

# 表計算ソフトを用いた機械学習教材に関する一検討

前田 篤彦<sup>1,a)</sup>

**概要:** 本論文では、機械学習に関して動かしながら学べる学習教材（実装例）を表計算ソフト上で適切に作成する方法を検討する。機械学習モデルを用いた計算処理は学習時と推論時の処理に分けられるが、比較的習得が困難な学習時のアルゴリズムに関する教材の実現方法に焦点をあてる。具体的には、従来の試みと異なり、機械学習モデルの学習過程を表計算ソフトの内部機能に頼らず、インタラクティブかつ少ないセル数で一覧性高く実装しやすい汎用的な方法を提案し、あわせて学習過程にグラフ描画等の可視化機能を連動させることと、表計算ソフトを使うからこそその注意点として学習時の処理のフローが把握しやすいよう配慮して実装すべきことを提案し、これらに関するいくつかの実装例を示す。

## Machine Learning Instructional Materials Using Spreadsheets

### 1. はじめに

日本の教育において、数理・データサイエンス・AIの重点化が進められている [1]。特に高等教育においては、情報系の学部・学科だけでなく、すべての大学・高専生に対して、それらへの関心を高め、適切に理解し活用する基礎的な能力を育成すべきという方針が打ち出されている。

本論文では、いわゆる理系だけでなく文系も含めた高等教育全般での活用を企図し、現代のAIの中心技術である機械学習に関する動かしながら学べる学習教材（実装例）を表計算ソフトを活用して作成する方法を検討する。プログラミング言語ではなく、表計算ソフトを活用する理由は以下のとおりである。

- 多くの学科で表計算ソフトの使い方をITリテラシー等の授業で最初に教えることが多く、利用ソフトウェアの使い方に関する講義の時間を減らすことができる。
- 一方、プログラミング言語を使うとなると、高等学校の「情報I」の授業でPython等を用いた授業が始まっているものの、カリキュラム全体のごく一部の時間を使って教えているのみであり [2], [3], [4], [5], 情報系や工学系以外の学科でも機械学習を学べる程度に改めて時間を割いて教える必要がある。
- 表計算ソフトでは複数の変数の値の変化を同時に確認

しやすく、その点では、複数のパラメータを持つ機械学習モデルと相性がよい [6]。

- AIのリテラシーレベルの教育では「データを読む、扱う、説明する」ことから教えることが一般的で [7], そのためにやはり表計算ソフトを用いる可能性が高く、その流れからも利用しやすい。

機械学習によって作成されるモデルを使った計算処理は学習時と推論時に分けられるが、本論文では、習得が比較的困難な学習時の計算やアルゴリズムに関する教材の実現方法に焦点をあてる。具体的には、従来までの試みと異なり、機械学習モデルの学習過程をブラックボックス化せず、インタラクティブかつ少ないセル数でノートPCのディスプレイ等でも一覧性高く実装しやすい汎用的な方法を提案し、あわせて学習過程にグラフ等の可視化機能を連動させることと、表計算ソフトを使うからこそその注意点として計算処理の流れが把握しやすいよう配慮して実装すべきことを提案し、いくつかの実装例を示す。

### 2. 関連研究

機械学習の基本である単回帰や重回帰に関してであれば、表計算ソフト Excel には、複数のデータから回帰式を計算する関数が最初から用意されている。また、散布図描画機能等にも回帰式を求める機能が備わっている。これらに関しては、オンラインドキュメントでどのような計算が行われているかが示唆されているものの [8], 実際の処理方法を学習者が確認することはできない。

<sup>1</sup> 神奈川工科大学 情報教育研究センター  
Kanagawa Institute of Technology, 1030 Shimo-ogino, Atsugi, Kanagawa, 243-0292, Japan  
<sup>a)</sup> maeda@cco.kanagawa-it.ac.jp

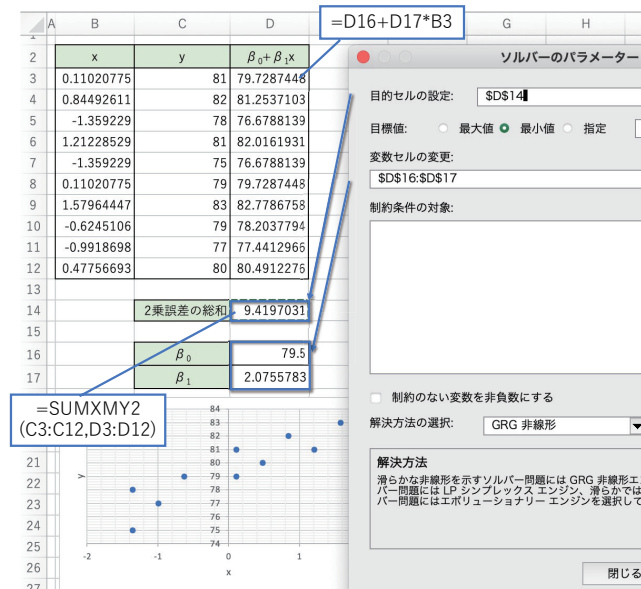


図 1 単回帰にソルバーを利用する例

Fig. 1 A simple linear regression example using Solver.

文献 [6] は、代表的な機械学習モデルであるニューラルネットワークの動作モデルの教材を表計算ソフトで作成した初期の例である。しかし、これは推論時の動作のみを対象としており、学習時は対象としていない。

Excel で機械学習モデルの学習時の計算処理を行う汎用的な方法としてアドインの一つであるソルバーを使う方法がある。これを利用した例として文献 [9], [10], [11] がある。単回帰を題材にこのソルバーの使用例を示す (図 1)。ここでは、2つの量  $x$  と  $y$  からなる観測データのセットが与えられたときに、 $x$  の値から  $y$  の値を予測するために次の単回帰モデルを利用する。

$$\hat{y} = \beta_0 + \beta_1 x \quad (1)$$

$\hat{y}$  は  $y$  の予測値、 $\beta_0$  と  $\beta_1$  は回帰直線の切片と傾きである。実際のデータセットには文献 [12] の表 1.1 の数値例を  $x$  のみ標準化して使っている。このアドインには数理最適化問題を解くためのいくつかの機能が備わっており、Excel のシート上で数式や変数を用いて目的関数 (損失関数) を定義したあと (同図ではセルの D14)、その値を最小化する等の設定をソルバーに対して行い、解を求めることができる。この方法では、学習のために損失関数を定義しなければならないことまでは実装例で示せるが、その後の計算についてはアドインの内部機能に任せることになり、やはり実際の処理を学習者が確認できなくなってしまう。

一方、機械学習の処理の繰り返し毎にシート上のセルを使用して実装する方法も提案されている [11], [13]。例えば文献 [11] ではニューラルネットワークを題材としているが、本論文では他のアプローチと比較しやすいよう、図 1 と同じデータと回帰モデルによる例を示す (図 2)。実装の詳細は説明しないが、このアプローチでは、 $n$  回目の繰

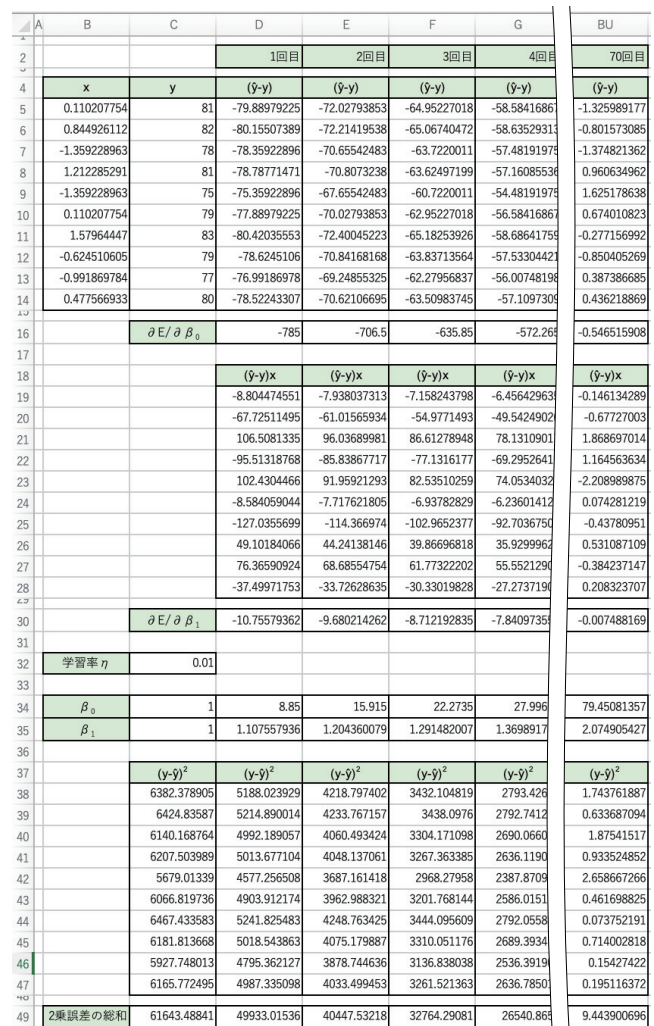


図 2 繰り返し処理毎にセルを使用して単回帰に勾配降下法を適用した例

Fig. 2 Simple linear regression with gradient descent using cells for each learning iteration.

り返しで得られたパラメータ  $\beta_0$  と  $\beta_1$  の値を新しいセルに書き込み、そのセルを参照して  $n + 1$  回目の繰り返しを行う (セルの 34, 35 行目)。しかし、同図が示すとおり、学習回数が減らせる標準化されたこのデータセットの場合でさえ 70 列ほどのセルが必要になっている。そのため一覧性が低く、初学者にとっては理解しにくいものになってしまう可能性がある。

その他、これまでの表計算ソフトを活用した機械学習の教材では学習過程におけるグラフ機能との連動や表計算シート上での計算式の適切な配置方法について述べている例は少ないが、本論文ではこれらについても言及する。

### 3. 表計算ソフト上で教材として実装するためのポイント

#### 3.1 繰り返し処理の実現方法

機械学習アルゴリズムの表計算ソフト上での従来の実装

が処理の繰り返し毎にセルを消費して実現されていたのは、パラメータの値を更新を繰り返すという、通常のプログラミング言語であれば当然可能な処理を実現することが簡単ではないからである。例えば、あるセルに格納されているパラメータ  $\beta$  の値を使って学習のための計算を行い、新しく得られた  $\beta$  の値の出力先として再び  $\beta$  のセルを指定したとすると、通常、Excel では循環参照エラーが生じてしまう。

そこで本論文では、学習によって得られる新しいパラメータの値を一旦他のセルに出力したあと、それを手動でコピーし、元のパラメータが格納されていたセルに通常のペーストではなく、「値」としてペーストする方法を提案する。これにより循環参照は避けることができる。ただし、この一連の操作を繰り返すのは大変面倒である。そこで、一度値としてペーストしたあとは、Excel の自動繰り返しのショートカットである F4 キーを押し続けることによって直前のコピー&ペーストを自動で反復して学習時の繰り返し処理を実現することとする。これにより、学習の過程を任意に進めたり止めたりして、インタラクティブに学習の過程を観察することも可能になる。

### 3.2 可視化機能との連動

上記のように学習の進行を自由に制御できる方法を採用するからこそ、表計算ソフトの可視化機能を連動させて学習中の状態を可視化できるようにすることを提案する。通常、表計算ソフトには、グラフ等によるデータの可視化機能が備わっている。学習時の途中経過をうまく可視化できれば、その処理内容を正しく理解するのに役立つはずである。ただし、学習過程と連動して何を可視化すべきかは機械学習のモデルや手法毎に異なるため、具体的事例で解説する。

### 3.3 処理の順序の明示化

どこから実行されるかが明確な通常のプログラミング言語と違い、表計算ソフトで複数の計算式や関数を用いた場合、どこから処理が始まっているかがわかりにくい。そのため、処理の工程が複雑になる場合には、一連の処理を適切なブロックに分け（セル上に分けて配置し）、それらの処理の順序を番号や矢印等で明示することを提案する。

## 4. 教師あり学習の例 1：単回帰

### 4.1 勾配降下法と繰り返し処理の実現方法

教師あり学習の表計算ソフト上での実装例として、単回帰に勾配降下法を適用する例を紹介する。単回帰の回帰係数は、一般的な統計学の教科書では勾配降下法を使わず正規方程式を解くか、それから導かれる回帰係数を算出するための公式を使って求められることが多い [12], [14]。しかし、ここでは、単回帰より複雑な機械学習モデルである

ニューラルネットワークをどのように学習させるのかを学んでいくための前準備として、勾配降下法を単回帰で学ぶケースを想定する。対象とするデータは図 1 や図 2 と同一で、回帰モデルも式 (1) である。

初めに損失関数を定義する。ここでは、 $N$  組の観測データ  $(x_i, y_i), i = 1, \dots, N$  から回帰直線を求めるために  $y$  と  $\hat{y}$  の二乗誤差を最小化する。すなわち、以下の式を損失関数とする。

$$E = \frac{1}{2} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2)$$

右辺先頭の  $1/2$  は、このあと求める  $E$  の偏導関数をシンプルにするためのものである。これを勾配降下法を使って最小化するために、まずは  $E$  を  $\beta_0$  と  $\beta_1$  でそれぞれ偏微分し、それらに十分小さな値の学習率  $\eta$  をかけたものを用いて、 $\beta_0$  と  $\beta_1$  の更新式をそれぞれ以下のように定義する。

$$\beta_0 := \beta_0 - \eta \frac{\partial E}{\partial \beta_0} \quad (3)$$

$$\beta_1 := \beta_1 - \eta \frac{\partial E}{\partial \beta_1} \quad (4)$$

あとは  $\beta_0$  と  $\beta_1$  に何らかの初期値を設定し、更新式 (3) と (4) に基づいて  $\beta_0$  と  $\beta_1$  の値の更新を繰り返す。 $\beta_0$  と  $\beta_1$  の現在値で  $\partial E / \partial \beta_0$  と  $\partial E / \partial \beta_1$  を求めたときに、どちらもほぼ 0 になれば終了である。 $E$  の  $\beta_0$  と  $\beta_1$  それぞれについての偏微分は以下となる。

$$\begin{aligned} \frac{\partial E}{\partial \beta_0} &= \frac{\partial}{\partial \beta_0} \left( \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \right) \\ &= \frac{1}{2} \sum_{i=1}^N \left( \frac{\partial}{\partial \beta_0} (y_i - \hat{y}_i)^2 \right) \\ &= \frac{1}{2} \sum_{i=1}^N \left( 2(y_i - \hat{y}_i) \frac{\partial}{\partial \beta_0} (y_i - (\beta_0 + \beta_1 x_i)) \right) \\ &= \sum_{i=1}^N (\hat{y}_i - y_i) \end{aligned} \quad (5)$$

$$\begin{aligned} \frac{\partial E}{\partial \beta_1} &= \frac{\partial}{\partial \beta_1} \left( \frac{1}{2} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \right) \\ &= \frac{1}{2} \sum_{i=1}^N \left( \frac{\partial}{\partial \beta_1} (y_i - \hat{y}_i)^2 \right) \\ &= \frac{1}{2} \sum_{i=1}^N \left( 2(y_i - \hat{y}_i) \frac{\partial}{\partial \beta_1} (y_i - (\beta_0 + \beta_1 x_i)) \right) \\ &= \sum_{i=1}^N (\hat{y}_i - y_i) x_i \end{aligned} \quad (6)$$

これらの式をもとにした Excel での実装例とその操作方法を図 3 に示す。 $\beta_0$  と  $\beta_1$  の初期値は最初はセルの C3 と C4 に、データセットの  $x$  は E3 から E12、 $y$  は F3 から F12 に格納されている。H3 から H15 には偏導関数 (5) による

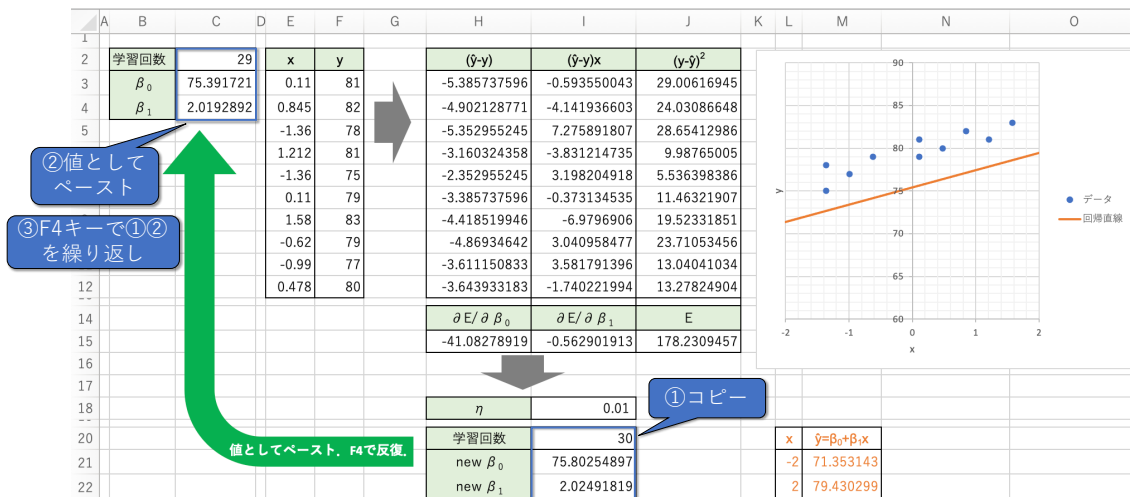


図 3 提案実装方法による勾配降下法を用いた単回帰とその操作方法  
Fig. 3 Proposed implementation of gradient descent for simple linear regression and the operation flow.

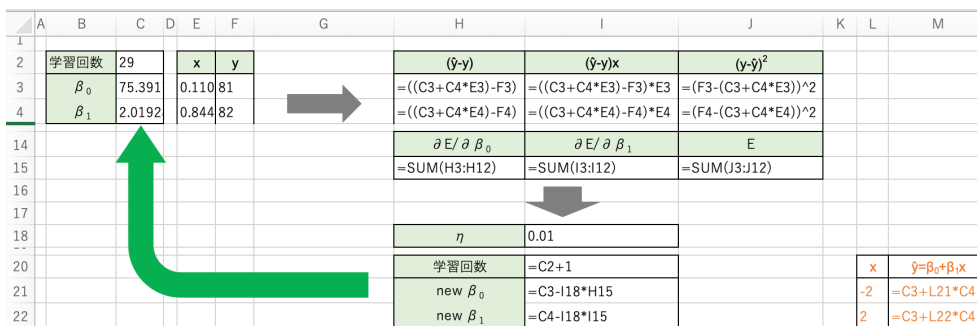


図 4 勾配降下法を用いた単回帰の実装における数式表示  
Fig. 4 Formulas in the proposed implementation of gradient descent for simple linear regression.

計算, I3 から I15 には偏導関数 (6) による計算, I21 と I22 には更新式 (3) と (4) それぞれの右辺の計算, そして J3 から J15 には損失関数 (2) による計算が実装されている (ただし右辺の 1/2 は省略している). Excel の具体的な計算式を図 4 に示す. なお, 5 行目から 12 行目のセルに入力されている計算式はそれより上の行と同じパターンであるため非表示にしている.

勾配降下法による繰り返し処理の 1 回目に関しては, 以下のように行う. まず, I20 から I22 を手動で選択してコピーし, C2 から C4 に値としてペーストする. I20 には「=C2+1」という計算式が格納されており, 学習回数がかかるようになっていく. このあと, 2 回目以降の繰り返し処理に関しては, キーボードの F4 キーを押下することで行う. 実際には,  $\frac{\partial E}{\partial \beta_0}$  と  $\frac{\partial E}{\partial \beta_1}$  がほぼ 0 になるまで, これらの値が格納されている H15 と I15 を観察しながら F4 キーを押し続けられればよい. 例えば  $\beta_0$  と  $\beta_1$  の初期値がどちらも 1 であれば, F4 キーを数秒押し続けるだけでよい. 同時に J15 の損失関数 E の値が小さくなることも確認できる.

#### 4.2 可視化機能との連動

本実装例では, 学習を繰り返すにつれ, データの散布図と回帰直線の関係も更新され, 適切な傾きと切片を持つ回帰直線に変化していく様子を図 3 の右側のグラフで観察することができる. これを実現するため, 基本的には Excel の散布図の機能を利用する. E3 から F12 にある  $x$  と  $y$  に関しては, 通常の方法で「系列」を作成してプロットする. 一方, 回帰直線に関しては, 勾配降下法の繰り返し処理を行うことで得られる最新の  $\beta_0$  と  $\beta_1$  の値を用いて, ある  $x$  のときの  $\hat{y}$  を式 (1) をもとに数パターン求めてセルに格納しておき (L21 から M22), 散布図ではこれに対する系列を用意し, ただしその描画方法については「データ系列の書式設定」から「マーカーなし」および「線 (単色)」を選択して直線として描画するようにする. 本実装例では,  $x$  が -2 と 2 のときの 2 点のみ計算しているが, 曲線で回帰する場合にはより多くの値の点を生成して描画すればよい.

#### 4.3 処理の順序の明示化

この実装例では, 勾配降下法の処理の流れを明示化する

	A	B	C	D	E	F	G	H	I	J
1										
2								$\Delta$	0.001	
3										
4		学習回数	17		x	y		$(\beta_0 + \Delta) + \beta_1 x$	$\beta_0 + (\beta_1 + \Delta)x$	$\beta_0 + \beta_1 x$
5		$\beta_0$	77.2899355		0.110481			$=(C5+I2)+C6*E5$	$=C5+(C6++I2)*E5$	$=C5+C6*E5$
6		$\beta_1$	2.04481856		0.844482			$=(C5+I2)+C6*E6$	$=C5+(C6++I2)*E6$	$=C5+C6*E6$
7										
8								$E(\beta_0 + \Delta, \beta_1)$	$E(\beta_0, \beta_1 + \Delta)$	$E(\beta_0, \beta_1)$
9								$=SUMXMY2(F5:F14,H5:H14)$	$=SUMXMY2(F5:F14,I5:I14)$	$=SUMXMY2(F5:F14,J5:J14)$
10										
11								$\frac{\partial E}{\partial \beta_0}$	$\frac{\partial E}{\partial \beta_1}$	
12								$=(H17-J17)/I2$	$=(I17-J17)/I2$	
13										
14								$\eta$	0.01	
15										
16								学習回数	$=C4+1$	
17								new $\beta_0$	$=C5-123*H20$	
18								new $\beta_1$	$=C6-123*I20$	
19										
20										
21										
22										
23										
24										
25										
26										
27										

図 5 数値微分を利用した勾配降下法による単回帰の数式表示

Fig. 5 Formulas in simple linear regression using gradient descent with numerical differentiation.

ために、パラメータ  $\beta_0, \beta_1$  とデータ  $x, y$  を格納したブロック、パラメータの現在値とデータを用いて  $\partial E/\partial \beta_0, \partial E/\partial \beta_1$ , 及び  $E$  を計算するブロック、 $\beta_0$  と  $\beta_1$  の新しい値を求めるブロックの3つを分離してセル上に配置している。そして、処理の流れがわかるようにそれらの間に矢印を配置している。今回は操作方法の説明のために番号を使用しており、それと混同する可能性があるため使わなかったが、番号で流れを示してもよい。

#### 4.4 数値微分版

式 (5) と (6) のような解析的な微分を理解することが困難な学習者向けには、数値微分を使ってもほぼ同じセル数で実装できる。以下の前方差分に基づく実装例の数式表示モードのみを図 5 に示す。

$$\frac{\partial E}{\partial \beta_0}(\beta_0, \beta_1) \approx \frac{E(\beta_0 + \Delta, \beta_1) - E(\beta_0, \beta_1)}{\Delta} \quad (7)$$

$$\frac{\partial E}{\partial \beta_1}(\beta_0, \beta_1) \approx \frac{E(\beta_0, \beta_1 + \Delta) - E(\beta_0, \beta_1)}{\Delta} \quad (8)$$

ここで  $\Delta$  は十分小さい正の数である。

### 5. 教師あり学習の例 2：ニューラルネットワーク

#### 5.1 誤差逆伝播法と繰り返し処理の実現方法

教師あり学習のもう一つの例として、ニューラルネットワークの実装例を紹介する。前述の単回帰では数値微分による勾配降下法の実装例も示したが、ニューラルネットワークでもパラメータの分だけ数値微分すれば同様の実装は可能である。しかしここではニューラルネットワークの学習方法として一般的な誤差逆伝播法の実装例を紹介する。図 6 にこの実装例とその操作方法を示すが、下方にモデルが図示されているように、入力層に  $x_1$  と  $x_2$ , 隠れ層は 1 層でノードが 2 つ, 出力層がノード 1 つの全結合のニューラルネットワークとなっている。活性化関数はすべてシグ

モイド関数である。入力データ  $x_1$  と  $x_2$  はセルの G4 から H7, 教師データ  $y$  は I4 から I7 に格納されており、これらには AND や XOR など論理演算の真理値表が入力されることを想定している。 $w_1, w_2, b_1, w_{11}, w_{12}, b_2, w_{21}, w_{22}, b_3$  がパラメータで E4 から E12 に格納されている。

誤差逆伝播法では学習の繰り返し毎にまず順伝播の計算が行われる。 $z_1, z_2$  がそれぞれ中間層の 1 番目と 2 番目のノードの出力で K4 から L7 に格納される。例えば  $z_1$  は以下のように求められる。

$$a_1 = w_{11}x_1 + w_{21}x_2 + b_1 \quad (9)$$

$$z_1 = \frac{1}{1 + \exp(-a_1)} \quad (10)$$

$\hat{y}$  がこのニューラルネットワークの最終的な出力で M4 から M7 に格納されており、計算式は  $z_1$  と同様である。 $\hat{y}$  は連続値であるが、0.5 以上なら 1, そうでなければ 0 の 2 値に N4 から N7 で変換している。損失関数は単回帰の実装例と同じく式 (2) で、M10 で計算している。ここまでで言及したセルの Excel の計算式も上記のとおりである。

誤差逆伝播法も結局は勾配降下法であり、「モデルが合成関数の形をとっているため、誤差関数の勾配は、微分における連鎖法則を用いて、簡単に導き出すことができ [15]」、計算効率も向上させられるという以外、図 3 の勾配降下法の実装と本質的な違いはない。P3 から X10 で、順伝播の計算結果を利用して逆伝播を計算している。例えば、 $E$  の  $w_1$  についての偏微分は  $\partial E/\partial w_1 = \sum_{i=0}^N (\hat{y}_i - y_i) \hat{y}_i (1 - \hat{y}_i) z_{1i}$  となる。P3 にはこの総和される項  $(\hat{y}_i - y_i) \hat{y}_i (1 - \hat{y}_i) z_{1i}$  が明記されており、P4 から P7 ではこの項の実際的な計算を行っている。P10 では P4 から P7 の総和を計算している。学習を進める方法も前述の実装例と同様で、X15 から X24 を E3 から E12 に値としてペーストし、その後は F4 キーで繰り返せばよい。

#### 5.2 可視化機能との連動

本実装例では、グラフは使わず、各パラメータがニューラルネットワークのどの位置のものかを可視化するだけに

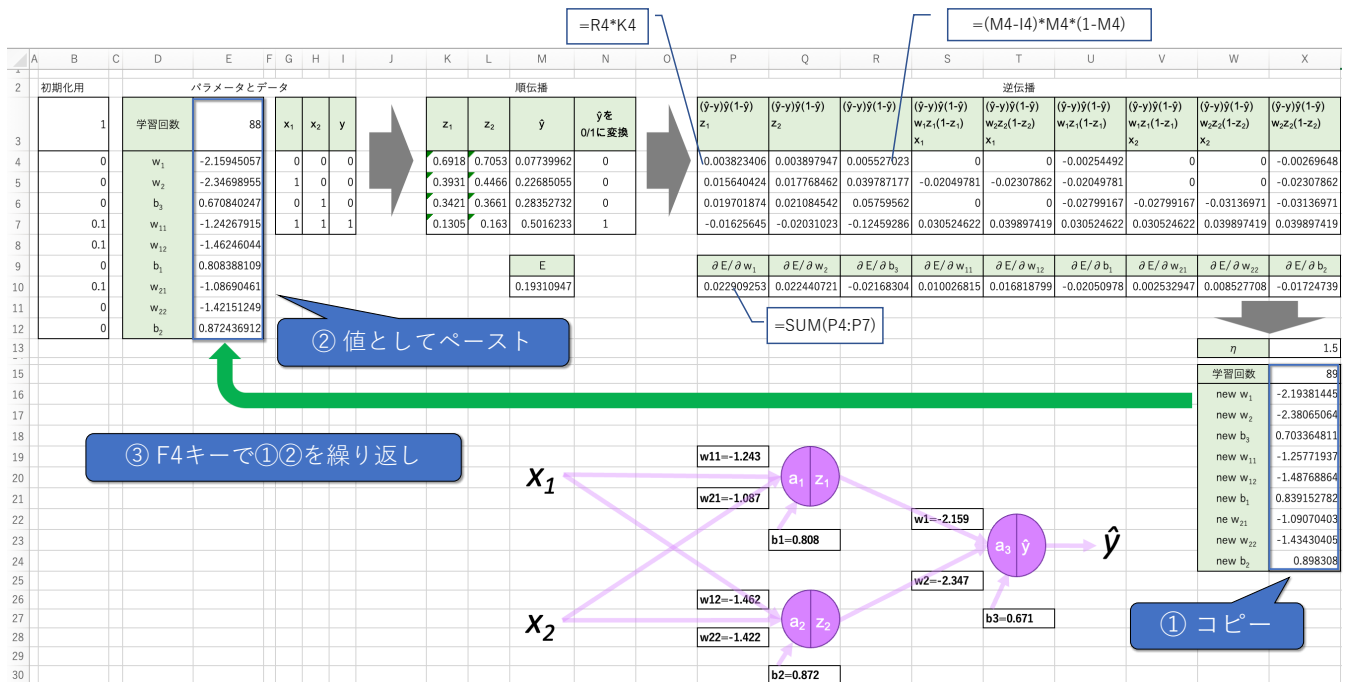


図 6 誤差逆伝播法を適用したニューラルネットワークの実装例とその操作方法  
Fig. 6 Proposed implementation of a neural network with backpropagation and the operation flow.

とどめている．具体的には，M18 から U30 付近にニューラルネットワークの図を円や矢印を配置して作成し，E3 から E12 にあるパラメータ同士の位置関係がわかりやすくなるように図中で参照している．これにより，F4 キーを押している間に，損失関数やパラメータの値の変化の速度が速くなったり遅くなったりする様子を見て，損失関数が凸関数ではないことを想像してもらうことや，ニューラルネットワークの前方の層と後方の層でパラメータが変化速度に差があることやそれに影響を与えている勾配の大きさの違い等を観察し，勾配消失問題を実感してもらうことを想定している．

### 5.3 処理の順序の明示化

データとパラメータのブロック，順伝播を計算するブロック，逆伝播を計算するブロック，新しいパラメータの値を得るブロックに分け，それらを矢印で繋いで処理の流れを明示化している．

## 6. 教師なし学習の例：K-means クラスタリング

### 6.1 アルゴリズムと繰り返し処理の実現方法

最後に教師なし学習の例として，K-means クラスタリングの実装例を紹介する．この手法は「 $d$  次元の  $N$  個のデータ  $D = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ,  $\mathbf{x}_i \in \mathbb{R}^d$  を，データ間の類似性（距離）を尺度に，あらかじめ定められた  $K$  個のクラスタに分類する [16]」手法である．まず，この手法のアルゴリズム

を文献 [16] の解説を引用し，以下に示す．

### K-means クラスタリングのアルゴリズム

**初期化：**  $N$  個のデータをランダムに  $K$  個のクラスタに振り分け，それぞれのクラスタの平均ベクトル（重心，セントロイド）を求め， $\mu_k (k = 1, \dots, K)$  とする．

(1)  $q_{ik}$  に関する最適化:  $\mu_k$  を固定し，帰属変数  $q_{ik}$  (筆者注:  $i = 1, \dots, N, k = 1, \dots, K$ ) を，

$$q_{ik} = \begin{cases} 1 & (k = \arg \min_j \|\mathbf{x}_i - \mu_j\|^2 \text{ の場合}) \\ 0 & (\text{それ以外の場合}) \end{cases}$$

に従って決める．

(2)  $\mu_k$  の最適化:  $q_{ik}$  を固定し，以下の式に従ってセントロイド  $\mu_k$  を決める．

$$\mu_k = \frac{\sum_{i=1}^N q_{ik} \mathbf{x}_i}{\sum_{i=1}^N q_{ik}}$$

(3) 繰り返し: 上記 (1) と (2) を収束するまで（状態変化がなくなるまで）繰り返す．

このようなアルゴリズムに基づく実装例とその操作方法を図 7 に示す．この例では，クラスタ数は 3 となっており，C7 から D9 にクラスタの平均ベクトル  $\mu_1, \mu_2, \mu_3$  の値が格納されている．データは 2 次元のデータが 10 個となっており，G4 から H13 に格納されている．上記アルゴリズムの「初期化」と異なり，任意の初期値を試しやすいよう，それぞれの平均ベクトルの初期値を直接設定できる

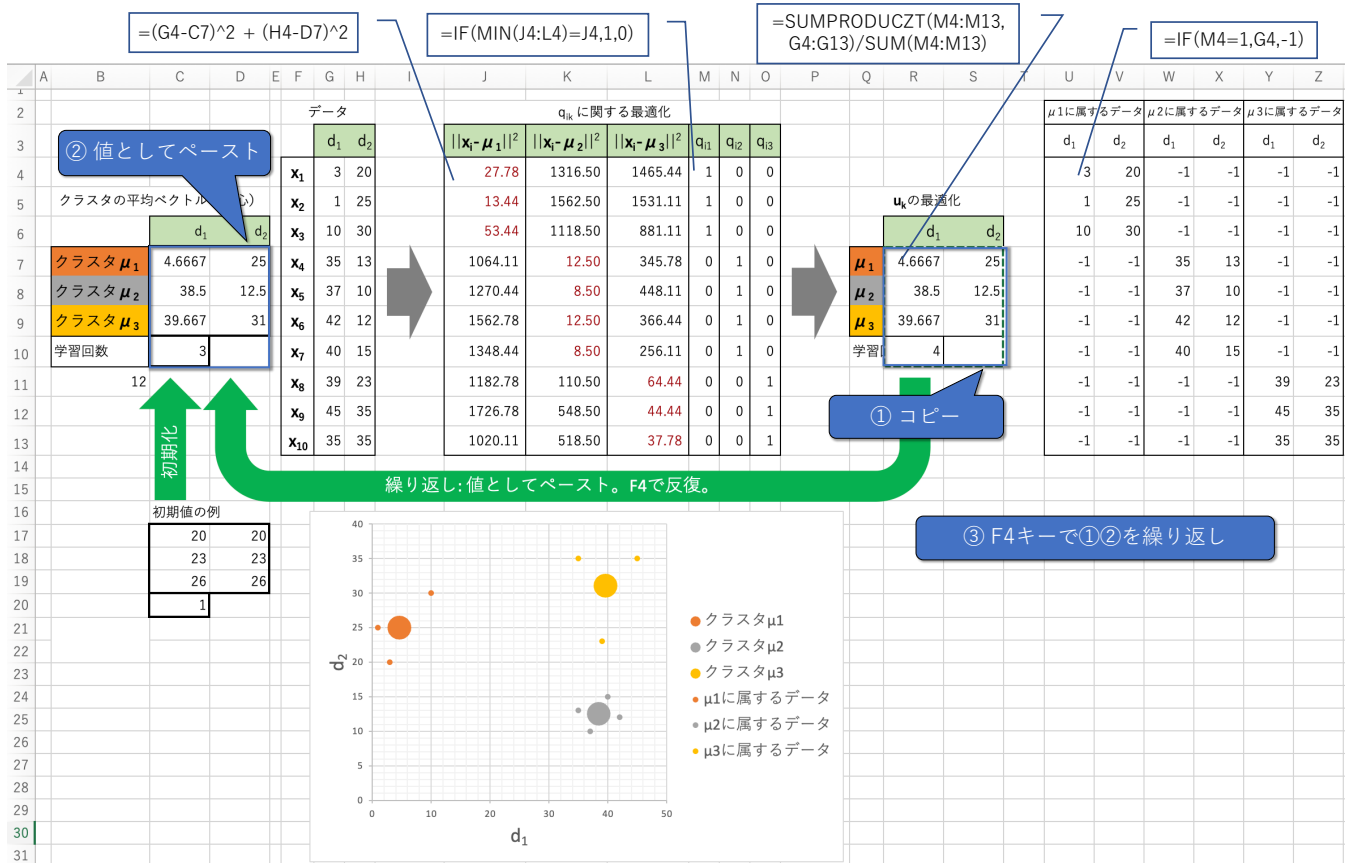


図 7 K-means クラスタリングの実装例とその操作方法  
Fig. 7 Proposed implementation of K-means clustering and the operation flow.

ようにしている。また、初期値を迅速に設定して動作確認できるように、C17 から D19 に初期値の一例を格納しておき、これをすぐにコピーできるようにしている。

アルゴリズムの「(1)  $q_{ik}$  に関する最適化」に関しては、まず、 $i = 1, \dots, N, k = 1, \dots, K$  における  $\|x_i - \mu_k\|^2$  を J4 から L13 で計算している。例えば  $\|x_1 - \mu_1\|^2$  は J4 で計算されており、Excel での計算式は「 $= (G4-C7)^2 + (H4-D7)^2$ 」となる。J4 から L13 の行毎にこの後最小値を求めることになるため、その値を自動的に色付けする処理も Excel の「条件付き書式」の機能で行っている。次に、これらの計算結果をもとに、M4 から O13 で帰属変数  $q_{ik}$  を求めている。例えば  $q_{11}$  は M4 で計算されており、Excel での計算式は「 $= IF(MIN(J4:L4)=J4, 1, 0)$ 」となる。

アルゴリズムの「(2)  $\mu_k$  の最適化」は、R7 から S9 で実行している。例えば  $\mu_1$  の 1 番目の要素は R7 で計算されており、Excel での計算式は「 $= SUMPRODUCT(M4:M13, G4:G13) / SUM(M4:M13)$ 」となる。

アルゴリズムの「(3) 繰り返し」に関しては、これまで紹介した実装例と同様の操作で行う。すなわち、R7 から S10 を C7 から D10 に値としてペーストする。その後はキーボードの F4 キーで、この一連の操作を繰り返す。操作者は、 $u_k$  の値が変化しなくなったときに操作をやめればよ

いが、本実装例では C17 から D20 の値を初期値とする場合は学習を 3 回繰り返すだけでよい。

## 6.2 可視化機能との連動

K-means クラスタリングのアルゴリズムでは、処理を繰り返す毎に、クラスタの平均ベクトル  $\mu_k$  が変化するとともに、データ  $x_i$  がどのクラスタに属するかも変化していく。この様子を繰り返し処理を進めながら観察できれば、K-means クラスタリングへの理解が深まるはずである。本実施例では、あるクラスタの平均ベクトル（重心）とそれに属するデータを同じ色で散布図として可視化している。ここでは、クラスタの平均ベクトル  $\mu_k$  毎に「系列」を変えることで散布図上で色を変えておき（これらの参照セルの範囲は C7 から D9）、それぞれのクラスタに属するデータ毎にやはり系列を用意し、属するクラスタと同じ色を設定している（これらの参照セルの範囲は U4 から Z13）。そして、U4 から Z13 の範囲に該当するデータをコピーすることで、各データの点の色を変更している。

なお、このようにデータの色を散布図上で動的に変更する方法については、K-means クラスタリングの理解という観点からは重要ではなく、学習者が深く理解する必要はない。従って、K-means クラスタリングのアルゴリズムを

実現しているセルの領域付近には散布図だけを配置しておき、色を変更している部分（本実装例ではU列からZ列）は場合によっては画面内に収まらなくてもよいと考える。

### 6.3 処理の順序の明示化

K-means クラスタリングの実装例でも、一連の処理の流れを把握しやすくするため、データや処理結果を単に羅列するのではなく、先述のアルゴリズムの説明に対応させて処理をブロックに分け、その間に矢印を配置し、各ブロックの処理内容の説明もブロックの上部に明記している。

## 7. 評価

本提案手法に基づいて機械学習の実装例を作成した場合、1)F4 キーを押下している間にどの程度の回数の学習を完了できるかと、2)一般的なサイズのディスプレイで、パラメータ、学習アルゴリズムの実現部、グラフ等による可視化部が一度に表示できるかを評価した。

Apple MacBook Air M1 2020 (CPU: Apple M1, メモリ: 8GB, OS: Monterey, ディスプレイ: 13.3 インチ, 解像度: 2,560 × 1,600 ピクセル), Microsoft Excel for Mac (version 16.67) にて、図 3 の単回帰、図 6 のニューラルネットワーク、図 7 の K-means クラスタリングの実装で、F4 キーを押下したあとに学習回数が 100 回分完了するまでの時間を計測した。このときキーボードの「キーのリピート」は最も速い設定で、「リピート入力認識までの時間」は 2 番目に短い、著者が普段から使っている設定であった。この計測は、PC の画面を録画し、その動画を動画編集ソフト Adobe Premiere Pro で 1 フレーム単位で確認することで行った。結果、単回帰の場合で平均 3.7 秒 ( $N = 10$ , 不偏分散 = 0.018), ニューラルネットワークの場合で平均 5.65 秒 ( $N = 10$ , 不偏分散 = 0.048), K-means クラスタリングの場合で平均 3.7 秒 ( $N = 10$ , 不偏分散 = 0.028) となった。

一方、一覧性に関しては、図 3 の単回帰はセルの一番左上から O22 まで、図 6 のニューラルネットワークは X30 まで、図 7 の K-means クラスタリングは Z31 までを使用しており、全ての実装例で、パラメータ格納部、学習アルゴリズムの実現部、グラフ等による可視化部の全てが、Excel の「リボン」を非表示にした状態で画面内に収まることを確認した。

## 8. 考察

学習者に F4 キーを押下し続けさせられる時間は、通常は数十秒程度までであろう。このことから、2 千回以下くらいまでの学習回数で所望の結果が得られるような事例であれば本提案手法は有効であると考えられる。そのため、この程度の学習回数で有効な結果が得られるように、機械学習におけるデータの前処理として一般的に実施されているこ

とだが学習データを標準化して必要な学習回数を減らす必要が多くなるであろう。とはいえ、勾配降下法や誤差逆伝播法などの仕組みを教えるための最小限の実装例であれば、本論文で紹介したように問題なく作成できるであろう。

一覧性に関しては、どの実装例においても、ノート PC の画面内に収めることができたため、個人のノート PC で演習を受けることが多い昨今の大学生の環境でも利用しやすいものになると思われる。単回帰からニューラルネットワークに講義を進める前に重回帰等の説明を挟みたい場合には、単回帰の実装に対して説明変数が追加される毎に、そのデータを格納する列とそれに対応するパラメータで損失関数を偏微分するための列を追加すればよく、画面内に収めることは十分可能であろう。

今後は、本論文で紹介した以外の機械学習手法に対しても本提案手法を問題なく適用できるかを検討するとともに、実装例を実際の授業で使用して学習者からのフィードバックを得ることを計画している。

## 参考文献

- [1] AI 戦略 2019 と数理・データサイエンス・AI 教育プログラム認定制度について, [https://www.mext.go.jp/content/20210315-mxt\\_senmon01-000012801-1.pdf](https://www.mext.go.jp/content/20210315-mxt_senmon01-000012801-1.pdf)
- [2] 坂村健ほか: 高等学校 情報 I, 数研出版, 2022.
- [3] 黒上晴夫, 堀田龍也, 村井純ほか: 情報 I, 日本文教出版, 2022.
- [4] 赤堀侃司, 東原義訓, 坂元章ほか: 新編 情報 I, 東京書籍, 2022.
- [5] 本郷健, 松原伸一ほか: 実践 情報 I, 開隆堂, 2022.
- [6] Thomas T. Hewett.: Teaching Students to Model Neural Circuits and Neural Networks Using an Electronic Spreadsheet Simulator. Behavior Research Methods, Instruments, & Computers. 17(2), pp. 339–344 (1985).
- [7] 数理・データサイエンス・AI 教育プログラム認定制度 (リテラシーレベル), [https://www.mext.go.jp/a\\_menu/koutou/suuri.datascience.ai/00002.htm](https://www.mext.go.jp/a_menu/koutou/suuri.datascience.ai/00002.htm)
- [8] <https://support.microsoft.com/ja-jp/office/slope-%E9%96%A2%E6%95%B0-11fb8f97-3117-4813-98aa-61d7e01276b9>
- [9] Hegazy, T., Ayed, A.: Neural Network Model for Parametric Cost Estimation of Highway Projects. Journal of Construction Engineering and Management. 124(3), pp. 210–218 (1998).
- [10] 涌井良幸, 涌井貞美: Excel でわかるディープラーニング 超入門, 技術評論社, 2017.
- [11] 涌井良幸, 涌井貞美: Excel でわかる機械学習超入門, 技術評論社, 2019.
- [12] 佐和隆光: 回帰分析 (新装版), 朝倉書店, 2020.
- [13] Rienzo, T.F., Athappilly, K.K.: Introducing Artificial Neural Networks through a Spreadsheet Model. Decision Sciences Journal of Innovative Education. 10(4), pp. 515–520 (2012).
- [14] 東京大学教養学部統計学教室: 統計学入門, 東京大学出版会, 1991.
- [15] Trevor Hastie, et al.: 統計的学習の基礎, 共立出版, 2014.
- [16] 平井有三: はじめてのパターン認識, 森北出版, 2012.