DDA3020 Homework 1

Instructions

- The **deadline** is 23:59, Oct 17, 2025.
- The weight of this assignment in the final grade is 20%.
- **Electronic submission**: Turn in solutions electronically via Blackboard. Be sure to submit your answers as one pdf file plus two python scripts for programming questions. Please name your solution files as "DDA3020HW1_studentID_name.pdf", "HW1_name_Q1.ipynb" and "HW1_name_Q2.ipynb" (".py" files are also acceptable).
- The complete and executable codes must be submitted. If you only fill in some of the results in your answer report for programming questions and do not submit the source code (.py or .ipynb files), you will receive 0 points for the question.
- Note that **late submissions** will result in discounted scores: 0-48 hours → 50%, more hours → 0%.
- Answer the questions in English. Otherwise, you'll lose half of the points.
- Collaboration policy: You need to solve all questions independently and collaboration between students is **NOT** allowed.

1 Written Problems (50 points)

1.1. (Logistic Regression, 25 points) Suppose we have training data $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{y}_i \in \mathbb{R}^k$ for $i = 1, 2, \ldots, N$. Consider the least-squares problem:

$$\min_{\mathbf{W} \in \mathbb{R}^{k \times d}, \mathbf{b} \in \mathbb{R}^k} \sum_{i=1}^{N} ||\mathbf{y}_i - \mathbf{W}\mathbf{x}_i - \mathbf{b}||_2^2.$$

1. Find the closed-form solution of the problem above. (5 points)
2. Show how to use gradient descent to solve the problem. Please state at least two possible stopping criterion. (5 points)

3. In a multi-class logistic regression setting, we aim to classify each sample into one of K categories. Let $\mathbf{x}_i \in \mathbb{R}^d$ be the feature vector for the i-th sample and $\mathbf{y}_i \in \mathbb{R}^K$ be its one-hot encoded label vector.

   The logit vector $\mathbf{z}_i \in \mathbb{R}^K$ for sample i is computed as a linear transformation of the features, using a weight matrix $\mathbf{W} \in \mathbb{R}^{d\times K}$:
$$\mathbf{z}_i = \mathbf{W}^T \mathbf{x}_i.$$

   The predicted probabilities $\hat{\mathbf{y}}_i$ are obtained by applying the Softmax function to the logits:

$$\hat{y}_{i,k} = \frac{\exp(z_{i,k})}{\sum_{j=1}^{K} \exp(z_{i,j})} \quad \text{for } k = 1, \ldots, K.$$

   The performance of the model is measured by the average cross-entropy loss over all N samples:

$$J(\mathbf{W}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{i,k} \log(\hat{y}_{i,k}).$$

   Task: Derive the gradient of the loss function $J(\mathbf{W})$ with respect to the weight matrix $\mathbf{W}$. Show your steps and prove that the gradient can be expressed in the following matrix form:

$$\nabla_{\mathbf{W}} J = \frac{1}{N} \mathbf{X}^T (\hat{\mathbf{Y}} - \mathbf{Y}),$$

   where $\mathbf{X} \in \mathbb{R}^{N\times d}$ is the matrix of input features, $\hat{\mathbf{Y}} \in \mathbb{R}^{N\times K}$ is the matrix of predicted probabilities, and $\mathbf{Y} \in \mathbb{R}^{N\times K}$ is the matrix of true one-hot labels. (15 points)

1.2. (Support Vector Machine, 25 points) In class we discussed support vector classification, but did not cover the regression problem. In this question you have to derive the dual form of SV regression (SVR). Your training data is $(x_1, y_1), \ldots, (x_n, y_n)$, where $x_i \in \mathbb{R}^m, y_i \in \mathbb{R}$.

Since the hinge loss that we used in class is only designed for classification, we cannot use that for regression. A frequently used loss function for regression is the epsilon sensitive loss:

$$L_\epsilon(x, y, f) = |y - f(x)|_\epsilon = \max(0, |y - f(x)| - \epsilon)$$

Here x is the input, y is the output, and f is the function used for predicting the label.

Using this notation, the SVR cost function is defined as

$$\frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} L_\epsilon\left(x_i, y_i, f\right)$$

where $f(x) = w^T x$, and $C, \epsilon > 0$ are parameters.

1. Introduce appropriate slack variables, and rewrite this problem as a quadratic problem (i.e. quadratic objective with linear constraints). This form is called the Primal form of support vector regression. (5 points)
2. Write down the Lagrangian function for the above primal form.(5 points)
3. Using the Karush Kunh Tucker conditions derive the dual form. (5 points)
4. Can we use quadratic optimization solvers to solve the dual problem? (5 point)
5. How would you define support vectors in this problem? (5 points)

2 Programming (50 points)

2.1. (Logistic Regression, 25 points) In this question, we will implement multi-class logistic regression using gradient descent. Follow the code template in *HW1_P2_Logistic.ipynb*. Please note: To deepen your understanding of the algorithm, you should not use the sklearn package to solve this question. In the `get_gradient` section, you can directly apply the formula derived in the written problem Question 1.1.3 above for implementation.

2.2. (Support Vector Machine, 25 points) In this problem, we will explore the fundamental application of Support Vector Machines (SVM) for a classic classification task using the *scikit-learn* library, as outlined in the *HW1_P2_SVM.ipynb* file. By completing the core implementation steps within the notebook, you will train a linear SVM classifier and then visualize its decision boundary.