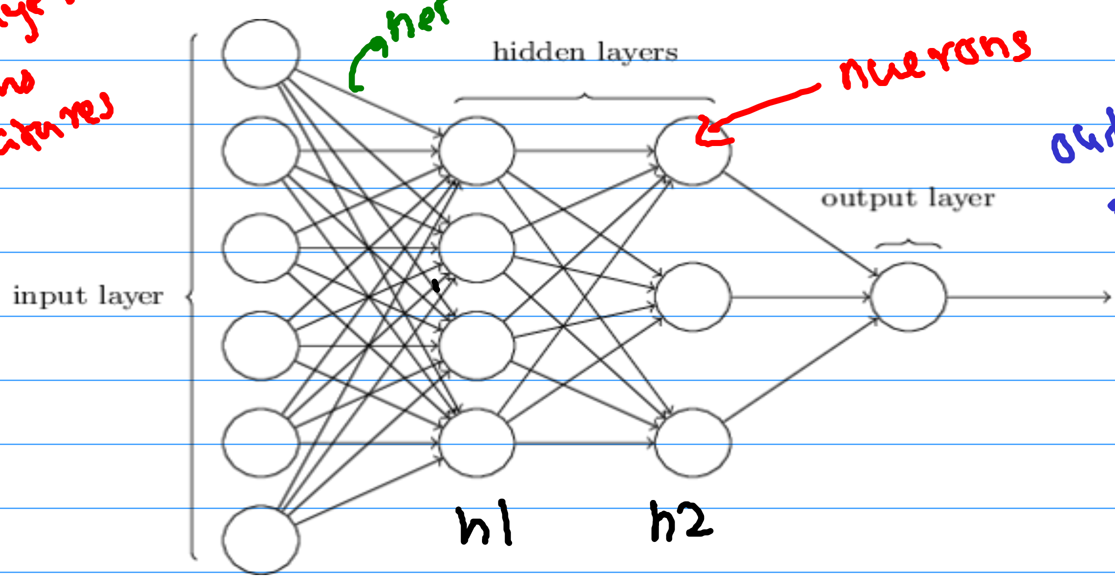
Anaconda module installation

(i) opencv
~~(ii) scikit-learn~~ inbuilt Library
(iii) tensorflow
(iv) Keras

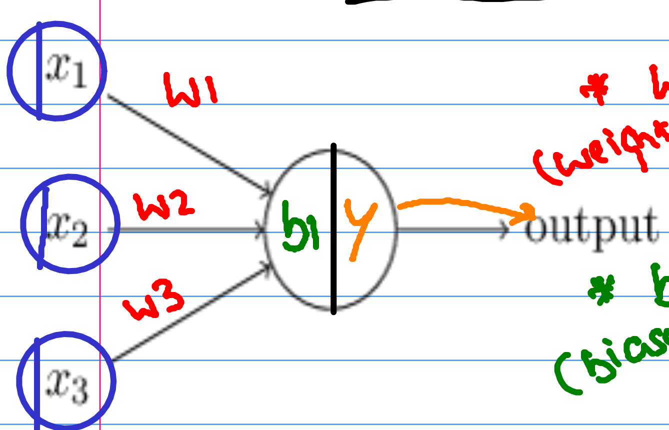
Nueral Network Architecture

input layer
neurons
= # features



output layer
neurons
= # labels

Perceptrone (Artificial neuron model)



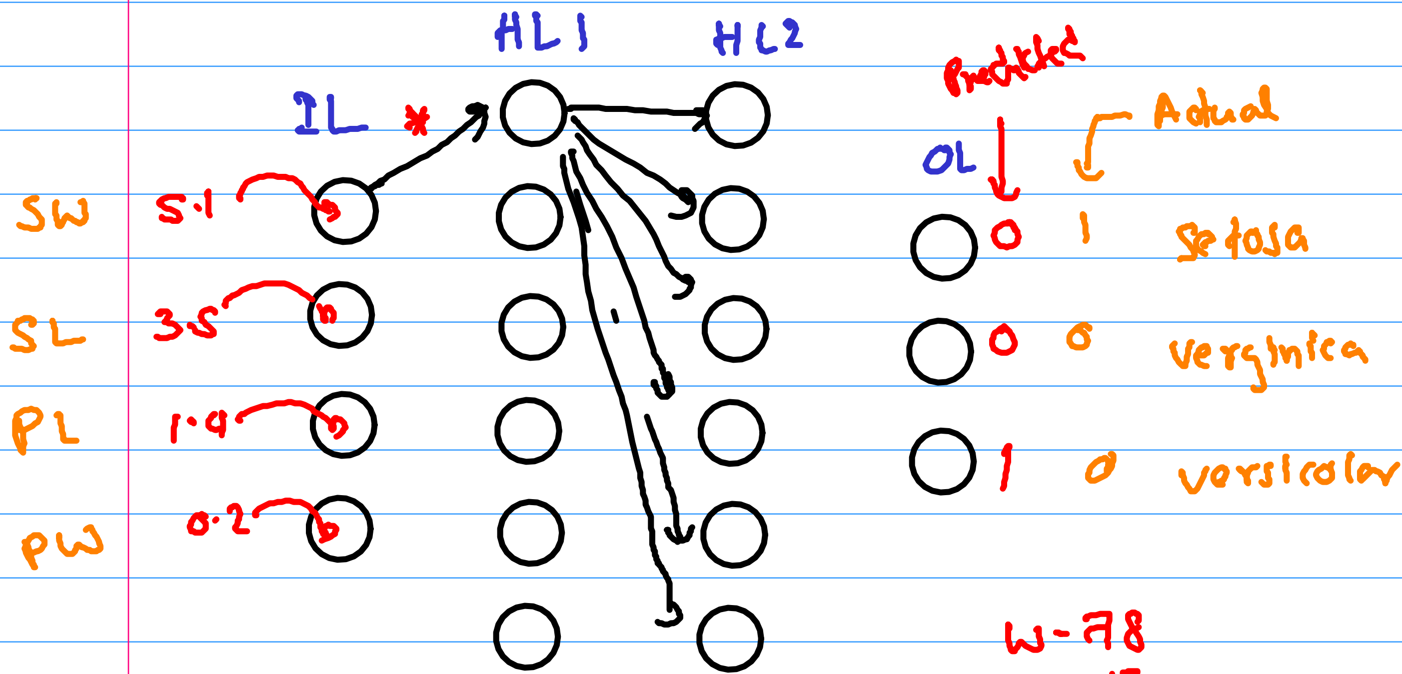
* W - All the nets in the NN have
(weights) identical weights

* b - All the neurons in the NN,
(bias) otherthan the neurons in the input layer have identical biases

$$z = \sum_{i=1}^{n} w_i \cdot x_i = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 x_3$$

$$Y = \begin{cases} 1 ; & z + b > 1 \\ 0 ; & z + b \leq 0 \end{cases} \qquad -\infty < w, b < +\infty$$

Training a NN

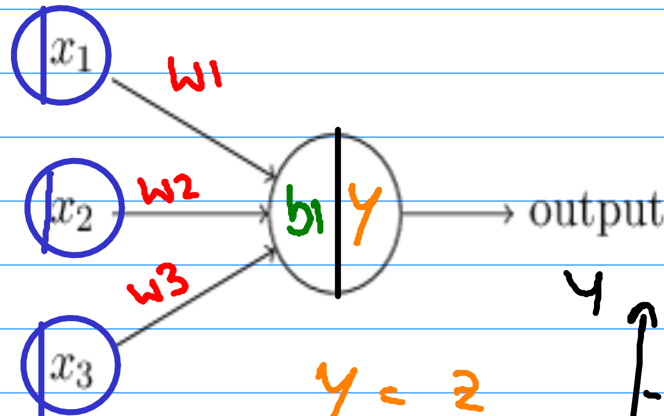


W ~ 78
b ~ 15

Activation Functions

(1) Linear
(2) Non Linear

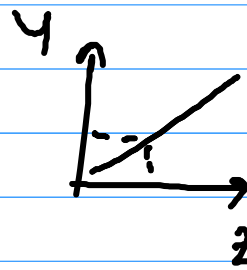
Linear Activation



$$z = \sum_{i=1}^{n} w_i \cdot x_i + b$$

$$z = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + b$$

$y = z$

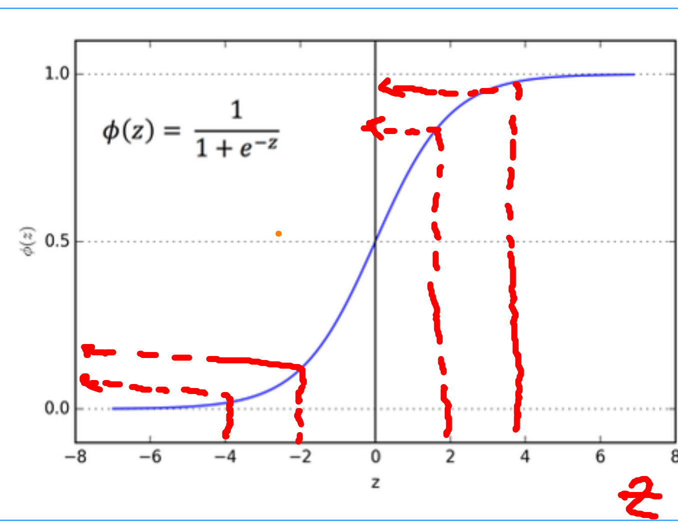


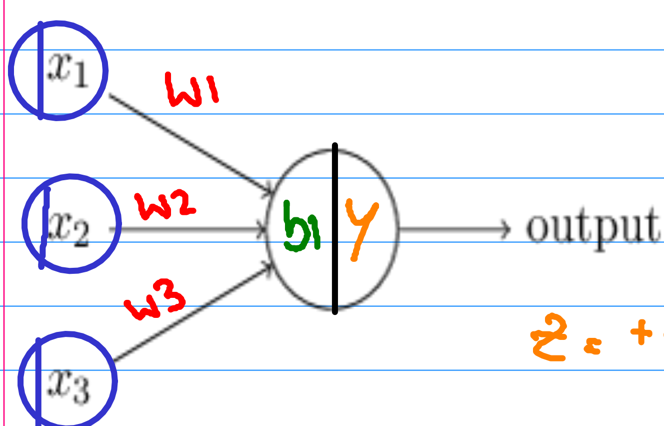
* mostly used in Regression problems

Non-Linear

(i) Sigmoid

$$y = \frac{1}{1+e^{-z}} \; ; \; z = \sum_{i=1}^{n} w_i x_i + b$$

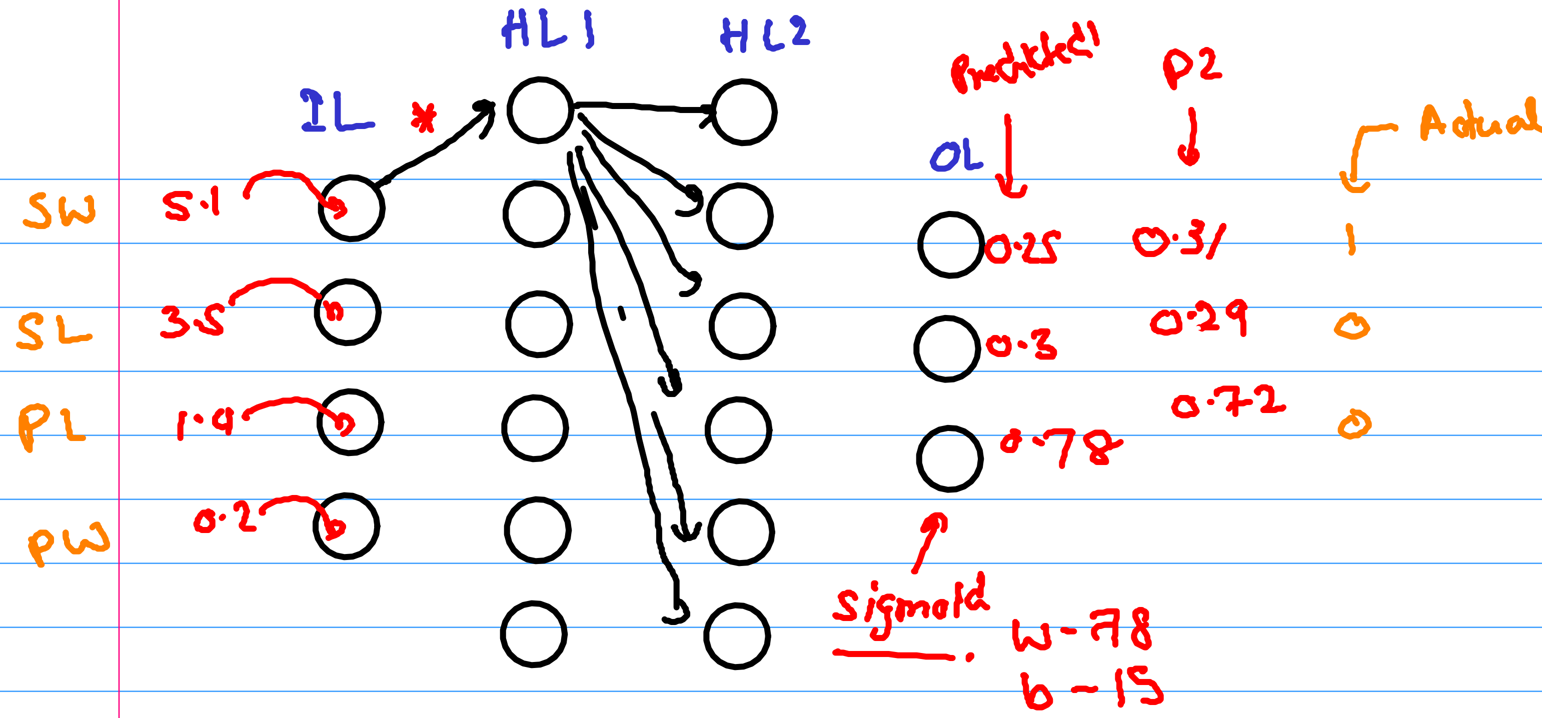




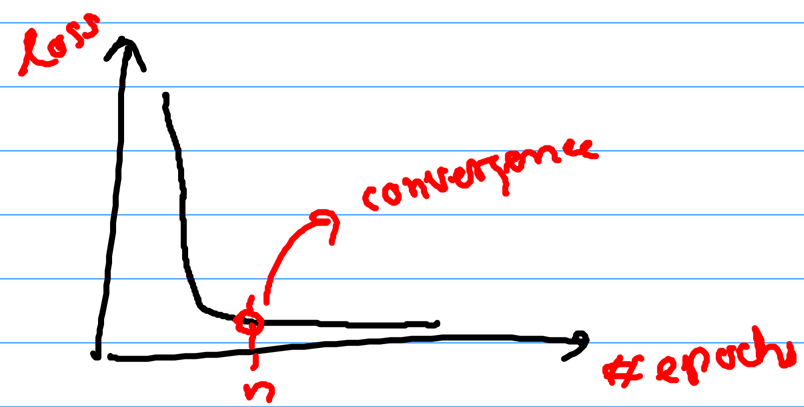
$z = +\infty \; ; \; y = \frac{1}{1+e^{-\infty}} = \frac{1}{1+1/e^{\infty}} = \frac{1}{1+\frac{1}{\infty}} = \frac{1}{1+0} = 1$

$z = -\infty \; ; \; y = \frac{1}{1+e^{\infty}}$

$y = \frac{1}{1+\infty} = \frac{1}{\infty} = 0$



Loss Function



softmax Activation

output layer

sigmoid	softmax
0.8	0.4 $0.8/(0.8+0.7+0.3)$
0.7	0.38 $0.7/(0.8+0.7+0.3)$
0.3	0.22 $0.3/(0.8+0.7+0.3)$

Loss Functions

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)}\log(\hat{y}^{(i)}) + (1-y^{(i)})\log(1-\hat{y}^{(i)})\right]$$

(i) crossentropy loss (classification)

y - actual result, $\hat{y}$ - Predicted result

output layer

Predicted	Actual
0.8	1
0.7	0
0.3	0

$$L = -\frac{1}{3}\Big[1\log(0.8) + (1-1)\log(1-0.8) + 0\log(0.7) + (1-0)\log(1-0.7) + 0\log(0.3) + (1-0)\log(1-0.3)$$

(ii) MSE (Regression problems)

output layer

Actual	predicted
0.8	2.2

$$L = \frac{(0.8-2.2)^2 + }{n}$$

Gradient Decent optimizer

$w + \Delta w$