

Comparison of Deep Learning vs. XGBoost Model

Data Size	Configuration	Training error	Validation error	Time of execution
1000	1 hidden layer (4n)	0.0875	0.0850	4.51 seconds
10000	1 hidden layer (4n)	0.0035	0.0060	15.19 seconds
100000	1 hidden layer (4n)	0.0012	0.0013	150.12 seconds
1000	2 hidden layers (4n)	0.1025	0.0700	5.93 seconds
10000	2 hidden layers (4n)	0.0038	0.0070	17.02 seconds
100000	2 hidden layers (4n)	0.0017	0.0017	162.76 seconds

Method Used	Dataset Size	Testing-set Predictive Performance	Time Taken (s)
Python + sklearn (5-fold CV)	1000	0.9520	0.29
Python + sklearn (5-fold CV)	10000	0.9730	0.85
Python + sklearn (5-fold CV)	100000	0.9865	6.35

Model Comparison and Conclusion:

The evaluation between deep learning models and XGBoost algorithm shows different strengths and weaknesses. The 1-layer and 2-layer neural networks produced acceptable validation errors of 0.0850 and 0.0700 respectively when working with 1,000 records but needed substantially longer execution time ranging from 4.5 to 6 seconds. XGBoost delivered a predictive performance of about 0.9520 while requiring only 0.29 seconds for training thus demonstrating both high efficiency and accuracy in small-scale operations.

The deep learning models decreased validation error as the dataset size grew from 10,000 to 100,000 records until they reached 0.0013 for the 1-layer model in the largest dataset. The execution time exceeded 150 seconds when processing 100,000 records. The XGBoost algorithm delivered outstanding results by reaching predictive accuracy levels of 0.9730 and 0.9865 for 10K and 100K datasets while maintaining execution times at 0.85 and 6.35 seconds respectively.

The 2-layer deep learning models failed to achieve better results than the 1-layer models across all dataset sizes.

The additional training duration of 2-layer deep learning models did not lead to better validation error results and sometimes produced slightly worse results compared to 1-layer models. The extra network complexity appears to have caused unnecessary redundancy or overfitting issues especially when working with smaller datasets.

Deep learning models need large datasets and extensive training time to reach low validation errors yet XGBoost provides a more practical solution for predictive modeling when strong generalization and computational efficiency matter. XGBoost stands as the best model throughout this because it delivers consistent accuracy together with fast training times for all dataset sizes.