WEEK 12 FINAL REPORT
Deep Learning vs XGBoost Comparative Study

Prepared by: Greeshma

Course: Deep Learning - Week 12 Assignment

Date: May 04, 2025

Introduction

Machine learning projects often require careful selection of models based on problem complexity, dataset size, execution time, and required accuracy. In this assignment, Deep Learning models and XGBoost models were evaluated on synthetic datasets of various sizes (1000, 10000, and 100000+ rows) to understand their performance under real-world scenarios.

Deep Learning models are widely known for their ability to capture complex patterns in data and are used in fields like speech recognition, computer vision, and natural language processing. However, they can be computationally intensive and may not be the most efficient choice for structured or tabular data.

On the other hand, XGBoost is one of the most powerful gradient boosting algorithms. It is known for speed, accuracy, and scalability, particularly in tabular datasets. It is widely used in data competitions and real-world applications where performance and efficiency are important.

This report presents a detailed comparison of Deep Learning and XGBoost (Python and R) models and provides analysis-based recommendations.

Deep Learning Results and Analysis

Dataset	Configuration	Train Error	Val Error	Time(s)
1000	1 hidden layer - 4 nodes	0.3800	0.3450	4.60
1000	2 hidden layers - 4 nodes each	0.4075	0.4300	7.10
10000	1 hidden layer - 4 nodes	0.1231	0.0965	2.82
10000	2 hidden layers - 4 nodes each	0.0792	0.0605	3.15
100000	1 hidden layer - 4 nodes	0.0021	0.0034	17.27
100000	2 hidden layers - 4 nodes each	0.0021	0.0034	16.39

Deep Learning models showed significant improvement in validation errors as dataset size increased. However, training time also increased substantially. Models with 1 hidden layer were found to be more efficient while maintaining similar performance to 2 hidden layer models, especially on large datasets.

Overall, Deep Learning is suitable when pattern recognition is difficult or when handling complex

datasets, but is less preferred for very large datasets due to higher training time.

XGBoost Python Results and Analysis

Dataset	Accuracy	Time(s)
100	0.9400	3.10
1000	0.9520	0.42
10000	0.9755	1.44
100000	0.9868	4.46
1000000	0.9917	51.62
10000000	0.9931	427.75

XGBoost Python demonstrated superior performance across all dataset sizes. Accuracy consistently improved with the increase in data volume, and even for datasets with 10 million records, XGBoost maintained excellent accuracy levels.

Moreover, training time remained very reasonable, and the model scaled exceptionally well. XGBoost Python is highly recommended for large datasets due to its balance of accuracy, speed, and scalability.

XGBoost R Results and Analysis

Method	Dataset Size	Accuracy	Time(s)
Direct CV	100	0.8784	0.97
Direct CV	1000	0.9290	1.77
Direct CV	10000	0.9578	2.44
Direct CV	100000	0.9710	8.12
Direct CV	1000000	0.9750	143.72
Direct CV	10000000	0.9757	547.55

XGBoost R also provided good accuracy across all dataset sizes. However, it had significantly higher training times compared to XGBoost Python. This makes it less suitable for very large datasets or real-time applications.

XGBoost R is preferred in scenarios where interpretability, control over cross-validation, and

explainability are critical requirements. It is suitable for research or small-to-medium scale projects.

Final Recommendation and Conclusion

After conducting a comprehensive comparison, it is clear that while Deep Learning models are effective in reducing validation error, their slower training times make them less ideal for large datasets. They are best reserved for problems that involve complex data patterns or non-tabular data such as images or text.

XGBoost Python stands out as the overall superior model for structured data. It offers rapid training, exceptional accuracy, and scalability to millions of rows. This makes it the recommended choice for large-scale and real-world data problems.

XGBoost R, although slower, holds value for projects needing careful validation and interpretability.
It is better suited for academic or research scenarios with small to medium datasets.

Based on all evidence and analysis, XGBoost Python is recommended as the best overall model for large datasets requiring efficiency, speed, and high accuracy.