

ブロックチェーン

2. ANN

3. 予測

4. 実験

5. 結論

論文紹介

Horse racing prediction using artificial neural networks

川口 晏璃 (Anri Kawaguchi)
u255010@st.pu-toyama.ac.jp

富山県立大学 電子・情報工学専攻 情報基盤工学部門

August 2, 2022

背景

人工ニューラルネットワーク（Artificial Neural Networks:ANN）は、天気予報、移動時間、株式市場などの予測に使用されている。また、スポーツの試合やレースなどの結果の予測にも適用されている。

目的

競馬に NN の教師あり学習アルゴリズムを適用し比較する。

ANN

ANN は、脳のニューロンと呼ばれる高度に接続された多数の要素で構成されていることから発想を得た。

データから機能（入力・出力）を実行することを学習する適応システムである。アダプティブとは、パラメータが操作中に変更されることを意味し、トレーニングフェーズと呼ばれる。

トレーニングフェーズ後、ANN のパラメータが修正され、システムが展開され問題を解決する。

NN は、入力のセットを適用すると、目的の出力のセットが生成されるように構成する必要がある。重みを設定する方法は、先験的な知識を用いて明示的に設定する方法と、教育パターンを与え、学習ルールによって重みを変更させることでトレーニングする方法がある。

2.1 学習アルゴリズム

4/14

学習アルゴリズム

- 1 NN にトレーニングセットと出力が提供される教師あり学習
- 2 外部ソースからの支援なしに入力データのパターン
- 3 特徴を発見することを目的とする教師なし学習

重みの更新は，一般的に次のように説明される．

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n)$$

2.1.1 最急降下法 (BP)

5/14

ブロックチェーン

2. ANN

3. 予測

4. 実験

5. 結論

この学習アルゴリズムでは、学習の反復が2つのフェーズで構成される。フォワードパスは、学習パターンごとにNNの出力を計算する。バックプロパゲーションは、出力層から入力層に向かって伝播し、重みが調整される。

$$\varepsilon(n) = \frac{1}{2}(\sum e_j^2(n)), e_j(n) = d_j(n) - y_j(n)$$

合計平均二乗誤差は、

$$E_{Av} = \frac{l}{N}(\sum_{n=1}^n \varepsilon(n))$$

与えられた入力 x_i に対する加重和 a_j 、および重み w_{ij} は、

$$a_j = \sum_{i=1}^n x_i w_{ij}$$

2.1.1 最急降下法 (BP)

6/14

シグモイド活性化関数を使用して，ニューロン j での出力は，

$$y_j = f(a_j) = \frac{1}{1 + e^{-a_j}}$$

重みは次の式に従って更新

$$w_{ij}(t) = w_{ij}(t-l) + \Delta w_{ij}(t)$$

$$\Delta w_{ij}(t) = \eta \frac{\partial E}{\partial w_{ij}}$$

2.1.2 運動量アルゴリズムを使用した最急降下法 (BPM)

7/14

バックプロパゲーションによるネットワークの収束は、多くの反復が必要となる。この問題を軽減するために運動量と呼ばれるパラメータを追加する。

$$w_{ij}(t) = w_{ij}(t-l) + \Delta w_{ij}(t) + \alpha \Delta w_{ij}(t-l)$$

ここで、 α は運動量項である。

ブロックチェーン

2. ANN

3. 予測

4. 実験

5. 結論

2.1.3 準ニュートン BFGS (BFG)

8/14

ニュートン法では更新ステップは次のように調整される.

$$W(t+l) = w(t) - H_t^{-l} g_t$$

ここで, H_t は重みとバイアスの現在の値でのパフォーマンスインデックスのヘッセ行列 (2 次導関数) である.

ニュートン法は, 共役勾配法よりも早く収束することがよくある. しかし広範囲で計算されるため, 計算コストが非常に高くなる.

2.1.4 Levenberg-Marquardt (LM)

9/14

ヘッセ行列を計算せずに2次のトレーニング速度に近づくように設計されている。誤差関数が二乗和であるという仮定で、ヘッセ行列を次のように近似することができる。

$$H = J^T J$$

そして、勾配は次のように計算できる。

$$g = J^T e$$

ここで、 J は重みとバイアスに関するネットワークエラーの一次関数を含む Jacobian 行列である。 e はネットワークエラーのベクトルである。

ヘッセ行列より計算コストが低くなる。

更新を次のように調整できる。

$$w(t+l) = w(t) - [J^T J + \eta I]^{-1} J^T e$$

2.1.5 共役勾配法 (CGD)

10/14

共役勾配アルゴリズムでは、ステップサイズは各反復で調整される.

$$P_0 = -g_0$$

重みベクトルは次のように更新.

$$w(t+l) = w(t) + \eta_t P_t$$

ここで,

$$p_t = -g_t + \beta_t p_{t-1}$$

$$\beta_t = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}$$

定数 β_t は, 今の勾配の 2 乗と, 前の勾配の 2 乗の比率.

3. 予測

11/14

予測するための分析手法では，ニューラルネットワークを使用．

利点

- 追加情報を追加する必要がない．
- 測定データから依存関係を自動的に学習．
- NN は一般化することができ，ノイズに耐性がある．

ブロックチェーン

2. ANN

3. 予測

4. 実験

5. 結論

4.1 データベース

12/14

レースに参加する馬ごとに NN が使用されており，出力は終了時間を予測．入力データを次のように正規化．

$$x_i = \frac{x_i - \text{mean}(x_i)}{\text{var}(x_i)}$$

入力データには，馬の体重，レースの種類，馬のトレーナー，騎手，レース中の馬の数，距離，トラックの状態，天気 の 8 つ．
この論文では，フィードバックつき多層 NN．

ブロックチェーン

2. ANN

3. 予測

4. 実験

5. 結論

4.3 トレーニングとテスト

13/14

2章で説明したトレーニングアルゴリズムを適用し、比較する。

Table 1: Race Showing Some Prediction (Race 7, January 18)

Show placement	Actual	BP	BPM	BFG	LM	CGD
1 st	Out of Respect	Out of Respect	Dahlgren chapel	Out of Respect	Out of Respect	Out of Respect
2 nd	Dahlgren chapel	Dahlgren chapel	Jet to Classics	Dahlgren chapel	Persian Ruler	Persian Ruler
3 rd	Persian Ruler	Jet to Classics	Out of Respect	Jet to Classics	Jet to Classics	Jet to Classics
4 th	Jet to Classics	Persian Ruler	Persian Ruler	Persian Ruler	Dahlgren chapel	Dahlgren chapel

Table 2: Experimental Results

	BP	BPM	BFG	LM	CGD
First Position in actual race results predicted correctly	39	39	35	29	32
Last Position in actual race results predicted correctly	30	29	27	22	37
1 horse in actual results predicted correctly	43	41	47	44	33
2 and more than 2 horses in actual results predicted correctly	33	23	25	23	37
No horse in actual results predicted correctly	24	31	28	33	30

BP アルゴリズムが 77%の精度で予測生成。

NN を構築し，学習アルゴリズムの分析からなる競馬予測を適用した．5つの教師あり NN を調査した．

結果は，すべてのアルゴリズムで平均 77%の精度で予測を生成．BP アルゴリズムは他と比べて精度が優れていたが，長いトレーニング時間が必要であった．LM アルゴリズムが最速であった．