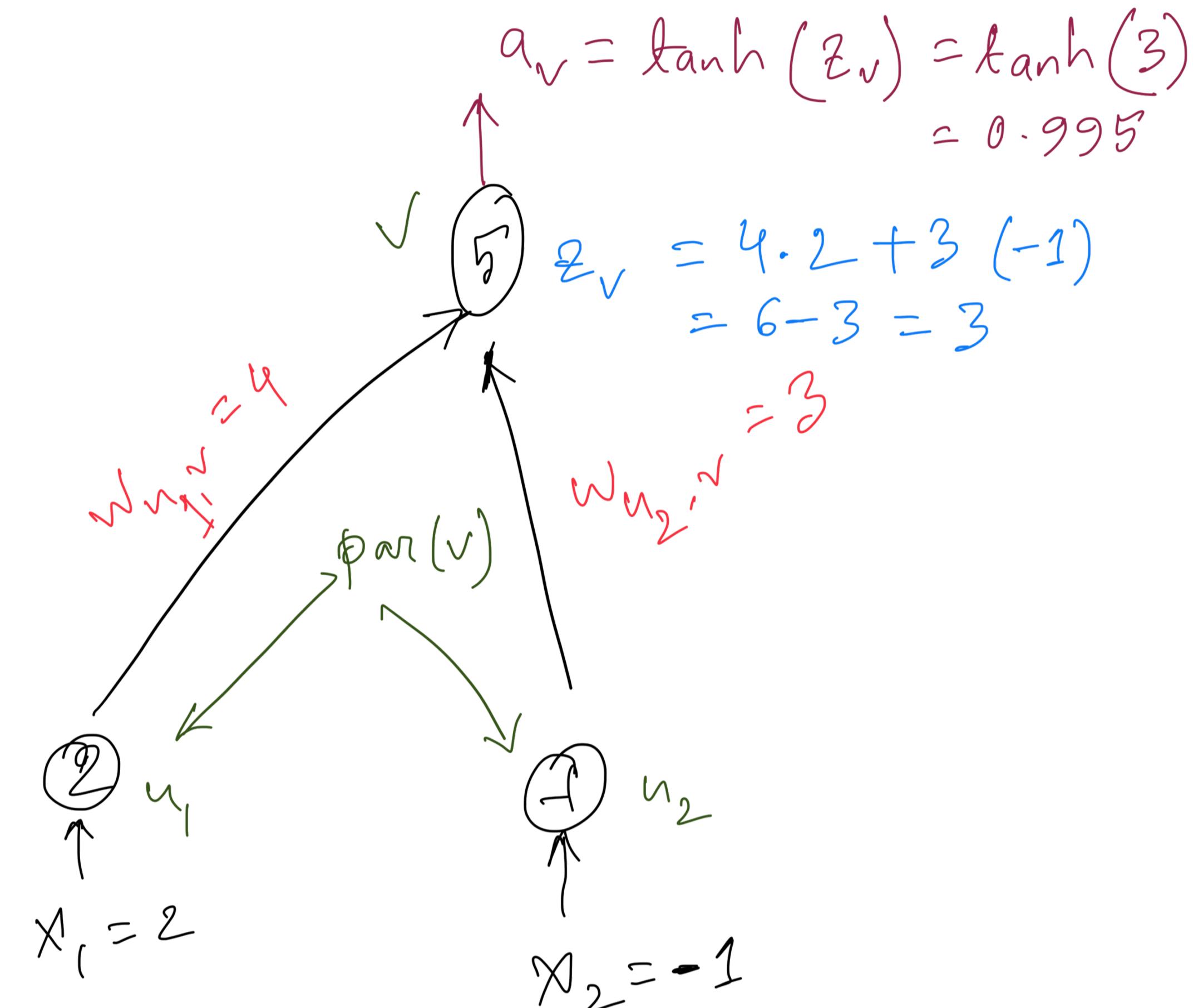


Figure : multi-layer network

Algorithm 27 : FORWARDPROPAGATION ( $x$ )

```

1: for all input nodes  $u$  do
     $a_u \leftarrow$  corresponding feature of  $x$ 
end for
4: for all nodes  $v$  in the network whose parents are computed do
5:    $z_v \leftarrow \sum_{u \in \text{par}(v)} w_{(u,v)} a_u$ 
6:    $a_v \leftarrow \tanh(z_v)$ 
end for
8: return  $z_y$  //assuming no non-linearity applied in final layer
  
```



Algorithm 28 : BACKPROPAGATION ( $x, y$ )

```

1: run FORWARDPROPAGATION ( $x$ ) to compute activations
2:  $e_y \leftarrow y - z_y$  //overall network error
3: for all nodes  $v$  in the network whose error  $e_v$  is computed do
    for all  $u \in \text{par}(v)$  do
         $g_{u,v} \leftarrow -e_v a_u$  //gradient of this edge
         $e_u \leftarrow e_u + e_v w_{v,u} (1 - \tanh^2(z_u))$  //compute "error" of parent node
    end for
end for
return all gradients  $g_e$ 
  
```

