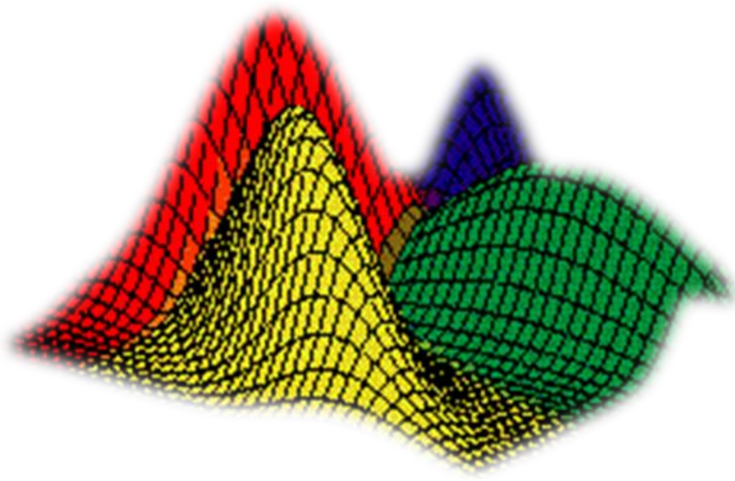


CSE 473: Pattern Recognition

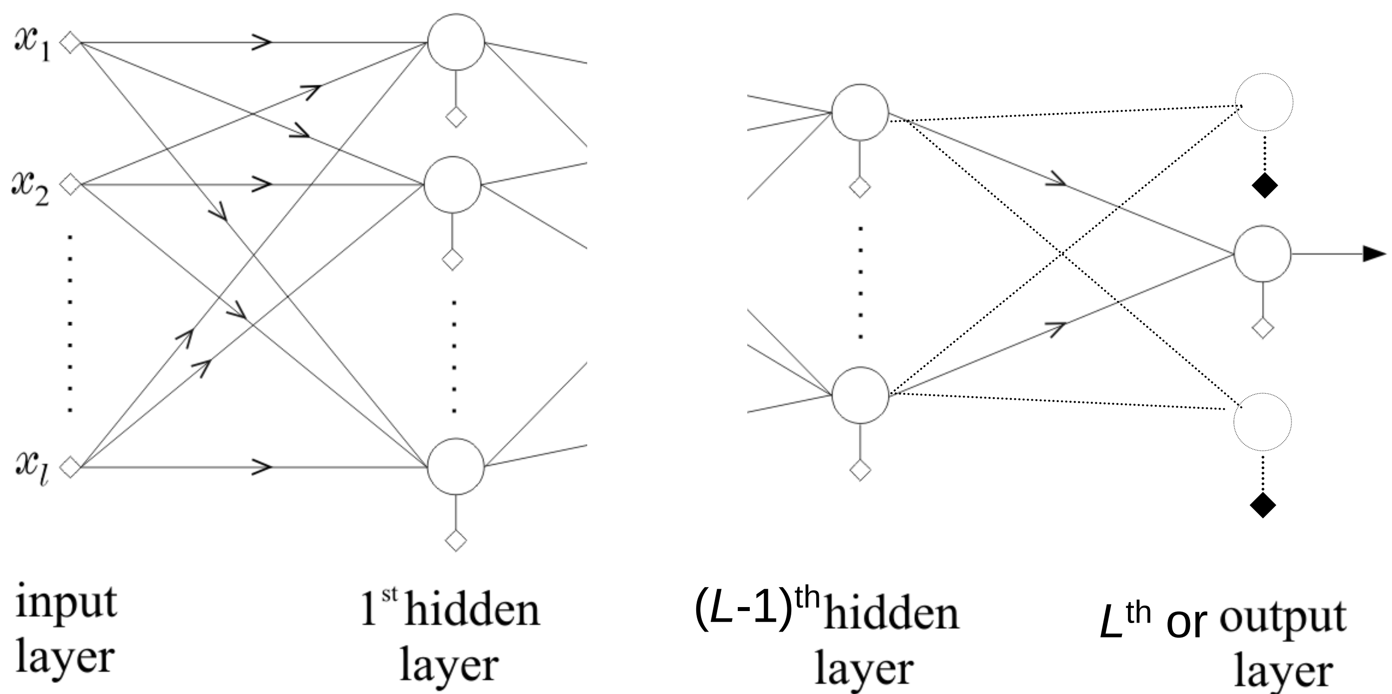
Non-Linear Classifier

Training of a Multi Layer Perceptron (MLP)

- use rationale and develop a structure that classifies **correctly all** the training patterns.

  OR

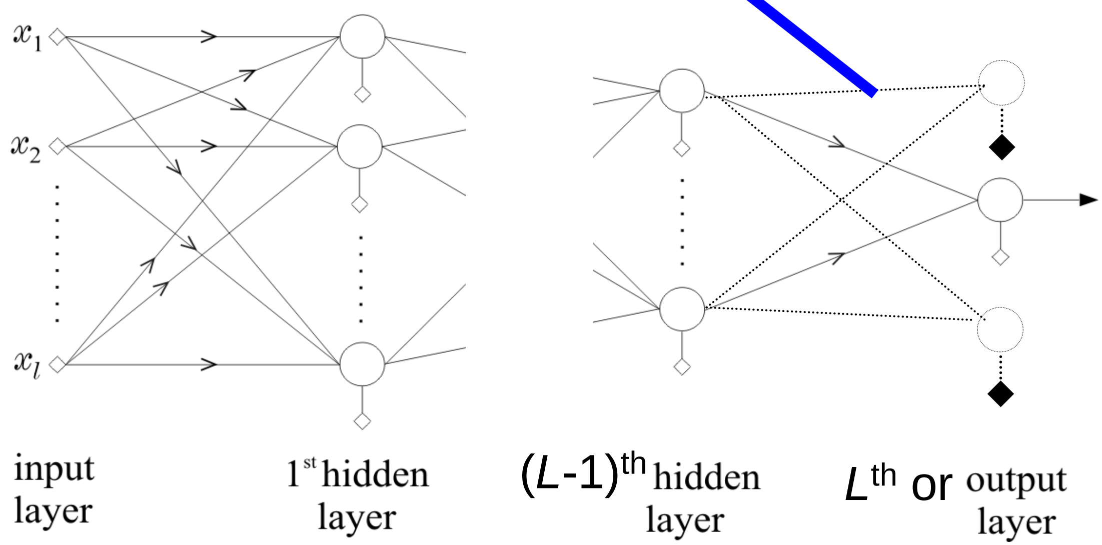
- choose a structure and compute the synaptic weights to optimize a cost function.



Iterative update of Synaptic weights: The Backpropagation Algorithm

- computes the weights iteratively, subject to a cost function is optimized

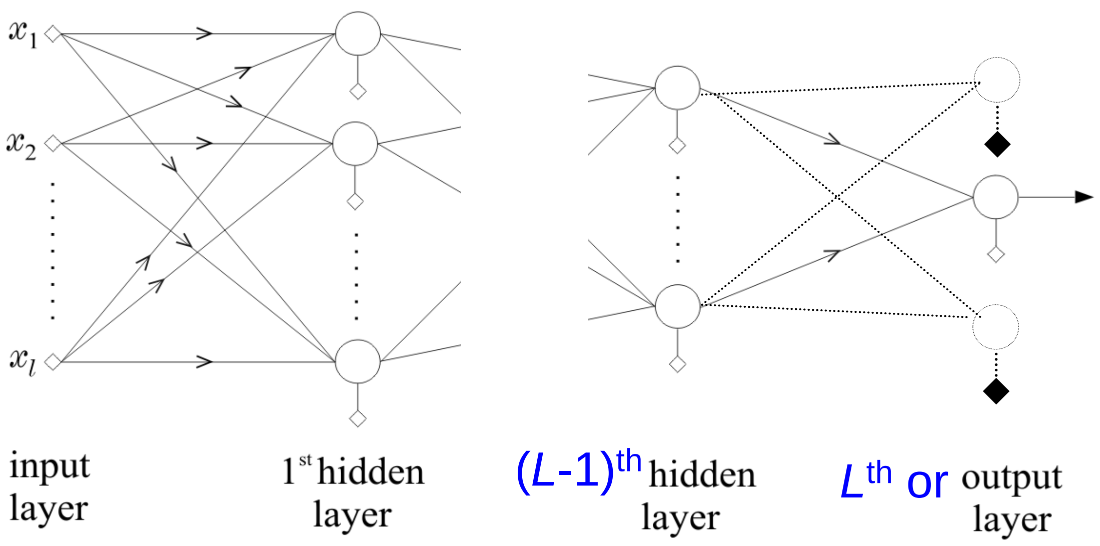
$$w_j^r(\text{new}) = w_j^r(\text{old}) + \nabla w_j^r$$



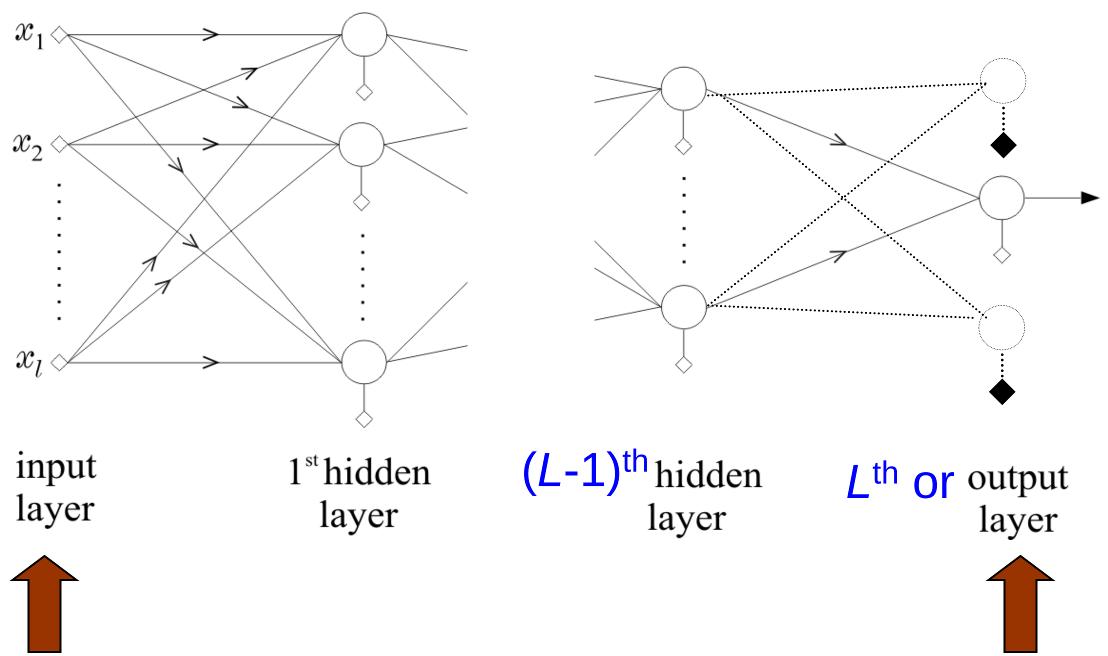
Iterative update of Synaptic weights: The Backpropagation Algorithm

Assume:

- Multiple layers
- more than one neurons in each layer
- any number of classes



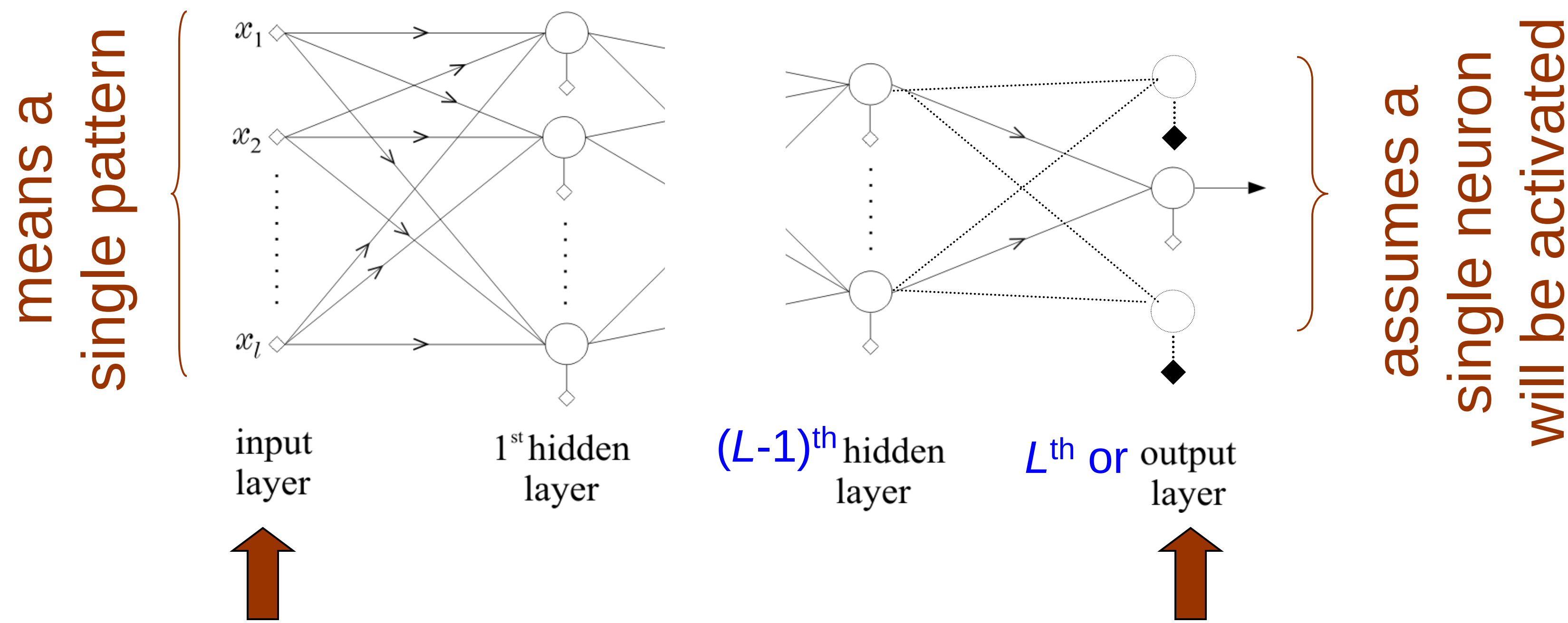
Iterative update of Synaptic weights: The Backpropagation Algorithm



Input vectors applied here

Estimated Outputs, one for each class

Iterative update of Synaptic weights: The Backpropagation Algorithm



Input vectors applied here

Estimated Outputs, one for each class

Iterative update of Synaptic weights: The Backpropagation Algorithm

Let:

- Data set has 4 classes

Training Sample#	Class#
Sample#1	1
Sample#2	3
Sample#3	2
Sample#4	4
Sample#5	2

Iterative update of Synaptic weights: The Backpropagation Algorithm

Let:

- Data set has 4 classes

Training Sample#	Class#	Class Vector
Sample#1	1	1 0 0 0
Sample#2	3	0 0 1 0
Sample#3	2	0 1 0 0
Sample#4	4	0 0 0 1
Sample#5	2	0 1 0 0

The Backpropagation Algorithm

- Recall the perceptron algorithm:
  - We update with this $\underline{w}(\text{new}) = \underline{w}(\text{old}) + \Delta\underline{w}$

The Backpropagation Algorithm

- Recall the perceptron algorithm:
  - We update with this $\underline{w}(\text{new}) = \underline{w}(\text{old}) + \Delta \underline{w}$

- Backpropagation updates multiple nodes for a number of layers:

$$w_j^r(\text{new}) = w_j^r(\text{old}) + \nabla w_j^r$$

The Backpropagation Algorithm

- Recall the perceptron algorithm:
  - We update with this $\underline{w}(\text{new}) = \underline{w}(\text{old}) + \Delta \underline{w}$

- Backpropagation updates multiple nodes for a number of layers:

$$w_j^r(\text{new}) = w_j^r(\text{old}) + \nabla w_j^r$$

r-th layer

j-th neuron

The Backpropagation Algorithm

- Another difference is the activation function:
- Perceptron algorithm uses unit activation function:

$$f(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases}$$

  - This function is not differentiable at x=0.

The Backpropagation Algorithm

- Another difference is the activation function:
- Perceptron algorithm uses unit activation function:

$$f(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases}$$

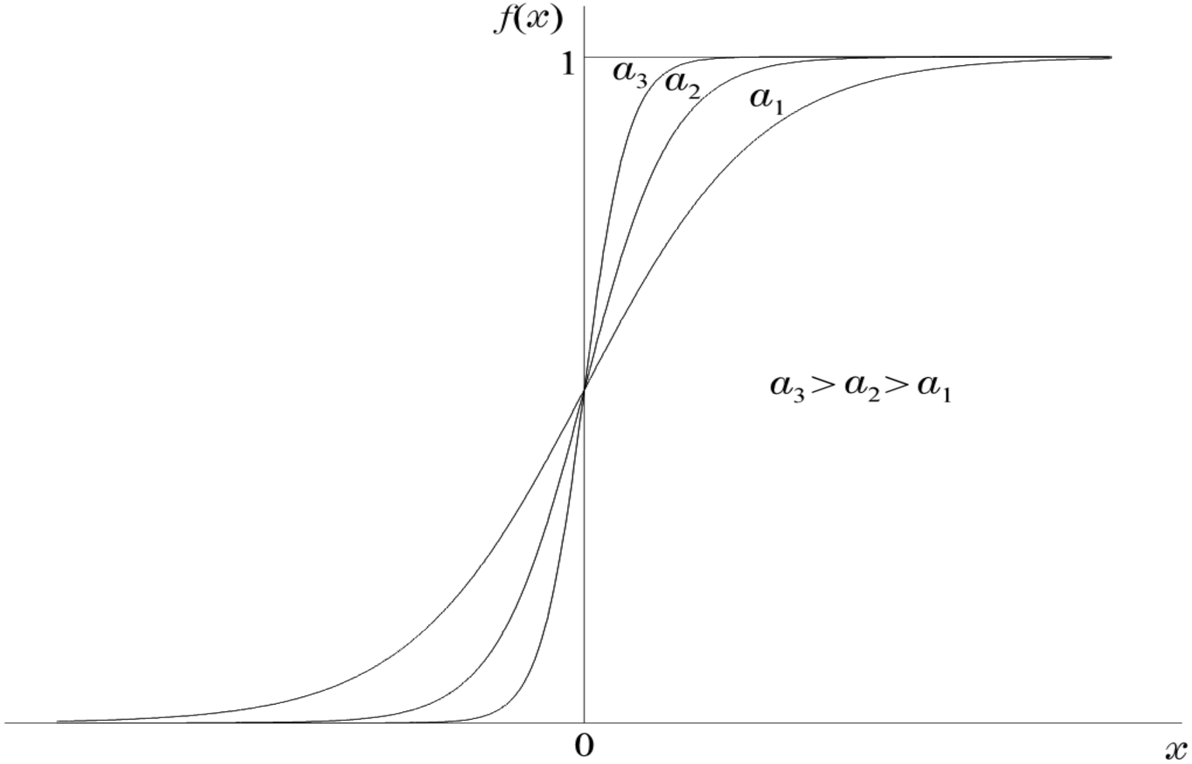
  – This function is not differentiable at x=0.

- Backpropagation uses logistic function:

$$f(x) = \frac{1}{1+\exp(-ax)}$$

Logistic function

The Logistic function



$$f(x) = \frac{1}{1 + \exp(-ax)}$$

The Backpropagation Algorithm

- Similar to perceptron algorithm: *Backpropagation also iteratively updates weights*

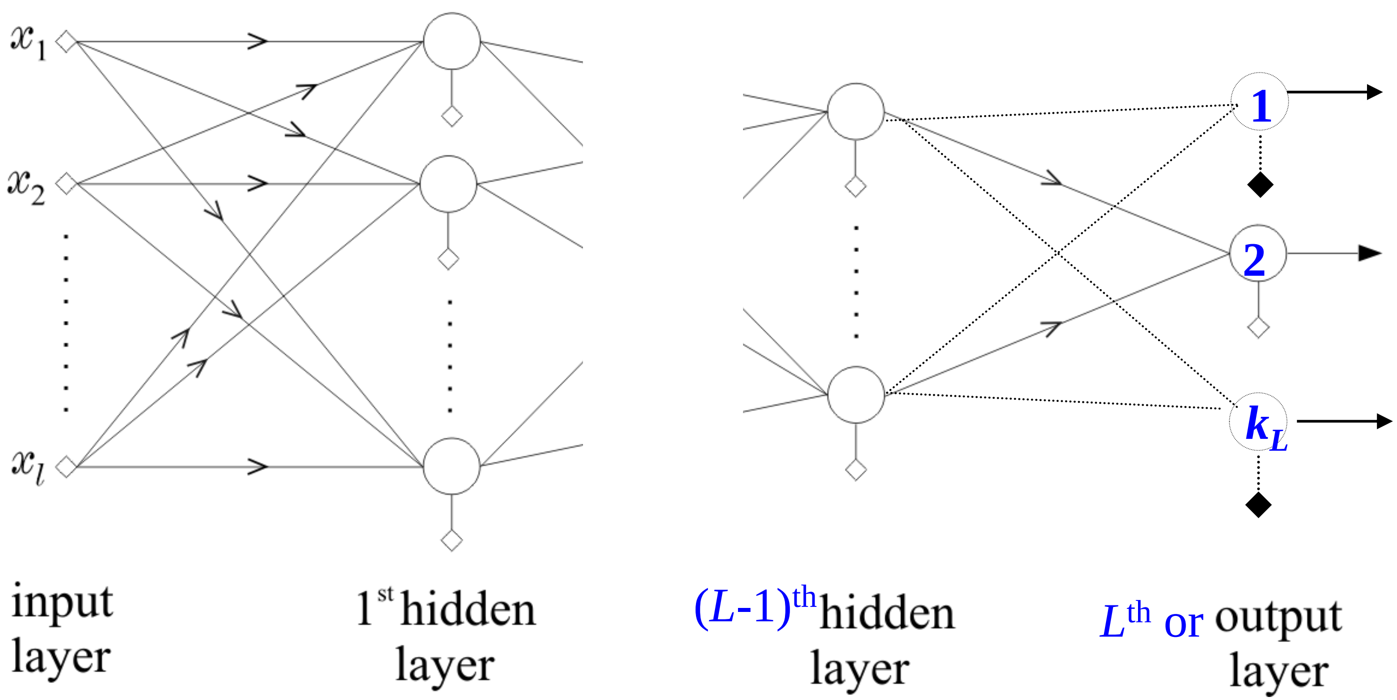
$$w_j^r(\text{new}) = w_j^r(\text{old}) + \nabla w_j^r$$

where, $$\nabla w_j^r = -\mu \frac{\partial J}{\partial w_j^r}$$

and $$J = \sum_{i=1}^{N} \varepsilon(i)$$

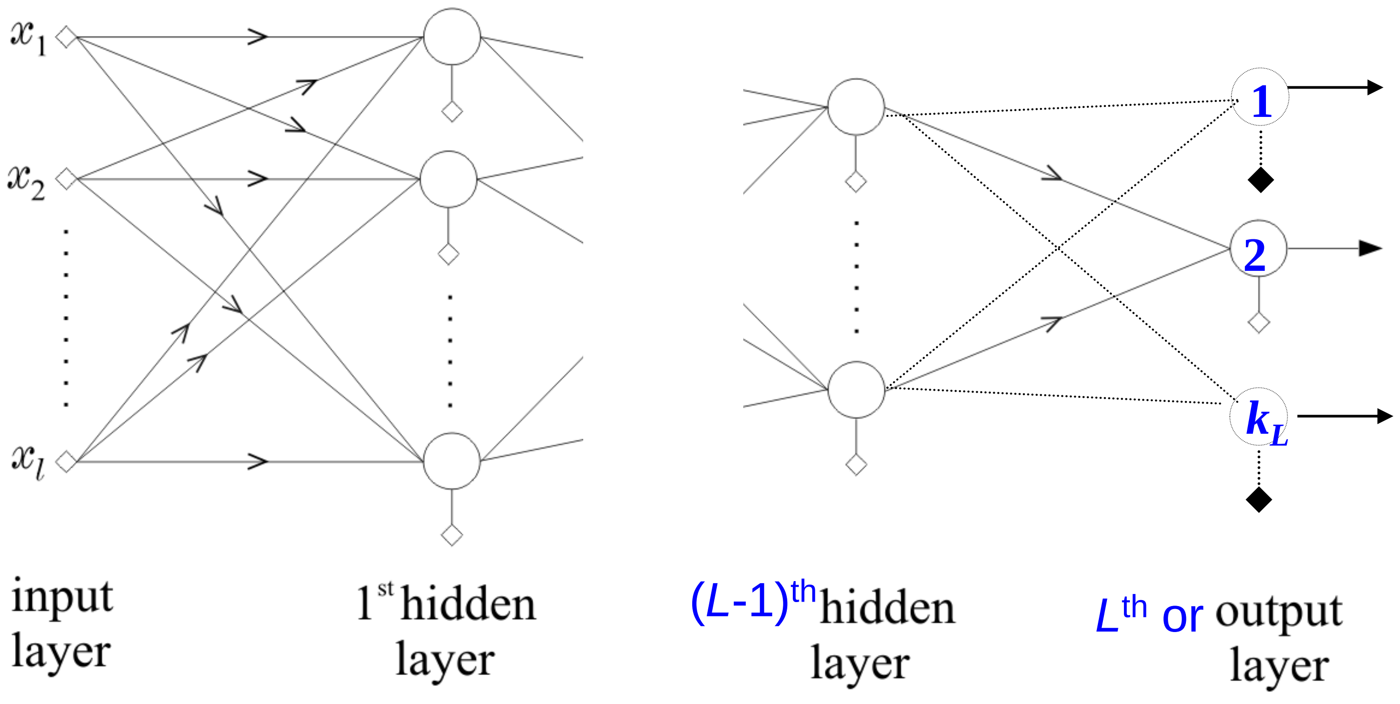
Define the Terms

- L layers of neurons
- k_r neurons in r^{th} layer
- k_0 nodes in the input layer = input feature dimension = l
- k_L nodes in the output layer = output class dimension



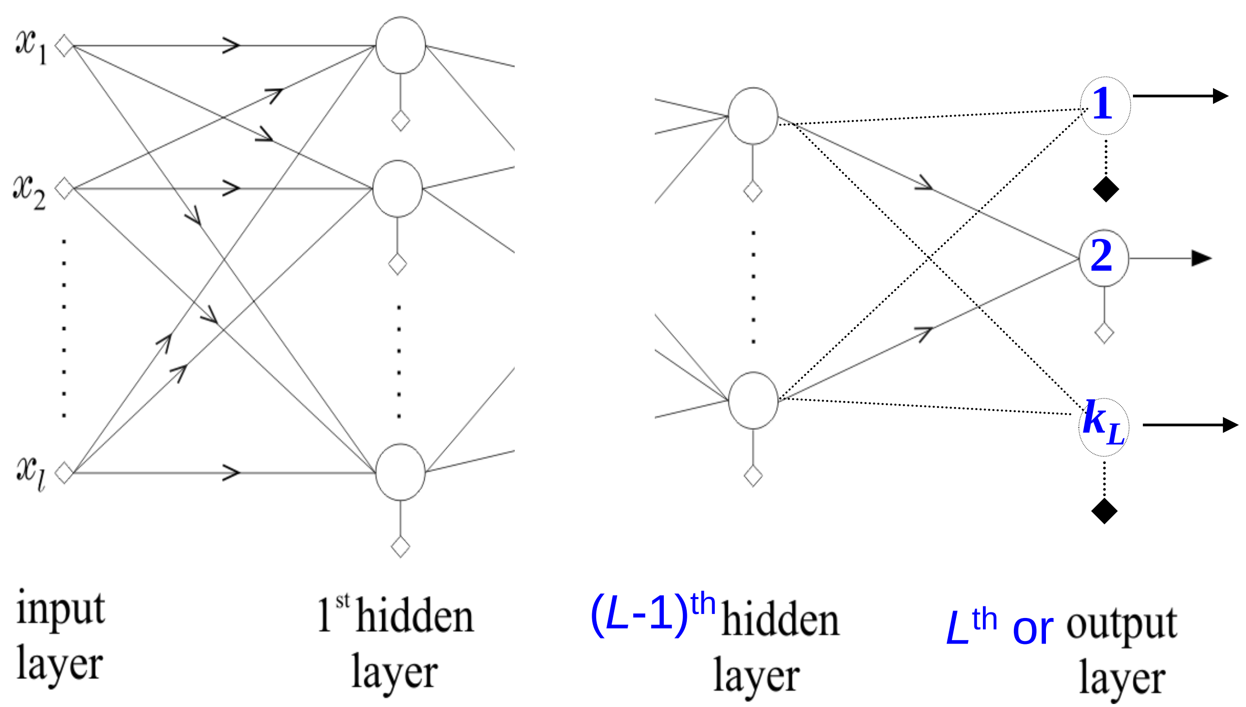
Define the Terms

- *Remember*: The number of classes is more than 2, it is K_L.
- Class value of a sample is no longer a single variable, rather it is a vector of k_L dimension.



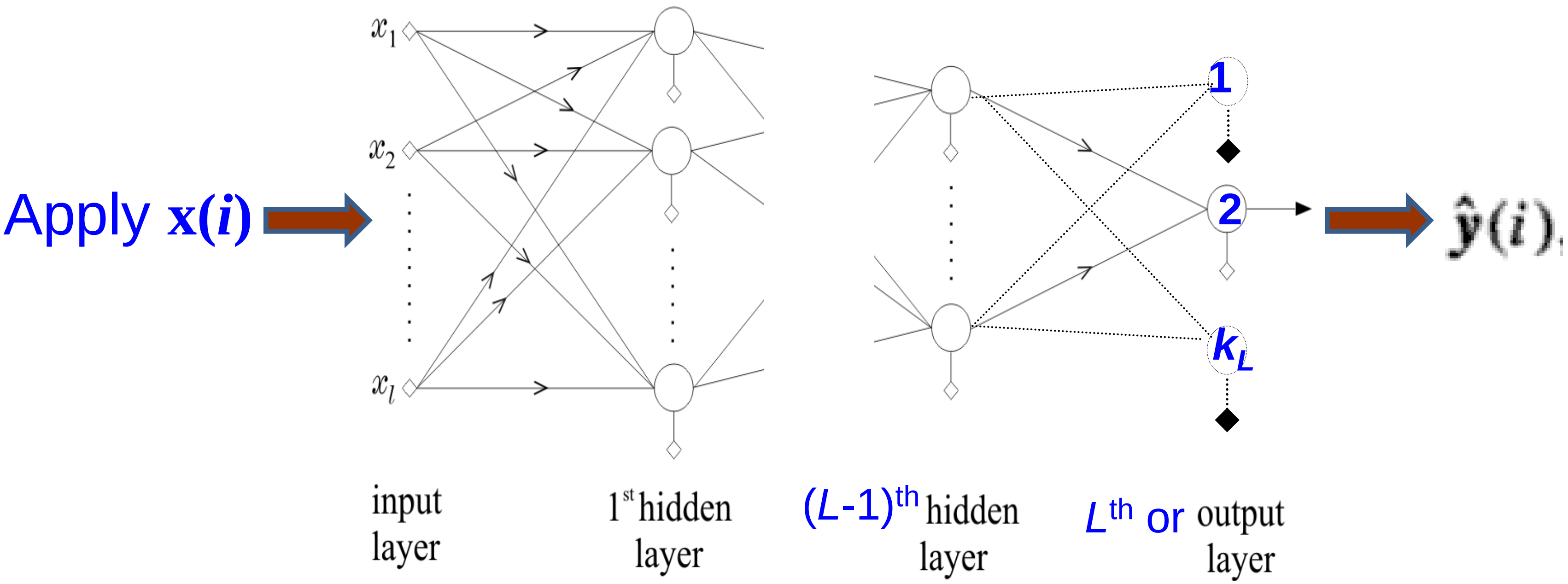
Define the Terms

- N training samples: $(\mathbf{x}(i), \mathbf{y}(i))$, for $i = 1, 2, 3, \ldots, N$
- Features of ith training sample: $\mathbf{x}(i) = \left[x_1(i), \cdots, x_{k_0}(i)\right]^T$
- Class of ith training sample: $\mathbf{y}(i) = \left[y_1(i), \cdots, y_{k_L}(i)\right]^T$



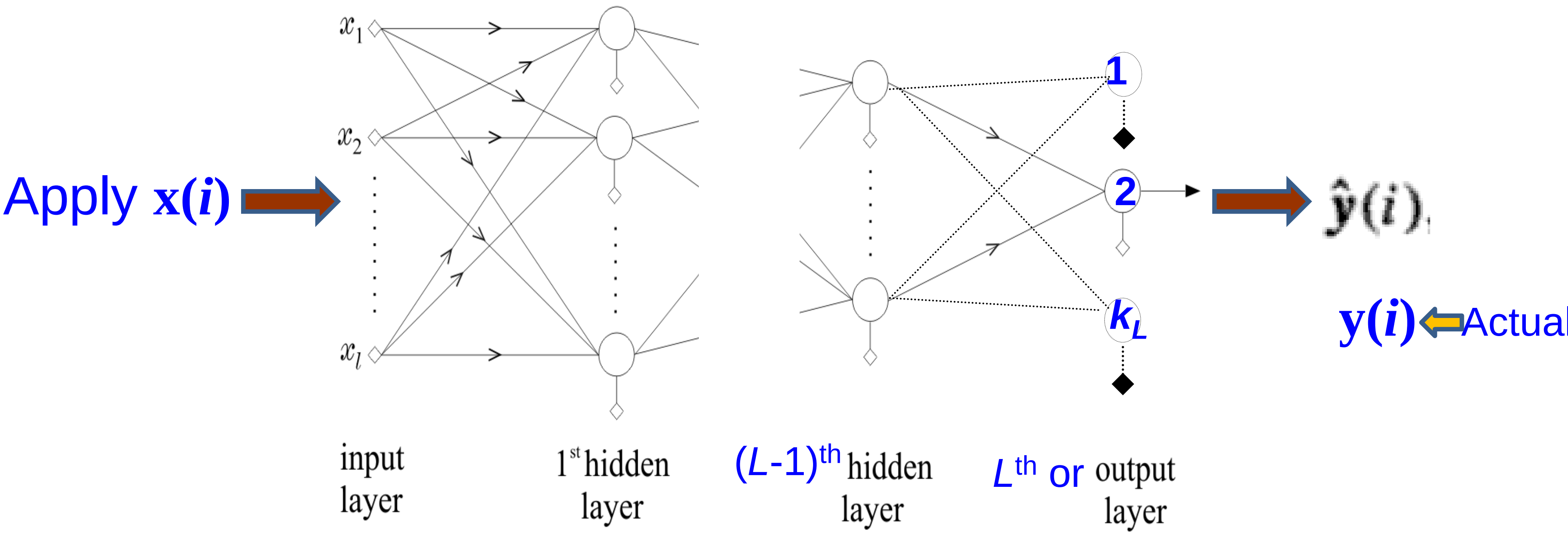
Define the Terms

- During training, apply i^{th} training vector $\mathbf{x}(i)$, and output is $\hat{\mathbf{y}}(i)$, instead of $\mathbf{y}(i)$)



Define the Terms

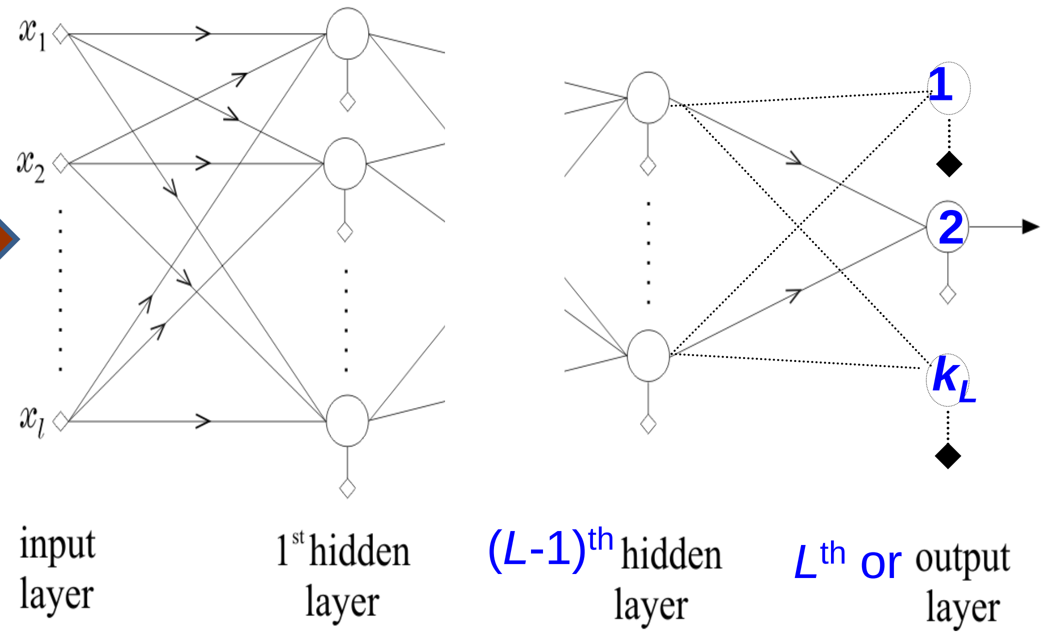
- During training, apply i^{th} training vector $\mathbf{x}(i)$, and output is $\hat{\mathbf{y}}(i)$, instead of $\mathbf{y}(i)$)



Define the Terms

- During training, apply i^{th} training vector $\mathbf{x}(i)$, and output is $\hat{\mathbf{y}}(i)$, instead of $\mathbf{y}(i)$)

Apply $\mathbf{x}(\boldsymbol{i})$



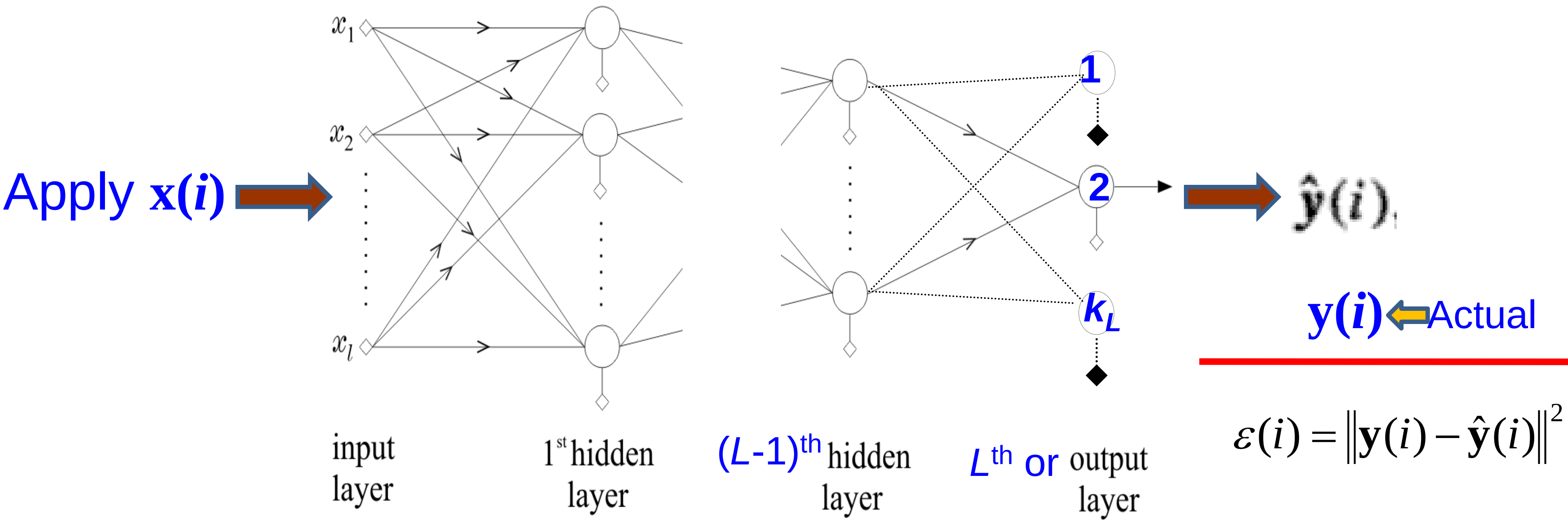
$\hat{\mathbf{y}}(i)$

$\mathbf{y}(\boldsymbol{i})$ Actual

Error = ?

Define the Terms

- During training, apply i^{th} training vector $\mathbf{x}(i)$, and output is $\hat{\mathbf{y}}(i)$, instead of $\mathbf{y}(i)$)



$$\varepsilon(i) = \left\| \mathbf{y}(i) - \hat{\mathbf{y}}(i) \right\|^2$$

Define the Terms

- During training, apply i^{th} training vector $\mathbf{x}(i)$, and output is $\hat{\mathbf{y}}(i)$, instead of $\mathbf{y}(i)$)

- Error for i^{th} vector:

$$\mathcal{E}(i) = \frac{1}{2}\sum_{m=1}^{k_L} e_m^2(i) \equiv \frac{1}{2}\sum_{m=1}^{k_L}(y_m(i) - \hat{y}_m(i))^2, \quad i = 1, 2, \ldots, N$$

- Total Error:

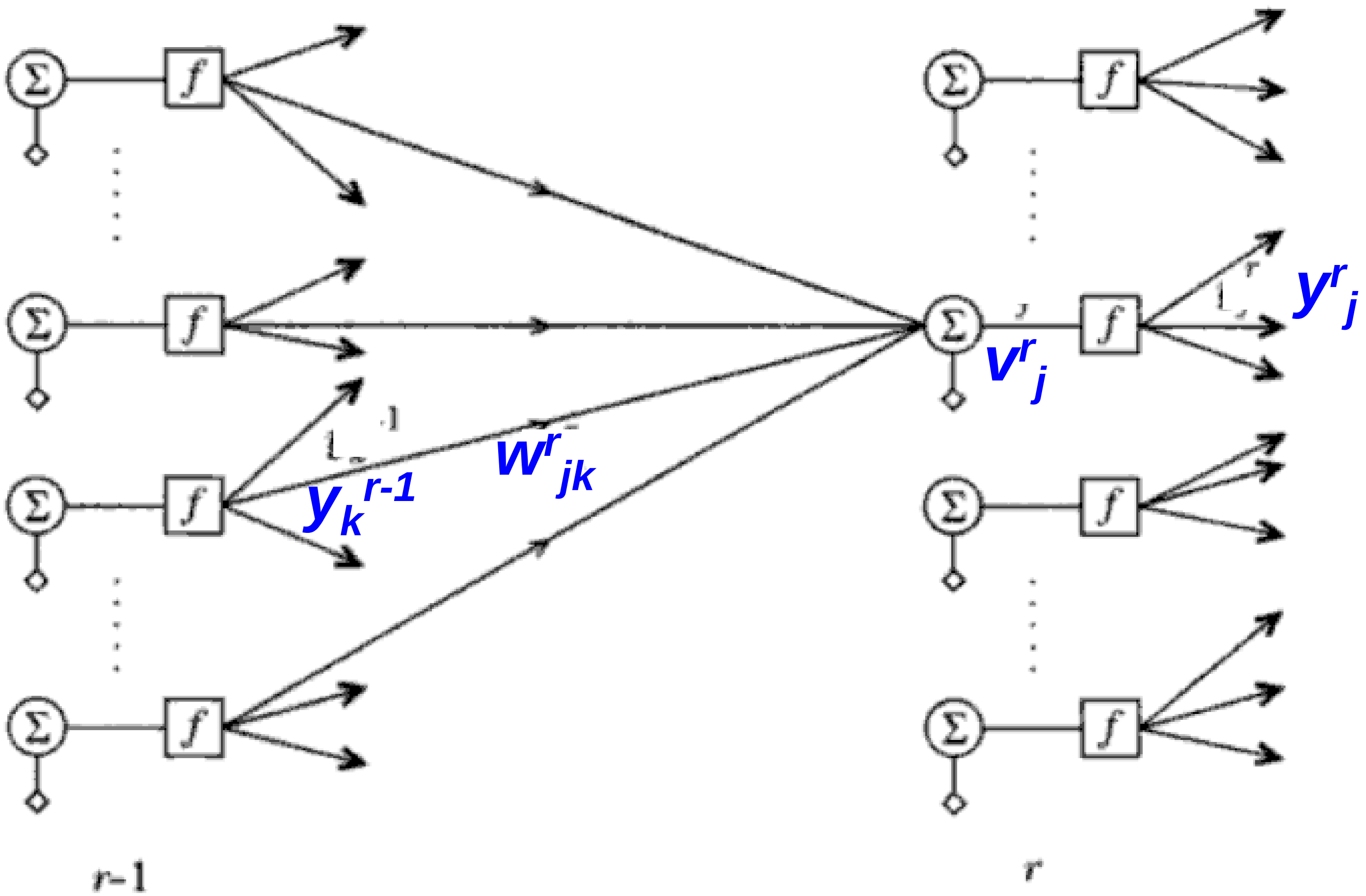
$$J = \sum_{i=1}^{N} \mathcal{E}(i)$$

Define the Terms

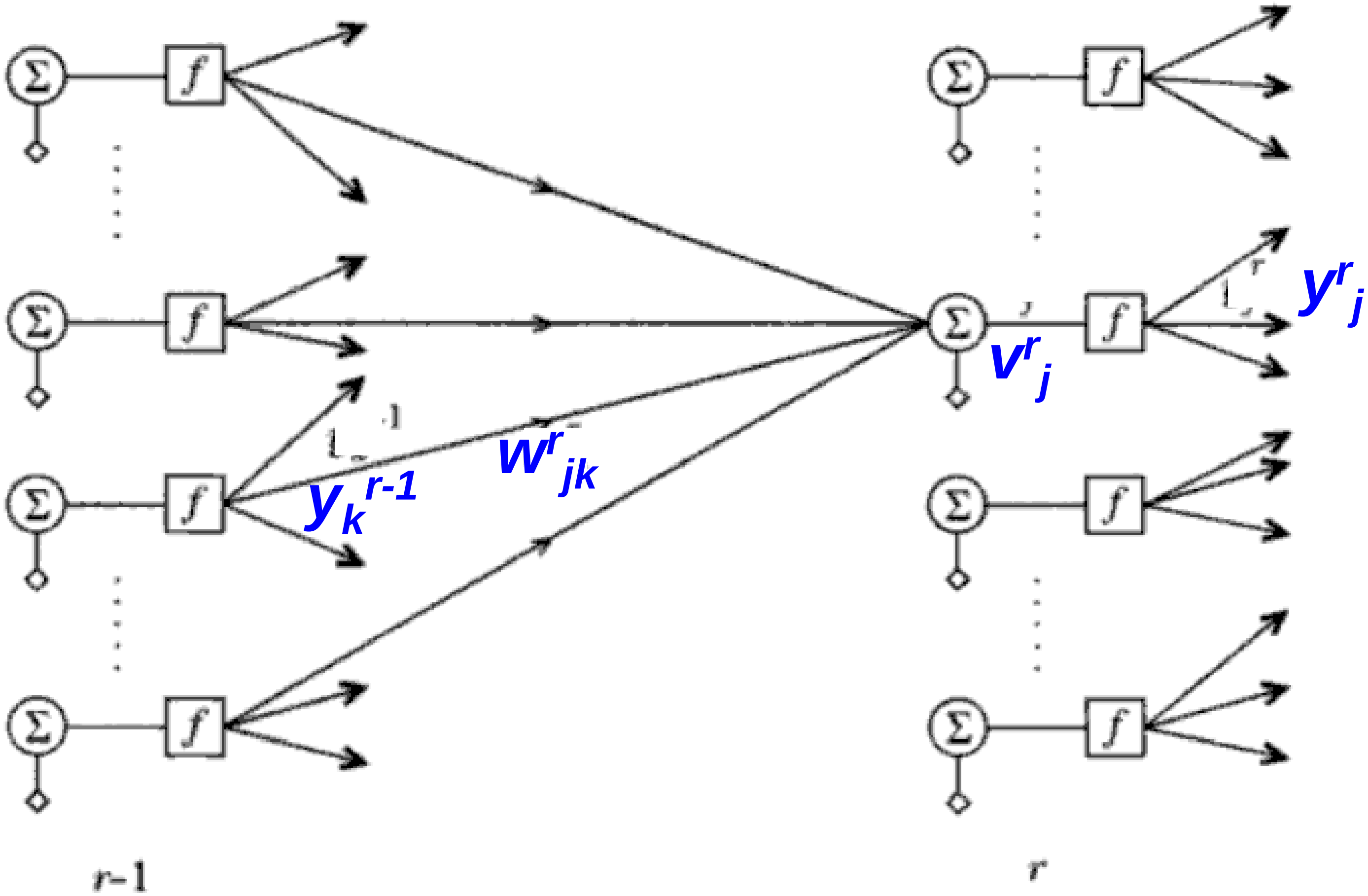
- We need to calculate $$\nabla \mathbf{w}_j^r = -\mu \frac{\partial J(i)}{\partial \mathbf{w}_j^r} = -\mu \sum_{i=1}^{N} \frac{\partial \varepsilon(i)}{\partial \mathbf{w}_j^r}$$

Define the Terms

- We need to calculate $$\nabla \mathbf{w}_j^r = -\mu \frac{\partial J(i)}{\partial \mathbf{w}_j^r} = -\mu \sum_{i=1}^{N} \frac{\partial \varepsilon(i)}{\partial \mathbf{w}_j^r}$$
- J depends on $\boldsymbol{w}_j^r$ and passes through v_j^r



Define the Terms



$$v_j^r(i) = \sum_{k=1}^{k_{r-1}} w_{jk}^r y_k^{r-1}(i) + w_{jo}^r \equiv \sum_{k=0}^{k_{r-1}} w_{jk}^r y_k^{r-1}(i)$$

Define the Terms

$$\frac{\partial \mathcal{E}(i)}{\partial \boldsymbol{w}_j^r} = \frac{\partial \mathcal{E}(i)}{\partial \upsilon_j^r(i)} \frac{\partial \upsilon_j^r(i)}{\partial \boldsymbol{w}_j^r}$$

Define the Terms

$$\frac{\partial \mathcal{E}(i)}{\partial \boldsymbol{w}_j^r} = \frac{\partial \mathcal{E}(i)}{\partial \upsilon_j^r(i)} \boxed{\frac{\partial \upsilon_j^r(i)}{\partial \boldsymbol{w}_j^r}}$$

Define the Terms

$$\frac{\partial \mathcal{E}(i)}{\partial \boldsymbol{w}_j^r} = \frac{\partial \mathcal{E}(i)}{\partial v_j^r(i)} \frac{\partial \boxed{v_j^r(i)}}{\partial \boldsymbol{w}_j^r}$$

Recall, $v_j^r(i) = \sum_{k=1}^{k_{r-1}} w_{jk}^r y_k^{r-1}(i) + w_{jo}^r \equiv \sum_{k=0}^{k_{r-1}} w_{jk}^r y_k^{r-1}(i)$

Define the Terms

$$\frac{\partial \mathcal{E}(i)}{\partial \boldsymbol{w}_j^r} = \frac{\partial \mathcal{E}(i)}{\partial \upsilon_j^r(i)} \frac{\partial \upsilon_j^r(i)}{\partial \boldsymbol{w}_j^r}$$

Recall, $\upsilon_j^r(i) = \sum_{k=1}^{k_{r-1}} w_{jk}^r y_k^{r-1}(i) + w_{jo}^r \equiv \sum_{k=0}^{k_{r-1}} w_{jk}^r y_k^{r-1}(i)$

Therefore, $$\frac{\partial}{\partial \boldsymbol{w}_j^r} \upsilon_j^r(i) \equiv \begin{bmatrix} \frac{\partial}{\partial w_{j0}^r} \upsilon_j^r(i) \\ \vdots \\ \frac{\partial}{\partial w_{jk_{r-1}}^r} \upsilon_j^r(i) \end{bmatrix}$$

Define the Terms

$$\frac{\partial \mathcal{E}(i)}{\partial \mathbf{w}_j^r} = \frac{\partial \mathcal{E}(i)}{\partial \upsilon_j^r(i)} \frac{\partial \upsilon_j^r(i)}{\partial \mathbf{w}_j^r}$$

Recall, $\upsilon_j^r(i) = \sum_{k=1}^{k_{r-1}} w_{jk}^r y_k^{r-1}(i) + w_{jo}^r \equiv \sum_{k=0}^{k_{r-1}} w_{jk}^r y_k^{r-1}(i)$

Therefore, $\dfrac{\partial}{\partial \mathbf{w}_j^r} \upsilon_j^r(i) \equiv \begin{bmatrix} \dfrac{\partial}{\partial w_{j0}^r} \upsilon_j^r(i) \\ \vdots \\ \dfrac{\partial}{\partial w_{jk_{r-1}}^r} \upsilon_j^r(i) \end{bmatrix} = \begin{bmatrix} +1 \\ y_1^{r-1}(i) \\ \vdots \\ y_{k_{r-1}}^{r-1}(i) \end{bmatrix}$

Define the Terms

$$\frac{\partial \mathcal{E}(i)}{\partial \boldsymbol{w}_j^r} = \frac{\partial \mathcal{E}(i)}{\partial v_j^r(i)} \frac{\partial v_j^r(i)}{\partial \boldsymbol{w}_j^r}$$

Recall, $v_j^r(i) = \sum_{k=1}^{k_{r-1}} w_{jk}^r y_k^{r-1}(i) + w_{jo}^r \equiv \sum_{k=0}^{k_{r-1}} w_{jk}^r y_k^{r-1}(i)$

Therefore, $\frac{\partial}{\partial \boldsymbol{w}_j^r} v_j^r(i) \equiv \begin{bmatrix} \frac{\partial}{\partial w_{j0}^r} v_j^r(i) \\ \vdots \\ \frac{\partial}{\partial w_{jk_{r-1}}^r} v_j^r(i) \end{bmatrix} = \boldsymbol{y}^{r-1}(i)$

Define the Terms

$$\frac{\partial \mathcal{E}(i)}{\partial \boldsymbol{w}_j^r} = \frac{\partial \mathcal{E}(i)}{\partial v_j^r(i)} \frac{\partial v_j^r(i)}{\partial \boldsymbol{w}_j^r}$$

$$\frac{\partial \mathcal{E}(i)}{\partial v_j^r(i)} \equiv \delta_j^r(i)$$

$$\frac{\partial}{\partial \boldsymbol{w}_j^r} v_j^r(i) \equiv \begin{bmatrix} \frac{\partial}{\partial w_{j0}^r} v_j^r(i) \\ \vdots \\ \frac{\partial}{\partial w_{jk_{r-1}}^r} v_j^r(i) \end{bmatrix} = \boldsymbol{y}^{r-1}(i)$$

Define the Terms

$$\frac{\partial \mathcal{E}(i)}{\partial \boldsymbol{w}_j^r} = \frac{\partial \mathcal{E}(i)}{\partial \upsilon_j^r(i)} \frac{\partial \upsilon_j^r(i)}{\partial \boldsymbol{w}_j^r}$$

$$\frac{\partial \mathcal{E}(i)}{\partial \upsilon_j^r(i)} \equiv \delta_j^r(i)$$

$$\frac{\partial}{\partial \boldsymbol{w}_j^r} \upsilon_j^r(i) \equiv \begin{bmatrix} \frac{\partial}{\partial w_{j0}^r} \upsilon_j^r(i) \\ \vdots \\ \frac{\partial}{\partial w_{jk_{r-1}}^r} \upsilon_j^r(i) \end{bmatrix} = \boldsymbol{y}^{r-1}(i)$$

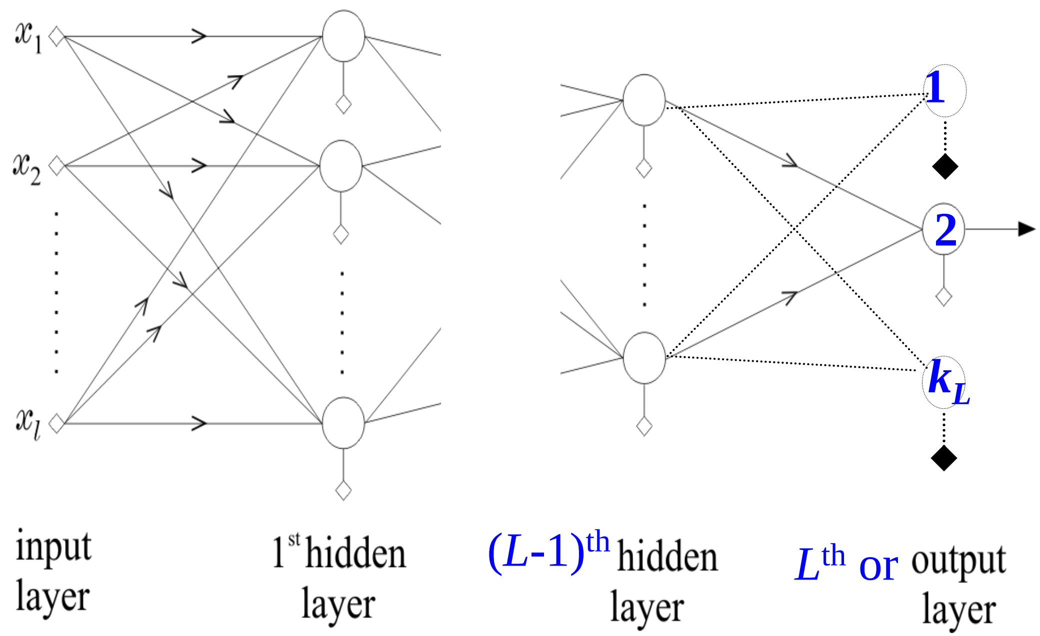
$$\Delta \mathbf{w}_j^r = -\mu \sum_{i=1}^{N} \frac{\partial \varepsilon(i)}{\partial \mathbf{w}_j^r(i)} \Rightarrow \boxed{\Delta \boldsymbol{w}_j^r = -\mu \sum_{i=1}^{N} \delta_j^r(i) \boldsymbol{y}^{r-1}(i)}$$

Define the Terms

$$\boxed{\Delta \boldsymbol{w}_j^r = -\mu \sum_{i=1}^{N} \delta_j^r(i)\, \boldsymbol{y}^{r-1}(i)}$$

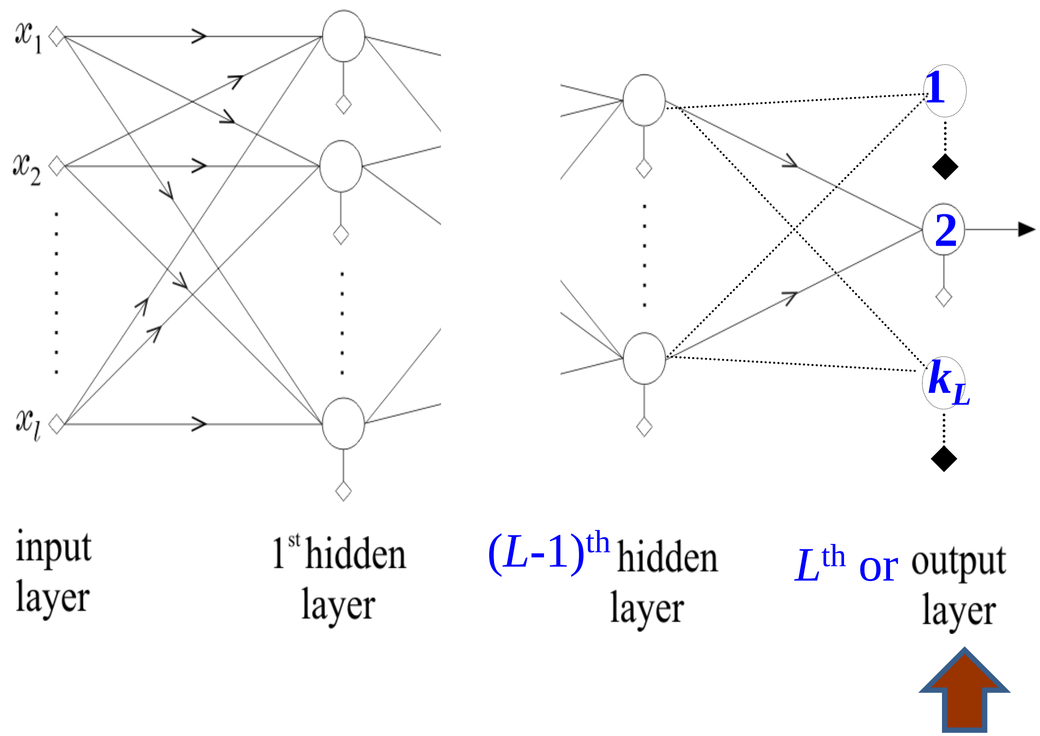
Define the Terms

$$\boxed{\Delta \boldsymbol{w}_j^r = -\mu \sum_{i=1}^{N} \delta_j^r(i)\, \boldsymbol{y}^{r-1}(i)}$$



Define the Terms

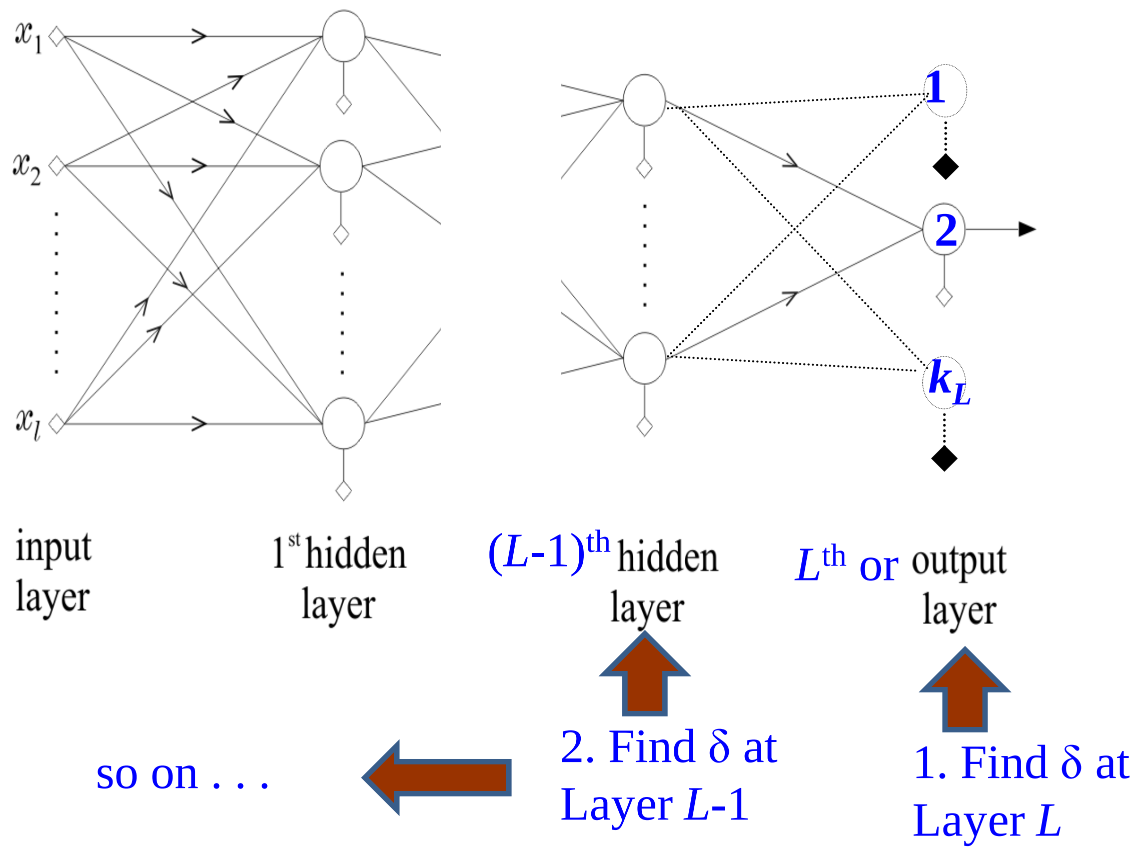
$$\boxed{\Delta \boldsymbol{w}_j^r = -\mu \sum_{i=1}^{N} \delta_j^r(i) \boldsymbol{y}^{r-1}(i)}$$



1. Find δ at Layer *L*

Define the Terms

$$\boxed{\Delta w_j^r = -\mu \sum_{i=1}^{N} \delta_j^r(i)\, y^{r-1}(i)}$$



Define the Terms

- Calculate $\delta_j^r(i) = \dfrac{\partial \varepsilon(i)}{\partial v_j^r(i)}$

Define the Terms

- Calculate $\delta_j^r(i) = \dfrac{\partial \varepsilon(i)}{\partial v_j^r(i)}$
- For $r = L$

$$\delta_j^L(i) = \frac{\partial \varepsilon(i)}{\partial v_j^L(i)}$$

Define the Terms

- Calculate $\delta_j^r(i) = \dfrac{\partial \varepsilon(i)}{\partial v_j^r(i)}$
- For $r = L$

$$\delta_j^L(i) = \frac{\partial \varepsilon(i)}{\partial v_j^L(i)}$$

$$\varepsilon(i) = \frac{1}{2}\sum_{m=1}^{k_L} e_m^2(i) \equiv \frac{1}{2}\sum_{m=1}^{k_L} \left(f(v_m^L(i)) - y_m(i) \right)^2$$

Define the Terms

- Calculate $$\delta_j^r(i) = \frac{\partial \varepsilon(i)}{\partial v_j^r(i)}$$
- For $r = L$

$$\delta_j^L(i) = \frac{\partial \varepsilon(i)}{\partial v_j^L(i)}$$

$$\varepsilon(i) = \frac{1}{2}\sum_{m=1}^{k_L} e_m^2(i) \equiv \frac{1}{2}\sum_{m=1}^{k_L} (f(v_m^L(i)) - y_m(i))^2$$

$$\delta_j^L(i) = \frac{1}{2} \times 2 \times (f(v_m^L(i)) - y_m(i)) \times f'(v_j^L(i))$$

Define the Terms

- Calculate $\delta_j^r(i) = \dfrac{\partial \varepsilon(i)}{\partial v_j^r(i)}$
- For $r = L$

$$\delta_j^L(i) = \frac{\partial \varepsilon(i)}{\partial v_j^L(i)}$$

$$\varepsilon(i) = \frac{1}{2}\sum_{m=1}^{k_L} e_m^2(i) \equiv \frac{1}{2}\sum_{m=1}^{k_L}(f(v_m^L(i)) - y_m(i))^2$$

$$\delta_j^L(i) = \frac{1}{2} \times 2 \times (f(v_m^L(i)) - y_m(i)) \times f'(v_j^L(i))$$

$$\delta_j^L(i) = e_j(i) f'(v_j^L(i))$$

Define the Terms

- Calculate $\delta_j^r(i) = \dfrac{\partial \varepsilon(i)}{\partial v_j^r(i)}$
- For $r = L$

$$\delta_j^L(i) = \frac{\partial \varepsilon(i)}{\partial v_j^L(i)}$$

$$\varepsilon(i) = \frac{1}{2}\sum_{m=1}^{k_L} e_m^2(i) \equiv \frac{1}{2}\sum_{m=1}^{k_L} (f(v_m^L(i)) - y_m(i))^2$$

$$\delta_j^L(i) = \frac{1}{2} \times 2 \times \boxed{(f(v_m^L(i)) - y_m(i))} \times f'(v_j^L(i))$$

$$\delta_j^L(i) = \boxed{e_j(i)}\, f'(v_j^L(i))$$

Define the Terms

- For $r < L$
- Calculate $\delta_j^{r-1}(i)$ from $\delta_j^{r}(i)$

Define the Terms

- For $r < L$
- Calculate $\delta_j^{r-1}(i)$ from $\delta_j^{r}(i)$

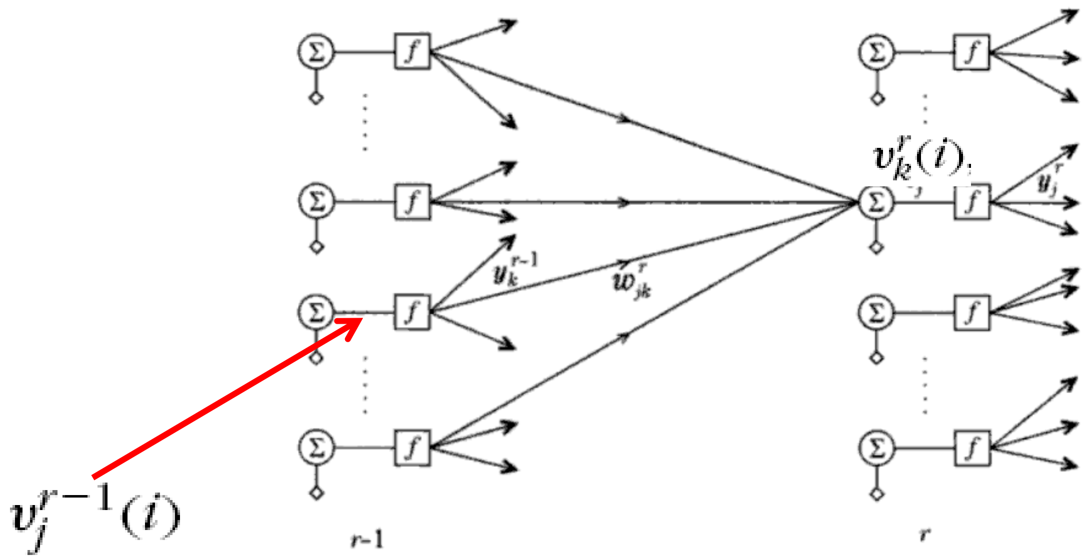
- We know,

$$\delta_j^{r-1}(i) = \frac{\partial \varepsilon(i)}{\partial v_j^{r-1}(i)}$$

Define the Terms

- We need to calculate, $$\delta_j^{r-1}(i) = \frac{\partial \varepsilon(i)}{\partial v_j^{r-1}(i)}$$

- However, $v_j^{r-1}(i)$ influences all $v_k^r(i)$, for $k = 1, 2, 3, \ldots, k_r$



Define the Terms

- We need to calculate,

$$\delta_j^{r-1}(i) = \frac{\partial \varepsilon(i)}{\partial v_j^{r-1}(i)}$$

- However, $v_j^{r-1}(i)$ influences all $v_k^r(i)$, for $k = 1, 2, 3, \ldots, k_r$

- Therefore,

$$\frac{\partial \varepsilon(i)}{\partial v_j^{r-1}(i)} = \sum_{k=1}^{k_r} \frac{\partial \varepsilon(i)}{\partial v_k^r(i)} \frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)}$$

Define the Terms

- For $r < L$
- Calculate

$$\delta_j^{r-1}(i) = \frac{\partial \varepsilon(i)}{\partial v_j^{r-1}(i)}$$

$$\frac{\partial \varepsilon(i)}{\partial v_j^{r-1}(i)} = \sum_{k=1}^{k_r} \frac{\partial \varepsilon(i)}{\partial v_k^{r}(i)} \frac{\partial v_k^{r}(i)}{\partial v_j^{r-1}(i)}$$

Define the Terms

- For $r < L$
- Calculate

$$\delta_j^{r-1}(i) = \frac{\partial \varepsilon(i)}{\partial v_j^{r-1}(i)}$$

$$\frac{\partial \varepsilon(i)}{\partial v_j^{r-1}(i)} = \sum_{k=1}^{k_r} \frac{\partial \varepsilon(i)}{\partial v_k^r(i)} \frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)}$$

$$\delta_j^{r-1}(i) = \sum_{k=1}^{k_r} \delta_k^r(i) \frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)}$$

Define the Terms

- For $r < L$
- Calculate

$$\delta_j^{r-1}(i) = \frac{\partial \varepsilon(i)}{\partial v_j^{r-1}(i)}$$

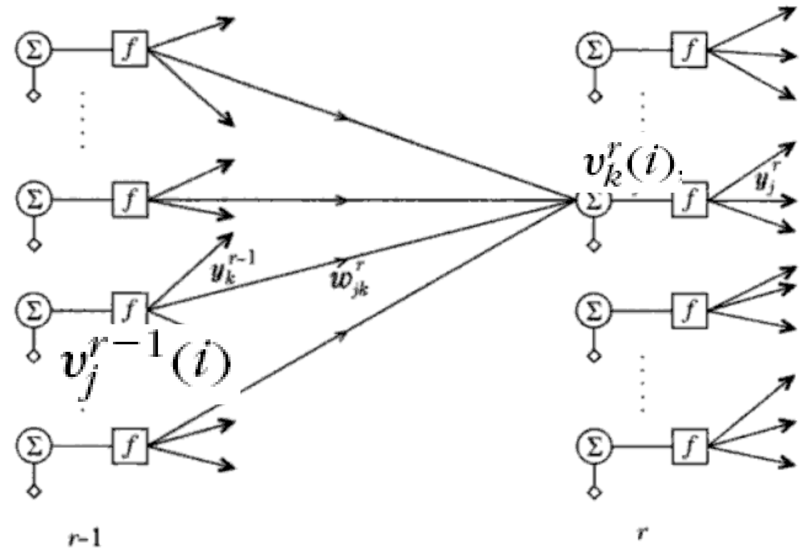
$$\frac{\partial \varepsilon(i)}{\partial v_j^{r-1}(i)} = \sum_{k=1}^{k_r} \frac{\partial \varepsilon(i)}{\partial v_k^{r}(i)} \frac{\partial v_k^{r}(i)}{\partial v_j^{r-1}(i)}$$

$$\delta_j^{r-1}(i) = \sum_{k=1}^{k_r} \delta_k^{r}(i) \frac{\partial v_k^{r}(i)}{\partial v_j^{r-1}(i)}$$

Define the Terms

$$\delta_j^{r-1}(i) = \sum_{k=1}^{k_r} \delta_k^r(i) \frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)}$$

$$\frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)} = \frac{\partial \left[\sum_{m=0}^{k_{r-1}} w_{km}^r y_m^{r-1}(i) \right]}{\partial v_j^{r-1}(i)}$$



where, $y_m^{r-1}(i) = f(v_m^{r-1}(i))$

Define the Terms

$$\delta_j^{r-1}(i) = \sum_{k=1}^{k_r} \delta_k^r(i) \frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)}$$

$$\frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)} = \frac{\partial \left[\sum_{m=0}^{k_{r-1}} w_{km}^r y_m^{r-1}(i) \right]}{\partial v_j^{r-1}(i)}$$ **where,** $y_m^{r-1}(i) = f(v_m^{r-1}(i))$

then,

$$\frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)} = w_{kj}^r f'(v_j^{r-1}(i))$$

Define the Terms

$$\delta_j^{r-1}(i) = \sum_{k=1}^{k_r} \delta_k^r(i) \frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)}$$

$$\frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)} = w_{kj}^r f'(v_j^{r-1}(i))$$

Define the Terms

$$\delta_j^{r-1}(i) = \sum_{k=1}^{k_r} \delta_k^r(i) \frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)}$$

$$\frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)} = w_{kj}^r f'(v_j^{r-1}(i))$$

Define the Terms

$$\delta_j^{r-1}(i) = \sum_{k=1}^{k_r} \delta_k^r(i) \frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)}$$

$$\frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)} = w_{kj}^r f'(v_j^{r-1}(i))$$

$$\delta_j^{r-1}(i) = \left[\sum_{k=1}^{k_r} \delta_k^r(i) w_{kj}^r\right] f'(v_j^{r-1}(i))$$

Define the Terms

$$\delta_j^{r-1}(i) = \left[\sum_{k=1}^{k_r} \delta_k^r(i) w_{kj}^r\right] f'(v_j^{r-1}(i))$$

$$\delta_j^{r-1}(i) = e_j^{r-1}(i) f'(v_j^{r-1}(i))$$

where, $$e_j^{r-1}(i) = \sum_{k=1}^{k_r} \delta_k^r(i) w_{kj}^r$$

Define the Terms

- Only remaining is the derivative of the logistic function:

$$f'(x) = \alpha f(x)(1 - f(x))$$

The Algorithm

- Initialization:
  - Start with small random weights
- Forward Computations: $v_j^r(i)$, $y_j^r(i) = f(v_j^r(i))$,
- Backward Computation: $\delta_j^L(i)$ and $\delta_j^{r-1}(i)$
- Update weight:

$$\boldsymbol{w}_j^r(\text{new}) = \boldsymbol{w}_j^r(\text{old}) + \Delta\boldsymbol{w}_j^r$$

$$\Delta\boldsymbol{w}_j^r = -\mu \sum_{i=1}^{N} \delta_j^r(i)\boldsymbol{y}^{r-1}(i)$$