Module 2におけるニューラルネットワーク学習の手順

mol-infer/2L-model

2021 年 3 月 3 日

目 次

1　はじめに

本稿では，本プロジェクト (`mol-infer/2L-model`) における Module 2 の手順を解説する．

与えられた化学グラフの集合を $D_\pi = \{G_1, G_2, \ldots, G_p\}$, 化学グラフを特徴ベクトルに変換する写像を f とする．$\mathcal{F}(D_\pi) \triangleq \{f(G_1), f(G_2), \ldots, f(G_p)\}$ と定義する (すなわち $\mathcal{F}(D_\pi)$ は特徴ベクトルの集合)．また, 注目する化学的性質を π と書くことにする．この π は，たとえば水可溶性，脂溶性，沸点，燃焼熱，水分配係数など様々である．Module 2 の入力と出力は以下の通りである．

入力: 特徴ベクトルの集合 $\mathcal{F}(D_\pi) = \{x_1, x_2, \ldots, x_p\}$，各化合物 $G_i \in D_\pi$ もといその特徴ベクトル $x_i = f(G_i) \in \mathcal{F}(D_\pi)$ が化学的性質 π に関して持つ値の集合 $\{a(x_1), a(x_2), \ldots, a(x_p)\}$，ニューラルネットワークの各種パラメータの値（隠れ素子の個数など）．

出力: 入力で指定された構造を持ち，$\mathcal{F}(D_\pi)$ における「多くの」特徴ベクトル $x \in \mathcal{F}(D_\pi)$ に対して化学的性質の値 $a(x)$ を「良く」推定するようなニューラルネットワーク．

出力の具体的な中身は，学習されたニューラルネットワークにおける各枝の重みと各ノードのバイアスである．

本稿の構成は以下の通りである．

- 第 2 節: 基本的な用語，およびパッケージのファイル構成の説明．
- 第 3 節: 簡単な実行例．
- 第 4 節: プログラムの入出力に関する詳細．

2　準備

2.1　用語の説明

特徴ベクトル.　各元素の種類の原子数等の化学物質を説明する数値，あるいはグラフの直径等の化学物質のグラフ表現のトポロジーに基づいて計算される数値のベクトル.

人工ニューラルネットワーク (ANN).　人工ニューラルネットワーク (artifician neural network, ANN)，または単にニューラルネットワーク (NN) とは，機械学習で最も確立した手法の 1 つである．これらは入力ベクトルに基づいて値を予測するために用いられる．この冊子では，ニューラルネットワークへの入力は，化合物の特徴ベクトルであり，出力は特定の化学的性質の予測値である.

本プロジェクトで用いるのはフィードフォーワード型のニューラルネットワークであり，これは非巡回有向グラフによって表すことができる．図 1 に例を示す.

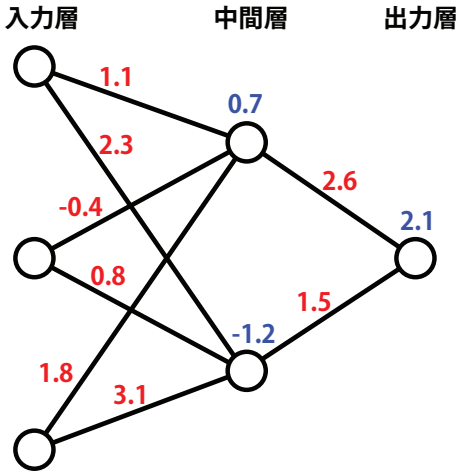

図 1: ニューラルネットワークの具体例．赤色の数値が重み，青色の数値がバイアスを表している．枝はすべて左から右に向いている.

入力層，隠れ層，出力層.　人工ニューラルネットワークの多層パーセプトロンモデルを仮定する．このモデルでは，ニューラルネットワークはいくつかの層で構成されている．最初の層は入力層で，入力層は場合によっては特徴ベクトルから数値データを得るため，特徴ベクトルの要素と同じ数のノードがある．次に数値は隠れ層を介して伝播される．隠れ層は 1 つの層の計算が次の層への入力として用いられる．最後に出力層が入力ベクトルに基づいた予測値を与える.

図 1 では，入力層は 3 つのノードを持つ．したがって入力する特徴ベクトルは 3 次元のものでなければならない．また 1 つの隠れ層を有し，この隠れ層は 2 つのノードを持つ．そして出力層は 1 つのノードを持つ.

重み.　ニューラルネットワークでは，ノード間を接続する枝（有向枝）にはそれぞれ数値が割り当てられ，その値を重みと呼ぶ．入力層から出力層への値の伝播には，これらの重みに基づく計算が含まれている．

図 1 では，枝の重みは赤によって示される．

バイアス.　ニューラルネットワークの隠れ層の各ノードにはバイアスと呼ばれる数値が割り当てられている．この数値は重みと共に，入力ベクトルに基づいて出力値を計算する過程で使用される．

図 1 では，ノードのバイアスは青によって示される．

活性化関数.　活性化関数はニューラルネットワークの各ノードに割り当てられており，与えられた入力ベクトルから出力値を計算する際に用いられる．特に各ノードの出力値は，重み付けされた対応する枝の重みと前の層からのノードの出力の線形結合を入力として与えられた活性化関数の値である．

本プロジェクトでは，**隠れ層の各ノードの活性化関数として Rectified Linear-Unit (ReLU) 関数に限定**して用いる．これは次の Module 3 (この Module 2 で学習したニューラルネットワークに基づいた化合物推定) では，活性化関数が ReLU であることを仮定しているからである．

2.2　ファイル構成

パッケージに含まれるファイルとその役割は表 1 の通りである．

表 1: Module 2 のパッケージに含まれるファイルとその役割

ファイル名	役割
2L_ANN.py	NN を学習するための Python スクリプト **（Module 3 に進むにはこのスクリプトの実行が必要）** • 使用する非標準ライブラリ: numpy，pandas，scikit-learn
Manual_Module_2_2L-model_jp.pdf Manual_Module_2_2L-model_jp.tex fig/ANN_sample_jp.eps	マニュアルの PDF，LaTeX ソースファイル および画像ファイル（日本語版）
Manual_Module_2_2L-model_en.pdf Manual_Module_2_2L-model_en.tex fig/ANN_sample_en.eps	マニュアルの PDF，LaTeX ソースファイル および画像ファイル（英語版）
サンプルデータ (水可溶性 ESOL に関するデータ; http://moleculenet.ai/datasets-1)	
data/Sl_all_eli.sdf	元となる SDF ファイル. 本モジュールでは取扱わない.
data/Sl_all_eli_desc_norm.csv*	Sl_all_eli.sdf に Module 1 の特徴ベクトル生成器を 適用して得られた csv ファイル. $\mathcal{F}(D_\pi)$ に相当.
data/Sl_values.txt**	水可溶性に関する観測値が書かれたテキストファイル. $\{a(x_1), \ldots, a(x_p)\}$ に相当.
data/Sl_all_eli_biases.txt data/Sl_all_eli_weights.txt	* および ** に対し, 本モジュールの学習スクリプト 2L_ANN.py を適用して得られた NN に関するデータファイル. 前者がノードバイアス, 後者が枝重みに関するファイルで, **この二つは次の Module 3 の実行に必要である.**

3　クイックスタート

次のコマンドを入力すれば，data/Sl_all_eli_desc_norm.csv を特徴ベクトル，data/Sl_values.txt を観測値とするようなデータセットに対し，2 つの隠れ層を持ち，それぞれの隠れ層におけるノード数を 20, 10 とするようなニューラルネットワークが, 乱数の種を変えていくつか学習される．なお一つのニューラルネットワークを学習するためのアルゴリズム (Adam 法) の反復回数は 10,000 である．学習されたニューラルネットワークのうち一つがある基準にしたがって選択される．選択されたニューラルネットワークにおける各ノードのバイアスは output_biases.txt，各枝の重みは output_weights.txt にそれぞれ出力される．

```
$ python 2L_ANN.py data/Sl_all_eli_desc_norm.csv data/Sl_values.txt\
                   output 10000 20 10
```

出力されたニューラルネットワークの**重みおよびバイアスに関するファイルは，Module 3 でも使用**する．

4　プログラムの入出力に関する詳細

4.1　入力

4.1.1　特徴ベクトル

特徴ベクトルは，我々が**FV 形式**と呼ぶフォーマットに基づく csv ファイルに記述されている必要がある．Module 1 の特徴ベクトル生成器は化合物に関する SDF ファイルから FV 形式の csv ファイルを生成するため，当該生成器を用いて生成されたファイルを用いれば問題はない．

以下，FV 形式の記述ルールを簡単に記しておく．

```
CID,n,cs,ch,nsH
244,8,6,2,8
307,10,6,4,8
657014,11,7,1,18
16704,9,9,0,10
```

- FV 形式では先頭行に記述子の名前をコンマ区切りで記述する．

- ただし最初の記述子は CID（化合物識別番号）でなければならない．CID の値が学習に用いられることはない．CID は識別のためのみに用いられる．
- 上記の例では，n（水素を除く原子の個数），cs（コアサイズ），ch（コアハイト），nsH（水素原子の個数）と四つの記述子が定められている．
- 二行目以降に，一つの行に一つの化合物の CID および特徴ベクトルを，コンマ区切りで記述する．したがって各化合物は，4 次元の特徴ベクトルで表されることになる．
- 化合物の順序は任意である．

4.1.2 化学的性質の値

化学的性質の値は，CID と値を羅列した csv ファイルに記述されなければならない．

```
CID,a
307,11.2
244,-0.5
657014,98.124
16704,-12.8
117,5.3
```

- 先頭行は CID,a でなければならない．
- 二行目以降に，一つの行に一つの化合物の CID および化学的性質の値をコンマ区切りで記述する．
- 化合物の順序は任意である．

4.2 実行

第 3 節で示したとおり，ニューラルネットワークを学習するには 2L_ANN.py を用いる．

4.2.1 引数

第 3 節で示したコマンドを再掲する．

```
$ python 2L_ANN.py data/Sl_all_eli_desc_norm.csv data/Sl_values.txt\
                   output 10000 20 10
```

各引数には以下を指定する．カッコ内の値は, 上記の例において指定されている値を示す.

- 第1引数: 特徴ベクトルに関するcsvファイルの名前 (`data/Sl_all_eli_desc_norm.csv`).
- 第2引数: 化学的性質の値に関するcsvファイルの名前 (`data/Sl_values.txt`).
- 第3引数: ニューラルネットワークの重み・バイアスを書き込むファイルステムの文字列 (`output`).
- 第4引数: 学習アルゴリズムにおける反復回数 (`10000`).
- 第5引数以降: 隠れ層（中間層）のノードの個数 (`20 10`).

引数を与えずに実行すれば（もしくは引数が適切に与えられなかった場合），引数に関する説明が英語で出力される．

```
$ python 2L_ANN.py
```

引数が適切に与えられると，ニューラルネットワークの学習が始まる．いくつかの乱数の種に対して5分割交差検定が行われ, 最も高い決定係数 (R^2 値) を達成した交差検定において, 五回の学習のうち試験集合に対して最も高い決定係数を実現するニューラルネットワークが採用され，その重みとバイアスが，ファイルに出力される．上記のコマンドの場合，そのファイルの名前は第3引数で指定された `output` に基づく，

- `output_weights.txt`（重み）
- `output_biases.txt`（バイアス）

である．これら出力されたファイルは，Module 3 の実行に必要となる．

データセットに関する注意． データセットは，

- 特徴ベクトルに関するcsvファイル，および
- 化学的性質の値に関するcsvファイル

の二つのファイルから構成されるが，CID は前者ファイルに記されたものすべて，かつそれのみが計算の対象となる．したがって，

前者ファイルに記された CID は，すべて後者ファイルに記されていなければならない．

しかし逆は成り立たなくともよい．つまり，化学的性質の値に関するcsvファイルには，特徴ベクトルのcsvファイルに記されていない,「余計な」CID に関する値が記されていても構わない．そのような値は無視される．

4.2.2 ハイパーパラメータの調整

ニューラルネットワークの学習は scikit-learn ライブラリ[1] における MLPRegressor を用いて行われる．2L_ANN.py の 125 行目以降で MLPRegressor インスタンスの初期化が行われるが，ハイパーパラメータの調整はここで行うことができる．一部のパラメータを以下のように設定している．

- activation: 'relu' **注意:** 学習されたニューラルネットワークを Module 3 以降で用いるには，'relu' (ReLU 関数) でなければならない．
- alpha: 10^{-5}
- early_stopping: False
- hidden_layer_sizes: 実行時に引数で指定した個数に基づく
- max_iter: 10^{10}
- random_state: 変数 ANN_seed の値．この変数の値として，タプル ANN_SEED (26 行目) に格納したすべての値が試される．
- solver: 'adam'

また交差検定の乱数の種は 135 行目で指定される (KFold インスタンスの初期化における random_state の値)．この乱数の種として，タプル SPLIT_SEED (27 行目) に格納したすべての値が試される．

4.3 出力

ふたたび以下のコマンドを取り上げ，その実行によって得られる出力について説明する．

```
$ python 2L_ANN.py data/Sl_all_eli_desc_norm.csv data/Sl_values.txt\
                   output 10000 20 10
```

4.3.1 標準出力

上記コマンドを実行すると端末（標準出力）に計算過程が出力される．

[1] https://scikit-learn.org/stable/

```
### (ANN_seed, split_seed)=(1,1), fold=1/5 ###
# t train test time
100 0.8486 0.8183 0.2774
200 0.9570 0.8509 0.8056
300 0.9757 0.8321 1.5904
400 0.9798 0.8231 2.0535
500 0.9766 0.8232 2.0819
600 0.9793 0.8232 2.1133
700 0.9775 0.8230 2.1433
800 0.9782 0.8233 2.1737
900 0.9777 0.8229 2.2041
1000 0.9775 0.8231 2.2346
2000 0.9777 0.8230 2.2629
3000 0.9775 0.8233 2.2916
4000 0.9775 0.8233 2.3210
5000 0.9774 0.8233 2.3511
6000 0.9774 0.8225 2.3810
7000 0.9774 0.8236 2.4108
8000 0.9774 0.8225 2.4409
9000 0.9771 0.8228 2.4729
10000 0.9771 0.8225 2.5017


### (ANN_seed, split_seed)=(1,1), fold=2/5 ###
# t train test time
100 0.8564 0.7537 0.2840
200 0.9648 0.8513 0.8229

(途中略)

========== COMPLETE ==========

Best R^2 value among all cross-validations: 0.859528

The performance of the selected neural network:
R^2 value for training set: 0.962288
R^2 value for test set: 0.884223

The data for the selected neural network is stored in
output_biases.txt and output_weights.txt.
They are required in Module 3.
```

- 先に述べたとおり, 2L_ANN.py ではいくつかの乱数の種について 5 分割交差検定が行わ

れる．乱数の種とは学習アルゴリズムに関するものと交差検定の分割に関するものの二種類である．前者は ANN_SEED (26 行目) で定め, 後者は SPLIT_SEED (27 行目) で定める．前者と後者に貯えられたすべての組について 5 分割交差検定が行われる．デフォルトでは前者に二つ, 後者に一つの値が入っていることから, $2 \times 1 = 2$ 回の交差検定が行われる．さらに一回の交差検定では五回の学習が行われるので, 全部で $2 \times 5 = 10$ 回の学習が行われることになる．

- 上記の出力例の前半は, 学習アルゴリズムに関する乱数の種 (ANN_seed) および交差検定の分割に関する乱数の種 (split_seed) をいずれも 1 としたときの学習過程を示している．学習アルゴリズムの反復回数を 10,000 と定めたが, 適当な回数が済んだ時点でのニューラルネットワークの決定係数の値 (訓練集合, 試験集合の双方) および学習が開始してからの経過時間が示されている．
- プログラムの最後に, 選択されたニューラルネットワークの性能が出力される．
  - 複数回の交差検定を通じて, 決定係数の最良値 (すなわち最大値) は 0.859528 である．
  - 当該交差検定では五回の学習を行なったはずだが, そのうち試験集合に対する決定係数が最も良かったニューラルネットワークが選択される．そのニューラルネットワークの訓練集合に対する決定係数の値は 0.962288, 試験集合に対する決定集合の値は 0.884223 である．
- 選択されたニューラルネットワークにおける枝重みが output_weights.txt，ノードのバイアスが output_biases.txt に出力される．繰り返すが, この二つのファイルは Module 3 の実行に必要である．

4.3.2 枝の重み

2L_ANN.py が出力する枝重みファイルの書式について説明する．

簡単のため，図 1 に示したニューラルネットワークが学習されたとする．このニューラルネットワークの枝重みは以下のようにファイルに出力される．

```
3 2 1
1.1 2.3
-0.4 0.8
1.8 3.1
2.6
1.5
```

- 最初の行はニューラルネットワークの構造，つまり入力層のノード数，各隠れ層のノード数，最後に出力層のノード数である．
- 2 行目以降は枝重みの値である．各行は 1 つのノードから出る枝重みの値を示す．

4.3.3 ノードのバイアス

同じく，2L_ANN.py が出力するノードのバイアスに関するファイルの書式について説明する．

やはり簡単のため，図 1 に示したニューラルネットワークが学習されたとする．このニューラルネットワークのノードのバイアスは以下のようにファイルに出力される．

```
0.7
-1.2
2.1
```

1 行につき 1 つのノードのバイアスの値が示されている．入力層のノードにはバイアスの値がないことに注意せよ．