Machine Learning

Summer Semester 2017, Homework 4

Prof. Dr. J. Peters, M. Ewerton, S. Parisi

TECHNISCHE UNIVERSITÄT DARMSTADT

Total points: 55 + 12 bonus
Due date: Tuesday, 18 July 2017 (before the lecture)

Name, Surname, ID Number

Problem 4.1 Support Vector Machines [25 Points + 4 Bonus]

In this exercise, you will use the dataset `iris-pca.txt`. It is the same dataset used for Homework 3, but the data has been pre-processed with PCA and only two kinds of flower ('Setosa' and 'Virginica') have been kept, along with their two principal components. Each row contains a sample while the last attribute is the label (0 means that the sample comes from a 'Setosa' plant, 2 from 'Virginica'). (You are allowed to use built-in functions for computing the mean, the covariance, eigenvalues, eigenvectors and for quadratic programming.)

a) **Definition [3 Points]**

Briefly define SVMs. What is their advantage w.r.t. other linear approaches we discussed this semester?

b) **Quadratic Programming [2 Points]**

Formalize SVMs as a constrained optimization problem.

c) **Slack Variables [2 Points]**

Explain the concept behind slack variables and reformulate the optimization problem accordingly.

d) **Slack Variables [7 Points]**

Solve the optimization problem of the previous question. Show all the intermediate steps and write down the final solution.

e) **The Dual Problem [4 Bonus Points]**

What are the advantages of solving the dual instead of the primal?

f) **Kernel Trick [3 Points]**

Explain the kernel trick and why it is particularly convenient in SVMs.

g) **Implementation [8 Points]**

Learn an SVM to classify the data in `iris-pca.txt`. Choose your kernel. Create a plot showing the data, the support vectors and the decision boundary. Show also the misclassified samples. Attach a snippet of your code.

Problem 4.2 Neural Networks [20 Points + 8 Bonus]

In this exercise, you will use the dataset `mnist_small`, divided into four files. The *mnist* dataset is widely used as benchmark for classification algorithms. It contains 28x28 images of handwritten digits (pairs `<input, output>` correspond to `<pixels, digit>`).

a) **Multi-layer Perceptron [20 Points]**

Implement a neural network with one hidden layer and train it using backpropagation on the provided dataset. Choose your loss function, activation function and optimizer, briefly explaining your choices. You can use off-the-shelf optimizers, loss and activation functions, but you have to write the network structure and the back-propagation algorithm by yourself. You are also free to choose a suitable number of neurons for the hidden layer.

Show how the misclassification error (in percentage) on the testing set evolves during the learning. Attach snippets of your code.

Hint: it can be helpful to check the network parameters or to plot the trend of your loss function if your network does not learn.

b) **Deep Learning [8 Bonus Points]**

In recent years, deep neural networks have become one of the most used tools in machine learning. Highlight the qualitative differences between classical neural networks and deep networks. Which limitations of classical NN does deep learning overcome? Give an intuition of the innovations introduced in deep learning compared to traditional NN. (Hint: Have a look at this paper. Search on Google Scholar and other scientific papers for more insights.)

Problem 4.3 Gaussian Processes [10 Points]

a) **GP Regression [10 Points]**

Implement a Gaussian Process to fit the target function $y = \sin(x) + \sin^2(x)$ with $x \in [0, 0.005, 0.01, 0.015, \ldots, 2\pi]$. Use a squared exponential kernel, an initial mean of 0 and assume a noise variance of 0.001. Begin with no target data points and, at each iteration, sample a new point from the target function according to the uncertainty of your GP (that is, sample the point where the uncertainty is the highest) and update it. Plot your GP (mean and two times standard deviation) after iterations 1, 2, 4, 8 and 16. In each figure, plot also the true function as ground truth and add a new marker for each new sampled point. Attach a snippet of your code.