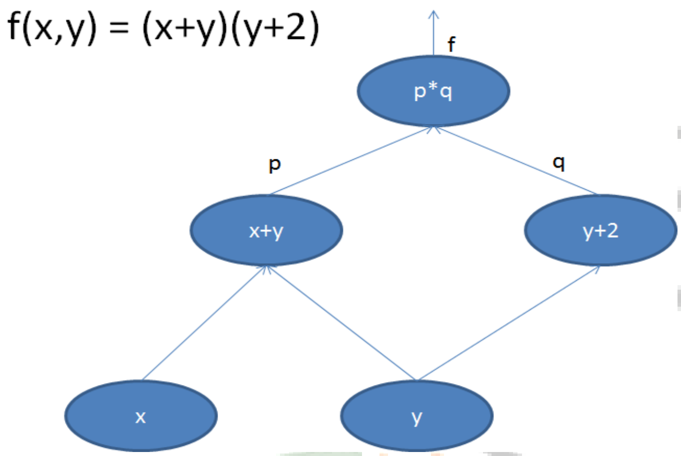
Practical Deep Learning (Backpropagation)

Problem 1: Forward vs Backward Differentiation

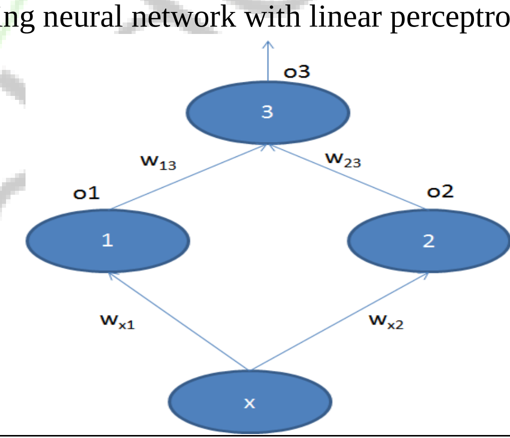
Forward-mode differentiation tracks how one input affects every node. Reverse-mode differentiation tracks how every node affects one output. That is, forward-mode differentiation applies the operator $\partial/\partial x$ to every node, while reverse mode differentiation applies the operator $\partial f/\partial$ to every node.



a. Apply forward differentiation operator $\partial/\partial y$ to each node in the graph i.e., compute $\partial y/\partial y$, $\partial p/\partial y$, $\partial q/\partial y$, $\partial f/\partial y$, $\partial x/\partial y$.
b. Apply forward differentiation operator $\partial/\partial x$ to each node in the graph i.e., compute $\partial y/\partial x$, $\partial p/\partial x$, $\partial q/\partial x$, $\partial f/\partial x$, $\partial x/\partial x$.
c. Apply backward differentiation operator $\partial f/\partial$ to each node in the graph i.e., compute $\partial f/\partial p$, $\partial f/\partial q$, $\partial f/\partial y$, $\partial f/\partial x$, $\partial f/\partial f$.
d. Which mode of differentiation is efficient to compute $\partial f/\partial x$, $\partial f/\partial y$.

Problem 2: Backpropagation in neural network with linear perceptrons

Given the following neural network with linear perceptrons, do the following:



www.algorithmica.co.in
Ph: +91-9246582537

a. Apply backward differentiation operator $\partial E/\partial$ to each weight in the graph i.e., Compute $\partial E/\partial w_{13}$, $\partial E/\partial w_{23}$, $\partial E/\partial w_{x1}$, $\partial E/\partial w_{x2}$ using back propagation.

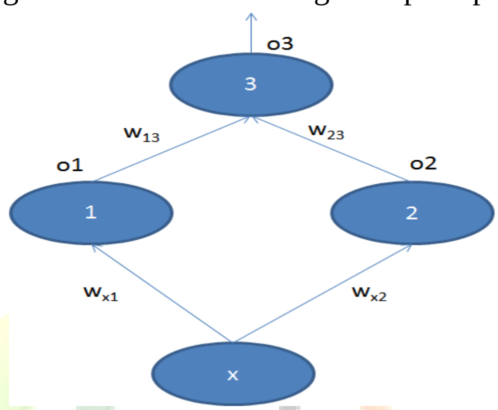
b. Given x=1, y = 6, apply gradient descent algorithm to find the values of weights that minimize the squared loss function, E, defined in the class. Assume following equation for gradient updates and also assume the initial weight values as 1.

$$w_{ij} = w_{ij} - 0.01 * \partial E/\partial w_{ij}$$

Repeat the algorithm for 3 iterations and observe how the weights gets adjusted

Problem 3: Backpropagation in neural network with sigmoid perceptrons

Given the following neural network with sigmoid perceptrons, do the following:



a. Apply backward differentiation operator $\partial E/\partial$ to each weight in the graph i.e., Compute $\partial E/\partial w_{13}$, $\partial E/\partial w_{23}$, $\partial E/\partial w_{x1}$, $\partial E/\partial w_{x2}$ using back propagation.

b. Given x=0.2, y = 1, apply gradient descent algorithm to find the values of weights that minimize the squared loss function, E, defined in the class. Assume following equation for gradient updates and also assume the initial weight values as 1.

$$w_{ij} = w_{ij} - 0.01 * \partial E/\partial w_{ij}$$

Repeat the algorithm for 3 iterations and observe how the weights gets adjusted