1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data size** | **Configuration** | **Training error** | **Validation error** | **Time of execution** |
| 1000 | 1 hidden layer 4 nodes | 0.255 | 0.2500 | 0.07 |
| 10000 | 1 hidden layer 4 nodes | 0.011125 | 0.016 | 0.84 |
| 100000 | 1 hidden layer 4 nodes | 0.000500 | 0.000750 | 4.34 |
| 1000 | 2 hidden layers of 4 nodes each | 0.237500 | 0.2200 | 0.04 |
| 10000 | 2 hidden layers of 4 nodes each | 0.007250 | 0.005500 | 1.12 |
| 100000 | 2 hidden layers of 4 nodes each | 0.000812 | 0.00125 | 5.16 |

2.

The 2 hidden layers model utilizing 100,000 samples proves to be the best deep learning model according to the results. The model configuration provides strong predictive ability through minimal overfitting by reaching a very low training error of 0.000812 and validation error of 0.00125. The model runs for 5.16 seconds which represents an acceptable execution time considering its enhanced accuracy performance. Model generalization for unseen data becomes possible through the training and validation errors staying closely matched in this model selection step. The two-layered deep architecture effectively captures advanced relationships within the expanded dataset which makes it the most suitable model for diabetes prediction.

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 | 0.9 | 0.13 |
| 1000 | 0.965 | 0.06 |
| 10000 | 0.9745 | 0.15 |
| 100000 | 0.9872 | 0.83 |

XGBoost delivers superior performance than all deep learning configurations for every dataset size examined. XGBoost achieves superior testing-set predictive performance ranging from 0.9 to 0.9872 which surpasses the best neural network model validation performance of 0.99875 achieved by the 2-layer model with 100,000 samples. XGBoost executes datasets of any size with remarkable speed where it completes 100,000 sample processing in 0.83 seconds while the equivalent neural network takes 5.16 seconds.

XGBoost provides superior performance for this diabetes prediction task because of multiple features which optimize its suitability. XGBoost processes tabular data formats better than neural networks because neural networks perform best when analyzing unstructured data types such as images and texts. The combination of built-in feature importance analysis and non-linear relationship detection without dependent hyperparameter adjustment provides XGBoost with advantages when working with medical datasets including few features. XGBoost provides optimal scaling capacity because it achieves exceptional performance with limited data and uses fewer computational resources when implemented with small datasets. The combination of advantages from XGBoost makes it the optimal model choice for this diabetes prediction application because it provides enhanced accuracy and faster training speeds and potentially better interpretability.