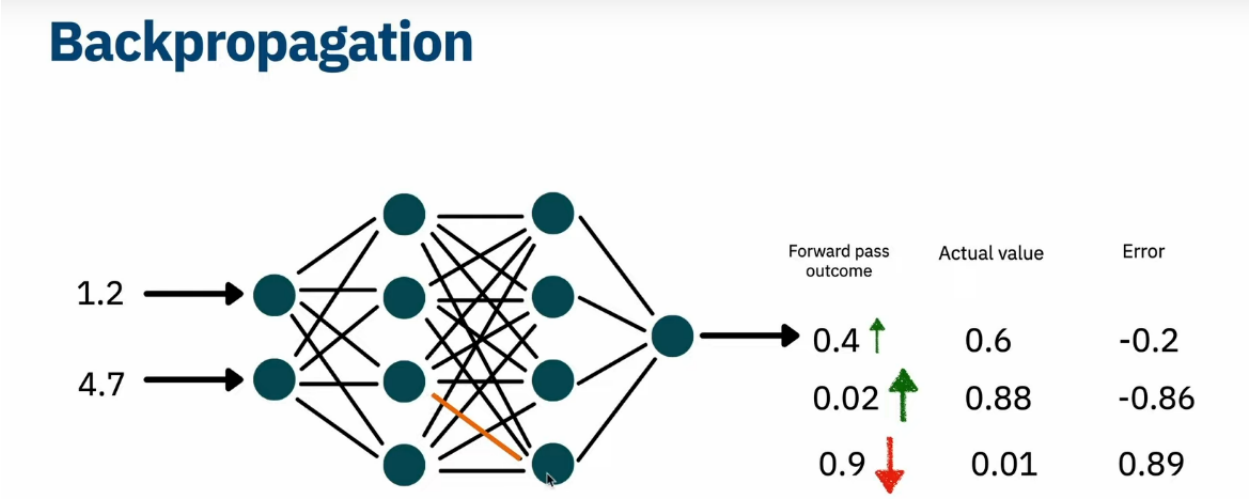
Back propagation:-

→ Back propagation is a fundamental concept in the training of neural networks, particularly in the context of supervised learning.

→ It is an optimization algorithm used to update the weights of the neural network to minimize the error between the predicted outputs and the actual targets (labels) for a given set of training data.

Goal of back propagation :-

→ to minimize the prediction error of a neural network by adjusting its weights based on the training data.



→ Here ,we have 2 inputs,and some hidden layers with weights and bias and an output layer.

→ Back propagation compares the predicted output (0.4, 0.02, 0.9) with the actual output (0.6, 0.88, 0.01) to calculate the errors.

→ It then adjusts the network's weights and biases based on these errors to reduce the discrepancy between predicted and actual values.

→ The process iterates multiple times to improve the neural network's accuracy in predicting future outputs.

(Back propagation maa sabai parameters like weights,bias and other parameters haru le kati ra kasari error/loss maa contribute gari rako xan vanni calculate garxa)

→ Back propagation calculates the relationship between the parameters and the error.

→ Then the **GRADIENT DESCENT** algorithm comes in play.
Gradient descent works by taking small steps in the direction that decreases the error, helping the neural network learn and improve its predictions.

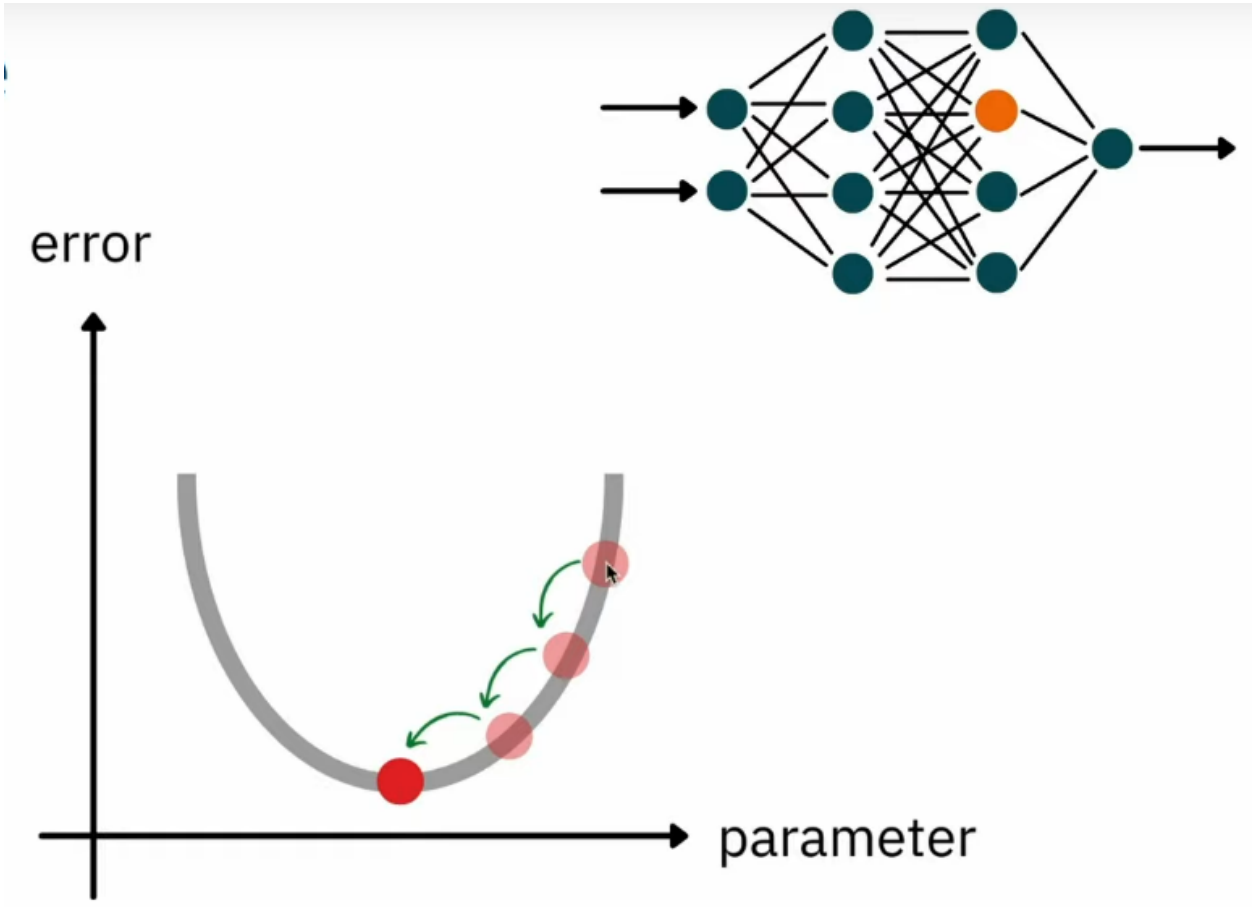
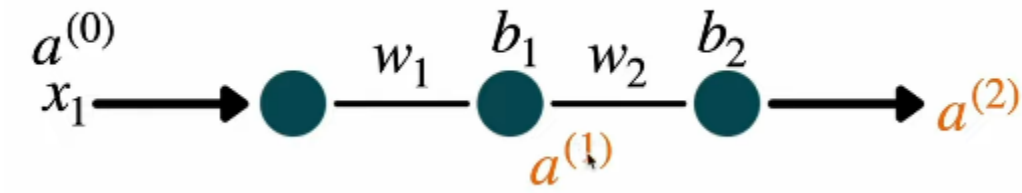


Fig:- one parameter affecting the error

Learning Rate:-

→Learning rate is a parameter that determines the step size taken during each iteration of the gradient descent algorithm in training a neural network.



Mathematical explanation of back propagation:-

→ True value : y

→ Cost : C = y – a(2)

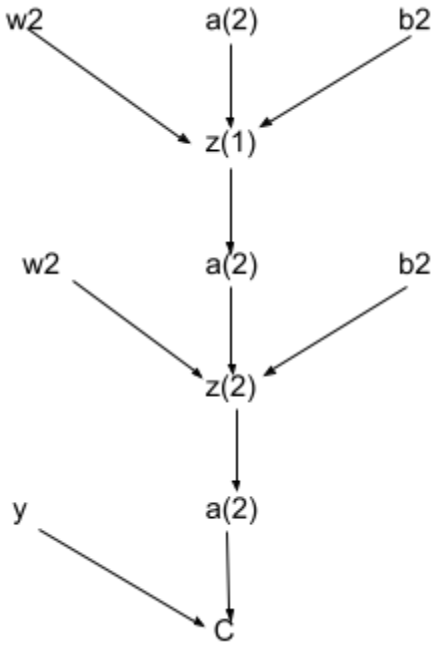
a(2) = a(z(2))

$z(2) = w_2 a(1) + b2$

a(1) = a(z(1))

$z(2) = w_1 a(0) + b1$

→ cost 'C' typically represents the loss or error between the predicted output and the actual target for a given input.



→**this tree defines how a single parameter affects the cost.**

→ Mathi ko tree ma euta parameter matrai change garda ,loss/cost lai kasari affect garxa..
→ So,tesko laagi we need to find the derivative of the cost with respect to all parameters .
→ jaslai partial derivative vaninxa…

→$\frac{dC}{dw_2}$=This is the partial derivative of cost(C),with respect to first weight (w_2).

→ this ratio just tells , how we should change the w_2 to get minimized cost (reduce error)

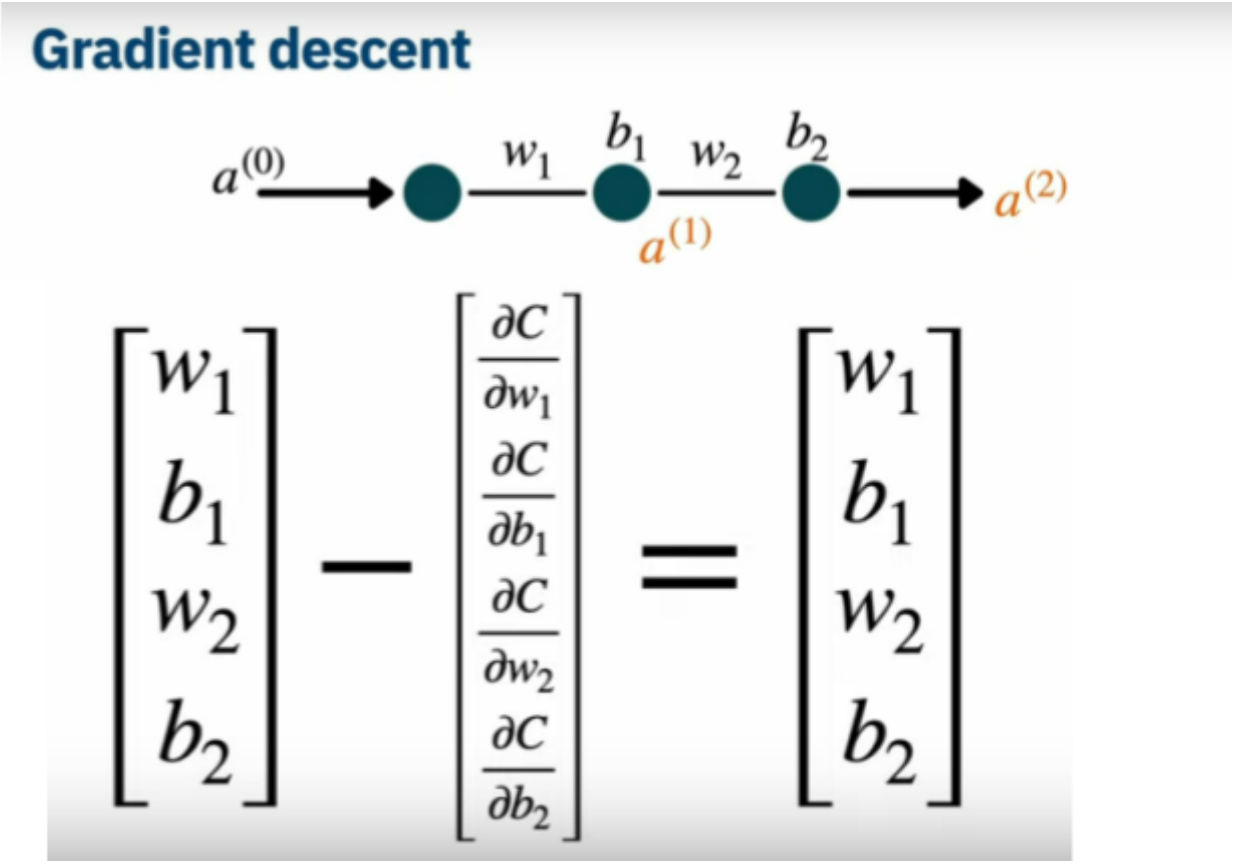
→$\frac{dC}{dw_2}=\frac{dz^2}{dw_2}*\frac{da^2}{dz^2}*\frac{dC}{da^2}$

→This value is calculated for all examples and then averaged.
As a result we get the gradient vector.

$$\text{Gradient} = \begin{bmatrix} \frac{\partial C}{\partial w_1} \\ \frac{\partial C}{\partial b_1} \\ \frac{\partial C}{\partial w_2} \\ \cdots \\ \frac{\partial C}{\partial w_n} \end{bmatrix}$$

→The gradient vector is a collection of partial derivatives of the loss function with respect to each parameter in the model.

→ yo chahi error/cost ghatauna use hunxa .



→ Yesma hamile simply weight vector maa gradient vector subtract gariyeko cha.

→**Normally, we have the following formula to update the weight vector:-**

→ *New Weight Vector = Current Weight Vector - (Learning Rate) * Gradient Vector*