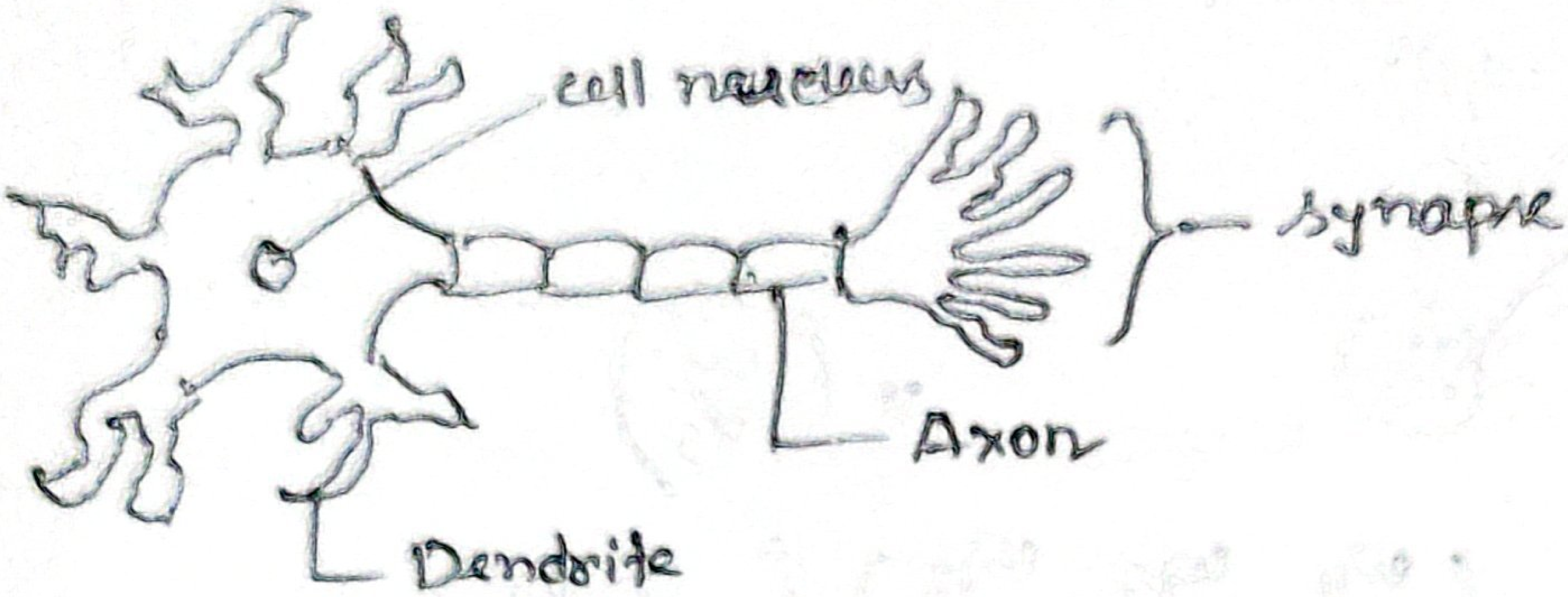
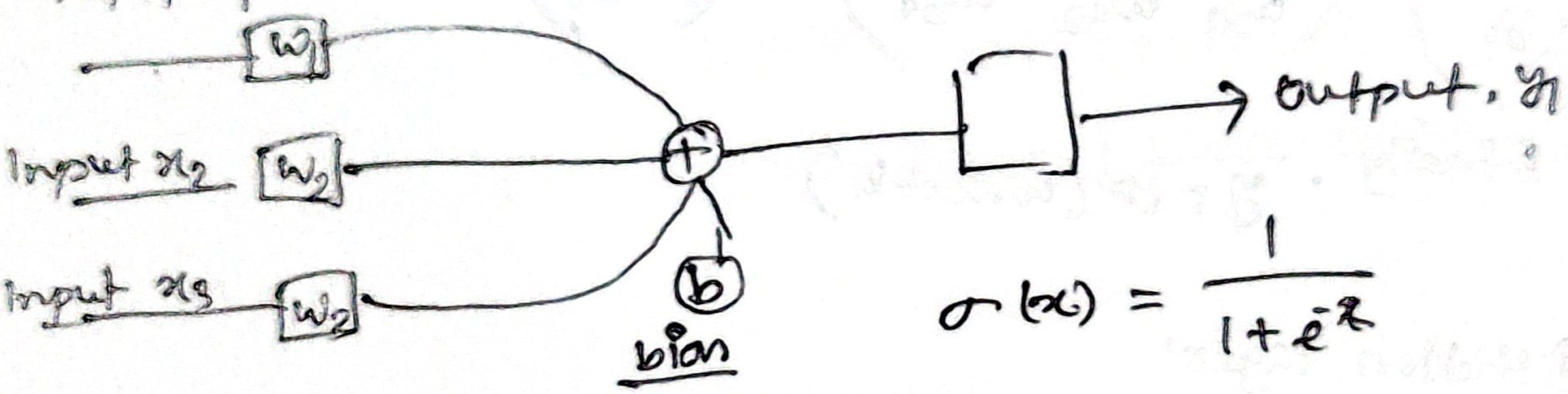
Deep learning 02.02.25




Linear



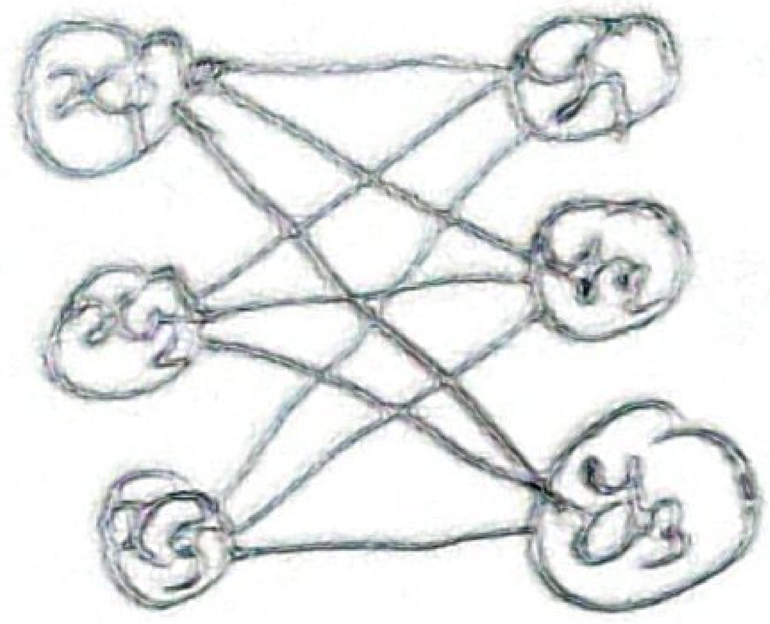

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$y_1 = \sigma(w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + b) = \sigma\left(\sum_i w_i x_i + b\right)$$

$$\boxed{mx+c}$$

Artificial Neuron (single layer perceptron)

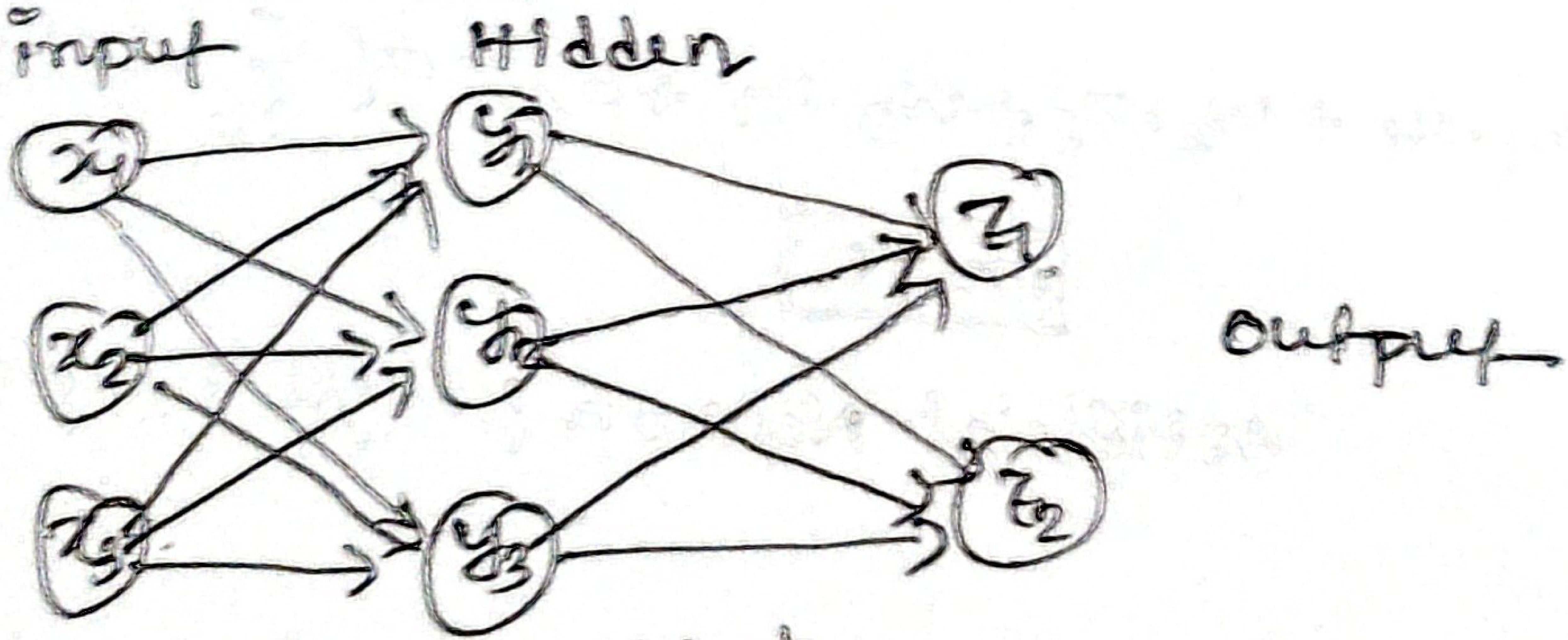
(4) multiple perception to create a layer:




$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{34} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

Finally. $y = \sigma(w.x+b)$

(5) Hidden layer:




$y =$ $w_1 . b_1$ $w_2 , b_2$

$y = \sigma(w_1 . x + b_1)$ ~~in~~ output

$z = \sigma(w_2 . y + b_2)$

↓

input for second layer.

$z = \sigma(w_2 . \sigma(w_1 . x + b_1) + b_2)$

## (4) Activation function:   Non-Linear

It is a mathematical function that takes the weighted sum of the inputs and Bias as input and then generates an output, typically used to add no-linearity to the model.

→ Sigmoid. $\sigma(x) = \frac{1}{1+e^{-x}}$ ; Range → (0 - 1)

→ Hyperbolic Tangent, $\sigma(x) = \tanh(x)$ ; Rang (-1 to +1)

→ Rectified Linear Unit, $\sigma(x) = \max(x, 0)$

(Relu) → ~~Also~~ (pos) → (-) ⇒ 0 [Linear response]

→ Softmax - mali-class operation, (0,1), all class. will be 1.

## # Relu: (Rectified Linear unit):

$\text{Relu}(x) = \max(0, x)$

make negative values to zero, positive value remain unchanged.

Ex

$x = -3, \ f(-3) = \max(0, -3) = 0$

$x = 2, \ f(2) = \max(0, 2) = 2$

* **Leaky ReLU: (Linear)**

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases}$$

$\alpha$ is a small positive coefficient such as, 0.01

Ex:
if, $\alpha = 0.01$, $x = -5$, then,

$f(x) = 0.01 \times -5 = -0.05$

* **ELU: (non-Linear)**

$$(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha(e^{x}-1) & \text{if } x < 0 \end{cases}$$

provide a non-zero gradient for negative values, which helps reduce the vanishing gradient effect.

* **Swish:** $(x) = x \cdot \sigma(x)$

$= x \cdot f(x)$. [combination of sigmoid]

$= x \cdot \frac{1}{1+e^{-x}}$

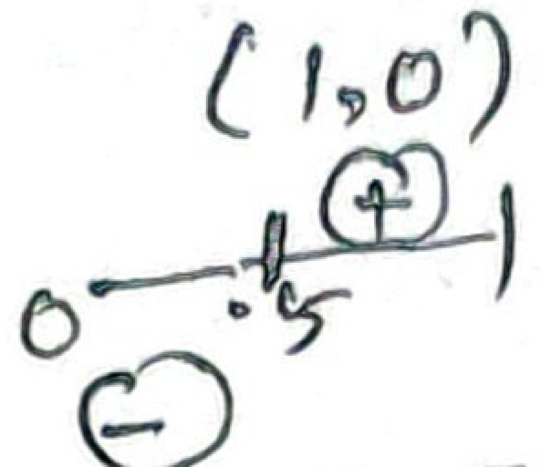
* $\text{Tanh}(x) = \frac{e^{x} - e^{-x}}{e^{x}+e^{-x}}$ Hidden layer $(-1, 1)$

much more complex

## Sigmoid : For logistic function : ~~and~~

→ used output layer.

→ Binary class classification.




→ vanishing gradient problem, non-zero centered Output, computational output is high.

$\boxed{\text{Softmax :}}$ $(z)_i = \dfrac{e^{z_i}}{\sum_j e^{z_j}}$

⇒ For multiclass problem classification

Let's assume :

Score for 1st class = 2.0

" " 2nd " = 1.0

" " 3rd " = 0.5

* $e^{2.0} \simeq 7.389$
* $e^{1.0} \simeq 2.718$
* $e^{0.5} \simeq 1.649$

total = 7.389 + 2.718 + 1.649 = 11.756

Probability for class − 1 = $\dfrac{7.389}{11.756} = 0.628$

" " " − 2 = $\dfrac{2.718}{11.756} = 0.231$

" " " − 3 = $\dfrac{1.649}{11.756} = 0.140$

Sum = 1.0, as expected.

⑧ For regression problem:

Linear activation problem: $f(x) = x$

input: $x$, output $= x$