

Excelでわかる ディープ ラーニング 超入門

AIのしくみを
やさしく理解できる!

涌井良幸 涌井貞美 著

初めてのAI学習に最適!

難しい数学計算はExcelに任せて
ディープラーニングのしくみを
動かしながら理解できる!

ご利用の前に必ずお読みください

本書は紙書籍『Excelでわかる ディープラーニング超入門』(ISBN978-4-7741-9474-5)を元に製作した電子書籍です。紙書籍とはデザインやレイアウトが異なり、ご覧になる端末により表示が異なる場合があります。表示設定は端末の標準設定を推奨します。配信後に補足訂正等でデータの再配布を行う場合があります。更新方法は購入先の電子書店のヘルプ等をご確認ください。

本書は情報提供のみを目的としており、掲載内容の運用結果について技術評論社および著者は一切の責任を負いません。掲載内容は特に断りのない限り執筆時点より以前の情報のため、変更される場合があります。特に、ソフトウェアはバージョンアップされる場合があり、本書での説明とは機能内容や画面図などが異なってしまうこともあります。

以上をあらかじめご承諾の上、ご利用をお願いします。

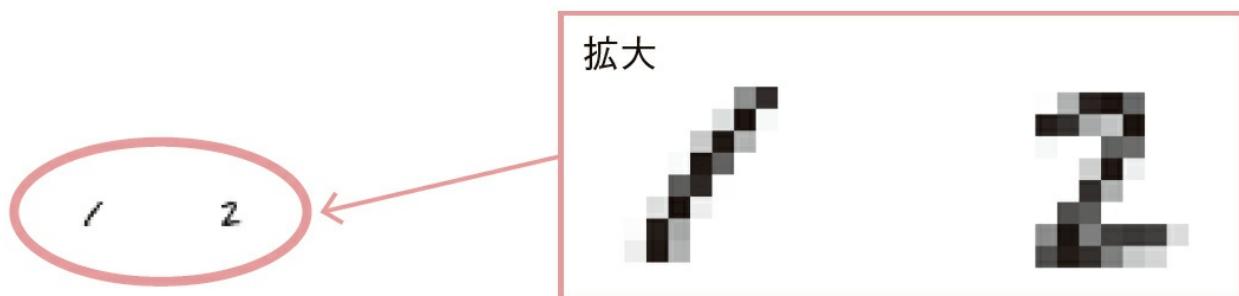
本文中に記載されている製品の名称は、すべて関係各組織、各社の商標または登録商標です。

はじめに

AI(人工知能)は毎日のようにマスコミの話題として取り上げられています。それほど社会へのインパクトが大きいテーマなのです。「AI、将棋で名人を破る」、「AIで株価予測」、「AIで健康管理」、「AIで自動運転」等々、枚挙に暇がありません。

このように話題の中心となるAIですが、その定義はあいまいです。昔から「AI炊飯器」などの家電も売られており、「何をいまさら」という意見もあります。しかし、今話題のAIはこれまでとは劇的に異なります。新しいアルゴリズムが開発されたのです。通称「ディープラーニング」とか「深層学習」と呼ばれるアルゴリズムです。

たとえば、次の2つの手書きの数字画像を見てください。



人には数字「1」、「2」と判別できますが、コンピューターにどのように区別させればよいでしょうか。

従来の20世紀型のアルゴリズムは機械に教え込むことが基本でした。

「1とはこんな数字の形であり、2とはそんな数字の形である」

ということを、事細かくコンピューターに教え込んだのです。

しかし、このアルゴリズムでは少し形が崩れた文字の判別は困難です。どんな悪筆に対しても対応できるように機械に教え込むのは無理なのです。ところが、ディープラーニングの手法を利用すると、それを実に簡単に実現できるのです。人間がどうにか判読できる文字でも、しっかり読んでくれます。しかも、驚くべきことに、人が事細かく教える必要はないのです。

本書はディープラーニングのしくみについて、Excelと対話しながら「わかり」「理解すること」を目的とした入門書です。基本的で具体的なテーマを追いながら、一步一步そのしくみを解き明かしていきます。

ディープラーニングについて多くの本が刊行されています。しかし、そのほとんどは使い方であったり、応用法であったりします。「なぜディープラーニングが形を区別できるのか」という根本的な問いに答える刊行物はほとんどありませんし、あっても難解です。これからディープラーニングがますます発展することが予想されますが、そのような時代にこそ、基本がしっかり見えていることが肝要です。本書はその基本を可視化することを目的としています。

ExcelはAI学習に最適といえます。ディープラーニングを支える「畳み込みニューラルネットワーク」は人工ニューロンから構成されていますが、面白いことにその一つひとつがExcelの一つひとつのセルに移し替えられるのです。すなわち、Excelのワークシートを眺めると、畳み込みニューラルネットワークの構造が容易に把握できるのです。本書はこのメリットを十分活用しながら、解説を進めます。

世界最強のプロ棋士のひとりを破った米グーグルのAI「アルファ碁」を開発した技術者デミス・ハサビス氏は次のように述べています。

「(AIの開発は)正しいはしごを登り始めた」

ハサビス氏が「正しいはしご」と呼ぶのは、まさにディープラーニングのことです。本書がその理解の一助になることを希望してやみません。

最後になりましたが、本書の企画から上梓まで一貫してご指導くださった技術評論社の渡邊悦司氏にこの場をお借りして感謝の意を表させていただきます。

2017年12月 著者

目 次

はじめに

本書の使い方

Excelサンプルファイルのダウンロードについて

1 章 初めてのディープラーニング

- §1. 置み込みニューラルネットワークのしくみは簡単
- §2. AIとディープラーニング

2 章 Excelの確認とその応用

- §1. 利用するExcel関数はたったの7個
- §2. Excelの参照形式
- §3. Excelソルバーの使い方
- §4. 回帰分析と最適化問題

3 章 ニューロンモデル

- §1. 神経細胞の働き
- §2. 神経細胞の働きを数式表現
- §3. 人工ニューロンと活性化関数
- §4. ステップ関数からシグモイド関数へ

4 章 ニューラルネットワークのしくみ

- §1. 読み物としてのニューラルネットワークのしくみ
- §2. ニューラルネットワークが手書き文字を識別
- §3. 訓練データの1つからニューラルネットワークの出力を算出
- §4. 正解と出力の誤差
- §5. ニューラルネットワークの目的関数
- §6. ニューラルネットワークの最適化
- §7. 最適化されたパラメーターを解釈
- §8. ニューラルネットワークをテストしよう
- §9. 現実の手書き文字にニューラルネットワークを応用

5 章 置み込みニューラルネットワークのしくみ

- §1. 読み物としての置み込みニューラルネットワークのしくみ
- §2. 置み込みニューラルネットワークが手書き数字を識別
- §3. 置み込みニューラルネットワークの入力層
- §4. 置み込みニューラルネットワークを特徴づける置み込み層
- §5. 置み込みニューラルネットワークのプーリング層
- §6. 置み込みニューラルネットワークの出力層
- §7. 正解と出力の誤差
- §8. 置み込みニューラルネットワークの目的関数
- §9. 置み込みニューラルネットワークの最適化
- §10. 最適化されたパラメーターを解釈

- §11. 置み込みニューラルネットワークをテストしよう
- §12. パラメーターに負を許容すると
- §13. 隠れ層の活性化関数を変更

付録

- A 訓練データ(1)
- B 訓練データ(2)
- C ソルバーのインストール法
- D パターンの類似度を数式表現

索引

本書の使い方

- ・本書はディープラーニングの基本となる畳み込みニューラルネットワークのしくみを、Excelを利用して理解することを目的とします。掲載のワークシートはExcel2013、2016で動作を確かめてあります。
- ・畳み込みニューラルネットワークのしくみがわかるることを目的としています。そこで、図を多用し、具体例で解説しています。そのため、厳密性に欠ける箇所があることはご容赦ください。
- ・ディープラーニングの世界はいろいろですが、本書は階層型ニューラルネットワークと畳み込みニューラルネットワーク(CNN)を画像認識に応用することを念頭に置いています。
- ・本書は「教師あり学習」のみを考えます。「教師なし学習」「強化学習」は入門編としては高度の内容になるからです。
- ・活性化関数はシグモイド関数を主に考えています。
- ・AIの文献の難読性のひとつは、文献による記号表現の不統一性にあります。本書では、Web上の文献で使われている最大公約数的な記号表現を採用しています。
- ・本書の理解にExcelの基本的な知識を前提としています。2章でそれを確認しているので、ご利用ください。

Excelサンプルファイルのダウンロードについて

本文中で使用するExcelのサンプルファイルをダウンロードすることができます。手順は次のとおりです。

- ①「<http://gihyo.jp/book/2018/978-4-7741-9474-5/support>」にアクセス
- ②「サンプルファイルのダウンロードは以下をクリックしてください」の下にある「excel_deeplearning_sample.zip」をクリック
- ③任意の場所に保存

■サンプルファイルの内容

項目名	ファイル名	概要
2章の内容をExcelで体験	2.xlsx	基本的な関数とソルバーの使い方を確認します
3章の内容をExcelで体験	3.xlsx	ニューロンの計算を体験
4章の内容をExcelで体験	4.xlsx	ニューラルネットワークのしくみを解説
5章の内容をExcelで体験	5.xlsx	畳み込みニューラルネットワークのしくみを解説

注意

- ・本書は、Excel2013、Excel2016で執筆しています。他のバージョンでの動作検証はしておりません。
- ・ダウンロードファイルの内容は、予告なく変更することがあります。
- ・ファイル内容の変更や改良は自由ですが、サポートは致しておりません。



1 章

初めての ディープラーニング

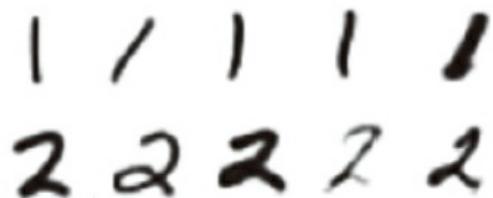
人工知能（AI）の分野で、近年大きな話題を集めているのがディープラーニング（深層学習）です。毎日のようにマスコミに取り上げられています。本章では、ディープラーニングの基本となるニューラルネットワークや畳み込みニューラルネットワークがどんなものか、また数学がどのように関与するかを鳥瞰してみましょう。

§1

畳み込みニューラルネットワークのしくみは簡単

近年、人工知能(AI)という言葉がマスコミを騒がせています。そのAIの実現手段の一つが「ディープラーニング」です。このディープラーニングとはどんなものかを見てみましょう。

次の図は数字1と2を表しています。我々人間にとって、すぐにそれとわかります。



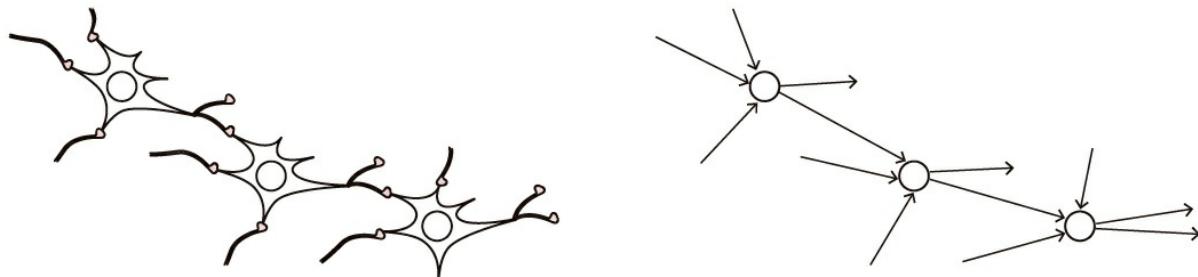
しかし、人が当たり前と思っている判断を、機械に行わせようと思うと、困難を極めます。大きさも筆跡も濃淡も、さまざまに変化に富んでいるからです。このような単純な問題でも、実用的な形の識別を自動的に実現しようとすると、我々は途方に暮れてしまうのです。

実際、20世紀までの論理では、文字や図形の識別に対処する理論作成は挫折の連続でした。その理由はコンピューターに教え込もうとする論理をとったからです。たとえば、手書きの数字「2」をコンピューターに認識させようとするとき、20世紀の論理は「『2』とはこのような特徴を持ったもの」と教え込もうとしたのです。



しかし、「教え込む」には現実はあまりに複雑です。先にも言及したように、字の形が余りにも多様すぎるからです。

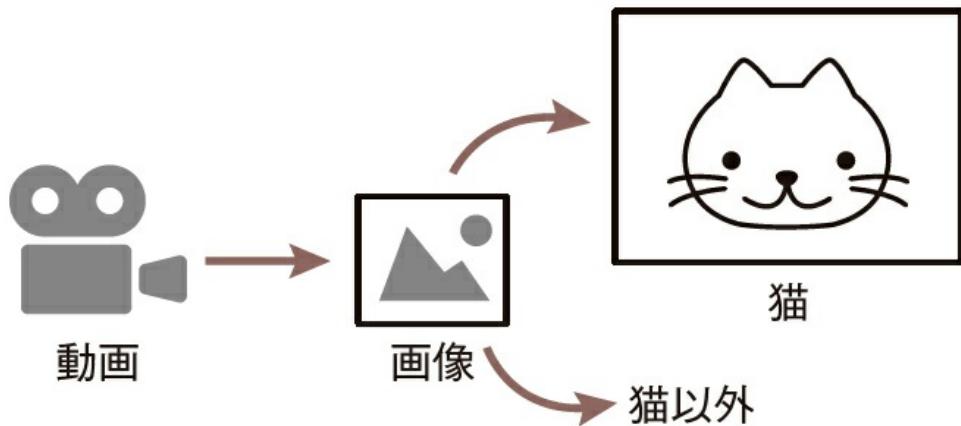
ところが、20世紀末の頃になって画期的な方法が開発されました。ニューラルネットワークと呼ばれるアルゴリズムです。動物の神経細胞を真似た人工ニューロンを積み重ね、ネットワークを作ります。そして、そのネットワークにたくさんの数字を読みませ、自ら学習させるのです。



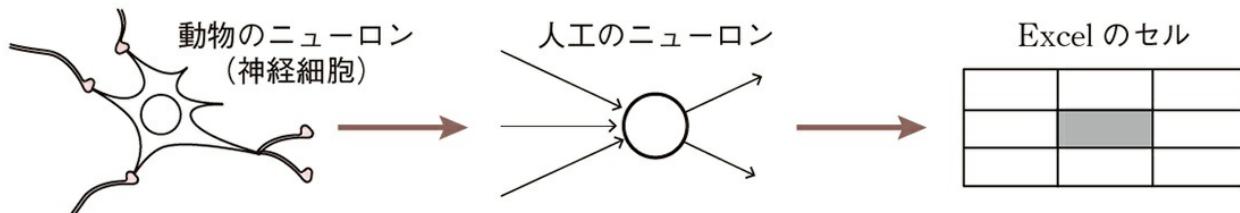
動物のニューロン（左）を真似た人工のニューロン（右）をネットワーク状につなげたのがニューラルネットワーク。

この方法は20世紀型のパターン認識のための論理に比べ、大きな成功を収めることになります。特に、ニューラルネットワークを多層に構造化した畳み込みニューラルネットワークと呼ばれる手法を用いると、人や猫ですらも写真や動画の中から認識できるようになったのです。ディープラーニングとはこのよくなじみで実現されたAI（人工知能）なのです。ちなみに、ディープラーニングは深層学習と直訳され

ます。



さて、「自ら学習」などというと難しく聞こえますが、ニューラルネットワークは数学的に大変簡単な理論です。面倒な計算部分をExcelに任せてしまうと、基本的なしくみは中学校の数学で理解できます。Excelの1つのセルが1つのニューロンに見立てられるからです。



Excelのセルは人工ニューロンの機能を果たす。

しきみが単純ということは大変ありがたいことです。これからは、人工知能が身近になることが予想されるからです。身近なもののしきみが見えなくては、使うのに危険があります。しきみがわかってこそ、正しい付き合い方が可能になるからです。

たとえば、マスコミ報道には次のような刺激的なタイトルが散見されます。

「人工知能で人は職を失う」

「人工知能が人間を支配する」

「人工知能が小説を書く」

しかし、しくみが理解されていれば、「それは、そんなもの」と客観的な判断が可能になります。無用な心配から解放され、刺激的なタイトルに踊らされることはなくなるのです。

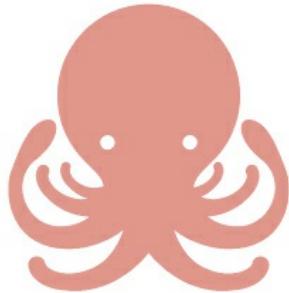
さて、ディープラーニングの基本は「畳み込みニューラルネットワーク」であり、畳み込みニューラルネットワークのアイデアは「動物の脳を真似ることから生まれた」と述べました。すると、逆にディープラーニングを学ぶことで、動物の脳、そして人の脳の理解につながるはずです。

例えば、「単純な神経細胞(ニューロン)がネットワークを組むと、なぜ『知能』が生まれるのか?」という困難な課題に対して、畳み込みニューラルネットワークは一つの解決の糸口を提供できるかもしれません。



また、「鍵刺激」といって、動物には本能的な行動を起こさせる特定の刺激がありますが、なぜそのようなことが起こるのかも、畳み込みニューラルネットワークは一つの解決のアプローチを提供するはずです。

例えば、イイダコというタコの仲間はラッキョウを見せると飛びついてきます。そこで、ラッキョウを疑似餌としてタコ釣りが楽しめます。しかし、どうしてイイダコがラッキョウに反応するのでしょうか。本書で調べる畳み込みニューラルネットワークのしくみがわかると、その理解の糸口がつかめるかもしれません。



イイダコはラッキョウを見ると飛びついてくる。そのしくみが畳み込みニューラルネットワークでわかるかもしれない。

さらにまた、動物の脳の中の1ニューロンが破壊されても、全体には大きな影響を示さない、という特性があります。そうでなければ動物の生命維持は不可能でしょう。これはコンピューターの世界とは大違います。コンピューターにおいて、そのCPUの配線の1本が切れただけでも、大変なことになります。このような動物とコンピューターの違いも、本書で調べる畳み込みニューラルネットワークを理解することで、納得がいくようになるかもしれません。

以上調べたように、畳み込みニューラルネットワークの理解は、さまざまな応用と発展の可能性を秘めています。Excelという便利な道具を利用して、これからこの世界に分け入ってみましょう。

Memo | ディープラーニングの誕生

2006年、トロント大学Hinton教授が発表した論文が発端といわれます。その後、その考え方を用いて開発されたソフトウェアが一般物体の認識のコンテスト(ILSVRC)で優勝し(2012年)、脚光を集めました。

§2

AIとディープラーニング

AI(人工知能)という言葉が氾濫しています。しかし、AIとは何かについての議論は、なかなか定まらないようです。そこで、ディープラーニングを理解するために、歴史を少し振り返ってみましょう。

■ AIの定義は定かでない

以前から「AI炊飯器」などというように、AIを冠した商品が販売されていました。その「AI」とは何かについては、統一的な定義はなされていません。そこで、次のような極論も生まれます。

「機械が一回でも判断すれば、それをAIと呼ぶ」

これを認めれば、例えば「暑いからスイッチを入れる」という単純なエアコンも「AI搭載エアコン」ということになります。

現在マスコミや経済界を騒がせているAIは、「ディープラーニング」と呼ばれるアルゴリズムを用いるものです。今世紀において、そのディープラーニングの輝かしい成果を紹介しましょう。

年	成果
2012 年	世界的な画像認識コンテスト ILSVRC で、ディープラーニングの手法を用いた手法が圧勝。
2012 年	Google の開発したディープラーニングの手法を用いた AI がユーチューブの画像から猫を認識する。
2014 年	Apple 社は Siri の音声認識を、ディープラーニングの手法を用いたシステムに変更。
2016 年	Google が開発したディープラーニングの手法を用いた AI「アルファ碁」が世界トップ級の棋士と勝負し、勝ちを収める。
2016 年	AUDI や BMW 社で、自動車の自動運転にディープラーニングの手法が利用される。

この表が示すように、ディープラーニングは人工知能(AI)の分野で大きな成功を収めているのです。定義の定かでないAIですが、その一つの実装手段であるディープラーニングは、確実に成果をあげています。

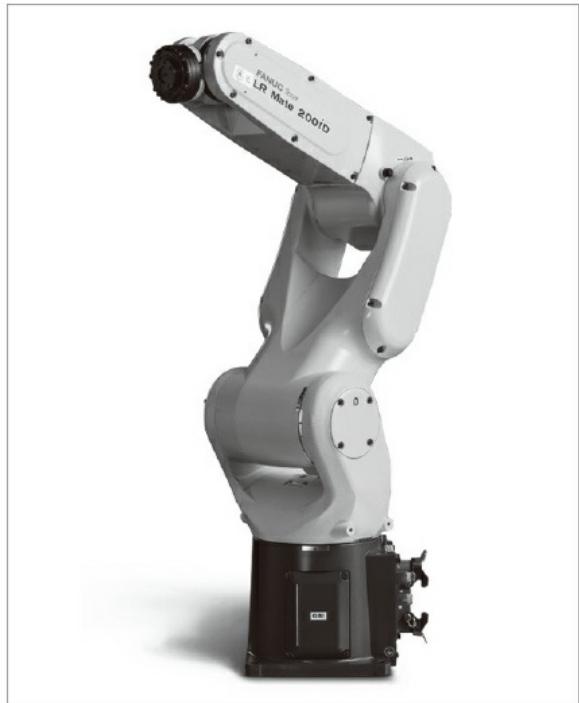
■ AIにもブームがある

すべての分野がそうであるように、AIにもブームがありました。それをまとめてみましょう。

空想ではなく、現実としての人工知能(AI)は1950年代から研究が始まったといわれます。それはコンピューターの開発の歴史と重なりますが、以下の3つのブームに分けられます。

世代	年代	キーとなる言葉	主な応用分野
第1世代	1950～1970	論理	パズルなど
第2世代	1980年代	知識	ロボット、自動翻訳
第3世代	2010年～	データ	パターン認識、音声認識

第2世代では、日本が主導権を握りました。代表的なものはエキスパートシステムと呼ばれます。さまざまな分野の達人の知識を教え込むタイプのAIです。その結果として、日本の産業用ロボットが世界を席巻することになります。



産業用ロボット

このロボットの多くには、人が教え込むタイプの人工知能が利用されている。各界の達人の技まで習得したロボットも多い。このようなAIをエキスパートシステムと呼んでいる。

本書のテーマになる第3世代は、表にも示しましたが、データが主役になります。先にも言及したように、ディープラーニングは自ら学習する論理を採用します。そこで、論理ではなく大量のデータが必要になるのです。IoTやビッグデータなどの時代を代表する言葉と共にAIが話題になるのはこのためです。

■ AI研究の明るい未来

ディープラーニングはAI(人工知能)の実現の大きな一歩です。序でも触れましたが、米グーグルの人工知能「アルファ碁」の生みの親であるデミス・ハサビス氏は「ディープラーニングと強化学習の2つの技術が確実に知能の研究を切り開く」と

明言しています。そして、「正しいはしごを登り始めた」と自信を示しています。実際、前の項（「AIの定義は定かでない」）に示したディープラーニングのやり遂げた偉業を見ると、それは夢の話ではないと思われます。本書で取り上げたディープラーニングの基本の話はその最初の一歩にすぎませんが、大きな飛躍の可能性を秘めています。

Memo | AIと誰が命名？

1956年、アメリカのニューハンプシャー州ハノーバーにあるダートマス大学で開催されたダートマス会議において、Artificial Intelligence（人工知能）という言葉が初めて提唱されました。その会議のメンバーのジョン・マッカーシーが提案した言葉です。マッカーシーは「人工知能の父」とも呼ばれるコンピューター科学の学者です。ちなみに、人工知能という言葉ではないのですが、同じ概念は1947年、「コンピューター科学の父」といわれるアラン・チューリングによって提唱されています。

2 章

Excel の確認と その応用

ニューラルネットワークや畳み込みニューラルネットワークで利用する Excel の知識は大変初等的です。多くの読者にとって周知のことと思われます。しかし、その初等的な知識がないと先に進めません。老婆心ながら、本章では一応の確認をすることにします。

また、それに絡めてデータ分析の基本になる回帰分析の方法を調べます。

§ 1

利用するExcel関数はたったの7個

本書のニューラルネットワークや畳み込みニューラルネットワークの解説で利用する関数はわずかです。すべて有名な関数であり、解説を要しないかもしれませんのが、確認しましょう。

■ 利用する関数は7個

次の7個の関数を使えば、ニューラルネットワークや畳み込みニューラルネットワークが簡単に構築できます。

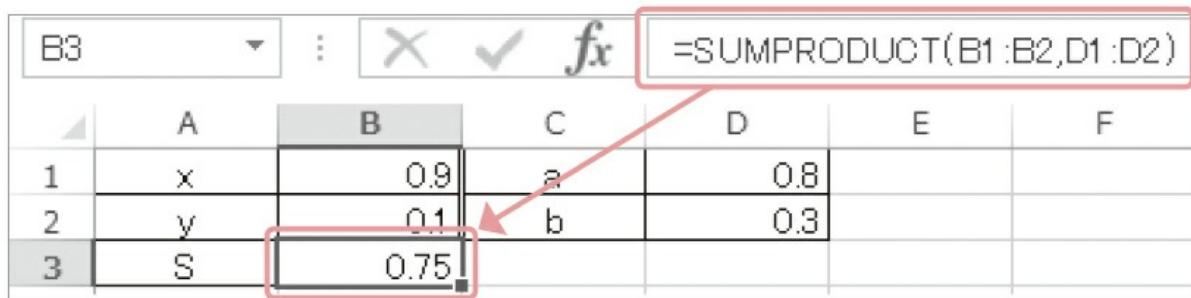
関数	意味	利用例
SUM	セル範囲の数値の和を計算します。	目的関数の計算
SUMPRODUCT	2つの指定した範囲にある数値の積和を計算します。	入力の線形和
SUMXMY2	2つの指定した範囲にある数値の差の平方和を計算します。	平方誤差の算出
EXP	指数関数の値を計算します。	シグモイド関数
MAX	指定した範囲の最大値を求めます。	MAX プーリング、ReLU
RAND	0以上1以下の乱数を発生。	初期値設定
IF	大小の判定。	画像の判定

この表の中で、後述の「入力の線形和」の計算に便利な**SUMPRODUCT**関数、最適化問題を解くときに便利な**SUMXMY2**関数(Sum of $(X - Y)^2$)、そして**EXP**関数の使い方を、例で確認します。

注 以下の例のワークシートは、ダウンロードサイト（→卷頭「Excelサンプルファイルのダウンロードについて」）に掲載されたファイル「2.xlsx」の中の「1_例1」「1_例2」「1_例3」タブにあります。

例1 $(x, y) = (0.9, 0.1)$ 、 $(a, b) = (0.8, 0.3)$ とします。このとき次の積和 S を、SUMPRODUCT関数を用いて求めましょう。

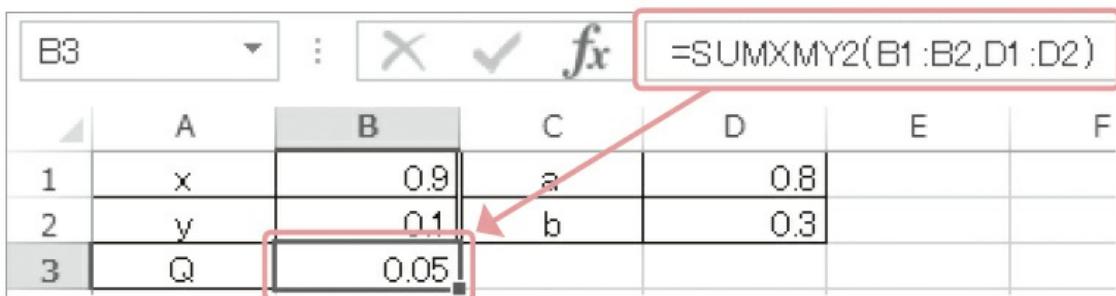
$$S = ax + by$$



	A	B	C	D	E	F
1	x	0.9	a	0.8		
2	y	0.1	b	0.3		
3	S	0.75				

例2 (例1)と同じく $(x, y) = (0.9, 0.1)$ 、 $(a, b) = (0.8, 0.3)$ とするとき、次の「差の平方和」 Q を、SUMXMY2関数を用いて求めましょう。

$$Q = (x - a)^2 + (y - b)^2$$

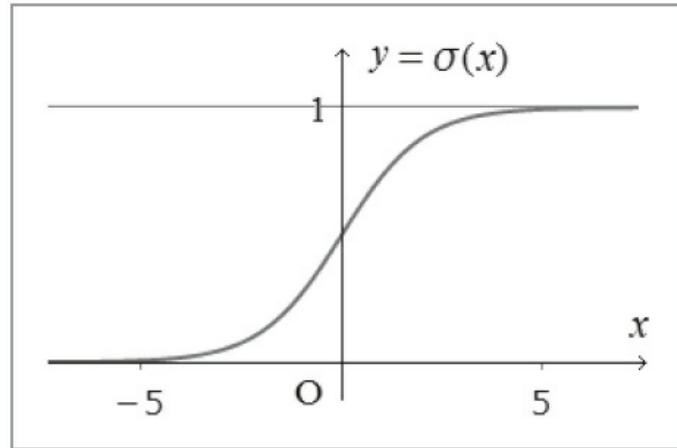


	A	B	C	D	E	F
1	x	0.9	a	0.8		
2	y	0.1	b	0.3		
3	Q	0.05				

例3 EXP関数を用いてシグモイド関数を作成しましょう。

シグモイド関数 $\sigma(x)$ は後述するように指数関数 e^x から次のように得られます（グラフは次図）。

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad \cdots (1)$$



ここで、 e はネイピア数と呼ばれる数で、次の値で近似されます。

$e \doteq 2.71828$ （「鮒一鉢二鉢」という覚え方が有名）

EXP関数は指数関数 e^x の値を求めるための関数です。式(1)からシグモイド関数は次のように表現できます。

B2	▼	⋮	X	✓	fx	=1/(1+EXP(-A2))
	A	B	C	D	E	
1	x	$\sigma(x)$				
2	1	0.731059				

§ 2

Excelの参照形式

表計算を使う上で必須の知識が「セルの参照方法」です。その方法は3種あります。3つともExcelでニューラルネットワークを実装するときに多用されます。

注 本節の各例のワークシートは、ダウンロードサイト（→卷頭「Excelサンプルファイルのダウンロードについて」）に掲載されたファイル「2.xlsx」の中の「2_例1」「2_例2」「2_例3」タブにあります。

■ セル参照

表計算において、計算式はセルの番地から構成されるのが普通です。このようにセルの番地を引用することをセル参照といいますが、その方法として、相対参照、絶対参照、複合参照の3つの方法があります。これらの違いをマスターすることが肝要です。

■ 相対参照

表計算の標準的なセル参照は相対参照です。あるセルに書かれた計算式を他のセルにコピーすると、その相対移動分だけ、計算式の番地が更新されます。次の例で確かめましょう。

例1 次のように社員3人の身長と体重の記録があります。この人たちの体格指

数（BMI）（＝体重kg ÷（身長m）²）を求めましょう。

最初に、1番の社員のBMIの計算式をセルD2に入力します。



	A	B	C	D	E
1	社員番号	身長(m)	体重(kg)	BMI	
2	1	1.75	64	20.9	
3	2	1.64	59		
4	3	1.67	71		

次に、セルD2に入力した計算式をD3:D4にコピーします。次の図に示すように、Excelは自動的に参照アドレスを変更してくれます。これが相対参照を利用するときの利点です。



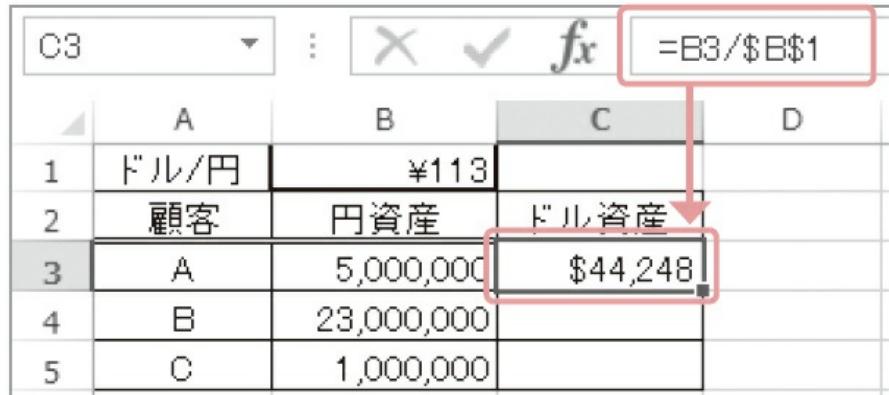
	A	B	C	D	E
1	社員番号	身長(m)	体重(kg)	BMI	
2	1	1.75	64	20.9	
3	2	1.64	59	21.9	
4	3	1.67	71	25.5	

■ 絶対参照

あるセルに書かれた計算式を他のセルにコピーするとき、計算式の番地を更新しない参照法が絶対参照です。参照するセルの番地に\$を付けます。

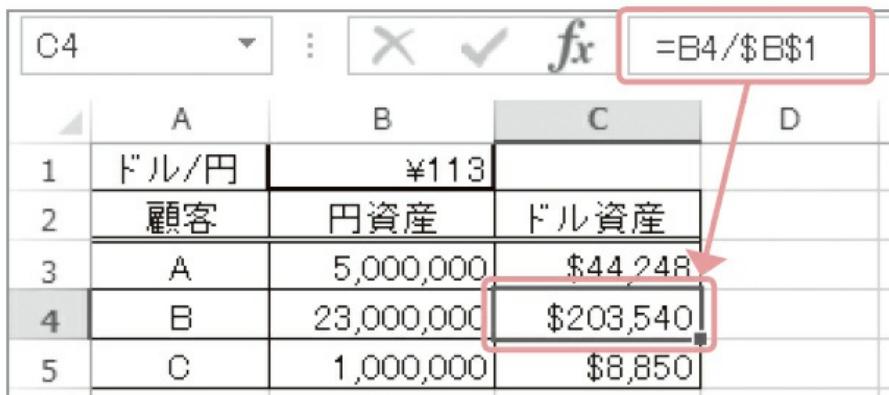
例2 資産管理会社が掌握している顧客A、B、Cの円資産を米ドルに換算しましょう。

最初に次のように、顧客Aのドル資産の換算式を入力します。ドル/円レートのセルのアドレスが\$B\$1と絶対参照になっていることに留意してください。



	A	B	C	D
1	ドル/円	¥113		
2	顧客	円資産	ドル資産	
3	A	5,000,000	\$44,248	
4	B	23,000,000		
5	C	1,000,000		

次に、セルC3をC4:C5にコピーします。次図に示すように、ドル/円レートのセルアドレスは固定されています。



	A	B	C	D
1	ドル/円	¥113		
2	顧客	円資産	ドル資産	
3	A	5,000,000	\$44,248	
4	B	23,000,000	\$203,540	
5	C	1,000,000	\$8,850	

このように、定数など、各セルの数式に共通の値の収められたセルは絶対参照しておくことが肝要です。

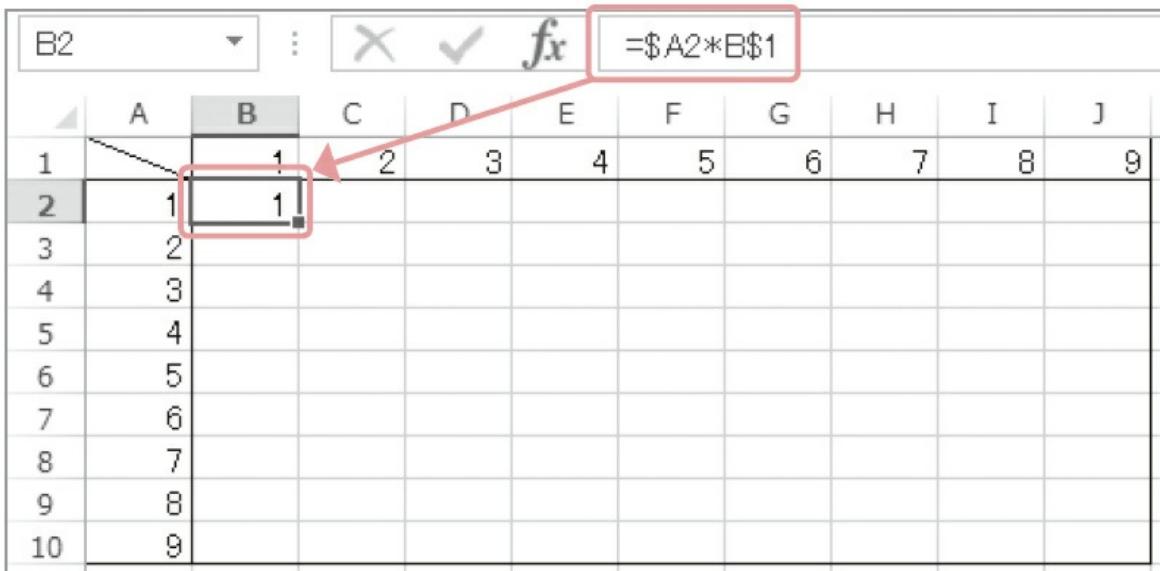
■ 複合参照

式の表現に相対参照と絶対参照とをミックスした参照法が複合参照です。参照するセルの行または列の番地の一方に\$を付けます。この参照法を利用して「九九の計算表」を作成してみましょう。

例3 表の上端(すなわち表頭)と表の左端(すなわち表側)の数字から、その積

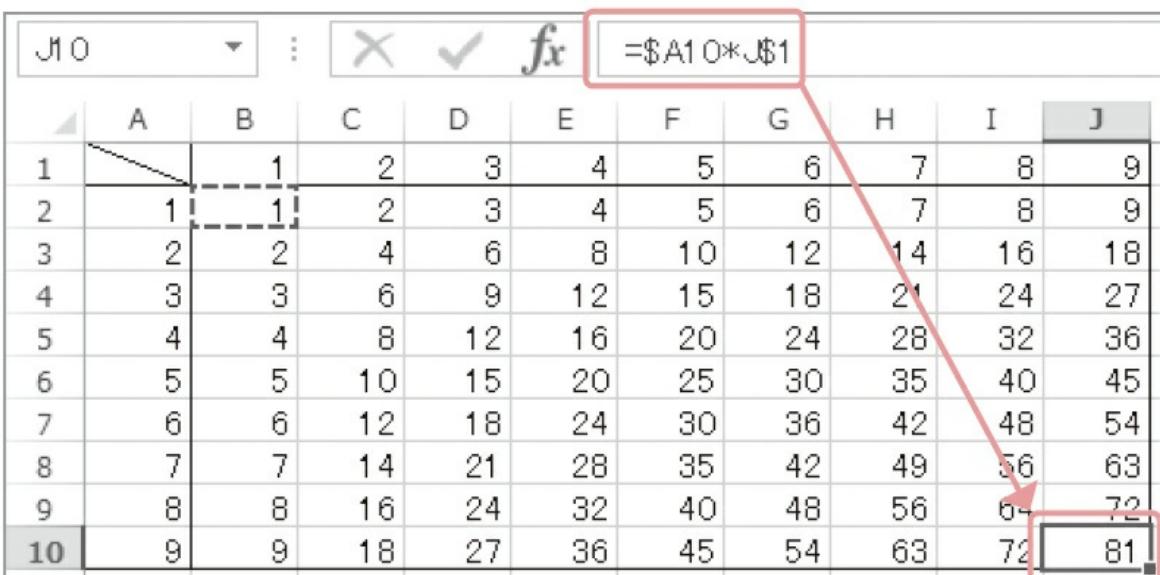
を求め、「九九の表」を完成しましょう。

最初に、次図のように 1×1 を作成します。



	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1	1								
3	2									
4	3									
5	4									
6	5									
7	6									
8	7									
9	8									
10	9									

次に、このセルの関数を次図のように表一杯にコピーします。



	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1	1	2	3	4	5	6	7	8	9
3	2	2	4	6	8	10	12	14	16	18
4	3	3	6	9	12	15	18	21	24	27
5	4	4	8	12	16	20	24	28	32	36
6	5	5	10	15	20	25	30	35	40	45
7	6	6	12	18	24	30	36	42	48	54
8	7	7	14	21	28	35	42	49	56	63
9	8	8	16	24	32	40	48	56	64	72
10	9	9	18	27	36	45	54	63	72	81

\$を付けていない行または列の番地が更新され、\$を付けている行または列の番地は更新されないことに留意してください。

§ 3

Excelソルバーの使い方

次節に後述するように、データ分析のために作成された数学モデルに含まれるパラメーターを決めるのに大変便利なツールがExcelに備えられているソルバーです。このソルバーの使い方を簡単な例で確認しましょう。

注 ソルバーはExcelのアドインであり、初期状態ではインストールされていない場合があります。その際には、巻末の付録Cを参照してください。

■ ソルバーを使ってみよう

例題を用いて、Excelソルバーの利用法を調べます。

例題1 関数 $y=3x^2+1$ の最小値とそれを実現する x をExcelのソルバーで求めましょう。

注 このワークシートは、ダウンロードサイト(→巻頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「2.xlsx」の中の「3_例題」タブにあります。

解

周知のように、解答は「 $x=0$ のとき、 y の最小値は1」です。それがソルバーを用いて得られることを確認します。それには、次のステップを追いましょう。

- ① x が与えられたときの関数 y の式を入力します。そして、 x に適当な初期値を設定しておきます(ここでは5を入れましたが、それに意味があるわけではありません)。

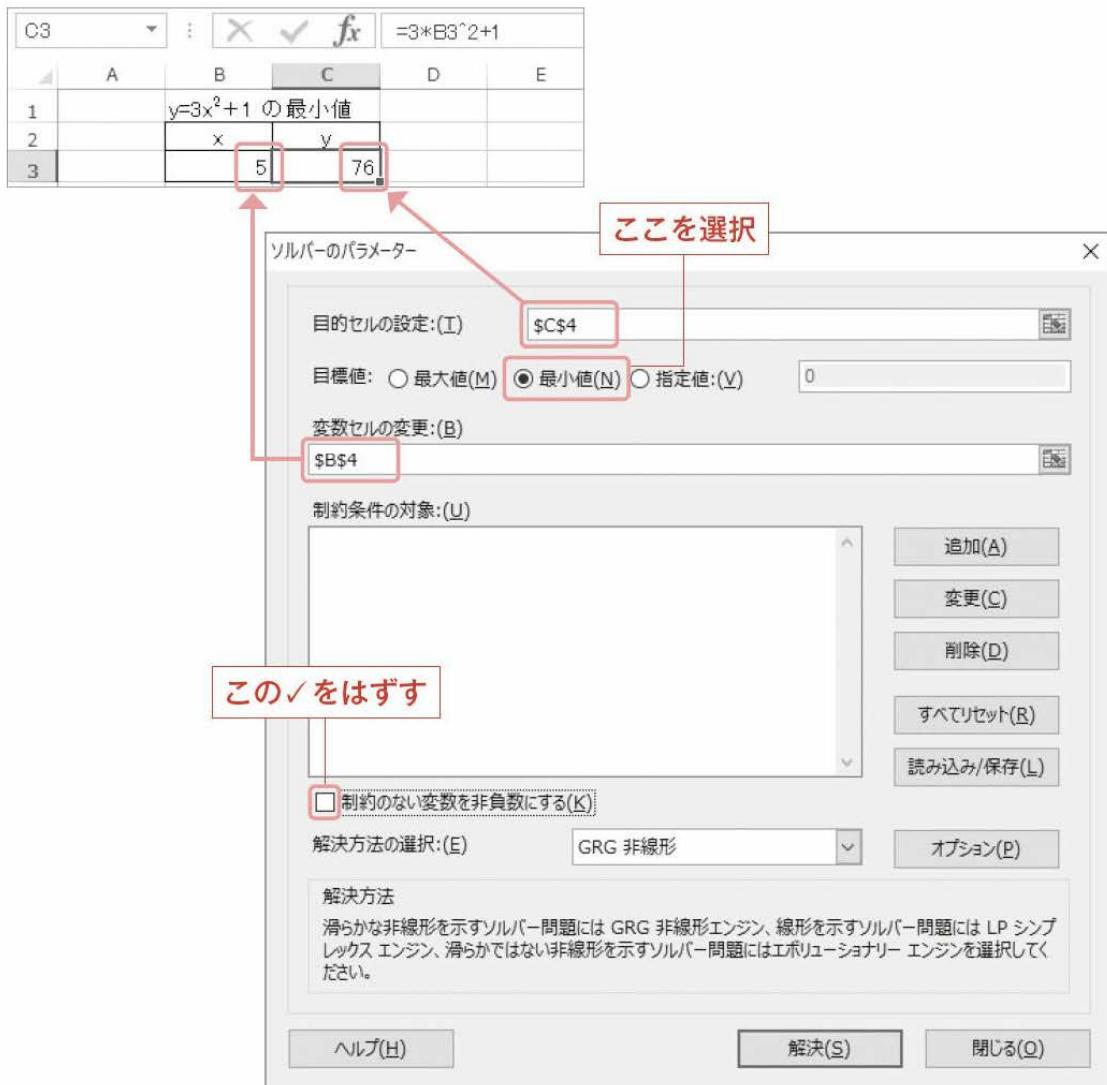
C3	A	B	C	D	E
			y=3x^2+1 の最小値		
1		x	y		
2		5	76		
3					

- ② ソルバーを起動します。それには「データ」リボンにある「ソルバー」の項目を選択します。

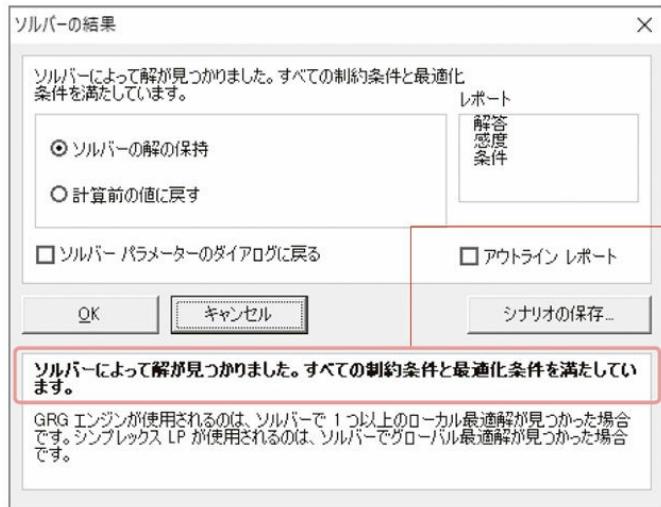


注 ソルバーがインストールされていないと、このメニューはありません(→付録C)。

ここで、次のように設定します。



- ③ ソルバーの「解決」ボタンをクリックし、実行します。ソルバーが最小値を探せば、次のメッセージが現れます。



このメッセージが大切

ソルバーの求めた解は最小値であることを保証しない。すなわち、極小解である可能性がある（この例の場合、その心配はない）。

「OK」ボタンをクリックすれば完了です。

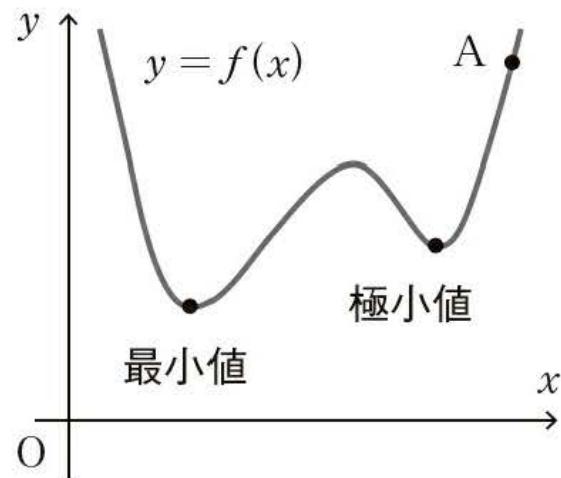
	A	B	C	D
1		$y=3x^2+1$ の最小値		
2				
3		x	y	
4		0	1	

ソルバーの計算結果

こうして、先にも述べた「 $x = 0$ のとき、 y の最小値は1」が得られました。

Memo ソルバーの求める「最小値」は極小解

関数 $y = f(x)$ のグラフが次の図のようだとします。このとき、 x の初期値として図に示した点 A の x 座標を与えると、ソルバーは極小値（次図）を求めてします。ソルバーは少しずつ変数を移動しながら小さい値を探していくからです。このような解を極小解といいます。



ソルバーを利用するときに、この極小解の存在には十分注意が必要です。回避するには初期値を色々変えたり、関数のグラフィイメージをある程度予測したり、という面倒な操作が必要になります。

§ 4

回帰分析と最適化問題

データ分析を理解するには「回帰分析」が最適です。すべてのモデルの原点がここにあるからです。データ分析の定石を調べながら、Excelのソルバーの使い方の復習もしましょう。

■ データを収める変数とモデルを定めるパラメーター

データを分析するには、数学的なモデルを作成します。このモデルはデータを収めるための変数と構造を決めるためのパラメーターがセットになっています。このパラメーターを決めるのが最適化と呼ばれる数学的技法です。

本書で扱うニューラルネットワークの決定は、数学的にいえば、最適化問題の一つです。ニューラルネットワークを規定するパラメーター（すなわち重みや閾値など）を、実際のデータに合致するようにフィットさせる問題なのです。

この最適化問題を理解するのに最もわかりやすい例題が回帰分析です。簡単な回帰分析を利用して、この最適化問題のしくみを調べましょう。これを理解すれば、ニューラルネットワークや畳み込みニューラルネットワークを決定する方法がすぐに理解できるようになります。

■ 回帰分析とは

複数の変数からなる資料において、特定の1変数に着目し、それを他の変数で説明する手法を回帰分析といいます。回帰分析にはいろいろな種類がありますが、考え方を知るために最も簡単な「線形の単回帰分析」と呼ばれる分析法

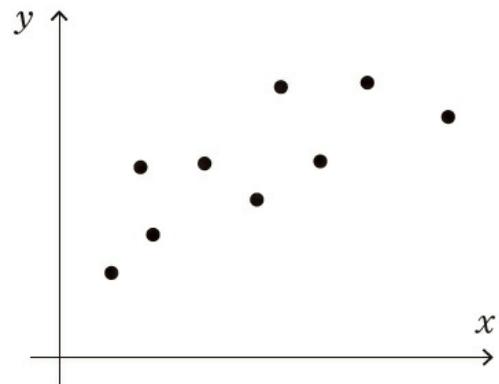
を調べることにします。

「線形の单回帰分析」とは2つの変数からなる資料を対象にします。いま、次のように2変数 x 、 y の資料とその相関図が与えられているとします。

(回帰分析というデータ分析モデルにおいて、この2変数 x 、 y が、先に示した「データを収めるための変数」の役割を担います。)

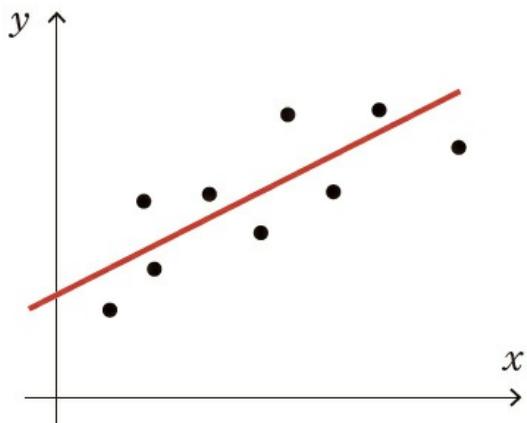
個体名	x	y
1	x_1	y_1
2	x_2	y_2
3	x_3	y_3
...
n	x_n	y_n

資料



左の資料の相関図

「線形の单回帰分析」は、上に示した相関図上の点列を直線で代表させ、その直線の式で2変数の関係を調べる分析術です。点列を代表するこの直線を回帰直線と呼びます。



回帰直線。この直線の式から、2変数の関係を調べる分析術が線形の单回帰分析。

回帰直線は次のように1次式で表現されます。

$$y = px + q \quad (p, q \text{ は定数}) \dots (1)$$

これを回帰方程式と呼びます。

x 、 y はデータの実際の値を入れるための変数で、右辺の x を説明変数、左辺の y を目的変数といいます。定数 p 、 q はこの回帰分析モデルを定めるパラメーターで、与えられたデータに適合するように決定されます。

注 p を回帰係数、 q を切片と呼びます。

■ 具体例で回帰分析の論理を理解

次の具体例を通して、回帰方程式(1)をどのように決定するか見てみましょう。その決定法は後に調べるニューラルネットワークや畳み込みニューラルネットワークの決定法と同一です。

例題1 次の資料は高校3年生の女子生徒7人の身長と体重の資料です。この資料から、体重 y を目的変数、身長 x を説明変数とする回帰方程式 $y = px + q$ (p 、 q は定数)を求めましょう。

番号	身長 x	体重 y
1	153.3	45.5
2	164.9	56.0
3	168.1	55.0
4	151.5	52.8
5	157.8	55.6
6	156.7	50.8
7	161.1	56.4

生徒7人の身長と体重の資料

注 このワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「2.xlsx」の中の「4_例題1」タブにあります。

解

求める回帰方程式を次のように置きます。

$$y = px + q \quad (p, q \text{ は定数}) \dots (2)$$

k 番の生徒の身長を x_k 、体重を y_k と表記しましょう。すると、この生徒の回帰分析が予測する体重(予測値といいます)は次のように求められます。

$$\text{予測値: } px_k + q$$

この予測値を表に示しましょう。

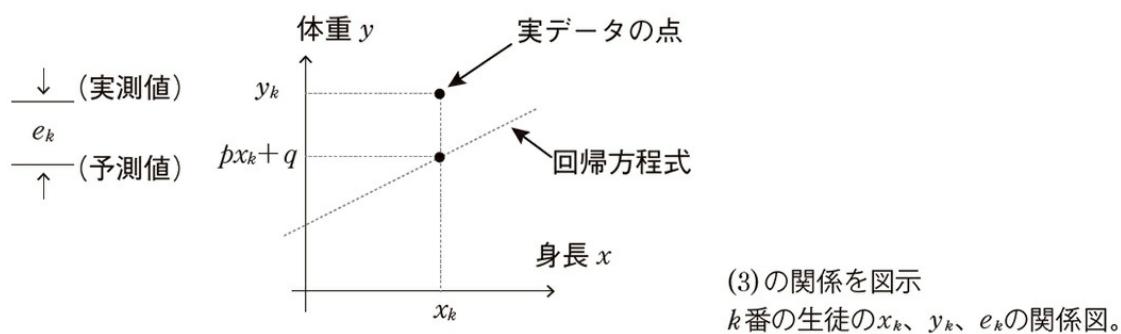
番号	身長	体重	予測値 $px + q$
1	153.3	45.5	153.3 $p + q$
2	164.9	56.0	164.9 $p + q$
3	168.1	55.0	168.1 $p + q$
4	151.5	52.8	151.5 $p + q$
5	157.8	55.6	157.8 $p + q$
6	156.7	50.8	156.7 $p + q$
7	161.1	56.4	161.1 $p + q$

体重 y の実測値と予測値。
数学的な最適化を考える際、実測値と予測値の違いを理解しておくことは大切。

実際の体重 y_k と予測値との誤差 e_k は次のように算出されます。

$$e_k = y_k - (px_k + q) \dots (3)$$

注 e は error の頭文字。



この e_k の値は正にも負にもなり、データ全体で加え合わせると打ち消し合ってしまいます。そこで、次の値 Q_k を考えます。これを資料 k 番目の平方誤差と呼びます。

$$Q_k = (e_k)^2 = \{y_k - (px_k + q)\}^2 \dots (4)$$

注 文献によって式(4)右辺には様々な定数係数が付きます。しかし、分析結果は同一です。

この平方誤差をデータ全体で加え合わせてみましょう。それはデータ全体の「誤差の総和」です。それを Q_T とすると、次のように式で表せます。

$$Q_T = Q_1 + Q_2 + \dots + Q_7 \dots (5)$$

前表の値を式(4)に代入すると、誤差の総和 Q_T は p 、 q の式で次のように表せます。

$$Q_T = \{45.5 - (153.3p + q)\}^2 + \{56.0 - (164.9p + q)\}^2 + \dots + \{56.4 - (161.1p + q)\}^2 \dots (6)$$

この誤差の総和(5)(すなわち(6))を目的関数といいます。最小化の目的となる関数だからです。注意すべきことは、この関数がパラメーター p 、 q の関数になっていることです。データ x 、 y に対して p 、 q は定数でしたが、最適化の際には変数になるのです。

さて、目標はこのパラメーター p 、 q の決定です。回帰分析では「目的関数(5)(すなわち(6))が最小になる p 、 q が解となる」と考えます。これは常識にも合致します。目的関数は誤差の総和であり、それが最小であることは良いモデルであると考えられるからです。平方誤差を最小にするこの最適化の方法を最小2乗法といいます。

この考え方方が与えられれば、後は簡単です。ソルバーで目的関数(6)が最小になるパラメーター p 、 q を探せばよいからです。以下に、ステップを追って p 、 q を求めてみましょう。

- ① 仮のパラメーター p 、 q を入力し、その値から回帰方程式(2)を用いて体重 y の予測値を計算します。

単回帰分析					
	A	B	C	D	E
1	<i>p</i>	1			
2	<i>q</i>	1			
3	番号	身長 x	体重 y	予測値	平方誤差
7	1	153.3	45.5	154.3	
8	2	164.9	56.0	165.9	
9	3	168.1	55.0	169.1	
10	4	151.5	52.8	152.5	
11	5	157.8	55.6	158.8	
12	6	156.7	50.8	157.7	
13	7	161.1	56.4	162.1	

仮のパラメーター p 、
 q として各々 1 を入
力。それを用いて式
(2) から y を算出

- ② 式(4)から、各女子生徒について平方誤差を算出します。

単回帰分析					
	A	B	C	D	E
1	<i>p</i>	1			
2	<i>q</i>	1			
3	番号	身長 x	体重 y	予測値	平方誤差
7	1	153.3	45.5	154.3	11837.4
8	2	164.9	56.0	165.9	12078.0
9	3	168.1	55.0	169.1	13018.8
10	4	151.5	52.8	152.5	9940.1
11	5	157.8	55.6	158.8	10650.2
12	6	156.7	50.8	157.7	11427.6
13	7	161.1	56.4	162.1	11172.5

式(4)から平方誤差
を算出

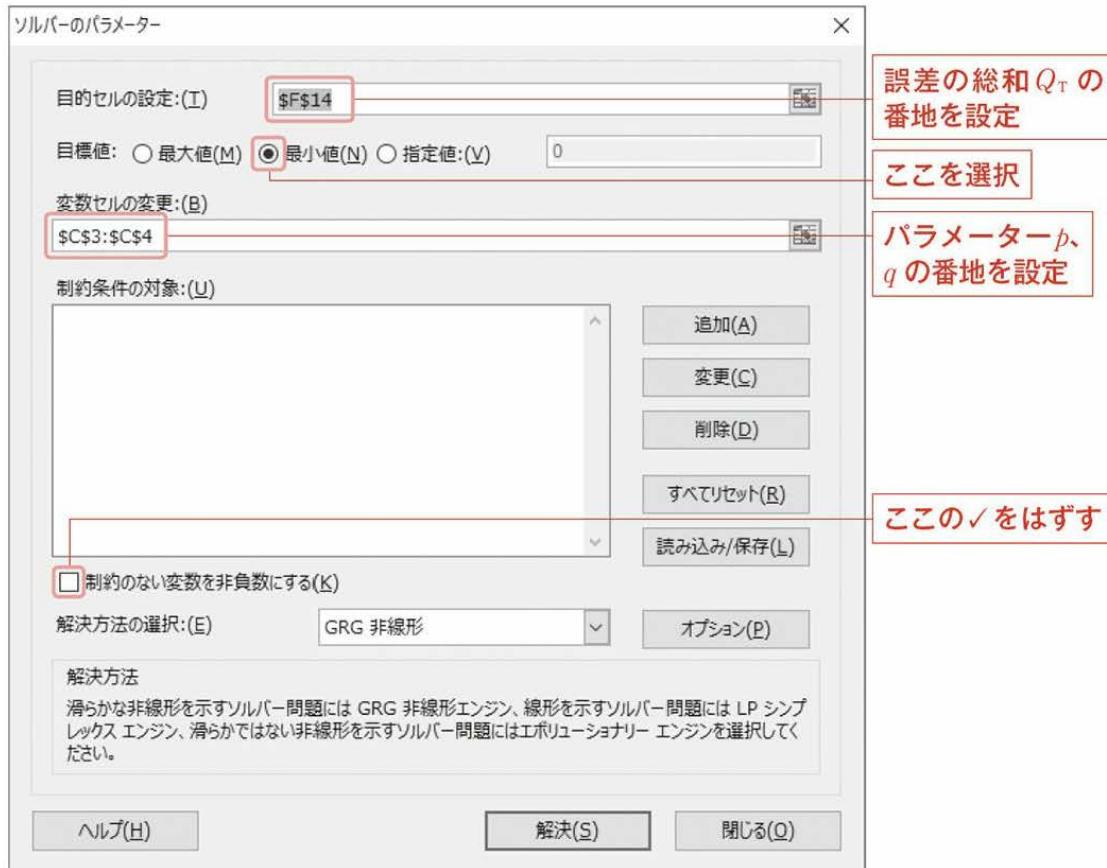
- ③ 平方誤差の総和 Q_T を SUM 関数で算出します (→ 式(5)(6))。

F14 ▾ : fx =SUM(F7:F13)

	A	B	C	D	E	F
1		单回帰分析				
2						
3		p	1			
4		q	1			
5						
6		番号	身長 x	体重 y	予測値	平方誤差
7	1	153.3	45.5	154.3	11837.4	
8	2	164.9	56.0	165.9	12078.0	
9	3	168.1	55.0	169.1	13018.8	
10	4	151.5	52.8	152.5	9940.1	
11	5	157.8	55.6	158.8	10650.2	
12	6	156.7	50.8	157.7	11427.6	
13	7	161.1	56.4	162.1	11172.5	
14				計 Q_T	80124.7	

平方誤差の総和

- ④ ソルバーを起動し、 Q_T の入ったセルを「目的セル」に、仮の値の入った p 、 q のセルを「変数セル」に次図のように設定します。



⑤ ソルバーを実行すると、次図のようにパラメーター p 、 q の値と平方誤差の総和 Q_T の値が得られます。

F14 : $=SUM(F7:F13)$

A	B	C	D	E	F
単回帰分析					
3	p	0.41			
4	q	-12.06			
6	番号	身長 x	体重 y	予測値	平方誤差
7	1	153.3	45.5	50.8	28.1
8	2	164.9	56.0	55.6	0.2
9	3	168.1	55.0	56.9	3.5
10	4	151.5	52.8	50.1	7.5
11	5	157.8	55.6	52.6	8.8
12	6	156.7	50.8	52.2	1.9
13	7	161.1	56.4	54.0	5.8
14				計	55.7

最適化されたパラメーター p 、 q の値

最適化された Q_T の値

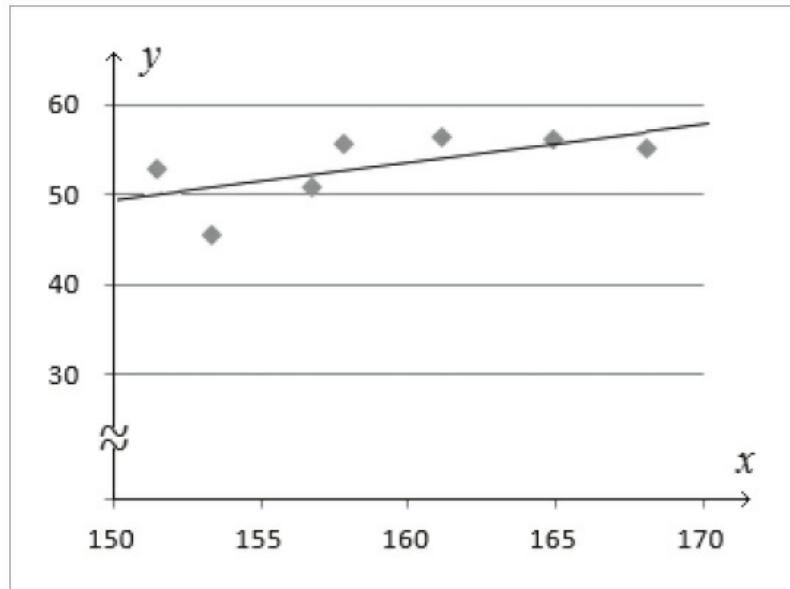
こうして、回帰係数と切片 p 、 q の値が得られました。

$$p = 0.41, q = -12.06 \dots (7)$$

また、回帰方程式は次のように表せます。

$$y = 0.41x - 12.06 \dots (8)$$

以上が[例題]の解答です。これを利用して、このデータの散布図と回帰直線の関係を図示しましょう。重なっていることが確かめられます。



例題の解となる回帰直線。

注意すべきことは、平方誤差の総和 Q_T が 0 にはならないことです。それは回帰直線が散布図にプロットされたすべての点を通らないことから明らかです。データとそれを説明するための回帰方程式とのせめぎ合いの中で、ぎりぎりの妥協値を(7)の p 、 q は表しているのです。

この回帰方程式を利用してみましょう。

例1 [例題1]で得られた回帰方程式を利用して、身長170cmの女子生徒の体重を予測してみましょう。

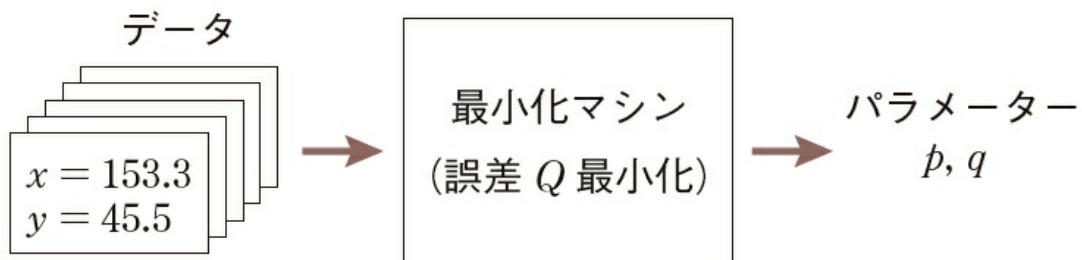
方程式(8)から、この女子生徒の体重は次のように予測されます。

$$\text{予測体重 } y = 0.41 \times 170 - 12.06 = 57.64 \text{ (答)}$$

■ 回帰分析が“わかれば”データ分析が“わかる

以上が線形の单回帰分析で用いられる回帰方程式の決定法と応用例です。大切なことは、これが“数学のデータ分析の典型例であり、「最適化問題」の解法のアイデアそのものである、ということです。ここで調べた最適化の方法は後の

ニューラルネットワークの計算に活かされます。



回帰分析は数学のデータ分析の典型例。分析モデルのパラメーター p, q がどのように決定されるかを確認しよう。

■ モデルのパラメーターの個数

再度、先の[例題1]を見てみましょう。モデルを規定するパラメーターの個数は p, q の2個です。そして、与えられた条件(データの大きさ)は7個でした。モデルのパラメーターの個数(いまは p, q の2個)が条件の個数(いまはデータの大きさ7個)より小さいのです。換言すれば、たくさんの条件を突きつけられ、妥協の産物として得られたのが回帰方程式なのです。その妥協とは、理想的には0となるべき目的関数(5)の値を最小にすることだけにとどめていることです。したがって、モデルとデータとの誤差 Q_T が0にならなくても心配する必要はありません。しかし、0に近いほど、データにフィットしたモデルといえます。

ちなみに、モデルのパラメーターの個数がデータの大きさより大きいと、どうなるでしょう。当然ですが、このときパラメーターは確定しません。したがって、モデルを確定するには、パラメーターの個数よりも大きなデータを用意しなければなりません。このことは後に調べるニューラルネットワークの世界では深刻です。ニューラルネットワークでは、パラメーターの個数が膨大になるからです。

■ 別の例題で確認

次の[例題2]は補外法(または外挿法)と呼ばれる数値解析で有名な技法を紹介しています。回帰分析と同じアイデアが用いられるので、いま調べた理論のよい復習になるでしょう。

例題2 変数 x と y の値が次の表のように与えられています。 y を x の1次式 $ax + b$ (a 、 b は定数)で予測し、 x が5のときの y の値を推定しましょう。なお、ここで最適化には最小2乗法を用いることにします。

x	y
1	13.3
2	15.8
3	19.4
4	22.3

注 推定したい x の値が1と4の間にあるときは補間法といいます。このワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「2.xlsx」の中の「4_例題2」タブにあります。

解

データを入力します。また、定数 a 、 b の値を適当に仮定します。

A	B	C	D	E	F	G	H
1	補外法						
2	a	1	x	y	ax+b	Q	
3	b	1	1	13.3			
4			2	15.8			
5			3	19.4			
6			4	22.3			

定数 a 、 b に適当な
値を代入

以上の準備の下に、次のステップを追いましょう。

注 以下の①～⑤は先の[例題1]の①～⑤のステップに準じています。

① 予測値 $ax + b$ の値を求めます。

A	B	C	D	E	F	G	H
1	補外法						
2	a	1	x	y	ax+b	Q	
3	b	1	1	13.3	2.00		
4			2	15.8	3.00		
5			3	19.4	4.00		
6			4	22.3	5.00		

絶対参照、相対参照の使い分けに
注意。

② 実測値 y と予測値 $ax + b$ の誤差として、平方誤差 Q を求めます。

A	B	C	D	E	F	G	H
1	補外法						
2	a	1	x	y	ax+b	Q	
3	b	1	1	13.3	2.00	127.69	
4			2	15.8	3.00	163.84	
5			3	19.4	4.00	237.16	
6			4	22.3	5.00	299.29	

y と $ax+b$ との差を平方。

③ 平方誤差 Q の総和 Q_T を算出します。

H7						fx	=SUM(H3:H6)
	A	B	C	D	E	F	G
1			補外法				H
2	a		1	x	y	ax+b	Q
3	b		1	1	13.3	2.00	127.69
4				2	15.8	3.00	163.84
5				3	19.4	4.00	237.16
6				4	22.3	5.00	299.29
7						Q _T	827.98

誤差の総和の算出にはSUM関数が便利。

④ ソルバーで平方誤差の総和 Q_T の最小値を求めます。

先の回帰分析と同様にソルバーを設定します。目的セルを総和 Q_T の関数の収められたH7に、変数セルを a 、 b の(仮の)値の収められたC2:C3にセットします(次の図)。

Excel シートとソルバーのパラメーター設定画面が表示されています。

Excel シート (H7 の計算式: =SUM(H3:H6)):

	A	B	C	D	E	F	G	H
1	補外法							
2	a		1	x	y	ax+b	Q	
3	b		1	1	13.3	2.00	127.69	
4				2	15.8	3.00	163.84	
5				3	19.4	4.00	237.16	
6				4	22.3	5.00	299.29	
7						Q _T	827.98	

ソルバーのパラメーター (赤い枠で囲った部分):

- 目的セルの設定: (I) \$H\$7
- 目標値: 最小値(N) 0
- 変数セルの変更: (B) \$C\$2:\$C\$3
- この✓をはずす 制約のない変数を非負数にする(K)
- 解決方法の選択: (E) GRG 非線形
- 解決方法: 滑らかな非線形を示すソルバー問題には GRG 非線形エンジン、線形を示すソルバー問題には LP シンプルクス エンジン、滑らかではない非線形を示すソルバー問題にはエボリューションナリー エンジンを選択してください。

⑤ ソルバーを実行し、定数 a 、 b を求めます。

ソルバーの算出した結果を見てみましょう。

A	B	C	D	E	F	G	H
1	補外法						
2	a	3.06	x	y	ax+b	Q	
3	b	10.05	1	13.3	13.11	0.04	
4			2	15.8	16.17	0.14	
5			3	19.4	19.23	0.03	
6			4	22.3	22.29	0.00	
7					Q _T	0.20	

ソルバーの算出値

ワークシートから、次の解が得られました。

$$a = 3.06, b = 10.05$$

これから、推定式が得られます。

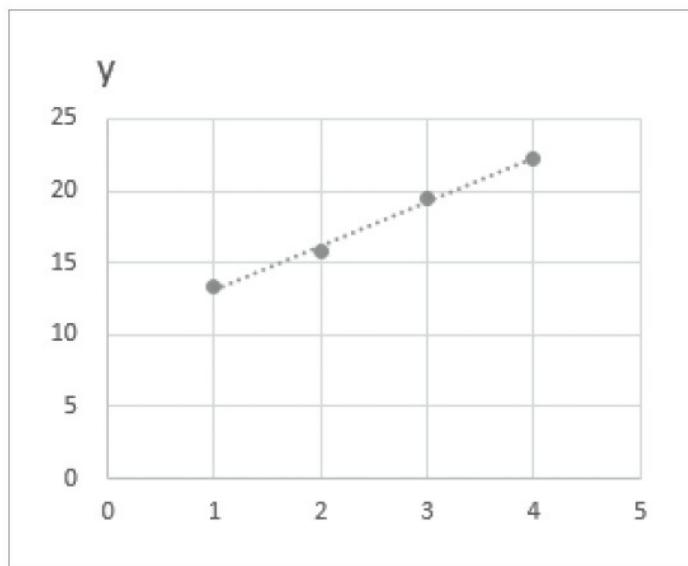
$$y = 3.06x + 10.05 \dots (9)$$

⑥ x が 5 のときの y の値を推定します。

上の式(9)に $x = 5$ を代入して、

$$y = 3.06 \times 5 + 10.05 = 25.35 \text{ (答)}$$

以上が、[例題2]の答です。式(9)のように推定式 $ax + b$ を y と置くと、それは直線を表します。この式(1)のグラフとデータの散布図の関係を図に示しましょう。



推定式(9)を点線で表示。データの散布図とよく重なっている。

Memo 定数と変数

回帰方程式(1)では、 x 、 y を順に説明変数、目的変数と呼び、 p 、 q は定数といいました。ところで、目的関数(5)(すなわち(6))においては p 、 q は変数として扱っています。そうであるからこそ、式(6)の最小値が考えられるわけです。

このように、立場によって、定数、変数は变幻自在です。データから見ると回帰方程式の x 、 y が変数であり、目的関数からみると p 、 q が変数なのです。

参考 Excelの回帰分析

Excelで回帰分析を実際にを行う際は、ここで調べた方法はお勧めできません。Excelにはいくつかの回帰分析専用ツールが備えられているからです。例えば、先の[例題1]では、次の図のように関数から簡単にパラメーターの値が得られます。

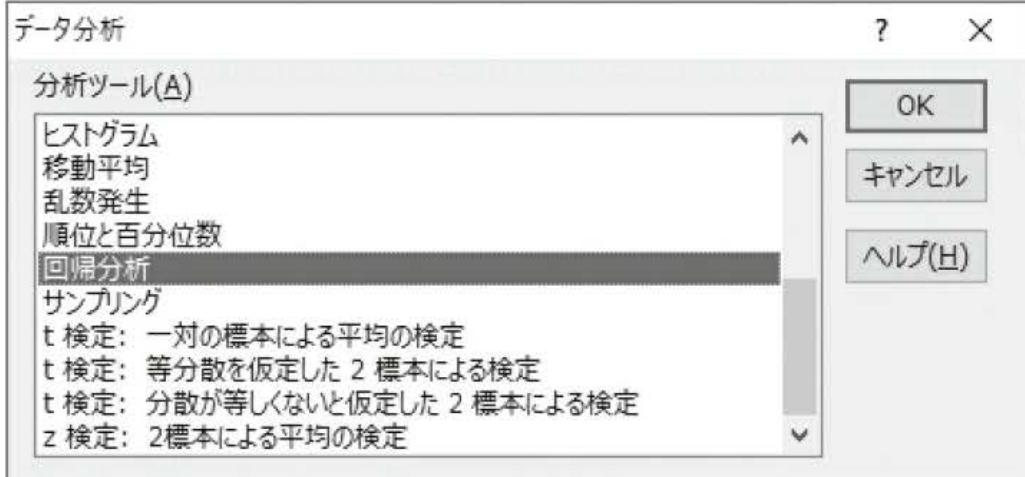
注 このワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「3.xlsx」の中の「4_memo」タブにあります。



C3	A	B	C	D	E	F	G
1			単回帰分析				
2							
3			<i>p</i>	0.41			
4			<i>q</i>	-12.06			
5							
6	番号	身長x	体重y				
7	1	153.3	45.5				
8	2	164.9	56.0				
9	3	168.1	55.0				
10	4	151.5	52.8				
11	5	157.8	55.6				
12	6	156.7	50.8				
13	7	161.1	56.4				

Excelには回帰分析専用の関数がいくつか用意されている。例えば、回帰係数はSLOPE、切片はINTERCEPTという関数が利用できる。

また、回帰分析のための専用ツールも用意されています。「データ」タブにある「データ分析」メニューをクリックし、そこで開かれる下記のボックスから、この機能が利用できます。



3 章

ニューロンモデル

ディープラーニングの基本単位となる人工ニューロン（本書ではニューロンと略します）について調べましょう。動物の神経細胞を数学的に模した単純なニューロンですが、ニューラルネットワークの最も基本となる部分です。

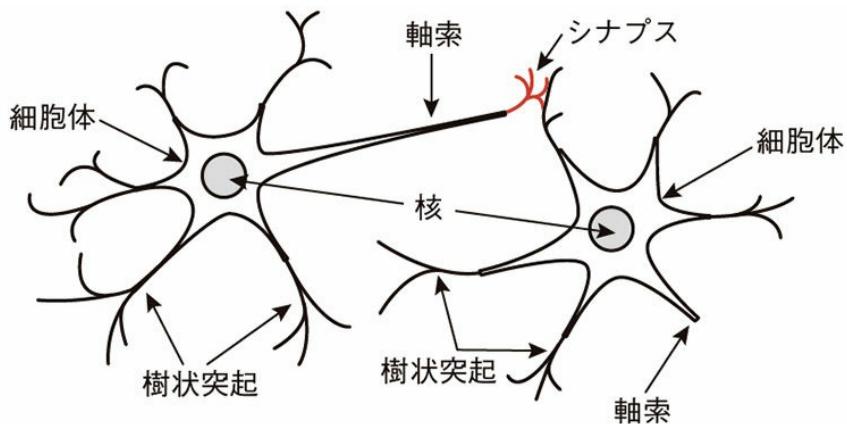
§ 1

神経細胞の働き

ニューラルネットワークは神経細胞をモデル化した人工ニューロンが出発点です。ここでは、人工ニューロンを考えるための出発点として、動物の神経細胞の基本的な働きを調べましょう。

■ 生物のニューロンの構造

動物の脳の中には多数の神経細胞(すなわちニューロン)が存在し、互いに結びついてネットワークを形作っています。すなわち、一つのニューロンは他のニューロンから信号を受け取り、また他のニューロンへ信号を送り出しています。脳はこのネットワーク上の信号の流れによって、さまざまな情報を処理しているのです。

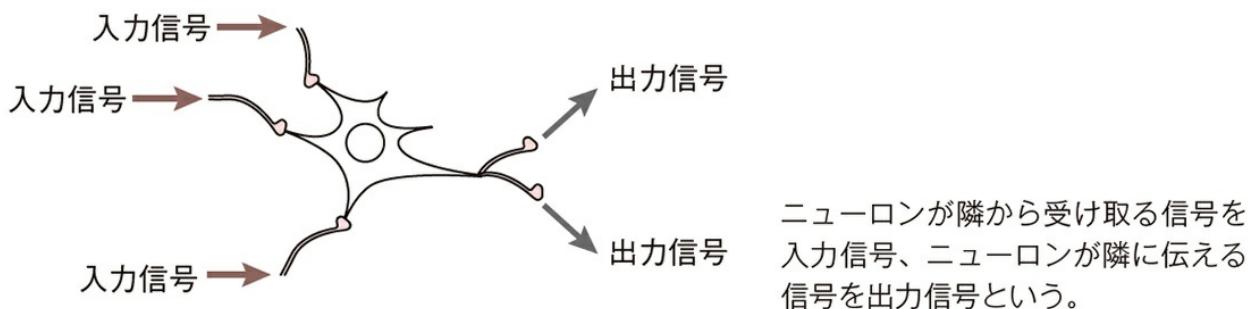


ニューロン(神経細胞)の模式図

神経細胞は、主なものとして細胞体、軸索、樹状突起からなる。樹状突起は他のニューロンから情報を受け取る突起であり、軸索は他のニューロンに情報を送り出す突起である。樹状突起が受け取った電気信号は細胞体で処理され、出力装置である軸索を通って、次の神経細胞に伝達される。ちなみに、ニューロンはシナプスを介して結合し、ネットワークを形作っている。

ニューロンが情報を伝えるしくみをもう少し詳しく見てみましょう。上の図に示すように、ニューロンは細胞体、樹状突起、軸索の3つの主要な部分から構成され

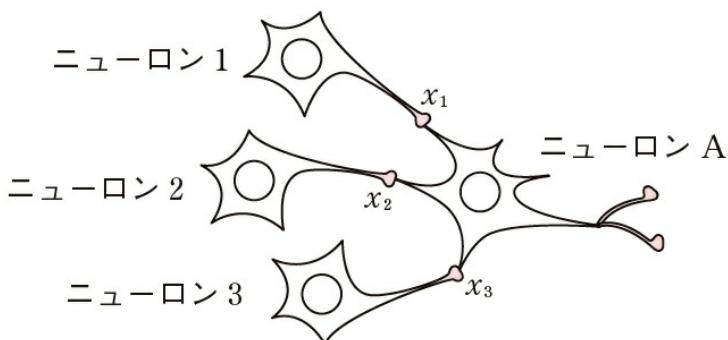
ています。他のニューロンからの信号（入力信号）は樹状突起を介して細胞体（すなわちニューロン本体）に伝わります。細胞体は受け取った信号（入力信号）の大小を判定し、今度は隣のニューロンに信号（出力信号）を伝えます。こんな単純な構造からどうやって「知能」が生まれるのか、大変不思議です。



■ ニューロンの複数入力の処理法

ニューロンは入力信号の大小を判定し、隣に出力信号を伝えるといいましたが、どのように入力信号の大小を判定し、どのように伝えるのでしょうか？

大切なことは、複数のニューロンから受け取る場合、入力信号はその渡されるニューロンごとに扱いが異なるという点です。いま、次図のようにニューロンAがニューロン1～3から信号を受け取るとしましょう。このとき、ニューロンAはニューロン1～3からの信号の和を求めるのですが、その和は重み付きの和になるのです。すなわち、各ニューロンからの信号に重み（weight）を付けるのです。

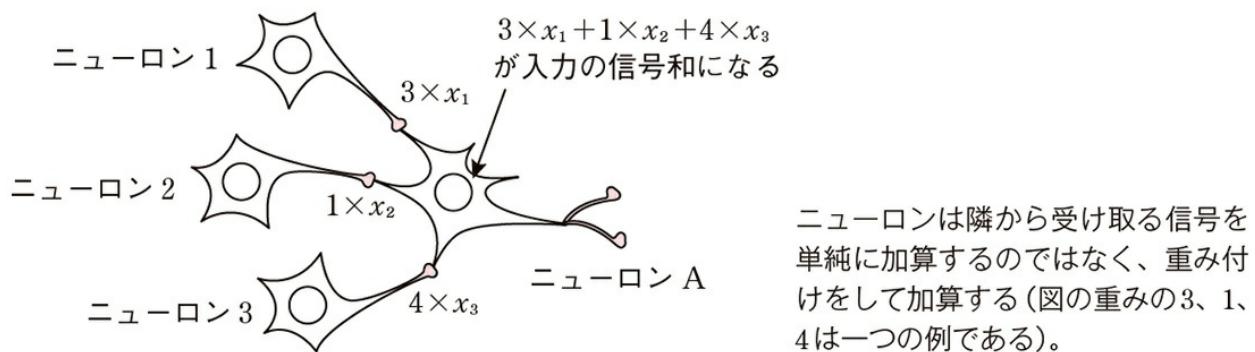


ニューロンAはニューロン1～3から受け取る信号 x_1 、 x_2 、 x_3 に重みを付けて処理する。

例えば、ニューロン1からの信号には「重み」3を、ニューロン2からの信号には「重み」1を、ニューロン3からの信号には「重み」4を付けるとします。この図のように、ニューロン1～3からの信号を x_1 、 x_2 、 x_3 とすると、ニューロンAの受け取る信号和は、次のように重み付きの和として表せることになります。

$$\text{重み付きの和} = 3 \times x_1 + 1 \times x_2 + 4 \times x_3 \dots (1)$$

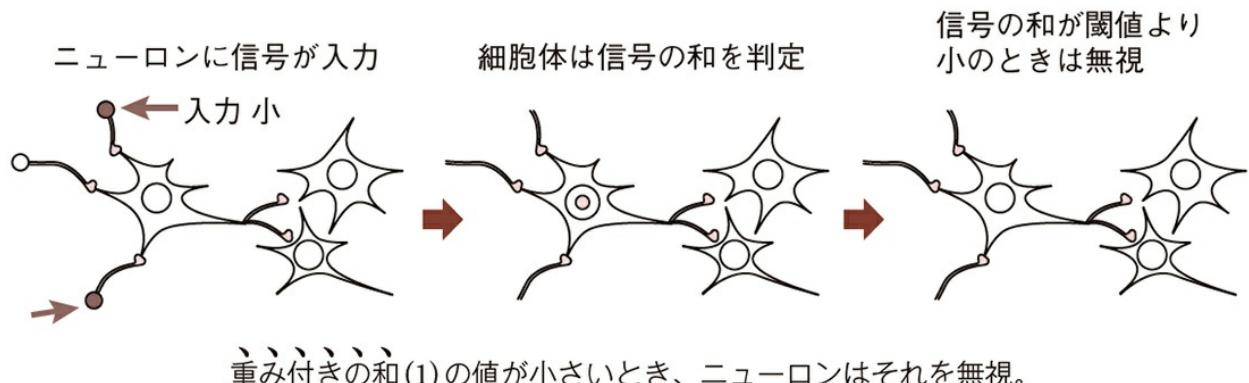
この重みを付けて信号を処理するというしくみこそがニューロンに知性を生じさせる源と考えられます。後にニューラルネットワークを考える際に、この重みをどのように決めるかが本質的な問題になります。



■ 発火

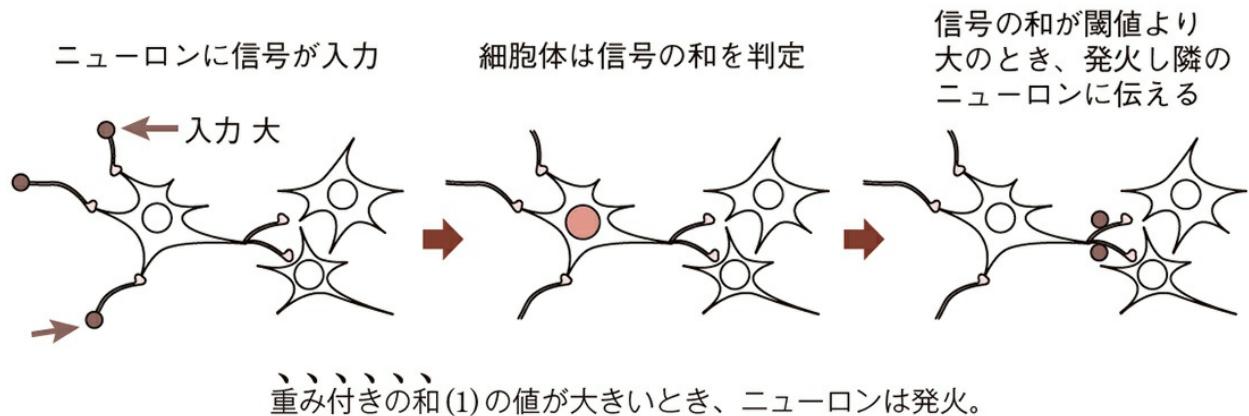
重み付きの和(1)を入力として受け取ったニューロンは、それをどのように処理するかを調べましょう。

複数のニューロンから得た入力の重み付きの和(1)が小さく、そのニューロン固有のある境界値（これを閾値と呼びます）を超えないければ、そのニューロンの細胞体は受け取った信号を無視し何も反応しません。



「小さい信号を無視する」という性質は、生命にとって大切なことです。そうしないと、ちょっとした信号の揺らぎにニューロンは興奮することになり、神経系は「情緒不安定」になってしまいます。閾値はそのニューロンの敏感度を表す個性なのです。

複数のニューロンから得た重み付きの和(1)が大きく、そのニューロン固有のある境界値(すなわち閾値)を超えたとしましょう。このとき、細胞体は強く反応し、軸索をつなげている他のニューロンに信号を伝えます。このようにニューロンが反応することを発火といいます。

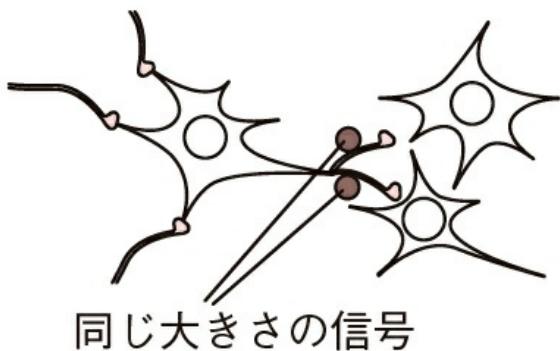


さて、発火したときのニューロンの出力信号はどのようなものでしょうか？面白いことに、それは一定の大きさなのです。たとえ重み付きの和(1)の値が大きくても、出力信号の値は一定なのです。また、該当ニューロンが複数の隣のニューロンへ

軸索をつなげても、隣の各ニューロンに渡す出力信号の値は一定なのです。

さらに面白いことは、この発火によって出力された信号の値はどのニューロンも共通していることです。ニューロンの場所や役割が違っても、その値は共通しています。現代的に言うと、「発火」で生まれる出力情報は0か1で表せるデジタル信号として表現できるのです。

発火したニューロン



発火したニューロンは軸索でつながったすべてのニューロンに同じ大きさの信号を伝える。

§2

神経細胞の働きを数式表現

前節(§1)では、動物の神経細胞の働きを調べました。大切なことは、その働きは簡単な数式で記述できることです。神経細胞の働きを、その簡単な数式に抽象化してみましょう。

■ ニューロンの働きをまとめると

§1ではニューロンの発火のしくみを調べました。それを整理してみましょう。

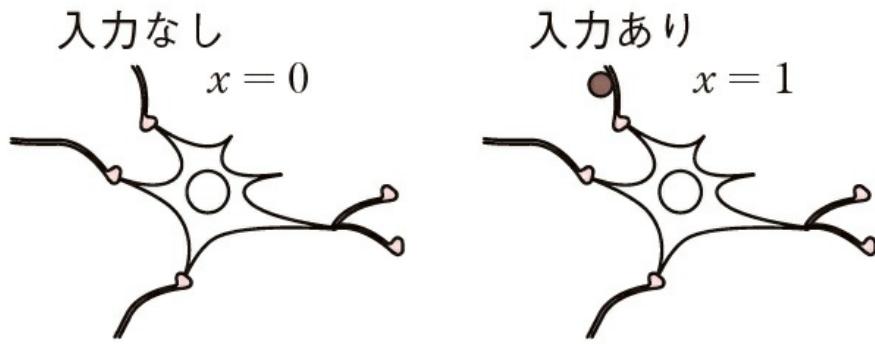
- (i) 隣の複数のニューロンからの重み付きの和の信号がニューロンの入力になる。
- (ii) その和信号がニューロンの固有の値(閾値)を超えると発火する。
- (iii) ニューロンの出力信号は発火の有無を表す0と1のデジタル信号で表現できる。

先にも述べたように、こんな簡単なしくみが組み合わさると「知能」が生まれるのです。

このように整理すると、ニューロンの発火のしくみを数学的に簡単に表現できることがわかります。

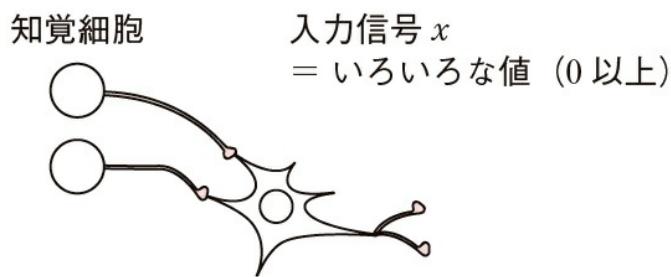
まず入力信号を数式で表現してみましょう。しくみ(iii)から、隣のニューロンから受け取る入力信号は「あり」「なし」の2情報で表せます。そこで、入力信号を変数 x で表すとき、 x は次のように表現できることになります。

$$\begin{cases} \text{入力信号なし} : x = 0 \\ \text{入力信号あり} : x = 1 \end{cases}$$



ニューロンへの入力信号は、デジタル的に
 $x = 0, 1$ で表現される。

注意することは、知覚細胞から直接つながるニューロンはこの限りではないという点です。例えば動物の視覚の場合でいうと、網膜上の視細胞に直接つながるニューロンはいろいろな値の信号を受け取ります。入力信号は感知した信号の大きさに比例したアナログ信号になるからです。

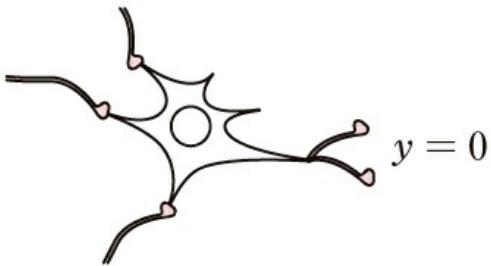


知覚神経に直接つながる神経細胞
 (ニューロン) の受け取る信号 x はア
 ナログ的。

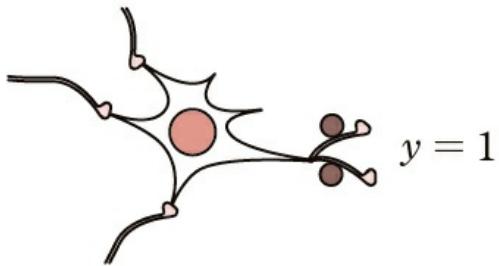
次に出力信号を数式で表現してみましょう。再びしくみ(iii)から、出力信号も発火の有無、すなわち、「あり」「なし」の2情報で表せます。そこで、出力信号を変数 y で表すとき、 y は次のように表現できます。

$$\begin{cases} \text{出力信号なし} : y = 0 \\ \text{出力信号あり} : y = 1 \end{cases}$$

出力なし（発火なし）



出力あり（発火あり）

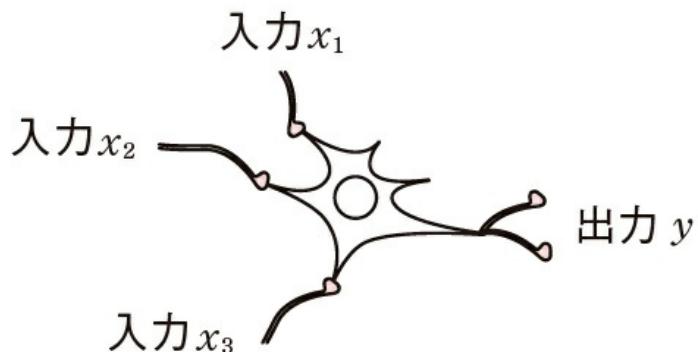


ニューロンの出力信号は、デジタル的に $y = 0$ 、 1 で表現される。この図では出力先が 2 つあるが、出力信号の大きさは同じ。

■ ニューロンの働きを数式で表現

最後に、「発火の判定」を数式で表現してみましょう。これがニューロンの最重要任務です。

具体例として、左隣の 3 つのニューロンからの入力信号を受け取り、右隣の 2 つのニューロンに出力信号を渡すニューロンについて調べます（次図）。

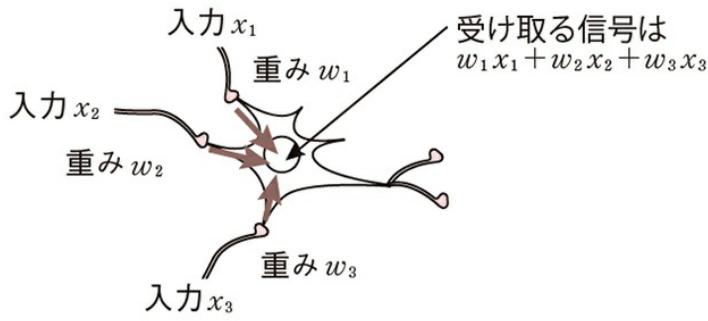


しくみ (i) (ii) から、ニューロンの発火の有無は他のニューロンからの入力信号の和で判定されます。その和の取り方は単純でないはずです。ニューロン間の結びつきには強弱があるからです。前節でも言及したように、この重みを考慮した信号和がニューロンへの入力信号になります。数学的にいふと、入力信号を

各々 x_1, x_2, x_3 で表し、その各々に付く重みを順に w_1, w_2, w_3 とするとき、処理される入力信号の和は次のように表現できます。

重み付きの和 $w_1x_1 + w_2x_2 + w_3x_3 \dots (1)$

注 「重み」は結合荷重、結合負荷とも呼ばれます。



他のニューロンからの入力信号 x_1, x_2, x_3 に対して、該当ニューロンは重み w_1, w_2, w_3 をかけて入力信号としている。それが(1)。

さて、しくみ(ii)から、受け取る信号和が閾値を超えるとニューロンは発火し、越えなければ発火しません。すると、「発火の判定」は式(1)を利用して、次のように表現できます。 θ をそのニューロン固有の閾値として、

$$\left. \begin{array}{l} \text{発火なし } (y = 0) : w_1x_1 + w_2x_2 + w_3x_3 < \theta \\ \text{発火あり } (y = 1) : w_1x_1 + w_2x_2 + w_3x_3 \geq \theta \end{array} \right\} \dots (2)$$

これがニューロン発火の数学的表現です。大変シンプルにまとめられるのです。こんな簡単な条件式で、その活動が表現されるニューロンが、どうして複雑な判断ができるのかを調べるのが、本書のミッションになります。

注 「閾」は英語で threshold。そこで、この値を示すのに頭文字 t に対応するギリシャ文字 θ がよく利用されます。

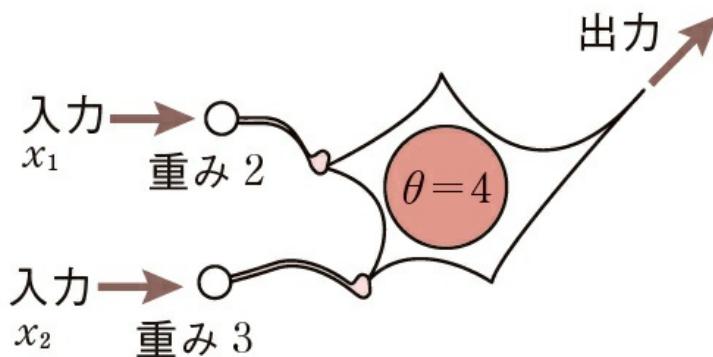
なお、式(2)の下の式の不等号には「=」がついています。この=が式(2)の上の式についている文献もあります。本書ではこれ以上、式(2)については深く触れないでの、このことが問題になることはありません。

例1 2つの入力 x_1, x_2 を持つニューロンを考えます。入力 x_1, x_2 に対する重みを順に w_1, w_2 とし、そのニューロンの閾値を θ とします。

いま、 w_1, w_2, θ の値が順に 2, 3, 4 と与えられたとき、重み付きの和

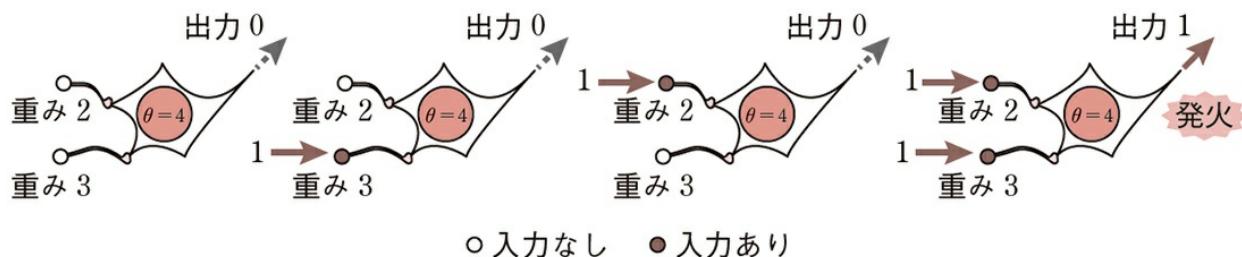
$$w_1 x_1 + w_2 x_2$$

の値とニューロンの発火の有無、そしてニューロンの出力値を求めましょう。



この(例1)の答を表にすると次のようにになります。

入力 x_1	入力 x_2	重み付きの和 $w_1 x_1 + w_2 x_2$	発火	出力
0	0	$2 \times 0 + 3 \times 0 = 0 (< 4)$	なし	0
0	1	$2 \times 0 + 3 \times 1 = 3 (< 4)$	なし	0
1	0	$2 \times 1 + 3 \times 0 = 2 (< 4)$	なし	0
1	1	$2 \times 1 + 3 \times 1 = 5 (> 4)$	あり	1



この(例1)で、出力は 0 と 1 のどちらかであることに留意してください。

■ Excelでニューロンの働きを再現

これからの準備として、この(例1)をExcelで計算してみましょう。

例題1 2つの入力 x_1, x_2 を持つニューロンを考えます。入力 x_1, x_2 に対する重みを順に w_1, w_2 とし、閾値を θ とします。2つの入力 x_1, x_2 を与えたときの出力を求めるワークシートを作成しましょう。ただし、 w_1, w_2, θ は任意に与えられるようにします

注 このワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「3.xlsx」の中の「2_例題1」タブにあります。

解

(例1)に合わせて、次図のようにパラメーター w_1, w_2, θ を順に2、3、4と設定しましょう。発火の条件式(2)から、ニューロンの出力 y (すなわち発火の有無)を簡単に求められます。次の図は入力 (x_1, x_2) が $(1, 1)$ の場合です。

		H3									
1	生物ニューロン										
2	重みと閾値			入力x			出力y				
3		w1	2	x1	1		1				
4		w2	3	x2	1						
5		θ	4								

Excelの1つのセル(セル番地H3)で1つのニューロンが表現できることを確認しましょう。

■ 例題で確認しよう

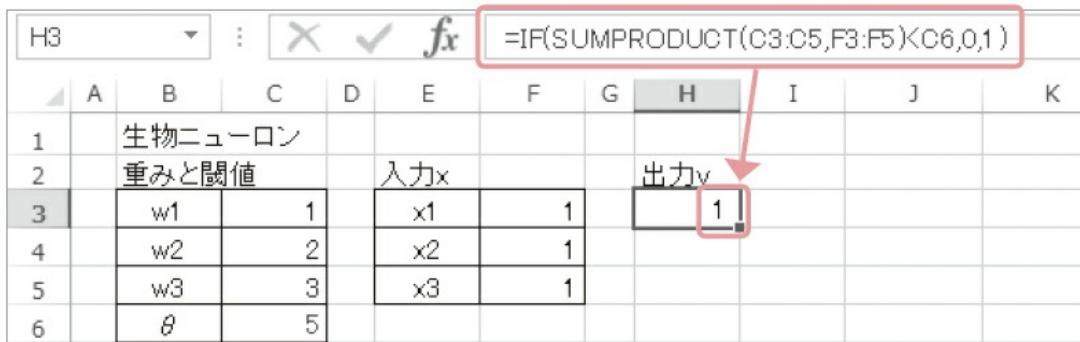
発火の条件式(2)に親しむことはニューロンの理解に大切です。[例題1]をアレンジした次の[例題2]を解いて確認しましょう。具体例として、条件式(2)の解説に利用したニューロンのモデルを考えます。

例題2 3つの入力 x_1, x_2, x_3 を持つニューロンを考えます。各入力に対する重みを順に w_1, w_2, w_3 とし、閾値を θ とします。3つの入力を与えたときの出力を求めるワークシートを作成しましょう。ただし、 w_1, w_2, w_3, θ は任意に与えられるようにします。

注 このワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「3.xlsx」の中の「2_例題2」タブにあります。

解

例として、パラメーターの組(w_1, w_2, w_3, θ)を(1, 2, 3, 5)と設定します。発火の条件式(2)から、ニューロンの出力 y (すなわち発火の有無)を算出します。次の図は入力の組(x_1, x_2, x_3)が(1, 1, 1)の場合です。



	A	B	C	D	E	F	G	H	I	J	K
1	生物ニューロン										
2	重みと閾値										
3	w1	1		x1	1			出力y			
4	w2	2		x2	1						
5	w3	3		x3	1						
6	θ	5									

パラメーターの組(w_1, w_2, w_3, θ)、入力の組(x_1, x_2, x_3)の値をいろいろ

ろと変えて、ニューロンに親しんでみましょう。先にも述べましたが、Excelの1つのセル(セル番地H3)で1つのニューロンが表現できることも確認しましょう。

Memo | ニューロンの計算に役立つSUMPRODUCT関数

ニューロンの計算では、式(1)の右辺の積和の形が頻出します。その計算にはSUMPRODUCT関数を利用するとよいでしょう。次のワークシートは積和 $1 \times 3 + 2 \times 2 + 3 \times 1$ を算出しています。

E3				fx		SUMPRODUCT(B3:B5,C3:C5)	
A	B	C	D	E	F	G	H
1	SUMPRODUCT関数						
2	x	y		積和			
3	1	3		10			
4	2	2					
5	3	1					

積和計算にはSUMPRODUCT関数が便利。

§3

人工ニューロンと活性化関数

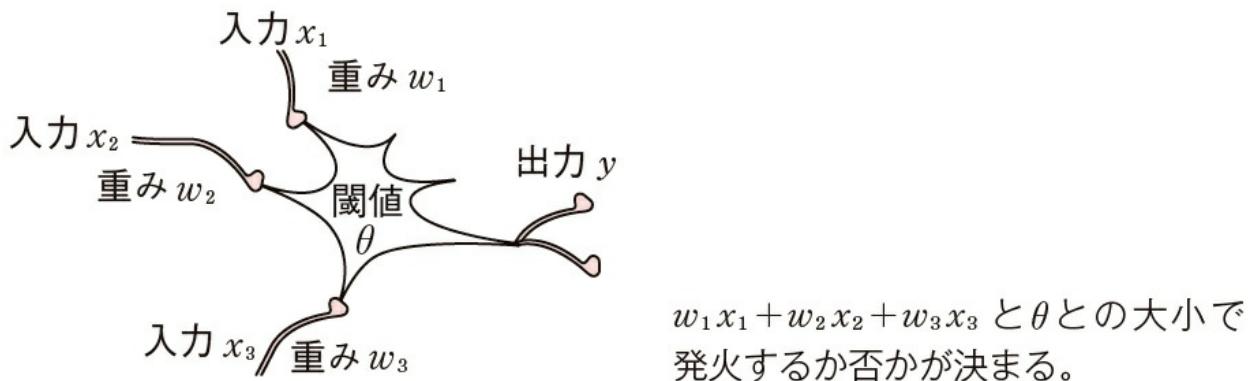
前の節(§2)では、動物のニューロン(神経細胞)の働きを条件式で表現しました。その条件式を関数で表現すると、ニューロンの働きがさらに整理されます。そして、シグモイドニューロンへと進化します。

■ ニューロンの働きをまとめると

前節(§2)ではニューロンの働きを簡単な数式に置き換えました。それを再掲しましょう。ニューロンへの入力を x_1, x_2, x_3 とし、それらに対する重みを順に w_1, w_2, w_3 とするとき、発火の条件式は次のように表せることを調べました。

$$\left. \begin{array}{l} \text{発火なし : } w_1 x_1 + w_2 x_2 + w_3 x_3 < \theta \\ \text{発火あり : } w_1 x_1 + w_2 x_2 + w_3 x_3 \geq \theta \end{array} \right\} \cdots (1)$$

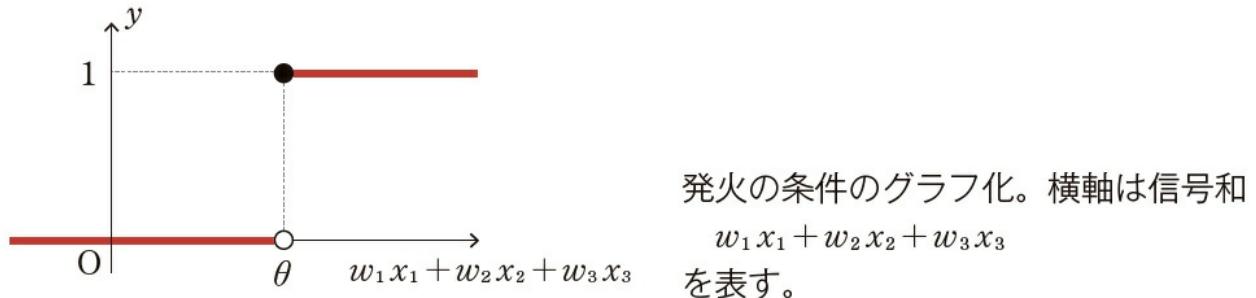
ここで θ はそのニューロン固有の値で閾値です。



■ 発火の条件を関数で表現

発火の条件(1)を関数で表現してみましょう。そのために発火の条件(1)を視覚

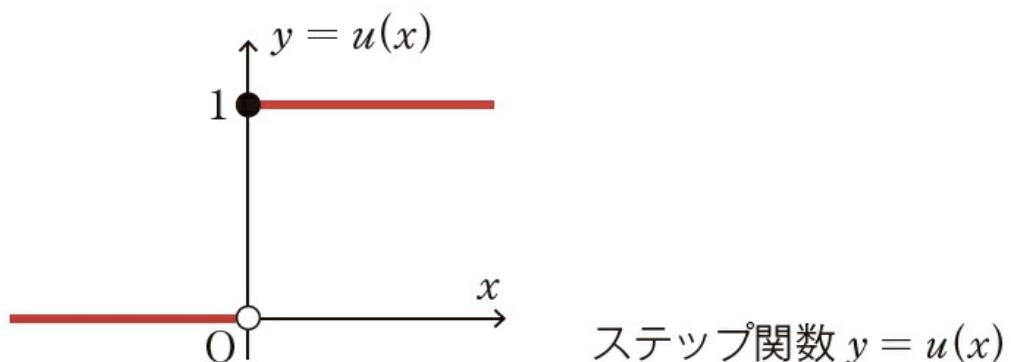
的に表現してみます。ニューロンへの重み付きの和を横軸に、そのニューロンの出力 y を縦軸にとると、発火の条件(1)は次のようにグラフ化できます。なお、出力 y は発火のとき1、発火しないときは0となる尺度を採用しています。



このグラフを関数として表現しましょう。このとき役立つのが次のステップ関数 $u(x)$ です。

$$u(x) = \begin{cases} 0 & (x < 0) \\ 1 & (x \geq 0) \end{cases} \dots (2)$$

ステップ関数のグラフは次のように描けます。



このステップ関数 $u(x)$ を利用すると、発火の条件(1)は次のように簡単に1つの式で表現できます。今後の発展の契機となる大切な式です。

発火の式: $y = u(w_1 x_1 + w_2 x_2 + w_3 x_3 - \theta) \dots (3)$

注 関数 u の引数 $w_1x_1 + w_2x_2 + w_3x_3 - \theta$ を、次節以降では「入力の線形和」と呼び、ローマ字 a で表します。

この式(3)が条件式(1)と同一であることを次の表で確かめてください。

$w_1x_1 + w_2x_2 + w_3x_3$	$w_1x_1 + w_2x_2 + w_3x_3 - \theta$	$u(x)$	y
θ より小	0 より小	0	0 (発火なし)
θ 以上	0 以上	1	1 (発火あり)

■ 人工ニューロン

以上のように数学的に整理すると、ニューロンの働きは一つの簡単な関数式で表されることがわかります。そこで、このように単純化されたニューロンの機能をコンピューター上で実現してみたくなります。それが人工ニューロンです。人工ニューロンとは式(3)を用いてコンピューター上で作動する仮想的なニューロンなのです。

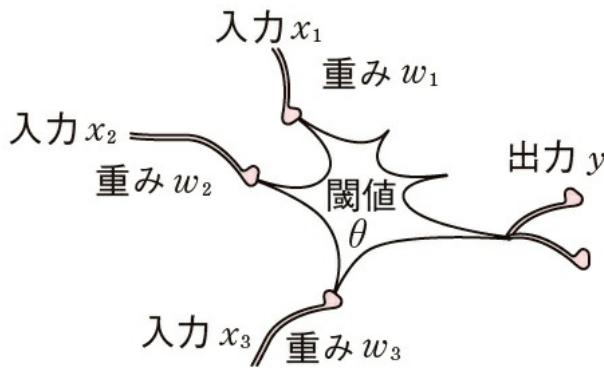
注 人工ニューロンは歴史的に形式ニューロンと呼ぶ文献もあります。なお、次節以降は人工ニューロンのことを単に「ニューロン」と略記します。

人工ニューロンを考えるとき、発火の条件式(3)を表現する関数 $u(x)$ を活性化関数(**activation function**)と呼びます。また、伝達関数(**transfer function**)とも呼ばれます。本書では、前者の「活性化関数」を用いることにします。

■ ニューロンの図を簡略化

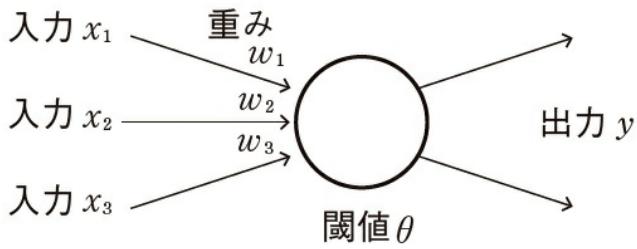
これまでにはニューロンを次図のように表現してきました。少しでもニューロンのイメ

ージに近づけたいのです。



ニューロンのイメージ(入力が3つ、出力が2つの場合)。軸索から出力先が2つに分岐しているが、出力値は同一。

しかし、人工ニューロンを考え、それをネットワーク状にたくさん描きたいときには、この図は不向きです。そこで、次のように簡略化した図を用います。こうすれば、たくさんの中のニューロンを描くのが容易です。



前図のニューロンの略式図。矢の向きで入出力を区別。ニューロンへの出力として2本の矢が出ているが、その値 y は同一値。

■ パーセプトロン

式(3)のように抽象化された人工知能を利用して、何らかの人工知能(AI)を実現しようとする試みが、20世紀中頃に行われました。パーセプトロン・モデルと呼ばれる人工知能です。パーセプトロンとは式(3)で示された人工ニューロンに学習機能を加味したものです。結果としては大きな成功は得られませんでした。理由として、ステップ関数が扱いにくいことが主な原因です。グラフからわかるように、それは不連続関数です。不連続関数は数学の最大の武器である微分法のアイ

デアの恩恵を受けにくいくらいです。このステップ関数を微分のしやすいシグモイド関数に置き換えると、人工ニューロンモデルは飛躍的な発展を遂げることになります。それがディープラーニングの始まりです。本書はこのシグモイド関数を前提として話を進めます。

注 パーセプトロンについて、これ以上の言及は避けます。歴史的には大切ですが、現代のディープラーニングの理解には本質でないからです。

Memo | ステップ関数はExcelにない！

式(2)で定義したステップ関数はExcelにはありません。これを実現するにはIF関数を応用するとよいでしょう。

$$\text{式(2)}: u(x) = \text{IF}(x < 0, 0, 1)$$

実装例を次図に示します。

	A	B	C	D	E
1	ステップ関数				
2	x	y			
3	2		1		

IF関数でステップ関数を実装。

§ 4

ステップ関数からシグモイド関数へ

ニューラルネットワークの基本となる「シグモイドニューロン」について調べます。これは先に調べた生物のニューロンを表すのに用いたステップ関数をシグモイド関数に置き換えた人工ニューロンのモデルです。

■ シグモイド関数

ステップ関数を用いた人工ニューロンの長所は、動物の神経細胞に忠実なモデルということです。しかし、前の節(§3)で調べたように、ステップ関数は滑らかでないという欠点があります。人類の発明した最大の数学の武器の一つである微分法のアイデアが使えないのです。

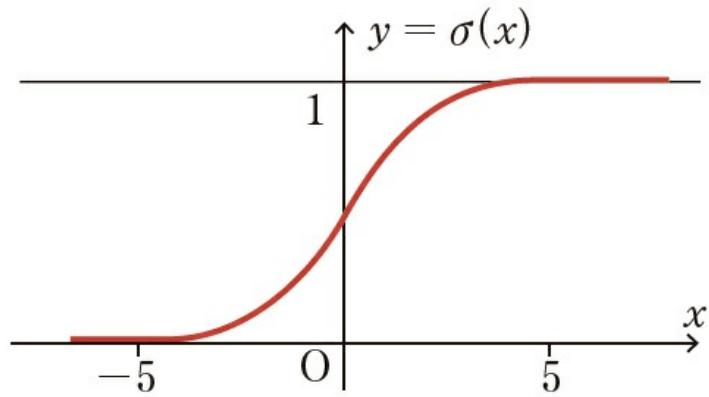
そこで、このステップ関数に似た、しかも滑らかな関数を考えましょう。それがシグモイド関数です。次のように定義されます。

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad \cdots (1)$$

注 2章§1で調べたように、 e はネイピア数(≈ 2.71828)と呼ばれる定数です。 e^x は指数関数です。

Excelでは $\text{EXP}(x)$ と表されます。

シグモイド関数(1)のグラフを見てみましょう。



シグモイド関数のグラフ。ステップ関数に似ているが、滑らかで数学的に扱いやすい。

このグラフからわかるように、シグモイド関数はステップ関数に似ていますが、どこも滑らかで、どの点でも微分が可能です。また、関数値は0と1の間に収まり、その値に割合や度合い、確率など、さまざまな数学的な解釈を施すことができます。

例題1 シグモイド関数 $\sigma(x)$ において、次の関数値をExcelで求めましょう。

- (1) $\sigma(0)$ (2) $\sigma(1)$ (3) $\sigma(-1)$ (4) $\sigma(10)$ (5) $\sigma(-10)$

注 このワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「3.xlsx」の中の「4_例題1」タブにあります。

解

図のように、式(1)をそのままの形で入力すればよいでしょう。

C3	▼	⋮	X	✓	fx	=1/(1+EXP(-B3))
	A	B	C	D	E	F
1	シグモイド関数					
2	x	$\sigma(x)$				
3	0	0.50				

指数関数 e^x の計算には EXP 関数を利用。

このワークシートで、 x の欄の数値を変更することで、次の概数が計算されます。

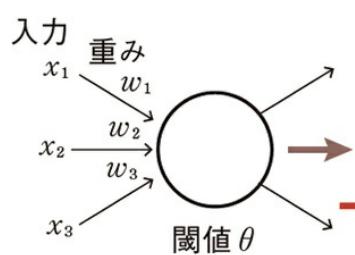
$$\sigma(0)=0.50, \sigma(1)=0.73, \sigma(-1)=0.27, \sigma(10)=1.00, \sigma(-10)=0.00$$

この[例題1]で注意すべき点は、 x の値が10にもなると、シグモイド関数 $\sigma(x)$ の値は1とみなせることです。また、 x の値が-10にもなると、シグモイド関数 $\sigma(x)$ の値は0とみなせることです。シグモイド関数がステップ関数のよい代替関数になっていることが納得できます。

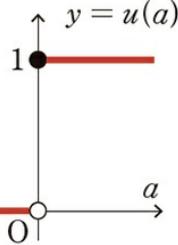
■ シグモイドニューロン

パーセプトロンのステップ関数をシグモイド関数に置き換えた人工ニューロンをシグモイドニューロンといいます。すなわち、シグモイドニューロンとは活性化関数にシグモイド関数を採用したニューロンのことです。

パーセプトロンモデル

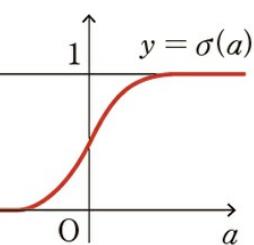
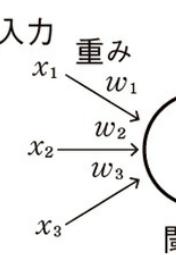


ステップ関数



シグモイドニューロン

シグモイド関数



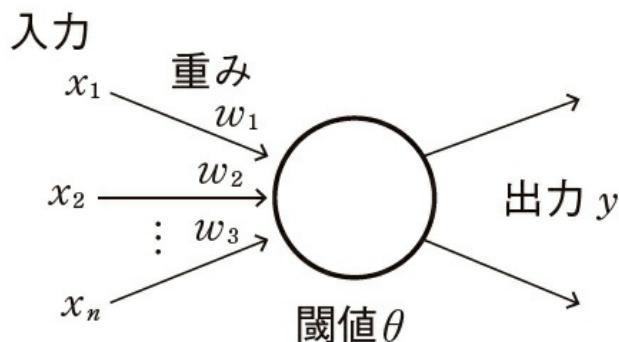
シグモイドニューロンは本書の中心的なニューロンモデルとなります。今後何も注釈を付けないとき、ニューロンといえばこのシグモイドニューロンを指します。そこで、その働きをここでまとめておきましょう。

入力信号 x_1, x_2, \dots, x_n (n は自然数)を考え、各入力信号には重み w_1, w_2, \dots, w_n が与えられたとする。閾値を θ とするとき、ニューロンの出力 y は

$$y \sigma(a) \dots (2)$$

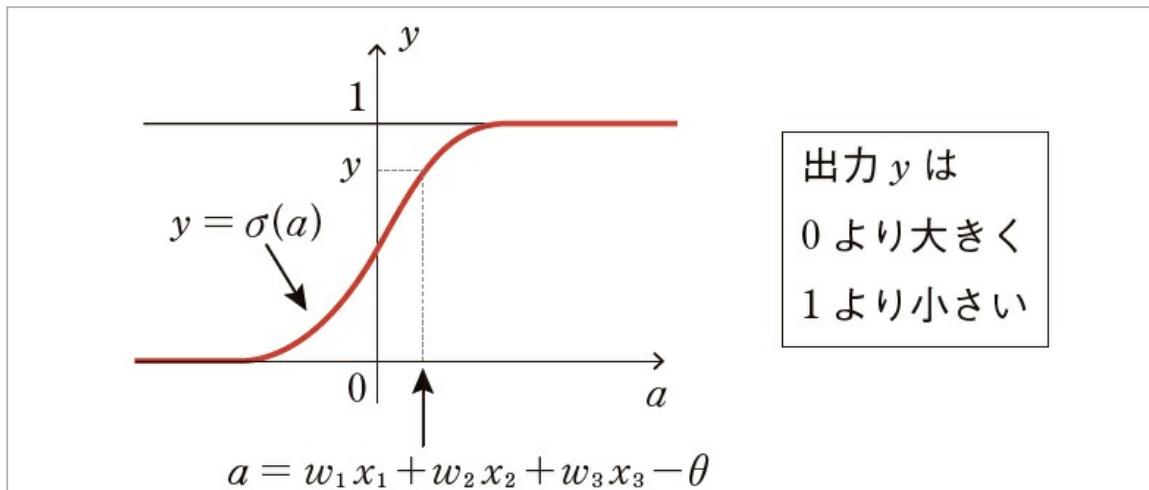
ここで、 σ はシグモイド関数であり、 a は「入力の線形和」と呼ばれ、次のように定義される。

$$a = w_1 x_1 + w_2 x_2 + \dots + w_n x_n - \theta \dots (3)$$



■ シグモイドニューロンの出力値は0と1の間の数

先の関数(1)のグラフの個所でも言及しましたが、シグモイドニューロンの出力値は0と1の間の任意の数になります。



このことは先に調べたパーセプトロンとは大きく異なります。パーセプトロンで用いたニューロンの出力は0、1の2値でした。それに対して0と1の間の任意の数を出力するシグモイドニューロンは、「神経細胞の発火」というデジタルなイメージとはかけ離れています。あえて解釈すれば、ニューロンの「活性度」、通俗な言葉を用いると「興奮度」を表すと解釈できます。いずれにせよ、シグモイドニューロンの出力は現実のニューロン(神経細胞)とは大きく異なるのです。

以上のことからわかるように、シグモイドニューロンを採用するということは、生物学的な神経モデルからは大きく離反することを意味します。



シグモイドニューロンの出力値はニューロンの「活性度」「興奮度」と解釈できる。

■ Excelでニューロンの働きを再現

それでは、具体的にシグモイドニューロンの出力の計算をしてみましょう。1つのセルが1つのニューロンを表現できることを確かめます。

例題2 2つの入力 x_1, x_2 を持つシグモイドニューロンを考えます。入力 x_1, x_2 に対する重みを順に w_1, w_2 とし、閾値を θ とします。2つの入力 x_1, x_2 を与えたときの出力を求めるワークシートを作成しましょう。ただし、 w_1, w_2, θ は任意に与えられるようにします。

注 このワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「3.xlsx」の中の「4_例題2」タブにあります。

解

次のワークシートはパラメーター(w_1, w_2, θ)に(2, 3, 4)を設定しています。式(2)、(3)からシグモイドニューロンの出力が求められますが、次の図は入力(x_1, x_2)に(1, 1)を考えた場合です。

1		シグモイドニューロン									
2		パラメータ		入力							
3		w1	2		x1	1		出力			
4		w2	3		x2	1					
5		θ	4								

例題2 3つの入力 x_1, x_2, x_3 を持つシグモイドニューロンを考えます。各入力に対する重みを順に w_1, w_2, w_3 とし、閾値を θ とします。3つの入力

を与えたときの出力を求めるワークシートを作成しましょう。ただし、 w_1, w_2, w_3, θ は任意に与えられるようにします。

注 このワークシートは、ダウンロードサイト（→卷頭「Excelサンプルファイルのダウンロードについて」）に掲載されたファイル「3.xlsx」の中の「4_例題3」タブにあります。

解

次の図はパラメーター(w_1, w_2, w_3, θ)に(1, 2, 3, 4)を設定しています。式(2)、(3)から、シグモイドニューロンの出力が求められますが、次図は入力(x_1, x_2, x_3)に(1, 1, 1)を考えた場合です。

	A	B	C	D	E	F	G	H	I	J	K
1		シグモイドニューロン									
2		パラメータ		入力				出力			
3		w1	1		x1	1		0.8808			
4		w2	2		x2	1					
5		w3	3		x3	1					
6		θ	4								

パラメーターの組(w_1, w_2, w_3, θ)、入力の組(x_1, x_2, x_3)の値をいろいろと変えて、シグモイドニューロンに親しんでみましょう。Excelの1つのセル(番地H3)で1つのシグモイドニューロンが表現できることも確認しましょう。

Memo シグモイドニューロンを一般化

活性化関数の候補はシグモイド関数だけではありません。疑似的な発火を実現できるようなグラフの形を持つ関数なら何でもよいのです。そこで、計算速度の速いReLUモデルが有名です。5章の最後で、この関数についても

触れることにします。

4 章

ニューラル ネットワークの しくみ

3章で調べた（人工）ニューロンを用いてネットワークを作成しましょう。簡単な「○」「×」2文字を識別することを具体例として話を進めます。Excelで計算すると、1つのセルが1つのニューロンに該当するので、ニューラルネットワークのしくみが理解しやすいでしょう。

(注) 本章からは人工ニューロンをニューロンと略します。

§ 1

読み物としてのニューラルネットワークのしくみ

ニューラルネットワークが画像の識別をどのように実現するのかを、人の動きに例えて調べてみましょう。全体を大まかに見渡しておくことは、次節以降の数値的な説明の理解を容易にするでしょう。

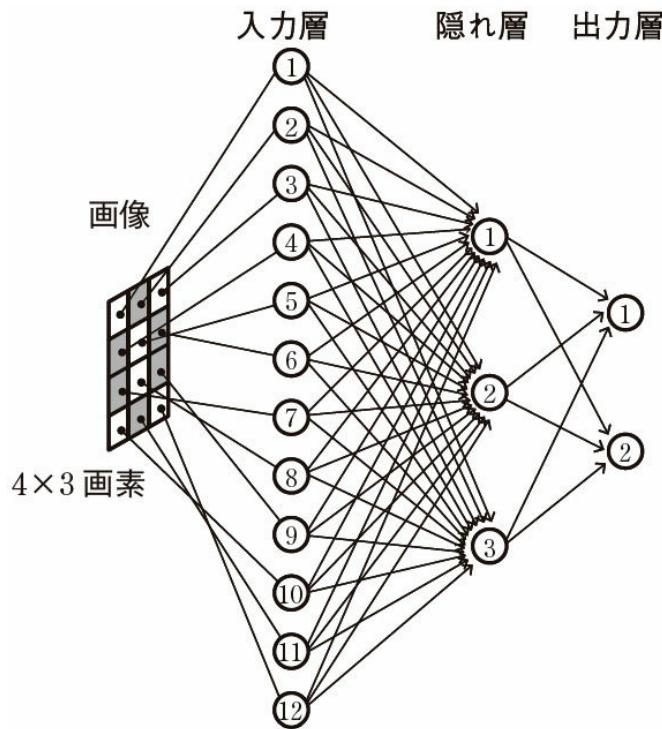
注 本書は階層型ニューラルネットワークと呼ばれるタイプしか扱いません。

■ 問題の明確化

本章では、次に示す[テーマ I]を具体的に調べることで、ニューラルネットワークのしくみを解説していきます。

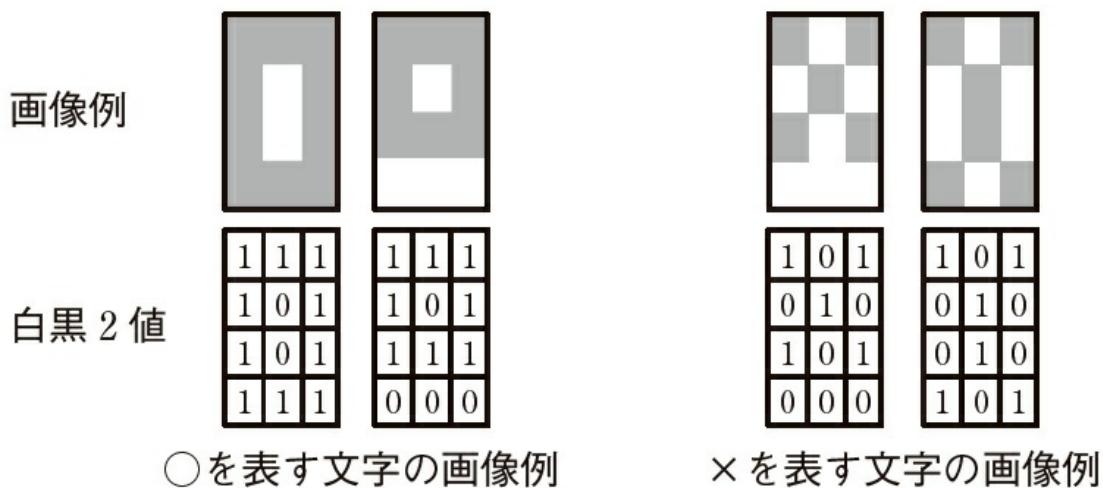
■ テーマ I 4×3画素の白黒2値画像として読み取った手書きの文字「○」、「×」を識別する畳み込みニューラルネットワークを作成しよう。

本章はこの課題に対するニューラルネットワークとして、次の形を採用します。



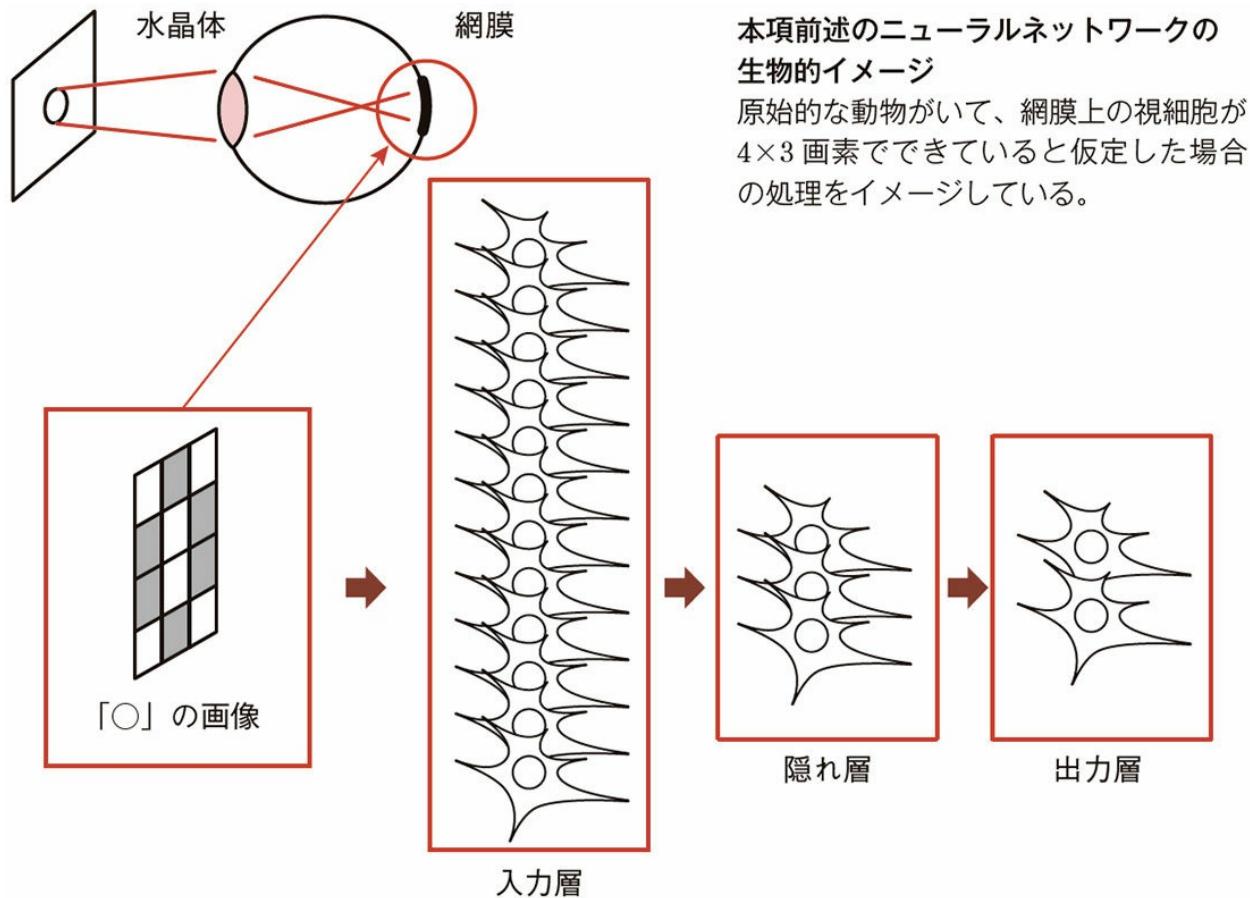
本章で調べるニューラルネットワーク。図の中の矢に対応する重みと、各ニューロンの閾値を決めることが大きな目的となる。なお、本節では、左の図のようにニューロンに番号を付ける。

上記[テーマ I]に示す「 $4 \times 3 = 12$ 画素の白黒2値画像」の手書き文字「○」、「×」とは、次に例示するような極めて単純な画像です。



本節は本項前述に示したニューラルネットワークが上記の文字を識別するしくみを、人の行動に例えて、解説することにします。

なお、次図は本項前述のニューラルネットワークを原始生物の視神経システムに見立てた図です。



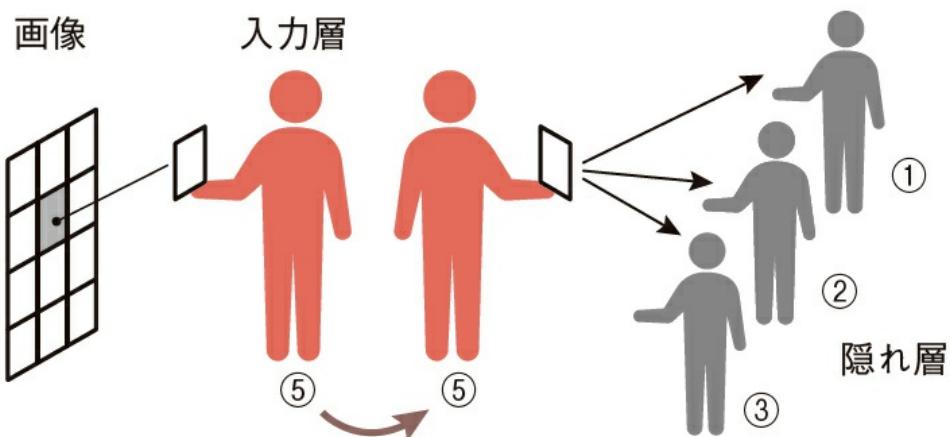
■ 各層のニューロンに役割を与える

解説しやすいように、ネットワークを構成するニューロンに番号を付けましょう。先に示したニューラルネットワークの図(本節の「問題の明確化」)のように、各層のニューロンについて、上から順に1、2、3と番号を振ります。また、縦の列にも名称を付けます。画像の隣の層を入力層、中間の層を隠れ層、そして右側の層を出力層と呼ぶことにします。

では、ニューラルネットワークの動作のしくみを人間の営みに置き換えて考えてみ

ましょう。最初に入力層について考えます。

この層にある12個のニューロンは、ネットワークへ画像情報を運ぶ「運搬係」の役割を担います。一人ひとりのスタッフは画像の一つひとつの画素を担当し、画素情報を加工せず、そのまま隠れ層の全員に報告する役割を担います。換言すれば、入力層のニューロンは入力信号を中間層に伝えるだけで、何の処理も行いません。彼ら運搬係の各スタッフは、ニューロン番号と同じ①～⑫で呼ぶことにします。



入力層の各ニューロンは信号の「運搬係」。この図は⑤番の運搬係を表す。運搬係のスタッフは受け持ちの画素情報をそのまま隠れ層の係全員に報告する。

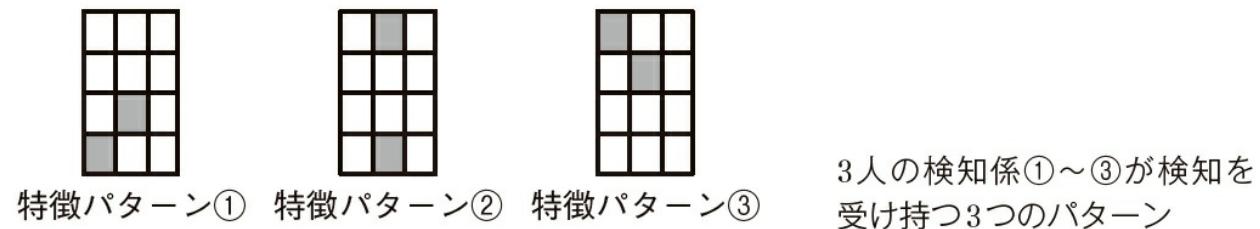
次に、隠れ層について考えます。

この層にある3つのニューロンは「検知係」の役割を担います。入力層から報告される画像パターンの中に受け持ちのパターンが含まれているかを調べ、その含有率を上の層に報告する役割を担います。

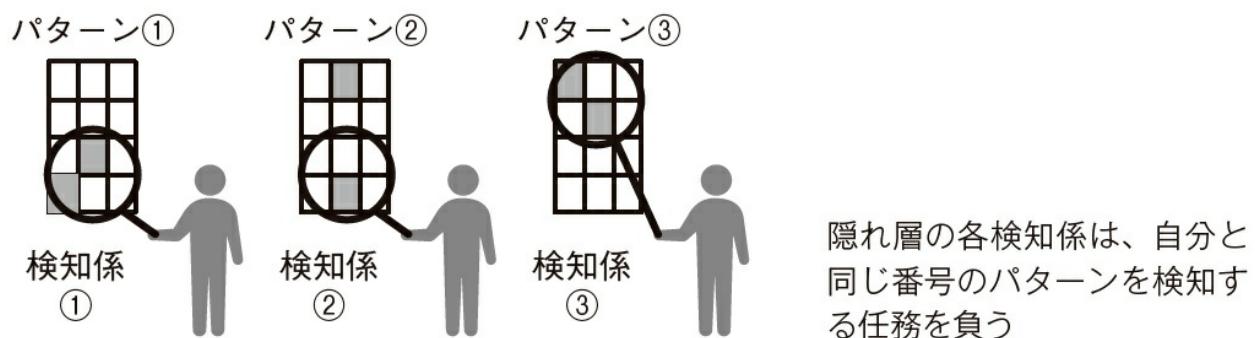
注 話を分かりやすくするために、「受け持ちのパターン」は予めわかっているものとしますが、実をいうとこの決定が本章の目的の一つなのです。ちなみに、含有率という言葉を用いましたが、これはイメージ的な表現であり、厳密な意味ではありません。

各検知係の受け持つパターンを「特徴パターン」と呼ぶことにしましょう。ここで

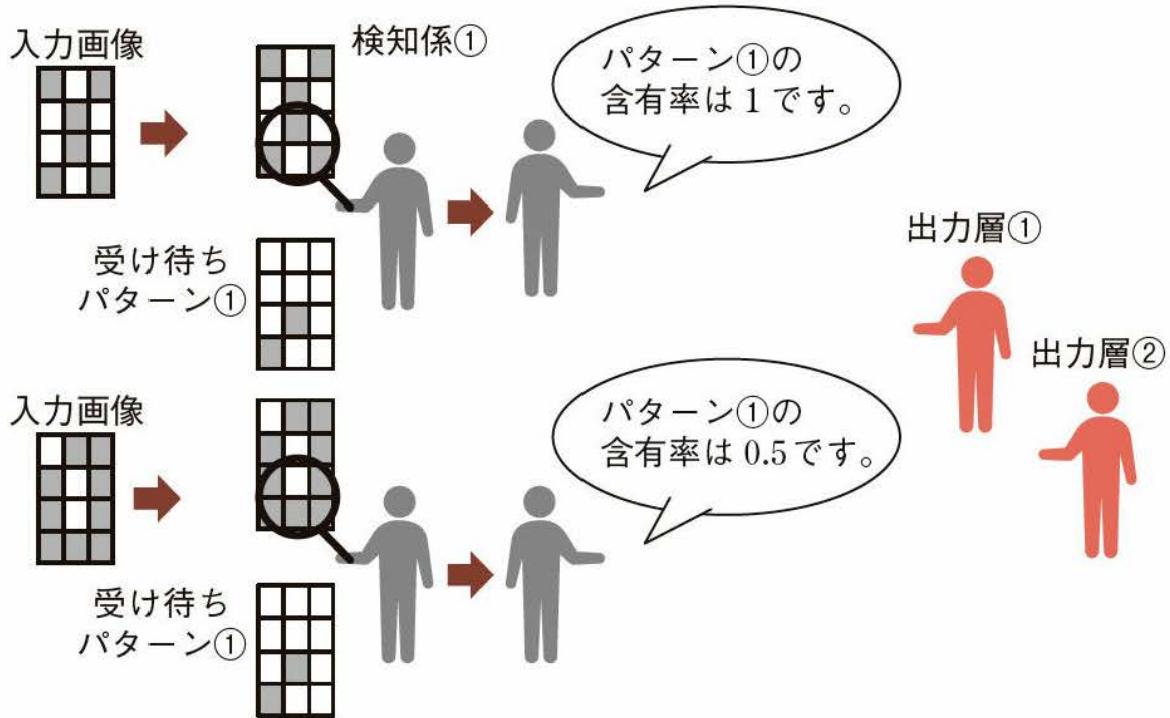
は、次のパターンを仮定することにします。



ニューロン番号①～③に該当する検知係①～③は、上記の特徴パターン①～③の検知を担当することにします。



次の図は、特徴パターン①を検出する役割を担う検知係①の働きを示しています。

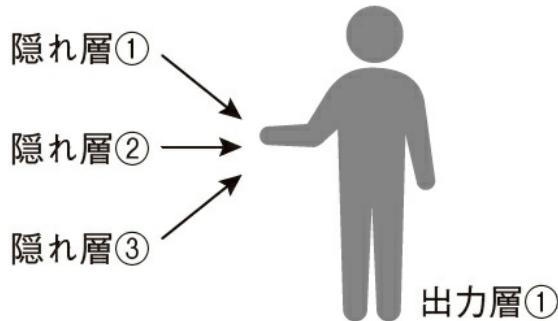


隠れ層の検知係①は受け待ちの特徴パターン①が画像にどれくらい含まれているかを調べ、その含有率を0～1の間の値で出力層に伝える。

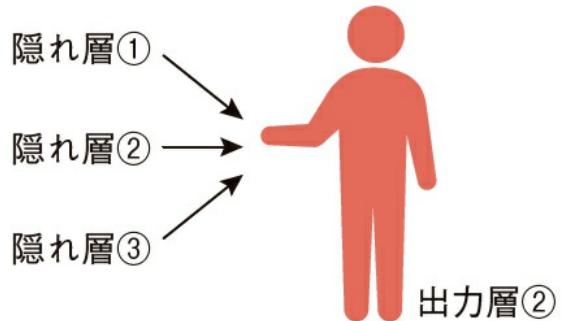
最後に出力層について調べましょう。

この層のニューロンは「判定係」の役割を担います。この係の各スタッフもニューロン番号と同じ①、②で呼ぶことにします。判定係①は文字「○」の判定を分担します。判定係②は文字「×」の判定を分担します。隠れ層の「検知係」から報告される3つの異なる特徴パターンの含有率を勘案して、判定係①は文字「○」である確信度を0と1の間の数値で表現します。判定係②は文字「×」である確信度を0と1の間の数値で表現します。

「○」と判定する判定係①



「×」と判定する判定係②



判定結果は、その判定の確信度に応じて0から1の間の値で出力する。

注 確信度という言葉を用いましたが、これはイメージ的な表現であり、厳密な意味ではありません。

■ ニューロン1個は知能を持たない！

「運搬係」12人、「検知係」3人と「判定係」2人の総勢17人の役割を調べました。「運搬係」は画素信号を隠れ層のスタッフ全員にそのまま届ける係、「検知係」は運搬係からもらった信号の中にある受け持ちの特徴パターンの含有率を判定係に報告する係、そして最後の「判定係」は検知係からもらった情報から文字「○」か文字「×」かの確信度を出力する係です。



注意すべきことは、各ニューロンを「人」に例えたからといって、そのニューロンは人のような知能を有してはいないことです。3章で調べたように、各ニューロンは単純

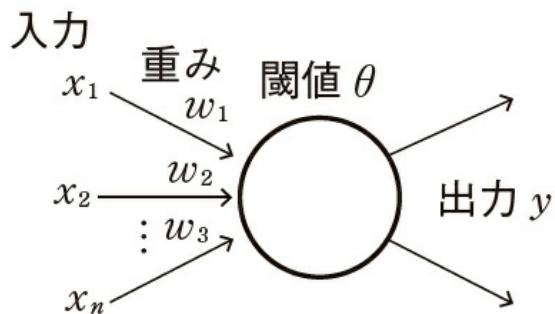
に次の働きをするだけです。

入力信号 x_1, x_2, \dots, x_n (n は自然数)を考え、各入力信号には重み w_1, w_2, \dots, w_n が与えられたとする。閾値を θ とするとき、ニューロンの出力 y は

$$y = \sigma(a) \dots (1)$$

ここで、 σ はシグモイド関数であり、 a は「入力の線形和」と呼ばれ、次のように定義される。

$$a = w_1x_1 + w_2x_2 + \dots + w_nx_n - \theta \dots (2)$$



では、どうやって、こんな単純なニューロン(すなわち17人のスタッフ)が寄り集まって文字識別という高度な処理が可能になるのでしょうか。その秘密は層間の各係のスタッフの関係(すなわち重み)の大小にあります。層ごとにそのしくみを調べてみましょう。

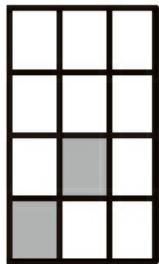
■ 隠れ層の役割は特徴抽出

入力層は単にネットワークの窓口です。受け取った入力値をそのまま隠れ層に渡します。その隠れ層の働きを考えることにします。

例として、前の項(「各層のニューロンに役割を与える」)に図示した隠れ層の

「検知係」①について調べます。この検知係①は、読み取った画像の中に、次図に示した特徴パターン①が含まれるかどうかを調べ、その含有率を0から1の間に数値化する役割を担います。

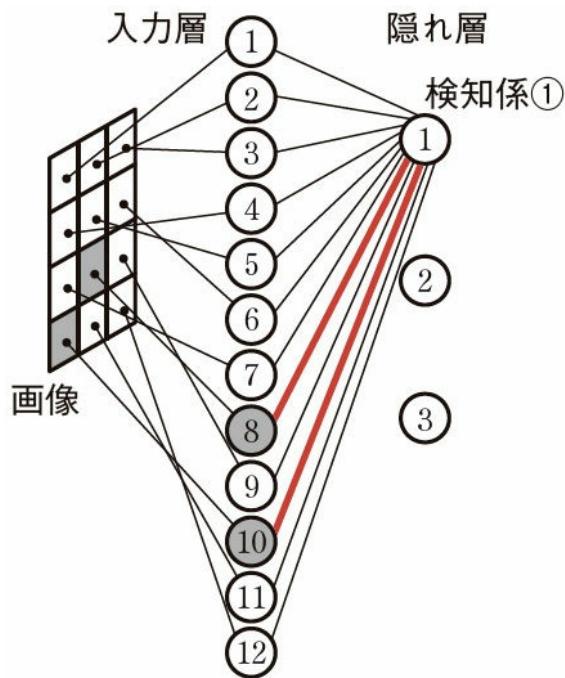
パターン①



検知係①の検知すべきパターン①。

では、どうやって含有率を算出するのでしょうか？ その秘密は入力層の運搬係と検知係①とを結ぶ矢の太さ(すなわち重みの大きさ)にあります。

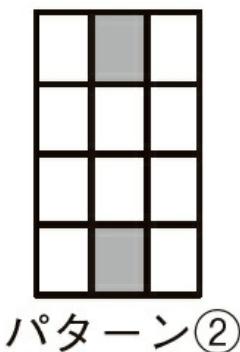
次の図を見てください。この図に示すように、入力層の運搬係⑧⑩と隠れ層の検知係①とを結ぶ矢を太くし(すなわち重みを大きくし)、検知係①に向けられた他の矢を細くして(重みを小さくして)みましょう。そうすれば、パターン①が画像に含まれているとき、「入力の線形和」(2)からわかるように、検知係①に伝わる信号は大きくなります。また、画像パターン①が画像に含まれていないとき、検知係①伝わる信号は小さくなります。



パターン①が画像に含まれるかどうかを調べるには、入力層のスタッフ⑧⑩と隠れ層の検知係①とが太い矢で結ばれていればよい。すなわち、隠れ層①のニューロンは入力層⑧⑩の入力の「重み」を大きくし、他の重みを小さくすればよい。

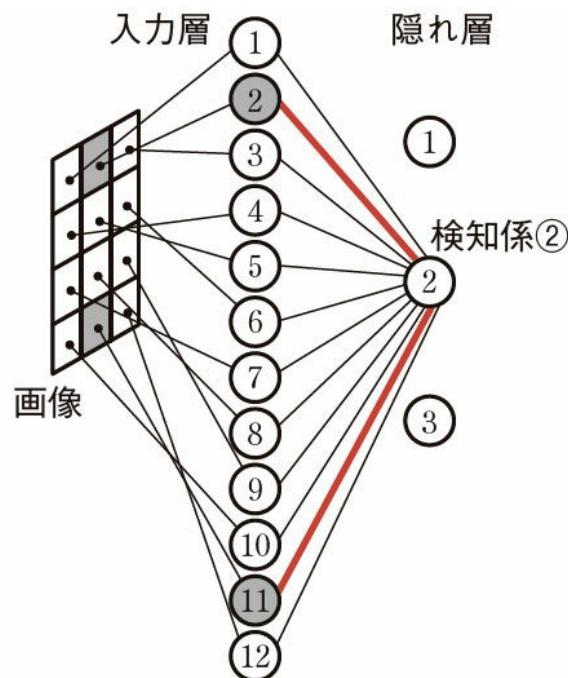
この例からわかるように、「入力の線形和」(2)の中の「重み」を調節することで、画像において、検知係が担当する特徴パターンの含まれ具合が判明し、活性化関数(1)で含有率に変換できます。

例1 前の項(「各層のニューロンに役割を与える」)に示した特徴パターン②(次図)を検知する「検知係」②については、どうすればこの特徴パターン②の含有率を検知係②は算出できるのか、調べてみましょう。



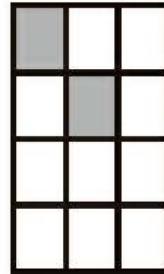
次図を見てください。入力層の運搬係②⑪と検知係②とを結ぶ矢の重みを大

きくし、検知係②に向けられた他の矢を細くして(重みを小さくして)みましょう。そうすれば、このパターン②が画像に含まれているとき、「入力の線形和」(2)から検知係②に伝わる信号は大きくなります。また、画像パターン②が画像に含まれていないとき、検知係②に伝わる信号は小さくなります。こうすることで、このパターン②が画像に含まれる大小を算出できます。この大小によって、活性化関数(1)でパターン②の含有率が求められます。これが(例1)の解答です。



パターン②が画像に含まれるかどうかを調べるには、入力層のスタッフ②⑪と隠れ層の検知係②とが太い矢で結ばれていればよい。すなわち、隠れ層②のニューロンは入力層②⑪の入力の「重み」を大きくし、他の重みを小さくすればよい。

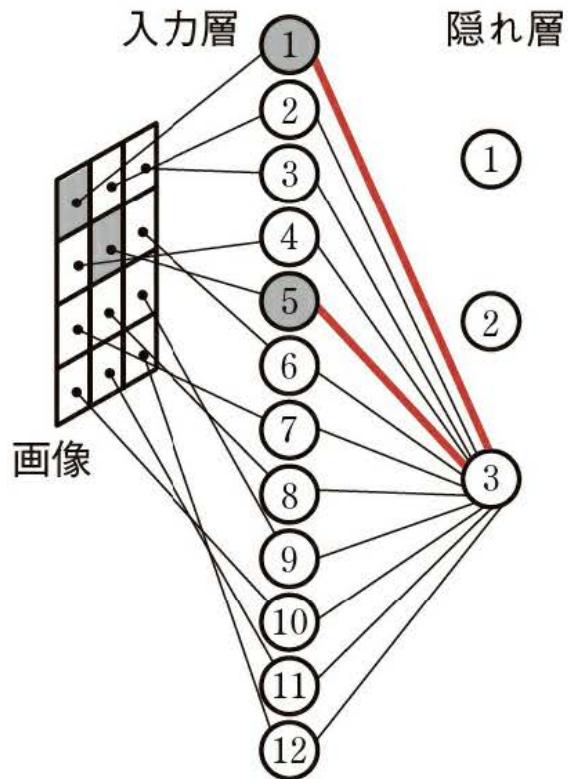
例題1 前の項(「各層のニューロンに役割を与える」)に示した特徴パターン③(次図)を検知する「検知係」③について、どうすればこの特徴パターン③の含有率を検知係③は算出できるのか、調べてみましょう。



パターン③

解

入力層の運搬係①⑤と検知係③とを結ぶ矢の重みを大きくし、他の矢を細くして(重みを小さくして)みましょう。そうすれば、パターン③が画像に含まれているとき、「入力の線形和」(2)から検知係③に伝わる信号は大きくなります。また、画像パターン②が画像に含まれていないとき、検知係②に伝わる信号は小さくなります。こうして、パターン③が画像に含まれる大小を算出できます。(答)



このようにして、検知係は、入力層と結ぶ「重み」の大小によって、与えられたパターンの含有率を算出できるのです。単純な人工ニューロンが、文字画像の中の情報を調べるということは、このように単純な操作によってなされます。

ところで、検知係が目的のパターンの含有率を算出することは、画像に含まれる特徴を抽出すると換言できます。このことをニューラルネットワークでは「隠れ層は特徴抽出する役割を担う」と表現します。

■ 出力層の「判定係」は受け持ち文字の確信度の出力

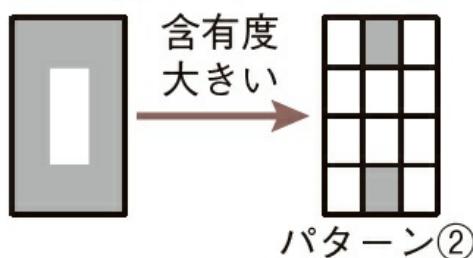
最後に出力層の判定係の働きを見てみましょう。

出力層の「判定係」は「検知係」から報告される特徴パターンの含有率から、自分が受け持つ文字かどうかの確信度を0と1の間に数値化します。先に定めた

ように、判定係①は「○」の確信度を、判定係②は「×」の確信度を数値化することにします。

ところで、判定係①が「○」と判定する役割を担っているということは、隠れ層の検知係②と太い矢を持つことを意味します。手書き文字「○」の文字には特徴パターン②が含まれている可能性が高いからです。

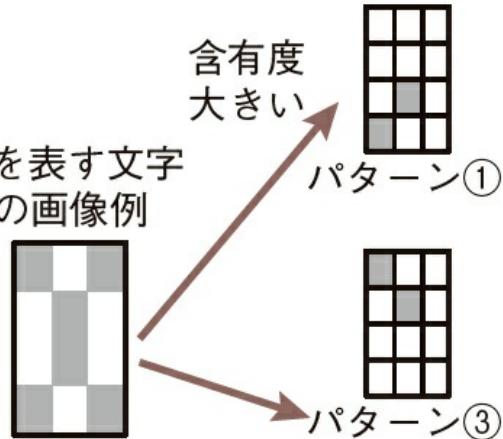
○を表す文字
の画像例



手書き文字「○」の文字には特徴
パターン②が含まれている可能
性が高い。

判定係②が「×」と判定する役割を担っているということは、隠れ層の検知係①③と太い矢を持つことを意味し、検知係②と細い矢をもつことを意味します。手書き文字「×」の文字にはパターン①③が含まれている可能性が高いからです。

×を表す文字
の画像例

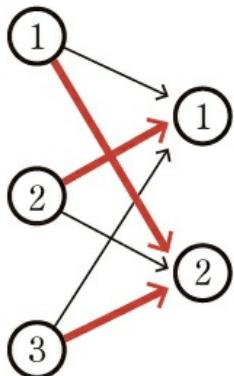


手書き文字「×」の文字にはパターン
①③が含まれている可能性が高い。

隠れ層のときと同様、太い矢とは、ニューロンの世界でいうと、「重み」が大きいことを意味します。細い矢ということは、式(2)の重みを小さくすることになります。隠

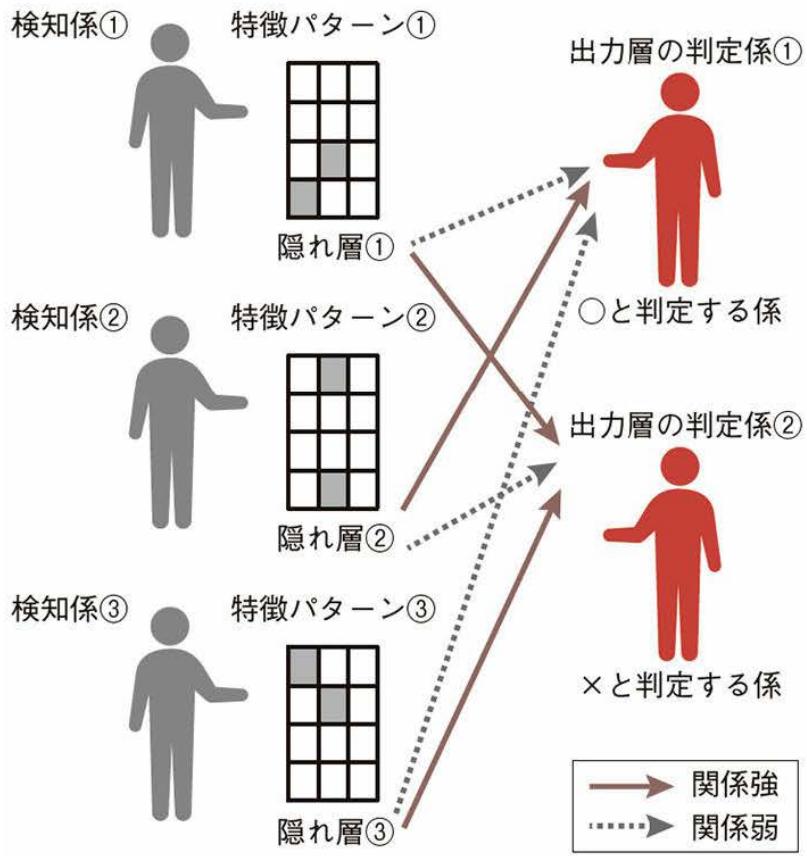
れ層のときと全く同じしくみで「入力の線形和」(2)から目的の情報を選別し、自分の任務とする判断がしやすいようにできるのです。

隠れ層 出力層



「○」と判定する判定係①には、パターン②の検知を受け持つ検知係②からの矢が太く(重みが大きく)なる。
「×」と判定する判定係②にはパターン①と③の検知を受け持つ検知係①③からの矢が太く(重みが大きく)なる。

擬人化された表現で図示すると、さらにわかりやすいでしょう。情報を伝えるパイプの細い太いで物事が決定されるのは、どこか人間社会にも似ています。

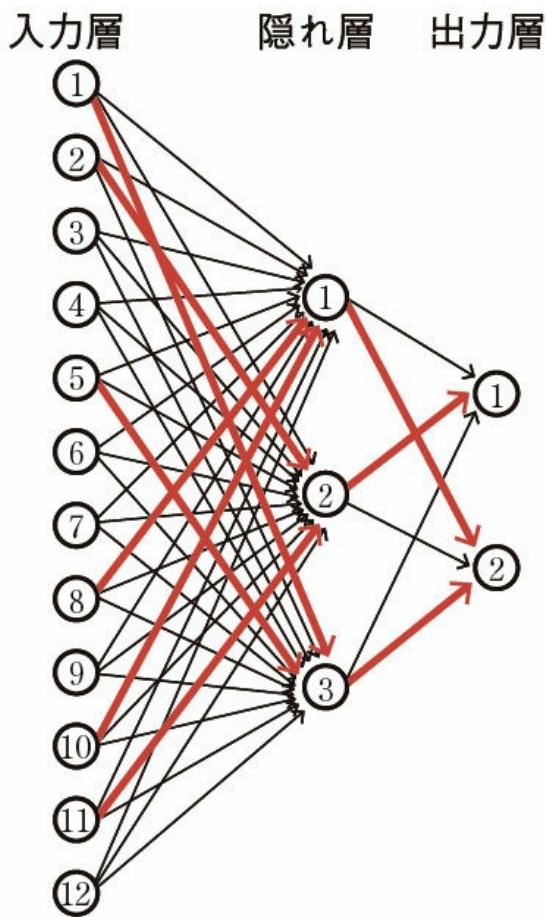


「○」と判定する出力層①は○の特徴パターンを持つ隠れ層②と親しく、隠れ層①③と疎遠のはず。

「×」と判定する出力層②は×の特徴パターンを持つ隠れ層①③と親しく、隠れ層②とは疎遠のはず。

■ まとめると

これまでの話をまとめてみましょう。結局、各層間の矢の太さ、すなわち「重み」の大きさが、ニューラルネットワークが画像を判別するカギになっていることがわかりました。その結果を次の図にまとめてみましょう。

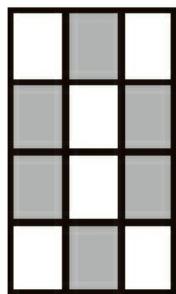


完成したニューラルネットワークのイメージ。太い矢は重みが大きいことを表わす。ニューラルネットワークが識別を可能にする秘密はこの矢の太さ(重みの大小)にある。

■ 秘密はネットワークの連携

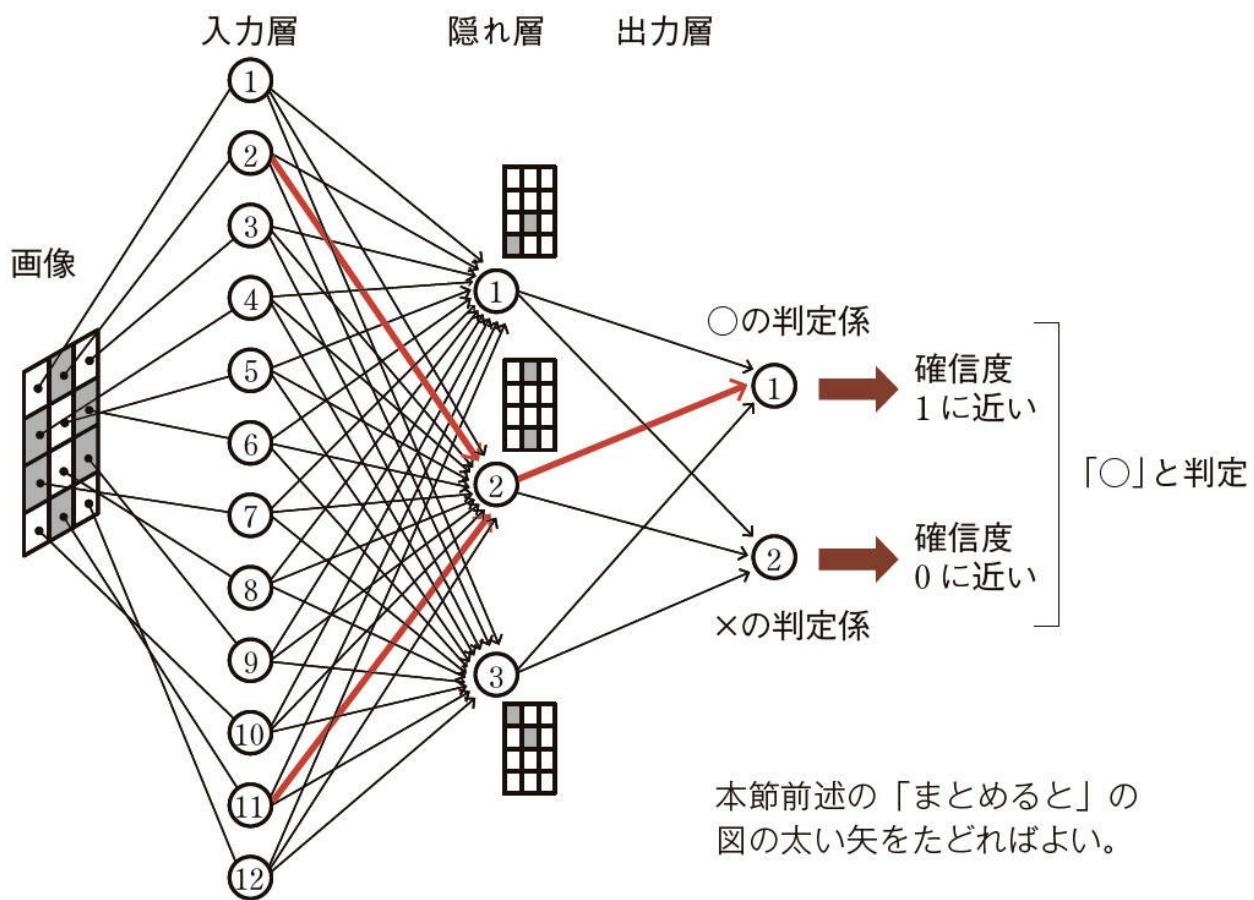
以上の話を具体化するために、次の例を調べましょう。

例2 次図の手書き文字「○」の画像が入力されたとき、ニューラルネットワークが「○」と判定するしくみを見てみます。

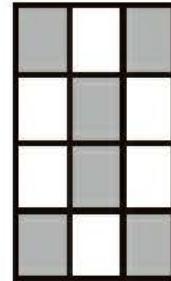


文字「○」を表す画像。

この文字画像には特徴パターン②が含まれています。そこで、運搬係の②、⑪は太い矢を持つ検知係の②に強い信号を送ります。すると、検知係②は太い矢を持つ出力層の判定係①に強い信号を送ります。こうして、「○」を判定する出力層の判定係①は「この画像は『○』」と確信し、確信度として1に近い値を出力します。それに対して、弱い信号しか受け取らなかった出力層の判定係②は「この画像は『×』」とする確信度として0に近い値を出力します。こうして、ニューラルネットワークは出力層の2者の確信度を比較することで「この文字画像は『○』である」と判定することになります。



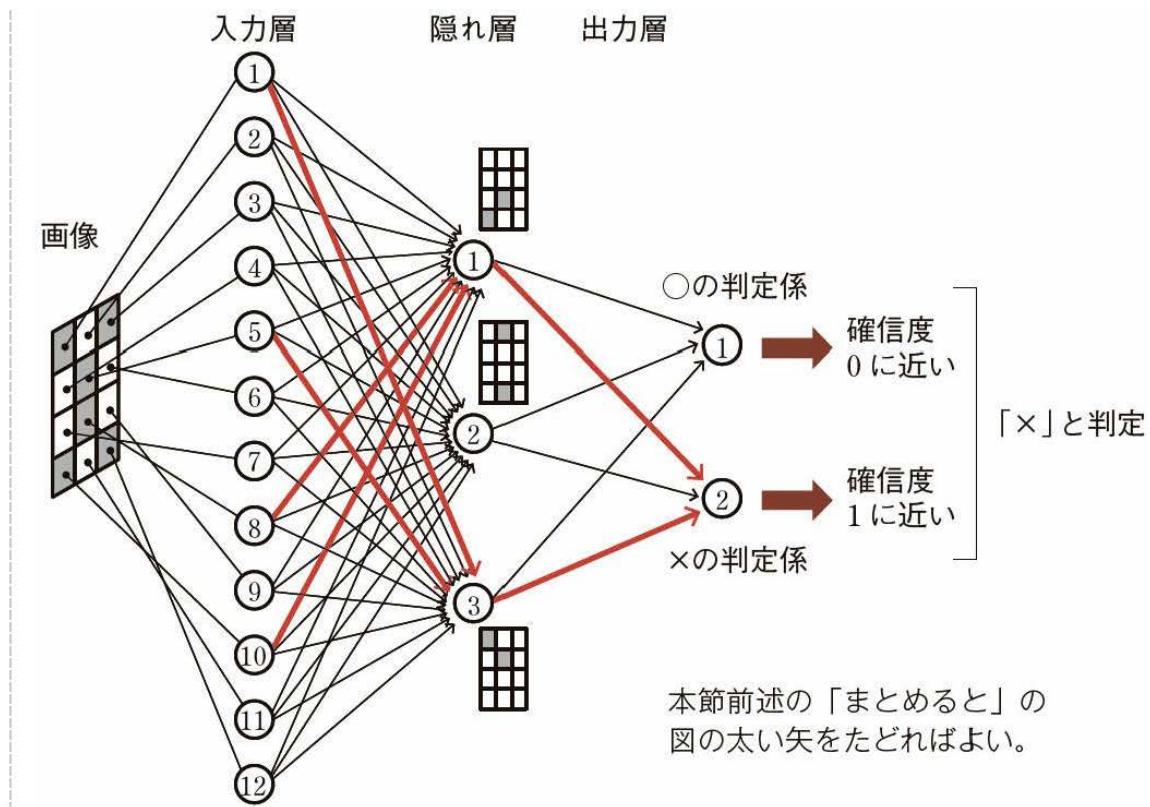
例題2 次の手書きの文字画像が入力されたとき、ニューラルネットワークが「×」と判定するしくみを説明しよう。



文字「×」を表す画像。

解

この文字画像には特徴パターン①と③が含まれています。左記の図の太い矢(情報伝達のパイプ)をたどることで、「×」と判定する判定係②に強い信号が伝わり、判定係②は1に近い確信度を出力します。それに対して、弱い信号しか受け取らなかった出力層の判定係①は「この画像は『○』とする確信度として0に近い値を出力します。こうして、ニューラルネットワークは出力層の2者の確信度を比較することで「この文字画像は『×』である」と判定することになります。(答)



■ 閾値の役割は不要な情報のカット

さて、隠れ層の検知係が入力層からの情報をえり分けるしきみが「重み」にあることを調べましたが、もう一つのパラメーターである「閾値」はどんな働きをするのでしょうか？

例えば、隠れ層の検知係のスタッフについて考えてみましょう。そのスタッフは自分と太い矢で結ばれた入力層の運搬係からの信号は大切です。しかし、それ以外の運搬係からの信号は雑音となります。その雑音をカットする役割が「閾値」なのです。閾値をちょうどよく設定することで、受け持つ目的の信号を取り込み、それ以外の信号を上手に抑え込むことができます。

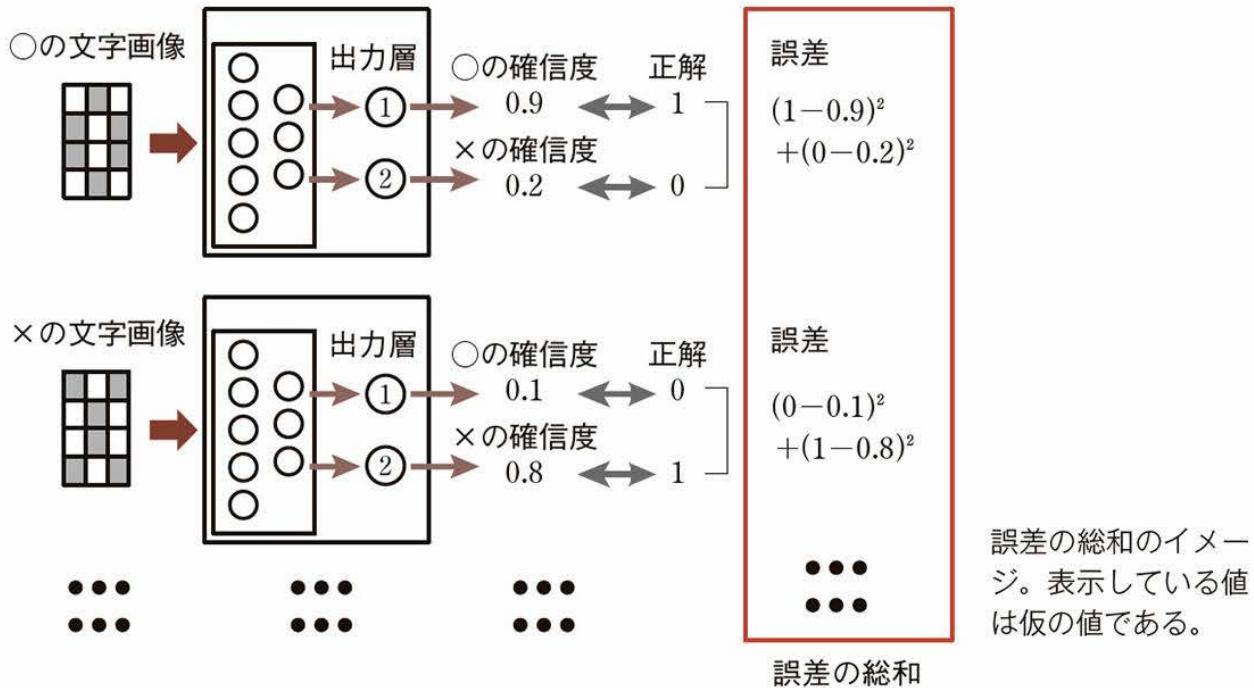
■ パラメーターの決め方

これまで、隠れ層の検知係が受け持つ「特徴パターン」は初めから与えられたものと仮定してきました。しかし、先にも注記したように、実際には何が画像の特徴なのかはわかりません。どうやって、画像の特徴が決められるのでしょうか？また、各ニューロンの重みについても、具体的にどう決定されるのでしょうか？

この疑問に答えるのが、ネットワーク自らが決定するというアイデアです。すなわち、重みや閾値は与えたデータからニューラルネットワーク自らが決定するのです。人が手取り足取りして教えるという操作はしません。

いま調べている例で考えてみましょう。「○」「×」の手書き画像のデータが何枚もあり、それらには1枚ずつ「○」か「×」かの正解がついていると仮定します。すると、最初にやることは、ニューラルネットワークに1枚1枚の画像を読ませ、「○」か「×」かの確信度を計算することです。次に、1枚1枚の画像に付けられた正解との誤差を算出し、画像データすべてにおいてこれらの誤差の総和を求めます。最後に、この誤差の総和が最小になるように、重みと閾値をコンピューターで決めるのです。

注 これら画像と正解のセットを教師データ、または訓練データといいます。



以上の数学的な手続きは数学モデルの最適化と呼ばれます。古くから研究されてきました。本書では、その最も簡単な方法として回帰分析を紹介しました(2章§4)。その最適化の手法からニューラルネットワークの重みと閾値が決定されるのです。別な表現をすれば、最適化の計算さえすれば、ニューラルネットワークは画像データから自分自身を決定するのです。これが先に述べた「ネット自らが決定」という表現になります。人が知識を与える余地はないのです。

■ コンピューター技術の発展の賜物

ネットワークの連係が判断を下すというしくみは、これまでの数学的アプローチとは異なる大きな可能性を秘めています。これまでの数学モデルは、それを規定するパラメーターをできるだけ少なくし、簡潔化しようとしてきました。それに対して、ニューラルネットワークのモデルは膨大な数のパラメーター(すなわち重みと閾値)が含まれています。このようなモデルを作成できるようになったのは、ひとえにコンピュー

ター技術の発展のおかげです。何百万という重みや閾値をコンピューターは嫌がらず計算してくれるからです。

■ ニューラルネットワークのアイデアのまとめ

ニューラルネットワークを構成するニューロン一つひとつは単純な働きをします。入力を線形和にまとめ(式(2))、その大小から0と1の間の数に変換します(式(1)) (式(1)(2)は本節の「ニューロン1個は知能を持たない！」参照)。このネットワークが画像の意味を判別できる「肝」の部分は重みをデータに合わせ調整することができます。特徴パターンが現れたときに、それが担当スタッフに報告されやすいように、矢の重み(すなわち情報伝達のパイプの太さ)を自らが決定するのです。

これはアリの社会にも似ています。アリー一匹一匹は大きな知能を持ちませんが、互いにネットワークを構成し、関係を築き合うことで複雑な社会をつくることができます。

次節以降、Excelと対話しながら、以上のことを見ていきましょう。

§2

ニューラルネットワークが手書き文字を識別

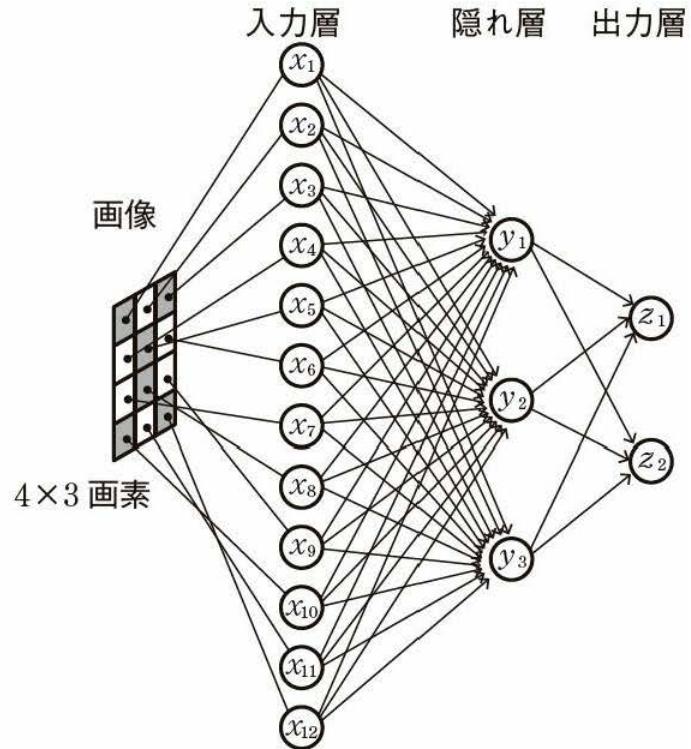
先の節ではニューラルネットワークのしくみを人の役割にたとえて調べました。本節からは、Excelを用いて具体的にそのしくみを調べることにしましょう。まずテーマを明確にし、そのためのデータを準備します。

■ これから調べる具体例

前の節(本章§1)で調べた次の「テーマ I」を具体例として、Excelを利用しながら議論を進めます。

テーマ I 4×3画素の白黒2値画像として読み取った手書きの文字「○」、「×」を識別する畳み込みニューラルネットワークを、64枚の手書き画像とその正解からなる訓練データから作成しましょう。

ニューラルネットワークとしては、§1で調べた次の図で表されるネットワークを利 用します。



本章の目標となるニューラルネットワークの形。なお、ニューロンに付された名称については、これから解説。

前節では各ニューロンを擬人化して、ニューラルネットワークのしくみを見てみました。これからはExcelを用いることで、具体的な数値でそのしくみを調べることにします。その中で、隠れ層や出力層の意味が数値的に明らかになるでしょう。

■ 各層の役割

前項に示すように、これから考えるニューラルネットワークは「入力層」「隠れ層」「出力層」の3層から成立します。それら各層がどのような働きをするかについては前節(§1)で擬人化して調べましたが、ここでももう一度確認します。

入力層の12個のニューロンは、ネットワークへ画像情報を運ぶ「運搬係」の役割を担います。換言すれば、入力層のニューロンは入力信号を中間層に伝えるだけで、何の処理も行いません。

隠れ層の3つのニューロンは「特徴抽出」の役割を果たします。入力された画像

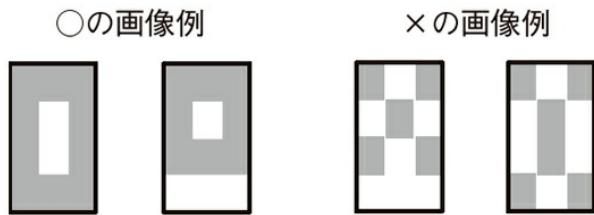
データから、特徴パターンを探し出します。

出力層の2つのニューロンは、上が「○」を、下が「×」に強く反応するように意図されています。隠れ層で抽出された特徴から、「○」「×」を総合的に判定し、その確信度を出力します。

以上のような解釈が可能なのは後に確かめられます。ここでは、ミステリー小説の「予告編」とと思っておいてください。

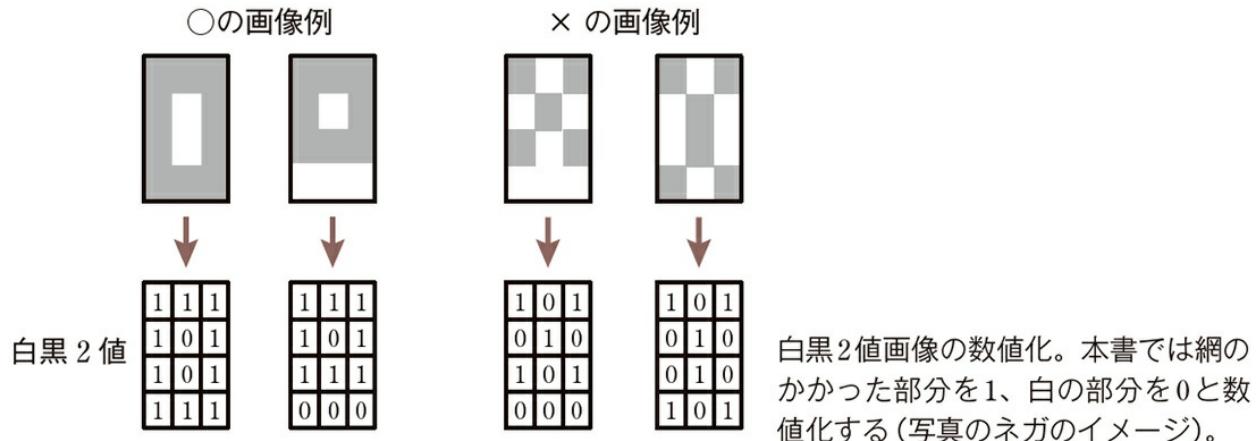
■ $4 \times 3 = 12$ ピクセルの白黒2値画像とは

先の節(§1)でも確認しましたが、「 $4 \times 3 = 12$ ピクセルの白黒2値画像」とは次のような画像を言います。



白黒2値画像の例。本書は、網を掛けた画素を1に、かかっていない画素を0と表現。

しかし、Excelはこのような模様を扱えません。模様を数値に置き換えることが必要です。本章の[テーマ I]では白黒2値画像なので、次の図のように0と1の2値のパターンに置き換えられます。



ところで、たとえ数値化されたとはいって、このような画像を読ませても、コンピューターは何を意味しているか理解できません。そこで、各手書き文字画像には、それが何を意味するかの「正解」を添えなければならないのです。この手書き文字画像と、その正解のセットを訓練データといいます。(また、学習データともいわれます。)いま調べている[テーマ I]の場合、手書きの「○」「×」画像と、その画像が「○」「×」どちらを表現するかの「正解」のセットが訓練データとなります。

白黒2値

1	1	1
1	0	1
1	0	1
1	1	1
1	1	1

正解

○

○

×

×

...

Excel が処理できる訓練データ

注 ニューラルネットワークで「教師なし学習」「強化学習」と呼ばれるタイプは、「正解」を直接には与えません。本書はこのような場合は考えません。

■ 画像データをExcelに入力

それでは訓練データをExcelに入力してみましょう。

例題 4×3画素の白黒2値画像として読み取られた手書き文字「○」と「×

を、正解と共にExcelに入力しましょう。

注 このワークシートは、ダウンロードサイト（→卷頭「Excelサンプルファイルのダウンロードについて」）に掲載されたファイル「4.xlsx」の中の「Data」タブにあります。なお、付録Aにその手書き数字のイメージを掲載してあります。

解

次の図に示すように、手書き文字画像とその正解を入力しましょう。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	○と×の識別																									
2																										
3	学習データ																									
4	番号	1	2	3	4	5	6	7		62	63	64														
5	文	1	1	1	0	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	
6	字	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	0	1	0	
7	画	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	1	0	
8	像	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	0	
9	正解	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	×	×	×		

正解の○と×の画像をランダムに配置していたほうが、のちの最適化の計算の収束性は良くなります。しかし、いま考えている訓練データは単純なので、結果がわかりやすいように、○と×を分離して入力しています。

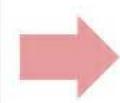
注 データを取り込んだワークシートのタブ名は「Data」としました。

Memo 数値で表された画像を模様で表示

Excelの「ホーム」タブにある「条件付き書式」を利用すると、次図のように数値パターンを濃淡の模様で表示できます。



7		
0	0	0
0	1	1
1	0	1
1	1	1
○		



7		
1	2	3
○		

数値データが与えられたときに、そのイメージをつくれるので便利です。

§3

訓練データの1つからニューラルネットワークの出力を算出

前節(§2)では、具体的なニューラルネットワークを提示し、そのための訓練データをExcelワークシートに入力しました。本節では、その最初の画像についてニューロンの出力値を算出しましょう。

注 この節の例題のワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「4.xlsx」の中の「例題」タブに一括して収められています。

■ 変数名の約束

ニューラルネットワークの出力を算出するにはニューロン間の関係を調べる必要がありますが、その前に関係を記述する際に必要な変数名について確認します。

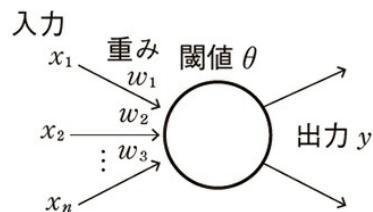
3章では、単独のニューロンを表現するのに、次の記号を導入しました。

x_i … i 番目の入力値

w_i … i 番目の入力にかけられる重み

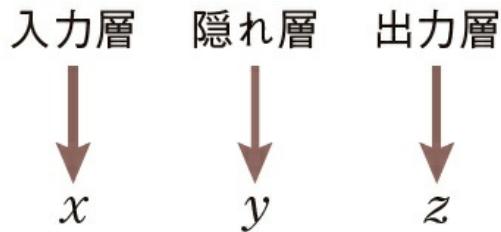
θ … 閾値

y … 出力



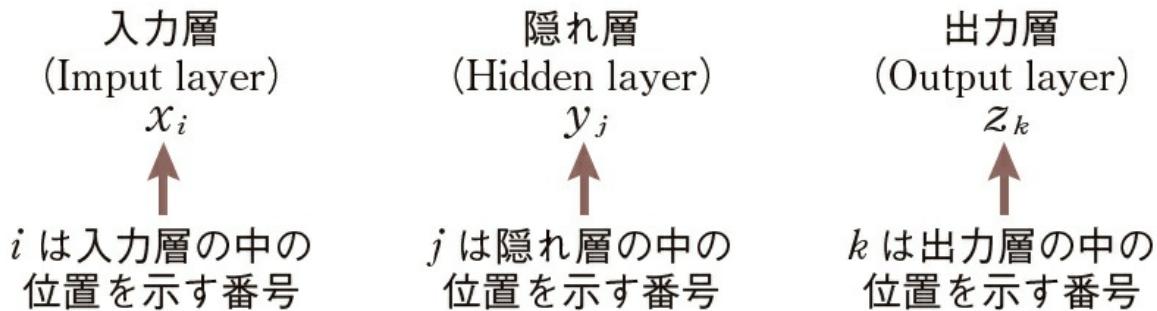
しかし、これらの記号はニューラルネットワークを記述するには力不足です。ネットワークの中でニューロンを議論するとき、どの層の何番目に位置するかの情報が必要になるからです。このことに留意しながら、ニューロンに新たな名称を与えましょう。

層の区別するために、入力層のニューロン名には x の文字を、隠れ層のニューロン名には y の文字を、出力層のニューロン名には z の文字を用いることにします。



3層を x 、 y 、 z の3文字で区別

各層の中のニューロンの位置は、該当層の上からの位置番号を用います。その番号を x 、 y 、 z に添え字として付加し、ニューロン名にします。

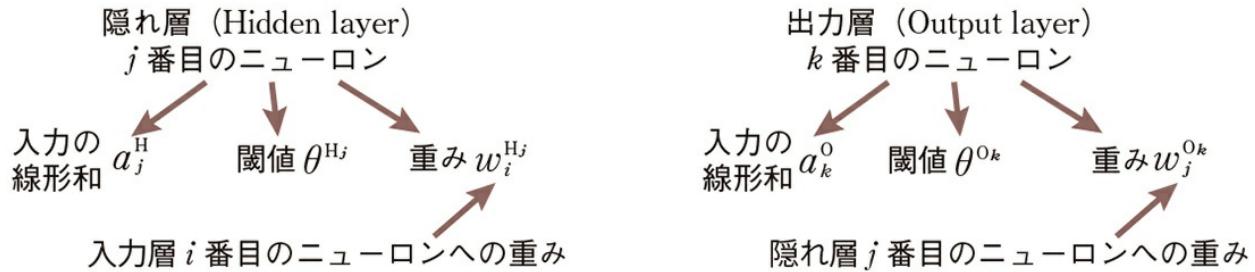


こうして名付けられたニューロンの出力はニューロン名と同一にします。すなわち、各ニューロンのニューロン名と出力変数名は同一にするのです。

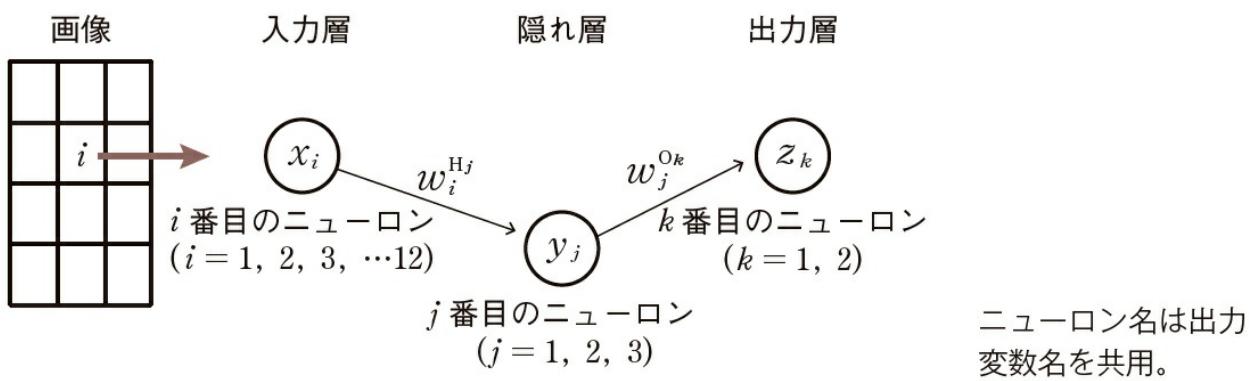


ニューロン名は出力変数名としても
使用する。

次に、ネットワークの各ニューロンに関する「重み」や「閾値」、そしてニューロンの「入力の線形和」を記述する変数名について考えます。これらは次の図のように約束します。



このように約束することで、次の図のように、層のニューロンの位置関係が示せます。



なお、各ニューロンについて、重み、閾値、入力の線形和の意味と役割は前の章(3章)で調べた単独のニューロン場合と同じです。以上のこととを表にまとめておきましょう。

記号名	意味
x_i	入力層 i 番目のニューロンへの入力を表す変数。入力層では、出力と入力は同一値なので、出力の変数にもなる。また、該当するニューロンの名称としても利用。
y_j	隠れ層 j 番目のニューロンの出力を表す変数。また、該当するニューロンの名称としても利用。
z_k	出力層 k 番目のニューロンの出力を表す変数。また、該当するニューロンの名称としても利用。
w_i^{Hj}	入力層の i 番目のニューロン x_i から隠れ層の j 番目のニューロン y_j に向けられた矢の重み。ニューラルネットワークを定めるパラメーターである。
w_j^{0k}	隠れ層の j 番目のニューロン y_j から出力層の k 番目のニューロン z_k に向けられた矢の重み。ニューラルネットワークを定めるパラメーターである。
θ_{Hj}	隠れ層 j 番目にあるニューロンの閾値。ニューラルネットワークを定めるパラメーターである。
θ^{0k}	出力層 k 番目にあるニューロンの閾値。ニューラルネットワークを定めるパラメーターである。
a^{Hj}	隠れ層 j 番目のニューロンに関する入力の線形和。
a^{0k}	出力層 k 番目のニューロンに関する入力の線形和。

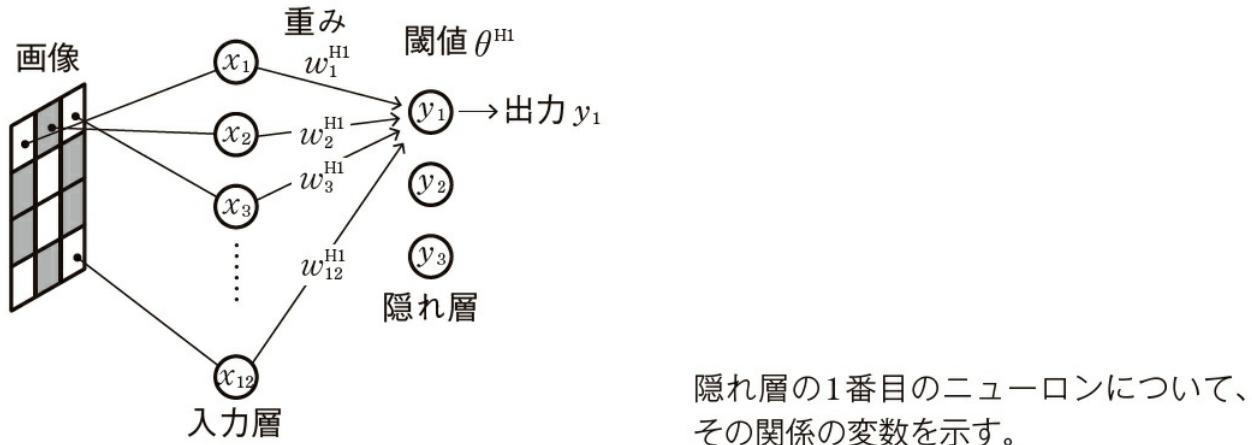
■ ネットワークを式で表現

ニューラルネットワークの中のニューロンの関係を式で表現する準備ができました。早速、その関係式を作成してみましょう。

ところで、ネットワークを構成する各ニューロンは3章で調べた単独のニューロンと同じ働きをします。そこで、関係式の作り方について、新しい話はありません。ただし、多数のニューロンが現れるので、その分、式は複雑になります。

まず、隠れ層のニューロンについて調べましょう。次図は隠れ層1番目のニューロ

ン y_1 について、変数の関係を示しています。



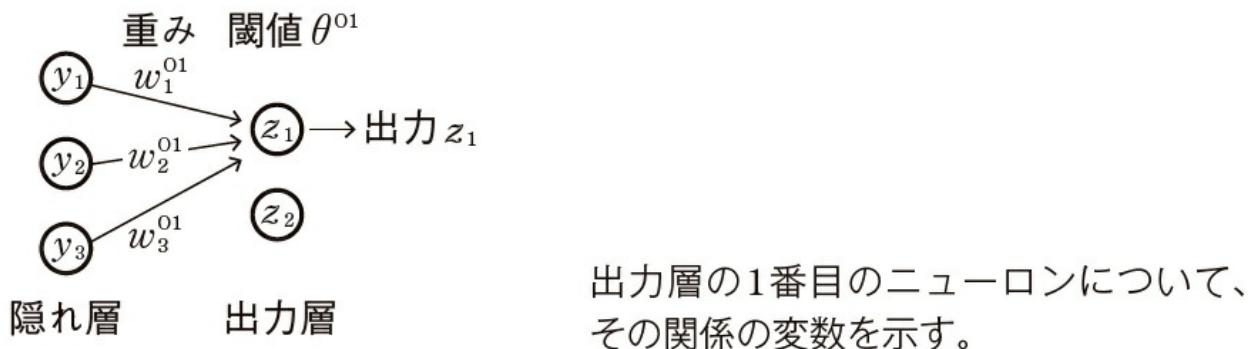
この図を参考にし、隠れ層についての全ての関係式が作成できます。

[隠れ層のニューロンに関する入力の線形和と出力]

$$\left. \begin{array}{l} a^{H1} = w_1^{H1} x_1 + w_2^{H1} x_2 + w_3^{H1} x_3 + \cdots + w_{12}^{H1} x_{12} - \theta^{H1} \\ a^{H2} = w_1^{H2} x_1 + w_2^{H2} x_2 + w_3^{H2} x_3 + \cdots + w_{12}^{H2} x_{12} - \theta^{H2} \\ a^{H3} = w_1^{H3} x_1 + w_2^{H3} x_2 + w_3^{H3} x_3 + \cdots + w_{12}^{H3} x_{12} - \theta^{H3} \end{array} \right\} \cdots (1)$$

$$y_1 = \sigma(a^{H1}), \quad y_2 = \sigma(a^{H2}), \quad y_3 = \sigma(a^{H3}) \quad (\sigma \text{ はシグモイド関数}) \cdots (2)$$

次に、出力層のニューロンについて調べましょう。次図は出力層の1番目のニューロンについて、変数の関係を示しています。



この図を参考にし、出力についての全ての関係式が作成できます。

〔出力層のニューロンの入力の線形和と出力〕

$$\left. \begin{aligned} a^{01} &= w_1^{01} y_1 + w_2^{01} y_2 + w_3^{01} y_3 - \theta^{01} \\ a^{02} &= w_1^{02} y_1 + w_2^{02} y_2 + w_3^{02} y_3 - \theta^{02} \end{aligned} \right\} \dots \quad (3)$$

$$z_1 = (a^{01}), z_2 = (a^{02}) \quad (\sigma \text{ はシグモイド関数}) \dots \quad (4)$$

■ 正解を変数化

画像を識別するための訓練データにおいて、各画像にはそれが何を意味するかの正解が付されています。今の[テーマ I]では、手書きの文字画像に「○」「×」のどちらかが付加されていることになります。ところで、「○」「×」では処理がしにくいので、計算しやすいように書き換えましょう。それが次の表に示す変数 t_1 、 t_2 の組です。

	意味	画像が「○」のとき	画像が「×」のとき
t_1	「○」の正解変数	1	0
t_2	「×」の正解変数	0	1

注 tはteacherの頭文字。教師データの正解なので、この変数名がよく用いられます。

このような組 t_1 、 t_2 で正解を表現すると、のちに紹介する誤差(→本章§4)を定義しやすくなります。

■ 1文字分の画像を作業用ワークシートに入力

ニューロン間の関係式が確認できました。訓練データの中の最初の1画像と正解について、順を追いながら処理してみましょう。まず作業用のワークシートに文

字1個分の画像と正解を入力します。

例題1 先の節で入力した訓練データの1番目の画像を、これから作業するワークシートの入力層に入力しましょう。なお、入力層は画素値をそのまま出力します。また、入力層の左下に付された正解の欄には、左側に上記変数の t_1 を、右側に t_2 の値を入力します。

解

前節(§2)で得た訓練データから、1番目の画像を次図の形式で新しい計算用ワークシートにコピーします。

注 前節(§2)で訓練データを取り込んだシート名は「Data」としています(→「画像データをExcelに入力」)。



		J8					
			:=Data!J5				
1		G	H	I	J	K	L
2	番号				1		
3	入力層				1	1	1
4					1	0	1
5					1	0	1
6					1	1	1
7	正解t1,t2						

訓練データの1番目の画像はJ3から始めることにする(→§2)。

次に、正解欄を作成します。先の表の t_1 、 t_2 を約束にしたがって、次図のように入力します。

	J7	=IF(Data!J9="O",1,0)		
1	G H I J K L M			
2	番号	1		
3	入力層	1 1 1 1 0 1 1 0 1 1 1 1		
4				
5				
6				
7	正解t1,t2	1 0	=IF(Data!J9="X",1,0)	

正解用のセルJ7には t_1 を、K7には t_2 の値をセット。この例は文字「○」を表す。

■ 仮のパラメーターを設定

ニューラルネットワークの計算の目標はパラメーター、すなわち重みと閾値の決定です。計算の最初でそれらの値がわかるはずもありません。しかし、それらの値がないとExcelの計算が進まず、これからの作業ができません。そこで、仮の重みと仮の閾値を設定しましょう。とりあえずの値を入力して話を進めるのです。

例題2 重みと閾値の仮の値を入力しましょう。なお、この重みと閾値は次の形式で入力します。

○と×の識別			
重みと閾値		番号	
隠れ層	1	0.06	0.17
		0.08	0.33
		0.15	0.92
		0.98	0.11
	2	0.08	0.91
		0.29	0.18
		0.35	0.12
		0.19	0.97
	3	1.00	0.16
		0.89	0.97
		0.94	0.12
		0.04	0.06
出力層	閾値	0.97	0.92
	1	0.18	0.92
	2	0.99	0.10
	閾値	1.00	0.94
正解t1,t2		1	0
隠れ層		1	2
出力 y		0.7	
出力層		1	2
出力 z		0.5	
出力層の閾値		出力層の閾値。左から順に 1 番目、2 番目のニューロンの閾値	
出力層の閾値		出力層の閾値。左から順に 1 番目、2 番目のニューロンの閾値	

解

上の図に示された重みと閾値の領域の1つにRAND関数を入力します。
RAND関数は0～1の乱数を発生させる関数です。

○と×の識別			
重みと閾値		番号	
	0.82	1	1
	1	0	1
		0	1
		1	1
		0	0

この関数をすべての重みと閾値の領域にコピーし、値を確定(値複写)します。こうして、上記のワークシートが得られます。

■ ニューラルネットワークの出力を計算してみよう

これまで入力した重みと閾値、そして1番目の画像を利用して、ニューラルネットワークの各ニューロンの出力を求めましょう。セル1個に対してニューロン1個が対応していることを確認しましょう。

例題3 これまでの準備のもとに、訓練データの1番目の画像について、隠れ層と出力層のニューロンの出力を算出しましょう。

解

式(1)～(4)を用いて、隠れ層と出力層のニューロンの出力 y_i ($i = 1, 2, 3$)、 z_i ($j = 1, 2$)を算出しましょう。

J10 の式: $=1/(1+EXP(-SUMPRODUCT(J3:L6,D3:F6)+D15))$

この式おかしい
右へコピーすると
おかしくなる

右にコピー

=1/(1+EXP(-SUMPRODUCT(\$D\$16:\$F\$16,J10:L10)+\$D\$18))

なお、仮の値で計算しているので、この出力値を議論するのは無意味です。

Memo | Excelのメリット

Excelでニューラルネットワークを実装するメリットは、式(1)～(4)を知らなくても計算式が作れることです。ニューロンの働きを頭にイメージしていれば、そのイメージに従って直接ワークシートに関数を入力できます。

§ 4

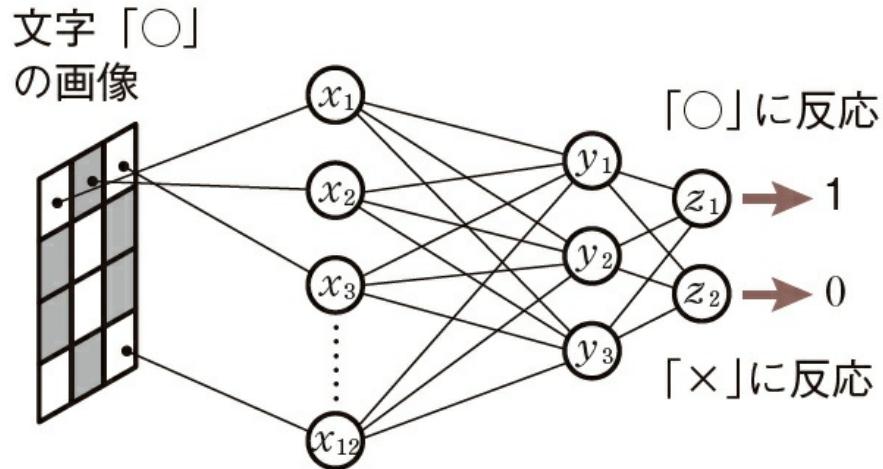
正解と出力の誤差

前節(§3)では、仮に与えられた重みと閾値を用いて、与えられた1枚の画像からニューラルネットワークの出力値を求めました。本節ではそこで算出した出力層の出力値が正解とどれだけ合致しているかを示す「目安」となる「平方誤差」について調べます。

■ ニューラルネットワークの出力値の意味

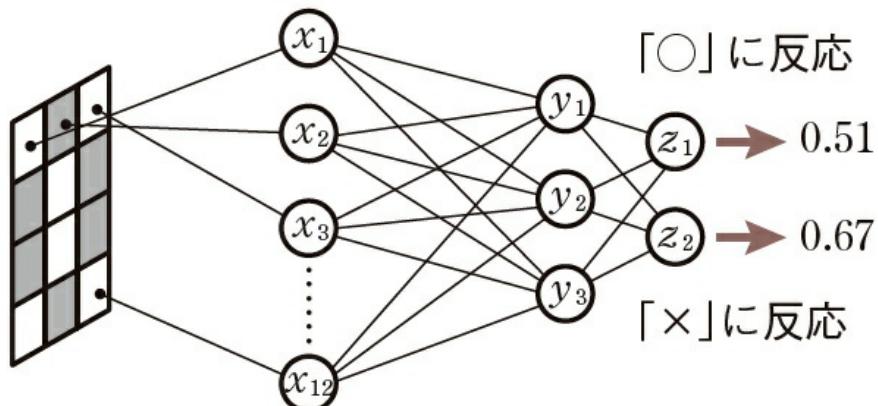
先に触れたように、[テーマ I]に示した出力層にある2つのニューロンは上から1番目が「○」を、2番目が「×」に反応するように意図されています。このことを念頭において、前節(§3)のニューラルネットワークの算出値を見てみましょう。

前節(§3)の[例題3]で取り上げた画像例は文字「○」を表しています。そこで、この場合、次の図のような出力を算出してくれるのが理想です。



出力層の1番目のニューロンは「○」を、2番目は「×」を検知する役割。そこで、「○」が読まれたなら、 $z_1 = 1$ 、 $z_2 = 0$ と算出されることが望ましい。

しかし、前節の算出結果は、次の図の通りです。シグモイド関数を利用する限り、当然ですが出力値は0か1にはなりません。



出力層の算出したこれら0.51、0.67とはどんな意味があるべきかを調べましょう。出力層のニューロンの出力 z_1 、 z_2 の値は、ニューラルネットワーク全体の結論です。また、シグモイドニューロンの出力値は、そのニューロンの「活性度」を表します(→3章§4)。今の場合、それはニューロン z_1 、 z_2 の「確信度」と言い換えられる

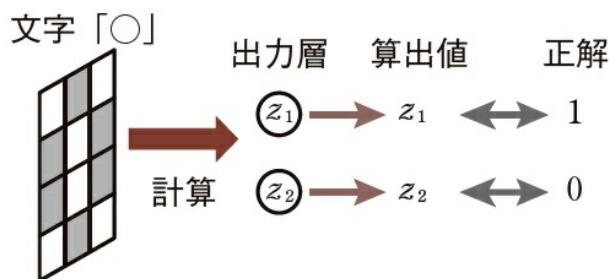
でしょう。すると、「○」の検知を役割とする出力層のニューロン z_1 の値が図のように 0.51 ということは、「たぶん 0.51 程度で○である」とニューラルネットワーク全体が確信したと考えられます。「×」の検知を役割とする出力層のニューロン z_2 の値が図のように 0.67 ということは、「たぶん 0.67 程度で×である」とニューラルネットワーク全体が確信したと考えられます。



注 この段階では「重み」、「閾値」は仮の値で、そのニューラルネットワークの算出値も仮の値です。そこで、 z_1 、 z_2 の値そのものを議論することに意味はありません。

■ 正解とニューラルネットワークの出力の誤差

以上の話から大切なことが見えてきます。文字「○」が読まれたとき、 z_1 の出力値と 1 との差が小さければ小さいほど、また、 z_2 の出力値と 0 との差が小さければ小さいほど、ニューラルネットワークはよい結果を算出したことになります。



文字「○」が読まれたとき、算出値 z_1 と 1 との差が小さければ小さいほど、算出値 z_2 と 0 との差が小さければ小さいほど、よい算出結果。

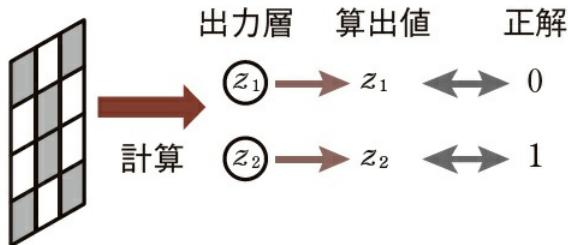
そこで、文字「○」が読まれたときのニューラルネットワークの算出値の誤差の評価として、次の値 Q が考えられます。

文字「○」が読まれたとき: $Q = (1 - z_1)^2 + (0 - z_2)^2 \dots (1)$

この値 Q が小さいとき、ニューラルネットワークは「よい値を算出した」ことになります。

文字「×」が読まれたときも同様です。 z_1 の出力値と 0 との差が小さければ小さいほど、また、 z_2 の出力値と 1 との差が小さければ小さいほど、ニューラルネットワークはよい結果を算出したことになります。

文字「×」



文字「×」が読まれたとき、算出値 z_1 と正解 0 との差が小さければ小さいほど、算出値 z_2 と正解 1 との差が小さければ小さいほど、よい算出結果。

そこで、文字「×」が読まれたときのニューラルネットワークの算出値の誤差の評価として、次の値 Q が考えられます。

文字「×」が読まれたとき: $Q = (0 - z_1)^2 + (1 - z_2)^2 \dots (2)$

この値 Q が小さいとき、ニューラルネットワークは「よい値を算出した」ことになります。

以上の式(1)(2)で定義した値 Q を、ニューラルネットワークの算出した値の平方誤差といいます。

注 文献によって、(1)(2)とは係数の違いがあります。多くの文献では誤差逆伝播法を意識して係数に $1/2$ を付けています。

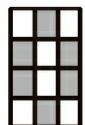
ここで、平方誤差(1)(2)が、その言葉の示す通り、平方の和であることに留意

してください。1画像だけの誤差評価ならば、わざわざ2乗(すなわち平方)の計算をする必要はありません。しかし、データ全体の誤差を見積もるとき、平方和であることが大切です。単に出力値と正解との差だけをとると、データ全体で誤差の和が相殺されてしまい、正しい誤差の評価ができなくなるからです。このことは、2章で調べた回帰分析と同じ事情です。

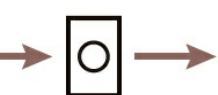
■ 平方誤差の式表現

以上の誤差(1)(2)の議論を図示すると、次のようになります。

○の文字画像 正解



正解



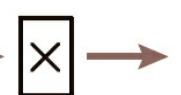
z_1 の正解 1
 z_2 の正解 0

$$\text{平方誤差 } Q = (1 - z_1)^2 + (0 - z_2)^2$$

×の文字画像 正解



正解



z_1 の正解 0
 z_2 の正解 1

$$\text{平方誤差 } Q = (0 - z_1)^2 + (1 - z_2)^2$$

ところで、前節(§3)では、画像の正解を表すのに変数 t_1 、 t_2 の組を用意しました。これは次の表の意味を持つ変数です。

	意味	画像が「○」のとき	画像が「×」のとき
t_1	「○」の正解変数	1	0
t_2	「×」の正解変数	0	1

この正解の組 t_1 、 t_2 を用いると平方誤差 Q の式(1)(2)は次のように1つにまとめられます。

$$Q = (t_1 - z_1)^2 + (t_2 - z_2)^2 \dots (3)$$

実際、例えば「×」の文字が読まれたとき、 $t_1 = 0$ 、 $t_2 = 1$ であり、式(3)は次のよ

うになります。

$$Q = (0 - z_1)^2 + (1 - z_2)^2$$

これは式(2)に一致します。

このように平方誤差を1つの式(3)として表現しておくと、Excelに式を入力する際に便利です。

■ 平方誤差をExcelで計算してみよう

これまで調べてきたことを実際にExcelで確認しましょう。

例題 前節(§3)で算出したニューラルネットワークの出力値から平方誤差 Q を求めましょう。

注 この節の例題のワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「4.xlsx」の中の「例題」タブに収められています。

解

式(3)を用いると、いま調べている画像についての平方誤差 Q は次のように求められます。

J20 : $=\text{SUMXMY2}(\text{J7:K7}, \text{J17:K17})$

A	B	C	D	E	F	G	H	I	J	K	L	
1	○と×の識別											
2	重みと閾値									番号	1	
3		1	0.06	0.17	0.12	入力層	1	1	1			
4		1	0.08	0.33	0.18		1	0	1			
5		1	0.15	0.92	0.12		1	0	1			
6		1	0.98	0.11	0.20		1	1	1			
7		2	0.08	0.91	0.12	正解t1,t2	1	0				
8		2	0.29	0.18	0.21							
9		2	0.35	0.12	0.22	隠れ層	1	2	3			
10		2	0.19	0.97	0.03	出力 y	0.77	0.92	0.97			
11		3	1.00	0.16	0.93							
12		3	0.89	0.97	0.11	出力層	1	2				
13		3	0.94	0.12	0.09	出力 z	0.51	0.67				
14		3	0.04	0.06	0.13							
15		閾値	0.97	0.92	0.94	誤差 Q	0.691					
16	出	1	0.18	0.92	0.06							
17	力	2	0.99	0.10	0.84							
18	層	閾値	1.00	0.94								
19												
20				Q _T								

式(3)を利用して、算出値の平方誤差が得られる

注 単純に見積もると、平方誤差 Q は0から2までの数値になります。当然ですが、0に近いほど誤差が小さいことになります。

Memo 平方誤差とSUMXMY2関数

Excelに備えられているSUMXMY2関数は平方誤差を算出する際に強力な武器になります。使い方については2章§1で調べましたが、上記ワークシートでその便利さを実感されると思います。

§5

ニューラルネットワークの目的関数

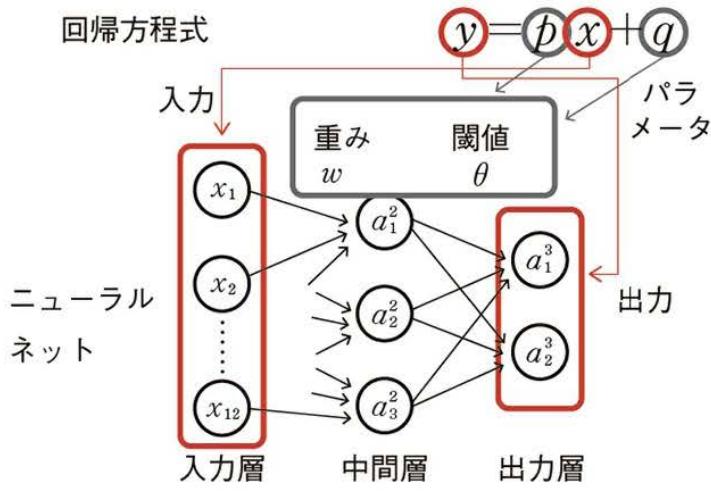
前節(§4)では、1つの画像とその正解からニューラルネットワークが算出する出力値と正解との平方誤差を算出しました。本節では、この誤差を訓練データ全体について加え合わせてみましょう。

■ モデルの最適化

これまで、重みと閾値に仮の値を利用してきました。では、これらの値はどのように決定されるのでしょうか？

一般的に、データを分析するための数学モデルはパラメーターで規定されます。2章§4では、その典型例として回帰分析を調べました。回帰分析では、回帰係数とその切片がパラメーターの役割を果たします。そして、そのパラメーターをデータにできるだけフィットするように決定する問題を最適化問題と呼ぶことを確認しました。

ニューラルネットワークの決定も、最適化問題の一つです。**モデルのパラメーターである「重み」と「閾値」を訓練データにできるだけ合致するように決定するのが目標**になるわけです。すなわち、重みと閾値は回帰分析と同じように決定されるのです。**誤差の総和を最小にするようにパラメーターは決められます。**



回帰分析でも、ニューラルネットワークでも、モデルの定め方は同一。回帰方程式の「回帰係数」、「切片」に相当するのがニューラルネットワークの「重み」と「閾値」。

では、このシナリオに準じて、これまで調べてきたニューラルネットワークの重みと閾値を決定していきましょう。この節では、訓練データ全体についての平方誤差 Q の総和がどのように求められるかを調べます。

■ ニューラルネットワークの目的関数

回帰分析でも調べましたが(→2章§4)、データ全体について平方誤差 Q を加え合わせ値 Q_T を目的関数と呼びます。いま考えている[テーマ I]のニューラルネットワークについて、その目的関数を式として示してみましょう。

前節(§4)で調べたように、1つの画像について、ニューラルネットワークの算出値と正解との誤差は次のように与えられます。

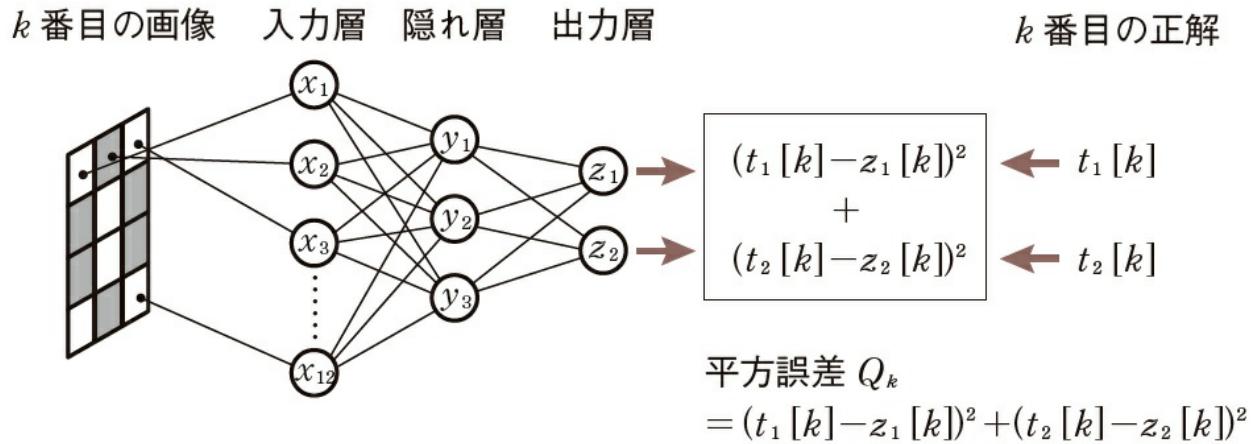
$$Q = \{(t_1 - z_1)^2 + (t_2 - z_2)^2\} \dots (1) \text{ (§4の式(3)再掲)}$$

ところで、訓練データのどの手書き文字画像に関するものなのか、この記号ではわかりません。そこで、 k 番目の画像の平方誤差を次のように表すことにします。

$$Q_k = \{(t_1[k] - z_1[k])^2 + (t_2[k] - z_2[k])^2\} \quad (k = 1, 2, \dots, 64) \dots (2)$$

ここで、 $t_1[k]$ 、 $t_2[k]$ は k 番目の手書き文字画像の正解を、 $z_1[k]$ 、 $z_2[k]$ は k 番目の手書き文字画像に対するニューラルネットワークの出力層ニューロンの

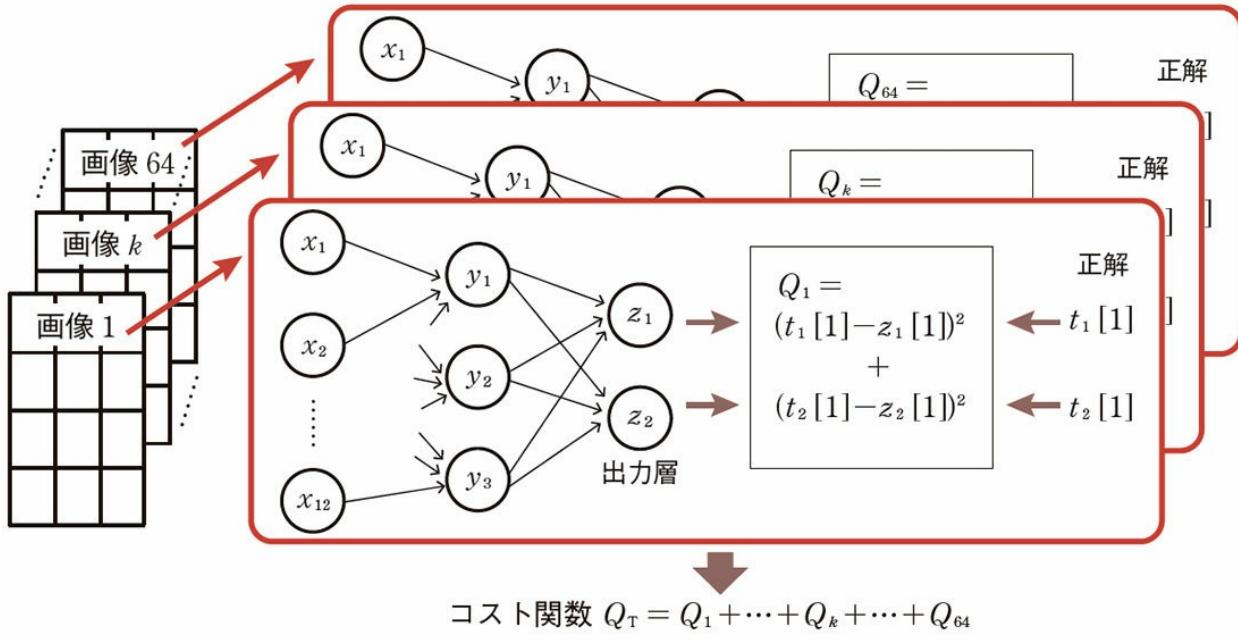
算出値を表します。また、値64はいま調べている訓練データの大きさ、すなわち画像の枚数です。



ニューラルネットワークを決定するために与えられた画像と正解のセット全体について、この Q_k を加え合わせたものが、訓練データ全体の誤差と考えられます。これがニューラルネットワークの目的関数 Q_T となります。

$$Q_T = Q_1 + Q_2 + \dots + Q_{64} \dots (3)$$

ここで、64は訓練データに含まれる手書き画像の枚数です。



目的関数の求め方。各画像についての平方誤差の総和が目的関数。

ちなみに、式(3)を重みと閾値の具体的な関数式で表現するのは現実的に無理です。

■ 目的関数をExcelで計算してみよう

実際に、いま調べている[テーマ I]に対するニューラルネットワークの目的関数の値をExcelで求めてみます。

例題 前節(§4)のワークシートで得られた1つの画像についての平方誤差を、訓練データ全体について加え合わせましょう。

注 この節の例題のワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「4.xlsx」の中の「例題」タブに収められています。

解

これまででは訓練データの1番目の画像について、各ニューロンの出力値を算出し、平方誤差を求めてきました。目的関数は全訓練データについての平方誤差の和なので、この1番目の画像について行った処理を全データについて実行しなければなりません。表計算ソフトの便利なところは、この処理をコピー操作で行えることです。次図のように、1番目の画像について行った処理を全画像についてコピーすればよいのです。

	H	I	J	K	L	M	N	O	Q	GQ	GR	GS
1												
2	番号		1			2				64		
3	入力層	1	1	1		0	1	1	0	0	0	1
4		1	0	1		1	0	1	0	1	0	1
5		1	0	1		1	0	1	0	0	1	0
6		1	1	1		1	1	1	1	1	0	0
7	正解t1,t2	1	0			1	0			0	1	
8												
9	隠れ層	1	2	3		1	2	3		1	2	3
10	出力 y	0.77	0.92	0.97		0.76	0.91	0.92	7	0.79	0.50	0.76
11												
12												
13												
14												
15												
16	出力層	1	2			1	2			1	2	
17	出力 z	0.51	0.67			0.51	0.66			0.41	0.63	
18												
19			誤差 Q			誤差 Q			誤差 Q			
20			0.69			0.68			0.31			

右にコピー

コピーが終了したら、目的関数(3)を算出します。上の図で求めた平方誤差 Q を SUM 関数で総和します。

Memo パラメーターの決め方

モデルのパラメーターを決定するには目的関数を最小化するという方法がニューラルネットワークではよく利用されます。パラメーターの決定には、他に最尤推定法と呼ばれる方法もあります。確率的に最も起こりやすいパラメーターを真の値とする決定法です。

The screenshot shows a Microsoft Excel spreadsheet with data representing a neural network. The network structure is as follows:

- Input Layer:** 3 nodes (values: 0.06, 0.17, 0.12)
- Hidden Layer 1:** 3 nodes (values: 0.08, 0.33, 0.18)
- Hidden Layer 2:** 3 nodes (values: 0.15, 0.92, 0.12)
- Hidden Layer 3:** 3 nodes (values: 0.98, 0.11, 0.20)
- Output Layer:** 3 nodes (values: 0.06, 0.17, 0.12)

Below the network, there is a table of weights and biases:

		番号	1	64
入力層	1	1	1	0 0 1
	2	1	0	1 0 1
	3	1	0	0 1 0
	4	1	1	1 0 0
正解t1,t2	1	0	0 1	
	2	1	0	1 2 3
	3	1	0	0.79 0.50 0.76
	4	1	0	1 2
隠れ層	1	2	0.41 0.63	
	2	1	0	0.69
	3	2	0.31	
	4	1	0	
出力 y	1	0.77	0.92	
	2	0.79	0.50	
	3	0.76	0.41	
	4	0.63	0.31	

The formula bar shows the formula `=SUM(J20:GS20)` for cell F20, which contains the value `31.19`. The cell `31.19` is highlighted in red.

先にも述べたように、この目的関数の値 Q_T は仮の重みと仮の閾値から得られたものです。したがって、この段階で目的関数の値の議論をするのは無意味です。

Memo 「教師なし学習」は難しい

ここで調べているニューラルネットワークのパラメーターの決め方は教師あり学習といわれます。正解が与えられているからです。ところで、宇宙人から届けられた文字画像の解読を例として考えればわかるように、その文字画像が何種類からできていて、そして正しい答えが何かなどは不明です。このような情報から正しく文字認識をするのは、教師あり学習よりもはるかに困難です。このような場合のパラメーターの決め方を教師なし学習といわれます。

§ 6

ニューラルネットワークの最適化

前節(§5)では、重みと閾値が与えられたときに、ニューラルネットワークの算出値と正解との誤差の総和(=目的関数)の求め方を調べました。本節では、それを最小化し、実際に重みと閾値を求めてみましょう。

■ 最適化の実行

前の節(§5)で得られた目的関数を用いて、ニューラルネットワークの重みと閾値を求めましょう。この操作を数学の世界では一般的に「最適化」と呼びますが、ニューラルネットワークでは学習と呼ぶことがあります。

最適化には、通常、そのための数学の知識が必要になります。しかし幸運なことに、いま調べている[テーマ I]のような単純なニューラルネットワークならば、Excelのソルバーを用いて簡単に最適化を行うことができます。数学上の技法は何も知らなくてよいのです。

■ Excelで目的関数を最小化

前節(§5)で求めた目的関数 Q_T をExcelのソルバーを利用して最小化してみましょう。なお、Excelソルバーの利用法は2章§3、2章§4を参照してください。

例題 これまで作成してきたワークシートを利用して、目的関数をソルバーで最小化してみましょう。

注 この節の例題のワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)からダウンロードしておいてください。

て」)に掲載されたファイル「4.xlsx」の中の「例題」タブに収められています。

解

ソルバーを呼び出し、目的関数のセルを「目的セル」に指定し、「重み」と「閾値」を「変数セル」に指定します。なお、「重み」も「閾値」もモデルに忠実にするために、非負数にしています。生命世界では、負のパラメーターは存在しないからです。

Excel のスクリーンショットで、シート「F20」が開かれています。セル F20 の値は「=SUM(J20:GS20)」で、計算結果は「31.19」です。

シート内には、重みと閾値の表と、各層の重みと閾値の表が並んでいます。また、誤差 Q の値が示されています。

「ソルバーのパラメーター」ダイアログボックスが表示されています。

- 目的セルの設定: (I)
- 目標値: 最小値(N) 最大値(M) 指定値(V)
- 変数セルの変更: (B)
- 制約条件の対象: (U) 制約のない変数を非負数にする(K)
- 解決方法の選択: (E) GRG 非線形
- 解: 解決(S)

ソルバーの計算が成功すると、次図のように重みと閾値が求められます。

A	B	C	D	E	F
1		○と×の識別			
2		重みと閾値			
3			0.26	0.00	0.92
4		1	0.00	2.51	0.03
5			0.00	6.52	0.00
6			4.75	0.00	0.26
7			0.05	4.97	0.04
8	隠 れ 層	2	0.57	0.00	0.37
9			0.68	0.03	0.40
10			0.01	5.00	0.02
11		3	2.92	0.00	2.13
12			0.00	5.24	0.00
13			0.00	0.06	0.02
14			0.02	0.00	0.03
15		閾値	10.65	4.01	7.69
16	出 力	1	0.00	11.57	0.05
17		2	14.75	0.00	10.51
18	層	閾値	5.66	6.07	
19					
20			Q _T	0.00	

ソルバーの算出値。目的関数 Q_T の値が0になっているので、この解は十分データにフィットしていることがわかる。なお、この結果のワークシートは、ダウンロードサイト（→巻頭「Excel サンプルファイルのダウンロードについて」）に掲載されたファイル「4.xlsx」の中の「例題_学習済」タブに収められている。

Memo | 計算結果は初期値に依存

数学的にいえば、目的関数 Q_T を最小とするパラメーターが最良のパラメーターということになります。しかし、ニューラルネットワークなどの複雑なモデルになると、「 Q_T の最小値」が不明なのです。パラメーターの初期値をいろいろ変更し、その中で最小なものを最善の解とするしか方法はありません。

ところで、初期値をいろいろと変えると、「 Q_T の最小値」は安定していませんが、パラメーターの値は時々に変化します。困ったことにも思えますが、生物学的には当然ともいえます。同じ考えに至るにしても、さまざまな思考過程があることに似ているからです。1章でも調べましたが、ニューラルネットワークを理解することで、動物の知能の理解にも役立つ応用例の一つといえるでしょう。

§ 7

最適化されたパラメーターを解釈

Excelでニューラルネットワークの計算をする最大のメリットは、算出結果を視覚的にすぐに確かめられることです。本節では、そのメリットを生かし、隠れ層、出力層の中身を見てみましょう。データ分析という観点からすると、最も興味深い課題になります。

■ 大きな重みに焦点を当てると

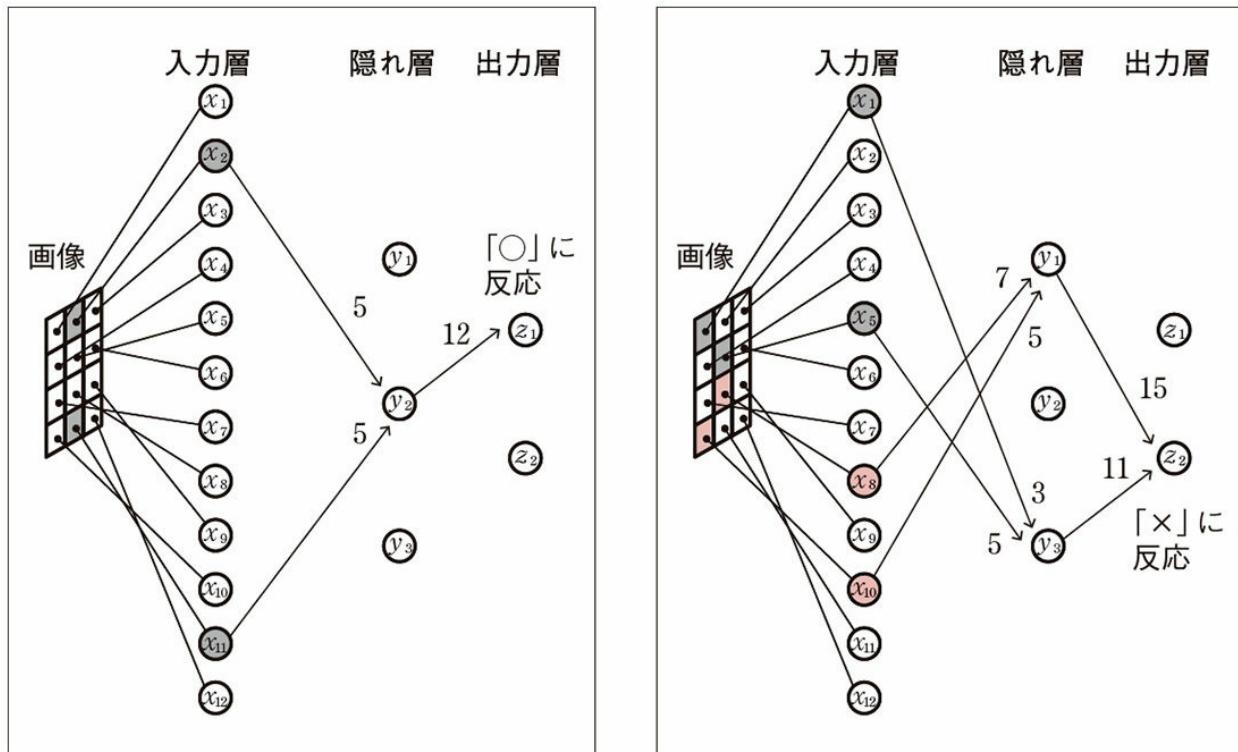
先の節(§6)で決定された重みと閾値はニューラルネットワークを決定するパラメーターです。その中で、「重み」はニューロンがその下の層のニューロンと結ぶ結合の強さを表しています。すなわち、情報交換のパイプの太さを表現しています。そこで、先の結果を用いて、大きさの順で上位2つの重みの値を取り上げ、他は無視した表を作成してみましょう(小数は四捨五入し、0になるものは無条件に無視)。

■隠れ層		■出力層	
ニューロン番号	重みの値	ニューロン番号	重みの値
1	7	1	12
	5		15
2	5	2	11
	5		
3	3		
	5		

出力層の1番目のニューロンに関する重みは、1つ以外すべて0なので無視する。ちなみに、このような雑音除去を実質的に担うのが閾値の役割。

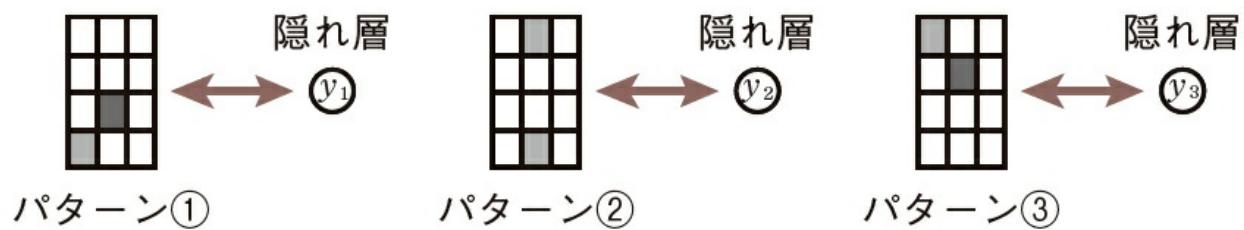
次に、この表に示された重みに関係するニューロンを矢で結んでみましょう。出

力層のニューロン1と2に分けてネットワークを図示すると、次のようにになります。それが次の図です。この図から、算出結果が読み解けます。



■ 隠れ層のニューロンの重みと特徴抽出

上の図において、まず入力層と隠れ層の結び付きを見てみましょう。隠れ層 y_1 には入力層の画素の組 (x_8, x_{10}) が強く結び付いていることがわかります。また、隠れ層 y_2 には入力層の画素の組 (x_2, x_{11}) が、隠れ層 y_3 には入力層の画素の組 (x_1, x_5) が強く結び付いていることがわかります。



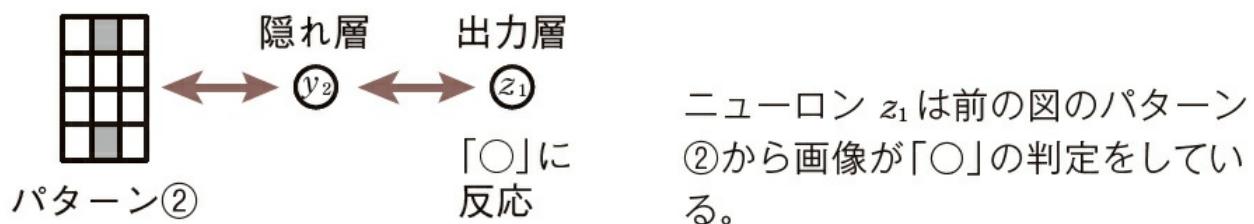
上の図のように、画素の組を順にパターン①、②、③と名付けましょう。すると、これらのパターンを持つ画像が読み込まれたなら、該当パターンと強く結びついた隠れ層のニューロンが興奮する（すなわち、出力が1に近い）ことになります。すると、隠れ層のニューロンの出力（すなわち興奮度）を調べれば、画像の中にパターン①～③が含まれるかどうかを判別できることになるのです。

隠れ層のニューロンは、このように重みを調整することで画像の中のパターン①～③を抽出するのです。先に「隠れ層は特徴抽出する」と述べましたが（→§1、2）、このようなしくみでその機能を果たしているのです。

注 このことを知った上で本章§1を読むと、そこで調べた「検知係」の意味がよく分かることでしょう。

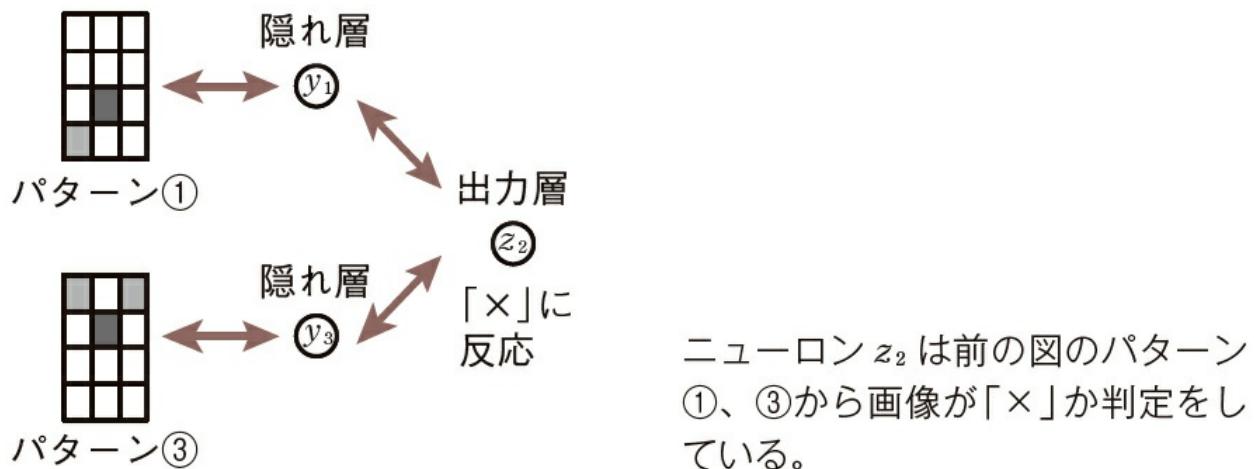
■ 出力層のニューロンは特徴抽出されたパターンから文字を判断

次に出力層を見てみましょう。最初にニューロン z_1 を調べます。このニューロンは「○」に興奮する（すなわち、出力が1に近い）ことが期待されています。前々図で矢をたどると、ニューロン z_1 と強く結びついている隠れ層のニューロンは y_2 です。ところで、そのニューロン y_2 は先に示した図のパターン②と強く結びついています。ということは、出力層のニューロン z_1 は、パターン②を用いて、画像が「○」かどうかを判定していることになります。



今度はニューロン z_2 を調べましょう。 z_2 は「×」に興奮する（すなわち、出力が1に

近い)ことが期待されているニューロンです。ニューロン z_2 と強く結びついているのは隠れ層のニューロン y_1 と y_3 です。その y_1 は先の図のパターン①と、 y_3 は先の図のパターン③とに強く結びついています。ということは、出力層のニューロン z_2 は、パターン①と③を組み合わせて、画像が「×」かどうかを判定していることになります。



■ 特徴抽出されたパターンの意味

パターン①を具体的な手書き文字「○」と「×」とで対比させてみましょう。いま述べたように、パターン①は「×」の判定に利用されますが、それは次図のように文字「×」に含まれ、「○」に含まれていない「部品」です。

手書き文字
「○」の例



含まない

隠れ層のニューロン
の抽出パターン①



含む

手書き文字
「×」の例



パターン②についても同様です。先に述べたように、パターン②は「○」の判定に利用されますが、実際それは次図のように文字「○」に含まれ、「×」に含まれていない「部品」です。

手書き文字
「○」の例



含む

隠れ層のニューロン
の抽出パターン②



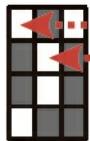
含まない

手書き文字
「×」の例



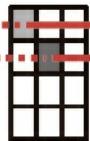
抽出パターン③についても同様です。抽出パターン③は「×」の判定に利用されますが、実際それは次図のように文字「×」に含まれ、「○」に含まれていない「部品」です。

手書き文字
「○」の例



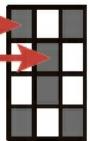
含まない

隠れ層のニューロン
の抽出パターン③



含む

手書き文字
「×」の例



こうして、ニューラルネットワークの出力層がどのように入力文字画像の「○」か「×」かを判定するしくみの全貌が見えました。特徴的な「部品」(パターン)の有無で判断を下していたわけです。

得られた結論は大変常識的です。原理的には人と同じことを実行しているわけです。本章の最初(§1)に擬人的な説明をしたことが、Excelの計算で確かめられたのです。

■ 閾値の意味は視覚化できない

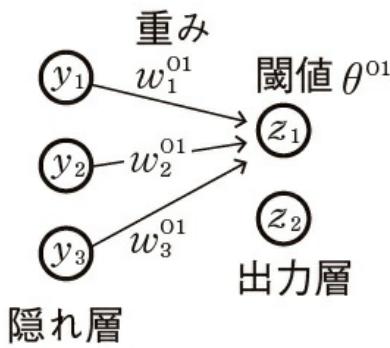
これまででは、算出結果の「重み」を調べてきました。次に「閾値」について見てみることにしましょう。先の[例題]の計算結果は次の値です。

隠れ層の閾値			出力層の閾値	
1	2	3	1	2
10.65	4.01	7.69	5.66	6.07

これらの値は、「重み」のときのようには視覚化できません。というのも、これらのパラメーターは最適化のための「黒子」の存在だからです。

例として、出力層の1番目のニューロン z_1 について調べてみましょう。そのニューロンの入力の線形和 a^{01} は次のように表現されます。ここで、その θ^{01} はニューロンの閾値です (→ 本章§3)。

$$a^{01} = w_1^{01} y_1 + w_2^{01} y_2 + w_3^{01} y_3 - \theta^{01} \quad \dots (1)$$



出力層1番目のニューロン z_1 について、その入力の線形和 a^{01} を書き下すための図。このニューロン z_1 は手書き文字「〇」を検知するためのニューロンである。

いま、「○」と「×」の手書き文字の画像が各々1枚ずつ読みされ、式(1)の
 $w_1^{01} y_1 + w_2^{01} y_2 + w_3^{01} y_3$ 」の部分の値が仮に8、3と得られたとします。

「○」のとき : $w_1^{01} y_1 + w_2^{01} y_2 + w_3^{01} y_3 = 8$

「×」のとき : $w_1^{01}y_1 + w_2^{01}y_2 + w_3^{01}y_3 = 3$

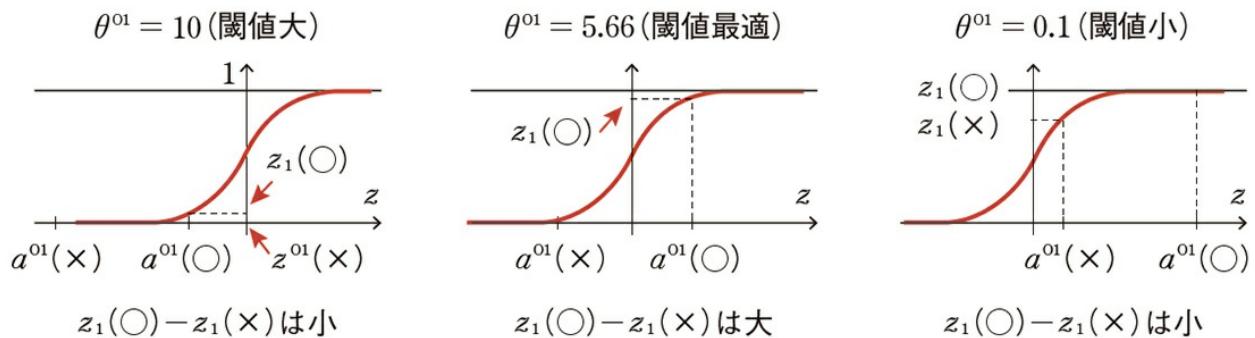
注 「○」に強く反応することを期待されているニューロン z_1 では、当然「○」の入力されたときの値は、「×」の入力されたときの値より大きくなります。

「○」が入力されたときの式(1)の値を $a^{01}(\circ)$ 、「×」が入力されたときの式(1)の値を $a^{01}(\times)$ とし、そのニューロン出力を順に $z_1(\circ)$ 、 $z_1(\times)$ としましょう。すると、これらは式(1)から次のように表せます。ここで $\sigma(a)$ はシグモイド関数です。

$$\text{「○」のとき: } a^{01}(\circ) = 8 - \theta^{01}, z_1(\circ) = \sigma(a^{01}(\circ))$$

$$\text{「×」のとき: } a^{01}(\times) = 3 - \theta^{01}, z_1(\times) = \sigma(a^{01}(\times))$$

すると、閾値の大小によって、次の図の位置関係が生まれます。



この図からわかるように、真ん中の図のように「閾値」 θ^{01} を設定すると、「○」と「×」に対するニューロンの出力差が大きくなります(左右の図の場合には出力差 $z_1(\circ) - z_1(\times)$ が小さく、区別がつきにくくなっています)。閾値を調整することで、「○」と「×」の違いがより鮮明に区別できるようになるのです。

こうして、閾値のニューラルネットワークにおける意味がわかりました。活性化関数を利用したニューロンによる識別をしやすくするために、各ニューロンの出力の違いを目立たせるように調整する役割を担っているのです。これはニューロン間の結びつきを表す「重み」では担えない役割です。

§ 8

ニューラルネットワークをテストしよう

これまで、訓練データを用いて、ニューラルネットワークを決定してきましたが、それはあくまで「訓練」用です。新しい画像に出会ったとき、そのニューラルネットワークが本当に正しい判定ができるかを調べましょう。

注 この節の例題のワークシートは、ダウンロードサイト（→卷頭「Excelサンプルファイルのダウンロードについて」）に掲載されたファイル「4.xlsx」の中の「テスト」タブに一括して収められています。

■ 新しい画像を入力

前の節（§6）で決定したニューラルネットワークが正しく動作することを次の例題で確認しましょう。

例題1 次に示すテスト用手書き文字の画像について、これまでに作成したニューラルネットワークが「○」か「×」かのどちらに判定するか調べてみましょう。



解

結果を示しましょう。

§ 7 で得た重みと閾値

○と×の識別のテスト

重みと閾値

1

0.26 0.00 0.92
0.00 2.51 0.03
0.00 6.52 0.00
4.75 0.00 0.26
0.05 4.97 0.04
0.57 0.00 0.37

閾値 10.65 4.01 7.69

出力層 閾値 1.00 0.00

出力層 1 2
出力 z 1.00 0.00

判定 ○

テストパターン

0	1	1
1	0	1
1	0	0
1	1	1

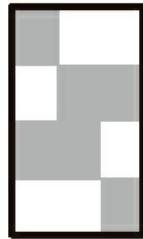
出力層のニューロン 1 がニューロン 2 よりも大きければ、入力された画像は文字「○」と判定

文字の判定結果 ○

=IF(J17>=K17,"○","×")

このテスト用の手書き文字は、ドット落ちしていて、○か×かは正確には判定できません。しかし、人ならば「○」と判定するでしょう。ところで、このワークシートも「○」と判定しています。前節までに確定したニューラルネットワークは、人と同様の判断を下したことになります。

例題2 次に示すテスト用手書き文字の画像について、これまでに作成したニューラルネットワークが「○」か「×」かのどちらに判定するか調べてみましょう。



解

結果を示しましょう。

§7で得た重みと閾値

$=IF(U17>=V17,"○","×")$

		E	F	G	S	T	U	V	W
1	○と×の識別のテスト								
2	重みと閾値								
3		0.26	0.00	0.92					
4		0.00	2.51	0.03					
5		0.00	6.52	0.00					
6		4.75	0.00	0.26					
7		0.05	4.97	0.04					
8		0.57	0.00	0.37					
9		0.68	0.03	0.40					
10		0.01	5.00	0.02					
11		2.92	0.00	2.13					
12		0.00	5.24	0.00					
13		0.00	0.06	0.02					
14		0.02	0.00	0.03					
15	閾値	10.65	4.01	7.69					
16	出力層	1	0.00	11.57	0.05				
17		2	14.75	0.00	10.51				
18	閾値		5.66	6.07					
19									
20									

テスト用手書き文字の画像を数値表現

出力層のニューロン2がニューロン1よりも大きければ、入力された画像は文字「×」と判定

文字の判定結果

このテスト用の手書き文字は、人ならば「×」と判定するでしょう。ところで、このワークシートも「×」と判定しています。前節までに確定したニューラルネットワークは、先の[例題1]同様、人と同様の判断を下したことになります。

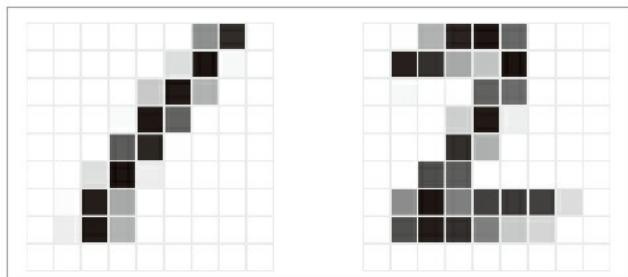
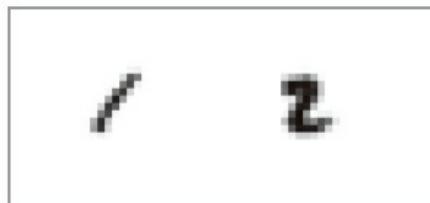
§ 9

現実の手書き文字にニューラルネットワークを応用

これまで、大変簡単な手書き文字の画像データについて、ニューラルネットワークを適用してきました。しかし、現実のデータに対しては、本節で見たニューラルネットワークには困難が生じます。それを調べましょう。

■ 実際の手書き文字にニューラルネットワークを適用

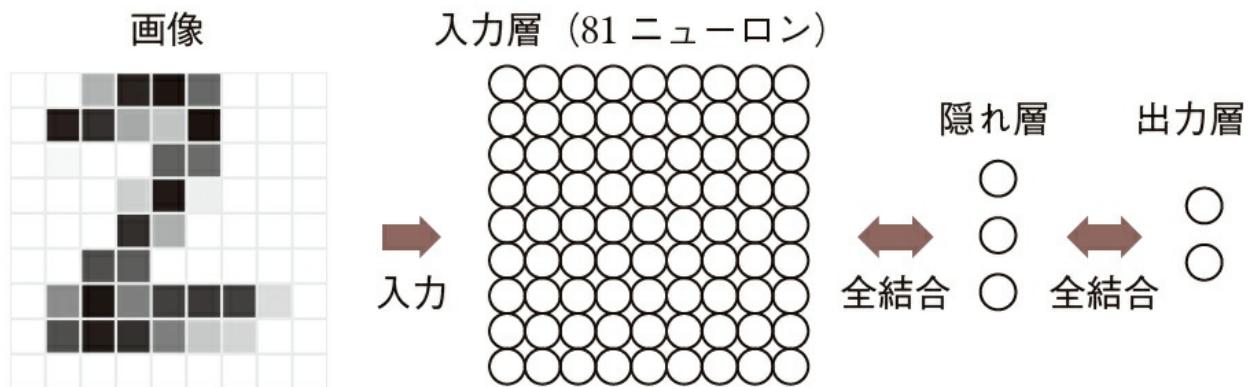
現実の手書き数字を区別するには、これまで調べてきたような 4×3 の解像度の画像では役に立ちません。例えば、次の文字は手書き数字「1」「2」ですが、この小さな画像でさえ解像度は 9×9 画素が必要です。実際、拡大すると、次の図に示すように 9×9 のマスに何とか収まります。



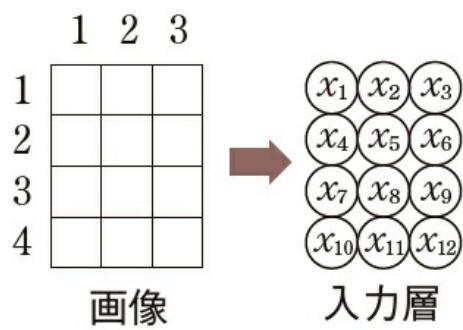
上の2つの手書き数字を拡大した図。
 9×9 のマスに収まる。

この小さな手書き数字1、2を、これまで調べてきたニューラルネットワークで識別してみましょう。すなわち、隠れ層に3ニューロン、出力層に2ニューロンを配置する

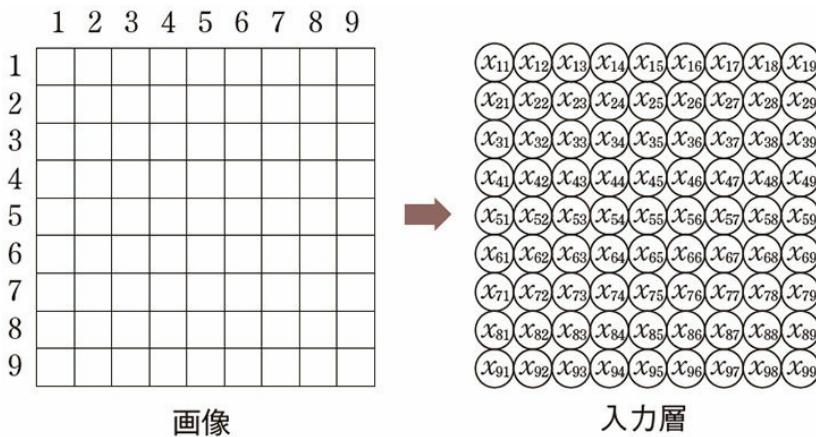
ニューラルネットワークを用いるのです。



ここで、これまでと大きく異なる点が1つ見えてきます。入力層のニューロン数が大きくなる点です。画素数が 9×9 なので、入力層のニューロンの数もそれに合わせて 9×9 個になります(次図)。これまで 4×3 画素だったので、 4×3 個のニューロンしか必要ありませんでした。



これまで調べてきた画素と入力層のニューロン。各々 4×3 個から構成されている。



画素数が 9×9 なら、入力層ニューロン数も 9×9 個が必要。

このニューロン数の増加はニューラルネットワークの理論自体には影響を及ぼしません。しかし、Excelに実装するとなると、大きな問題が生じます。決定すべきパラメーター数が増えてしまうことから来る、ソルバーの能力オーバーの問題です。

画像の解像度が9×9画素になると、隠れ層の各ニューロンが入力層から向けられる矢の数は9×9本となります。すると、隠れ層のニューロンがこれらの矢に与える「重み」の個数は、次の値になります。

$$3 \times (9 \times 9) = 243\text{個}$$

したがって、Excelに実装すると、次の図のようなワークシートになります。それに訓練データを与え、最適化のためにソルバーを実行してみましょう。すると、その下のメッセージが出力され、異常終了します。ソルバーは200までのパラメーターしか扱えないからです。

入力層のニューロン数は
9×9個

隠れ層の重みの個数は
3×(9×9) = 243個



このオーバーフローの問題はExcelソルバーの能力が低いため、とも考えられます。しかし、実用のことを考えると、そう単純な話ではありません。実際に利用される画像では1000万画素が普通です。そこで、何の工夫もなく、これまでのニューラルネットワークをコンピューターに実装すると、Excelに限らず計算が不可能になるのです。すなわち、ニューラルネットワークに何か工夫が必要なことがわかります。そ

ここで登場するのが畳み込みニューラルネットワークです。次章では、このネットワークについて調べましょう。

5 章

畳み込みニューラル ネットワークのしくみ

畳み込みニューラルネットワークの仕組みについて調べましょう。畳み込みニューラルネットワークは現在話題のディープラーニングの基本となるネットワークで、本書の目的となるテーマです。

§1

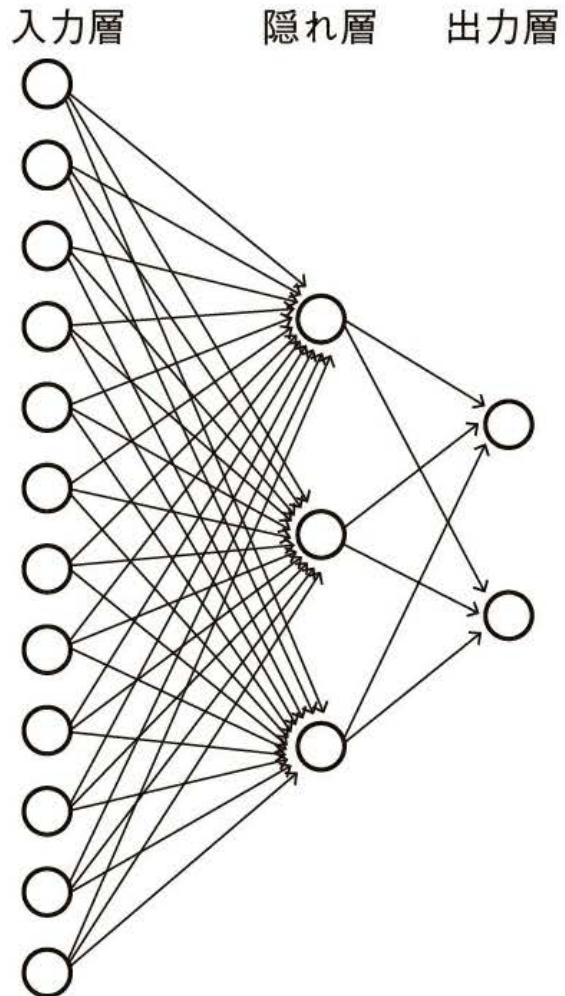
読み物としての畳み込みニューラルネットワークのしくみ

ディープラーニングとは隠れ層(中間層)が幾重にも重なったニューラルネットワークのことです。深層学習と訳されています。隠れ層に構造を持たせ、より効率的に学習が進むようにしたニューラルネットワークです。その中で特に近年脚光を浴びているのが畳み込みニューラルネットワーク(Convolutional Neural Network、略してCNN)です。

ここでは、この畳み込みニューラルネットワークがどのようなしくみで機能するかを、4章§1の「読み物としてのニューラルネットワークのしくみ」に登場してもらった「運搬係」「検知係」「判定係」の3つの役割のスタッフに解説してもらいましょう。

■ ニューラルネットワークの問題点

前章で調べたニューラルネットワークを見てみましょう。これは12個のニューロンからなる入力層、3個のニューロンからなる中間層(隠れ層)、そして2個のニューロンからなる出力層から構成されています。



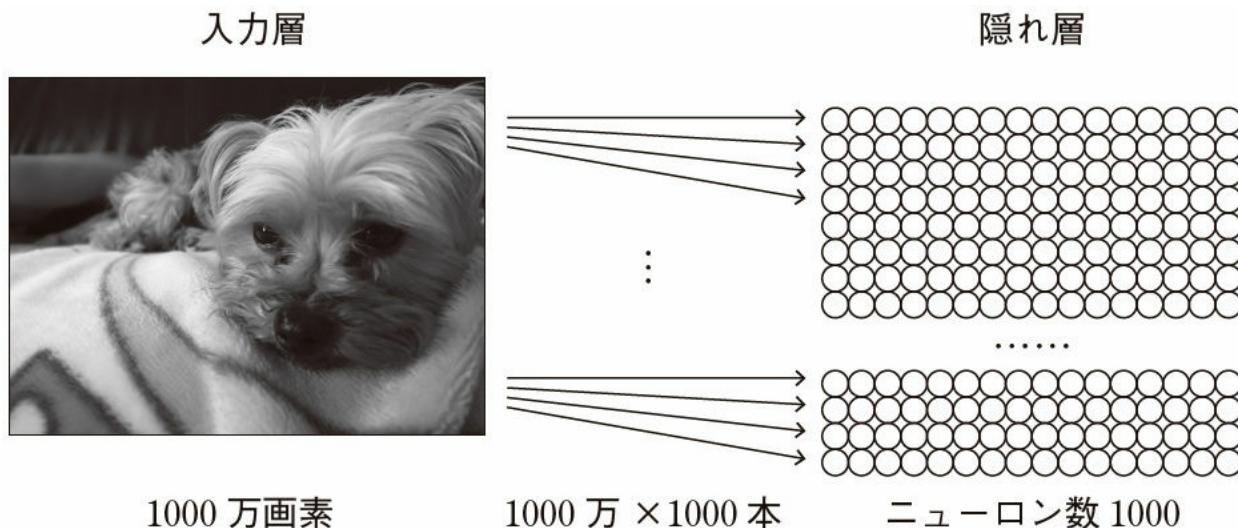
4章で調べたニューラルネットワーク。こんな簡単な形でも、一応文字識別が可能。

こんな簡単なニューラルネットワークでも、 4×3 画素に収まる簡単な手書き文字「○」、「×」を区別できました。しかし、たかだか2つの簡単な文字を区別するためにも、隠れ層は入力層と $3 \times 12 (= 36)$ 本の矢を結ぶことになります。まさにこの本数がニューラルネットワークの問題点となるのです。

例えば、写真から猫と犬とを識別するニューラルネットワークを作ることを考えてみてください。現代では安価なデジカメでも1000万画素の解像度を持ちます。すると、入力層から隠れ層の一つのニューロンに向ける矢の本数は1000万となります。「○」、「×」の区別と違って、猫と犬とを識別するには隠れ層には少なくとも1000個のニューロンを配置する必要があるでしょう。すると、隠れ層と入力層の間

の矢の本数は次のように膨大な数になります。

1000万画素×隠れ層のニューロン数1000個＝百億(本)



このような膨大な数のパラメーターを決定するには、スーパーコンピュータでも手こずります。また、それに対応するデータを用意するのも至難です。以上の問題を解決するのが畳み込みニューラルネットワークです。

注 データの大きさがパラメーターの個数以上でないと、モデルは決定されません(→2章§4)。

■ 特徴抽出がポイント

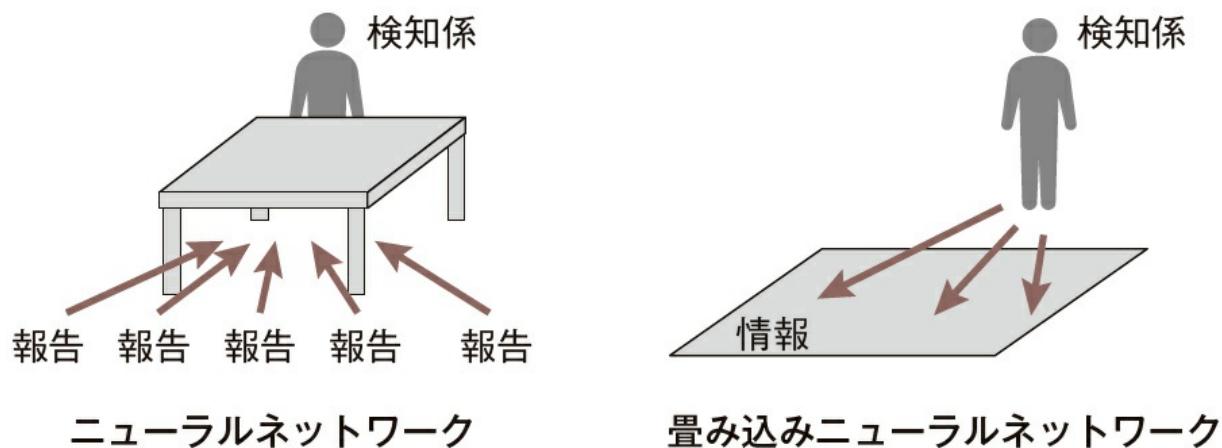
畳み込みニューラルネットワークのアイデアの基本は4章で調べたニューラルネットワークと同じです。ただ、隠れ層が行う「特徴抽出」に一工夫がなされます。

前章で調べたように、ニューラルネットワークを構成する各層を人の役割に見立てると、隠れ層には「検知係」が対応しました。そのスタッフは入力された画像から目的とする特徴パターンを検知し、その「含有度」を上の層に報告します。

畳み込みニューラルネットワークの隠れ層にも同じように「検知係」のスタッフが常駐し、特徴パターンの検知作業をします。その検知作業の方法がニューラルネ

ットワークと畳み込みニューラルネットワークでは大きく異なるのです。

ニューラルネットワークの検知係は椅子に座って下の層から全員の報告を待つて、「ヨッコイショ」と仕事をするタイプです。それに対して畳み込みニューラルネットワークの検知係は活動的で、積極的に報告を拾いに行くタイプです。この積極性こそがネットワークを簡潔にし、モデルのパラメーターを大幅に低減してくれるのです。



ニューラルネットワークの検知係は報告を待つタイプ。それに対して畳み込みニューラルネットワークの検知係は積極的に情報を拾いに行くタイプ。

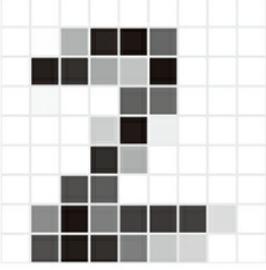
■ 例で調べよう

話を具体的にするために、本章は次のテーマを用いて話を進めることにします。

テーマ II $9 \times 9 = 81$ 画素のモノクロ画像として読み取った手書きの数字

「1」、「2」を識別する畳み込みニューラルネットワークを作成しよう。

最初に、 $9 \times 9 = 81$ 画素のモノクロ画像として読み取った手書きの文字「1」、「2」とはどのようなものか、次に例示しましょう。

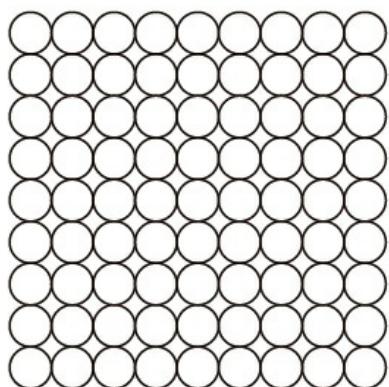


0	0	23	197	228	198	18	0	0
0	11	254	254	216	251	254	51	0
0	130	223	53	8	6	253	215	0
0	210	175	0	0	0	69	254	2
4	253	198	0	0	0	14	254	13
0	200	244	27	0	0	94	252	1
0	11	248	243	198	200	254	144	0
0	0	42	223	254	234	153	0	0
0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0
0	1	74	223	247	147	0	0	0
0	230	208	83	59	252	0	0	0
0	7	0	0	154	142	0	0	0
0	0	0	47	241	16	0	0	0
0	0	0	210	77	0	0	0	0
0	0	173	159	0	0	0	0	0
0	110	249	123	190	198	190	34	0
0	173	236	208	123	53	40	0	0

さて、これから調べるネットワークでは、前章と比較しやすいように、隠れ層の「検知係」の人数は前章と同じ3人とします。出力層には、2文字「1」、「2」を識別する「判定係」2人がいることも、前章と同様です。入力層には画素数と同じ人員が必要になるので、 $9 \times 9 = 81$ 人の「運搬係」が必要になります。

入力層 (81 ニューロン)



運搬係

隠れ層



検知係



出力層



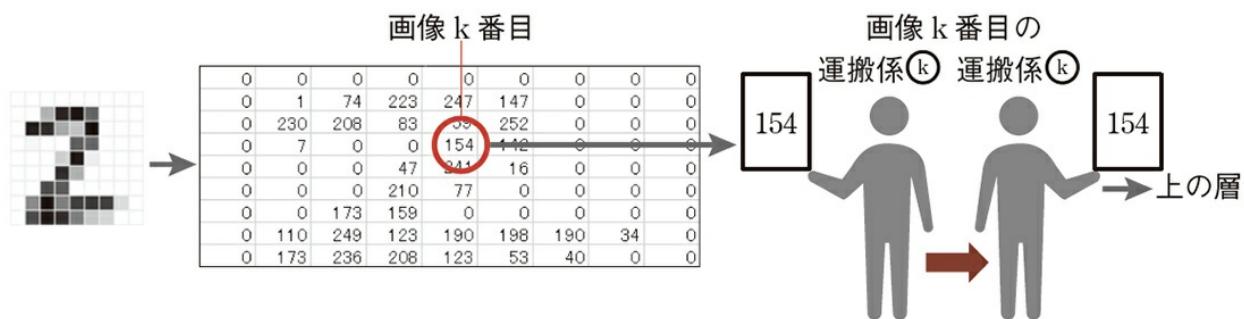
判定係



$9 \times 9 = 81$ 画素のモノクロ画像で読み取られた「1」「2」の手書き数字を識別する畳み込みニューラルネットワークの枠組み。

■ 置み込みニューラルネットワークの入力層の役割

最初に入力層について考えます。ニューラルネットワークでも調べたように(→4章§1)、この層はネットワークへ画像情報を運ぶ「運搬係」が対応します。彼らの役割はニューラルネットワークのときと全く同じです。各画素に1人ずつスタッフが配置され、その任務は入力された画素情報(すなわち信号の大きさ)について何も加工せず、そのまま上の層に報告することです。ただし、総勢81人($=9 \times 9$ 人)が任務にあたります。

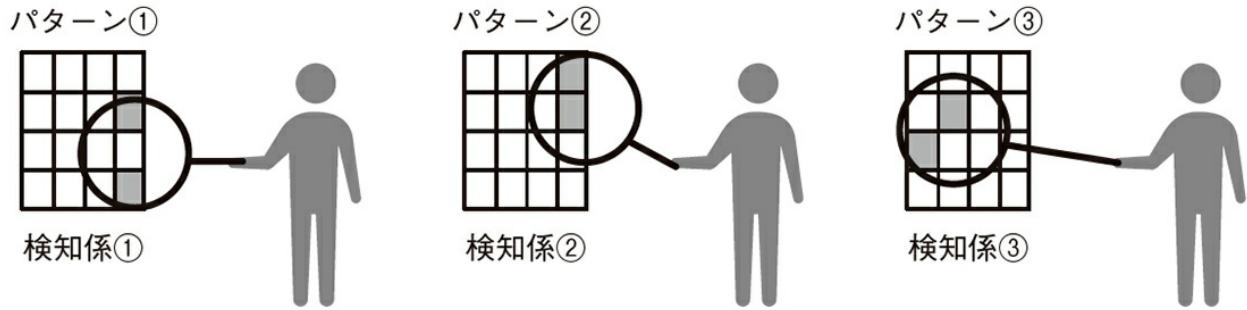


入力層の k 番目の「運搬係」。入力層の各ニューロンは画素情報をそのまま隠れ層の係に報告する。なお、次節以降では、ソルバーを有効に働かせるために、大きさを100分の1にして出力する方法を採用。

■ 置み込みニューラルネットワークの隠れ層の役割

次に隠れ層について考えます。この層にあるニューロンには、ニューラルネットワークのときと同様、画像上の特定のパターンの有無を調べ、その含有率を上の層に報告する「検知係」が対応します。この例題では隠れ層に3つのニューロン①～③があるので、3つの特徴パターンに関心を払うスタッフ①～③がいることを意味します。

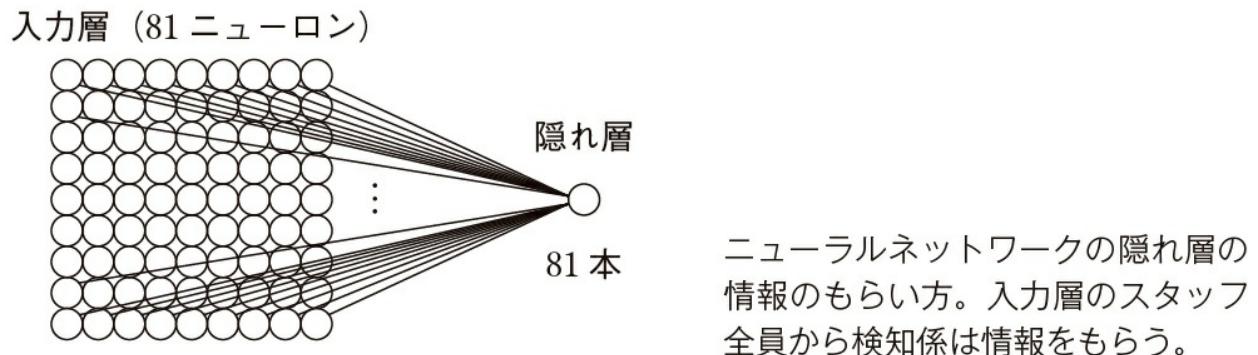
注 ここで含有率といつてもイメージ的な表現であり、厳密な意味ではありません。



各々異なる特定のパターンの読み取りを役割とする係員。この図のパターンはあくまで仮の例であり、実際は後に調べる「最適化」で決められる。読み取りの結果は、該当パターンの含有率として上の層(出力層)全員に報告する。

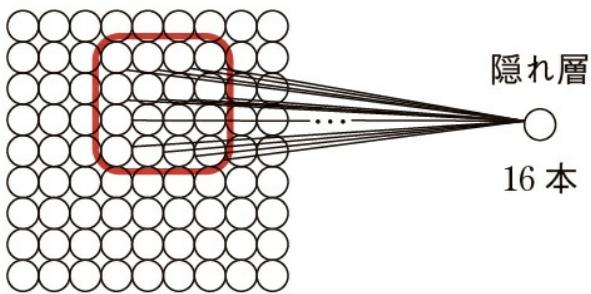
ここまで検知係についての説明は、基本的にニューラルネットワークの場合と同じです。では、ニューラルネットワークと畳み込みニューラルネットワークとでは、どこが根本的に異なるのでしょうか。その答えは、入力層から情報を受け取る方法にあります。

ニューラルネットワークは入力層の運搬係の全スタッフから情報を一括して受け取ります。



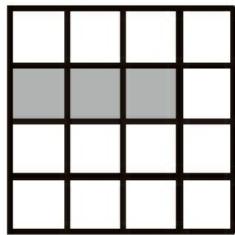
それに対して、畳み込みニューラルネットワークでは、入力層の運搬係から情報を小分けにして受け取るのです。こうすることで、「検知係」が同時に応する入力層のスタッフの人数が小さくなり、受け持つ特徴パターンを探索する手間を大幅に減らせます。

入力層 (81 ニューロン)



畳み込みニューラルネットワークの隠れ層の情報のもらい方。入力層のスタッフを小分けにしグループ化して情報をもらう。

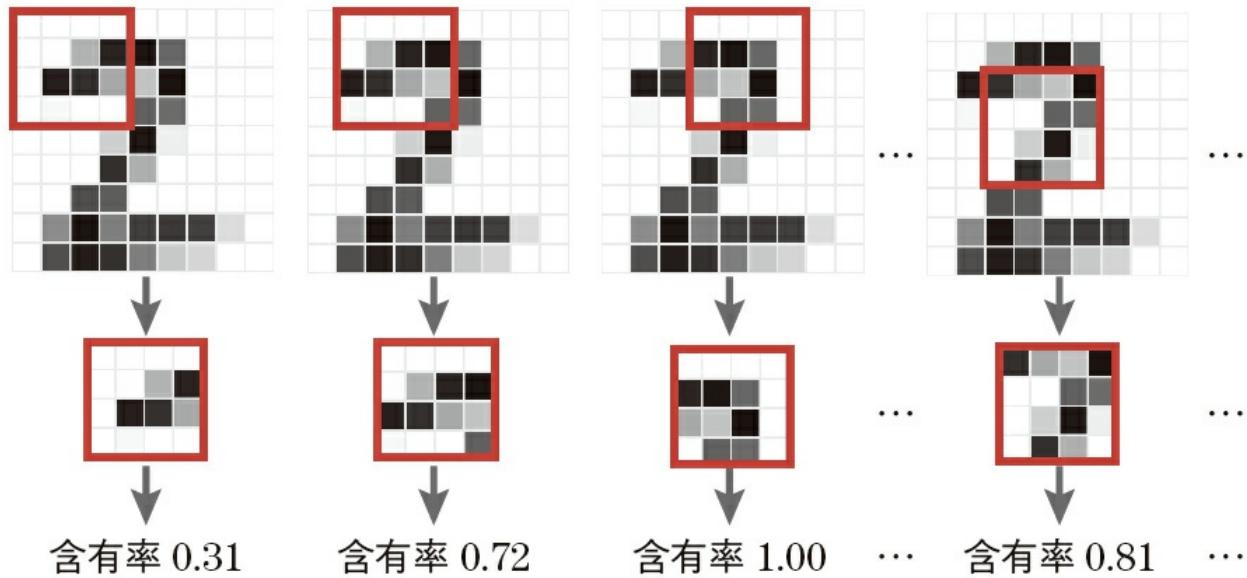
ここで突然ですが、隠れ層の3人の検知係の1人 \otimes が次のパターンの画素を検知する役割を担っているとしましょう。このパターンは $4 \times 4 (=16)$ の画素から成り立っていると仮定することにします。



隠れ層の検知係 \otimes が受け持つ特徴パターン

検知するパターンの大きさとして、現実的な畳み込みニューラルネットワークでは 5×5 の大きさが多用されます。しかし、本書では簡単にするために、検知する特徴パターンの大きさを 4×4 で多少スケールダウンします。

この検知係 \otimes は入力係から渡された画像を特徴パターンの大きさに小分けし、順次スキャンしていきます。そして、小分けした区画ごとにニューラルネットワークのときと同じ処理を施します。すなわち、受け持ちの特徴パターンの含有率を各検知係は独自に算出するのです。



ちなみに、畳み込みニューラルネットワークの場合、含有率は「特徴パターンとの類似度」と表現した方が似つかわしいかもしれません。上の図を見ればわかるように、ニューラルネットワークに比べ、調べる範囲が小さいからです。

さて、上の図からわかるように、畳み込みニューラルネットワークでは入力画像を小分けしたので、その出力は含有率の表になります。それに対して、ニューラルネットワークの場合、隠れ層の各検知係は入力画像を一括処理するので、出力は1個の含有率だけでした。

0.92

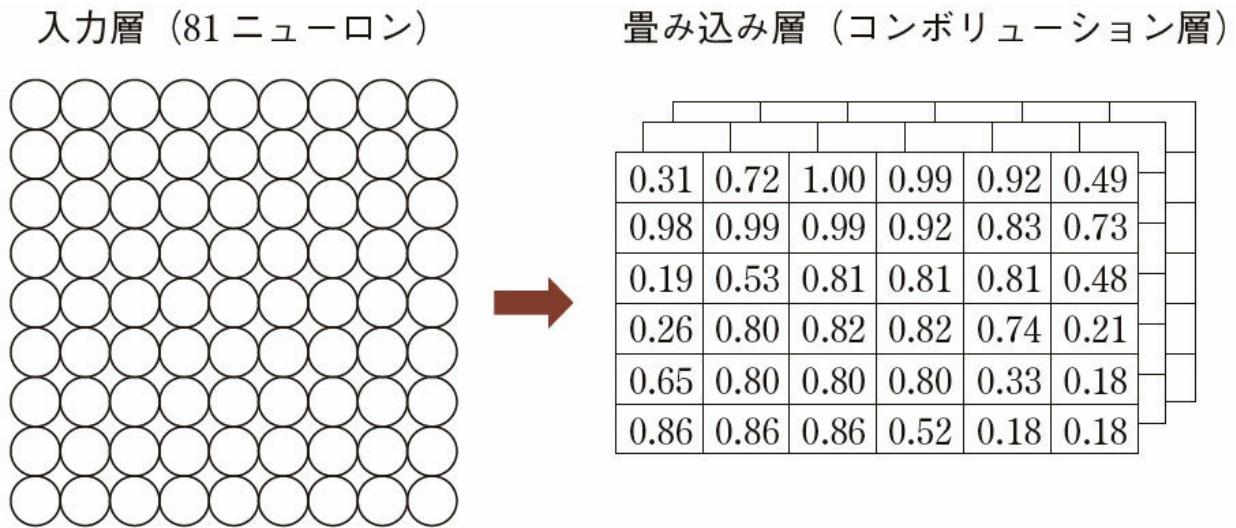
0.31	0.72	1.00	0.99	0.92	0.49
0.98	0.99	0.99	0.92	0.83	0.73
0.19	0.53	0.81	0.81	0.81	0.48
0.26	0.80	0.82	0.82	0.74	0.21
0.65	0.80	0.80	0.80	0.33	0.18
0.86	0.86	0.86	0.52	0.18	0.18

ニューラルネットワークの隠れ層のニューロンの出力は「数」。

畳み込みニューラルネットワークの隠れ層のニューロンの出力は「表」。

隠れ層には検知係のスタッフ3人を仮定しているので、1人につき1枚、計3枚の

表が出力されます。この表の1セットが一つの層を形成します。これが畳み込み層と呼ばれる層です。コンボリューション層とも呼ばれます。



調べる画素を小分けにしたぶん、隠れ層の検知係は3枚の表として出力を提供する。これが畳み込み層を形成する。

■ プーリング層で情報を濃縮

通常、検知係が算出した畳み込み層の表はまだ大きなものです。今の例ではたかだか9×9画素の画像でしたが、実際は100×100画素以上の画像を扱います。そこで、検知係はさらに情報の縮約を試みます。

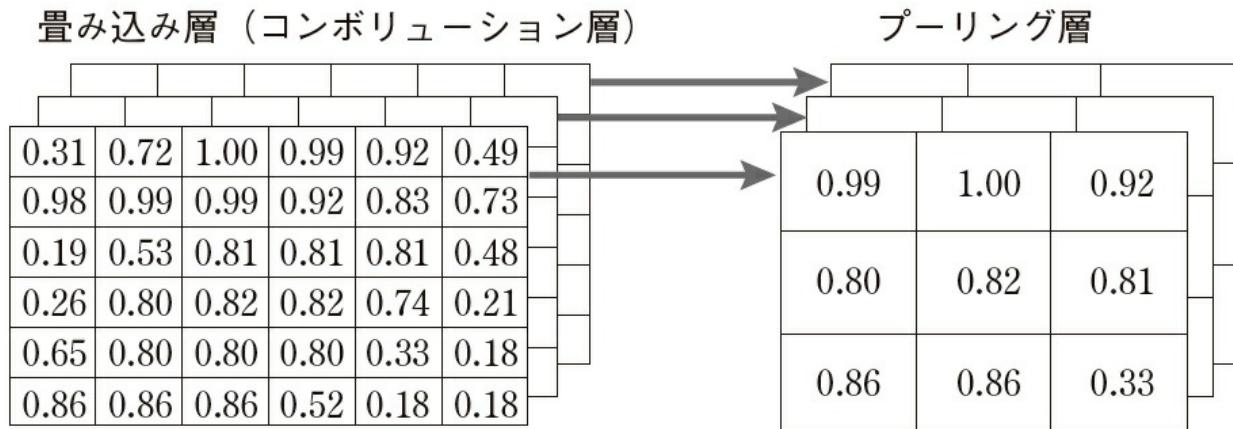
情報の縮約法として、例えば、次のような方法が考えられます。畳み込み層として得られた各出力表を2×2の大きさに区分けし、その区分けされた4つの中から最大値を代表として選出するのです。こうすることで、表は4分の1に縮約されます。先の検知係 \otimes が出力した表を例にして調べてみましょう。



0.31	0.72	1.00	0.99	0.92	0.49
0.98	0.99	0.99	0.92	0.83	0.73
0.19	0.53	0.81	0.81	0.81	0.48
0.26	0.80	0.82	0.82	0.74	0.21
0.65	0.80	0.80	0.80	0.33	0.18
0.86	0.86	0.86	0.52	0.18	0.18

0.99	1.00	0.92
0.80	0.82	0.81
0.86	0.86	0.33

畳み込み層にある3枚の表にこの処理を施してみましょう。こうして縮約された表の一群はプーリング層と呼ばれます。情報内容としては荒くなりますが、入力層の81画素の情報がコンパクトな情報にまとめられたことになります。



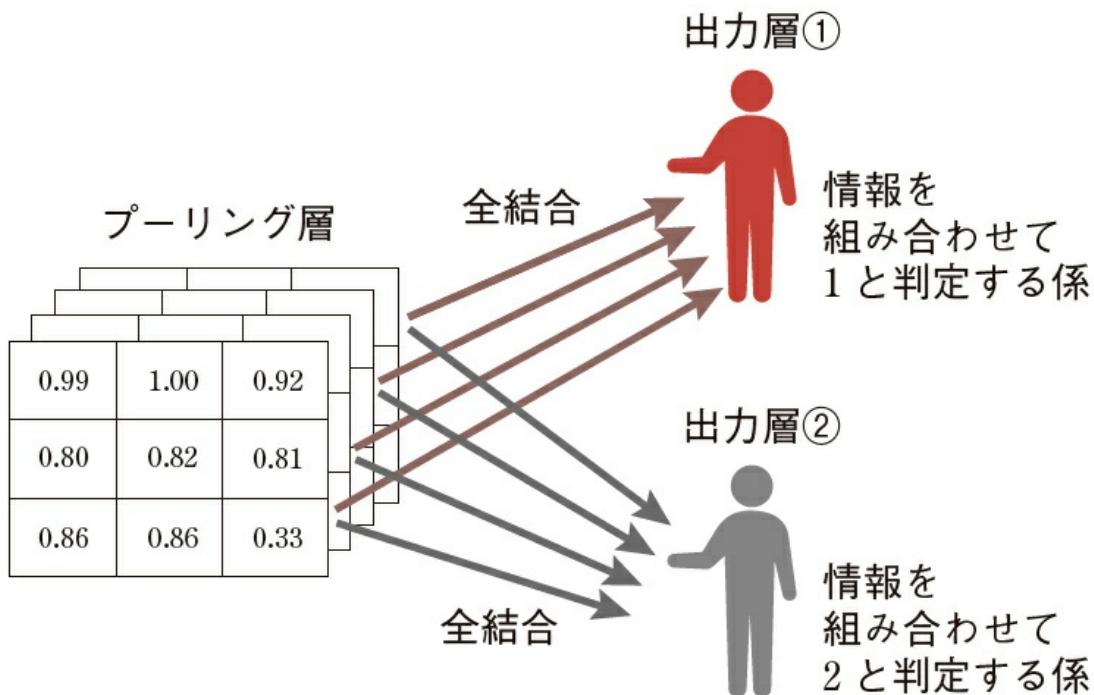
畳み込み層の情報はプーリング層でさらに4分の1に縮約される。

隠れ層は以上の畳み込み層とプーリング層の2つの層から構成されることになります。

■ 畳み込みニューラルネットワークの出力層の役割

次に出力層について考えましょう。この層にあるニューロンは「判定係」の役割を担いますが、その役目はニューラルネットワークのときと全く同じです。下の層(プ

リング層)からの情報を組み合わせ、自分が受け持つ文字とマッチするかの確信度を0と1の間の数値で表現します。いま調べている[テーマⅡ]では「1」と「2」の2文字の識別なので、「1」の確信度を数値化する判定係①と、「2」の確信度を数値化する判定係②の2者を配置します。



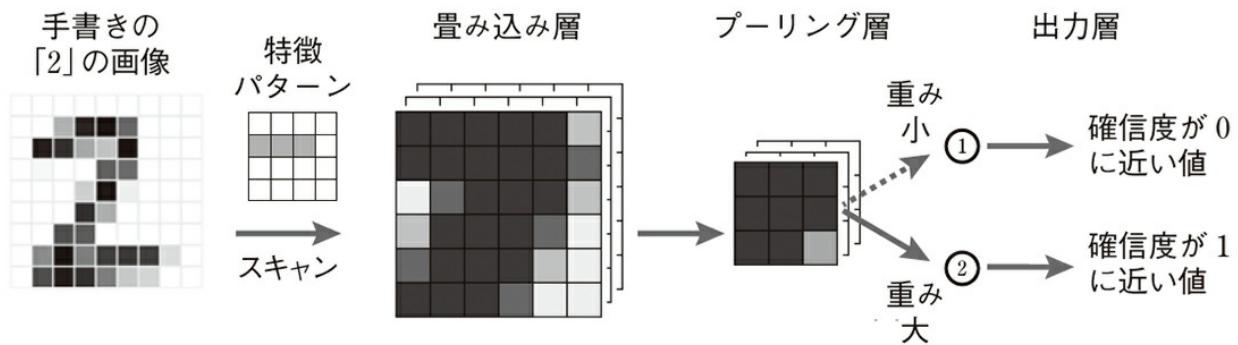
プリング層からの情報の組み合わせ方はニューラルネットワークのときと同様です。プリング層の各成分から引かれる矢に重みを付け、担当の数字かどうかの判断がしやすいように値を設定します。

こうして、出力層が完成しました。出力層のスタッフのくだす結論が畳み込みニューラルネットワーク全体の結論になります。

■ まとめてみよう

話をまとめてみましょう。次の図はこれまで作成した畳み込みニューラルネットワ

ークを利用して、手書きの数字を識別するしくみを表現しています。



大きい値を濃く表示している。データにマッチした特徴パターンを用いると、右に行くほど情報が濃縮していくことがみてとれる。

■ 閾値の役割

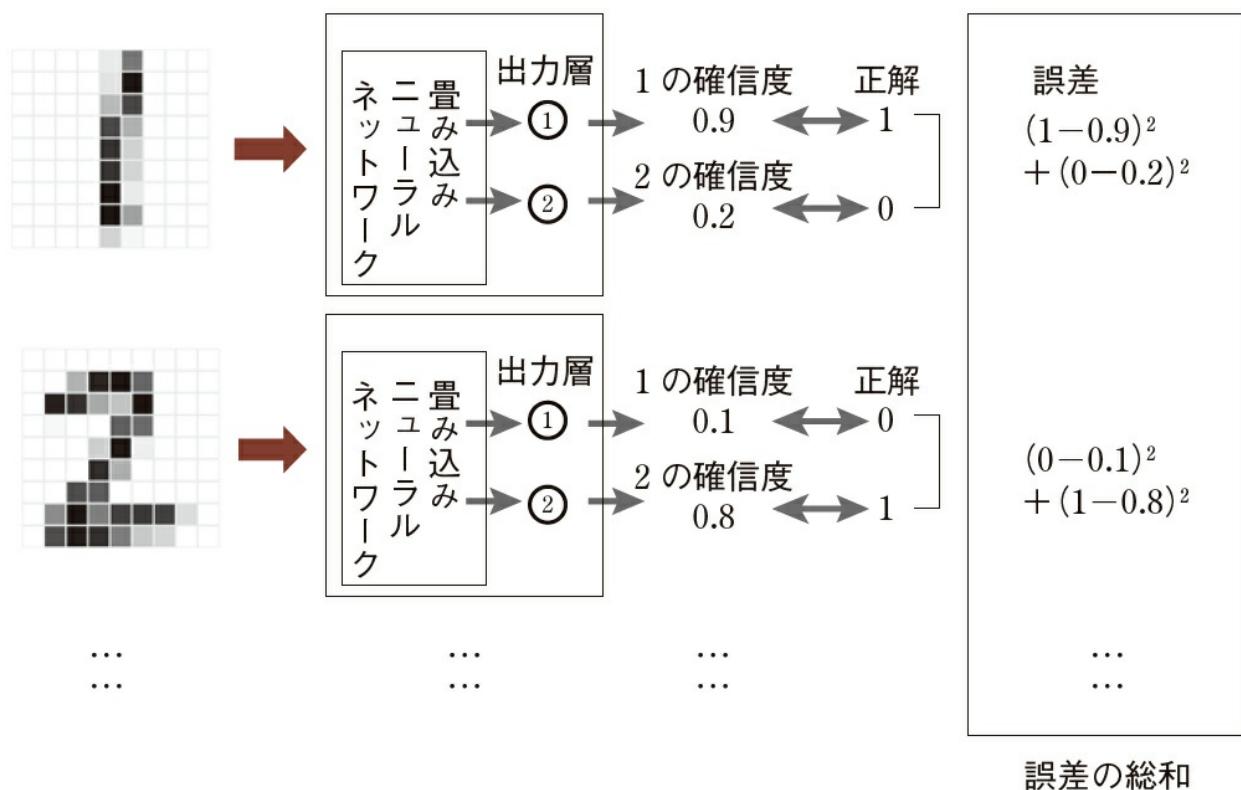
「特徴パターン」や「重み」を中心に話を進めてきましたが、ネットワークで大切な働きをする「閾値」について確認しましょう。

ニューラルネットワークのときと同様、閾値は黒子的な役割を演じ、イメージでは表現しにくいものです。しかし、その黒子の存在がなければ、ネットワークは機能しません。というのも、ネットワークの各係が情報を出力するとき、必要な情報だけをしっかり提示し、不要な情報をできるだけカットするよう、各スタッフに働きかける役割をするからです。それはニューラルネットワークの場合と全く同じ役割です（→4章§1、§7）。

■ パラメーターの決定法

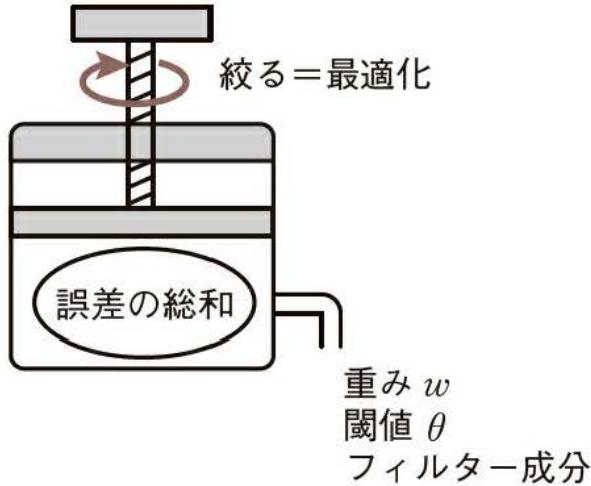
これまででは、特徴パターンや重み、閾値は与えられたものと仮定して話を進めてきました。しかし、それをどのような値として設定してよいかは、最初は不明です。それを決定するには、「ネットワーク自らが決定する」方法が用いられます。

これまで調べてきた[テーマⅡ]を例にして見てみましょう。訓練データには「1」「2」の手書き画像データが何枚もあり、それらには1枚ずつ「1」か「2」の正解がついています。すると、我々がやるべきことは、ニューラルネットワークに1枚1枚の画像を読ませ、「○」か「×」かの確信度を計算させることです。そして、1枚1枚の画像に付けられた正解との誤差を算出すればよいのです。後は、画像データすべてにおいてこれらの誤差の総和が最小値になるように、特徴パターン、重み、閾値をコンピューターで決めることがあります。



重みや閾値を決定するしくみ。表示している値は仮のものである。

以上の計算法は、ニューラルネットワークのときと同じで、数学モデルの最適化と呼ばれる技法であり、学習とも呼ばれます。人が教え込むことは何もありません。



人、すなわちコンピューターがやることは最適化の計算だけ。畳み込みニューラルネットワークに知恵を教え込むことはない。これが大きな特徴であり、「自らが決定する」と表現される中身である。

Memo AIは無定義用語

現代はAIブームと言われますが、そのAIについて明確な定義はなされていません。1章でも述べたように、以前には「AI炊飯器」「AI洗濯機」などという商品も販売されていましたが、その「AI」とはわれわれが想像するAIとは大きく異なっています。

われわれが想像するAIには、有名な古典漫画「鉄腕アトム」のイメージが投影されます。「人と話す」「考える」「感情がわかる」がAIの必要要素に思われます。しかし、現実にはそのようなAIは存在しません。

AIとは「知的な機械、特に、知的なコンピュータプログラムを作る科学と技術」(日本人工知能学会のホームページより)などという定義もありますが、そもそも「知的」とは何かについて議論が必要でしょう。要するに、AIは無定義用語なのです。

人工知能の研究には2つの立場があるといわれます。ひとつは人間の知能そのものを持つ機械を作ろうとする立場であり、もうひとつは人間が知能を使ってすることを機械にさせようとする立場です。実際の研究のほとんどは後者の立場に立っています。そうだからこそ、産業界ではAIが最大関心事のひとつ

になるわけです。そして、その分野に、本書のテーマである畳み込みニューラルネットワークの技術が確実に浸透しています。

§2

畳み込みニューラルネットワークが手書き数字を識別

前の節(§1)では畳み込みニューラルネットワークがどんなものを、人の役割にたとえて物語風に調べました。本節からは、Excelを用いながら、具体的に式を追ってしくみを調べることにしましょう。本節では、準備として訓練データを用意します。

■ ニューラルネットワークの限界

画像識別をニューラルネットワークで行うとき、対象となる画像が持つ画素数が大きくなると、隠れ層が重くなり、コンピューターの能力を使い切ってしまいます。前章最後の節(4章§9)では、その具体例を確認しました。この問題を解決するのが畳み込みニューラルネットワークです。近年、マスコミ等の話題をさらっている技法です。動的な中間層を設定することで、コンピューターの資源を有効活用するテクニックを利用します。

■ 手書き数字データの準備

本章の具体的なテーマとして、前節(§1)でも紹介した次の[テーマⅡ]を利用します。このテーマに沿いながら、畳み込みニューラルネットワークのしくみを、Excelを利用しながら順次調べていくことにしましょう。

■ **テーマⅡ** 9×9=81画素のモノクロ画像として読み取った手書きの数字

「1」、「2」を識別する畳み込みニューラルネットワークを、190枚の手書き画像

とその正解からなる訓練データから作成しよう。

この[テーマⅡ]は単純なものであり、実際にははるかに複雑な構造を持つものが利用されます。しかし、この内容の理解があれば、複雑なものにも応用が利くはずです。

最初に述べたように、本節では訓練データを準備します。訓練データとしては、次に示す「1」、「2」の計190文字を数字画像として利用します。ちなみに、数字画像が何を示すかの正解を付加しなければなりませんが、人はこれらの画像を見て1か2を区別できるので、紙面では省略します。

```
/ 1 1 1 / 2 2 2 2 2 1 1 / 1 2 1 2 2 1 / 1 3  
1 1 2 1 2 2 2 3 1 1 1 1 1 2 3 1 / 1 2 1  
1 1 2 2 2 2 2 2 / 1 1 1 2 2 2 2 2 1 2 2  
1 2 2 2 2 2 1 2 / 2 2 2 2 2 2 2 2 1 1 2  
1 2 1 1 2 2 1 1 2 2 2 1 2 2 2 2 1 / 2 2 1  
2 7 2 1 1 2 2 1 1 1 2 2 2 1 2 1 1 2 /  
1 2 2 2 2 3 2 2 2 1 1 1 2 2 / 2 2 2 /  
2 1 2 1 2 2 1 1 1 2 2 2 1 / 1 1 1 1 1  
1 2 1 2 2 1 1 1 2 1 2 1 2 1 2 2 2 2 2 /  
1 1 2 2 1 2 2 1 2 1
```

注 この手書き数字の「0」と「1」は、MNISTデータから1と2の手書き数字を190文字だけピックアップし、判別できる最小の解像度(9×9画素)に縮小したものです。拡大図は付録Bに掲載しました。なお、MNISTデータについては、下記《メモ》を参照してください。

Memo | MNISTデータ

MNIST(Mixed National Institute of Standards and Technology)とは28×28画素の60000枚の学習サンプルと10000枚のテストサンプルからなる手書き数字画像データベースのことです。NISTとはアメリカ国立標準技術研究所で、この機関が用意してくれた見本データがNMISTなのです。本書

では、これを解読可能な最小の解像度の9×9に変換したものを利用します。

なお、MNISTの詳細は次のWebを見てみましょう。

<http://yann.lecun.com/exdb/mnist/>

それでは、Excelに上記の訓練データを入力してみましょう。

例題 9×9画素のモノクロ画像として読み取った手書き数字「1」と「2」をExcelに入力しましょう。

注 この例題のワークシートは、ダウンロードサイト（→卷頭「Excelサンプルファイルのダウンロードについて」）に掲載されたファイル「5.xlsx」の中の「Data」タブに収められています。なお、付録Bにその手書き数字イメージを掲載しております。

解

本書指定のサイトからExcelワークシート(5.xlsx)をダウンロードすると、次の訓練データが得られます。内容については、ダウンロードしたファイルを開いて確かめてください。

	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	/
番号	1										2										3
画像	0	0	0	0	0	0	107	213	0	0	0	0	171	5	0	0	0	0	0	0	
	0	0	0	0	0	33	249	7	0	0	0	0	193	94	0	0	0	0	0	0	
	0	0	0	0	52	242	74	0	0	0	0	0	193	138	0	0	0	0	0	0	
	0	0	0	5	245	153	0	0	0	0	0	0	78	138	0	0	0	0	0	0	
	0	0	0	159	221	1	0	0	0	0	0	0	193	138	0	0	0	0	0	0	
	0	0	32	248	17	0	0	0	0	0	0	0	193	138	0	0	0	0	0	0	
	0	3	234	83	0	0	0	0	0	0	0	0	236	138	0	0	0	0	0	0	
	0	17	244	74	0	0	0	0	0	0	0	0	229	115	0	0	0	0	0	0	
正解	1										1									1	

訓練データはダウンロード
したものを利用

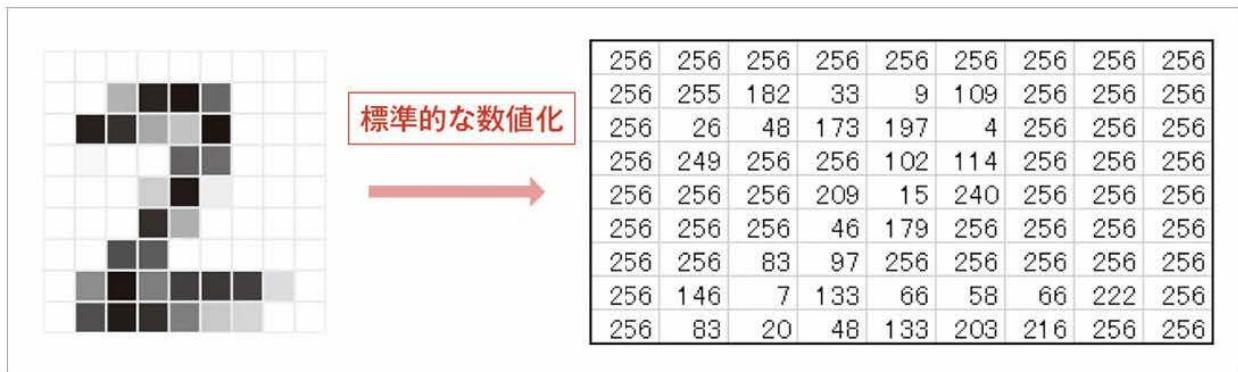
BMM	BMN	BMO	BMP	BMQ	BMR	BMS	BMT	BMU	BMV	BMW	BMX	BMY	BMZ	BNA	BNB	BNC	BND	BNE
	189									190								
0	0	0	28	129	72	1	0	0	0	0	0	0	0	74	88	0	0	0
0	0	12	252	118	242	29	0	0	0	0	0	0	0	189	57	0	0	0
0	0	0	15	0	234	72	0	0	0	0	0	0	0	228	38	0	0	0
0	0	1	140	77	253	13	114	197	83	0	0	0	0	252	11	0	0	0
0	49	250	225	254	163	6	101	235	2	0	0	0	2	228	0	0	0	0
0	38	249	238	133	251	249	184	7	0	0	0	0	0	40	202	0	0	0
0	7	76	0	0	19	5	0	0	0	0	0	0	0	52	159	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	6	225	0	0	0	0
R28C21	2									1								

注 この訓練データを取り込んだワークシートのタブ名は「Data」とします。

■ 画像の確認

ここで、数値化された画像の形式について確認します。モノクロ画像は各画素に0～255までの数値が割り振られ、明るさを表します。標準では0が黒を、255が白を表現します。しかし、手書き数字との対応を画面上で調べるとき、それでは不便です。そこで、本書では、白と黒の数値を反転して表示しています。フィルム写真のネガを示していると考えると、わかりやすいでしょう。このような変換をしても得られる結果は同じですが、反転した方が直感的な解釈がしやすいと思われます。このことは、すでに4章のニューラルネットワークのときにも利用しています。

<標準的なデータの数値化>



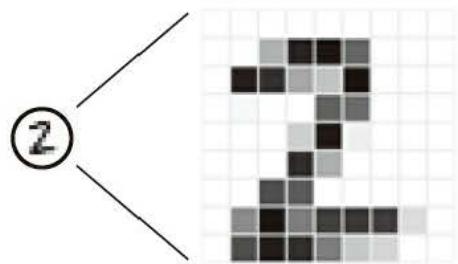
<本書で用いる明暗反転した数値化>



注 上の表と下の表の対応する数値を加えると、値は256になります。ちなみに、明暗を反転した画像は昔のアナログフィルムの「ネガ」に相当します。

■ 訓練データの正解

例題のためにダウンロードしたファイルを見ると、そこに「正解」の欄があります。それは、対象の画像が何を表しているかを示すものです。何度も述べているように、このような正解とセットのデータを訓練データ(**training data**)といいます。正解が必要なのは、与えられた画像が何を意味するか、コンピューターには不明だからです。



正解
= 2

この手書き数字が2であることを
コンピューターに教えるためには
正解「2」が必要。このセットの集
まりが訓練データ。

§ 3

畳み込みニューラルネットワークの入力層

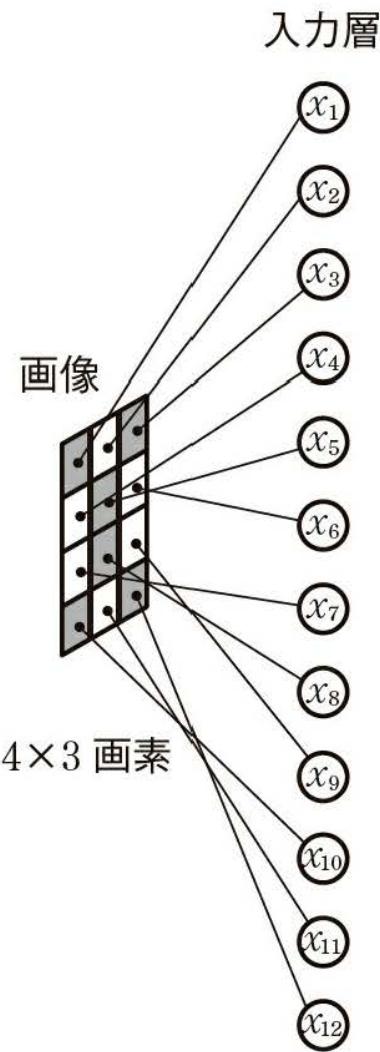
実際にExcelで畳み込みニューラルネットワークを動作させるためには、ニューロンの関係を明示しなければなりません。そのことを、まず入力層で調べます。

■ 入力層を格子状に表現

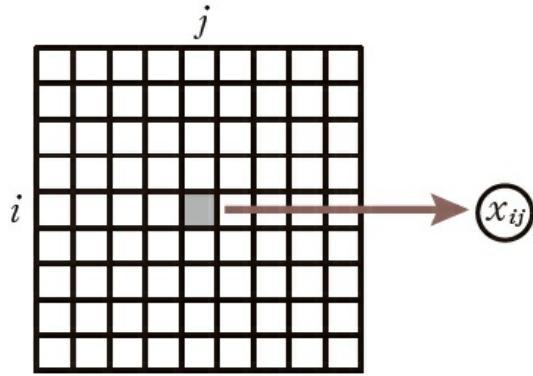
4章のニューラルネットワークのときにも調べましたが、入力層にあるニューロンはデータに何も作用しません。画素からの信号を(定数倍を除いて)そのままネットワークに取り入れる働きをします。その入力層にあるニューロンの名称として、4章と同じく、ローマ字 x を用いることにします。

注 次節以降の話ですが、前章同様、隠れ層は y 、出力層は z を基本的に利用します。

ところで、前章で調べたニューラルネットワークでは、次の図に示すように、入力層のニューロンの名称に単純に1～12までの添え字を変数名に配分しました(12は画素数です)。

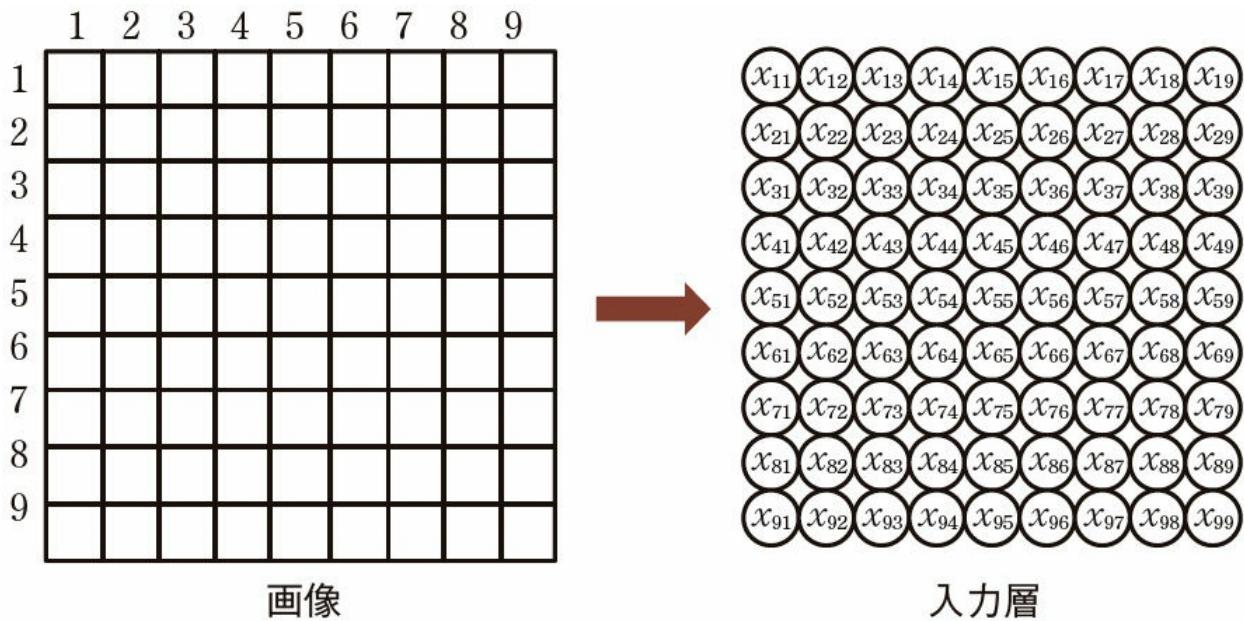


しかし、画素数が大きくなると、この命名法は分かりにくくなります。本章で扱う画素数は $9 \times 9 = 81$ なので、81番までの番号を x に添え字として付けていくことになり、見にくくなるからです。また、画像との対応が不明になってしまいます。そこで、画像の i 行 j 列にある画素に対応する入力層のニューロンには、 x_{ij} という名称を付けることにします。



画像の中の i 行 j 列にある画素の信号を受ける入力層のニューロン名が x_{ij} 。

こうすることで、画素と入力層のニューロンとの対応が一目瞭然になります。対応をすべて書き出すと、次のようにになります。

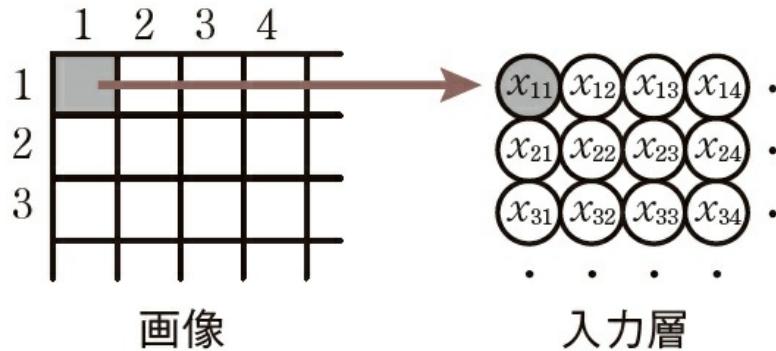


ニューロンの名称は、ニューラルネットワークのときと同様、出力変数の名称としても利用することにします。すると、入力層のニューロンは信号に何も作用を加えないでの、一般的には次の関係が成立します。

$$x_{ij} = \text{入力層 } i \text{ 行 } j \text{ 列にあるニューロンの名称}$$

$$= \text{入力層 } i \text{ 行 } j \text{ 列にあるニューロンの出力値} \dots \dots \dots (1)$$

例1 x_{11} は画像の1行1列の位置にある画素信号を受けるニューロン名、及びその出力です(次図)。



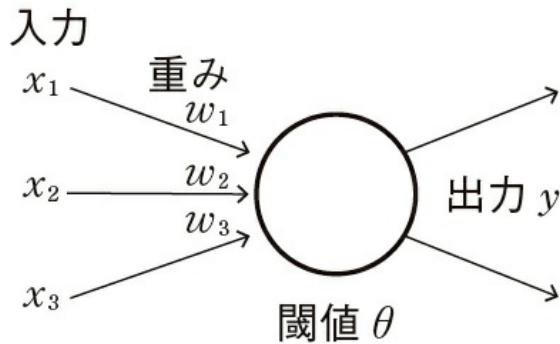
■ 入力層のニューロンをExcelで表現

のちのソルバーによる最適化のために、本書は入力層の出力を画素信号の100分の1に変換することにします。こうすることで、Excelのソルバーの計算スピードと収束性が向上するからです(→下記の《メモ》参照)。

Memo 出力値をなぜ画素値の1/100倍にしたのか？

例えば、次図のようなニューロンを考えてみましょう。3つの入力信号 x_1, x_2, x_3 を考え、各入力信号には重み w_1, w_2, w_3 が与えられるとします。閾値を θ とするとき、ニューロンが得る入力の線形和 a は次の形をとります。

$$a = w_1 x_1 + w_2 x_2 + w_3 x_3 - \theta$$



さて、入力信号 x_1, x_2, x_3 を各々 k 倍し、重み w_1, w_2, w_3 を $1/k$ 倍するとき、この和 a の値は同じです。

$$a = \frac{1}{k} w_1 (kx_1) + \frac{1}{k} w_2 (kx_2) + \frac{1}{k} w_3 (kx_3) - \theta$$

すなわち、入力信号の大きさ x_1, x_2, x_3 をどんな尺度で表示しても、重みの大きさを変えることで、和 a の値は変わらないのです。本節では、画像からの入力信号を勝手に $1/100$ 倍しましたが、重みに条件がなければ、数学的には全く問題はないのです。

では、どうして $1/100$ 倍したのでしょうか？ それはソルバーのアルゴリズムの問題です。ソルバーは変数セルの値を少しずつ変更しながら最適値を探しますが、そのデフォルト値は 0.0001 です。したがって、このデフォルト値を変更しない場合、ソルバーが扱う値が大きいと、その最適化に手間どってしまうのです。入力層のニューロンの値を画素の 100 分の 1 としたのは、このような事情です。

例題 訓練データの1番目の数字画像について、画素からの信号を $1/100$ 倍して出力することにして、入力層のニューロンの出力を求めましょう。また、その正解は (t_1, t_2) の形式で出力層セルの左下に配置しましょう。読み込み画像が「1」のときに $(1, 0)$ とし、「2」のときに $(0, 1)$ とします。

注 この例題のワークシートは、ダウンロードサイト（→卷頭「Excelサンプルファイルのダウンロードについて」）に掲載されたファイル「5.xlsx」の中の「例題」タブに収められています。

解

次図のように、セル番地L3から始まる9×9のセル範囲に題意に即した出力値を用意します。更に、セル番地L12とその右隣に、手書き数字の正解を設定します。

J	K	L	M	N	O	P	Q	R	S	T
番号		1								
入力層		0	0	0	0	0	0	1.07	2.13	0
		0	0	0	0	0	0.33	2.49	0.07	0
		0	0	0	0	0.52	2.42	0.74	0	0
		0	0	0	0.05	2.45	1.53	0	0	0
		0	0	0	1.59	2.21	0.01	0	0	0
		0	0	0.32	2.48	0.17	0	0	0	0
		0	0.03	2.34	0.83	0	0	0	0	0
		0	0.17	2.44	0.74	0	0	0	0	0
正解(t ₁ ,t ₂)		1	0							

1番目の画像が読みこまれたときの、入力層の出力。画素値を1/100倍している

正解の欄

これを訓練データにわたってコピーする必要がありますが、その操作は最後に回します（→本章§8）。

■ 正解の表現法

訓練データでは、各文字画像に、その画像が何を表しているかの正解が付けられています。それを左記の例に示すようにセットしたわけです。このセットの形式は、4章で調べたニューラルネットワークのときと同様です。のちに調べる最適化のための誤差表現をしやすくするために、このような表現を利用します。

なお、4章でも調べたように、変数 t_1 、 t_2 は次のように定義されています。使い方については、のちの§7で調べましょう。

	意味	画像が「1」	画像が「2」
t_1	「0」の正解変数	1	0
t_2	「1」の正解変数	0	1

§ 4

畳み込みニューラルネットワークを特徴づける畳み込み層

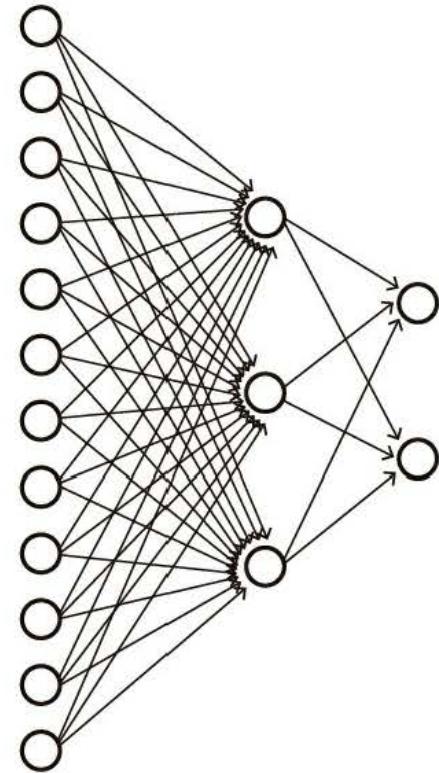
畳み込みニューラルネットワークの中心テーマとなる「畳み込み層」と呼ばれる層の役割を調べます。Excelで畳み込みニューラルネットワークを実装すると、そのしくみがよく見えます。

■ 畳み込みニューラルネットワークは入力層を小分けに調べる

本章§1では、畳み込みニューラルネットワークとはどのようなものかについて擬人的な説明をしました。ここでは具体的に式と数値で話を進めます。重複する部分もありますが、詳しく調べることにしましょう。

先に調べたように(→4章§9)、ニューラルネットワークを現実データに適用すると、ニューロンを結ぶ矢の数が膨大になり、決めるべきモデルのパラメーター数がコンピューターの能力を超えてしまいます。その原因は入力層のニューロンと隠れ層のニューロンがすべて結合されていたからです。このような結合の仕方を全結合といいます。

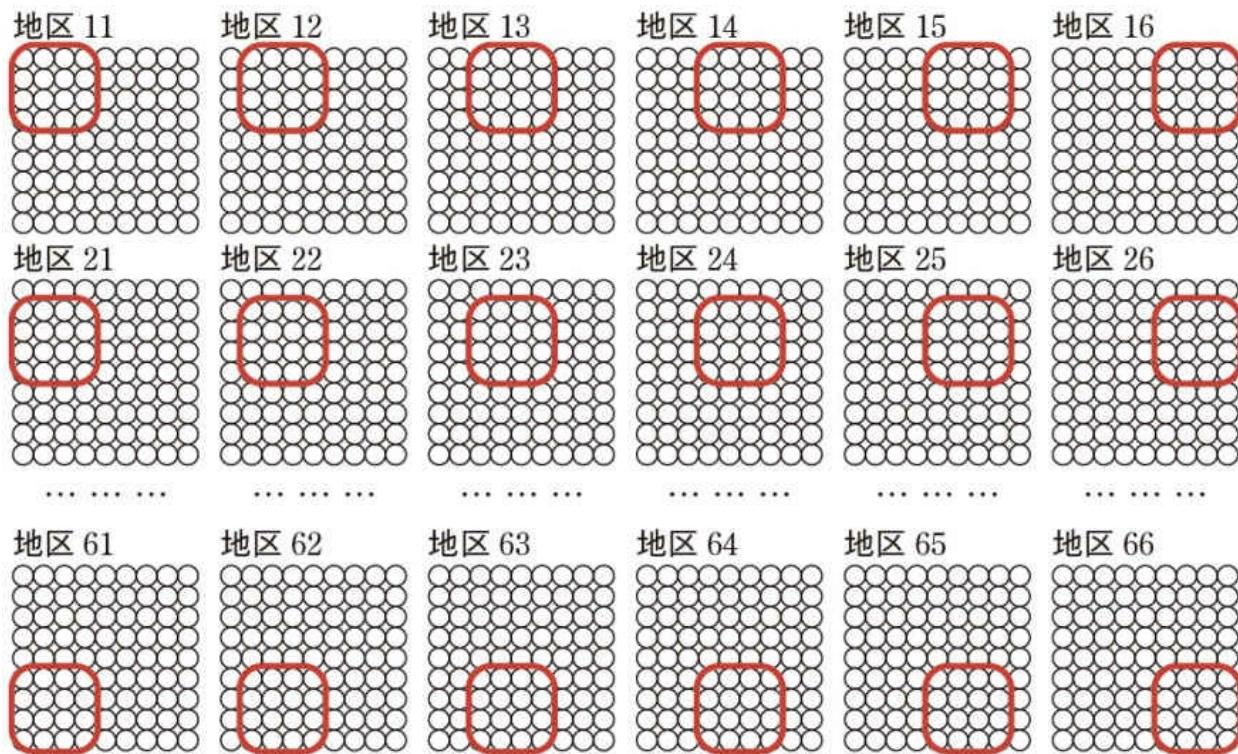
入力層 隠れ層 出力層



4章で調べたニューラルネットワーク。ニューラルネットワークでは、上の層と下の層のニューロンがすべて矢で結ばれている。すなわち、全結合している。

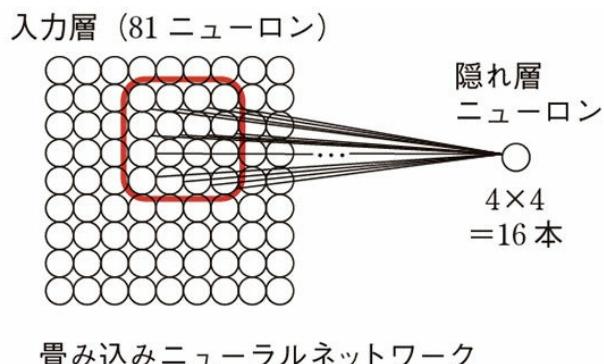
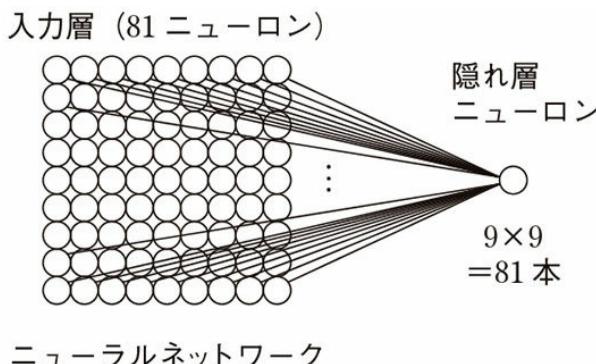
畳み込みニューラルネットワークがこの問題を解決する手段として取り入れた方法が、入力層のニューロンを小分けにして調べる方法です。入力層のニューロンを小分けすることで、隠れ層のニューロンが調べる矢の本数が少なくてすみます。すると、その矢に課す「重み」の個数も少なくてすむのです。

では、実際に 9×9 の画素数の画像を次図のように 4×4 の大きさに小分けしてみましょう。画像は $6 \times 6 (=36)$ 地区に分けられます。



注 先にも言及したように(→§1)、多くの畳み込みニューラルネットワークは 5×5 の大きさを小分けの大きさとして採用します。本書では記述を簡略化するために、より小さな 4×4 の大きさを採用します。

小分けした各地区について、隠れ層のニューロンは、ニューラルネットワークのときと同じ処理を行います。小分けしたぶん、その隠れ層のニューロンが矢に課す「重み」は 4×4 個だけですむことになります。(全結合すると、 9×9 個の「重み」が必要になります！)



小分けして処理する代償として、1つのニューロンの出力する値は単一ではなくなります。先の図が示すように、隠れ層のニューロンが調べる地区は地区11、地区12、地区13、…、地区66の計 $6 \times 6 = 36$ 個なので、次のような表形式で表現されることになります。

地区 11 の出力	地区 12 の出力	…	地区 16 の出力
地区 21 の出力	地区 22 の出力	…	地区 26 の出力
……	……	……	……
地区 61 の出力	地区 62 の出力	…	地区 66 の出力

隠れ層のニューロンの出力は表の形式になる。

ところで、隠れ層には複数のニューロンが配置されています。そこで、隠れ層のニューロンの出力する表は複数枚になります。この出力表のセットを畳み込み層といいます。

地区 11 の出力	地区 12 の出力	…	地区 16 の出力	
地区 11 の出力	地区 12 の出力	…	地区 16 の出力	力
地区 11 の出力	地区 12 の出力	…	地区 16 の出力	力
地区 11 の出力	地区 12 の出力	…	地区 16 の出力	力
地区 21 の出力	地区 22 の出力	…	地区 26 の出力	力
……	……	……	……	力
地区 61 の出力	地区 62 の出力	…	地区 66 の出力	…

畳み込み層の出力表
隠れ層のニューロンが入力層のニューロンを小分けして処理した結果の表の集まりである。

この図で、畳み込み層を構成する1枚1枚の表を「畳み込み層の出力表」と呼びことにします。

先に示したように、小分けした各地区について、隠れ層のニューロンは、ニューラルネットワークのときと同じ処理を行います。すると、ニューラルネットワークで調べたように、各地区について特徴抽出をすることになります。そして、小分けしたぶん、調べる範囲が絞られ、そのぶん特徴抽出をしやすくなるのです。その特徴抽出の

結果が上記の出力表にまとめられているのです。

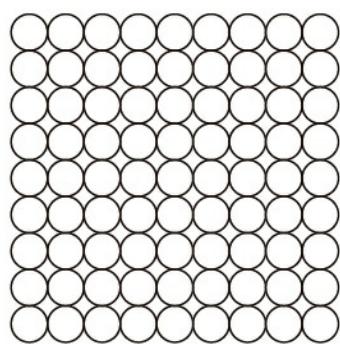
■ 隠れ層に3つのニューロンを配置

以上が畳み込み層の概略です。以下では、具体的に話を進めましょう。

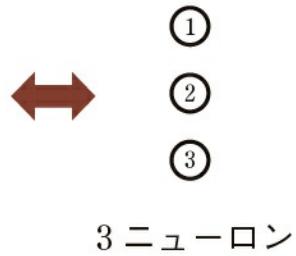
最初に、畳み込みニューラルネットワークの構造を決定します。4章と比較しやすいように、本章でも隠れ層のニューロン数は3個とします。

注 隠れ層に何個のニューロンが必要かは、理論的には決められません。訓練データの性質によるからです。試行錯誤するしかありません。

入力層 (81 ニューロン)



隠れ層



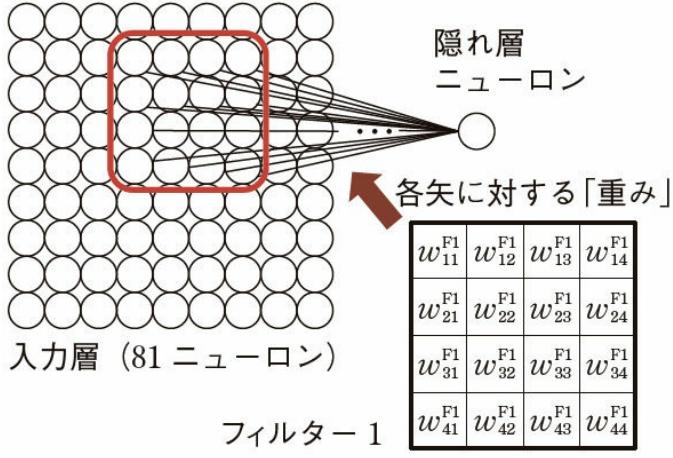
3 ニューロン

§1で調べたように、隠れ層に3つのニューロンを配置。

いま、隠れ層の1番目のニューロン①について見てみましょう。

6×6個の小分けされた各地区に対して、隠れ層のニューロンは、4章で調べたニューラルネットワークのときと同じ処理を行います。このとき、小分けされた入力層のニューロンに与える「重み」は次図のように表で表現できます。これら「重み」の表を本書ではフィルターと呼ぶことにします。そして、フィルターを構成する各値を「フィルターの成分」と呼ぶことにします。

注 文献によっては、フィルターをカーネルなどと呼んでいます。



フィルターは小分けされた全地区に共通。フィルターの成分の記号の w は重み (weight)、F はフィルター (Filter) の頭文字。F1 は隠れ層のニューロン①用のフィルターであることを表す。

隠れ層には3つのニューロン①～③を仮定しました。そこで、このようなフィルターは次図のように3枚用意されます。

フィルター 1

w_{11}^{F1}	w_{12}^{F1}	w_{13}^{F1}	w_{14}^{F1}
w_{21}^{F1}	w_{22}^{F1}	w_{23}^{F1}	w_{24}^{F1}
w_{31}^{F1}	w_{32}^{F1}	w_{33}^{F1}	w_{34}^{F1}
w_{41}^{F1}	w_{42}^{F1}	w_{43}^{F1}	w_{44}^{F1}

フィルター 2

w_{11}^{F2}	w_{12}^{F2}	w_{13}^{F2}	w_{14}^{F2}
w_{21}^{F2}	w_{22}^{F2}	w_{23}^{F2}	w_{24}^{F2}
w_{31}^{F2}	w_{32}^{F2}	w_{33}^{F2}	w_{34}^{F2}
w_{41}^{F2}	w_{42}^{F2}	w_{43}^{F2}	w_{44}^{F2}

フィルター 3

w_{11}^{F3}	w_{12}^{F3}	w_{13}^{F3}	w_{14}^{F3}
w_{21}^{F3}	w_{22}^{F3}	w_{23}^{F3}	w_{24}^{F3}
w_{31}^{F3}	w_{32}^{F3}	w_{33}^{F3}	w_{34}^{F3}
w_{41}^{F3}	w_{42}^{F3}	w_{43}^{F3}	w_{44}^{F3}

右上の添え字 Fk は隠れ層のニューロン k の用いるフィルターを表す。

さて、隠れ層の3つのニューロンは、ニューラルネットワークのときと同様、「閾値」を持ちます。これを次のように表現しましょう。

θ^{Fk} … 隠れ層 k 番目のニューロンの閾値 ($1 \leq k \leq 3$)

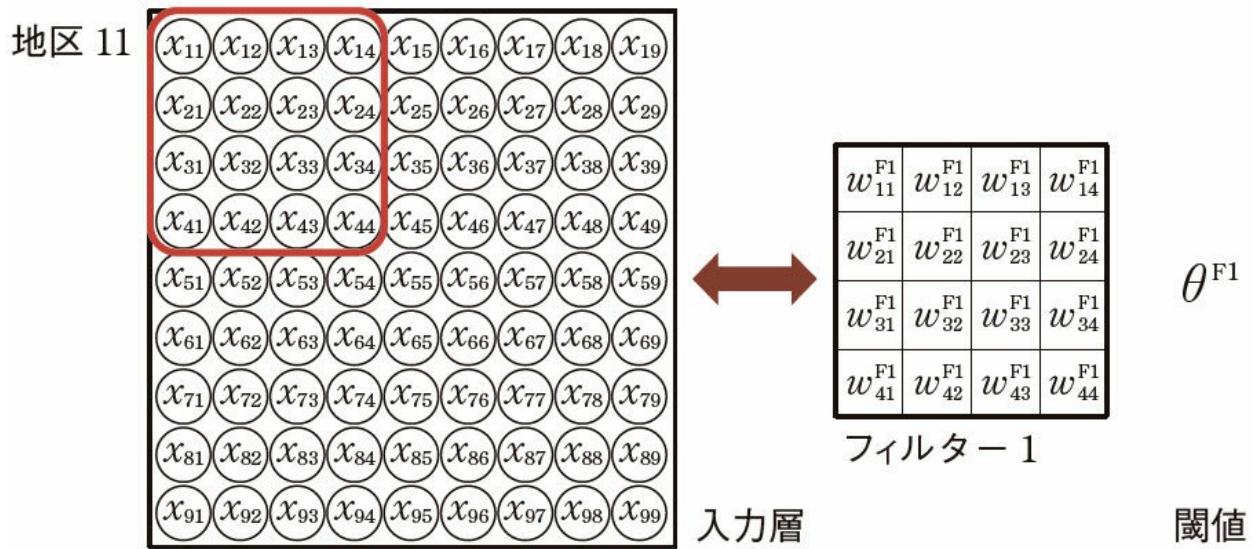
隠れ層のニューロン①に関するパラメーター記号をまとめましょう。

フィルターの成分 w_{11}^{F1}	フィルター名
フィルター内的位置	閾値 θ^{F1}

■ 隠れ層のニューロンの出力を式に表現

隠れ層のニューロンの目を通して、小分けした各部分を調べてみましょう。具体的に、隠れ層の1番目のニューロン①が、入力層の地区11をどのように扱うかを調べてみます。

最初に、扱う記号名の位置関係を次図に示します。



前にも述べましたが、小分けした部分に対する処理の仕方はニューラルネットワークと同じです。すると、この地区11を入力層としたとき、隠れ層のニューロン①に関する「入力の線形和」 y_{11}^{F1} は次のように表せます。

$$a_{11}^{F1} = w_{11}^{F1} x_{11} + w_{12}^{F1} x_{12} + w_{13}^{F1} x_{13} + \cdots + w_{44}^{F1} x_{44} - \theta^{F1} \quad \cdots (1)$$

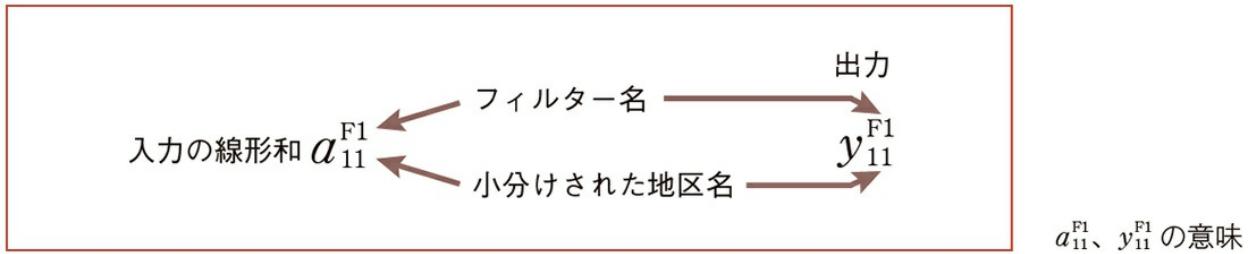
すると、隠れ層のニューロン①の出力値 a_{11}^{F1} は、 σ をシグモイド関数として、次の

ように表せます。

$$y_{11}^{F1} = \sigma(a_{11}^{F1}) \cdots (2)$$

この y_{11}^{F1} は畳み込み層1枚目の出力表の1行1列目の成分になります。

注 先にも示したように、本書は隠れ層の出力には文字 y を用います。



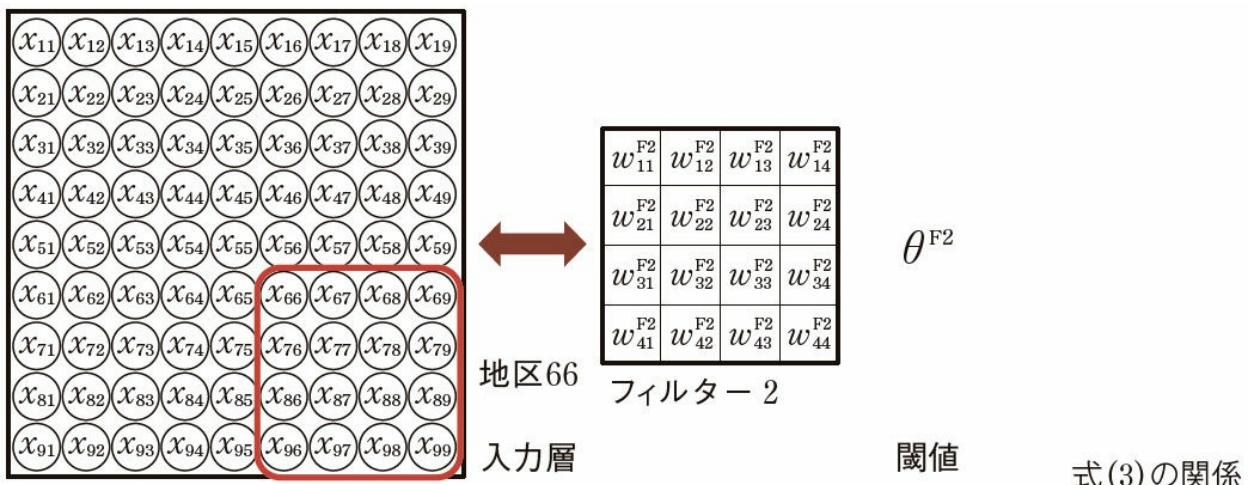
例1 入力層の小分けした地区66(本節の「畳み込みニューラルネットワークは入力層を小分けに調べる」の図参照)に対して、出力層2番目のニューロン②の出力 y_{66}^{F2} を式で表現しましょう。

地区66からこのニューロン②への「入力の線形和」 a_{66}^{F2} は次のようにになります。

$$a_{66}^{F2} = w_{11}^{F2} x_{66} + w_{12}^{F2} x_{67} + w_{13}^{F2} x_{68} + \cdots + w_{44}^{F2} x_{99} - \theta^{F2} \cdots (3)$$

すると、このニューロンの出力は次の値になります。

$$y_{66}^{F2} = \sigma(a_{66}^{F2}) \cdots (4)$$



この式(4)の y_{66}^{F2} は畳み込み層2枚目の出力表の6行6列目の成分になります。

■ 畳み込み層を具体化

式(2)(4)で得られた隠れ層の出力結果は、次の図のような表にまとめられます。隠れ層に3つのニューロンを仮定したので、出力の表は3枚になります。この1枚1枚が本節の「畳み込みニューラルネットワークは入力層を小分けに調べる」に示した「畳み込み層の出力表」の具体例です。

畳み込み層					
y_{11}^{F3}	y_{12}^{F3}	y_{13}^{F3}	y_{14}^{F3}	y_{15}^{F3}	y_{16}^{F3}
y_{11}^{F2}	y_{12}^{F2}	y_{13}^{F2}	y_{14}^{F2}	y_{15}^{F2}	y_{16}^{F2}
y_{21}^{F1}	y_{22}^{F1}	y_{23}^{F1}	y_{24}^{F1}	y_{25}^{F1}	y_{26}^{F1}
y_{31}^{F1}	y_{32}^{F1}	y_{33}^{F1}	y_{34}^{F1}	y_{35}^{F1}	y_{36}^{F1}
y_{41}^{F1}	y_{42}^{F1}	y_{43}^{F1}	y_{44}^{F1}	y_{45}^{F1}	y_{46}^{F1}
y_{51}^{F1}	y_{52}^{F1}	y_{53}^{F1}	y_{54}^{F1}	y_{55}^{F1}	y_{56}^{F1}
y_{61}^{F1}	y_{62}^{F1}	y_{63}^{F1}	y_{64}^{F1}	y_{65}^{F1}	y_{66}^{F1}

隠れ層のニューロンが処理した結果を表に整理したもの。各表を「畳み込み層の出力表」と呼ぶことは、本節前述の「畳み込みニューラルネットワークは入力層を小分けに調べる」で言及。

隠れ層の3つのニューロンの活躍した結果がこの3枚の表です。小分けした分、「重み」の個数は少なくてすみますが、出力は3枚の表となるのです。この隠れ層のニューロンの出力結果が、畳み込み層と呼ばれるのです。先に示したように、これら畳み込み層の1枚1枚の表を、畳み込み層の出力表と呼ぶことにしますが、フィルターの目を通した特徴抽出の結果の集大成です。

■ 置み込み層の出力の公式化とその意味

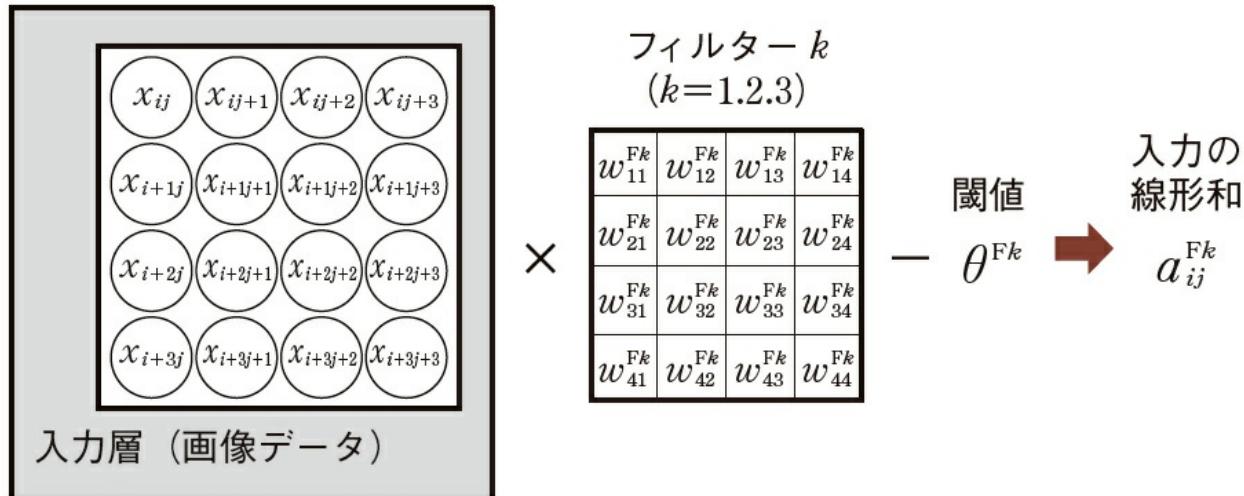
式(1)～(4)の作り方さえ理解していれば、Excelのワークシートで置み込み層を実装するは容易です。しかし、他の文献を読む際の確認として、あえてこれまでの結果を公式としてまとめましょう。

隠れ層 k 番目のニューロンが入力層の地区 ij を入力とするとき、その線形和を a_{ij}^{Fk} 、出力値を y_{ij}^{Fk} とすると、次のように表せる。

$$a_{ij}^{Fk} = w_{11}^{Fk} x_{ij} + w_{12}^{Fk} x_{ij+1} + w_{13}^{Fk} x_{ij+2} + \cdots + w_{44}^{Fk} x_{i+3j+3} - \theta^{Fk} \quad \cdots (5)$$

$$y_{ij}^{Fk} = \sigma(a_{ij}^{Fk}) \quad (\sigma \text{ はシグモイド関数}) \quad \cdots (6)$$

式(5)の記号の関係を図示します。



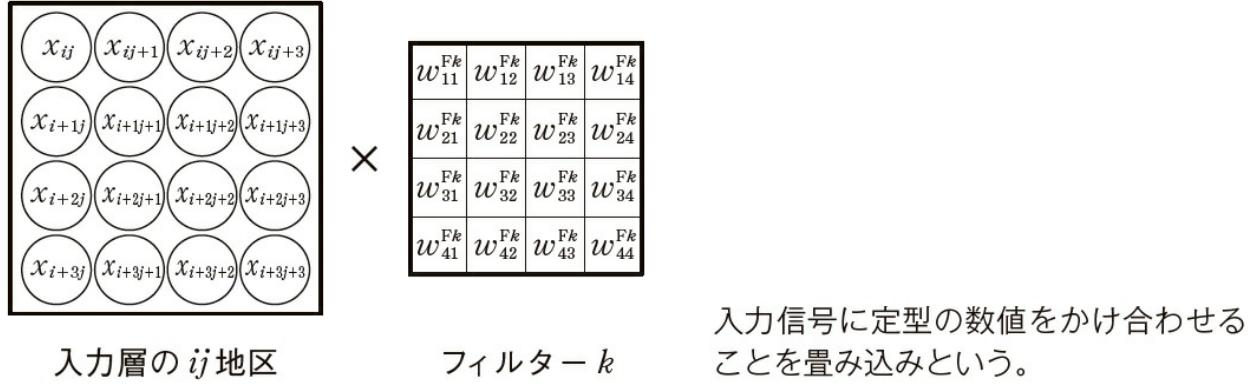
入力層の該当地区にあるニューロンの出力とフィルターとをかけ合わせ、閾値を引いたものが線形和(5)。

■ 置み込みの意味

公式(5)を見てください。式の中で、閾値を除いた次の和を考えます。

$$w_{11}^{Fk} x_{ij} + w_{12}^{Fk} x_{ij+1} + w_{13}^{Fk} x_{ij+2} + \cdots + w_{44}^{Fk} x_{i+3j+3} \cdots \quad (7)$$

この和をフィルターによる画像成分の畳み込みといいます。

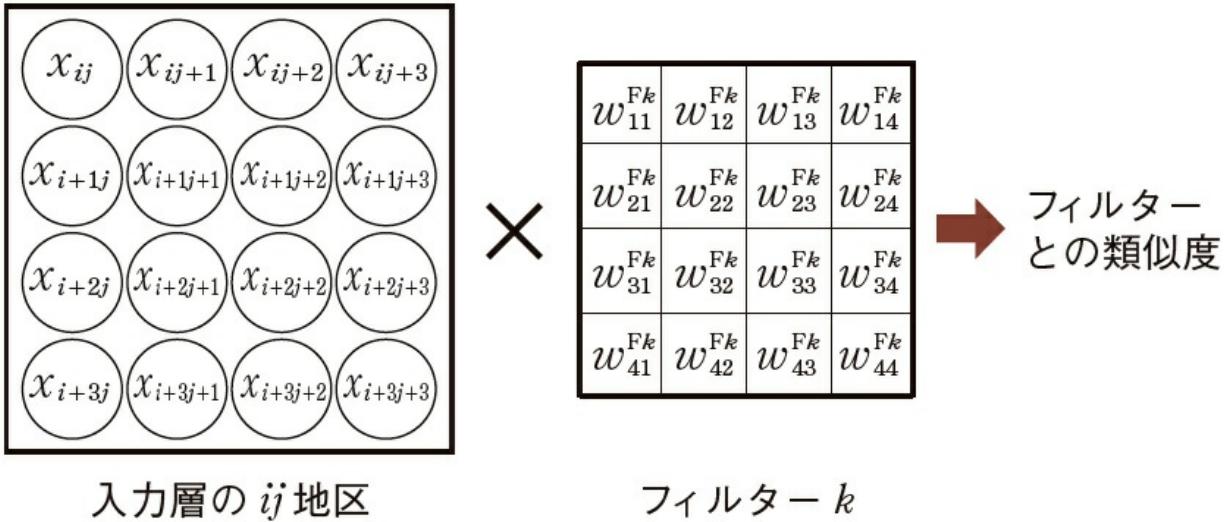


フィルターの成分を決定するのが、本章の目的の一つですが、ここではフィルターを固定して、この畳み込みの意味を考えてみましょう。

数学の世界で、式(7)は次の2つのベクトルの内積です。

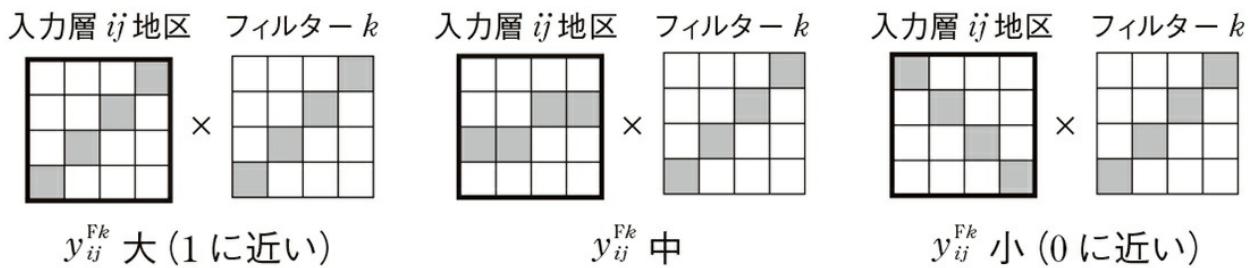
$$(x_{ij}, x_{ij+1}, x_{ij+2}, \dots, x_{i+3j+3}) \cdot (w_{11}^{Fk}, w_{12}^{Fk}, w_{13}^{Fk}, \dots, w_{44}^{Fk})$$

よく知られているように、2つのベクトルの内積の値が大きいと2つのベクトルは似ていることを示します(→付録D)。ということは、式(7)の和は、調べている地区とフィルターとの類似度を算出したことになります。



ところで、式(5)はこの(7)から定数の閾値を引いたものですから、類似度という意味に変わりはありません。さらに、それを用いて算出した式(6)の結果の y_{ij}^{Fk} も、シグモイド関数 σ が単調増加であることを考えると、やはり類似度という意味を保持します。ようするに、式(6)で得られた y_{ij}^{Fk} は入力層の区画とフィルターとの類似度を表しているのです。

畳み込みニューラルネットワークの算出結果を解釈するとき、この「類似性」という見方は大切になります。畳み込み層の出力表の各成分は、調べている地区の画素パターンとフィルターの画素パターンとが似ているかどうかを表しているのです。



■ 畳み込み層の効果

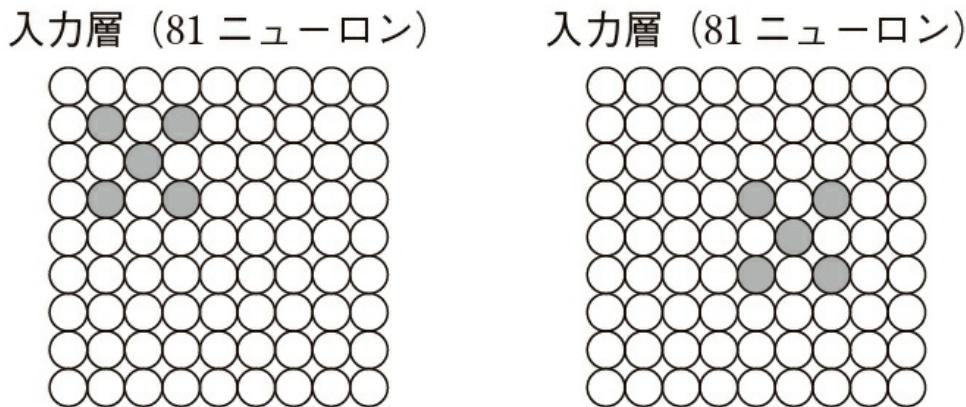
フィルターを利用して畳み込み層を作成するメリットは「パラメーターを少なくする

ため」と説明してきました。実際、ニューラルネットワーク及び畳み込みニューラルネットワークの隠れ層で利用するパラメーター数を比較してみましょう。

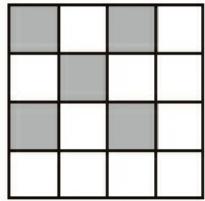
	ニューラルネットワーク	畳み込みニューラルネットワーク
重み w	$3 \times (9 \times 9) = 243$ 個	$3 \times (4 \times 4) = 48$ 個
閾値	3 個	3 個
計	246 個	51 個

数学的な意味では、このようなパラメーター数の減少は計算効率を高めるという効果を生みます。それから派生することですが、さらにありがたいことは、畳み込みニューラルネットワークの隠れ層は効率よく特徴抽出が可能になるということです。

効率よい特徴抽出ということを調べるために、入力層の異なる場所にある次の「 \times 」のパターンの特徴抽出を考えてみましょう。



ニューラルネットワークのしくみでは、同一の左右のパターンは異なるものとみなされます。ニューラルネットワークは入力層全体を一度に扱うからです。しかし、畳み込みニューラルネットワークでは、次のフィルターを用いことで同一のものとして特徴抽出できます。このフィルターで画像全体が小分けされてスキャンされるからです。



上の図の2つのパターンはこのフィルターで探すことができる。

この効率よい特徴抽出こそが、畳み込みニューラルネットワークの大切な性質なのです。

■ Excelで畳み込み層を計算しよう

入力層の出力と畳み込み層のフィルターとの関係が調べられたので、それを用いて実際に計算を進めてみましょう。それには次の例題のステップを追うことになります。

例題1 フィルターの成分と閾値の仮の値を入力しましょう。なお、これらのパラメーターの値は次のワークシートの形式で入力します。

注 この例題のワークシートは、ダウンロードサイト（→卷頭「Excelサンプルファイルのダウンロードについて」）に掲載されたファイル「5.xlsx」の中の「例題」タブに収められています。

	A	B	C	D	E	F	G	H
11								
12				F1	0.16	0.06	0.18	0.18
13					0.17	0.01	0.17	0.19
14					0.16	0.04	0.08	0.12
15					0.16	0.07	0.16	0.17
16	畳 み 込 み 層	フ ィ ル タ ー	F2		0.14	0.12	0.20	0.06
17					0.05	0.03	0.06	0.19
18					0.02	0.06	0.04	0.12
19					0.10	0.02	0.08	0.05
20			F3		0.12	0.04	0.09	0.05
21					0.08	0.15	0.16	0.10
22					0.15	0.03	0.18	0.01
23					0.03	0.08	0.1	0.14
24			θ		0.09	0.15	0.13	

隠れ層1番目のニューロンの
フィルター(隠れ層2、3番目
も同様)

隠れ層のニューロンの閾値。
左から順に1番目、2番目、3
番目のニューロンの値

解

上の図に示されたフィルターと閾値の領域の1つにRAND関数を入力します。RAND関数は0~1の乱数を発生させる関数です。

E12	fx	=0.2*RAND()						
A	B	C	D	E	F	G	H	I
11								
12				F1	0.09			
13								

パラメーター領域の一角に
RAND関数を入力。ここでは、それを0.2倍している

ここでは、Excelのソルバーが計算しやすいように、初期値に入力する
RAND関数も小さくしています(すなわち、0.2倍しています)。

次に、この関数をすべてのフィルターと閾値の領域にコピーし、値を確定
(昔風にいうと「値複写」)します。こうして、[例題1]に示されたワークシート
が得られます。

例題2 本章§3の例題(→「入力層のニューロンをExcelで表現」)で求め
た入力層のニューロンの出力値と、本節例題1で設定したパラメーターの仮

の値を利用して、畳み込み層の出力表を作成しましょう。

注 この例題のワークシートは、ダウンロードサイト（→卷頭「Excelサンプルファイルのダウンロードについて」）に掲載されたファイル「5.xlsx」の中の「例題」タブに収められています。

解

畳み込み層の作り方にのっとって、隠れ層のニューロン①の出力表を作成しましょう。それには、小分けした区画11（本節の「畳み込みニューラルネットワークは入力層を小分けに調べる」の図で地区11と呼んだところ）について、1番目のフィルターを用いて、式(1)(2)から出力を算出します。

手書き数字1、2の識別(未学習)		番号	1	0	0	0	0	0	0	1.07	2.13	0
倍率 0.01		入力層	0	0	0	0.05	2.45	1.53	0	0	0	0
			0	0	1.59	2.21	0.01	0	0	0	0	0
			0	0	0.32	2.48	0.17	0	0	0	0	0
			0	0.03	2.34	0.83	0	0	0	0	0	0
			0	0.17	2.44	0.74	0	0	0	0	0	0
			0	0	0	0	0	0	0	0	0	0
正解t1,t2			1	0								
		F1	0.16	0.06	0.18	0.18						
			0.17	0.01	0.17	0.19						
			0.16	0.04	0.08	0.12						
			0.16	0.07	0.16	0.17						
畳	フ	F2	0.14	0.12	0.20	0.06						

この図が示すように、セル番地L12に次の関数を埋め込んでいます。

$$=1/(1+\text{EXP}(-\text{SUMPRODUCT}(\E12:\H15,L2:O5)+E24))$$

参照番地の指定に複合参照を利用していることに留意しましょう。

次に、このセル番地L12の関数を、セル範囲L12:Q17の6×6の区画にコピーしましょう。こうして隠れ層のニューロン①に関する出力表が完成します。

手書き数字1、2の識別(未学習)											正解t1,t2											
番号	入力層										正解t1,t2											
	L	M	N	O	P	Q	R	S	T	1	0	0	0	0	0	0	1.07	2.13	0	0		
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
5	0	0	0	0.05	0.245	1.53	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
6	0	0	0	1.59	2.21	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
7	0	0	0.32	2.48	0.17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
8	0	0.03	2.34	0.83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
9	0	0.17	2.44	0.74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
11	正解t1,t2										1	0										
12	F1	0.16	0.06	0.18	0.18	0.48	0.60	0.73	0.80	0.84	0.75	0.55	0.72	0.80	0.83	0.81	0.69	0.65	0.75	0.48	0.48	
13		0.17	0.01	0.17	0.19	0.65	0.79	0.84	0.83	0.75	0.69	0.75	0.83	0.83	0.76	0.69	0.54	0.48	0.48	0.48	0.48	
14		0.16	0.04	0.08	0.12	0.78	0.68	0.72	0.64	0.49	0.48	0.83	0.82	0.80	0.73	0.58	0.48	0.48	0.48	0.48	0.48	
15		0.16	0.07	0.16	0.17	0.78	0.68	0.72	0.64	0.49	0.48	0.83	0.82	0.80	0.73	0.58	0.48	0.48	0.48	0.48	0.48	
16	F2	0.14	0.12	0.20	0.06	0.78	0.68	0.72	0.64	0.49	0.48	0.83	0.82	0.80	0.73	0.58	0.48	0.48	0.48	0.48	0.48	
17		0.05	0.03	0.06	0.19	0.78	0.68	0.72	0.64	0.49	0.48	0.83	0.82	0.80	0.73	0.58	0.48	0.48	0.48	0.48	0.48	

同様のことを実行して、隠れ層のニューロン2、3の出力表を作成します。

L24		fx		=1/(1+EXP(-SUMPRODUCT(\$E\$20:\$H\$23,L2:O5)+\$G\$24))																					
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T				
1	手書き数字1、2の識別(未学習)						倍率 0.01		番号		1														
2											0		0		0		0		0		1.07		2.13		
3											0		0		0		0.33		2.49		0.07		0		
4											0		0		0		0.52		2.42		0.74		0		
5											0		0		0.05		2.45		1.53		0		0		
6											0		0		1.59		2.21		0.01		0		0		
7											0		0		0.32		2.48		0.17		0		0		
8											0		0.03		2.34		0.83		0		0		0		
9											0		0.17		2.44		0.74		0		0		0		
10											0		0		0		0		0		0		0		
11									番号		1														
12	置 み 込 み 層			F1		0.16	0.06	0.18	0.18			F1		0.48	0.60	0.73	0.80	0.84	0.75						
13				0.17		0.01	0.17	0.19			0.55		0.72	0.80	0.83	0.81	0.69								
14				0.16		0.04	0.08	0.12			0.65		0.79	0.84	0.83	0.75	0.65								
15				0.16		0.07	0.16	0.17			0.75		0.83	0.83	0.76	0.69	0.54								
16				F2		0.14	0.12	0.20	0.06			0.83		0.82	0.80	0.73	0.58	0.48							
17				0.05		0.03	0.06	0.19			0.78		0.68	0.72	0.64	0.49	0.48								
18				0.02		0.06	0.04	0.12			F2		0.46	0.51	0.62	0.69	0.70	0.68							
19				0.10		0.02	0.08	0.05			0.48		0.62	0.70	0.69	0.71	0.59								
20				F3		0.12	0.04	0.09	0.05			0.55		0.71	0.69	0.74	0.64	0.59							
21				0.08		0.15	0.16	0.10			0.67		0.68	0.74	0.68	0.62	0.52								
22				0.15		0.03	0.18	0.01			0.70		0.69	0.72	0.64	0.54	0.46								
23				0.03		0.08	0.11	0.14			0.63		0.67	0.61	0.57	0.47	0.46								
24				θ		0.09	0.15	0.13			F3		0.47	0.56	0.62	0.74	0.73	0.72							
25				O1		P1	0.13	0.11	0.18			0.52		0.61	0.74	0.75	0.75	0.63							
26				0.10		0.07	0.13			0.57		0.68	0.78	0.76	0.70	0.58									
27				0.12		0.16	0.03			0.62		0.77	0.74	0.73	0.60	0.51									
28				P2		0.09	0.10	0.04			0.74		0.74	0.63	0.54	0.47	0.47								
29				0.19		0.16	0.19			0.72		0.69	0.67	0.59	0.47	0.47									

これを訓練データ全体にコピーする必要がありますが、その操作は最後に回します。ちなみに、この段階での畳み込み層の出力を見ても意味はありません。フィ

ルターの成分と閾値が仮の値だからです。

§5

畳み込みニューラルネットワークのプーリング層

畳み込み層のニューロンが算出した出力表の各成分は、フィルターを通して見た画素情報が凝縮されています。しかし、実際の画像を扱うときには、まだまだ情報量は多すぎます。そこで、さらに情報の縮約を行いましょう。それがプーリング層の役割です。

■ プーリング層

畳み込み層の出力表は、画像を構成する $9 \times 9 (= 81)$ 画素の情報を $6 \times 6 (= 36)$ 個の情報に縮約した表と考えられます。しかし、まだ大きな情報です。そこで、隠れ層のニューロンの出力表をさらに縮約してみましょう。それを行うのがプーリング層です。

例として、畳み込み層のニューロン①の出力表を見てみましょう。前節で調べたように、この表は 6×6 の成分を持ちます。それを次図のように 2×2 の区画に分割してみます。

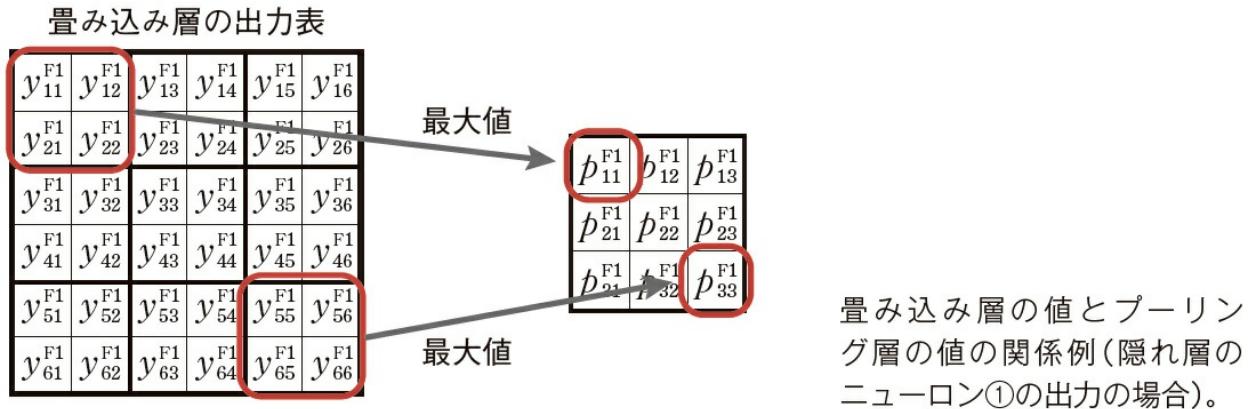
隠れ層のニューロンの出力層

y_{11}^{F1}	y_{12}^{F1}	y_{13}^{F1}	y_{14}^{F1}	y_{15}^{F1}	y_{16}^{F1}
y_{21}^{F1}	y_{22}^{F1}	y_{23}^{F1}	y_{24}^{F1}	y_{25}^{F1}	y_{26}^{F1}
y_{31}^{F1}	y_{32}^{F1}	y_{33}^{F1}	y_{34}^{F1}	y_{35}^{F1}	y_{36}^{F1}
y_{41}^{F1}	y_{42}^{F1}	y_{43}^{F1}	y_{44}^{F1}	y_{45}^{F1}	y_{46}^{F1}
y_{51}^{F1}	y_{52}^{F1}	y_{53}^{F1}	y_{54}^{F1}	y_{55}^{F1}	y_{56}^{F1}
y_{61}^{F1}	y_{62}^{F1}	y_{63}^{F1}	y_{64}^{F1}	y_{65}^{F1}	y_{66}^{F1}

2×2に
区画化
→

y_{11}^{F1}	y_{12}^{F1}	y_{13}^{F1}	y_{14}^{F1}	y_{15}^{F1}	y_{16}^{F1}
y_{21}^{F1}	y_{22}^{F1}	y_{23}^{F1}	y_{24}^{F1}	y_{25}^{F1}	y_{26}^{F1}
y_{31}^{F1}	y_{32}^{F1}	y_{33}^{F1}	y_{34}^{F1}	y_{35}^{F1}	y_{36}^{F1}
y_{41}^{F1}	y_{42}^{F1}	y_{43}^{F1}	y_{44}^{F1}	y_{45}^{F1}	y_{46}^{F1}
y_{51}^{F1}	y_{52}^{F1}	y_{53}^{F1}	y_{54}^{F1}	y_{55}^{F1}	y_{56}^{F1}
y_{61}^{F1}	y_{62}^{F1}	y_{63}^{F1}	y_{64}^{F1}	y_{65}^{F1}	y_{66}^{F1}

そして、各区画の最大値をその区画の代表値として採用します。次の図は、隠れ層のニューロン①の出力表を分割し、分割された左上の区画と右下の区画に着目している図です。その各区画から代表値を取り出し、それを p_{11}^{F1}, p_{33}^{F1} としていることを示しています。



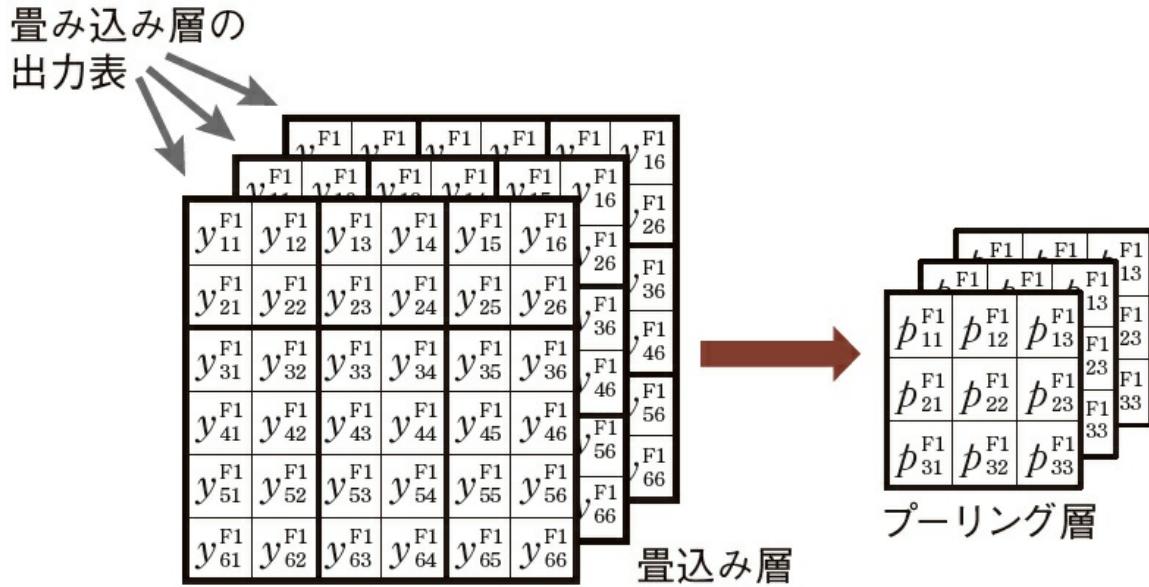
この代表値の選び方として有名なもの一つが区画の最大値を選出する方法です。式で示すと、次のように表現できます(他の区画についても同様です)。

$$p_{11}^{F1} = \text{Max}(y_{11}^{F1}, y_{12}^{F1}, y_{21}^{F1}, y_{22}^{F1}) \dots (1)$$

$$p_{33}^{F1} = \text{Max}(y_{55}^{F1}, y_{56}^{F1}, y_{65}^{F1}, y_{66}^{F1})$$

このようにして、畳み込み層に凝縮された画素情報をさらに縮約する方法を **MAXプーリング**(max pooling)と呼びます。

以上の操作を畳み込み層全体について実施してみましょう。畳み込み層の3枚の出力表は、大きさ3×3(=9)の3枚の表に縮約されることになります。



このように、畳み込み層を縮約した新たな表が作る層をプーリング層と呼びます。プーリング層を構成する表をプーリング表と呼ぶことにします。

畳み込み層の出力表の各成分は、前節で調べたようにフィルターとの類似度を表します。その出力表からMAXプーリングで最大値を選出するということは、フィルターに似たパターンを濃縮することを意味します。畳み込みニューラルネットワークはこのように情報凝縮していくのです。

例1 次の図の左側が畳み込み層の出力値のとき、右が最大プーリングの結果です。

0.48	0.60	0.73	0.80	0.84	0.75
0.55	0.72	0.80	0.83	0.81	0.69
0.65	0.79	0.84	0.83	0.75	0.65
0.75	0.83	0.83	0.76	0.69	0.54
0.83	0.82	0.80	0.73	0.58	0.48
0.78	0.68	0.72	0.64	0.49	0.48

畳み込み層の出力

プーリング

0.72	0.83	0.84
0.83	0.84	0.75
0.83	0.80	0.58

プーリング表

MAX プーリングの例

■ Excelでプーリング層を計算しよう

前節(§4)で得られた畳み込み層の出力表から、プーリング層を作成しましょう。

例題 前節(§4)で得られた畳み込み層から、プーリング層を形作るプーリング表を求めましょう。

注 この例題のワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「5.xlsx」の中の「例題」タブに収められています。

解

MAXプーリングを行うには、ExcelのMAX関数が便利です。先に作成した畳み込み層の出力表に、この関数を 2×2 区画ごとに適用します。

		L30											
12			K	L	M	N	O	P	Q				
13		畳 込 層	F1	0.48	0.60	0.73	0.80	0.84	0.75				
14				0.55	0.72	0.80	0.83	0.81	0.69				
15				0.65	0.79	0.84	0.83	0.75	0.65				
16				0.75	0.83	0.83	0.76	0.69	0.54				
17				0.83	0.82	0.80	0.73	0.58	0.48				
18			F2	0.78	0.68	0.72	0.64	0.49	0.48				
19				0.46	0.51	0.62	0.69	0.70	0.68				
20				0.48	0.62	0.70	0.69	0.71	0.59				
21													
22													
23													
24													
25													
26													
27													
28													
29													
30			P1	0.72									
31													
32													

範囲 L12:M13 の最大値を
セル L30 に MAX 関数で
求める。

MAXプーリングのための関数入力が面倒なのは、いま入力したセルの関数をそのままコピーして使えないことです。仕方がないので、隣のセルには、手作業でまた新たにMAX関数を入力します。

M30				fx		=MAX(N12:O13)			
I	J	K	L	M	N	O	P	Q	R
12	疊 込 層	F1	0.48	0.60	0.73	0.80	0.84	0.75	
13			0.55	0.72	0.80	0.83	0.81	0.69	
14			0.65	0.79	0.84	0.83	0.75	0.65	
15			0.75	0.83	0.83	0.76	0.69	0.54	
16			0.83	0.82	0.80	0.73	0.58	0.48	
17			0.78	0.68	0.72	0.64	0.49	0.48	
18		F2	0.46	0.51	0.62	0.69	0.70	0.68	
19			0.48	0.62	0.70	0.69	0.71	0.59	
29			0.72	0.69	0.67	0.59	0.47	0.47	
30		P1	0.72	0.83					
31									
32									

範囲 N12:O13 の最大値
をセル M30 に MAX 関
数で求める

先に入力したセルの関数をコピー
してはならない。隣のセルには手
作業でまた新たに MAX 関数を入
力。

同様な操作をプーリング層全体に行います。多少面倒な作業でしたが、
こうしてプーリング層が完成します。

I	J	K	L	M	N	O	P	Q	R
12	疊 込 層	F1	0.48	0.60	0.73	0.80	0.84	0.75	
13			0.55	0.72	0.80	0.83	0.81	0.69	
14			0.65	0.79	0.84	0.83	0.75	0.65	
15			0.75	0.83	0.83	0.76	0.69	0.54	
16			0.83	0.82	0.80	0.73	0.58	0.48	
17			0.78	0.68	0.72	0.64	0.49	0.48	
18		F2	0.46	0.51	0.62	0.69	0.70	0.68	
19			0.48	0.62	0.70	0.69	0.71	0.59	
20			0.55	0.71	0.69	0.74	0.64	0.59	
21			0.67	0.68	0.74	0.68	0.62	0.52	
22			0.70	0.69	0.72	0.64	0.54	0.46	
23			0.63	0.67	0.61	0.57	0.47	0.46	
24	疊 込 層	F3	0.47	0.56	0.62	0.74	0.73	0.72	
25			0.52	0.61	0.74	0.75	0.75	0.63	
26			0.57	0.68	0.78	0.76	0.70	0.58	
27			0.62	0.77	0.74	0.73	0.60	0.51	
28			0.74	0.74	0.74	0.63	0.54	0.47	
29			0.72	0.69	0.67	0.59	0.47	0.47	
30		P1	0.72	0.83	0.84				
31			0.83	0.84	0.75				
32			0.83	0.80	0.58				
33	プ リ ン グ 層	P2	0.62	0.70	0.71				
34			0.71	0.74	0.64				
35			0.70	0.72	0.54				
36		P3	0.61	0.75	0.75				
37			0.77	0.78	0.70				
38			0.74	0.74	0.54				

手作業でプーリング層
を完成

この結果を訓練データに含まれる画像すべてにコピーする必要がありますが、これまで通り、その操作は最後に回します。また、この段階での畳み込み層の出力を見ても、何も意味ある情報は得られません。パラメーターが仮の値だからです。

Memo MAXプーリングの公式化

式(1)(→本節の「プーリング層」)の作り方さえ理解していれば、Excelのワークシートでプーリング層を作成するは容易です。しかし、他の文献を読む際の確認として、あえて公式としてまとめましょう。具体的に、隠れ層のニューロン①に関する場合を調べましょう。

y_{11}^{F1}	y_{12}^{F1}	y_{13}^{F1}	y_{14}^{F1}	y_{15}^{F1}	y_{16}^{F1}
y_{21}^{F1}	y_{22}^{F1}	y_{23}^{F1}	y_{24}^{F1}	y_{25}^{F1}	y_{26}^{F1}
y_{31}^{F1}	y_{32}^{F1}	y_{33}^{F1}	y_{34}^{F1}	y_{35}^{F1}	y_{36}^{F1}
y_{41}^{F1}	y_{42}^{F1}	y_{43}^{F1}	y_{44}^{F1}	y_{45}^{F1}	y_{46}^{F1}
y_{51}^{F1}	y_{52}^{F1}	y_{53}^{F1}	y_{54}^{F1}	y_{55}^{F1}	y_{56}^{F1}
y_{61}^{F1}	y_{62}^{F1}	y_{63}^{F1}	y_{64}^{F1}	y_{65}^{F1}	y_{66}^{F1}



p_{11}^{F1}	p_{12}^{F1}	p_{13}^{F1}
p_{21}^{F1}	p_{22}^{F1}	p_{23}^{F1}
p_{31}^{F1}	p_{32}^{F1}	p_{33}^{F1}

プーリング表 1

畳み込み層の出力表 1

この図から、プーリング表の i 行 j 列の値 p_{ij}^{F1} は、畳み込み層の出力表の値を利用して、次のように求められます (i, j は 1 から 3 までの整数)。

$$p_{ij}^{F1} = \text{Max}(y_{2i-1, 2j-1}^{F1}, y_{2i-1, 2j}^{F1}, y_{2i, 2j-1}^{F1}, y_{2i, 2j}^{F1})$$

これを一般化すると、次のように表すことができます。ここで、 k は隠れ層のニューロンの番号 (すなわち畳み込み層の出力シートの番号) です。

$$p_{ij}^{Fk} = \text{Max}(y_{2i-1, 2j-1}^{Fk}, y_{2i-1, 2j}^{Fk}, y_{2i, 2j-1}^{Fk}, y_{2i, 2j}^{Fk})$$

かなり面倒な式です。繰り返しますが、Excelで畳み込みニューラルネットワークを実装する際には、このような式が不要です。

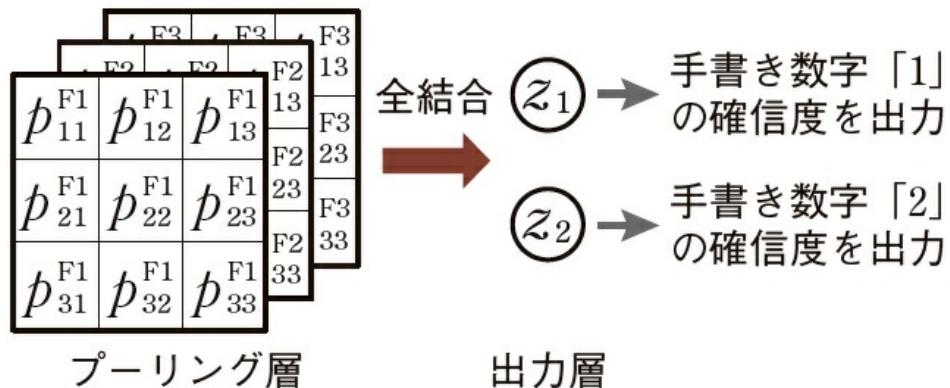
§ 6

畳み込みニューラルネットワークの出力層

本節では畳み込みニューラルネットワークの出力層について調べましょう。この層の機能はニューラルネットワークの場合と同じです。下の層(プーリング層)と全結合し、プーリング層で濃縮された情報をまとめ、ネットワーク全体の判断を出力します。

■ 出力層はニューラルネットワークと同じ機能

前の節(§5)で、プーリング層は入力情報を濃縮することを調べました。その濃縮情報からネットワーク全体の判断を出力するのが出力層の役割です。今の[テーマⅡ]は手書き数字「1」「2」の区別なので、出力層は2つのニューロンから成り立ちます。



図に示したように、出力層の上から順にニューロン名を z_1 、 z_2 とし、それらの出力も同じ z_1 、 z_2 と表すことにします。

出力層1番目のニューロン z_1 は、手書き数字「1」が入力されたときに1が、そうでないときには0が算出されることを期待される変数です。2番目のニューロン z_2

は、手書き数字「2」が入力されたときに1が、そうでないときには0が算出されることを期待される変数です。

■ 出力層とプーリング層は全結合で結ばれる

ニューラルネットワークにおいては、出力層のニューロンと隠れ層の出力とは全結合しました。同様に畳み込みニューラルネットワークでも、出力層のニューロンとプーリング層の各成分とは全結合します。プーリング層の各表(すなわちプーリング表)の値と出力層のニューロンとはすべて矢で結ばれるのです。こうすることで、ニューラルネットワークのときと同じくして特徴抽出の結論を出力することが可能になります。

例1 出力層1番目のニューロンに関して、入力の線形和 a^{01} とこれから得られる出力 z_1 を具体的に書き下してみましょう。

入力の線形和とその出力の定義から、次のように書き下せます。

$$a^{01} = w_{1-11}^{01} p_{11}^{F1} + w_{1-12}^{01} p_{12}^{F1} + \cdots + w_{2-11}^{01} p_{11}^{F2} + w_{2-12}^{01} a_{12}^{P2} + \cdots + w_{3-11}^{01} a_{11}^{P3} + w_{3-12}^{01} a_{12}^{P3} + \cdots - \theta^{01} \quad \cdots (1)$$

$$z_1 = \sigma(a^{01}) \quad (\sigma \text{ をシグモイド関数}) \quad \cdots (2)$$

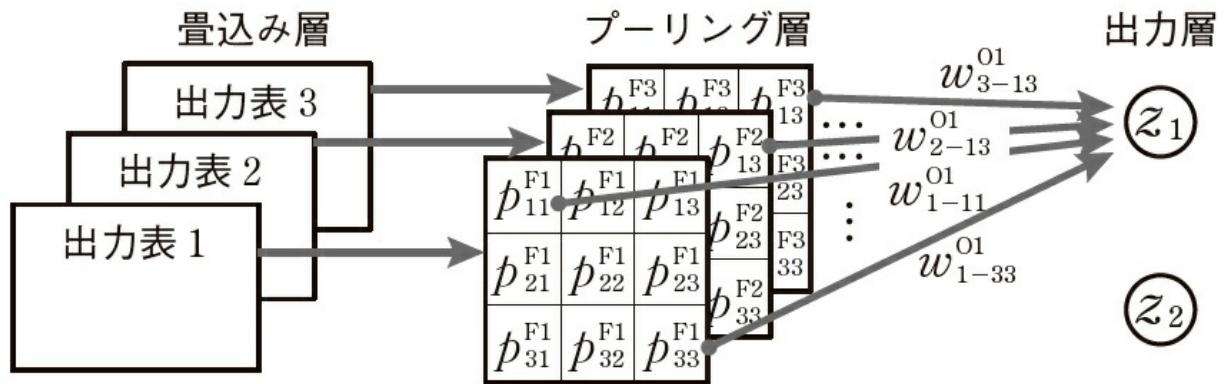
ここで、係数 w_{k-i}^{01} は出力層1番目のニューロンが k 枚目のプーリング表の i 行 j 列目にある値に課す重みです。また、 θ^{01} は出力層1番目のニューロンが持つ閾値です。

例2 w_{2-13}^{01} は出力層1番目のニューロンが2枚目のプーリング表の1行3列目にある値に課す重みです(次図)。

重み w_{2-13}^{O1}
 2 はプーリング層
 の表の番号
 13 はプーリング層の
 行番号と列番号

O1 は出力層の 1 番目のニューロン

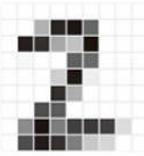
式(1)は複雑なので、変数間の関係を図で例示しましょう。



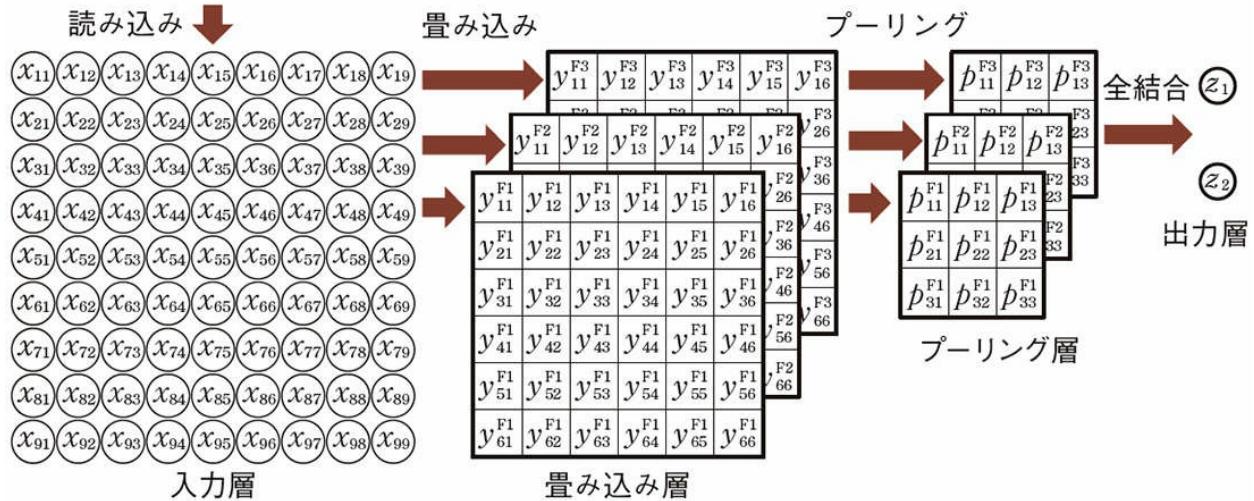
出力層のニューロンの出力 z_1 を書き下すための変数とパラメーターの関係。

■ 入力層から出力層までまとめよう

こうして、いま調べている[テーマⅡ]に対する畳み込みニューラルネットワークの概形が完成しました。これまでバラバラに描いてきた図を統合してみましょう。それが次の図です。本章の[テーマⅡ]の畳み込みニューラルネットワークの形が完成したのです。



手書き数字 (9×9 画素)



■ Excelで出力層を計算しよう

前節(§5)で得られたプーリング層から、出力層の計算をしましょう。こうして、画像1枚について、その表す数字が「1」か「2」かを判断するワークシートが完成します。

そのために、まず、出力層の各ニューロンがプーリング表の各成分に与える重みと、そのニューロンの閾値の仮の値を設定します。その上で式(1)(2)から、出力層のニューロンの出力を計算します。この出力が畳み込みニューラルネットワークの判断の結論になります。

例題1 出力層の重みと閾値の仮の値を入力しましょう。なお、これらのパラメーターの値は次のワークシートの形式で入力します。

注 この例題の結果のワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードにつ

いて」に掲載されたファイル「5.xlsx」の中の「例題」タブに収められています。

	A	B	C	D	E	F	G	H
25		O1	P1		0.13	0.11	0.18	
26					0.10	0.07	0.13	
27					0.12	0.16	0.03	
28			P2		0.09	0.10	0.04	
29					0.19	0.16	0.19	
30					0.07	0.04	0.06	
31			P3		0.04	0.18	0.17	
32					0.15	0.11	0.03	
33	出				0.07	0.17	0.08	
34	力	O2	P1		0.11	0.10	0.19	
35	層				0.10	0.10	0.12	
36					0.08	0.14	0.06	
37			P2		0.15	0.08	0.11	
38					0.06	0.18	0.00	
39					0.16	0.14	0.08	
40			P3		0.11	0.05	0.09	
41					0.00	0.05	0.04	
42					0.15	0.06	0.07	
43			θ		0.08	0.12		

出力層1番目のニューロンの重みの仮の値（出力層2番目も同様）

出力層のニューロンの閾値の仮の値。左から順に1番目、2番目の値

解

上の図に示されたフィルターと閾値の領域の1つにRAND関数を入力します。RAND関数は0～1の乱数を発生させる関数です（→2章§1）。

	A	B	C	D	F	F	G	H	I	J	K	L	M	N	
25		O1	P1	0.03								0.52			
26												0.57			
27												0.62			
28					P2							0.74			
29												0.72	0.69	0.67	
30												P1	0.72	0.83	0.84
31													0.83	0.84	0.75
32													0.83	0.80	0.58
33			出									P2	0.62	0.70	0.71
34			力										0.71	0.74	0.64
35			層										0.70	0.72	0.54
36					O2	P1						P3	0.61	0.75	0.75
37													0.77	0.78	0.70
38													0.74	0.74	0.54
39												出力層	z1	z2	
40															
41												誤差	Q		
42															
43												θ			

本章§4と同様、Excelのソルバーが計算しやすいように、初期値に入力するRAND関数も小さくしています(すなわち、0.2倍しています)。

次に、この関数をすべてのフィルターと閾値の領域にコピーし、値を確定(値複写)します。こうして、[例題1]の題意にある仮の値のワークシートが得られます。

例題2 前節(§5)で求めたプーリング層から、訓練データの手書き数字の1番目の画像について、出力層の2つのニューロンの出力を算出しましょう。

注 この例題のワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「5.xlsx」の中の「例題」タブに収められています。

解

前節で得られたプーリング表を利用して、出力層1番目のニューロンについて

て、その出力値 z_1 を求めます。計算式は式(1)(2)に従います。なお、重みや閾値については、[例題1]で作成した仮の値を利用します。

同様の操作を出力層2番目のニューロンについて行い、その出力値 z_2 を求めます。

注 L40をM40にそのままコピーしてはいけません。

25		O1	P1	0.13	0.11	0.18				0.52	0.61	0.74	0.75	0.75	0.63				
26				0.10	0.07	0.13				0.57	0.68	0.78	0.76	0.70	0.58				
27				0.12	0.16	0.03				0.62	0.77	0.74	0.73	0.60	0.51				
28		P2		0.09	0.10	0.04				0.74	0.74	0.74	0.63	0.54	0.47				
29				0.19	0.16	0.19				0.72	0.69	0.67	0.59	0.47	0.47				
30				0.07	0.04	0.06				P1	0.72	0.83	0.84						
31		P3		0.04	0.18	0.17					0.83	0.84	0.75						
32				0.15	0.11	0.03					0.83	0.80	0.58						
33	出			0.07	0.17	0.08				P2	0.62	0.70	0.71						
34	力	O2	P1	0.11	0.10	0.19					0.71	0.74	0.64						
35	層			0.10	0.10	0.12					0.70	0.72	0.54						
36				0.08	0.14	0.06				P3	0.61	0.75	0.75						
37		P2		0.15	0.08	0.11					0.77	0.78	0.70						
38				0.06	0.18	0.00					0.74	0.74	0.54						
39				0.16	0.14	0.08				出力層	z1	z2							
40		P3		0.11	0.05	0.09					0.89	0.85							
41				0.00	0.05	0.04				誤差	Q								
42				0.15	0.06	0.07													
43		θ		0.08	0.12														

プーリング層の各値に、出力層のニューロンは重みを付けている

以上で、手書き数字の1番目の画像について、仮の重みと閾値から出力層のニューロンの出力が得られました。

Memo | プーリング法いろいろ

本節の解説では、プーリングの方法として最大プーリングを利用しました。対象の領域の最大値を代表値として採用する情報の縮約法です。プーリング方法にはこれ以外にもいろいろあります。有名なものを次の表に記載します。

名称	解説
最大プーリング	対象の領域の最大値を採用する縮約法。
平均プーリング	対象の領域の平均値を採用する縮約法。
L2 プーリング	例えば 4 つの出力 y_1 、 y_2 、 y_3 、 y_4 に対して $\sqrt{y_1^2 + y_2^2 + y_3^2 + y_4^2}$ を採用する縮約法。

§ 7

正解と出力の誤差

1枚の数字画像を対象にして、畳み込みニューラルネットワークが「1」か「2」かの判断を下すワークシートが出来上がりました。本節では、その出力の評価を数値化します。その考え方は、回帰分析(2章§4)やニューラルネットワーク(4章§4)のときと同じです。

■ 正解の表現法の確認

先に調べたように(→本章§3)、手書き数字画像の正解は次図のように、 t_1 、 t_2 の2つの変数の組として表現しています。

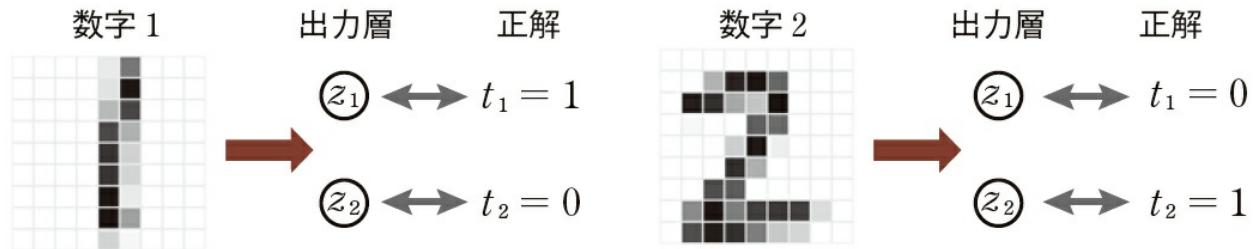
J	K	L	M	N	O	P	Q	R	S	T
番号	1									
入力層		0	0	0	0	0	0	1.07	2.13	0
		0	0	0	0	0	0.33	2.49	0.07	0
		0	0	0	0	0.52	2.42	0.74	0	0
		0	0	0	0.05	2.45	1.53	0	0	0
		0	0	0	1.59	2.21	0.01	0	0	0
		0	0	0.32	2.48	0.17	0	0	0	0
		0	0.03	2.34	0.83	0	0	0	0	0
		0	0.17	2.44	0.74	0	0	0	0	0
正解(t_1, t_2)		1	0							

正解の欄

そこで調べたように、この変数 t_1 、 t_2 は次の表のよう定義されました。

	意味	画像が「1」	画像が「2」
t_1	「1」の正解変数	1	0
t_2	「2」の正解変数	0	1

ニューラルネットワーク(4章)のときにも確認しましたが、このような正解の表現を用いると、出力層の2つのニューロンとピッタリと対応がとれます。次の図は、2つの画像例について各変数値を示しています。



■ 算出値と正解との誤差の表現

出力層1番目のニューロン z_1 は、手書き数字が「1」と思われるとき1に、そうでないときには0に近い値が算出されることが期待されます。2番目のニューロン z_2 は、手書き数字が「2」と思われるとき1に、そうでないときには0に近い値が算出されることが期待されます。すなわち、出力層のニューロンの出力 z_1, z_2 には、畳み込みニューラルネットワークの算出した手書き数字「1」、「2」の確信度がセットされていると考えられます。

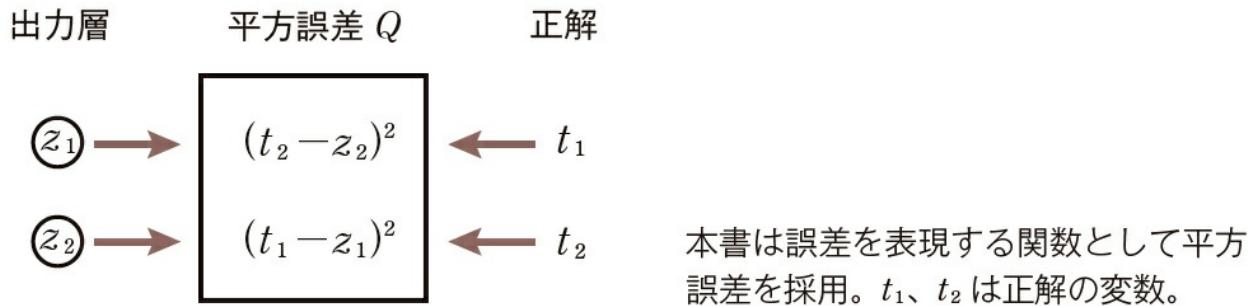
期待される値		
	画像が「1」のとき	画像が「2」のとき
z_1	1 に近い値	0 に近い値
z_2	0 に近い値	1 に近い値

そこで、1つの手書き数字画像について、ネットワークが算出した計算値と正解との誤差 Q は次のように表せます。これを平方誤差と呼ぶことは、すでに2章§4や

4章§4でも調べました。

$$Q = (t_1 - z_1)^2 + (t_2 - z_2)^2 \dots (1)$$

注 多くの文献には、この式(1)に係数1/2が付いています。それは微分計算を簡潔にするためです。本書では微分計算をしないので略し簡潔にします。



■ Excelで平方誤差を計算しよう

前節(§6)のワークシートを用いて、実際に平方誤差(1)を算出してみましょう。

例題 前節(§6)で得られた出力層のニューロンの出力値から、訓練データの1番目の手書き数字画像について、平方誤差を計算しましょう。

注 この例題のワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「5.xlsx」の中の「例題」タブに収められています。

解

次図のワークシートに示すように、平方誤差 Q の算出にはExcelのSUMXMY2関数が威力を発揮します。

Excel screenshot showing the calculation of handwritten digit recognition. The formula `=SUMXMY2(L11:M11,L40:M40)` is highlighted in the formula bar.

正解のペア (Correct pair) is highlighted in red in cell L11.

平方誤差の算出 (→式 (1)) (Calculation of squared error (→Equation (1))) is highlighted in red in cell Q41.

出力層の出力値のペア (Pair of output values of the output layer) is highlighted in red in cell P41.

Handwritten digit recognition table:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
1	手書き数字1、2の識別(未学習)										番号	1						
2							倍率	0.01				0	0	0	0	0	0	
3												0	0	0	0	0.33		
4												0	0	0	0.05	2.45	1.53	
5												0	0	0	1.59	2.21	0.01	
6												0	0	0.32	2.48	0.17	0	
7												0	0.03	2.34	0.83	0	0	
8												0	0.17	2.44	0.74	0	0	
9												0	0	0	0	0	0	
10																		
11																		
12																		
13																		
14																		
29																		
30																		
31																		
32																		
33																		
34																		
35																		
36																		
37																		
38																		
39																		
40																		
41																		
42																		
43																		

以上で、手書き数字の1番目の画像について、仮の重みと閾値から得られた出力層のニューロンの出力値の評価が数値化されました。

§8

畳み込みニューラルネットワークの目的関数

前節では、1つの数字画像について、その出力値の平方誤差を算出しました。畳み込みニューラルネットワークを最適化するには、訓練データ全体について平方誤差を求め、加え合わせなければなりません。

■ 目的関数 Q_T を求める

考えているネットワークにおいては、出力層の出力が z_1, z_2 の2つ、それに対応して訓練データには正解 t_1, t_2 とすると、前節§7で調べたように、正解と出力値との誤差 Q は次の式で表せることを調べました。

$$Q = (t_1 - z_1)^2 + (t_2 - z_2)^2 \dots (1)$$

ところで、この議論は手書き数字画像1枚を対象にした議論です。訓練データ全体で考えると、ネットワーク全体について、これを加え合わせなければなりません。すなわち、 k 番目の手書き数字画像についての誤差 Q_k を誤差の式(1)から求め、それらの総和 Q_T を求めます。

$$Q_T = Q_1 + Q_2 + \dots + Q_{190} \dots (2)$$

注 この190は今考えている[テーマⅡ]の題意にある訓練データに含まれる画像の枚数です(→本章§2)。



訓練データ全体について考えるとき、この誤差の総和 Q_T が、ネットワークが算出した理論値と正解との誤差になります。この Q_T を目的関数と呼びます。この論理は2章の回帰分析や4章のニューラルネットワークのときと同様です。

■ Excelで目的関数を計算しよう

訓練データ全体についての誤差の総和である目的関数の値を算出します。

例題 前節(§7)までに作成したワークシートを利用して、目的関数(2)の値を算出しましょう。

注 この例題のワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「5.xlsx」の中の「例題」タブに収められています。

解

1番目の手書き数字画像とその正解に関して、これまで作成してきたワークシートを以下にまとめましょう。

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	手書き数字1、2の識別																		
2												番号	1						
3													0	0	0	0	0	1.07	2.13
4													0	0	0	0	0.33	2.49	0.07
5													0	0	0	0.05	0.52	2.42	0.74
6													0	0	0	0.05	2.45	1.53	0
7													0	0	0	1.59	2.21	0.01	0
8													0	0	0.32	2.48	0.17	0	0
9													0	0.03	2.34	0.83	0	0	0
10													0	0.17	2.44	0.74	0	0	0
11													0	0	0	0	0	0	0
12													正解t1,t2	1	0				
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			
21																			
22																			
23																			
24																			
25																			
26																			
27																			
28																			
29																			
30																			
31																			
32																			
33																			
34																			
35																			
36																			
37																			
38																			
39																			
40																			
41																			
42																			
43																			

仮の値

これまで作成したワークシート

この1番目の画像に関するワークシートを、訓練データ全部にコピーします。これで、全訓練データについてのワークシートが完成です。

J	K	L	M	N	O	P	Q	R	S	T	BMW	BMX	BMY	BMZ	BNA	BNB	BNC	BND	BNF	
入力層	1	番号	1	0	0	0	0	0	1.07	2.13	0	0	0	0	0.74	0.88	0	0	0	
	2			0	0	0	0	0.33	2.49	0.07	0	0	0	0	1.89	0.57	0	0	0	
	3			0	0	0	0	0.52	2.42	0.74	0	0	0	0	2.28	0.38	0	0	0	
	4			0	0	0	0.005	2.45	1.53	0	0	0	0	0	0.252	0.11	0	0	0	
	5			0	0	0	0.159	2.21	0.01	0	0	0	0	0	0.002	2.28	0	0	0	
	6			0	0	0.32	2.48	0.17	0	0	0	0	0	0	0.4	2.02	0	0	0	
	7			0	0.03	2.34	0.83	0	0	0	0	0	0	0	0.52	1.59	0	0	0	
	8			0	0.17	2.44	0.74	0	0	0	0	0	0	0	0.06	2.25	0	0	0	
	9			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	10			0	0	0	0	0	0	0	0	0	0	0	0	0.001	0	0	0	
正解t1,2	11		1	0							1	0								
	12		F1	0.48	0.60	0.73	0.80	0.84	0.75						0.48	0.75	0.78	0.63	0.77	0.56
	13			0.55	0.72	0.80	0.83	0.81	0.69						0.48	0.80	0.80	0.62	0.81	0.52
	14			0.65	0.79	0.84	0.83	0.75	0.65						0.50	0.82	0.80	0.61	0.81	0.50
	15			0.75	0.83	0.83	0.76	0.69	0.54						0.51	0.81	0.78	0.61	0.79	0.48
	16			0.83	0.82	0.80	0.73	0.58	0.48						0.52	0.80	0.77	0.61	0.78	0.48
	17			0.78	0.68	0.72	0.64	0.49	0.48						0.52	0.73	0.68	0.58	0.71	0.48
	18		P2	0.46	0.51	0.62	0.69	0.70	0.68						0.46	0.66	0.65	0.60	0.62	0.50
	19			0.48	0.62	0.70	0.69	0.71	0.59						0.46	0.69	0.68	0.62	0.64	0.49
	20			0.55	0.71	0.69	0.74	0.64	0.59						0.47	0.71	0.68	0.62	0.64	0.48
置込層	21			0.67	0.68	0.74	0.68	0.62	0.52						0.48	0.69	0.67	0.61	0.63	0.47
	22			0.70	0.69	0.72	0.64	0.54	0.46						0.50	0.67	0.67	0.59	0.63	0.46
	23			0.63	0.67	0.61	0.57	0.47	0.46						0.50	0.66	0.62	0.59	0.56	0.46
	24		P3	0.47	0.56	0.62	0.74	0.73	0.72						0.47	0.62	0.74	0.66	0.66	0.52
	25			0.52	0.61	0.74	0.75	0.75	0.63						0.47	0.63	0.76	0.66	0.69	0.50
	26			0.57	0.68	0.78	0.76	0.70	0.58						0.48	0.64	0.76	0.65	0.68	0.48
	27			0.62	0.77	0.74	0.73	0.60	0.51						0.49	0.64	0.74	0.64	0.67	0.47
	28			0.74	0.74	0.74	0.63	0.54	0.47						0.48	0.66	0.73	0.64	0.65	0.47
	29			0.72	0.69	0.67	0.59	0.47	0.47						0.49	0.57	0.69	0.58	0.64	0.47
	30		P1	0.72	0.83	0.84									0.80	0.80	0.81			
ブーリング層	31			0.83	0.84	0.75									0.82	0.80	0.81			
	32			0.83	0.80	0.58									0.80	0.77	0.78			
	33		P2	0.62	0.70	0.71									0.69	0.68	0.64			
	34			0.71	0.74	0.64									0.71	0.68	0.64			
	35			0.70	0.72	0.54									0.67	0.67	0.63			
P3	36			0.61	0.75	0.75									0.63	0.76	0.69			
	37			0.77	0.78	0.70									0.64	0.76	0.68			
	38			0.74	0.74	0.54									0.66	0.73	0.65			
	39		出力層	z1	z2										z1	z2				
誤差	40			0.89	0.85										0.89	0.85				
	41			Q	0.74										Q	0.74				

全画像にわたってコピー

こうして、訓練データ全体について、1枚ずつの誤差 Q が算出されます。これら誤差 Q を足し合わせましょう。これが目的関数 Q_T の値となります。以下では、それをセルF45に格納しました。これで 本章の目的である[テーマⅡ]について、それを実現する畳み込みニューラルネットワークのためのワークシートが完成です。

全画像について加え合わせた誤差の総和が目的関数の値。それをセルF45に格納。これが目的関数の値

目的とするワークシートは完成しましたが、しかし、まだ最も大切な作業が残っています。パラメーター（すなわち、フィルターの各成分、重み、閾値）の決定です。これまでのすべての計算は、仮のパラメーターの上でなされました。ネットワークの出力値の意味を議論するには真のパラメーターの上でなされなければならないのです。

■ 計算でモデルの有効性を確認

これからが畳み込みニューラルネットワークの作業のクライマックスです。先にも触れていますが、モデルがデータ解析に役立つか否かは、実際にこのモデルを用いて計算し、与えられたデータを上手に説明できるか否かにかかっています。これから行う最適化の作業の成否が、畳み込みニューラルネットワークというデータ解析モデルの命運を担うわけです。

Memo | 論理は回帰分析と同じ

以上、見てきたように、畳み込みニューラルネットワークの作業は複雑に見えますが、基本は2章で調べた回帰分析と同じです。誤差の総和を求め、そ

れを最小にするようにパラメーターを決定するのです。

§ 9

畳み込みニューラルネットワークの最適化

ワークシートが完成しました。本節では、そのワークシートを用いて畳み込みニューラルネットワークを決定しましょう。数学的にその決定法を「最適化」と呼びますが、そのしくみは回帰分析やニューラルネットワークのときと同じです。

■ 最適化

前節までに、目的の〔テーマⅡ〕のためのExcelワークシートを完成しました。しかし、かんじんのネットワークを決定するパラメーターの値が定まっていません。フィルターの成分や、重み、閾値に仮の値を用いていたのです。ここで、Excelの用意した最適化ツール(ソルバー)を利用して、決定しましょう。ソルバーを用いて誤差の総和、すなわち目的関数 Q_T を最小化する値を求めるのです。

■ Excelソルバーで最適化

それでは、目的関数をソルバーで最小化してみましょう。

例題 前節(§8)に完成したワークシートを用いて、畳み込みニューラルネットワークを最適化しましょう。

注 この例題のワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「5.xlsx」の中の「例題」タブに収められています。

解

次のワークシートに示すように、目的関数 Q_T の入力されたセルを「目的セルの設定」に、畳み込みニューラルネットワークのパラメーター（フィルターの成分や、重み、閾値）のセル範囲を「変数セルの変更」欄に設定します。この準備のもとで、「解決」ボタンをクリックすれば、最適化がなされます。

注 「制約のない変数を非負数にする」に✓を入れます。これを外す場合については、後の§12を参考しましょう。

Excel シートのスクリーンショットです。F45 セルに =SUM(L41:BNE41) と入力されています。

ソルバーのパラメーターアクションが実行されています。

目的関数のセルを設定: \$F\$45 (最小値)

変数セルの変更: \$E\$12:\$H\$23,\$E\$24:\$G\$42,\$E\$43:\$F\$43

制約条件の対象: (U)

フィルター、重み、閾値のセルを設定

この✓を入れる

このメッセージが大切

パソコンによっては 20 分以上を要する

ソルバーの結果

ソルバーによって解が見つかりました。すべての制約条件と最適化条件を満たしています。

ソルバーの解の保持 (選択済)

計算前の値に戻す

ソルバー パラメーターのダイアログに戻る

レポート

解答 策定 条件

アウトライン レポート

OK キャンセル シナリオの保存...

ソルバーによって解が見つかりました。すべての制約条件と最適化条件を満たしています。

GRG エンジンが使用されるのは、ソルバーで 1 つ以上のローカル最適解が見つかった場合です。シンプレックス、LP が使用されるのは、ソルバーでグローバル最適解が見つかった場合です。

ソルバーの計算結果を示しましょう。これが目標とするパラメーター（すなわち、フィルターの成分、重み、閾値）の値です。

ソルバーの計算結果

				F45							
				fx				=SUM(L41:BNE41)			
A	B	C	D	E	F	G	H	I	J	K	L
12				F1	0.00	0.00	0.00	0.00			
13					0.00	0.00	0.00	1.82			
14					0.00	0.00	0.00	0.52			
15					0.00	0.00	0.26	2.42			
16	重 み 込 み 層	F2			0.00	0.00	0.00	0.87			
17					0.00	0.00	0.00	0.86			
18					0.00	0.00	0.00	0.75			
19					0.00	0.00	0.00	0.00			
20		F3			0.00	0.10	0.00	0.04			
21					0.00	2.61	0.95	0.01			
22					1.32	0.08	0.00	0.00			
23					0.00	0.11	0.0	0.00			
24		θ			1.52	3.80	2.64				
25	出 力 層	O1	P1		0.00	0.00	0.00				
26					0.00	0.00	0.00				
27					0.00	0.00	0.00				
28			P2		0.00	0.00	0.00				
29					4.19	0.00	0.00				
30					0.00	0.00	0.00				
31			P3		0.00	0.00	0.00				
32					0.00	0.00	0.00				
33					0.00	0.00	0.00				
34		O2	P1		0.00	0.01	0.27				
35					0.01	0.02	0.87				
36					0.00	2.22	0.49				
37			P2		0.00	0.00	0.19				
38					0.00	0.10	0.00				
39					0.00	0.91	0.51				
40			P3		2.46	0.00	0.01				
41					0.00	0.00	0.00				
42					2.44	0.00	0.69				
43		θ			2.31	5.35					
44				Q _T	37.25						

これが本章の目的の
パラメーターの値

目的関数 Q_T の値

注 このワークシートは、ダウンロードサイト（→卷頭「Excelサンプルファイルのダウンロードについて」）に掲載されたファイル「5.xlsx」の中の「例題_学習済」タブに収められています。

目的関数 Q_T の値 37.25 の大小の議論は難しいのですが、こうして得られたパラメーターを利用すると、訓練データの 99.5% を正しく判別できることが確かめられます。

注 畳み込みニューラルネットワークが算出した値の正誤の判定法については§11を参照しましょう。

§10

最適化されたパラメーターを解釈

前節(§9)で、本章のテーマである[テーマⅡ]に対する畳み込みニューラルネットワークが完成しました。結果として得られたパラメーターの意味を調べてみましょう。

■ フィルターを見てみよう

畳み込み層で用いたフィルターの中身を見てみましょう。各フィルターの中で、大きい方の2つに網をかけてみました。

フィルター 1

0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.82
0.00	0.00	0.00	0.52
0.00	0.00	0.26	2.42

フィルター 2

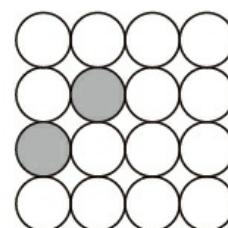
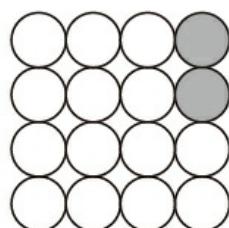
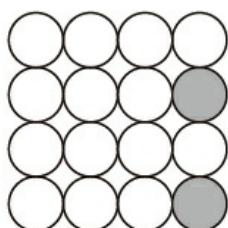
0.00	0.00	0.00	0.87
0.00	0.00	0.00	0.86
0.00	0.00	0.00	0.75
0.00	0.00	0.00	0.00

フィルター 3

0.00	0.10	0.00	0.04
0.00	2.61	0.95	0.01
1.32	0.08	0.00	0.00
0.00	0.11	0.00	0.00

この表から、3つのフィルターの特徴がわかります。すなわち、訓練データの中の手書き数字画像において、隠れ層のニューロンはこの3つのパターンを画像の特徴と判定したのです。すなわち、次の3つのパターンを特徴抽出したのです。

特徴パターン 1 特徴パターン 2 特徴パターン 3



■ 出力層の重みを見てみよう

次に、出力層のニューロンがプーリング層の表(すなわちプーリング表)の各成分に課した重みを見てみましょう。次の図では、1以上の重みを課したプーリング表の値に網をかけています。出力層のニューロン1と2とでは、重みのかけ方が大きく異なっています。

■出力層のニューロン1

フィルター1の
プーリング表への重み

0.00	0.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00

フィルター2の
プーリング表への重み

0.00	0.00	0.00
4.19	0.00	0.00
0.00	0.00	0.00

フィルター3の
プーリング表への重み

0.00	0.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00

■出力層のニューロン2

フィルター1の
プーリング表への重み

0.00	0.01	0.27
0.01	0.02	0.87
0.00	2.22	0.49

フィルター2の
プーリング表への重み

0.00	0.00	0.19
0.00	0.10	0.00
0.00	0.91	0.51

フィルター3の
プーリング表への重み

2.46	0.00	0.01
0.00	0.00	0.00
2.44	0.00	0.69

この図の網のかかった部分だけに着目して、プーリング層から出力層に矢を描いてみましょう。この図から、手書き数字1を判定する役割のニューロン z_1 はフィルター2に対応するプーリング表と強く結びついていることがわかります。また、手書き数字2を判定する役割のニューロン z_2 はフィルター1、3に対応するプーリング表と強く結びついていることがわかります。

フィルター 1

0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.82
0.00	0.00	0.00	0.52
0.00	0.00	0.26	2.42

→ フィルター 1 から
得られるプーリング表

プーリング表

p_{11}^{F1}	p_{12}^{F1}	p_{13}^{F1}
p_{21}^{F1}	p_{22}^{F1}	p_{23}^{F1}
p_{31}^{F1}	p_{32}^{F1}	p_{33}^{F1}

フィルター 2

0.00	0.00	0.00	0.87
0.00	0.00	0.00	0.86
0.00	0.00	0.00	0.75
0.00	0.00	0.00	0.00

→ フィルター 2 から
得られるプーリング表

出力層

p_{11}^{F2}	p_{12}^{F2}	p_{13}^{F2}
p_{21}^{F2}	p_{22}^{F2}	p_{23}^{F2}
p_{31}^{F2}	p_{32}^{F2}	p_{33}^{F2}

 z_1 z_2

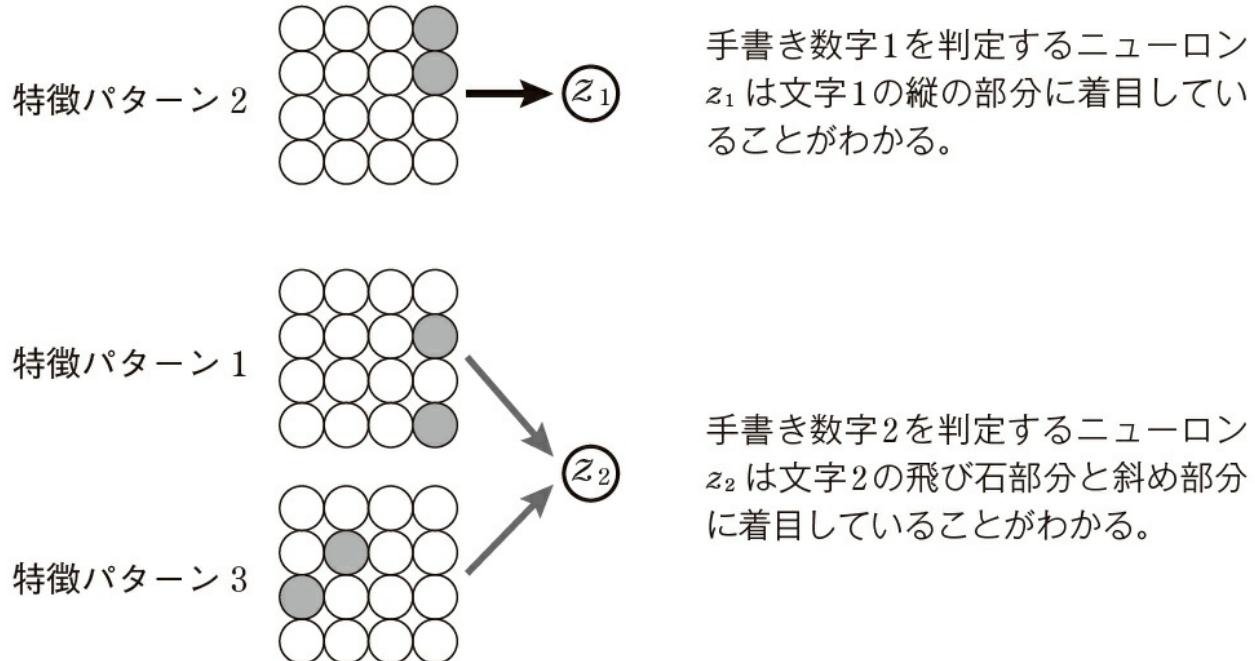
フィルター 3

0.00	0.10	0.00	0.04
0.00	2.61	0.95	0.01
1.32	0.08	0.00	0.00
0.00	0.11	0.00	0.00

→ フィルター 3 から
得られるプーリング表

p_{11}^{F3}	p_{12}^{F3}	p_{13}^{F3}
p_{21}^{F3}	p_{22}^{F3}	p_{23}^{F3}
p_{31}^{F3}	p_{32}^{F3}	p_{33}^{F3}

さて、先のフィルターの考察から、フィルター 1、2、3 は順に本節の「フィルターを見てみよう」の特徴パターン 1、2、3 を抽出しています。これらから、出力層のニューロン z_1 はフィルター 2 の抽出した特徴パターン 2 と強く結びついていることがわかります。また、出力層のニューロン z_2 はフィルター 1、3 の抽出した特徴パターン 1、3 と強く結びついていることがわかります。こうして、出力層のニューロンがどのようなパターンから手書き数字を識別しているかがわかりました。



■ 閾値は縁の下の力持ち

最適化によって、閾値は次のように得られました。

畠込み層 の閾値	1	2	3
	1.52	3.80	2.64

出力層 の閾値	1	2
	2.31	5.35

先にも述べたように、閾値は畠込みニューラルネットワークを陰で支える役割を持ちます。手書き数字の1と2とをしつかり区別するために、不要な情報をカットする役割を演じているのです。

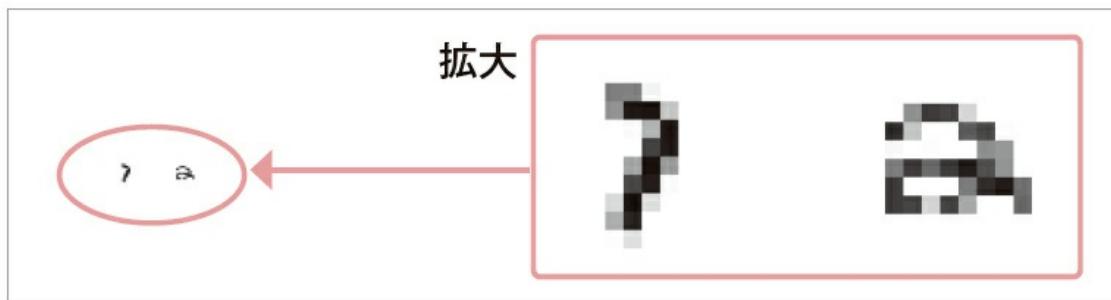
§ 11

畳み込みニューラルネットワークをテストしよう

先の節(§9)で決定された畳み込みニューラルネットワークは訓練データを用いて決定されました。ここでは、新たな手書き数字を正しく識別できるかを調べてみましょう。

■ 新たなデータを用意

訓練データの中にはない新たな手書き数字の画像に対して、作成した畳み込みニューラルネットワークが正しく機能するかを確かめてみましょう。ここでは、次の2つの数字画像をテストデータとして利用します。



人間には何とか左側の数字は1、右側の数字は2と読みますが、作成した畳み込みニューラルネットワークにはどう読めるかを、次の例題で調べましょう。

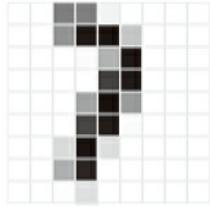
例題1 これまで作成した畳み込みニューラルネットワークが、上記の左の手書き数字をどう判読するか調べましょう。

注 この例題のワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「5.xlsx」の中の「テスト」タブに収められています。

解

まず、上記手書き数字の画像を数値化しましょう。

左側の数字



0	0	122	121	12	0	0	0	0	0
0	0	103	243	251	55	0	0	0	0
0	0	0	4	74	243	0	0	0	0
0	0	0	0	188	238	0	0	0	0
0	0	0	87	254	96	0	0	0	0
0	0	0	182	234	4	0	0	0	0
0	0	36	252	33	0	0	0	0	0
0	0	77	221	0	0	0	0	0	0
0	0	2	31	0	0	0	0	0	0

この数値データ画像をこれまでに作成したワークシートの画像データ部分にはめ込みます(パラメーターは最適化後のものを利用)。そして、次の基準にしたがって数字を判定します。

$$z_1 > z_2 \text{ のとき手書き数字は1, } z_1 < z_2 \text{ のとき手書き数字は2 } \dots \text{ (1)}$$

出力層のニューロンの出力値は「確信度」を表すと考えられるからです(→本章§1、§7)。確信度の大きな方が判読結果として採用されるわけです。

このワークシートからわかるように、畳み込みニューラルネットワークは手書き数字を「1」と判定しています。人間の感性と同じです。

例題2 これまで作成した畳み込みニューラルネットワークが、本項前出の図の右の手書き数字をどう判断するか調べましょう。

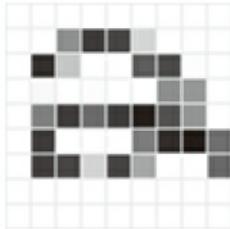
注 この例題のワークシートは、ダウンロードサイト(→巻頭「Excelサンプルファイルのダウンロードについて」)からダウンロードできます。

に掲載されたファイル「5.xlsx」の中の「テスト」タブに収められています。

解

まず、手書き数字の画像を次のように数値化しましょう。

右側の数字



0	0	0	0	0	0	0	0	0
0	0	107	195	184	40	0	0	0
0	209	56	0	0	158	178	2	0
0	6	0	2	0	0	108	99	0
0	122	194	143	195	233	171	102	0
0	194	0	0	0	125	212	237	138
0	193	184	42	186	84	0	0	145
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

先の例題と同じく、この数値データ画像を§6までに作成したワークシートの画像データ部分にはめ込みます(パラメーターは最適化後のものを利用)。そして、出力層のニューロンの値を求め、左記の式(1)にしたがって、手書き数字を判定します。

数字の判定結果。 $z_1 < z_2$ なので
画像を 2 と判定している

§9 で決定された
バラメーターの値

これまでに作成した
ワークシート

	A	B	C	D	E	F	G	H	I	U	V	W	X	Y	
1	手書き数字1、2の識別テスト									番号	2				
2							倍率	0.01				0	0	0	
3												0	0	0	
4												0	1.07	1.95	
5												0	0	1.84	
6												0	0.56	0	
7												0	0.06	0.02	
8												0	1.22	1.43	
9												0	1.94	0	
10												0	1.93	1.84	
11												0	0	0.42	
12												0	0	0	
13												0	0	0	
14												0	0	0.26	
15												0	0	2.42	
16	畳 み 込 み 層				F1	0.00	0.00	0.00	0.00			0.889	0.863	0.507	
17						0.00	0.00	0.00	1.82			0.921	0.973	0.999	
18						0.00	0.00	0.00	0.52			0.322	0.375	0.938	
19						0.00	0.00	0.00	2.42			0.930	0.999	0.997	
20					F2	0.00	0.00	0.00	0.87			0.213	0.364	0.767	
21						0.00	0.00	0.00	0.86			0.319	0.866	0.502	
22						0.00	0.00	0.00	0.75			0.108	0.089	0.093	
23						0.00	0.00	0.00	0.00			0.012	0.101	0.111	
24					F3	0.00	0.10	0.00	0.04			0.062	0.088	0.337	
25						0.00	2.61	0.95	0.01			0.073	0.108	0.299	
26						1.32	0.08	0.00	0.00			0.097	0.331	0.487	
27						0.00	0.11	0.0	0.00			0.031	0.101	0.122	
28						9	1.52	3.80	2.64			0.191	0.992	0.993	
29					O1	P1	0.00	0.00	0.00			0.973	0.332	0.094	
30							0.00	0.00	0.00			0.122	0.308	0.537	
31					P2	0.00	0.00	0.00	4.19			0.941	0.989	0.953	
32							0.00	0.00	0.00			0.940	0.578	0.516	
33							0.00	0.00	0.00			0.987	0.930	0.570	
34	出 力 層				P3	0.00	0.00	0.00	0.00			P1	0.973	0.989	
35							0.00	0.00	0.00				0.998	0.999	
36							0.00	0.00	0.00			P2	0.112	0.190	
37							0.01	0.02	0.87				0.108	0.552	
38							0.00	2.22	0.49			P3	0.992	0.983	
39							0.00	0.00	0.19				0.999	0.983	
40							0.00	0.10	0.00				0.987	0.957	
41							0.00	0.91	0.51				z1	z2	
42							2.46	0.00	0.01				0.136	0.989	
43							0.00	0.00	0.00				判定	2	
							9	2.31	5.35						

畳み込みニューラルネットワークは手書き数字を「2」と判定しています。これも人間の感性と一致しています。

Memo ニューラルネットワークの出力の解釈

ニューラルネットワークや、そのひとつである畳み込みニューラルネットワークの結論は出力層のニューロンの値に現れます。そのニューロンの活性化関数にシグモイド関数を利用すると、0、1の2値にはなりません。ようするに、白黒

の判断をしないのです。「ニューロンの出力は確信度と解釈できる」と何度も述べましたが、この辺の事情を指します。それは、悩みながら「たぶん、こちらが正しい」とする人間の所作に似ています。

Memo | さまざまな解

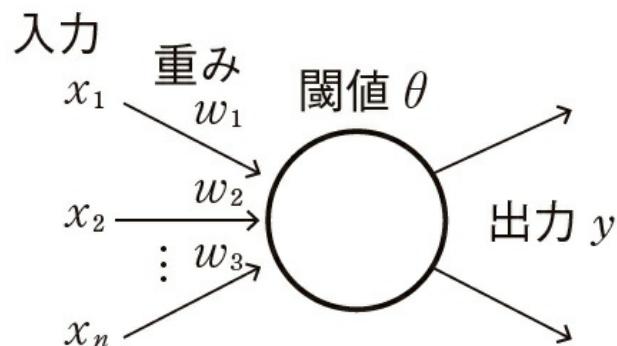
最初に設定した仮の値を変化させると、最適化後のパラメーターの真の値は変化します。しかし、判定の結果は同一です。それは生まれや育ちによって、同じ結論を出すのに様々な考え方をとる人の世界に似ています。結論が同じでも、それに至る道は一つではないことを、ニューラルネットワークは再確認させてくれます。

12 パラメーターに負を許容すると

これまで、フィルターの成分や重み、閾値について負の数は考えませんでした。しかし、モデルを最適化するということを至上命令とするなら、それらを負に設定することも可能です。

■ パラメーターに負の数も許容

ニューラルネットワークの基本は(人工)ニューロンです。何度か調べたように、このニューロンは次の関係で入力から出力値を算出します。



入力信号 x_1, x_2, \dots, x_n (n は自然数)を考え、各入力信号には重み w_1, w_2, \dots, w_n が与えられたとする。閾値を 0 とするとき、ニューロンの出力 y は

$$y = \sigma(a) \dots (1)$$

ここで、 a は入力の線形和と呼ばれ、次のように定義される。

$$a = w_1 x_1 + w_2 x_2 + \dots + w_n x_n - \theta \dots (2)$$

ところで、これまでの重み、閾値については非負数、すなわち 0 以上の数だけを

調べてきました。フィルターの成分も重みと考えられるので、これまでにはパラメーター全般について負の数は排除していたのです。

負の数を排除してきた理由は簡単明瞭です。上記のニューロンを生命の世界で考えれば、負の量は存在しないからです。しかし、もし生命という制約を離れ、モデルを最適化するということを最優先にするなら、負の数を用いることは数学的に大変有利です。それだけ自由度が増すからです。

■ バイアス

まず、パラメーターに負を許容するための準備として「バイアス」という言葉を導入します。入力の線形和の式(2)を見てください。

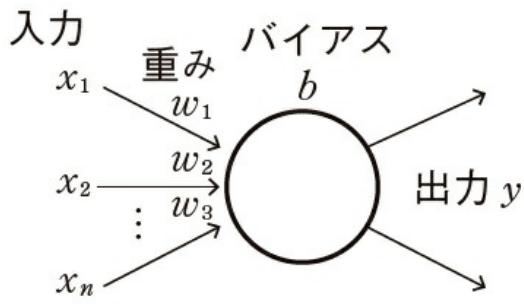
$$a = w_1x_1 + w_2x_2 + \dots + w_nx_n - \theta \dots (2) \text{ (再掲)}$$

ここで θ は「閾値」と呼ばれ、生物的にはニューロンの個性を表現する値です。直感的にいえば、 θ が大きければそのニューロンは刺激に対して興奮しにくく(すなわち鈍感)、小さければ興奮しやすい(すなわち敏感)という感受性を表します。

ところで、 θ だけマイナス記号が付いていて美しくありません。美しさが欠けることは数学が嫌うところです。また、マイナスは計算ミスを誘発しやすいという欠点を持ちます。そこで、 $-\theta$ を b と置き換えます。

$$a = w_1x_1 + w_2x_2 + \dots + w_nx_n + b \dots (3)$$

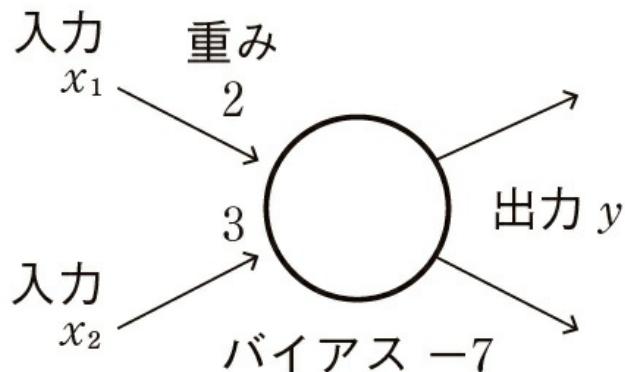
こうすれば式として美しく、計算ミスも起こりにくくなります。このように閾値 θ を変更したパラメーター b をバイアス(bias)と呼びます。



入力 x_1 、 x_2 、 x_3 、重み w_1 、 w_2 、 w_3 、バイアス b から入力の線形和は式(3)で算出される。

このように修正すると係数につく符号がすべて正になり、数学的には対等になります。閾値に負の数を許容しても、形式的には何の不都合も生じなくなるのです。

例1 次の図のニューロンがあります。図に示すように、入力 x_1 の重みは 2、入力 x_2 の重みは 3 とします。また、バイアスは -7 とします。



このとき、入力 (x_1, x_2) が $(1, 2)$ 、 $(2, 1)$ のとき、入力の線形和 a 、出力 y を電卓で求めてみましょう。ただし、活性化関数はシグモイド関数とし、 e の値を $e = 2.72$ として近似します。

入力 x_1	入力 x_2	入力の線形和 a	出力 y
1	2	$2 \times 1 + 3 \times 2 - 7 = 1$	0.73
2	1	$2 \times 2 + 3 \times 1 - 7 = 0$	0.50

■ 負の数を許容して最適化してみよう

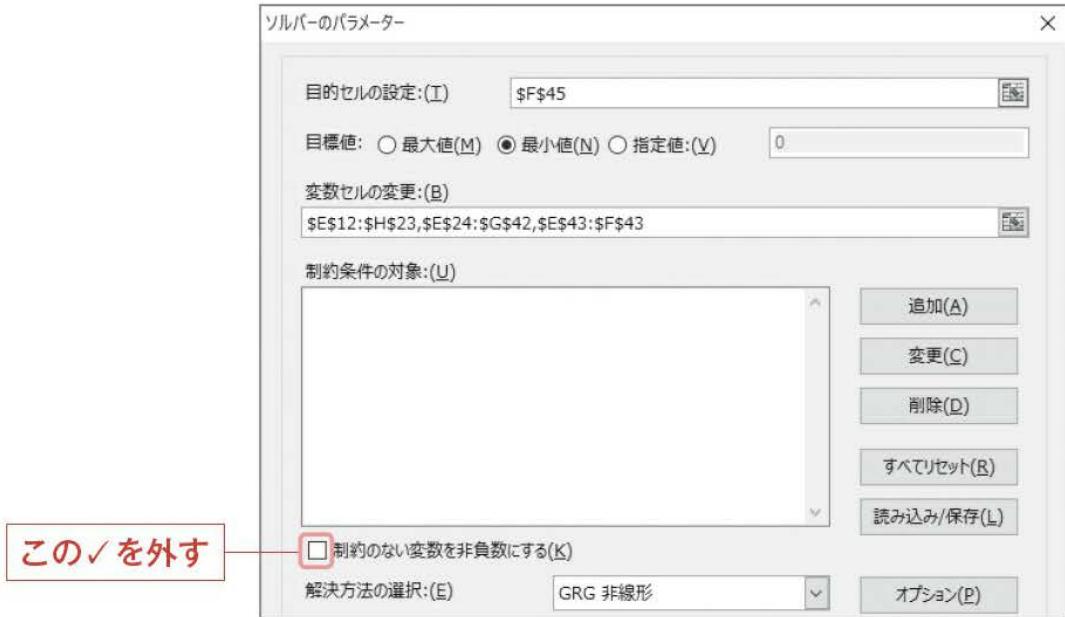
パラメーターに負の数を許容すると、一般的に計算は速くなり、最適化も深まります。しかし、仮想世界のニューロンで処理するため、得られた量の解釈はしづらくなります。具体例でそれを確かめてみましょう。

例題1 §9において、パラメーターに負の数を許容し、畳み込みニューラルネットワークのパラメーターを決定しましょう。

注 この例題のワークシートは、ダウンロードサイト（→卷頭「Excelサンプルファイルのダウンロードについて」）に掲載されたファイル「5.xlsx」の中の「負許可」タブに収められています。

解

パラメーターに負を許容しても、これまで作成してきたExcelのワークシートは変更を要しません。唯一変更する点は、ソルバーのオプション設定です。次の図のように、「制約のない変数を非負数にする」のチェックを外すだけですみます。



では、この設定の下で、これまで作成してきたワークシートを用いて最適化を実行してみましょう。パラメーターの初期値は§4、6で作成したものを探用します。

§9までに作成した関数に
変更はない

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
1	手書き数字1、2の識別(未学習)																
2						倍率	0.01										
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	
29																	
30																	
31																	
32																	
33																	
34																	
35																	
36																	
37																	
38																	
39																	
40																	
41																	
42																	
43																	
44																	
45																	

目的関数の値

目的関数の値は、次のようになりました。

$$Q_T = 4.03$$

非負数で最適化したときの値 $Q_T = 37.25$ (§9参照) に比べて、約1割になって
います。負の数を認めた方が、はるかにモデルはデータに適合することがわかりま

す。しかし、良いことばかりではありません。解釈がしづらくなるのです。それを見るために、前節(§10)と同様、得られたパラメーターの値を調べてみましょう。

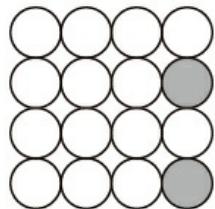
■ フィルターを見てみよう

最適化によって得られたフィルターの中身を見てみましょう。各フィルターの中で、大きい方の2つに網をかけてみました。

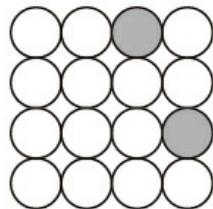
フィルター 1				フィルター 2				フィルター 3			
-1.27	-4.76	-19.02	4.66	-0.37	0.64	37.12	0.03	5.95	0.42	-6.68	-1.40
-2.22	0.19	-9.41	28.16	0.19	-0.52	0.79	-4.80	4.87	19.66	-2.61	5.56
4.79	3.50	0.10	13.01	-0.10	-2.29	0.41	7.47	3.08	0.29	-4.20	0.04
-2.61	-0.30	-4.97	18.89	-5.66	0.14	2.88	3.43	-0.32	0.03	-0.29	11.44

これらの表から、3つのフィルターの特徴がわかります。すなわち、訓練データの中の手書き数字画像において、次の3つのパターンを特徴抽出したのです。

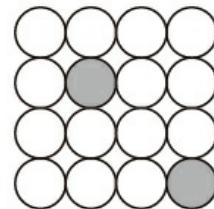
特徴パターン 1



特徴パターン 2



特徴パターン 3



§10では特徴パターンを実際の手書き文字と比べて解釈ができました。しかし、ここで求めた特徴パターンを簡単には解釈できません。困ったことにも思えますが、逆に「新しいものの見方の発見」につながると考えれば、畳み込みニューラルネットワークの評価につながります。

■ 出力層の重みを見てみよう

次に、出力層のニューロンがプーリング層の表(すなわちプーリング表)の各値に課した重みを見てみましょう。次の図では、1番大きな値の重みを課したプーリング層の表の値に網をかけています。

■出力層のニューロン 1

フィルター 1 の
プーリング表への重み

11.66	-0.95	2.43
9.33	-3.05	-11.60
1.91	-20.51	-1.71

フィルター 2 の
プーリング表への重み

2.10	-3.63	0.19
11.06	0.57	-12.28
-1.59	-0.55	-6.01

フィルター 3 の
プーリング表への重み

0.28	-3.50	5.98
10.55	-0.37	0.02
-2.85	-20.71	-6.29

■出力層のニューロン 2

フィルター 1 の
プーリング表への重み

-9.60	4.63	1.50
-11.25	4.27	15.11
-1.71	9.38	5.98

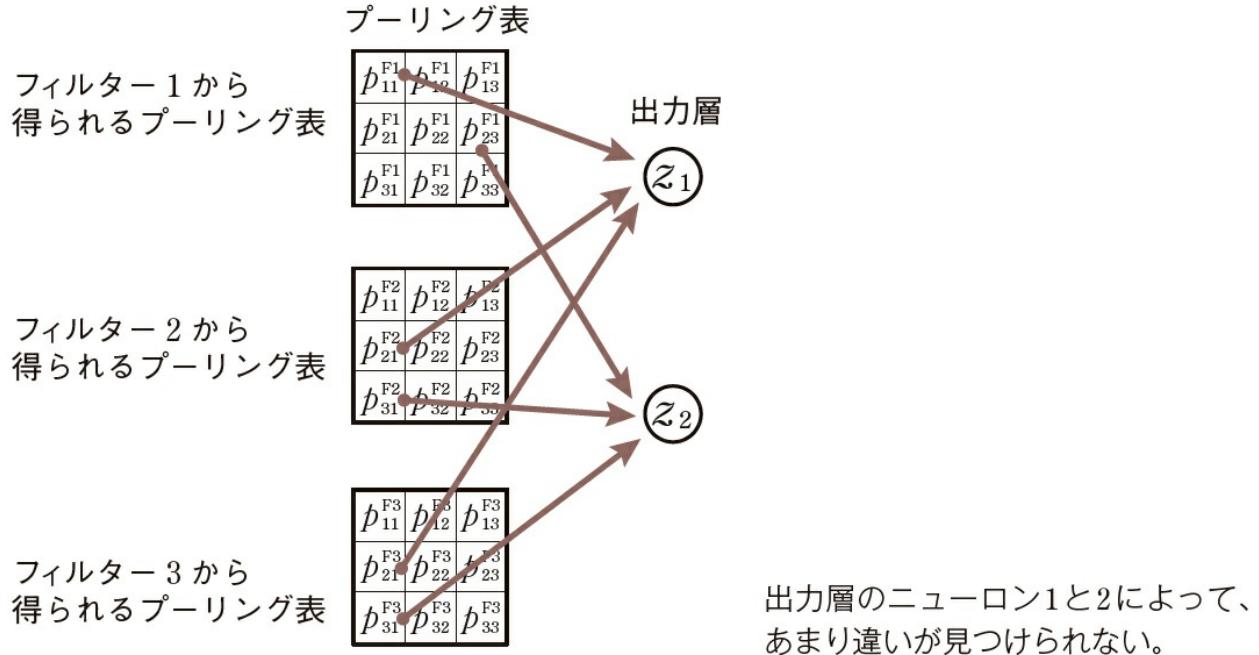
フィルター 2 の
プーリング表への重み

3.51	0.41	-1.29
-1.56	-6.47	0.00
25.95	3.53	9.18

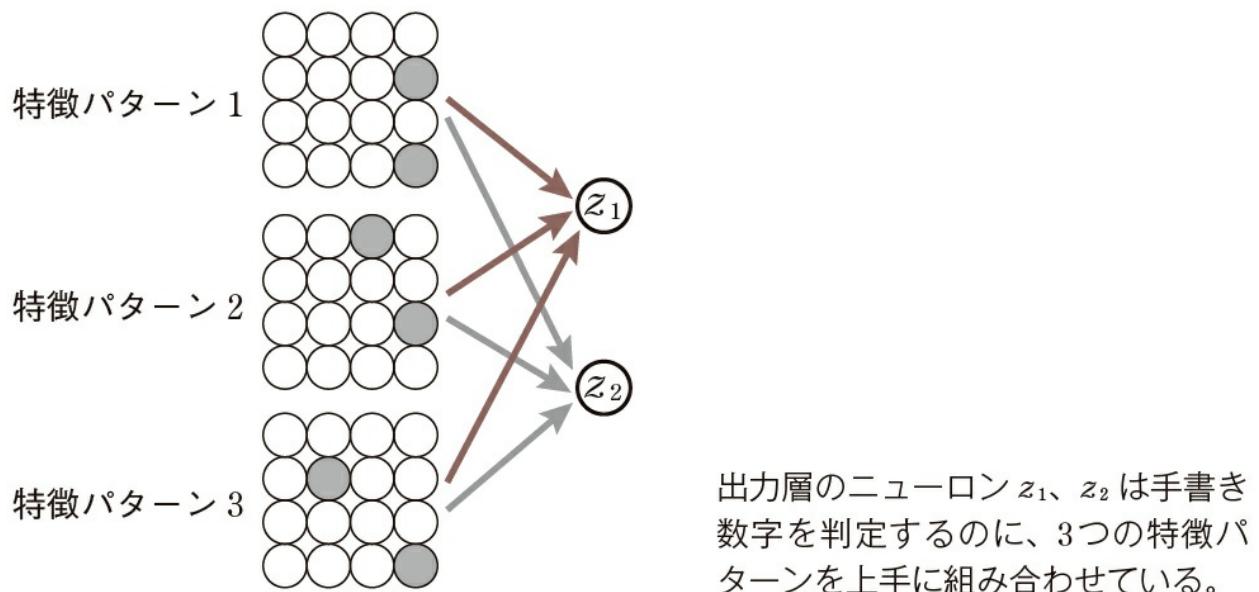
フィルター 3 の
プーリング表への重み

-3.68	0.65	-2.98
0.00	0.47	-0.52
8.45	2.15	5.05

パラメーターに負の値を用いなかった場合に比べて、出力層のニューロン z_1 と z_2 がプーリング層の各シートにかける重みは一様になっています。すなわち、出力層のニューロンとプーリング層の各表の相性に大きな違いがないのです。



もう少し詳しく見てみましょう。フィルター1、2、3は順に先の「フィルターを見てみよう」の特徴パターン1、2、3を抽出しています。すると、上記のことから、出力層のニューロン z_1 、 z_2 はこれらの特徴パターンと一緒に結びついていることがわかります。出力層のニューロン z_1 、 z_2 は抽出した特徴パターンの選り好みをしていないのです。

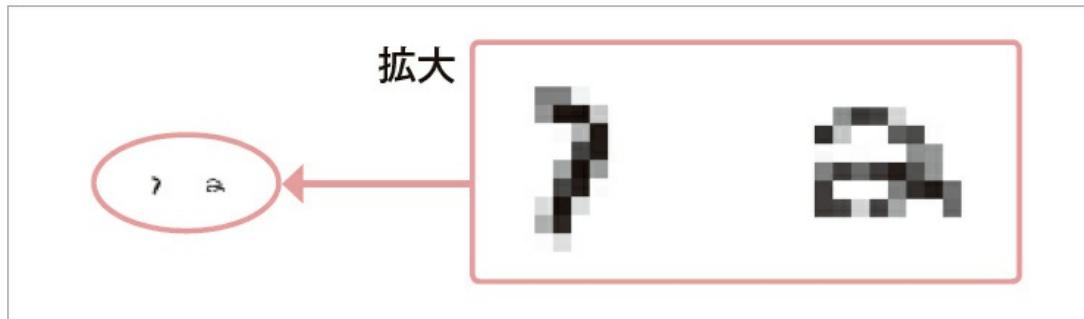


このことから、次のことがわかります。負の数を許容して最適化すると、出力層のニューロンは特徴パターンから直接文字を判定するのではなく、特徴パターンを上手に組み合わせて判定している、ということです。その組み合わせの妙については、直感的に理解することは困難でしょう。

注 負の重みについて、入門的な本書の性格上、考察していません。本来はしなければならないですが、負の値はイメージ的な解説が困難です。

■ 新たなデータでテスト

訓練データにない新たな手書き数字の画像に対して、作成した畳み込みニューラルネットワークが正しく機能するかを確かめてみましょう。§11と同じ次の2つの数字画像をテストデータとして利用します。



人間が読めば左の数字は1、右の数字は2でしょうが、ここで作成した畳み込みニューラルネットワークがどう判定するか見てみます。

例題2 前の図の2つの手書き数字を、ここで作成した畳み込みニューラルネットワークがどう判断するか調べましょう。

注 この例題のワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)

に掲載されたファイル「5.xlsx」の中の「負許可_テスト」タブに収められています。

解

テスト方法は§11と同じなので、結論だけを以下に示します。ワークシートは先の[例題1]で作成したものを利用しています。

数字の判定結果										前図の左の手書き数字の数値データ (1/100 されている)												
L42		fx		=IF(L40>M40,"1","2")																		
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R					
1	手書き数字1、2の識別テスト				倍率		0.01	No	1													
2									0	0	1.22	1.21	0.12	0								
3									0	0	1.03	2.43	2.51	0.55								
4									0	0	0	0.04	0.74	2.43								
5									0	0	0	0	1.88	2.38								
6									0	0	0	0.87	2.54	0.96								
7									0	0	0	1.82	2.34	0.04								
8									0	0	0.36	2.52	0.33	0								
9									0	0	0.77	2.21	0	0								
10									0	0	0.02	0.31	0	0								
11																						
12	置 み 込 み 層	F1	-1.27	-4.76	-19.02	4.66	yF1		1.00	1.00	1.00	0.00	0.03	0.46								
13			-2.22	0.19	-9.41	28.16			0.99	1.00	1.00	0.00	0.28	0.17								
14			4.79	3.50	0.10	13.01			1.00	1.00	1.00	0.00	0.00	0.00								
15			-2.61	-0.30	-4.97	18.89			1.00	1.00	0.00	0.00	0.00	0.00								
16		F2	-0.37	0.64	37.12	0.03			1.00	1.00	0.00	0.00	0.00	0.00								
17			0.19	-0.52	0.79	-4.80			1.00	0.00	0.00	0.00	0.00	0.01								
18			-0.10	-2.29	0.41	7.47			1.00	1.00	1.00	1.00	0.00	0.00								
19			-5.66	0.14	2.88	3.43			1.00	1.00	1.00	1.00	0.07	0.13								
20		F3	5.95	0.42	-6.68	-1.40			1.00	1.00	1.00	1.00	0.00	0.12								
21			4.87	19.66	-2.61	5.56			1.00	1.00	1.00	1.00	0.07	0.13								
22			-0.39	-2.79	-1.29	0.11			1.00	1.00	1.00	1.00	0.05	0.18								
23			-0.32	0.03	-0.29	11.44			0.99	1.00	1.00	0.37	0.12	0.23								
24		θ	4.36	1.18	11.26				0.00	1.00	1.00	1.00	1.00	0.13								
25	出 力 層	O1	P1	11.66	-0.95	2.43	yF3		0.00	0.91	0.00	0.27	1.00	1.00								
26				9.33	-3.05	-11.60			1.00	1.00	0.00	0.93	1.00	1.00								
27				1.91	-20.51	-1.71			1.00	0.00	0.00	1.00	1.00	1.00								
28			P2	2.10	-3.63	0.19			1.00	0.00	0.72	1.00	1.00	0.00								
29				11.06	0.57	-12.28			0.41	0.00	1.00	1.00	0.99	0.00								
30				-1.59	-0.55	-6.01			P1	1.00	1.00	0.46										
31			P3	0.28	-3.50	5.98				1.00	1.00	0.00										
32				10.55	-0.37	0.02			P2	1.00	1.00	0.00										
33				-2.85	-20.71	-6.29			P3	1.00	1.00	1.00										
34		O2	P1	-9.60	4.63	1.50				1.00	1.00	0.13										
35				-11.25	4.27	15.11				1.00	1.00	0.25										
36				-1.71	9.38	5.98				1.00	1.00	1.00										
37			P2	3.51	0.41	-1.29				1.00	1.00	1.00										
38				-1.56	-6.47	0.00				1.00	1.00	1.00										
39				25.95	3.53	9.18				1.00	1.00	1.00										
40			P3	-3.68	0.65	-2.98				z1	z2											
41				0.00	0.47	-0.52				1.00	0.02											
42				8.45	2.15	5.05				z1 > z2												
43		θ		-11.17	28.21					1												

$z_1 > z_2$ なので、数字画像を1と判定している。

畳み込みニューラルネットワークは本項前出の図の左の手書き数字を「1」と判定しています。人間の感性と同じです。

次に、本項前出の図の右側の数字を判定させてみましょう。これも、そのテストのための設定の仕方は同一なので解説は省略し、結論だけを以下に

示します。

数字の判定結果					前々図の右の手書き数字の 数値データ (1/100 されている)												
W42		fx			=IF(W40>X40,"1","2")												
A	B	C	D	E	F	G	H	I	U	V	W	X	Y	Z	AA	AB	AC
1	手書き数字1、2の識別テスト					No	2										
2					倍率	0.01					0	0	0	0	0	0	0
3											0	0	1.07	1.95	1.84	0.4	
4											0	2.09	0.56	0	0	1.58	1.71
5											0	0.06	0	0.02	0	0	1.01
6											0	1.22	1.94	1.43	1.95	2.33	1.7
7											0	1.94	0	0	0	1.25	21
8											0	1.93	1.84	0.42	1.86	0.84	
9											0	0	0	0	0	0	
10											0	0	0	0	0	0	
11																	
12	重 み 込 み 層	F1	-1.27	-4.76	-19.02	4.66					yF1	1.00	1.00	1.00	1.00	1.00	0.96
13			-2.22	0.19	-9.41	28.16					0.05	0.00	1.00	1.00	0.00	0.00	
14			4.79	3.50	0.10	13.01					0.13	1.00	1.00	1.00	1.00	0.96	
15			-2.61	-0.30	-4.97	18.89					1.00	1.00	1.00	1.00	1.00	1.00	
16		F2	-0.37	0.64	37.12	0.03					0.00	0.91	1.00	0.99	1.00	0.90	
17			0.19	-0.52	0.79	-4.80					0.00	1.00	0.97	0.00	0.00	0.00	
18			-0.10	-2.29	0.41	7.47					yF2	0.00	0.00	1.00	1.00	0.94	0.09
19			-5.66	0.14	2.88	3.43					1.00	1.00	1.00	1.00	0.99	0.00	
20		F3	5.95	0.42	-6.68	-1.40					1.00	0.12	1.00	1.00	1.00	0.80	
21			4.87	19.66	-2.61	5.56					0.01	0.00	0.02	1.00	1.00	1.00	
22			3.08	0.29	-4.20	0.04					1.00	1.00	1.00	1.00	1.00	1.00	
23			-0.32	0.03	-0.29	11.44					0.18	0.00	0.03	1.00	1.00	1.00	
24	出 力 層	θ	4.36	1.18	11.26						yF3	0.01	1.00	1.00	1.00	1.00	0.01
25		O1	P1	11.66	-0.95	2.43					1.00	1.00	1.00	1.00	1.00	1.00	
26				9.33	-3.05	-11.60					0.00	0.36	0.95	0.72	1.00	1.00	
27				1.91	-20.51	-1.71					1.00	1.00	1.00	1.00	1.00	1.00	
28			P2	2.10	-3.63	0.19					0.99	0.23	0.00	0.00	1.00	1.00	
29				11.06	0.57	-12.28					1.00	1.00	0.98	1.00	0.07	0.00	
30				-1.59	-0.55	-6.01					P1	1.00	1.00	1.00			
31			P3	0.28	-3.50	5.98						1.00	1.00	1.00			
32				10.55	-0.37	0.02						1.00	1.00	1.00			
33				-2.85	-20.71	-6.29					P2	1.00	1.00	0.99			
34	出 力 層	O2	P1	-9.60	4.63	1.50						1.00	1.00	1.00			
35				-11.25	4.27	15.11					P3	1.00	1.00	1.00			
36				-1.71	9.38	5.98						1.00	1.00	1.00			
37			P2	3.51	0.41	-1.29						1.00	1.00	1.00			
38				-1.56	-6.47	0.00						1.00	1.00	1.00			
39				25.95	3.53	9.18						z1	z2				
40			P3	-3.68	0.65	-2.98						0.00	1.00				$z_1 < z_2$
41				0.00	0.47	-0.52											
42				8.45	2.15	5.05											
43		θ	-11.17	28.21							判定	2					

$z_1 < z_2$ なので、数字画像を1と判定している。

畳み込みニューラルネットワークは手書き数字を「2」と判定しています。これも人間の感性と一致しています。

§

13 隠れ層の活性化関数を変更

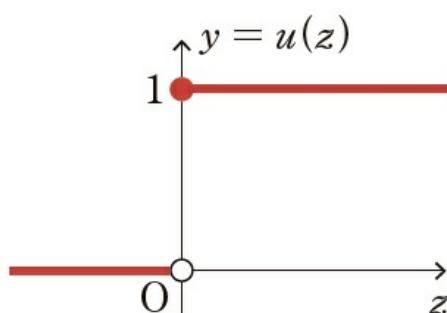
これまでにはニューロンの活性化関数としてシグモイド関数を利用してきました。しかし、考えてみればそれにこだわる必然性はありません。ここでは、シグモイド関数の代わりに「ランプ関数」を利用してみましょう。

■ ランプ関数とReLU

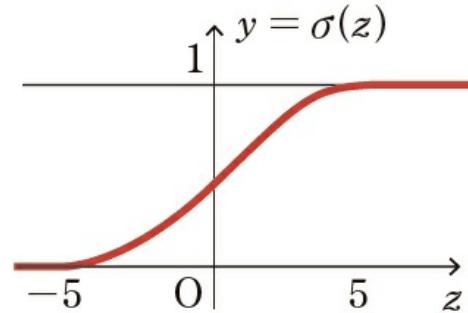
これまでにはニューロンとしてシグモイドニューロンを利用してきました。シグモイドニューロンとはシグモイド関数 $\sigma(x)$ を活性化関数として利用する人工ニューロンです(→3章)。

$$\text{シグモイド関数: } \sigma(x) = \frac{1}{1+e^{-x}} \quad \cdots (1)$$

シグモイド関数はステップ関数に似ている、微分計算が簡単、などの理由で広く採用されています。



ステップ関数

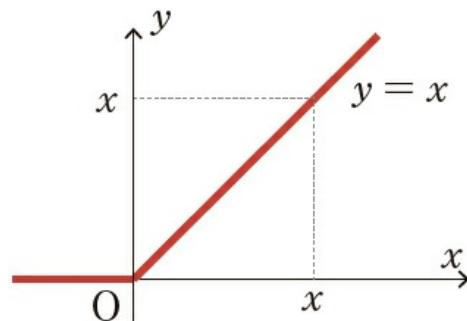


シグモイド関数

ところで、これらの条件を満たす関数はシグモイド関数だけではありません。電子工学の分野で有名なランプ関数もそれに該当します。ランプ関数とは次のよう

に定義される関数です。グラフをその右側に描きましょう。

$$\text{ランプ関数} : f(x) = \begin{cases} x & (x \geq 0) \\ 0 & (x < 0) \end{cases}$$

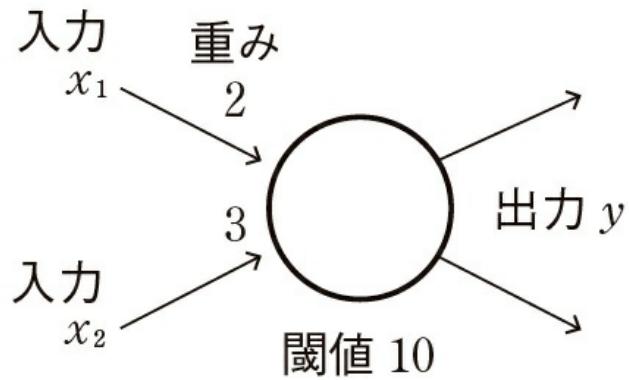


このランプ関数にはシグモイド関数にないメリットがあります。計算が単純であるということです。実用的な畳み込みニューラルネットワークは膨大なニューロンから構成されているので、計算がしやすいということは大変ありがたい性質です。

計算がしやすいランプ関数を活性化関数とするニューロンをReLU (Rectified Linear Unit)と呼びます。

シグモイド関数の大きなメリットに0と1の間に値が収まることが挙げられます。そのおかげで、その出力値を活性度とか、確信度、興奮度などさまざまに解釈できました。しかし、ReLUの出力は0と1の間に収まりません。出力の解釈がしづらいという欠点があります。そこで、解釈が不要な隠れ層(中間層)のニューロンとしてよく利用されます。

例1 次の図のニューロンがあります。図に示すように、入力 x_1 の重みは2、入力 x_2 の重みは3とします。また、閾値は10とします。



ReLU の例

このとき、入力(x_1, x_2)が(3, 1)、(1, 3)のとき、入力の線形和 a 、出力 y を求めましょう。

入力 x_1	入力 x_2	入力の線形和 a	出力 y
3	1	$2 \times 3 + 3 \times 1 - 10 = -1$	0
1	3	$2 \times 1 + 3 \times 3 - 10 = 1$	1

■ ReLUを用いて最適化してみよう

ReLU関数を活性化関数として、これまで用いてきた手書き文字「1」、「2」の判別のための畳み込みニューラルネットワークを作成してみましょう。

例題1 これまで作成してきたExcelのワークシートにおいて、隠れ層のニューロンをReLUに変更します。このとき、これまで用いてきた訓練データから畳み込みニューラルネットワークのパラメーターを決定しましょう。

注 この例題のワークシートは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)

に掲載されたファイル「5.xlsx」の中の「ReLU」タブに収められています。

解

§9までに作成したExcelのワークシートで、畳み込み層のシグモイド関数をすべてランプ関数に書き換えます。ここでは、次のようにランプ関数を実装しましょう。

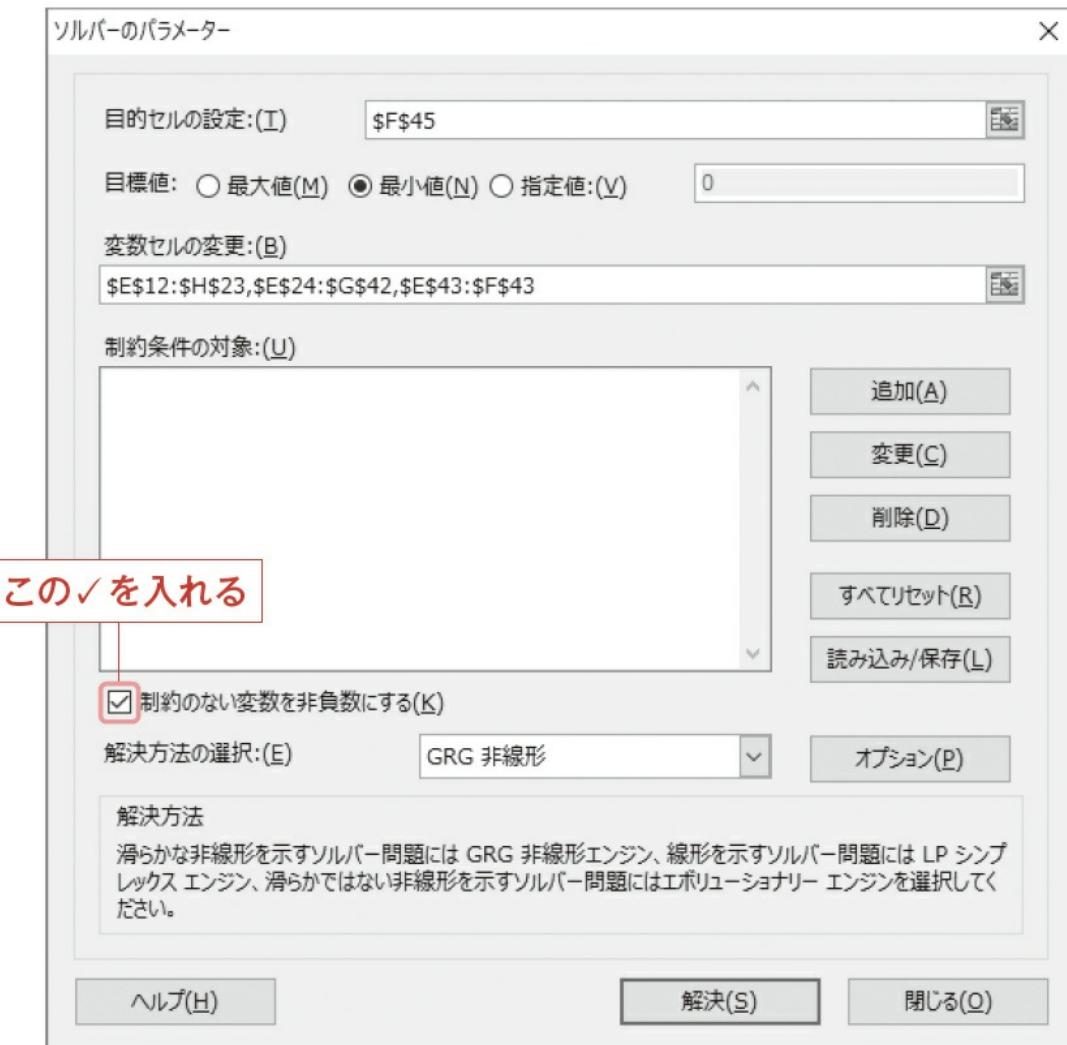
$f(a) = \text{MAX}(0, a)$ (a は入力の線形和)

手書き数字1、2の識別(ReLU利用)										
1	A	B	C	D	E	F	G	H	I	J
2										K
3										L
4										M
5										N
6										O
7										P
8										Q
9										R
10										S
11	§4、6で作成した初期値を利用									
12	F1	0.01	0.04	0.02	0.02					
13		0.01	0.01	0.03	0.03					
14		0.03	0.04	0.04	0.05					
15		0.02	0.01	0.04	0.04					
16	F2	0.03	0.05	0.03	0.04					
17		0.01	0.02	0.01	0.01					
18		0.03	0.03	0.04	0.04					
19		0.05	0.03	0.03	0.02					
20	F3	0.04	0.03	0.02	0.01					
21		0.05	0.01	0.03	0.01					
22		0.04	0.00	0.02	0.05					
23		0.01	0.03	0.0	0.03					
24	B	0.04	0.02	0.04						
25	O1 P1	0.05	0.04	0.00						
26		0.03	0.02	0.04						
27		0.00	0.02	0.01						
28	P2	0.03	0.01	0.05						
29		0.00	0.00	0.02						

この図は、訓練データの1番目の画像について、隠れ層の活性化関数をランプ関数にしています。§8で示したように、この1番目の画像部分を全訓練データぶんコピーし、畳み込みニューラルネットワークのワークシートが完成します。

注 本書では、出力層についてはシグモイド関数を利用します。

ワークシートが完成したなら、§9で示したように、ソルバーを用いて最適化します。その際、本節では解釈がしやすいように、非負数のパラメーターで最適化してみましょう。



では、この設定のもとで、先に作成したワークシートを用いて最適化を実行してみます。パラメーターの初期値は§4、6で利用したものを探用しています。

F45		X ✓		fx	=SUM(L41:BN41)												
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	手書き数字1、2の識別(ReLU利用)				倍率		0.01										
2																	
3																	
4																	
5																	
6		ソルバーの算出した 最適化解															
7																	
8																	
9																	
10																	
11																	
12	層 フ イ ル タ ー	F1	0.00	0.00	0.00	0.38											
13			0.00	0.00	1.01	0.76											
14			0.00	0.08	1.32	0.93											
15			0.00	0.05	0.00	0.13											
16		F2	0.09	0.00	0.00	0.00											
17			0.00	0.00	0.00	0.00											
18			0.43	0.22	0.02	0.54											
19			0.00	0.02	0.00	0.00											
20		F3	0.00	0.00	0.00	0.00											
21			0.00	0.00	0.00	0.00											
22			0.00	0.00	0.00	0.76											
23			0.00	0.00	0.0	1.07											
24	層 出 力 層 P リ ン グ	θ	0.00	1.38	0.87												
25			O1	P1	0.00	0.00	0.00										
26					0.00	0.00	0.00										
27					0.00	0.00	0.00										
28		P2	0.00	0.00	0.00	0.00											
29					0.00	0.00	0.00										
30					0.00	0.00	0.00										
31			P3	1.01	0.00	0.00											
32					0.00	0.00	0.00										
33		O2	0.02	0.00	0.00												
34			P1	0.00	0.00	0.00											
35					0.00	0.00	0.43										
36					0.00	0.00	2.27										
37		P2	0.06	0.00	0.00												
38					1.10	0.29	0.00										
39					1.04	2.19	0.00										
40			P3	0.00	0.00	0.00											
41					0.00	0.00	0.23										
42					0.00	0.00	0.95										
43		θ	2.48	2.27													
44					Q_T	37.86											
45																	

最適化されたときの
目的関数 Q_T の値

Memo ランプ関数の「ランプ」の意味

ランプ関数はそのグラフが傾斜路(ramp)に似ていることから命名されています。「高井戸ランプ付近で渋滞1km」などと、高速道の道路情報で「ランプ」はよく使われますが、これは立体交差部分が傾斜路になっているからで

す。

目的関数の値は次の値になっています。

$$Q_T = 37.86$$

§9で調べたシグモイドニューロンに比べて、ほぼ同じ適合度です。隠れ層の解釈を厳密に考えない限り、計算の速いReLUを利用することは大きなメリットがあることになります。

■ フィルターを見てみよう

最適化によって得られたフィルターの中身を見てみましょう。各フィルターの中で、相対的に大きいと思われる0.4以上の値について、網をかけてみました。

フィルター 1

0.00	0.00	0.00	0.38
0.00	0.00	1.01	0.76
0.00	0.08	1.32	0.98
0.00	0.05	0.00	0.13

フィルター 2

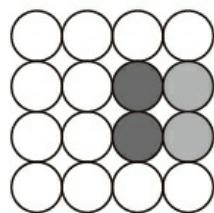
0.09	0.00	0.00	0.00
0.00	0.00	0.00	0.00
0.43	0.22	0.02	0.54
0.00	0.02	0.00	0.00

フィルター 3

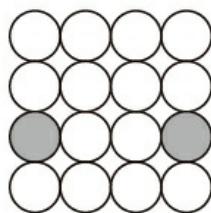
0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.76
0.00	0.00	0.0	1.07

この表から、3つのフィルターの特徴がわかります。すなわち、訓練データの中の手書き数字画像において、次の3つのパターンを特徴抽出したのです。

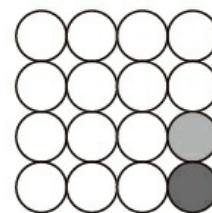
特徴パターン 1



特徴パターン 2



特徴パターン 3



■ 出力層の重みを見てみよう

次に、出力層のニューロンがプーリング表に課した重みを見てみましょう。次の図では、比較的大きい値と思われる0.9以上の重みに網をかけています。出力層のニューロン1と2とによって、重みのかけ方が大きく異なっています。

■出力層のニューロン 1

フィルター 1 の
プーリング表への重み

0.00	0.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00

フィルター 2 の
プーリング表への重み

0.00	0.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00

フィルター 3 の
プーリング表への重み

1.01	0.00	0.00
0.00	0.00	0.00
0.02	0.00	0.00

■出力層のニューロン 2

フィルター 1 の
プーリング表への重み

0.00	0.00	0.00
0.00	0.00	0.43
0.00	0.00	2.27

フィルター 2 の
プーリング表への重み

0.06	0.00	0.00
1.10	0.29	0.00
1.04	2.19	0.00

フィルター 3 の
プーリング表への重み

0.00	0.00	0.00
0.00	0.00	0.23
0.00	0.00	0.95

この図の網のかかった部分だけに着目して、プーリング層から出力層に矢を描いてみましょう(同じ表から複数の矢があるときは1本で代表させています)。この図から、手書き数字1を判定する役割のニューロン z_1 はフィルター3のプーリング表と強く結びついていることがわかります。また、手書き数字2を判定する役割のニューロン z_2 はフィルター1～3のすべてのプーリング表と強く結びついていることがわかります。

フィルター 1 から
得られるプーリング表

p_{11}^{F1}	p_{12}^{F1}	p_{13}^{F1}
p_{21}^{F1}	p_{22}^{F1}	p_{23}^{F1}
p_{31}^{F1}	p_{32}^{F1}	p_{33}^{F1}

フィルター 2 から
得られるプーリング表

p_{11}^{F2}	p_{12}^{F2}	p_{13}^{F2}
p_{21}^{F2}	p_{22}^{F2}	p_{23}^{F2}
p_{31}^{F2}	p_{32}^{F2}	p_{33}^{F2}

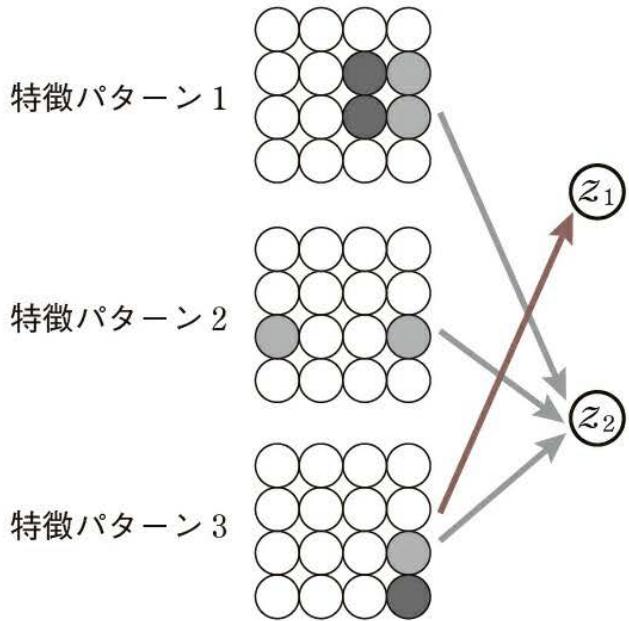
フィルター 3 から
得られるプーリング表

p_{11}^{F3}	p_{12}^{F3}	p_{13}^{F3}
p_{21}^{F3}	p_{22}^{F3}	p_{23}^{F3}
p_{31}^{F3}	p_{32}^{F3}	p_{33}^{F3}

z_1

z_2

さて、先のフィルターの考察から、フィルター 1、2、3 は順に前項の特徴パターン 1、2、3 を抽出しています。これらから、出力層の 1 番目のニューロン z_1 はフィルター 3 と強く結びついています。それに対して、出力層の 2 番目のニューロン z_2 はフィルター 1、2、3 と均等に結びついていることがわかります。すなわち、出力層のニューロン z_1 は特徴パターン 3 と、出力層のニューロン z_2 は特徴パターン 1、2、3 と次の図のように結びついていることがわかります。



手書き数字1を判定する役割を担うニューロン z_1 は、空白に隣接する右下の縦のパターンに着目していることがわかる。

手書き数字2を判定する役割を担うニューロン z_2 は、3つのパターンを組み合わせて判断していることがわかる。

このことから、出力層のニューロンがどのようなパターンから手書き数字を識別しているかがわかります。出力層のニューロン z_1 は特徴パターン3の有無で画像が「1」と判断しているのです。それに対して、出力層のニューロン z_2 は特徴パターン1、2、3を組み合わせて判断しているのです。

ReLUモデルでは実在の神経細胞モデルから離れたぶん、ニューロン z_2 の直感的な解釈がしにくくなっています。

■ 新たなデータでテスト

訓練データの中にはない新たな手書き数字の画像に対して、畳み込みニューラルネットワークが正しく機能するかを確かめてみましょう。§11と同じ、次の2つの数字画像をテストデータとして利用します。



人間が読めば左の数字は1、右の数字は2でしょうが、ここで作成した畳み込みニューラルネットワークがどう判定するか見てみます。

例題2 前図の左の手書き数字を、ここで作成した畳み込みニューラルネットワークがどう判断するか調べましょう。

注 この例題のワークシートは、ダウンロードサイト（→卷頭「Excelサンプルファイルのダウンロードについて」）に掲載されたファイル「5.xlsx」の中の「ReLU_テスト」タブに収められています。

解

テスト方法は§11と同じなので、結論だけを以下に示します。ワークシートは先の[例題1]で作成したものを利用しています。

手書き数字1、2の識別テスト										No	1							
1	(ReLU利用)	倍率	0.01									0	0	1.22	1.21	0.12	0	
2												0	0	1.03	2.43	2.51	0.55	
3												0	0	0	0.04	0.74	2.43	
4												0	0	0	0	1.88	2.38	
5	前図のテスト用画像 数値データ											0	0	0.87	2.54	0.96		
6												0	0	0	1.82	2.34	0.04	
7												0	0	0.36	2.52	0.33	0	
8												0	0	0.77	2.21	0	0	
9												0	0	0.02	0.31	0	0	
10																		
11																		
12	畳み込み層	F1	0.00	0.00	0.00	0.38						F1	3.39	5.38	6.49	3.92	0.30	0.00
13			0.00	0.00	1.01	0.76						1.07	3.63	7.67	5.87	0.23	0.00	
14			0.00	0.08	1.32	0.93						1.05	5.51	9.05	3.98	0.07	0.00	
15			0.00	0.05	0.00	0.13						2.67	8.17	7.61	1.22	0.00	0.00	
16		F2	0.09	0.00	0.00	0.00						4.81	8.30	3.50	0.06	0.00	0.00	
17			0.00	0.00	0.00	0.00						6.09	6.68	0.53	0.00	0.00	0.00	
18			0.43	0.22	0.02	0.54						F2	0.00	0.00	0.05	0.00	0.00	0.00
19			0.00	0.02	0.00	0.00							0.00	0.00	0.04	0.00	0.17	0.00
20		F3	0.00	0.00	0.00	0.00							0.00	0.00	0.00	0.00	0.00	0.00
21			0.00	0.00	0.00	0.00							0.00	0.00	0.00	0.00	0.00	0.00
22			0.00	0.00	0.00	0.76							0.00	0.00	0.00	0.00	0.00	0.00
23			0.00	0.00	0.00	1.07							0.00	0.00	0.00	0.00	0.00	0.00
24		θ	0.00	1.38	0.87							F3	0.00	1.71	3.53	0.00	0.00	0.00
25	出力層	O1	P1	0.00	0.00	0.00							0.06	3.29	1.97	0.00	0.00	0.00
26				0.00	0.00	0.00							1.74	3.57	0.00	0.00	0.00	0.00
27				0.00	0.00	0.00							3.22	1.26	0.00	0.00	0.00	0.00
28			P2	0.00	0.00	0.00							3.42	0.00	0.00	0.00	0.00	0.00
29				0.00	0.00	0.00							1.14	0.00	0.00	0.00	0.00	0.00
30				0.00	0.00	0.00						P1	5.38	7.67	0.30			
31			P3	1.01	0.00	0.00							8.17	9.05	0.07			
32				0.00	0.00	0.00							8.30	3.50	0.00			
33				0.02	0.00	0.00						P2	0.00	0.05	0.17			
34		O2	P1	0.00	0.00	0.00							0.00	0.00	0.00			
35				0.00	0.00	0.43							0.00	0.00	0.00			
36				0.00	0.00	2.27						P3	3.29	3.53	0.00			
37				P2	0.06	0.00	0.00						3.57	0.00	0.00			
38					1.10	0.29	0.00						3.42	0.00	0.00			
39					1.04	2.19	0.00											
40				P3	0.00	0.00	0.00						z1	z2		$z_1 > z_2$		
41					0.00	0.00	0.23						0.71	0.10				
42					0.00	0.00	0.95						判定	1				
43					2.48	2.27												

$z_1 > z_2$ なので、数字画像を1と判定している。

畳み込みニューラルネットワークは本項前出の図の左の手書き数字を「1」と判定しています。人間の感性と同じです。

次に、本項前出の図の右側の数字を判定させてみましょう。これも、そのテストのための設定の仕方は同一なので解説は省略し、結論だけを以下に示します。

$z_1 < z_2$ なので、数字画像を 2 と判定している。

畳み込みニューラルネットワークは手書き数字を「2」と判定しています。これも人間の感性と一致しています。

Memo 目的関数の値の評価

平方誤差の総和である目的関数の値はニューラルネットワークがどれだけ訓練データとズれているかの目安を与えます。しかし、ただ小さいからといつ

て、ニューラルネットワークがより良い判断をするとは限りません。§12で調べた負を許容するパラメーターの場合、非負の場合(§9)よりも、目的関数 Q_T の値ははるかに小さくなりました。しかし、訓練データについての正答率は却つて低くなります。畳み込みニューラルネットワークを含むニューラルネットワークの持つこのような「癖」を知っておくことは、応用上大切でしょう。

そもそも、ニューラルネットワークと訓練データとの適合具合を表す目的関数は一種ではありません。本書では誤差の目安として平方誤差の和を用いましたが、近年有名な目的関数にクロスエントロピーがあります。クロスエントロピーは情報科学から生まれた概念ですが、最適化のための計算の収束性が良いことが知られています。2つの尺度で誤差を見積もったとき、どちらが優れているかの基準を明確化することは困難です。

参考 本章で利用した記号のまとめ

本章の関係式で利用する記号は多彩なので、ここに整理します。

記号名	意味
x_{ij}	入力層 i 行 j 番目のニューロンの出力を表す変数。また、その名称としても利用。通常、入力層はデータ加工をしないが、本書では画素値を 100 分の 1 に変更して出力。
y_{ij}^{Fk}	畳み込み層の k 枚目の i 行 j 列にある成分。
z_k	出力層 k 番目のニューロンの出力を表す変数。また、その名称としても利用。
w_{ij}^{Fk}	隠れ層 k 番目のニューロンが用いるフィルターの i 行 j 列成分。ネットワークを定めるパラメーターである。
w_{k-ij}^{On}	出力層の n 番目のニューロンがプーリング層の k 枚目の表の i 行 j 列成分に課す重み。
θ^{Fk}	隠れ層 k 番目のニューロンの閾値。
θ^{On}	出力層 n 番目にあるニューロンの閾値。
a_{ij}^{Fk}	入力層を小分けしたとき、その ij 地区から隠れ層 k 番目のニューロンへの入力の線形和。
a^{On}	出力層 n 番目のニューロンに関する入力の線形和。
p_{ij}^{Fk}	プーリング層にある k 枚目のプーリング表の i 行 j 列成分。

何度も言及しているように、Excelで畳み込みニューラルネットワークを実装する際には、このような記号の意味を細かく理解する必要はありません。大切なことはニューロン間の関係をしっかり理解しておくことで、それが頭にあればワークシート作成は容易です。

§ A

訓練データ(1)

4章の例題で作成するニューラルネットのための訓練データです。文字「〇」と「×」を4×3画素で描いています。画素は0と1の2値です。

注1 本文では網をかけた画素を1、白部分を0としています。

注2 数値化されたデータは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「4.xlsx」の中の「Data」タブに収められています。

§ B

訓練データ(2)

5章の例題で作成するニューラルネットのための訓練データです。手書き数字1、2を9×9画素で写しています。ここでは、データ内容が見やすいように、その数字パターンを拡大して表示します。

注 本文では網が濃い画素を256に近い値、薄い部分を0に近い値としています。

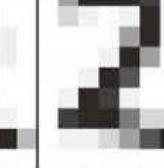
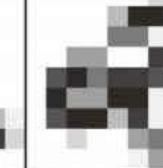
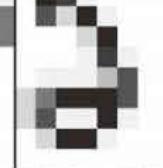
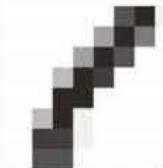
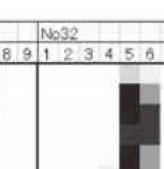
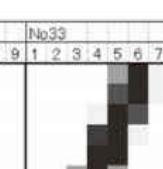
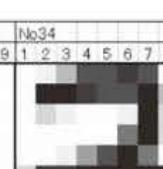
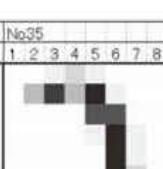
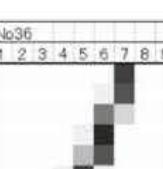
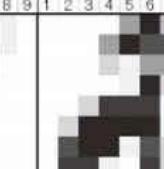
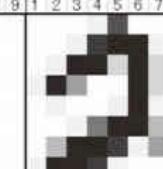
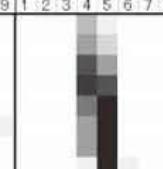
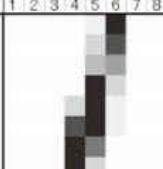
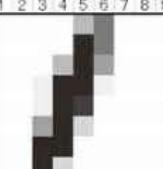
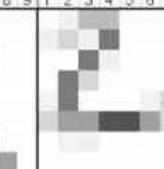
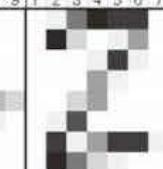
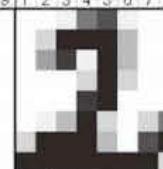
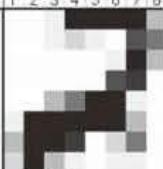
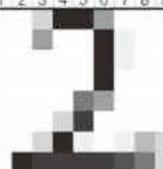
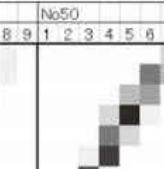
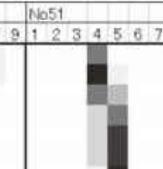
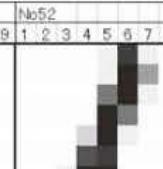
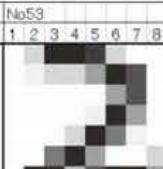
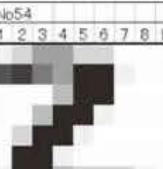
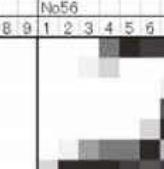
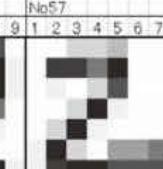
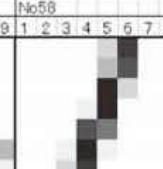
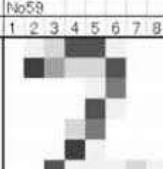
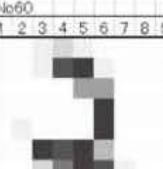
注 数値化されたデータは、ダウンロードサイト(→卷頭「Excelサンプルファイルのダウンロードについて」)に掲載されたファイル「5.xlsx」の中の「Data」タブに収められています。

	No1	No2	No3	No4	No5	No6
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 数字 画像						
正解	1	1	1	1	2	2

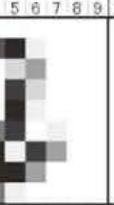
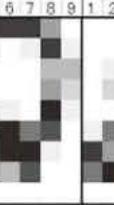
	No7	No8	No9	No10	No11	No12
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 数字 画像						
正解	2	2	1	1	1	1

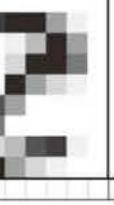
	No13	No14	No15	No16	No17	No18
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 数字 画像						
正解	2	1	2	2	1	1

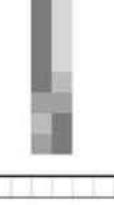
	No19	No20	No21	No22	No23	No24
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 数字 画像						
正解	1	2	1	1	2	1

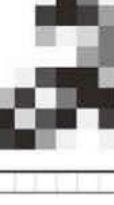
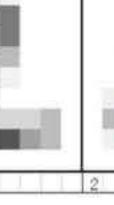
	No25	No26	No27	No28	No29	No30
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	2	2	2	2	1	1
	No31	No32	No33	No34	No35	No36
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	1	1	1	2	2	1
	No37	No38	No39	No40	No41	No42
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	1	2	2	1	1	1
	No43	No44	No45	No46	No47	No48
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	2	2	2	2	2	2
	No49	No50	No51	No52	No53	No54
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	1	1	1	1	2	2
	No55	No56	No57	No58	No59	No60
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	2	2	2	1	2	2

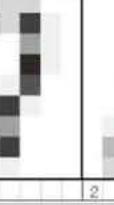
	No61	No62	No63	No64	No65	No66
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	1	2	2	2	1	2

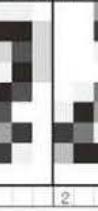
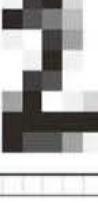
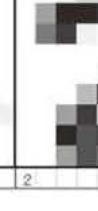
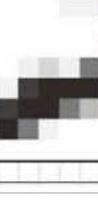
	No67	No68	No69	No70	No71	No72
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	1	2	1	2	2	2

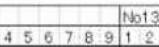
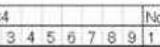
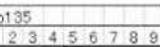
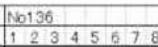
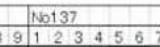
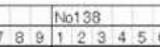
	No73	No74	No75	No76	No77	No78
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	2	2	2	2	2	1

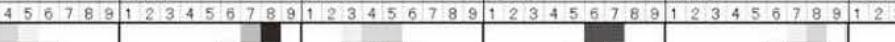
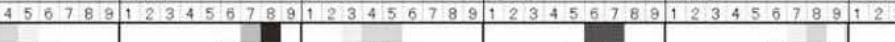
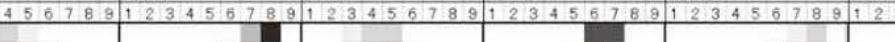
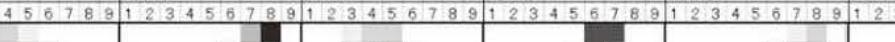
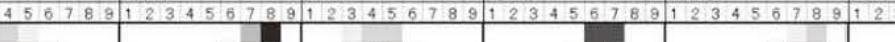
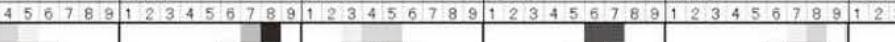
	No79	No80	No81	No82	No83	No84
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	1	2	1	2	1	1

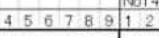
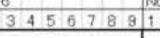
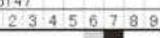
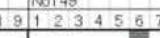
	No85	No86	No87	No88	No89	No90
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	2	2	1	1	2	2

	No91	No92	No93	No94	No95	No96
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	2	1	2	2	2	2

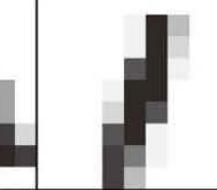
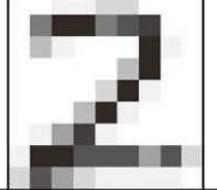
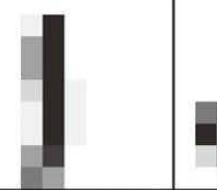
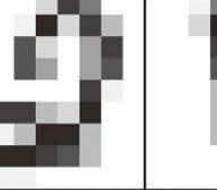
	No97	No98	No99	No100	No101	No102
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	1	2	2	1	2	1
	No103	No104	No105	No106	No107	No108
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	2	1	1	2	2	1
	No109	No110	No111	No112	No113	No114
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	1	1	2	2	2	1
	No115	No116	No117	No118	No119	No120
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	1	2	1	1	2	1
	No121	No122	No123	No124	No125	No126
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	1	2	2	2	2	2
	No127	No128	No129	No130	No131	No132
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	2	2	2	1	1	1

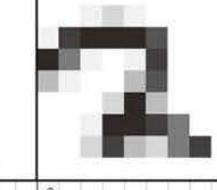
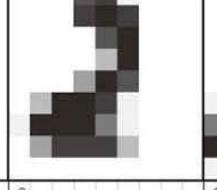
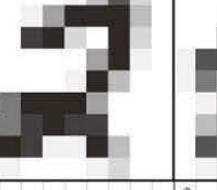
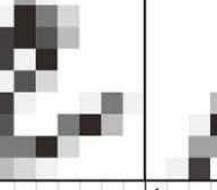
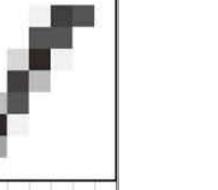
	No133	No134	No135	No136	No137	No138
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	1	2	2	1	2	2

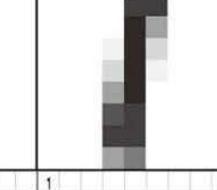
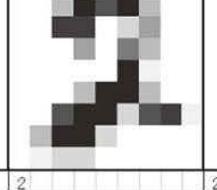
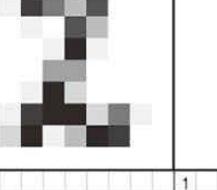
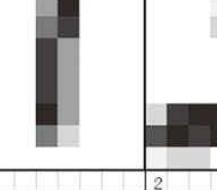
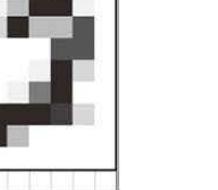
	No139	No140	No141	No142	No143	No144
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	2	1	2	1	2	1

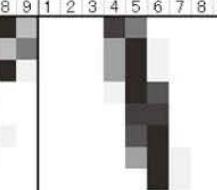
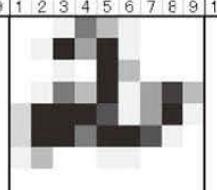
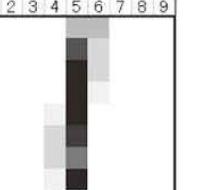
	No145	No146	No147	No148	No149	No150
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	2	2	1	1	1	2

	No163	No164	No165	No166	No167	No168
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
数字 画像						
正解	1	2	2	1	1	1

	No169	No170	No171	No172	No173	No174
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9						
正解	2	1	2	1	2	1

	No175	No176	No177	No178	No179	No180
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9						
正解	2	2	2	2	2	1

	No181	No182	No183	No184	No185	No186
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9						
正解	1	1	2	2	1	2

	No187	No188	No189	No190
	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9				
正解	2	1	2	1



ソルバーのインストール法

本書の計算の強力な助手は、Excelに備わっているアドインのひとつ「ソルバー」です。このアドインによって、高度な数学を用いることなく、畳み込みニューラルネットワークのしくみを数値的に理解できるのです。

ところで、新しいパソコンの場合、ソルバーがインストールされていない場合があります。それは「データ」タブに「ソルバー」メニューがあるかどうかで確かめられます。



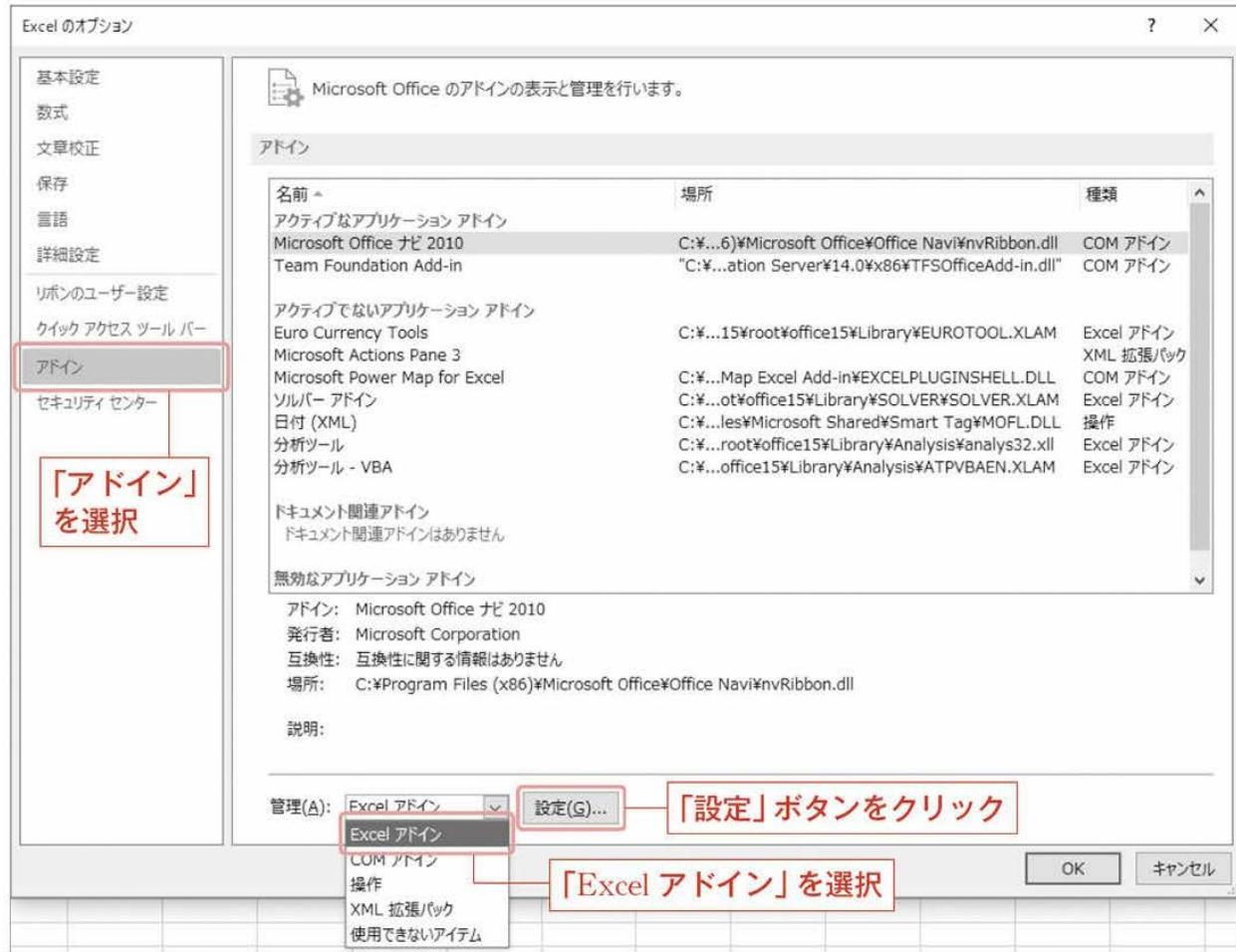
「ソルバー」のメニューがない場合には、インストール作業をする必要があります。ステップを追って調べてみましょう。

注 Excel2013、2016の場合について調べます。

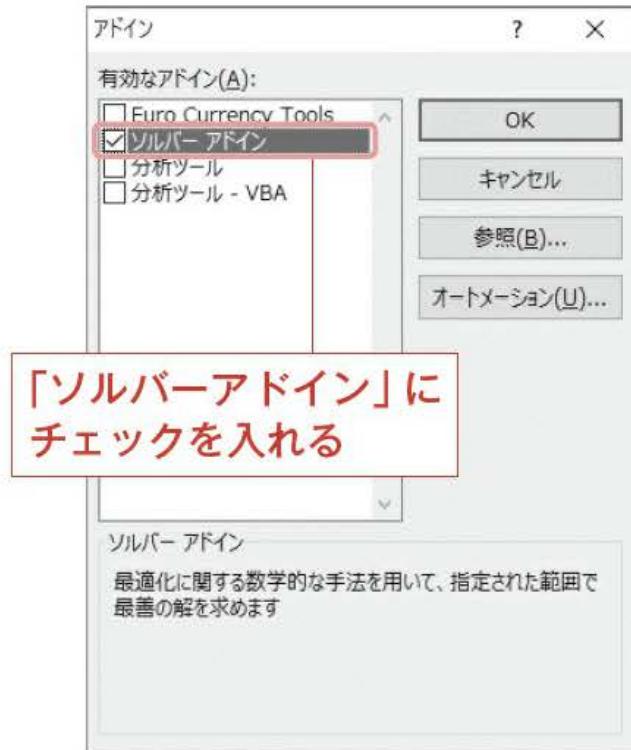
- ① 「ファイル」タブの「オプション」メニューをクリックします（次図）。すると、次の②のボックスが表示されます。



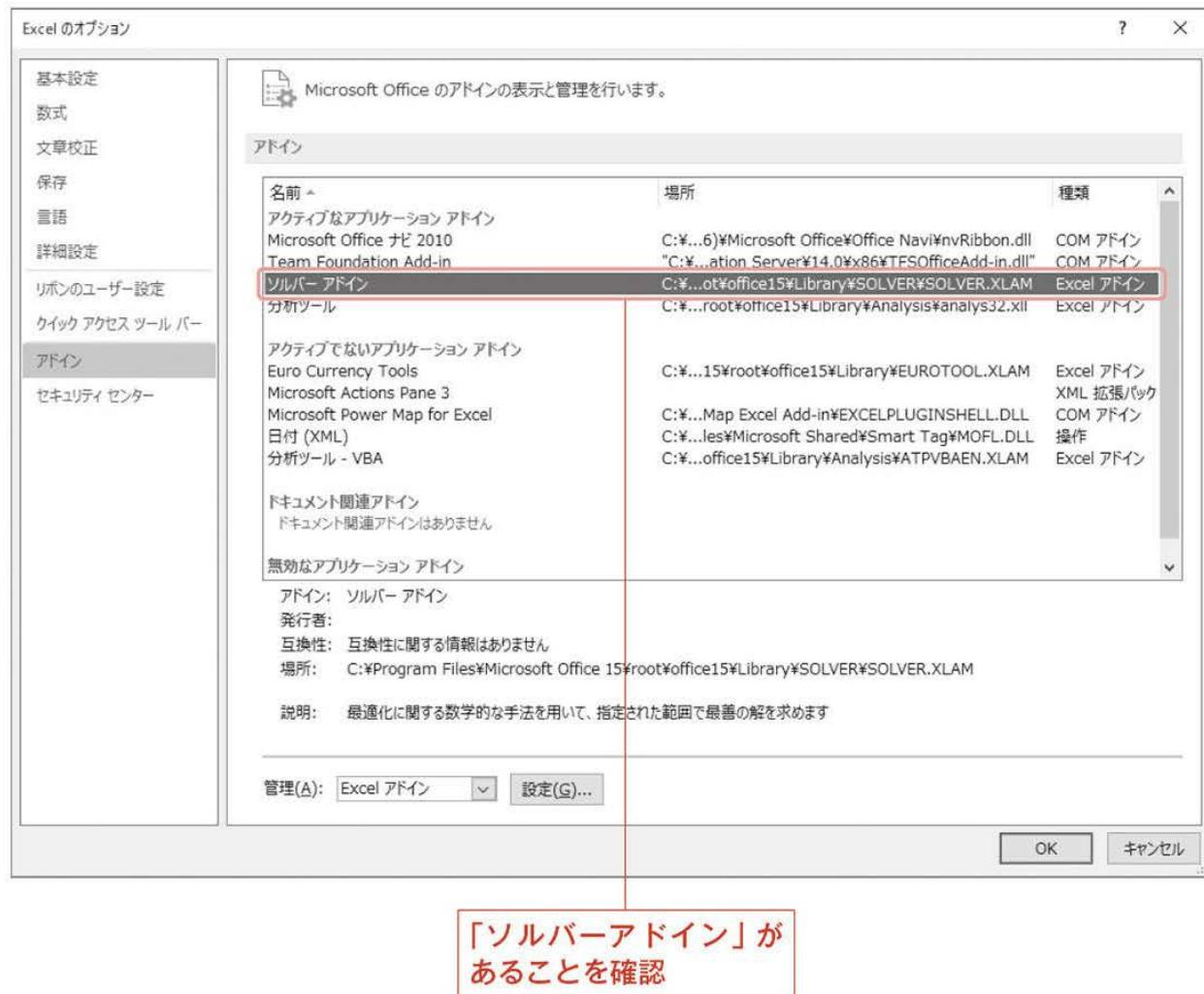
- ② 「Excelのオプション」ボックスが開かれるので、左枠の中の「アドイン」を選択します。さらに、得られたボックスの中の下にある、「Excelアドイン」を選択し、「設定」ボタンをクリックします。



③「アドイン」ボックスが開かれるので、「ソルバーアドイン」にチェックを入れ、「OK」ボタンをクリックします。



- ④ インストール作業が進められます。正しくインストールされたことは②のボックスが
次のようになっていることで確かめられます。



§ D

パターンの類似度を数式表現

畳み込みニューラルネットワーク(5章§4)では、次の定理が利用されました。

4×4画素からなる2つの数の並び A 、 F が次図のように与えられているとき、 A 、 F の類似度は次のように求められる。

$$\text{類似度} = w_{11}x_{11} + w_{12}x_{12} + w_{13}x_{13} + \dots + w_{44}x_{44} \dots (1)$$

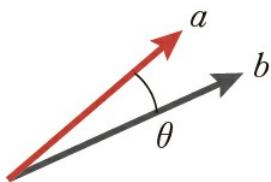
A	x_{11}	x_{12}	x_{13}	x_{14}
	x_{21}	x_{22}	x_{23}	x_{24}
	x_{31}	x_{32}	x_{33}	x_{34}
	x_{41}	x_{42}	x_{43}	x_{44}

F	w_{11}	w_{12}	w_{13}	w_{14}
	w_{21}	w_{22}	w_{23}	w_{24}
	w_{31}	w_{32}	w_{33}	w_{34}
	w_{41}	w_{42}	w_{43}	w_{44}

この定理は、ベクトルの性質を利用して説明できます。

大きさを固定した2つのベクトル a 、 b が似ているとき、その内積 $a \cdot b$ は大きくなるという特徴があります(次図)。

$$a \cdot b = |a| \parallel b \parallel \cos\theta \quad (\theta \text{は2つのベクトルのなす角})$$



平面のイメージで考えると、2つのベクトルの内積は、それらの矢の長さをなす角の余弦をかけたもの。角が0に近いほど余弦は大きい値をとる。すなわち、似ているときには値が大きくなると考えられる。

この内積の性質を利用するため、 A 、 F を次のようにベクトルとみなします。

$$A = (x_{11}, x_{12}, x_{13}, x_{14}, x_{21}, x_{22}, x_{23}, \dots, x_{44})$$

$$F = (w_{11}, w_{12}, w_{13}, w_{14}, w_{21}, w_{22}, w_{23}, \dots, w_{44})$$

すると、2つのベクトルの内積 $A \cdot F$ は上記の式(1)の右辺に一致します。すなわち、式(1)を類似度と解釈できるのです。

索引

記号

$\$ \quad \sigma(x) \quad \theta \quad \theta^{F1} \quad \theta^{Fk} \quad \theta^{Hj} \quad \theta^{Ok} \quad \theta^{On}$

英字

activation function $a_{11}^{F1} \quad a_{ij}^{Fk} \quad a^{Hj} \quad \text{AI} \quad a^{Ok} \quad a^{On} \quad \text{Artificial Intelligence}$
bias BMI CNN $e \quad e^x \quad \text{Excel} \quad \text{ReLUを用いた最適化} \quad \text{回帰分析}$
画像データの入力 仮のパラメーターの設定 参照形式 指数関数の計算 出力層の計算
ソルバー ソルバーで最適化 ソルバーのインストール 畳み込み層の計算 データ分析
手書き数字を入力 入力層のニューロン ニューロンの出力 ニューロンの働きを再現
プーリング層の計算 平方誤差の計算 目的関数の計算 目的関数の最小化 文字の判定
EXP $\text{EXP}(x)$ Filter Fk Hidden layer Hinton IF ILSVRC Input layer
INTERCEPT L2プーリング MAX max pooling MAXプーリング
MAXプーリング層の公式化 MNIST NIST Output layer $p_{ij}^{Fk} \quad Q \quad Q_T \quad \text{RAND}$
Rectified Linear Unit ReLU SLOPE SUM SUMPRODUCT SUMXMY2 t_1
 t_2 threshold training data transfer function weight $w_{11}^{F1} \quad w_{ij}^{Fk} \quad w_i^{Hj}$
 $w_j^{Ok} \quad w_{k-ij}^{On} \quad x_i \quad x_{ij} \quad y_{11}^{F1} \quad y_{ij}^{Fk} \quad y_j \quad z_k$

ア行

アナログ信号 アメリカ国立標準技術研究所 アラン・チューリング アルファ碁 閾値
運搬係 エキスパートシステム オーバーフロー 重み 重み付きの和 重み付きの和の式

力行

カーネル 回帰係数 回帰直線 回帰分析 回帰分析専用ツール 回帰方程式

階層型ニューラルネットワーク 外挿法 鍵刺激 学習 学習データ 確信度 隠れ層
隠れ層 y 画像 仮想的なニューロン 画像の数値化 活性化関数 活性度
仮の閾値 仮の重み 仮のパラメーター 関数 $u(x)$ 含有率 強化学習
教師あり学習 教師データ 教師なし学習 極小解 クロスエントロピー 訓練データ
形式ニューロン 結合荷重 結合負荷 検知係 検知するパターンの大きさ 興奮する
興奮度 誤差逆伝播法 誤差の総和 誤差の評価 コンピューター科学の父
コンボリューション層

サ行

最小2乗法 最小値 最大プーリング 最適化 最適化問題 細胞体 最尤推定法
差の平方和 産業用ロボット 散布図 閾値 識別 軸索 シグモイド関数
シグモイドニューロン シグモイドニューロンの一般化 視神経システム 指数関数 実測値
自動運転 シナプス 縮約法 樹状突起 出力 出力信号 出力信号 y
出力信号あり 出力信号なし 出力層 出力層の重み 出力表 出力変数名
条件付き書き式 ジョン・マッカーシー 白黒2値画像 神経細胞 神経細胞の発火 信号
人工知能 人工知能の父 人工ニューロン 信号和 深層学習 水晶体 推定
推定式 数値化 ステップ関数 正解 正解変数 積和 絶対参照 切片
説明変数 セル参照 線形の単回帰分析 線形和 全結合 相対参照 ソルバー
ソルバーのインストール

タ行

ダートマス会議 ダートマス大学 置み込み 置み込み層 効果 出力表
置み込みニューラルネットワーク 隠れ層の役割 出力層 出力層の役割 入力層の役割
単回帰分析 知覚細胞 地区 知能 中間層 ディープラーニング 定数
データの数値化 手書き数字 手書き文字 デジタル信号 デミス・ハサビス 伝達関数

動物のニューロン 特徴抽出 特徴パターン 特徴パターンの類似度 トロント大学

ナ行

ニューラルネットワーク テスト 問題点 入力 入力x 入力信号 入力層 入力の組
入力の線形和 ニューロン イメージ 活性度 出力値 働き 働きを再現 発火
略式図 ネイピア数 ネットワークの連携 ネットワーク自らが決定する 濃縮 濃縮情報

ハ行

パーセプトロンモデル バイアス パターンの類似度 発火式 判定 発火あり
発火なし パラメーター 組 決定法 初期値 判定係 フィルター
フィルターとの類似度 フィルターの成分 プーリング層 プーリング表 プーリング法
複合参照 負の数 不連続関数 平均プーリング 平方誤差 平方誤差の式 ベクトル
変数 変数名 補外法 補間法

マ行・ヤ行・ラ行

自ら学習 網膜 目的関数 目的関数 Q_T 目的変数 モデルの最適化
モデルの有効性 矢の太さ 予測値 ランプ関数 類似性 類似度

Profile

涌井 良幸 (わくい よしゆき)

1950年、東京都生まれ。東京教育大学(現・筑波大学)数学科を卒業後、千葉県立高等学校の教職に就く。

教職退職後はライターとして著作活動に専念。

涌井 貞美 (わくい さだみ)

1952年、東京生まれ。東京大学理学系研究科修士課程修了後、富士通、神奈川県立高等学校教員を経て、サイエンスライターとして独立。

●装丁

小野貴司

●本文

BUCH⁺

Profile

涌井 良幸 (わくい よしゆき)

1950年、東京都生まれ。東京教育大学（現・筑波大学）数学科を卒業後、千葉県立高等学校の教職に就く。
教職退職後はライターとして著作活動に専念。

涌井 貞美 (わくい さだみ)

1952年、東京生まれ。東京大学理学系研究科修士課程修了後、富士通、神奈川県立高等学校教員を経て、サイエンスライターとして独立。

本書へのご意見、ご感想は、技術評論社ホームページ (<http://gihyo.jp/>) または以下の宛先へ、書面にてお受けしております。電話でのお問い合わせにはお答えいたしかねますので、あらかじめご了承ください。

〒162-0846 東京都新宿区市谷左内町21-13
株式会社技術評論社 書籍編集部
『Excelでわかるディープラーニング超入門』係
FAX: 03-3267-2271

●装丁：小野貴司

●本文：BUCH+

Excelでわかる ディープラーニング超入門

2018年1月5日 初版 第1刷発行

著 者 涌井良幸・涌井貞美
発 行 者 片岡 嶽
発 行 所 株式会社技術評論社
東京都新宿区市谷左内町21-13
電話 03-3513-6150 販売促進部
03-3267-2270 書籍編集部
印刷／製本 株式会社加藤文明社

定価はカバーに表示しております。

本の一部または全部を著作権の定める範囲を超えて複写、複製、転載、テープ化、あるいはファイルに落とすことを禁じます。
造本には細心の注意を払っておりますが、万一、乱丁（ページの乱れ）や落丁（ページの抜け）がございましたら、小社販売促進部までお送りください。
送料小社負担にてお取り替えいたします。

電子版書籍について

本書は紙の書籍『Excelでわかる ディープラーニング超入門』(ISBN978-4-7741-9474-5)を電子書籍化したものです。紙書籍とは一部レイアウトやデザインが異なります。本書の更新履歴や補足情報は[技術評論社ウェブサイト](#)をご参照ください。

本書の一部または全部を著作権法の定める範囲を超えて複写、複製、転載、テープ化、ファイルに落とすことを禁じます。造本には細心の注意を払っておりますが、万一、ページの乱れやページの抜け等がございましたら、小社クロスメディア事業部までお知らせください。

電子版奥付

書名

Excelでわかる

ディープラーニング超入門

電子版発行日

2017年12月23日 初版 第1刷発行

著 者

涌井良幸・涌井貞美

発行者

片岡 巖

発行所

株式会社技術評論社

東京都新宿区市谷左内町21-13

電話

03-3113-6150 販売促進部

03-3113-6180 クロスメディア事業部

電子版製本

株式会社リ・ポジション

©2017 涌井良幸、涌井貞美

ISBN978-4-7741-9528-5