Neural Network
Lecture 4

Jeong-Yean Yang

2020/10/22

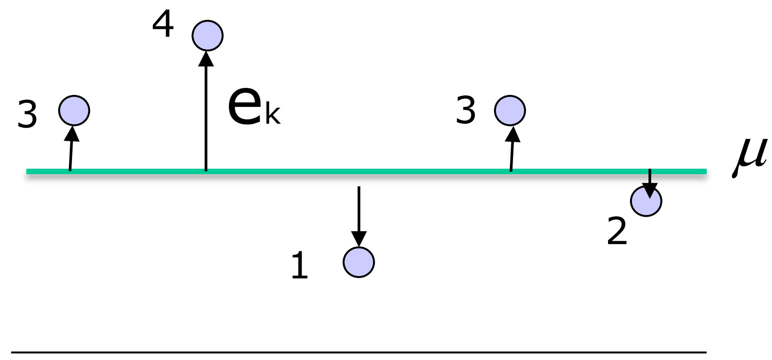
1 Neural Network with RBF

Mean and Variance

- Mean value

$$\mu = \frac{1}{N}\sum_{k=1}^{N} x_k$$

$$ex)\mu = \frac{3+4+1+3+2}{N=5} = 2.6$$



– Find the best value to minimize error.

$$J = \sum_{k=1}^{N} e_k^2 = \sum_{k=1}^{N} (x_k - \mu)^2$$

$$\frac{\partial J}{\partial \mu} = -2\sum_{k=1}^{N} (x_k - \mu) = 0$$

$$\sum_{k=1}^{N} x_k - \mu N = 0$$

$$\therefore \mu = \frac{1}{N}\sum_{k=1}^{N} x_k$$

Mean and Variance

- Variance

$$\sigma^2 = \frac{1}{N}\sum_{k=1}^{N}(x_k - \mu)^2 \qquad \sigma : \text{Standard Deviation}$$

- Oops, It is very similar to Cost function, J

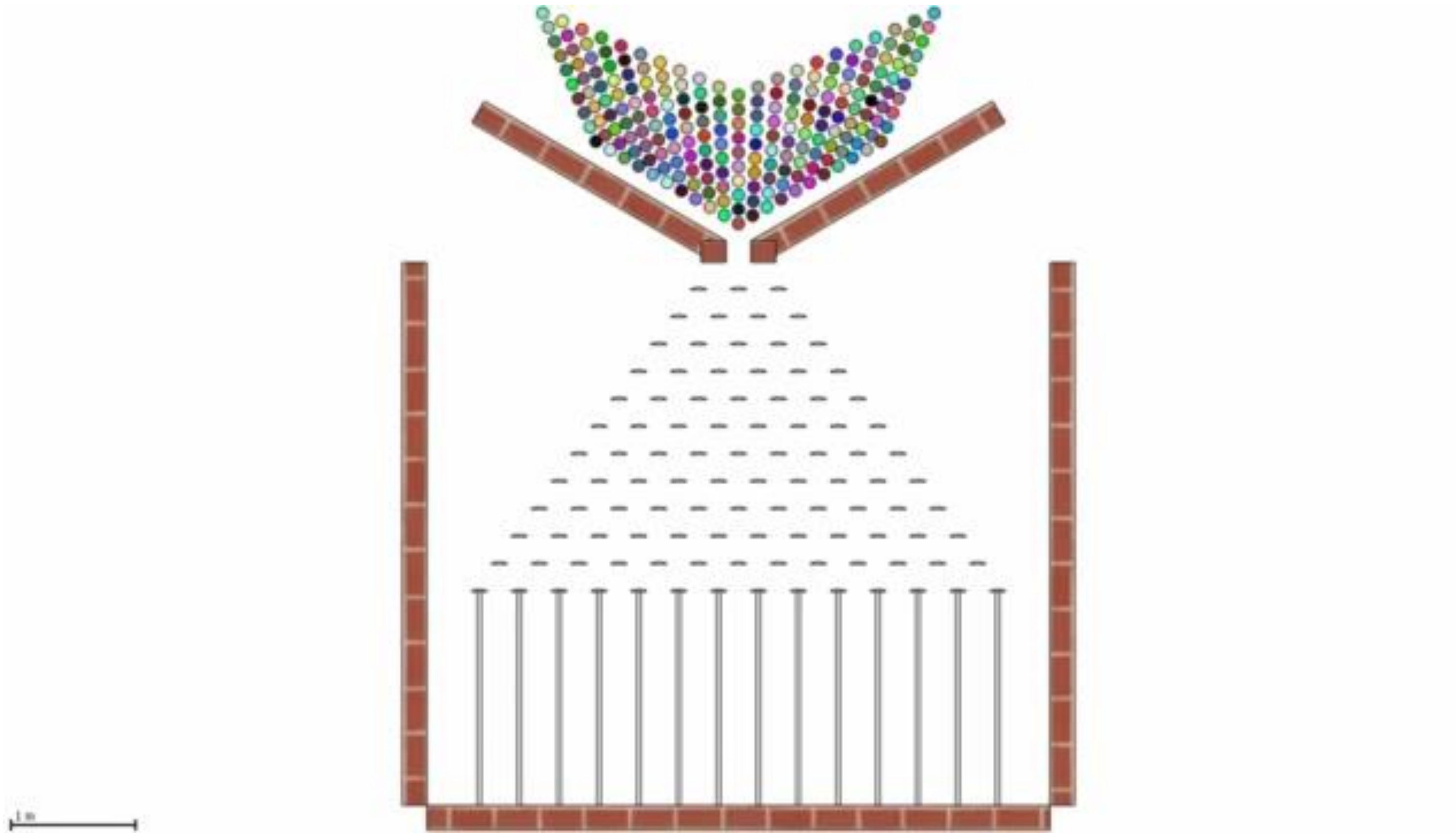
$$J = \sum_{k=1}^{N} e_k^2 = \sum_{k=1}^{N}(x_k - \mu)^2$$

In another form,

$$\therefore \sigma^2 = \frac{J}{N} = \frac{1}{N}\sum_{k=1}^{N} e_k^2 = \frac{1}{N}\sum_{k=1}^{N}(x_k - \mu)^2$$

  – Variance is the mean value of Cost function
  – Cost function: sum of squared errors

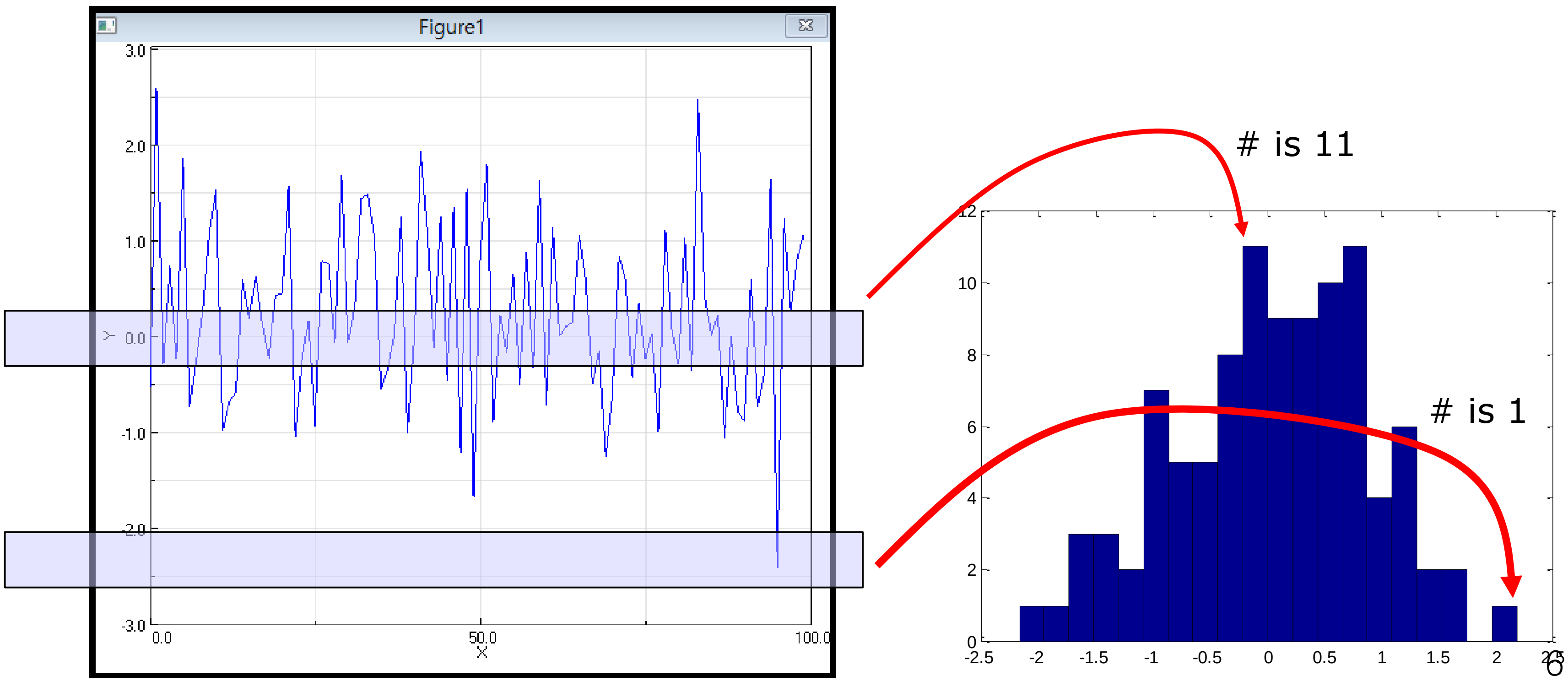
Gaussian Distribution



$$\mathrm{PDF}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2 \right),$$ Probability Density Function

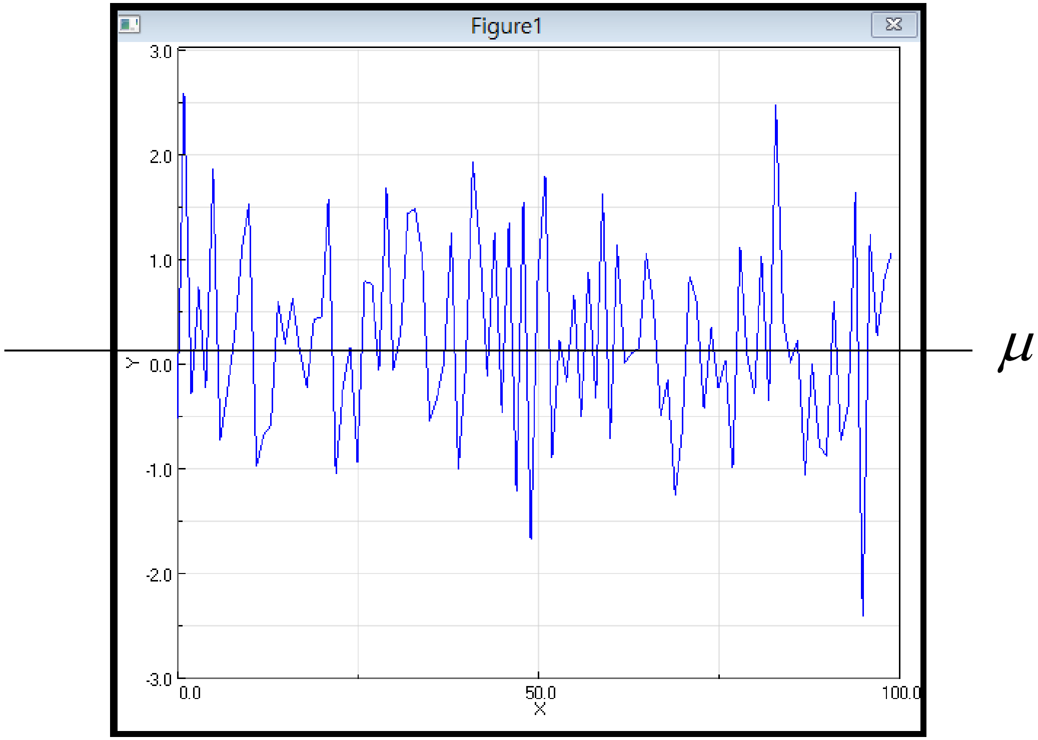
Gaussian Distribution Example

- randn with Matlab or Out Program
  - X=randn(100,1)
  - plot(X)



Gaussian Distribution with mean and variance

- Why Gaussian distribution?



In many cases, error is distributed in Gaussian distribution.

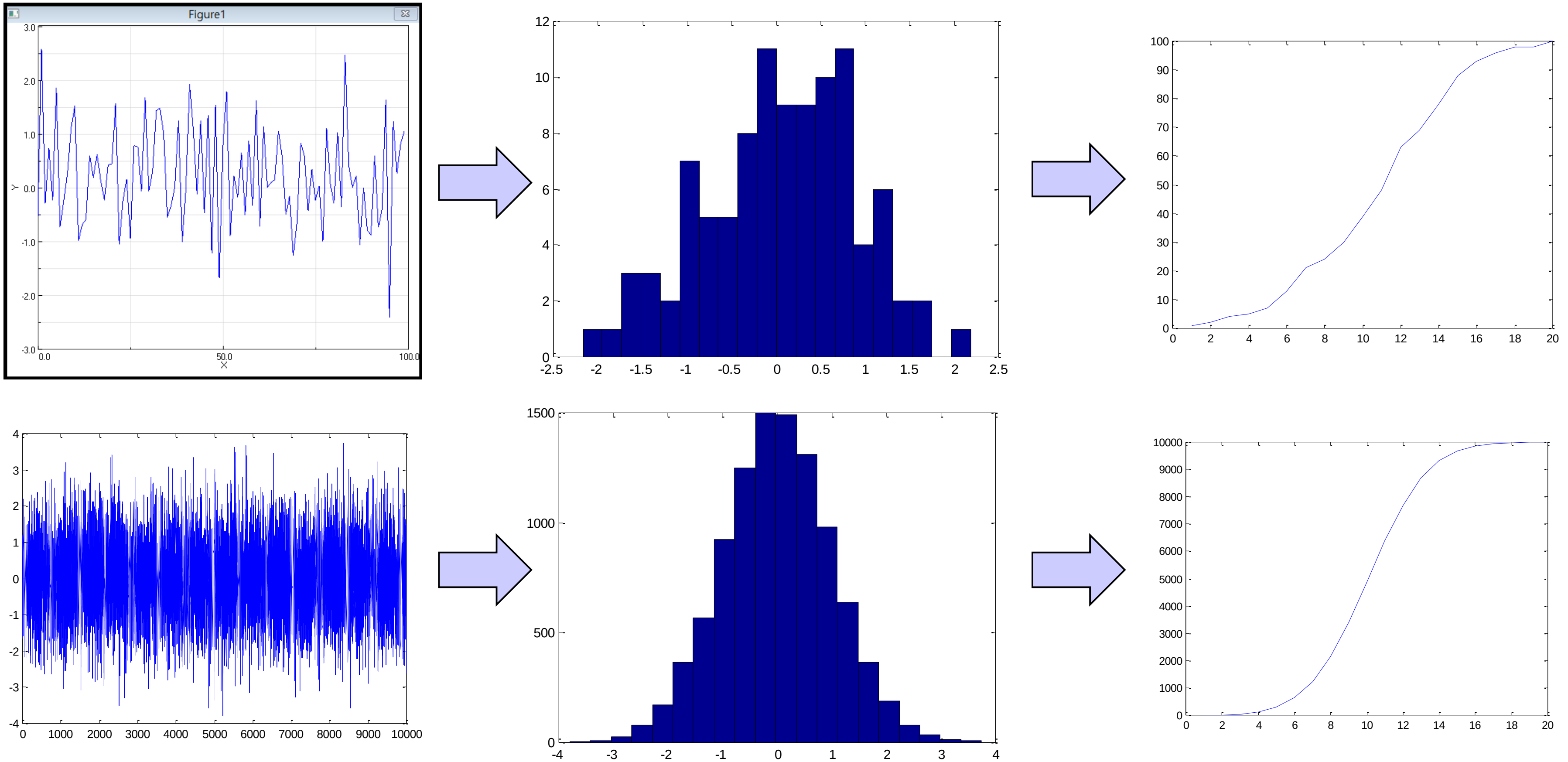
→ That is why Gaussian distribution is used for Nonlinear function in NN.

NN: Cumulative Gaussian function

$$\mathrm{Cum}(x) = \int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma} \right)^2 \right) dx$$

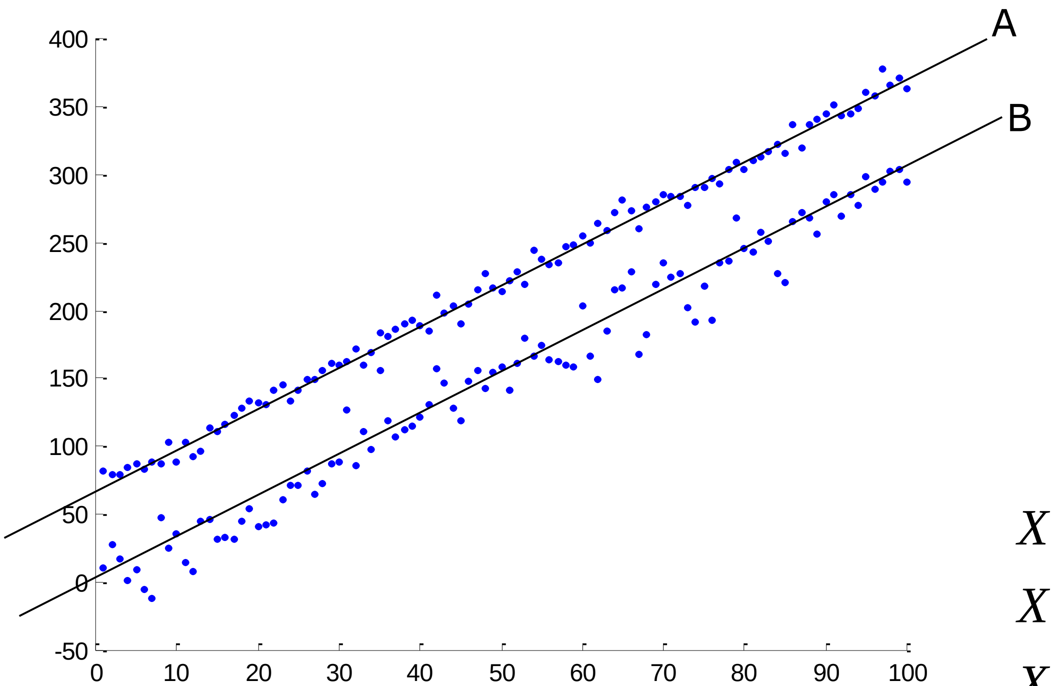
Similar to
Sigmoidal function

randn(100,1)



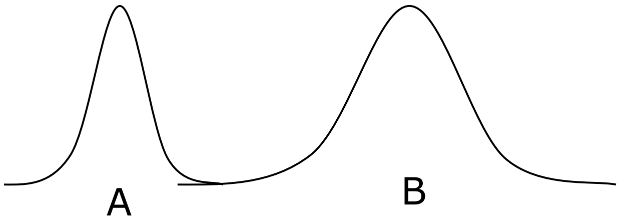
NN: Gaussian Function Sum

- Think line regression example



$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 \right)$$

$$x \sim N(\mu, \sigma)$$



$$X_A \sim N(\mu_1, \sigma_1)$$

$$X_B \sim N(\mu_2, \sigma_2)$$

$$X = X_A + X_B = N(\mu_1, \sigma_1) + N(\mu_2, \sigma_2)$$

Radial Basis Function (RBF)

- Sum of Gaussian function expresses any function.

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 \right)$$

$$x \sim N(\mu, \sigma)$$

- RBF is simplified Gaussian Function

$$\Phi(x) = \exp\left(-\left(\frac{x-\mu}{\sqrt{2}\sigma}\right)^2 \right) = \exp\left(-\left(\frac{x-w}{b}\right)^2 \right)$$

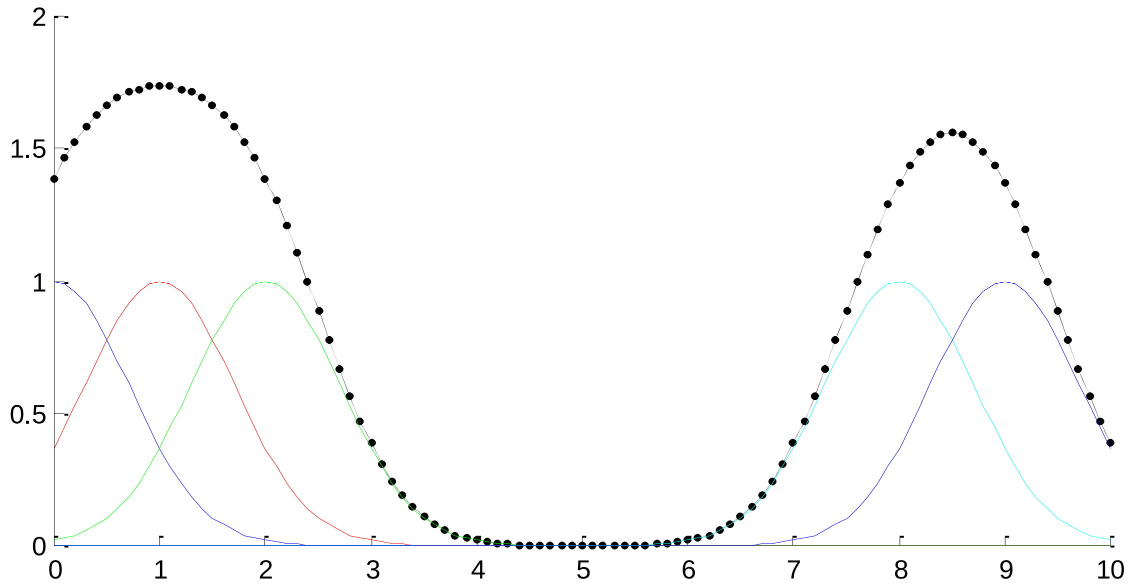
$$w, b = weight$$

$$ex)\ \Phi(x=\mu) = 1$$

RBF Superposition

- Sum of RBF can estimate a given data, D

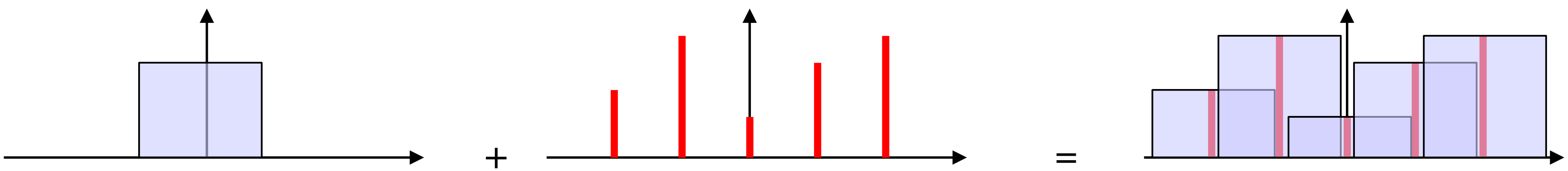
$$\hat{y}(x) = \sum_k \Phi_k(x) = \sum_k \exp\left(-\left(\frac{x - w_k}{b_k} \right)^2 \right)$$



b=1 and various w

Example: Kernel Function Superposition

- Window:
  - Sample with Window Size and build kernel value
- Dirchlet Window(Rectangular Window)
  - [dir-i-**kley]**

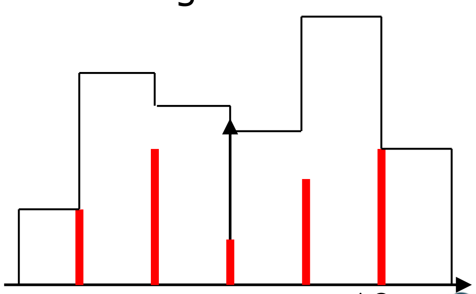


Window, w

Signal, s

Windowed signal

=



- **RBF Kernel is thought as Window**

RBF Feed Forward Method

- Feed forward method with data,

$$\hat{y}(x) = \sum_{k \in D} y_k \Phi_k(x) = \sum_k y_k \exp\left(-\left(\frac{x - x_k}{b} \right)^2 \right)$$

$$b = const$$

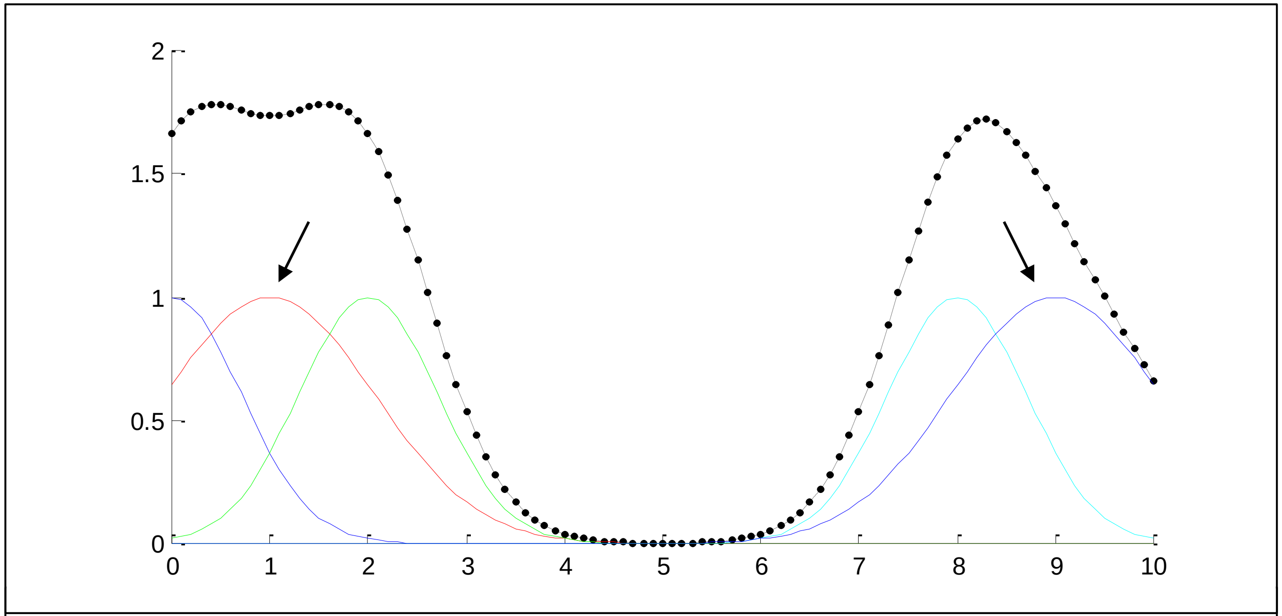
- Example) Measure the distance

D=(deg, distance)

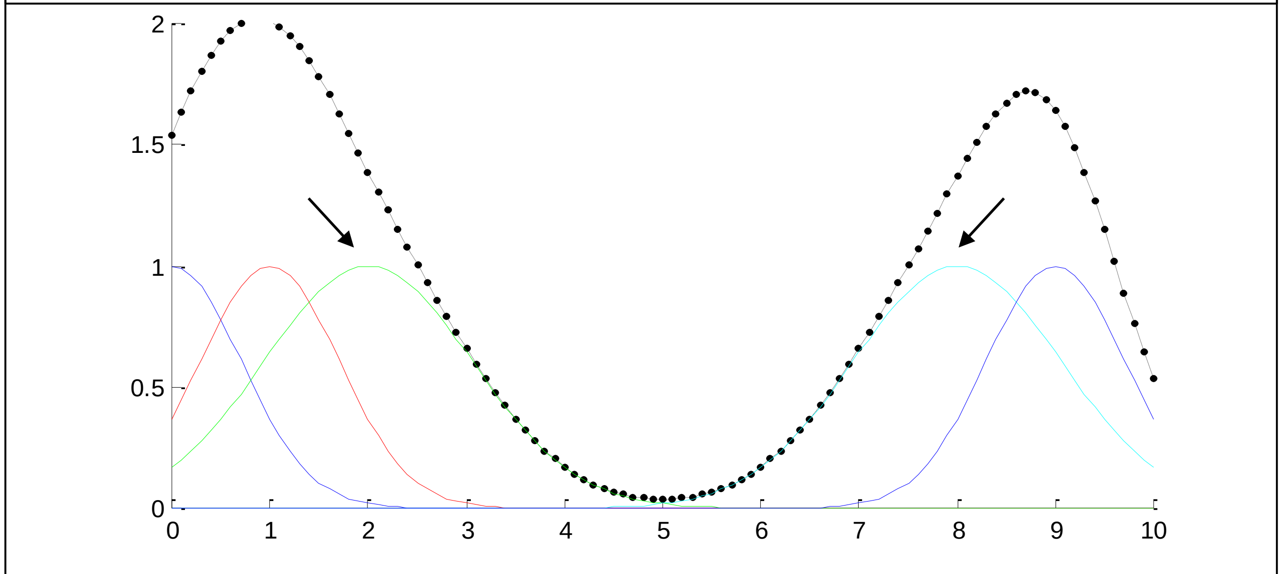
(-80, 8)
(-20 , 3)
(0, 2.5)
(20, 3)
(80, 8)

$$\hat{y}(x) =$$
$$8\exp(-(x+80)^2) +$$
$$3\exp(-(x+20)^2) +$$
$$2.5\exp(-(x+0)^2) +$$
$$3\exp(-(x-20)^2) +$$
$$8\exp(-(x-80)^2)$$

RBF Feed Back method



various b and w
W=[0,1,2, 8,9]
b=[1,1.2,1,1,1.2]

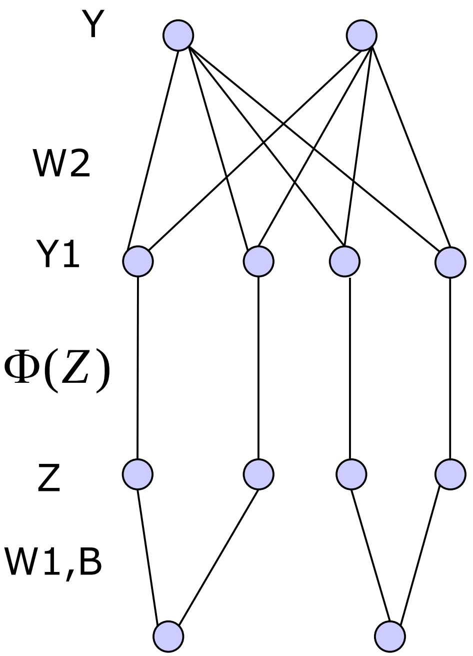


various b and w
W=[0,1,2, 8,9]
b=[1,1,1.2,1.2,1]

RBF Feedback Network Structure

- RBF Network

$$\hat{y}(x) = W_2 \Phi(x) = W_2 \exp\left(-\left(\frac{x - W_1}{B} \right)^2 \right)$$

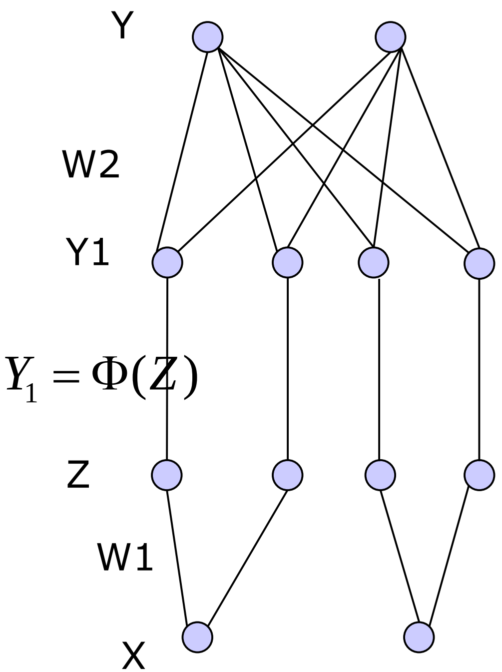


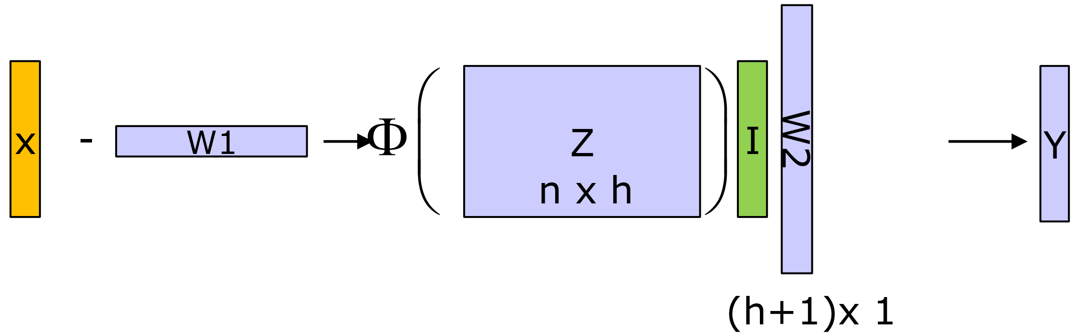
$$Y_1 = \Phi(Z)$$

$$Z = \frac{x - W_1}{B}$$

$$Y = W_2 Y_1 = W_2 \Phi(\mathrm{x}) = W_2 \exp\left(-Z^2 \right)$$

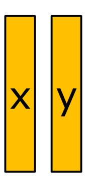
Network Model by Matrix Expression II



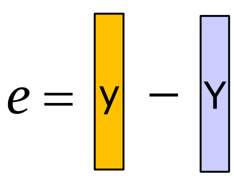


$$\underset{n\times 1}{x} - \underset{1\times h}{W_1} = \underset{n\times h}{Z} \xrightarrow{\ \Phi\ } \underset{n\times h}{Y_1} = \Phi(\underset{n\times h}{Z})$$

$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times 1}{W_2} \longrightarrow \underset{n\times 1}{Y}$$



Data

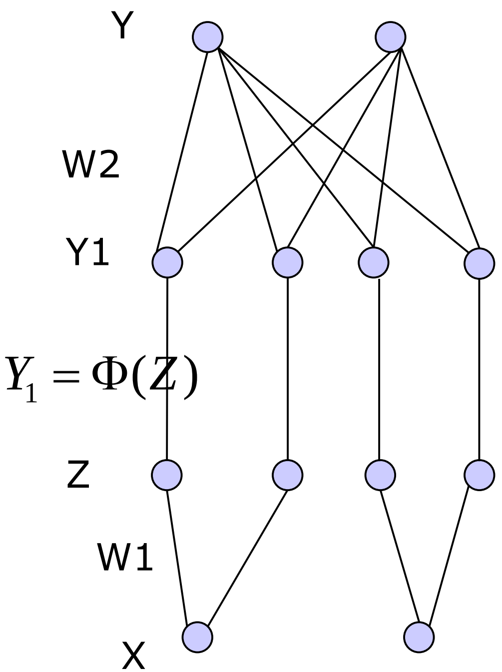


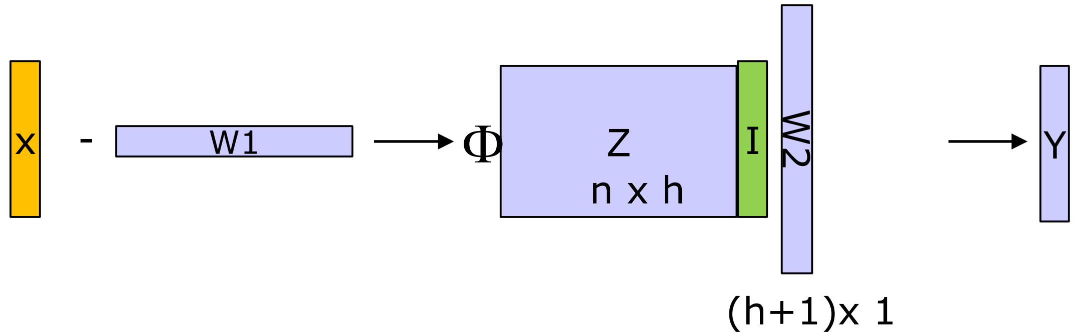
Neural Network

$$Y_1 = \Phi(Z)$$

$$Y = [Y_1 \quad I]W_2$$

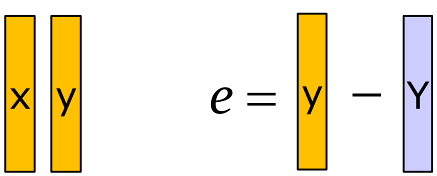
Network Model by Matrix Expression II



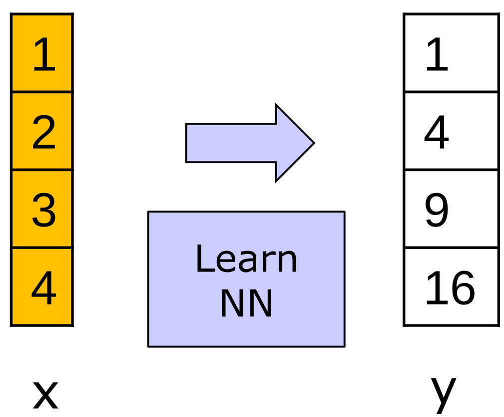


$$\underset{n\times 1}{x} - \underset{1\times h}{W_1} = \underset{n\times h}{Z} \xrightarrow{\Phi} \underset{n\times h}{Y_1} = \Phi(\underset{n\times h}{Z})$$

$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times 1}{W_2} \longrightarrow \underset{n\times 1}{Y}$$

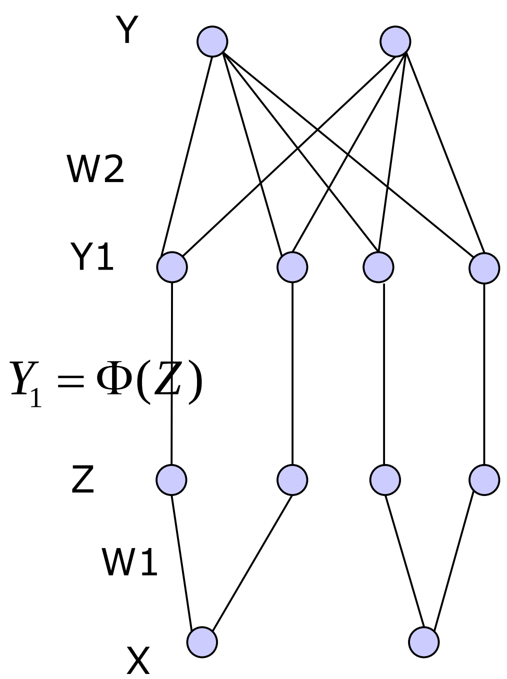


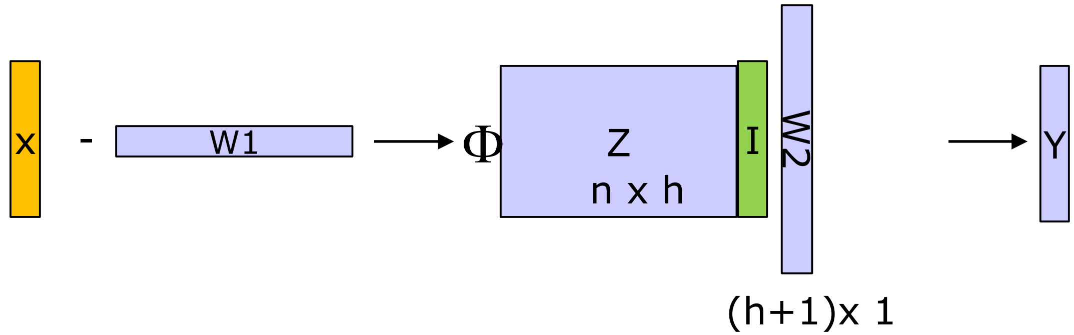
Data



$$y = x^2$$

Network Model by Matrix Expression II





$$\underset{n\times 1}{x} - \underset{1\times h}{W_1} = \underset{n\times h}{Z} \xrightarrow{\Phi} \underset{n\times h}{Y_1} = \Phi(\underset{n\times h}{Z})$$

$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times 1}{W_2} \longrightarrow \underset{n\times 1}{Y}$$

x (n):

1
2
3
4

\-

W1 (h):

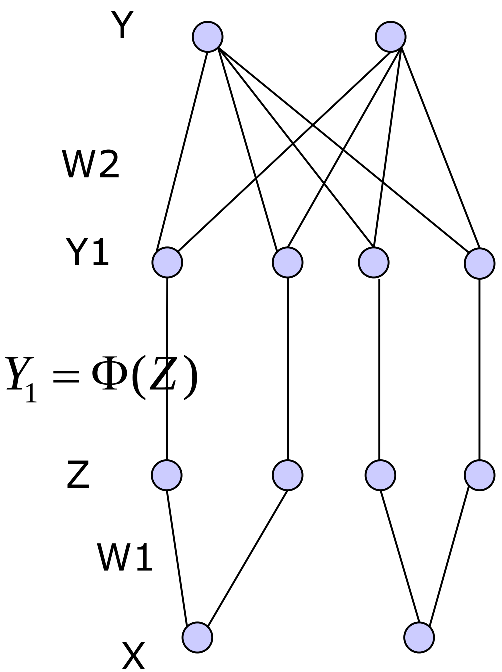
w11	w12	w13	w14	w15

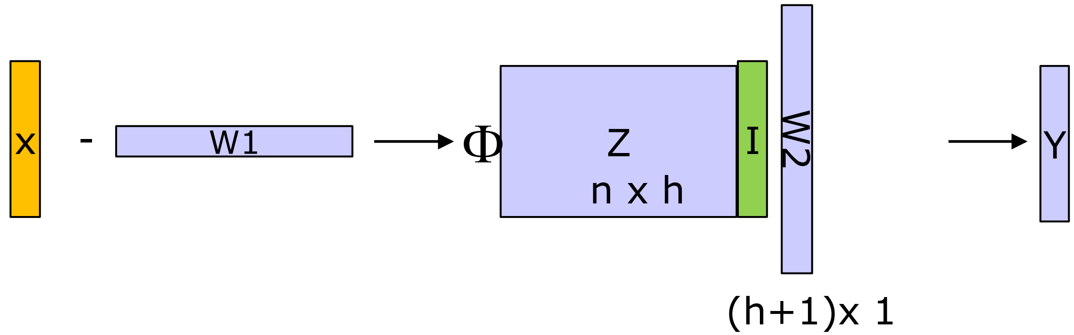
=

1-w11	1-w12	1-w13	1-w14	1-w15
2-w11	2-w12	2-w13	2-w14	2-w15
3-w11	3-w12	3-w13	3-w14	3-w15
4-w11	4-w12	4-w13	4-w14	4-w15

$$\times \frac{1}{b}$$

Network Model by Matrix Expression II





$$\underset{n\times 1}{x} - \underset{1\times h}{W_1} = \underset{n\times h}{Z} \xrightarrow{\Phi} \underset{n\times h}{Y_1} = \Phi(\underset{n\times h}{Z})$$

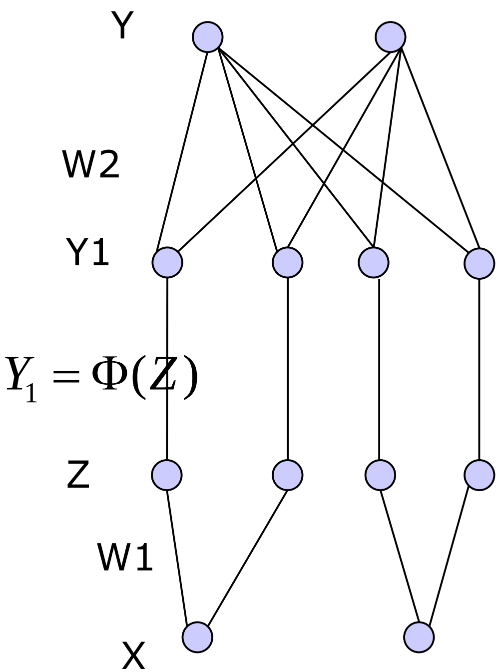
$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times 1}{W_2} \longrightarrow \underset{n\times 1}{Y}$$

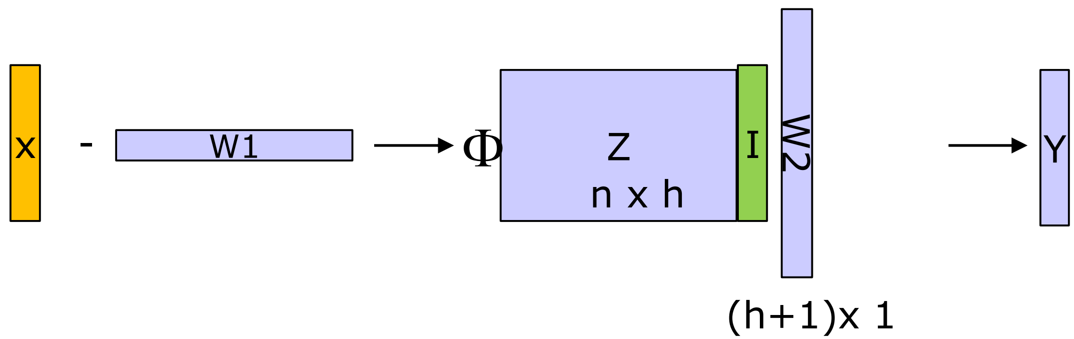
$$Y_1 = \Phi\left(\underset{n \times h}{M} \times \frac{1}{b} \right)$$

where M (n rows, h columns) is:

1-w11	1-w12	1-w13	1-w14	1-w15
2-w11	2-w12	2-w13	2-w14	2-w15
3-w11	3-w12	3-w13	3-w14	3-w15
4-w11	4-w12	4-w13	4-w14	4-w15

Network Model by Matrix Expression II





$$\underset{n\times 1}{x} - \underset{1\times h}{W_1} = \underset{n\times h}{Z} \xrightarrow{\Phi} \underset{n\times h}{Y_1} = \Phi(\underset{n\times h}{Z})$$

$$\downarrow$$

$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times 1}{W_2} \longrightarrow \underset{n\times 1}{Y}$$

h

n

Φ

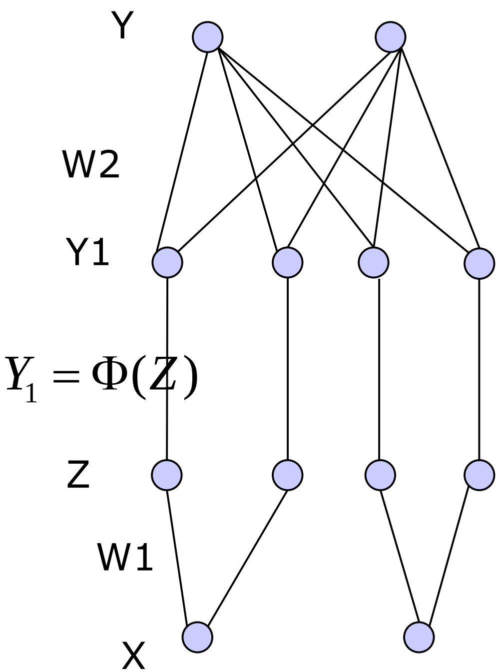
1-w11	1-w12	1-w13	1-w14	1-w15
2-w11	2-w12	2-w13	2-w14	2-w15
3-w11	3-w12	3-w13	3-w14	3-w15
4-w11	4-w12	4-w13	4-w14	4-w15

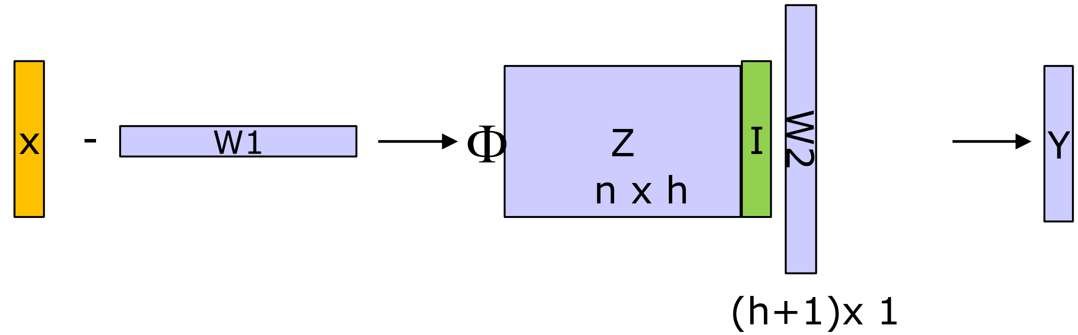
$\times \frac{1}{b}$

1
1
1
1

w2,1
w2,2
w2,3
w2,4
w2,5
w2,6

Network Model by Matrix Expression II





$$\underset{n\times 1}{x} - \underset{1\times h}{W_1} = \underset{n\times h}{Z} \xrightarrow{\Phi} \underset{n\times h}{Y_1} = \Phi(\underset{n\times h}{Z})$$

$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times 1}{W_2} \longrightarrow \underset{n\times 1}{Y}$$

$$\hat{y} =$$

$$\begin{pmatrix}
\Phi(1-w_{11})w_{2,1} + \Phi(1-w_{12})w_{2,2} + \Phi(1-w_{13})w_{2,3} + \Phi(1-w_{14})w_{2,4} + \Phi(1-w_{15})w_{2,5} + w_{2,6} \\
\Phi(2-w_{11})w_{2,1} + \Phi(2-w_{12})w_{2,2} + \Phi(2-w_{13})w_{2,3} + \Phi(2-w_{14})w_{2,4} + \Phi(2-w_{15})w_{2,5} + w_{2,6} \\
\Phi(3-w_{11})w_{2,1} + \Phi(3-w_{12})w_{2,2} + \Phi(3-w_{13})w_{2,3} + \Phi(3-w_{14})w_{2,4} + \Phi(3-w_{15})w_{2,5} + w_{2,6} \\
\Phi(4-w_{11})w_{2,1} + \Phi(4-w_{12})w_{2,2} + \Phi(4-w_{13})w_{2,3} + \Phi(4-w_{14})w_{2,4} + \Phi(4-w_{15})w_{2,5} + w_{2,6}
\end{pmatrix}$$

Simpler than sigmoidal function

Differentiation with W2 = Same with sigmoidal function

$$J = \frac{1}{2}\sum_k {e_k}^2 = \frac{1}{2} e^T e$$

$$\frac{\partial J}{\partial W_2} = e^T \frac{\partial e}{\partial W_2}$$

Lemma 5.

$$c = x^T x$$

$$\frac{\partial c}{\partial z} = 2x^T \frac{\partial x}{\partial z}$$

$$\frac{\partial J}{\partial W_2} = e^T \frac{\partial e}{\partial W_2} = -e^T \frac{\partial Y}{\partial W_2} = -e^T \frac{\partial [Y_1 \quad I]W_2}{\partial W_2} = -e^T{}_{1\times n}[Y_1 \quad I]_{n\times(h+1)}$$

$$Transpose \rightarrow Vector = \left(\frac{\partial J}{\partial W_2}\right)^T_{(h+1)\times 1} = -\left([Y_1 \quad I]_{n\times(h+1)}\right)^T e_{n\times 1} = -[Y_1 \quad I]^T e$$

Differentiation with W1

$$J = \frac{1}{2}\sum_k e_k{}^2$$

$$\frac{\partial J}{\partial w_{11}} = \sum_k e_k \frac{\partial e_k}{\partial w_{11}} = \sum_k e_k \frac{\partial (y_k - \hat{y}_k)}{\partial w_{11}} = -\sum_k e_k \frac{\partial \hat{y}_k}{\partial w_{11}}$$

$$= -e_1 \partial \frac{\Phi(1 - w_{11})w_{2,1} + \Phi(1 - w_{12})w_{2,2} + \Phi(1 - w_{13})w_{2,3} + \Phi(1 - w_{14})w_{2,4} + \Phi(1 - w_{15})w_{2,5} + w_{2,6}}{\partial w_{11}}$$

$$-e_2 \partial \frac{\Phi(2 - w_{11})w_{2,1} + \Phi(2 - w_{12})w_{2,2} + \Phi(2 - w_{13})w_{2,3} + \Phi(2 - w_{14})w_{2,4} + \Phi(2 - w_{15})w_{2,5} + w_{2,6}}{\partial w_{11}}$$

$$-e_3 \partial \frac{\Phi(3 - w_{11})w_{2,1} + \Phi(3 - w_{12})w_{2,2} + \Phi(3 - w_{13})w_{2,3} + \Phi(3 - w_{14})w_{2,4} + \Phi(3 - w_{15})w_{2,5} + w_{2,6}}{\partial w_{11}}$$

$$-e_4 \partial \frac{\Phi(4 - w_{11})w_{2,1} + \Phi(4 - w_{12})w_{2,2} + \Phi(4 - w_{13})w_{2,3} + \Phi(4 - w_{14})w_{2,4} + \Phi(4 - w_{15})w_{2,5} + w_{2,6}}{\partial w_{11}}$$

Differentiation with W1

$$J = \frac{1}{2}\sum_k e_k{}^2$$

$$\frac{\partial J}{\partial w_{11}} = \sum_k e_k \frac{\partial e_k}{\partial w_{11}} = \sum_k e_k \frac{\partial (y_k - \hat{y}_k)}{\partial w_{11}} = -\sum_k e_k \frac{\partial \hat{y}_k}{\partial w_{11}}$$

$$= -e_1 \frac{\partial \Phi(1 - w_{11}) w_{2,1}}{\partial w_{11}} - e_2 \frac{\partial \Phi(2 - w_{11}) w_{2,1}}{\partial w_{11}} - e_3 \frac{\partial \Phi(3 - w_{11}) w_{2,1}}{\partial w_{11}} - e_4 \frac{\partial \Phi(4 - w_{11}) w_{2,1}}{\partial w_{11}}$$

$$= -e_1 \Phi'(1 - w_{11}) w_{2,1} - e_2 \Phi'(2 - w_{11}) w_{2,1} - e_3 \Phi'(3 - w_{11}) w_{2,1} - e_4 \Phi'(4 - w_{11}) w_{2,1}$$

Can you find the PATTERN?

Differentiation with W1

$$J = \frac{1}{2}\sum_k {e_k}^2$$

$$\frac{\partial J}{\partial w_{1j}} = \sum_k e_k \frac{\partial e_k}{\partial w_{ij}} = \sum_k e_k \frac{\partial (y_k - \hat{y}_k)}{\partial w_{ij}} = -\sum_k e_k \frac{\partial \hat{y}_k}{\partial w_{ij}}$$

$$= -e_1 \Phi'(1 - w_{1j}) w_{2,1} - e_2 \Phi'(2 - w_{1j}) w_{2,1} - e_3 \Phi'(3 - w_{1j}) w_{2,1} - e_4 \Phi'(4 - w_{1j}) w_{2,1}$$

$$= -\sum_k e_k \Phi'(x_k - w_{1j}) w_{2,1}$$

- RBF is much simpler than Sigmoidal function
- Remind that Differentiation of Exponent is also simple
- $$\Phi(x) = \exp(x)$$ $$\Phi'(x) = \exp(x) = \Phi(x)$$

Differentiation with W1

$$J = \frac{1}{2}\sum_k e_k{}^2$$

$$\frac{\partial J}{\partial w_{1j}} = \sum_k e_k \frac{\partial e_k}{\partial w_{1j}} = \sum_k e_k \frac{\partial (y_k - \hat{y}_k)}{\partial w_{1j}} = -\sum_k e_k \frac{\partial \hat{y}_k}{\partial w_{1j}}$$

$$= -e_1\Phi'(1 - w_{1j})w_{2,1} - e_2\Phi'(2 - w_{1j})w_{2,1} - e_3\Phi'(3 - w_{1j})w_{2,1} - e_4\Phi'(4 - w_{1j})w_{2,1}$$

$$= -\sum_k e_k\Phi'(x_k - w_{1j})w_{2,j} = -\sum_k \left(eW_{2,h\times 1}^T\right)_{kj} \Phi'(x_k - w_{1j})$$

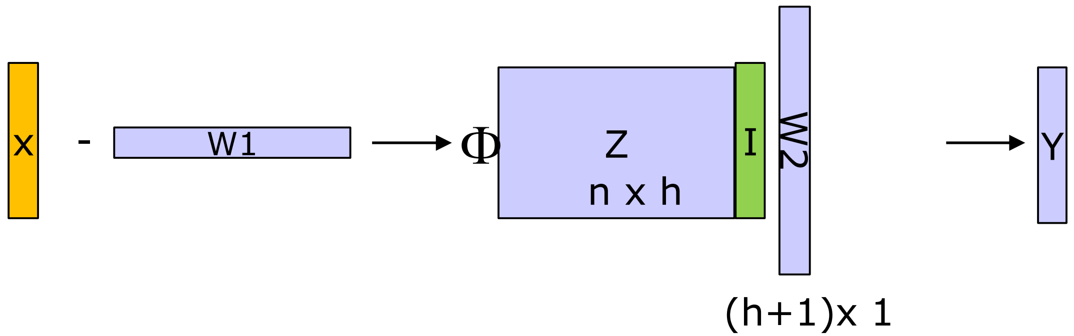
$$= -\sum_k \left(eW_{2,h\times 1}^T\right)_{kj} \Phi'(\mathrm{Z}_{kj})$$

$$= -\sum_k \left(eW_{2,h\times 1}^T\right)_{kj} \frac{\partial}{\partial w_{1j}}\left(\mathrm{Exp}\left(-\left(\frac{x_k - w_{1j}}{b}\right)^2\right)\right) = -2\sum_k \left(eW_{2,h\times 1}^T\right)_{kj} \Phi(\mathrm{Z}_{kj})\,\mathrm{Z}_{kj}$$

$$\therefore \frac{\partial J}{\partial W_1} = -2\sum_k \left(eW_2^T\right)_k \circ \Phi_k \circ Z_k = -2\sum_k \left(eW_2^T\right)_k \circ Y_{1,k} \circ Z_k$$

Where j?

Network Model by Matrix Expression II

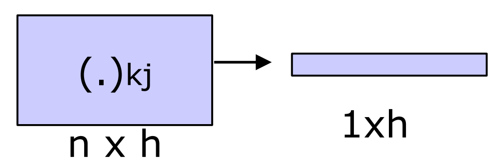


$$\underset{n\times 1}{x} - \underset{1\times h}{W_1} = \underset{n\times h}{Z} \xrightarrow{\Phi} \underset{n\times h}{Y_1} = \Phi(\underset{n\times h}{Z})$$

$$\downarrow$$

$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times 1}{W_2} \longrightarrow \underset{n\times 1}{Y}$$

$$\frac{\partial J}{\partial w_{1j}} = 2\sum_k \left(eW^T_{2,h\times 1}\right)_{kj} \Phi(Z^i_{kj})\, Z^i_{kj} = 2\sum_k (.)_{kj}$$

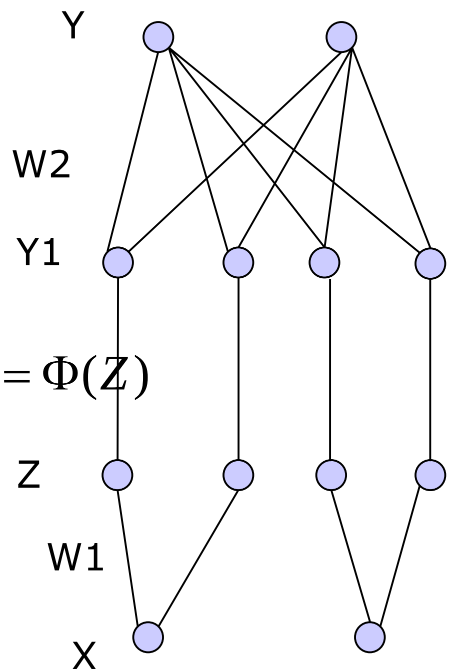


$$\sum_k (.)_{kj}$$

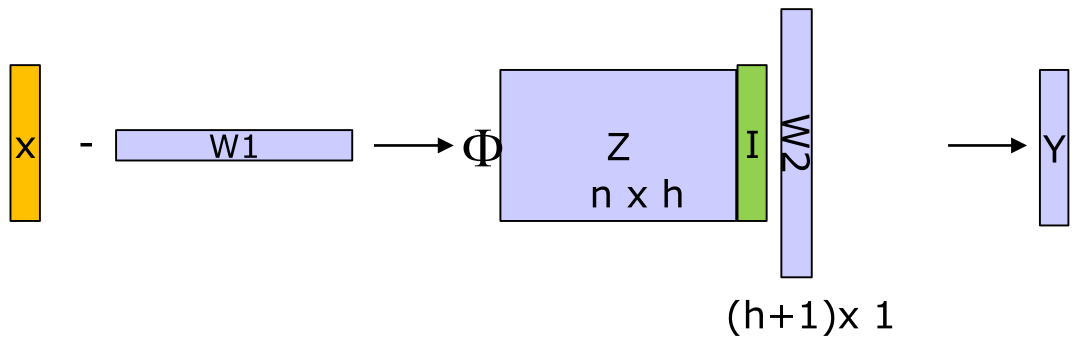
$$\frac{\partial J}{\partial W_1} = \left[\frac{\partial J}{\partial w_{11}}, \frac{\partial J}{\partial w_{12}}, \frac{\partial J}{\partial w_{13}}, \dots \frac{\partial J}{\partial w_{1h}}\right]_{1\times h} = \left[2\sum_k (.)_{k1}, 2\sum_k (.)_{k2}, 2\sum_k (.)_{k3}, \dots, 2\sum_k (.)_{kh}\right]_{1\times n}$$

$$\therefore \frac{\partial J}{\partial W_1} = 2\sum_k \left(eW_2^T\right)_k \circ Y_{1,k} \circ Z_k$$

Network Model by Matrix Expression II



$$Y_1 = \Phi(Z)$$



$$\underset{n\times 1}{x} - \underset{1\times h}{W_1} = \underset{n\times h}{Z} \xrightarrow{\Phi} \underset{n\times h}{Y_1} = \Phi(\underset{n\times h}{Z})$$

$$\downarrow$$

$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times 1}{W_2} \longrightarrow \underset{n\times 1}{Y}$$

$$e_{n\times 1}\left(\underset{(h+1)\times 1}{W_2}\right)^T = e_{n\times 1}\left(W_2^T\right)_{1\times(h+1)} = \left(eW_2^T\right)_{n\times(h+1)}$$

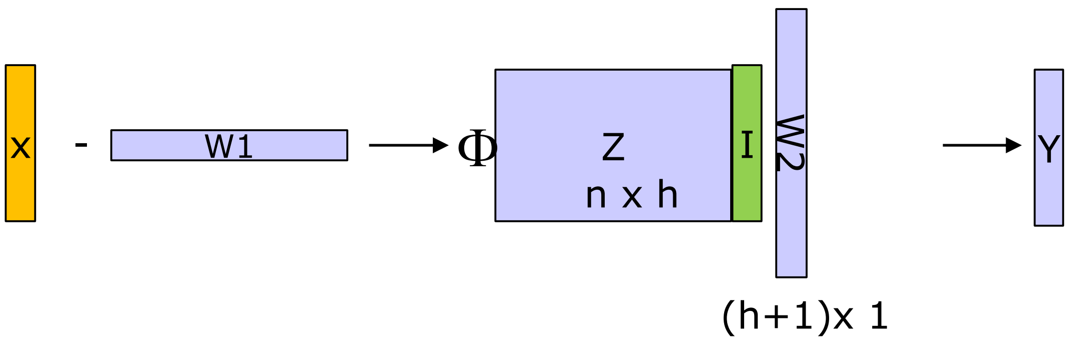
$$\Rightarrow e_{n\times 1}\left(\underset{h\times 1}{W_2}\right)^T = e_{n\times 1}\left(W_2^T\right)_{1\times h} = \left(eW_2^T\right)_{n\times h}$$

$$J = \frac{1}{2}\sum_k \left(y_k - \hat{y}_k\right)^2 = \frac{1}{2}\sum_k e_k{}^2 = \frac{1}{2}e^T e$$

$$Vector: \frac{\partial J}{\partial W_1} = -2\sum_{k=1..n}\left(eW_2^T\right)_k \circ Y_{1,k} \circ Z_k$$

$$Vector: \frac{\partial J}{\partial W_2} = -[Y_1 \quad I]^T e$$

Example: l7rbf.py



$$\underset{n\times1}{x} - \underset{1\times h}{W_1} = \underset{n\times h}{Z} \xrightarrow{\Phi} \underset{n\times h}{Y_1} = \Phi(\underset{n\times h}{Z})$$

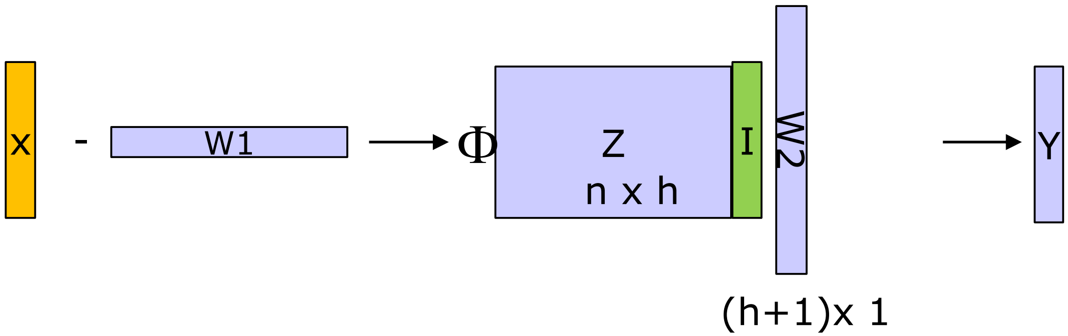
$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times1}{W_2} \longrightarrow \underset{n\times1}{Y}$$

```
n=11
h=200

x= linspace(0,10,n).T()
y= -0.1*pow(x-2,2);

# x
X=x
I=ones(n,h)

W1=20*randn(1,h)      #mean
b=1
W2=randn(h+1,1)
Z =array(n,h)

Y1=array(n,h+1)
Y1[:,h+1]    = 1;

alpha=0.1;
```

$$J = \frac{1}{2}\sum_k (y_k - \hat{y}_k)^2 = \frac{1}{2}\sum_k e_k{}^2 = \frac{1}{2}e^T e$$

$$Vector : \left.\frac{\partial J}{\partial W_1}\right|_{1\times h} = -2\sum_{k=1\ldots n} \left(eW_2^T\right)_{k\times h} \circ Y_{1,k\times h} \circ Z_{k\times h}$$

$$Vector : \left.\frac{\partial J}{\partial W_2}\right|_{(h+1)\times1} = -\left[[Y_{1,n\times h} \quad I_{n\times1}]^T\right]_{(h+1)\times n} e_{n\times1}$$



$$\underset{n\times 1}{x} - \underset{1\times h}{W_1} = \underset{n\times h}{Z} \xrightarrow{\Phi} \underset{n\times h}{Y_1} = \Phi(\underset{n\times h}{Z})$$

$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times 1}{W_2} \longrightarrow \underset{n\times 1}{Y}$$

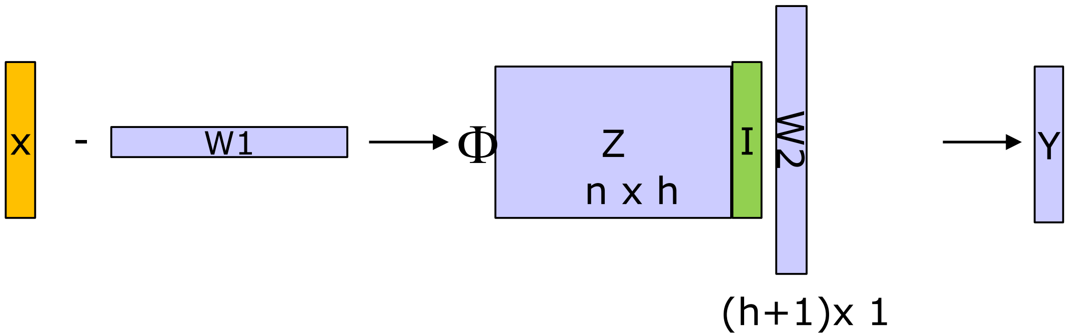
$$J = \frac{1}{2}\sum_k \left(y_k - \hat{y}_k\right)^2 = \frac{1}{2}\sum_k {e_k}^2 = \frac{1}{2} e^T e$$

$$Vector: \frac{\partial J}{\partial W_1} = -2\sum_k \left(e W_2^T\right)_k \circ Y_{1,k} \circ Z_k$$

$$Vector: \frac{\partial J}{\partial W_2} = -[Y_1 \quad I]^T e$$

$$Z = \frac{x - W_1}{b}$$

```
for i in range(0,2000):
    for i in range(1,n+1):
        Z[i,:]  = (X[i,:]-W1[1,:])/b

    Y1[:,1:h]   = exp(-Z.mul(Z))
    Y           = Y1*W2
    e           = y-Y
    J           = e.T()*e

    dW2  = -Y1.T()*e
    dW1  = -2*((e*W2[1:h,:].T()).mul(Y1[:,1:h])).mul(Z)
    dW1  = sum(dW1,1);

    W1   = W1-alpha*dW1
    W2   = W2-alpha*dW2
```



$$\underset{n\times 1}{x} - \underset{1\times h}{W_1} = \underset{n\times h}{Z} \xrightarrow{\Phi} \underset{n\times h}{Y_1} = \Phi(\underset{n\times h}{Z})$$

$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times 1}{W_2} \longrightarrow \underset{n\times 1}{Y}$$

$$J = \frac{1}{2}\sum_k (y_k - \hat{y}_k)^2 = \frac{1}{2}\sum_k e_k{}^2 = \frac{1}{2}e^T e$$

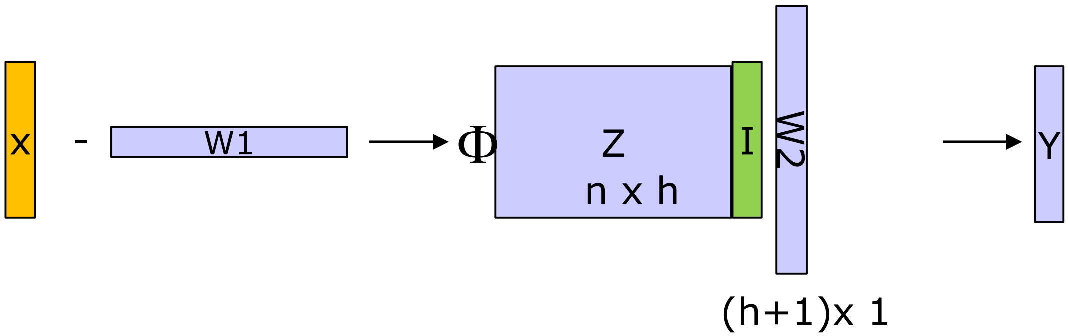
$$Vector: \frac{\partial J}{\partial W_1} = -2\sum_k \left(eW_2^T\right)_k \circ Y_{1,k} \circ Z_k$$

$$Vector: \frac{\partial J}{\partial W_2} = -[Y_1 \quad I]^T e$$

$$Z = \frac{x - W_1}{b}$$

$$e_{n\times 1}\left(\underset{(h+1)\times 1}{W_2}\right)^T \Rightarrow e_{n\times 1}\left(\underset{h\times 1}{W_2}\right)^T = \left(eW_2^T\right)_{n\times h}$$

```
for i in range(0,2000):
    for i in range(1,n+1):
        Z[i,:]   = (X[i,:]-W1[1,:])/b

    Y1[:,1:h]    = exp(-Z.mul(Z))
    Y            = Y1*W2
    e            = y-Y
    J            = e.T()*e

    dW2  = -Y1.T()*e
    dW1  = -2*((e*W2[1:h,:].T()).mul(Y1[:,1:h])).mul(Z)
    dW1  = sum(dW1,1);

    W1   = W1-alpha*dW1
    W2   = W2-alpha*dW2
```



$$\underset{n\times 1}{x} - \underset{1\times h}{W_1} = \underset{n\times h}{Z} \xrightarrow{\Phi} \underset{n\times h}{Y_1} = \Phi(\underset{n\times h}{Z})$$

$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times 1}{W_2} \longrightarrow \underset{n\times 1}{Y}$$

$$J = \frac{1}{2}\sum_k \left(y_k - \hat{y}_k\right)^2 = \frac{1}{2}\sum_k {e_k}^2 = \frac{1}{2}e^T e$$

$$Vector: \frac{\partial J}{\partial W_1} = -2\sum_k \left(eW_2^T\right)_k \circ Y_{1,k} \circ Z_k$$

$$Vector: \frac{\partial J}{\partial W_2} = -[Y_1 \quad I]^T e$$

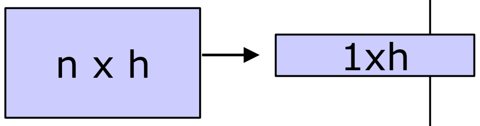
$$Z = \frac{x - W_1}{b}$$

X=[0 1 2
3 4 5]

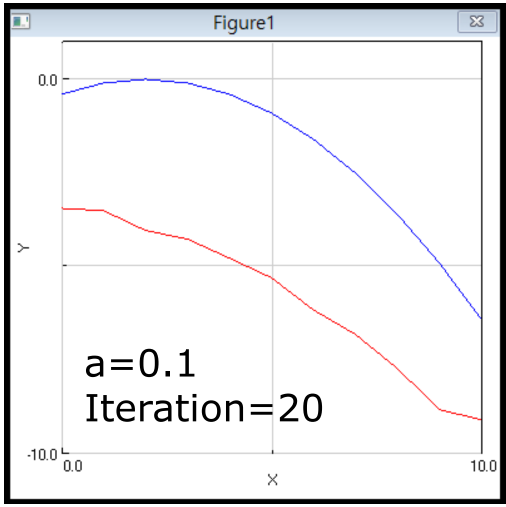
sum(X,1)
=[3 5 7]

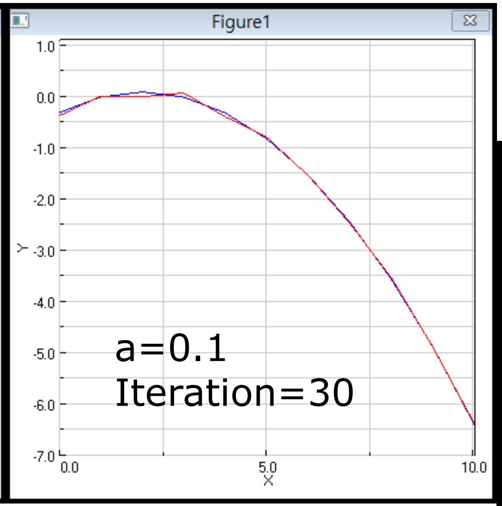
sum(X,2)
=[3
12]

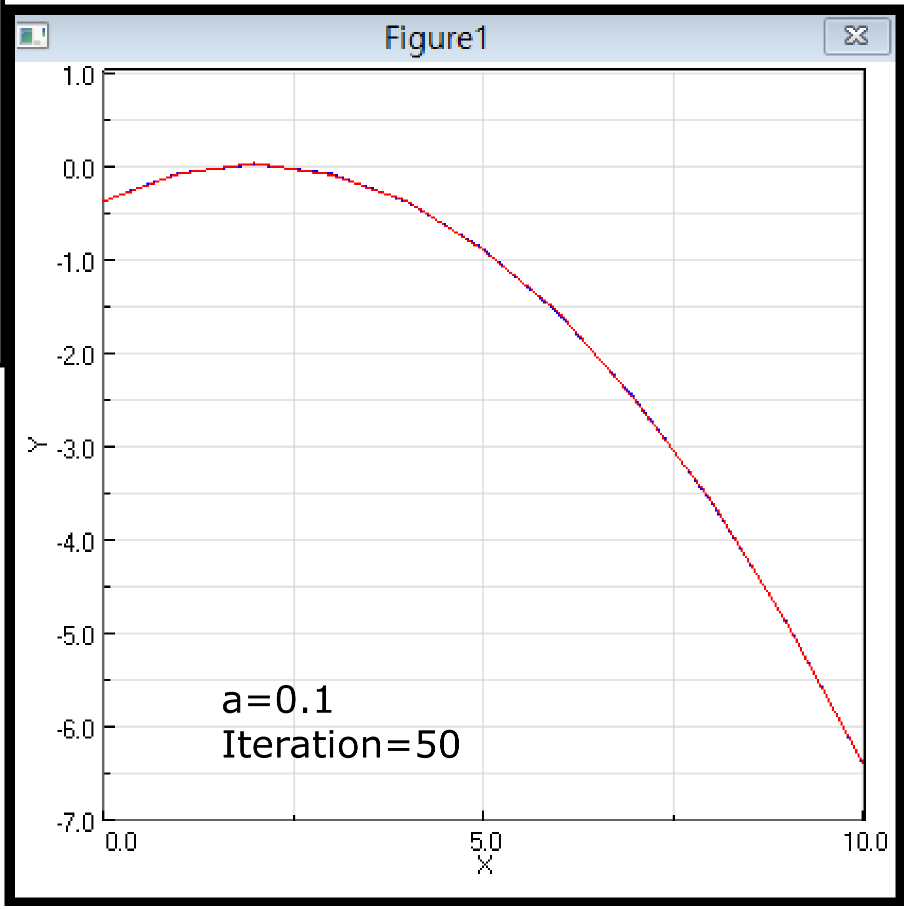
```
for i in range(0,2000):
    for i in range(1,n+1):
        Z[i,:]   = (X[i,:]-W1[1,:])/b

    Y1[:,1:h]    = exp(-Z.mul(Z))
    Y            = Y1*W2
    e            = y-Y
    J            = e.T()*e

    dW2  = -Y1.T()*e
    dW1  = -2*((e*W2[1:h,:].T()).mul(Y1[:,1:h])).mul(Z)
    dW1  = sum(dW1,1);

    W1   = W1-alpha*dW1
    W2   = W2-alpha*dW2
```



RBF Result





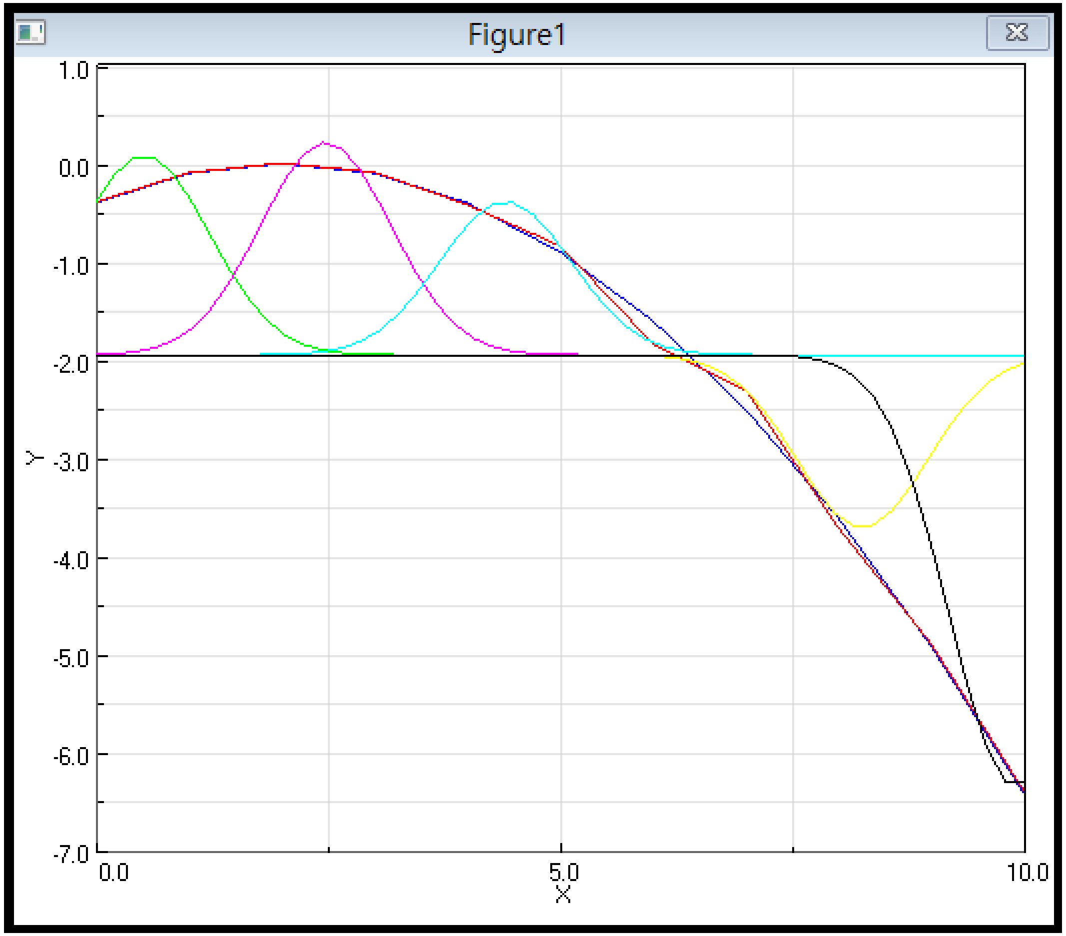


RBF shows the best performance

2 RBF Weight

Secrets of RBF Network: "Initial Weight" is important

- RBF Neural Network is the sum of Weighted RBF

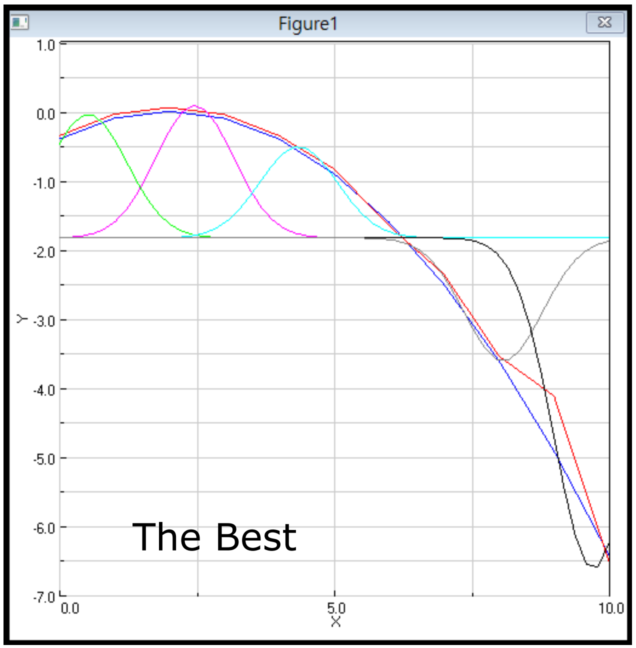


See example, l4rbf5k

-Only 5 kernels are used.

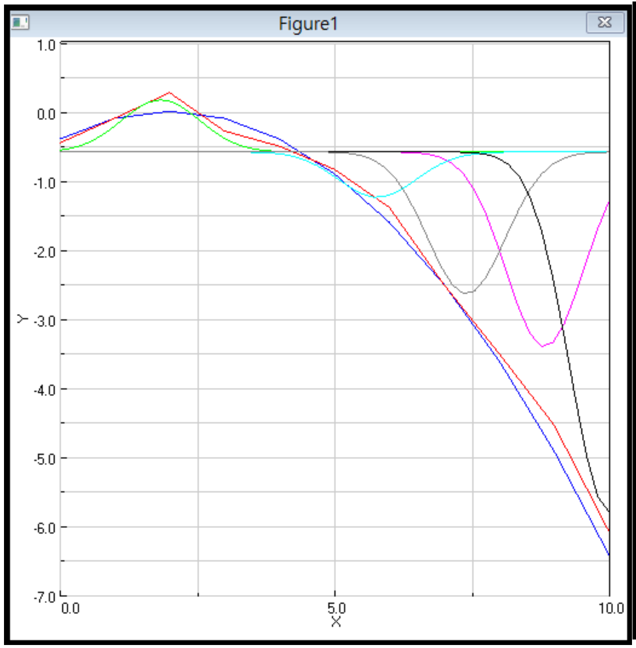
Comparison Three cases
See, Red line is the Result

W1=linspace(0,10,h)



W1=[0 2.5 5 7.5 10]

W1=10*rand(1,h)

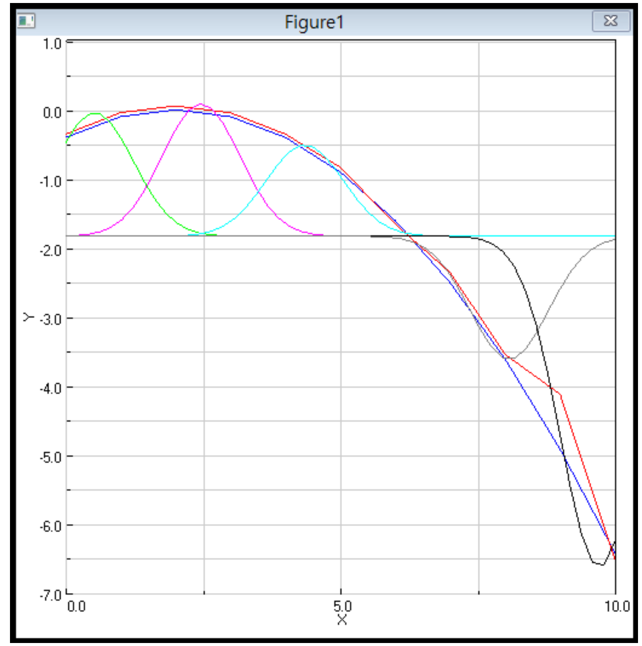


W1=[2.2, 1.5, 6.8, 8.2, 7.8]

W1=10*randn(1,h)

W1=
[9.3679, 16.7770, -5.3647, 8.0107, 2.2742]

Kernels are Fighting



- dW is given for kernel's development.
- Each kernel wants to grow.
- But, Good for the one is NOT Good for others

Initial Guess of Weight is mean value

$$\hat{y}(x) = \sum_k \Phi_k(x) = \sum_k \exp\left(-\left(\frac{x - w_k}{b_k}\right)^2\right)$$

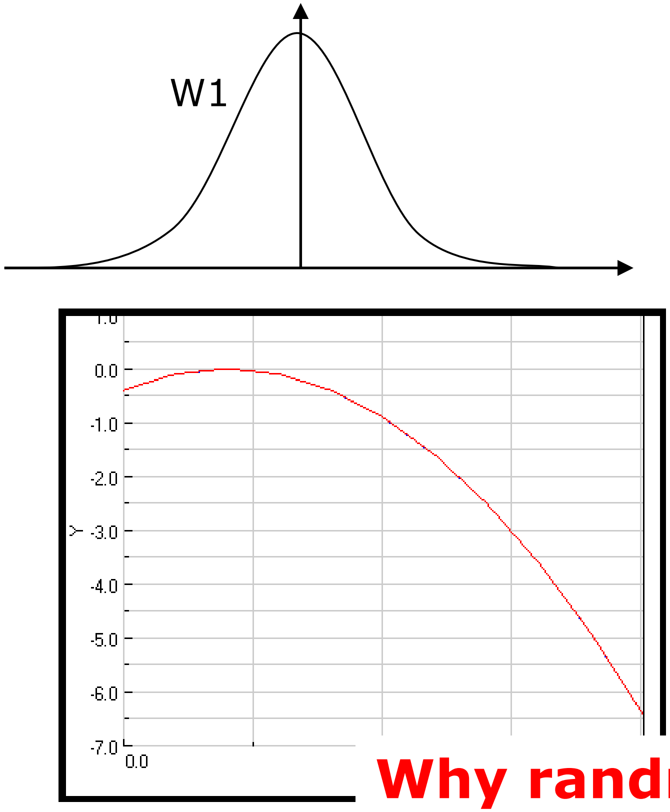
- W1=linspace(0,10,h) is balanced guess between 0~10
- rand or randn(Gaussian random) might be Good or Bad.
- Thus, case 1 is the best.

- Question:
  - 1. NN performance looks very sensitive to initial guess. Then, how we use it?
  - 2. If we increase kernel number, what happens?

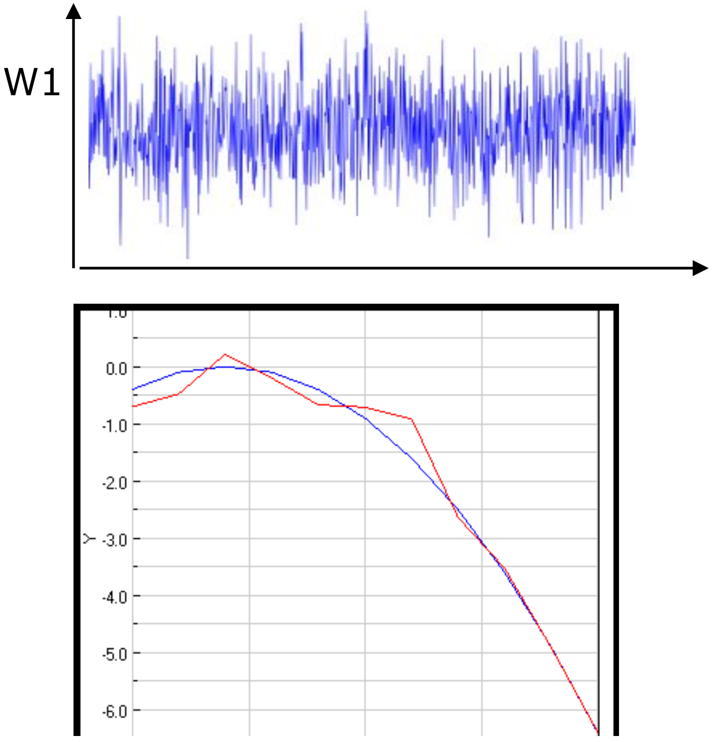
Back to l4rbf.py

- Compare two cases

W1=20*randn(1,h)

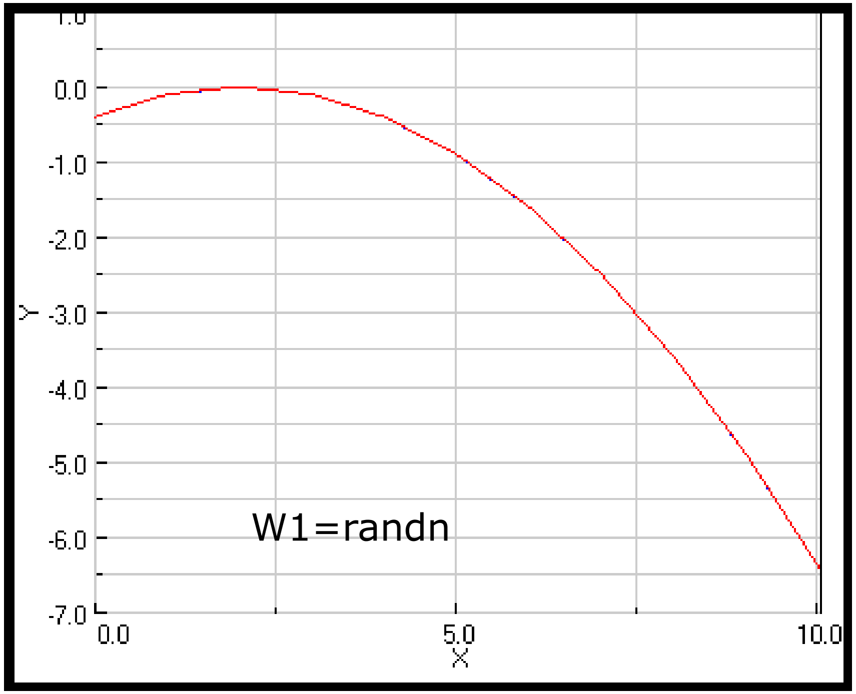


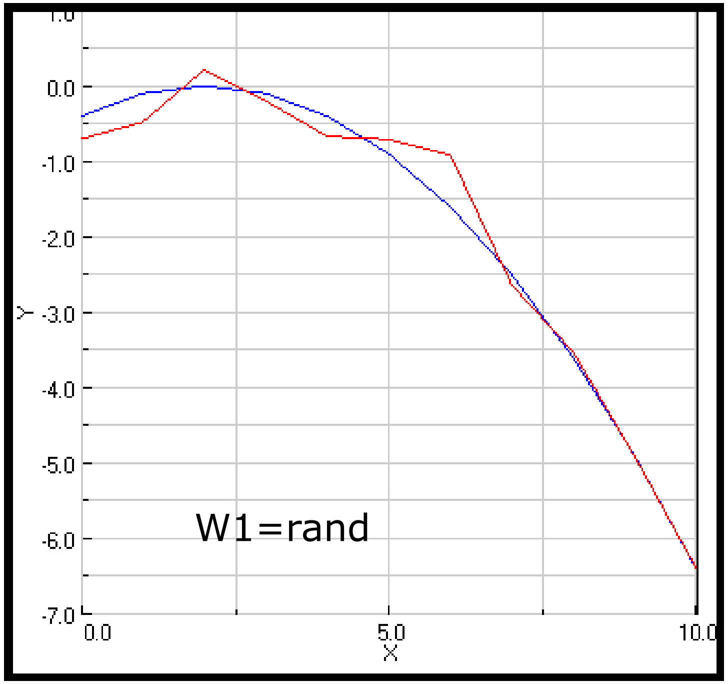
W1=20*rand(1,h)
(White noise)



Why randn is better than rand?

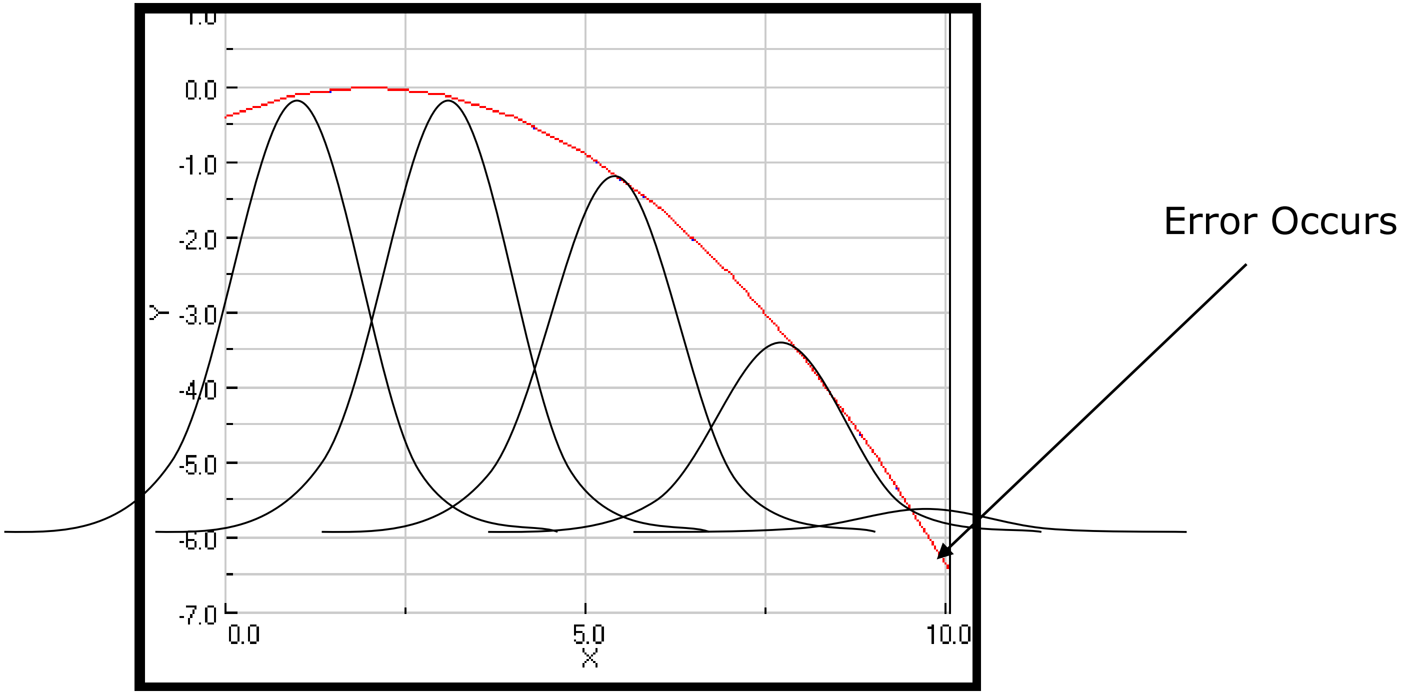
Why randn is better than rand **in this case**?





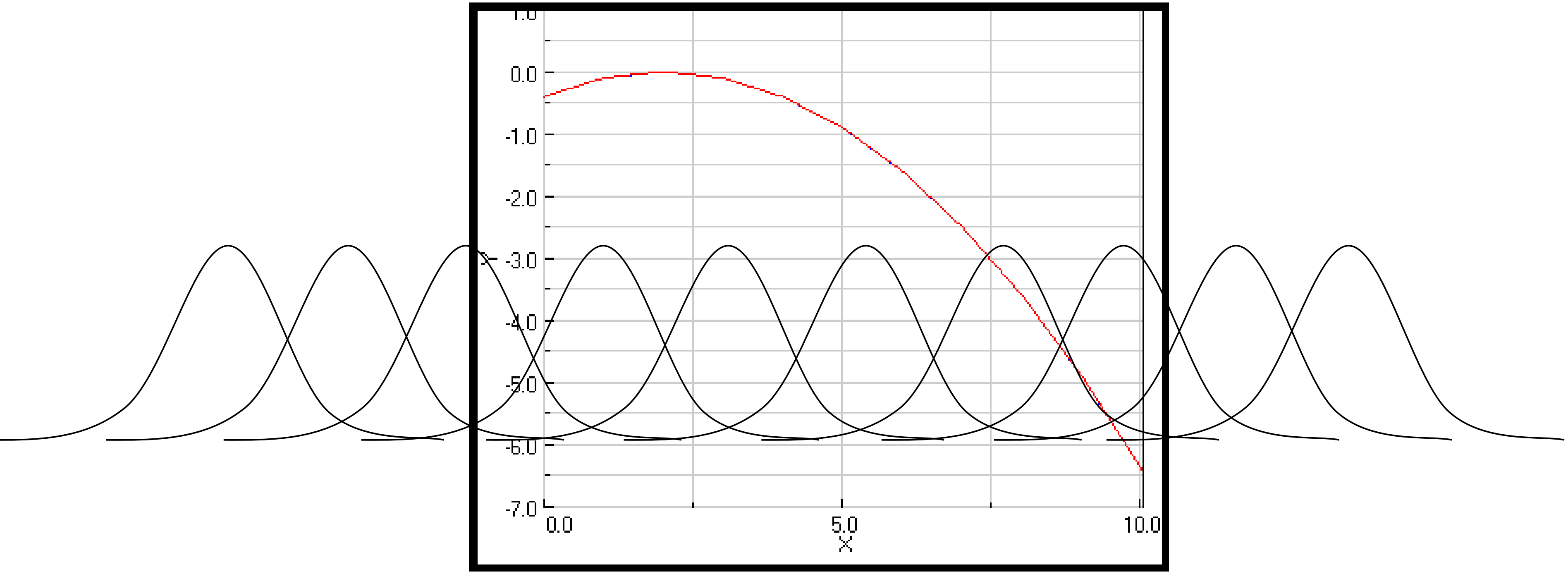
- randn() generates W1 around zero. Thus, estimation error around zero is Better.
- rand() seems to have lack of initial kernel around zero. Thus, big errors around zero.

Why Not 10 but 20 is taken?



- Graph is drawn from 0 to 10.
- But, W1 = 10 randn
- X<0 or X>10 has larger errors

Why Not 10 but 20 is taken?



- Kernels < 0 or Kernels >10 makes an good effect on Neural network(NN) estimation.

3 Neural Network with Multiple Sample

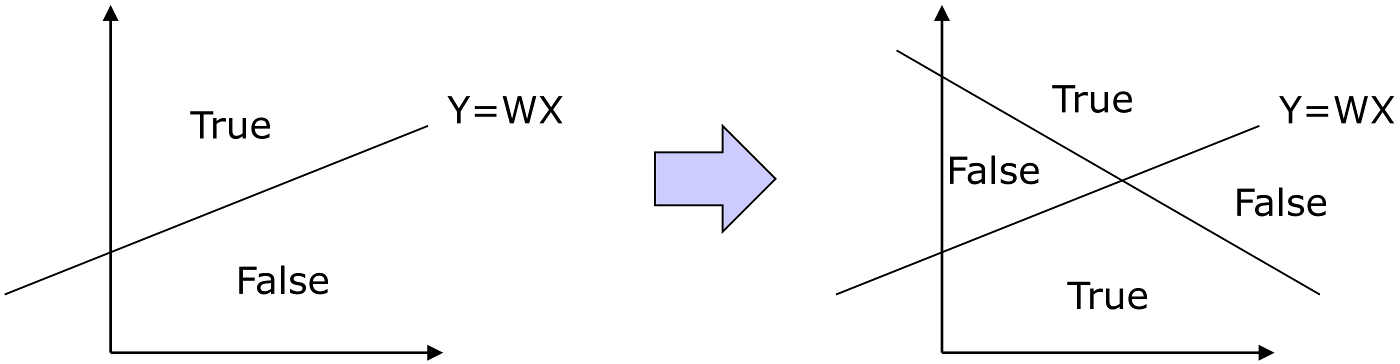
XOR Problems

- XOR operation

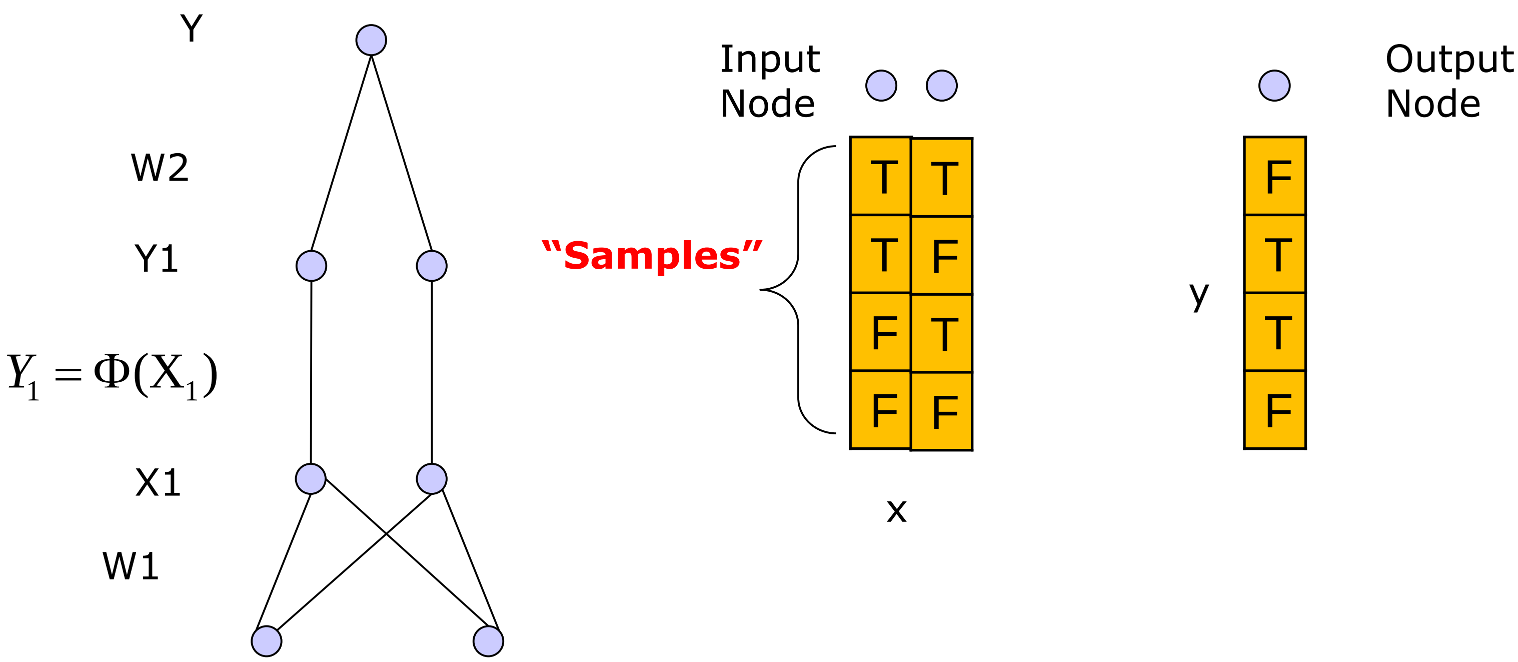
A	B	XOR
1	1	0
1	0	1
0	1	1
0	0	0

Special use
A=0 or 1
XOR(A,A)=0

- NN was challenged with XOR Problem by Minsky

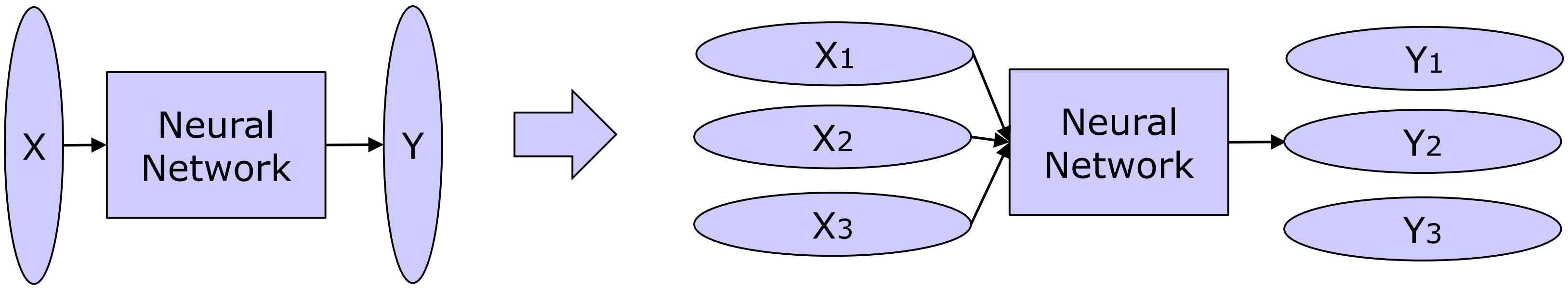


Build Network Structure
Multiple Samples Inputs X=(x1,x2)



- See that 4 cases are thought as sample number

See the Network Differences
Sigmoidal and RBF Network VS. XOR Example



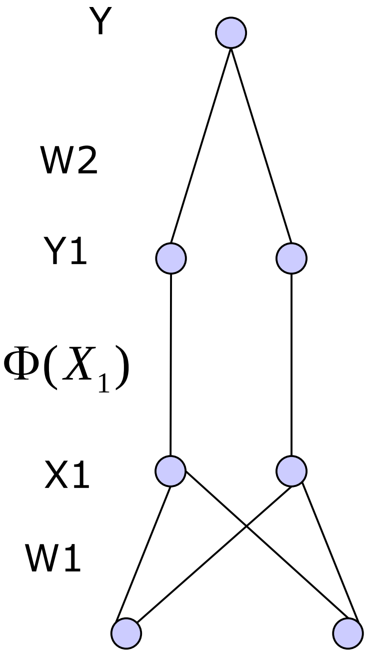
Input, X　　Output, Y　　Input, X　　Output, Y

Sigmoidal(logsig) and RBF
Network Example

It is the first Example that
We uses multiple samples as inputs

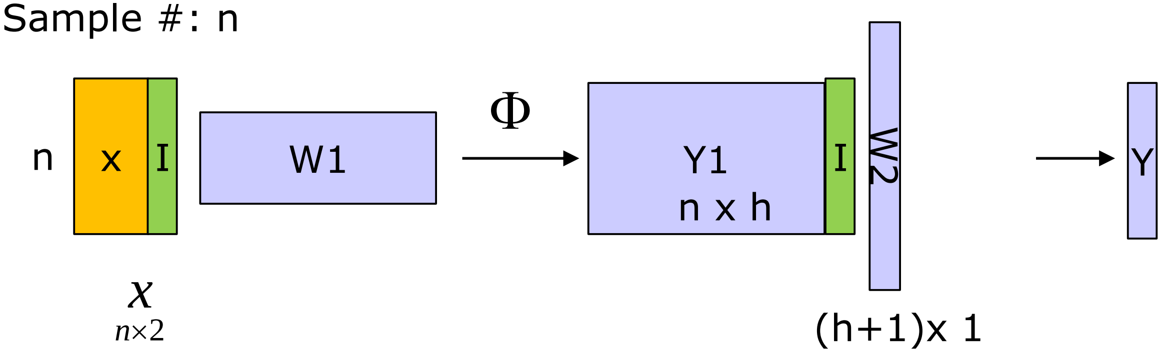
- Remind that Regression Example used Multiple Samples.
- We build NN with Multiple Samples, not with one sample

XOR NN



$$Y_1 = \Phi(X_1)$$

Sample #: n



$$\underset{n\times 2}{x} \downarrow$$

$$\underset{n\times 3}{X} = [x \quad I] \rightarrow \underset{3\times h}{W_1} \xrightarrow{\Phi} \underset{n\times h}{Y_1}$$

$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times 1}{W_2} \longrightarrow \underset{n\times 1}{Y}$$

$$\Phi = \text{logsig}$$

x

T	T
T	F
F	T
F	F

y

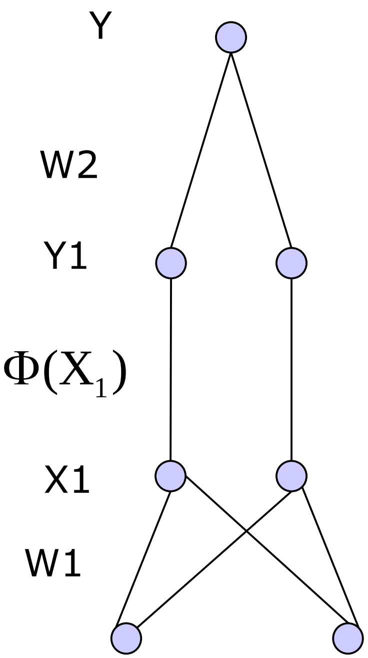
F
T
T
F

e= y

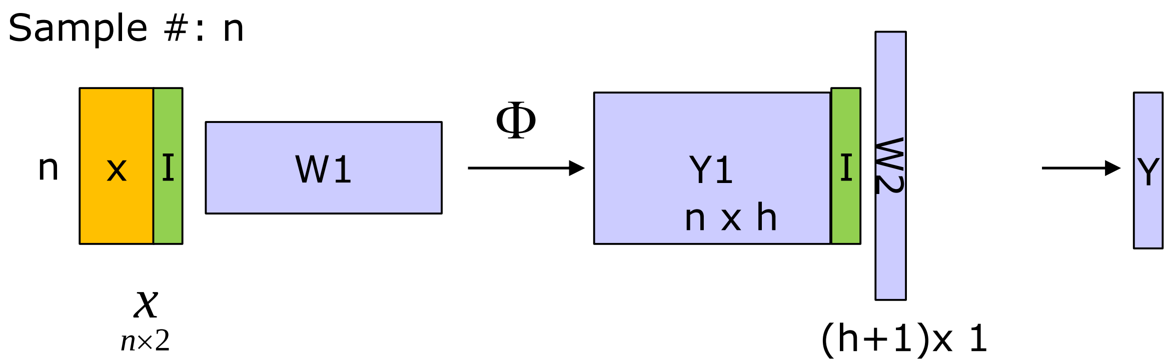
F
T
T
F

\- Y

XOR NN



$$Y_1 = \Phi(X_1)$$



$$\underset{n\times2}{x}$$

$$\underset{n\times3}{X} = [x \quad I] \rightarrow \underset{3\times h}{W_1} \xrightarrow{\Phi} \underset{n\times h}{Y_1}$$

$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times1}{W_2} \longrightarrow \underset{n\times1}{Y}$$

$$\Phi = \text{logsig}$$

n

T	T	1
T	F	1
F	T	1
F	F	1

x

h

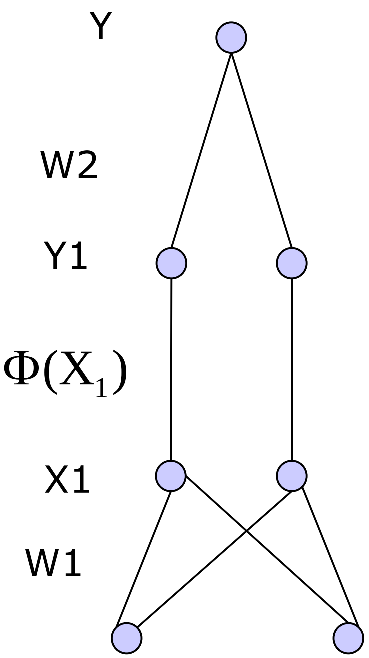
w11	w12	w13	w14	w15
w21	w22	w23	w24	w25
w31	w32	w33	w34	w35

W1

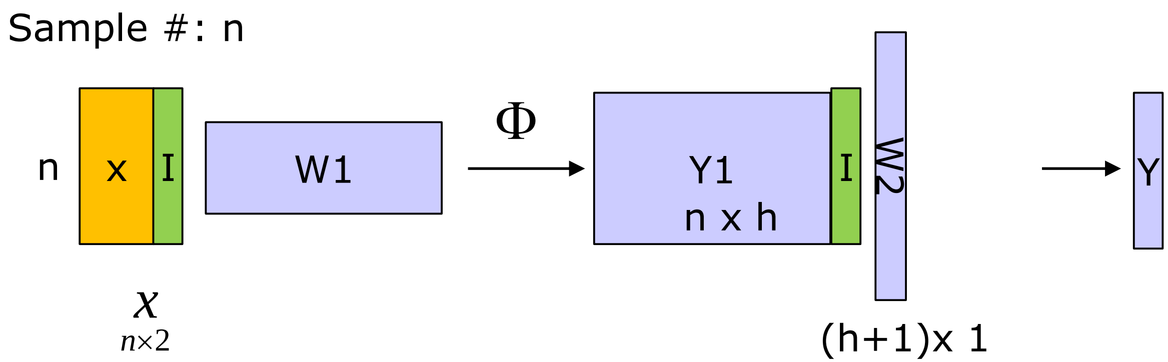
=

Tw11 +Tw21 +w31	Tw12 +Tw22 +w32	Tw13 +Tw23 +w33	Tw14 +Tw24 +w34	Tw15 +Tw25 +w35
Tw11 +Fw21+w 31	Tw12 +Fw22 +w32	Tw13 +Fw23 +w33	Tw14 +Fw24 +w34	Tw15 +Fw25 +w35
..	…	…	…	…

We know the Patterns from Sigmoidal Example



$$Y_1 = \Phi(X_1)$$



$$\underset{n\times 2}{x} \downarrow$$

$$\underset{n\times 3}{X} = [x \quad I] \rightarrow \underset{3\times h}{W_1} \xrightarrow{\Phi} \underset{n\times h}{Y_1}$$

$$\underset{n\times(h+1)}{[Y_1 \quad I]} \rightarrow \underset{(h+1)\times 1}{W_2} \longrightarrow \underset{n\times 1}{Y}$$

$$\Phi = \text{logsig}$$

$$J = \frac{1}{2}\sum_k (y_k - \hat{y}_k)^2 = \frac{1}{2}\sum_k {e_k}^2 = \frac{1}{2} e^T e$$

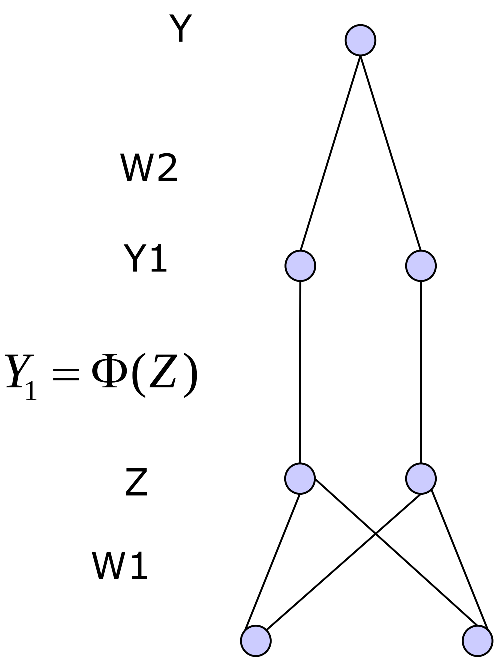
$$Matrix: \frac{\partial J}{\partial W_1} = -[x \quad I]^T \left(e W_{2,h\times 1}^T \circ \Phi'\right) \Rightarrow$$

$$\left.\frac{\partial J}{\partial W_1}\right|_{3\times h} = -\left([\underset{n\times 2}{x} \quad \underset{n\times 1}{I}]_{n\times 3}\right)^T \left(\left(e_{n\times 1} W_{2,h\times 1}^T\right)_{n\times h} \circ \Phi'_{n\times h}\right)$$

$$Vector: \frac{\partial J}{\partial W_2} = -[Y_1 \quad I]^T e$$

$$\left.\frac{\partial J}{\partial W_2}\right|_{(h+1)\times 1} = -\left([\underset{n\times h}{Y_1} \quad \underset{n\times 1}{I}]^T\right)_{(h+1)\times n} e_{n\times 1}$$

Matlab XOR Example



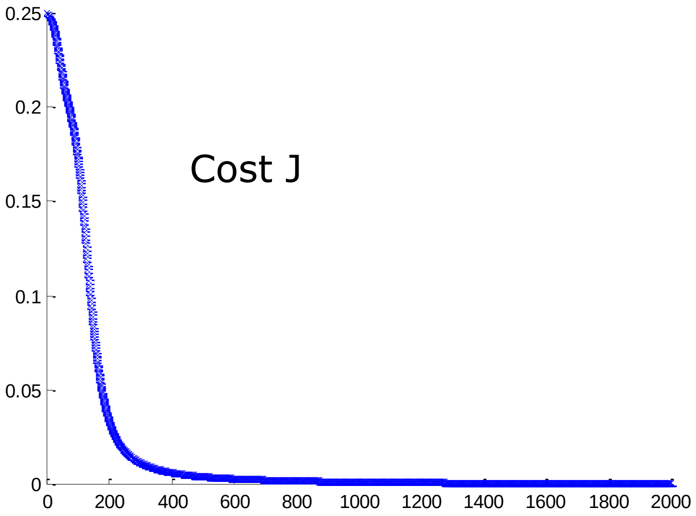

```
p=[0  0
   1  0
   0  1
   1  1
   ];

t = [ 0 1 1 0 ]';

in = 2;  % input node
hn = 2; % node at hidden
on = 1;  % output node
```

$$W_1 \qquad W_2$$
$$3\times 2 \qquad 3\times 1$$

XOR circuit is designed
with two hidden nodes

This case uses only two hidden states.
It is the minimum condition for XOR

Matlab XOR Example




```
E =

   -0.0243    0.0235    0.0235   -0.0283
```

```
W1 =

    5.1002   -3.2019
    5.1002   -3.2019
   -2.0101    4.7616
```

```
W2 =

   11.1509
   11.6657
  -16.5765
```

```
for k=1:data
   uk = p(k,:)';
   yk = t(k,:)';

    % Forward
   X1 = ([ uk' 1]*W1)';
   Y1 = logsig(X1);
   X2 = ([ Y1' 1]*W2)';
    Y2 = logsig(X2);

    % Error
    E(k) = yk-Y2;

    % Get dW
    dsig2      = Y2.*(1-Y2);
   dY2_dW2 = [ Y1 ;1]*dsig2';
   dsig1     = dsig2*Y1.*(1-Y1);
    dY1_dW1= -[ uk ;1]*dsig1';

    dW1  = dW1-eta/data*E(k)*dY1_dW1;
    dW2  = dW2-eta/data*E(k)*dY2_dW2;
```