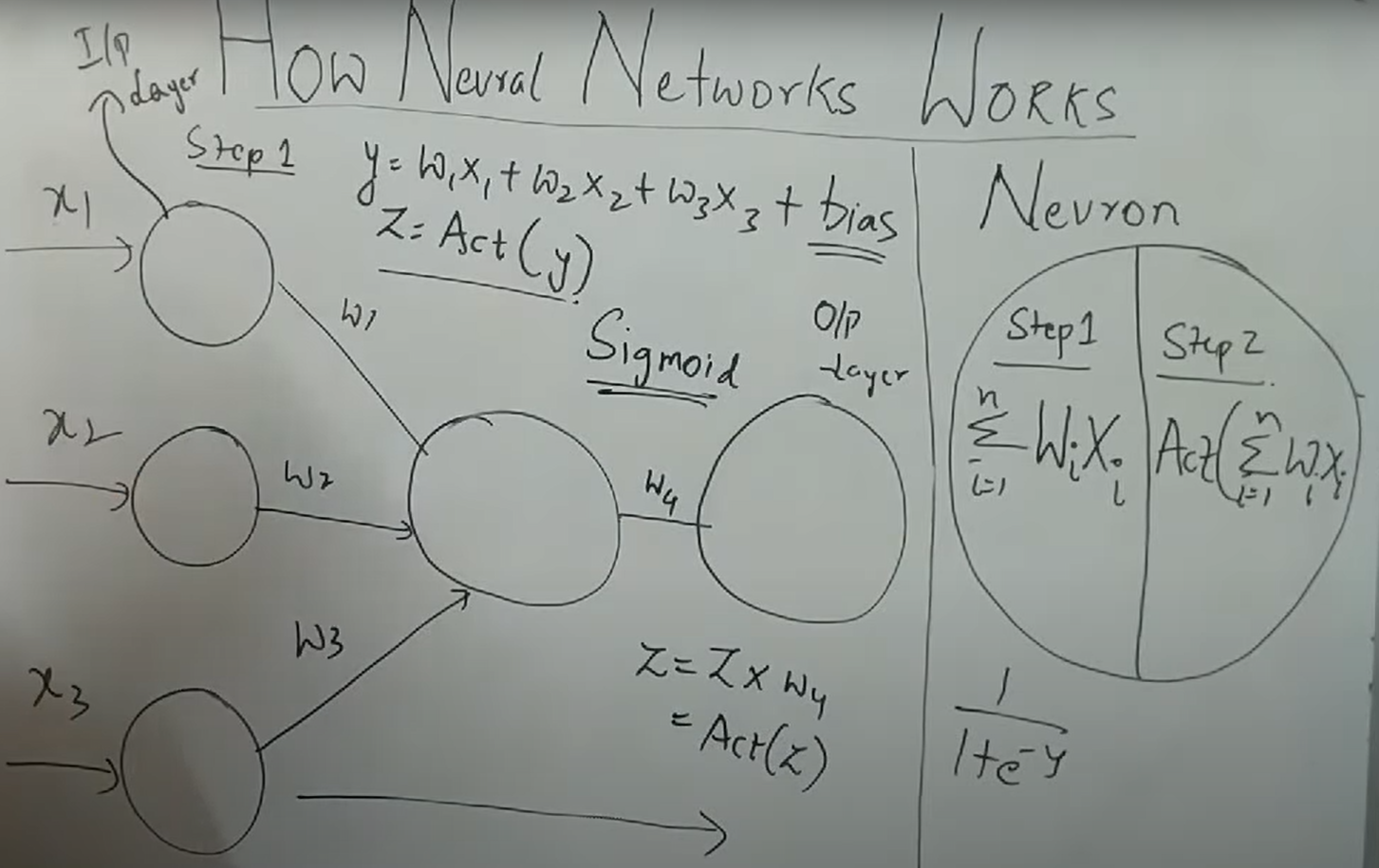
**Unit 2  
Neural Network Working**

📊 Diagram: Working of a Neural Network



**2.1 Forward Propagation**

Forward propagation is the process of moving inputs through the layers of a neural network to produce an output.  
• Input values (x1, x2, x3...) are multiplied with weights (w1, w2, w3...).  
• Weighted sum + bias is calculated at each neuron.  
• The result passes through an activation function to introduce non-linearity.  
• This continues layer by layer until the output layer generates final predictions.

**2.2 Activation Function**

Activation functions decide whether a neuron should be activated and introduce non-linear properties to the network.  
Common activation functions:  
• Sigmoid: squashes output between 0 and 1, often used in binary classification.  
• ReLU (Rectified Linear Unit): outputs 0 if input < 0 else input, widely used for hidden layers.  
• Tanh: outputs between -1 and 1, useful for centered outputs.  
These functions help neural networks approximate complex non-linear functions.

**2.3 Backward Propagation**

Backward propagation (backprop) is the process of updating weights based on the error from output.  
• The loss is calculated by comparing predicted output with actual output.  
• Error is propagated backward from the output layer to hidden layers.  
• Gradients of loss with respect to each weight are computed using the chain rule.  
• Weights are updated using gradient descent to minimize the loss.  
This process allows the network to learn by reducing prediction errors iteratively.

**2.4 Cost Function and Loss Function**

• Loss Function: Measures error for a single training example (e.g., Mean Squared Error, Cross-Entropy Loss).  
• Cost Function: Average of loss values across the entire training dataset.  
Purpose:  
– Guides weight updates by quantifying prediction error.  
– Lower cost indicates better performance of the model.  
Choosing the right cost/loss function depends on the type of problem (regression, classification, etc.).

**Summary**

• Forward propagation moves inputs through layers to generate outputs.  
• Activation functions introduce non-linearity (Sigmoid, ReLU, Tanh).  
• Backpropagation computes gradients and updates weights iteratively.  
• Loss function measures error for a sample, cost function averages across dataset.  
• Together, these steps enable neural networks to learn patterns effectively.