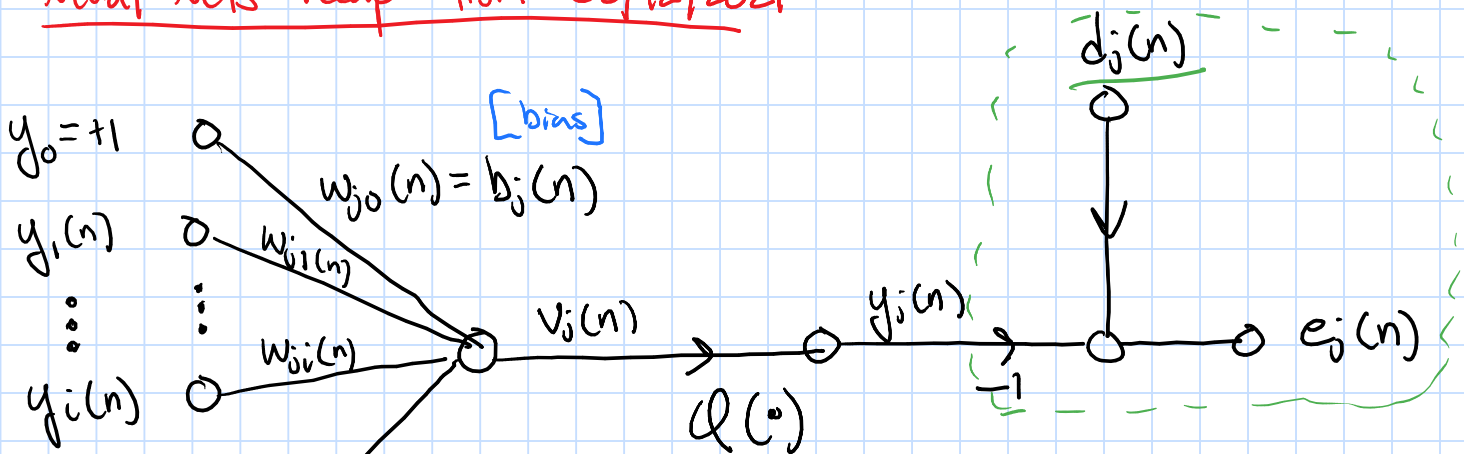
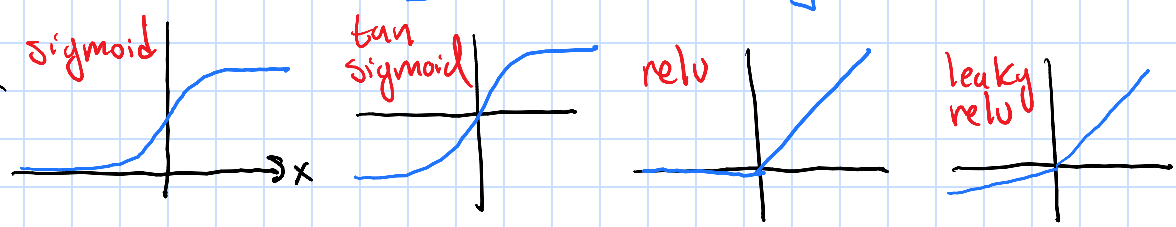
Neural Nets recap from 03/12/2021

$$e_j(n) = d_j(n) - y_j(n)$$



[activation function]



$$f(x) = 1/(1+e^{-x})$$
$$f' = f(1-f)$$

Goal: $W(t+1) = W(t) + \underbrace{\Delta W(t)}_{-\eta \frac{\partial E}{\partial w}}$ [gradient descent]

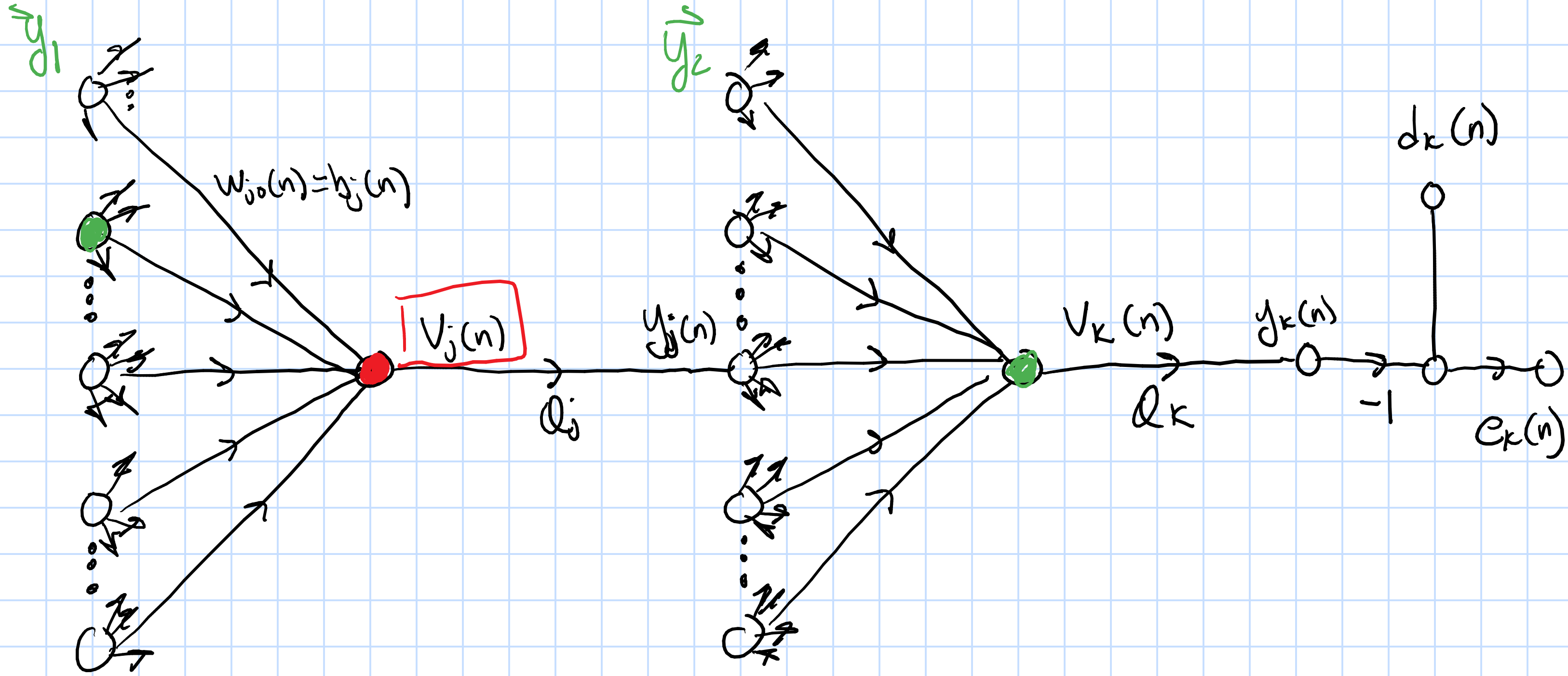
$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

$$\frac{\partial E}{\partial e_j(n)} = e_j(n) \qquad \frac{\partial y_j(n)}{\partial v_j(n)} = \varphi_j'(v_j(n))$$

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \qquad \frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n)$$

$$\therefore \quad \boxed{\frac{\partial E}{\partial w_{ji}(n)} = -e_j(n)\,\varphi_j'(v_j(n))\,y_i(n)}$$

Definition: Local gradient $\quad \delta_j(n) = \frac{\partial E}{\partial v_j(n)} = e_j(n)\,\varphi_j'(v_j(n))$



Case I: Neuron j is an output node

When neuron j is located at the output layer of the network, it is supplied with a desired output (response) of its own. In this case, $e_j(n) = d_j(n) - y_j(n)$ is straightforward to find for the local gradient $\delta_j(n)$.

Case II: Neuron j is a hidden node

When neuron j is a hidden node, we are not provided a desired signal. The error at the hidden node would need to be determined recursively and working backwards in terms of the error signals of all the neurons to which that hidden neuron is directly connected. This is where backpropagation gets interesting!

Neural Networks and Learning Machines

Redefine the local gradient @ j

$$\boxed{\delta_j(n)} = \frac{\partial E(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\cancel{\partial y_j(n)}}^{-1} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} \qquad (7)$$

$$= -\frac{\partial E(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} = -\boxed{\frac{\partial E(n)}{\partial y_j(n)}} \cdot \varphi_j'(v_j(n))$$

[neuron j is a hidden node]

To calculate $\frac{\partial E}{\partial y_j}$, we need

$$E(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n) \qquad (5)$$

[neuron k is an output]

We need to diff (5) wrt $y_j(n)$

$$\frac{\partial E}{\partial y_j(n)} = \sum_k e_k(n) \boxed{\frac{\partial e_k(n)}{\partial y_j(n)}} \quad \star$$

$$= \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \cdot \frac{\partial v_k(n)}{\partial y_j(n)}$$

Note:

$$e_k(n) = d_k(n) - y_k(n)$$

$$= d_k(n) - \varphi_k(v_k(n))$$

① $$\frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi_k'(v_k(n))$$

② $$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n)$$

$$v_k(n) = \sum_{j=1}^{m} w_{kj}(n)\, y_j(n)$$

6

Combining

$$\frac{\partial E(n)}{\partial y_j(n)} = -\sum_k e_k(n)\, \varphi_k'(v_k(n))\, w_{kj}(n)$$

$$= -\sum_k \delta_k(n)\, w_{kj}(n)$$

[j is hidden]

Combining (6) and (7)

$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n)\, w_{kj}(n)$$

$$\begin{pmatrix}\text{weight} \\ \text{correction for} \\ w_{ji}(n)\end{pmatrix} = \eta \begin{pmatrix}\text{local} \\ \text{gradient} \\ \delta_j(n)\end{pmatrix} \times \begin{pmatrix}\text{Input signal} \\ \text{of neuron} \\ y_i(n)\end{pmatrix}$$

$$f(x) = 1/(1+e^{-x})$$

$$f' = f(1-f) \leq \frac{1}{4}$$