to identify patterns within an image.

Recurrent neural networks (RNNs) are identified by their feedback loops.

These learning algorithms are primarily leveraged when using time-series data to make predictions about future outcomes, such as stock market predictions or sales forecasting.

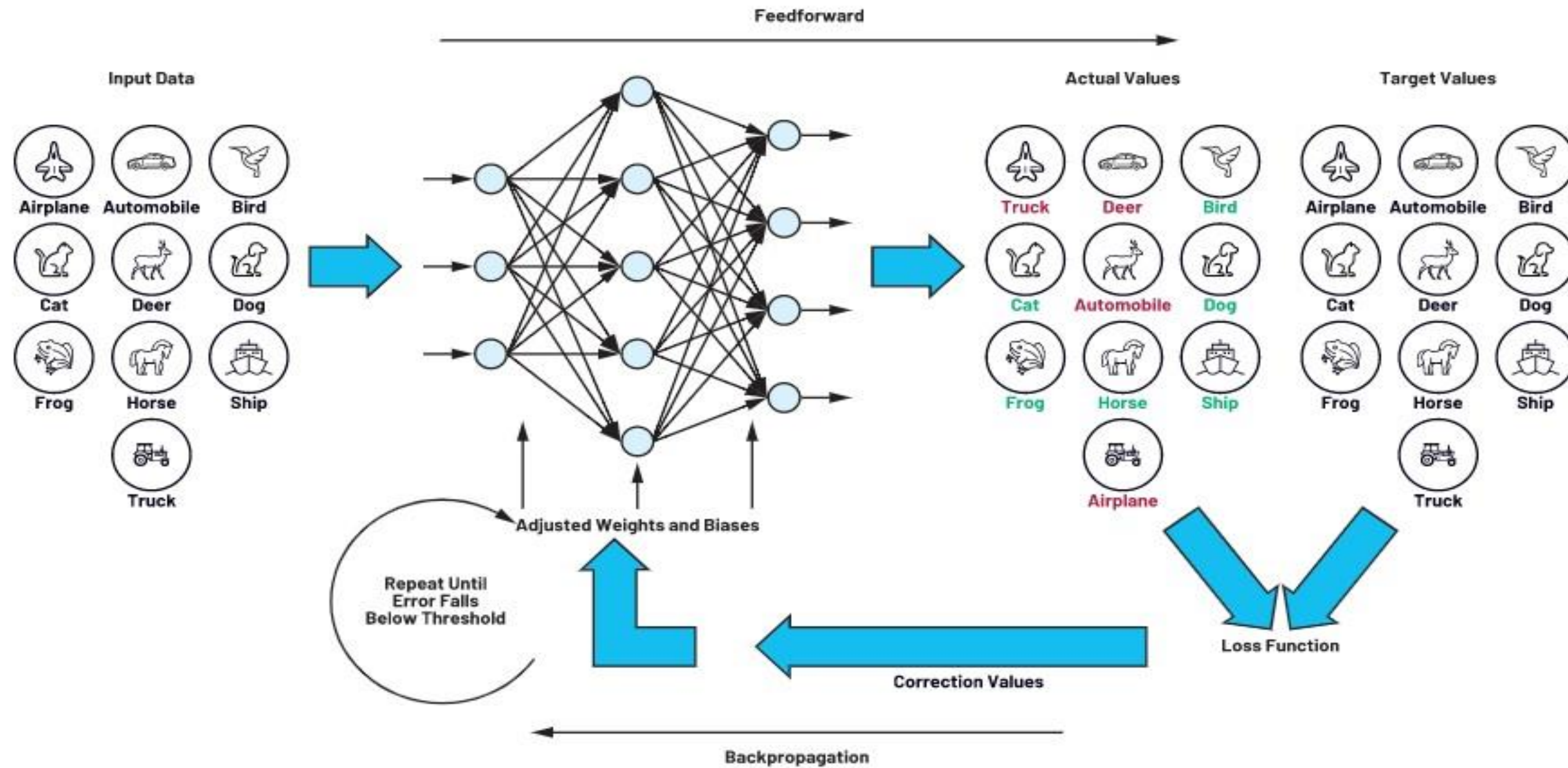


Neural Networks-Learning Process





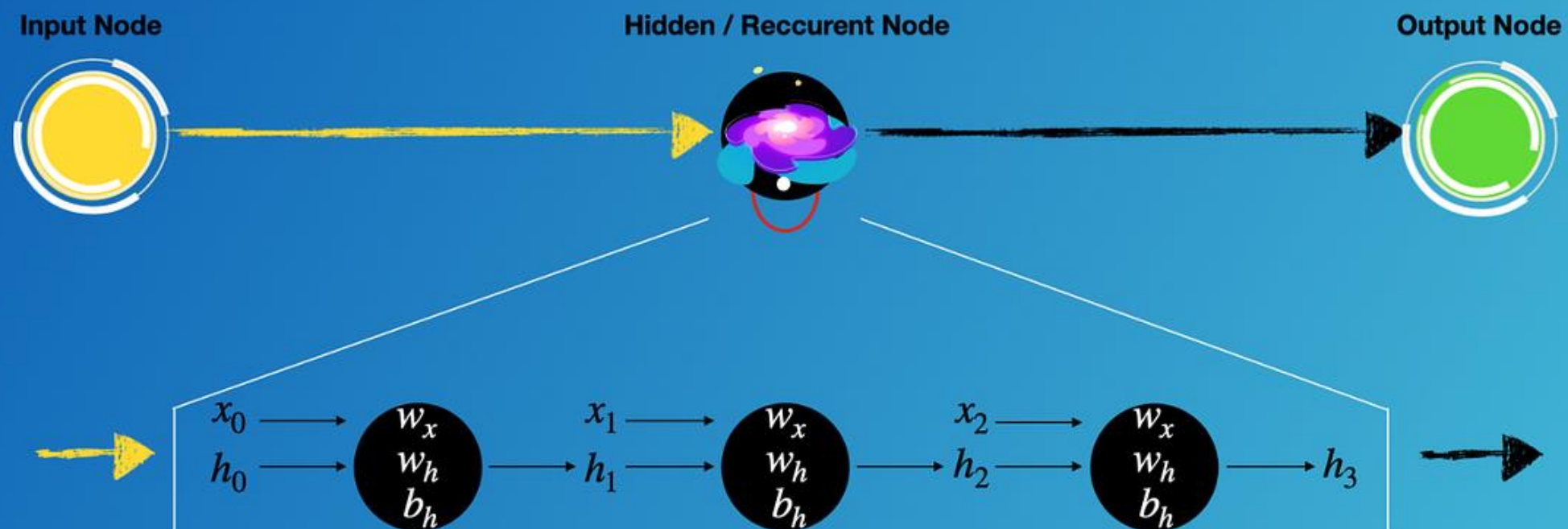
Neural Networks-Trained CNN





Neural Networks- RNN

RECURRENT NEURAL NETWORKS (RNN)





Neural Networks-Applications



1. Aerospace
2. Defence
3. Facial Recognition
4. Healthcare
5. Stock Market Prediction
6. Signature Verification and Handwriting Analysis
7. Social Media
8. Weather Forecasting

14