Week 12 - Neural Network

Deep Learning Model - Python

Data size	Configuration	Training error	Validation error	Time of execution(s)
1000	1 hidden layer 4 nodes	16838507.60	19117599.37	10.58
10000	1 hidden layer 4 nodes	5881283.44	6195782.84	45.76
100000	1 hidden layer 4 nodes	44551305.75	44301406.17	366.74
1000	2 hidden layers of 4 nodes each	36404138.46	37360560.18	11.92
10000	2 hidden layers of 4 nodes each	3608301.71	4033195.52	47.54
100000	2 hidden layers of 4 nodes each	38298279.46	38157371.64	363.59

XGBoost - R code

Data size	Configuration	Training error	Validation error	Time of execution(s)
1000	XGBoost (Git Demo Params)	20576375	23883824	0.04444814
10000	XGBoost (Git Demo Params)	616872251	638614024	0.05348611
100000	XGBoost (Git Demo Params)	56935639286	56445635627	0.14031887

Extra (3 Hidden Layers with dropout)

Data size	Training error	Validation error	Time of execution(s)
1000	852,027,467.59	799,803,699.18	12.68
10000	89,418,029,883.55	88,894,042,681.98	55.51

100000	8,772,232,717,850.74	8,791,813,609,110.16	452.31

Comparison

The best performance outcome using deep learning methods emerged when researchers used the 2-layer 4-node arrangement for testing various data sizes. The chosen 2-layer 4-node design reached the best validation error rate while maintaining practical run times to become the most dependable deep learning structure in testing. Models with three hidden layers containing 64 nodes per layer combined with regularization/dropout performed extensively poorly. Training along with validation MSE reached extremely high levels because of overfitting combined with improper regularizer settings or training instabilities.

Results of testing deep learning models against XGBoost models depended on data sample sizes. With 1,000 observations the simple deep learning model using a single hidden layer achieved a slightly lower validation MSE than XGBoost while running much slower than the XGBoost algorithm. With a dataset size of 10,000 observations the deep learning model delivered both lower training error rates and validation error rates than tree-based models because it was properly adjusted. XGBoost surpassed all deep learning settings at 100,000 observations through faster execution time and achieved substantially lower validation MSE.

Another experiment was conducted using XGBoost alongside a model constructed from 3 hidden layers with dropout included. Throughout the experiments the proposed model demonstrated inferior performance than all other dataset sizes. The excessive error values indicate that the complex deep architecture surpassed the simple input features' limitations which resulted in unstable and unproductive learning procedures.

Conclusion

The results suggest that while deep learning models can be effective particularly at moderate data sizes XGBoost is more robust and scalable, especially when the dataset is large and the features are relatively simple. The data used in this task likely lacked the complexity or high dimensionality needed to justify deeper neural network architectures, making XGBoost the better choice overall for this problem.