

基於 Feedforward Neural Network 預測籃球員位置

吳定軒

109318095

roy87816@yahoo.com.tw

施昀楷

109318044

ian860916@gmail.com

台北科技大學電機工程系碩一 412 研究室

第十二組

Abstract—使用前饋神經網路來預測籃球員位置，將適當的人才放在最適合的位置，發揮最大的效果。

Keywords—前饋神經網路

I. INTRODUCTION

在快速變遷的社會底下，不只科技日新月異，就連運動也跟以往不同，如今各國的籃球聯盟(NBA、CSKA、LEGA、CBA、P league.....等等)，球隊不再僅依靠球員身材來區分位置，且隨著時代演進，會投三分的中鋒亦或是會搶籃板的後衛等特色鮮明的球員也越來越多，所以我們團隊決定使用前饋神經網路結合球員的各項數據，試著對球員位置進行分類，同時也可找出潛力新秀除了當前位置以外是否適合其他位置。

II. PROBLEM STATEMENT

由於現今的籃球選手風格多樣，位置的界線越來越模糊，我們想知道在此情況下，是否仍能依靠球員數據進行位置分類及預測，因此我們將身高、體重、投籃命中率、三分命中率、平均得分、平均籃板、平均助攻、抄截、阻攻等九筆數據做為輸入參數，藉由前饋神經網路對球員主要位置進行判斷，以找出符合現今球員位置的分界。

III. DATA PROCESSING

A. 訓練資料

為了準確的預測，需要一個可信度高且資料量齊全的資料來源，所以我們使用了 NBA 官方網站，本年度的數據來製作我們的訓練資料[1]。

總共使用了十種數據，其中身高、體重、投籃命中率、三分命中率、平均得分、平均籃板、平均助攻、抄截、阻攻等九種數據做為輸入參數，而球員所屬位置則作為 Groundtruth。

由於輸入參數的尺度不同，可能會使得參數的重要性有所差異，為了防止 overfitting 的情況發生，將輸入數據做了 standardize，先將每個輸入變數 x 減去樣本平均數 μ ，再除以樣本標準差 σ ，如公式 1，使輸入資料的平均值為 0，標準差為 1。

$$\text{Standardize} = \frac{x - \mu}{\sigma} \quad (1)$$

過程中我們發現有些樣本可信度較低，其原因為上場時間太少導致數據波動過大，例如：命中率為百分之百；數據皆為零的狀況，這些情況應當不考慮，所以我們將其剔除。

B. 測試資料

測試資料共分為兩個部分，第一個部分為 2021 年 NBA 選秀前六十名潛力新秀[2] 的資料。第二個部分為台灣本土聯盟 P.LEAGUE+ 共四十名球員的資料[3]。

C. Groundtruth

我們將球員位置進行編碼，前鋒為 0、中鋒為 1、後衛為 2，再對其進行 One-hot，將其以三個參數 F、C、G 表示，如表 1：

表 1：位置的 One-hot 編碼

參數 位置	F	C	G
前鋒(0)	1	0	0
中鋒(1)	0	1	0
後衛(2)	0	0	1

IV. NEUEAL NETWORKS STRUCTURE AND TECHNICAL DETAILS

起初我們先使用 2 的倍數來架構網路，並且在層與層之間進行 dropout，為了滿足網路的對稱性在倒數第二層加上一層含有九個神經元的網路(對應 9 種輸入參數)，最後使用 softmax 分為三類。

Layer (type)	Output Shape	Param #
dense_61 (Dense)	(None, 256)	2560
dropout_46 (Dropout)	(None, 256)	0
dense_62 (Dense)	(None, 512)	131584
dropout_47 (Dropout)	(None, 512)	0
dense_63 (Dense)	(None, 1024)	525312
dropout_48 (Dropout)	(None, 1024)	0
dense_64 (Dense)	(None, 512)	524800
dropout_49 (Dropout)	(None, 512)	0
dense_65 (Dense)	(None, 256)	131328
dropout_50 (Dropout)	(None, 256)	0
dense_66 (Dense)	(None, 9)	2313
dropout_51 (Dropout)	(None, 9)	0
dense_67 (Dense)	(None, 3)	30
Total params: 1,317,927		
Trainable params: 1,317,927		
Non-trainable params: 0		

每一層的 activation function 使用了 relu，並將每個權重的初始值以 random uniform 來設定，並採用 cross-entropy 來計算 loss，梯度下降方面由於 SGD 的訓練時間較長，經過反覆測試後採用 Adam 所表現的效果較佳，最後為了預防 overfitting 使用了早停技術。

訓練過程中我們發現資料集中各個位置的樣本數有一定差距，從表 2 中能看出中鋒的樣本數過少，造成我們很難預測出中鋒的結果，所以為了預防此狀況發生，我們對其權重依照比例反比進行加成，我們先將樣本數 {177,53,209} 找出最小公倍數，再各別除以樣本數，求出各自的加成權重 {1.96,6.38,1.66}。

表 2 各類別樣本數及加成權重

	前鋒(0)	中鋒(1)	後衛(2)
樣本數	177	53	209
加成權重	1.96	6.38	1.66

V. EXPERIMENTS

在附檔中共含十份檔案，分別為 source1.ipynb、source2.ipynb、model.hdf5、data1.txt、data2.txt、data3.txt、data1.csv、data2.csv、data3.csv、requirement.txt，下表是各個檔案之簡介。

表 3 各檔案簡介

檔名	內容
source1.ipynb	Model_train
source2.ipynb	Model_test
model.hdf5	Final.ver model
data1.txt/ data1.csv	Train_data(NBA)
data2.txt/ data2.csv	Test_data(NBA.Draft)
data3.txt/ data3.csv	Test_data(P.League+)
requirement.txt	version

A. 訓練/驗證資料的準確率及損失結果

我們總共的 data 數量有 432 筆，其中的 90% 作為訓練資料，10% 作為驗證資料。根據圖 1 可以看出準確率不管是在訓練資料及驗證資料上都表現出良好的成果，訓練資料的準確率為 87%，驗證資料的準確率為 82%，而在圖 2 中 loss 的表現上也確實降低。

表 4 訓練/驗證之準確率/損失

Train Accuracy	Val Accuracy	Train Loss	Val Loss
87%	82%	0.31	0.63

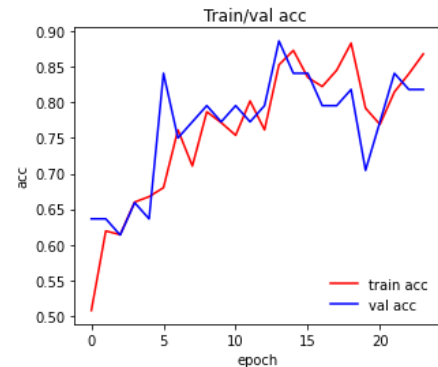


圖 1 訓練/驗證之準確率

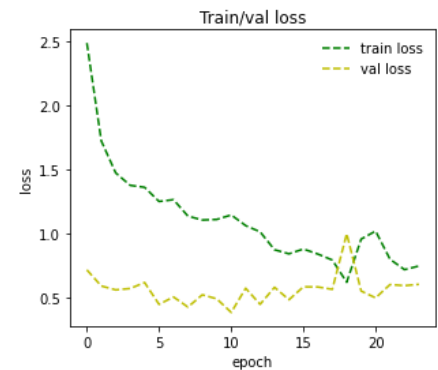


圖 2 訓練/驗證之損失

B. 測試資料

測試資料分為兩部份，2021 年 NBA 選秀前六十名潛力新秀及 P.League+ 聯盟四十位選手資料，我們想知道此模型是否可以運用在不同聯盟。

一、預測 NBA 選秀前六十名潛力新秀

預測 NBA 選秀前六十名潛力新秀的準確率為 76%，值得注意的是在預測錯誤的 26% 中，我們發現第二高的機率與 Ground-truth 完全相符，我們也列出幾個例子。

正確：

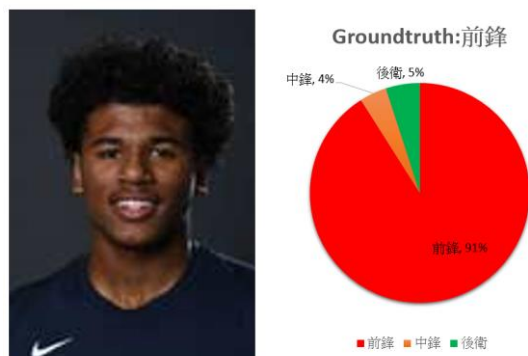


圖 3 正確分類之樣本-1

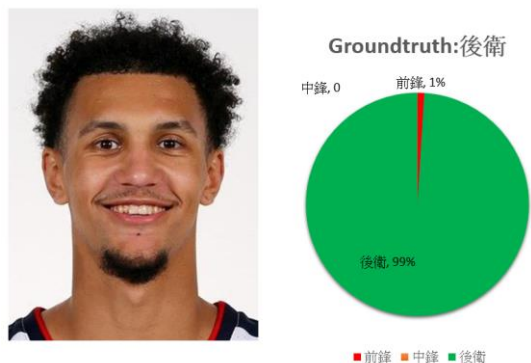


圖 4 正確分類之樣本-2

錯誤：

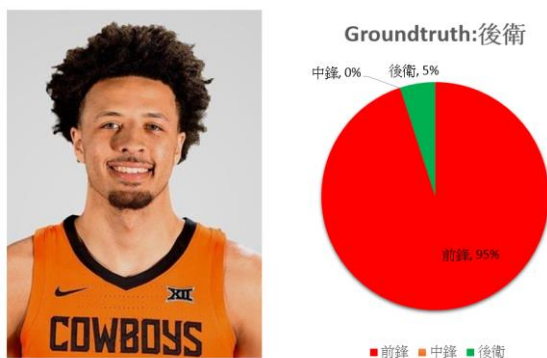


圖 5 錯誤分類之樣本-1



圖 6 錯誤分類之樣本-2

從圖 5、圖 6 中可以發現即使是錯誤的分類，第二高的機率也與 Ground-truth 完全相符。

二、 預測 P.League+聯盟四十位選手資料

預測 P.League+聯盟四十位選手資料的準確率為 51.7%，以結果而言是非常差的，我們認為其原因在 P.League+各方面的數據與訓練集中的資料差異過大，NBA 中鋒的平均身高為 215cm，P.League+中鋒的平均身高為 200cm，加上 P.League+的樣本數不夠多，才會導致此情形發生。

表 5 不同測試集之準確率

Association	Accuracy
NBA.Draft	76%
P.League+	51.7%

正確：

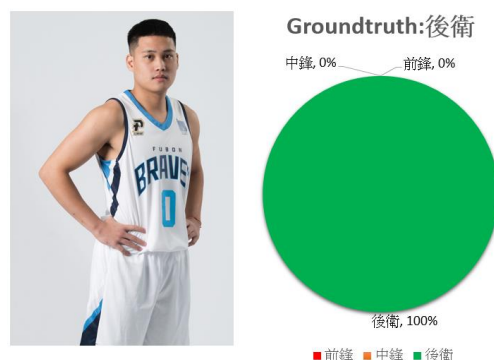


圖 7 正確分類之樣本(P+)-1

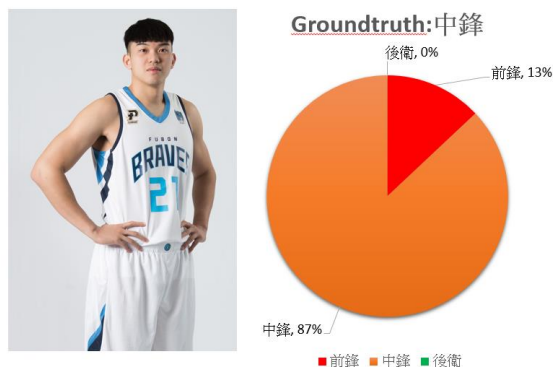


圖 8 正確分類之樣本(P+)-2

錯誤：

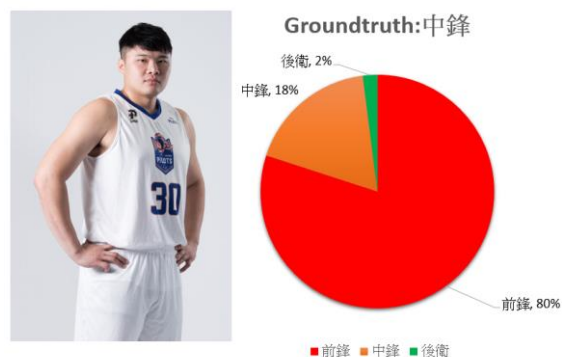


圖 9 錯誤分類之樣本(P+)-1

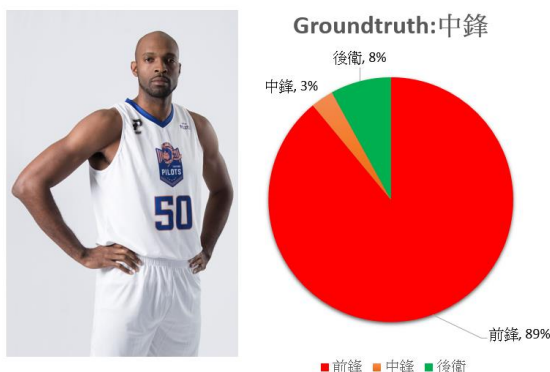


圖 10 錯誤分類之樣本(P+)-2

相較於 NBA.Draft 資料集來說，P.League+由於與訓練集的資料差異過大，所以即使第二高機率的類別也不一定與 Groundtruth 相同。

C. 其他測試

論文中使用了加成權重{1.96,6.38,1.66}，起初沒有考慮加成權重時，預測結果中完全沒有出現中鋒的狀況，我們進一步思考，發現訓練時未考慮樣本數{177,53,209}之間的差異，所以造成了這樣的錯誤。

預測	實際	對錯	前鋒	中鋒	後衛
2	2	correct	[0.01	0.	0.99]
0	0	correct	[0.93	0.05	0.02]
2	2	correct	[0.21	0.	0.79]
2	2	correct	[0.02	0.	0.98]
2	0	wrong	[0.45	0.	0.55]
0	1	wrong	[0.93	0.07	0.]
0	0	correct	[0.85	0.09	0.06]
0	2	wrong	[0.91	0.	0.09]
2	2	correct	[0.04	0.	0.96]
2	2	correct	[0. 0.	1.]	

圖 11 w/o 加成權重

從圖 9 可以發現，預測時完全沒有出現中鋒(1)的情況。

從訓練及驗證上來說，有考慮加成權重時的 Train Accuracy 及 Val Accuracy 會比沒考慮時高上 5% 左右。

表 6 加成權重對訓練的影響

method	Train Accuracy	Val Accuracy
w/o 加成權重	82%	78%
w/ 加成權重	87%	82%

在測試集上來說，不管是 NBA.Draft 或 P.League+ 的表現，都是考慮了加成權重之後的準確率較高，P.League+ 更是相差了將近 15%。

表 7 加成權重對不同測試集的影響

method	NBA.Draft	P.League+
w/o 加成權重	71%	37%
w/ 加成權重	76%	51.7%

在梯度下降法方面起初我們是使用 SGD 來進行運算，嘗試了各種 learning rate = {0.01,0.1,1,10}，我們發

現除了 0.1 以外的 learning rate 效果都很差，我們推測其原因在於太小的 learning rate 會造成陷入局部最低點無法跳出，太大的 learning rate 跳過全局最低點導致無法到達。

表 8 不同 lr 對 Acc/Loss 的影響

SGD lr	Accuracy	Loss
0.01	48%	2.45
0.1	88%	0.78
1	32.74%	2.82
10	28.93%	20.87

我們也試著使用不同的梯度下降方法，發現使用 Adam 跟 RMSprop 時，不管是訓練或測試的準確率都與 SGD 匹敵(表 9)，其中 Adam 所需的訓練週期僅需 SGD 及 RMSprop 的不到一半(表 10)，所以在多方考量下我們決定使用 Adam。

表 9 不同梯度下降法對測試資料的 Acc

method	NBA.Draft	P.League+
SGD(lr=0.1)	78%	48%
Adam	76%	51.7%
Adagrad	26%	45%
RMSprop	76%	55%

表 10 不同梯度下降法所需訓練週期

	Accuracy	Epoch
SGD(lr=0.1)	88%	80
Adam	87%	20
RMSprop	86%	37

VI. CONCLUSION

經過我們的多方嘗試，我們成功地設計出了可以預測及分類籃球員位置的模型，不僅限於身高及體重上，而是考慮球員整體的數據來進行預測。起初我們也在考慮究竟是使用哪一種(SGD、RMSprop、Adam)方法來設計網路，可以達到最好的效果，但經過我們討論認為不應該反覆的使用相同資料集進行測試，這樣會使得測試集不夠公正，所以我們決定選擇運算速度最快的方法(Adam)來建構我們的網路。

我們唯一的遺憾是在於此模型並不能適用在各個籃球聯盟上，就如我們上文所說，各國籃球聯盟(NBA、CSKA、LEGA、CBA、P league.....等等)的強度不同，像是亞洲選手普遍與歐美選手的身體素質差異過大，更進一步來探討，各個籃球聯盟的打法也不相同，有些著重於防守；有些著重於進攻等等，風俗民情的不同導致數據上的差異，所以非常可惜我們的模型只適用於歐美選手。

將來我們希望能夠獲取更多的相關資訊，使模型不只適用於歐美選手，因為目前所看到的所有數據集，都是我們一個一個從網站上擷取下來的，有些聯盟甚至不提供上述提到的數據，這也是為什麼我們的模型只適用於歐美選手，若可以獲取更多的資料集，相信我們一定能更加全面的預測，達到我們期望的效果。

VII. REFERENCE

- [1] <https://tw.global.nba.com/teamindex/>
- [2] <https://www.nbadraft.net/nba-mock-drafts/>
- [3] <https://pleagueofficial.com/>