1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data size** | **Configuration** | **Training error** | **Validation error** | **Time of execution** |
| 1000 | 1 hidden layer 4 nodes | 0.2488 | 0.2 | 0.13 |
| 10000 | 1 hidden layer 4 nodes | 0.009 | 0.0125 | 0.57 |
| 100000 | 1 hidden layer 4 nodes | 0.0005 | 0.0006 | 3.56 |
| 1000 | 2 hidden layers of 4 nodes each | 0.2488 | 0.195 | 0.03 |
| 10000 | 2 hidden layers of 4 nodes each | 0.2385 | 0.252 | 0.14 |
| 100000 | 2 hidden layers of 4 nodes each | 0.001 | 0.0012 | 2.04 |

2.

Among the deep learning models presented the model with 1 hidden layer of 4 nodes trained on 100,000 data points demonstrates the best performance according to the results. The model demonstrates outstanding generalization ability because it reaches the lowest training error (0.0005) and validation error (0.0006) while maintaining minimal overfitting. The model with one hidden layer of 4 nodes and 100,000 data points achieves superior error metrics than the 2 hidden layers model while maintaining efficient computational performance and reasonable execution time of 3.56 time units. Deployment readiness for this model becomes stronger because its training and validation errors prove consistent with one another.

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 | 0.9470 | 0.22 |
| 10000 | 0.9646 | 0.45 |
| 100000 | 0.9711 | 6.84 |

XGBoost outperforms the deep learning models across every dataset level as shown by performance metrics. The testing-set predictive performance of XGBoost reached 0.9470 for 1000 data points and 0.9646 for 10000 data points and 0.9711 for 100000 data points thereby demonstrating superior performance than deep learning alternatives. The validation errors of 0.0006 achieved by the best deep learning model with one hidden layer and four nodes on 100,000 data points corresponds to approximately 0.9994 performance if measured on the same scale. XGBoost demonstrates efficient execution times of 6.84 time units for processing the largest dataset which matches closely with the 3.56 time units of the deep learning model.

XGBoost demonstrates superior performance in this situation because of multiple factors. XGBoost achieves better data pattern extraction by combining multiple decision trees than neural network architectures since it employs an ensemble based approach. Small datasets show good results when evaluated with XGBoost because the model achieves 0.9470 accuracy with 1000 data points while deep learning models require larger data samples for equivalent performance levels. The better sample efficiency and reduced computational requirements of XGBoost allow it to achieve high performance levels. The combination of superior predictive accuracy with effective computational power and stable performance with different dataset sizes makes XGBoost the optimal choice as a model solution for this scenario.