

# パラメータファイルの作成方法

## 1 はじめに

---

本文書は、BIPL-SDNN ライブラリにおける選択的不感化ニューラルネット（以下 SDNN）のパラメータ設定ファイルの記述方法について述べるものである。SDNN そのものの説明や、用語の説明はライブラリパッケージに含まれる他の文書を参照して欲しい。

パラメータファイルは、shift-jis で記述し、拡張子は.txt を使用する必要がある。

## 2 パラメータの基本

---

全てのパラメータは、「パラメータ名 = パラメータ内容」で定義される。パラメータ内容は、数値、文字列、数値ベクトル、文字列ベクトルのいずれかである。数値、文字列はそのまま、ベクトルは全体を[]で囲み、各要素間を,で区切る。また、同じ要素が連続する場合、:を用いて記述を省略できる。例えば[1,1,2,3,3,3]は、[1:2,2,3:3]と記述できる。同一のパラメータが存在しないよう、<>を用いたタグ付けを採用している。

## 3 各パラメータの詳細

---

### <SDNN> タグ

全てのパラメータは SDNN タグ内に記述する。

- input\_number パラメータ  
SDNN の入力次元数。  
2 以上の整数を指定する。

### <SD> タグ

選択的不感化操作に関するパラメータが記述される。

- method パラメータ  
選択的不感化における入力の組み合わせの指定方法。mutualとfileのいずれかを指定する。  
mutual：全ての入力の組み合わせで不感化を行う（相互不感化）。  
file：ファイルを通じて入力の組み合わせを指定する。ファイルの記述方法は後述（4章）。

- filename パラメータ  
method パラメータで file を指定した場合に必要な。  
不感化の組み合わせが書かれたファイル名。

#### <PC> タグ

パターンコーディング操作に関するパラメータが記述される

- n パラメータ  
パターンコーディング操作で入力を変換したパターンの素子数。  
2 以上の偶数を指定。計算量が許す限り大きい値とすることを推奨。
- type パラメータ  
コーディングパターンの作成方法の指定。全ての入力次元に対しての作成方法を、文字列ベクトルの形で指定する。入力次元数とベクトルの要素数が同じ必要がある。  
要素は,
  - NUMERICAL(入力分割数,RANDOM\_INVERSE(反転数))  
数値入力, [0,1]の範囲を分割し, それぞれに対応するパターンを作成する。  
パターン作成方法はランダム反転法を用いる。
  - NUMERICAL(入力分割数,INTERPOLATE(代表ベクトル数))  
数値入力, [0,1]の範囲を分割し, それぞれに対応するパターンを作成する。  
パターン作成方法は補間法を用いる。
  - SYMBOL(入力種類数,RANDOM\_INVERSE(反転数))  
シンボル入力, シンボル間に序列がある場合に使用する。  
パターン作成方法はランダム反転法を用いる。
  - SYMBOL(CORRELATION\_TREE(相関木ファイル名))  
シンボル入力, 相関木が書かれたファイルを元に入力パターンを作成する。相関木ファイルの作り方は後述（5章）。

のいずれかである必要がある。
- random\_seed パラメータ  
コーディングパターン作成時に用いる乱数の種。Unsigned int の数値ベクトルか,  
random\_device を指定。random\_device を指定した場合は, ハードウェアエントロピーソースの乱数を 10 個生成し, それを乱数種として利用する。

## <NN> タグ

出力ニューラルネットに関するパラメータが記述される.

- type パラメータ  
出力ニューラルネットの種類. PP か SP を指定する.  
PP : 並列パーセプトロン (関数近似用)  
SP : 単純パーセプトロン (パターン識別用)
- train\_method パラメータ  
学習の終了条件. 以下のいずれかを指定.  
rmse(a,b) 学習サンプルの平均二乗誤差平方根が a 以下になる || 学習を b 回繰り返す  
(PP 専用)  
for(n) 学習を n 回繰り返す  
+ を用いて複数の条件を組み合わせられる. 例えば rmse(0.01,100) + for(30)と指定した場合,  
平均二乗誤差平方根が 0.01 を下回った後 30 回追加学習を行う.
- random\_seed パラメータ  
出力ニューラルネット初期化時の乱数種. <PC> タグ内の random\_seed パラメータと同様.
- initial\_value\_range パラメータ  
出力ニューラルネット初期化時の重みの幅, [最小値,最大値]で指定. 最小値, 最大値共に int  
整数.  $\pm 10$  程度にとどめたほうが, 近似 識別精度が高いことが経験上知られている.

\* 出力ニューラルネットが PP の場合設定するパラメータ

- output\_range パラメータ  
出力範囲, [最小値,最大値]で指定する. 数値はどちらも double 型.
- output\_quantization\_step\_size パラメータ  
出力の刻み幅, double 型数値で指定.

\* 出力ニューラルネットが SP の場合設定するパラメータ

- category\_number パラメータ  
識別クラス数. 2 以上の整数を指定.
- multi\_class\_recognition\_method パラメータ  
多クラス認識手法. 1v1 か 1vR のいずれかを指定する.  
1v1 : one-vs-one method  
1vR : one-vs-rest method

#### <OPTION> タグ

その他, ライブラリの設定に関するパラメータが記述される.

- print\_progression パラメータ  
学習の進行状況を画面表示するか否か. Y か N で指定.

#### <MULTI\_THREAD> タグ

- use パラメータ  
OpenMP による並列演算を使用するか否か. Y か N で指定.
- thread\_number パラメータ  
何個の CPU スレッドを使用するか. 2 以上の整数で指定.

```
<SDNN>
{
    input_number = 2
    <SD>
    {
        method = file
        filename = testsd.txt
        <PC>
        {
            n = 200
            type = [NUMERICAL(100, RANDOM_INVERSE(3)):2]
            random_seed = random_device
        }
    }
    <NN>
    {
        type = PP
        train_method = for(300)
        random_seed = random_device
        initial_value_range = [-5, 5]
        output_range = [-0.2, 1.2]
        output_quantization_step_size = 0.005
    }
    <OPTION>
    {
        print_progression = Y
        <MULTI_THREAD>
        {
            use = Y
            thread_number = 6
        }
    }
}
```

2変数関数近似を行う場合のパラメータの例

## 4 不感化設定ファイルの作成方法

本ライブラリは選択的不感化を行う入力の組み合わせをファイル経由で設定できる。

ファイルは shift-jis で記述, .csv 拡張子を使う必要がある。

1 行に一つの組み合わせを, 「不感化される入力番号,不感化する入力番号」の形で記述する。

入力番号は, 0 を始点とする。例として, 3 入力相互不感化を行う場合は,

```
0, 1
0, 2
1, 0
1, 2
2, 0
2, 1
```

3入力相互不感化の不感化設定ファイルの例

となる。

## 5 相関木ファイルの作成方法

序列関係にないシンボルを入力とする場合, 相関木を用いて適切な入力同士の関係を定義することで, SDNN が入出力関係を容易に学習できるようになる。

他のファイル同様 shift-jis で記述し, .csv 拡張子を用いる必要がある

1 行には, 入力シンボルに対応するノードを親ノードから順番に,

「親ノード,相関係数,子ノード,,,相関係数,入力対応ノード」として記述する。

1 行上のノードと同じノード, 相関係数の場合は, 一部記述を省略できる。

相関木ファイルの例を以下に示す。

```
Color,      0,      warm,  0.7,   red
,           ,           ,  0.7,   orange
,           ,           ,  0.7,   pink
,           0,      cold,  0.5,   blue
,           0,      cold,  0.5,   light_blue
```

相関木ファイルの例

このとき, 葉ノードである red,orange,pink,blue,light\_blue が入力シンボルとして利用できる。これらは red から順に 0,1,2,,,と番号が振られており, SDNN の入力にはこちらの番号を使用する。