

# バックプロパゲーションにおける 6F-5 高速アルゴリズムの研究 — 追記学習 —

益岡 竜介 渡部 信雄 木本 隆 川村 旭 浅川 和雄  
(株)富士通研究所

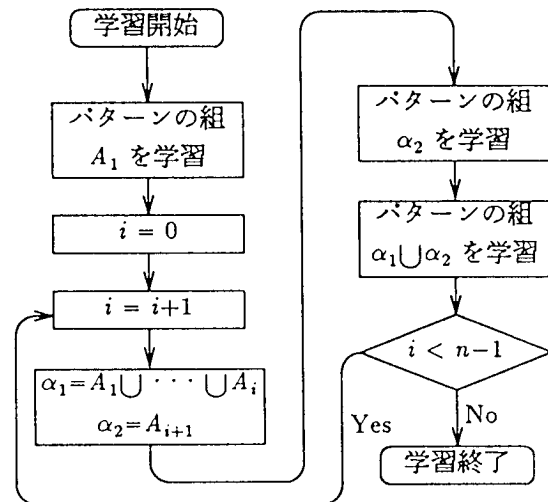
## 1. はじめに

各種の問題に対するバックプロパゲーションの有効性が示されつつあるが、実用上で問題になるのは、学習の際に、一度に全てのデータを揃える必要があること、その計算量の多さなどである。そこで、学習データを幾つかの組に分けその組ごとに学習と復習を繰り返すというアルゴリズムを開発した。このアルゴリズムは、学習データが新たに増えるという状況にも対応でき、計算量を減らすことができた。この学習アルゴリズムを追記学習と名付けた。

## 2. 追記学習の概要

学習データを幾つかの組に分ける。例えば、大量のデータを小さなデータの組に分ける。あるいは今までの学習データに新たな学習データが増えた場合に、今までの学習データと新たな学習データをそれぞれを1つずつの組にしたりする。そのようにして、得られたデータの組を  $\{A_1, A_2, \dots, A_n\}$  とする。このとき第1図が追記学習のアルゴリズムとなる。

- (d)  $\alpha, \beta$  のパターンを合わせて学習させる。
- (e)  $\gamma$  のパターンを学習させる。
- (f)  $\alpha, \beta, \gamma$  のパターンを合わせて学習させる。
- (g)  $\delta$  のパターンを学習させる。
- (h)  $\alpha, \beta, \gamma, \delta$  のパターンを合わせて学習させる。



第1図 追記学習のアルゴリズム

## 3. 実験の詳細

以下に追記学習の方法を具体的に実験を用いて説明する。実験はニューラル・ネットワークにバックプロパゲーション法を用いて、8ドット×8ドットのアلفベット26文字を認識させるものである。具体的には以下のようにアルファベットを学習させていく。

- (a) A から Z までのパターンを4つのパターンの組に分ける。

$$\alpha = \{A, B, C, D, E, F, G\}$$

$$\beta = \{H, I, J, K, L, M\}$$

$$\gamma = \{N, O, P, Q, R, S, T\}$$

$$\delta = \{U, V, W, X, Y, Z\}$$

- (b)  $\alpha$  のパターンを学習させる。
- (c)  $\beta$  のパターンを学習させる。

## 4. 実験の設定

同じ状態のネットワークからそれぞれ通常の学習と追記学習でアルファベットの認識を学習させて、学習の速さを比較した。

### 4.1. ネットワークの構造

使ったネットワークの構造は以下のようである。

- 入力層 64 ユニット
- 中間層 15 ユニット
- 出力層 26 ユニット

閾値は、中間層と出力層に入れた。

### 4.2. 重みと閾値の初期値と学習

重みと閾値の初期値は、 $[-0.1, 0.1]$  からランダムに取った。ネットワークの学習は、学習させるパターンを変える以外は通常のバックプロパゲーションを用いた。閾値も重みと同様に学習させた。

#### 4.3. データの構造

入力信号は各アルファベットについて  $8 \times 8$  ドットのフォントのデータを 0, 1 の 2 値としたものを与えた。教師信号は各アルファベットにつきそれぞれ異なる一つのユニットだけが 1 でその他は全て 0 としたものを与えた。データの数は A から Z までの 26 個である。

| 入力信号            | 教師信号                                |
|-----------------|-------------------------------------|
| 0 0 0 1 1 0 0 0 | 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
| 0 0 1 0 0 1 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
| 0 1 0 0 0 0 1 0 |                                     |
| 0 1 1 1 1 1 1 0 |                                     |
| 0 1 0 0 0 0 1 0 |                                     |
| 0 1 0 0 0 0 1 0 |                                     |
| 0 1 0 0 0 0 1 0 |                                     |
| 0 1 0 0 0 0 1 0 |                                     |
| 0 0 0 0 0 0 0 0 |                                     |

第 2 図 入力信号と教師信号の例 (A)

#### 4.4. 学習定数

学習定数は以下のように取った。

##### (1) 学習レート

A から Z の 26 文字全てを学習するとき 0.2 とした。後は学習させるパターン数に反比例してとった。すなわち,  $p$  をパターン数としたときに学習レート  $\epsilon$  を  $\epsilon = 5.2/p$  とした。

##### (2) モーメンタム

一律にモーメンタム  $\alpha$  を  $\alpha = 0.4$  とした。

#### 4.5. 学習の収束判定

学習の収束判定は, 教師信号からの差が各ユニットで 0.4 未満になったときとした。

#### 5. 実験結果

この追記学習では, 学習一回ごとに学習させるパターン数が異なるために, 単純に学習回数を比較したのでは, 計算に必要なリソースを測る事が出来ない。ここでは, 通常の学習との比較のために, 二つの量による比較の表を用意した。

##### [A] 総計算量

浮動少数点の足し算と掛け算の回数としての計算量は, シグモイド関数の計算に 30 回要するとして, 一回の学習に関して

$$7199p+5460$$

回要する。ここで  $p$  は学習させるパターン数である。総計算量とは, 一回ごとの計算量を学習が完了するまで, 足しあわせたものをものとする。

##### (B) パターン提示回数

計算に必要なリソースを測るもっと簡単な指標として, パターン提示回数を考えた。これは一回ごとの学習させるパターン数を学習が完了するまで, 足しあわせたものである。

表の中で seed とあるのは乱数の発生用の seed である。この値でネットワークの最初の重みが定まる。

| seed  | [A]      |          | [B]  |      |
|-------|----------|----------|------|------|
|       | 通常       | 追記       | 通常   | 追記   |
| 0     | 65020150 | 38952890 | 8710 | 5026 |
| 10000 | 68125590 | 47059531 | 9126 | 5989 |
| 20000 | 65214240 | 47608185 | 8736 | 6179 |
| 30000 | 70454670 | 41415003 | 9438 | 5341 |
| 40000 | 69678310 | 37243844 | 9334 | 4788 |
| 50000 | 70260580 | 36918230 | 9412 | 4750 |
| 60000 | 72783750 | 36473876 | 9750 | 4668 |
| 70000 | 67737410 | 34772793 | 9074 | 4471 |
| 80000 | 63661520 | 37970987 | 8528 | 4909 |
| 90000 | 62496980 | 33950915 | 8372 | 4373 |

第 3 図 追記学習における計算量の表

#### 6. おわりに

この場合約 2 倍の高速化がなされた。いろいろなデータにより, 結果は変動するが追記学習の方が計算量を必要としたという場合はほとんどなかった。

データの追加にいつでも対応できながら, なおかつ最終的な計算量が最初からやり直した場合より大きく減っているというのは大きな利点である。また大量の学習データに対してもデータを幾つかの組に分割して学習させることができ, 学習が容易になる。

この追記学習のアルゴリズムは仮想インピーダンス制御法<sup>1</sup> などの他のアルゴリズムとも組み合わせることができ, 更なる高速化を期待できる。またアナログのデータに対しても実験しており, 約 5 倍の高速化という結果を得ている。

#### 謝辞

日頃から御指導頂く棚橋純一部門長, 林弘部長ならびに人工知能第二研究室の皆様へ感謝します。

#### References

- 長田茂美, 関口実, 吉沢英樹, 渡部信雄, 木本隆, 浅川和雄, “ニューロコンピュータによるロボットの行動制御,” 情報処理学会基礎研究会, vol. 11-2, pp. 1-8, 1988/11/21.