野球の勝敗予測システムの設計
ー投手起用を例としてー

2018SE105 横井秀哉
指導教員：野呂昌満

1　はじめに

スポーツ界ではデータ解析を用いる動きが盛んになっている．[1][2][3] 野球においても相手打者の打球方向のデータを分析して，打球が飛びそうなところに守備位置を変えて守っていたり，相手投手との相性をデータで分析してなど導入が進んでいる．こうしたデータは，チームとしてだけでなく選手も活用している．球場にトラックマンと呼ばれる弾道測定器を導入して専用のレーダーによってボールをトラッキングしあらゆるデータを取得している．例えば，投手の場合はリリースポイントの位置，球速，ボールの回転数，変化球の変化量を計測してチームの強化に活用しており，今後のプロ野球では IT やデータの活用がますます重要になってくるだろう．

近年は様々な例で人間を模倣できる人工知能が注目されている．福田ら [3] は，野村スコープというコンテンツを模倣して人工知能を用いた配球予想システムの開発を行った．このように人工知能を応用することは，これから野球をもっと楽しむためのコンテンツとして重要になっていくだろう．

本研究では，先発投手のデータを分析する．野球の試合において先発投手は 9 イニングのうち半分以上のイニングを投げるのでとても重要な役割を担っている．先発投手がどのようなピッチングをするかで試合の結果が大きく変わると考える．

本研究では，先発投手の投球内容をデータ分析し，試合に勝ったのか負けたのか結果を予想するシステムを作成し，その有効性の確認を行うことである．本研究の技術課題は以下のとおりである．

1. ニューラルネットワークにおける特徴量の決定
2. ニューラルネットワークの設計
3. 設計したニューラルネットワークの妥当性検証

2　システムの設計

2.1　ニューラルネットの構造

ニューラルネットワークの構造としては入力層，隠れ層，出力層である．全体で 3 層のニューラルネットワークを設計する．入力層は 1 層でユニット数は２５９個である．２５９とは特徴量の数のことである．隠れ層は 1 層でユニット数は入力層よりも多く２５６個とした．隠れ層の活性化関数は，勾配消失問題を起こしにくく正の領域では微分した値が 1 になり，負の領域では微分した値が 0 になる ReLU 関数を 用いる．隠れ層の活性化関数では ReLU 関数を用いるが，本研究の出力は「勝利」,「負け」,「引き分け」の３つの分類であり出力ユニットは３つであるので出力層の活性化関数はソフトマックス関数を用いる．ソフトマックス関数は出力が 0 と 1 の間を取るように変換し確率として扱うことができるので出力層の活性化関数として最適だと考えた．そして，誤差関数を最小化させるために最適化アルゴリズムは Adam を用いる．

2.2　対象データ

本研究では, 野球速報の Web サイトを対象にシステムの設計を行う．Web サイトで公開されている情報の中から必要な情報をを抽出する．Web サイトは使いたいデータがまとめて掲載されていた nf3-Baseball Data House を使用する．Web サイトから抽出する特徴量は以下の 9 個である．

- 投球数
- 投球回
- 自責点
- 被安打数
- 被本塁打
- 奪三振数
- 与四死球数
- ホーム/ビジター
- 投手名

この特徴量は，先発投手の投球成績を見る上で必要だと想定されるものを抽出した．投球数, 投球回, 自責点, 奪三振数は投手がどの程度試合を作り上げたのか見るため使用した．被安打数，被本塁打数，与四死球数はランナーの出塁などで失点に繋がり勝敗予測に影響するため使用した．

またホーム/ビジターも特徴量に追加した．これは，各球団のホーム球場の試合は投手が慣れているマウンドで投げることが出来ることやファンの応援が試合に影響していると考えた．実際に成績をみても大半の球団はホームで勝ち越しをしているので予測精度が上がるのではと考えた．以上，９個の特徴量を用いていく．特徴量の相互関係を学習していくことで，以下の図 1 のようなニューラルネットワークを使い勝敗予測システムを提案する．

3　実現

3.1　開発環境

ニューラルネットを作成するにあたり，Python には多くのライブラリが存在する．その中で今回は近年公開されたライブラリであり，インターネット上にも多くのプログ

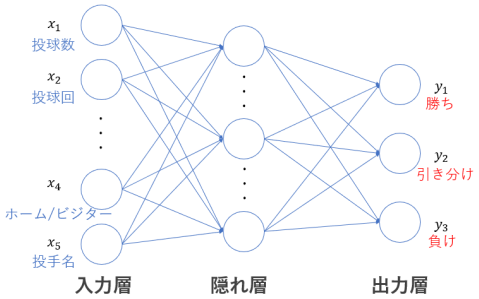

図 1　ニューラルネットワークの構造

ラムが掲載されており，参考にしやすいというメリットがある Pytorch を用いた．[3]

3.2　データセットの作成

実装には大量のデータが必要となる．ニューラルネットに学習させるための野球についてのデータを収集する Web スクレイピングプログラムを Python で作成した．

データの取得には nf3-Baseball Data House という Web ページの各球団の先発一覧表を使用する．この Web ページの URL を指定し，その URL が無効でないときに HTML からデータ収集を行った．

次に，データ取得には今回必要のない情報まで取得しているので Pandas という Python ライブラリを用いて表形式にしてデータの加工を行う．

本研究で使用しないデータの削除，中止になった試合は NaN として一行で書き込まれていたので NaN の削除，勝敗引き分けと先発投手名は文字列になってしまっているので，ダミー変数化をして「0」か「1」に分類する．このような加工を行うことで１７１６行１６２列の表を作成した．

4　実験

4.1　結果

保存したデータより５０％を訓練データ，残りの５０％をテストデータにランダムで分けた．入力を１５９個，出力を 3 個としてディープラーニングを行った結果以下の図 2 と図 3 なった．

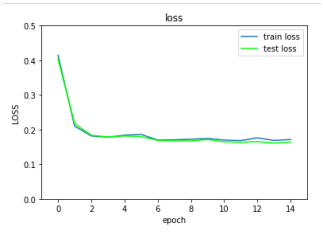

図 2　Loss 関数

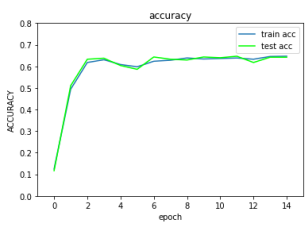

図 3　Accuracy

まず図 2 は損失関数の値をグラフにしたものである．青い線が訓練中の損失値，緑の線がテスト中の損失値である．このグラフはエポック数を重ねるごとに値が小さくなっているが損失関数は約２０％程度となった．そして，図 3 は正答率のグラフである．青い線が訓練中の正答率，緑の線がテスト中の正答率である．正答率は約６５％となった．エポックが進むごとに緩やかに正答率が上がっていき，訓練データとテストデータの損失値の開きが少ないので過学習なく学習を進めていくことができた．

5　考察

今回は，過学習をすることなく学習を進めていきエポックごとに緩やかに精度をあげていった．しかし，本研究は訓練データ 858 個，テストデータを 858 個，球団や日付などをランダムに分けて用いたがエポック数が進んでも正答率が６５％で頭打ちとなってしまった．６５％ではシステムの精度として高いとは言えないだろう．精度が上がらなかった要因としてはデータ不足や特徴量などが挙げられる．データ不足により投手の様々な投球内容を学習することが出来なかったと考える．本研究で使用した特徴量は試行錯誤的に決定したが試合の勝敗に影響する別の特徴量があると考える．相関係数などを用いて勝敗への影響度が強いデータを調べ，特徴量として抽出するなど工夫した分析をしていく必要がある．また，２０２１年シーズンは９回打ち切りというルールがあったので，先発投手が早めにマウンドを降りる試合が多くなってしまったり，引き分けの試合が多くなったので予想が複雑化し正答率が上がらなかったかもしれない．

6　おわりに

本研究は，先発投手の投球内容をデータ分析し，試合の勝敗結果を予想するシステムを作成し，その有効性の確認を行うことを目的とした．

今後の課題としては，さらに精度を上げていくために特徴量の追加やニューラルネットのハイパーパラメータの最適化することである．

参考文献

[1] Michael Woodham，Jason Hawkins，Ankita Singh，Shayok Chakraborty：『When to Pull Starting Pitchers in Major League Baseball? A Data Mining Approach』．18th IEEE International Conference on Machine Lerning and Applications(ICMLA)，Department of Computer Science，Florida State University，2019.

[2] Ganeshapillai Gartheeban,John Guttag：『A data-driven method for in-game decision making in MLB:when to pull a starting pitcher』．ACM SIGKDD International Conference，2013.

[3] 福田大知，野尻雅音，齋藤直紀，鈴木才都，稲垣隼基：『システム情報科学実習グループ報告書』．公立はこだて未来大学，2016.