Cairo University
Faculty of Computers and Artificial Intelligence
AI321 - Theoretical Foundations Of Machine Learning

ML Project Phase - 3

Name	ID
Marwan Osama	20220324
Ammar Mohammed	20220261
Hassan El-Hamid	20220523
Moahmmed Mahmoud	20220732
Youssef Reda	20220392

Comparative Analysis of Machine Learning Models

Introduction

The objective of this report is to present the results of testing various machine learning models on a MNIST dataset and to compare their performances in terms of accuracy and computational efficiency. The models evaluated include:

1. Support Vector Machine (SVM) implemented from scratch and using a built-in library.
2. Neural Networks with two different architectures.
3. Random Forest Classifier.
4. k-Nearest Neighbors (kNN) Classifier.

Evaluation Metrics

The models were compared based on the following criteria:

1. **Accuracy**: The proportion of correctly classified instances out of the total instances.
2. **Execution Time**: Computational efficiency during model training and evaluation.

Model Performance Summary

1. Support Vector Machine (SVM)

- **From Scratch Implementation**:
    - Accuracy: **0.93**
    - Observations: Implementing the SVM algorithm from scratch demonstrated strong accuracy but took significantly more computational time compared to the built-in library implementation due to the absence of optimization at the library level.
- **Built-In Implementation**:
    - Accuracy: **0.94**
    - Observations: Achieved slightly better accuracy with much faster execution due to optimized internal algorithms.

2. Neural Network

- **According to the following Comparison Table:**

Architecture Performance Comparison:

====================================

Architecture	Mean CV Score	CV Std	Test Score	Convergence
(200, 100, 50)	0.979100	0.003402	0.9837	53
(200, 100)	0.978400	0.002264	0.9830	48
(200,)	0.978267	0.001898	0.9811	41
(100, 50)	0.975950	0.001328	0.9798	49
(100, 100)	0.975850	0.002392	0.9784	46
(100, 50, 25)	0.975683	0.001837	0.9801	48
(100,)	0.974683	0.001883	0.9782	54
(50, 100)	0.970917	0.003888	0.9765	63
(50,)	0.967300	0.002975	0.9723	92

Best Architecture: (200, 100, 50)

- **The Chosen Architecture (Hidden Layers: 200, 100, 50 Neurons)**:
  - Accuracy: **0.9837**
  - Observations: A deeper architecture improved accuracy slightly but at the cost of increased computational time a little bit.

3. Random Forest Classifier

- **Performance**:
  - Accuracy: **0.97**
  - Observations: The Random Forest model achieved high accuracy and required significantly less computational time than the Neural Network and SVM models.

4. k-Nearest Neighbors (kNN)

- **Performance**:
  - Accuracy: **0.97**
  - Observations: Similar to Random Forest, the kNN model delivered high accuracy with minimal computational overhead.

Comparison

Model	**Accuracy**	**Computational Efficiency**
SVM (Custom)	0.93	Low
SVM (Built-In)	0.94	Moderate
Neural Network (Arch 1)	0.9782	Moderate
Neural Network (Arch 2)	0.9873	Low
Random Forest	0.97	High
kNN	0.97	High

Conclusion

The results indicate that:

1. **Neural Networks** are most suitable for achieving the highest accuracy (0.987 with a deeper architecture). However, they require considerable computational resources. And the Architecture and Depth of
2. **Random Forest** and **kNN** are excellent choices for scenarios where computational efficiency is a priority, offering a strong balance between speed and accuracy (0.97).
3. Implementing **SVM** from scratch provided valuable insights into the algorithm but is not practical for real-world applications compared to optimized library versions.

Recommendations

1. Neural Networks are the best method for the MNIST Dataset while SVM achieves low performance due to low accuracy and high latency in execution time
2. Use **Neural Networks** for applications where accuracy is critical, and computational resources are available.
3. Choose **Random Forest** or **kNN** for fast prototyping or when computational efficiency is essential.
4. Optimize and leverage built-in libraries for SVM if it is to be used in production environments.
