CS 4701 **Proposal**
Title: **Studying modern NBA positions using neural nets**
Name: Dilip Thiagarajan
NetID: dt372

1 Description

Since the its inception to about 2010, the NBA gameplay was anchored by giants standing tall near the hoop, exemplified by players at center (position) like Wilt Chamberlain, Bill Russell, Kareem Abdul-Jabbar, and Shaquille O'Neal. However, in recent years, the league has been dominated by players on the other end of the height spectrum, being deceptively quick and accurate sharpshooters who extend the floor more widely than before, and teams have had to adjust accordingly by taking out players too slow to keep up. As a result, the center role has dramatically changed, exposing a broad spectrum within the center role, as well as fracturing other roles in a similar manner.

As a result, in studying the different archetypes of players using characteristic statistics, one would find that the traditional split of 5 roles (point guard, shooting guard, small forward, power forward, center) is not an accurate description of the roles the 5 players on the court usually play in today's game. As an example, in 2010, Muthu Alagappan used this kind of analysis to support his argument that **positions don't reflect playing styles**, and are an oversimplification of the true set of roles in basketball.

In his study, Alagappan claimed that there were 13 characteristic roles, and in most further studies aiming to reproduce his results, there have always been about 8 or more characteristic roles that accurately describe a player's playing style, using statistics such as points/assists/steals/rebounds per 100 possessions, player efficiency rating, win shares, and more. What I aim to do with this project is further verify and expand on these studies using neural nets.

2 Approach

As mentioned above, I plan to approach this problem using neural nets. Specifically, I'll analyze the data using autoencoders, which essentially tries to use an encoder/decoder pair to learn a latent, low-dimensional representation of the input data (generally high dimensional). In previous studies, techniques such as clustering and linear discriminant analysis have been used, and while the results were promising, I believe the approximation is not as accurate as it could be, given how the style of the NBA game has fluctuated, especially in recent years.

I will be scraping data from websites such as basketball-reference, which has almost all advanced metrics and per-possession data I will need to sufficiently describe each player. I will eliminate outliers by only considering players who played more than about 50 games.

3 Plan

To evaluate my approach, I'll validate my model parameters using a uniformly sampled subset of my data. If necessary, I'll use more sophisticated validation schemes, such as k-fold validation. Given that autoencoders are essentially trying to memorize the data using a low-dimensional representation, validation is easy in both a discriminative and generative approach. Specifically, I will use a variational autoencoder to evaluate samplings from the latent representation, and how they compare with the original input (generative approach). Additionally, I'll evaluate the final decoding (output) against the original input (discriminative approach).

One thing that might pose an issue is the dimensionality of the original data. Autoencoders work well in general when the input data lies in a low-dimensional subspace, and it's possible that the dimension of the subspace my data will occupy is not a significant reduction from the original space. If I have time, one thing I might try is to impose connections on the player data using the teams they played on (i.e. constructing a graph based on teams, where each team forms a connected, separate component of the graph), and then using that as input. This inherent similarity metric imposition might add further weighting to players who really define the various playing styles more closely. Several studies have been done on autoencoders for graph input, so I don't forsee it being too hard to implement and evaluate if I have about 2 weeks time leftover.

4 Timeline

By the end of this month, I'll have my data retrieved and organized in a way convenient for analysis in Python (specifically, using the PyTorch framework). Additionally, I'll have a toy implementation of a plain autoencoder working. By the middle of November, I'll have the implementation working for the whole dataset, and by the end of November, I'll have the variational autoencoder working for the whole dataset. With the latter two steps, I'll have descriptive evaluations prepared, based on the validation approach described above (using a uniformly sampled subset of the data).