1.

Data size	Configuration	Training error	Validation error	Time of execution
1000	1 hidden layer 4 nodes	0.1441	0.1332	3.34
10000	1 hidden layer 4 nodes	0.0068	0.0071	11.34
100000	1 hidden layer 4 nodes	0.0020	0.0019	95.26
1000	2 hidden layers of 4 nodes each	0.1610	0.1457	3.13
10000	2 hidden layers of 4 nodes each	0.0034	0.0041	13.34
100000	2 hidden layers of 4 nodes each	0.0014	0.0014	97.24

2.

A 2-layered network comprising 4 nodes per layer using 100,000 training points demonstrates the most effective configuration according to the outcome. The tested model configuration demonstrates the best performance by reaching a training error of 0.0014 alongside a validation error of 0.0014. The model demonstrates excellent generalization capabilities because its training and validation errors match perfectly at a very low level. The extended training duration of 97.24 seconds for this model justifies itself through its superior accuracy performance. Both model architecture with two layers instead of one as well as increased sample size from 100,000 to 10,000 or 1,000 allow for superior performance through complex pattern recognition in the data without sacrificing generalization ability.

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	100	0.86	0.89
	1000	0.949	0.58
	10000	0.9752	1.25
	100000	0.9874	4.03

The deep learning model with 2 hidden layers and 100,000 training samples proves superior in its results when compared to XGBoost and other deep learning models. The XGBoost model demonstrates effective testing predictions with 0.9874 accuracy but the deep learning model achieves lower error measurements of 0.0014 for training and validation using 100,000 samples. The accuracy calculation from deep learning error rates (0.9986) demonstrates superior performance than XGBoost. The deep learning model shows superior scalability along with enhanced performance benefits when data increases in size despite XGBoost taking shorter time for training at 4.03 seconds compared to 97.24 seconds for handling 100,000 samples. The deep learning model generates substantial improvements which outpace XGBoost because its error rate decreases from 0.1457 at 1,000 samples to 0.0014 at 100,000 samples but XGBoost shows less significant progression with error reduction from 0.86 to 0.9874. The combination of superior performance with enlarged datasets coupled with minimal error rates makes the deep learning model stand out as the top selection regardless of training speed.