Report

Homework 1

Goal: Develop models for 10-class classification problems with medium and large input space

Andrea Massignan

November 24, 2023

Project Overview

The goal is to provide two different solution for a 10-class classification problem.

1 Dataset

The datasets used in this project consists in training sets and blind test sets. The training sets are composed by 50000 samples, each one with 100 features for Dataset 1 and 1000 features for Dataset 2. The blind test sets are composed by 10000 samples, each one with 100 features for Dataset 1 and 1000 features for Dataset 2. The labels are 10, one for each class.

- `X_train`: 50000 samples for each dataset.
- `n_features`: 100 for Dataset 1 and 1000 for Dataset 2.

2 Data Exploration

The datasets, that are in a csv file, the elements $V^{(i)}$ belonging to the X column are feature vectors that represent the input data.

The Y column contains the associated labels C_k(k=0,...9).

Thus, each line i in column X is a feature vector structured as:

$[V_1^{(i)}, V_2^{(i)}, ..., V_j^{(i)}, ... , V_d^{(i)}]$

Where d is the size of the feature vector with sizes described before.

And $i = 1, ..., n$, where n is the number of samples in the dataset as specified before.

These are loaded with an helper function called `load_data` that returns the training set in the X matrix and the labels in the Y vector.

Then the training set is split into `X_train` and `Y_train` with a test size of 0.2 and different random states.

Since the dataset is already vectorized, no further preprocessing is needed at the moment.

3 Metrics

The metrics used to evaluate the models are the accuracy and the confusion matrix.

3.1 Precision

Precision is a measure of the accuracy of the positive predictions made by a model. It is the ratio of true positive predictions to the total number of positive predictions made by the model (both true positives and false positives). Precision is calculated using the formula:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Precision provides insight into the model's ability to avoid making false positive predictions. A high precision indicates that when the model predicts a positive class, it is likely to be correct.

3.2 Recall (Sensitivity or True Positive Rate)

Recall measures the ability of a model to correctly identify all relevant instances of the positive class. It is the ratio of true positive predictions to the total number of actual positive instances in the dataset (including both true positives and false negatives). Recall is calculated using the formula:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Recall is especially important when the cost of false negatives (missing a positive instance) is high, as it focuses on minimizing false negatives.

3.3 F1-Score

The F1-score is the harmonic mean of precision and recall. It provides a balanced measure that considers both false positives and false negatives. The F1-score is calculated using the formula:

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1-score is useful when there is an uneven class distribution or when false positives and false negatives have different consequences.

3.4 Support

Support refers to the number of actual instances of each class in the dataset. It is the count of true positive and true negative instances for each class. Support is not directly involved in the calculation of precision, recall, or F1-score, but it provides context by showing how many instances belong to each class.

Classification

4 Data Loading and Splitting

4.1 Loading Dataset 1

The data loading process begins with dataset 1, where the feature matrix X and label vector Y are extracted using the `load_data` function. An informative message is printed to indicate the commencement of this operation. The resulting shape of matrix X is then presented, providing insights into the number of samples and features within dataset 1.

4.2 Loading Dataset 2

An analogous procedure is employed for dataset 2. The feature matrix $X2$ and label vector $Y2$ are obtained through the `load_data` function. A corresponding message is printed, and the shape of $X2$ is displayed to convey the dataset's sample and feature dimensions.

4.3 Splitting Data

The subsequent steps involve splitting the loaded data into training and testing sets:

```
X_train1, X_test1, Y_train1, Y_test1 = train_test_split(X, Y, ...)
  ...
X_train4, X_test4, Y_train4, Y_test4 = train_test_split(X2, Y2, ...)
```

Utilizing the `train_test_split` function, both dataset 1 and dataset 2 are partitioned into training and testing sets. Each dataset undergoes this process twice, resulting in four sets per dataset. The `test_size` parameter dictates the proportion allocated for testing (20%), while `random_state` ensures reproducibility.

The code concludes by acknowledging the completion of data loading and splitting.

The resulting variables (X_train1, X_test1, Y_train1, Y_test1, etc.) are now ready for subsequent machine learning tasks.