

ニューラルネットワーク作成による数学学習の動機付け

日本大学・生物資源科学部/大阪市立大学・数学研究所 濱田 龍義

Tatsuyoshi Hamada, College of Bioresource Sciences, Nihon University/OCAMI

1 はじめに

2016年から日本大学生物資源科学部で教鞭をとることとなった。生物資源科学は生命、環境、資源、食品などを中心とする研究分野である。神奈川県藤沢キャンパスにおいて12学科、約7000人の学生が学んでいる。そのうち1年生は約1500人ほどである。1952年に農獣医学部として発足し、1996年に生物資源科学部に改組されたが、これまでに多数の生物資源科学、農学、獣医学分野の専門家を輩出してきた。

現在、生物資源科学部の一般教養において数学の講義である「線形代数」、「解析学」、計算機実習を含む講義として「情報科学」、「ネットワーク入門」等を担当している。言うまでもなく、線形代数学と解析学は、すべての科学の基礎となる分野であり、統計学、経済学等を理解するために必要な基礎的分野と言って良い。一般に理系分野においては線形代数学を4単位、解析学を4単位設置して、必修科目としている場合が多い。一方で、私学の生物資源科学や農学分野においては、線形代数学や解析学は必修化されていない場合が多く、設置されている講義も前期もしくは後期のみの2単位程度であることが多いようである。

赴任した当初は「線形代数」の履修率が1割を切っている状況であったが、同僚の先生方の協力を仰ぐことによって、現在では「線形代数」については全体の3割弱、「解析学」については全体の2割強の学生が履修するようになった。それでも、他の理系学部と比べて履修者が少ない状況は変わっていない。

一方で、生物資源科学においては、どの分野を選択しても、統計学やデータサイエンスを多用する。特に主成分分析や多次元尺度法を用いる分野が多いが、“行列”や“固有値”という概念を講義で学ばずに取り組むため、ブラックボックスとしてソフトウェアの使い方だけを覚えて利用している学生も多い。そこで、講義「ネットワーク入門」において、学生の興味も高い人工知能に関連した「ニューラルネットワークによる機械学習」をテーマとして選択し、学部レベルの数学への導入として位置づけた。

2 講義「ネットワーク入門」について

「ネットワーク入門」は一般教養の半期1単位の実習科目として開講している。“ネットワーク”という言葉から最初に連想されるのはTCP/IPネットワークであると思う。しかし、大学のコンピュータ環境に慣れていない1年生には負荷が高いこと、コンピュータ・サイエンスの基礎についての予備知識も期待できないことから断念した。

どのようなテーマで実習を進めるかを検討していたときに会ったのが、斎藤 康毅氏による入門書 [1] である。この書籍は最新の技術を解説しているわけではない。しかし、プログラミング言語 Python を用いて数値計算ライブラリ NumPy¹や2次元グラフィックスライブラリ matplotlib²を用いて実装しながら、ディープラーニングを理解する上での要点を丁寧に解説している。ただ、書籍の内容のすべてを半年の実習で取り扱うことは質、量ともに厳しい。そこで、ニューラルネットワークの実装、学習を通して、線形代数学や解析学に触れてもらうことを目標として、以下のようなシラバスを設計した。

1. ガイダンス
2. コンピュータの使い方
3. 簡単な論理回路
4. Python 基本文法
5. 行列の基本
6. NumPy と matplotlib
7. パーセプトロン
8. ニューラルネットワーク
9. 活性化関数
10. 手書き数字認識
11. ニューラルネットワークの学習
12. 損失関数
13. 勾配法
14. 誤差逆伝播法
15. これまでの学習内容の確認

なお、学生向けの参考文献として大関真之氏による入門書を紹介した [2]。白雪姫のお后様と魔法の鏡との会話によって進められる機械学習の解説は、ユーモアを交えて軽妙であり、高校生でも読めるわかりやすいものとなっている。数式に慣れていない学習者にも推薦できる。

コンピュータ実習室での演習には Anaconda³を利用した。Anaconda はデータサイエンスや機械学習などに特化した Python のパッケージである。演習に用いる NumPy

¹<http://www.numpy.org/>

²<https://matplotlib.org/>

³<https://www.anaconda.com/>

と matplotlib も同梱されている。また, Spyder⁴と呼ばれる Python 用の簡易開発環境も同梱されている。Anaconda は Windows だけでなく MacOS にも対応しているオープンソースソフトウェアである。プログラミング環境の選択には, 自宅で学習環境を構築することができることも重要視している。Windows 限定となるが, Anaconda と似たパッケージとして WinPython⁵というプロジェクトも存在する。こちらは USB メモリー等にインストールができるので利便性が高い。

最初のプログラミング環境は Anaconda Prompt から導入を行った。Python 実行環境を設定された cmd.exe と考えて良い。基本的に演習は Windows のメモ帳でプログラミングを行うが, 慣れてくれば Spyder を紹介している。Spyder を使う利点は, エラーメッセージの指示箇所がわかりやすいことと, 変数エクプローラー等で変数の値を確認できることなどである。メモ帳に慣れた学生は, あえて Spyder を使わない場合も多いようだ。

その他に講義資料の作成に L^AT_EX 環境として Luatex-j⁶, もしくは ptex2pdf⁷を用いた。ネットワーク図等の図形描画に PGF/TikZ⁸, 論理回路の描画に circuitikz⁹を用いている。グラフの描画は matplotlib, 場合によっては GeoGebra¹⁰も利用した。この講究録についても PGF/TikZ, GeoGebra を用いている。また, 課題提出確認は, 大学で契約している Google 社の G Suite for Education に含まれている Google Classroom¹¹を活用した。

3 ニューラルネットワークを理解するための準備

生物資源学部の入試範囲は高校の数学 I, II, A, B (確率分布と統計的な推測を除く)なので, これらの科目は履修していると仮定する。ただし, 農業高校出身者は数学 I のみを履修している場合が多いので, 可能な限り個別に対応する必要がある。

3.1 論理演算

ニューラルネットワークを理解する上で, 論理演算の基礎知識は欠かせない。また, 最低限必要な数学はベクトルや行列の演算, 合成関数, 確率, 多変数関数の微分積分, ベクトル解析などが考えられる。基本的には行列の演算と関数の理解に, 時間を費やすことになる。

論理演算, 特に 論理和, 論理積, 否定, 否定論理積, 排他的論理和などの確認を行っている。数学 I の「命題と論証」で基礎的なことは学んでいるはずだが, 初めて学ぶ排他的論理和については, 若干戸惑う学生もいるようである。

⁴<https://www.spyder-ide.org/>

⁵<https://winpython.github.io/>

⁶<https://ja.osdn.net/projects/luatex-ja/>

⁷<https://www.preining.info/blog/software-projects/ptex2pdf/>

⁸<https://www.ctan.org/pkg/pgf>

⁹<https://www.ctan.org/pkg/circuitikz>

¹⁰<https://www.geogebra.org/>

¹¹<https://edu.google.com/intl/ja/k-12-solutions/classroom/>

3.2 パーセプトロン

論理演算について確認を行った後にパーセプトロンについて取り上げた。パーセプトロンは 1957 年に Frank Rosenblatt によって考案された神経回路網を模したモデルである。まずは単層パーセプトロンと出力の評価式について学ぶ。Python でパーセプトロンを実装し、論理演算を構成することによって重みベクトル $\mathbf{w} = (w_1, w_2)$ とバイアス b の意味を理解することを目指す。

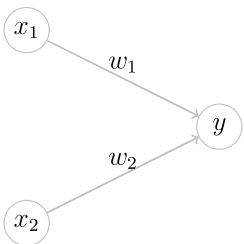


図 1: 2 入力単層パーセプトロン

以下は上記 2 入力単層パーセプトロンの評価式である。

$$y = \begin{cases} 0 & (w_1x_1 + w_2x_2 + b \leq 0) \\ 1 & (w_1x_1 + w_2x_2 + b > 0) \end{cases}$$

排他的論理和の実現に伴い、パーセプトロンの限界について学び、多層パーセプトロンへと導入する部分は大変興味深い。[1] の魅力の 1 つでもある。ここで、評価式の重みベクトル $\mathbf{w}$ は線形分離のための境界超平面（2 入力の場合は直線，3 入力の場合は平面）に対する法ベクトルとして定義される。

数学 III を履修していない学生にとっては、活性化函数の取扱がつまづきやすい部分かもしれない。パーセプトロンでは階段函数を活性化函数として用いていたのだが、これを非線形なシグモイド函数や ReLU 函数で置き換えていくことが重要である。数学 III で学ぶ合成函数の考え方を理解していれば、特に難しいものではないが、多変数函数についても経験がないことを考えると、戸惑う学生がいても不思議ではない。

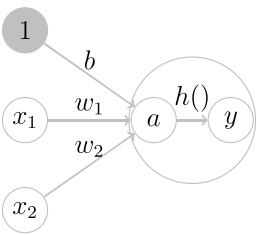


図 2: 活性化函数 $h()$

以下の評価式は単層パーセプトロンと活性化函数の関係をベクトルの内積で表したも

のである．活性化関数 $h()$ は階段関数を用いている．

$$\begin{aligned}\mathbf{w} &= (w_1, w_2) \\ \mathbf{x} &= (x_1, x_2) \\ y &= h(\mathbf{w} \cdot \mathbf{x} + b) \\ h(x) &= \begin{cases} 0 & (x \leq 0) \\ 1 & (x > 0) \end{cases}\end{aligned}$$

単層パーセプトロンの評価式をベクトルの内積として捉えることは，この後，行列演算に結びつける上で重要な部分である．しかし，線形代数を履修していない学生が多い状況下では難しい部分である．対策として，ベクトルと行列の積について手計算による演習を繰り返し行っている．NumPy による計算も行うが，簡単な 2×2 もしくは 2×3 型や 3×2 型の行列について，前もって手計算による習熟が重要である．

3.3 ニューラルネットワーク

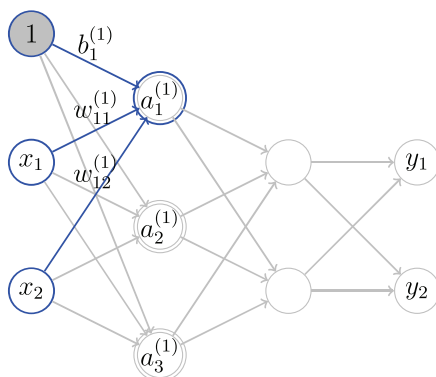


図 3: 3 層ニューラルネットワークにおける第 1 層への入力

ニューラルネットワークを記述する上で行列は重要な道具となる．NumPy は高速な行列演算に対応している．NumPy を用いて，手書き文字データセット MNIST¹² を分類するニューラルネットワークの実装を行う．ニューラルネットワークの第 1 層への入力を行列で記述すると以下のような評価式となる．ニューラルネットワークは「アフィン写像」と「活性化関数による合成関数」の繰り返しで構成される．

$$(a_1^{(1)} \quad a_2^{(1)} \quad a_3^{(1)}) = (x_1 \quad x_2) \begin{pmatrix} w_{11}^{(1)} & w_{21}^{(1)} & w_{31}^{(1)} \\ w_{12}^{(1)} & w_{22}^{(1)} & w_{32}^{(1)} \end{pmatrix} + (b_1^{(1)} \quad b_2^{(1)} \quad b_3^{(1)})$$

¹²<http://yann.lecun.com/exdb/mnist/>

3.4 活性化関数

活性化関数として、よく知られているものは次の関数である。

- 階段関数: $h(x) = \begin{cases} 0 & (x \leq 0) \\ 1 & (x > 0) \end{cases}$
- シグモイド関数: $h(x) = \frac{1}{1 + e^{-x}}$
- ReLU 関数: $h(x) = \begin{cases} 0 & (x \leq 0) \\ x & (x > 0) \end{cases}$
- ソフトマックス関数: $h_k(x) = \frac{e^{x_k}}{\sum_{i=1}^n e^{x_i}}$

matplotlib は画像の表示や関数のグラフ作成を行うことができるので、画像を行列もしくはベクトルとして扱うことを確認したり、グラフの作成を演習として課している。数学 III を履修していない学生も多いので、ネイピア数 e についても説明が必要となる。適切な導入を行えば、シグモイド関数のグラフの作成については問題ないようであった。しかし、応用として正規分布の確率密度関数によるグラフの描画についての課題は、ほとんどの学生が混乱していたようである。統計を学ぶ上で重要な関数であるため、さらなる習熟が必要な部分である。

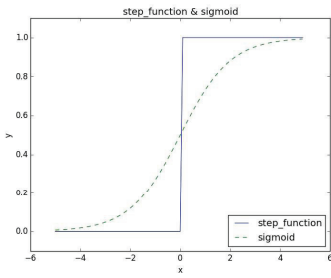


図 4: 階段関数とシグモイド関数

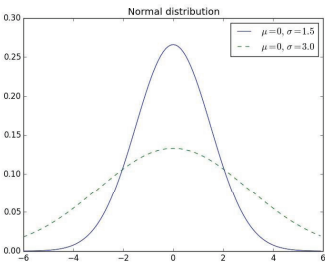


図 5: 正規分布の確率密度関数

また、応用としてツールキット matplotlib mplot3d¹³を用いることで、2変数関数のグラフを描画して対話的に観察することも可能である。ソフトマックス関数は MNIST 等の分類問題の出力層の活性化関数として用いられる。演習では単純化して2入力のソフトマックス関数のグラフである曲面を描画し、2変数関数に関する導入として用いた。

¹³https://matplotlib.org/mpl_toolkits/mplot3d/

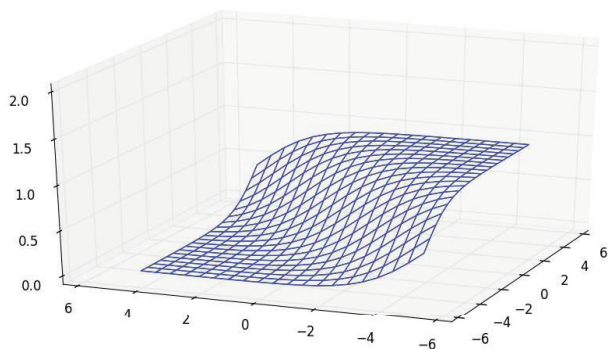


図 6: 2 入力ソフトマックス関数

3.5 損失関数と勾配法

ニューラルネットワークの性能を評価するには損失関数を用いる。 y_k はニューラルネットワークの出力値, t_k は教師データの値である。

- 2乗和誤差: $E = \frac{1}{2} \sum_k (y_k - t_k)^2$
- 交差エントロピー誤差: $E = - \sum_k t_k \log y_k$

また、勾配法は最適なパラメータを探すための基礎的な方法である。GeoGebraを利用することで、始点を変更したり、曲面を描く函数を対話的に変更したりすることを誰でも手軽に行うことができる。ニューラルネットワークの学習について学ぶ上で損失

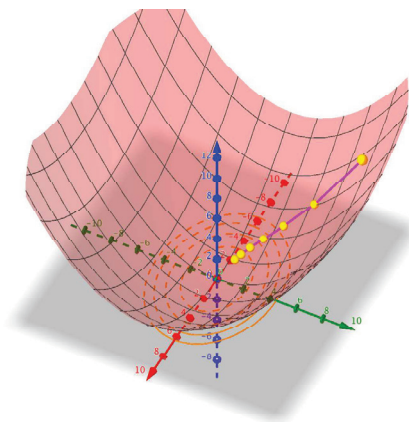


図 7: 勾配法の解説図

函数と勾配法，誤差逆伝播法は重要な部分であるが，半年間の講義では深く取り上げることができなかった．今後，検討を必要とする部分である．

4 授業に関するアンケート

以下は，学部で行われている授業アンケートの設問と回答である．

1. この授業科目を1週（回）受けるに当たり，授業時間以外で学習（予習，復習，課題等）にどのくらい取り組みましたか？
 (1) 3時間以上 (2) 2～3時間 (3) 1～2時間 (4) 1時間未満 (5) していない
2. この授業科目に関し，授業時間外（授業終了直後を含む）に，担当教員に対し質問等をしましたか？
 (1) 7回以上 (2) 5回程度 (3) 3回程度 (4) 1回程度 (5) 全くしていない
3. この授業科目を受けるに当たり，図書館をどのような目的で利用しましたか？
 (1) 図書や文献資料を閲覧・借りた (2) 図書館で予習・復習をした (3) レファレンスサービスを利用した (4) 相互利用サービスを利用した (5) 全く利用しなかった
4. この授業科目に関し，授業時間外に，学生間で共に学習しましたか？
 (1) 何度もした (2) 数回した (3) 1回した (4) 休み時間に話す程度 (5) 全くしていない
5. 授業時間外の学習（予習，復習等）の方法について，担当教員から具体的（シラバスに明記を含む）に示されましたか？
 (1) 強くそう思う (2) そう思う (3) どちらとも言えない (4) そう思わない (5) 全く思わない
6. 課題（レポート，小テスト等）に対し，担当教員から学生へのフィードバック（評価や講評等の開示）はありましたか？
 (1) 詳細な講評，模範解答が示され，評価が開示された (2) 講義中に講評や解説，個人への評価の開示があった (3) 評価の開示があったのみ (4) ほとんどなかった (5) 課題は課されなかった
7. この授業科目は，シラバス（授業計画）通り行われていたと思いますか？
 (1) 強くそう思う (2) そう思う (3) どちらとも言えない (4) そう思わない (5) 全く思わない
8. 担当教員は，授業に際し十分な準備を行い，意欲的に授業を進めていたと思いますか？
 (1) 強くそう思う (2) そう思う (3) どちらとも言えない (4) そう思わない (5) 全く思わない

9. 担当教員は、学生の理解度や反応を考慮して授業を進めていたと思いますか？
 (1) 強くそう思う (2) そう思う (3) どちらとも言えない (4) そう思わない (5) 全く
 そう思わない
10. この授業科目の難易度及び進行は適切でしたか？
 (1) 非常に適切であった (2) かなり適切であった (3) 難しく理解できない点があっ
 た (4) 難しく、進行も速いので理解できない点があった (5) 全体的に難しく不適
 切であった
11. この授業科目で学んだ内容をどのくらい理解できたかと思いますか？
 (1) 9割以上 (2) 8割程度 (3) 6割程度 (4) 4割程度 (5) 2割以下
12. この授業を5段階で評価してください。
 (1) とても良い (2) 良い (3) 普通 (4) 悪い (5) とても悪い

設問	(1)	(2)	(3)	(4)	(5)	記入ミス	回答者数	評価平均
1	0	1	6	15	3	1	25	2.20
2	3	0	9	4	10	0	26	2.31
3	1	4	1	0	20	0	26	1.69
4	3	5	3	8	7	0	26	2.58
5	4	12	6	3	1	0	26	3.58
6	7	10	8	0	0	1	25	3.96
7	3	16	6	1	0	0	26	3.81
8	5	16	4	1	0	0	26	3.96
9	1	9	11	1	4	0	26	3.08
10	1	4	7	10	4	0	26	2.54
11	0	1	6	11	8	0	26	2.00
12	3	3	18	1	1	0	26	3.23

アンケートの回答分布表には『「評価平均」とは、「無記入・記入ミス」を除く有効回答データの「5」から「1」に各点を与え、平均を算出したもの。』とある。また、はっきりとした数値はわからないが、全体平均（アンケートを行った講義全体の平均値）が各設問ごとに回答分布表にグラフ表示されている。

設問1については、アンケートを行っている学部の講義の平均が2弱ということで、全体に学生の学習時間の少なさが目立つ。

設問2の担当教員への質問については、全体平均が約1.5と、かなり低いことを考えると、検討している方だろうか。強く興味を持ち、頻繁に質問をする学生と、質問をするだけの準備ができていない学生との乖離が目立つ。

設問3が著しく低いのが、この設問の回答が平均値を取るにそぐわないものであることを考慮しても、学生による図書館の利用率が低いことが目立つ。図書館に適切なコンピュータ環境がないこと、適切な書籍が図書館に収録されていないこと等もあるが改善

が望まれる。現在、限られた予算で少しずつ必要な書籍を揃えつつある。今後、学部全体としても何らかの対策が必要な項目ではないかと考えられる。

設問4の授業時間外の学生間の共同学習についてもモチベーションの高い学生と、そうでない学生との乖離が目立つ。

設問9, 10については担当者として厳しく捉えないといけないかもしれない。学生の過半数が難しく進行が速いと感じているようである。また、半分以上理解できた学生も少ないようである。

5 まとめ

機械学習やディープラーニングは、学生にとっても注目の高い話題であり、学部レベルの数学学習の動機付けとして格好の題材と思われる。数学学習における、数学ソフトウェアによる教育支援は今後も増えてくると思われるが、手で計算することと、計算機で計算することの区別は重要であり、手で計算できることによって、計算機による結果を理解することができるということを、再確認する必要があるのではないだろうか。また、数学および数学教育に関わる人間にとっても、計算機がどのように動いているのかについて、基礎理論について学ぶことの重要性がますます増していると思われる。

ただ、一般に、数学と計算機の学習を同時に学生に要求することは、難しい場合が多いようである。当該講義については学習内容を絞り込むことを考えている。特に、プログラミングの初歩的なことについては、取り扱う量を増やしても、かえって学生の負荷が大きいように思われた。特定の目標を持って取り組むことは学生のモチベーションを維持する上でも重要であると思われる。非常に強い興味をもって課題に取り組んでくれた学生も多く、今後も引き続き改善しながら取り組んでいきたい。学生が数学に興味を持つような新しい題材についても、さらなる検討を進めたいと考えている。

参考文献

- [1] 「ゼロから作る Deep Learning ——Python で学ぶディープラーニングの理論と実装」, 斎藤康毅, 2016, O'Reilly Japan, Inc.
- [2] 「機械学習入門 ボルツマン機械学習から深層学習まで」, 大塚真之, 2016, オーム社.