1.

Data size	Configuration	Training error	Validation error	Time of execution
1000	1 hidden layer 4 nodes	0.0682	0.0639	3.11
10000	1 hidden layer 4 nodes	0.0323	0.0377	3.04
100000	1 hidden layer 4 nodes	0.0177	0.0141	8.91
1000	2 hidden layers of 4 nodes each	0.0168	0.0129	9.38
10000	2 hidden layers of 4 nodes each	0.0087	0.0077	64.67
100000	2 hidden layers of 4 nodes each	0.0105	0.0122	76.09

2.

The model with two hidden layers containing four nodes each reached the best performance by processing 10,000 training samples. The model configuration with 2 hidden layers and 4 nodes each delivers the best validation error (0.0077) and requires 64.67 seconds for training. The 1 hidden layer model trained on 100,000 samples exhibits the minimum training error of 0.0177 yet the 2 hidden layer deep model delivers better generalization because it has a lower validation error. The validation error rate of 0.0087 is lower than the training error rate of 0.0087 which indicates a well-fitting model that avoids overfitting. The deep architecture requires longer execution time but delivers superior predictive performance which makes it the best configuration among all tested options.

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	0.95	0.291959
	10000	0.9739	0.838179
	100000	0.9868	4.16

The XGBoost results demonstrate outstanding predictive accuracy which increases from 0.95 for 1,000 samples toward 0.9868 for 100,000 samples. The model demonstrates steady performance growth with increased data availability and executes quickly with 4.16 seconds for 100,000 samples while the best deep learning model requires 76.09 seconds. The implementation of 5-fold cross-validation generates better performance estimates than the basic train-validation split used by deep learning models.

Several performance factors demonstrate XGBoost surpasses deep learning models in terms of superiority. The predictive accuracy of XGBoost stands out as superior since it reaches 0.9739 on 10,000 samples whereas the best deep learning model achieves a validation error of 0.0077 (which would equate to 0.9923 accuracy if interpreted correctly but this conflicts with training error results). The computational efficiency of XGBoost stands out because it needs 0.84 seconds to process the same dataset size which takes 64.67 seconds for the other model. The XGBoost model delivers scalable efficiency with steadily improved results as data size increases because it demonstrates better performance characteristics than deep learning models in this context. Data overfitting emerges in deep learning techniques particularly in the 100,000 sample test where validation error surpasses training error. XGBoost proves to be the best solution for this problem due to its exceptional accuracy alongside its efficient computation and dependable results.