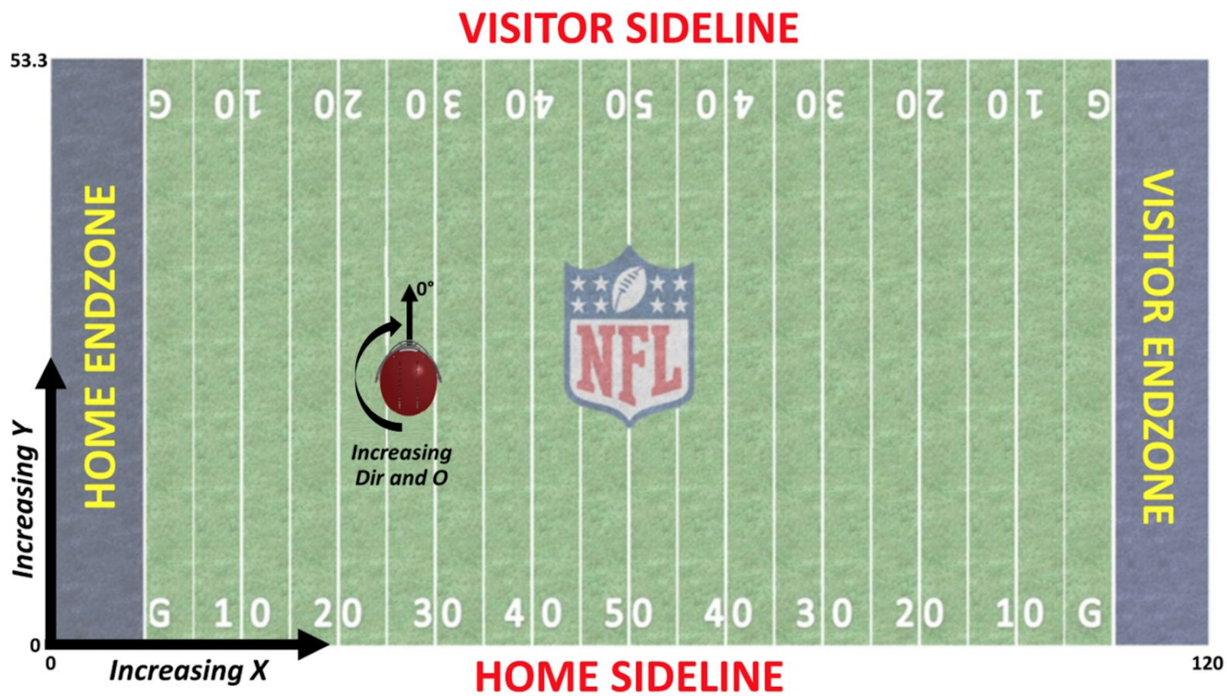
Big Time Rush:
Using Various Deep Learning Architectures to Analyze NFL Rushing Plays

Gokul Ajith (gajith), Harrison Boyer (hboyer), Benjamin Deckey (bdeckey), Akhil Trehan (atrehan1)



Introduction: Using a combination of over 40 metrics for thousands of rushing plays in NFL history, our project goal is simply to predict how far the team will rush for a given play. The interesting aspect of this project is that the data is modeled from the image above, where positions, speed, orientations, field position, etc and this graph replication can lead to insights into the sport that does not solely depend on personnel on the field. These positional metrics can also be combined with overall play conditions including weather, score, location, etc to gain a deeper understanding of the exact conditions under which a play occured. From a specific football standpoint, deeper insight into rushing plays will help teams, media, and fans better understand the skill of players and the strategies of coaches, creating a new analytical approach to football strategy as a whole that can even make the game safer. It will also assist the NFL and its teams evaluate the ball carrier, his teammates, his coach, and the opposing defense, in order to make adjustments as necessary. However, the deeper reason we chose this project was that vast sports data of this specificity is not always available, and this dataset presents the opportunity to analyze a vast number of metrics (as highlighted above from positional attributes to game attributes to weather conditions to even field type) to determine variances in overall similar outcomes. Simply the positions of all players on the field give an interesting graph problem to represent using a model, but the generated weights with additional features can also be very useful in determining which aspects of analysis are worth honing in on over others. This is our biggest motivator, as we attempt to see if deep learning can provide advanced insights into a combination of metrics that can generalize aspects of a professional sport to create a prediction model as to the overall success of the play.

Data: The dataset we used comes from a competition started by the NFL. This data has information from thousands of rushing plays in the NFL, with each rushing play having 22 data points that represent various positional properties of each of the players on the field including X, Y location, speed, distance, acceleration, and orientation. Additionally, each play also has data on 40 other game properties including score, home/away teams, down and distance, field position, formation, turf type, and weather. Our data is large in the sense that each of the 5000 plays (graphs) has 22 nodes that all need to know relative positions of each other and have individual properties to consider.

Methodology: Initially, the positional attributes of all the players on the field enticed us to create a graph model. This was implemented with custom preprocessing objects and DGL with which represented each play in our dataset as its own graph, with every player on the field as a node containing positional data connected by edges that connected the offense to the defense. These graphs were then batched and passed into a Graph Convolutional Network with message passing between the edges. The idea here was to see if this message passing from every source player to destination player would yield a sense of which side would gain an advantage and by how much. We believed that our results from the GCN model could be improved on. Thus we additionally utilized a CNN architecture. For this model, we transformed each play into an image, where the image height were the two types of teams, offensive and defensive, and the width was the size of each type of team respectively. We theorized that the CNN would analyze each player as a pixel where its RGB values are the associated player's features.

GCN Architecture:

- Lifting layer (relu) to 200
- 3 GCN layers (200,200) with message function as a fully-connected linear layer
- Linear layer to 1

- Dropout (0.2)
- For loss, we used Mean Squared Error, and for accuracy, we used Mean Absolute Error to less penalized outliers.

CNN Architecture:

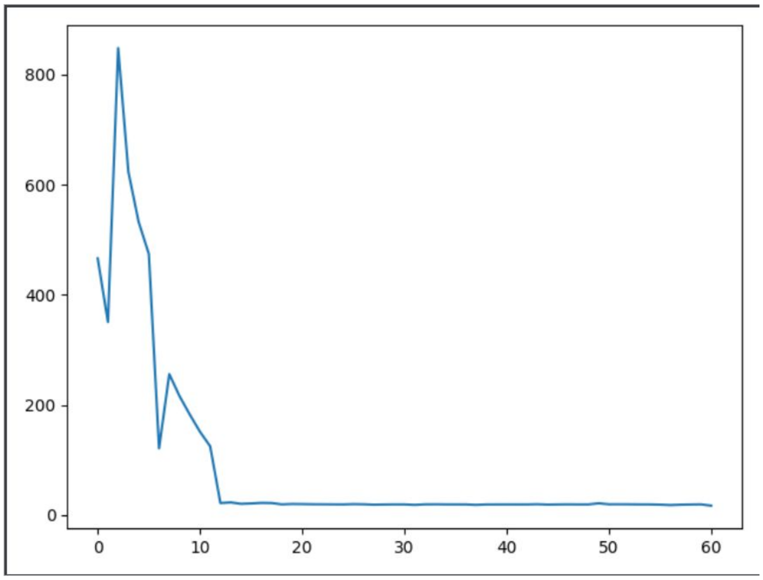
- Lifting dense layer to 100
- CNN layer (20)
- CNN layer (10)
- CNN layer (5)
    - Batch normalization
    - Relu
    - Max 2D pooling
- Flattening layer
- Dense layer to 50
- Dense layer to 1
- For loss we used MSE, and for accuracy we used MAE.

Challenges: Our biggest challenge was the limited data we had access to. We believe that in order for our models to perform better we need a much more robust dataset with definitely more plays. Because of our limited data we found that loss convergence was the main issue with both our MPNN and our Convolution model. In convolution, we additionally had the main problem of representing each play as an image to function with traditional convolution architecture.

Results:

For our GCN the training typically converged after around a few epochs, with an MAE of about 8 yards per play. However, after a closer examination of our testing results, we noticed that our model was relatively good at predicting the plays that only yield a few yards, and failed at predicting plays with a very high yardage yield. This is actually indicative of our model learning that in the NFL, the majority of these rushing plays will be small plays of 3-7 yards, and it predicts these to within one yard often. However, the hardest plays are those that break the first line of defense, as then the initial features which are being trained on have broken down and external factors represent whether the play will go for 20 or 50 yards. For these plays, our model tended to predict still high but conservatively, inflating our testing MAE.

We saw a similar trend in our CNN, where the loss converges after around epoch 10. Our MAE for the CNN was typically around 3 yards per play.



CNN loss per epoch

Reflection:

REFLECTION FOR MPNN:

While the model could not be tested to the extent that we wished due to the data constraints, the MPNN model lends itself to a lot of future work. It has been made generic enough to handle multiple features as well as to have various calculations for the message passing through edges. With more time, we would have attempted to calculate momentum to pass through these edges, but anyone could build off this model with ideas to optimize learning. There is also much room for hyperparameter tuning as future work also.

The results of the model from an ethical standpoint is important to consider as well. If this model were to be widely accepted as accurate, it would change strategic football computation for the whole NFL. This would mean that team coaches on both sides could use self-attention from this model to understand what aspects allow them to gain an advantage over the other side. This could also be used to optimize personnel changes, reducing risks of injury and improving safety in the game.

REFLECTION FOR CONVOLUTION:

We attempted to create an out of the box convolution approach to the prediction of how many yards an offensive team would be able to rush in any given rushing play. One of the largest challenges with this model was its tradition reliance on image data. In order to compensate for this, we decided to represent each play as a conceptual image where the image height was the type of the team (offensive/defensive) and the image width was the number of players on each team (11). Here each player's features would represent its "RBG" value. We also had problems with the limited amount of data that we theorized to have caused early convergence in the CNN model. Due to the vast majority of low label values, the model failed to recognize outliers, and predictions converged to an average value. Although results were adequate, we believe that with more varied data we can achieve more accurate predictions. Other experimental architectures could be used such as graph convolutional neural nets cited in *Learning Convolutional Neural Networks for Graphs by* Mathias Niepert, Mohamed Ahmed, Konstantin Kutzkov.

Ethically if this technology were to expand into repeatable precise correct predictions then there is a possibility of athletes being undervalued by the model. In the case where the model functions correctly for enough majority to be adapted by influential parties interested in prediction of yards then there most certainly be times when organizations will believe that a player may underperform or over-perform than they really will. This can be detrimental to both teams and players.

Link to Github:
https://github.com/gokulajith/BigTimeRush