1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data size** | **Configuration** | **Training error** | **Validation error** | **Time of execution** |
| 1000 | 1 hidden layer 4 nodes | 0.00400 | 0.015130 | 1.09 |
| 10000 | 1 hidden layer 4 nodes | 0.00110 | 0.001515 | 3.505 |
| 100000 | 1 hidden layer 4 nodes | 0.00092 | 0.001012 | 14.969 |
| 1000 | 2 hidden layers of 4 nodes each | 0.00200 | 0.006722 | 1.20 |
| 10000 | 2 hidden layers of 4 nodes each | 0.00030 | 0.001406 | 3.40 |
| 100000 | 2 hidden layers of 4 nodes each | 0.00058 | 0.000551 | 22.00 |

2.

The model configuration with two hidden layers containing four nodes each performed best with training on 100,000 data points. The proposed model configuration demonstrates the best generalization performance by reaching a validation error of 0.000551. The two-layer architecture achieves superior performance over the single hidden layer model since it presents validation error that is 45% lower than the 0.001012 error rate from the model with 100,000 data points. The training error value of 0.00058 matches closely with the validation error value which indicates that the model does not overfit. The extended training duration (22.00 seconds compared to 14.969 seconds) justifies itself through better validation performance. Model performance improvements come from growing the dataset from 1,000 to 100,000 samples because the validation errors decrease by approximately two orders of magnitude no matter what architecture is used.

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 | 0.34 |
| 1000 | 0.9470 | 0.27 |
| 10000 | 0.9764 | 0.81 |
| 100000 | 0.9873 | 4.05 |

The predictive power of XGBoost models increases steadily as the size of available dataset grows from 100 to 100,000 observations. The model accuracy starts at 0.86 with 100 observations before reaching 0.947 with 1,000 observations and 0.9764 with 10,000 observations and ends at 0.9873 with 100,000 observations. The execution times stay within reasonable limits with 0.27-0.34 seconds for datasets containing 100 observations and 4.05 seconds for the largest dataset of 100,000 observations. XGBoost showcases its ability to provide quick training operations alongside accurate prediction results.

XGBoost stands as the optimal model choice for this specific problem when compared to deep learning models. The evaluation of XGBoost's performance relies on multiple elements where its predicted accuracy (0.9873) surpasses the validation errors (0.000551) of the deep learning models. The training process for XGBoost takes only 4.05 seconds to process 100,000 samples while the best deep learning model needs 22 seconds to complete its execution. XGBoost maintains solid performance across limited datasets since it reaches an accuracy level of 0.947 with only 1,000 samples yet deep learning models generate higher prediction errors when dealing with restricted data. The decision process for deploying XGBoost becomes easier because it demands less attention to hyperparameter tuning and architectural choices when compared to deep learning systems. XGBoost stands as the optimal solution for this problem because of its advantageous features.