Machine Learning	Dr. Mario Fritz and Dr. Björn Andres
Exercise Sheet 4 - 24.11.2014	due: 01.12.2014

Exercise 12 - Regression in Practice

We consider the regression problem where the input $\mathrm{X} \in \mathbb{R}^d$ and the output $\mathrm{Y} \in \mathbb{R}$. We use either L_1 or L_2-loss.

a. **(2 points)** In the following you have to implement least squares and ridge regression (both L_2-loss).

- `w = LeastSquares(Designmatrix,Y)`:
  - input: design matrix $\Phi \in \mathbb{R}^{n \times D}$ and the outputs $Y \in \mathbb{R}^n$ (column vector)
  - output: weight vector w of least squares regression as column vector
- `w = RidgeRegression(Designmatrix,Y,Lambda)`:
  - input: the design matrix $\Phi \in \mathbb{R}^{n \times D}$, the outputs $Y \in \mathbb{R}^n$ (column vector), and the regularization parameter $\lambda \in \mathbb{R}^+$
  - output: weight vector w of ridge regression as column vector. Use the non-normalized version: $w = (\Phi^T \Phi + \lambda \mathbb{1}_D)^{-1} \Phi^T Y$.

Note that the matlab code for L_1-loss (with and without L_2-regularizer) is provided in the zip-file. It requires the installation of CVX - you can find a link on the course webpage.

b. **(1 Point)** Write a Matlab function `Basis`(X, k):

- input: the input data matrix $X \in \mathbb{R}^{n \times 1}$ and the maximal frequency k of the Fourier basis.
- output: the design matrix $\Phi \in \mathbb{R}^{n \times (2k+1)}$ using the Fourier basis functions:

$$\phi_0(x) = 1, \qquad \phi_{2l-1}(x) = \frac{1}{l} \cos(2\pi l x), \qquad \phi_{2l}(x) = \frac{1}{l} \sin(2\pi l x), \quad l = 1, \ldots, k.$$

c. In the first example we have only one feature, thus we want to learn a function $f : \mathbb{R} \to \mathbb{R}$. First plot the training data (`plot(Xtrain, Ytrain,'.');`).

- **(2 Points)** Which loss function (L_1 or L_2) is more appropriate for this kind of data ? Justify this by checking the data plot. Use in the next part only the regression method with your chosen loss (that is either ridge regression or L_1-loss with L_2-regularizer).
- **(4 Points)** Use the basis functions with $k = 1, 2, 3, 5, 10, 15, 20$ from part b) to fit the regularized version of the loss chosen in the previous part. Use regularization parameter $\lambda = 10$. Plot the resulting functions (use `x = 0 : 0.01 : 1`) for all values of k together with the training data,

$$f_k(x) = \langle \phi(x), w^k \rangle = \sum_{i=1}^{2k+1} w_i^k \phi_i(x).$$

Save the plots using the command `saveas(gcf, 'PlotFunctions', 'png')`.

Compute the loss, that is

$$\frac{1}{n} \sum_{i=1}^{n} L(Y_i, f(X_i)),$$

on the training and test data (variable names `trainloss` and `testloss`) and plot training and test loss as a function of k. Save the plots using the command `saveas(gcf, 'PlotLoss', 'png')`. Save your training and test loss in a file `LossFirstEx`, use

```
save LossFirstEx trainloss testloss
```

Repeat the same for $\lambda = 0$ (unregularized version) - append a 0 to all filenames e.g. `LossFirstEx0`.

How does increasing k affect the estimated functions f_k ? What is the difference in terms of k of the regularized and unregularized regression method. The last two questions have to be answered on paper.

d. The second example is a real dataset. The task is to predict the total number of violent crimes per 100K population (output variable $Y \in \mathbb{R}$) from a set of features (input variables $X \in \mathbb{R}^{99}$) capturing all sorts of properties of the cities and their population.

- **(2 Points)** Use a linear design with an offset, that is $f(x) = \langle w, x \rangle + b$, (add a feature which is 1 for every data point, `X = [X, 1(size(X, 1), 1)]`) and fit the data using least squares regression. Compute the training loss, that is $\frac{1}{n} \sum_{i=1}^{n} \|Y - X * w\|_2^2$, on the training set. Save this `trainloss` with `saveLossSecondExtrainloss`.
- **(2 Points)** You are now free to use any set of basis functions and any regression method. Write a function `Prediction2(X)` which given a set of testpoints - a matrix $X \in \mathbb{R}^{n \times 99}$ outputs the predictions of your chosen learning method as a column vector $f \in \mathbb{R}^{n \times 1}$. For the best results on our hidden test set (which have to be better than the results of linear least squares) we have the following prizes
  1. **(10 Bonus Points)** for the winner
  2. **(5 Bonus Points)** for the second best prediction
  3. **(3 Bonus Points)** for the third best prediction

Zip the m-files (`Basis` etc.), your plots (png files) and the matlab data files (.mat) and send them in one file by email with the subject "[ML] EX4 Group[A/B/C]" to your respective tutor until the lecture on Monday. The zip file should also contain a README file with the names and matriculation numbers of those submitting the assignment.

Hints:

- In order to have several plots in one figure you have to use
  `figure, hold on`
  ... all your plotting commands ...
  `hold off`
- `gcf` is a handle to the current figure (save the figure just after it was created).
- In order to distinguish the curves for different values of k draw them in different colors using:
  `Colors = jet(NVals),`
  `for k = 1 : NVals, plot(x, Output(:, k), 'linestyle', '-', 'color', Colors(k, :)), end`
  where `Output(:, k)` is a matrix containing the estimated function values at `x`.
- `norm(x)` computes the Euclidean norm of a vector `x`.
- Linear system, `Ax = b`, can be solved in Matlab using the backslash operator `x = A\b`.
- More details on the features for the second task can be found in the data-file.

Exercise 13 - Derivation of a dual problem

Let $(x_i, y_i)_{i=1}^n$ be a training sample for a binary classification task, that is $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$. The so-called hard-margin Support Vector Machine (SVM) without offset corresponds to the optimization problem

$$\begin{aligned} \min_{w \in \mathbb{R}^d} \quad & \frac{1}{2} \|w\|_2^2 \\ \text{subject to :} \quad & y_i \langle w, x_i \rangle \geq 1, \qquad i = 1, \ldots, n \end{aligned}$$

a. **(3 Points)** Derive the dual problem.

b. **(1 Point)** Which problem, dual or primal, would you solve depending on n (number of training samples) versus d (number of features) ?

Hints:

- Note that inequality constraints have the form $g(x) \leq 0$