

情報工学実験Ⅱレポート（探索アルゴリズム2）

曜日 & グループ番号: 月 & グループ 0

2011 年 12 月 23 日

概要

このレポート（ファイル）は、「情報工学実験Ⅱ・探索アルゴリズムその 2[1]」の実験レポートの骨組みを例示している。あくまでも例示であって、全てをこの通りに従う必要はないが、指示された項目を含めた上で、報告書として他者が読みやすいレポートとなるように工夫する事。

グループメンバ

（補足：レベル毎に 全員が協力して実施 した上で、レベル毎にレポートをまとめる担当者を決め、全体を一つのレポートとして整理すること。）

- 945734J 當間愛晃: 担当 Level1.1, 1.2, 3.4
- 945700 hoge: 担当 Level2.1, 2.2, 3.5
- 945700 hoge: 担当 Level3.1, 3.3
- 945700 hoge: 担当 Level3.2

提出したレポート一式について

レポート一式は「naha:/home/home/teacher/tnal/jikken1-fri/e945734/」にアップロードした。提出したファイルのディレクトリ構成は以下の通りである。

（補足：必ず下記のように整理しろという指定ではない。自分たちでやりやすいように Level 毎に整理しても構わない）

```
./src/      # 作成したプログラム一式
./report/   # レポート関係ファイル・図ファイルを含む。
```

1 Level1: 線形分離可能な OR 問題への適用

1.1 課題説明

2 入力 1 出力で構成される単純パーセプトロン (ニューラルネットワーク) を用いて、4 つの教師信号を用意した OR 問題へ適用し、重みが適切に学習可能であることを確認する。また、学習が収束する様子をグラフとして示す。

1.2 OR 問題を学習させた際の誤差収束度合いについて

1.2.1 実験結果

NN では重みを更新する毎に誤差が減るように学習を行うが、その学習の様子は初期の重みをどのように設定したか、学習に用いたパラメータをどのように設定したか、といった対象問題以外の要素に影響して学習の様子が変化する。シード値を変えた際の学習収束回数を表 1 に示す。シード値を 10 回変更して学習させた際の重みを更新する様子を図 1 に、その平均をプロットした平均推移値を図 2 に示す。なお、平均値を求める際には * * して * * した。具体的には云々。

表 1: OR 問題の学習に要した回数

シード値	収束した回数
100	hoge
200	hoge
300	hoge
400	hoge
500	hoge
600	hoge
700	hoge
800	hoge
900	hoge
1000	hoge
10 試行の平均値	hoge

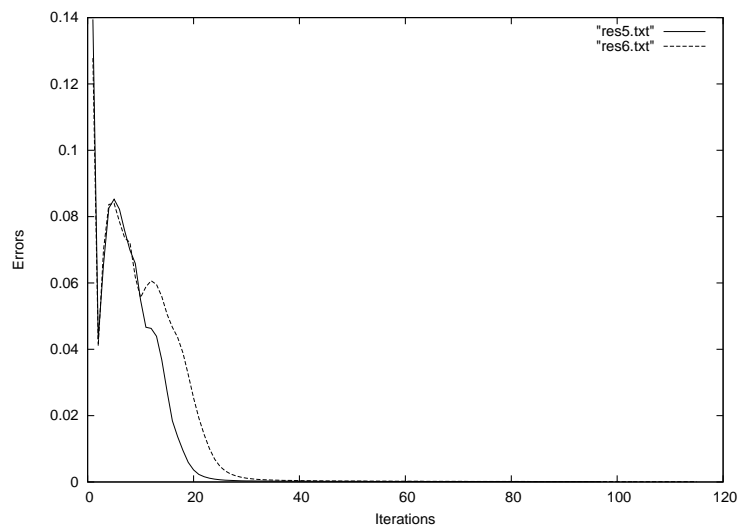


図 1: 重みを更新する様子

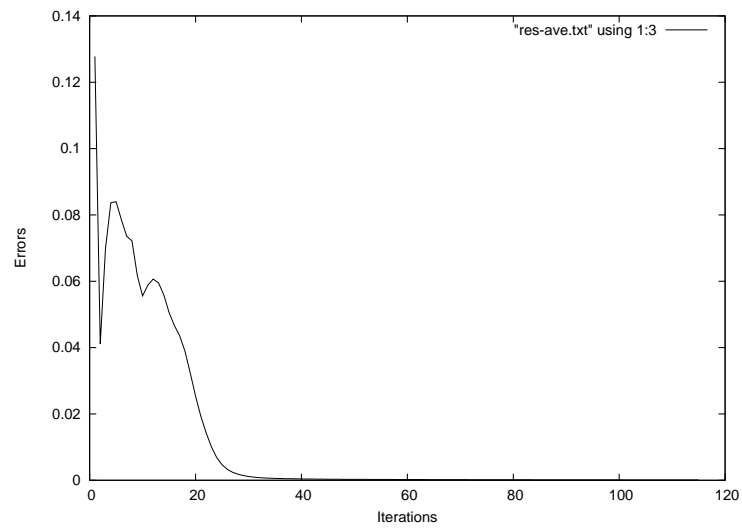


図 2: 重みを更新する様子 (平均値)

1.2.2 考察

(結果から分かったことは？)

2 Level2: 線形分離不可能な ExOR 問題への適用

2.1 課題説明

階層型ニューラルネットワークを ExOR 問題へ適用し、線形分離できない問題においても学習可能であることを確認する。特に Level2 では、この問題を解決するために中間層を導入することで拡張した階層型ニューラルネットワークにより学習可能であることを確認する。

2.2 階層型 NN による学習

2.2.1 最適なパラメータを探すためのアプローチ

ETA, ALPHA, HIDDEN を指定された範囲内でランダムな値に設定し、プログラムを実行した。パラメータの値の決定や、iteration 数の平均値を求める処理は次のシェルスクリプト (ソースコード 1) で行った。ExOR 問題は線形分離不可能であるため、HIDDEN の最小値を 2 としている。最大値は特に指定されていないが、HIDDEN を極端に大きくしすぎると各パラメータの最適な組み合わせを見つけ出すのが困難になると考えたため、最大値を 16 とした。

ソースコード 1: 本 level で使用したシェルスクリプト

```
1 #!/bin/sh
2 #シェルスクリプトの要件
3 # 1.ETA,ALPHA,HIDDENの値として範囲内でランダムな組み合わせを選択肢
4 # 2.それを元にseed値を変えて10パターンの結果を得る。
5 # 3.FINISHのiterationの回数を抽出し、そのときのパラメータの値を一緒に表示
6 # 4.10パターンのiterationの平均をとる。
7 #
8 #bp_mo_exor.cの変更
9 # 文字列処理を行うためにfprintf(stderrを以下のように変更した。
10 # fprintf(stdout,"FINISH 2: iteration = %4d, error = %.10f\n",ite,err);
11 #
12 export LANG=C
13
14 ((eta_inte=${RANDOM}%2))
15 #eta_deci='echo "scale=2; ($RANDOM*3) / 100000" | bc'
16 ((eta_deci=${RANDOM}%100))
17 ETA='expr ${eta_inte}.${eta_deci}'
18 sed -i -e "s:#define ETA.*:#define ETA $ETA:g" bp_mo_exor.c
19
20 #alpha_deci='echo "scale=2; ($RANDOM*3) / 100000" | bc'
21 ((alpha_deci=${RANDOM}%100))
22 ALPHA='expr 0.${alpha_deci}'
23 sed -i -e "s:#define ALPHA.*:#define ALPHA $ALPHA:g" bp_mo_exor.c
24
25 ((HIDDEN=${RANDOM}%15+2))
26 sed -i -e "s:#define HIDDEN.*:#define HIDDEN $HIDDEN:g" bp_mo_exor.c
27
28 sed -i -e "s/
29 //g" bp_mo_exor.c
30
31 gcc bp_mo_exor.c -o bp_mo_exor
32 echo "ETA:$ETA ALPHA:$ALPHA HIDDEN:$HIDDEN"
33 ./bp_mo_exor 1000 > fuga.txt 2> log.txt
34 ./bp_mo_exor 2000 >> fuga.txt 2>> log.txt
35 ./bp_mo_exor 3000 >> fuga.txt 2>> log.txt
36 ./bp_mo_exor 4000 >> fuga.txt 2>> log.txt
37 ./bp_mo_exor 5000 >> fuga.txt 2>> log.txt
38 ./bp_mo_exor 6000 >> fuga.txt 2>> log.txt
39 ./bp_mo_exor 7000 >> fuga.txt 2>> log.txt
40 ./bp_mo_exor 8000 >> fuga.txt 2>> log.txt
41 ./bp_mo_exor 9000 >> fuga.txt 2>> log.txt
42 ./bp_mo_exor 10000 >> fuga.txt 2>> log.txt
43
44 sum=0
45 cat log.txt | awk '{print $5}' | tr -d "," > figures.txt
46 exec < figures.txt
47 while read line
48 do
49     ((sum=${sum}+${line}))
50 done
51
52 ((avarage=${sum}/10))
53
54 cat log.txt
55 echo "the avarage is ${avarage}."
```

このシェルスクリプトを数十回実行し，iteration 数の平均値が比較的小さい実行結果を 10 個分記録した．その中で平均値が最小なものを選択し，グラフ化することにした．

2.2.2 実行結果

表 2 にシード値 10 パターンで試した際の収束に要した学習回数と，その最小の平均回数を示す．

シード値	収束した回数
1000	37
2000	45
3000	52
4000	44
5000	50
6000	53
7000	39
8000	29
9000	38
10000	39
10 試行の平均値	42

表 2: 階層型 NN による ExOR 問題の学習に要した回数

各パラメータが $ETA = 1.26$, $ALPHA = 0.94$, $HIDDEN = 16$ の時，表 2 のような結果が得られた．その時の学習曲線は図 3 のようになる．図 3 は gnuplot を用いて seed 値別にプロットしたグラフを元に smooth unique オプションで平均化を行い，得られたものである．

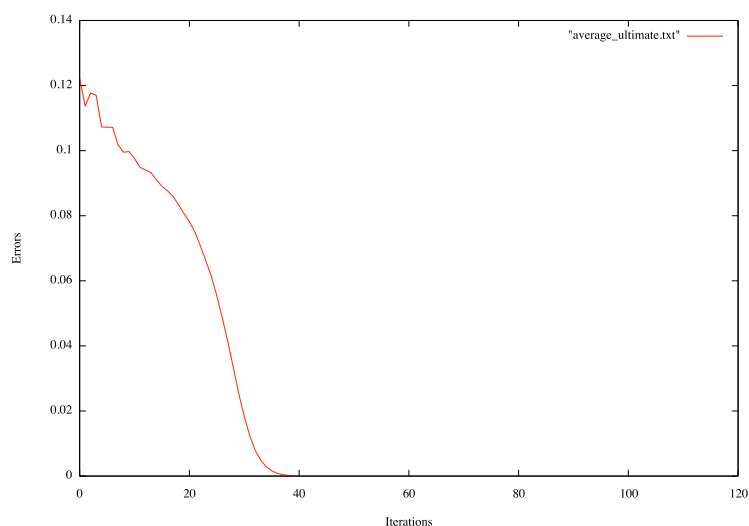


図 3: 重みを更新する様子（平均値）

2.2.3 考察

ここでは，各パラメータと iteration 数の関係について考察する．

下記に平均 iteration 数が比較的小さい実行結果とそのときのパラメータの値を示す．

— 平均 iteration 数が小さい実行結果 —

```
/Users/e135761/search/info2/nn/bp_mo% cat suitable.txt | grep -e 'ETA' -e 'the'
ETA:1.89 ALPHA:0.82 HIDDEN:14
the avarage is 102.
ETA:1.79 ALPHA:0.80 HIDDEN:10
the avarage is 120.
ETA:1.20 ALPHA:0.90 HIDDEN:5
the avarage is 115.
ETA:1.22 ALPHA:0.86 HIDDEN:14
the avarage is 114.
ETA:0.85 ALPHA:0.94 HIDDEN:15
the avarage is 55
ETA:1.80 ALPHA:0.83 HIDDEN:10
the avarage is 102.
ETA:1.74 ALPHA:0.77 HIDDEN:14
the avarage is 137.
ETA:1.26 ALPHA:0.94 HIDDEN:16
the avarage is 42.
ETA:1.60 ALPHA:0.93 HIDDEN:10
the avarage is 46.
ETA:0.84 ALPHA:0.93 HIDDEN:16
the avarage is 70.
ETA:0.69 ALPHA:0.94 HIDDEN:13
the avarage is 78.
```

10 回の実行結果における共通点は、いずれも ALPHA 値が 1 に近い値をとるという点だった。
続いて、平均 iteration 数が大きい実行結果を示す。

— 平均 iteration 数が大きい実行結果 —

```
/Users/e135761/search/info2/nn/bp_mo% cat unsuitable.txt | grep -e 'ETA' -e 'the'
ETA:0.14 ALPHA:0.29 HIDDEN:9
the avarage is 5344.
ETA:0.38 ALPHA:0.25 HIDDEN:2
the avarage is 12870.
ETA:0.37 ALPHA:0.94 HIDDEN:2
the avarage is 20202.
ETA:0.0 ALPHA:0.78 HIDDEN:9
the avarage is 100000.
ETA:0.81 ALPHA:0.49 HIDDEN:2
the avarage is 10938.
ETA:0.16 ALPHA:0.22 HIDDEN:4
the avarage is 6317.
ETA:0.57 ALPHA:0.25 HIDDEN:2
the avarage is 11891.
ETA:1.70 ALPHA:0.27 HIDDEN:2
the avarage is 33521.
ETA:1.49 ALPHA:0.44 HIDDEN:2
the avarage is 30505.
ETA:1.68 ALPHA:0.72 HIDDEN:2
the avarage is 40170.
```

平均 iteration 数が小さいときのパラメータと平均 iteration 数が大きいときのそれとを比較すると、HIDDEN の値が小さいと平均 iteration 数が大きくなり、反対に値が大きいと平均試行回数が小さくなるという結果になった。また *ETA* に関しては、0 を設定しない限り iteration 数に大きな影響を与えないことが伺える。これら実行結果より、最も効率良く学習が収束するパラメータの組み合わせは、表 3 のようになる。

<i>ETA</i>	<i>ALPHA</i>	<i>HIDDEN</i>
0 以外の数値	1 に近い値	できるだけ大きい数値

表 3: 最適なパラメータの組み合わせ

3 Level3: 応用事例：文字認識問題への適用

3.1 課題説明

階層型 NN を文字認識に適用し、考察する。特に、用意された教師データと認識のしやすさに関する関係性や、学習最適化のためのパラメータのチューニングおよび、より柔軟性の高い認識方法に関する検討を行う。

3.2 Level3.1: パラメータのチューニング

3.2.1 最適なパラメータを探すためのアプローチ

指定された条件下において学習が効率良く行われるパラメータの組み合わせを探すため、**して**することでパラメータを調整した。

（補足：全パターンを調べても良いし、いくつかのパターンを調べても良いが、どのような方法で調整したら良いかを考えよう。その上で、自分たちがどのように取り組んだのか（＝アプローチ）を説明しよう。）

3.2.2 実行結果

（補足：ボーナスポイントの確認がありますので、シード値 10 パターンで試した際の収束に要した学習回数と、その平均回数が分かるように明示してください。）

表 4: 階層型 NN による文字認識問題の学習に要した回数

シード値	収束した回数
100	hoge
200	hoge
300	hoge
400	hoge
500	hoge
600	hoge
700	hoge
800	hoge
900	hoge
1000	hoge
10 試行の平均値	hoge

3.2.3 考察

3.3 Level3.2: パラメータと収束能力の関連性について

3.3.1 関係性を確認するためのアプローチ

3 つのパラメータがどのような関係にあるかを検証するため、ケース 1,2,, を設定し、学習曲線からその関係性について考察する。

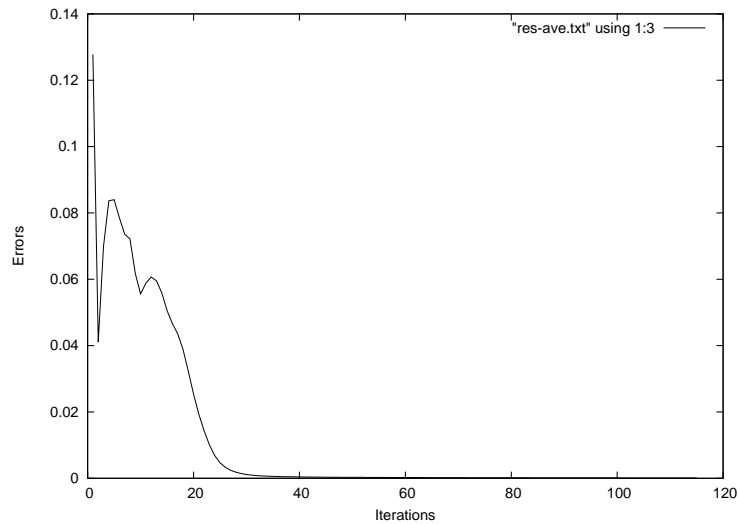


図 4: 重みを更新する様子 (平均値)

3.3.2 結果

3.3.3 考察

3.4 Level3.3: 任意の評価用データを用いた評価

3.4.1 アプローチ

(仮説 1)

学習時のデータ (教師データ) との違いが少ない程認識率が高く、逆に教師データとの違いが多い程認識率が低くなる
との仮定の下、如何に示す評価データを用意した。

(仮説 2)

学習時のデータ (教師データ) との違いが多い程認識率が高く、逆に教師データとの違いが少ない程認識率が低くなる
との仮定の下、如何に示す評価データを用意した。

(補足: それ以外でも構いませんが、検証したい内容に応じた仮説を立て、その仮説を検証するためのデータを用意
しましょう)

3.4.2 結果

3.4.3 考察

3.5 Level3.4: 認識率を高める工夫

3.5.1 対象とする問題点

3.5.2 改善方法の提案

3.5.3 考察

4 その他: 実験の内容・進め方に関するコメント等

(補足 : 今後の為に参考にしたいので、情報工学実験 2・探索アルゴリズム 1,2 で扱った内容、実験の進め方等について意見があれば書いてください (当然、どのような意見であってもレポートの評価を下げる事はしません。)。「授業評価アンケート」の際に書いてもらっても構いません。)

(補足 : 参考文献は thebibliography 環境を使って列挙し、本文中で適切な箇所で引用するようにしましょう。例えば下記文献は、アブストラクト中で引用しています)

参考文献

[1] 情報工学実験 2: 探索アルゴリズムその 2 (當間)

<http://www.eva.ie.u-ryukyu.ac.jp/~tnal/2011/info2/search2/>