Week_12_Assignment
Week 12 - Neural Network
HDS-5230-07
Shreya Samir Kurani

# Week 12 – Deep Learning vs XGBoost Model Comparison

## 1. Deep Learning Results (Error-Based)

In this analysis, I present performance of deep learning trained on synthetic dataset of different sizes. By default, the models produced accuracy metrics which were converted to error metrics with the formula error = 1 - accuracy to match the assignment requirements. Below is a summary of the training error, validation error and execution time for each model configuration.

| Data Size | Configuration | Training Error | Validation Error | Time of Execution |
|---|---|---|---|---|
| 1000 | 1 Hidden layer (4 nodes) | 0.2613 | 0.235 | 11.64 s |
| 10000 | 1 Hidden layer (4 nodes) | 0.0038 | 0.002 | 19.48 s |
| 100000 | 1 Hidden layer (4 nodes) | 0.0012 | 0.0012 | 180.01 s |
| 1000 | 2 Hidden layer (4 nodes each) | 0.0412 | 0.07 | 9.86 s |
| 10000 | 2 Hidden layer (4 nodes each) | 0.0034 | 0.001 | 23.4 s |
| 100000 | 2 Hidden layer (4 nodes each) | 0.0016 | 0.0018 | 179.34 s |

Hence, this suggests that the overall model performance improves with more data (except for a small jump from 1,000 to 10,000 observations), and that the biggest improvement happens when going from 1,000 to 10,000 observations. Adding a second hidden layer seems to help more for the smaller dataset size

## 2. XGBoost Results (Week 11 Reference)

To compare with the results of the Week 11 assignment, the following table shows the results from the XGBoost (Python implementation with 5-fold cross validation). These metrics constitute a baseline for comparing the deep learning approaches in terms of performance.

| Data Size | Testing-Set Accuracy | Time of Execution |
|---|---|---|
| 100 | 0.98 | 0.02 s |
| 1000 | 0.998 | 0.04 s |
| 10000 | 0.9856 | 0.06 s |
| 100000 | 0.9707 | 0.4 s |

## 3. Superior Deep Learning Model

I then look at the empirical results and considered the model with 2 layer (4 nodes each) deep learning trained on 100,000 rows to be the most effective deep learning configuration. It achieved a validation error of only 0.0018 with a training error of 0.0016, showing good generalization ability. The execution time of 179.34 seconds is computationally intensive, and it is worth it for applications where predictive accuracy is the most important consideration.

The performance of the 2 hidden layer architecture seems to improve because it has a greater capacity to model complex relationships in the data, especially as the dataset size increases. This observation is consistent with what one should expect theoretically of how model complexity depends on dataset size.

## 4. Deep Learning vs XGBoost – Comparative Analysis

The different types of models are shown to follow different performance patterns with respect to data sample size. For smaller datasets (1,000 rows), XGBoost outperforms the best deep learning model by the speed as well as accuracy. XGBoost achieves 0.998 accuracy in 0.04 second whereas the best deep learning model achieves 0.93 accuracy in 9.86 seconds.

When dataset size gets increased, deep learning models, especially ones with 2 hidden layers outperform XGBoost with respect to accuracy. On the order of 100,000 rows, the deep learning model has a validation accuracy of 0.9982, a fair difference of 0.1994 from Xgboost's 0.9707 meaningfully for predictive performance.

Nevertheless, XGBoost remains quite advantageous in the time of execution, regardless of the dataset size. Although comparable deep learning models needed around 180 seconds, the largest model trained using XGBoost took only 0.4 seconds, implying a 450X difference, which can amount to a lot in time critical applications and resource limited environment.

## 5. Final Recommendation and Judgment Basis

Based on the requirements of the use case, I recommend two different recommendations:-

When there are no concerns with computational resources and obtaining maximum predictive accuracy is imperative, we recommend the deep learning model that has two hidden layers with larger datasets ($\geq$ 100,000 observations). As its generalization capabilities are superior than its competitive models, it is good for complex prediction tasks where marginal improvements in accuracy lead to important business or research outcomes.

On the other hand, XGBoost is more practical if efficiency, rapid iteration or limited computational resources are important factors. Despite the fact that it is not able to achieve such high predictive performance as other methods, it is uniquely fast — which makes it especially well suited to deploying models that can be trained and deployed very rapidly in situations where the needed dataset may be smaller or even very small in the first place.

These recommendations are made on the basis of quantitative evaluation of performance metrics (training/validation error), training time requirements and observed scaling behavior over dataset sizes. In order to meet the assignment specifications, the error values were calculated using the formula error = 1 - accuracy taken from model history outputs.

The results imply an important tradeoff between computational efficiency and predictive performance that must be well considered based on the particular application issues.