Robert Earland

9/16/2023

**Report on Neural Network Model**

**Introduction (Analysis Purpose)**

With the introduction of deep learning, I believe it has presented plenty of neural network architectures tailored for various tasks, ranging from natural language processing to image classification. In this analysis, I will examine the effectiveness of the neural network models for this specific task and hopefully gain some insight on potential alternatives. The purpose of this analysis is to evaluate the effectiveness of the chosen neural network model in addressing the problem, to understand its strengths and limitations, and to explore potential alternative models that might be equally or more effective.

**Results**

The dataset is comprised of over 34,000 organizations that have received funding. It included various metadata such as application type, affiliation, income amount, and more. Data preprocessing included steps like dropping unnecessary ID columns, binning application types and classifications with lower counts, one-hot encoding categorical variables, and scaling. The architecture consisted of an input layer, two hidden layers (with 80 and 30 nodes), and an output layer. The activation functions used were "ReLU" for the hidden layers and "Sigmoid" for the output layer. The model was trained over 100 epochs. This number was chosen to allow the model to learn the patterns in the dataset effectively without overfitting. The desired accuracy for the assignment was 75%, but I was unable to achieve that. The model achieved an accuracy of 73% on the test dataset instead and other methods shown within Jupyter notebook either show the same percentage or slightly lower. The challenges I faced led me to believe the model showed signs of slight overfitting as evidenced by a widening gap between training and validation accuracy over epochs. Also, achieving a higher accuracy required hyperparameter tuning.

**Summary of Overall Results**

Overall, the neural network model delivered an accuracy of 73% rather than the desired 75%. However, the challenge remains to push the model's performance further while ensuring it does not overfit. The preprocessing steps ensured that the data was in an optimal format for the neural network to process, but I’m under the impression that there is still some tweaking to do with the model's architecture and parameters for better results. I tried all the methods I could think of but am open to suggestions from grading officials.

**Alternate Model Proposition**

I thought an alternative approach to address the problem would be using the Gradient Boosted Trees model, specifically the XGBoost implementation. The choice would be influenced by factors like the size of the dataset, computational constraints, and interpretability of the model. For this challenge I used all the resources I had at my disposal including ChatGPT, W3scshools, Stack Overflow, and YouTube for this challenge but I guess it came down to having a deeper understanding of how to implement the changes I needed.