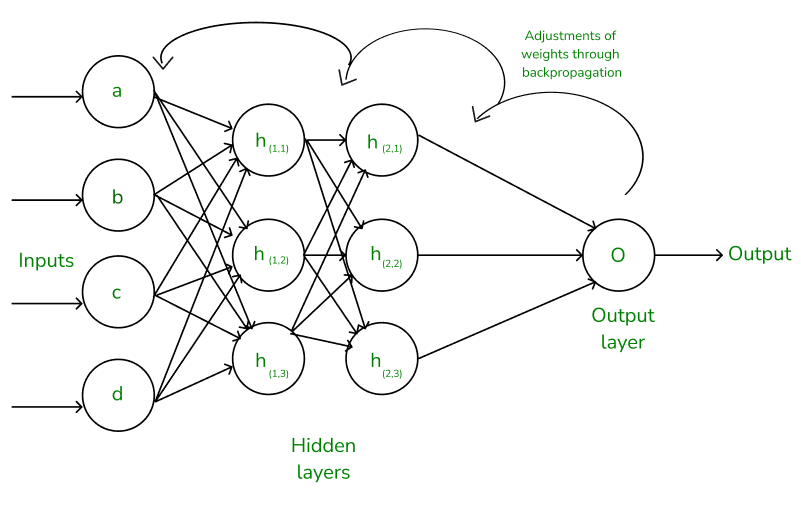
**TERMWORK – 7**

**BACKPROPOGATION**

Machine learning models learn from data and make predictions. One of the fundamental concepts behind training these models is backpropagation. In machine learning, backpropagation is an effective algorithm used to train artificial neural networks, especially in feed-forward neural networks.

Backpropagation is an iterative algorithm, that helps to minimize the cost function by determining which weights and biases should be adjusted. During every epoch, the model learns by adapting the weights and biases to minimize the loss by moving down toward the gradient of the error.



**Advantages of Using the Backpropagation Algorithm in Neural Networks:**

1. Ease of Implementation
2. Simplicity and flexibility
3. Efficiency
4. Generalization
5. Scalability.

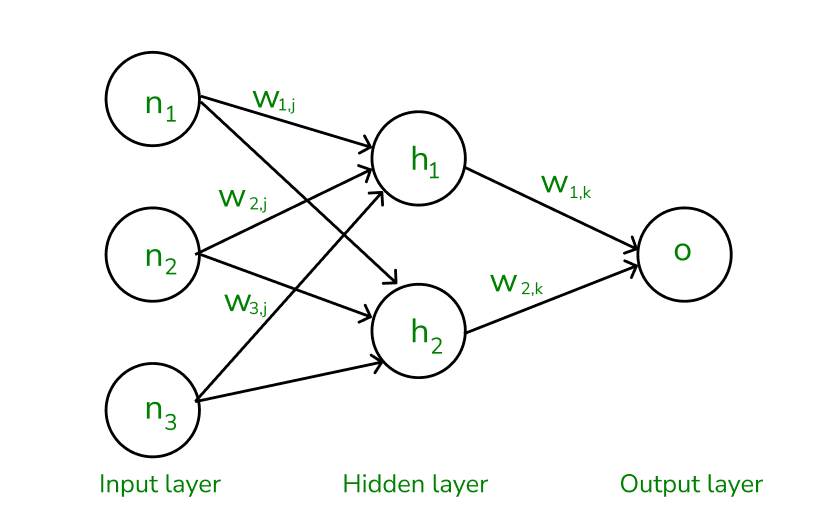
**Working of Backpropagation Algorithm**

The Backpropagation algorithm works by two different passes; they are:

1. Forward pass
2. Backward pass

**How does Forward pass work?**

* In forward pass, initially the input is fed into the input layer. Since the inputs are raw data, they can be used for training our neural network.
* The inputs and their corresponding weights are passed to the hidden layer. The hidden layer performs the computation on the data it receives. If there are two hidden layers in the neural network, for instance, consider the illustration fig(a), h1 and h2 are the two hidden layers, and the output of h1 can be used as an input of h2. Before applying it to the activation function, the bias is added.
* To the weighted sum of inputs, the activation function is applied in the hidden layer to each of its neurons. One such activation function that is commonly used is ReLU can also be used, which is responsible for returning the input if it is positive otherwise it returns zero. By doing this so, it introduces the non-linearity to our model, which enables the network to learn the complex relationships in the data. And finally, the weighted outputs from the last hidden layer are fed into the output to compute the final prediction, this layer can also use the activation function called the softmax function which is responsible for converting the weighted outputs into probabilities for each class.



*The forward pass using weights and biases*

**How does backward pass work?**

* In the backward pass process shows, the error is transmitted back to the network which helps the network, to improve its performance by learning and adjusting the internal weights.
* To find the error generated through the process of forward pass, we can use one of the most commonly used methods called mean squared error which calculates the difference between the predicted output and desired output. The formula for mean squared error is:

𝑀𝑒𝑎𝑛𝑠𝑞𝑢𝑎𝑟𝑒𝑑𝑒𝑟𝑟𝑜𝑟=(𝑝𝑟𝑒𝑑𝑖𝑐𝑡𝑒𝑑𝑜𝑢𝑡𝑝𝑢𝑡–𝑎𝑐𝑡𝑢𝑎𝑙𝑜𝑢𝑡𝑝𝑢𝑡)2*Meansquarederror*=(*predictedoutput*–*actualoutput*)2

* Once we have done the calculation at the output layer, we then propagate the error backward through the network, layer by layer.
* The key calculation during the backward pass is determining the gradients for each weight and bias in the network. This gradient is responsible for telling us how much each weight/bias should be adjusted to minimize the error in the next forward pass. The chain rule is used iteratively to calculate this gradient efficiently.
* In addition to gradient calculation, the activation function also plays a crucial role in backpropagation, it works by calculating the gradients with the help of the derivative of the activation function.