Logistic Regression

Activation Functions

Output from every neuron is generated after applying activation Function to the values being calculated with set of inputs and their weights.

Most Popular Activation Functions are ReLU (Rectified Linear Units) and Sigmoid, Softmax and tanh

Activation functions should be differentiable. Non-linear Activation functions help you to bring non-linearity in the system

To transform/squash your input to a different space/domain and do some kind of thresholding

Shape of Your Input: **X** (n x m)

Number of IO pairs available for training the network

No of ***Data Points*** in Single Input

$x_1^{(1)}$	$x_1^{(2)}$	$x_1^{(3)}$	$x_1^{(4)}$	..	..	..	..	$x_1^{(m-1)}$	$x_1^{(m)}$
$x_2^{(1)}$	$x_2^{(2)}$	$x_2^{(3)}$	$x_2^{(4)}$	..	..	..	..	$x_2^{(m-1)}$	$x_2^{(m)}$
$x_3^{(1)}$	$x_3^{(2)}$	$x_3^{(3)}$	$x_3^{(4)}$	..	..	..	..	$x_3^{(m-1)}$	$x_3^{(m)}$
$x_4^{(1)}$	$x_4^{(2)}$	$x_4^{(3)}$	$x_4^{(4)}$	..	..	..	..	$x_4^{(m-1)}$	$x_4^{(m)}$
:	:	:	:	..	..	..	..	:	:
:	:	:	:	..	..	..	..	:	:
:	:	:	:	..	..	..	..	:	:
:	:	:	:	..	..	..	..	:	:
:	:	:	:	..	..	..	..	:	:
:	:	:	:	..	..	..	..	:	:
$x_{n-1}^{(1)}$	$x_{n-1}^{(2)}$	$x_{n-1}^{(3)}$	$x_{n-1}^{(4)}$	..	..	..	..	$x_{n-1}^{(m-1)}$	$x_{n-1}^{(m)}$
$x_n^{(1)}$	$x_n^{(2)}$	$x_n^{(3)}$	$x_n^{(4)}$	..	..	..	..	$x_n^{(m-1)}$	$x_n^{(m)}$

Shape of Your Output: Binary Classification

Number of Outputs corresponding to inputs for training the network

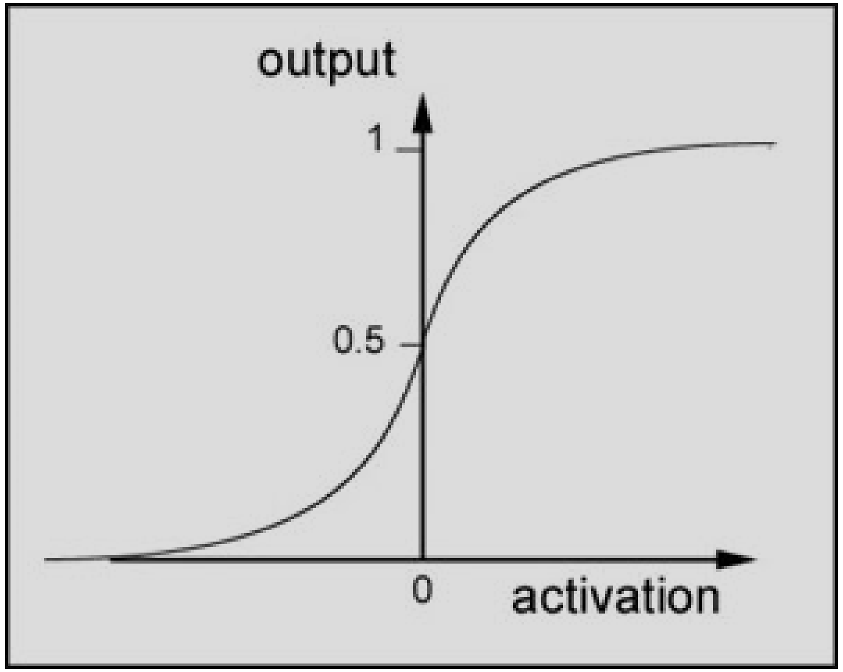
$y^{(1)}$	$y^{(2)}$	$y^{(3)}$	$y^{(4)}$	..	..	..	..	$y^{(m-1)}$	$y^{(m)}$

Y.Shape = (1 , m)

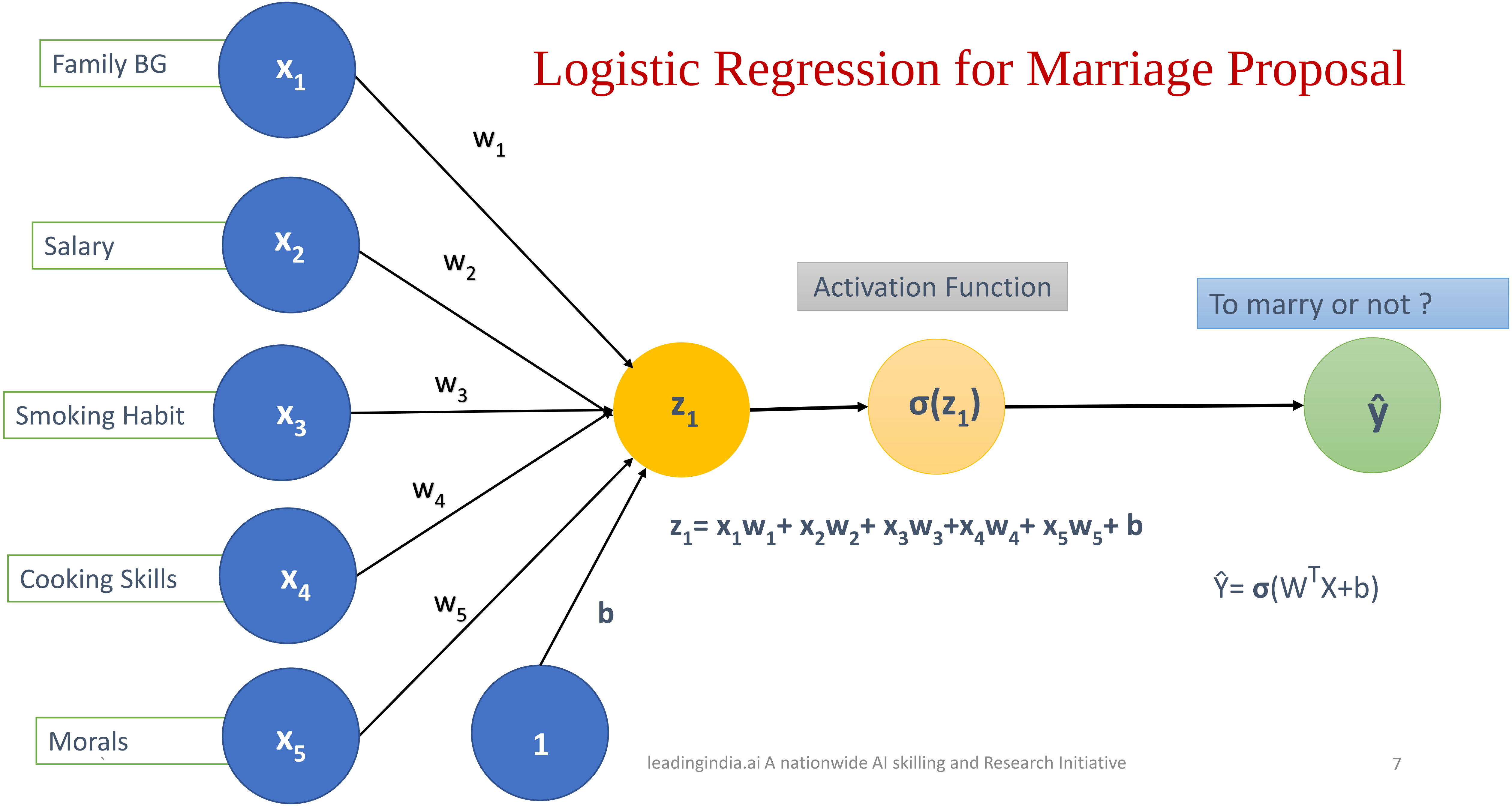
Logistic Regression

Sigmoid Activation Function

- $\hat{y}=\sigma\,(w^Tx + b)$ where $\sigma(z)=\frac{1}{1+e^{-z}}$
- If z is very large then e^{-z} is close to zero and $\sigma(z) = \frac{1}{1+0} \approx 1$
- If z is very small then e^{-z} is large and $\sigma(z) = \frac{1}{1+Large\ Number} \approx 0$



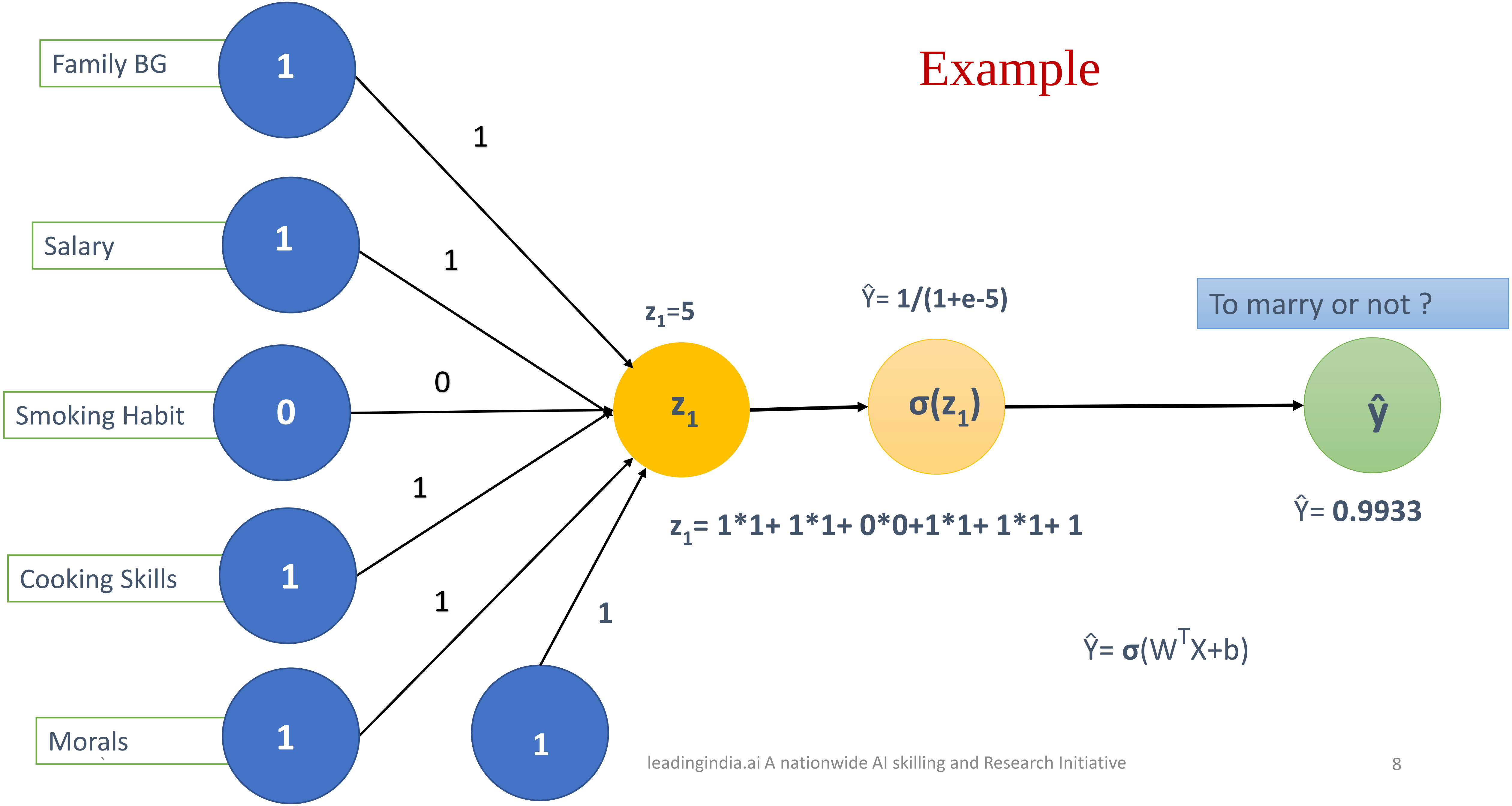
Logistic Regression for Marriage Proposal



$$z_1 = x_1w_1 + x_2w_2 + x_3w_3 + x_4w_4 + x_5w_5 + b$$

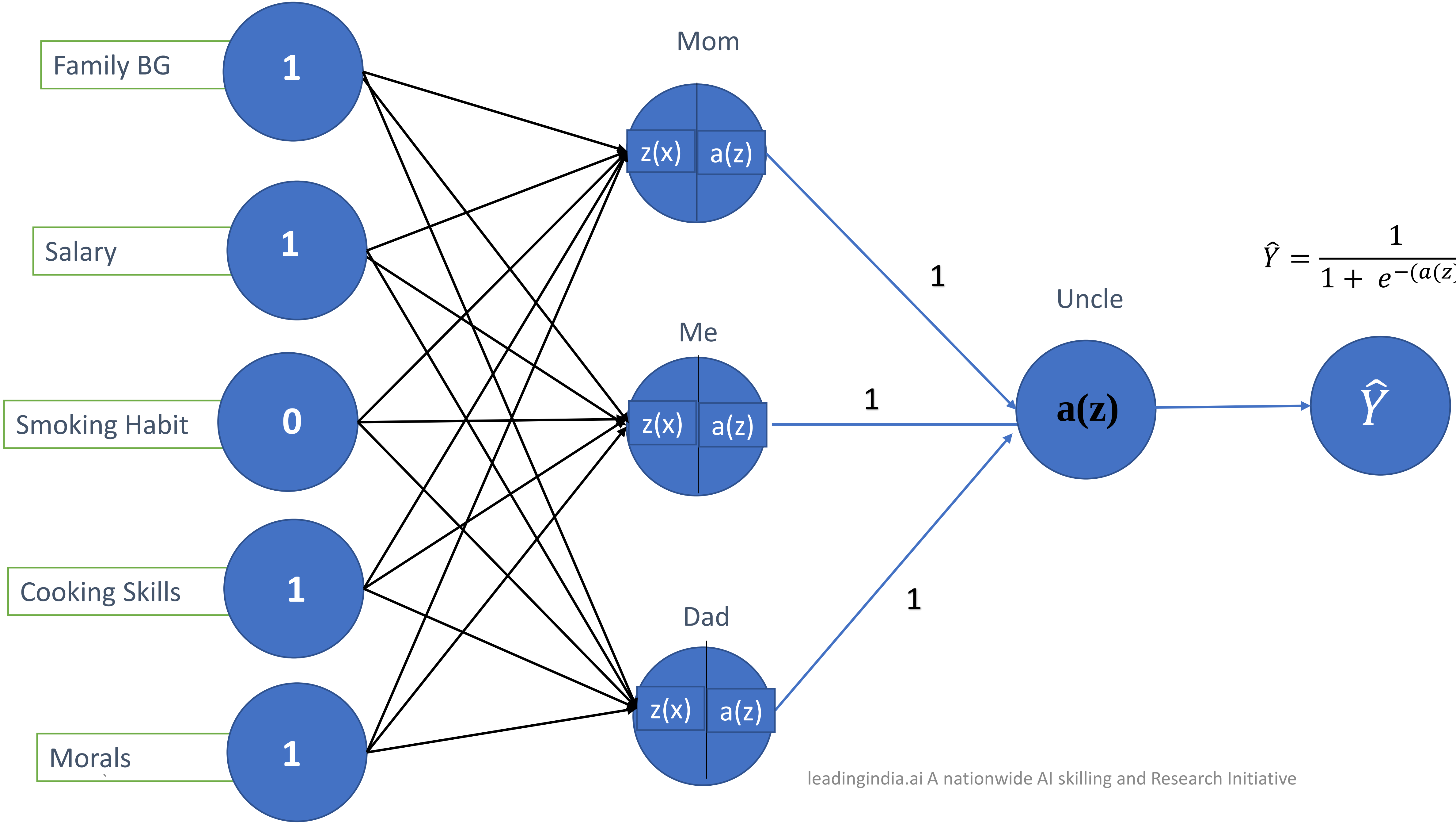
$$\hat{Y} = \sigma(W^TX+b)$$

Example



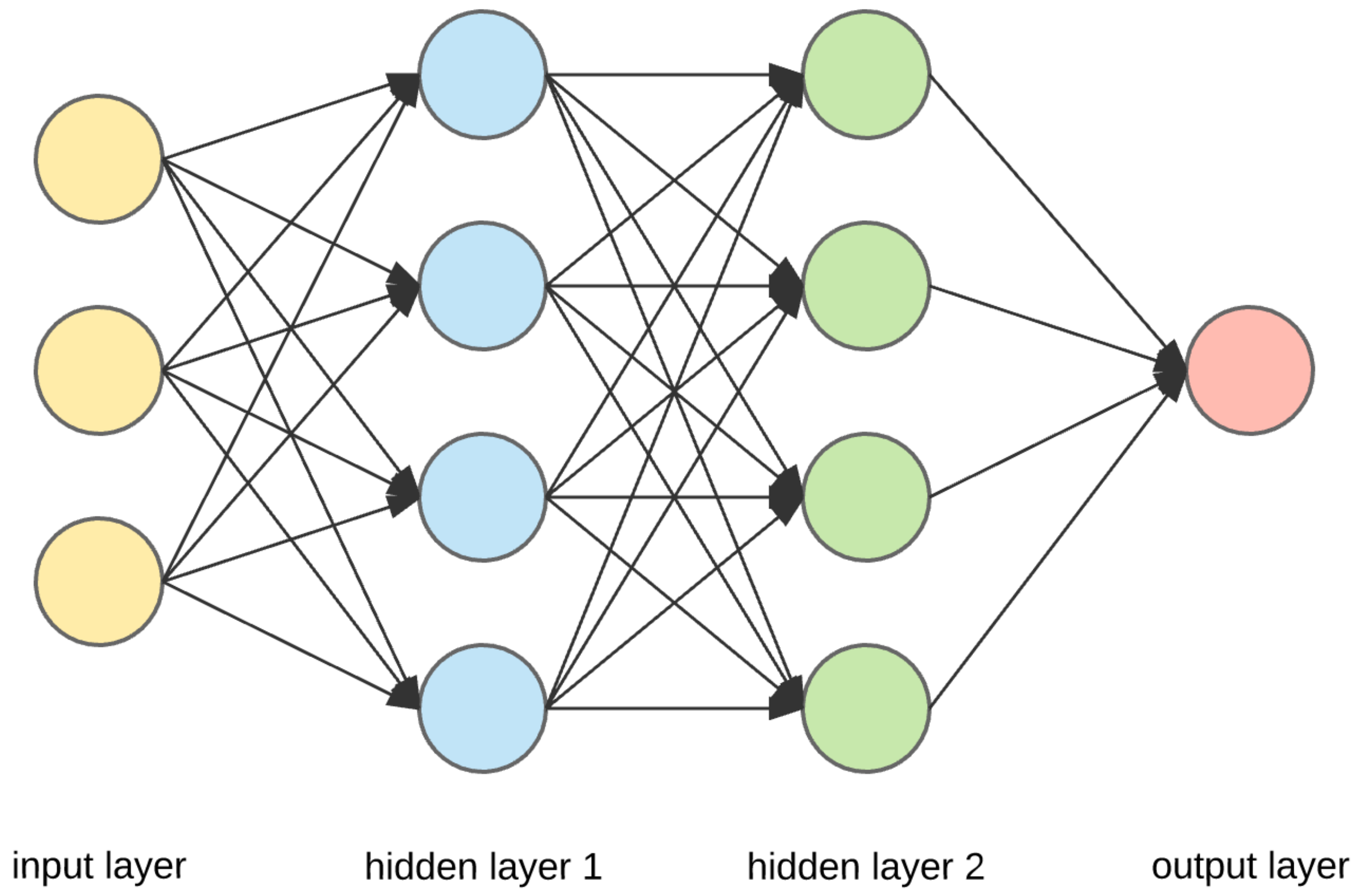
Family BG
1
Salary
1
Smoking Habit
0
Cooking Skills
1
Morals
1
1
1
1
0
1
1
1
$z_1=5$
z_1
$\hat{Y}$= 1/(1+e-5)
$\sigma(z_1)$
To marry or not ?
$\hat{y}$
$\hat{Y}$= 0.9933
z_1= 1*1+ 1*1+ 0*0+1*1+ 1*1+ 1
$\hat{Y}= \sigma(W^TX+b)$

5 * 3 Connections and weights



$$\hat{Y} = \frac{1}{1 + e^{-(a(z))}}$$

A Deep Neural Network



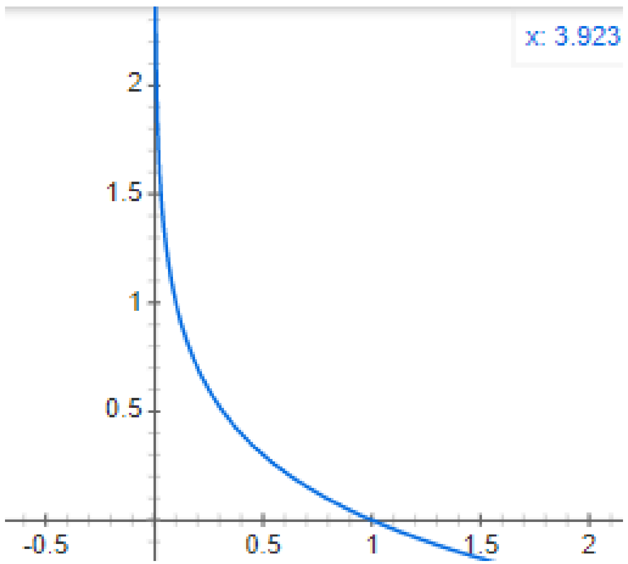
Loss Function for logistic regression

$L(\hat{y}, y) = -((y\log \hat{y}) + (1-y)(\log(1-\hat{y}))$

If $y=1$ $L(\hat{y}, y) = -(\log \hat{y})$ Here, we want $\log \hat{y}$ large , want $\hat{y}$ large that means $\hat{y} \approx 1$

If $y=0$ $L(\hat{y}, y) = -(\log(1-\hat{y})$ Here we want $(\log(1-\hat{y})$ large, want $\hat{y}$ small means $\hat{y} \approx 0$

It helps us also to resolve the local minima problem and gives us a convex problem



Cost Function: Average of Loss across the whole input

$\hat{y}^{(i)} = \sigma\,(w^T x^{(i)} + b)$ where $\sigma(z^{(i)}) = {}^{1}/_{1+e^{-z^{(i)}}}$

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L\,(\hat{y}^{(i)}, y^{(i)})$$

$$= -\frac{1}{m}\sum_{i=1}^{m} (y^{(i)}\log \hat{y}^{(i)} + (1 - y^{(i)}) \log (1 - \hat{y}^{(i)}))$$

Our objective is to find w,b that minimize J(w,b)