**Subject Name: Machine Learning, Mid-Sem, MM:40 Mar, Batch 1**

**Question 1 : [12.5 Points] Answer the following :**

a) **[5 Points]:** Show that the maximum likelihood solution is equivalent to the least square solution. Also determine the conditions which follow this inference.

b) **[2.5 Points]**: How Least square or maximum likelihood solution changes by the introduction of weight decay regularizer ? Please also comment on Lasso regularizer and its geometric interpretation with respect to weight decay regularizer.

c) **[5 Points]**: Show that predictive distribution in Bayesian linear regression is given by the following expression :

$$p(t \mid \boldsymbol{x}, \boldsymbol{t}, \alpha, \beta) = N(t \mid \boldsymbol{m}_N^T \boldsymbol{\phi}(\boldsymbol{x}), \sigma_N^2(\mathrm{x})) \qquad \text{where} \quad \sigma_N^2(\boldsymbol{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\boldsymbol{x})^T S_N \boldsymbol{\phi}(\boldsymbol{x})$$

Please also show how you arrive at the variance as aforementioned.

**Question 2 : [12.5 Points] Python Exercise on Gaussian Process Regression (GPR) for posterior prediction.**

Generate 10 data points (these points will serve as training data points) with negligible noise (corresponds to noiseless GP regression). Use the following python function with default noise variance.

```
import numpy as np
def generate_noisy_points(n=10, noise_variance=1e-6):
    np.random.seed(777)
    X = np.random.uniform(-3., 3., (n, 1))
    y = np.sin(X) + np.random.randn(n, 1) * noise_variance**0.5
    return X, y
```

**Write Python script for the following questions :**

a) **[1.5 Points]** Plot the variation of X wrt y

b) **[4 Points]** Generate 100 test data points and Draw 10 function samples from the GP prior distribution. Show the mathematical representation of GP prior distribution.

**c)** **[7 Points]** Compute and Plot the GP posterior distribution given the original (training) 10 data points. Use the RBF kernel function. Please also show the mathematical representation of posterior distribution.

**Question 3 : [15 Points]**

**a) [5 Points]**

**Question:** Derive the gradient descent training rule assuming that the target function representation is:

$o_d = w_0 + w_1 x_1 + \ldots + w_n x_n.$

Define explicitly the cost/error function E, assuming that a set of training examples D is provided, where each training example $d \in D$ is associated with the target output $t_d$.

**b) [5 Points]**

**Question:** Prove that the LMS training rule performs a gradient descent to minimize the cost/error function E defined in **3 a)**

**c) [5 Points]**

**Question:** Explain the principle of the gradient descent algorithm. Accompany your explanation with a diagram. Explain the use of all the terms and constants that you introduce and comment on the range of values that they can take.