Machine Learning (Problem set 3)

Ali Alipour, *University of Tehran*

This report covers a series of machine learning tasks, focusing on the classification of signals and datasets, model evaluation, and the application of various algorithms such as support vector machines (SVM) and decision trees. The exercises explore essential concepts in machine learning, such as handling imbalanced datasets, normalizing data, and reducing overfitting using regularization techniques. In the first section, we analyze electrocardiogram (ECG) signal classification, extracting features and addressing the challenge of imbalanced classes. We implement different machine learning models, evaluate their performance using confusion matrices, and refine the models using data normalization techniques. The second part of the report delves into the use of SVM with various kernels, including linear, radial b asis function (RBF), and polynomial kernels. These models are compared based on their ability to classify non-linear datasets and the effectiveness of regularization methods in improving model performance. Finally, we explore decision trees, with a focus on the pre-pruning technique to prevent overfitting. The tree models are evaluated in terms of accuracy, complexity, and their ability to generalize well on unseen data. Through these exercises, we aim to understand the trade-offs between model complexity and performance and how to fine-tune machine learning algorithms for better results.

Question 1

In this question, we aim to classify ECG signals. Initially, we perform feature extraction on the ECG signals, obtaining 169 different features.

Part A

We need to add the required libraries to proceed with the exercise. After extracting the data from the Excel format, we convert it to a DataFrame and separate the labels from the data. Using the provided code, we then determine the number of each label in the dataset.

Part B

The dataset is highly imbalanced, as observed from the distribution of the data. This imbalance is a common issue in machine learning where certain classes have significantly fewer samples than others. This can lead to biased models that perform poorly on the minority classes, as the model is optimized to minimize overall error, which is heavily influenced by the majority class.

Part C

Next, we split the data into training and test sets and fit the model. The errors on the training and test sets are computed. The confusion matrix shows that the network has reduced the error on the majority class but still struggles with the minority class.

Part D

After normalizing the data using standard normalization, we re-train the network. The performance of the model is assessed using precision, recall, and F1-score for each class.

Question 2

In this exercise, we generate the required data and display it. We then design a neural network to classify the generated data.

Part A

Using the provided code, we generate the data and plot the points. We observe that the created data is imbalanced.

Part B

We define a Madaline network as required and report the accuracy of the model.

Part C

We repeat the process with a different set of neurons and observe the accuracy of the model.

Part D

As we increase the number of neurons, the model's ability to separate the data improves, although this also increases the complexity and training time. It is crucial to strike a balance between model complexity and performance.

Question 3

In this question, we download the CIFAR-10 dataset and display six random images from the dataset.

Part A

We load the dataset, normalize the data, and convert it to decimal form for training and testing.

Part B

After adding the required libraries, we split the dataset into training and testing sets. The process shows that the dataset is prepared for model training.

Part C

We design the required model and train it using different solvers (SGD, RMSProp, Adam) and compare their performances.

Part D

We observe that Adam performs better and converges faster, though more epochs might be needed to achieve a better solution.

Question 4

In this question, we load the required libraries and dataset, build a decision tree model, and report the accuracy on both training and test sets.

Part A

Using the provided code, we load the dataset and separate the training and testing sets.

Part B

We build the decision tree model, observe the training process, and report the accuracy.

Part C

We discuss the pre-pruning technique used to stop the decision tree from overfitting by setting stopping criteria based on tree size, the number of nodes, or impurity reduction.

Part D

A model with depth 2 is designed, and we compare its accuracy with a fully trained tree. The shallower tree has lower accuracy on the training data but performs better on the test data, indicating it generalizes better.

1. First, we define the necessary functions and libraries. We generate random data points within a specified radius and display the corresponding images.
2. Next, we repeat the process for a second scenario, generating and displaying the random data points again.
3. Using the generated data, we organize them into a DataFrame and assign labels to them.
4. We then write a manual code to split the data into training and testing sets.
5. Using the Logistic Regression class, we manually implement a logistic regression algorithm to classify the data.
6. Afterward, we increase the features using a mapping function to enhance the classification process.
7. Finally, we implement a classification algorithm using radial data generated from uniform and normal distributions and report the results.

1. First, we load the dataset. We define both the Gaussian kernel and the estimated kernel function.

2. We extract the "duration" column from the dataset and represent it as a list. We proceed to plot the results of the kernel density estimations for different values of "duration."

3. We use the `kernelDensity` function to compute the kernel density and plot the results.

4. Lastly, we perform Parzen window estimation for a subset of 250 data points and visualize the results.

References

[Alipour Fraydani, 2024] Alipour Fraydani, A. (2024). Homework on Machine Learning problem set 3, University of Tehran. *Unpublished Manuscript*, Department of Electrical Engineering, University of Tehran.