In the Name of God

Convex Optimization I (25756-1)

CHW 2

Fall Semester 1402-03

Department of Electrical Engineering

Sharif University of Technology

Instructor: Dr. R. Amiri

Due on December 30, 2023 at 23:59

Introduction

In this assignment, we will explore the application of Support Vector Machines (SVM) and Neural Networks (NN) in convex optimization. SVM is a popular machine learning algorithm used for classification and regression tasks. It is based on the idea of finding a hyperplane that separates different classes while maximizing the margin between them. Neural networks, a class of machine learning models inspired by the human brain's structure, have gained widespread use for various tasks, including image recognition, natural language processing, and regression problems. Unlike SVM, neural networks are not inherently convex, and their training involves iterative optimization methods.

1- Soft-Margin SVM and CVXPY

In cases where the data is not perfectly separable, a soft-margin SVM introduces a margin of tolerance, allowing for some misclassifications. This is achieved by introducing slack variables. The optimization problem for soft-margin SVM can be formulated as follows:

1. **Primal Formulation for Soft-Margin SVM:**
   Consider a soft-margin SVM for linearly separable data with slack variables $\xi_i \geq 0$:

$$
\begin{aligned}
\text{Minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\
\text{Subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, N \\
& \xi_i \geq 0, \quad i = 1, 2, \ldots, N
\end{aligned}
$$

   Here, $\mathbf{w}$ represents the weight vector, b is the bias term, and C is the regularization parameter controlling the trade-off between a wide margin and misclassification. Load the svm_train dataset. This dataset has 863 rows and 3 columns. The first two columns represent features and the third column represents labels associated with the features. Solve Soft-Margin SVM using CVXPY with $C = 1$. Report the optimal $\mathbf{w}$ and b.

2. **Dual formation of Soft-Margin SVM:**
   Show that the Lagrange dual function could be written as:

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2C}\boldsymbol{\alpha}^T \mathbf{Y}(\mathbf{X}\mathbf{X}^T)\mathbf{Y}\boldsymbol{\alpha}$$

   subject to the constraints:

$$0 \leq \alpha_i \leq C \quad \text{for } i = 1, 2, \ldots, n$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

   where:

   - $\boldsymbol{\alpha}$ is the vector of Lagrange multipliers,
   - $\mathbf{Y}$ is a diagonal matrix with elements y_i,
   - $\mathbf{X}$ is a matrix with rows $\mathbf{x}_i$.

   Now, Solve the dual optimization problem using CVXPY. Check whether the optimized $\mathbf{w}$ and b are the same as before or not.

2- SVM in Machine Learning

Convex optimization plays a crucial role in training Support Vector Machines. The primal and dual formulations of the SVM optimization problem are inherently convex, allowing for efficient and guaranteed optimization. For this section, you are required to use scikit-learn library in Python. Implement an SVM classifier using scikit-learn and perform the following tasks:

1. **Data Preparation:** You are provided with a dataset named age_salary. It gives information about 400 people of different ages and salaries and the commodities they purchased. Split the dataset into training and testing sets. Remember to use only age and salary columns as your feature matrix. Don't forget to scale the data before training your model.
2. **SVM Training:** Train an SVM classifier on the training set using linear SVM.
3. **Visualization:** Visualize the decision boundary of the trained SVM models along with the support vectors. Use different colors for different labels.
4. **Model Evaluation:** Evaluate the performance of the trained models on the testing set. Report accuracy, precision, recall, and F1-score. You are allowed to use Classification_report from scikit-learn.
5. **Reset:** This time use make_circles from scikit_learn in order to create dataset. Repeat steps 1-4 for this dataset. Is linear kernel the best option here?

3- Optimization Methods

Please complete "OptimizationMethods.ipynb" file. In this notebook, you will learn more advanced optimization methods that can speed up learning.

4- Data Classification with One Hidden Layer (Optional)

Please complete "classification.ipynb" file. In this notebook, you will implement a 2-class classification neural network with a single hidden layer and forward and backward propagation.