Soccer outcome prediction Using Recurrent Neural Networks

Udit Sharma, *Department of Computer Engineering, Rochester Institute of Technology*

Abstract— **Predicting the outcome of a game has always been an area of immense interest among sports fans as well as the bookmakers. To make an informed guess, researchers have previously tried to use numerous machine learning techniques like Bayesian nets, logistic regression, decision trees classifier and even basic neural nets but none have explored using Recurrent Neural Networks (RNNs) for sports prediction. This paper intends to predict the result of an English Premier League (EPL) game using the 'European Soccer Database' from Kaggle.com. Football pundits have touted form as one of the most crucial factors when trying to predict the outcome of a game. To take this parameter into account and give it more weight, we have formulated this as a time-series problem, trying to leverage the natural architecture of RNN combined with Long-Short Term Memory (LSTM) for sequential problems.**

Index Terms **— Artificial Intelligence, Long-Short Term Memory (LSTM), Machine Learning, Recurrent Neural Networks (RNN), Sports prediction**

I. INTRODUCTION

Soccer is one of the most popular sports in the world, even more so in Europe. Fanbases of larger clubs are in millions and many even travel hundreds of miles to watch away games taking place in other countries. Putting together revenues from stadium attendance, broadcasting rights and sponsors, the European soccer market is estimated to be more than £25 billion (approx. $32 billion) with the English Premier League (EPL) being, by far, the largest contributor. Hand in hand with the game goes the betting industry. The trading in the European betting industry is estimated to be upwards of 700 billion dollars, 70% of this is said to come from football. 5 major leagues are operating in Europe: English, Spanish, German, Italian and French. This paper concerns with the EPL, the top tier league in England.

With such tremendous money involved, the fans and betters must make an informed decision. Football games are full of unpredictability. One moment of magic from a player may change the result completely in their favor. On the other hand, a team may dominate the whole game with abundant chances of scoring created and still end up losing the game. The development of new Machine Learning (ML) techniques coupled with powerful computing technologies has helped fans, betters and pundits to analyze the past data to predict the outcome of the game or the final scoreline.

.

This paper discusses how Recurrent Neural Networks (RNNs) can be used to predict the outcome of an EPL game. This is a multiclass classification problem with win, loss and draw as the possible outcomes.

II. RELATED WORK

Sports analytics has always remained an exciting field of research. Numerous attempts have been made in the field of football and other sports trying to predict different parameters like the outcome, expected goals, number of fouls, free kicks, etc. In this section, we will discuss a few methods most relevant to this research. As far as other sports are concerned, [1] has experimented with several ML algorithms like decision trees, K-Nearest Neighbors (KNN) classifier, Support Vector Classifier (SVC) and reports that SVC performed the best among all the above techniques used to predict the March Madness winners, the annual National Collegiate Athletic Association (NCAA) basketball tournament. Reference [2] draws a comparison to prove that Deep Learning (DL) can prove to perform better than traditional ML techniques. Reference [3] aims at building an Expected goals (xG) model using neural networks, random forests and Support Vector Machines (SVM) using shots data and ELO team ratings and showed results comparable to bookmakers' odds. Many researchers have also analyzed how Artificial Neural Networks (ANN) can be used to predict the outcome of sports [4]. This paper also devises the 'SRP-CRISP-DM' framework for result prediction based on the standard CRISP_DM [5]. Further, Prediction of football scores using ML techniques like Naïve Bayes, LogitBoost, KNN, random forest and ANN is performed in [6]. This paper also concluded that the ANNs performed the best amongst all these classifiers. Lastly, [7] discusses how RNNs along with Long-Short Term Memory (LSTM) can be used to predict the outcome of a football game. Experiments were run using different hyperparameters and accuracy as high as 88% is reported.

III. PROPOSED METHOD

The problem of predicting the outcome of a football game is a classification problem. According to experts, the performance of a team in previous matches is a very important criterion in predicting the output of the next match. Therefore, this paper treats this as a time-series problem. RNNs have a natural architecture for solving time-series problem, hence we decided to choose RNNs for this experiment. Each previous 'matchday' is considered a timestep. A matchday is an instance when each team plays against one other team in the league. So, if there are 20 teams in a league, 10 matches take place in what is called a matchday. Since most matches in a matchday take place simultaneously and are not independent of each other, it is not logical to feed all those 20 matches to a time series model. Therefore, we intend to build a separate for each team. The neural network is fed a feature vector consisting of the statistics of the match at that timestep. We decide to use 4 previous matches to predict the result of the 5th match, therefore, there are 4 timesteps and at each time step, a feature vector consisting of the statistics of the match is pushed into the network and the network produces an output which is be fed as the input to the next layer. The architecture of a many-to-many RNN is shown in Fig. 1. The network receives a new input, x, at every time step, t, computations take place in the hidden layer, h and an output, y for that timestep is produced.

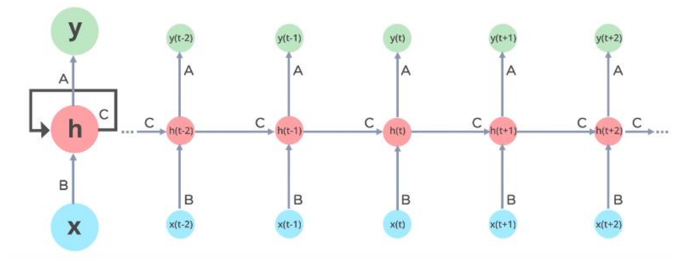

Fig. 1: Architecture of an RNN

A general equation for updates in an RNN can be written as follows,

$$h(t) = g(W * x(t)) + C \quad (1)$$

Further, C can be written as,

$$C = V * x(t-1) \quad (2)$$

where, W and V are weight matrices for t and t-1 layer respectively and x is the input.

We use the many-to-many RNN which produces output after each time step and feed the last output and cell state to the next state. RNNs also face the problem of exploding gradient and vanishing gradient when the gradients become too big or too small respectively while trying to learn long term dependency. To get around this, we use the LSTM. LSTMs consist of contextual state cells that can act as long-term or short-term memory. The output of the LSTM network is regulated by the state of these cells and helps the network to consider the historical context of inputs instead of just the last output.

The architecture of an RNN is generally a simple one. We intend to use the LSTM followed by a fully connected layer. At the last fully connected layer, we have the softmax activation function which returns one of the 3 class (win, loss or draw) from the home team perspective and categorical cross-entropy loss is used to evaluate the function for errors.

IV. RESULTS

A. Dataset

European Soccer Data from Kaggle.com is being used as the dataset for this experiment. This is an SQLite database which consists of match, player and team data from 11 leagues across Europe. For this problem, we are only concerned about the EPL The match table consists of each match stats that took place from the 2008-09 season to the 2015-16 season. Some relevant fields in this table include date, stage (Matchday 1 – 38), ratings of 11 players from each team who took part in the game, shots on-target, shots off-target and cards issued, if any. The player attributes table consists of individual player stats such as overall player rating and specific skill ratings like dribbling, crossing, shooting, etc. The team attributes table consists of team-level statistics. Some of the features that we have experimented on are build-up speed, defense pressure, and chances created. For each team, there is data for 304 matches across the 8 seasons as the team plays 38 matches each season. We use the first 7 years' data for training and validation and last year's data for testing.

B. Hyperparameter tuning

Several combinations of batch size, hidden units, size of embedding, units in each layer, learning rate, epochs were tested. We include only the most successful ones here. Below is a discussion of various parameters used and how they were tuned.

Dropout: Initially, experiments were conducted without taking the dropout and we were able to achieve a near 100% accuracy on the training set but a poor accuracy was seen on the test set. Clearly, the model was overfitting on the training set. Then, dropout was added, slowly incremented and the best results on the testing set were found for a 30% dropout.

Hidden units: Different numbers of hidden units were tried, 4, 8, 16 and 64 LSTM units were tested. For a larger number of LSTM units, the model was able to learn faster but again led to overfitting due to the small amount of available data. The best results were found using 4 LSTM units. Next, we were able to get the best results with 1 fully connected layer with 128 units. Fig. 2 shows the confusion matrix for the best results obtained.

Look back: This defines how many matches prior should be considered by the model to predict the next match. We tested our model using lookbacks of 3, 4 and 5 and was able to get the best accuracy using a lookback of 4.

		Predicted		
		Loss	Win	Draw
Actual	Loss	5	4	1
	Win	4	14	1
	Draw	3	2	4

Fig. 2: Confusion matrix - Best results by the model

Accuracy of 60.5% was achieved using our model. A lower accuracy can be attributed to smaller amount of data which made it hard for the neural network to train. Alternatively, SVM produced a test accuracy of 89.47% on the same features and data.

V. CONCLUSION

An RNN with LSTM has the potential to perform well on such a problem if we have enough data to train a neural network. SVM producing a much higher accuracy reinforces the validity of conclusion that neural network was unable to learn properly due to less amount of data. Future work could include training the same model using more data. Alternatively, other features like cards or possession statistics can be used for this prediction.

REFERENCES

[1] Jordan Gumm, Andrew Barrett, and Gongzhu Hu. “A machine learning strategy for predicting march madness winners”. *16th IEEE/ACIS International Conference on. IEEE. (2015).*

[2] Pablo Bosch. “Predicting the winner of NFL-games using Machine and Deep Learning.”. M.S. Business Analytics Dissertation, Vrije universiteit, Amsterdam, Nederland, 2018.

[3] C. Herbinet “Predicting Football Results Using Machine Learning Techniques”. Individual Project, Dept. Computing, Imperial College of Science, Technology and Medicine, London, 2018.

[4] R.P. Bunker and F. Thabtah, “A machine learning framework for sport result prediction”, Applied Computing and Informatics, vol. 15, pp 27–33, 2019.

[5] C. Shearer, The CRISP-DM model: the new blueprint for data mining, J. DataWarehousing 5 (4), pp 13–22, 2000.

[6] Josip Hucaljuk and Alen Rakipović. “Predicting football scores using machine learning techniques” *MIPRO, 2011 Proceedings of the 34th International Convention. IEEE. (2011).*

[7] Daniel Pettersson , Robert Nyquist. “Football Match Prediction using Deep Learning”, M.S. Dissertation, Dept. Elect. Engg., Chalmers University of Technology, Gothenburg, Sweden, 2017.

[8] Burak Galip Aslan and Mustafa Murat Inceoglu. “A comparative study on neural network based soccer result prediction”. *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007).*

[9] Junyoung Chung et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling”. *arXiv (2014).*

[10] Anito Joseph, Norman E Fenton, and Martin Neil. “Predicting football results using Bayesian nets and other machine learning techniques”. *Knowledge Based Systems 19. (2006).*

[11] Link to dataset: https://www.kaggle.com/hugomathien/soccer