Predicting the football match winner using LSTM model of Recurrent Neural Networks

Geetanjali Tewari
School of Computer Science and Engineering
Vellore Institute of Technology
Tamil Nadu, India 632014
Email: geetanjali.tewari11@gmail.com

Krishna Kartik Darsipudi
School of Computer Science and Engineering
Vellore Institute of Technology
Tamil Nadu, India 632014
Email: krishnakartik1@gmail.com

***Abstract*—Football is one of the most widespread and popular sport, therefore predicting the results of a football match poses an interesting challenge. Prediction is also very useful in helping managers and clubs in making right decision to win leagues and tournaments. This field is growing because of the demands for good predictive accuracy due to large monetary amounts involved in betting. Instead of using the traditional methods based on statistical analysis and machine learning techniques, in this research paper we want to focus on Deep Learning techniques that help predict the result with a high accuracy. We will be using LSTM model of Recurrent Neural Network in our research.**

***Index Terms*—Football Match Prediction, Machine Learning, Deep Learning, LSTM, Recurrent Neural Networks**

I. Introduction

Sports are one of the favourite areas of interest of the population and one of the most popular sports is Football. Hence predicting the results of a football match especially in Champions league is a trend set involving huge monetary amounts in betting. Therefore researchers are working on finding the best prediction technique to predict the result of football match. In this paper we have used deep learning to attain the highest accuracy.

With deep learning comes a hoard of advantages such as its highly scalable nature, this means that the performance of deep learning algorithms dont plateau after a certain amount of data has been trained unlike traditional machine learning algorithms and its renowned capability in supervised learning from labelled data. Since deep learning algorithms basically consist of deep neural networks, they seek to exploit the unknown structure in the input distribution in order to discover good representations, often at multiple levels, with higher-level learned features defined in terms of lower-level features. The hierarchy of concepts allows the computer to learn complicated concepts by building them out of simpler ones. Artificial Neural Networks are the quintessential deep leaning algorithms that take a set of inputs and feed them forward through 1 or more hidden layers to get the output, but this model will not work for us, the explanation of which is as follows.

Lets consider an example for the scenario that is considered in this paper, i.e., football match result prediction, if a team A won the match against team B today and if they are set to play again in a couple of days, there is no provision for an Artificial Neural Network to remember this loss as it happened in a different period of time. If this remembrance has to be coded using an ANN, there has to be a separate set of weights for each instance of time and a method to calculate the current output of that node with the previous value of that node as input. This is where Recurrent Neural Networks come into play. A Recurrent Neural Network is capable of combining various layers of an ANN together with the help of a recurrence function, which uses the current input and the previous output at that layer as inputs. Similarly, this recurrence function can be used to combine all the different time frames into a single layer. So, the input at the present instance of time and the value already obtained after the previous instance of time will be used to calculate the present output. This feature of RNNs acts like a remembering function because the previous output affects the present output in some way. Thus, a RNN considers the sequence before giving an output. But even this has certain drawbacks that can potentially reduce the accuracy.

RNN is still essentially an ANN. Hence, once the final output has been calculated, back-propagation has to be performed to find the error at each level. Thus, the error term for a particular layer is somewhere a product of all previous layers errors. When dealing with activation functions like the sigmoid function, the small values of its derivatives (occurring in the error function) gets multiplied multiple times as we move towards the starting layers. As a result of this, the gradient almost vanishes as we move towards the starting layers, and it becomes difficult to train these layers. A similar case is observed in Recurrent Neural Networks. RNN remembers things for just small durations of time, i.e. if we need the information after a small time it may be reproducible, but once a lot of words are fed in, this information gets lost somewhere. This issue can be resolved by applying a slightly tweaked version of RNNs, the Long Short-Term Memory Networks (LSTM). When RNNs are

faced with new information, they transform the existing information completely by applying a function. Because of this, the entire information is modified, on the whole, i.e. there is no consideration for important information and not so important information. LSTMs are a type of RNNs that make small modifications to the information by multiplications and additions. With LSTMs, the information flows through a mechanism known as cell states. This way, LSTMs can selectively remember or forget things. Thus, in this research paper, we will predict the outcome of a football match with the help of LSTMs and try to tweak it for better results.

Second section of the paper discusses about the related work done which also serves as the base for our proposed methodology. Section 3 explains about the dataset and methodology followed and , and section 4 elaborates our experiments and result analysis. The last section contains our conclusion on predicting the football match winner using LSTM technique of neural network.

II. Related Work

Previously, methods like logistic regression [1] were used to predict the outcome of a football match. Prasetio. D et. al. performed multivariable logistic regression on precisely 4 attributes in of the football dataset. Since it is logistic regression, the output is a sigmoid function and tuples with values above 0.5 were classified as a home win and the others were classified as an away win. So, they estimated the coefficients of the different variables in training and applied them to new tuples in the testing data.

The usage of machine and other predictive learning techniques can be seen in the theses by Michael j Bailey[2]. Here, statistical methods are used to perform analysis on the American Football league and cricket. T-tests, analysis of variance and person coefficients for normally distributed data, chi-square tests for comparison of proportions and Wilcoxon rank sum or Kruskal Wallis tests for non-parametric data.

In 2011,hucaljuk. J et. al. focussed on feature selection ,because the predicting the outcomes of a football match is a difficult problem [3] because determining the number of factors playing an important role is difficult. The initial dataset contained more than 30 features which were reduced down to 20 and then 6 different classifiers-Naive Bayes, Baysian networks, LogitBoost, KNN, Random forest and ANN were applied on the dataset. The highest accuracy was found to be 65% which was of ANN.

In 2014, Chinwe Peace et. al. employed a data mining tool called rapid miner [5] to predict the football match results. This tool allows including as many features as possible. Two techniques ANN and logistic regression were applied thought knowledge discovery database which yielded 85% and 93% prediction accuracy respectively. ANN could predict win, loss and draw while LR could only predict win or lose. A comparison between statistical , machine learning approach and KDD approach has been made which shows that KDD approach give us higher accuracy of 75% where as other approach gives accuracy of only 55% .Here better accuracy results were achieved than[3].

In the research of finding new methodologies of prediction,Jongho Shin et.al. proposed a new way of data selection which with the data collected by the video game industry[5] . Therefore making the use of virtual data collected from a video game (FIFA2015),two types of models were performed, supervised and unsupervised, in supervised learning, two approaches were described: Real Predictor: Machine learning techniques were applied on real data and Virtual Predictor: Machine Learning techniques were applied on virtual data. The hit rate in both of them were 0.75 and 0.83 respectively, which showed that data collected by video games industry could be used to solve real problems. In unsupervised learning, k means clustering was used to identify 5 different types of playing strategies , based on the skills present in the team. It was observed that top teams have very offensive strategies, which was in contrast to NBA, where best teams had good defensive strategies. It was also observed that weaker teams perform better against stronger teams if they use defensive strategies.

Coming to the other methods of classification we walk though [6] the journal of machine learning research in which Felix A. et al. discusses about learning Precise timing with LSTM Recurrent Networks. In order to predict the outcome of the tasks which require the previous information stored time to time, an approach called Recurrent Neural network can be used. LSTM (Long Short-term Memory) is used with RNN because it outperforms RNN on tasks involving long time gaps.Here in this paper an extention to the LSTM has been provided which involved the peephole connections in the internal cells of the LSTM, which enables the layers to see the cell state of LSTM. This makes LSTM very efficient for short term memory dependencies .

A blog written by Colah in 2015 [7] and Analytics Vidhya[8] explains us about the Recurrent Neural Network, how they address the issue of storing the information time to time and reusing it for a new iteration. For the information which have uncertain time gaps: very long or very short,LSTM approach comes to rescue. LSTM approach solves 2 kinds of problems: the problem of long term dependencies, that is when we need more than just the previous information to predict the outcome. LSTM, has a cell state, though which the information can be passed, added and deleted. Therefore whenever any data or information is input to a cell state, in order to modify what to be stored for the output through the state, the first step is to decide what information will be thrown away from the state. This is done by a sigmoid layer called forget gate layer. Then the information to be stored is decided by a sigmoid layer called input gate layer.Then

we put the cell state through tanh layer and multiply with output of sigmoid gates to get our decided output. There are several variations to LSTM techniques for example peephole connections (for short term memory dependencies),coupled forget and input gates,combination of forget and input gate into a single update layer.

III. Dataset

The dataset for the prediction of the football match winner has been taken from [9] , from where we have taken the data of the English Premier League from seasons 2010-11 to 2016-17. The advantage with these football datasets are that the number of matches played in the total tournament and the number of matches played by each team is fixed. This helped us in preprocessing the data and removing the unnecessary information. Each of the league datasets have 380 rows and around 65 attributes and are in chronological order. From this, manual feature generation was done where, information from two or more attributes was merged to produce a new attribute. For example, the values from the 'FTAG'(Full Time Away Goals) and 'FTHG'(Full Time Home Goals), two new attributes 'HTGS'(Home Team Goals Scored) and 'ATGS'(Away Team Goals Scored) were calculated by summing over the attributes through all the rows. Once the necessary attributes had been generated, all the numeric attributes which increased with each week in the league were scaled with the help of the match week. Also, the competitors of each match had their form calculated by finding the win streak of the previous 5 matches and new attributes were produced to record if any of the team were on a hat trick or had won five matches in a row. Finally, for the decision attribute, i.e. the result of the match, One Hot Encoding was done for classification purpose. The various rows that are considered in the final dataset is listed below -

1) HTGS-Home team goal score
2) ATGS-Away team goal score
3) HTGC-Home team goals conceded
4) ATGC-Away team goals conceded
5) HTP-Home team points
6) ATP-Away team points
7) HTGD-Home team goal difference(HTGS-HTGC)
8) ATGD-Away team goal difference(ATGS-ATGC)
9) HM1(-4) - Result of the previous 4 Matches that the Home team has won
10) AM1(-4) - Result of the previous 4 Matches that the Away team has won
11) HtWinStreak3 - Home Team Win Streak 3 (Hat trick)
12) AtWinStreak3 - Away Team Win Streak 3 (Hat trick)
13) HtLossStreak3 - Home Team Loss Streak 3 (Hat trick)
14) AtLossStreak3 - Away Team Loss Streak 3 (Hat trick)
15) HtWinStreak5 - Home Team Win Streak 5
16) AtWinStreak5 - Away Team Win Streak 5
17) HtLossStreak5 - Home Team Loss Streak 5
18) AtLossStreak5 - Away Team Loss Streak 5
19) HTGD - Home Team Goal Difference
20) ATGD - Away Team Goal Difference
21) DiffPts-(HTP-ATP)
22) DiffFormPts- HTFormPts- ATFormPts

IV. Methodology

Every neural network model will have a few parameters that don't change throughout the execution of the model and even through the calculation of accuracy. These parameters are called hyper-parameters. Once, the model has been executed with a certain set of hyper-parameters, these parameters can be tweaked and played around with, to get better insights and accuracies into how the neural network model behaves.

Recurrent Neural Networks have an ability to simulate remembering older instances of data. Hence, it makes sense to use it in applications like Natural Language Processing. In the previous mentioned application, if a sentence is given as an instance, it is pretty obvious that the model should make sense of the sentence in its original order. This can be achieved by feeding the model, word by word, taking advantage of the properties of RNN. For normal datasets in the form of relational databases, a set of attributes can be sent in chunks to the model. This will simulate a sequence of attributes within an instance of data. The equation for the value of a state of a RNN cell is given by

$$h_t = f(h_{t-1}, X_t)$$

where h_t is the new value of the state, h_{t-1} is the previous value of the state and X_t is the input. Now, RNNs can not only remember the sequence in one instance, but also the sequence of the various instances coming in, because the nodes storing the weights and the activation functions are same. With respect to football match outcome prediction, the dataset is a relational database as mentioned in section III. Also, the order of the various columns is unimportant as it doesn't affect the outcome of the match. This aspect of sequence dependence has also been tested in the section V.

Long Short-Term Memory (LSTM) cells are a development over RNNs. In RNN, the output of every state is directly sent as the h_{t-1} to calculate the value of the new state. LSTMs are cells that implement RNNs and in addition, also perform some calculations with this previous output and the current input to generate the new output. The LSTM cells are mainly composed of three gates

- Forget Gate - This is responsible for forgetting the unnecessary information when moving to a new sequence. This gate uses a sigmoid function.
- Input Gate - This is responsible for providing new information to the cell. This gate uses a tanh function and a sigmoid function.
- Output Gate - This is responsible for shaping the final output from the cell state calculated in the previous two states. This also uses a tanh function and a sigmoid function to give the final output of the state.

Fig. 1. LSTM Cell

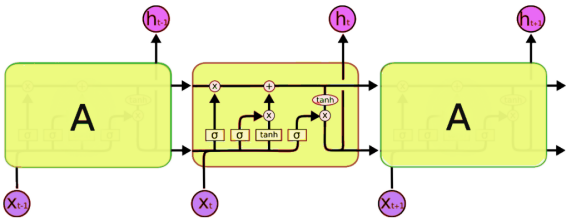

Figure 1 gives a pictorial representation of the LSTM cell. The leftmost sigmoid (σ) function is the forget gate. The input gate is represented after this. It consists of a sigmoid (σ) function and a tanh function. The results of both of these are matrix multiplied and added to the cell state. Finally the output gate also consists of a tanh function and a sigmoid (σ) function. But unlike the input gate, the tanh function is applied to the cell state and the σ function is applied to the product of h_{t-1} and X_t. These two are multiplied to get the final output.

The implementation of the above described models is done using the Tensorflow library in Python. This library handles all data in different forms of arrays. Tensorflow is capable of handling extremely large data with the help of a NVIDIA GPU. A code written in Tensorflow works in two steps. Initially,a computational graph is formed, where all the connections among various layers of the neural network and the calculations of the weights are done but none of these variables have values. They are initialized with Tensorflow objects without any numberical values. The second step is the execution, where a session function is run and all the variables that have to be calculated are passed as parameters. This generates numerical values that are assigned to the variables.

V. Experiments and Results

The experiment has been performed on the above dataset using basicLSTM cell from the tensorflow library. The model was run on various set of hyper-parameters to find out the best model. The hyper-parameters include -

1) n_classes - number of output classes
2) batch_size - number of rows being fed to the model in one iteration
3) hm_epochs - number of epochs the model runs for
4) chunk _size - number of attributes in each chunk
5) n _chunks - number of chunks in one instance of data
6) rnn _size - number of LSTM cells in the hidden layer of the recurrent neural network

Initially, Both the Artificial Neural Network and Recurrent Neural Networks were run for different batch and chunk sizes to support the theme of this paper that LSTMs perform better than ANN. So, table I shows the various train and test accuracies for 2 batch sizes namely 30 and 60. Table II shows the various train and test accuracies for batch sizes from 1 to 124. Table III shows the various train and and test accuracies with chunk_size = 3 and n_chunks = 9, for batch sizes from 1 to 124. It's pretty clear from this, that ANN (table I) begins to overfit the training data as the train accuracy is extremely high whereas the test accuracy is still lesser than both the RNN models (table II and III). Now, the accuracies in II are higher and as discussed in section IV were initially expected to be high because the various attributes in the table need not necessarily be in any order. Hence this model, with chunk size as 27 was considered for further analysis. IV shows the accuracies of the model with varying size of the hidden layer. Here,for increasing rnn_size, though the train accuracy is increasing, there is no sufficient increase in the test accuracy. This suggests that the model is over-fitting the data. Hence, for the present dataset the best model values are as follows -

1) n_classes - 2
2) batch_size - 1
3) hm_epochs - 10
4) chunk_size - 27
5) n_chunks - 1
6) rnn_size - 64

TABLE I
ANN

Batch Size	30	60
Accuracy Train	0.99462366	0.983871
Accuracy Test	0.79125	0.76125

TABLE II
RNN WITH CHUNK SIZE 27

Batch Size	1	30	60	124
Accuracy Train	0.9811828	0.7688172	0.74139786	0.7354839
Accuracy Test	0.8075	0.69875	0.69375	0.69

TABLE III
RNN WITH CHUNK SIZE 3 AND N_CHUNKS AS 9

Batch size	1	30	60	124
Accuracy train	0.922043	0.6973118	0.6655914	0.6704301
Accuracy test	0.76125	0.6525	0.64625	0.6525

TABLE IV
RNN WITH DIFFERENT SIZES OF HIDDEN LAYER

RNN Size	32	64	128	256
Accuracy Train	0.96505374	0.9811828	0.98387	0.9919355
Accuracy Test	0.79875	0.8075	0.79875	0.805

VI. Comparison With Existing Models

In [3] the accuracies a host of machine learning techniques that were used to predict the outcome of a football match.

According to the accuracies, Hucaljuk et. al. showed that the Artificial Neural Network classified the winners with the best accuracy of 68.8%. Compared to this, the accuracy achieved with the LSTM form of RNNs show a major improvement with a test accuracy of 80.75%.

VII. Conclusion

The popularity and international effect that football has, makes it an interesting problem to solve. Moreover, the number of factors that affect the outcome of a match is enormous.
From the results, we can say that RNNs with LSTM show a visible and obvious advantage over the original ANN and the traditional machine learning . Hence, other than the mainstream uses of LSTMs, this particular path of using it for prediction of the outcome for sporting events has also showed promising results.
This model can still be improved. One way to do that is, by using a better set of attributes. They can include statistics of each individual player. This can also help in predicting the form a particular player from season to season. Larger datasets will also help in better training the neural network.

References

[1] Prasetio, D. (2016, August). Predicting football match results with logistic regression. In *A*dvanced Informatics: Concepts, Theory And Application (ICAICTA), 2016 International Conference On(pp. 1-5). IEEE.

[2] Bailey, M.J. (2005). Predicting Sporting Outcomes: A Statistical Approach. Swinburne University of Technology: Faculty of Life and Social Sciences.

[3] Hucaljuk, J., & Rakipovi, A. (2011, May). Predicting football scores using machine learning techniques. In MIPRO, 2011 Proceedings of the 34th International Convention (pp. 1623-1627). IEEE.

[4] Igiri, C. P., & Nwachukwu, E. O. (2014). An improved prediction system for football a match result. IOSR Journal of Engineering (IOSRJEN), 4, 12-20.

[5] Shin, J., & Gasparyan, R. (2014). A Novel Way to Soccer Match Prediction. Stanford University:Department of Computer Science

[6] Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. Journal of machine learning research, 3(Aug), 115-143.

[7] https://www.analyticsvidhya.com/

[8] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[9] http://football-data.co.uk/data.php