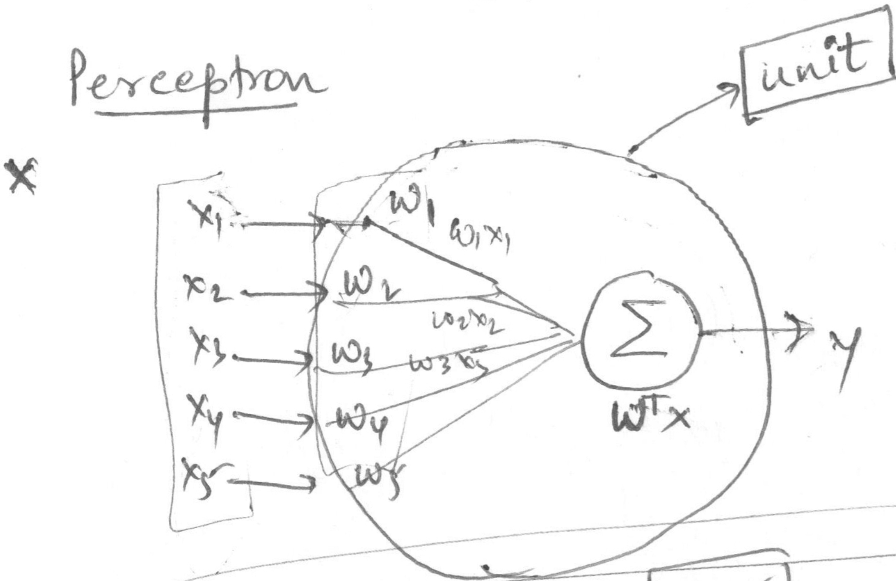
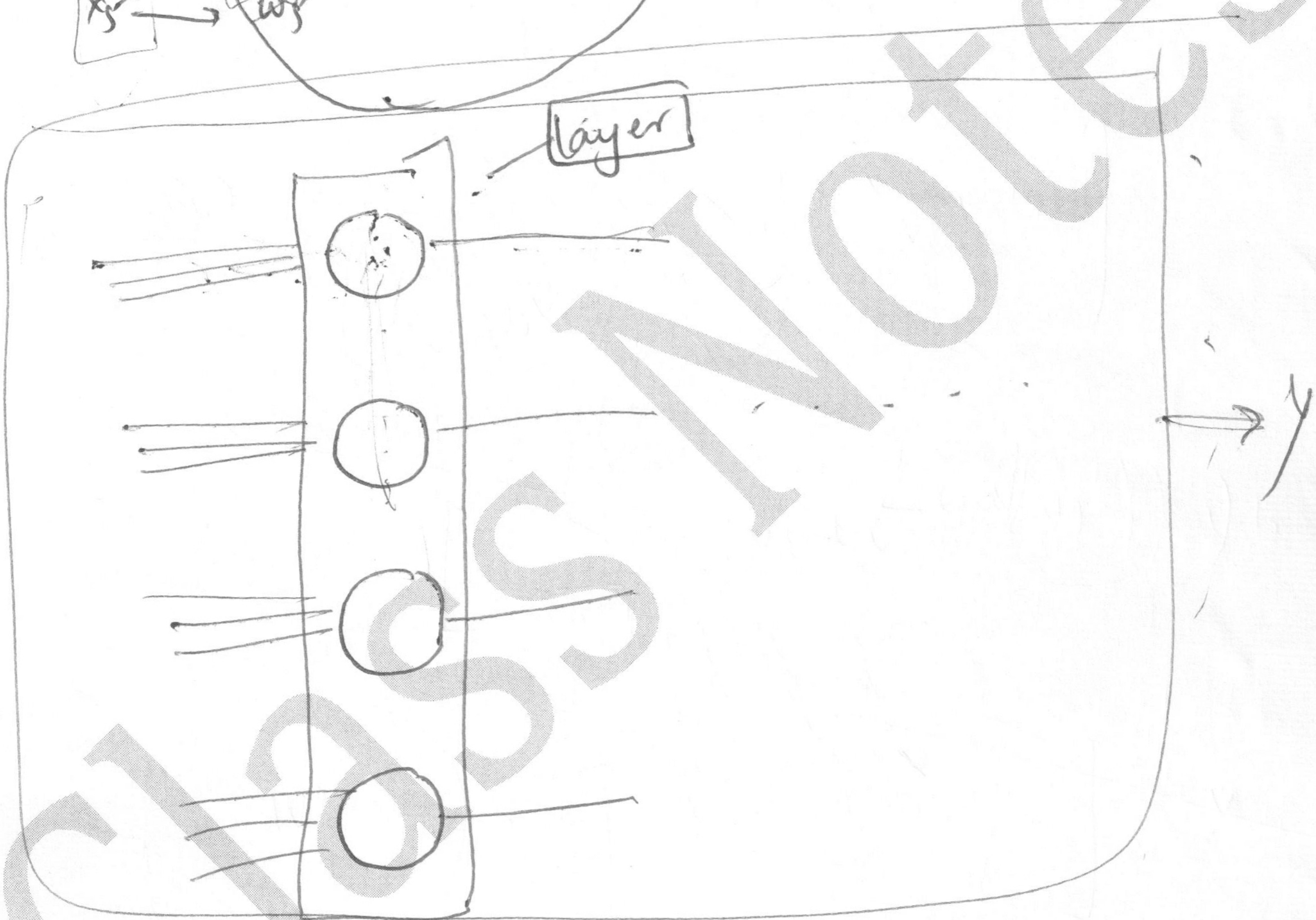
Perceptron

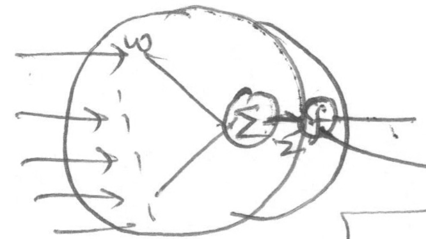


linear unit.



Width of a layer → # units.

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

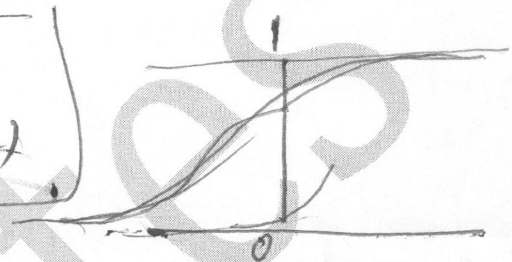


$z = w^T x$

activation fn.

Sigmoid fn.

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

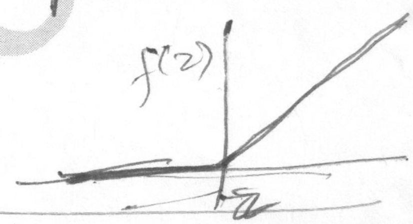


tanh unit

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

RELU unit.

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

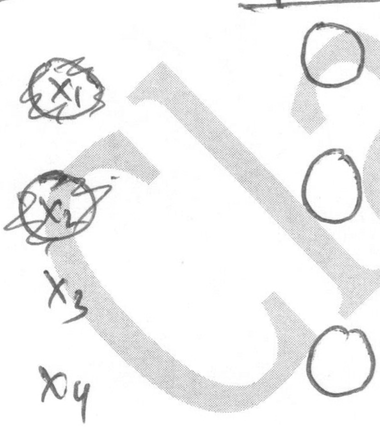


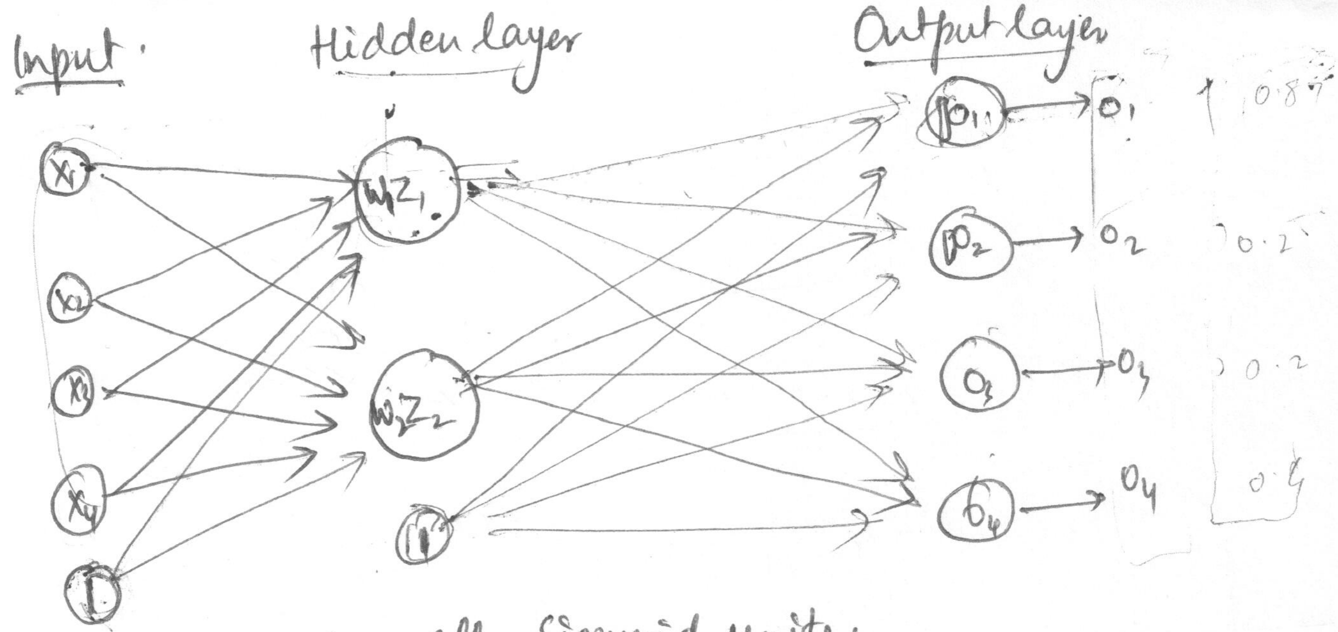
Multi-layered Perceptron (MLP) → K-way classification

$K = 4$

$x \in R^4$

Input layer Hidden layer Outputlayer



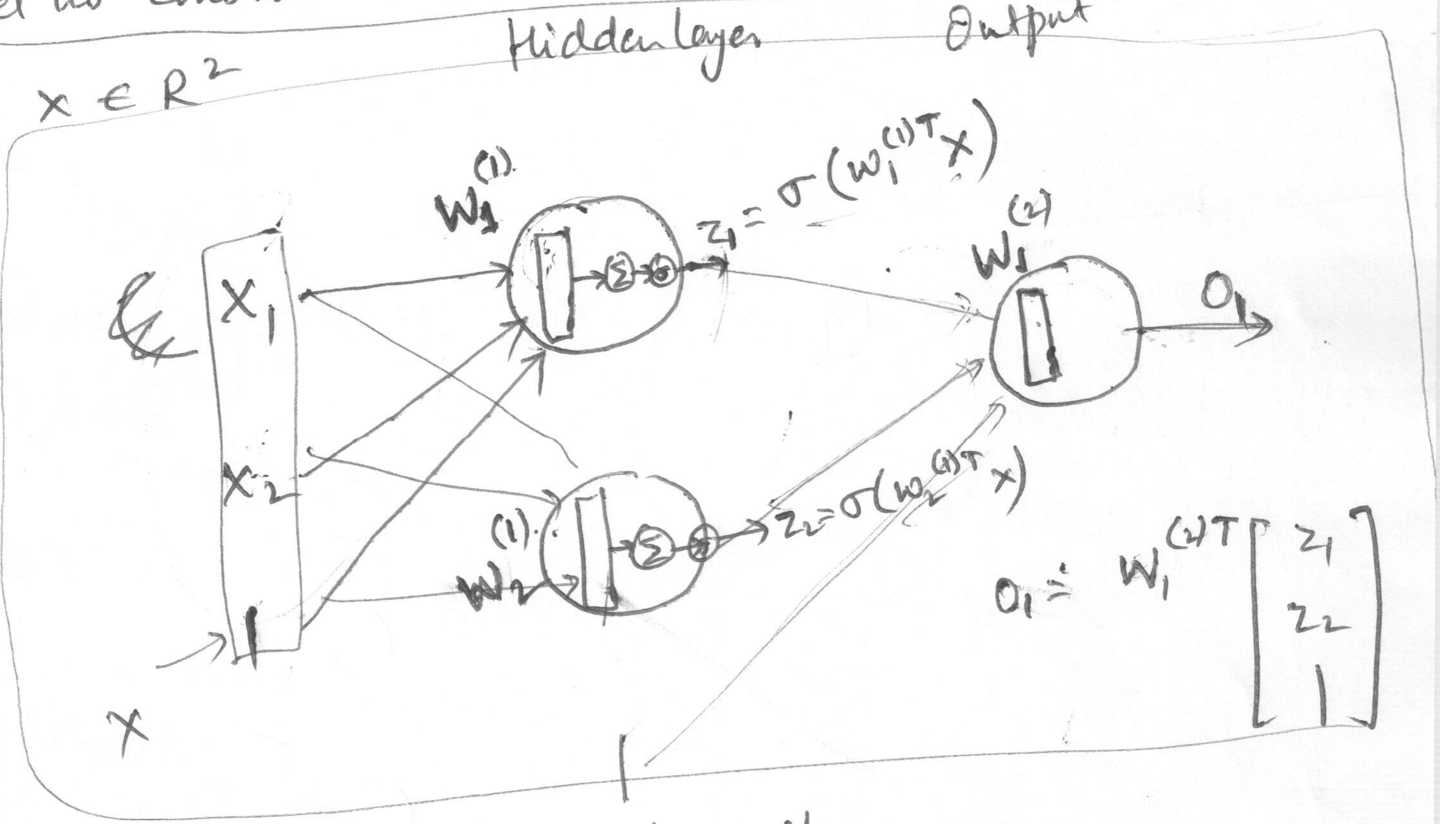


$z_1, z_2, o_1 \ldots o_4$ are all sigmoid units.

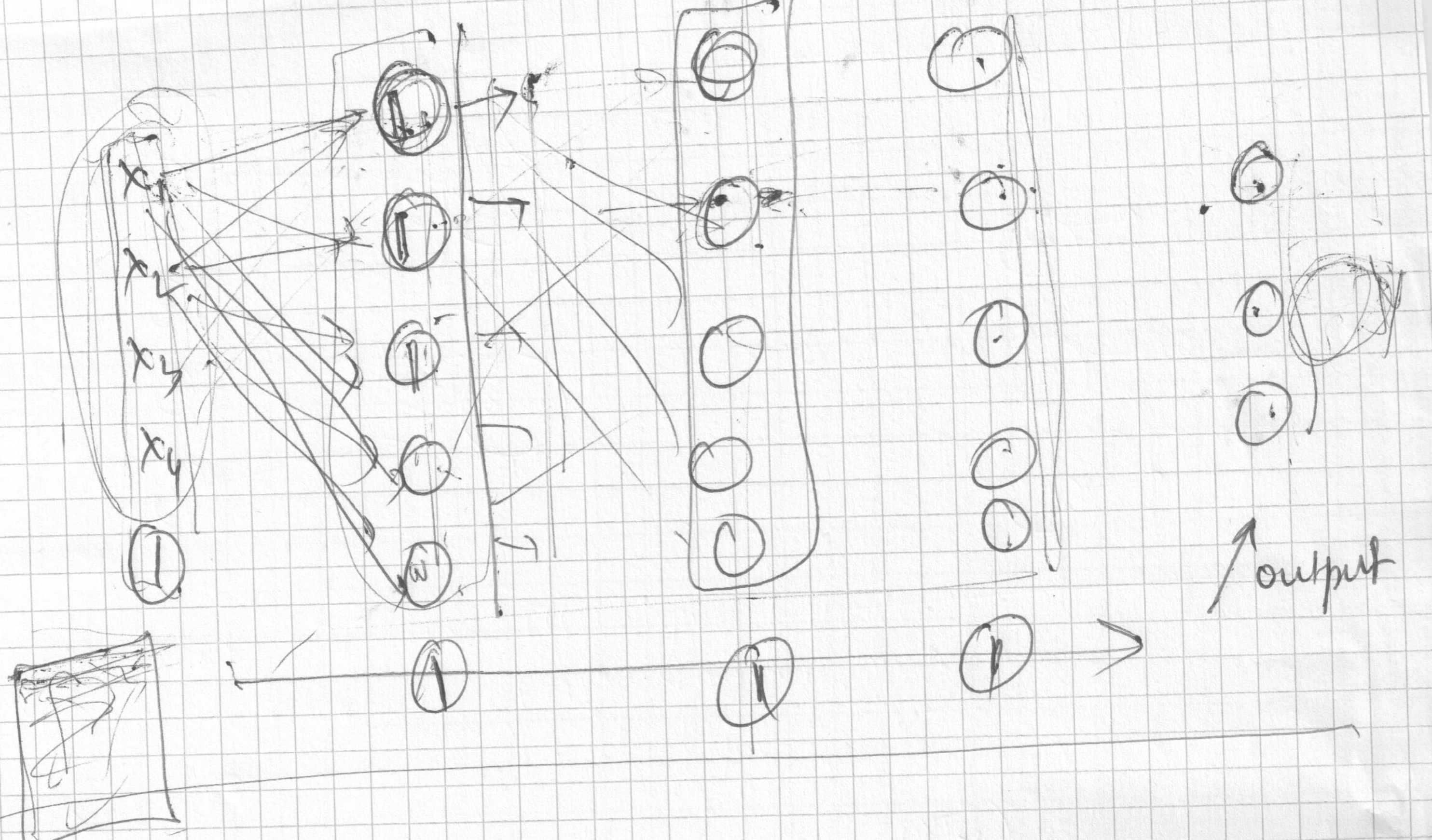
---

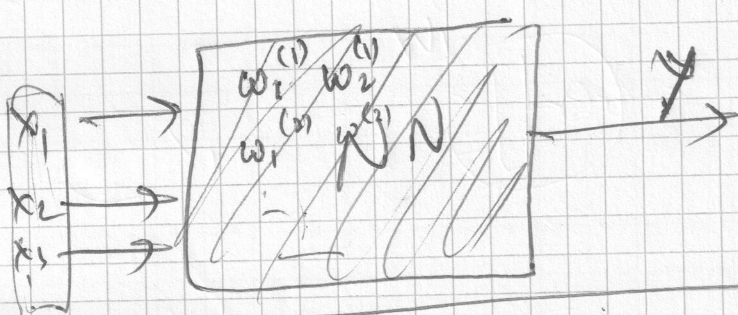
let us consider an even simpler network.

let $x \in R^2$

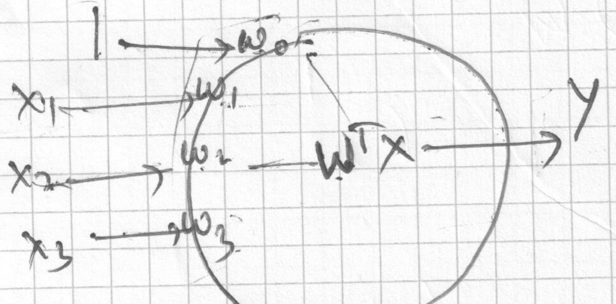


activation - <u>output of a unit</u>





$L(w^{(1)}, w^{(2)} \ldots)$



Assuming we have only 1 training example

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

Initialize the weight vector.

$$w^T x = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

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

$$\frac{d}{dw_0} L(w) = -(y - w^T x)$$

$$\frac{d}{dw_1} L(w) = -(y - w^T x)\, x_1$$

new weight.

$$w_0 = w_0 - \eta \frac{d}{dw_0} L(w)$$

$$= w_0 - \eta\,(-y - w^T x)$$

$$= w_1 - \eta\,(-(y - w^T x)\, x_1)$$