

Deep Learning 入門

小野 潔

概要

近年注目されている Deep Learning は突然現われたわけではなく、人工知能の数々の挫折から生み出された技術である。Deep Learning はパターン認識を自動学習で行う 5 ～ 30 層から構成されるニューラルネットである。機械学習のアルゴリズムの中には数学で証明されていないものもあり、Deep Learning に至っては、物を判別できる理由すら満足に説明できていない。『手書き数字のパターン認識』は Deep Learning の一つである Auto Encoder で実現できる。Deep Learning の転移学習は基礎研究なしで応用研究への転用やシステムに取り込むことができる。インテックでは、特定分野のカスティングボードを握るため、Deep Learning 転移学習の可能性を探っている。

1. はじめに

近年注目されている Deep Learning(深層学習) は、ニューラルネットワークという脳の仕組みをヒントにした人工知能 (Artificial Intelligence) の一分野である。Deep Learning はまだ理論が解明されていないのに関わらず、画像認識／パターン認識 [1] の分野では既存の分析手法を凌駕した。さらに今年、囲碁の世界チャンピオンに勝ち、世界を驚かせた。

Deep Learning は機械学習によるモデリングを行うため、本稿では機械学習から説明する。機械学習は人の学習能力と同

等の機能をコンピューターで実現しようとする人工知能の技術である。データベースや Web からサンプルデータを採取し、有用なパターン、ルール、知識表現、判断基準を抽出する。機械学習はデータを解析するため、統計学との関連が深く、両者は補完し合う関係である。差異を簡単に言えば、統計学は分布を仮定し、少数サンプリングに基づきエレガントな数式で表現することを目指す。それに対して機械学習は分布を仮定しない代わりに、大量データからビジネスに役に立つモデルを目指すと言える。また、機械学習のアルゴリズムの中には数学で証明されていないものもあり、後から証明されるケースも少なくない。

Deep Learning に至っては、物を判別できる理由すら説明できていない。

Deep Learning は突然現われたわけではなく、人工知能の数々の挫折から生み出された技術である。本稿では前半は統計学と人工知能の歴史、さらに Deep Learning の衝撃を述べ、後半は手書き数字を解読する Deep Learning のアーキテクチャーとその実装法を紹介する。本稿は深層学習やニューラルネットワークの動向を SE / プログラマー向けにテクニカルでわかりやすい解説を目指した [2]。

2. 人工知能のブームと挫折の歴史

最初に近代統計学と人工知能の歴史を、表1に示す。近代統計学は19世紀後半にゴルトンが生物統計学を形づくり、ピアソンが記述統計学を大成させ、20世紀初めにフィッシャーが仮説検定法を編み出した推計統計学から始まる。表1に記載がないベイズ統計学は近代統計学より古く18世紀半ばに既に発表されたが、従来の統計学とは対立した。ベイズ統計学は、新たに取得した情報によって確率の更新を認めているため、人工知能、制御理論、統計物理学等の応用分野では、基礎数学である。

一方、第一次 AI ブームは世界初のコンピュータ ENIAC(1946) が完成した僅か10年後の1950年代から始まる。この時代では既に機械学習の基礎原理となるニューラルネットワーク、遺伝子アルゴリズム、決定木等の元アルゴリズムが発表された。しかし1970年初頭に当時の人工知能では現実問題を解けないことがわかり、最初の AI の冬の時代を迎える。1980年代には通産省が第5世代コンピュータプロジェクトを発足させ、第二次 AI ブームの幕開けとなる。専門家の知識をルールベースとし、ルール推論を行うエキスパートシステムが全盛期を迎えた。これも1990年頃になると、専門家からのルール取得／保守が困難であることが判明し、第5世代コンピュータプロジェクトも大きな成果を残さずに終息した。2010年頃から IBM の Watson と Deep Learning がきっかけとなり、第三次 AI ブームは始まった。特に Deep Learning は技術イノベーションにつながる可能性を大きく秘めており、世界の IT 企業が巨額の資金を投資し、AI 研究所を次々と設立した。(Google は6億ドルで Deep Mind 社(2013)を買収。中国検索サイトの百度(Baidu) は、シリコンバレーに3億ドルの予算で Deep Learning の研究所(2013)を設立。)

表1 近代統計学と人工知能の歴史

西暦	分野	イベント/人物/企業	内容
1877年	統計	ゴルトン	回帰分析の概念確立
1896年	統計	ピアソン	相関係数
1904年	統計	スピアマン	因子分析
1935年	統計	フィッシャー	実験計画法
1936年	統計	マハラノビス	判別分析
1938年	統計	サーストン	因子分析
1938年	統計	フィッシャー	コレスポンデンス分析
1946年	コンピュータ	IBM	世界初のコンピュータ-ENIAC
(1956~1960年代)	人工知能	第一次AIブーム	探索・推論問題の解法(定理証明(1957)、遺伝子アルゴリズム(1958))
1956年	人工知能	ダートマスワークショップ	初めて人工知能(Artificial Intelligence)という言葉が出現
1958年	ニューロ	ローゼンブラット	パーセプトロンを発表(ニューラルネットワークの最初のニューロンモデル)
	統計	カイザー	因子のバリマックス回転
	統計	ラオ	成長曲線モデル
	統計	トーガソン	多次元尺度法
1962年	ニューロ	ミンスキー	パーセプトロンは線形分離しか適用できないことを指摘
1963年	人工知能	モーガン、ソニキスト	AID、回帰木
1964年	統計	ルース、ターキー	コンジョイント分析
1967年	統計	マックイーン	K-平均クラスター
1970年	統計	ヨレスコグ	共分散構造分析
1970年代	人工知能	AIの冬の時代到来	機械翻訳絶望、現実問題がとげず
1972年	統計	コックス	生存時間分析
1973年	人工知能	スタンフォード大学	MYCIN(マイン)、エキスパートシステムの開発
1975年	人工知能	ホランド	遺伝的アルゴリズム
1980年代	人工知能	第二次AIブーム	知識工学の時代 エキスパートシステム、自然言語・画像・音声理解システム
1982~1994	人工知能	通産省	第5世代コンピュータプロジェクト(1981年)に570億円
1984年	ニューロ	コホーネン	自己組織化マップ
1986年	ニューロ	ラメルハート、マクレランド、ヒントン	ニューラルネットワークの中間層以降を学習させる誤差逆伝播法を発表 ニューラルネットワークで非線形分離問題も解くことが可能に
	ニューロ	ホップフィールド	ニューラルネットワークによる最適化問題と連想記憶モデルを発表
	ニューロ	コホーネン	ニューラルネットワークによる自己組織化マップを発表
1990年代	人工知能	再びAIの冬の時代到来	知識(ルール)取得と維持が困難
1992年	人工知能	ヴァブニック	サポートベクターマシンは最強のパターン認識モデル、分類器にも適用可
1997年	ゲーム	IBM	IBMのディープ・ブルーがチェスの世界チャンピオンに勝利
2000年	統計	パール	グラフィカルモデリング
2006年	ニューロ	ヒントン	オートエンコーダー(自己符号化器)を発表。Deep Learningの発端。
2010年~現在	人工知能	第三次AIブーム	自己学習、表現の時代 ビックデータ出現、Web広がり、Deep Learningの発見
2010年	ゲーム		「あから2010」が将棋の女流棋士に勝利
2011年	人工知能	IBM	質問応答システム Watson が米国クイズ番組「Jeopardy(ジェバディ)」でクイズ王に勝利
2012年	ニューロ	大規模画像認識コンテスト(ILSVRC)	Deep Learning がコンペティションで圧勝(以後3年連続優勝)
	ニューロ	Google	トロント大学ヒントン教授と学生の会社を買収
	ニューロ	ニューヨークタイムズ誌	トップ記事で、グーグル猫を掲載
2013年	ニューロ	Facebook	ニューヨーク大学のヤン教授を所長に招き人工知能研究所を設立
2014年	ニューロ	Google	Deep Mind Technologies(英国)を6億ドルで買収
	ニューロ	Baidu(中国)	スタンフォード大学のアンドリュース教授を所長に迎えてシリコンバレーにDeep Learningの研究所を開設(3億ドルの研究予算)
	ニューロ	Facebook	人工知能のVicarious社に4,000万ドルの投資
	人工知能	ドワンゴ、リクルート	日本の各社が相次いでAI研究所設立
	人工知能	ロシアのAI	Eugene(ユーゼン) 君13歳がチューリングテストに合格
2016年	ゲーム	Google	AlphaGoが囲碁の世界チャンピオンに勝利
.....
2045年頃	人工知能	AIが人類を越す年	シンギュラリティ(技術的特異点)

3. Deep Learning (深層学習) の登場

Deep Learning は2012年に開催された一般物体認識のコンテスト (ILSVRC [3]) で、トロント大学 Hinton 教授が他のグループに比べて誤識別率が10%以上低い値となる大差で優勝し、その高精度が注目された。そして Deep Learning を一躍、世間に名を知らしめたのは『Google 猫』[4]である。2012年6月 Google は人間がプログラムすることなく、コンピューター自身で猫を認識できたと発表した。YouTube 動画からランダムに 200×200pixel の画像を1,000万枚 (人の顔、猫) 集め、9階層を有するネットワークで1,000台 (16,000コア) のコンピューターで3日かけて学習させた。その結果、人の顔、猫の顔、人の体の写真に反応するニューロンが生成できた。使用 CPU パワーを1990年頃の水準に直せば6,000年以上の学習期間を要すると言われている。

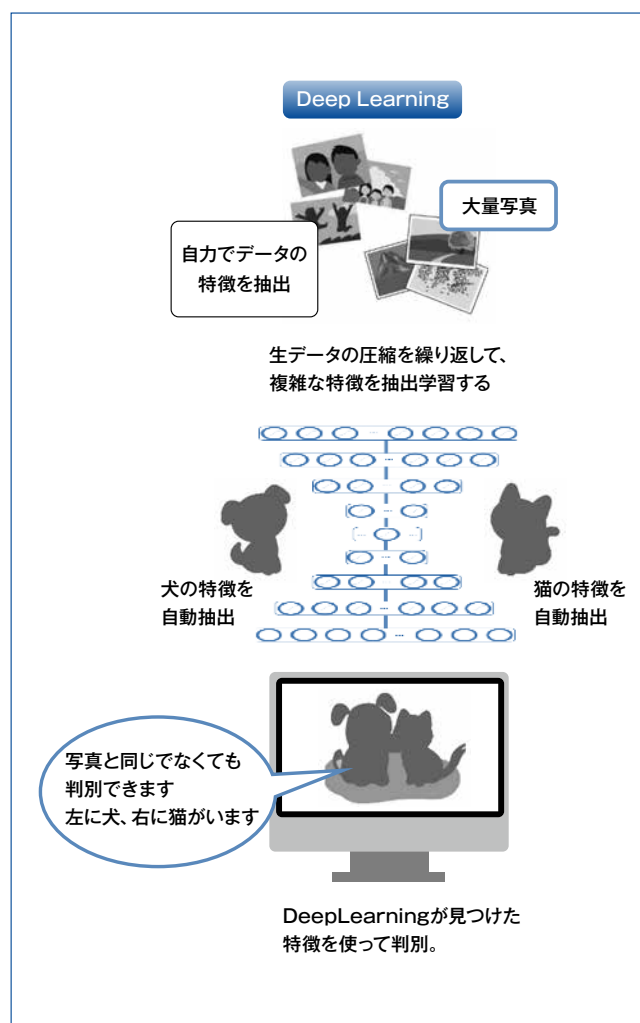


図1 Deep Learningの特徴抽出

1980年代のニューラルネットワークは1~2層で構成されるのに対して Deep Learning は5~30の深層で構成され、入力 / 出力層も10~30個から500~5000個のユニット数に拡大された。Deep Learning は従来の統計学や機械学習と全く相違する特徴を有する。従来のデータ分析はあらかじめ人が考えて組んでいたが、Deep Learning はコンピューター自らデータから特徴を突き止める (完全自律型学習)。また従来の分析では準備段階でデータクリーニングが欠かせなかったが、Deep Learning は生データから特徴抽出と識別を行う (図1)。さらに利用するほどに自己教示学習 (Self-Taught Learning) を続け、未知データもコンピューター自らパターンを学習する。課題は長い学習時間 (数日間) や大量のパラメータである。『Google 猫』の深層ネットのパラメータ数は数十億個と言われる。

4. AlphaGo(アルファ碁) の衝撃

2014年 Google が6億ドルでベンチャー企業の Deep Mind 社を傘下に収めた。翌年 Deep Mind 社は Deep Q-Network(DQN) という Deep Learning と強化学習を組み合わせた新しい機械学習を発表した。この機械学習の能力は凄まじく、TV ゲームを学習させると数時間で人よりも強くなる。この技術を AlphaGo に適用し、囲碁の世界チャンピオンに勝利するという人工知能の歴史に残る偉業を成し遂げた。

AlphaGo の開発ではプロの3,000万種類の打ち手を見せて学習させ、対戦する人間の動きを57%の確率で予測できるようにした。それ以上の棋譜がないため、AlphaGo に自己対戦を数百万回繰り返させた。AlphaGo は経験を重ねる中で徐々に人間の直観を身に着けたと言われる。

5. Deep Learningのアーキテクチャー [5]

ニューラルネットワークの原型は1969年代に研究されたパーセプトロンという単純なニューロンモデルである (図2)。このモデルは線形判別しかできないことが後日証明された。1980年代に単純パーセプトロンを階層的に組み合わせたニューラルネットワークが開発された。ニューラルネットワークは入力層、隠れ層 (中間層)、出力層のアーキテクチャーを有し、ネットワークを形成する多数のニューロンの重みを変化させることで、非線形関数を近似できる。各ニューロンの重みを解くために、誤差逆伝播法 (Backpropagation) が開発された。これは得られ

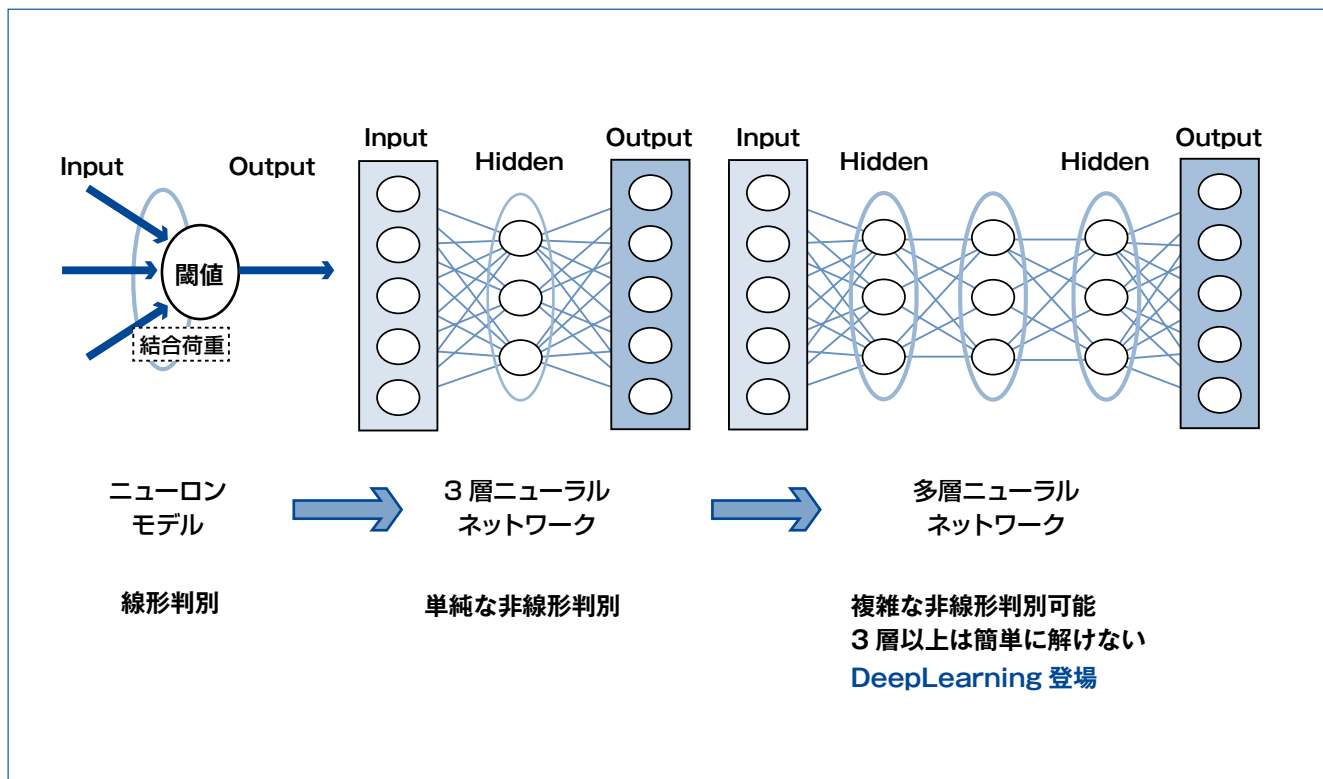


図2 ニューロンモデルからニューラルネットワークへの進化

る出力誤差を小さくする方向にパラメータを、上層から下層へフィードバックする方法である。しかし誤差逆伝播法は層を通る過程で誤差が分散され、入力層に近い層のパラメータが更新されづらく、複雑な非線形関数の生成ができない。この点を改善したのが Deep Learning である。

本章では Deep Learning の特徴抽出を説明するため、Deep Learning の初期モデルの一つである Auto Encoder(自己符号器)の実装を紹介する。「手書き数字の判別」は統計学の主成分分析が用いられるが、主成分分析は線形の次元圧縮、Auto Encoder は非線形の次元圧縮を行う。

Auto Encoder は入力層と出力層に同じニューロン数、しかも入力層と出力層に同じ数値を入れて、「教師なし学習」⁽¹⁾を行うニューラルネットワークである。入力層と出力層に同じ手書きデータを入れるところが Auto Encoder のミソである。実は入力層と出力層を同じ数値にすると、ニューロン数が少ない隠れ層で情報圧縮が起き、特徴が蓄積される。図3では入力層(5ニューロン)の情報圧縮した値が隠れ層(2ニューロン)に生成される。

隠れ層が2層以上あると、誤差逆伝播法で述べたとおりに最適解に収束する保証はない。そのため Deep Learning では前準備で各ユニットのパラメータ等の初期値の計算を行うこと

で解決した。2層以上の場合、Deep Learning はまず隠れ層1層だけ作り、次に出力層を取り除き、隠れ層を入力とみなし、もう1層積み上げる。つまり入力層側から順次1層ずつ分離し、各層ごとに教師なし学習を反復し、単層の Auto Encoder を積み上げる。実際のニューロン数は膨大であり、Hinton は画像の次元を2,000→1,000→500→30と圧縮し、30→500→1,000→2,000と復元した。

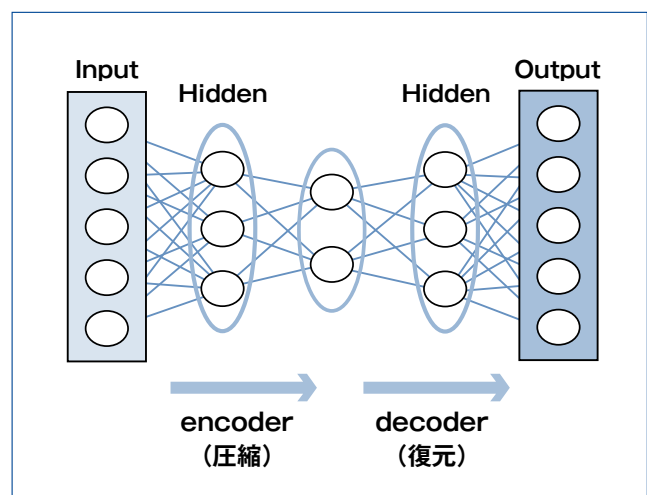


図3 Auto Encoderの基本アーキテクチャー

(1)「教師なし学習」は外部から正解データを与えられず、機械がデータからパターンを見つける。統計学のクラスター分析と同じ機能。

6. 手書き数字のデータ

本稿の目的の一つはSE / プログラマーの方に Deep Learning の理論やプログラムの実装法を紹介することである。この章では『Deep Learning による手書き数字の判別プログラム』を説明する。手書きの数字の画面データは米標準技術局 [6] からダウンロードできる。手書き数字は $28 \times 28 = 784$ ピクセルの画像データ (図4①) なので、0-783のピクセル番号を割り当てる。次に256階調グレースケールを画面に変換し、ピクセル番号に割り当てることで、0-255の整数値を取る784次元ベクトルからなるデータセット (図4②) を作成できる。Auto Encoderはこの次元を情報量をできるだけ落とさないように圧縮 (削減) する。

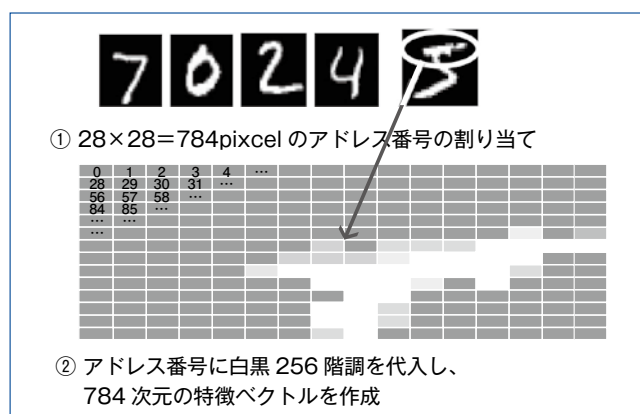


図4 手書き数字を784(28×28)次元の特徴ベクトルへ変換

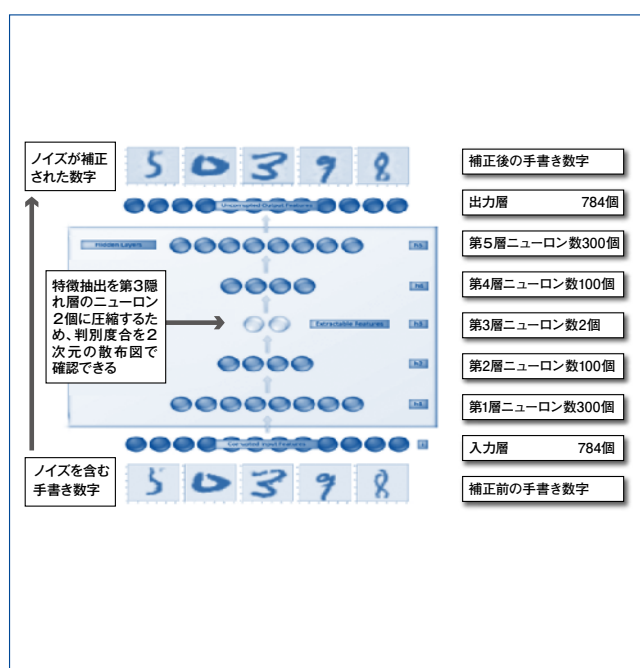


図5 Auto Encoder多層ネットと数字修正

7. Deep Learningの実装

ここでは Auto Encoder を改良した Stacked Denoising Auto Encoder [7] を紹介する。両者は同じアーキテクチャーを有するが、Stacked Denoising Auto Encoder はノイズ (画素の欠損) を混ぜたデータを学習させる。ネットワーク構造は5階層隠れ層から構成され (図5)、第1隠れ層から第3隠れ層にあがるにつれて、隠れ層のニューロン数は少なくなる。ここは特徴抽出を行う圧縮 (encoder) 部分である。逆に第3隠れ層から第5隠れ層にあがるにつれて隠れ層ニューロン数は増加する。ここは手書き数字の復元する解読部分 (decoder) である。この Stacked Denoising Auto Encoder ではノイズ (画素の欠損) を学習させるので、第3層を通過し数字をデコードした際に手書きの数字が補正される。ノイズが含まれる手書き数字 (図5の下の手書き数字) が出力されると、手書き数字の画像がわずかに明瞭になる (図5の上と下の手書き数字を比較)

図6に統計言語 SAS で書かれたプログラムを示す [8]。Auto Encoder を積み重ねるアーキテクチャーを実現するために、『ネット層を固定する freeze』と『ネット層を解放する thaw』の2種類のコマンドを利用する。最初にすべての各層を freeze (固定) させる。次に入力層から第1層の隠れ層のみを thaw (解放) し学習を行い、第1層のニューロンの重みづけを算出する。次に第1層の重みを固定し、第2層を解放し

```
proc neural /* Deep Learning のアーキテクチャは neural プロシジャーで構築する */
data=autoencoderTraining dmbat=work:autoencoderTrainingCat;
performance compile details cpucount=4 threads=yes; /* ENTER VALUE FOR CPU COUNT */
/*DO NOT EXCEED NUMBER OF PHYSICAL CORES */

/* DEFAULTS: ACT= TANH COMBINE= LINEAR /* IDS ARE USED AS LAYER INDICATORS - SEE FIGURE 6 */
archi MLP hidden= 5; /* 隠れ層は 5 層を設定 */
hidden 300 / id= h1; /* 第 1 隠れ層は 300 ニューロンを設定 */
hidden 100 / id= h2; /* 第 2 隠れ層は 100 ニューロンを設定 */
hidden 2 / id= h3 act= linear; /* 第 3 隠れ層は 2 ニューロンを設定 */
hidden 100 / id= h4; /* 第 4 隠れ層は 100 ニューロンを設定 */
hidden 300 / id= h5; /* 第 5 隠れ層は 300 ニューロンを設定 */
input 784 / id= i level= int end= end;
target 784 / id= t level= int end= end;

/* BEFORE PRELIMINARY TRAINING WEIGHTS WILL BE RANDOM */
initial random 123;
prelia 10 preiter= 10;

/* TRAIN LAYERS SEPARATELY /* 全各層を一層固定する */
freeze h1-h2; /* 第 1 隠れ層から第 2 隠れ層を固定する */
freeze h2-h3; /* 第 2 隠れ層から第 3 隠れ層を固定する */
freeze h3-h4; /* 第 3 隠れ層から第 4 隠れ層を固定する */
freeze h4-h5; /* 第 4 隠れ層から第 5 隠れ層を固定する */
train technique= congru maxtime= 10000 maxiter= 1000; /* 学習 */

freeze i-h1; /* 入力層から第 1 隠れ層を固定する */
thaw h1-h2; /* 第 1 隠れ層から第 2 隠れ層を開放する */
train technique= congru maxtime= 10000 maxiter= 1000; /* 学習 */

freeze h1-h2; /* 第 1 隠れ層から第 2 隠れ層を固定する */
thaw h2-h3; /* 第 2 隠れ層から第 3 隠れ層を開放する */
train technique= congru maxtime= 10000 maxiter= 1000; /* 学習 */

freeze h2-h3; /* 第 2 隠れ層から第 3 隠れ層を固定する */
thaw h3-h4; /* 第 3 隠れ層から第 4 隠れ層を開放する */
train technique= congru maxtime= 10000 maxiter= 1000; /* 学習 */

freeze h3-h4; /* 第 3 隠れ層から第 4 隠れ層を固定する */
thaw h4-h5; /* 第 4 隠れ層から第 5 隠れ層を開放する */
train technique= congru maxtime= 10000 maxiter= 1000; /* 学習 */

/* RE-TRAIN ALL LAYERS SIMULTANEOUSLY /* 各層層の学習を終えたあと、全層を開放する */
thaw i-h1;
thaw h1-h2;
thaw h2-h3;
thaw h3-h4;
thaw h4-h5;

/* Deep Learning では最終に微調整のために再度、学習を行う。ここもミソ。 */
```

図6 SASプログラム

第2層を計算する。要は freeze と thaw 繰り返すことで、Auto Encoder を積み上げることと同等になる。最後にすべての層を解放し、各ニューロンの微調整を行えば、各ユニットの重みづけが確定する [3]。

(実行環境) プログラム : SASver 9.4、
OS: Windows 2012 Standard、
CPU Xeon 2.70 Ghz(4 コア)、メモリ 8 GB
(学習比率) 学習データ6万件、検証データ1万件
(学習時間) 2日間

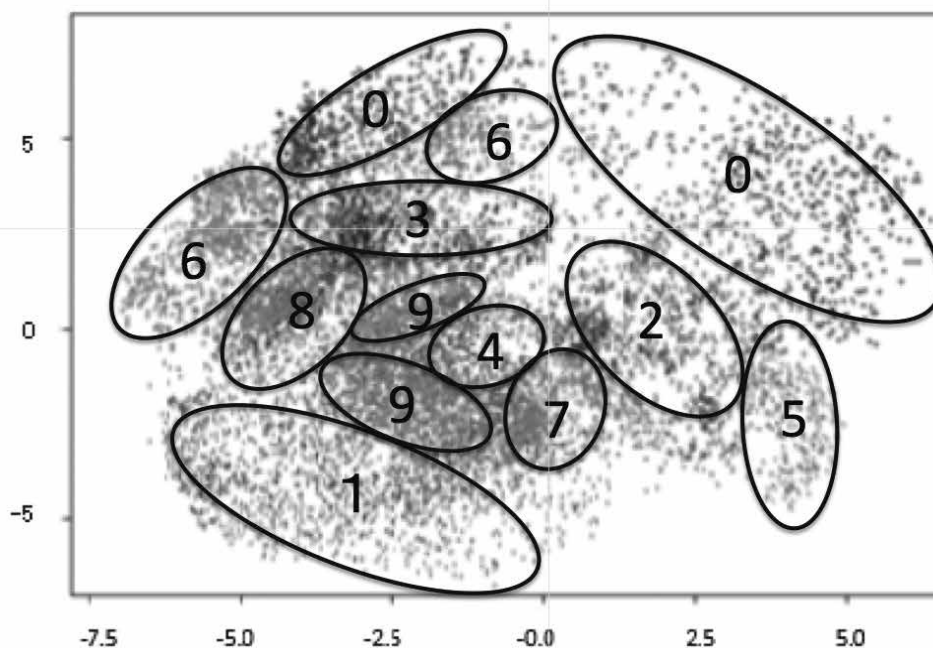
8. 手書き数字の判別力

Deep Learning の隠れ層のニューロン2個に生成された値を散布図にすると、特徴抽出を確認できる。図7は特徴抽出された検証データの手書き数字の散布図である。実際は色別で数字の判別を示すが、白黒印刷のため実際の数字の分布をおおよそ楕円で囲んだ。このプログラムは手書き数字の特徴を抽出し、それに基づいて新しい手書き数字を判別し、同時にノイズの補正機能を有すると言える。

9. 課題

Deep Learning は、画像認識・音声認識の分野で他の統計手法とは一桁も相違する精度を有する。応用分野は無限に存在し、将来は自動車の自動運転技術への応用が期待されている。ただ Deep Learning を実社会に応用する上で留意点がある。Deep Learning は学習結果を人が理解可能な形で取り出すことができないため人工知能が明らかに誤りと思える判断をした場合にも、その原因の解析は困難である。判別・分類の応用には適しやすいが、予測ビジネスや医療分野への応用では理由を説明できないことが弱点になる。

Deep Learning では、ユーザーがデータの構造を見極めて適切な層の数や形のアーキテクチャーを設計する必要がある。そのため Deep Learning の性能を発揮させるには、モデルに合わせて関数や大量のパラメータを調整しなければならないが、隠れ層数やニューロン数の決定は容易でない。『Google 猫』は手書き数字のパターン認識技術の延長線にあるが、開発には想像もつかない労力を要する。



数字0の手書きは、丸印と縦長楕円印の2つに分類されるため、左の判定領域は2か所存在する。また数字1,9,7の手書きは類似するため、判定領域は隣り合う。

図7 手書き数字の判定

10. おわりに

本稿では2012年までの Deep Learning の発展のさわりを紹介した。実は2013年以降に膨大な論文が発表され、先端企業(Google, Facebook, 百度等)には特許戦争が発生しており、現在は理論が混沌としている⁽²⁾。

最先端の Deep Learning のネットワークを構築するのは、巨額な資本投資(百億円単位)と優秀な頭脳の集団が必要である。この数年の世界を巻き込んだ AI ブームは Deep Mind 社をはじめ、多くの人工知能研究拠点を設立した。日本は出遅れており、日本の全拠点を結集しても、Google の研究規模には及ばない。

インテックが Deep Learning 研究を始めたとしても、とても先端企業には太刀打ちできない。ただ Deep Learning の転移学習(Transfer Learning)を利用すれば、最初からネットワークを構築しなくとも Deep Learning を実用システムに取り込むことができる。転移学習は学習済みのネットワークを他のタスクに転用する手法である。大量の画像で学習させたネットワークは画像の特徴抽出器としても優秀であり、他データで再学習すれば他への転用が容易となる。既に先端企業は学習済みのネットワークを利用できるクラウド環境⁽³⁾を提供している[9]。インテックは特定分野のキャスティングボードを握るため、Deep Learning 転移学習の可能性を探ると共に、Deep Learning の権利といった法律の動向にも注目している。

(2) 本稿は「教師なし学習」のAuto Encoderを説明したが、2012年以降は学習効率が良い「教師つき学習の畳み込みニューラルネットワーク」がメインである。しかし人が「数万の教師あり学習データ」に答えを割り振る必要があるため、最近では少量データの「教師あり学習」の後に大量データの「教師なし学習」を行う「半教師あり学習」が有望視されている。

(3) 転移学習は大きなビジネス課題を抱えている。最初のネットワーク構築には膨大な労力と資本を要するが、それを守る特許権も著作権も存在しない。そのため一部の先端企業は、学習済みネットワークを自社環境内での利用させ、顧客とノウハウの囲い込みを始めている。

参考文献

- [1] C.M. ビシヨプ (元田浩) : パターン認識と機械学習 (上・下巻) , 丸善出版, (2012)
- [2] 小野潔、松澤一徳 : SAS による新しい大規模統計学 I & II , pp223-240, SAS ユーザー総会 2015, SAS Insititute Japan, (2015)
- [3] Stanford Vision Lab:Large Scale Visual Recognition Challege 2012, ILSVRC2012, 2014/7/2, <http://image-net.org/challenges/LSVRC/2012/>, (参照 2016/06)
- [4] Quoc V.Le, Marc' Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S.Carrado, Jeff Dean, Andrew Y.Ng:Building High-level Features Using Large Scale Unsupervised Learning,Research at Google, 2012, http://static.googleusercontent.com/media/research.google.com/ja//archive/unsupervised_icml2012.pdf, (参照 2016/06)
- [5] 麻生英樹、安田宗樹、前田新一、岡野原大輔、岡谷貴之、久保陽太郎、ボレガラ・ダヌシカ : 深層学習 , 近代科学社 ,(2015)
- [6] Yann LeCun, Connna Cortes, Christopher J.C.Burges:THE MNIST DATABASE of handwritten digits, The National Institute of Standards and Technology,1998/11, <http://yann.lecun.com/exdb/mnist/>, (参照 2016/06)
- [7] Geoffrey E. Hinton, R. R. Salakhutdinov: Reducing the Dimensionality of Data with Neural Networks, Science 313(5786), pp504-507, (2006)
- [8] Patrick Hall, Jared Dean, Ilknur Kaynar Kabul, Jorge Silva : An Overview of Machine Learning with SAS® Enterprise Miner™, SAS Institute Inc., 2014, <http://support.sas.com/resources/papers/proceedings14/SAS313-2014.pdf>, (参照 2016/06)
- [9] TensorFlow is an Open Source Software Library for Machine Intelligence, TensorFlow, 2016/5, <https://www.tensorflow.org/>, (参照 2016/06)

本論文には他社の社名、商号、商標および登録商標が含まれます。



小野 潔

ONO Kiyoshi

- 社会システム戦略事業部
社会システムプラットフォーム開発部
- データ分析、モデル構築に従事 データサイエンティスト
- 人工知能学会員、日本不動産金融工学学会員、
SAS ユーザー会世話人／論文審査員、
日本不動産金融工学学会評議員