超平面を用いたDeep Learning情報処理過程の解析とシグナルフロー線図による表現

武川, 公
原岡, 剛一
長井, 聡
福田, 博也

(Citation)
神戸大学大学院人間発達環境学研究科研究紀要, 14(1):139-141

(Issue Date)
2020-09-30

(Resource Type)
departmental bulletin paper

(Version)
Version of Record

(JaLCDOI)
https://doi.org/10.24546/81012463

(URL)
https://hdl.handle.net/20.500.14094/81012463

神戸大学大学院人間発達環境学研究科研究紀要
第14巻第１号　2020

研究報告

超平面を用いたDeep Learning情報処理過程の解析とシグナルフロー線図による表現

Analysis of the Deep Learning Process using Hyperplanes and a Signal Flow Graph

武川　公＊　　原岡　剛一＊＊　　長井　聡＊＊＊　　福田　博也＊＊＊＊
Akira TAKEKAWA＊　　Goichi HARAOKA＊＊
Satoshi NAGAI＊＊＊　　Hiroya FUKUDA＊＊＊＊

要約：Deep Learningには，多数のユニットが使われている。各ユニットは入力とそれに対する重みを掛け算して和を取るニューロン部と，その値を活性化関数で変換して出力とする出力部からなっている。この研究では，深層学習におけるユニットの働きを，超平面を用いて解釈することを行った。その結果，各ニューロンは，重みというユニットに固有な法線ベクトルを持つ超平面を表していることが分かった。また，n個の特性量が入力ベクトルとして与えられるとき，法線ベクトルへの入力ベクトルの射影の長さが計算され，その値が各ニューロンに記憶される。この過程を各層において繰り返しながら，計算は最終層に向かって進み，最終層で出力結果を得る。この一連の情報処理過程において，ユニットによる多数の超平面が形成され，それらの組み合わせによって非線形の判断が実行され，画像が認識されることが分かった。さらにDeep Learningの情報処理過程をシグナルフロー線図で表した。この線図によって，Deep Learningの情報処理過程を視覚的に理解することが可能となった。

キーワード：人工知能，ディープラーニング，超平面，誤差逆伝播

１．はじめに

人工知能の研究の歴史は，人間がどのように事物を認識し，その知識に基づいて探索し，推論するのかを分析し，コンピュータにも同じ手法を使って認識，探索，推論を行わせようとしたのが初めである。しかし人間の認識行動は多種多様であり，それらをコンピュータプログラムとして定式化して，物事を判断させることはなかなか難しい問題であった。1960年代には盛んに，この試みが行われたが，結果的には実用化させることが出来なかった[(1)]。次に1980年代に入ると，エキスパートシステムと言われる手法の人工知能が現れた。これは専門家の思考プロセスをif thenのルールで表し，そのルールのデータベースを運用して，専門家と同じ判断を機械に模擬させようとする試みであった。この試みは一定の成果[(2)(3)(4)]を収めたが，コンピュータが知能を持ち，専門家を上回る結果を生み出すことは出来なかった。

2000年以降になると膨大な知識を機械に学習させ，その中から特徴を抽出し，機械自身に判断させる手法が考え出された。これはすでに人間の神経細胞を模擬し，判断させると言うニューラルコンピュータ[(5)]という考え方で示されていたが，これを多層化して何段にも深い学習をさせ，その特徴を記憶させるという手法がDeep Learning（深層学習）となって現れた[(6)(7)(8)(9)]。Deep Learningは，ユニット（または，ノードnode）と言われる一つの神経細胞を層内に多数持っており，それを多層化させて各ユニット内のデータに対する重みを決定している。その計算過程はブラックボックス[(10)]と言われて，論理が明快でないという批判があった。そこで我々は以下のことを明らかにする。

(1) この重みは，n個の特徴量で表されるn次元空間における超平面の法線ベクトルを表す。
(2) この法線ベクトルとデータのベクトルとの内積を取り，その値を活性化関数（activation function）で変換したものはユニットの出力値であり，その値が次のユニットの入力値となっている。
(3) この計算過程を，入力層から出力層まで繰り返して，すべての法線ベクトルを決定することがDeep Learningの情報処理過程である。
(4) 最後に，Deep Learningで行われるプロセスの全体をシグナルフロー線図[(11)]で表し，複雑な情報処理過程を視覚的に理解する。

２．Deep Learningのユニット

Deep Learningの計算過程は図１のようなユニットを組み合わせることによって実行される。ユニットは入力部と，それを統合する統合部（ニューロン，neuron）と，出力部からなっている。入力部は，ユニットにn個の特徴量のデータが与えられるとき，$\boldsymbol{x}=(x_1, x_2, \cdots, x_n)$という行ベクトルで表わされる。ニューロンでは，このベク

＊甲南大学大学院FIRST特別研究員，姫路独協大学名誉教授
＊＊神戸大学医学部附属病院美容外科准教授
＊＊＊神戸大学大学院人間発達環境学研究科研究生
＊＊＊＊神戸大学大学院人間発達環境学研究科准教授

（2020年３月31日　受付
2020年５月11日　受理）

トルの各要素に重み $w_1, w_2, \cdots, w_n$ を掛け，それらの総和にバイアス w_0 を加えたものを z として出力する。その計算は $\boldsymbol{w}=(w_1, w_2, \cdots, w_n)$ を用いると，$z=\boldsymbol{x}\boldsymbol{w}^{\mathrm{T}}+w_0$ で与えられる。ここで上側の添え字 T は転置行列を表す。

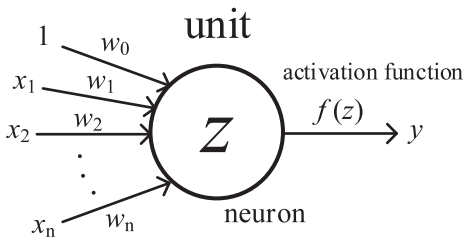

図１　ユニットの入出力関係

$\boldsymbol{x}\boldsymbol{w}^{\mathrm{T}}$ はベクトル $\boldsymbol{x}$ とベクトル $\boldsymbol{w}$ の内積である。得られた z は適当な活性化関数 $f(z)$ によって変換され，ユニットの出力値 y が得られる。この活性化関数の導入は $\boldsymbol{w}$ の要素 w_i で微分することを可能にするためのものである。近年，活性化関数として ReLU（Rectified Linear Unit）関数[6]が主に用いられている。ReLU 関数は入力が正または 0 のとき，その値をそのまま出力し，負の値のときには 0 を出力するものである。この場合 z が非負の範囲では，ユニットの出力は線形関数で次のユニットに信号が伝えられることになる。また，この関数を用いることによって，ユニットは，その重みを調整し，ユニットの発火（出力が正の状態）を制御出来ることも示している。以下の議論においては，活性化関数として ReLU 関数が使用されることを想定している。

n 次元空間の任意の点，座標 $(x_1, x_2, \cdots, x_n)$ から法線ベクトル $(w_1, w_2, \cdots, w_n)$ を持つ超平面への距離は，$|(w_1, w_2, \cdots, w_n)|=1$ とすると $|\boldsymbol{x}\boldsymbol{w}^{\mathrm{T}}+w_0|$ で与えられる。それゆえに $|z|$ は座標 $(x_1, x_2, \cdots, x_n)$ から法線ベクトル $\boldsymbol{w}$ の超平面への距離であることが分かる。

以上に述べたことを式で表すと式 (2. 1)，(2. 2) となる。

$$z=\boldsymbol{x}\boldsymbol{w}^{\mathrm{T}}+w_0 \tag{2. 1}$$

$$y=f(z) \tag{2. 2}$$

結局，一つのユニットによって，座標 $\boldsymbol{x}$ に対する超平面の法線ベクトルとその超平面への距離が決定されることになる。

３．第 i 層 j 番目のユニットの入出力関係

Deep Learning においては，まず入力部とそれを受け取る第 1 層があり，この第 1 層には多数のユニットが置かれている。また最終の出力層にはいくつかのユニットが並んでおり，このユニットの出力部が最終層を形成している。この入力層と出力層の間に中間層があり，多数のユニットを含んだいくつもの層が置かれている。

図２はこの多数のユニットの中の一つのユニットである第 i 層 j 番目のユニットの入出力関係を表したもので，第 i 層 j 番目のユニットの表現は，アルファベットの右側の (i, j) で表すものとする。

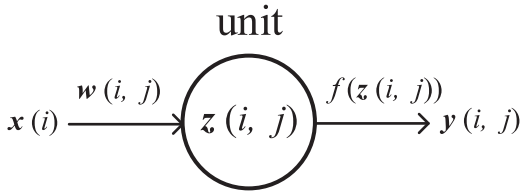

図２　i 層 j 番目のユニットの入出力関係

このとき入力データ $\boldsymbol{x}(i)$，その重み $\boldsymbol{w}(i, j)$，第 i 層全体のニューロンの出力 $z(i)$，全体のユニットの出力 $y(i)$ は式 (3. 1)，(3. 2)，(3. 7)，(3. 8) のようになる。

$$\boldsymbol{x}(i)=\big(x(i, 1), x(i, 2), \cdots, x(i, m(i-1))\big) \tag{3. 1}$$

ここで $m(i-1)$ は $(i-1)$ 層のユニットの出力データ数である。

$$\boldsymbol{w}(i, j)=\big(w(i, j, 1), w(i, j, 2), \cdots, w(i, j, m(i-1))\big) \tag{3. 2}$$

$$z(i, j)=\boldsymbol{x}(i)\boldsymbol{w}(i, j)^{\mathrm{T}}+w_0(i, j) \tag{3. 3}$$

ここで $\boldsymbol{z}(i)$，$\boldsymbol{w}(i)^{\mathrm{T}}$，$\boldsymbol{w}_0(i)$ を次式のように置くと，

$$\boldsymbol{z}(i)=\big(z(i, 1), z(i, 2), \cdots, z(i, m(i))\big) \tag{3. 4}$$

$$\boldsymbol{w}(i)^{\mathrm{T}}=\big(w(i, 1)^{\mathrm{T}}, w(i, 2)^{\mathrm{T}}, \cdots, w(i, m(i))^{\mathrm{T}}\big) \tag{3. 5}$$

$$\boldsymbol{w}_0(i)=\big(w_0(i, 1), w_0(i, 2), \cdots, w_0(i, m(i))\big) \tag{3. 6}$$

ここで $w_0(i, j)$ は各ユニットのバイアス値である。結局，i 層の出力値および i + 1 層の入力値は次のように表される。

$$\boldsymbol{z}(i)=\boldsymbol{x}(i)\boldsymbol{w}(i)^{\mathrm{T}}+\boldsymbol{w}_0(i) \tag{3. 7}$$

$$\boldsymbol{y}(i)=f(\boldsymbol{z}(i)) \tag{3. 8}$$

$$\boldsymbol{x}(i+1)=\boldsymbol{y}(i) \tag{3. 9}$$

式 (3. 7)～(3. 9) において $i=1, 2, \cdots, L$ を繰り返して代入すると，最終層 $y(L)$ が求められる。これが出力層である。出力層のユニット数 m は出力カテゴリー数と一致しており，Deep Learning の最終層において第 L 層の出力結果と教師層のデータ t との比較が行われる。ここで t は $t=(t_1, t_2, \cdots, t_m)$ である。

そのとき，評価関数 E は次の値で評価されるものとする。

$$E=\frac{1}{2}|(y(L)-t)|^2 \tag{3. 10}$$

この E を最小にするように，各ユニットの重み $\boldsymbol{w}(i, j)$ は決定される。実際の $\boldsymbol{w}(i, j)$ の決定においては，出力カテゴリーに対応するあらゆる種類のデータ $\boldsymbol{x}(1)$ が入力層に入力され，式 (3. 10) によって繰り返し学習が行われ，それぞれのカテゴリー入力に対応する出力層のカテゴリーのユニットが高い値（確率値）を持つように調整される。

４．誤差逆伝播法（back propagation）[6]

評価関数 E を最小にするためにはユニットの重み $\boldsymbol{w}(i, j)$ の各要素，式 (3. 2) の $w(i, j, \ell)$ を適切な値に導く必要がある。このために $w(i, j, \ell)$ のすべてに対して，次の更新式に従って $w(i, j, \ell)$ の要素を少しずつ変化させ，E の最小値を求める。ここで ℓ は $w(i, j)$ ユニットの中の任意の要素である。また η は学習率（learning rate）であり，事前に外部から与えられるものである。

$$w(i, j, \ell) = w(i, j, \ell) - \eta \frac{\partial E}{\partial w(i, j, \ell)} \qquad (4.1)$$

式 (4. 1) の計算をするにあたって，$\partial E/\partial w(i, j, \ell)$ の計算をする必要があるが，数式によって，それを求めると煩雑な計算になる。それゆえ，ここでは計算グラフ（computational graph）によってそれを求める方法を述べる。計算グラフの利点は，逆方向の伝播によって微分を効率よく計算できることにある。これは局所的な微分は最終層から第一層に向けて伝播していく原理を用いたものである。たとえば $L-2$ 層の $w(L-2, j, \ell)$の各要素の評価関数 E に対する変化を考えるとき，それは $\partial E/\partial w(L-2, j, \ell)$ で表されるが，これは合成関数の微分法によれば，$\partial E/\partial w(L-1, j, \ell')\cdot\partial w(L-1, j, \ell')/\partial w(L-2, j, \ell)$ と表される。ここで，ℓ' は $\boldsymbol{w}(L-1, j)$ 中の任意の要素を表したものである。これは $L-1$ 層の変化分 $\partial E/\partial w(L-1, j, \ell')$ に，$L-1$ 層に対する $L-2$ 層の重みについての局所的な微分を掛け算したものとなっている。このようにして，第１層までの重みについての微分が簡単に求められ，E を最小にする計算は容易になる。

５．Signal flow chart による Deep Learning の記述[11]

第 i 層におけるユニット間の関係式は，式 (3. 7)〜(3. 9) で述べたが，ここでは Deep Learning の処理の全体像をシグナルフロー線図によって記述する。図３はそのシグナルフロー線図である。まず第１層の入力 $\boldsymbol{x}(1)$ が与えられると，それに対して重み $\mathbf{w}(1)$ が掛けられる

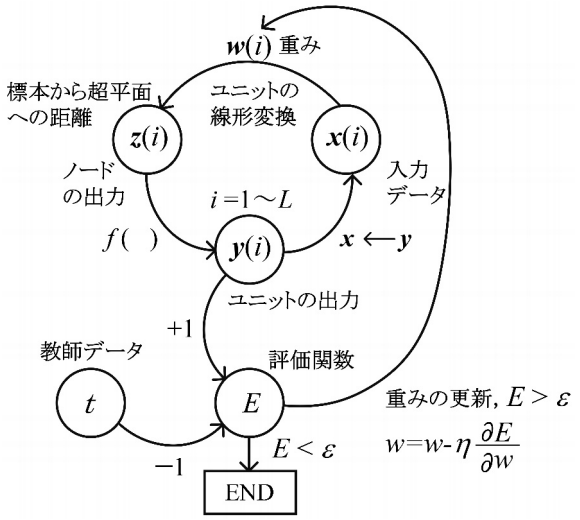

図３　Deep Learning に対するシグナルフロー線図

その結果，出力 $\boldsymbol{z}(1)$ が得られる。$\boldsymbol{z}(1)$ は活性化関数 $f(\boldsymbol{z}(1))$ によってユニットの出力 $\boldsymbol{y}(1)$ となる。次に $\boldsymbol{y}(1)$ は第２層への入力となって $\boldsymbol{x}(2)$ が与えられる。これを L 層まで繰り返し，最終層で $\boldsymbol{y}(L)$ の結果を得る。この $\boldsymbol{y}(L)$ と教師データが比較され，評価関数の値が十分に小さくなければ，重みは式 (4. 1) に従って更新される。このことを多数回繰り返して，最終的に L 層からなる最適な重みの集合 $\boldsymbol{w}_L$ とバイアス $\boldsymbol{w}_{0L}$ が決定される。この重み $\boldsymbol{w}_L$ とバイアス $\boldsymbol{w}_{0L}$ の中にデータ $\boldsymbol{x}(1)$ の特徴が埋め込まれることになる。

以上のことから，Deep Learning は，標本のデータに対して線形変換の処理をユニットごとに繰り返しながら，それぞれのユニットに対して，評価関数を最小にする，超平面の法線ベクトル $\boldsymbol{w}_L$ とバイアス $\boldsymbol{w}_{0L}$ を決定している情報処理であることが理解できる。

６．おわりに

Deep Learning には多数のユニットが使われているが，そのユニットの一つについて，入出力関係を数式で記述するとき，一つのユニットは n 個の特徴量によって作られる n 次元空間の一つの超平面を表しており，重みベクトルはその超平面の法線ベクトルであることがわかった。Deep Learning における多数の超平面の重みベクトルは，教師データと Deep Learning の最終層の出力値とから導かれる評価関数を最小にするように決定される。このようにして，最終的に求められた全ユニットの重みベクトルの集合は，全カテゴリー項目を含んだ入力ベクトルに対応している。このことから Deep Learning は多数の超平面を組み合わせることによって，画像を認識するという，非線形の判別をしていることが分かった。線形判別を多層組み合わせることによって非線形の判別を可能にすることは，多層パーセプトロンですでに例証されている[6]。最後に Deep Learning の情報処理過程をシグナルフロー線図で表した。これによって線形変換の重みの調整を繰り返しながら，全体として教師データに一致するように重みが決定されて行くプロセスを視覚的に捉えることが出来た。これは Deep Learning の情報処理過程の理解に役立つものであった。

参考文献

(1) 松尾豊：「人工知能は人間を超えるか」，角川 EPUB 選書021, pp. 60-82（2016）

(2) 北村新三・辻茂樹・田中克己：「東洋医学における弁証論治エキスパートシステム」，システムと制御，30巻，６号別冊，pp. 37-39（1986）

(3) 武川公・北村泰彦：「人工知能を用いた古典ギリシャ語（特にκοινη）の文法解析」，第27回情報科学技術研究集会論文集，pp. 337-341（1991）

(4) Shortliffe, E, Hance: “Computer-based medical consultation”, MYCIN. American Elsevier (1976)

(5) 合原一幸：「ニューラルコンピュータ」，東京電機大学出版局（1988）

(6) V. Nair and G. E. Hinton: “Rectified linear units improve restricted Boltzmann machines”, In International Conference on Machine Learning (ICML), pp. 807-814 (2010)

(7) Nikhil Buduma（牧野聡 訳）：「実践 Deep Learning―Python と TensorFlow で学ぶ次世代の機械学習アルゴリズム」，オライリー・ジャパン（2018）

(8) 斎藤康毅：「ゼロから作る Deep Learning 2」，自然言語処理編，オライリー・ジャパン（2018）.

(9) 「Newton 別冊　ゼロからわかる人工知能」（2018）

(10) 「AI の「ブラックボックス」問題とは」，日刊工業新聞，https://www.nikkan.co.jp/articles/view/0000446022（2020-2-1参照）

(11) 北村新三・武川公・松永公廣：「制御工学」，森北出版（1987）