

情報工学実験Ⅱレポート（探索アルゴリズム2）

金曜日グループ7

平成 27 年 1 月 10 日

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。）

- 135712D 山城義弘: 担当 Level1.1, 1.2
- 135706K 喜友名 朝稔: 担当 Level2.1, 2.2
- 135719B 澤崎夏希: 担当 Level2.3, 3.1
- 135740K 大城朝貴: 担当 Level4.1, 4.2
- 945734J 當間愛晃: 担当 Level1.1, 1.2, 3.4

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）

```
./scr/      # 作成したスクリプト一式
./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 Level1.1: OR 問題が解けることの確認

1.1.1 学習が収束する回数

指定されたシード値を用いた際の、学習が終了した回数を表 1 に示す。

表 1: OR 問題の学習に要した回数

シード値	収束した回数
1000	97
2000	91
3000	112
4000	109
5000	94
6000	100
7000	101
8000	115
9000	114
10000	95

また、横軸を学習回数、縦軸を誤差としたときの学習結果を 1 と 2 に示す。

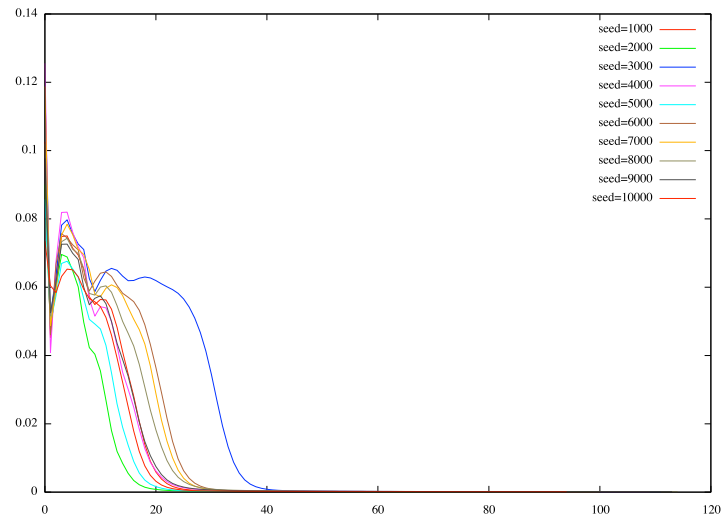


図 1: シード値を変更して得られた 10 回分の学習結果

1.1.2 考察

1 より、シード値が変わっても、収束するまでの学習回数はおおよそ 100 回ぐらいになる。全体的に、学習回数が増加していくと誤差は収束していくが、序盤は誤差の変化が大きい。

1 と 2 より、シード値の増減で大幅なグラフの変更はなかった。

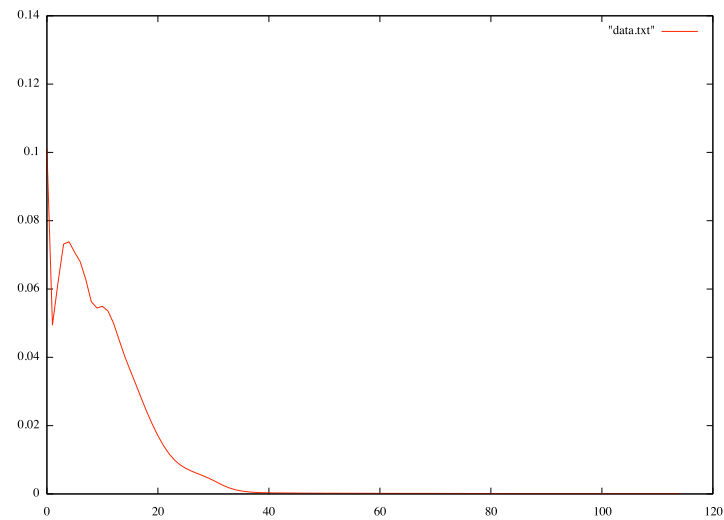


図 2: 10 回分の学習結果を平均した結果

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

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、最初は 1 つの値づつ、大まかな値から設定して調べ、徐々に値の細かくしていくことでパラメータを調整した。

2.2.2 実行結果

2.2.3 考察

alpha や ETA は高い値の方が誤差が低い結果となったが、HIDDEN に関しては必ずしも荘ではなく、16 という比較的低い数字で誤差が小さくなったので、中間層のユニットは多ければ多い程良いというわけではないと考えられる。

表 2: 階層型 NN による ExOR 問題の学習に要した回数

シード値	収束した回数
1000	15622
2000	110
3000	100000
4000	100000
5000	100000
6000	17180
7000	27237
8000	4394
9000	6665
10000	100000
10 試行の平均値	47130.8

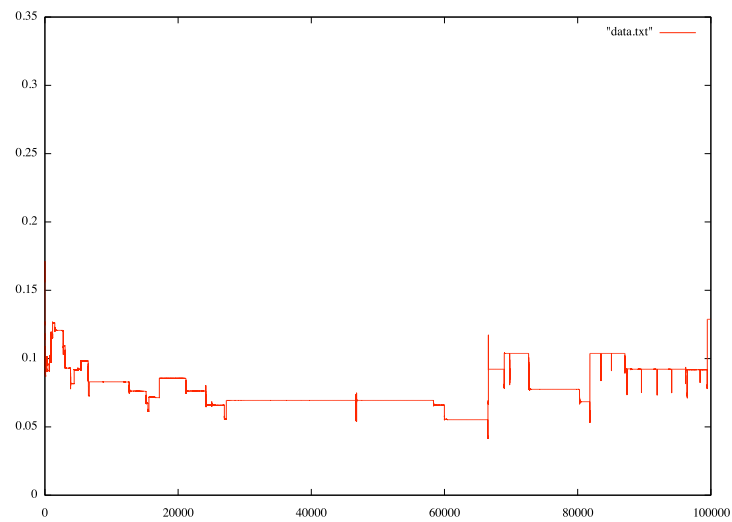


図 3: 重みを更新する様子 (平均値)

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

最適なパラメータを探すために、3つのパラメータをそれぞれを探索するスクリプトを作成した。

- HIDDEN.sh :HIDDEN の値を 1 ずつ書き換えながら平均値を出力
- ALPHA.sh :引数に「ALPHA の初期値」「ALPHA の刻み値」「ALPHA の最大値」を取り、連続的に実行する。
- ETA.sh :ALPHA.sh と同様に ETA を小刻みに変更しながら実行を行う。

3.2.2 実行結果

- H:31
- E:1.4
- A:0.5727

表 3: 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
1000	83
2000	75
3000	68
4000	139
5000	46
6000	67
7000	96
8000	122
9000	113
10000	162
10 試行の平均値	971

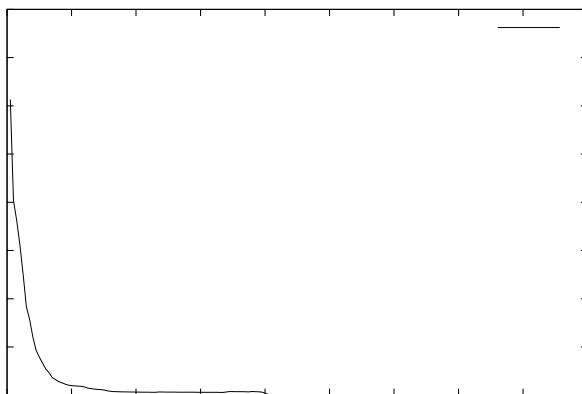


図 4: 重みを更新する様子 (平均値)

3.2.3 考察

今回の探索には「SEARCH.sh」を実行して探索したものである。このスクリプトは各パラメータを逐次変更しながら数値が減少する方向に移動し数値の増加幅がある程度を超えると次の段階へ数値を進めるものである。

3.3 Level3.2: パラメータと収束能力の関連性について

学習と各パラメータの影響について、それぞれのパラメータを変更しながらその平均値をグラフにして観察した。

3.4 結果:HIDDEN

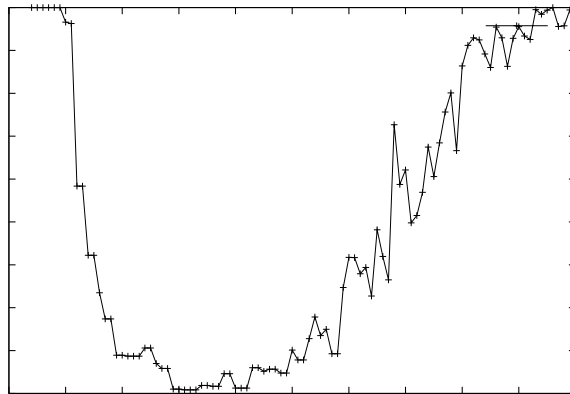


図 5: HIDDEN の増加に伴う推移

3.5 結果:ETA

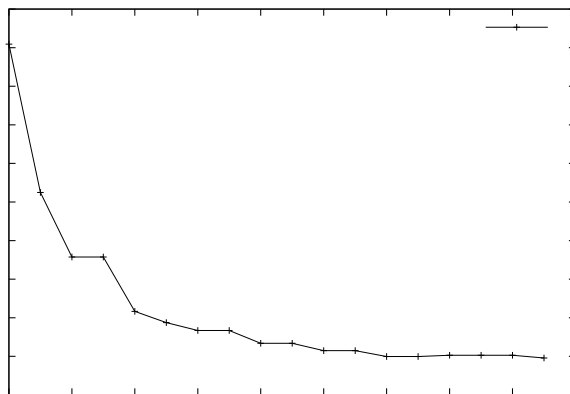


図 6: ETA の増加に伴う推移

3.6 結果:ALPHA

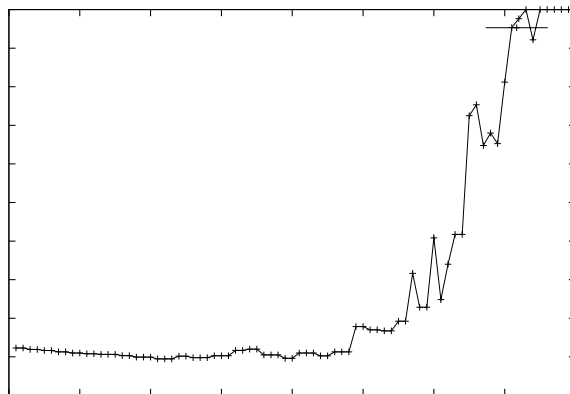


図 7: ALPHA の増加に伴う推移

3.6.1 考察

出力結果を見ると、HIDDEN の数値が結果に大きく影響することがわかる。HIDDEN が低い場合、少し数値を増やすだけで値が大幅に更新されるが、30 を超えた辺りから増加に転じた。このパラメータは増加させることで非常に学習に影響をおよぼすが、性能の向上には限界がある。しかし、値の刻み幅も 1 だけなのでまずこのパラメータを調整することにした。次に ETA は数値を増やしていくことで次第に最小値へ近づくように変化するパラメータであることがわかった。値の影響は大きいですが、刻み幅を狭くしてもそれほど結果に表れなかった。一方で ALPHA は増やせば増やすほど収束までの時間がかかった。これは α が学習の際にどれくらい値を増やすか、という数値になっているため、精度は増すが学習に時間がかかるという結果になったことがわかる。

3.7 Level3.3: 任意の評価用データを用いた評価

3.7.1 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、明らかに見た目が違う評価データは用意せず教師データを拡大、位置移動、縮小のような認識しやすい評価データと、認識しにくい斜めにした評価データを用意した。

3.7.2 結果

ソースコード 1: 学習時のデータ拡大

```
1 nn> e
2 filename? —> data.num2/eval-4.txt
3 correct = 0100000000
4 000111110000
5 000001000000
```

```

6 000001000000
7 000001000000
8 000001000000
9 000001000000
10 000001000000
11 000001000000
12 000001000000
13 000001000000
14 000001000000
15 000111110000
16 CHECK filename data.num2/eval-4.txt
17 EVA o[0] = 0.03804, correct[0] = 0.1
18 EVA o[1] = 0.64843, correct[1] = 0.9
19 EVA o[2] = 0.18650, correct[2] = 0.1
20 EVA o[3] = 0.03347, correct[3] = 0.1
21 EVA o[4] = 0.16344, correct[4] = 0.1
22 EVA o[5] = 0.26652, correct[5] = 0.1
23 EVA o[6] = 0.06497, correct[6] = 0.1
24 EVA o[7] = 0.07104, correct[7] = 0.1
25 EVA o[8] = 0.07718, correct[8] = 0.1
26 EVA o[9] = 0.02484, correct[9] = 0.1
27 EVA sum_error = 0.85847

```

ソースコード 2: 学習時のデータ位置移動

```

1 nn> e
2 filename? —> data.num2/eval-5.txt
3 correct = 01000000000
4 0000000000000
5 0000000001110
6 0000000000100
7 0000000000100
8 0000000000100
9 0000000000100
10 0000000000100
11 0000000000100
12 0000000000100
13 0000000000100
14 0000000001110
15 0000000000000
16 CHECK filename data.num2/eval-5.txt
17 EVA o[0] = 0.06909, correct[0] = 0.1
18 EVA o[1] = 0.13674, correct[1] = 0.9
19 EVA o[2] = 0.08040, correct[2] = 0.1
20 EVA o[3] = 0.44300, correct[3] = 0.1
21 EVA o[4] = 0.08931, correct[4] = 0.1
22 EVA o[5] = 0.30921, correct[5] = 0.1
23 EVA o[6] = 0.05398, correct[6] = 0.1

```



```

24 EVA o[7] = 0.04762, correct[7] = 0.1
25 EVA o[8] = 0.12550, correct[8] = 0.1
26 EVA o[9] = 0.24100, correct[9] = 0.1
27 EVA sum_error = 1.64158

```

ソースコード 3: 学習時のデータ斜め

```

1 filename? —> data.num2/eval-6.txt
2 correct = 0100000000
3 00000000000000
4 000000100000
5 0000000001000
6 000000100010
7 000001000000
8 010010000000
9 000100000000
10 000010000000
11 000000000000
12 000000000000
13 000000000000
14 000000000000
15 CHECK filename data.num2/eval-6.txt
16 EVA o[0] = 0.13168, correct[0] = 0.1
17 EVA o[1] = 0.22826, correct[1] = 0.9
18 EVA o[2] = 0.35467, correct[2] = 0.1
19 EVA o[3] = 0.14085, correct[3] = 0.1
20 EVA o[4] = 0.06984, correct[4] = 0.1
21 EVA o[5] = 0.17521, correct[5] = 0.1
22 EVA o[6] = 0.00947, correct[6] = 0.1
23 EVA o[7] = 0.34620, correct[7] = 0.1
24 EVA o[8] = 0.21842, correct[8] = 0.1
25 EVA o[9] = 0.15081, correct[9] = 0.1
26 EVA sum_error = 1.61026

```

ソースコード 4: 学習時のデータ縮小

```

1 nn> e
2 filename? —> data.num2/eval-7.txt
3 correct = 0100000000
4 00000000000000
5 00000000000000
6 000011100000
7 000001000000
8 000001000000
9 000001000000
10 000001000000
11 000001000000
12 000001000000

```

```
13 000011100000
14 000000000000
15 000000000000
16 CHECK filename data.num2/eval-7.txt
17 EVA o[0] = 0.08543, correct[0] = 0.1
18 EVA o[1] = 0.73866, correct[1] = 0.9
19 EVA o[2] = 0.08469, correct[2] = 0.1
20 EVA o[3] = 0.04289, correct[3] = 0.1
21 EVA o[4] = 0.14736, correct[4] = 0.1
22 EVA o[5] = 0.14030, correct[5] = 0.1
23 EVA o[6] = 0.03224, correct[6] = 0.1
24 EVA o[7] = 0.19359, correct[7] = 0.1
25 EVA o[8] = 0.07265, correct[8] = 0.1
26 EVA o[9] = 0.10494, correct[9] = 0.1
27 EVA sum_error = 0.52962
```

3.7.3 考察

Iを拡大, 縮小, 斜め, 位置移動をしたものを評価用データとして認識テストを行った結果より, 拡大, 縮小はほかのと比べエラーが低めだが, 斜めそして予想に反して学習時のデータを平行移動させただけのデータではエラーがとても高くIとしては到底認識できていないことが分かった. これよりこの文字の認識には学習時のデータの形はもちろんだがそれよりも位置が非常に重要になっていると予想できる.

3.8 Level3.4: 認識率を高める工夫

3.8.1 対象とする問題点

入力されたデータが想定していた入力と比べてサイズが異なったり, 位置がずれている等, 文字の一部が欠けている以外にも多様な要因によるデータ(情報)の劣化による認識率の低下.

3.8.2 改善方法の提案

- 学習時のデータをより多く様々な可能性を考え用意することで認識率を高める.
- 逆に認識したいデータとは全く違うデータを学習させ, 認識率が低い場合に認識したいデータと判断する方法.

3.8.3 考察

一つ目の提案では, 例えばIを認識したいとき, 斜め, 拡大, 縮小, 位置変更をしたIをそれぞれ複数学習させることにより, その学習時のデータの組み合わせにより様々な形のデータにも認識率を高めることができると考えた. 二つ目の提案は, 認識率が高いことにこだわらず逆に認識率が低いことに着眼点をおくことにより, 逆転の発想でデータの認識率を高めることができると考えた.

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/>