Machine Learning Coursework Report

Yasin Lester M00809695

Implementations and Insights from Two-Fold Cross-Validation

**Machine Learning Coursework Report**

**Introduction**

The goal of this coursework was to implement machine learning algorithms from scratch to classify handwritten digits using the Optical Recognition of Handwritten Digits Data Set from the UCI Machine Learning Repository. The system was tested using two-fold cross-validation, as outlined in the coursework brief.

Two algorithms were implemented for this task:

1. K-Nearest Neighbours (KNN)
2. Multi-Layer Perceptron (MLP)

The performance of these algorithms was evaluated based on their accuracy across the two folds.

**K-Nearest Neighbours (KNN)**

**Implementation**

The KNN algorithm was implemented with the following features:

* **k-Value:** Set to 3.
* **Distance Metric:** Euclidean distance was used to calculate the similarity between data points.
* **Weighted Voting:** Neighbour contributions were weighted by the inverse of their distance raised to a power of 4.5.
* **Normalization:** Similar to the MLP, the dataset was normalized to ensure consistency.
* **Two-Fold Validation:** The dataset was split and evaluated using two-fold validation.

**Results**

* **Fold 1 Accuracy:** 98.04%
* **Fold 2 Accuracy:** 98.65%
* **Average Accuracy:** 98.35%

The KNN algorithm provided consistent results across the two folds, with an accuracy close to that of the MLP.

**Multi-Layer Perceptron (MLP)**

**Implementation**

The MLP algorithm was designed with the following structure:

* **Input Layer:** 64 nodes, corresponding to the 64 features in the dataset.
* **Hidden Layer:** 512 nodes with ReLU(Rectified Linear Unit) activation to capture complex patterns.
* **Output Layer:** 10 nodes, representing the digit classes (0-9).
* **Learning Rate:** Set to 0.01.

Key components of the implementation:

* **Forward Pass:** Computes activations for the hidden and output layers using ReLU activation.
* **Backpropagation:** Updates weights using the error gradients.
* **Normalization:** The dataset was normalized to the range [0, 1] for improved convergence.
* **Two-Fold Validation:** The dataset was split into training and testing sets for two-fold validation.
* **Random Initialization of Weights:** The weights in my implementation are initialized randomly at the start of training. This is crucial for breaking symmetry and ensuring effective learning. However, it also means that the results can vary slightly with each execution due to differences in the initial weights.

**Results**

* **Fold 1 Accuracy:** 97.65%
* **Fold 2 Accuracy:** 99.47%
* **Average Accuracy:** 98.56%

The MLP achieved high accuracy, demonstrating its ability to learn complex patterns in the dataset.

**Discussion**

**Algorithm Comparison**

The MLP achieved slightly higher accuracy than the KNN because it can learn and model complex relationships in the data through iterative training. However, the MLP requires significantly more computational resources and runtime. The MLP program takes approximately 2 minutes to execute fully, compared to the KNN, which completes much faster because it does not involve a training phase.

Additionally, the MLP’s random initialization of weights can lead to slight variations in results between runs, making it less deterministic than the KNN. In contrast, KNN provides consistent results because it relies solely on deterministic distance calculations and voting.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Fold 1 accuracy | Fold 2 accuracy | Average accuracy |
| KNN | 98.04% | 98.65% | 98.35% |
| MLP | 97.65% | 99.47% | 98.56% |

**Advantages and Limitations**

**MLP:**

* Advantages: High accuracy, ability to model non-linear relationships.
* Limitations: Requires more computational resources and careful tuning of hyperparameters.

**KNN:**

* Advantages: Simple to implement, no training phase required.
* Limitations: Computationally expensive during prediction, sensitive to the choice of k.

**Dataset Usage**

The dataset was split into two subsets for training and testing in a two-fold cross-validation setup. Each subset was normalized to ensure consistent feature scaling.

**Potential Improvements**

* Exploring additional algorithms such as Support Vector Machines or Decision Trees.
* Hyperparameter tuning to further optimize performance.
* Adding confusion matrices for deeper analysis of classification results.

**Conclusion**

Both the MLP and KNN algorithms successfully classified handwritten digits with high accuracy. The MLP slightly outperformed the KNN in this task. The implementation of these algorithms from scratch provided valuable insights into their workings and effectiveness.

**Self-Marking Sheet**

|  |  |  |
| --- | --- | --- |
| Component | Marks allocated | Self-marking |
| Self-Marking Sheet | 10 | 10 |
| Running Code | 10 | 10 |
| Two-Fold Test | 5 | 5 |
| Quality of Code | 15 | 8 |
| Report | 20 | 12 |
| Quality of Results | 20 | 11 |
| Quality of Algorithm | 20 | 11 |
| Total | 100 | 67 |