

リモートセンシングから大気パラメータを得る ニューラルネットワークの実装

Implementation of a Neural Network for Retrieving Atmospheric Parameters from Remote Sensing

フィリップ バロン ヤナ メンドロック 笠井康子

P. Baron, J. Mendrok, and KASAI Yasuko

要旨

教師あり学習を行うフィードフォワードニューラルネットワーク (NN) の数値モデルを実装した。目的は、将来打ち上げられるリム放射サウンダ JEM/SMILES の観測の逆問題を解く、NN 型リトリバルアルゴリズムの性能を調べることにある。このモデルは研究目的で設計された。柔軟性と拡張性が重視されているが、同時にその計算能力が衛星データ解析に多く用いられる規模のネットワークに十分対応する必要があるという点についても考慮されている。プログラムは Python で書かれている。本稿では NN の生成・使用手続きを示したあと、学習手続きのアルゴリズムについて詳述する。MLP の学習は、Levenberg-Marquardt 法または最急降下法を用い、モデルパラメータの最適値を入出力の例にしたがって探索する。本モデルには、データの基準化を行う関数やその主成分を使用するための関数が備わっている。MLP が教師信号に過度に従うことを防止するため、複数の手法がある。コスト関数に正規化項を追加すれば、ベイズ統計の法則を用いたハイパーパラメータの最適化が行える。また相互検証データセットを用いた学習の早期打ち切り手続きも設定可能である。アルゴリズム構築の正しさをシミュレーションによって実証し、結果について論じる。

A numerical model of a supervised feedforward Neural Network (NN) has been implemented. The purpose is to study the capabilities of a NN based retrieval algorithm to inverse the measurements performed by the future JEM/SMILES limb sounder. The model has been designed for research purpose with a special care given to its flexibility and extension facility, but keeping in mind that the computational performances must allow the use of a network with the size of those commonly used for satellite data inversion. The code is written in the Python language. The procedure to create and use a NN is presented and the algorithms of the training procedure are described in detail. The MLP is trained using either the Levenberg-Marquardt or the steepest descent method to find the optimal value of the model parameters according to some examples of the inputs and outputs. The model also provides a set of functions to scale the data or to use their principal components. In order to prevent the MLP to overfit the training data, several solutions are available. A regularization term can be added to the cost function with the possibility to optimize the hyperparameters using a Bayesian method. Also, an early stopping procedure can be set using a cross-validation data set. The correctness of the algorithms implementation is demonstrated with simulations and the results are discussed.

【キーワード】

ニューラルネットワーク, 多層パーセプトロン, 教師あり学習, 衛星リモートセンシング, リトリバル

Neural network, Multilayer Perceptron, Supervised learning, Satellite remote sensing, Retrieval

1 背景

超伝導サブミリ波リム放射サウンダ (SMILES) は 2009 年に打ち上げが予定されている高感度放射計で、中層大気 (約 10 km~80 km) の化学の研究を目的としている [1]。SMILES は国際宇宙ステーションの日本実験棟「きぼう」(JEM) に搭載され、微量気体を地球規模で観測する。独立行政法人宇宙航空研究開発機構 (JAXA) では現在、最適推定法 (OEM) [2] によるデータ処理チェーンを開発中である。このプロセスは非線形フォワードモデルを使って逆問題を解くに当たり、時間のかかる逐次解法を用いている [3]。

ニューラルネットワーク (NN) を用いたリトリバルアルゴリズムが JEM/SMILES のデータをどの程度処理できるのかを調べる研究が現在、NICT において実施されている。そのために、NN の特殊なグループである、教師あり学習を行うフィードフォワード NN [4] が開発された。フィードフォワードネットワークはニューロン層が継続的に連結されたものであり、第 1 層から最終層まで一切のフィードバックをすることなく情報が処理されていく。本稿では主として多層パーセプトロン (MLP)¹ を扱う。これはリトリバルによく使用される、特殊なケースのフィードフォワードネットワークである。また「教師あり」とは、モデルパラメータが学習手続きを通して幾つかの入出力例について最適化されることを意味する。

NN は、非線形のリトリバル問題を解く代替的な手法である。NN の出力計算は高速であると同時に、必要なメモリ量が少ない。計算時間コストは、データ処理段階の前に独立して実施される学習段階に費やされる。学習データはフォワードモデルを使って作成でき、実測値を必要としない。また、解の変数に関する荷重関数も不要である。

衛星観測に関する理論的研究では、MLP を用いたアルゴリズムが従来の統計的手法に近い性能 (精度、確度、垂直分解能など) を発揮することが示されている [5] - [9]。実際に幾つかのアルゴリズムが実測値に適用され、既に成功している [6] [10] - [14]。なかでも特に注目されているのが、衛星 Odin に搭載されたサブミリ波放射計 (SMR: Submillimeter Radiometer) の観測値に対する逆問題解析である [15]。JEM/SMILES の観測値は Odin/SMR に類

似すると思われるが、そのリトリバル手法は同等の問題に対応できるはずである。

本稿ではモデルそのものに的を絞り、JEM/SMILES データ処理への適用については扱わない。なお、モデルの用途は JEM/SMILES のリトリバル解析にとどまらず、非線形回帰問題の解析、より一般的には関数近似問題や分類問題の解析にも使用できる [16]。

2 では、本モデルの基本的特徴、その生成法及び使用法について述べる。モデルの実現についても触れる。3 では学習手続きのアルゴリズムについて、また 4 ではネットワークの学習時に学習データに追従しすぎるといった一般的な問題の回避方法について論じる。最後に 5 ではアルゴリズム適用の有効性をシミュレーションによって実証する。

1 MLP は少なくとも一つの中層を持つ。中間ニューロンは入力の変換可能な非線形関数を実行し、出力ニューロンは線形演算を実行する。MLP は任意の正則関数がある精度で近似できることが示されている。

2 基本的特徴

学習を施した MLP は、JEM/SMILES のリトリバル問題における未知の逆関数を次式のように近似することができる。

$$y_n \approx \text{MLP}(x_m, w_p) \quad (1)$$

ここに、 y_n は未知の大気状態を表す n 次元ベクトルである。これには、未知の装置パラメータも含む。 x_m はノイズを含んだ観測値を表す m 次元ベクトル、 w_p は MLP パラメータの p 次元ベクトルである²。

MLP は、サンプリングした q 個の要素のグリッド (すなわち学習データセット $\{x_m, y_n\}_q$) に対して定義される任意の正則関数の補間器として作用する。このアナロジーから本モデルの限界が明らかとなる。具体的には、i) m 次元空間のうち学習データセットが張る空間の外では誤った結果となる。ii) 学習データが希薄な場合に良好な結果が得られないことがある。iii) 学習データに含まれるノイズや誤りに影響される。

JEM/SMILES のリトリバル解析では、学習データセットがフォワードモデルを使って生成される。そこには測器による観測誤差と大気変動の

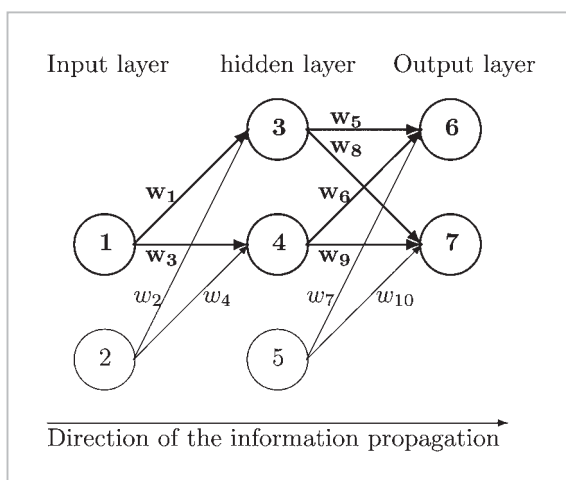


図1 1×2×2個のニューロン(具体的には、入力ニューロン1個、出力ニューロン2個及び中間層のニューロン2個)とバイアスを持つ2層ネットワークの模式図

バイアスは、発火の強さが常に1である仮想ニューロンと実ニューロンとをつなぐ結合の重みで表される。実ニューロンは濃い円(1, 3, 4, 6, 7)で、バイアスは薄い円(2, 5)で表している。結合の重み($w_1 \sim w_{10}$)はモデルにおける可変パラメータである。

統計に関する知識が盛り込まれる。なお、そのような作業を行わずに実際の観測値を使用することも可能である[12]。

図1に示すように、MLPはニューロンが多層に結合した構造をしている。第1層と最終層をそれぞれ入力層及び出力層という。両層の間に中間層(隠れ層ともいう)がある。入力層のニューロンは何の演算も行わず、単に第1中間層のニューロンへの入力となる。それ以外の層のニューロンは次式のような演算を実行する。

$$a[j] = f\left(\sum_i w_{ji} a[i] + \theta[j]\right) \quad (2)$$

ここに、 $a[i]$ と $a[j]$ は*i*番目と*j*番目のニューロンが持つ発火の強さである。番号「*i*」は、*j*番目のニューロンが所属する層の直前の層にあるすべてのニューロンに対応する。また w_{ji} は*i*番目と*j*番目のニューロンをつなぐ結合の重み、 $\theta[j]$ はバイアスである。重み w とバイアス θ は本モデルの自由パラメータである。最終層ニューロンの発火がMLPの出力となる。

デフォルトの f は中間ニューロン(中間層に存在するニューロン)において $f(x) = \tanh$ であり、出力ニューロンでは恒等関数である。ユーザは独

自の発火関数を定義することができる。ただし、同一層の全ニューロンに対して同じ関数を与えねばならない。

本稿では、ネットワークの大きさは中間層と出力層を合わせた層数と定義する。ニューロンの個数は $N_i \times N_1 \times N_2 \dots \times N_o$ の形で表す。ここに、 N_i は入力層のニューロン数、 N_1 と N_2 は第1及び第2の中間層のニューロン数、 N_o は出力層のニューロン数である。例えば図1は1×2×2個のニューロンを持つ2層ネットワークを示している。なお、バイアスは一定の発火強度を持つ仮想ニューロンで表されるが、ニューロンの表示においてバイアスはカウントされない。

2 本稿では下付きの添え字はベクトルないし行列の次元を、また上付きの添え字 T と -1 はそれぞれ転置と逆行列を表す。

2.1 データの基準化と圧縮

学習手続きの質と頑健性(外部影響の受けにくさ)は、学習アルゴリズムを適用するデータを基準化及び圧縮することによって改善できる[6][9][17]。本モデルはこれらの変換が実施できる。これらのオプションはオンにしてもユーザから見えないためにモデルの入出力は変わらない。

基準化入力データ \tilde{x}_m は次式によって計算される。

$$\tilde{x}_m = \frac{x_m - \text{mean}(\tilde{x})_m}{3 \times \text{std}(\tilde{x})_m} \quad (3)$$

ここに \tilde{x} は入力を代表するサンプルである。

データの圧縮には主成分分析(PCA)を用いる。圧縮した入力 κ_a は、ある代表サンプル \tilde{x} の共分散行列に対する最初の*a*個の固有関数から成る基底で x_m を表したものである。すなわち、

$$\kappa_a = U_{a,m} x_a \quad (4)$$

行列 $U_{a,m}$ の行は、最初の*a*個の正規直交固有関数を固有値(成分の分散)の大きい順に並べたものである。

圧縮レベル(a/m の比)は、次式の逆変換を実施して得られる再構成入力 x_m^r の誤差に左右される。

$$x_m^r = (U^T)_{m,a} \kappa_a \quad (5)$$

自乗誤差の和 $\sum_{i=1,m}(x^r[i] - x[i])^2$ に関するしきい値はユーザが指定する。

この方法は、分散が小さい(固有値が小さい)主成分は情報の重要度が低く、したがって無視できることを前提にしている。言い換えると、観測値において分散の小さいパラメータの情報は圧縮手続きにおいて捨てられる可能性がある。観測値で分散が小さくなる原因として、学習データセットの大気パラメータにばらつきが少ないことや、観測値の感度が悪いことが考えられる。

ここでは入力に対する基準化及び主成分(PC)変換にしか触れなかったが、出力 y_n についても同様の操作を行うことができる。

2.2 実装について

本モデルは研究目的で作成され、柔軟性と拡張性に重点が置かれている。一方で、本モデルは衛星観測値の逆問題解析によく使用される規模のMLPを扱える必要がある。例えば Odin/SMR の観測値に対する逆問題解析には $30 \times 10 \times 1$ のMLPと4,000件の例を持つ学習データセットが使用され[10]、1ネットワークごとに一つの大気パラメータを取得する。Airesら[18]の解析では $671 \times 50 \times 30$ のMLPと2,700件の例を持つ学習データセットを使用し、大気サウンディング赤外干渉計(IASI: Infrared Atmospheric Sounding Interferometer)の観測値をシミュレートしたものから温度プロファイルを求めている。Odin/SMRの学習手続きでは4,000個の方程式からパラメータ321個分のパラメータの最適値を推定しなければならない。しかしIASIの場合、システムの大きさは35,130個のパラメータと81,000個の方程式になる。

こうした制約の下、Pythonを使ってプログラムコードが書かれた。Pythonはインタープリタのオブジェクト指向言語である(<http://www.python.org/>)。コードそのものは三つのファイルで構成されるが、NumPyという外部ライブラリ(<http://www.scipy.org/>)を利用することで基本的な数学関数の使用や配列操作が可能になり、インタープリタ言語の欠点である遅いループを減らすことができる。PythonがNumPy及び2次元プロットライブラリmatplotlib(<http://matplotlib.sourceforge.net/>)によって今回の分析にとって実用的かつ十分な環境となる点に留意したい。本モ

表1 $4 \times 3 \times 2$ のMLPと q 個の例を持つ学習データセット $\{X_{t_4,q}, Y_{t_2,q}\}$ で関数 $Y_2 = f(X_4)$ を近似する処理シーケンス。コードの意味は本文中に説明している

```
1> import mmlp
2> m = mmlp.Mlp([4,3,2], bias=1,
               act_hidden='tanh', act_last='linear')
3> m.setPcx(Xt, 1.e-6)
4> m.setScaleY(Yt)
5> m.initweights()
6> m.fitweights(Xt,Yt)
7> m.run(X)
8> Y = m.getY()
```

デルはすでにLinuxとWindowsにおいて試験されている。現在のところ、Odin/SMRネットワークの学習に要するCPU時間は通常のPCで数分であり、IASIの学習で数日である。後者のケースが本モデルの限界だと考えてよい。しかしコードのインプリメンテーションは今後更に改善される上、そのパフォーマンスはJEM/SMILESのリトリバルを実行するのに十分良好である。

MLPはPythonのオブジェクトインスタンスである。これは、各ニューロンの発火の強さ、重みやトポロジーといったアトリビュート(属性やプロパティのこと)を変更及び参照する一連のメソッド(関数のこと)を提供する。モデルを設定ないし実行する様々な操作はPythonのスクリプトで書くことができ、通常のプログラムとして実行できる。スクリプトはPythonシェルから直接実行することもできる。その場合、MLPは対話形式で実行されるため、操作が終わるたびにMLPの属性分析を行うことができる。

例として表1に幾つかの命令を示す。これは関数 $Y_2 = f(X_4)$ を $4 \times 3 \times 2$ のMLPで近似するスクリプトである。なお、表に示したメソッド群は完全なものではない。

この例では、最初にMLPモジュールをPythonのモジュールとしてインポートしている(1行目)。次に、バイアスを持つ $4 \times 3 \times 2$ のMLPのオブジェクトを m という名前で作成する(2行目)。中間ニューロンと出力ニューロンに対してそれぞれ \tanh と linear の発火関数を設定している。入力データは q 個の例を持つサンプル $X_{t_4,q}$ の主成分に変換され(3行目)、出力データはサンプル $Y_{t_2,q}$ に関してスケールされる(4行目)。出力データの

圧縮レベルに対する制御しきい値は 10^{-6} に設定している。重みを無作為に初期化し(5行目)、学習データセット $\{X_t, Y_t\}_q$ に従って重みを計算する(6行目)。最後に、このモデルを使って入力 X_4 を計算し(7行目)、出力 Y_2 を求める(8行目)。

3 学習アルゴリズム

3.1 回帰アルゴリズム

学習手続きの目的は、 q 個の入力 x_m 及び期待値 z_n を持つ学習データセット $\{x_m, z_n\}_q$ から MLP パラメータ w_p の最適値を推定することであった。解は、次式で与えられるコスト関数 ϵ^2 を最小化する。

$$\begin{aligned} \epsilon^2 &= \frac{\beta}{2} [\tilde{z}_o - \tilde{y}_o(\{x_m\}_q, w_p)]^T [\tilde{z}_o - \tilde{y}_o(\{x_m\}_q, w_p)] \\ &+ \frac{\alpha}{2} w_p^T w_p \quad (6) \\ &= \beta E_y^2 + \alpha E_w^2 \end{aligned}$$

ここに、 $o = n \times q$ 、 \tilde{z}_o は期待値の集合 $\{z_n\}_q$ で作られるベクトル、 \tilde{y}_o はすべての学習入力 $\{x_m\}_q$ をモデルに与えたときの出力である。解を安定化させるためと、 w_p は小さな値が好ましいために E_w の項を含めている。

コスト関数が最小となるパラメータは、最急降下(SD)法(勾配法ともいう)[2][4]又は Levenberg-Marquardt(LM)法[2][19]を用いて求める。いずれも逐次アルゴリズムであり、逐次操作のたびに重みを下記のように修正する。

$$dw_p = -\eta \left(\frac{\partial \epsilon^2}{\partial w_p} \right)_p \quad (7)$$

コスト関数の導関数は次式で与えられる。

$$\left(\frac{\partial \epsilon^2}{\partial w_p} \right)_p = -\beta J_{o,p}^T [\tilde{z}_o - \tilde{y}_o(\{x_m\}_{1\dots q}, w_p)] + \alpha w_p \quad (8)$$

ここに、 $J_{o,p} = \frac{\partial \tilde{y}_o}{\partial w_p}$ は標準的な誤差逆伝播(バックプロパゲーション)法[4]を用いて計算される。またパラメータ η は次式となる。

$$\eta = \begin{cases} \mu^{-1}, & \text{SD 法の場合} \\ [\beta(J^T J)_{p,p} + (\mu + \alpha)I_{p,p}]^{-1}, & \text{LM 法の場合} \end{cases}$$

ここに、 $I_{p,p}$ は $p \times p$ 単位行列、 μ はステップ幅パラメータの逆数(SD法の場合)又は Marquardt パ

ラメータ(LM法の場合)である。LM法でのパラメータ η は、コスト関数のヘッセ行列のガウス近似から導かれる。具体的には次式となる。

$$\frac{\partial^2 \epsilon^2}{\partial w_p^2} \approx [\beta(J^T J)_{p,p} + \alpha I_{p,p}] \quad (9)$$

μ の初期値はユーザが設定し(デフォルト値は10)、コスト関数の最小値に向かって収束するよう逐次操作が終わるごとに更新される。SD法及びLM法のいずれでも μ の更新には Marquardt の方法が使用される。 w_q の更新後にコスト関数(式6)が増加した場合は、あるユーザ定義値(デフォルト値は10)をパラメータ μ に乗じ、新たな重み推定を行う(式7)。コスト関数が減少した場合は μ を別のユーザ定義値で割り、ヤコビアン及びヘッセ行列を計算してから w_q を更新する。収束状態に達するまでこの過程を繰り返す。LM法の場合は収束状態に達した後も μ をゼロに設定することで新たなループが実行できる。

二つの連続する逐次操作 i から $i+1$ までのコスト関数変化が小さいときに収束状態に達したという。収束の判定基準は $|\epsilon^2(i+1) - \epsilon^2(i)| < (\text{thr} \times \epsilon^2(i))$ で定義される。しきい値 thr はユーザによって定義される(デフォルト値は 10^{-5})。収束状態に達する前に逐次過程を停止する他のメカニズムについては 4 で幾つか提示している。

逐次過程の開始時においてバイアスはゼロに設定され、結合の重みは -1 から 1 までの値が無作為に設定される。入力データを基準化すると、文献[20]で提唱される方法を使って入力層と第1中間層の間の重みとバイアスを初期化することも可能になる。

LM法は安定しており、SD法よりも短時間で解に収束する。一方、SD法ではヘッセ行列(式9)の計算と逆演算が不要であるため、2.2で触れたIASIのケースのようにパラメータが多いネットワークの学習に適している。

行列 $J_{o,p}$ (式8)は規模が大きいが、ヘッセ行列(式9)とコスト関数のヤコビアン(式8)しか使用しないためにメモリには保存されない。本モデルにおいて学習データセットは q 個の要素を持つ n 個の部分集合に分割される。それぞれの部分集合は一つのニューロン出力に対応する。ヘッセ行列とヤコビアン行列は次式を用いて計算される。

$$(J^T J)_{p,p} = \sum_{i=1,m} (J_i^T)_{p,q} (J_i)_{q,p}$$

及び

$$\left(\frac{\partial \epsilon^2}{\partial w_p} \right)_p = -\beta \sum_{i=1,n} (J_i^T)_{p,q} \tilde{e}_q^i + \alpha w_p$$

ここに、 $J_{i,p,q}$ はヤコビアン行列、 \tilde{y}_q^i は q 個の学習データについて推定される i 番目のネットワーク出力の発火である。

3.2 ベイズの正規化

正規化した回帰アルゴリズムを確率的に見れば、ハイパーパラメータ α 及び β (式 6) はそれぞれ MLP パラメータ w_p が学習前に持つ分散の逆数及び誤差 $\hat{E}_y = [\tilde{z}_o - \tilde{y}_o(\{x_m\}_q, \hat{w}_p)]$ の逆数と解釈される。パラメータ \hat{w}_p は w_p の最確値である。 \hat{w}_p を用いて推定されるパラメータには今後「 \wedge 」記号を付けることにする。

ハイパーパラメータの最適値の推定は、ベイズの定理[21][22]を用いて行う。

$$\alpha = \frac{\gamma}{2\hat{E}_w^2} \quad (10)$$

$$\beta = \frac{o - \gamma}{2\hat{E}_y^2}$$

ここに $\hat{H}_{p,p}$ はコスト関数のヘッセ行列(式 9)であり、また、

$$\begin{aligned} \gamma &= p - \alpha \text{trace}(\hat{H}_{p,p}^{-1}) \\ &= \sum_{k=1}^p \frac{\hat{\lambda}_k}{\hat{\lambda}_k + \alpha} \end{aligned} \quad (11)$$

は $\hat{\lambda}_k$ of $\beta (J^T J)_{p,p}$ の固有値 $\hat{\lambda}_k$ から計算できる有効パラメータ数である。

LM 法を使用する場合は、逐次過程の各段階が終わるごとに現段階で使用できる w_p と $H_{p,p}$ の値を用いてハイパーパラメータを更新するアルゴリズムが提唱されている[23]。SD 法を選択する場合にはヘッセ行列の計算が必要になる。しかしそのアルゴリズムでは α の値が過大になって解に対して制約がかかりすぎることがある。そのため、より安定であるとされる次の二つの方法がある。

一つ目[24]は、下記の逐次過程からパラメータを推定する方法である。

$$\gamma(i+1) = \sum_{k=1}^p \frac{\lambda_k}{\lambda_k + \alpha(i)} \quad (12)$$

$$\alpha(i+1) = \frac{\gamma(i+1)}{2E_w^2}$$

ここに、 i は反復番号である。ループ回数 i_{max} はユーザ定義値であり、デフォルトは 1 に設定されている。この値は文献[23]で提唱されている解である。パラメータ β は最後の推定値 $\gamma(i_{max})$ を用いて(式 10)から算出される。

二つめの方法[25]では α を次式で近似する。

$$\alpha \approx \alpha(0) \frac{p}{p - \gamma(0)}$$

γ と β はこの新しい α を使って計算される。

α 、 β 、 γ に対する初期値はそれぞれ 0、1、 p が使われる。文献[25]の記載のように、ハイパーパラメータの更新が行われるのはユーザが指定した反復回数後である(デフォルト値は 3)。

ハイパーパラメータが更新されるとコスト関数の定義が変わるため、Marquardt ループサイクルを続ける前にその推定をやり直す必要がある。新しいコスト関数の値は γ の値に等しくなる(式 6 と式 10 を参照)。

有効パラメータ数 γ はネットワークの自由度である。それは常にパラメータ w_p の個数より小さい。そのためその最適値は、ネットワークの最適な大きさを推定する際の客観的基準を与える。ニューロン数が極めて少ないネットワークの場合、 γ の値はパラメータ w_p の個数に近い。 γ はネットワークが大きくなるにつれて増大し、最適な自由度に近づいていく。ネットワークにとって最良の大きさは、パラメータ w_p の個数が γ の最大値を若干超えるときの大きさである。

学習手続きの全体を表 2 にまとめる。

4 過剰学習の問題

過剰学習は、ノイズの多いデータで MLP を学習させるときに多くみられる問題である。学習データのノイズパターンは完璧に再現されるが、新しい入力データを与えると MLP の性能が低下する。本モデルには、この問題に対処し、MLP

表2 学習アルゴリズムの主要ステップのまとめ。[*]印はオプション

1. *scale the input/output data (Eq 3)
2. *convert the input/output to PC (Eq 4)
- 3a compute the cost function ϵ^2 (Eq 6)
- 3b calculate the Jacobian of ϵ^2 (Eq 8)
- 3c *calculate the Hessian matrix (Eq 6)
4. update the weights and biases w_p (Eq 9)
5. in case of LM:
 - 5a. if $\mu \neq 0$ and the convergence is reached, set $\mu_{prev} = \mu$
set $\mu = 0$ and return to step 3b.
 - 5b. if $\mu = 0$ and the convergence is reached, stop the process
 - 5c. if $\mu = 0$ and ϵ^2 increases, set $\mu = \mu_{prev}$ and return to step 3a.
- 6 in case of SD:
 - 6a. if the convergence is reached, stop the process
 7. if ϵ^2 decreases, increase μ
set w_p to the previous value and return to step 4.
- 8a. *in case of Bayesian regularization, if ϵ^2 increases, update the hyperparameters (Eq 10) and ϵ^2
- 8b if ϵ^2 increases, decrease μ and return to step 3b

The convergence is reached if one of the following tests is true:

1. small change of the cost function:
 $|\epsilon^2(i+1) - \epsilon^2(i)| < (\text{thr} \times \epsilon^2(i))$,
2. number of iterations superior to a threshold,
3. early stopping criterion fulfilled
(see section 4.2)

が学習データを滑らかに再現する方法が幾つか実装されている。

4.1 構造的安定化と正規化

MLP の複雑さ、言い換えると複雑なパターンを持つ関数を表す能力は、パラメータ w_p の個数とその振幅によって制御される。モデルの複雑さは、パラメータ数を超えられない自由度によって決まる。モデルの自由度をコントロールする類似の方法に、正規化したコスト関数(式 6)を用いる方法とパラメータ数を操作する方法がある。なお、モデルの自由度が低すぎると関数を正しく近似できないので注意が必要である。

リトリーバル問題を解くには 2 層のネットワークで十分である [14][18]。3.2 で述べたようにベイズ流の学習手続きを用いれば自由度 γ (有効パラ

メータ数)の最適値が求まり、ひいては最良の中間ニューロン数を推定することが可能になる。

4.2 学習の早期打ち切り

学習アルゴリズムの逐次過程は収束状態の到達前に打ち切ることができる。早期打ち切り手続きを実行することでパラメータ w_p が過大な値になることを防止することができる。学習手続きを打ち切るには最大反復回数を定義すればよい。あるいは、モデルの一般化能力が劣化し始めたことを示す相互検証データセットを使用するという方法もある。ユーザが設定したある反復回数の間、相互検証コスト関数が減少しないと、学習過程が打ち切られる。

5 アルゴリズムの検証

5.1 ニューロンの発火強度と出力のヤコブ行列

ニューロンの発火強度と出力のヤコビアンを計算を行う本アルゴリズム実装を、既知パラメータを持つ参照 MLP を用いて検証した。参照 MLP は図 1 のような $1 \times 2 \times 2$ の単純なネットワークである。パラメータ w_p の値を表 4 に示す。発火関数は中間ニューロンが \tanh で、出力ニューロンが恒等関数である。図 2 に、 -50 から 50 の入力に対する二つのネットワーク出力を示す。

このネットワークは規模が小さいためにニューロンの発火強度と出力のヤコビアンを計算する数式を解析的に容易に導くことができる。表 3 に、中間ニューロン及び出力ニューロンの発火強度を

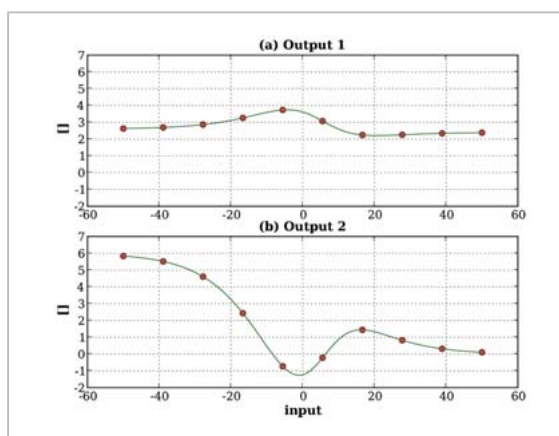


図2 参照 MLP の第 1 出力(右)と第 2 出力(左)

赤い四角のデータはアルゴリズムの検証に使用される。

表3 図1のようなMLPについてニューロンの発火強度を算出する解析的数式 $a[3]$ と $a[6]$ 及びパラメータ $w[1]$, $w[5]$, $w[7]$, $w[9]$ に対する第1出力 $o[1] \equiv a[6]$ のヤコビアン。中間ニューロン及び出力ニューロンによってそれぞれ \tanh と線形の発火関数が実行される

$$\begin{aligned}
 a[3] &= \tanh(w[1] \times a[1] + w[2]) \\
 a[6] &= w[5] \times a[3] + w[6] \times a[4] + w[7] \\
 J[1,1] &= \frac{\partial o[1]}{\partial w[1]} = a[1] \times w[5] \times (1 - a^2[3]) \\
 J[1,5] &= \frac{\partial o[1]}{\partial w[5]} = a[3] \\
 J[1,7] &= \frac{\partial o[1]}{\partial w[7]} = 1 \\
 J[1,9] &= \frac{\partial o[1]}{\partial w[9]} = 0
 \end{aligned}$$

表4 参照MLPにおけるパラメータ w_p の値と、同じモデル出力を与える他の二つの組合せ

バイアスの番号は太字で示した。

index	w_p	other solutions	
1	0.05	-0.05	0.10
2	0.10	-0.10	-0.50
3	-0.10	-0.10	0.05
4	0.50	0.50	0.10
5	1.90	-1.90	-2.00
6	2.00	2.00	1.90
7	2.50	2.50	2.50
8	-10.00	10.00	7.00
9	-7.00	-7.00	-10.00
10	3.00	3.00	3.00

求める関数 $a[3]$ と $a[6]$ を示した。またパラメータ $w[1]$, $w[5]$, $w[7]$, $w[9]$ に対する第1モデル出力 $o[1] \equiv a[6]$ のヤコビアン関数の計算も示してある。

-50 から 50 までの等間隔の 10 点についてモデル出力と解析関数を計算した。全ニューロンの発火強度及び出力ヤコブ行列の要素をチェックした。比較を行ったところ振幅は数値ノイズのレベル ($\sim 10^{-16}$) であり、両計算値の差は無視できることが分かる。ニューロン 3, 4, 6, 7 の発火強度及び出力 1 のヤコビアン行列に関する結果を表 5 に示す。各パラメータの最小値と最大値及び最大誤差も記載する。

表5 -50 から 50 の間に等間隔の 10 個の入力について解析的計算とモデルの比較

モデルは図1のMLPである。 $a[3]$, $a[4]$, $a[6]$, $a[7]$ の発火強度及び 10 個の重みとバイアスに対する第1出力 $o[1] \equiv a[6]$ のヤコビアンに関する結果を示す。左側が変数の範囲で、右側が最大誤差である。

	value range	error
Neuron output		
$a[3]$	-0.98, 0.98	1.1e-16
$a[4]$	-1.00, 1.00	1.0e-16
$a[6]$	2.20, 3.74	4.4e-16
$a[7]$	-1.25, 5.83	1.7e-15
Jacobian for $o[1]$		
J[1,1]	-19.29, 14.89	1.5e-14
J[1,2]	0.04, 1.89	4.4e-16
J[1,3]	-4.30, 15.72	9.7e-15
J[1,4]	0.00, 1.99	4.4e-16
J[1,5]	-0.98, 0.98	1.1e-16
J[1,6]	-1.00, 1.00	1.1e-16
J[1,7]	1.	0.
J[1,8:10]	0.	0.

5.2 コスト関数の最小化

モデルパラメータ w_p の計算における LM 法と SD 法のアルゴリズム実装を前節で定義した参照MLPを用いて検証した。同じ大きさと同じ発火関数を持つ新たなMLPを用意し、参照モデルが再現されるように学習を施した。学習データは参照MLPの10個の出力から成る。これは-50と50の間の等間隔の入力を用いて算出されたものである(図2を参照)。学習について3種類の設定を定義した。具体的には、(1)LM法を使用、(2)基準化した学習データを用いてLM回帰法を実施、(3)基準化したデータを用いてSD回帰法を実施。設定内容と結果を表6にまとめた。コスト関数には正規化項を用いていない($\beta=1$, $\alpha=0$)。二つの連続する逐次操作においてコスト関数の変化が小さくなった時点で逐次過程を打ち切る。具体的な値は設定による(表6を参照)。反復総数は40,000が上限である。各設定を100回繰り返す。パラメータ w_p の初期値はNguyenアルゴリズムを使って毎回変更する(3.1を参照)。

ニューロンの置換及び w_p の符号変更を行ってもモデル出力が不変となる場合がある点是指摘しておく価値がある(表4を参照)。そのため求めた

表6 3種類の設定に対する内容と結果

結果は、各設定について実行した100回の平均値。学習手続き終了時におけるコスト関数を示す。コスト関数が0.01を超える実行は失敗(incorrect)とした。成功(コスト関数が減少)した逐次操作の回数と成功した逐次操作の割合も示す。後者は成功した逐次操作の回数を総数で割った比である。

Setting	1	2	3
regression method	LM	LM	SD
X and Y Scaling	no	yes	yes
number of runs	100	100	100
convergence threshold	1.e-7	1.e-7	1.e-8
μ initial	100	10.	50.
μ increase rate	2.	2.	150
μ decrease rate	10.	10.	300.

Results			
cost function	4.0e-29	3.7e-29	6.5e-05
number of incorrect runs	12	0	0
number of iterations	25	13	18704
successful iteration ratio	0.36	0.67	0.47

パラメータを参照モデルのものと直接比較してもあまり意味がない。学習手続きの質は、検定用データセットから推論される学習後出力の自乗誤差の和によって推定する。検定用データは-50から50までの等間隔の100個の入力を用いて計算する。

図3の上の図は、第1の学習設定を用いて学習させたMLPの誤差を示したものである。誤差は実行番号と検定用セットの入力に対して示してある。100回の実行中、88回が 10^{-15} 未満という無視できる大きさの誤差のモデルを生成した。それ以外の12回の実行では、全入力に対して誤差が0.5を超える誤ったMLPモデルが生成されている。そのような誤ったモデルは明確に検知され、見逃されることはない。

図3の下図は、コスト関数(学習データセットに対して計算)とMarquardtパラメータにおける変動の様子を代表的な5回の実行について反復番号に対して示したものである。4回の実行においては30回から60回の反復で約 10^{-27} という小さいコスト関数に収束している。5回目のケースは上記の12回の誤った学習を施したMLPの一つに該当する。 10^{-2} を超えるコスト関数によって別の極小値にトラップされている。

図4は第2の学習設定に対する結果である。データの基準化によって結果が改善されている。すなわち学習後の全MLPが 10^{-14} 未満の誤差で参

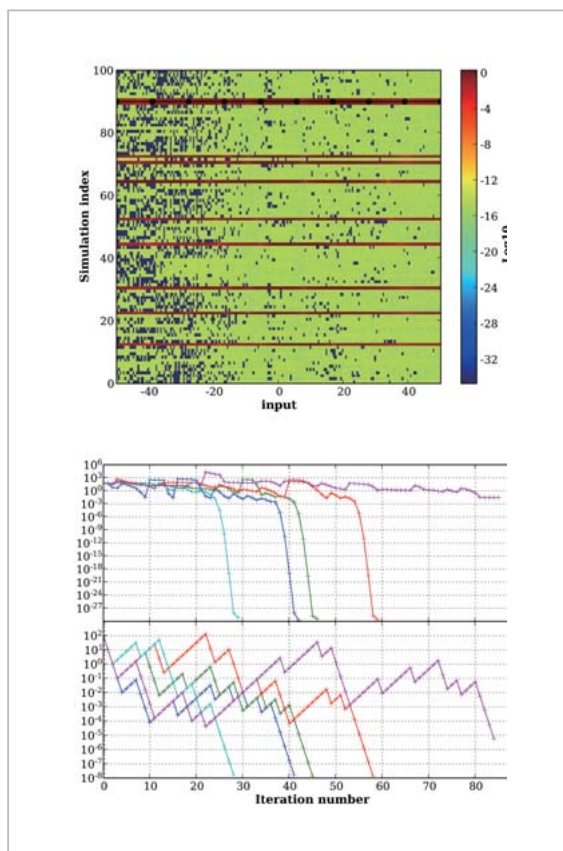


図3 第1の学習設定に対する結果

上：学習後のMLPの出力誤差。ここでいう誤差とは、各検定データについて推定される両出力の自乗誤差の和の平方根である。Levenberg-Marquardt法を使用し、学習データは基準化していない。学習データの位置は黒い丸で示した。
下：学習データセットに対して推定されるコスト関数及び100回の実行中の五つのネットワークに対するパラメータ μ を反復番号に対して表示した。

照モデルを完璧に再現している。いずれのケースも35回以内で収束状態に達している。成功した逐次操作数の平均は13である。

最初の2種類の設定で得られた結果は、Levenberg-Marquardt法によるアルゴリズム実装が有効であることを示している。

最後の学習設定に対する結果を図5に示す。学習は基準化したデータにSD法を施して行った。収束はなかなか進まず、99回の実行において収束状態に達する前に学習が打ち切られた。成功した逐次操作数の平均は約18,000である。シミュレーション番号(Simulation index)が10の付近にある濃い赤色の横線に相当する実行では、コスト関数の変化が小さいために学習手続きが収束したと勘違いし、逐次過程が早期に打ち切られた。とはい

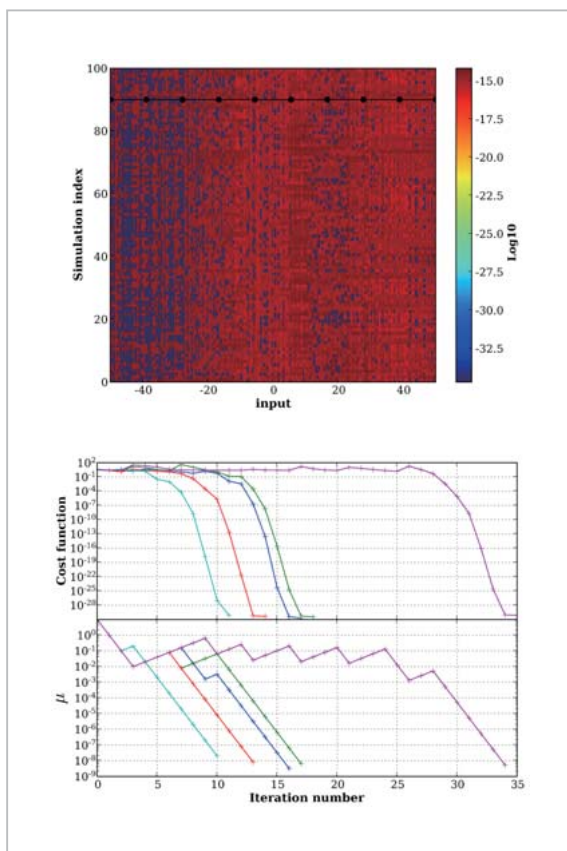


図4 第2の学習設定に対する結果

上：図3の上の図と同じだが、基準化した学習データを用いた。
 下：図3の下図と同じだが、基準化した学習データを用いた。

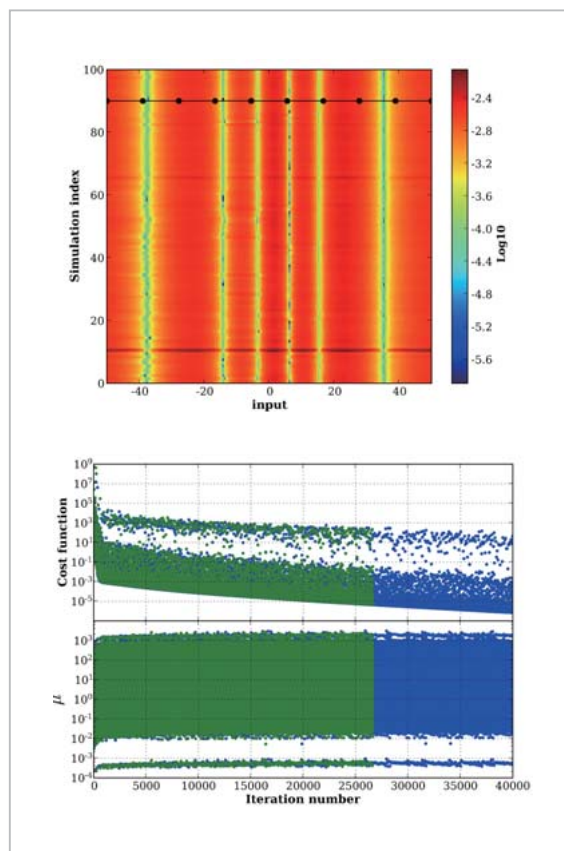


図5 最後の学習設定に対する結果

上：図3の上の図と同じだが、最急降下法及び基準化した学習データを用いた。
 下：図3の下図と同じだが、最急降下法及び基準化した学習データを用いた。

え、学習後の全 MLPが 10^{-2} 未満の誤差で誤差パターンを再現しているため、収束は安定している。最急降下法のアルゴリズムは正しく実現されているものの、逐次操作が終わるたびに行われるステップ幅パラメータ (μ^{-1}) の更新を改善することによって収束速度を早める必要があると結論づけられる。

5.3 ベイズの正規化の検証

この節では、ベイズ的正規化アルゴリズムを提唱する文献[23]に示される三角波関数の近似法を流用する。三角波関数は -0.5 から 0.5 の間の三つの線形部分で構成される (図6)。学習データセットは等間隔の100個の入力と、0.01の分散のノイズを持つ出力から成る。我々は幾つかの $1 \times N_h \times 1$ ネットワークに学習を施した。 N_h は 2, 3, 4, 5, 6, 8, 10, 14 に設定した。各ネットワークについて α と β の初期値はそれぞれ 10^{-6} と 1 である。

その値は逐次操作が 1 回終わるごとに (式 12) のように更新する。Marquardt パラメータ μ の初期値は 1 に設定し、収束状態に応じて 2 を乗じるか、または 10 で割る。

図6の上の図は、コスト関数に正規化項を加えずに学習させた $1 \times 6 \times 1$ のネットワークによる関数近似の様子である。 -0.4 から -0.2 にかけて過剰学習の影響がみられる。下の図は同じ近似だが、基準化したコスト関数とベイズ的アルゴリズムを使用した。過剰学習の影響は消え、近似された関数はノイズのない元の関数とよく一致している。

八つの学習済みネットワークの結果を表7にまとめる。これらのネットワークのパフォーマンスは、ノイズのない学習データセットについて評価した自乗誤差の和 Ea によって推定される。MLPの大きさを増やすと誤差 Ea は 0.16 まで下がる。ネットワークの中間ニューロン数が 4 を超えると誤差 Ea は一定に近くなる。中間ニューロンの数

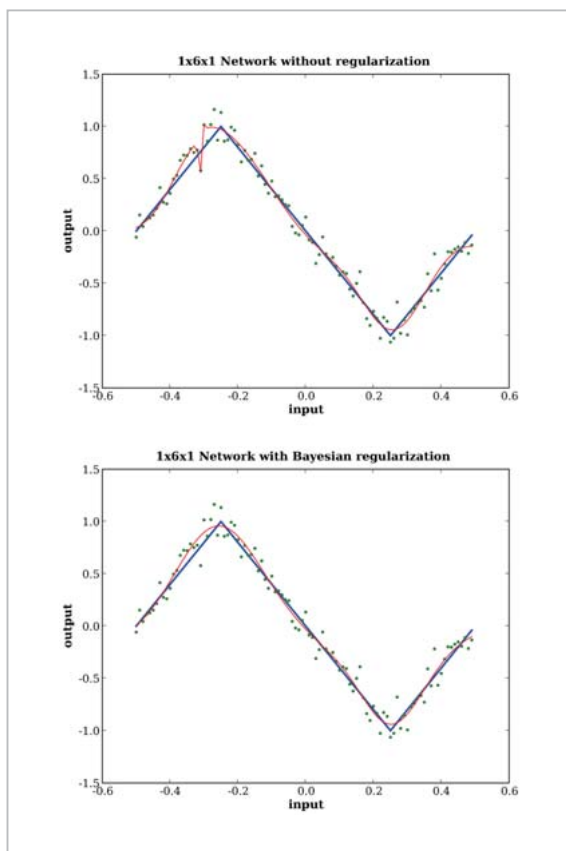


図6 上：ノイズの多い三角波関数(点線)を正規化なしの1×6×1で近似した結果(赤線)

ノイズのない元の関数も示す(濃い青線)。計算は文献[23]のものを改めて実施した。下：上の図と同じだがベイズ的正規化の方法を適用した。

が増えると有効パラメータ数は10.3まで上がり、ネットワークの中間ニューロン数が4を超えるとやはり一定に近くなる。 γ の最大値は中間ニューロン数が3のケースに対応するが、このことは、四つの中間ニューロン(13個のパラメータ)を持つモデルがこの関数の近似にとって適切な選択であることを示している。したがって γ の推定値は誤差 E_a のばらつきと合致している。

オリジナルの分析で得られた結果も表7に示している。誤差 E_a の最小値は0.11で、今回の分析の計算値(0.17)より小さい。 E_a におけるこのような違いは、学習データにノイズが混入することによって説明できる。図7は、ノイズの出現が異なる100の学習データセットを使って1×6×1のMLPを学習させたときの誤差のヒストグラムを示す。ノイズの分散はどのデータセットにおいても0.01である。100件のケース中の15件にお

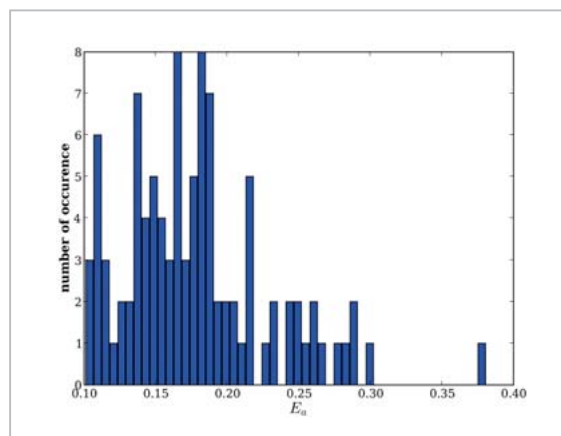


図7 100セットの学習データを用いたときの1×6×1ネットワークの誤差のヒストグラム

学習データによる違いは、初期重みと学習データ上のノイズである。正規化したMLP出力の誤差は $E_a=0.17$ である。

表7 1×Nh×1個のニューロンから成る八つのネットワークの学習結果

Nw はパラメータ w_0 の個数、 E_a はMLP出力と無ノイズの関数の間の自乗誤差の和である。またパラメータ γ は有効パラメータ数である。

Nh	Nw	E_a	γ
2	7	0.56 (0.50)	5.6 (5.7)
3	10	0.26 (0.19)	8.7 (8.5)
4	13	0.16 (0.11)	10.2 (9.8)
5	16	0.16 (0.11)	10.3 (9.9)
6	19	0.17 (0.11)	10.3 (9.9)
8	25	0.17 (0.11)	10.3 (9.9)
10	31	0.17 (0.11)	10.3 (9.9)
14	43	0.17 (0.11)	10.4 (9.9)

いて誤差は0.10と0.12の間である。オリジナルの分析結果は今回の分析の最良ケースに対応する。

6 今後の展望

本稿ではフィードフォワードネットワークのモデルを適用し、それについて説明した。アルゴリズムが正しく構築されたことが示された。ネットワークパラメータの計算に用いる最急降下アルゴリズムは改善が必要である。このほか、よりよい圧縮アルゴリズムの採用や、コスト関数の正規化パラメータにスカラーではなく共分散行列を使用

するといった改善が施されることになろう。

しかし、本モデルの現状からすれば、JEM/SMILESのリトリバル研究を開始することが可能である。研究の次のステップは、10,000種類の大气シナリオと測器ノイズによる観測シミュレーションのデータベース作成である。データベースの一部は学習手続きに、残りは結果の検証に使用されることになる。Odin/SMRの分析に用いた設定と同様に、ネットワークは一つのパラメータのみを与えるように設定する。SMILESの観測値を

十分に表現するように入力の変圧レベルを定義し、中間ニューロンの最良の個数を推定する。

高度分解能、リトリバル確度及びフォワードモデルパラメータ数の誤差についてリトリバル手法の性能を研究する。

謝辞

独立行政法人日本学術振興会には大変お世話になった。ここに感謝の意を申し述べたい。

参考文献

- 1 M. Shiotani, H. Masuko, the SMILES Science Team, and the SMILES Mission Team, "JEM/SMILES mission plan. NASDA Rep. Version 2.1, National Space Development Agency (NASDA), Communications Research Laboratory (CRL)", Nov.,15 2002.
http://www2.nict.go.jp/y/y222/SMILES/Mission_Plan/
- 2 C. D. Rodgers, "Inverse Methods for Atmospheric Sounding: Theory and practice, volume 2 of Series on Atmospheric, Oceanic and Planetary Physics", World Scientific. Singapore-New Jersey-London-Hong-Kong, 2000.
- 3 Y. Kasai, C. Takahashi, S. Ochiai, P. Baron, J. Urban, T. Motoki, Y. Irimajiri, and A. Kleinboehl, "SMOCO: a retrieval code for Super-conductive Sub-Millimetr Limb Emission Sounder (SMILES) on the International Space Station", JQSRT, 2007. submitted.
- 4 C. M. Bishop, "Neural networks and their applications", Rev Sci Instrum, 65(6):1803-1832, Jun. 1994.
- 5 F. Aires, C. Prigent, W. B. Rossow, and M. Rothstein, "A new neural network approach including first guess for retrieval of atmospheric water vapor, cloud liquid water path, surface temperature, and emissivities over land from satellite microwave observations", J. Geophys. Res., 106:14887-14908, 2001.
- 6 W. J. Blackwell, "A neural-network technique for the retrieval of atmospheric temperature and moisture profiles from high spectral resolution sounding data", IEEE Transactions on geoscience and remote sensing, 43:2535-2546, 2005.
- 7 C. Clerbaux, J. Hadji-Lazaro, S. Payan, C. Camy-Peyret, J. Wang, D. P. Edwards, and M. Luo, "Retrieval of CO from nadir remote-sensing. measurements in the infrared by use of four. different inversion algorithms", Applied Optics, 41(33), 2002.
- 8 C. Jiménez, "A neural network technique for retrieving atmospheric species from microwave limb sounders", PhD thesis, Chalmers University, Göteborg, Sweden, 2003.
- 9 C. Jimépnéz and P. Eriksson, "A neural network technique for retrieving atmospheric species from microwave limb sounders", Radio Sci., 36(5):941-953, 2001.
- 10 C. Jiménez, P. Eriksson, and D. Murtagh, "First inversions of observed sub-millimetre limb sounding radiances by neural networks", J. Geophys. Res., 108(24):4791, 2003.
- 11 F. Karbou, F. Aires, C. Prigent, and L. Eymard, "Potential of Advanced Microwave Sounding Unit-A (AMSU-A) and AMSU-B measurements for atmospheric temperature and humidity profiling over land", Journal of Geophysical Research, 110(D9):7109+, Apr. 2005.

- 12 M. D. Müller, A. K. Kaifel, M. Weber, S. Tellmann, J. P. Burrows, and D. Loyola, "Ozone profile retrieval from GOME data using a neural network approach (NNORSY)", *J. Geophys. Res.*, 108:4497, 2003.
- 13 J. Escobar Munoz, "Base de Données pour la Restitution de Paramètres Atmosphériques à l'Échelle Globale - Étude sur l'Inversion par Réseaux de Neurones des Données des Sondeurs Verticaux Atmosphériques Satellitaires présents et à venir", PhD thesis, Université de Paris VII, France, 1993.
- 14 C. Jiménez, P. Eriksson, and D. Murtagh, "Inversion of Odin limb sounding sub-millimeter observations by a neural network technique", *Radio Sci.*, 38(4):8602, 2003.
- 15 D. P. Murtagh, U. Frisk, F. Merino, M. Ridal, A. Jonsson, J. Stegman, G. Witt, P. Eriksson, C. Jiménez, G. Mégie, J. de La Noë, P. Ricaud, P. Baron, J. R. Pardo, A. Hauchecorne, E. J. Llewellyn, D. A. Degenstein, R. L. Gattinger, N. D. Lloyd, W. F. J. Evans, I. C. McDade, C. S. Haley, C. Sioris, C. von Savigny, B. H. Solheim, J. C. McConnell, K. Strong, E. H. Richardson, G. W. Leppelmeier, E. Kyrölä, H. Auvinen, and L. Oikarinen, "An overview of the Odin atmospheric mission", *Can. J. Phys.*, 80(4):309-318, 2002.
- 16 M. Van der Baan and C. Jutten, "Neural networks in geophysical applications", *Geophysics*, 65:1032-1047, 2000.
- 17 F. Aires, W. B. Rossow, N. Scott, and A. Chedin, "Remote sensing from the iasi instrument. 1 compression, de-noising, and first-guess retrieval algorithms", *J. Geophys. Res.*, 107(D22), 2002.
- 18 F. Aires, A. Chedin, N. Scott, and W. B. Rossow, "A regularized neural network approach for retrieval of atmospheric and surface temperatures with the iasi instrument", *Journal of Applied Meteorology*, 41(2):144-159, 2002.
- 19 M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm", *IEEE Transactions on Neural Networks*, 5:989-993, 1994.
- 20 D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights", *International Joint Conference of Neural Network*, 3:21-26, 1990.
- 21 D. J. C. MacKay, "Bayesian interpolation", *Neural Comput.*, 4(3):415-447, 1992.
- 22 D. J. C. MacKay, "Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks", *Network: Computation in Neural Systems*, 6:469-505, 1995.
- 23 F. D. Foresee and M. T. Hagan, "Gauss-newton approximation to bayesian learning", In *Proceedings of the 1997 IEEE International Conference on Neural Networks.*, pages 1930-1935, 1997.
- 24 S. Gutjahr, "Improving the determination of the hyperparameters in bayesian learning", In *Proceedings of the ACNN '98, Brisbane*, 1998.
- 25 Jan Poland, "On the robustness of update strategies for the bayesian hyperparameter alpha", 2001.

Philippe Baron

電磁波計測研究センター環境情報セン
シング・ネットワークグループ特別研
究員 Ph.D.

大気リモートセンシングのためのフォ
ワード・リトリーバルモデルの開発



Jana Mendrok

電磁波計測研究センター環境情報セン
シング・ネットワークグループ専攻研
究員 Ph.D.

放射伝達モデルと雲のリモートセンシ
ング



かさい やすこ
笠井康子

電磁波計測研究センター環境情報セン
シング・ネットワークグループ主任研
究員 博士(理学)

大気分光リモートセンシング