BIPL_SDNN　ライブラリの使い方

1 はじめに

本文書は，選択的不感化ニューラルネット（SDNN）のクラスを含む静的ライブラリ，BIPL_SDNN ライブラリの使い方を説明するものである．

本ライブラリは C++11，コンパイラには Visual Studio 2015 Professional を用いている．
gcc 等他のコンパイラでコンパイル可能かどうかは確認していない．また，ベータ版では VS 2015 以外のコンパイラでの対応，動作保障はしない．

2 導入方法

Bin フォルダ内の，BIPL.SDNN.lib と BIPL_SDNN.h ファイルがライブラリ本体です． 一般の静的ライブラリと同様に導入してください．
ベータ版で提供しているライブラリは，Release X64 でコンパイルしています．

X86 等でのライブラリが必要な場合や，自身の環境でライブラリのコンパイルがしたい場合は，source フォルダ内にあるプロジェクトファイル，ソースファイルを利用してください．プロジェクトの参照フォルダは，利用環境に合わせて変更してください．

© 2016 堀江 和正

3 SDNN による学習，テストの実行方法

bipl::sdnn::SDNN クラスを用いて，SDNN の機能を実行する．

本クラスを用いて，SDNN の学習，テストを実行する場合，以下の手順になる．

1. bipl::sdnn::SDNN クラスの実体化
2. InitSDNN 関数を用いて SDNN の初期化
3. 学習，テストサンプルの準備
4. Train 関数を用いてサンプルの学習
5. Estimate 関数を用いて未知のサンプルの推定，識別

関数の詳細は次章．

4 クラス関数詳細

4.1 INITSDNN 関数

```
void InitSDNN(const std::string &parameter_filename);
```

パラメータファイルの設定に従い，SDNN の初期化を行う．

パラメータファイルの作り方については別文書参照．

＜入力＞

parameter_filename：パラメータファイル名

＜出力＞

なし

＜エラー＞

誤ったパラメータ設定を行っていた場合，コンソールから正しいパラメータ入力を求める．

4.2 TRAINONESAMPLE 関数

```
void TrainOneSample(const std::vector<double> &input, const double target);
```

1 サンプルを 1 回学習する．機械学習での利用を想定している．

＜入力＞

input：サンプル入力

target：サンプル目標値

© 2016 堀江 和正

＜出力＞

なし

4.3 TRAIN 関数

```
void Train(const std::vector<std::vector<double>>input, const std::vector<double> target);
```

SDNN の設定に従い，学習サンプルの学習を実行する

＜入力＞

input：サンプル入力リスト，入力ベクトルを要素としたベクトル

target：目標値リスト，目標値を要素としたベクトル．

input[i]の入力に対する目標値が target[i]になっている必要あり

＜出力＞

なし．

4.4 ESTIMATE 関数

```
double Estimate(const std::vector<double> &input);
```

未知のサンプルの識別，推定を行う．

＜入力＞

input：サンプル入力ベクトル

＜出力＞

推定，識別結果

その他の関数については，doc/doxygen/html 内の実装仕様書を参照．

5 入力に関する注意

本ライブラリでは，数値入力，シンボル入力共に可能である．

数値入力の場合は，[0,1]の範囲に収まるように正規化する必要がある．

シンボル入力の場合はシンボルに対応する番号（0 以上の整数）を使用する．

この番号は，コーディングパターンの作成方法によって割り振り方が異なる．

ランダム反転法の場合は，シンボルの序列順に，相関木法の場合は，相関木ファイルの上から順に番号を割り振る．

© 2016 堀江 和正