**Predicting NFL Fantasy Football Performance Using a Neural Network**

Adrian Cortez

Department of Mathematics and Computer Science, Olivet Nazarene University

COMP/ENGR: 347: Machine Learning and AI

Prof. Ryan Younger

December 12, 2025

## Introduction

The spectacle of sports competitions can hold a captivating power. It exemplifies the human spirit, bringing out competitiveness and passion like no other. For millions of devoted fans, this passion can be expressed in the game of Fantasy Football, an activity where the fan transforms from a passive observer of the NFL to an active manager of a virtual team. Participants draft a roster composed of real-life professional NFL players, and their weekly points are awarded based on the actual statistical performance of those players in their games. This allows fans to participate in the complex, strategy-driven competition that they observe in sports competitions. In recent years, fantasy football has become an increasingly popular activity for sports fans, and making informed decisions based on accurate projections is critical for winning. Traditional methods of predicting fantasy performance, such as expert analysis or simple trends, often fail to account for complex interactions between a player's statistics and team context.

This project aims to develop a neural network model to predict NFL players' fantasy points for upcoming seasons using historical player statistics. Neural networks are particularly suited for this project because they can learn complex, nonlinear relationships between a variety of different features. This includes factors such as passing, rushing, and receiving statistics, while also considering player age, how injury prone they might be, how much they are used on a team, and more. These patterns often involve subtle, hidden interactions that neural networks can learn. By leveraging this approach, the goal is to generate more accurate and adaptable projections that help users make informed draft and lineup decisions.

**Data Summary**

This project uses the *NFL Fantasy Data 1970-2024* dataset from Kaggle, which provides over 50 years of detailed NFL player statistics and fantasy scoring. This dataset includes CSV files acquired from Pro Football Reference, a leading comprehensive online statistics and historical database from the NFL. After cleaning the data and filtering for complete records, the project focuses on seasons from 1990 to 2024, to better reflect the modern style of football. According to ELDORADO (2016), NFL scoring and offensive production have trended upwards, especially since the early 1990s, reflecting rule changes and evolving offensive strategies that favor passing and scoring. This dataset includes thousands of player entries with features such as passing, rushing, receiving, and other production categories. Key variables include passing statistics, such as passing yards, touchdowns, interceptions, and yards per attempt. Rushing and receiving statistics are included as well, such as rushing attempts, rushing yards, receiving yards, receptions, touchdowns, and more. More advanced statistics are included, such as touches, games played, and scrimmage yards. Furthermore, contextual factors are included, such as age, team, and position. Most importantly, fantasy points are represented in this dataset, including total, weekly, and average fantasy points.

To further prepare our data to feed into our neural network, the categorical position variable (QB, RB, WR, TE) was converted into one-hot encoded columns. This allows our model to learn how different positions affect fantasy scoring. Missing and incomplete values were removed, and players with zero games played were excluded because they do not contribute meaningful patterns to model training. To ensure model stability and improved convergence during training, features were standardized using z-score scaling. Overall, the final cleaned and encoded dataset was suitable for deep learning, consistent with best practices in sports analytics.

**How Neural Networks Work**

The predictive model used in this project is a fully connected neural network implemented using TensorFlow and Keras. It is known as a 'feed-forward neural network', which is a neural network where information moves in one direction. It starts at the input, then the hidden layers, and finally the output. A neural network consists of layers of neurons that transform input data through a series of weighted computations. Each neuron receives numeric inputs, multiplies them by learned weights, and applies an activation function that determines the neuron's output. This is similar to a biological neuron, which fires electric impulses along neural pathways when a threshold is exceeded. The output is passed forward to the next layer, contributing to the calculations that ultimately produce the model's prediction. During training, the model uses backpropagation to compute how much each weight contributed to the prediction error, and an optimizer updates those weights to reduce the difference between predicted and actual fantasy points. By stacking layers, the network can learn complex, non-linear patterns.

## Model Architecture Description

The model uses sequential architecture, meaning layers are stacked in a straight line from input to output. This works well for regression tasks, where the goal is to transform a set numeric features into a single continuous prediction. The model architecture begins with the input layer. The shape of the data is determined by the number of standardized features (passing, rushing, receiving stats, age, etc.). These features provide the model with the statistical profile of each player.

The first hidden layer contains 64 neurons and utilizes a Rectified Linear Unit (ReLU) activation. ReLU is used because it handles nonlinear patterns efficiently and avoids vanishing gradients. The vanishing gradients problem is a common issue in training neural networks, where gradients, or the values used to update the weights during backpropagation, become extremely small and cause the network to stop learning meaningful patterns. 64 neurons are used in this first layer, because it must first learn the broad, high-level relationship across input features. Using a larger first layer gives the network enough capacity to detect complex patterns. Next, there is a dropout layer, where 10 percent of neuron outputs are dropped at random. This helps prevent overfitting, especially with noisy football data (injuries, outliers, and inconsistent usage). This is also known as regularization to improve generalization to unseen seasons.

The second hidden layer uses 32 neurons and ReLU activation once more. We decrease from 64 to 32 neurons in this layer, so we can force the model to compress and refine the representations learned in the first layer. This helps our model learn more important patterns, while reducing the risk of overfitting. This layer is also a Dense layer, where each neuron receives input from every output of the previous layer. This allows the network to learn complex combinations of features, because every neuron has access to information from the prior layer. Another dropout layer is added after this for regularization, which further improves generalization.

Furthermore, the output layer contains 1 neuron, which is all that's needed for our task where the goal is to predict a single numeric value. Unlike classification models, the output layer does not use an activation function, meaning it outputs a linear value. This makes it so the model isn't restricted to a range and can freely learn the full spectrum of possible fantasy outcomes. This neuron computes the final weighted sum, which produces the model's prediction: the player's projected fantasy points for the next season.

Before training, the model is compiled with specific settings that determine how the network learns and how performance is evaluated. The optimizer that this network uses is called Adam, which sets the learning rate to 0.001. Adam automatically adjusts the learning rate for each weight during training. By setting the learning rate to 0.001, we can balance speed and stability during training. The model updates weights gradually enough to avoid overshooting good solutions, but not inefficiently. For our noisy NFL statistics, a smaller learning rate helps the network learn more stable patterns rather than reacting to large outliers. Additionally, the loss function configured is Mean Squared Error (MSE). MSE measures the average squared difference between predicted and actual fantasy points. Because errors are squared, the model is penalized more heavily for large mistakes. Lastly, the evaluation metric is the Mean Absolute Error (MAE). This treats all errors proportionally, unlike MSE, making it easy to interpret. MAE represents the average number of fantasy points the model's predictions deviate from real outcomes.

## Evaluation Results and Interpretation

### Learning Curves

The neural network was trained for 20 epochs using an 80/20 training-testing split with an additional validation split applied during training. As observed in Figure 1 (Appendix), the learning curves show a rapid decrease in both training and validation loss early in training, indicating that the model learned meaningful patterns quickly. Over the course of training, both training and validation losses steadily decreased and stabilized, with small fluctuations. This suggests that the model converged effectively and learned generalizable relationships rather than overfitting. Based on training and validation loss curves, the model performed well, capturing important patterns.

### Test Set Performance

Performance on the unseen test set further confirms the model's effectiveness. The model achieved a Test MAE of 1.78, meaning that, on average, predictions differ from actual fantasy outcomes by fewer than two fantasy points per player. This can be observed in Figure 2 (Appendix). The Test MSE of 8.30 and corresponding RMSE of 2.88 show that even larger prediction errors remain small. The model also achieved an R-squared score of 0.9987, indicating that it explains over 99.8% of the variance in fantasy point outcomes. More variance means more information, so this high value demonstrates that the neural network captures the underlying relationships between player statistics and fantasy production extremely well.

Overall, these results suggest that the baseline neural network is both highly accurate and can generalize well on unseen data. Low error metrics and strong validation performance indicate that the architecture and training configuration are well suited for predicting fantasy football performance.

**Bias-Variance Analysis**

In this project, the neural network must balance bias (underfitting) and variance (overfitting). A model with high bias would be too simple to capture the complex relationships in our fantasy football data. This would be represented in high training and validation errors. A model with high variance, on the other hand, would memorize specific player seasons instead of learning generalizable patterns. This would be represented by training loss dropping while validation loss fluctuates or increases. It is evident from our results that the learning curves decrease smoothly, while validation loss initially improves and then stabilizes with small fluctuations. This indicates moderate variance, which is expected for neural networks, but not severe overfitting. To mitigate this, the implementation of more regularization techniques such as early stopping, dropout, and a reduced learning rate can help control variance by preventing the network from becoming overly complex.

## Ethics and Impact Analysis

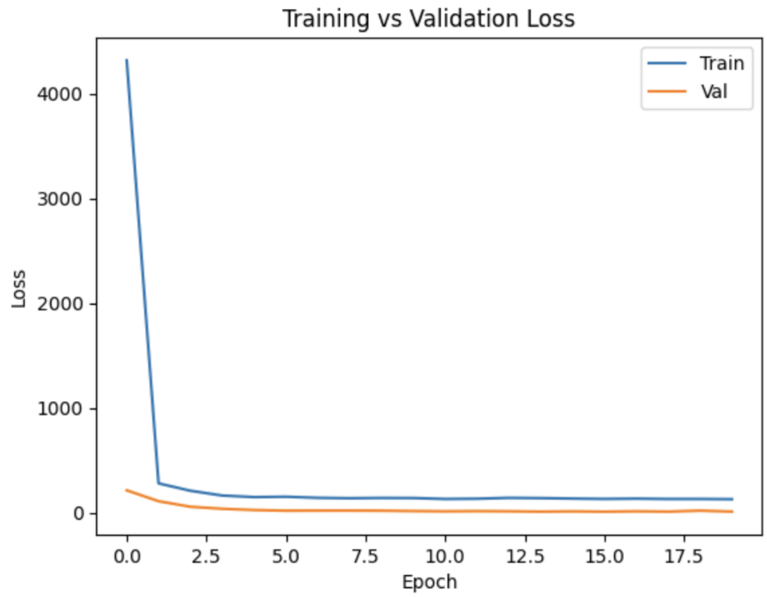
When creating a machine learning or AI model, one must consider the ethical implications and impact. This project uses publicly available NFL player statistics and does not involve private information, reducing concerns over privacy issues. However, fantasy football projections can influence the perception people have on a player and the public narratives that subsequently follow. This can create an unrealistic expectation or misinterpretation of a player's true value. Moreover, while the neural network produces accurate predictions, it cannot account for unpredictable factors such as injuries, team strategy, or personal circumstances. Over relying on the model without human judgment may lead to biased or misleading conclusions. Therefore, model predictions should be viewed as a tool to assist with the strategy involved in fantasy football, rather than a definitive conclusion. Overall, when used responsibly and transparently, this model has positive impacts by enhancing decision making in fantasy football.

## Conclusion

In summary, this project successfully met the goal of implementing a neural network model that is capable of accurately predicting NFL players' fantasy points. By effectively leveraging the complex relationships hidden in NFL statistics, the model was able to achieve a strong predictive accuracy, as evidently seen in the MAE and R-squared scores. When used responsibly, fans of the NFL can use this tool to further their passion and competitiveness inspired from the captivating nature of sports. Looking forward, the implementation of this project in the field of sports demonstrates how the world of AI is continuing to push the boundaries of what was previously known.

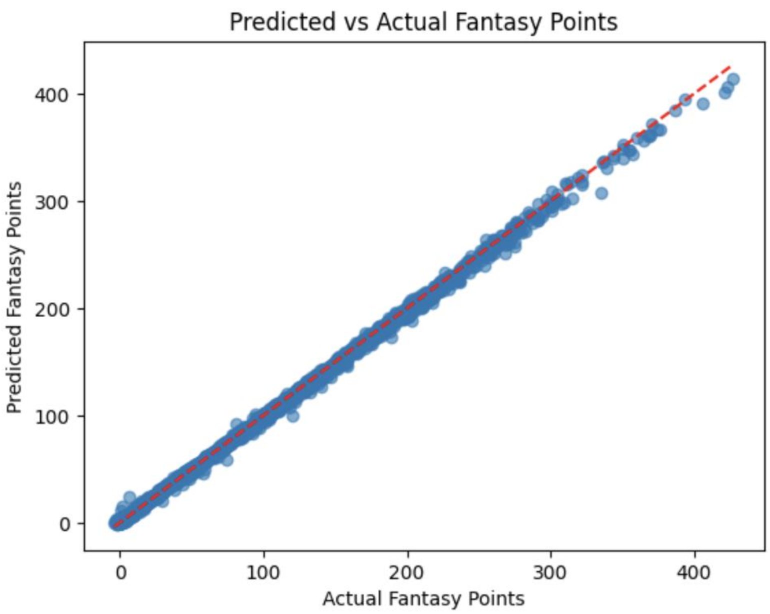
## Appendix

### Figure 1: Training vs Validation Loss




Note: This scatter plot compares validation and training loss from the baseline run.

### Figure 2: Actual vs Predicted Fantasy Points




## References

ELDORADO. (2016, January 27). *A complete history of NFL points, scores, and scoring*.

ELDORADO. https://www.eldo.co/a-complete-history-of-nfl-points-scores-and-scoring.html