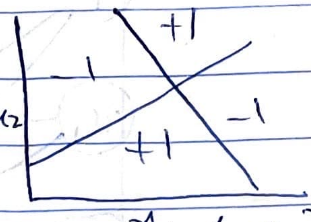
Lecture-18

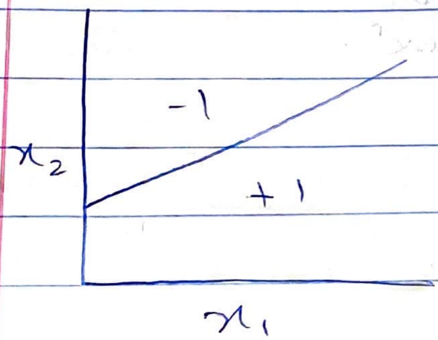
Decomposing XOR

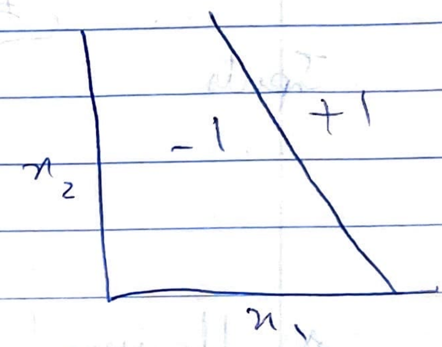
$f = XOR(h_1, h_2) \rightarrow$

$= h_1\bar{h}_2 + \bar{h}_1 h_2$



$h_1(x) = \text{sign}(w_1^T x)$ $\qquad$ $h_2(x) = \text{sign}(w_2^T x)$

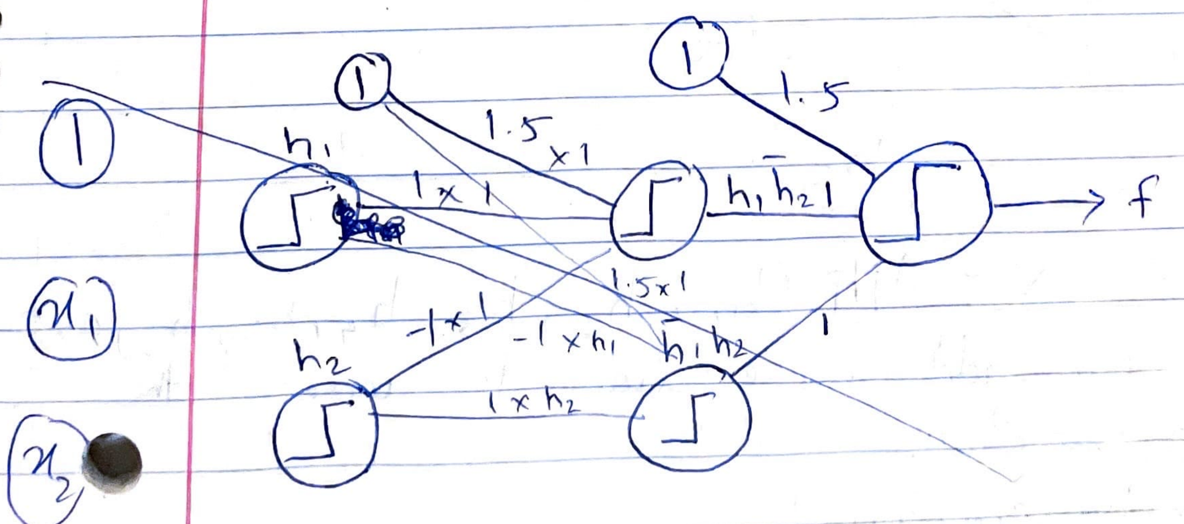




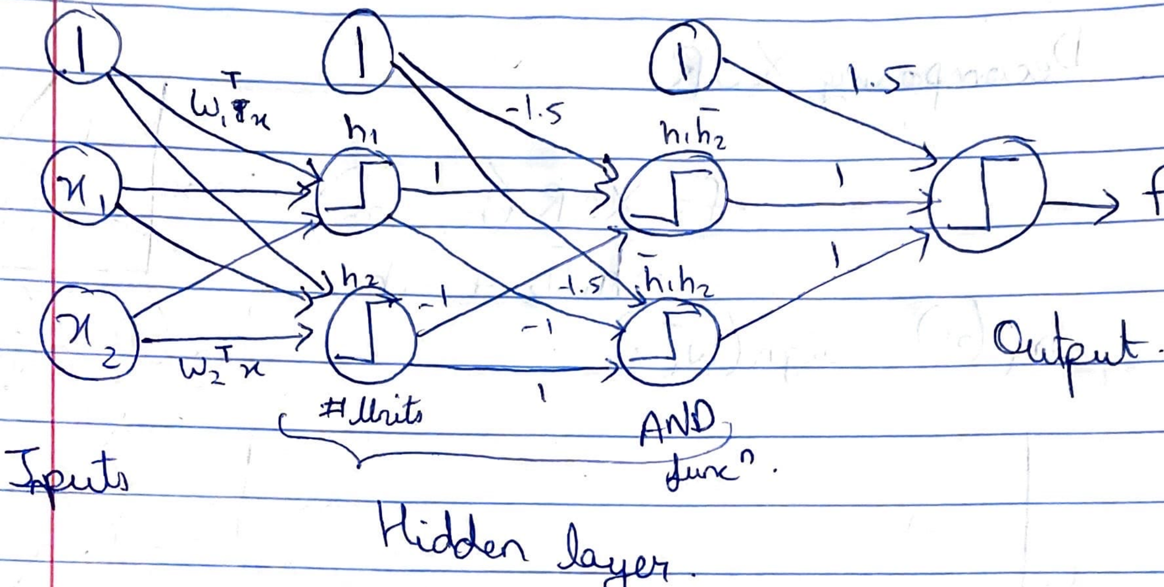
negation

$f = h_1\bar{h}_2 + \bar{h}_1 h_2$

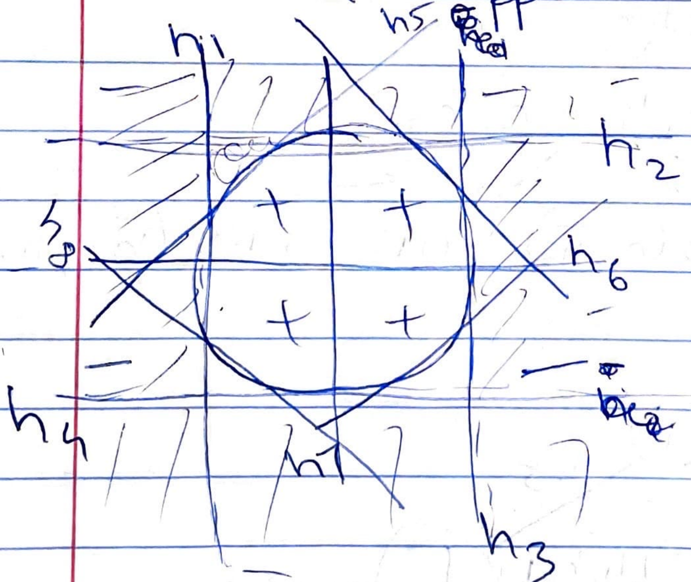
AND$(h_1, \bar{h}_2)$ OR AND$(\bar{h}_1, h_2)$.



The Multi-layer Perceptron (MLP)



☆ Universal Approximation



$$r = \text{AND}(h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8)$$

We can add more lines to increase the approximation.

like $h_9, h_{10}, h_{11}, h_{12} \ldots$

Approximation vs Generalization.

More nodes per hidden

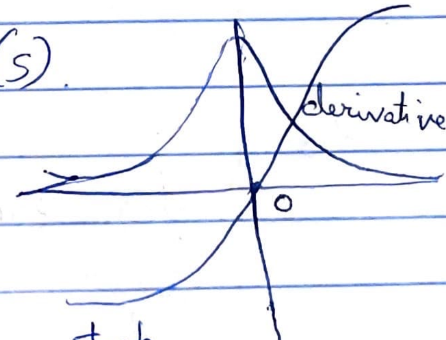
Approximation ↑

Generalization ↓

Fitting the Data / Minimizing E_{in}.

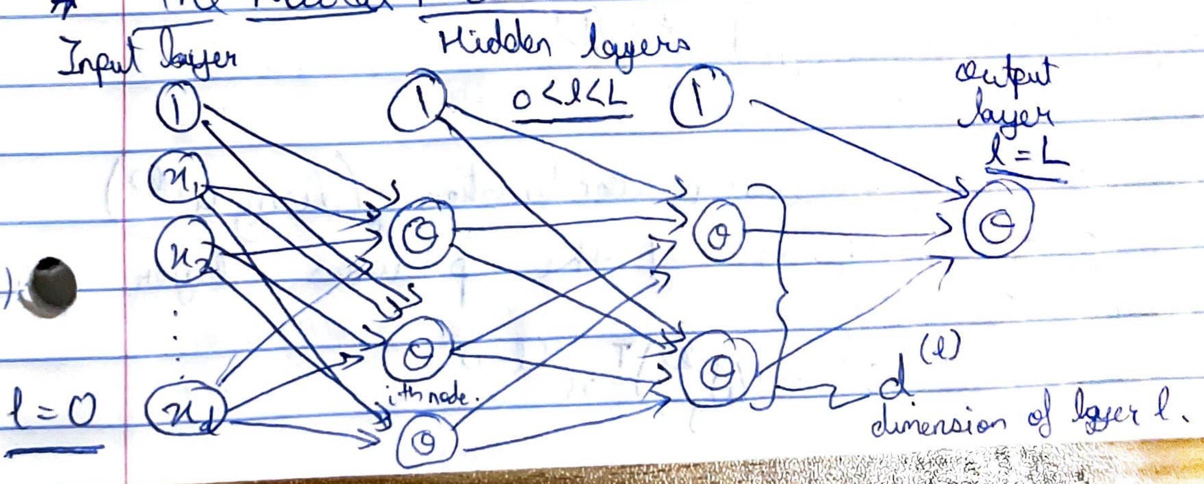
E_{in} is not smooth due to use of $\text{sign}(\cdot)$ func^n.
⇒ cannot use gradient descent.

Replace $\text{sign}(s)$ with $\tanh(s)$.

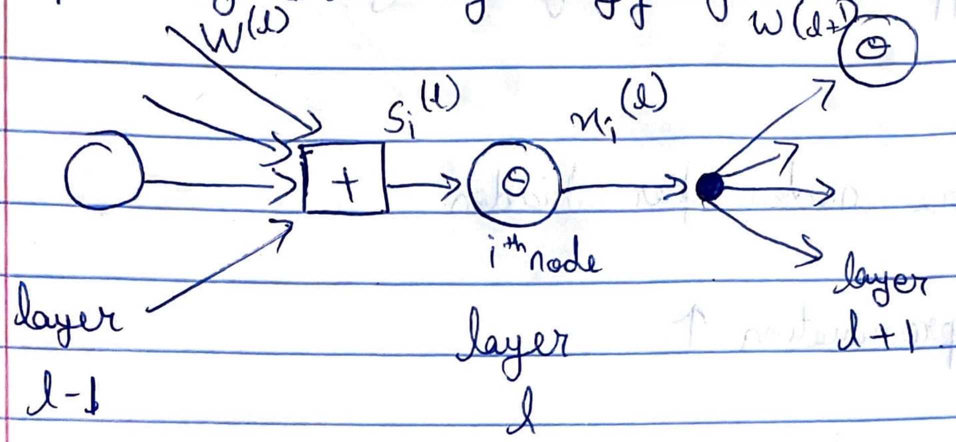


$$\text{sign}(s) \approx \tanh(s)$$

The Neural Network



Anatomy and Physiology of a Hidden Node.



layer l
has dimension $d^{(l)}$
ie $d^{(l)} + 1$ nodes.

$$W^{(l)} = \begin{bmatrix} w_1^{(l)} & w_2^{(l)} & \cdots & w_d^{(l)} \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

for each i^{th} node we need to compute linear signal $s_i^{(l)}$ to it and the apply tan function to it.

★ The linear signal $s^{(l)}$

$s^{(l)}$ is a linear combination (using $w^{(l)}$) of the outputs of the previous layer $x^{(l-1)}$

$$s^{(l)} = (w_j^{(l)})^T x^{(l-1)}$$