Statistical Machine Learning: Exercise 4

Support Vector Machines, Gaussian Processes, and Neural Networks

Prof. Georgia Chalvatzaki, MSc. Snehal Jauhri, MSc. Ali Younes

TECHNISCHE UNIVERSITÄT DARMSTADT

Summer Term 2022

Due date: August 9th, 23:59 PM

Task 1: Support Vector Machines (35 Points)

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 `numpy` functions (e.g., `numpy.linalg.eig`). For quadratic programming we suggest `cvxopt`.

1a) Definition (3 Points)

Briefly define how SVMs work and what are they used for. What is their advantage w.r.t. other linear approaches we discussed this semester?

1b) Quadratic Programming (2 Points)

Formalize SVMs as a constrained optimization problem.

1c) Slack Variables (2 Points)

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

1d) Optimization with Slack Variables (8 Points)

Solve the optimization problem of the previous question, i.e. derive the optimal solution for the parameters and how you can use them to classify. Show all the intermediate steps and write down the final solution.

1e) The Dual Problem (4 Points)

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

1f) Kernel Trick (6 Points)

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

1g) Implementation (10 Points)

Implement and learn an SVM to classify the data in `iris-pca.txt`. Choose your kernel and decide whether to use slack variables or not. Create a plot showing the data, the support vectors and the decision boundary. Show also the misclassified samples. Attach a snippet of your code.

Task 2: Gaussian Processes (20 Points)

2a) GP Regression (10 Points)

Implement a Gaussian Process to fit the target function $y = 2\cos(x) + \sin^2(x)$ 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 0.005. 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 the standard deviation) after iterations 1, 2, 5, 10 and 15. 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.

2b) GP Posterior Update Derivation (10 Points)

Let us suppose that the target values $t_N = (t_1, ..., t_N)^T$, corresponding to input values $x_1, ..., x_N$, comprise the observed training set, and our goal is to predict the target variable t_{N+1} for a new input vector x_{N+1}. This requires evaluating the predictive distribution $p(t_{N+1}|t_1, .., t_N, x_1, ..., x_N, x_{N+1})$. Starting from the joint distribution over target points $p(t_1, ..., t_N, t_{N+1})$, derive the mean and the covariance of the required conditional distribution.

Task 3: Neural Networks (35 Points)

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>`).

3a) Multi-layer Perceptron (30 Points)

Implement a neural network and train it using backpropagation on the provided dataset. Choose your loss and activation functions and your hyperparameters (number of layers, neurons, learning rate, ...), briefly explaining your choices. You **cannot** use any library beside `numpy`! That is, you have to implement by yourself the loss and activation functions, the backpropagation algorithm and the stochastic gradient descent optimizer. Experiment around with hyperparameters and check what works and what does not.

Show how the misclassification error (in percent) on the testing set evolves during learning. An acceptable solution achieves an error of 8% or less. Attach snippets of your code.

Hint: If your network does not learn, check how the network parameters change and plot the trend of your loss function. You will get full points, if you include a thorough discussion of the observed problems.

3b) Perceptron vs Logistic Regression vs MLP (5 Points)

Compare perceptron, Logistic regression and the Multi-layer perceptron. Discuss when to use each of them and the key properties.

3c) [Bonus] Skip-connections and Stochastic gradient descent (10 Points)

Skip-connections are connections in the neural network that skip some of the layers in the neural network and feeds the output of one layer as the input to the next layers. Modify the neural network by adding skip connections. Discuss the results. When skip-connections are useful.