Comparison of Deep Learning and XGBoost Models

1. **Based on the results, which model do you consider as superior, among the deep learning models fit?**

The configuration with two hidden layers of four nodes each consistently outperformed the single-layer model in terms of predictive accuracy as the dataset size increased among the deep learning models assessed in this study. More training data improved both models, but the 2-layer model's generalization and learning ability were better.

The 2-layer deep learning model, for example, obtained a validation error of 0.07360 at a dataset size of 100,000 records, which was significantly less than the 0.15935 error that the single-layer model recorded. This suggests that the network performed better on unseen examples because the extra layer enabled it to capture more intricate relationships within the data. It is noteworthy that, at this scale, the 2-layer model trained marginally faster (46.81 seconds) than the single-layer model (50.04 seconds). This is probably because, despite its slightly higher complexity, the 2-layer model converged more efficiently during training.

Although both models suffered more from the lack of data, the 2-layer model generally maintained comparable or better performance even at smaller dataset sizes like 10,000 and 1,000. All things considered, the findings indicate that the 2-layer deep learning architecture is the more successful option among the deep learning configurations examined in this assignment since it better balances model complexity and learning capacity.

2. **Next, report the results (for the particular numbers of observations) from applying XGboost (week 11 – provide the relevant results here in a table). Comparing the results from XGBoost and deep learning models fit, which model would you say is superior to others? What is the basis for your judgment?**

Data Size	Model Type	Training Error	Validation Error	Time (sec)
1,000	**DL - 1 layer**	**0.33250**	**0.31500**	**2.01**
1,000	**DL - 2 layers**	**0.40000**	**0.40500**	**2.69**
1,000	**XGBoost**	**0.00125**	**0.01000**	**0.03**
10,000	**DL - 1 layer**	**0.25450**	**0.25050**	**6.69**
10,000	**DL - 2 layers**	**0.26588**	**0.26100**	**8.43**
10,000	**XGBoost**	**0.00000**	**0.00400**	**0.07**

100,000	**DL - 1 layer**	**0.15978**	**0.15935**	**50.04**
100,000	**DL - 2 layers**	**0.07300**	**0.07360**	**46.81**
100,000	**XGBoost**	**0.00001**	**0.00055**	**0.44**

Comparative Analysis and Judgment:

The findings unequivocally show that XGBoost performs better than both 1-layer and 2-layer deep learning configurations for datasets of all sizes (1,000, 10,000, and 100,000 records). Three important performance metrics—execution time, validation error, and training error—make this clear.

XGBoost completed the training process in just 0.03 seconds, achieving an exceptionally low validation error of 0.01000 and a training error of just 0.00125 at 1,000 records. In contrast, the deep learning models not only took a lot longer to train (2.01 to 2.69 seconds), but they also did not perform well in terms of accuracy. The 2-layer model, in particular, showed instability at small data volumes, as evidenced by its unexpectedly higher validation error (0.40500) than the 1-layer model (0.31500).

The deep learning models performed better as the dataset grew to 10,000 records, but XGBoost continued to hold a significant advantage. It produced a validation error of only 0.00400, while the best two-layer deep learning model recorded a validation error of 0.26100, which is more than 65 times higher. Furthermore, XGBoost kept its speed advantage, finishing the task in 0.07 seconds as opposed to deep learning's 6–8 seconds.

At 100,000 records, the performance disparity peaked. With a validation error of only 0.00055 and a training error of nearly zero (0.00001), XGBoost not only maintained its superior accuracy but also showed remarkable speed, training in just 0.44 seconds. By contrast, the deep learning models required over **46 seconds** to train, and still delivered **higher validation errors**: 0.15935 for the 1-layer model and 0.07360 for the 2-layer model.

Conclusion:

For this structured data binary classification task, XGBoost is unquestionably the best model when taking into account all metrics, including accuracy, execution speed, and scalability. It is more resilient to changing data volumes, learns more quickly, and generalises better. On the other hand, deep learning models are computationally costly and prone to higher errors, particularly at smaller scales, even though they get better with more data.

This makes XGBoost the more dependable and effective option, especially in real-world settings where accuracy and speed are crucial.