1.

Data size	Configuration	Training error	Validation error	Time of execution
1000	1 hidden layer 4 nodes	0.238	0.2528	0.041
10000	1 hidden layer 4 nodes	0.0111	0.012	0.688
100000	1 hidden layer 4 nodes	0.0006	0.0006	4.475
1000	2 hidden layers of 4 nodes each	0.226	0.238	0.044
10000	2 hidden layers of 4 nodes each	0.2376	0.238	0.186
100000	2 hidden layers of 4 nodes each	0.0006	0.0007	9.212

2.

Among the deep learning models evaluated the settings which utilized one hidden layer with four nodes alongside 100000 training examples produced the most advantageous outcomes. The model demonstrates the best performance by reaching training error levels of 0.0006 and validation error levels of 0.0006 while avoiding overfitting. The 2-hidden-layer model with 100,000 data points requires 9.212 seconds to execute but produces the same accuracy as the single hidden layer model that finishes in 4.475 seconds. Both 100,000-data models demonstrate good generalization based on their equivalent training and validation errors although the single hidden layer solution performs equally well while requiring less computational resources. The experimental results indicate that model performance improves substantially more from larger data sets than from network complexity additions because the error rates decrease significantly after using 100,000 data points instead of 1,000 data points.

3.

Method Used	Dataset Size	Testing Set Performance	Time Taken for the model to be fit
XGBoost in Python via scikit-learn and 5-fold CV	1000	0.949	0.31
	10000	0.9746	0.78
	100000	0.9871	2.74

The XGBoost results provided enable me to conduct a comparison with the deep learning models presented in the previous table. The XGBoost model shows testing set performance of 0.949, 0.9746, and 0.9871 for dataset sizes of 1000, 10000, and 100000 respectively, with execution times of 0.31, 0.78, and 2.74 seconds. The performance metric used in XGBoost appears as accuracy because higher values approach 1 whereas deep learning error metrics operate with lower values for better results.

XGBoost demonstrates superior performance than deep learning models across every metric evaluation. The XGBoost model delivers 0.9871 accuracy which corresponds to 0.0129 error during 2.74 seconds of execution time for the 100,000 dataset size while the best deep learning model (1 hidden layer, 4 nodes) reaches 0.0006 error but requires 4.475 seconds. The performance of XGBoost remains strong even when working with limited data since it reaches 94.9% accuracy with only 1,000 samples while deep learning models experience errors between 0.238-0.252. XGBoost demonstrates superiority through three key advantages: it maintains high performance with limited data availability and trains faster than other models particularly for large datasets and delivers results comparable to other methods without needing complex architecture adjustments. The deep learning models need datasets containing 100,000 samples to achieve competitive results but require longer computational time than XGBoost making it the more practical solution for this problem.