LE VAN HUNG (黎文雄)

Student ID: 0860831

Deep Learning - Homework 0

Exercise 1:

a. Train the coefficients of the polynomial function:

$$y(x,w) = w_0 + w_1 x + w_2 x^2 + ... + w_m x^m$$

Step1: Load data set → split data for training set and testing set

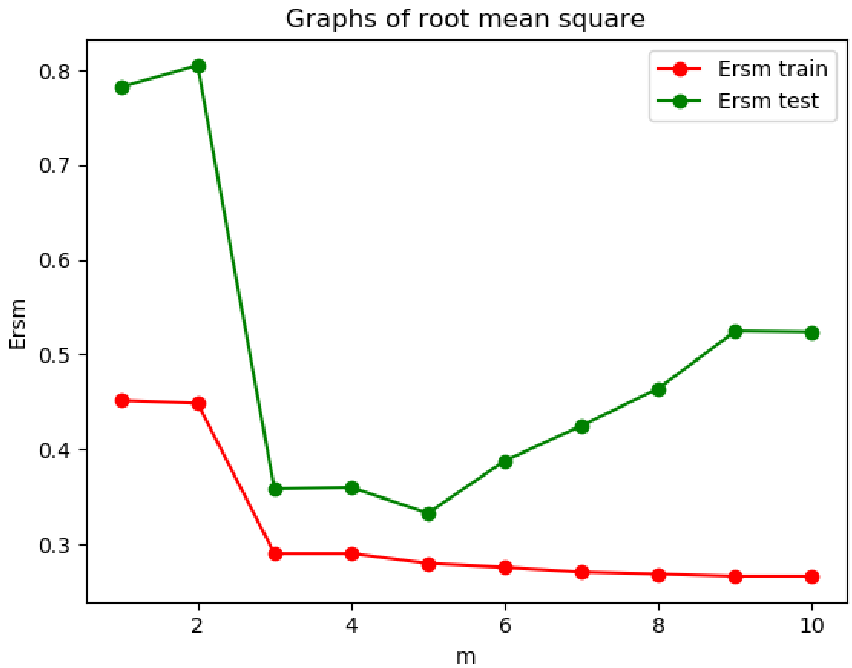
Step2: Calculate weight by using normal equation

- $X = \begin{bmatrix} x_1^{\ 0} & x_1^{\ 1} & ... & x_1^{\ m} \\ x_2^{\ 0} & x_2^{\ 1} & ... & x_2^{\ m} \\ ... & ... & ... & ... \\ x_n^{\ 0} & x_n^{\ 1} & ... & x_n^{\ m} \end{bmatrix}$
- $\text{weight} = (\mathrm{X}^T X)^{-1} X^T y$ here y is corresponding target value

Step3: Calculate E_{RSM}

- $E(\mathrm{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{y\left(x_n, weight - t_n\right)\right\}^2$
- $E_{RSM} = \sqrt{2E / N}$

Step4: plot the figure of root-mean-square error depend on degree of polynomial function (m = 1 →10)



b. Training coefficient with various regularization coefficient λ

Step1: Calculate weight by using regularization

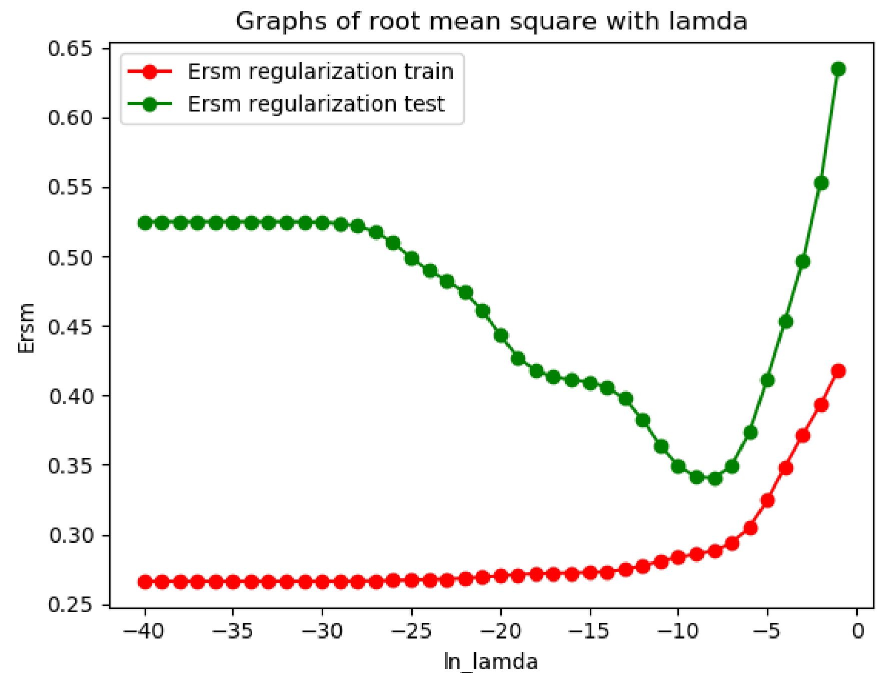
- $$X = \begin{bmatrix} x_1^{\ 0} & x_1^{\ 1} & \dots & x_1^{\ m} \\ x_2^{\ 0} & x_2^{\ 1} & \dots & x_2^{\ m} \\ \dots & \dots & \dots & \dots \\ x_n^{\ 0} & x_n^{\ 1} & \dots & x_n^{\ m} \end{bmatrix}$$

- $\text{weight} = (\mathrm{X}^T X + \lambda I)^{-1} X^T y$ here y is corresponding target value

Step2: Calculate E_{RSM}

- $E_{RSM} = \sqrt{2E / N}$

Step3: with each λ, try to do Step1 and Step2 again and plot graph of root mean square depend on $\ln\lambda$ ($\ln\lambda$ = -10 → 0)



Exercise 2:

Training the coefficients of the following polynomial function

$$y(x,w) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

Step1: load data from data.mat file and split data. Use first 40 samples of each class for training and last 10 samples of each class for testing

Step2: Set up matrix X

$$X = \begin{bmatrix} 1 & x_1^1 & \dots & x_4^1 & x_1^1 x_1^1 & \dots & x_4^1 x_4^1 & x_1^1 x_1^1 x_1^1 & \dots & x_4^1 x_4^1 x_4^1 \\ 1 & x_1^2 & \dots & x_4^2 & x_1^2 x_1^2 & \dots & x_4^2 x_4^2 & x_1^2 x_1^2 x_1^2 & \dots & x_4^2 x_4^2 x_4^2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots\dots \\ 1 & x_1^n & \dots & x_4^n & x_1^n x_1^n & \dots & x_4^n x_4^n & x_1^n x_1^n x_1^n & \dots & x_4^n x_4^n x_4^n \end{bmatrix}_{nxm}$$

Here: - n is number of data point

- $m = 1 + D + D^2 + D^3$

Step3: Calculate weight by using normal equation

$$\text{weight} = (X^T X)^{-1} X^T y$$ here y is corresponding target value

Step4: Calculate E_{RSM}

- $$E(\mathrm{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{ y\left(x_n, weight - t_n\right)\right\}^2$$
- $$E_{RSM} = \sqrt{2E/N}$$

```
weight=
[[  8.07692332  -6.84725349   2.12779874   7.29239444 -10.72044166
    0.55520822   0.82094265   1.149691    -2.91747342   0.82081316
   -1.03418235  -3.06400158   6.16405432   1.14939053  -3.06405305
   -2.63528857   4.61946965  -2.91733199   6.16376786   4.61915252
   -6.2379328    0.10000033  -0.14622713  -0.24797724   0.44114968
   -0.14624464   0.09640282   0.1935125   -0.24054802  -0.24805571
    0.19336511   0.28120081  -0.44878721   0.44121318  -0.24054778
   -0.44883269   0.65273815  -0.14624464   0.09640282   0.1935125
   -0.24054801   0.09641454  -0.0453777    0.01151567  -0.26069199
    0.19348727   0.01152397  -0.09017744   0.05207061  -0.24055479
   -0.26068315   0.0520708    0.06133339  -0.24803896   0.19330026
    0.28126877  -0.44880401   0.19348727   0.01152397  -0.09017744
    0.05207061   0.2813655   -0.09025069  -0.14806395   0.12150728
   -0.44884141   0.0520425    0.12156113   0.21119985   0.44117565
   -0.24053581  -0.44880311   0.65273256  -0.24055479  -0.26068533
    0.05207152   0.06133339  -0.44884033   0.05204457   0.12144905
    0.21119571   0.65273257   0.06133253   0.21120711  -1.74062434]]
the mean square error of training set:  0.1628032581192223
the mean square error of testing set:  0.19715308113373076
```

Exercise 3: implement Bayesian linear regression

a. Plot 5 sample curves of the function y(x,w) for data size N=1,2,4,25,80,100

Step1: Calculate the maximum posterior weight.

The posterior distribution is given by $p(w|t) = N(w|m_N, S_N)$

Here:
$$m_N = \beta S_N \Phi^T t$$
$$S = inv(\alpha I + \beta \Phi^T \Phi)$$

$$\Phi = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \cdots & \phi_{M-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_N) & \phi_1(x_N) & \cdots & \phi_{M-1}(x_N) \end{bmatrix}$$

Because posterior distribution is Gaussian, so the maximum posterior weight vector is simply given by $weight = m_N = \beta S_N \Phi^T t$

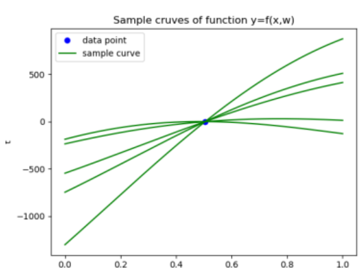
step2: Plot 5 sample curves of the function y(x,w)

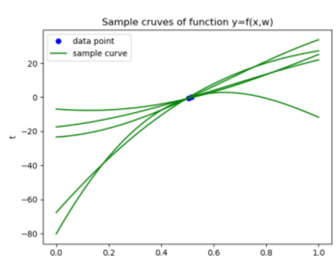
use the following function:

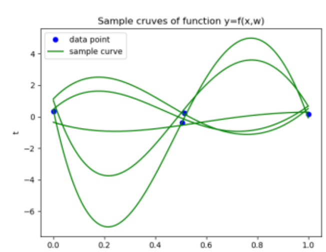
numpy.random.multivariate_normal (mean_vector, covariance_matrix, number_sample)

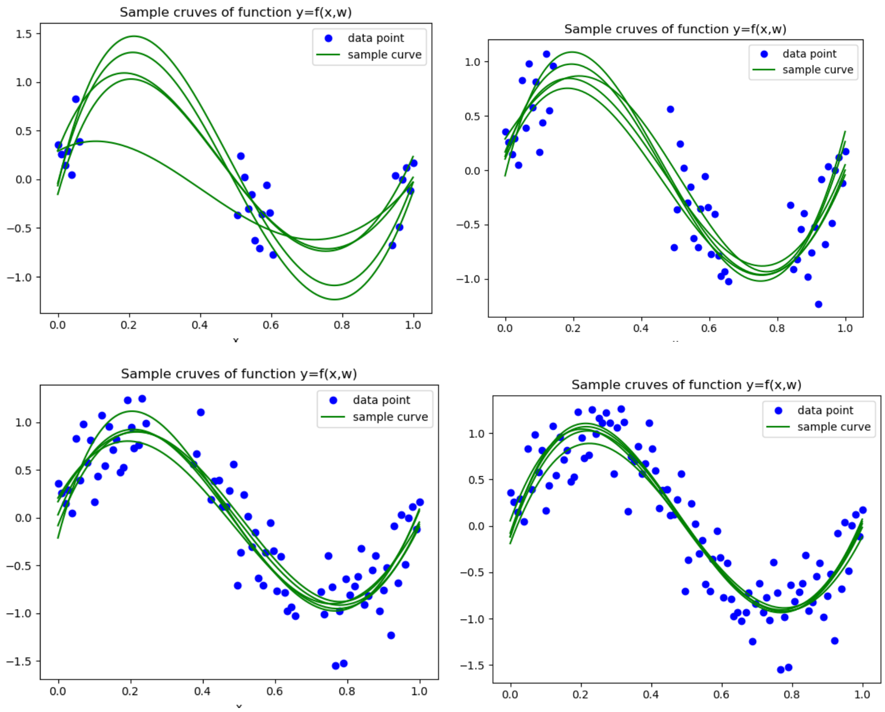
we obtain sample_weight(9x5) with each row of this matrix include a sample weight vector.

By using dataset with size of data increased gradually (N=1,2,4,25,50,80,100) I obtain the figures below.









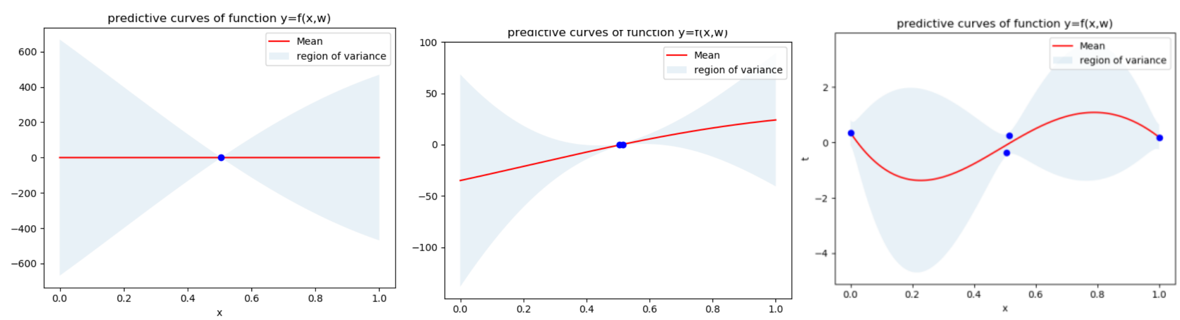
b. Show the mean curve and the region of variance

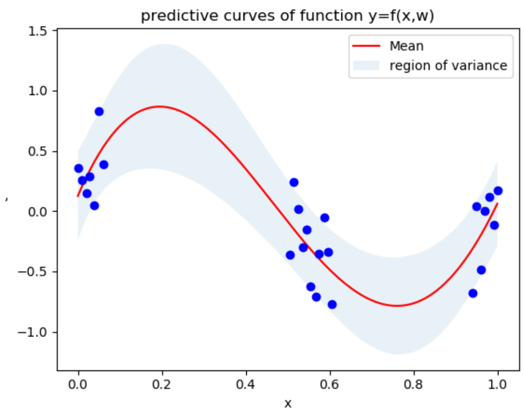
The predictive distribution defined by:

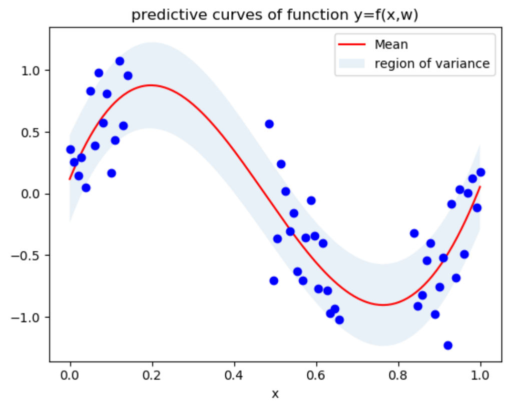
$$p(t \mid x, \tau, \alpha, \beta) = N(t \mid m_N^T \phi(x), \sigma_N^2(x))$$

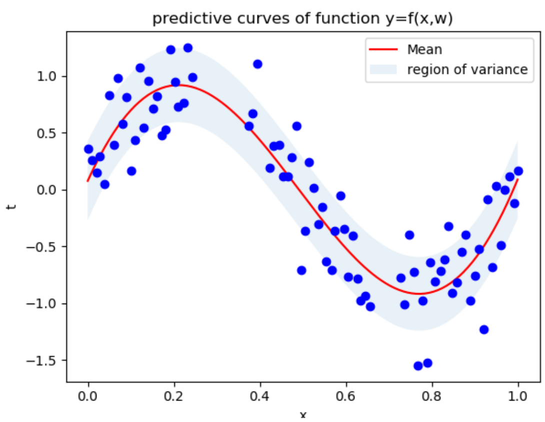
Here, the variance is given by: $\sigma_N^2(x) = \dfrac{1}{\beta} + \phi(x)^T S_N \phi(x)$

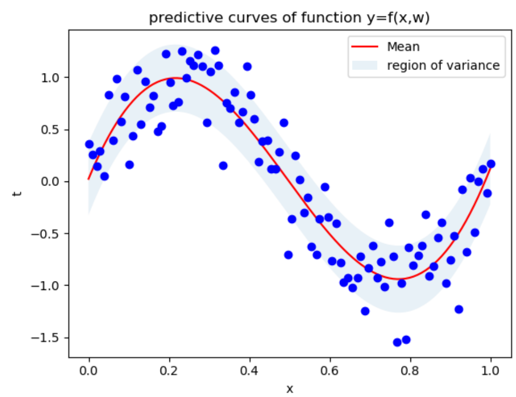
By using dataset with size of data increased gradually (N=1,2,4,25,50,80,100) I obtain the figures below.





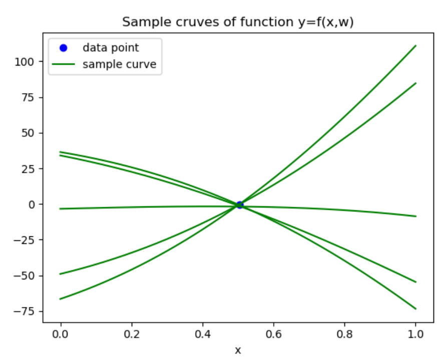


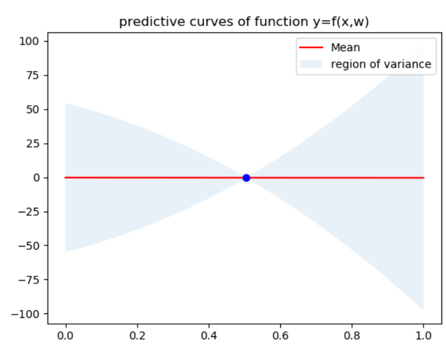


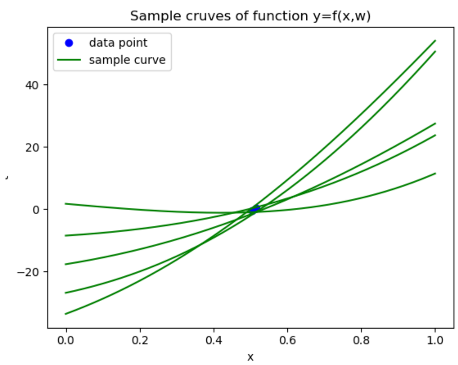


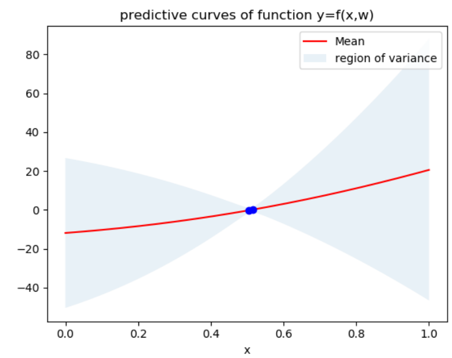
c. Repeat (a) and (b) using a different set of basis functions.

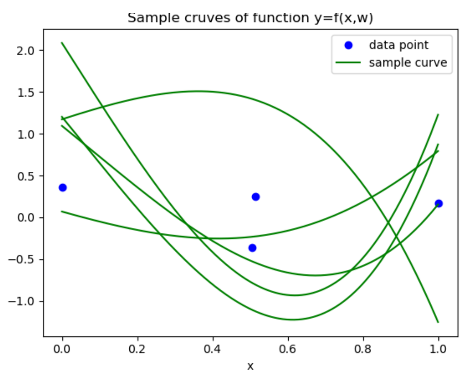
By using dataset with size of data increased gradually (N=1,2,4,25,50,80,100) I obtain the figures below.

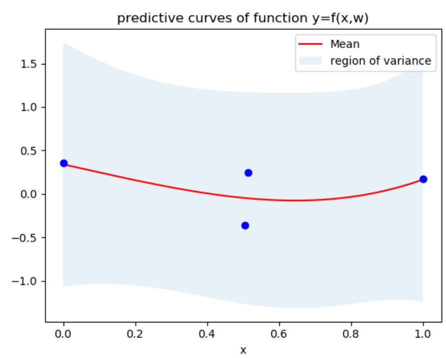


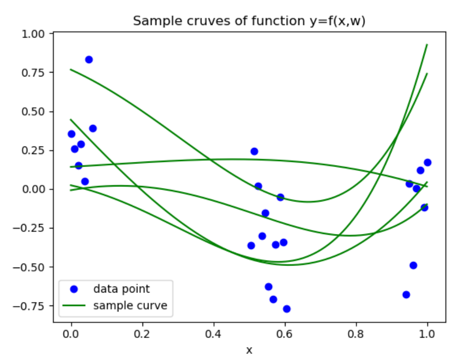


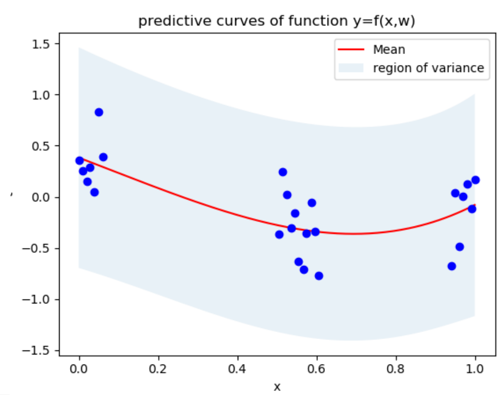

Sample cruves of function y=f(x,w)
data point
sample curve
x

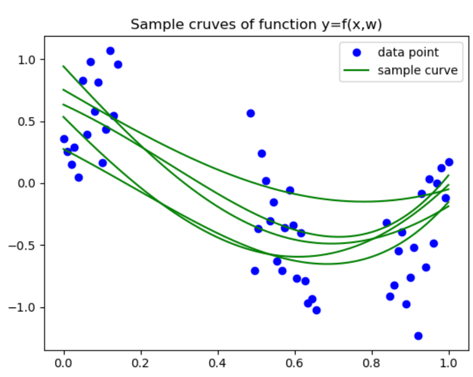

predictive curves of function y=f(x,w)
Mean
region of variance
x

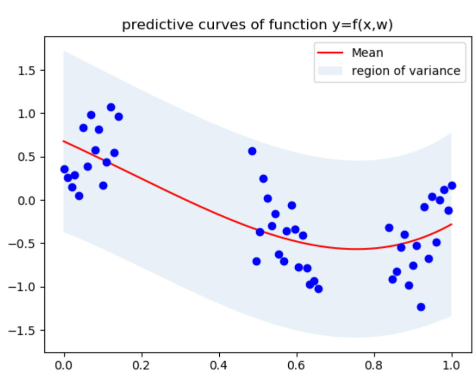

Sample cruves of function y=f(x,w)
data point
sample curve
x

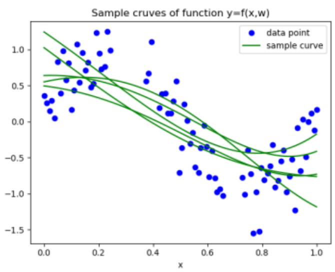

predictive curves of function y=f(x,w)
Mean
region of variance
x

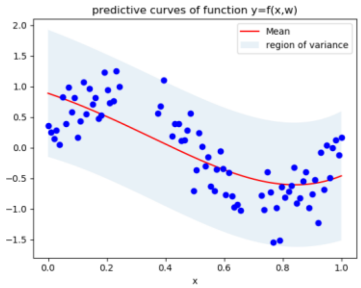

Sample cruves of function y=f(x,w)
data point
sample curve
x


predictive curves of function y=f(x,w)
Mean
region of variance
x


Sample cruves of function y=f(x,w)
data point
sample curve
x


predictive curves of function y=f(x,w)
Mean
region of variance
x


Sample cruves of function y=f(x,w)
data point
sample curve
x


predictive curves of function y=f(x,w)
Mean
region of variance
x