



















Backpropagation- forward pass and Gradient calculations

- As seen in the previous slide we perform the forward pass to calculate
 - linear combination(or a function) of inputs
 - sigmoid (activation function) of the above function
- We use the value from the output layer to predict the value of y
- we calculate loss function Using MSE or Cross entropy
- to update weights $w_t = w_{t-1} + \nabla L$, we need to find gradient of L w.r.t all weights

Backpropagation-backward pass

- during backward pass we propagate the gradient through all nodes
 - first, we take a sum of products of all incoming gradient from the next nodes and weights of current node

- $$\sum_{i=1}^{NodesInNextLayer} \frac{\partial L}{\partial h_{layer,i}} w_{layer,i}$$

- e.g.
$$\left(\frac{\partial L}{\partial h_{21}} + \frac{\partial L}{\partial h_{22}} \right) w_3$$

- where h_{21}, h_{22} are nodes in the next layer in MLP connect to current node h_{11}
- w_3 is the weight of the current layer h_{11}

- multiply this sum with the derivative of the sigmoid function

- $$\left[\left(\frac{\partial L}{\partial h_{21}} + \frac{\partial L}{\partial h_{22}} \right) w_3 \right] \left[\sigma(h_{11})(1 - \sigma(h_{11})) \right]$$

- pass this value to previous node as gradient

Backpropagation - weight calculations

- using the new gradient we find the new weights as

$$w_{new} = w_{old} - \eta \left(\frac{dL}{dNode} \sigma(Node) \right)$$

- $\frac{dL}{dNode}$ the node gets from the previous node during back propagation
- $\sigma(Node)$ is its own output
- w_{old} is the node's current weight
- η is the learning rate