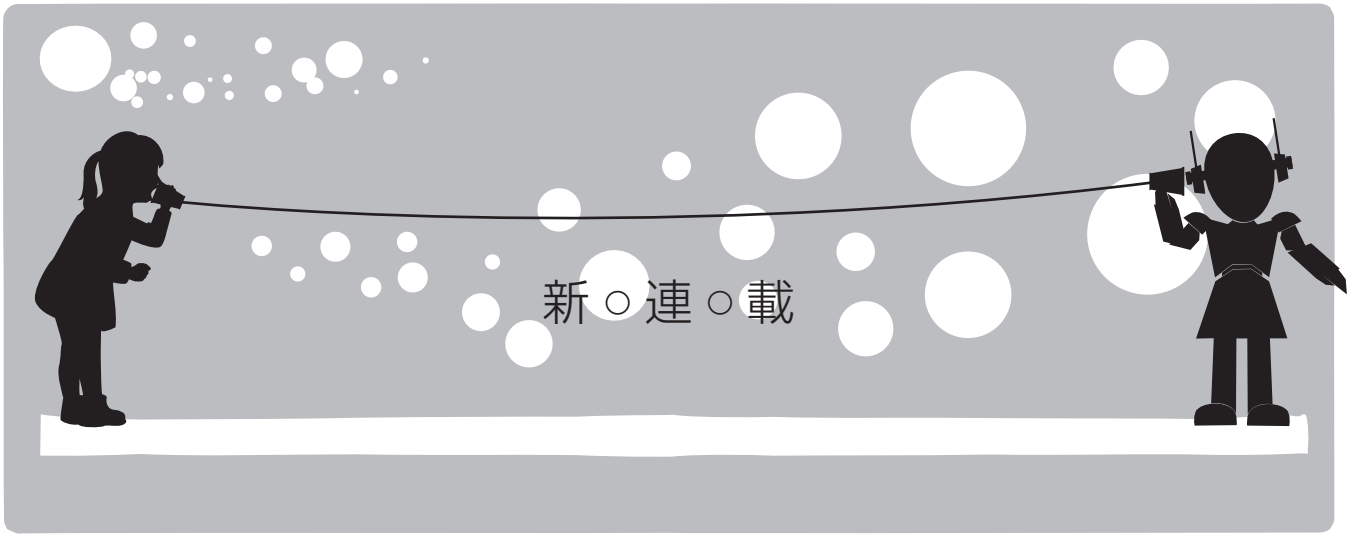


講座○機械学習超入門○全６回
開講にあたって

編集幹事 間下以大

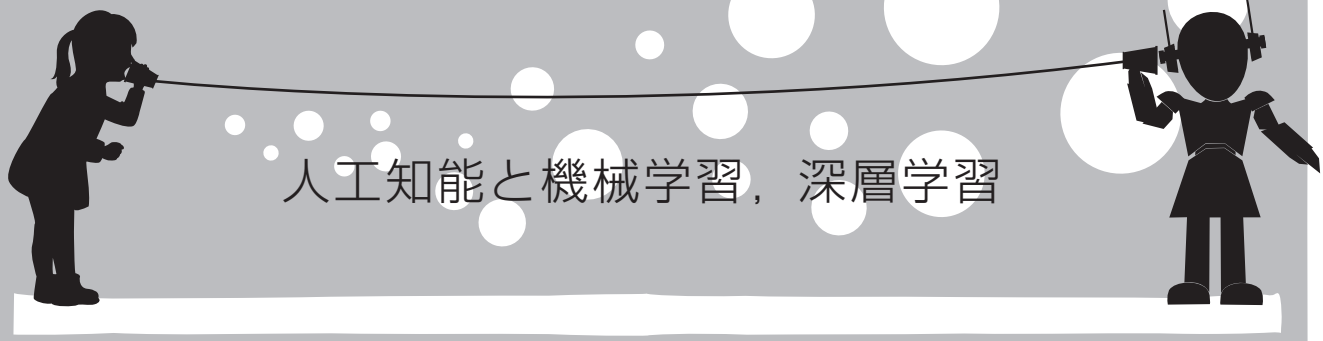
近年の深層学習（Deep Learning）と人工知能ブームは未だとどまる所を知らず，日々新たな技術が開発されています．また，アメリカや中国の情報系企業を中心に人材の争奪線が行われており，筆者が2017年7月に参加したCVPR2017では，GoogleやFacebookといった世界中の名だたる情報系企業が人材募集を行っていました．

このような状況の中で，これから研究や事業に機械学習を導入したいが，どのようにすべきかとお悩みの方は多いと思います．本講座はそのような方々を想定し，前半では人工知能と機械学習，近年の深層学習についての簡単な解説から始まり，機械学習の基本的な考え方を解説します．後半ではそれらを踏まえて深層学習を学び，利用するための考え方について説明する予定です．機械学習のさまざまなアルゴリズムの詳細は他の書籍におまかせし，機械学習の考え方を中心に解説を行う予定です．

本講座は私，間下以大編集幹事が執筆の機会を頂きました．本講座が皆様のお役に立てば幸いです．

予定目次○全６回

第１回○人工知能と機械学習，深層学習○○○間下以大○大阪大学
第２回○機械学習とは線を引くアルゴリズムのこと○○○間下以大○大阪大学
第３回○線を引くだけでは足りない場合もある○○○間下以大○大阪大学
第４回○特徴量の設計が腕の見せ所○○○間下以大○大阪大学
第５回○深層学習で何が変わったのか○○○間下以大○大阪大学
第６回○深層学習は万能ではない○○○間下以大○大阪大学



人工知能と機械学習，深層学習

正会員 間下以大†

1．まえがき

人工知能（Artificial Intelligence: AI）という言葉は人によっても意味するものに大きく幅があり，とても曖昧である．さらに近年の人工知能ブームで言葉が一人歩きしており，なんとなくのイメージだけでAIが語られることも多い．つまるところ胡散臭いわけだが，一方で将棋や囲碁でトッププロを破る[1)2)]など，革新的な技術としての実績も多数ある．

人工知能には強いAIと弱いAIという考え方がある．あるいは，汎用人工知能（Artificial General Intelligence: AGI）と呼ばれる場合もある．強いAIやAGIとは人と同様の知能を持ち，あらゆる状況に対応して判断，行動を行うコンピュータのことである．平たく言えばドラえもんやC3POのようなサイエンスフィクションに登場する，人のように振る舞うロボットの知能のことを意味する．一方弱いAIや特化型人工知能と呼ばれるものは，特定の問題に対して適切な出力や判断を行うコンピュータのことである．自動車の自動運転のように，一見とても複雑で高度な判断を行っているように見えても，現在の人工知能技術はすべてこちらである．それを人工知能で何でもできるかのように喧伝されているから胡散臭く思われるわけである．

実際のところ，人工知能の対象とする問題の大半はパターン認識（Pattern Recognition）と機械学習（Machine Learning）の問題として扱われる．また，本講座のテーマである深層学習（Deep Learning）は機械学習の一手法である．人工知能，機械学習，深層学習はよく同時に使われ，混同されることも多いが，関係は図1のようになる．人工知能という言葉や分野の定義は研究者でも人に依って定義が異なるため，パターン認識と機械学習より大きな概念になっているが，実際のところ，現在のほとんどの人工知能は機械学習の問題として扱うことができる．近年では機械学習を用いて複雑な物理シミュレーションを高速に計算できるようにする[3)]，NP困難な組合せ最適化問題を解く[4)]といった例のように，人工知能の範囲外にも機械学習が応用されている．

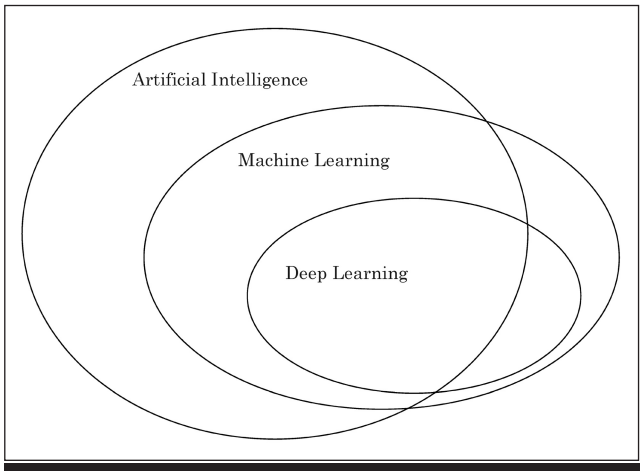


図1　人工知能，機械学習，深層学習の関係

このように，深層学習は機械学習の一手法であることから，パターン認識と機械学習についてある程度の知識があれば，現在の人工知能ブームに乗り遅れずかつ胡散臭い煽りにだまされないはずである．本稿ではパターン認識と機械学習の基礎的な考え方に関する解説とこれまでの機械学習と現在大ブームとなっている深層学習がどう違うのかについての解説を行う．

パターン認識と機械学習はどう違うのかと思われる方のために，Bishopによる世界的な名著“パターン認識と機械学習”（通称PRML）[5)]のまえがきの冒頭を引用すると，「パターン認識は工学を起源とするが，機械学習は計算機科学の分野から生じている．しかし，これらの研究活動内容は同じ分野を二つの側面から見た物とみなせ，両分野ともこの10年間に大きく発展した」とある．つまり何かを分類するような工学的な側面がパターン認識であり，計算機に知能的な処理を行わせるという計算機科学的な側面が機械学習である．そして，実際に用いられる技術や数理は呼び名が違う場合もあるが，ほぼ同じものである．本稿では前処理等の個々の問題に強く依存する処理も含む場合にはパターン認識と呼び，前処理等が行われた後の知識表現や判

† 大阪大学
"A Machine Learning Primer (1): Artificial Intelligence and Machine Learning" by Tomohiro Mashita (Osaka University, Osaka)

映像情報メディア学会誌 Vol. 72, No. 2, pp. 236～240（2018）

断基準を生成する処理を機械学習と呼んでいる．

上述のPRMLは非常に難しい本としても有名であり，よほど数学に強い方でない限り，これから機械学習を勉強しようという方にお勧めではない．その難しさは副読本として，光成滋生著“パターン認識と機械学習の学習”[6]というものも存在するほどである．“パターン認識と機械学習の学習”の普及版はインターネットで公開されている．これから勉強する方のためにPRML以外のパターン認識と機械学習の書籍を紹介しておくと，パターン認識としては，石井らによる“わかりやすい　パターン認識”[7]と“続・わかりやすい　パターン認識　教師なし学習入門”[8]が良い．また，機械学習として紹介するなら，Hastieらによる“統計的的学習の基礎”[9]をお勧めする．PRMLとこちらは著名な日本人研究者による翻訳が出版されており，日本語で読むことができる．もう少し薄い方が良いという方には“東京大学工学教程情報工学機械学習”[10]が良いと思う．

2．パターン認識

パターン認識とは，画像や音声といった特定の入力に対して，入力データに対してさまざまな処理を施し，個人のラベルや文字といった一定の規則に従う情報を判別して出力する技術である．入力となるデータには，ノイズやパターンとは関係のない情報も大量に含まれる．例えば，画像認識では対象となる物体の画素は必要だが，それ以外の背景にあたる部分のデータは不要である．このような，余計なデータを含んだデータの中から求める出力を得るためさまざまなアルゴリズムが駆使される．今回の講座のテーマである深層学習はそういったアルゴリズムの一つであり，深層学習を含む機械学習は多数のデータから必要な出力を得るためのルールや判断基準を決める技術である．このルールや判断基準を決定境界（Decision Boundary）と呼ぶ．筆者は，この決定境界を決めることを“線を引く”と呼んでいる．これについては，第2回で解説予定である．また，“線を引く”とは別の方法については第3回で解説予定である．

パターン認識のアルゴリズムはさらに図2のように分けられる．先ほど，深層学習は機械学習のアルゴリズムの一つと述べたが，これまでのパターン認識を大きく変えているため，この図のようにした．これらのアルゴリズムについて例を交えて簡単に解説する．

(1) ルールベース

ルールベースとは，アルゴリズムの動作のすべてが人によって決定されるアルゴリズムである．例えば，入力変数についてその値がある閾値を超えた場合と越えない場合に出力を変えるようなアルゴリズムである．If-thenルールと言っても良い．これらのアルゴリズムでは基本的に閾値等による条件分岐処理の組合せによって求める出力を得るため，閾値も含めてすべて人（プログラマ）がその内容を決定

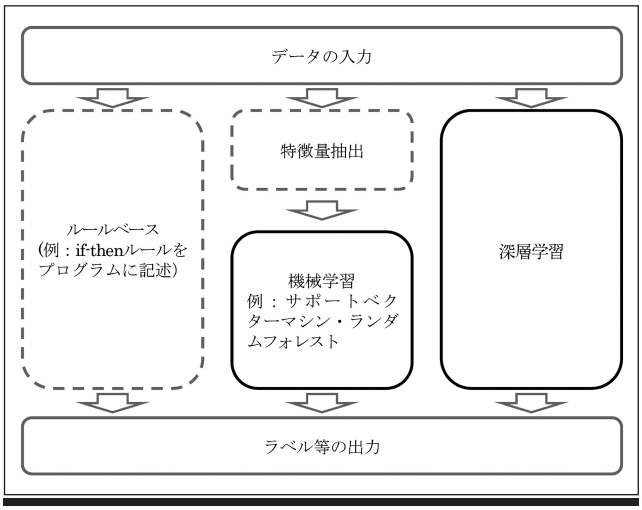


図2　パターン認識アルゴリズムの分類

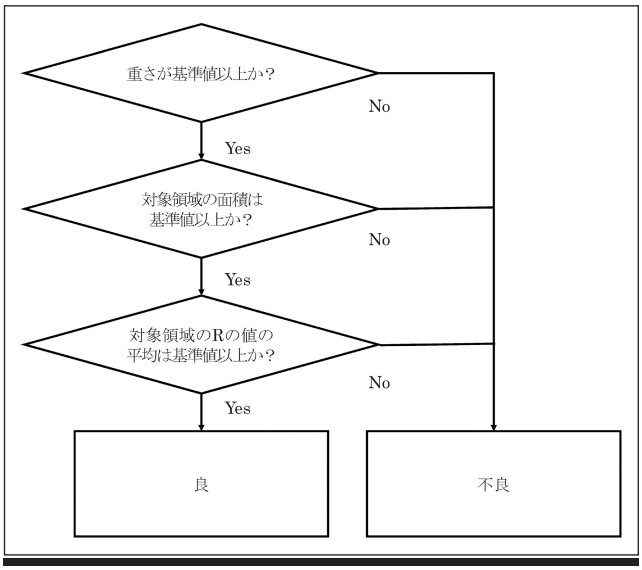


図3　if-thenルールの例

する．もっとも原始的なタイプのアルゴリズムであり，昔のゲームのAIなどはこのように作られていたと思われる．

例として，林檎をカメラや計り等で計測し，品質によって良と不良に分類するアルゴリズムを考えるとする．ルールベースのアルゴリズムでは，林檎が基準の重さ以上なら上質といったようなアルゴリズムなる．そして，その基準となる値は人によって決定される．もう少し複雑にするなら，図3のように，色合い，大きさなどについても基準を与えることになる．

(2) 特徴量抽出と機械学習

特徴量抽出と機械学習は深層学習以前のパターン認識の手法を意味している．これに含まれるアルゴリズムでは，出力を決定するルールはデータを使ってアルゴリズムで決定される．最もシンプルな例を考えるならば，先ほどのルールベースの閾値をデータから自動的に決定するものと

思えばよい．そしてその閾値を自動的に決めるアルゴリズムが機械学習である．実際には閾値を決めるだけでは上手く分類できないことも多いので，より複雑な決定境界を求められるアルゴリズムを用いることが多い．

機械学習のアルゴリズムは数多くの手法が開発されている．幾つか例を挙げると，K最近傍法（K-Nearest Neighbor: K-NN），Support Vector Machine（SVM），Random Forest, AdaBoostなどがあり，初期の浅いニューラルネットもこれに含まれる．機械学習のアルゴリズムにはそれぞれに特徴があり，対象となるデータの性質や量，利用可能な計算資源，求められる処理時間等によって使い分けられる．

機械学習によって決定境界をデータから自動的に決めることは可能になったが，入力となるデータに関しては人の手による加工が必要であった．例えば，画像認識の場合，観測された画像をそのまま入力とすることもできるが，特徴量抽出と呼ばれる処理を行うことで，より良い結果がえられた．特徴量は対象とする問題によってさまざまであり，その設計方法にも特別な方法論などはない．また，特徴量を設計するには，対象となる問題とデータについて充分な知識が必要であった．つまり，特徴量の設計こそがパターン認識における腕の見せ所と言える．これについては，第4回で解説予定である．

特徴量と識別器の秀逸な組合せの例として，有名なViolaとJonesによる顔認識アルゴリズム[11]を紹介する．このアルゴリズムでは，高速な顔認識を実現するために，Haar like特徴量と呼ばれる特徴量を使い，Adaboostと呼ばれる識別器を用いた．この手法では，比較的単純な特徴量であるHaar like特徴と単純な弱識別器を多数重ね合わせることで高性能を実現するAdaboostの組合せが非常に良く機能しており，高速な処理を実現している．また，顔画像に対してHaar like特徴がシンプルながらも相性が良かったと思われる．

ルールベースで用いた林檎の品質判定の例を，特徴量抽出と機械学習を用いた場合で考えてみる．先ほどの例では，キズや痛みがある場合を考慮していないが，品質を判定するならキズや痛みの判別は重要である．キズや痛みは色の変化等で判別できるとし，特徴量の例として赤チャネルのヒストグラムを採用する．そうすると，図4のようにキズのある方は赤い色以外の山ができる．キズのない林檎でも軸や模様で赤以外も多少ある．このヒストグラムはキズや痛みの具合によって変化する．変化の程度はさまざまであるし，良品質の方も林檎の色づき具合によってヒストグラムの形がかなり変化するため，ルールベースのような単純な基準による判別は難しそうである．また，ヒストグラムのbin毎に閾値を設定し，場合によってはさらに細かい条件を考えるのは現実的でない．このような場合に機械学習のアルゴリズムはどこで分ければ最も間違いが少ないかを決定する．間違いの少ない判断基準を決定するために，あらかじめ正しく良品質，不良品質の判定が行われた画像から抽出した特徴ベクトルを用意し，それらを最も適切に分類するように識別器のパラメータを決定する．このパラメータを決定する処理を学習（Learning）と呼ぶ．この例ではヒストグラムが特徴量ベクトルに当たる．必要に応じて重さや面積など，他の特徴も合わせて特徴量ベクトルとしても良い．このヒストグラムを特徴量として用いることは人間が設計している．これは例なので実際にこのような特徴量が上手く機能するかはわからないが，おそらく，ルールベースよりはより高度な識別が可能になるはずである．特徴量を求めた後にルールベースを適用することも可能だが，多くの場合特徴量は多次元のベクトルであり，先ほどの例のように，その各次元に対してそれぞれに適切なルールを設定してプログラムとして実装するのは現実的ではない．例えば，ある特徴量の分布が図5（a）のような分布だったとする．ここで●と×を分離する境界線を描くとすると，ルールベースではせいぜい図5（b）程度である．機械学習を用いて上手く学習ができれば図5（c）のような複雑な境界線を描くことも可能になる．実際の特徴量は非常に高次元のベクトルであり，図5のように分布を目で確認することは困難なので，実際の問題でのルールベースと機械学習の差は大きい．

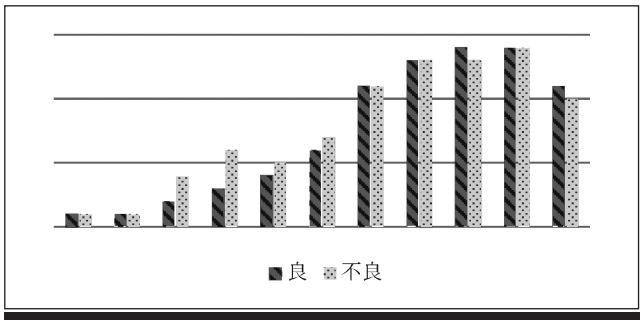


図4　ヒストグラム特徴

(3) 深層学習

深層学習の登場以後，上手くやれば入力から自動的に必要な情報を学習し，決定ルールも自動で決めるようなアルゴリズムが現れた．画像や音声といったセンサから得られるデータをそのまま入力とするだけで結果がえられるようなアルゴリズムである．つまり，特徴量の設計という人の手で行っていた部分の自動化が行われたわけである．このような学習は表現学習（Representation Learning）とも呼ばれる．

特徴量抽出のようなデータの変換が不要なアルゴリズムはEnd-to-endなどとも呼ばれ，盛んに開発が行われている．深層学習によって最も変わったのはこの点である．では，End-to-endで良い結果がえられるのならデータと計算資源があればよく，そこになんの工夫もないのかというと，もちろんそうではない．複雑な問題に対して，良い性能の識

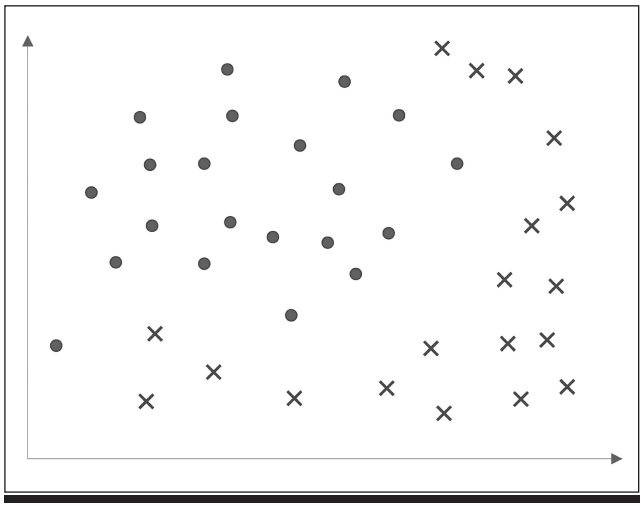


図5（a）　特徴量分布の例

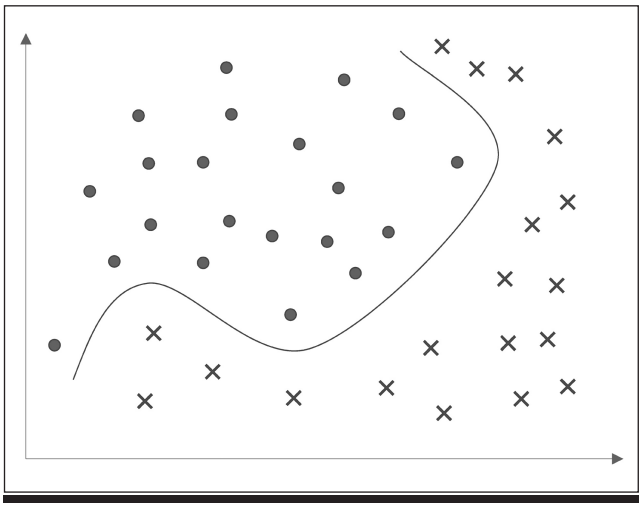


図5（c）　機械学習による決定境界の例

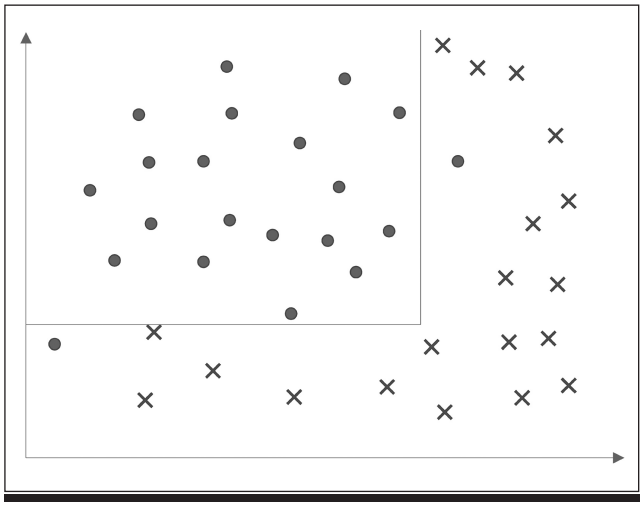


図5（b）　ルールベースによる決定境界の例

別器を作るにはネットワークの構造やデータの与え方，損失関数の定義などに工夫が必要である．これらをハイパーパラメータのチューニングと言ってしまうのは簡単だが，そこには機械学習の研究で積み重ねられてきた高次元のデータに対する理解や対象となる問題の知識が利用されている．この工夫などについては，第5回，第6回で解説の予定である．

もう一度林檎の品質判定を例にすると，深層学習を用いた識別の場合，大量の良/不良の判別済みの林檎画像を用意する．適切なネットワークと充分な計算資源を用意して学習を行う．ヒストグラムを作るとか赤さを計算するといったことはしない．ネットワークはおそらく畳み込みニューラルネットワーク（Convolutional Neural Network）が良く，問題も比較的シンプルなので極端に深いネットワークは必要ないと思われる．学習させるデータにもよるが，先ほどのヒストグラムの例では判別できなかった小さなキズも見逃さない識別器ができるかもしれない．ただし，学習データに含まれていなかった種類のキズは判定できないと思われる．

ここで，もう一度図2に戻ると，破線で囲まれた人の手で決定していた処理が機械学習，深層学習と変わる毎にデータから決定する範囲が広がっている．しかし，データから決定する範囲が広がるにつれて，必要とされるデータ量や計算資源も大きくなっている．それぞれに必要とされる知識も合わせると**表1**のようになる．すなわち，機械学習＋特徴量の時代では，そこそこのデータとそこそこの計算機，機械学習やデータサイエンスに関する知識，対象データに関する知識が必要であったのに対して，深層学習を使う場合には，文字通り桁違いのデータと計算資源が必要になる．加えて，難しい問題を解決するような識別器を実現するには対象に対する深い知識も必要である．

これらを踏まえると，従来のルールベースや特徴量＋機械学習といったアプローチはコスト面で優秀であり，問題の程度や必要とされる性能によっては，従来の手法を用い

表1　ルールベース，特徴量＋機械学習，深層学習のコスト比較

	ルールベース	特徴量＋機械学習	深層学習
計算資源	極小	中（PC程度）	大（PC程度～1,500万円以上）
データ	不要	必要	膨大な量が必要
対象の知識	要	要	要
機械学習・データサイエンスの知識	不要	必要	必要

る方が合理的な場合もあり得るだろう．林檎の例でも，単に大きさや重さで判断するだけで充分ならルールベースが最良の選択と思われる．

3．教師あり学習と教師なし学習

機械学習は教師あり学習（Supervised Learning）と教師なし学習（Unsupervised Learning）に分類され，教師のありなしとは正解となるデータを持っているか持っていないかということを意味する．教師なし学習と教師あり学習はパターン認識においてどちらも非常に重要なであり片方だけを学べば良いというものではないが，本講座では基本的には教師あり学習を取り扱う．

教師なし学習について簡単に説明しておくと，正解となるデータは持たずに，データに含まれる特定の性質を残したまま低次元のデータに変換するなど，より見やすい，あるいは扱いやすい形へのデータの変換やデータの分析を主な目的とする．代表的なアルゴリズムは，クラスタリングや主成分分析，自己組織化写像等である．この，データが特定の性質を持っているということがパターン認識では非常に重要な意味を持っている．特定の性質とは，データが取り得る空間中に特定の偏りが発生していることであり，その偏りこそがパターンの本質とも言える．2次元や3次元といった低次元のデータなら目で見てその偏りを捉えることができるかもしれないが，大抵のパターン認識の問題において入力となるデータは非常に高次元であり，目で見ることはできない．例えば，128 × 128=16384pixel grayscaleの画像データについて考える．これは画像としてはとても粗いものだが，パターン認識のデータとして考えると，ある一枚の画像は16384次元の空間中の1点と考える．ここで，例えば，顔画像など特定の種類の画像を集めて，各画像を一つの点として16384次元の空間中にプロットすると，一部分に偏って点が存在するはずである．パターン認識ではそのような偏りを記述することで対象とするデータ上手く認識しようとする．もちろん，そういった高次元のデータの集合を目で見ることはできないので，データを理解するには数学，統計学，などさまざまな方法を駆使する必要がある．教師なし学習とはそのような技術とも言える．

なお近年では，深層学習を教師データを使わずに，あるいは少量の教師データで学習させる試みも行われていて，こちらを教師なし学習と呼ぶ場合もある．この場合，目的としている出力は教師あり学習で得られるような出力であり，データの偏りを教師データなしで捉えるという意味では教師なし学習になるが，アルゴリズムの目的としては教師あり学習に近いと言える．

4．むすび

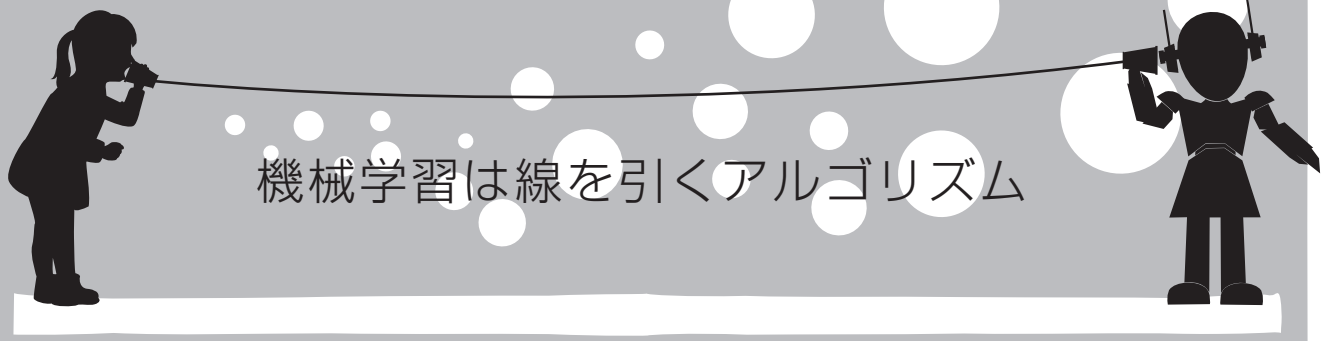
今回の講座では深層学習とパターン認識，機械学習がどういったものかについて解説した．その中で，ルールベース，従来の機械学習，深層学習がどのように違うのかを例をあげて解説した．また，教師あり学習と教師なし学習についても簡単に解説した．次回以降，機械学習のアルゴリズムについて幾つかの代表例を挙げながら解説し，講座の後半では深層学習についても解説を行う．

（2017年12月25日受付）

〔文　献〕

1）AlphaGO，https://deepmind.com/research/alphago/
2）Alpha Zero，https://arxiv.org/pdf/1712.01815.pdf
3）T. Okamoto, Y. Uranishi, T. Mashita, P. Ratsamee, K. Kiyokawa, H. Takemura: "Realtime Generation of Caustic Images Using a Deep Neural Network", The 16th IEEE International Symposium on Mixed and Augmented Reality（2017）
4）A. Milan, S.H. Rezatofighi, R. Garg, A.R. Dick and I.D. Reid: "Data-Driven Approximations to NP-Hard Problems", The Thirty-First AAAI Conference on Artificial Intelligence pp.1453-1459（2017）
5）C.M. Bishop：“パターン認識と機械学習上，下”丸善出版（2012）
6）光成滋生：“パターン認識と機械学習の学習－ベイズ理論に負けないための数学”，暗黒通信団普及版（2017）
7）石井健一郎，前田英作，上田修功，村瀬洋：“わかりやすいパターン認識”，オーム社（1998）
8）石井健一郎，上田修功：“続・わかりやすいパターン認識－教師なし学習入門”，オーム社（2014）
9）T. Hastie, R. Tibshirani, J. Friedman: "統計的学習の基礎－データマイニング・推論・予測”，共立出版（2014）
10）中川裕志：“東京大学工学教程情報工学機械学習”，丸善出版（2015）
11）P. Viola and M.J. Jone: "Robust real-time face detection", International journal of computer vision 57.2: 137-154（2004）

間下（ましたともひろ）　以大　大阪大学大学院基礎工学研究科博士後期課程修了．現在，同大学サイバーメディアセンター准教授．2012年，オーストリア・グラーツ工科大学客員研究員．コンピュータビジョン，機械学習，拡張現実に関する研究に従事．博士（工学）．正会員．



機械学習は線を引くアルゴリズム

正会員 間下以大†

1．まえがき

筆者は機械学習のアルゴリズムを一言で説明するために，“線を引く”アルゴリズムと言っている．もちろんこれは極端な解釈ではあるが，この“線を引く”ということを理解していれば．多くのアルゴリズムの理解に役立つはずである．わかりやすくするため”線を引く”と述べたが，線を引くアルゴリズムは識別モデル（discriminative model）と呼ばれる．そして”線を引く”とは異なるアルゴリズムもあり，それらは生成モデル（generative model）と呼ばれる．今回の講座ではこの線を引くアルゴリズム（識別モデル）について解説する．生成モデルについては第3回で解説予定である．

2．識別モデルと生成モデル

識別モデルと生成モデルの違いを簡単に述べるために，図1のような●と○のどちらかのラベルに分類する問題を考える．識別モデルの場合，図2のように線を引くことで分類を実現する．識別モデルはまさに図2のように線を引くことで線の右側と左側（実際には正負で変わる）に空間を分割し，そのどちらに属するかで未知の入力を識別する．生成モデルの場合，図3のように各ラベルの分布の形状やデータが生成される過程をモデルとして表現する．図3の破線は各データが生成される確率の高さを表す等高線である．そうすることで，新しい入力に対してそのデータがそれぞれのラベルに属する度合いを計算し，分類結果を得る．通常，ある入力に対して各ラベルに属する度合い（確率）を出力する（確率）モデルとして表現する．

機械学習で入力となるのは特徴量ベクトルと呼ばれる元のデータに何らかの処理を行い，識別を行うのに有用と考えられる情報である．前回の例では林檎の識別に色のヒストグラムを特徴量ベクトルとした．この特徴量ベクトルはパターン認識において非常に重要であり，識別の性能に関してはどのような機械学習のアルゴリズムを用いるかよりも目的に適した特徴量をどのように得るかが重要であることが多い．本稿の図では紙面上で図示するために1次元や2次元で特徴量ベクトルを表現しているが，通常はもっと高次元のデータになる．そしてその特徴量空間は1次元低い部分空間によって二つに分けることができる．例えば，2次元空間は1次元の線によって二つに分けられ，3次元空間は

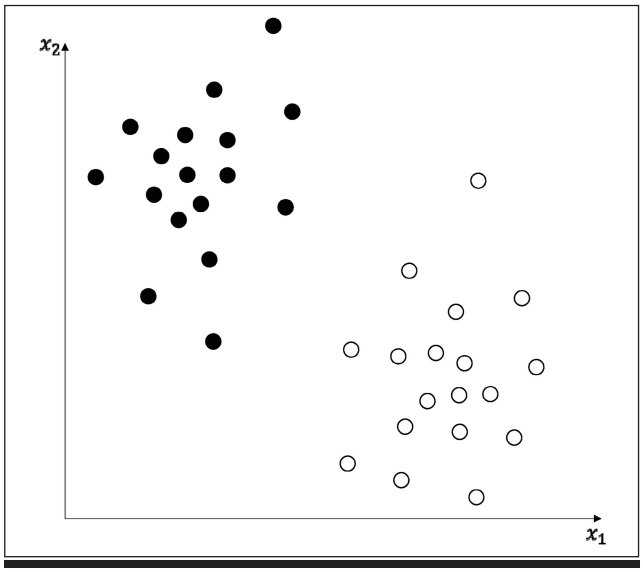


図1　2クラス分類の問題例

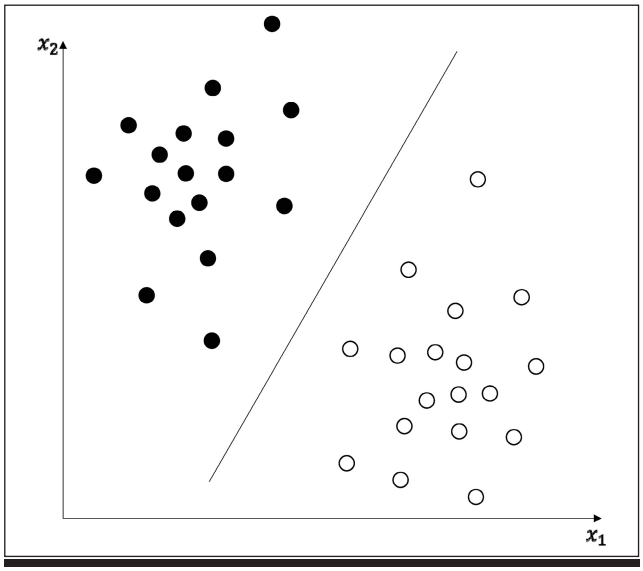


図2　識別モデルによる分類

† 大阪大学
"A Machine Learning Primer (2): Machine Learning is Line Drawing Algorithm" by Tomohiro Mashita (Osaka University, Osaka)

映像情報メディア学会誌 Vol. 72, No. 3, pp. 412～417（2018）

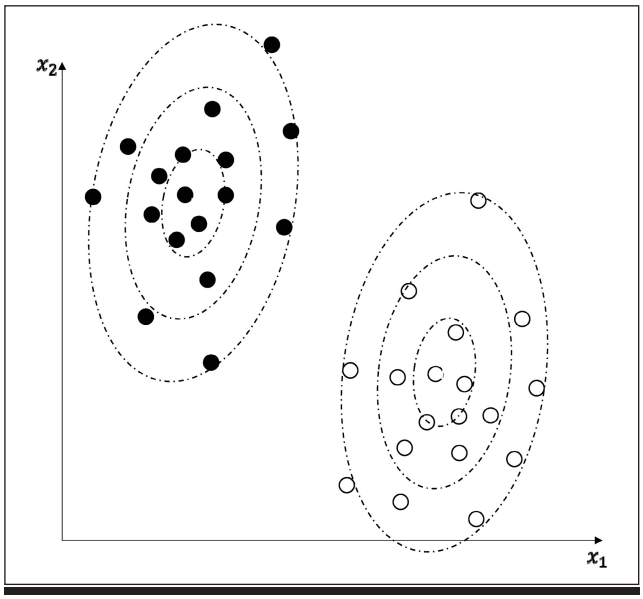


図3　生成モデルによる分類

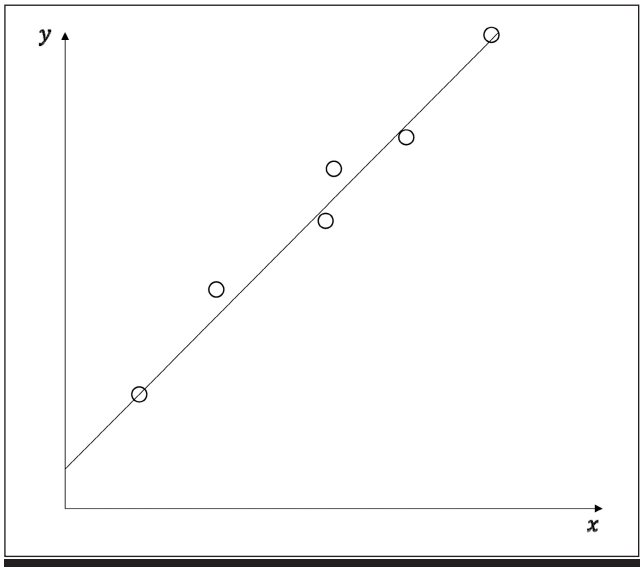


図4　回帰の例

2次元の平面によって二つに分けられる．この1次元低い部分空間は超平面（Hyper plane）と呼ばれ，“線を引く”の“線"とはこの超平面のことである．

識別モデルはさらに分類と回帰に分けられる．分類モデルでは入力に対してあらかじめ定義されている非連続な値のどれかが出力される．通常，その離散値をラベルとして扱い，ラベル付け問題を推定する．出力が0か1といった二つの値を持つ分類を2クラス分類と呼び，三つ以上の場合多クラス分類と呼ぶ．

一方，回帰モデルでは入力に対して出力される値が連続値を持つ．問題によっては出力も多次元となることもある．

単純な例を用いて図示すると，2次元（x_1, x_2）の2次元の入力に対する2クラス分類を図で示すと図2のようになり，1次元（x）から1次元（y）への回帰を図で示すと図4のようになる．どちらの場合においても，2次元の平面上に線が引かれていることがわかる．最初に述べたように，識別モデルのアルゴリズムはこの線を引くアルゴリズムであり，さまざまなアルゴリズムが発明されている．実際の問題に適用するアルゴリズムを決定する場合，識別性能は各アルゴリズムが描く線の複雑さや学習データの数，次元数，性質等によって変わってくる．問題によっては計算リソースや求められる処理時間等に制限があり，その制限によっても適切なアルゴリズムは変わる場合もある．

3．線はどのようにして決定されるのか

もう一度図1のように，二つのラベル（●と○）に分類する問題を考える．先ほど説明したように二つのラベルを分類するには図2のような直線を引くことができればよい．直線の方程式は

$$y - ax - b = 0$$

として表すことができる．つまり，ある入力$(x, y) = (x', y')$があるとすると，$y' - ax' - b$の値が正になる空間と負になる空間に分けられ，正の場合には●，負の場合には○と見なすことで分類結果がえられる．ここで，$y - ax - b = 0$としているが，実際に分類を行うためには具体的なa, bの値を決める必要があり，このa, bの値を既知のラベル付けされたサンプルデータから決定するのが機械学習である．この，既知のラベル付けされたサンプルデータは教師付データ（supervised data）や学習データ（training data）等と呼ばれる．実際の問題では高次元の入力になり，N次元空間を分割する超平面は

$$y = w_0 + \sum_{i=1}^{N} w_i x_i$$

という形で表現され，$w_i (i = 0, 1, ..., N)$を求める．

この例は最もシンプルな場合であり，1本の直線（線形な超平面）で正しく分離することが可能な分類対象となっている．これを，線形分離可能と呼ぶ．実際の分類問題では線形分離可能な場合は少ないが，線によって分類するという機械学習の性質が良く表されている．また，線形分離できない問題であっても線形分離の組み合わせによって適切な決定境界を得たり，入力データを何らかの形に変換したりすることで線形分離可能な状態にすることがよく行われる．回帰問題の場合も線を引くという点でまったく同様であり，入力と出力の対応を表す線のパラメータを求めることで未知の入力に対する推定値を得ることができるようになる．

また，多値の場合も線が複数になるだけであり，本質的には同じ問題である．アルゴリズムによって他クラスにそのまま適用可能なものと2クラス分類の組み合わせが必要なものがあり，2クラス分類を組み合わせる方法には1対他（Onve-vs-the-rest）や1対1（One-vs-One）といった組み合わせ方法がある．

3.1 過学習と汎化性能

先ほど述べたように，実際のパターン認識の問題では線形分離できることは稀である．そのため，機械学習のアルゴリズムは複雑な決定境界を表現できるようになっている．しかし，複雑な決定境界が常に良い性能の識別器になっているとは限らない．

教師あり学習では未知の新たな入力に対して正しい識別を行うために，既知の正解付きデータから決定境界を求める．学習用のデータが図5(a)のような状態であった場合，緩やかな曲線で決定境界を表現すると図5(b)のようになる．一方，複雑な曲線で表現すると，図5(c)のような決定境界も作成可能である．学習用のデータに対する性能という点では，(c)の方が誤識別がなく良い決定境界と言えるが，図5(b)で誤識別となっているデータは何らかの理由で誤識別となるような値のデータとなったのかもしれない．ここで何らかの理由とは，データの計測時のエラー，教師ラベルの付け間違い，そもそも特徴量が良くない等，さまざまな理由が考えられる．実際のパターン認識の問題ではさまざまな理由でこのような外れ値が発生している．このような外れ値を識別するために複雑な決定境界を用いるとその外れ値付近で誤識別が発生しやすくなり，学習データに対する性能は良いが，未知のデータに対しての性能は悪くなるという状態になる．このような学習を過学習(Overfitting)と呼び，未知のデータに対する性能を汎化性能と呼ぶ．

過学習にならないために，交差検証(Cross validation)という技術が用いられる．交差検証には幾つか方法があるが，代表的な方法は学習データをK個に分割し，一つを評価用残りを学習用のデータとして学習と評価を行う．これをK個に分割されたデータセットについてそれぞれ行い，その平均を用いる方法である．

3.2 ハイパーパラメータ

後述する機械学習のアルゴリズムは学習によって決定されるパラメータの他に，あらかじめ与えるパラメータを持っている．例えば，多項式回帰の多項式の次数，K最近傍法のKの値等が該当する．深層学習におけるネットワークの深さや，活性化関数の選択，畳み込みニューラルネットのチャネル数等も含まれる．これらは総称してハイパーパラメータと呼ばれ，その値によって識別器の汎化性能や処理時間が大きく変化することが多い．そのため，実際の問題では単に識別器に学習データを入力するだけでなく，ハイパーパラメータのチューニングが必要になる．このチューニングは人手による試行錯誤によって行われることが多く，ハイパーパラメータの数が増えるほどチューニングは難しくなる．また，一度の学習とテストで確認できるのは一つのハイパーパラメータの値なので，学習に時間がかかる場合，パラメータの試行錯誤そのものが難しい場合もある．深層学習が黒魔術と呼ばれる原因の一つはこの

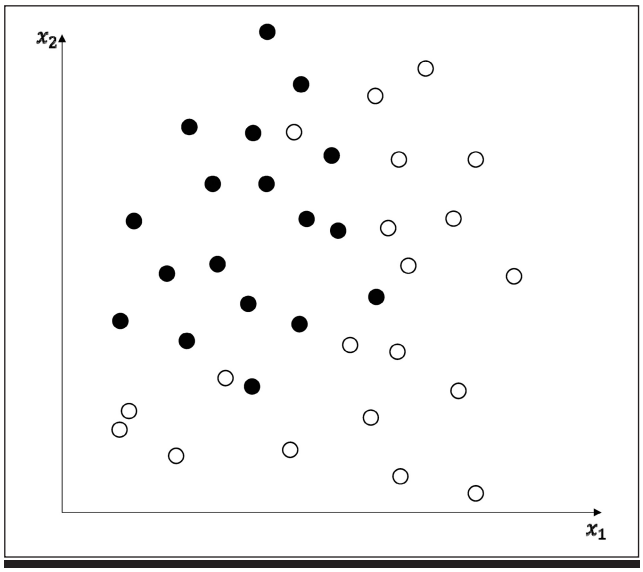


図5(a) 複雑な境界となる問題の例

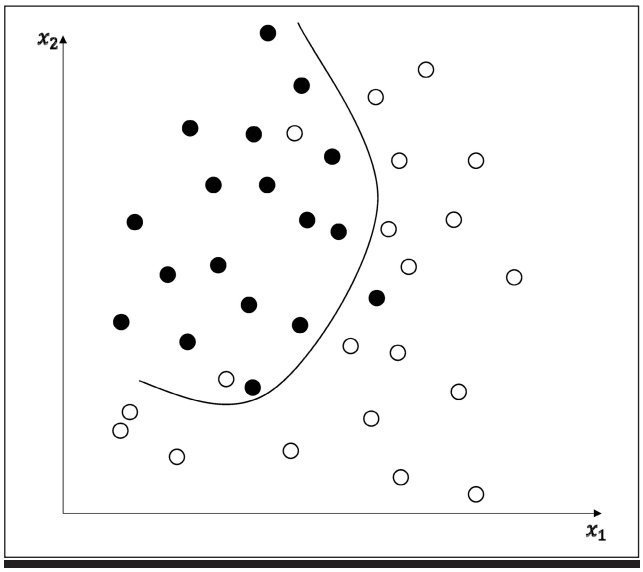


図5(b) 緩やかな曲線による決定境界

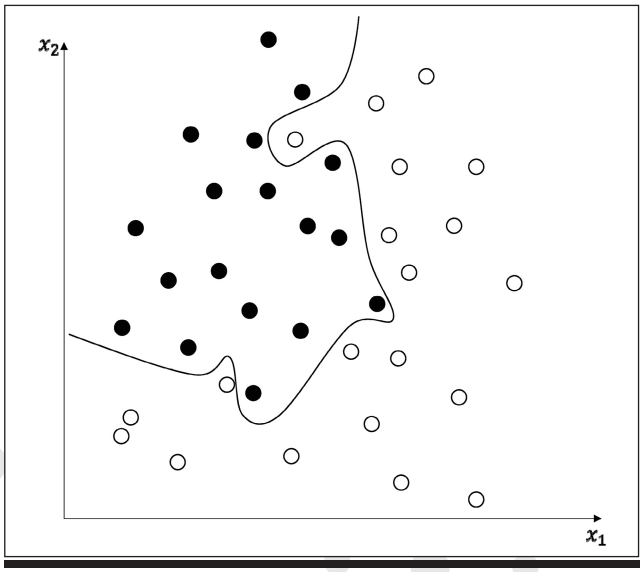


図5(c) 複雑な曲線による決定境界

チューニングの難しさにある．一方，後述するサポートベクターマシンやRandom Forest等は比較的ハイパーパラメータが少なく，チューニングが容易である．

4．機械学習のアルゴリズム

4.1　線形判別分析

先ほど直線を引くと述べたが，図1のような分布の場合，与えられているサンプルデータに対して正しく分離することができる直線は，図6に示すようにいくつも存在する．例えば，図6の線aや線bのような決定境界の場合，片方のラベル対しては確実に分類できるが，もう片方のラベルについては学習データから少し離れた位置の入力に対して誤識別をすることがわかる．また図6の線cもサンプル点に近く，その近いサンプル点から少し外れただけで誤識別が起きそうである．このデータの場合，線dのような決定境界が良さそうである．このように，無数に考えられる線の候補の中で未知のデータに対しても良い結果を出力できる線を決定する必要がある．適切な決定境界を決定する方法の一つが，線形判別分析（Linear Discriminant Analysis: LDA）である．線形判別分析では，クラス間の分散を分母に，クラス内の分散を分子にとった評価関数を最小化するような評価軸を作成し，その評価軸上の正負で2クラスの判別を行う．分散が大きくなる軸，小さくなる軸とは図7のような軸であり，データの広がりが大きい方向と小さい方向と考えればよい．先ほどの例で，クラス内の分散が小さくなる軸とは図8の軸aやbであり，クラス間分散を大きくする軸とは図8の軸cのような．各クラスの中心の距離が大きくなる軸になる．これらを総合した結果，図8の軸dに示すように，分母であるクラス間の分散を大きく，クラス内の分散を小さくとるような軸で入力を評価する．言い換えればその軸と直交する超平面（図8の線e）が決定境界となる．なお，図6の線aやbを悪い決定境界としたが，実際にはそうでない場合もある．例えば，誤識別に対するリスクが極端に異なる場合，そのリスクを考慮すると図6のaやbのような決定境界が適切となる場合もある．

4.2　最近傍法，K最近傍法

サンプルデータを直接利用したシンプルな手法が最近傍法やK最近傍法である．最近傍法は入力された未知のデータと最も近い距離にある教師データと同じラベルや値を出力する方法，K最近傍法は最も近い上位K個のデータから出力を決定する方法である．K個の最近傍サンプルを等しく扱ってもよいし，距離に応じて重みを付けることもできる．回帰問題の場合は，距離の逆数を重みにする等して出力値の重み付き平均をとれば連続値としても値を得ることができる．K最近傍法によって得られる決定境界はKが小さい場合には細かく複雑になり，Kが大きくなるとゆるやかな曲線となる．また，K最近傍法は上述のような決定境界のパラメータを持たずに決定境界が決まるため，ノンパ

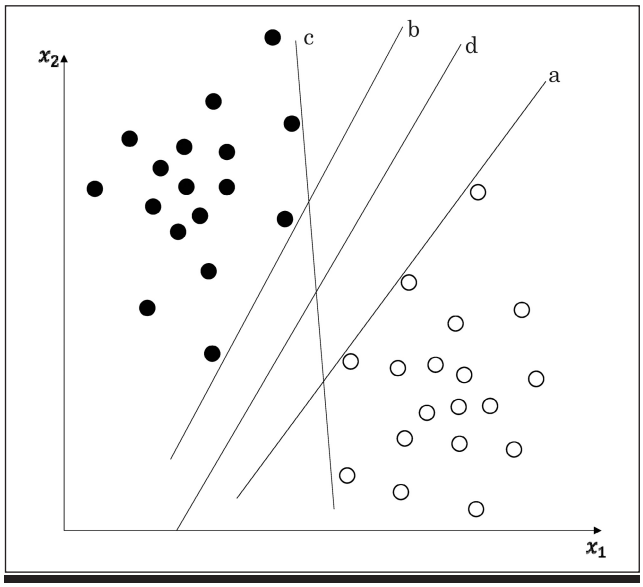


図6　複数の決定境界候補の例

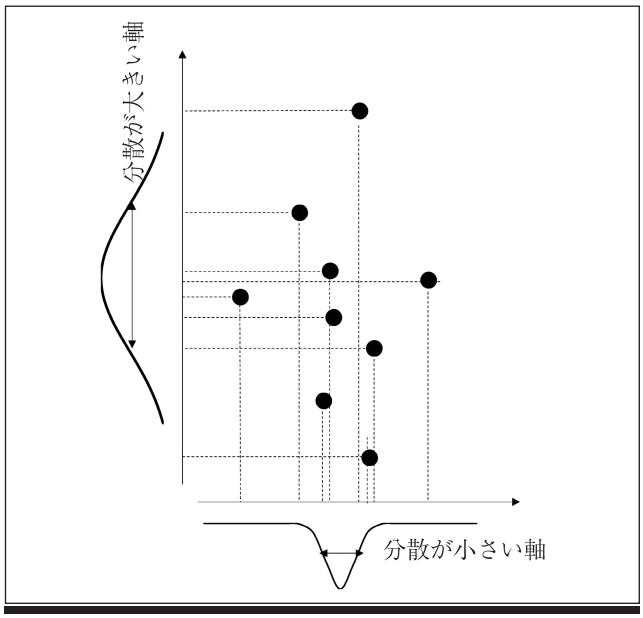


図7　分散が小さい軸と大きい軸

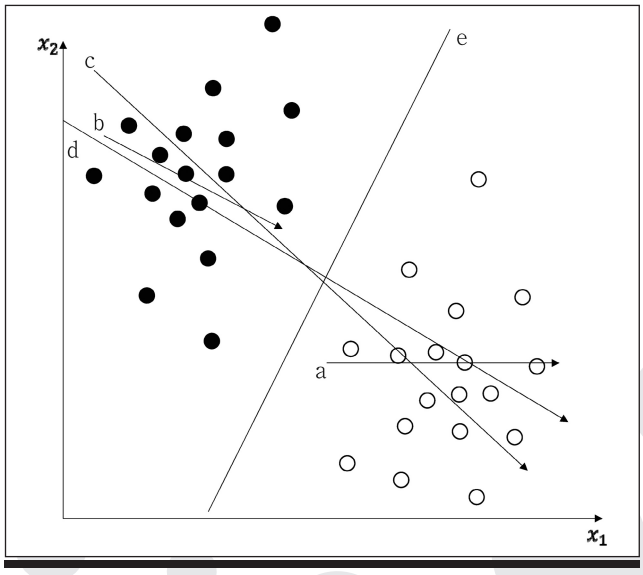


図8　線形判別分析の例

ラメトリックモデルと呼ばれる手法の一つである．

最近傍法では，未知のデータが入力される毎に各教師データとの距離を計算し，それらをソートして最近傍点を決定するため計算量が大きい．最近傍法による識別器の性能を上げるにはなるべく多くの教師付データを用意する必要があるため，計算時間と性能のトレードオフが問題となる．計算を効率化するためにkd-treeといった探索アルゴリズムがある．

最近傍法に限らないが，距離の計算についても注意が必要な場合がある．例えば，長さと重さといった同一に扱うことができない多次元の値を入力とする場合，重み係数を決める等の方法で一つの距離尺度にする必要があり，その距離尺度によって性能が左右される可能性がある．

このように，最近傍法は機械学習のアルゴリズムの中でも最も単純な方法であり効率的ではないが，かといって性能が悪いとは限らない．例えば，充分に密度の高い教師データが得られており，処理時間に制限がない場合，優秀な識別器となる．このことは機械学習において重要であり，最新の複雑な識別器を高性能な計算機で時間をかけて学習した結果，最近傍法の方が汎化性能が良いということも起こりえる．特に次元の低いデータは比較的教師データの密度を上げやすいため，最近傍法を基準として性能を評価することも考慮すべきである．

4.3　サポートベクターマシン

深層学習が大ブームとなる以前はサポートベクターマシン（Support Vector Machine）が良く用いられる識別器の一つであった．SVMはマージン最大化という考え方に基づいて決定境界が求められる．線形判別分析でも述べたように，線形分離可能な2クラス分類問題であったとしても，その決定境界は一つではなく，どのように定めるかによって識別器の汎化性能が変化する．これについて，SVMでは図9に示すように，決定境界と各クラス内のサンプルとの最短距離（マージン）を最大化する基準によって汎化性能の良い決定境界を得る．

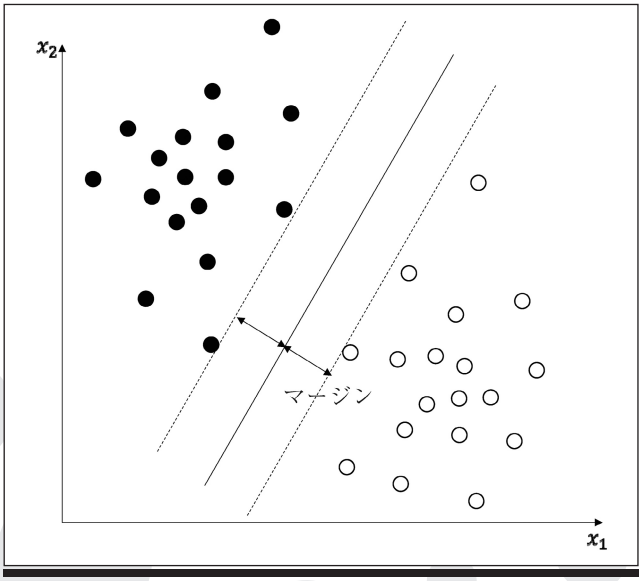


図9　サポートベクターマシンにおけるマージン

上記の説明は線形分離可能な場合についてであったが，多くの問題は線形分離可能ではない．SVMではそのような問題に対応するため，ソフトマージンとカーネル法と呼ばれる技術を用いる．ソフトマージンは決定境界を越えるようなサンプルをある程度許容する方法である．カーネル法は線形分離できない場合，つまり直線で二つのクラスを分離できない場合に，xをカーネル関数と呼ばれる関数によって別の空間に変換し，その空間内で線形に分離する技術である．

カーネル関数による変換では通常，非線形で入力よりも高次元の空間に変換され，その場合元の空間での決定境界は非線形なものになる．図10のような線形分離できない場合に，図10の×印からの距離の二乗を加えた3次元の空間で考えると，線形分離できる．×印からの距離を横軸にとり，縦軸にその2乗をとると図11のようになり，線形分離が可能になっていることがわかる．そして，図10の空間

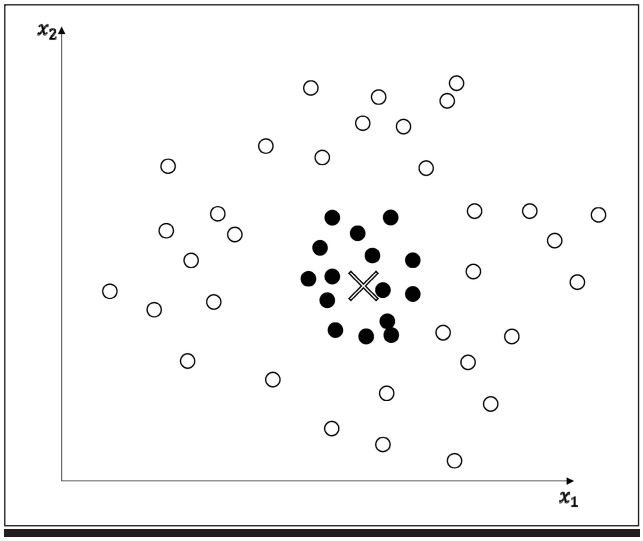


図10　線形分離できない例

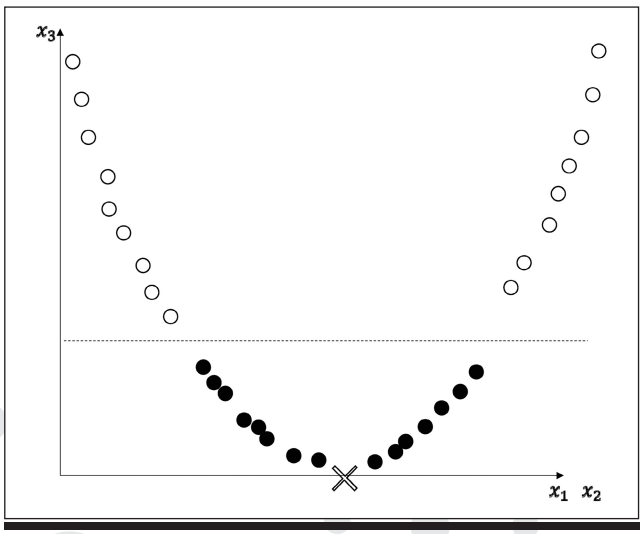


図11　カーネル法による線形分離の例

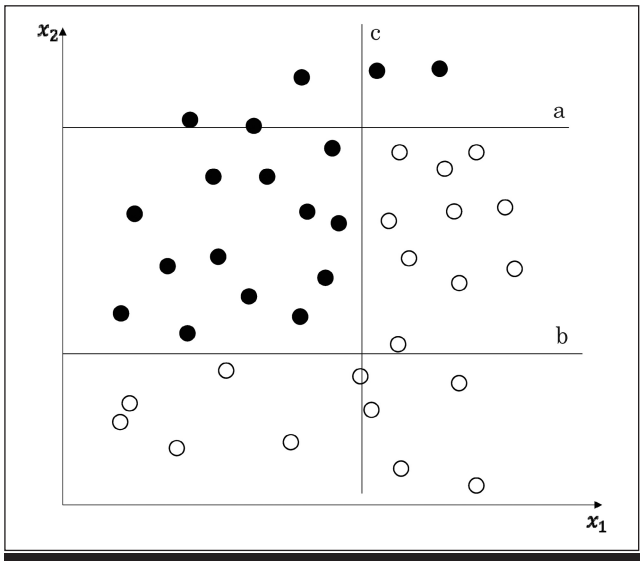


図12　弱識別器の多数決による識別例

では×印を中心に円を描くような曲線が描かれる．このようなカーネル法はSVMに限らず，特徴量ベクトルを拡張する方法としてさまざまな識別器に応用される．

上記の例は，カーネル法をシンプルに表した一例であり，カーネル関数はさまざまなものがある．そのため，問題によって適切なものを選択する必要がある．実際にSVMを用いている例ではRBF（Radial Basis Function）カーネル（別名，ガウシアンカーネル）を用いることが多い．SVMはチューニングの必要なハイパーパラメータが少なく汎化性能に優れている識別器を作成しやすいとされており，先述のように深層学習以前には良く用いられる識別器の一つであったと思われる．

4.4　Adaboost

Adaboostはboostingと呼ばれる技術を用いた識別器である．boostingでは多数の弱識別器（weak classifier）の合議（committee）によって良い識別性能を実現する．例えば，ある2クラス分類問題に対して一つの弱識別器の性能が50%を少し越える程度であったとしても，それらを多数組み合わせることで良い性能が得られる．図12に簡単な例を示す．図12では直線a～cがそれぞれ弱識別器である．個々の直線では●と○を正しく分類できていないが，直線の組み合わせによって正しく識別されることがわかる．Adaboostでは前の識別器で誤識別されたサンプルの重みをつけることで新たな識別器を適応的（Adaptive）に生成する．

Adaboostは前回の講座でも紹介したViola and Jonesの顔識別[2]でも用いられている．この手法では，Haar like特徴量とAdaboostの組み合わせが良く機能しており，高速な処理を実現している．また，Adaboostの適応的な性質を利用したEnsemble tracking[3]では，画像中の追跡対象の変化に応じて弱識別器を再構成することで適応的な物体追跡を実現している．

4.5　Random Forest

Random Forest[4]は決定木（decision tree）を弱識別器として多数用い，合議による識別を行う識別器である．決定木は第1回で解説したif-thenルールのように，ある値についての判断を繰り返して求める出力を行う方法であり，データからそのルールを学習させることもできる．決定木は処理の過程が単純で理解しやすいが，識別器としての性能は良くないことが多い．Random Forestではこの決定木を多数用いることで高性能な識別器を実現する．Random Forestや先述のBoostingのように，多数の識別器を用いる方法を集団学習（ensemble learning）と呼び，多数の識別器の合議によって汎化性能が良くなることが知られている．また，Random Forestはハイパーパラメータが少なく使いやすい識別木と言える．

5．むすび

今回の講座では，教師付学習による識別モデルについて解説した．識別モデルと生成モデル，分類と回帰等，機械学習の基本的概念を説明したのち，代表的なアルゴリズムについて簡単な解説を行った．各アルゴリズムの詳細については，第1回で引用した書籍を参照して頂きたい．ニューラルネットも機械学習のアルゴリズムであり，同列に解説することもできるが，本講座の後半で解説予定のため今回の講座では省略した．

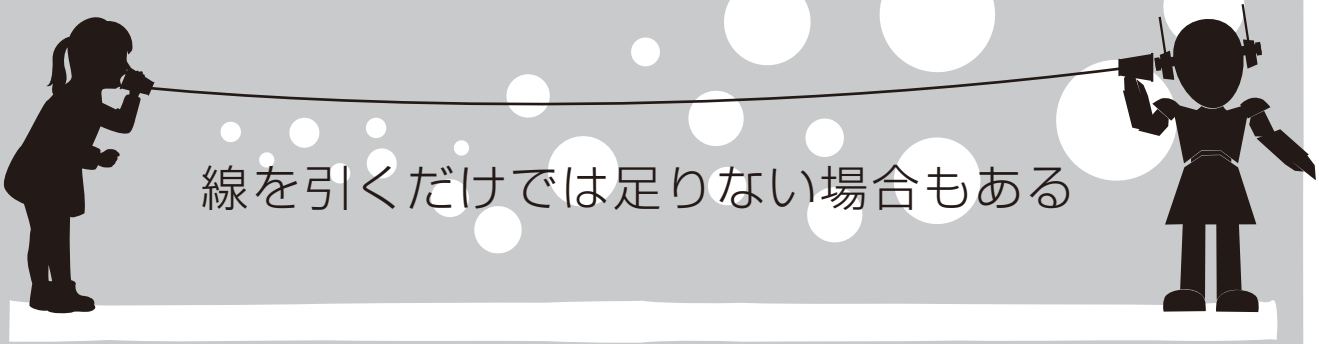
機械学習のアルゴリズムは幾つも存在するが，その性質はそれぞれ異なる．深層学習ブームで機械学習＝深層学習と考えたり，深層学習が常に最良の選択と考えたりする人もいるようだが，問題によっては最近傍法が最も適切な場合もあるし，決定木が最適な手法の場合もある．第1回の講座でも述べたように用意できる教師データの数，実際に使える計算リソース，性能によって適切なものを選択すべきであり，そのためには識別器と問題の特徴をよく理解することが重要である．

（2018年3月13日受付）

〔文　献〕

1）Y. Freund and R.E. Schapire: "A decision-theoretic generalization of on-line learning and an application to boosting", Journal of computer and system sciences, **55**, 1（1997）, pp.119-139
2）P. Viola and M.J. Jones: "Robust real-time face detection", International journal of computer vision, **57**, 2, pp.137-154（2004）
3）S. Avidan: "Ensemble tracking", IEEE transactions on pattern analysis and machine intelligence, **29**, 2（2007）
4）L. Breiman: "Random forests", Machine learning, **45**, 1, pp.5-32（2001）

間下（ましたに）　以大（ともひろ）　大阪大学大学院基礎工学研究科博士後期課程修了．現在，同大学サイバーメディアセンター准教授．2012年，オーストリア・グラーツ工科大学客員研究員．コンピュータビジョン，機械学習，拡張現実に関する研究に従事．博士（工学）．正会員．



線を引くだけでは足りない場合もある

正会員 間下以大†

１．まえがき

第1回では人工知能と機械学習，深層学習の概略について紹介し，その違いについて紹介した．第2回では機械学習とは高次元の特徴量空間に線（超平面，hyper plane）を引くためのアルゴリズムであることを紹介した．今回は線を引くのではなく，データの集合を記述する方法について解説する．これは，第2回の最初に述べたように機械学習のアルゴリズムは識別モデル（discriminative model）と生成モデル（generative model）に分類される内の生成モデルに該当する．

２．識別モデルと生成モデル

識別モデルと生成モデルの違いは第2回で解説した．識別モデルではデータの識別を目的としているため，ある超平面に対する正負が基本的な考え方になる．モデルによっては確率を出力する場合もある．一方，生成モデルではサンプルデータからデータの全体像を記述しようとする．データの全体像を記述できればさまざまなことに利用できる．識別はその一つであり，新たな入力が識別対象のクラスに属する確率を求め，その大小で識別を行う．他にも，データの全体像がモデル化されているため，発生する確率の低いデータを発見したりすることができる．また，データ生成過程を記述することができれば，生成過程のパラメータを変更することで，計測が難しい条件下のデータを生成したり，条件の異なる問題にモデルを応用することで必要となるサンプルデータの数を少なくしたり等が可能になる．あるいは条件の異なる未知のデータの生成自体が目的となる場合もある．ただし，これらは正しく生成モデルを記述できた場合の話である．通常，生成モデルは識別モデルよりも複雑であり，学習には多量のデータと計算資源が必要である．すなわち，単に識別を行うだけならば多くの場合識別モデルの方が効率が良い．

３．確率論

生成モデルは対象データの全体像を記述し，識別等に利用すると述べた．このデータの全体像を記述するための道具が確率論である．まずは確率論の基本につい簡単に説明する．

確率の例題として，以下を考える．

例題1.1：二つの箱b_1, b_2があり，b_1の箱には当たりが60%，ハズレが40%の割合で入っている，一方，b_2の箱には当たりが30%，ハズレが70%の割合で入っている．箱を無作為に選んだ場合に当たりを引く確率を求めよ．

これは，X＝当たりとなるすべてのYについて足し合わせることで求める確率が得られる．すなわち，

$$
\begin{aligned}
p(X=\text{当たり}) &= \sum_Y p(X=\text{当たり}, Y) \\
&= \frac{1}{2}\times\frac{6}{10}+\frac{1}{2}\times\frac{3}{10} \\
&= \frac{9}{20}
\end{aligned}
$$

となる．これを確率の加法定理と呼ぶ．この，他の変数（この例の場合はY）について，足し合わせることを周辺化と呼ぶ．また，右辺の$p(X=\text{当たり}, Y)$は，箱b_1を選ぶか当たりを引く確率であり，同時確率（joint probability）と呼ぶ．なお，ここでYはb_1, b_2についてどちらかが選ばれれば片方は選ばれないという条件の下に加法定理が成り立っている．また，X, Yはそれぞれ確率変数と呼ぶ．

確率の加法定理

$$p(X)=\sum_Y p(X,Y)$$

同時確率

X, Yが独立のとき

$$p(X,Y)=p(X)p(Y)$$

さらにもう一つの例題として，以下を考える

例題1.2：例題1.1の箱について，無作為に箱を選びクジを引いた結果が当たりであったとする．このとき，選んだ箱がb_1である確率を求めよ．

† 大阪大学
"A Machine Learning Primer (3); There is a Case Where Line Drawing is Not Enough" by Tomohiro Mashita (Osaka University, Osaka)

この場合の確率は条件付き確率（conditional probability）$p(Y=b_1|X=$当たり$)$で表される．同時確率とは異なり，真ん中が縦棒になっている．これは，カッコ内の右側の条件が与えられた時に左側である確率を意味しており，

条件付き確率

$$p(Y=y|X=x)=\frac{p(Y=y,X=x)}{p(X)}$$

として定義される．例題の場合，分母のは例題1.1で得られているため，

$$p(Y=b_1|X=\text{当たり})=\frac{\frac{6}{20}}{\frac{9}{20}}=\frac{6}{9}$$

となる．さらに，同時確率$p(Y=y,X=x)$は次式のように，条件付き確率$p(Y=y|X=x)$にその条件が発生する確率$p(X=x)$を掛けることで得られ，これを確率の乗法定理と呼ぶ．

確率の乗法定理

$$p(X,Y)=p(Y|X)p(X)$$

さらに，

確率の対称性

$$p(X,Y)=p(Y,X)$$

から

ベイズの定理

$$p(Y|X)=\frac{p(X|Y)p(Y)}{p(X)}$$

が得られる．ベイズの定理はパターン認識と機械学習で非常に重要な定理である．パターン認識の問題に当てはめて説明すると，左辺は事後確率（posterior probability）と呼ばれ，Xが観測されたときにクラスYに属する確率である．右辺の$p(X|Y)$は観測されたXがYに属しているとした場合の確からしさ（尤度，likelihood）を意味する．分母の$p(X)$は$\sum(p(X|Y)p(Y))$とすることができ，これは，上記の尤度の和を1にする正規化係数と見なすことができる．さらに右辺の$p(Y)$は事前確率（prior probability）と呼ばれ，クラスYが発生する確率を意味する．ベイズ主義と呼ばれる考え方ではこの事前確率を主観的に操作することでより良い結果を得る．逆に，観測から得られる情報だけで決定することを頻度主義と呼ぶ．ベイズ主義は意図を持って確率をコントロールするように見えるため，不自然に感じる人もいるかも知れないが，パターン認識の問題では観測や特徴量の設計に人の意図が加わっているものなので，その設計の延長に確率モデルの設計があると考えることもできる．

上記の確率の基本定理はXとYが独立であることを前提に成り立っているが，実際の問題の場合，独立でない可能性も考える必要がある．

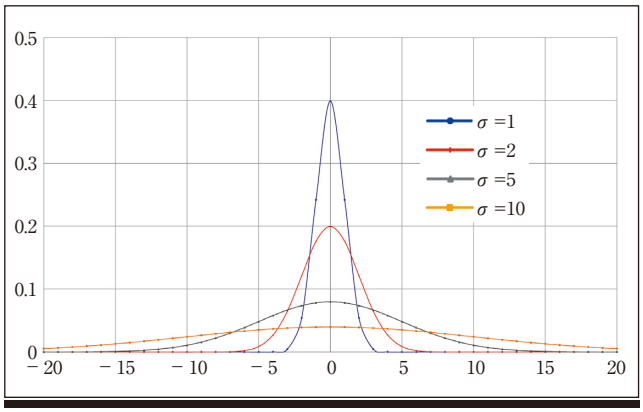


図1　ガウス分布

3.1　確率分布

確率分布は確率変数の値について，その値が起きる確率を表したものである．確率分布を利用した生成モデルでは，ある確率分布を仮定してそのパラメータを用意したデータ等から推定する．すなわち，分布の形状はユーザによって与えられ，その位置や大きさ等のパラメータを求める．

最も代表的な確率分布は次式のガウス分布（Gaussian distribution）である．

ガウス分布

$$N(x|\mu,\sigma^2)=\frac{1}{(2\pi\,\sigma^2)^{\frac{1}{2}}}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

ガウス分布は正規分布（normal distribution）とも呼ばれ，平均μと分散σ^2の二つのパラメータで定義される．ガウス分布の形状は図1のような形をしている．

xがd次元ベクトルとなる，多変量の場合は

$$N(x|\mu,\Sigma)=\frac{1}{(2\pi)^{\frac{d}{2}}}\frac{1}{|\Sigma|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$

となる．μは平均ベクトル，Σは共分散行列（covariance matrix）とよばれ$d\times d$の行列であり，$|\Sigma|$は行列式である．ガウス分布はパターン認識の問題で非常によく使われる．これは実際の問題にフィットしやすいだけでなく，数学的に扱いやすいことも理由である．扱いやすさの一例として，二乗誤差を用いて回帰曲線のフィッティングが，ノイズがガウス分布に従うと仮定した尤度の最大化と見なすことができるというものがある．これは，ガウス分布に従う尤度の計算が対数尤度を用いることで，実際の計算がガウス分布の式の指数部分で済み，すなわち線形代数で計算できること意味している．これは非常に便利で実用的な性質であり，多くの問題でガウス分布を仮定したモデルが使われる．また，第2回で解説した線形判別分析は等分散なガウス分布を仮定している．

どんな問題に対してもガウス分布が当てはまるわけではないので，注意が必要である．すなわち，ガウス分布を仮

定しているということは図1を見てもわかるように，ほとんどの値が平均付近に集中していることを仮定している，ところが，実際のデータに他の値から大きく外れたデータ（外れ値，outlier）が多く含まれていると，仮定されているガウス分布に適合していないと言える．この場合，ガウス分布のパラメータ（平均，分散）が外れ値の影響を強く受けることになり，適切なモデルが得られなくなる．同様のことは深層学習の損失関数（loss function）の設計にも言える．深層学習でも二乗誤差を使うことが多いが，対象としている問題の損失関数として適切であるかはよく考える必要がある．確率分布が上手くフィットしているかはカルバック・ライブラ情報量（Kullback-Leibler divergence）を使って確認すると良い．確率分布はガウス分布以外にもさまざまなものがあり，各分布の事前分布として用いられる分布もある．詳しくはPRML[1]等の文献を参照して頂きたい．

複雑なデータの全体像を表現するため，複数の確率分布を重ね合わせて用いることもある．代表例は混合ガウス分布（Mixtures of Gaussians）である．

混合ガウス分布

$$p(x)=\sum_{k=1}^{K}\omega_k N(x\,|\,\mu_k,\Sigma_k)$$

$$0\le\omega_k\le 1$$

$$\sum_{k=1}^{K}\omega_k=1$$

混合ガウス分布では複数のガウス分布を重みパラメータω_kを掛けて重ね合わせる．混合ガウス分布の例を図2に示す．混合ガウス分布のパラメータ推定には通常，EMアルゴリズムと呼ばれる手法を用いる．EMアルゴリズムでは各分布のパラメータ（ガウス分布の場合は平均と分散）の推定と各分布の重みパラメータω_kを交互に推定することで適したパラメータを求める．初期値となる各分布の平均と分散の最初の値によって得られる結果が異なることがある．また，混合数Kは通常，ハイパーパラメータになる．

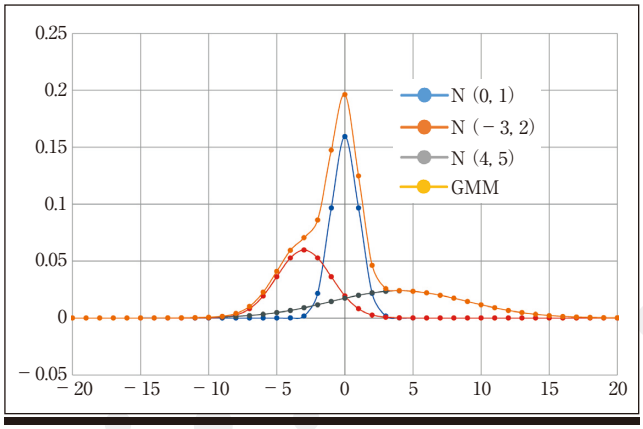


図2　混合ガウス分布の例

3.2　ハイパーパラメータとオーバフィッティング

確率モデルを用いた生成モデルの場合，その分布の形状は比較的単純であり，第2回で説明したオーバフィッティングが発生することは少ないと考えられる．またハイパーパラメータのチューニングも少ないと言える．しかし，混合ガウス分布のような複数の分布の重ね合わせ等でモデルを記述した場合は，その重ね合わせの数や初期値などに工夫が必要な場合もある．また，次章で述べるGenerative Adversarial Network（GAN）[2]はニューラルネットで生成モデルを実現する方法であるが，他の深層学習と同様に多数のパラメータが存在する．

4．サンプリング

生成モデルの形状を決める要素として確率分布について説明してきたが，モデルのパラメータを決める学習データについても注意が必要である．一般的に学習に用いるデータは均質にサンプリングされたものと仮定されている．しかし，実際のデータではサンプリングに偏りが発生することもある．生成モデル，識別モデルに限らず，サンプリングの偏りは時として大きな落とし穴となるので，注意が必要である．ベイズ確率を用いるならば，サンプリングの偏りをあらかじめ事前確率として与えるなどの工夫を考えるべきである．

一方，データの偏りがサンプリングではなくそのデータの特徴を強く表している場合もあるので，実際の問題を扱う場合にはその問題とデータに対する深い理解が不可欠である．

5．Generative Adversarial Network

近年の深層学習の発達の中で，大きな発明の一つがGoodfellowらによるGenerative Adversarial Network（GAN）[2]である．日本語でもGANと呼ばれることが多いが，敵対的生成ネットワークと呼ばれることもある．GANによって，確率モデルでは表現できない高次元で複雑なデータの生成が可能になった．

GANは深層学習を用いたモデルであるため，モデルの構築や学習時の工夫には深層学習に関する知識が必要になる．深層学習自体の解説は第5回，第6回を予定しているので本稿ではGANの基本的な概念と派生した手法を幾つか紹介する．先に深層学習について勉強したいという方は，GANを提案したGoodfellowらによる著書，“Deep Learning”[3]を参考にされると良い．日本語版[11]も出版されている．また，岡谷による“深層学習”[4]や人工知能学会監修の“深層学習”[5]も良い．本稿で解説するGANについては，“はじめてのGAN”[6]とGoodfellowによる“NIPS2016 Tutorial: Generative Adversarial Network”[7]を参考にした．詳しく知りたい方はこれらを参照して頂きたい．

GANの基本概念は図3に示すように，Generator Gと

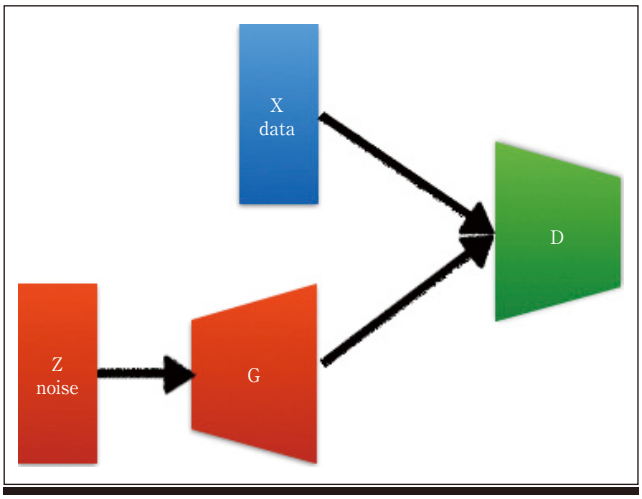


図3　GANの概念図（はじめてのGAN[6]より引用）

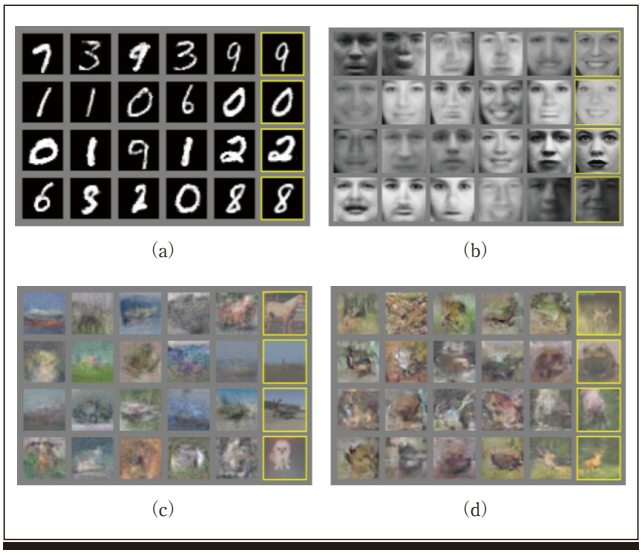


図4　GANで生成された画像（Goodfellowら[2]より引用）

Discriminator Dで構成される．入力はData Xとnoise Zである．GeneratorはZを入力として，Dを騙すようなデータを生成する．一方，Dには本物のデータXか，生成されたデータのどちらかが入力となり，Dはそれらを本物か偽物かを識別するように学習する．うまく学習が進むとGはXと区別がつかないようなデータを生成し，Dの識別率は50%に近づく．この状態でGはZを入力として，Xとよく似ているがXとは異なるデータが生成される．Goodfellowらによる結果を図4に示す．図4では各図の右端が訓練データでそれ以外が生成されたデータであり，よく似た画像が生成されていることがわかる．特に図4(b)は人の顔がよく生成されている．

GANを発展させた技術の中に，1章で述べたような，データの生成過程で特定の情報をコントロールするパラメータを獲得する方法[8]や明示的に与えた条件にしたがってデータを生成する方法[9]等が提案されている．また，シミュレーション画像を実画像のように変換する方法[10]も提案されている．このような技術を応用することで，機械学習で最もコストがかかると言える，教師付データの獲得や，そもそも観測が難しい確率の低いデータの獲得のコスト低減が期待されている．

GANは乱数からデータを生成できるため非常に有用な技術であることは間違いないが，そのモデルの学習は非常に難しいことが知られている．また，他の深層学習と同様に学習データも大量に必要であるなど，課題も多い．現在の研究でも日々新たなモデルや学習方法が考案されており，今後の発展が期待される．

6．むすび

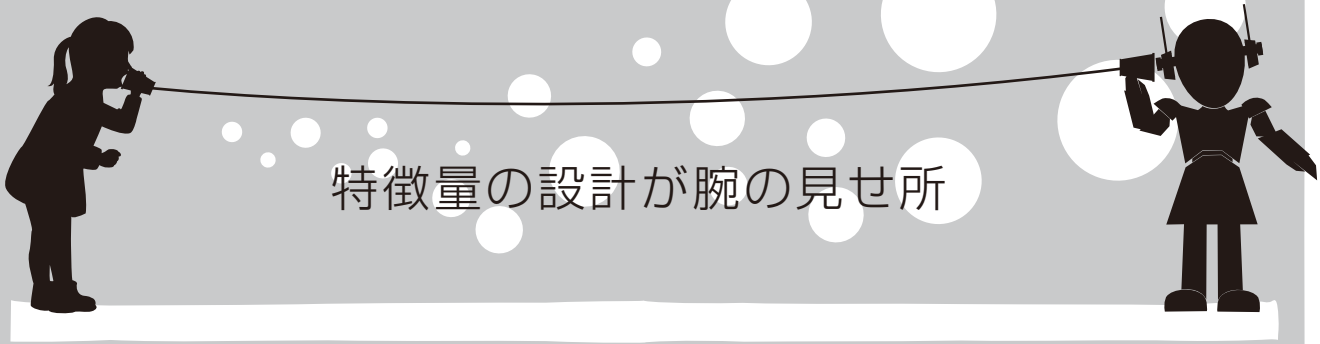
今回の講座では生成モデルの解説を行った．生成モデルはGANの出現以降盛んに研究されており，さまざまな利用方法が開発されている．一方で学習が難しいことでも知られている．複雑なモデルを上手く学習させるには繰り返し述べているように，機械学習に関する知識と対象に関する知識の両方が必要となる．次回第4回では対象に関する知識の表現である特徴量について解説を行う予定である．

（2018年5月17日受付）

〔文　献〕

1）C.M. Bishop：“パターン認識と機械学習上，下”丸善出版（2012）
2）I. Goodfellow, J.P.-Abadie, M. Mirza, B. Xu, D.W.-Farley, S. Ozair, A. Courville and Y. Bengio: "Generative adversarial nets", In Advances in neural information processing systems, pp.2672-2680（2014）
3）I. Goodfellow, Y. Bengio, A. Courville: "Deep Learning (Adaptive Computation and Machine Learning series)", The MIT Press（2016）
4）岡谷貴之：“深層学習”，講談社（2015）
5）人工知能学会監修：“深層学習”，近代科学社（2015）
6）はじめてのGAN，https://elix-tech.github.io/ja/2017/02/06/gan.html
7）I. Goodfellow: "NIPS 2016 Tutorial: Generative Adversarial Networks", https://arxiv.org/abs/1701.00160
8）Xi Chen, et al.: "Infogan: Interpretable representation learning by information maximizing generative adversarial nets", Advances in Neural Information Processing Systems（2016）
9）M. Mirza and S. Osindero: "Conditional generative adversarial nets", arXiv preprint arXiv:1411.1784（2014）
10）A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang and R. Webb: "Learning from simulated and unsupervised images through adversarial training", In the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3, 4, p.6（July 2017）
11）I. Goodfellow, Y. Bengio, A. Courville（著），岩澤有祐，鈴木雅大，中山浩太郎，松尾豊（監訳）：“深層学習”，KADOKAWA（2018）

間下　以大（ましたともひろ）　大阪大学大学院基礎工学研究科博士後期課程修了．現在，同大学サイバーメディアセンター准教授．2012年，オーストリア・グラーツ工科大学客員研究員．コンピュータビジョン，機械学習，拡張現実に関する研究に従事．博士（工学）．正会員．



特徴量の設計が腕の見せ所

正会員 間下以大†

1．まえがき

パターン認識において特徴量の設計は非常に重要であり，識別性能を最も大きく左右する．しかしながら，特徴量設計に明確な方法論等はなく，本講座で紹介した，よくわかるパターン認識[1]のコラム“特徴量に王道なし”にも述べられているように，対象となる問題を深く知る以外にない．ただし，先行研究を参考にしたり，不変量などの基本的な考え方，画像や音声，言語といった各分野で培われた知見を使ったりすることは重要である．また，深層学習で表現学習と呼ばれる方法では，特徴量の設計が不要になるとされる場合もあるが，只データとモデルがあれば良いのではなく，データの与え方や損失関数の設計が特徴量設計の代わりとなっている場合も多い．本稿では，画像認識においてよく用いられる特徴量を紹介し，その後，特徴量設計の考え方が深層学習においてどのように用いられるかを解説する．

2．画像認識における特徴量

画像を対象とした問題では，さまざまな特徴量が用いられてきた．ここでの問題とは，画像に何が写っているかを認識するような典型的なパターン認識の問題だけでなく，画像間の対応付け問題なども含んでいる．

2.1　輝度勾配に基づく特徴量

画像を対象とした問題では輝度勾配に基づく特徴量が非常に多い．輝度勾配とは画像の縦または横方向の微分値である．実際には差分で表現され，代表的なsobelオペレータによる微分画像の例を図1に示す．輝度勾配に基づいた特徴量がよく用いられる理由として，一般的な画像は照明やシャッタスピード等の撮影条件によるバイアスが大きいことが考えられる．そしてこのバイアスの影響はシーンに対しておよそ均一であるため，近傍の画素との差分によって低減することができる．信号処理的には微分を行うことでノイズの影響が大きくなるが，近年の画像はセンサの性能

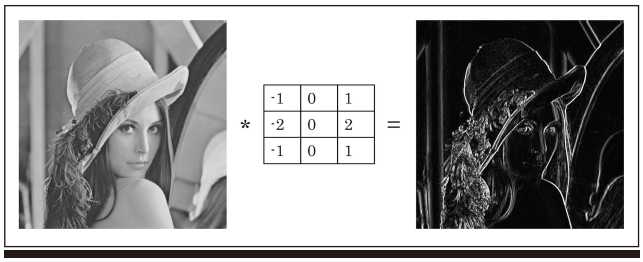


図1　Sobelオペレータによる微分（x方向）

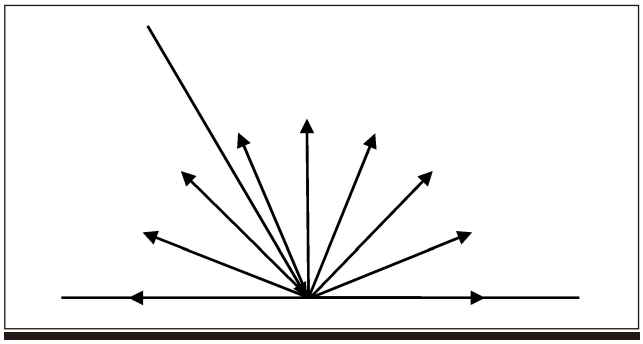


図2　Lambert反射

が良いため微分であっても影響が少ないものと考えられる．

このような輝度勾配を元に考えられた特徴量がHistogram of Oriented Gradient（HoG）である[2]．HoGでは，その名の通り，輝度勾配のヒストグラムを特徴量ベクトルとする．HoGについての解説は当会誌[3]で過去に行われている．

輝度勾配は上記のバイアスが取り除かれるだけでなく，さまざまな情報を持っている．物体表面の輝度は入射する光の方向や量によって変化する．この性質について，入射した光がすべての方向に等しく拡散して反射する，ランバート反射（図2）を仮定したとすると，ある点にはいる光線の密度は光線と面の角度によって変化する．実際の映像では四方八方から光が入射しており，物体の表面はランバート面ではないため，面の輝度から厳密に面の向きを計算することは難しいが，その輝度の変化が対象の幾何学的性質を強く表していることに代わりはない．例として，図3のような立方体に対して主に右上から光が当たっている状況を考えると，各面に照射される光の密度は，各面と光線方向によって決まり，光の方向と面の向きからおよそ図のようになる．これに対して輝度勾配を求めると，エッジと呼ばれ

† 大阪大学
"A Machine Learning Primer (4): Show off Your Skills in Designing Feature" by Tomohiro Mashita (Osaka University, Osaka)

映像情報メディア学会誌 Vol. 72, No. 5, pp. 732～735（2018）

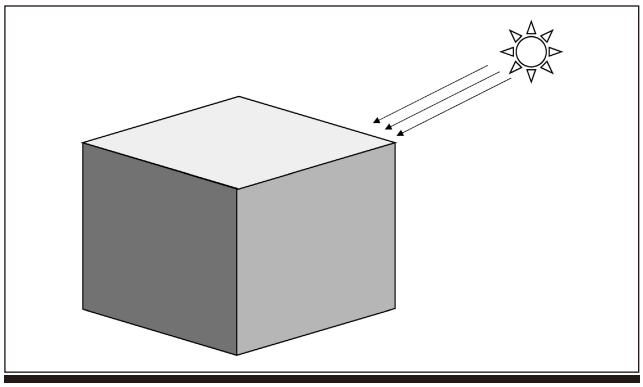


図3　物体の面の陰影

る輝度が大きく変化する部分が残る．すなわち，同じ面上の輝度は変化がないため輝度勾配がほぼ0となり面の継ぎ目で輝度が変化するため値が大きくなる．この輝度勾配が大きく変化するところがエッジであり，エッジには面の継ぎ目や背景との境界線などシーンの形状に関する情報が含まれている．この例の場合，特殊な例外を除くと光の当たる方向や強さが変化しても同様にエッジを求めることができ，光の当たり方の影響を取り除くことが可能になる．

エッジ情報は微分によって得られると述べたが，2階微分によっても得られる．図3の中央あたりの横一列の輝度値をプロットすると，**図4**(a)のようになる．これを1階微分すると，図4(b)になり，さらに微分を行うと図4(c)となる．2階微分によるエッジ検出の場合，値が正から負，負から正へと変化する点(ゼロ交差，Zero-cross)がエッジになる．2階微分の計算は通常，Laplacian Filterによって得られる．実際のアルゴリズムでは，Laplacian of Gaussian(LoG)やDifference of Gaussian(DoG)等によって計算される．DoGは後述するSIFT特徴[4]で用いられている．

2.2　局所特徴量

画像中のある点とその周囲の情報にだけ着目し，特定の処理を行うことで特徴量ベクトルを得るアルゴリズムの総称が局所特徴量である．局所画像特徴量や局所特徴量記述子と呼ばれたりすることもある．英語の場合はLocal FeatureやLocal Feature Descriptor, Local Image Descriptorなどと呼ばれる．なお，筆者は画像から特徴量を計算するアルゴリズムのことをLocal Feature Descriptorと認識しているが，人によっては，計算によって得られた特徴量ベクトルのことをLocal Feature Descriptorと呼ぶので注意が必要である．局所特徴量は，その名の通り画像中の特定の位置とその周囲(例えば3×3や16×16ピクセル)の領域から特徴量を計算する．狭い領域から得られる情報であるため画像中の1点に対する特徴量だけで何らかの画像認識するのではなく，画像中の多数の特徴量の値と位置関係などから物体認識を行ったり，2枚の画像間の対応付けを行ったりする．

局所特徴量は多数のアルゴリズムが提案されており，最も代表的なアルゴリズムは前節でも触れた，SIFT[4]やHOG[2]であろう．また，本講座の第1回で述べたHaarもその一つであると言える．その他にも，SURF[5]，ORB[6]，KAZE[7]，AKAZE[8]，LIFT[9]等，多数提案されている．詳しく知りたい方はサーベイ論文[10]や文献[11]を参考にすると良い．局所特徴量記述子でも上述の微分の考え方に基づいて，画素の差分を計算するものが多いが，連続する2点間の差分とは限らないものもある．その設計も用途によって異なり，例えば，ロボットや拡張現実で用いられる自己位置と環境地図の同時推定(Simultaneously Localization and Mapping: SLAM)を用途として設計されたものは処理が高速なものが多い．そのため，どのアルゴリズムが適切かは用途によっても異なる．

近年の局所特徴量では，機械学習によるもの[9]が提案されており，学習を用いない局所特徴量をHandcrafted feature descriptorと呼ぶ人もいる．機械学習を用いた特徴量は高性能であることを示す結果が得られているが，もちろん事前の学習が必要であり，またシーンや学習データに特化した学習を行うものがほとんどであるため，こちらも用途を良く考えて用いるべきである．

2.3　不変量

特徴量を設計する際，必要な情報をなるべく残し，それ以外の識別に悪影響を及ぼす要素を取り除く必要がある．その際に不変量(invariant)という考え方が重要になる．不変量とは，さまざまな要因によって異なる値になっている複雑な集合$x_i \in X$が，ある変換(写像)fによって得られる同じ値Yのことを言う．パターン認識の場合，識別性能を低下させる要素によってさまざまな値となっている入力が，特徴量抽出(写像)によってそれらを取り除き，識別に重要な要素を残すことが良い特徴量の設計となる．例えば，**図5**

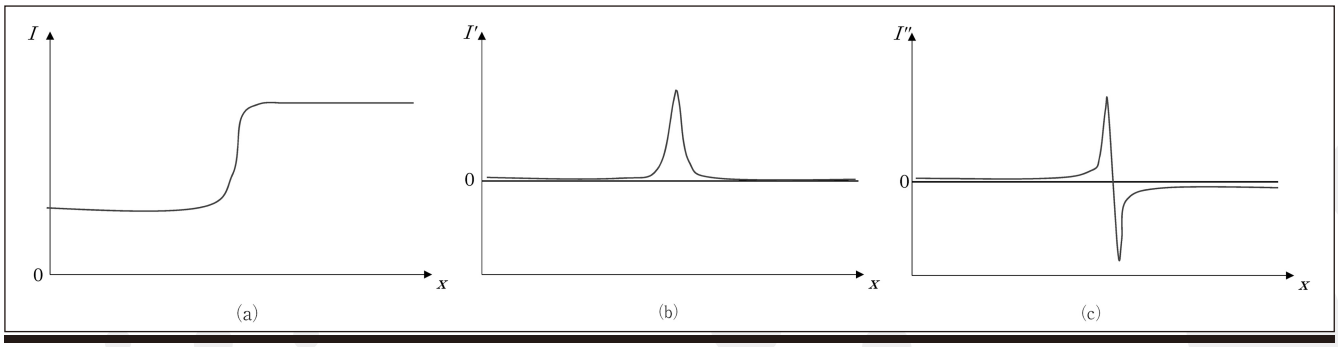


図4　微分によるエッジ検出

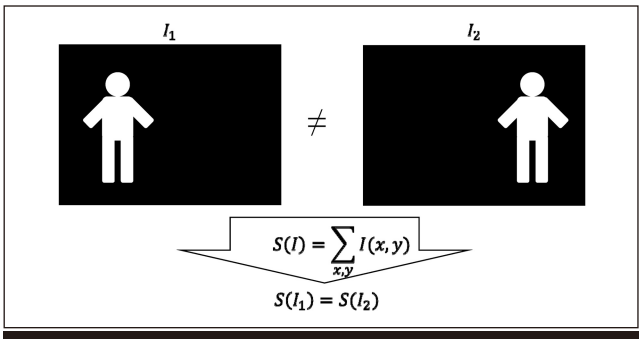


図5　不変量の例

のような画像中に人物が映っているかどうかを識別する問題を考えた場合，画像中のどの位置に映っているかによって画像（入力）は変化する．ここで，入力画像中の前景領域の面積を計算すると，対象物体の大きさが一定であれば，面積は物体の画像中の位置によらず一定の不変量である．このような値を用いることで，さまざまな入力に対して安定した処理を行う．

前述の輝度勾配も，シーン全体の輝度値にかかるバイアスに対して不変な値を求める写像と言える．この，シーン全体の輝度値に対するバイアスは，実際には撮影時の光量，光源方向，シャッタスピード，絞り等さまざまな要因で発生する．これらの影響を取り除くことができるため，輝度勾配は有用な特徴量となる．一方，暗室のようなこれらの要因を排除したり固定したりできる環境で撮影された画像には一般に用いる必要がなく，むしろS/N比を悪化させる要因となりえる．

不変量を得る方法はさまざまであるが，その一つは積分である．先述の対象領域の面積も積分によって得られる．対象となるデータがあるパラメータθ_1, θ_2で表現されており，観測は$g(\theta_1, \theta_2)$で得られていたとする．ここで，θ_2の影響をなくした値が欲しい場合，θ_2に関する積分，$G(\theta_1) = \int_{-\infty}^{\infty} g(\theta_1, \theta_2) d\theta_2$によって$\theta_2$に関する不変量$G(\theta_1)$が得られる．他にも，フーリエ変換やウェーブレット変換等による周波数解析，主成分分析（Principal Component Analysis, PCA），モーメント，曲率等さまざまな方法で不変量が得られる．図5の例における面積は図5中の式にあるように積分であり，0次モーメントと等しい．SIFTは回転不変，スケール不変になるように設計されており，またDoGに基づいているためシーンの輝度変化にも頑健とされている．

3．深層学習と特徴量

表現学習によって特徴量設計が不要になるという考え方は部分的には正しいと言えるが，設計の類がまったく不要になるわけではない．大量のデータを与えることによって特徴量設計の試行錯誤は少なくなったと言える．しかし，これまでのパターン認識で特徴量の設計によってなされていた，どのようなデータが入力され，どのような不変量が必要となりどのように出力を変化させるべきかという点については，教師データの与え方や損失関数の設計によって実現されている．

教師データの与え方によって入力の変化の影響を受けないようにあるいは受けるようにする方法の一つはデータ拡張（Data augmentation）である．データ拡張はその名の通り，データ数を擬似的に拡張させ，データ不足を補う方法として知られている．この方法では，画像の場合，画像の拡大や縮小，反転といった処理を加えたデータを与えることで，データの数を擬似的に増やす．そのため，データ数を増やすための方法であると思われがちであるが，上述の不変量の観点で考えると，モデルにどのような不変量を学習させ，求める出力を得るかを設計する段階であると言える．

データ拡張の例として細胞画像処理を考える．細胞画像は顕微鏡で撮影された画像であるため，画像の上下や回転に意味はなく，サンプルの向きによって偶然に決定される．また，おそらく細胞の左右を反転させたところでその画像の持つ意味が反転するようなことはないと思われる．このような場合，画像の回転や反転によってデータ数を増加させることができ，対応するラベルを元の画像と同じにすることで，回転や反転に対して頑健なモデルが学習される．しかし，その画像が特定の倍率で画像上の細胞のみかけの大きさに意味があり，細胞の大きさを識別する場合に有用な特徴となるならば，画像の拡大や縮小はすべきでない．

もう一つの例として，監視カメラによる人物の追跡のような問題を考えた場合，画像の上方向に頭があり，その下に胴体がありといった情報が重要であり，画像に上下の方向が存在すると考えられる．このような場合に，学習データとなる人物の画像を回転させる等のデータ拡張は行うべきでない．一方，人の身長や体格にはある程度のばらつきがあり，また撮影する距離によっても画像上の大きさは変化するため，その範囲内の拡大縮小などは有効かもしれない．そして，この与えられる入力とモデルの出力を評価する損失関数は合わせて設計される．すなわち，何らかの変化に対して影響を受けない出力が必要ならば，教師データはその変化に対して一定であり，逆にその影響を反映した出力が必要ならばそのような値を教師データとする．また損失関数も，求める出力とそうでない出力の違いをより正しく表現するものが望ましい．

このように考えると，多様な背景に対して影響を受けない出力が欲しい場合は，多様な背景を学習データと一定の出力となる教師データを用いれば良く，深層学習における特徴量の設計は与えるデータの設計となっていることがわかる．

損失関数の設計の例として，Simo-SerraとIizukaらによる，ラフスケッチから線画を生成する手法[12]をあげる．この手法では，損失関数にloss mapというヒストグラムを用いた関数を利用し，これによって出力の細線化に成功している．図6にSimo-SerraとIizukaらによる，loss mapの有無による出力の違いを示す．図から明らかなように，loss mapを用いた場合とそうでない場合では結果が大きく異

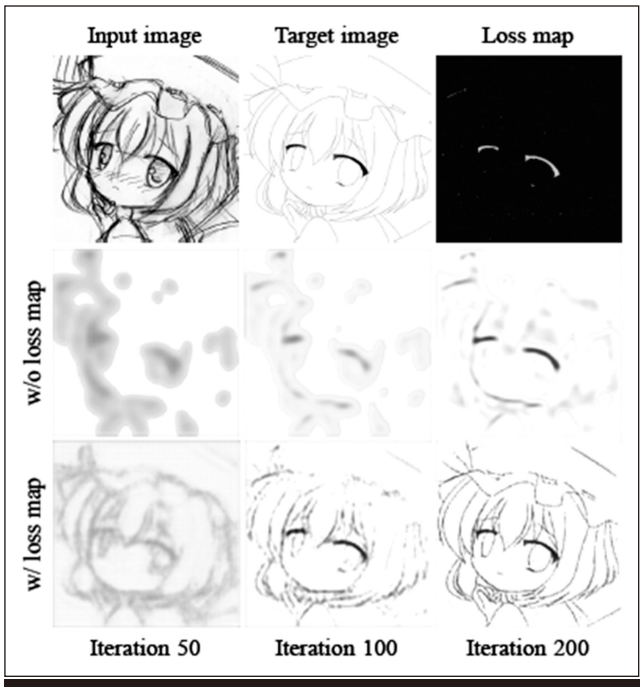


図6　損失関数の効果の例[12)]

なっており，loss mapを用いた損失関数によって求める出力（線画）とそうでない出力の違いを定義することに成功していると言える．

上述の例から，データ拡張を含むデータの与え方と損失関数がモデルの出力に大きな影響を与えることがわかる．そして，その設計には対象とする問題とデータについて，データに含まれる潜在的なパラメータや目的とする出力の定義や表現など，深い知識が用いられていることがわかる．このようなデータや問題に関する深い知識をモデルの学習に組み込むことがパターン認識において最も重要であり，この点において本講の最初に述べた“特徴量の設計が最も重要”と同じである．また，その設計方法に一般的な方法論がないことも同じである．このよう考えると，本講座の第1回のパターン認識アルゴリズムの分類（図7）について述べたように，人によって設計されていた特徴量抽出が深層学習に置き換えられたかのように思えるが，特徴量設計の本質である問題やデータに関する知識を計算機の処理に組み込むという作業の一部は学習データや損失関数の設計等に移ったと言える．正確には，ルールベースやこれまでの機械学習でも抽出される特徴や識別器の出力は学習データに依存していたが，深層学習においては学習データと損失関数の設計の重要性が増したと言える．

4．むすび

今回の講座では，画像を対象とした機械学習における特徴量について解説を行った．輝度勾配を用いた特徴と局所特徴量について解説を行い，特徴量設計の基本的な考え方の一つとして不変量についての解説を行った．特徴量についてはさまざまなものが考案されているため，より詳しく知りたい方は文献[11)]等を参考にされると良い．そして，特

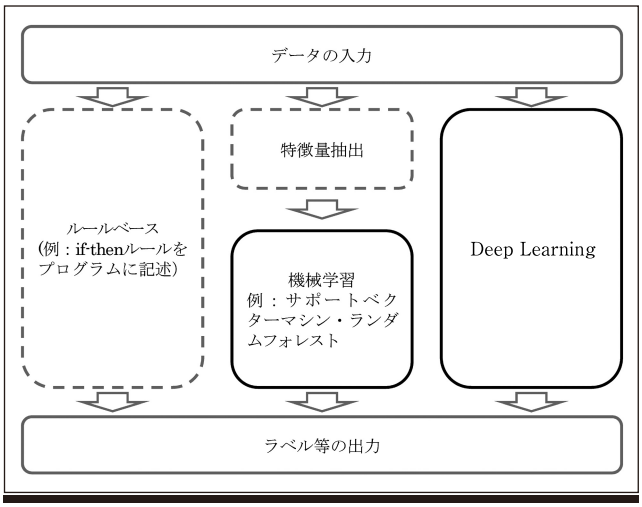


図7　パターン認識アルゴリズムの分類

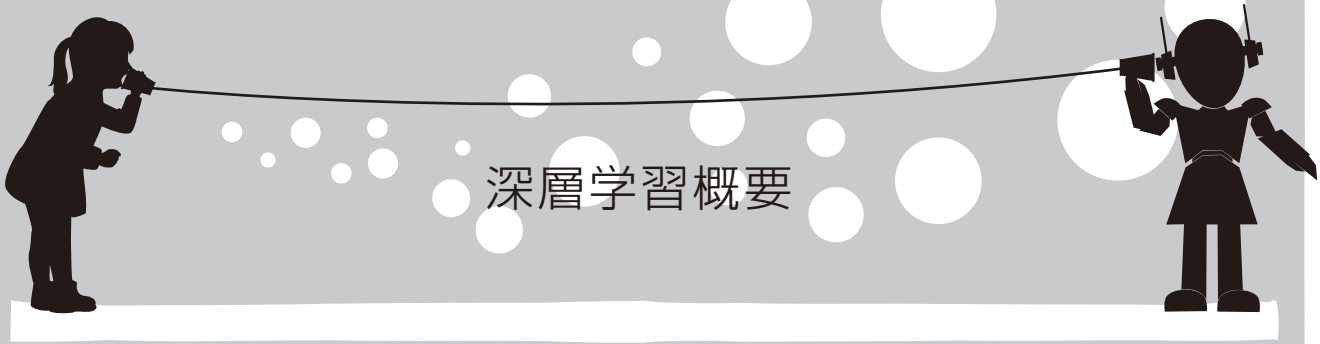
徴量設計の考え方が深層学習においてどのように適用されるかについて解説を行い，教師データの与え方やデータ拡張，損失関数の設計に対象の知識が用いられることを例を用いて示した．

（2018年7月27日受付）

〔文　献〕

1）石井健一郎，前田英作，上田修功，村瀬　洋：“わかりやすいパターン認識”，オーム社（1998）
2）N. Dalal and B. Triggs: "Histograms of oriented gradients for human detection", Computer Vision and Pattern Recognition, CVPR 2005, IEEE Computer Society Conference on. 1. IEEE（2005）
3）山崎俊彦：“画像の特徴抽出：Histogram of Oriented Gradients（HoG）”，映情学誌，64，3，pp. 322-329（2010）
4）D.G. Lowe: "Object recognition from local scale-invariant features", Computer vision, The proceedings of the seventh IEEE international conference on. 2. IEEE（1999）
5）H. Bay, T. Tuytelaars and L.V. Gool: "Surf: Speeded up robust features", European conference on computer vision, Springer, Berlin, Heidelberg（2006）
6）E. Rublee, et al: "ORB: An efficient alternative to SIFT or SURF", Computer Vision（ICCV），2011 IEEE international conference on. IEEE（2011）
7）P.F. Alcantarilla, A. Bartoli and A.J. Davison: "KAZE features", European Conference on Computer Vision. Springer, Berlin, Heidelberg（2012）
8）P.F. Alcantarilla and T. Solutions: "Fast explicit diffusion for accelerated features in nonlinear scale spaces", IEEE Trans. Patt. Anal. Mach. Intell 34.7, pp.1281-1298（2011）
9）K.M. Yi, et al: "Lift: Learned invariant feature transform", European Conference on Computer Vision. Springer, Cham（2016）
10）T. Tuytelaars and K. Mikolajczyk: "Local invariant feature detectors: A survey", Foundations and trends® in computer graphics and vision 3.3, pp.177-280（2008）
11）S. Krig: "Computer vision metrics", Springer（2016）
12）E. Simo-Serra, et al: "Learning to simplify: fully convolutional networks for rough sketch cleanup", ACM Transactions on Graphics（TOG）35.4, p.121（2016）

間下　以大（ました　ともひろ）　大阪大学大学院基礎工学研究科博士後期課程修了．現在，同大学サイバーメディアセンター准教授．2012年，オーストリア・グラーツ工科大学客員研究員．コンピュータビジョン，機械学習，拡張現実に関する研究に従事．博士（工学）．正会員．



深層学習概要

正会員 間下以大†

１．まえがき

第4回までの講座で，機械学習における識別器と特徴量設計と深層学習の関係について解説を行ってきた．5回目となる本講座では深層学習の基本を理解すべく，ニューラルネットの基本と代表的なネットワークアーキテクチャについて解説する．

２．ニューラルネットの基本

良く知られているようにニューラルネットは，生物の神経細胞（ニューロン）の結合を模して作られた技術である．その基本構成（ユニット）は非常にシンプルであり，図1に示すような幾つかの入力$x_1, \ldots, x_N$に対して重みwをかけて足し合わせ，その結果であるuを入力とした関数$f(u)$の値zを出力する．関数fは活性化関数（activation function）と呼ばれる．また，uの計算にはバイアスbが加えられている．そしてこのユニットの出力が次のユニットの入力となり，さらにその出力が次の入力となり，と繰り返して最終的な出力を得る．

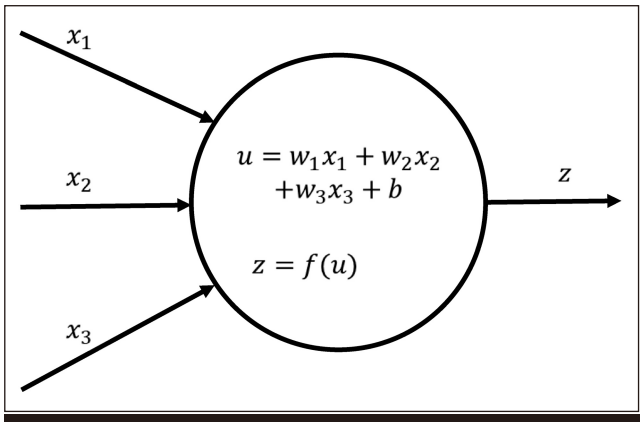


図1　ニューラルネットのユニット

活性化関数には単調増加する非線形関数が用いられ，代表的なものは，ロジスティックシグモイド関数（logistic sigmoid function），双曲線正接関数（hyperbolic tangent function, tanh），正規化線形関数（rectified linear function）等である．各関数の形を図2に示す．近年では，計算量の小ささや結果が良いことから正規化線形関数がよく使われている．正規化線形関数を用いたユニットはReLU（Rectified Linear Unit）と呼ばれる．また，回帰問題等では出力層の活性化関数に恒等写像を用いる場合もある．

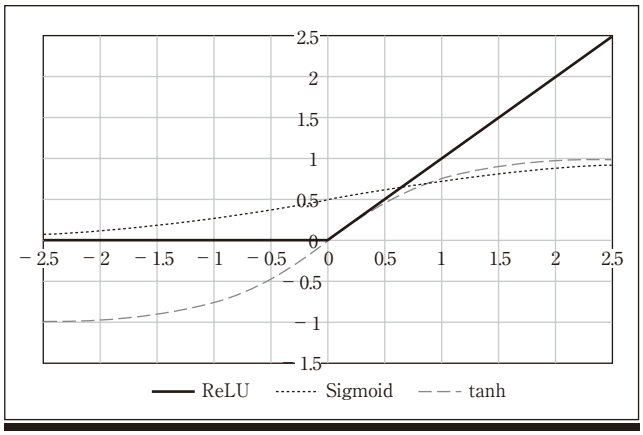


図2　代表的な活性化関数の形状

対象とする問題が多クラス分類問題の場合，最終層はSoftmax関数が用いられる．この場合，最終層Lのユニット数は識別するクラスの数Kと同じにし，Softmax関数は最終層の各ユニットの出力u_j^Lと次式によって出力y_kを決定する．

$$y_k = \frac{\exp(u_k^L)}{\sum_{j=1}^{K} \exp(u_j^L)}$$

この関数では，$\sum_{j=1}^{K} y_j = 1$となり，最終的な出力の値を確率として扱うことができる．

2.1　順伝搬型ネットワーク

最も基本的な構成として，順伝搬型ネットワークの例を図3に示す．図3では左から右へと伝搬する構成になっており，一番左側を入力層（input layer），一番右側を出力層（output layer），中間を隠れ層（hidden layers）または中間

† 大阪大学
"A Machine Learning Primer (5): A Brief Overview of Deep Learning" by Tomohiro Mashita (Osaka University, Osaka)

映像情報メディア学会誌 Vol. 72, No. 6, pp. 900～904（2018）

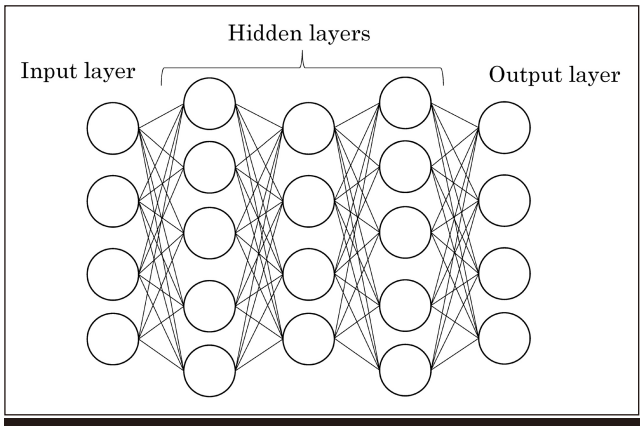


図3　順伝搬型ネットワークの例

層（internal layers）と呼ぶ．図3の例は順伝搬型ネットワークの中でも多層パーセプトロン（multilayer perceptron, MLP）と呼ばれるものである．前述のように，ユニットの構成は非常に単純だが，層を重ねることで複雑な入力と出力の対応関係（写像，map）を構成することが可能になる．第2回で述べたパターン認識のアルゴリズムによって“線を引く”や第4回で述べた特徴量抽出はこの写像に含まれる．そのため，特徴量抽出とその後の識別や回帰の処理をすべてニューラルネットで構成することができ，表現学習やend-to-endといった学習が可能になった．一方で図の各ユニットを繋ぐエッジ毎に重みパラメータが存在し，入力層を除く各ユニットにバイアスパラメータが存在するため，非常に多数のパラメータを学習によって決定する必要がある．

ニューラルネットのパラメータは，他の機械学習と同じくデータを用いて決定される．ニューラルネットの場合，損失関数（loss function）によって出力を評価し，その勾配に基づいて損失関数の誤差を逆伝搬させることでパラメータを更新していく．

誤差の逆伝搬には通常，勾配降下法が用いられる．勾配降下法は他の2次微分を利用する方法等とくらべて収束が遅いとされているが，深層学習の場合，2次微分の計算自体が困難であるため勾配降下法が用いられる．実際には，大量の学習データすべてを用いて勾配を計算するのではなく，その一部を使って勾配を計算する，確率的勾配降下法（Stochastic Gradient Descent: SGD）が用いられる．

このように深層学習は，単純なユニットの組み合わせと勾配降下法による学習という比較的単純な技術が基本となっているため，機械学習の中でも理解し易い技術といえる．しかし，深層学習ではパラメータが非常に多いことから，求める性能が得られる前に局所解に陥ったり，勾配がなくなったりすることによってパラメータの更新が行われなくなることがある．深層学習ではこのような状態を回避し，より効率的な学習を行う方法が多数存在する．実際の問題に適用する場合，ネットワークの構造や学習方法の多数の選択肢の中から対象となる問題に適切なものを探る必要がある．

3．代表的なネットワークアーキテクチャ

深層学習で用いられるニューラルネットのアーキテクチャには代表的なものが幾つかある．先述のMLPは最も基本的なアーキテクチャと言える．これ以外にも，自己符号化器（Autoencoder: AE），畳み込みニューラルネット（Convolutional Neural Network: CNN），回帰結合型ニューラルネット（Recurrent Neural Network: RNN），第3回で解説した敵対的生成ネットワーク（Generative Adversarial Network: GAN）等である．

3.1　自己符号化器（AE）

自己符号化器は，中間層で一度ユニット数が少なくなってから入力と同じユニット数になるような構造をしている．そして最大の特徴は出力の評価に入力となる値を用いて学習を行うことである．最も単純な例を図4に示す．図のように，出力された$\tilde{x}$は損失関数で入力と比較し，その誤差が最小になるように学習が進む．中間層は入力よりも少ないユニット数であるため，中間層の出力は入力データの情報をなるべく保持したまま次元を削減した値となる．自己符号化器はこの性質を利用して事前学習に利用されたりする．図の例では損失関数を二乗誤差としており，この場合，中間層の出力は主成分分析と実質的に同じものが得られることが知られている．他の応用例として，xにノイズを加えてから入力し，ノイズのないxで評価することで，ノイズ除去の効果を得られるように学習させる方法などがある．

3.2　畳み込みニューラルネット（CNN）

畳み込みニューラルネットは，時系列のデータや画像のような格子状のデータといった，各要素の並びに重要な情報が含まれているデータに対して用いられるアーキテクチャである．日本語においてもCNNと呼ばれることも多い．畳み込みニューラルネットの基本的なアーキテクチャ

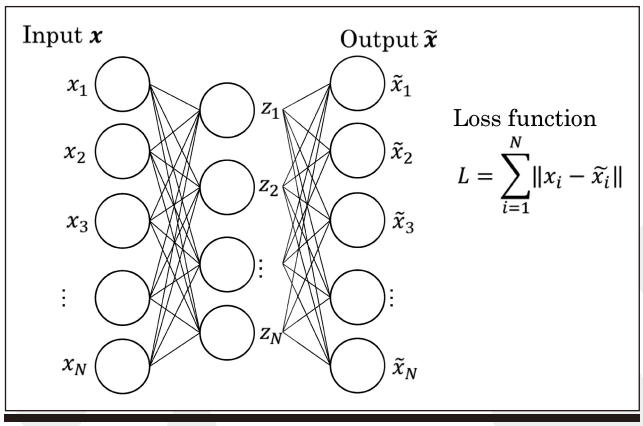


図4　自己符号化器の例

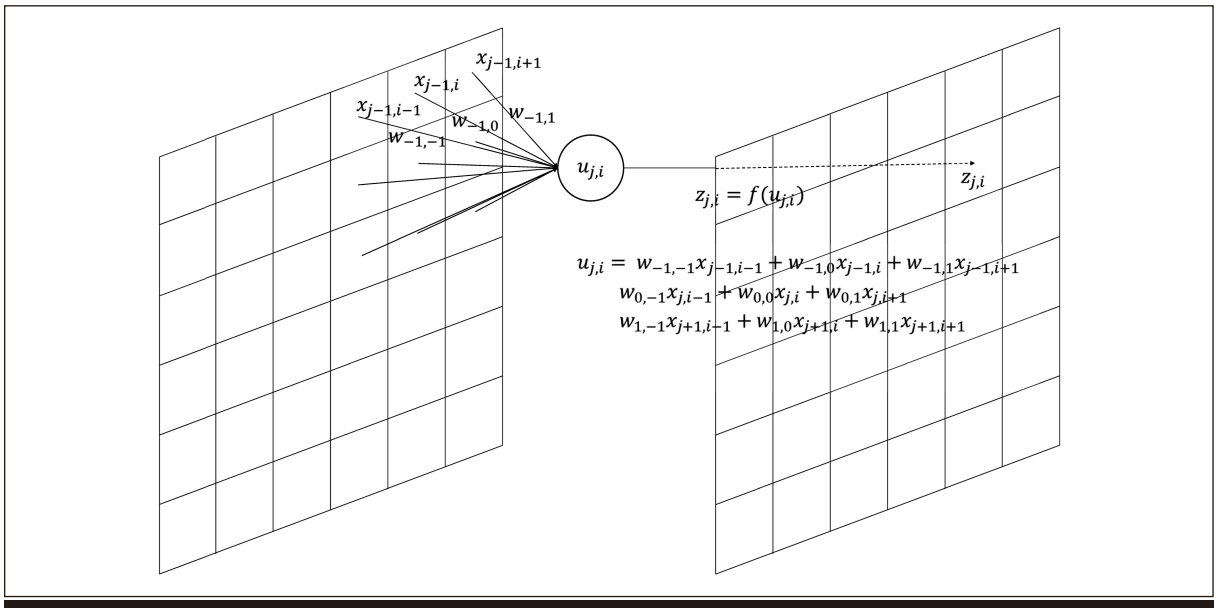


図5　畳み込みニューラルネット

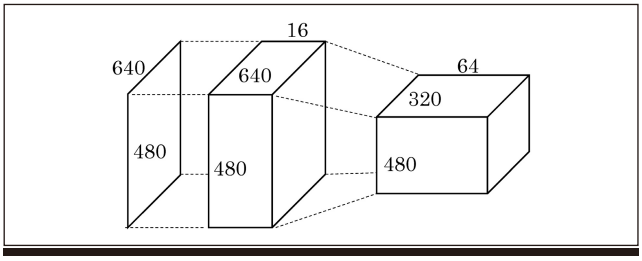


図6　畳み込みニューラルネットのモデルを表す図の例

は，図5のような構造であり，格子状にならんだ画素に対してその名の通り畳み込みと同じ処理を行う．この処理は画像処理のフィルタリングと同様にすべての画素に対して同じ重みパラメータを用いて行われる．すなわち一つの畳み込みニューラルネットによって1枚の格子状の入力から別の格子状に並んだ出力が得られる．畳み込みニューラルネットを用いた論文では図6に示すような1チャネルの画像入力（一番左の640 × 480）に対して，多数のチャネルが生成されているような図が用いられる．このような図では，出力されるチャネル数分の異なる重みを学習する畳み込みニューラルネットが用いられていることを意味する．すなわち，図6の入力画像に対する処理では，16種類の畳み込みニューラルネットが用いられている．2段階目の処理では，畳み込みの適用を1画素おき（ストライドと呼ばれる）にすることで，出力サイズが半分になっている．畳み込みニューラルネットではさらにプーリング（pooling）と呼ばれる処理も用いられる．代表的なmaxpoolingでは入力となる2 × 2といった範囲の中の最大値を出力する．畳み込みニューラルネットはこの畳み込み層とプーリング層によって，位置の変化に対して不変になる性質を持っていると言われている．

3.3　回帰結合型ニューラルネットRNN

回帰結合型ニューラルネットは長さが変化する系列を扱うニューラルネットである．長さが変化する系列とは，音声，動画像，言語等が該当する．音声データの場合，同じ文章を読み上げたとしてもそれぞれのデータによって読み終わるまでの時間が異なるし，動画像の場合も同様である．また自然言語処理の場合，文や単語の長さが異なる．静止画像の場合，画素数や解像度，倍率等が異なることはあっても，一定のサイズに正規化することは可能であり，極端な正規化でなければ，そこから必要な情報を読み取ることは可能である．しかし，音声や動画像の長さを無理矢理一定にしてしまうと，周波数等の情報が変化する．また，言語のようなデータの場合，すべての文を50文字に正規化するといったことはできない．

系列データは順序のついたデータ$\{x_1, x_2, \ldots, x_t\}$として扱う．ここで，長さが変化する$T$とはの値が変化することであり，各$x_t$の次元は一定である．動画像の場合各フレームが$x_t$に該当し，動画像の長さが$T$になる．

回帰結合型ニューラルネットの基本的な構造は図7のようなものであり，入力として各x_tが順に与えられる．そして，最大の特徴が中間層の出力h_tと次の入力x_{t+1}を合わせて中間層の入力とすることである．このような処理は回帰構造を展開して考えることができ，図8のようになる．

回帰結合型ニューラルネットは長さが変化する系列データを扱うことができるが，長期依存姓と呼ばれる，tの値が離れたデータの依存性を学習できない問題があった．この問題に対応したのがLong Short Term Memory（LSTM）と呼ばれるネットワークである．LSTMには中間層に幾つかのゲートと呼ばれるユニットを加える工夫がなされている．LSTMについての詳細は各書籍等[10) 11)]を参考にして頂

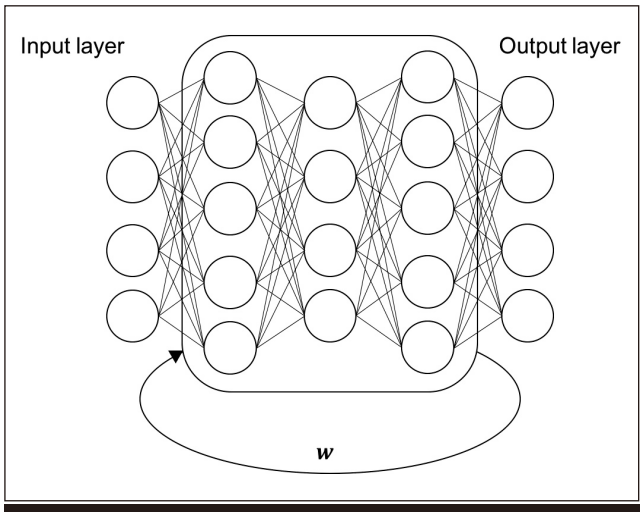


図7　回帰結合型ニューラルネットのアーキテクチャ

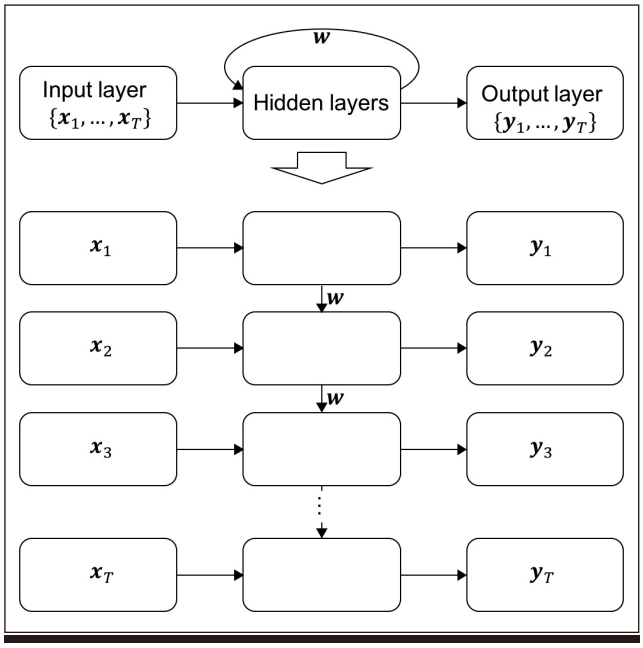


図8　回帰結合型ニューラルネットの展開

きたい．

回帰結合型ニューラルネット（RNN）は，先述の通りRecurrent Neural Networkの訳であるが，日本語では再帰型ニューラルネットと呼ばれる場合もある．紛らわしいことに，Recursive Neural Network（再帰型ニューラルネット）というものもある．こちらもRNNと同様に時系列データを扱うものであり，再帰ツリー状に展開することができる．そして，両者の日本語訳は回帰（Recurrent）と再帰（Recursive）が入れ代わっている場合もある．良く用いられているのはRecurrent Neural Networkの方ではあるが，念のため注意が必要である．

4．深層学習後のパターン認識

深層学習の応用例をみてみると，これまでの機械学習で行われてきた特徴量抽出と識別器の組み合わせという設計を踏襲している例が非常に多いことがわかる．例えば，画像認識の場合，後述する畳み込みニューラルネットによって特徴量抽出を行い，出力層とその直前の層で多層パーセプトロンによる回帰やsoftmaxによるクラス識別を行うという設計がなされている．具体例を挙げると，GoogleNet[1]等に代表される物体識別を行うニューラルネットのほとんどはそのような構成となっている．また，カメラの位置姿勢推定を行うPoseNet[2]では，GoogleNetを画像の特徴量抽出のために利用しており，GoogleNetでsoftmaxが用いられている部分を全結合層（1層のパーセプトロン）に置き換えることで回帰による位置姿勢推定を行っている．PoseNetの例のように，モデル内最終層付近の多層パーセプトロンを明示的にRegressorと呼ぶ例も見られる．

すでに述べたようにディープニューラルネットはシンプルな関数を多数組み合わせることで写像を行う関数と言える．つまり深層学習とは，極端な言い方をすれば，計算機上で表現される何かの数値データから別の何かの数値データへの変換を行う関数を求める仕組みである．このことから，変換前と変換後のそれぞれの集合をある程度満たすデータと計算資源があればその変換が得られるようになったと言える．そのため，従来のパターン認識では，対象を理解して特徴量を設計し，適切な識別器で学習を行うのが定石であったが，深層学習の登場後は，対象となる問題に対して充分な計算資源と教師ありデータを用意してニューラルネットに学習させるのが定石となった．

この変化の結果，問題に対する設計者の知識を反映させる点が特徴量設計からデータの与え方と損失関数の設計に変化したと言える．第4回でも述べたように，かつてのパターン認識の実用的な問題に対する研究では特徴量設計に工夫されることが多かったが，現在はデータの与え方や学習の手順，損失関数の設計などが工夫されることが多い．ただし，その工夫を行うためには対象（データ）に対する知識が重要であることは特徴量設計と変わらない．

また，上記の定石がそのまま適用できるような問題，つまり，

- 公開されているデータセットをそのまま問題に利用することができる等，教師つきデータが容易に入手できる．
- 損失関数やデータの与え方等に工夫を加える必要がない．
- ネットワークの構造が典型的なもので充分な性能が得られる．

という問題の場合，よほど研究対象が新規なものでもない限り，研究としての新規性は低くなったと言える．逆にこれらの3点が画像認識のような応用に近いパターン認識での典型的な問題設定になった．そのため，すでに述べたよ

うに，個々の問題に関しては損失関数やデータの与え方に工夫がなされるようになった．ネットワークの構造については，Resnet[3]やDensenet[4]といった効率的で性能の良いネットワークに関する研究が注目されるようになった．データに関しては，データセットの提案[5]や転移学習[6]，ドメイン適応[7]，データ生成[8]のように，データ不足に貢献できる研究が注目されている．

先述の写像という観点では，従来ではあまり行われなかった機械学習の使われ方も増えた．その一つが物理シミュレーション等を機械学習で置き換える手法[9]である．精密な物理シミュレーションには多くの計算資源と時間が必要とされる場合が多いが，事前にシミュレーションの結果を多数用意し，ニューラルネットに学習させておくことで高速に出力を得ることができる．ただし，教師データによって写像を得るという機械学習の性質上，機械学習によって処理の高速化は可能だがシミュレーション性能の向上はできない．また，教師データから大きく外れた設定に対する性能は大きく低下する可能性が高いと言える．そのため用途は専ら高速化となる．

第3回で紹介したGANは深層学習によって現れた，新しく最も興味深い機械学習の考え方の一つである．GANの応用，派生による興味深い研究は現在も次々と発表されており，今後も注目が必要である．

深層学習はすでに述べたように，非常にシンプルな構造が基本となっている．そのため，従来の機械学習に比べて技術習得までの時間が大きく短縮された．また，ライブラリー等も充実しており，以前と比べると新たに機械学習に取り組む人にとってよい条件が整っている．そのため，人工知能分野に多くの人と企業が流入しており，本講座で多数引用しているCVPRでは，6,000人と急速に参加者数が増えている．また，本稿執筆時にNIPS2018の参加登録が11分で売り切れになったという情報がtwitterで話題になっていた．自明ではあるが，このような開発環境の整備と人工知能分野の爆発的な拡大が最も大きな変化である．

5．むすび

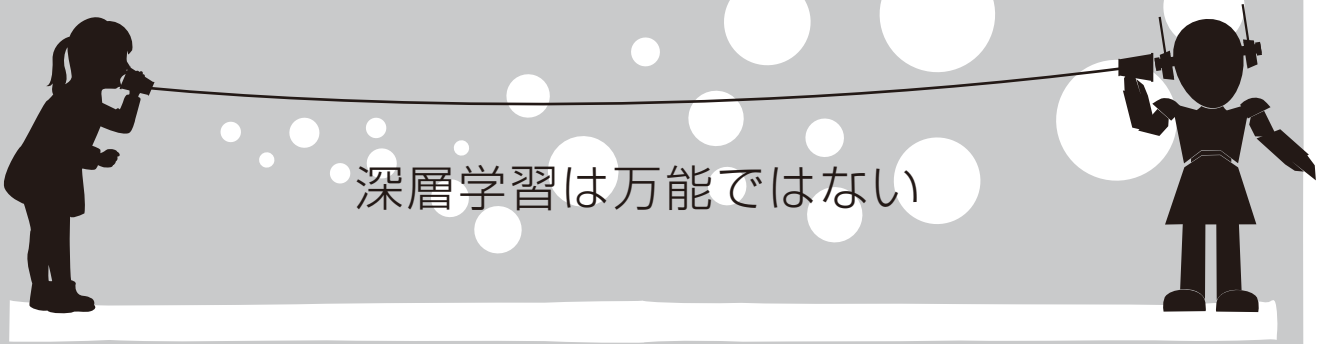
本講座では，ニューラルネットの基本と深層学習で用いられるアーキテクチャの特徴や使われ方について解説を行い，その後深層学習の登場による変化について述べた．ニューラルネットの基本や各アーキテクチャについては，これまでも本講座で紹介してきた，岡谷による“深層学習”[10]や，Goodfellowによる“Deep Learning”[11]に基づいているため，詳細な説明や数学的説明についてはこれらを参照されると良い．

（2018年9月6日受付）

〔文　献〕

1) Szegedy, Christian, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich: "Going deeper with convolutions", in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp.1-9（2015）
2) A. Kendall, M. Grimes and R. Cipolla: "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization", Proceedings of the International Conference on Computer Vision, pp.2938-2946（2015）
3) K. He, X. Zhang, Sh. Ren, J. Sun: "Deep residual learning for image recognition", in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp.770-778（2016）
4) G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger: "Densely Connected Convolutional Networks", in proceedings of IEEE Conference on Computer Vision and Pattern Recognition（CVPR）, 1, 2（2017）
5) T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, T. Pajdla: "Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions", The IEEE Conference on Computer Vision and Pattern Recognition, pp.8601-8610（2018）
6) A. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, S. Savarese: "Taskonomy: Disentangling Task Transfer Learning", in Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition, pp.3712-3722（2018）
7) K. Saito, K. Watanabe, Y. Ushiku, T. Harada: "Maximum Classifier Discrepancy for Unsupervised Domain Adaptation", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.3723-3732（2018）
8) A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, R. Webb: "Learning from Simulated and Unsupervised Images through Adversarial Training", in proceedings of IEEE Conference on Computer Vision and Pattern Recognition（CVPR）, 2, 4, p.5（2017）
9) J. Tompson, K. Schlachter, P. Sprechmann, K. Perlin: "Accelerating Eulerian Fluid Simulation with Convolutional Networks", Proceedings of the 34th International Conference on Machine Learning, PMLR 70:3424-3433（2017）
10) 岡谷貴之“深層学習”，講談社（2015）
11) I. Goodfellow, Y. Bengio and A. Courville: "Deep Learning（Adaptive Computation and Machine Learning series）”, The MIT Press（2016）

間下　以大（ました　ともひろ）　大阪大学大学院基礎工学研究科博士後期課程修了．現在，同大学サイバーメディアセンター准教授．2012年，オーストリア・グラーツ工科大学客員研究員．コンピュータビジョン，機械学習，拡張現実に関する研究に従事．博士（工学）．正会員．



深層学習は万能ではない

正会員 間下以大†

1．まえがき

第1回から第4回までの講座では機械学習の基本的な考え方について，第5回の講座では深層学習の基礎とよく使われるアーキテクチャについて，さらに深層学習の登場後にパターン認識・機械学習がどのように変化したかについて解説を行った．最終回である第6回では，“深層学習は万能ではない”とし，深層学習が利用できる場合とそうでない場合や，現在研究の最先端で取り組まれている問題について，本講座第1回～第5回を振り返りつつ議論したい．

2．パターン認識，機械学習の定石

第5回で解説したように，深層学習の登場後パターン認識・機械学習ではディープニューラルネットの学習を大量のデータと計算資源によって行うことで解決するという方法が定石となった．この定石が適用できる条件は，

(1) 公開されているデータセットをそのまま問題に利用することができる等，教師付データが容易に入手できる
(2) 損失関数やデータの与え方等に工夫を加える必要がない
(3) ネットワークの構造が典型的なもので充分な性能が得られる

である．この，データ，損失関数，モデル，に計算資源を加え，それぞれについて深層学習が適用できる場合とそうでない場合について解説する．

2.1 データ

ある問題を深層学習で解決する場合，その問題の入力となるデータが存在する空間と出力となるデータが存在する空間をある程度満たす必要がある．

ここでいう空間とは，例えば，画像や音声といった信号が量子化されたものであり，画像の場合は画素数がその次元数にあたる．このことについては第1回で述べた．このデータは，第1回の例のように128 × 128=16,384 pixelのグレースケールという，画像としては非常に低解像度なものでも16384次元という広い空間であり，画像が取り得る値すべてを埋めるデータを考慮するというのは現実的でないし，その多くは意味を持たないノイズのようなものである．すなわち実際の問題では対象とするデータは偏りをもって存在するため，上記のデータが存在する空間をある程度埋めるとは，その偏って存在する空間を埋めるという意味になる．

解決すべき問題があり，学習用の教師付データがあったとすると，そのデータが充分かどうかを判断する必要がある．この判断を行うには，対象としている問題のデータの広がりや複雑さを見積もる必要がある．ここで必要となるのが，その問題に関する知識である．このことも第1回で述べた．この問題の広がりや複雑さの予測は非常に難しく，人工知能の父として有名なMinsky教授は当初，“コンピュータビジョンを大学生の夏休みのプロジェクト程度の難しさと考えていた”という逸話がある．実際にはコンピュータビジョンが人工知能における大きな分野の一つであり，盛んに研究が続いている問題であることは言うまでもなく，この逸話からも問題の複雑さを見積もることの難しさがわかる．この問題の複雑さは深層学習においても同様に重要な要素であり，後述するモデルのキャパシティの問題として扱われる．

また，問題の複雑さとデータの関係についてはデータの数よりも密度で考える方がよく，また出力となる教師付データの複雑さも関係する．すなわち，高次元で複雑なデータから複雑な出力を得たい場合にはより高密度のデータが必要であり，比較的単純な問題の場合には必要なデータの密度も少なくて良い．これを図示すると図1のようになる．第2回で解説した機械学習によって線を決定するという考え方に基づくと，複雑な曲線を描くには多数の学習データと多数のパラメータが必要であり，少数の学習データや少数のパラメータでは比較的単純な曲線になる．具体例として，顔画像認識を考えてみると，顔を検出するだけの人工知能と顔検出に加えて表情を識別する人工知能の場合では，問題の複雑さが異なり，必要となるデータの密度も異なる．当然，後者ではさまざまな表情の入力データと

† 大阪大学
"A Machine Learning Primer (Final study): Deep Learning is not universal" by Tomohiro Mashita (Osaka University, Osaka)

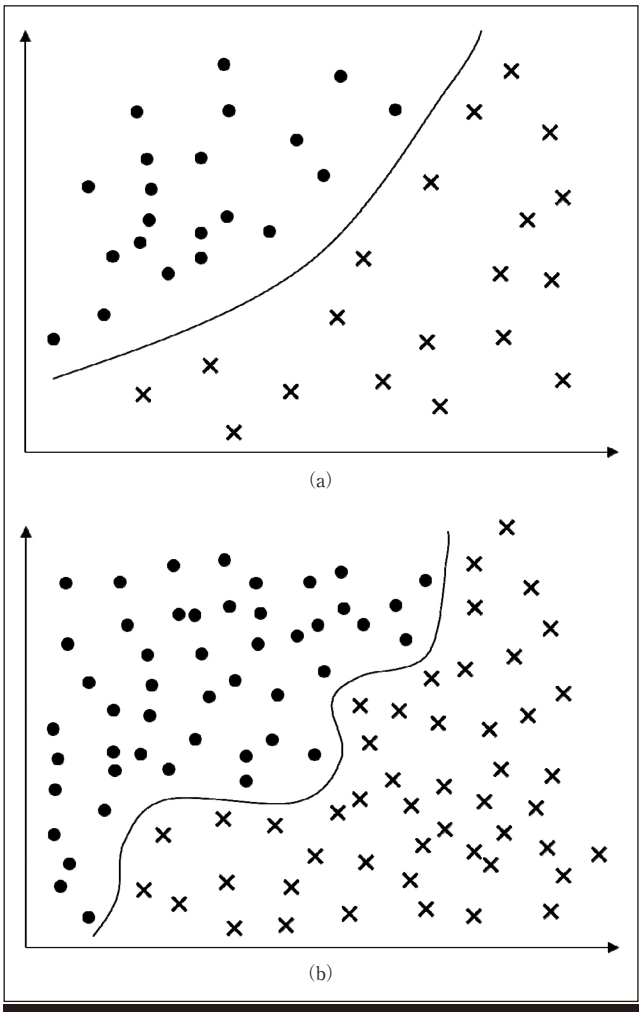


図1　点の密度によって異なる決定境界の例

それに対応する教師データを用意する必要があり，複雑な問題になる．

データを用意する際に忘れられがちなのが，データのサンプリングである．機械学習に用いられるデータのサンプリングに偏りがある場合，学習結果もその偏りを反映したものになる．このことは，本講座第3回で少し述べた．図1の例で，本来得るべき理想的なサンプリングが図1(a)であったとし，そのサンプリングが何らかの理由で偏った結果，図2の破線で示したサンプルが得られなかったとする．その場合，得られる決定境界は図2の二重線のようになり，実際の運用において破線で示したサンプルが入力された場合，誤識別となる可能性が高い．先ほどと同じく顔認識の例で考えると，何らかの理由で黒髪の人物ばかりで学習を行うと，金髪等の人物は検出できなくなるといった事態が考えられる．このデータの偏りは識別アルゴリズムの設計や学習の過程で気付くのが難しい．このような問題に気付くには，実際のデータをよく観察したり，学習したモデルを実際に運用してその結果を詳細に観察したりする必要があり，またデータの準備段階から充分に注意する必要がある．

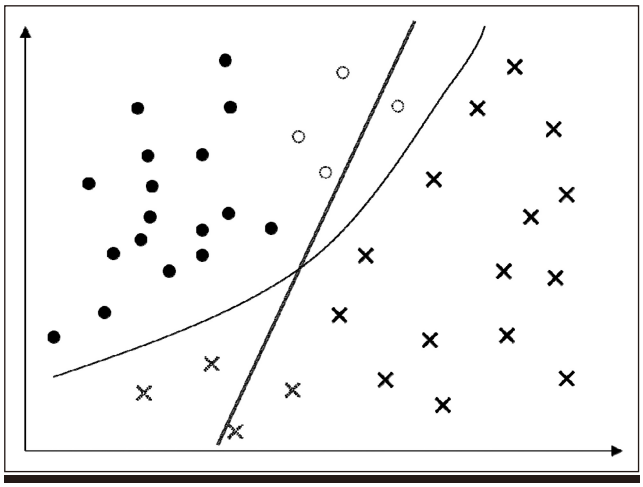


図2　サンプリングが偏った場合の例

データを用意するコストも深層学習を用いる際の障害の一つである．インターネット等で公開されているデータセットを利用できない場合，自らデータセットを用意する必要がある．しかし，データセットの作成は非常にコストのかかる作業であり，特に深層学習では大量のデータを用意する必要があるため，そのコストも比例して大きいものになる．データセットの作成コストが問題になる場合は，サポートベクタマシン等の比較的少量のデータでも性能が期待できる手法の利用も検討してみるべきである．また，近年では多種多様なデータセットが用意されているが，目的を少し変えた人工知能を開発しようとすると，そのデータセットが上手く適用できないといったことが起きる．そういった場合はやはり自らデータセットを用意するか，シミュレーションや他のデータセットによる学習を目的とする学習に適応させる，ドメイン適応(Domain Adaptation)といった方法が必要となってくる．なお，ドメイン適応は近年の最先端の深層学習に関する研究でよく議論されているトピックの一つである．

2.2　損失関数

損失関数の設計は，データの性質が直接表現されることから深層学習以前の特徴量の設計に該当する部分があり，出力されるデータの性質をどのように定義づけるかといった意味がある．これについては第4回で述べた．深層学習の場合は二乗損失等の代表的な損失関数を用いることが多く，それらは出力されたデータをすべて均等に扱っている．しかし，実際の問題では均等に扱うべきでない場合も多い．例えば，出力された画像の誤差は各画素の差の二乗和(Sum of Squared Difference: SSD)として扱われることが一般的であるが，画像から抽出したい特定の情報や領域と背景等のその他の情報や領域を同様に扱って良いかは問題による．また，出力された画像が人に見せることを目的としているならば，人に知覚されやすい周波数や領域を重視して損失関数を設計するというのも一つの戦略である．損

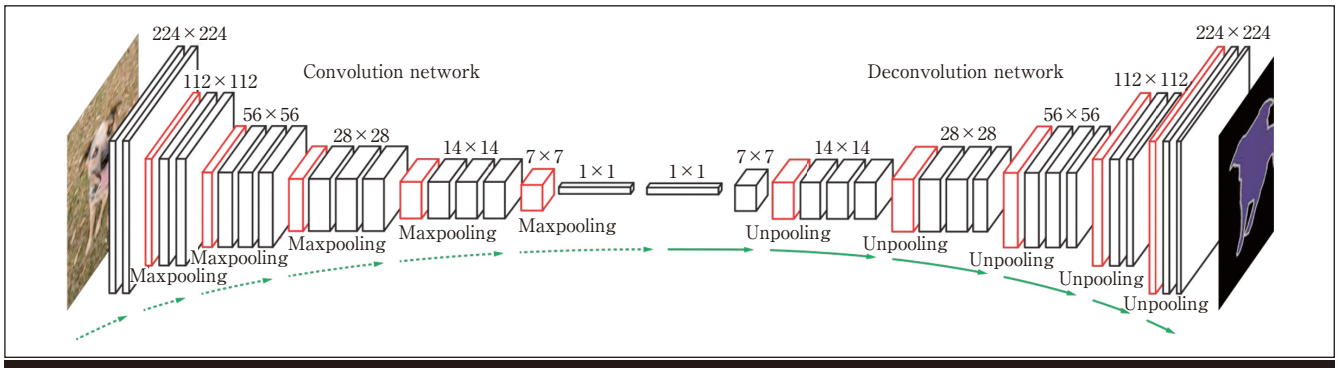


図4　Conv-deconv型ネットワークの例[2)]

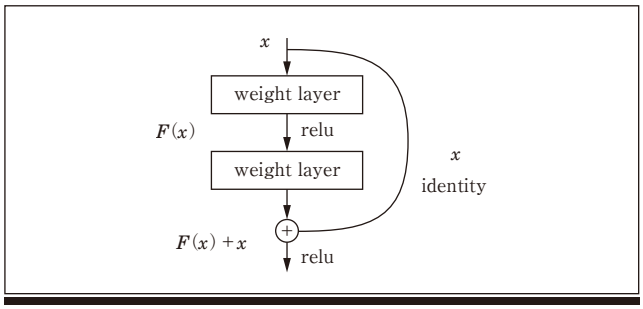


図3　ResNetの基本構成となるbuilding blockの例[1)]

失関数の設計は個々の問題によって異なるため，見落とされがちではあるが，出力データの性質を表現する唯一の部分であるためよく考えて設計する必要がある．

2.3　モデル

代表的な深層学習のアーキテクチャについては，第5回で解説したとおり，画像を入力とする場合にはCNN（畳み込みニューラルネット，Convolutional Neural Network），時系列データにはRNN（回帰結合型ニューラルネット，Recurrent Neural Network）やその改良であるLSTM（Long Short Term Memory）といった定石がある．

CNNの中でもGoogleNet，ResNet，U-Netなど幾つか良く使われているモデルがある．このうち，ResNetとU-Netについて簡単に紹介する．

ResNetは図3に示す基本構成を多数繰り返すような構造になっており，その構造はレイヤを飛び越えて接続するSkip connectionが特徴である．通常，深いネットワークモデルの場合，入力側のレイヤが上手く学習できないことが多いが，ResNetの場合はこのSkip connectionによって非常に深く，パラメータの多いモデルの学習が可能になっている．実際に用いられる層の深い例では，152層のモデル（ResNet152）が用いられる．このような深くパラメータの多いモデルをキャパシティが大きいなどと呼ぶ．キャパシティが大きいとパラメータのチューニングや初期値の選び直しなどの試行錯誤が少なくなると考えられ，とりあえず学習が上手く行くから使っていると思われるような研究をよく見かける．

CNNによる畳み込みの後に逆畳み込み（Deconvolution）を行う，Conv-deconv型のネットワーク（図4）は，画像生成を行うGAN（敵対的生成ネットワーク，Generative Adversarial Network）等，画像のようにアレイ状に並んだデータから別のアレイ状のデータを出力する時に用いられる．近年では，1枚の画像から深度画像を出力したり，ピクセル単位での画像のセグメンテーションを行ったりする等の研究が盛んである．

U-Netは図5のような構造をしており，主に画像のセグメンテーションに用いられるConv-deconv型ネットワークの一種である．最初に提案されたのが細胞画像処理の分野であったため，特に細胞画像のセグメンテーションで良く用いられている．

U-Netの特徴は各サンプリング解像度のデータがdeconvolutionのレイヤへ接続されている点である．通常のCNNではプーリングによってダウンサンプリングを行い，位置の移動に対して不変な特徴を獲得しているとされる．この特徴は“画像に何が写っているか”といった問題では有用であるが”，画像のどこに目的の物体があるか”といった問題の場合，対象の位置情報が重要であるため，むしろ位置に不変な変換では問題がある．すなわち，画像中の細胞を検出したり，追跡したりするようなタスクの場合にはconv-deconv型は適さないと言える．U-Netはこれを各解像度間の接続によってconv-deconv型では失われる位置情報を保存していると考えられる．

conv-deconv型とU-Netの例のように，一見似たような構造であっても，その設計によって保存される情報と不変となる情報は異なるため，適切なモデルの選択が重要である．

画像の場合，CNNに基づくモデルを使うのが定石ではあるが，3次元データの場合は状況が異なる．一口に3次元データと言っても，代表的なものでも図6のように点群，メッシュ，ボクセルといった種類があり，それぞれに用途や特徴が異なる．2次元画像にもっとも近いのはボクセルデータであり，深層学習を適用するならCNNを3次元にすれば良いと思われるが，2次元から3次元になることで

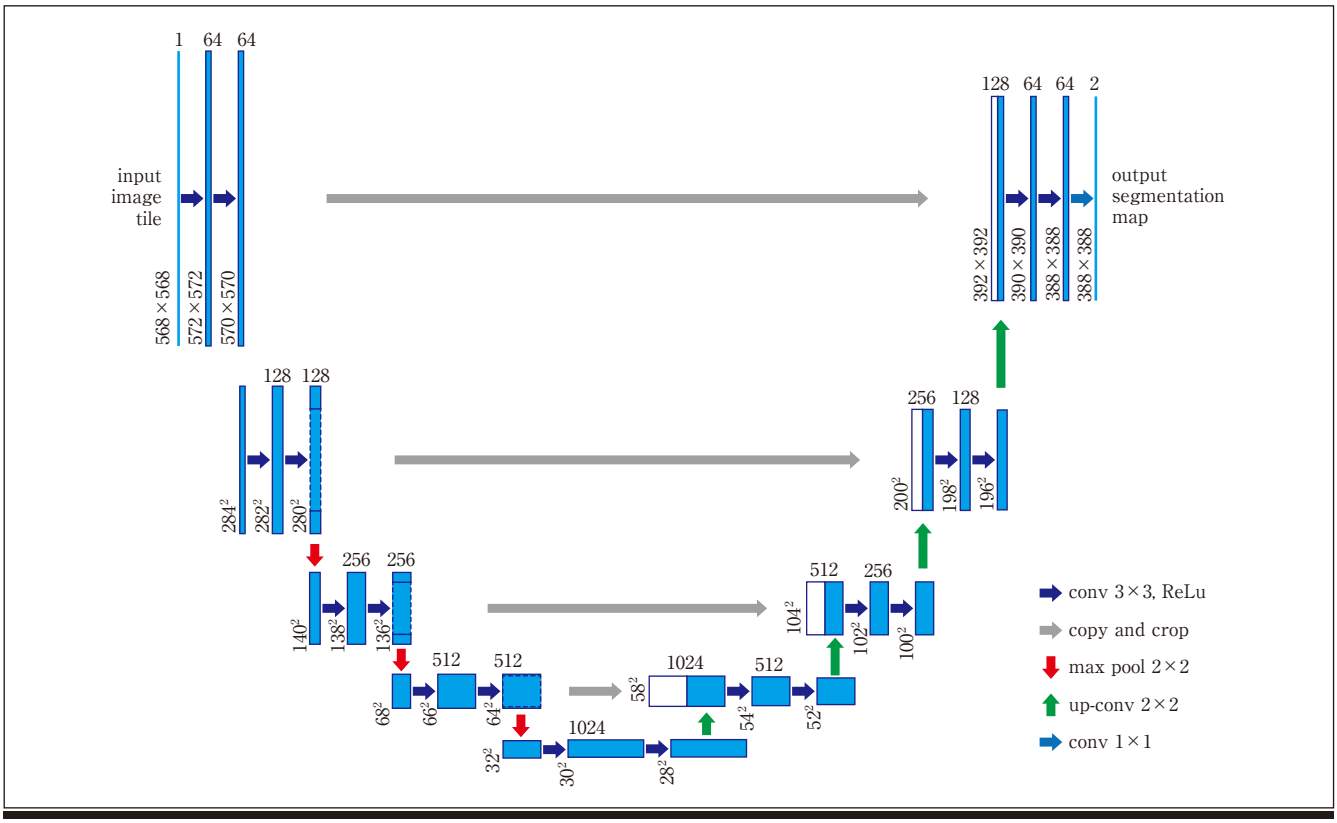


図5　U-Netの例[3]

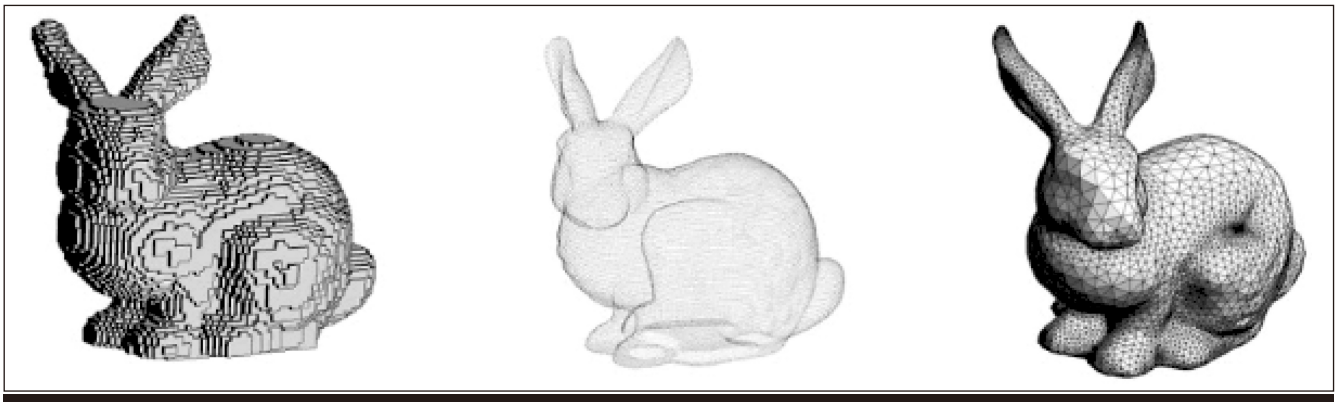


図6　左からVoxel，Point colud，Meshの例[4]

データのサイズが爆発的に大きくなるため，2次元画像ほどの高解像度のものはあまり見かけない．

点群データやメッシュデータはデータ構造が画像とはまったく異なり，データの並び順や隣り合うデータとの関連性が持つ意味は比較的薄い．そのため，2次元画像では非常に強力なCNNの適用がむずかしく，現在の最先端の研究でも模索されているようである．点群に関する2018年の研究例では，データの並びに依存しないネットワークとしてPointNet++等が提案されている．また，メッシュを扱うネットワーク[4][5]も提案されている．

2.4　計算資源

深層学習を用いて研究や開発を行う際に無視できないのが計算資源の確保である．近年ではAmazon Web Service（AWS）やGoogle Cloud Platform（GCP），Microsoft Azureといったクラウドサービスも充実しているが，これらにも利用料金が必要である．いずれにせよ長時間の学習を繰り返し行うには計算資源（＝お金）が必要になる．一方で複雑なモデルを学習するために，数週間や数ヵ月といった長時間の学習を必要とする場合もあり，例えば，第5回でも紹介したTaskonomy[6]では，およそ48000 GPU時間の計算を行っている．仮に1 GPU時間を1ドルとしてもおよ

そ約500万円である．また，Iizuka等の研究[7]では，学習にNVIDIAのGPUであるTesla K80を4枚搭載した計算機でおよそ2ヵ月の計算を行っている．こういった研究の追試を行ったり，別のデータに適用したりするだけでも同等の計算資源が必要であることから，計算資源の確保も重要である．また，比較的少ない計算資源で済む効率的なモデルや学習方法の開発は重要な研究課題と言える．

3．人工知能の課題

深層学習によって人工知能は急激に発展し，特定の問題では人間以上の性能を発揮するようになった．しかし，前節で述べたように，人工知能を適用するにはさまざまな課題が残されている．また，第1回で述べた汎用人工知能（Artificial General Intelligence）のような課題もある．

現在の人工知能にとって壁となっているのは，人間が行うような対象の理解とそれに基づく推論のように思われる．

例えば，自然言語処理では一つの文を他の言語の文へ変換するといったことは可能になっているようであるが，多数の文から成り立つ文章の文脈を読み取って翻訳するといったことは未だに難しいようである．また，映像においても同様であり，特定の行動等の検出は可能だが，長い映像の要約等は未解決の問題のようである．このように考えると，複数の理論的な推論を基に成立するような判断は難しいと思われる．このような問題は演繹的方法による人工知能と帰納的方法による人工知能として議論されている．現在の人工知能研究は，基本的には大量のデータと深層学習による帰納的方法が大成功した成果であり，さらにその結果，演繹的方法が必要と思われる課題が新たに露出しつつある状態のように思われる．

なんらかの問題を人工知能で解決する場合，前節で述べたようなデータ，損失関数，モデル，計算資源の定石に当てはまるか，あるいは当てはまらない場合でも何らかの方法で解決可能であるかが，その問題が人工知能で解決可能かを判断する基準になる．また，前述のように対象とする問題が現在の人工知能技術で解決可能な問題として定義されるかも重要な判断基準である．ただし，難しい問題と考えられていた囲碁や将棋が今やトッププロ以上の実力となっているなど予想を覆す例は多数あるため，この判断は非常に難しい．やってみると思いの外簡単に解決される場合もあれば，問題を深く分析すると上述のような演繹的方法に基づく判断が含まれる問題であったりする．結局の所，経験による判断であったり，とりあえず試してみたりといったアプローチになることが多い．

（2018年11月25日受付）

〔文　献〕

1）K.He, , X.Zhang, , S.Ren, , and J. Sun: "Deep residual learning for image recognition", In Proceedings of the IEEE conference on computer vision and pattern recognition, pp.770-778（2016）
2）H. Noh, S. Hong and B, Han: "Learning deconvolution network for semantic segmentation", in Proceedings of the IEEE international conference on computer vision, pp.1520-1528（2015）
3）O. Ronneberger, P. Fischer and T. Brox: "U-net: Convolutional networks for biomedical image segmentation", in International Conference on Medical image computing and computer-assisted intervention, pp.234-241（2015）
4）https://www.youtube.com/watch?v=vziVsrCaMHY
5）H. Kato, Y. Ushiku and T. Harada: "Neural 3d mesh renderer", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.3907-3916（2018）
6）A.R. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik and S. Savarese: "Taskonomy: Disentangling Task Transfer Learning", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.3712-3722（2018）
7）S. Iizuka, E. Simo-Serra and H. Ishikawa: "Globally and Locally Consistent Image Completion", ACM Transactions on Graphics（SIGGRAPH）（2017）

間下（ましたに）　以大（ともひろ）　大阪大学大学院基礎工学研究科博士後期課程修了．現在，同大学サイバーメディアセンター准教授．2012年，オーストリア・グラーツ工科大学客員研究員．コンピュータビジョン，機械学習，拡張現実に関する研究に従事．博士（工学）．正会員．