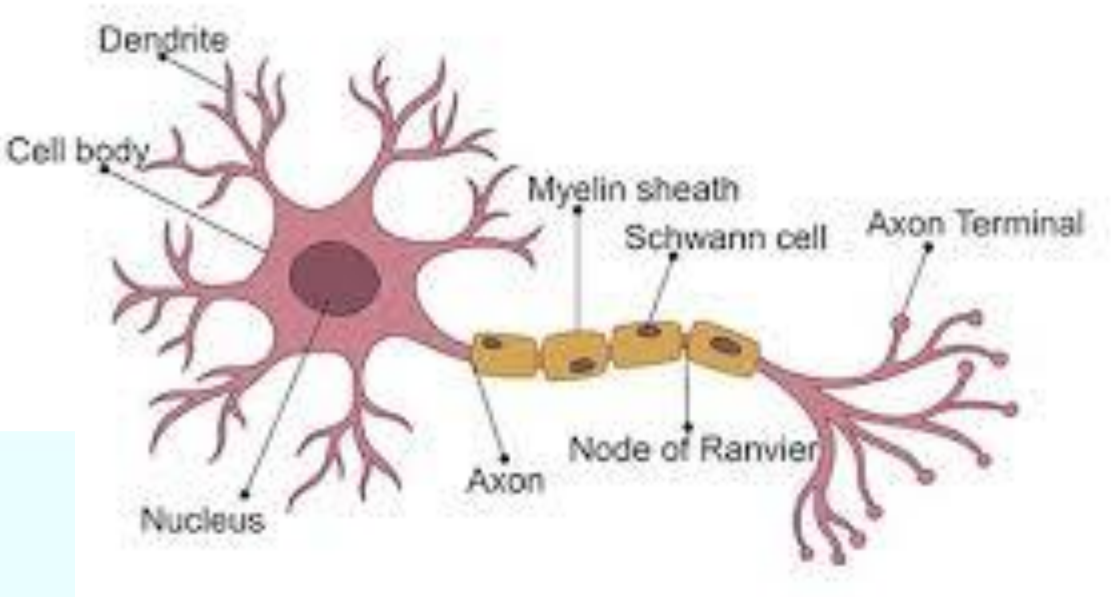
MM 225 – AI and Data Science

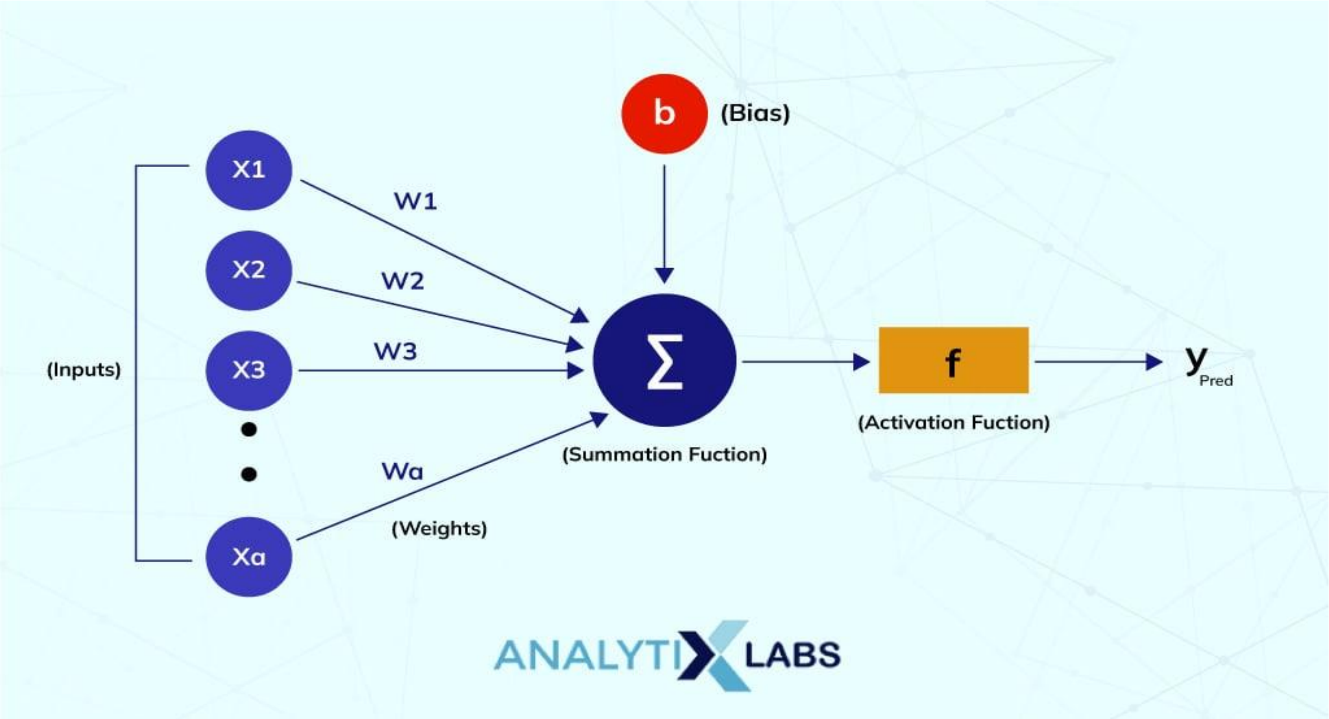
Day 30 : Artificial Neural Network

Instructors: Hina Gokhale, MP Gururajan, N. Vishwanathan

15 OCTOBER 2024

Nature Inspired Neural Network





The case of Logistic Regression

Regression model :

$$y^{(i)} = \boldsymbol{w} \cdot \boldsymbol{x}^{(\boldsymbol{i})} + \ \epsilon_i$$

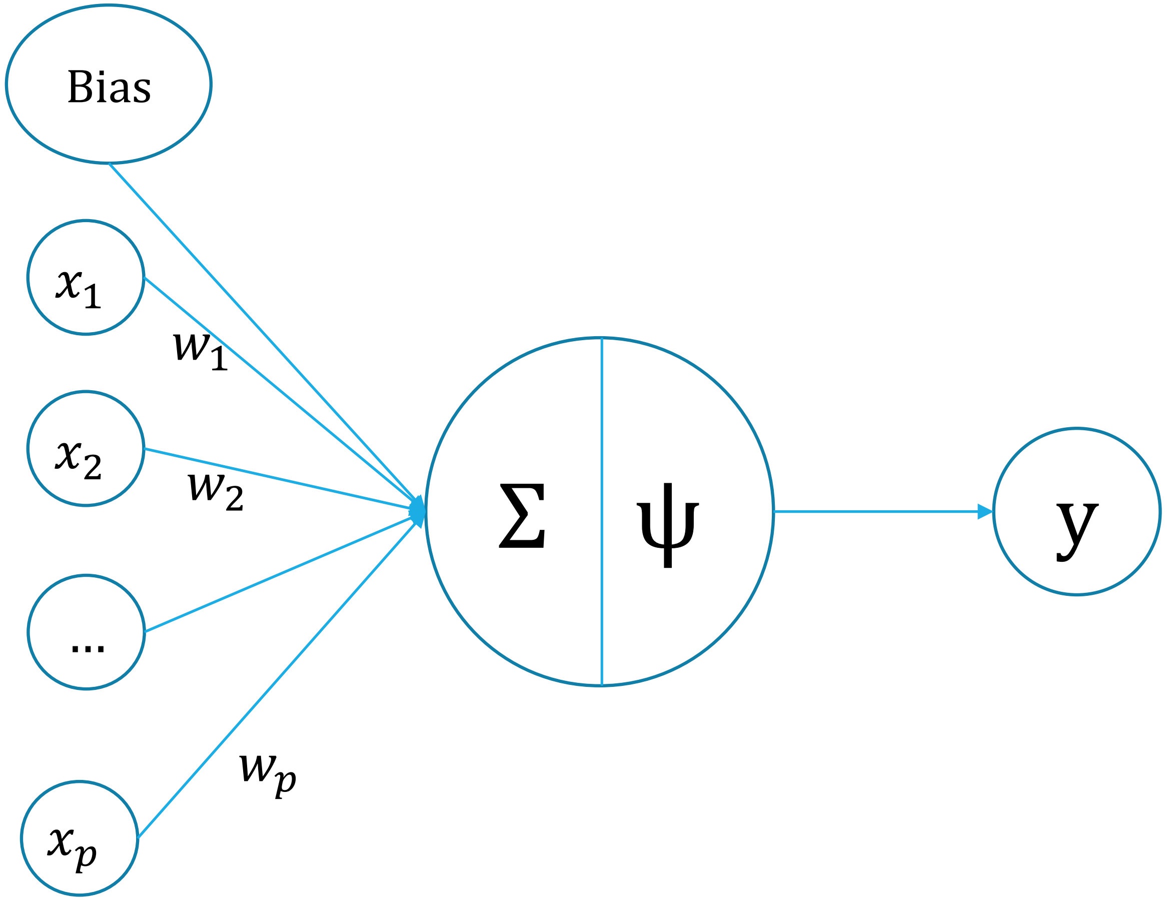
For logistic regression the LHS is “transformed” through sigmoid transformation as follows:

Let $z_i = \boldsymbol{w} \cdot \boldsymbol{x}^{(\boldsymbol{i})}$, then,

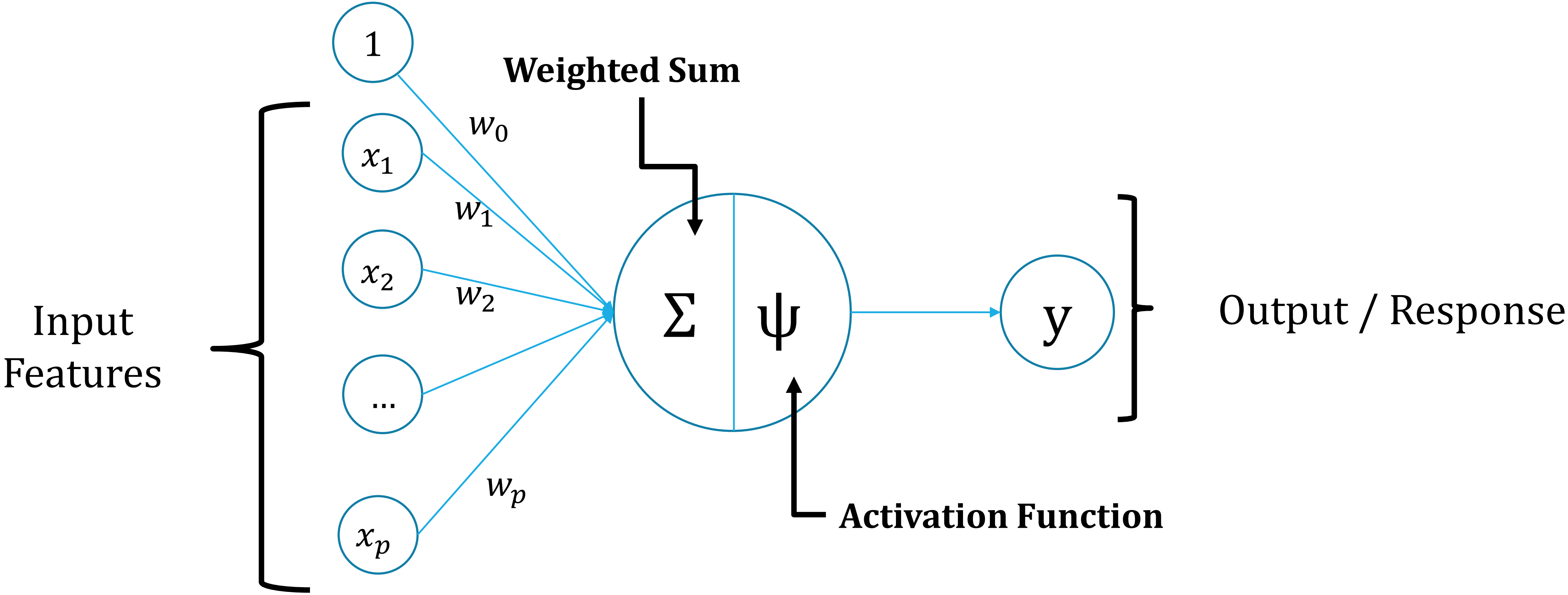
$$\psi(z_i) = \frac{1}{1+e^{-z_i}} = \frac{1}{1+\exp(-z_i)}$$

Logistic Regression Model - Graphically

$y^{(i)} = \boldsymbol{\psi}(\boldsymbol{w} \cdot \boldsymbol{x}^{(i)})$



Sigmoid as Activation Function



Activation Function

There are many choices for activation function

Choice depends on the output range

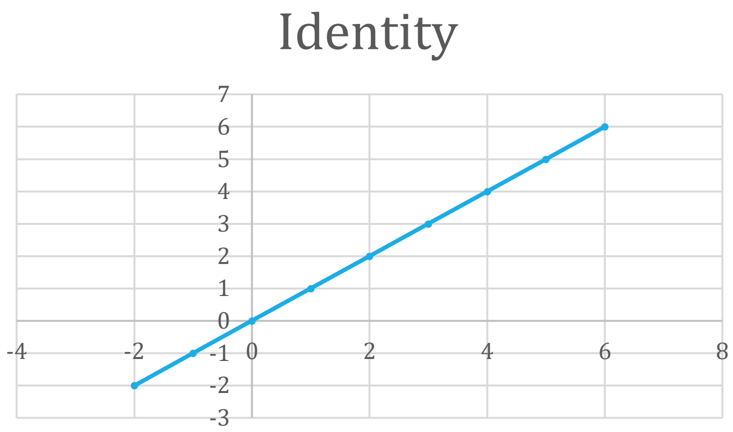
Example: Logistic Regression:

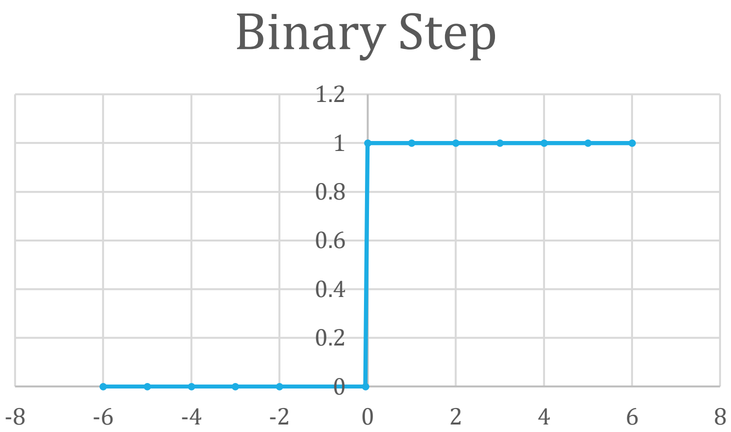
- Output ∈ {0,1}
- Made a choice as sigmoid function ψ(t).
  - Range is (0,1)
  - therefore, output based on threshold values

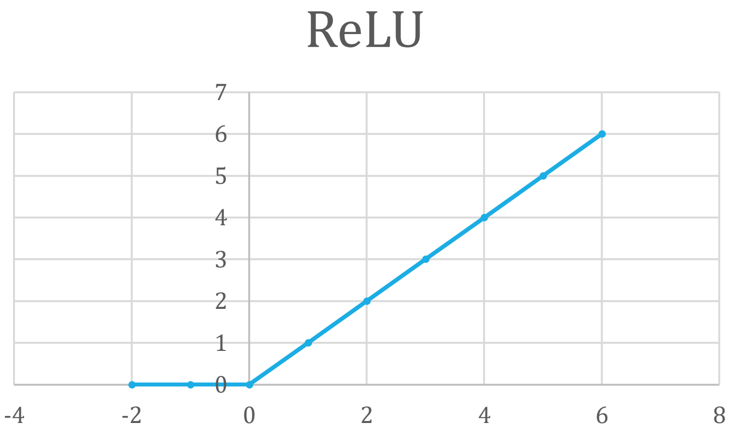
Activation Functions g(t)

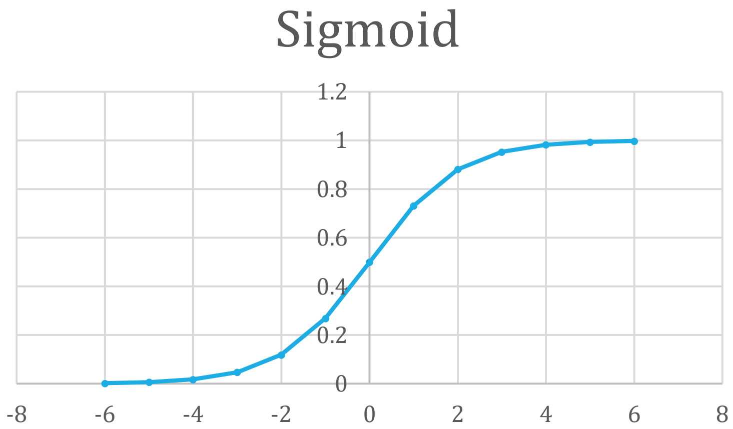
Name	Function g(t)	Gradient	Range
Identity	$g(t) = t$	1	(- ∞, ∞)
Binary Step	$g(t) = \begin{cases} 0\ if\ t < 0 \\ 1\ if\ t \geq 0 \end{cases}$	---	{0,1}
Sigmoid (logistic)	$g(t) = \dfrac{1}{1 + \exp(-t)}$	$g(t)(1 - g(t))$	(0,1)
Hyperbolic Tangent	$g(t) = \tanh(t) = \dfrac{e^t - e^{-t}}{e^t + e^{-t}}$	$1 - (g(t))^2$	(-1,1)
Rectified Linear Unit (ReLU)	$g(t) = \max(0, t)$	$\begin{cases} 0\ if\ t \leq 0 \\ 1\ if\ t > 0 \end{cases}$	[0, ∞)

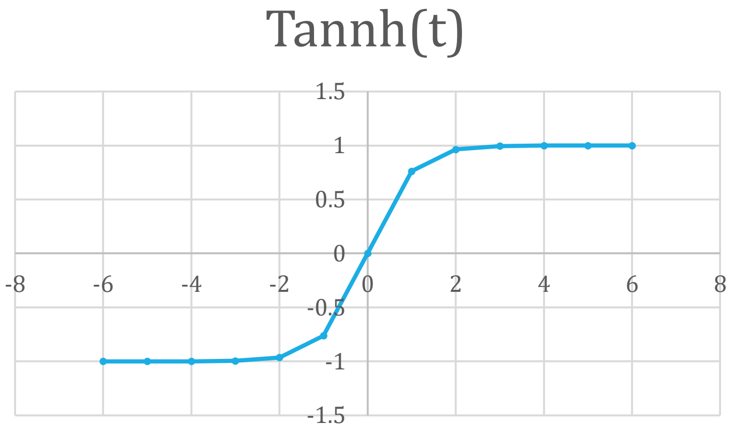
Wikipedia: "Activation Function"


Identity
7
6
5
4
3
2
1
0
-1
-2
-3
-4
-2
0
2
4
6
8

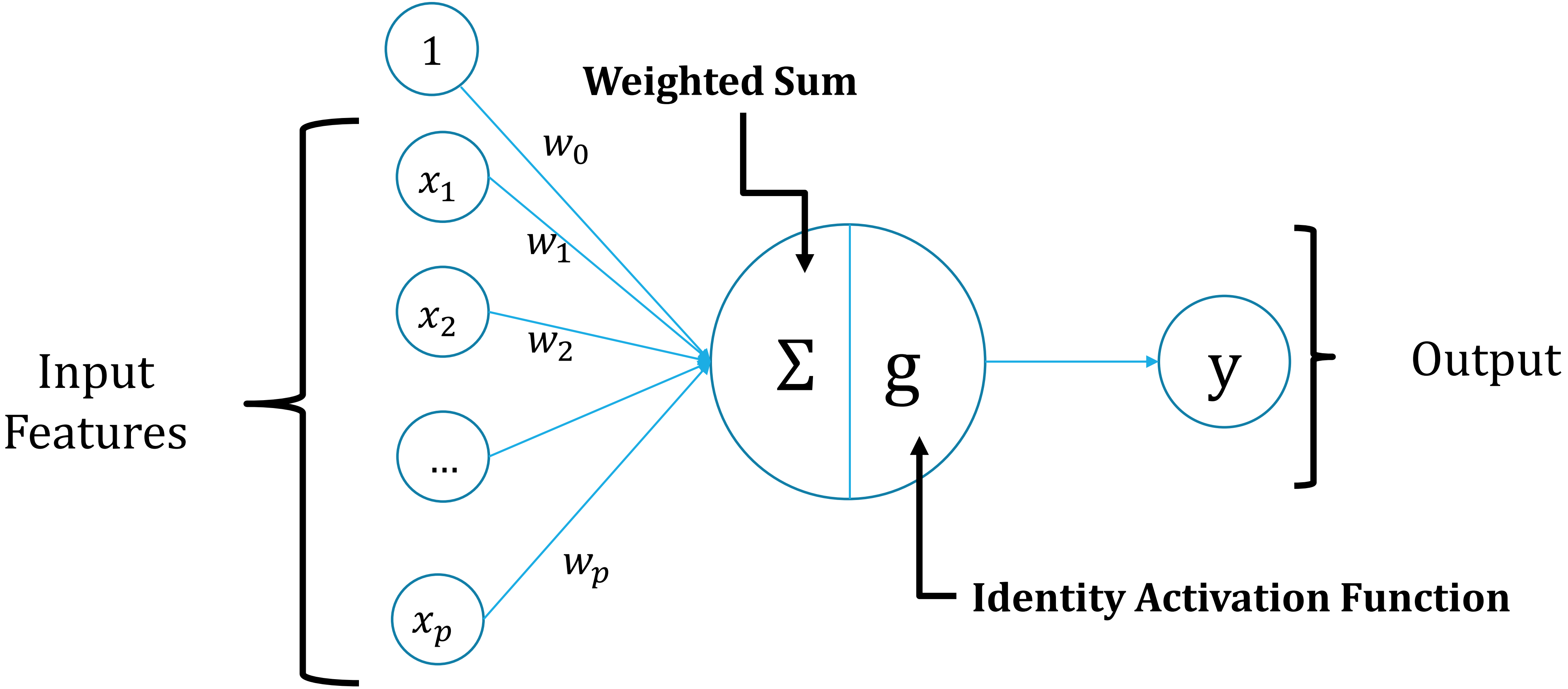

Binary Step
1.2
1
0.8
0.6
0.4
0.2
0
-8
-6
-4
-2
0
2
4
6
8


ReLU
7
6
5
4
3
2
1
0
-4
-2
0
2
4
6
8

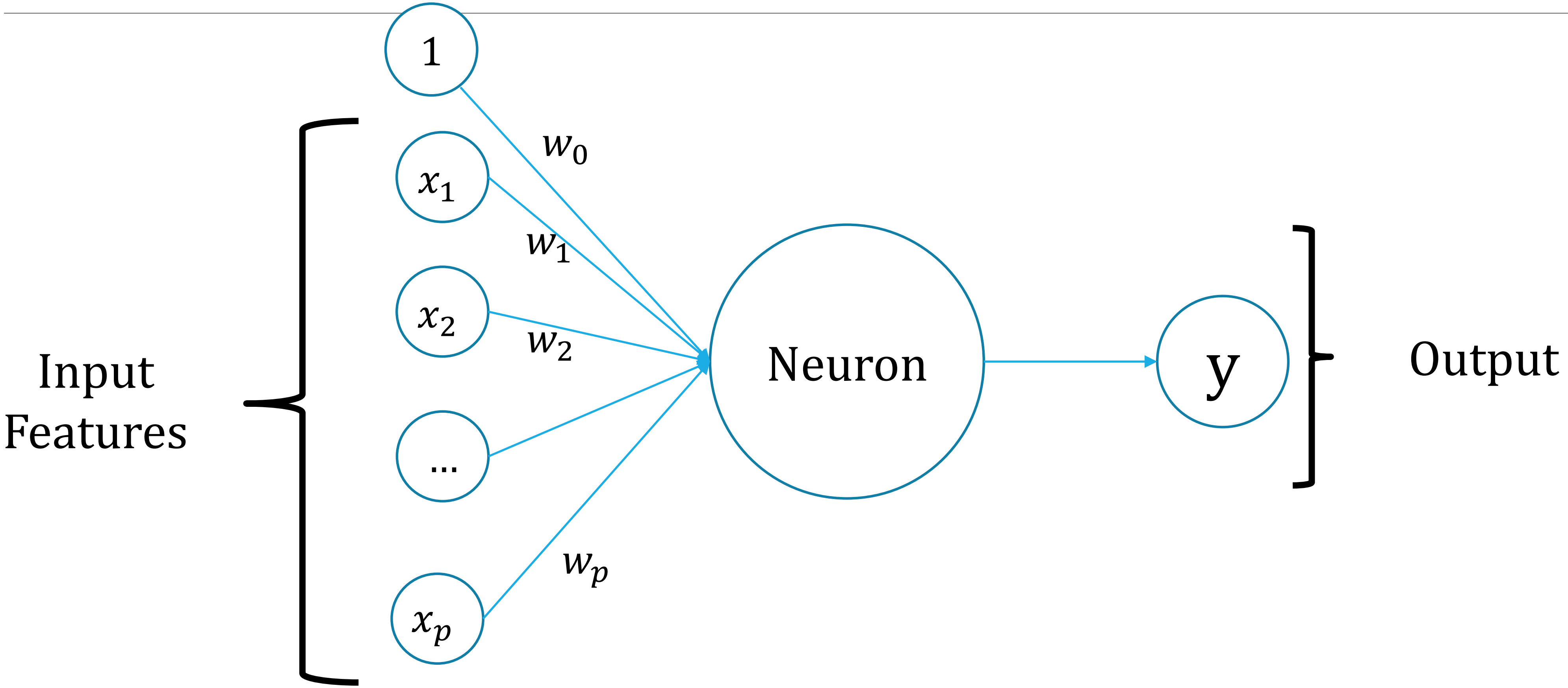

Sigmoid
1.2
1
0.8
0.6
0.4
0.2
0
-8
-6
-4
-2
0
2
4
6
8


Tannh(t)
1.5
1
0.5
0
-0.5
-1
-1.5
-8
-6
-4
-2
0
2
4
6
8

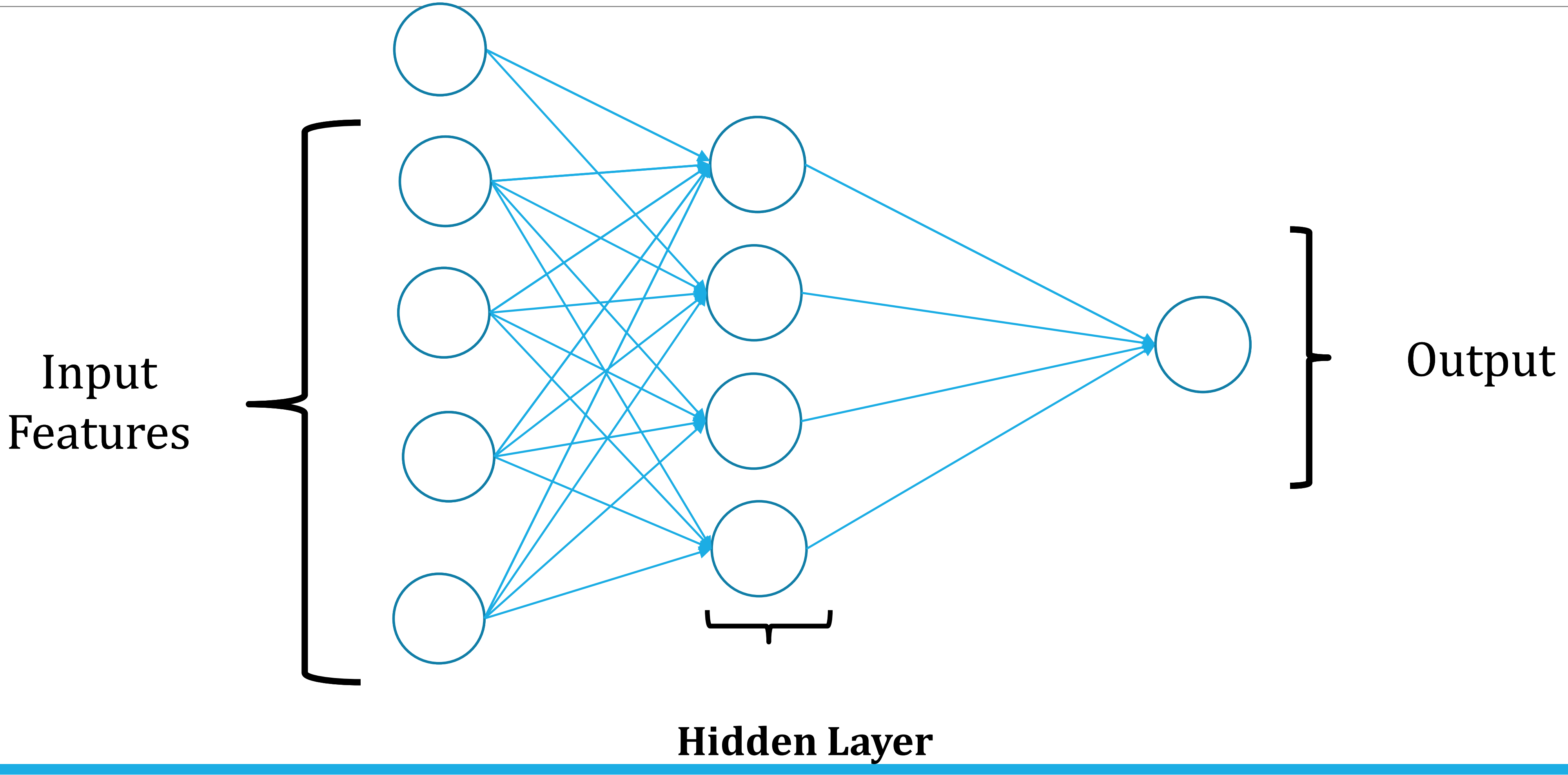
Representation of Linear Regression



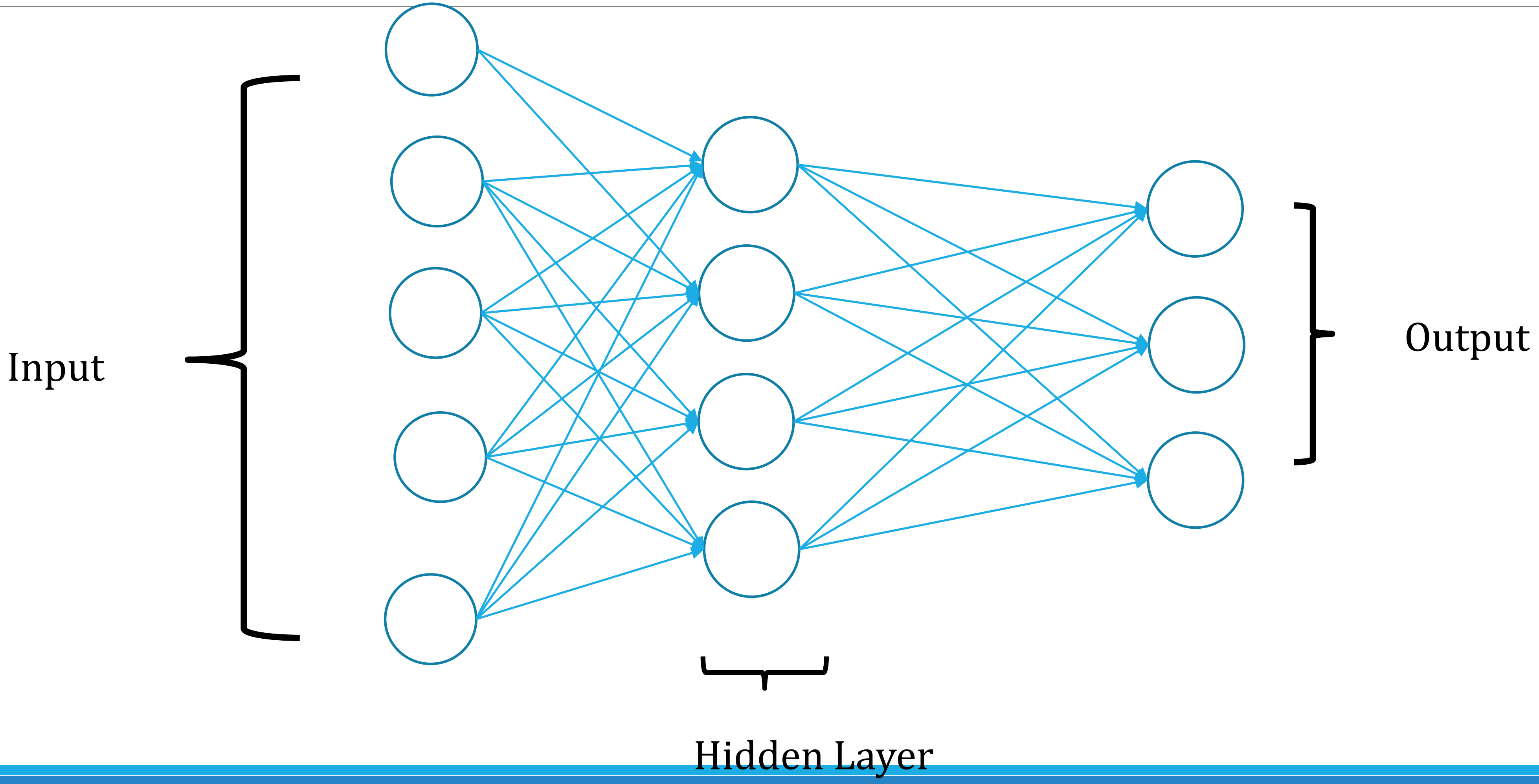
General form



Stacking up more neurons



Simple NN



Weight Estimation

Objective is to minimize the error function $\mathcal{E}$

- $\mathcal{E}$ depends on the output and therefore on activation function g

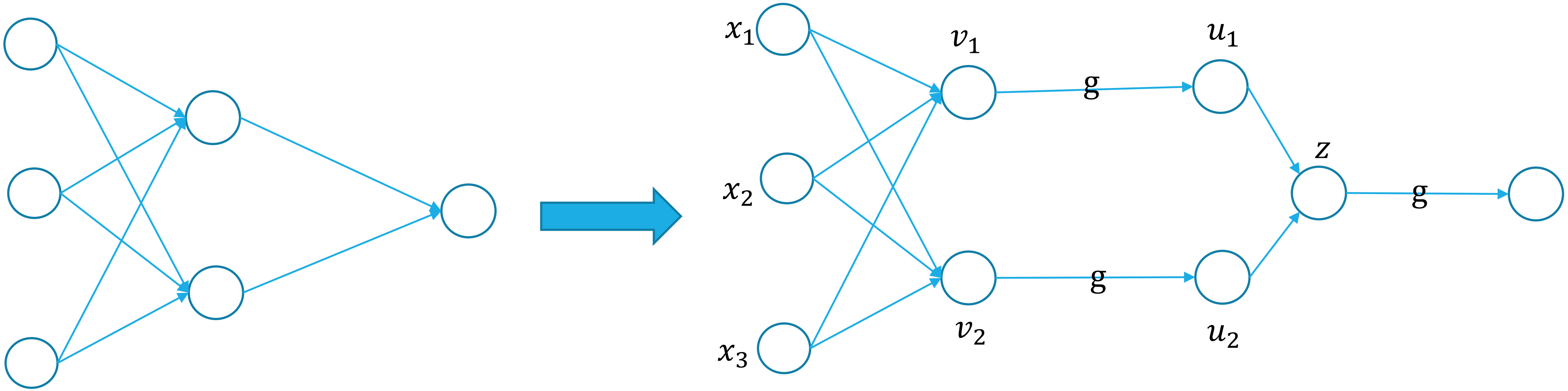
Method of Gradient Descent to be applied

- This requires derivative of activation function wrt weights

Approach is backwards.

- First estimate weights that calculate final output
- Move backwards to estimate previous weights that calculated hidden layer

Expand NN



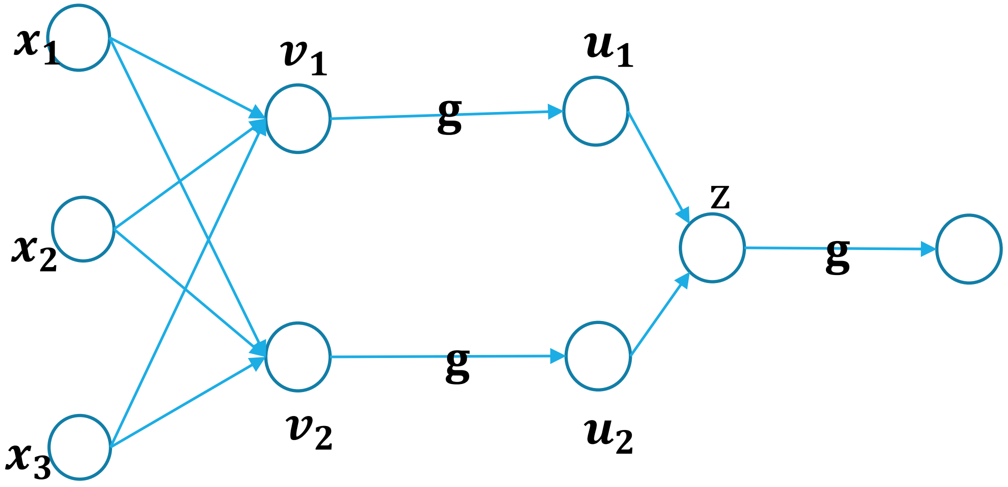
ANN and equation -- Example

$$d = g(z)$$

$$z = w_{31}u_1 + \ w_{32}u_2$$

$$u_k = g(v_k)$$

$$v_k = \sum_{j=1}^{3} w_{kj}x_j, for\ k = 1{,}2$$



Summary

Artificial Neural Network as a perceptron

Logistic Regression as ANN

Variety of Activation functions

Stacking up neurons to make up a network of artificial neurons

Brief introduction to process of Weight Estimation

Thank you…