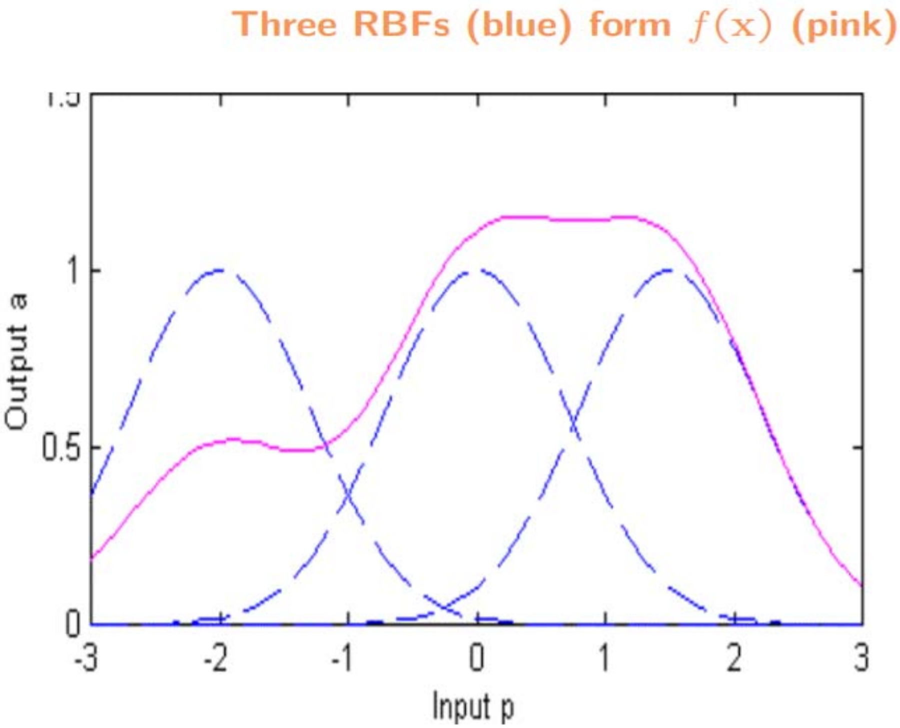
Multivariate Regression

Dr. Pritam Anand.
Assistant Professor,
DA-IICT, Gandhinagar.

Gaussian Basis Function



- $\phi_M(X) = \exp(\frac{-1}{2s_M}||X - cM||^2)$
- $\phi_{M-1}(X) = \exp(\frac{-1}{2s_{M-1}}||X - c_{M-1}||^2)$

…. …. … …. …….. ..

…….. …… …… ….. …..

- $\phi_2(X) = \exp(\frac{-1}{2s_2}||X - c_2||^2)$
- $\phi_1(X) = \exp(\frac{-1}{2s_2}||X - c_1||^2)$
- $\phi_0(X) = 1$

Gaussian Basis Function

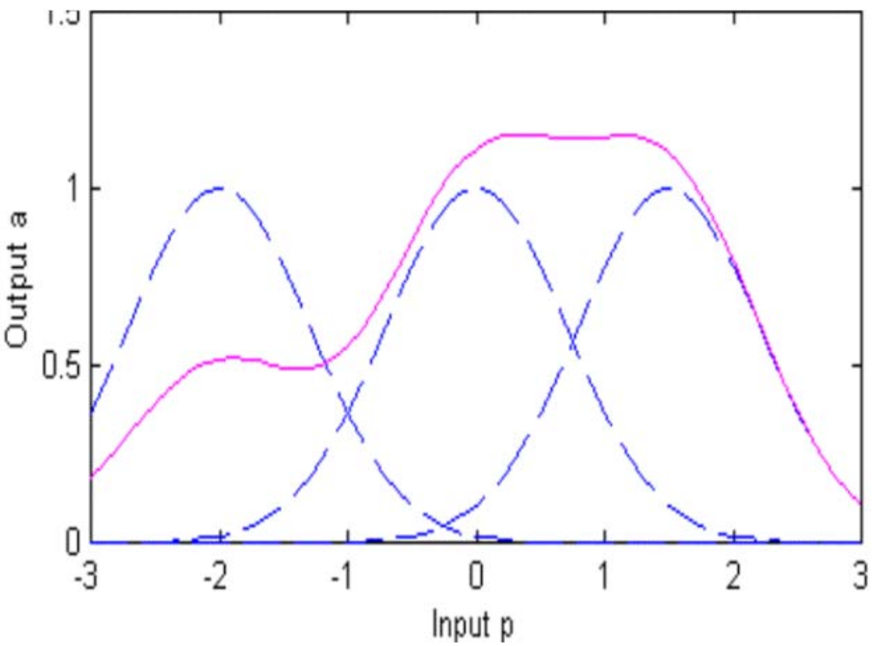
- $\phi_M(X) = \exp(\frac{-1}{2s_M}||X - cM||^2)$
- $\phi_{M-1}(X) = \exp(\frac{-1}{2s_{M-1}}||X - c_{M-1}||^2)$

…. ….　　…　　….　　……..　　..

……..　　……　　……　　…..　　…..

- $\phi_2(X)$ = $\exp(\frac{-1}{2s_2}||X - c_2||^2)$
- $\phi_1(X)$ = $\exp(\frac{-1}{2s_2}||X - c_1||^2)$
- $\phi_0(X)$ = 1

Three RBFs (blue) form $f(x)$ (pink)



$$f(X) = \beta_M\phi_M(X) + \beta_{M\,\_\,1}\phi_{M-1}(X) \ldots \ldots . + \beta_2\phi_2(X) + \beta_1\phi_1(X) + \beta_0$$

Gaussian Basis Function

$$f(X) = \beta_M\phi_M(X) + \beta_{M\ \_\ 1}\phi_{M-1}(X) \ldots\ldots.. + \beta_2\phi_2(X) + \beta_1\phi_1(X) + \beta_0$$

$$= W^T\phi\ (X)$$

where ,

$$W = \begin{bmatrix} \beta_M \\ \beta_{M-1} \\ - \\ - \\ - \\ \beta_1 \\ \beta_0 \end{bmatrix} \text{ and } \phi\ (X) = \begin{bmatrix} \phi_M(X) \\ \phi_{M-1}(X) \\ - \\ - \\ - \\ \phi_1(X) \\ \phi_0(X) \end{bmatrix}$$

For given Training Set T = { (x_1,y_1), (x_2,y_2),….., (x_N,y_N)}, we need to solve

$$Min_{(\beta_1,\beta_0)} \quad J(W) = \quad \sum_{i=1}^{N} (yi-(\; W^T \phi \; (X_i) \;))^2 \qquad .....(1)$$

If we assume

$$A = \begin{bmatrix} \phi_0(X_1), \; \phi_1 \, (X_1), \dots \dots \dots \dots, \phi_M(X_1) \\ \phi_0(X_2), \; \phi_1 \, (X_2), \dots \dots \dots \dots, \phi_M(X_2) \\ \\ \\ \phi_0(X_N), \; \phi_1 \, (X_N), \dots \dots \dots \dots, \phi_M(X_N) \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ \\ \\ y_N \end{bmatrix}$$

and u = W= $\begin{bmatrix} \beta_0 \\ \beta_1 \\ \\ \\ \\ \\ \\ \\ \beta_M \end{bmatrix}$

$$\boxed{u = W = (A^T A)^{-1} \; A^T Y}$$

For given Training Set T = { (x_1,y_1), (x_2,y_2),....., (x_N,y_N)}, we need to solve

$$Min_{(\beta_1,\beta_0)} \quad J(W) = \quad \sum_{i=1}^{N} (yi-(\ W^T\phi\ (X_i)\ ))^2 \qquad .....(1)$$

If we assume

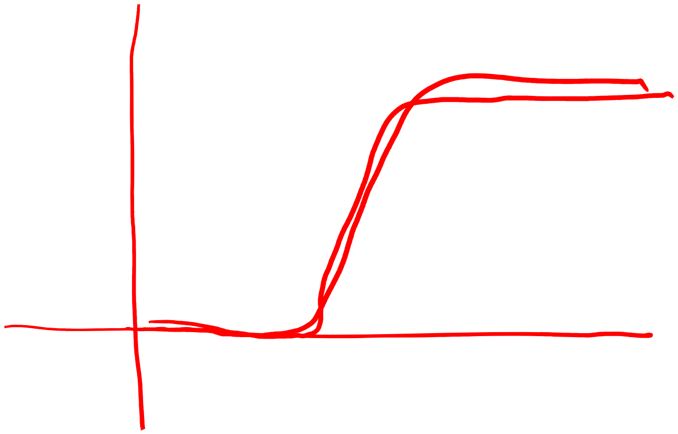
$$A = \begin{bmatrix} \phi_0(X_1),\ \ \phi_1\,(X_1), \dots \dots \dots \dots, \phi_M(X_1) \\ \phi_0(X_2),\ \ \phi_1\,(X_2), \dots \dots \dots \dots, \phi_M(X_2) \\ \\ \\ \phi_0(X_N),\ \ \phi_1\,(X_N), \dots \dots \dots \dots, \phi_M(X_N) \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ \\ \\ y_N \end{bmatrix} \qquad \text{and u = W=} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \\ \\ \\ \\ \\ \\ \\ \beta_M \end{bmatrix}$$

$$\boxed{u = W= (A^TA)^{-1}\ \ A^TY}$$

Sigmoidal function

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

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



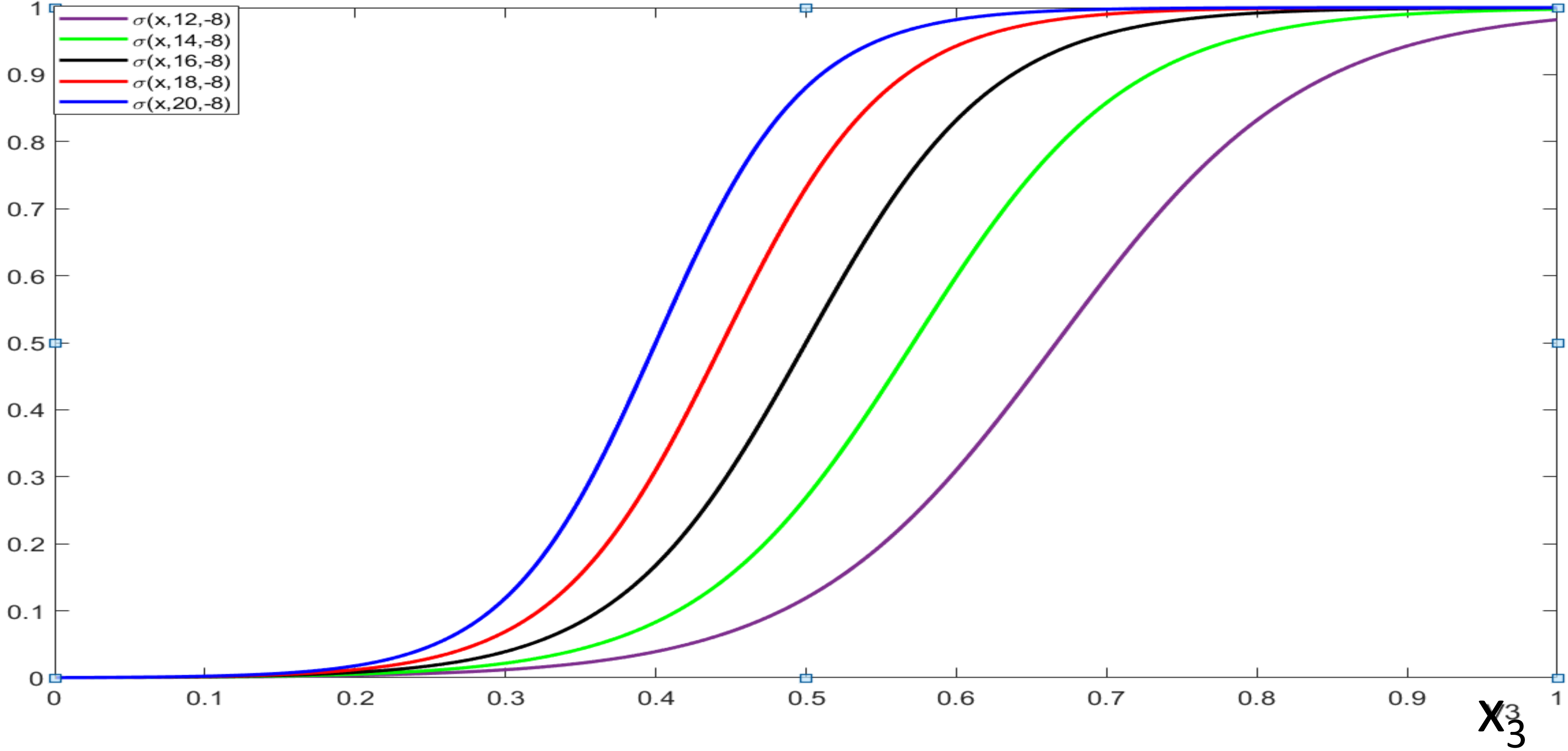
$$\sigma(x, \beta_1, \beta_0) = \frac{1}{1+e^{-(\beta_1 x+\beta_0)}} = \frac{e^{\beta_1 x+\beta_0}}{1+e^{\beta_1 x+\beta_0}}$$

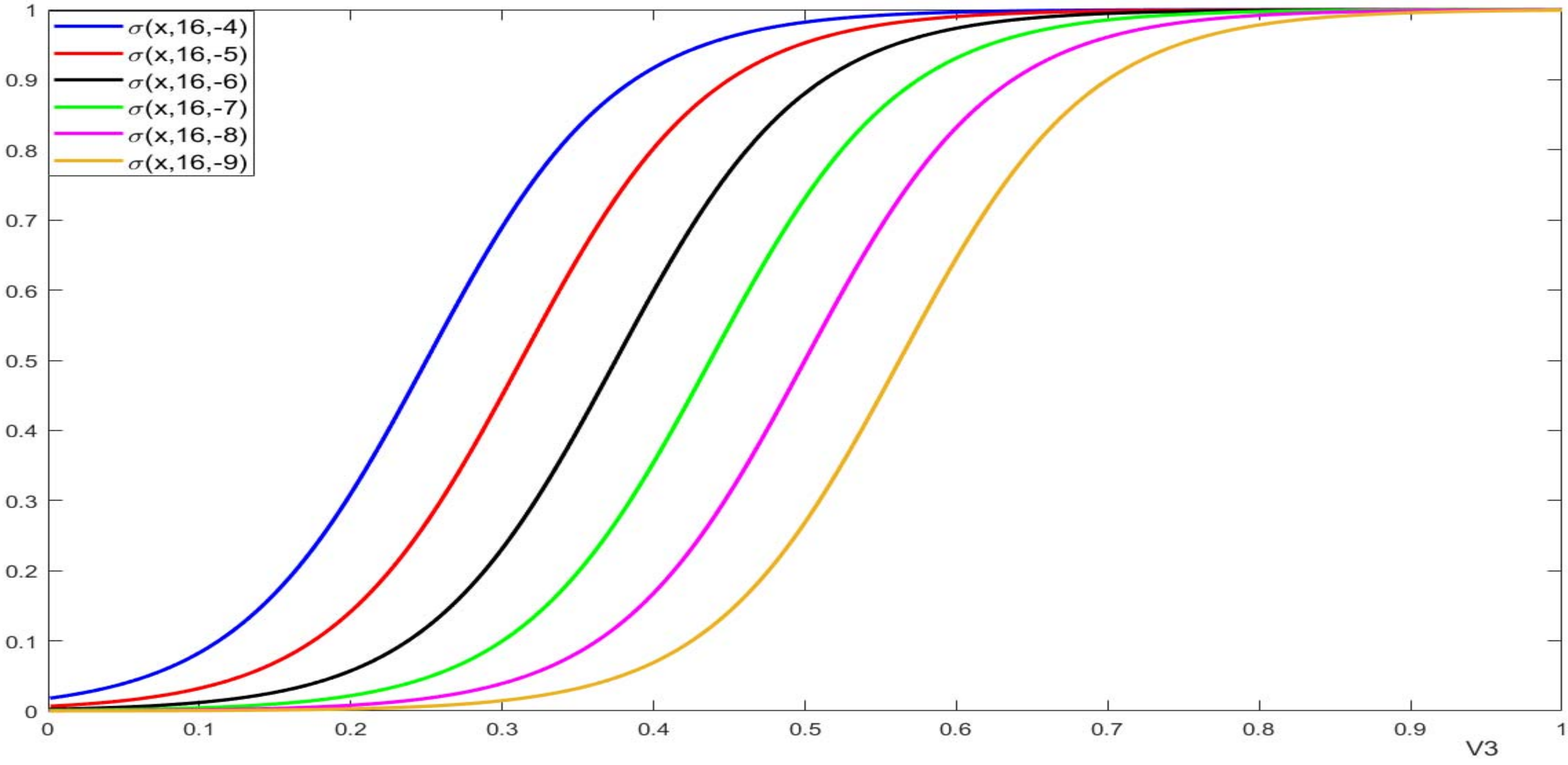
$$\sigma'(x) = \sigma\cancel{(x)}\,\sigma(1-x)$$

$$\sigma(x)\,(1-\sigma(x))$$

$\sigma(x, \beta, \beta_0)$

(




σ(x,16,-4)
σ(x,16,-5)
σ(x,16,-6)
σ(x,16,-7)
σ(x,16,-8)
σ(x,16,-9)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
V3

Sigmoidal Function

$$\sigma(x, c_M, dM) = \boxed{\frac{1}{1+e^{-\boxed{(c_M x+d_M)}}} = \frac{e^{(c_M x+d_M)}}{1+e^{(c_M x+d_M)}}}.$$

$$\sigma(x, \underset{M-1}{c_1}, \underset{M-1}{d_1}) = \frac{1}{1+e^{-(c_{M\_1} x+d_{M-1})}} = \frac{e^{(c_{M\_1} x+d_{M-1})}}{1+e^{(c_{M\_1} x+d_{M-1})}}$$

… … … … … … … … … … … … … … … … … …. .

$$\sigma(x, c_1, d_1) = \frac{1}{1+e^{-(c_1 x+d_1)}} = \frac{e^{(c_1 x+d_1)}}{1+e^{(c_1 x+d_1)}}$$

$$\sigma(x, c_0, d_0) = \frac{1}{1+e^{-(c_0 x+d_0)}} = \frac{e^{(c_0 x+d_0)}}{1+e^{(c_0 x+d_0)}} = 1.$$

Sigmoidal Function

$c_m^T x + dm$

$$\phi_M(X) = \sigma(X, c_M, dM) = \frac{1}{1+e^{\underline{-(c^T{}_M X+d_M)}}} = \frac{e^{(c^T{}_M X+d_M)}}{1+e^{(c_M X+d_M)}}.$$

$$\phi_{M-1}(X) = \sigma(X, c_1, d_1) = \frac{1}{1+e^{\underline{-(c^T{}_{M\_1} X+d_{M-1})}}} = \frac{e^{(c^T{}_{M\_1} X+d_{M-1})}}{1+e^{(c^T{}_{M\_1} X+d_{M-1})}}$$

$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.$$

$$\phi_1(X) = \sigma(X, c_1, d_1) = \frac{1}{1+e^{-(c^T{}_1 X+d_1)}} = \frac{e^{(c^T{}_1 X+d_1)}}{1+e^{(c^T{}_1 X+d_1)}}$$

$$\phi_0(X) = \sigma(X, c_0, d_0) = \frac{1}{1+e^{-(c^T{}_0 X+d_0)}} = \frac{e^{(c^T{}_0 x+d_0)}}{1+e^{(c^T{}_0 X+d_0)}} = 1.$$

$$f(X) = \beta_M \phi_M(X) + \beta_{M\_1} \phi_{M-1}(X) \dots \dots\dots + \beta_2 \phi_2(X) + \beta_1 \phi_1(X) + \beta_0$$

For given Training Set T = { (x_1,y_1), (x_2,y_2),….., (x_N,y_N)}, we need to solve

$$Min_{(\beta_1,\beta_0)\; W} \quad J(W) = \sum_{i=1}^{N} (y_i - (W^T \phi (X_i)))^2 \qquad \text{.....(1)}$$

If we assume

$$A = \begin{bmatrix} \phi_0(X_1), & \phi_1(X_1), \ldots\ldots\ldots\ldots, \phi_M(X_1) \\ \phi_0(X_2), & \phi_1(X_2), \ldots\ldots\ldots\ldots, \phi_M(X_2) \\ & \\ \phi_0(X_N), & \phi_1(X_N), \ldots\ldots\ldots\ldots, \phi_M(X_N) \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ \\ \\ y_N \end{bmatrix} \qquad \text{and u = W=} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \\ \\ \\ \\ \\ \beta_M \end{bmatrix}$$

$$\boxed{u = W = (A^T A)^{-1} \; A^T Y}$$

Credit Card Dataset

Income	Limit	Rating	Cards	Age	Balance
14.891	3606	283	2	34	333
106.025	6645	483	3	82	903
104.593	7075	514	4	71	580
148.924	9504	681	3	36	964
55.882	4897	357	2	68	331
80.18	8047	569	4	77	1151
20.996	3388	259	2	37	203
71.408	7114	512	2	87	872
15.125	3300	266	5	66	279
71.061	6819	491	3	41	1350
63.095	8117	589	4	30	1407

Evaluation

Income (hundred thousand dollar)	Balance (thousand dollar)
0.550798	5.651202
0.708148	7.321263
0.290905	5.167304
0.510828	5.609367
0.892947	9.406379
0.896293	9.379439
0.125585	2.734997
0.207243	4.876649
0.051467	3.584138
0.44081	5.437239

Training data

Income (x) (hundred thousand dollar)	Balance (y) (thousand dollar)
0.96703	9.675083
0.547232	6.293266
0.972684	9.730614
0.714816	7.474346
0.697729	7.342933
0.216089	4.619033
0.976274	9.765597
0.00623	4.012784
0.252982	4.762698
0.434792	5.626166
0.779383	7.989045
0.197685	4.552625
0.862993	8.705537
0.983401	9.835217
0.163842	4.43522
0.597334	6.622444
0.008986	4.01882
0.386571	5.37153
0.04416	4.096522
0.956653	9.574695

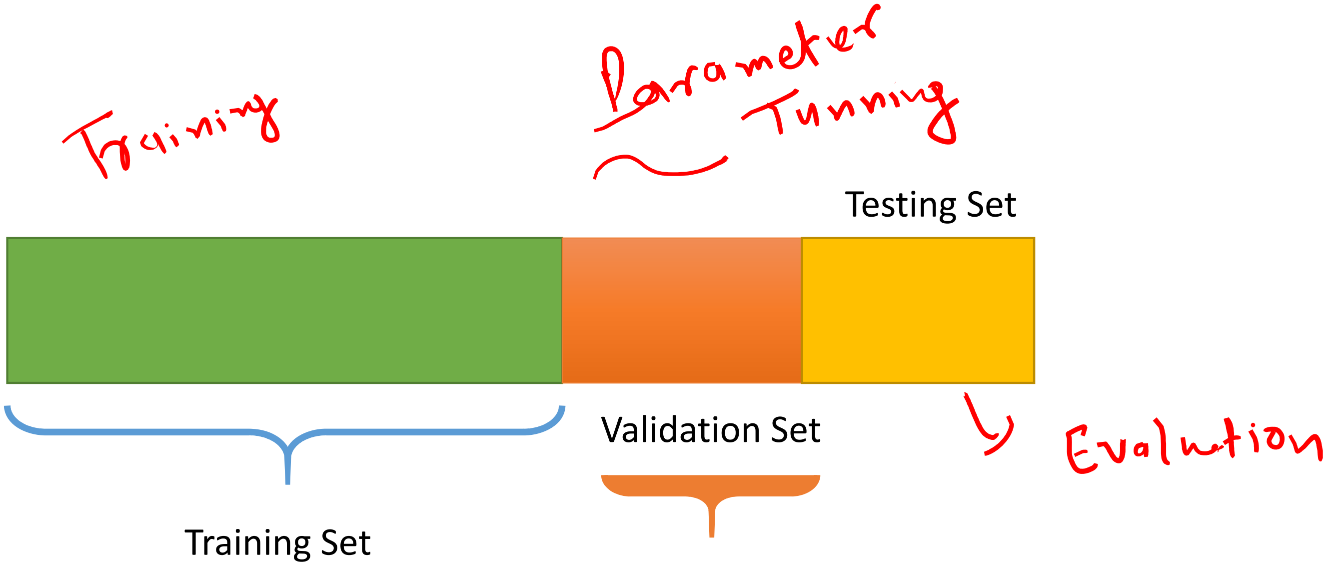
Testing Data

Income (x) (thousand dollar)	Balance (y) (thousand dollar)	Estimated f(x) (thousand dollar)
0.96703	9.675083	9.41205399
0.547232	6.293266	6.47467789
0.972684	9.730614	9.45161938
0.714816	7.474346	7.64728226
0.697729	7.342933	7.5277212
0.216089	4.619033	4.15763074
0.976274	9.765597	9.47673972
0.00623	4.012784	2.68921946
0.252982	4.762698	4.41577473
0.434792	5.626166	5.68791617
0.779383	7.989045	8.09906511
0.197685	4.552625	4.02885271
0.862993	8.705537	8.6840969
0.983401	9.835217	9.52660279
0.163842	4.43522	3.79205019
0.597334	6.622444	6.8252457
0.008986	4.01882	2.70850244
0.386571	5.37153	5.35051307
0.04416	4.096522	2.95461902
0.956653	9.574695	9.33944573

Evaluation Metrics

- $SSE = \sum_{i=1}^{k}(yi - f(xi))^2$
- $NMSE = \frac{\sum_{i=1}^{k}(yi-f(xi))^2}{\sum_{i=1}^{k}(yi-\bar{y})^2}$
- $RMSE = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(yi - f(xi))^2}$
- $MAE = \frac{1}{k}\sum_{i=1}^{k}|(yi - f(xi))|$
- $MAPE = \frac{1}{k}\sum_{i=1}^{k}\left(\frac{(yi-f(x_i))}{y_i} \times 100\right)$
- $R^2 = \frac{\sum_{i=1}^{k}(f(x_i)-\overline{f(x)})^2}{\sum_{i=1}^{k}(yi-\bar{y})^2}$

Cross-validation



Training

Parameter Tunning

Testing Set

Validation Set

Evaluation

Training Set

k-fold Cross Validation

Income	Limit	Rating	Cards	Age	Balance
14.891	3606	283	2	34	333
106.025	6645	483	3	82	903
104.593	7075	514	4	71	580
148.924	9504	681	3	36	964
55.882	4897	357	2	68	331
80.18	8047	569	4	77	1151
20.996	3388	259	2	37	203
71.408	7114	512	2	87	872
15.125	3300	266	5	66	279
71.061	6819	491	3	41	1350
63.095	8117	589	4	30	1407

← Testing Set

← Training Set

$$\text{Test RMSE}_1 = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(yi - f_1(xi))^2} = 0.9426$$

k-fold Cross Validation

Income	Limit	Rating	Cards	Age	Balance
14.891	3606	283	2	34	333
106.025	6645	483	3	82	903
104.593	7075	514	4	71	580
148.924	9504	681	3	36	964
55.882	4897	357	2	68	331
80.18	8047	569	4	77	1151
20.996	3388	259	2	37	203
71.408	7114	512	2	87	872
15.125	3300	266	5	66	279
71.061	6819	491	3	41	1350
63.095	8117	589	4	30	1407

← Testing Set

Test RMSE_2= $\sqrt{\frac{1}{k}\sum_{i=1}^{k}(yi - f_2(xi))^2} = 0.8725$

k-fold Cross Validation

Income	Limit	Rating	Cards	Age	Balance
14.891	3606	283	2	34	333
106.025	6645	483	3	82	903
104.593	7075	514	4	71	580
148.924	9504	681	3	36	964
55.882	4897	357	2	68	331
80.18	8047	569	4	77	1151
20.996	3388	259	2	37	203
71.408	7114	512	2	87	872
15.125	3300	266	5	66	279
71.061	6819	491	3	41	1350
63.095	8117	589	4	30	1407

← Testing Set

Test RMSE$_5$= $\sqrt{\frac{1}{k}\sum_{i=1}^{k}(yi - f_5(xi))^2} = 0.8721$

k-fold Cross Validation

Income	Limit	Rating	Cards	Age	Balance
14.891	3606	283	2	34	333
106.025	6645	483	3	82	903
104.593	7075	514	4	71	580
148.924	9504	681	3	36	964
55.882	4897	357	2	68	331
80.18	8047	569	4	77	1151
20.996	3388	259	2	37	203
71.408	7114	512	2	87	872
15.125	3300	266	5	66	279
71.061	6819	491	3	41	1350
63.095	8117	589	4	30	1407

← Testing Set

Test RMSE= 0.8354 ± 0.123

Artifical

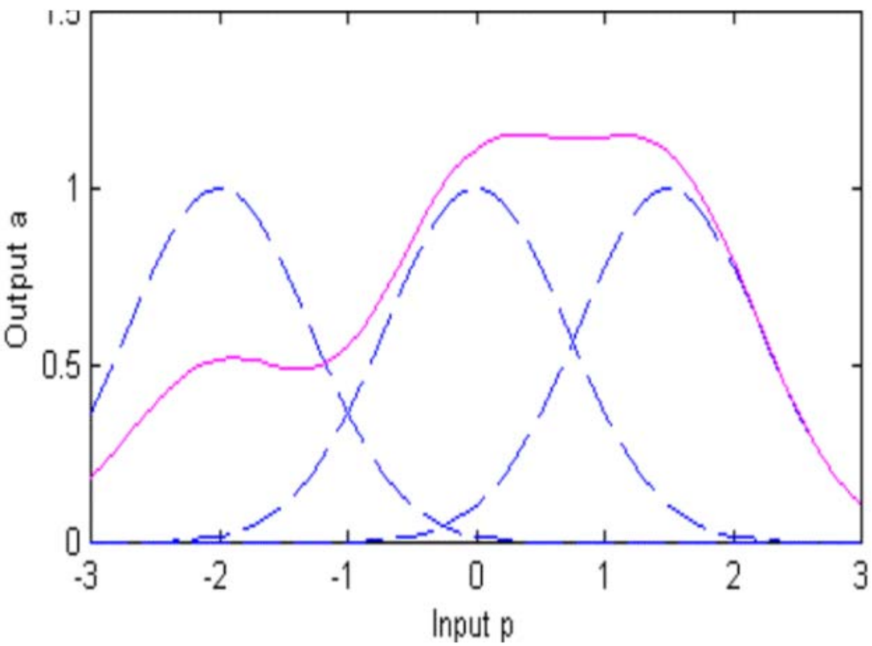
- $\phi_M(X) = \exp(\frac{-1}{2s_M}||X - cM||^2)$
- $\phi_{M-1}(X) = \exp(\frac{-1}{2s_{M-1}}||X - c_{M-1}||^2)$

…. …. … …. …….. ..

…….. …… …… ….. …..

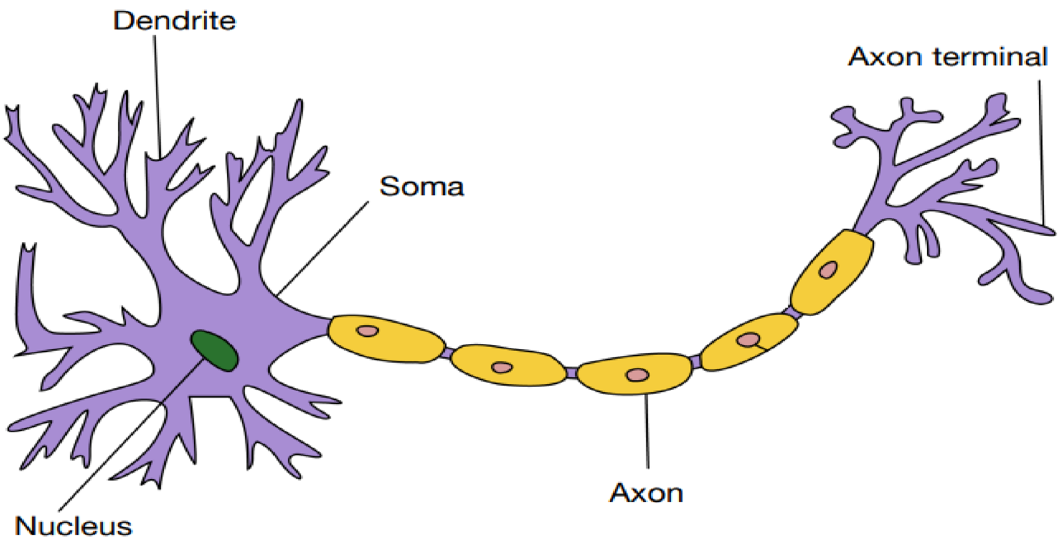
- $\phi_2(X)$ = $\exp(\frac{-1}{2s_2}||X - c_2||^2)$
- $\phi_1(X)$ = $\exp(\frac{-1}{2s_2}||X - c_1||^2)$
- $\phi_0(X)$ = 1

Three RBFs (blue) form $f(\mathrm{x})$ (pink)



$$f(X) = \beta_M\phi_M(X) + \beta_{M\ \_\ 1}\phi_{M-1}(X) \ldots \ldots . . + \beta_2\phi_2(X) + \beta_1\phi_1(X) + \beta_0$$

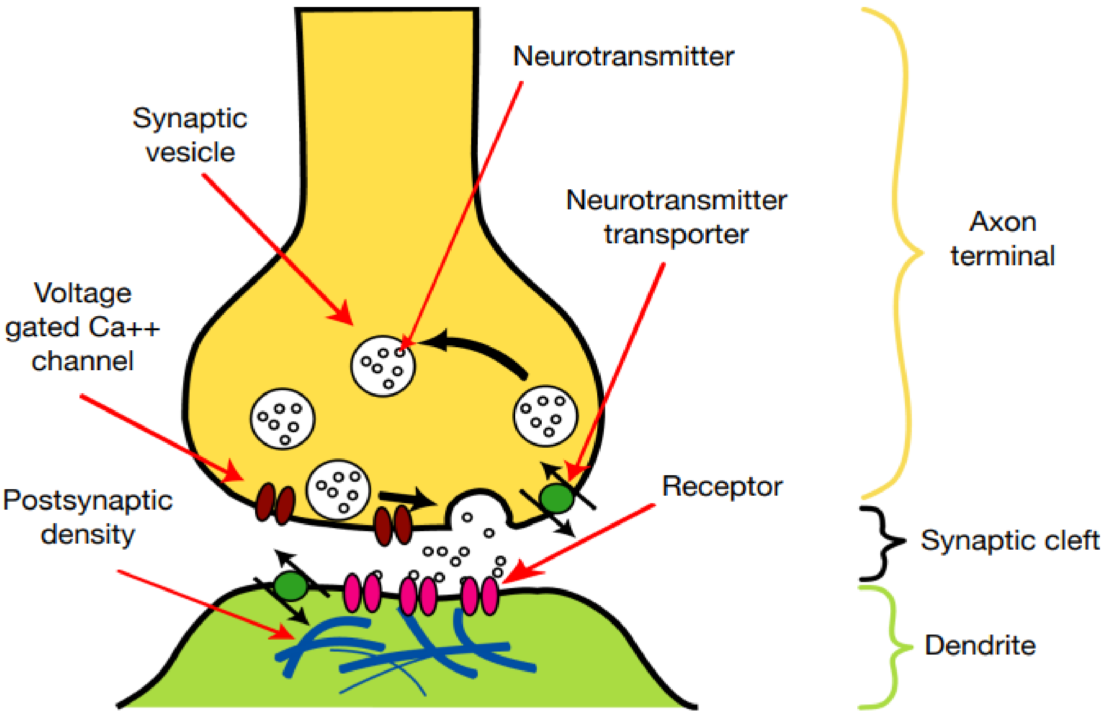
- The human brain is made up of about 100 billion neurons



- Neurons receive electric signals at the dendrites and send them to the axon

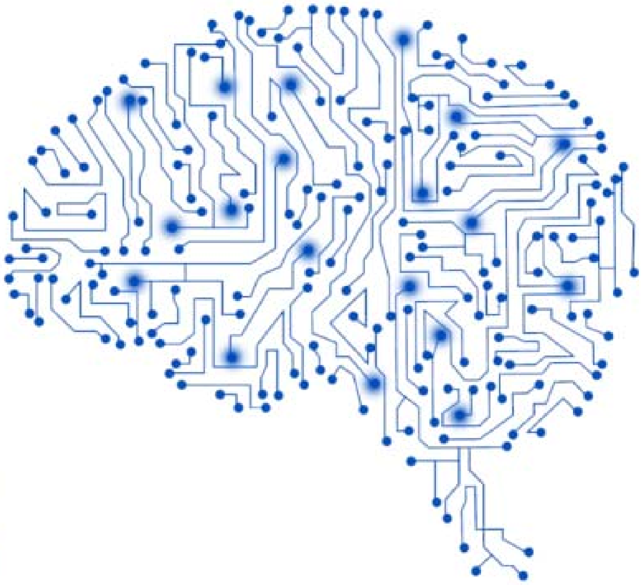
http://mt-class.org/jhu/slides/lecture-nn-intro.pdf

- The axon of the neuron is connected to the dendrites of many other neurons

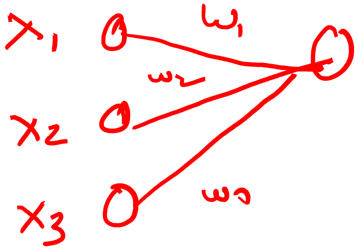


http://mt-class.org/jhu/slides/lecture-nn-intro.pdf

- Similarities
  - Neurons, connections between neurons
  - Learning = change of connections, not change of neurons
  - Massive parallel processing



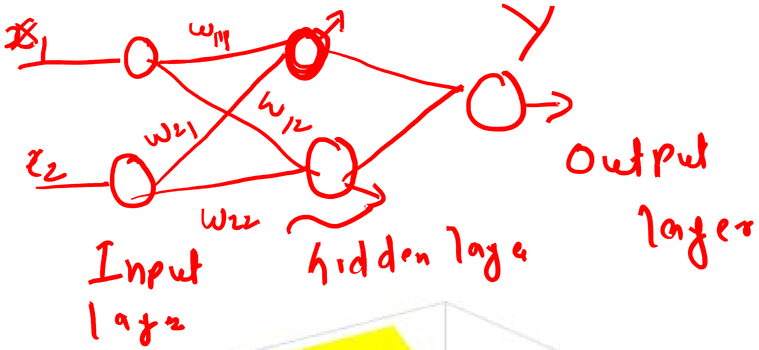
- But artificial neural networks are much simpler
  - computation within neuron vastly simplified
  - discrete time steps
  - typically some form of supervised learning with massive number of stimuli



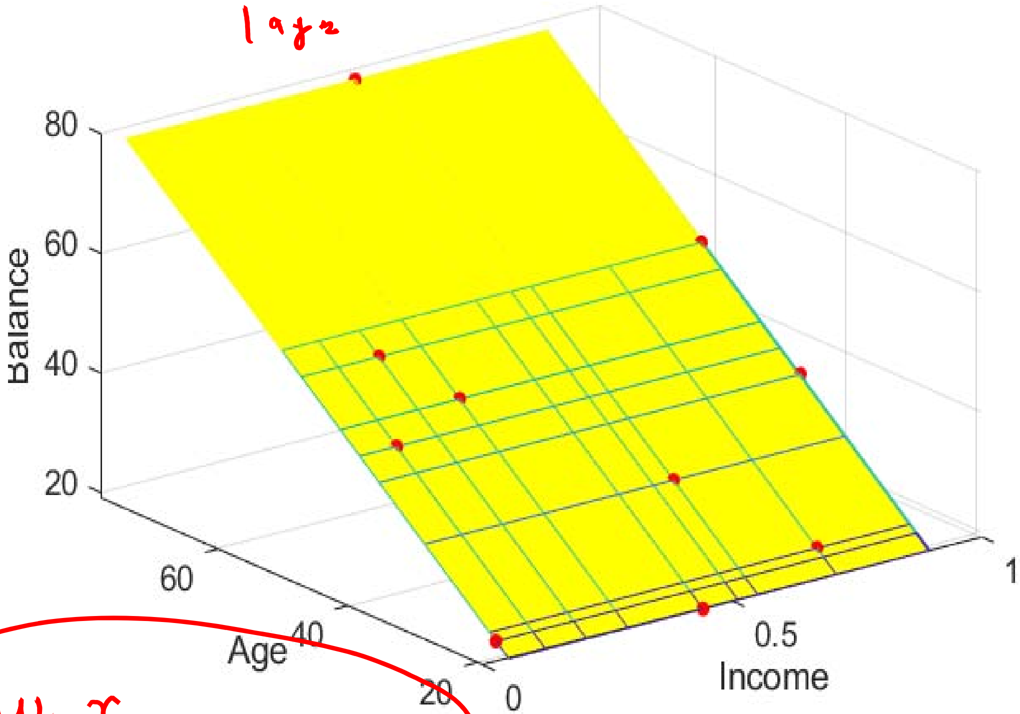
$x_1 w_1 + x_2 w_2 + x_3 w_3$

http://mt-class.org/jhu/slides/lecture-nn-intro.pdf

Multiple Linear Regression model

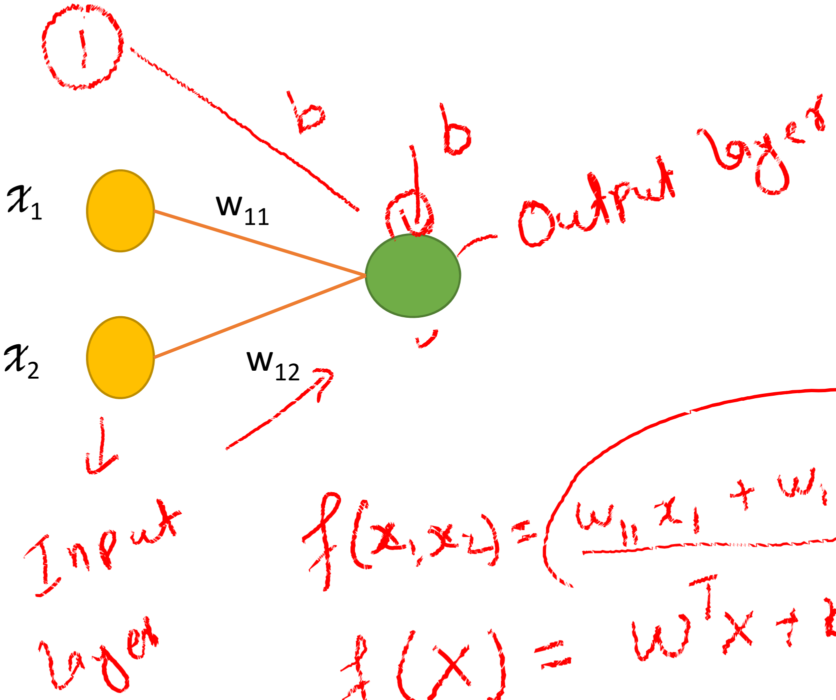


x_1 Age	x_2 Income (hundred thousand dollar)	y Balance (thousand dollar)
32	0.550798	5.651202
22	0.708148	7.321263
45	0.290905	5.167304
78	0.510828	5.609367
54	0.892947	9.406379
39	0.896293	9.379439
42	0.125585	2.734997
51	0.207243	4.876649
21	0.051467	3.584138
19	0.44081	5.437239



$w_{11}x_1 + w_{21}x_2 +$

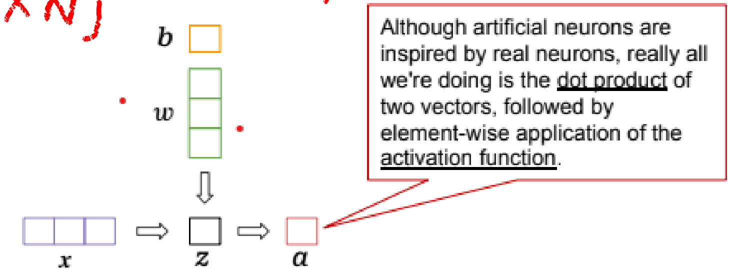
Artificial Neural Network



$f(x_1)$ Y_1

$f(x_2)$ Y_2

$f(x_N)$ Y_N



Although artificial neurons are inspired by real neurons, really all we're doing is the dot product of two vectors, followed by element-wise application of the activation function.

$$f(x_1, x_2) = w_{11}x_1 + w_{12}x_2 + b - Y_1$$

$$f(X) = w^T x + b$$

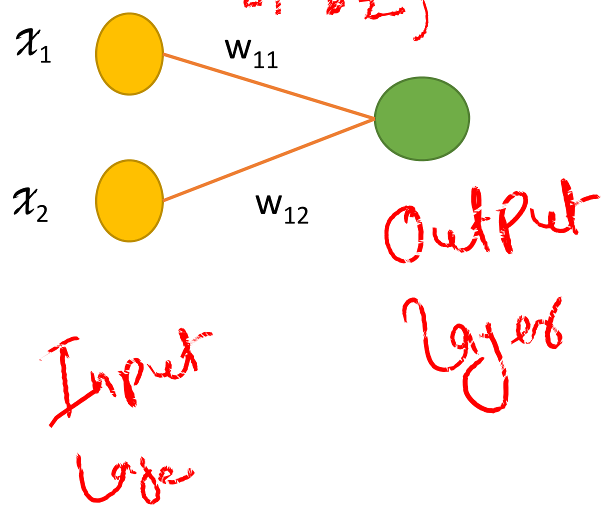
$$f(x_2)$$

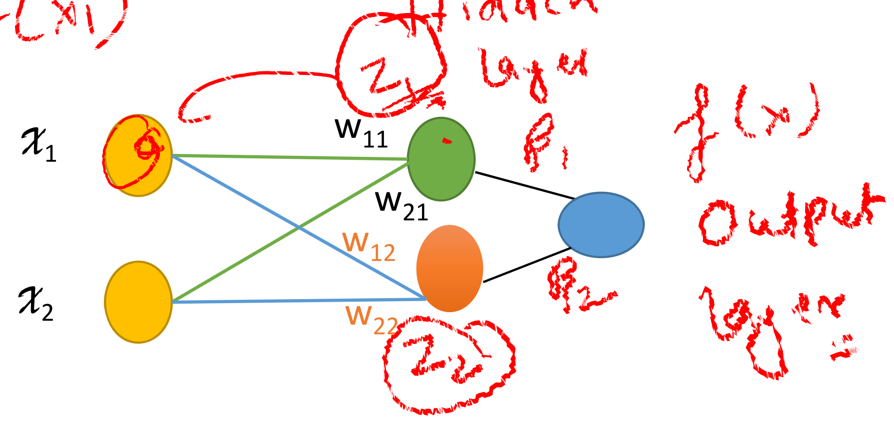
$$f(x)_{\theta}$$

$$\frac{1}{N}\sum_{i=1}(y_i - f(x_i))^2$$

Artificial Neural Network

$$\min_{(\beta_1,\beta_2,w_{12},w_{21},w_{11},w_{22},b_1,b_2)} J \quad \sum_{i=1}^{N}\Big(y_i - \underbrace{\big[\beta_1\,\sigma(w_{11}x_1+w_{21}x_2+b_1)+\beta_2\,\sigma(w_{21}x_{1}+w_{22}x_2+b_2)}_{f(x_i)}\Big)^2$$





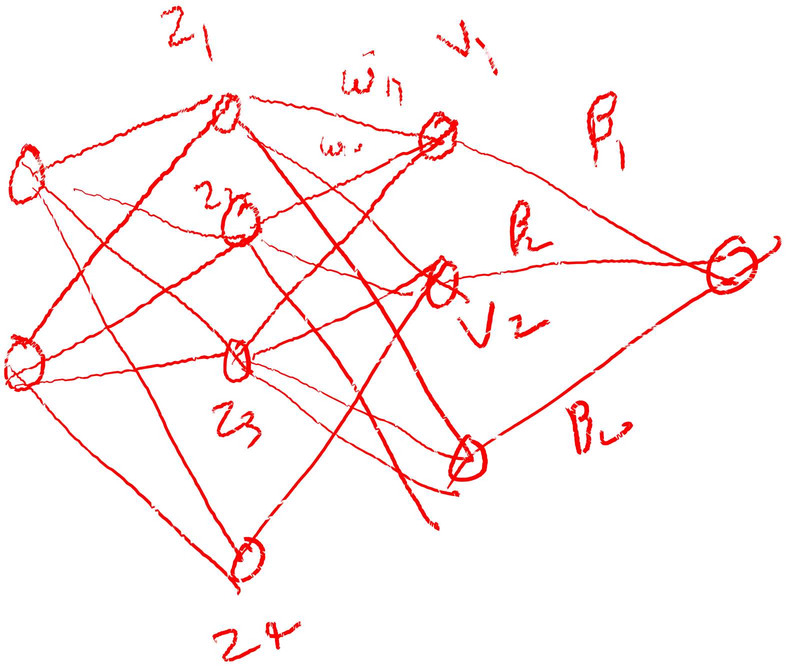
$$Z_1 = \sigma(w_{11}x_1 + w_{21}x_2 + b_1)$$

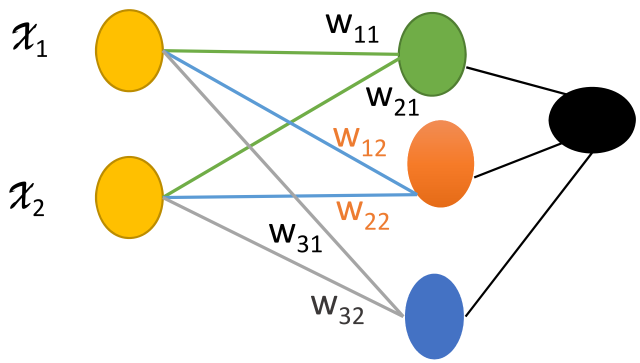
$$Z_2 = \sigma(w_{12}x_1 + w_{22}x_2 + b_2)$$

$$\frac{\partial J}{\partial w_{12}} = \frac{\partial J}{\partial Z}\cdot\frac{\partial Z}{\partial w_{12}}$$

$$f(x) = \beta_1 Z_1 + \beta_2 Z_2$$

Artificial Neural Network





$$\underline{\sigma\left(\bar{w}_{11} z_1 + \bar{w}_{12} z_2 + \bar{w}_{13} z_3 + b_1\right)} +$$

Introduction: Activation Functions

Many activation functions have been proposed, including:

- ***linear*** activation function: $g(z) = z$
- ***step*** activation function: $g(z) = \begin{cases} 0 & z < 0, \\ 1 & z \geq 0 \end{cases}$
- ***sigmoid*** activation function: $g(z) = \frac{1}{1 + e^{-z}}$
- ***<u>ReLU</u>*** activation function (ReLU stands for Rectified Linear Unit): $g(z) = \max(0, z)$
- ***tanh*** activation function (tanh is the hyperbolic tangent): $g(z) = \tanh z$

Apart from the ***linear*** activation function, these activation functions are non-linear, which is important to the power of neural networks.

Introduction: Activation Functions

Apart from the ***linear*** activation function, these activation functions are non-linear, which is important to the power of neural networks.

Activation functions and their derivatives -