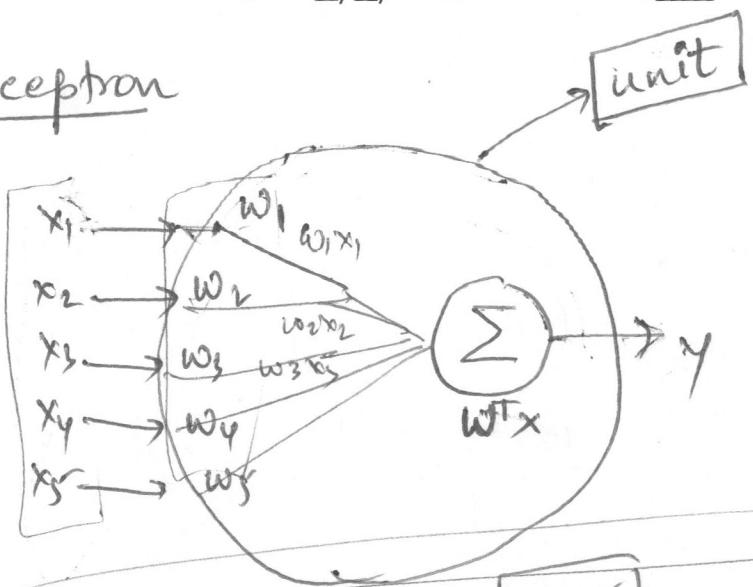
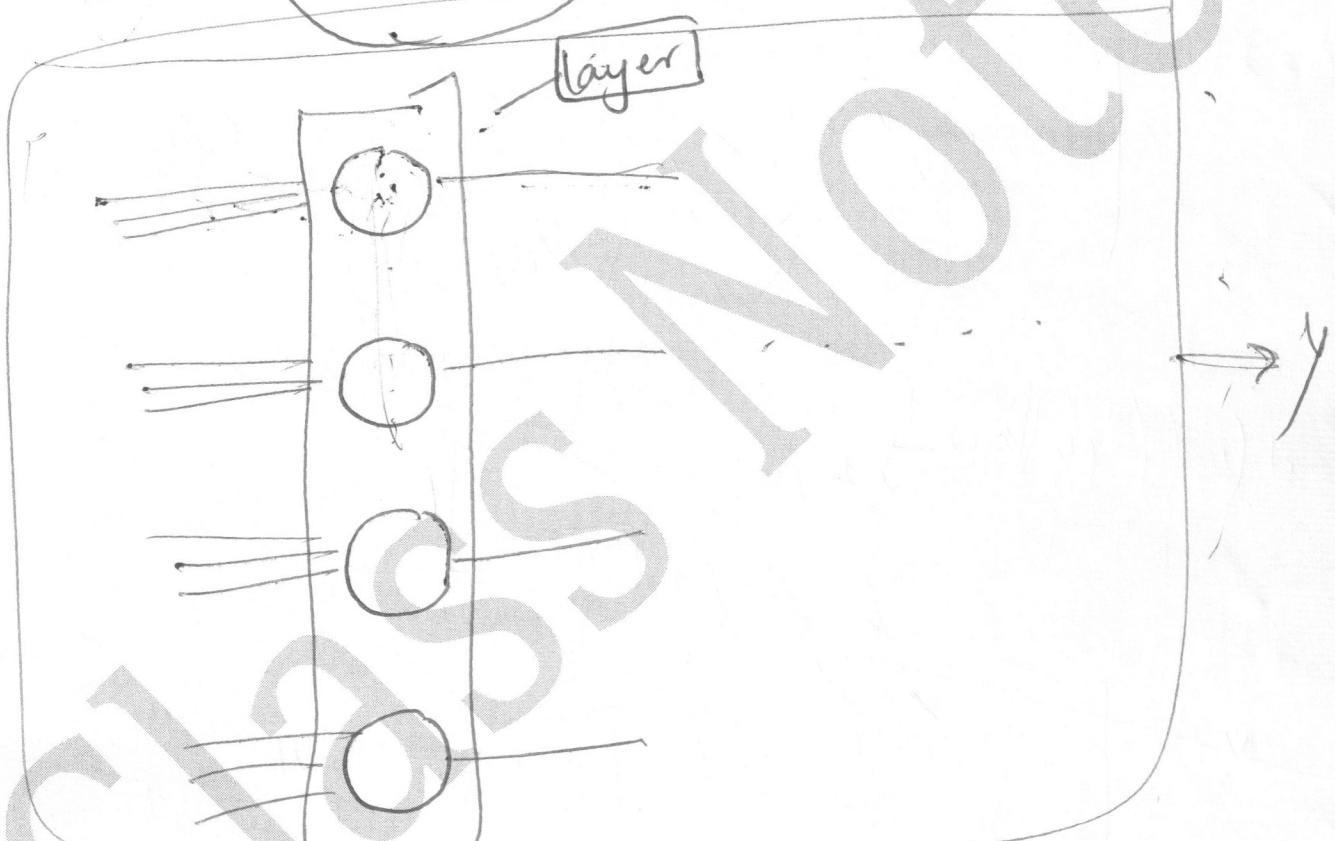


Perception

X



linear unit



Width of a layer → # units.

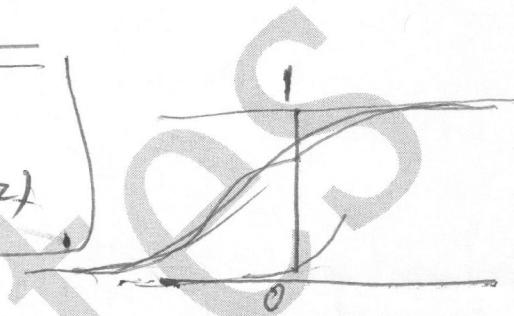
Instead of linear unit, we use a non-linear unit.



$$z = w^T x$$

Sigmoid fn.

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



tanh unit

$$f(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)}$$

ReLU unit.

$$f(z) = \max(0, z)$$



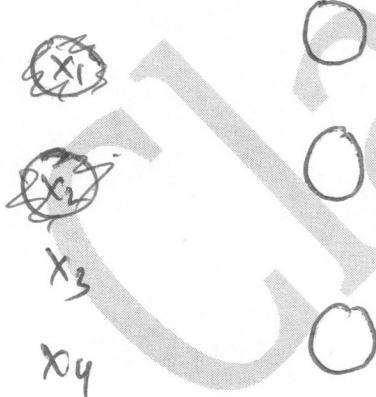
Multi-layered Perception (MLP) \rightarrow K-way classification
K=4

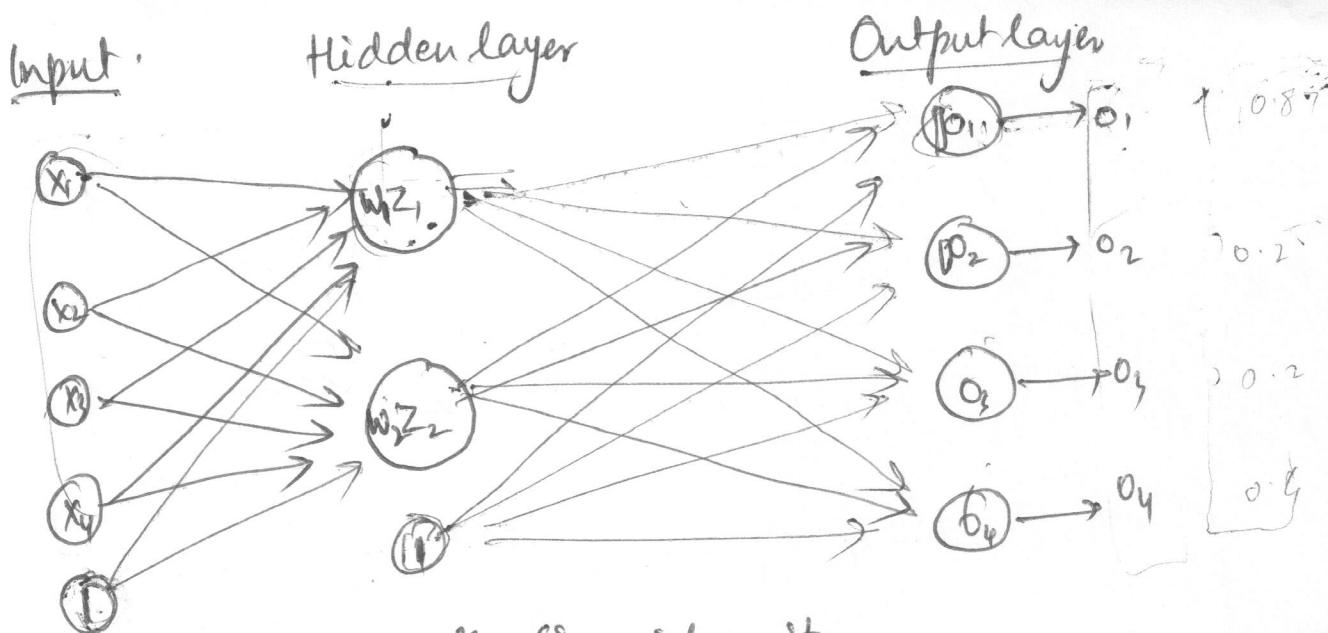
$$x \in \mathbb{R}^n$$

Input layer

Hidden layer

Output layer



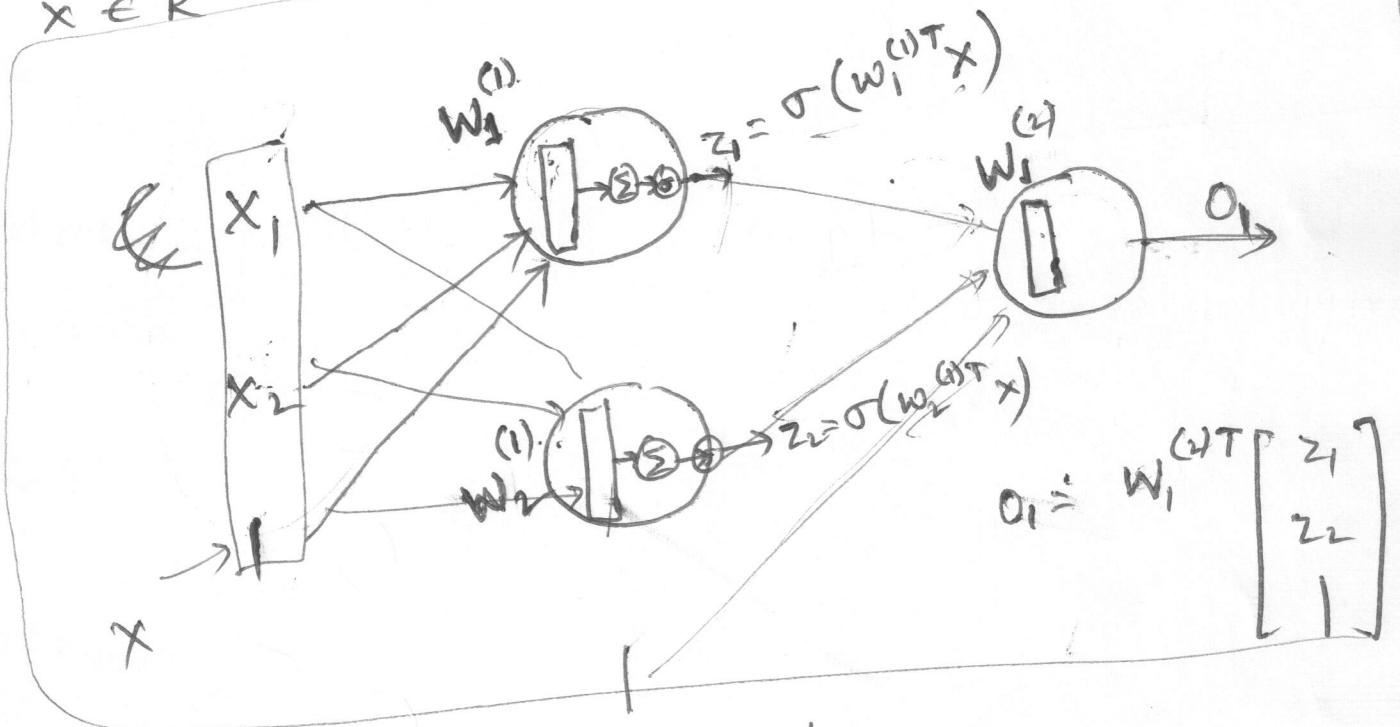


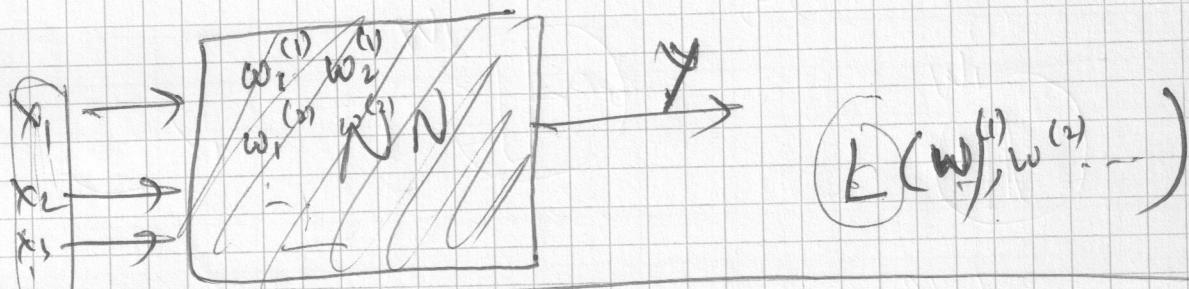
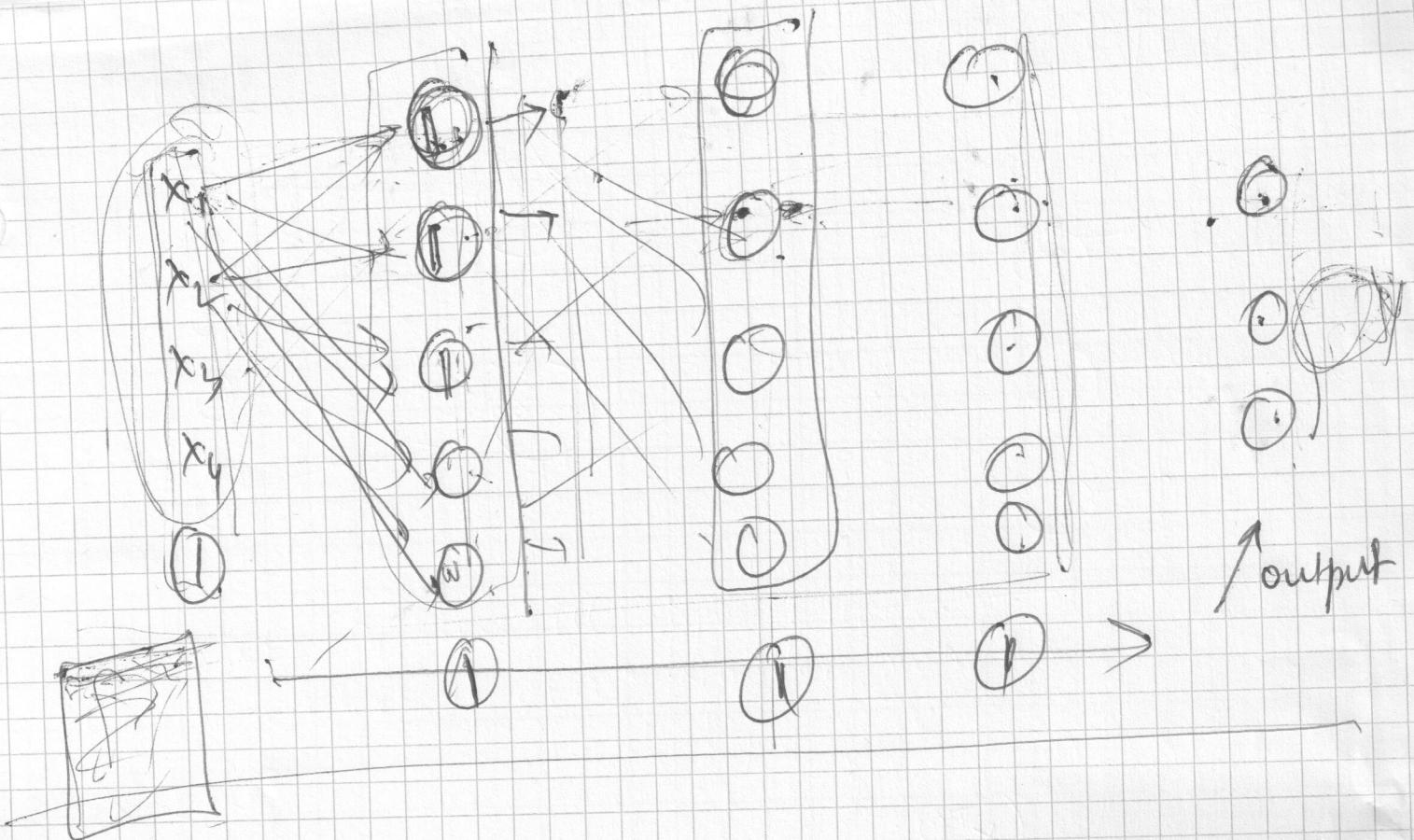
$z_1, z_2, o_1 \dots o_4$ are all sigmoid units.

Let us consider an even simpler network.

Input: Hidden layer: Output

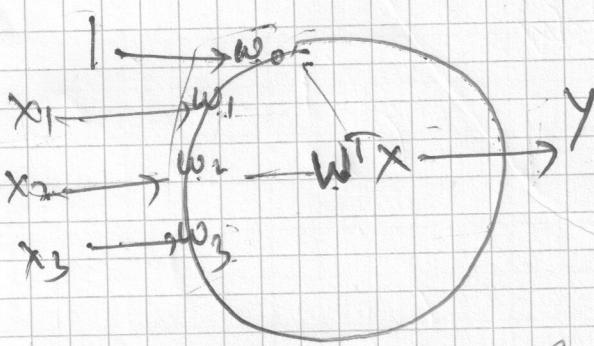
let $x \in \mathbb{R}^2$





Assuming we have only 1 training example

$$\mathbf{x} \rightarrow \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad y$$



Initialize the weight vector.

$$L(w) = \frac{1}{2} (y - w^T x)^2$$

$$\begin{aligned} \frac{\partial L(w)}{\partial w_0} &= -(y - w^T x) \\ \frac{\partial L(w)}{\partial w_1} &= -(y - w^T x) x_1 \\ &\vdots \end{aligned}$$

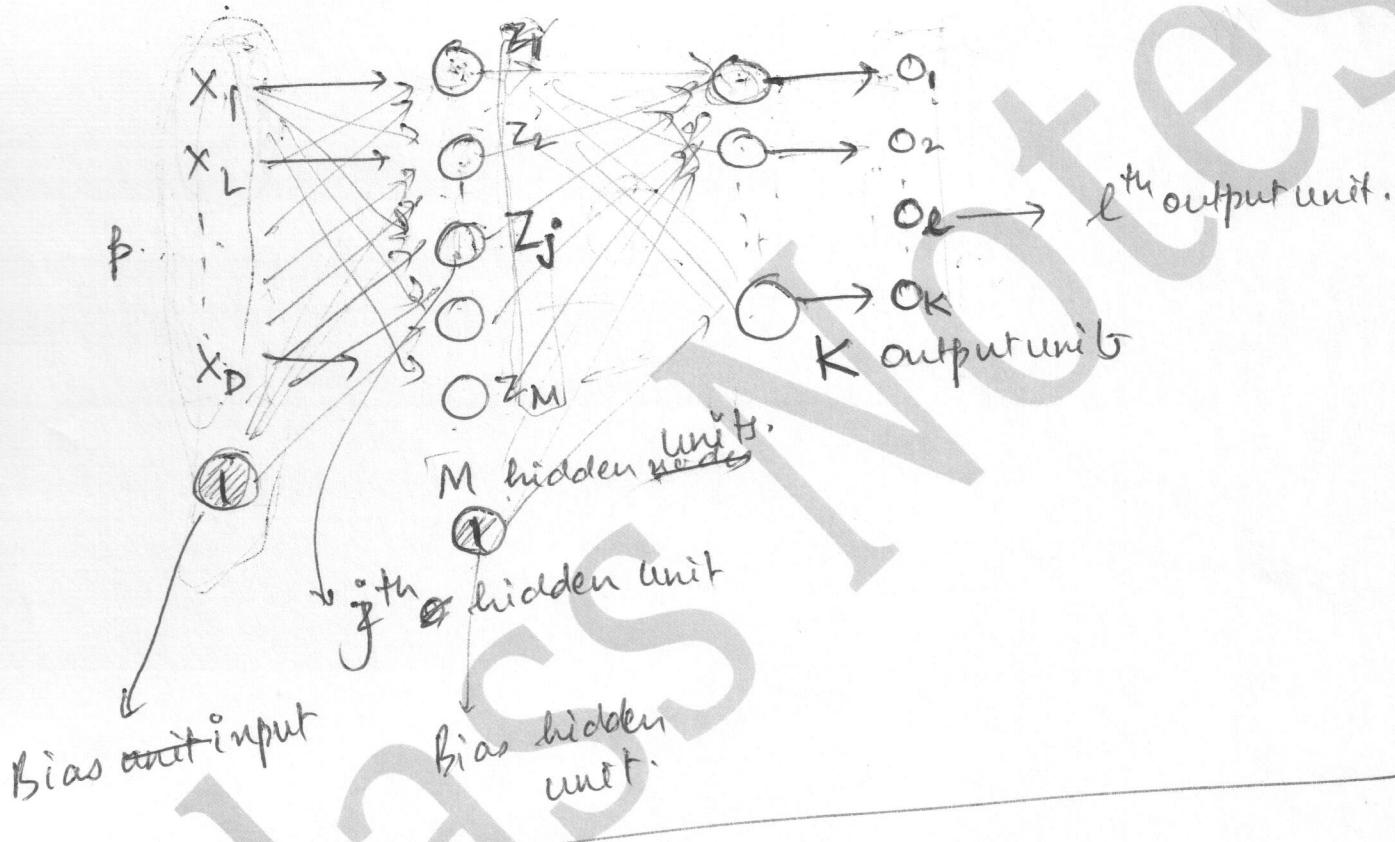
$$\begin{aligned} w^T x &= w_0 + w_1 x_1 \\ &\quad + w_2 x_2 \\ &\quad + w_3 x_3 \end{aligned}$$

new weight:

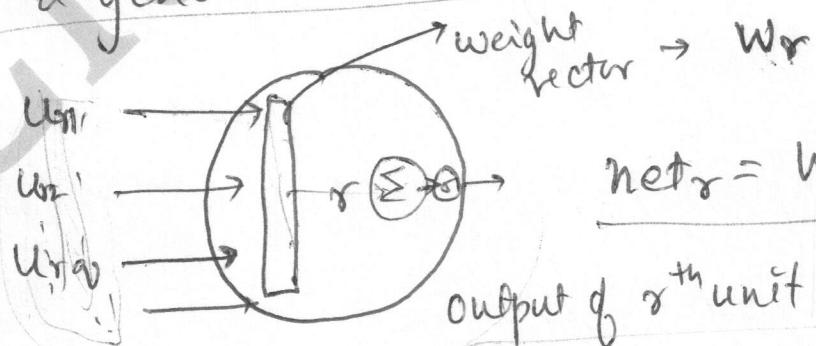
$$\begin{aligned} w_0 &= w_0 - \eta \frac{\partial L(w)}{\partial w_0} \\ &= w_0 - \eta (-y - w^T x) \\ &= w_0 - \eta ((y - w^T x)x_1) \end{aligned}$$

How to train a multi-layered perceptron? (MLP)
 For k-way classification

$$x \in \mathbb{R}^D \quad x \in \mathbb{R}^D \quad y \in \{1, \dots, K\}$$



Consider a general unit. r^{th} unit



$$net_r = W_r^T U_r = \sum_{i=1}^n w_{ri} u_{ri}$$

output of r^{th} unit will be $\sigma(net_r)$

$$\sigma(net_r) = \frac{1}{1 + \exp(-net_r)}$$

for the j^{th} hidden unit

$$z_j = \sigma(\text{net}_j) = \sigma(W_j^T X) \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

For the l^{th} output unit

$$o_l = \sigma(\text{net}_l) = \sigma(W_l^T z) \quad z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_M \end{bmatrix}$$

Predicted class = $\text{argmax}[o_1, o_2, \dots, o_K]$

For the j^{th} hidden unit: W_j is a $(p+1)$ -length vector

For the l^{th} output unit: W_l is a $(M+1)$ length vector

Training: Given N training examples

$$\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_N & y_N \end{bmatrix}$$

to find [learn] w_j 's and w_l 's.

$$j: 1 \dots M \quad l: 1 \dots K$$

$$J(w_j \text{ s } w_l \text{ s }) = \text{_____}$$

$$H(y=2)$$

$$\begin{bmatrix} o_1 \\ o_2 \end{bmatrix}$$

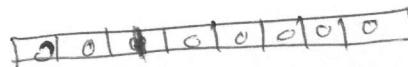
$$\begin{bmatrix} \vdots \\ o_l \\ \vdots \\ o_K \end{bmatrix}$$

Convert y_i into a vector

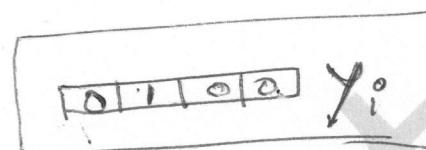
↑
label for
 i^{th} example.

One-Hot-Encoding
1 of K Encoding

K



example let $K=4$
 $y_i = 2$



loss for the i^{th} example:

$$J_i(\quad) = \frac{1}{2} \sum_{l=1}^K (y_{il} - o_{il})^2$$

→ output at l^{th}
output unit for
 i^{th} training example

Total loss $J = \sum_{i=1}^N \frac{1}{2} \sum_{l=1}^K (y_{il} - o_{il})^2$

We will assume that $N=1$, drop the subscript i .

$$J = \frac{1}{2} \sum_{l=1}^K (y_l - o_l)^2$$

for a general unit

We need

$$\frac{\partial J}{\partial w_{rl}}$$

$$w_{rl} \\ \downarrow \\ w_{rlg}$$

$$(w_{rlg} \leftarrow w_{rlg} - \eta \frac{\partial J}{\partial w_{rlg}})$$

Obs. 1

Not the
same z
as before

$$\begin{aligned} \frac{d}{dz} \sigma(z) &= \frac{d}{dz} \left(\frac{1}{1+e^{-z}} \right) \\ &= -\frac{1}{(1+e^{-z})^2} \frac{d}{dz} e^{-z} = \frac{1}{(1+e^{-z})^2} e^{-z} \\ &= \left(\frac{e^{-z}}{1+e^{-z}} \right) \frac{1}{(1+e^{-z})} \\ &= \left(1 - \frac{1}{1+e^{-z}} \right) \left(\frac{1}{1+e^{-z}} \right) \\ &= \underline{\sigma(z)(1-\sigma(z))} \end{aligned}$$

Obs. 2 For any r^{th} unit.

$$\frac{\partial J}{\partial w_{rq}} = \frac{\partial J}{\partial \text{net}_r} * \frac{\partial \text{net}_r}{\partial w_{rq}}$$

$$\text{net}_r = \sum_q w_{rq} u_{rq} = w_{r1} u_1 + w_{r2} u_{r2} + \dots + w_{rq} u_{rq} + \dots$$

$$\frac{\partial \text{net}_r}{\partial w_{rq}} = u_{rq}$$

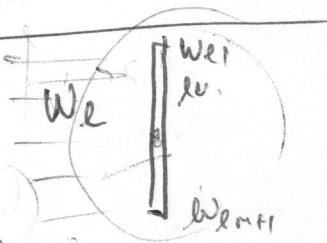
$$\frac{\partial J}{\partial w_{rq}} = u_{rq} \frac{\partial J}{\partial \text{net}_r}$$

Consider the l^{th} output unit.

We want

$$\frac{\partial J}{\partial w_{ij}} = z_{ej} \frac{\partial J}{\partial \text{net}_l}$$

* $\sigma_l = \underline{\sigma(\text{net}_l)}$



$$\frac{\partial J}{\partial \text{net}_e} = \frac{\partial J}{\partial o_e} \frac{\partial o_e}{\partial \text{net}_e} \rightarrow o_e(1-o_e)$$

$$= \left(\frac{\partial J}{\partial o_e} \right) o_e(1-o_e)$$

$$\frac{\partial J}{\partial o_e} = \frac{\partial}{\partial o_e} \left[\frac{1}{2} \sum_{e=1}^k (y_e - o_e)^2 \right]$$

$$= \frac{1}{2} \frac{\partial}{\partial o_e} (y_e - o_e)^2$$

$$= \frac{1}{2} * 2(y_e - o_e)(-1)$$

$$\Rightarrow -(y_e - o_e)$$

$$\boxed{\frac{\partial J}{\partial w_{ej}} = -z_{ej} o_e(1-o_e)(y_e - o_e)}$$