Training of Neural Network (This Lecture And Next Week)

Notation (e chapter 7)

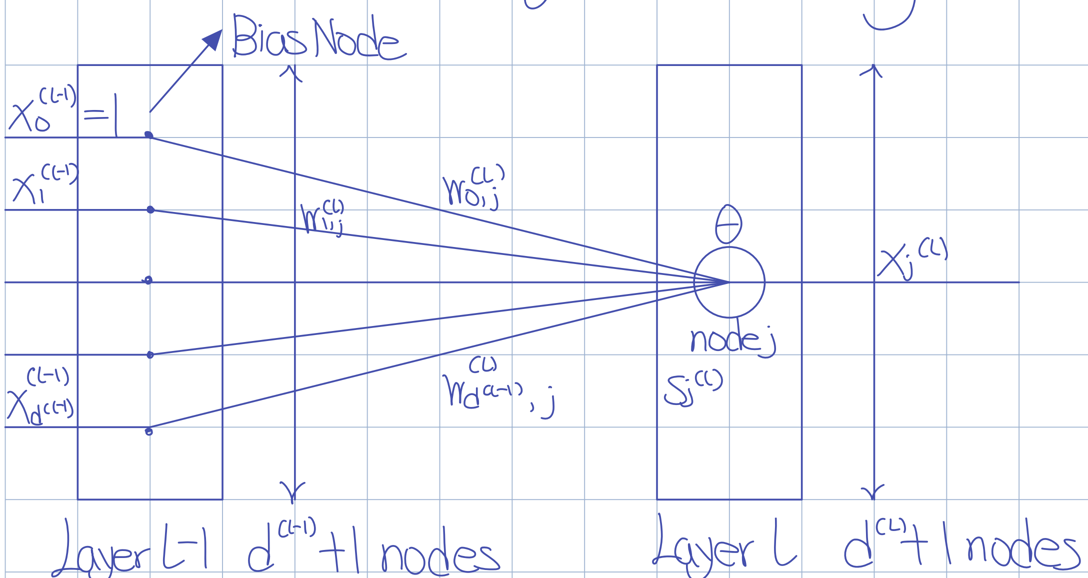
Input Layer: $l=0$
Hidden Layers: $1 \le l \le L-1$
Output Layer: $l=L$

L layers neural network

$w_{i,j}^{(l)} \equiv$ weight connecting node i in layer $(l-1)$ to node j in layer l

Not Power, But Superscript

$d^{(l)} \equiv$ # of nodes in layer l (not counting bias node)



$x_j^{(l)} \equiv$ output from node j in layer l

$s_j^{(l)} \equiv$ input into node j in layer l

$$s_j^{(l)} = w_{0,j}^{(l)} + \sum_{i=1}^{d^{(l-1)}} w_{i,j}^{(l)} x_i^{(l-1)} \qquad *$$

$$x_j^{(l)} = \theta(s_j^{(l)})$$

$\theta(s) \equiv$ Activation Function

$\theta(s) = \tanh(s)$

$= \mathrm{ReLU}(s)$

Vector Notation

$$\underline{s^{(l)}} = \begin{bmatrix} s_1^{(l)} \\ \vdots \\ s_{d^{(l)}}^{(l)} \end{bmatrix} \qquad \underline{\theta(s^{(l)})} = \begin{bmatrix} \theta(s_1^{(l)}) \\ \vdots \\ \theta(s_{d^{(l)}}^{(l)}) \end{bmatrix}$$

$$\underline{x^{(l)}} = \begin{bmatrix} 1 \\ \\ \theta(s^{(l)}) \end{bmatrix}$$

Weight Matrix

$$\underline{W^{(l)}} = \left\{ w_{i,j}^{(l)} \right\}_{\substack{0 \le i \le d^{(l-1)} \\ 1 \le j \le d^{(l)}}}$$

Bias Node From Layer $(l-1)$ to Nodes In Layer (l)

$$\underline{W^{(l)}} = \begin{bmatrix} w_{0,1}^{(l)} & w_{0,2}^{(l)} & \cdots & w_{0,d^{(l)}}^{(l)} \\ w_{1,1}^{(l)} & w_{1,2}^{(l)} & \cdots & w_{1,d^{(l)}}^{(l)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{d^{(l-1)},1}^{(l)} & w_{d^{(l-1)},2}^{(l)} & \cdots & w_{d^{(l-1)},d^{(l)}}^{(l)} \end{bmatrix}$$

jth column (incident to node j in Layer l

In Vector Form

$$s^{(l)} = \boxed{W^{(l)}}^T x^{(l-1)}$$

Given, Input Vector

$$x^{(0)} = [1 \quad x_1 \quad x_2 \quad \ldots \quad x_d]^T$$

$$x^{(0)} \xrightarrow{W^{(1)}} s^{(1)} \longrightarrow x^{(1)}$$

$$x^{(1)} \xrightarrow{W^{(2)}} s^{(2)} \longrightarrow x^{(2)}$$

$$x^{(L-1)} \xrightarrow{W^{(L)}} s^{(L)} \longrightarrow x^{(L)}$$ (Final Output)

Forward Propagation Algorithm

Input $x^{(0)} = [1 \quad x_1 \quad x_2 \quad \ldots \quad x_d]^T$

for $l = 1, 2, \ldots, L$, do

$$s^{(l)} = \boxed{W^{(l)}}^T x^{(l-1)}$$

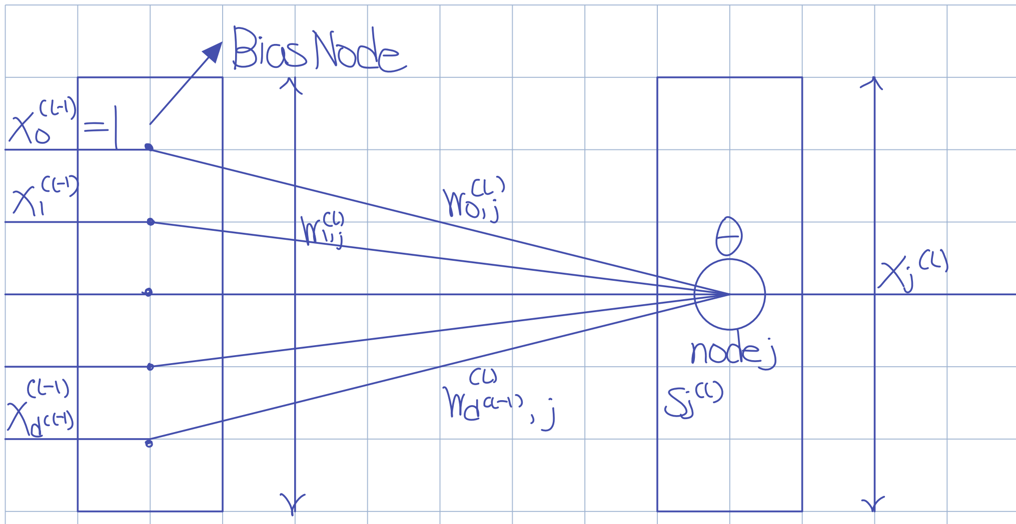
$$x^{(l)} = \boxed{\theta(s^{(l)})}$$

end

output $x^{(L)}$

Computational Complexity of Forward Propogation

of computation needed in layer l:



$d^{(l-1)}+1$ Computation for one node

$\underbrace{(d^{(l-1)}+1)\, d^{(l)}}$, Computation for one layer

Operations for computing $\underline{s^{(l)}}$

Activation Function $d^{(l)}$

$(d^{(l-1)}+1)\, d^{(l)} + d^{(l)}$ for computing $\underline{x^{(l)}}$

Total # of Computations

$$\sum_{l=1}^{L} (d^{(l-1)}+1)\, d^{(l)} + \sum_{l=1}^{L} d^{(l)}$$

# of edges = V   # of nodes = Q

≈ few millions

Linear

Summary So Far

Introduced Neural Network

Model Parameters

$$\Omega = \{W^{(1)}, W^{(2)}, \dots, W^{(l)}\}$$

Input, $\underline{x}^{(0)} = \begin{bmatrix} 1 & x_0 & x_1 & \dots & x_d \end{bmatrix}^T$

Output, $\underline{x}^{(l)}$

Loss Function

Regression, $d^{(l)} = 1$, $g(x^{(l)}, y) = (x^{(l)} - y)^2$

Classification,

Output $\underline{x}^{(l)} = \begin{bmatrix} \hat{P}_1 & \hat{P}_2 & \dots & \hat{P}_c \end{bmatrix}$, $d^{(l)} = c$

Loss $g(\underline{x}^{(l)}, y) = -\log \hat{P}_y$

Log Loss Function

Define $e_n(\Omega) \equiv$ Loss on training sample $(\underline{x}_n, y_n)$

$e_n(\Omega) = g(\underline{x}_n^{(l)}, y_n) = (\underline{x}_n^{(l)} - y_n)^2$ Regression

Average Lose

$$E_{in}(\Omega) = \frac{1}{N} \sum_{n=1}^{N} e_n(\Omega)$$

$$\Omega^* = \underset{\Omega}{\operatorname{argmin}} \, E_{in}(\Omega)$$

Gradient Descent Until Converge

Backpropagation