1.

Data size	Configuration	Training error	Validation error	Time of execution
1000	1 hidden layer 4 nodes	0.0019	0.0019	96.9944
10000	1 hidden layer 4 nodes	0.0019	0.0019	101.4970
100000	1 hidden layer 4 nodes	0.0019	0.0019	92.4563
1000	2 hidden layers of 4 nodes each	0.0016	0.0016	87.8583
10000	2 hidden layers of 4 nodes each	0.0016	0.0016	87.6890
100000	2 hidden layers of 4 nodes each	0.0016	0.0016	89.7752

2.

The deep learning model with two hidden layers of 4 nodes each proves superior because it delivers error rates of 0.0016 on training and validation datasets which outperforms the single hidden layer model with 0.0019 error rates. The two-layer model outperforms single-layer models in all data conditions by showing superior pattern recognition capabilities in the data. The two-layer model demonstrates better performance by reaching convergence while achieving superior results than the single hidden layer model which makes it the preferred choice for this particular problem.

3.

Method used	Dataset size	Testing-set predictive performance	Time taken for the model to be fit
XGBoost in Python via scikit-learn and 5-fold CV	1000	0.943	0.32
	10000	0.9742	0.81
	100000	0.986690	6.00

Evaluation shows that XGBoost outperforms other models because it reaches 0.9867 predictive accuracy on 100,000 observations within the 6-second timeframe which surpasses deep learning models requiring ~90 seconds. The deep learning models achieve low error rates of 0.0016-0.0019 but XGBoost outperforms them because it demonstrates consistent improvement as dataset size increases. The better predictive performance and faster computations of XGBoost prove this model to be the most suitable choice for solving this specific problem. The practical benefit of XGBoost stems from its ability to deliver these outcomes without needing additional architectural choices that determine optimal layer setups since real-world deployment demands high performance and efficiency.