

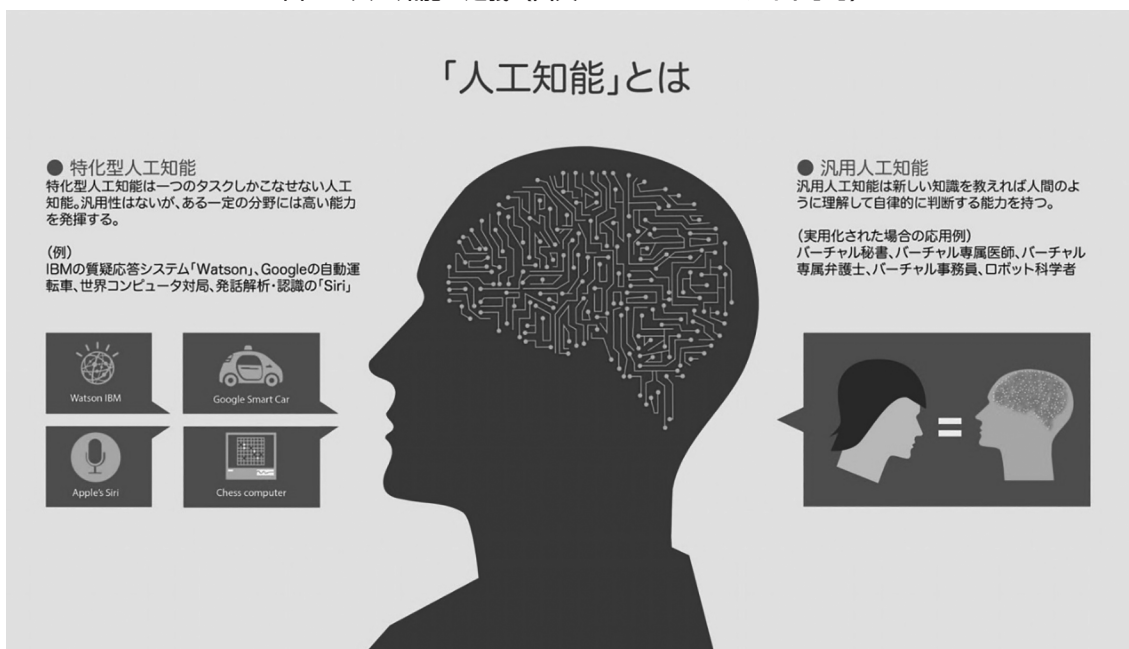
AI の概要， 変遷及び今後の展望

大 槻 明， 坂 野 徹

1. AI (Artificial Intelligence) の概要

CATALYST サイト[1]によると，人工知能（AI）は，図 1 に示す通り特化型人工知能と汎用型人工知能の 2 つに大別される。

図 1 人工知能の定義（出典：CATALYST サイト[1]）



この特化型人工知能の一つとして機械学習，ニューラルネットワーク (Neural network, NN) 及びディープラーニング (Deep learning, DL) が含まれており，2018 年度現在において，AI と呼ばれているものの殆どがこの特化型人工知能である。つまり，機械学習や NN によって実装された人工知能が人間のツールとして決められたタスクを実行するにとどまる。

対して，汎用型人工知能は，人間の感情や創造力までも模倣可能な真の意味での人工知能 (Artificial Intelligence) であり，AI 自らで考え，意思決定をすることが可能となるものである。しかし，汎用型人

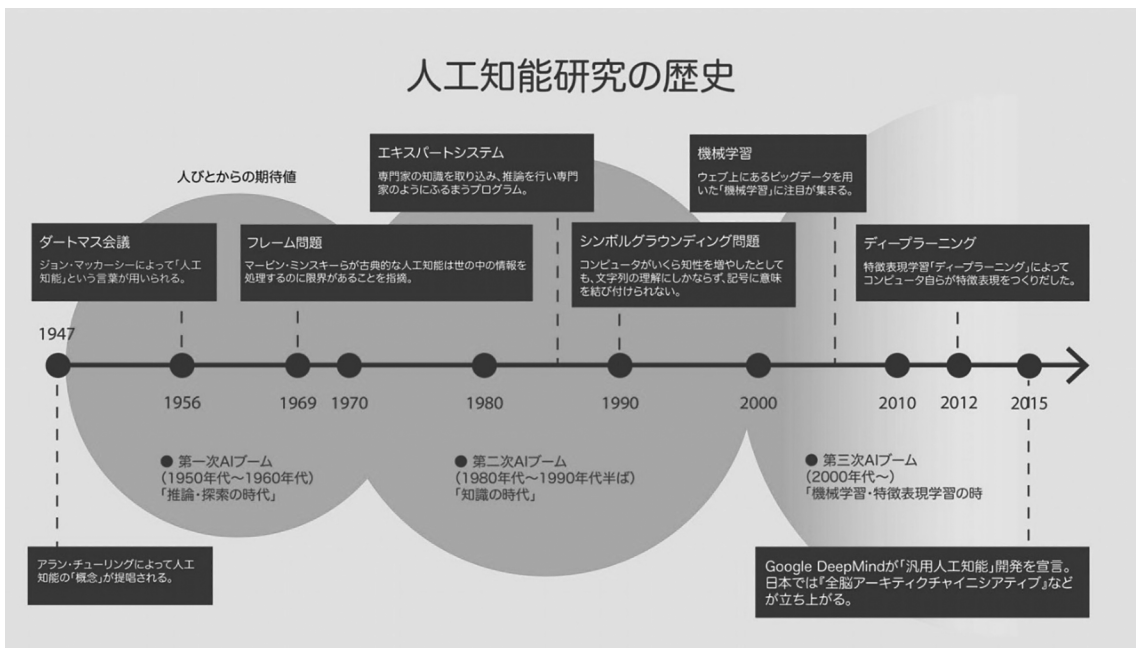
工知能が実現されるためには、人間の知覚、随意運動、思考、推理、記憶など、脳の高次機能を司っているといわれる大脳新皮質が、脳科学的に、また情報学的にモデル化されなければならず、すでにこの手のプロジェクトは始まってはいるが、脳の構造（とくに記憶や思考の過程）は、2018年現在においてもまだ解明されていない[2-3]。

現在のAIブームを牽引しているものはDLであるが、この基礎となっているニューラルネットワーク（ニューロン）は、脳の一部の構造だけを模倣しているに過ぎない。一番知りたいのは脳の「構造」よりも、どうやって知識を引っ張り出して統合して思考して出力しているのか？といったことであり、その神経回路の仕組みが分かれば本当の意味での「人工知能」が誕生すると考えられる。

2. AIの変遷

CATALYST サイト[1]によると、2018年現在のAIブームは第3次AIブームと呼ばれるものであり、AI研究の発端は1950年頃までさかのぼる。次章以降に、第1次～第3次AIブームの概要について述べる。

図2 人工知能研究の歴史（出典：CATALYST サイト[1]）

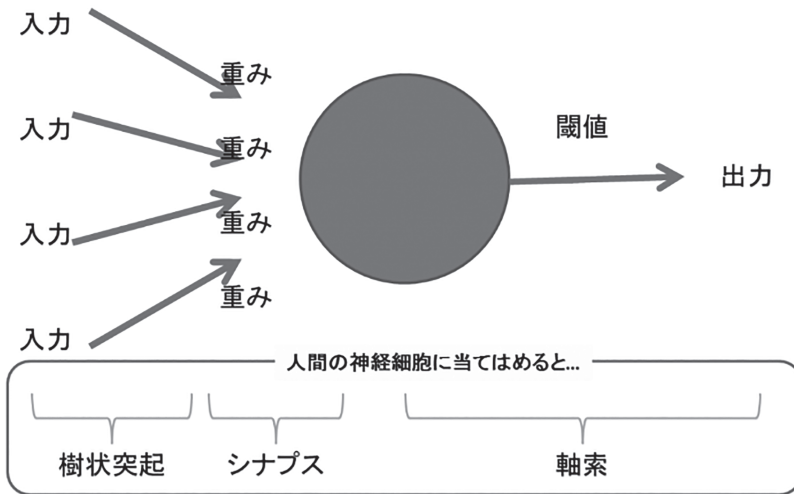


3. 第1次AIブーム（1950年代）

アラン・チューリング (A. M. Turing) [4]によって初めて人工知能 (AI) の概念が提唱された時代であり、機械学習の原型であるパーセプトロンの概念が提唱された時代である。

脳のニューロン構造をモデル化する試行自体は1943年ごろから始まっており、神経生理学者・外科医であるウォーレン・マカロックと論理学者・数学者であるウォルター・ピッツが世界で初めて形式ニューロンを発表した[5] (図3)。

図3 ニューロン (出典: <https://sitest.jp/blog/?p=6727>)
形式ニューロン



その後 1957 年にアメリカの心理学者フランク・ローゼンブラットが形式ニューロンを発展させる形で考案したのがパーセプトロン (Perceptron) である [6]. 入力層と出力層のみの 2 層からなるものを単純パーセプトロンと言い, 単純パーセプトロンを用いてコンピュータの論理演算を実現した一例を表 1~3 に示す. 表 1 が AND ゲート, 表 2 が OR ゲート, 表 3 が NAND ゲートをそれぞれ表す.

表 1 AND ゲート

X_1	X_2	Y
0	0	0
1	0	0
0	1	0
1	1	1

表 2 OR ゲート

X_1	X_2	Y
0	0	0
1	0	1
0	1	1
1	1	1

表 3 NAND ゲート

X_1	X_2	Y
0	0	1
1	0	1
0	1	1
1	1	0

AND ゲート (表 1) は, X_1 と X_2 の入力と共に 1 の時のみ y に 1 を出力する論理演算であり, OR ゲート (表 2) は, X_1 と X_2 のどちらかの入力が 1 の時に y に 1 を出力する論理演算である. そして, NAND ゲートは, y の出力が AND ゲートの逆になる論理演算である.

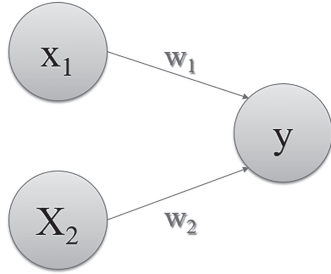
さらに, パーセプトロンを具体的に説明するために, Python でパーセプトロンを実装したイメージを図 4 に示す. w_1, w_2 は重みであり, 入力値 x_1, x_2 にこれら重みがそれぞれ掛け合わされる. そして重み付けされた x_1, x_2 の合計値が閾値 θ を超えた場合は 1 を出力し, θ を超えなかった場合は 0 を出力するようなイメージである.

図4 Perceptron を Python で実装したイメージ

```

① def AND(x1, x2):
    w1, w2, theta = 0.5, 0.5, 0.7
    tmp = x1 * w1 + x2 * w2
    if tmp <= theta:
        return 0
    elif tmp > theta:
        return 1
② print(AND(0, 0))
0
② print(AND(1, 0))
0
② print(AND(0, 1))
0
② print(AND(1, 1))
1

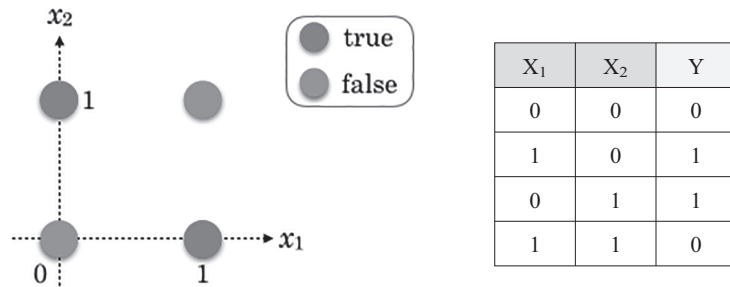
```



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

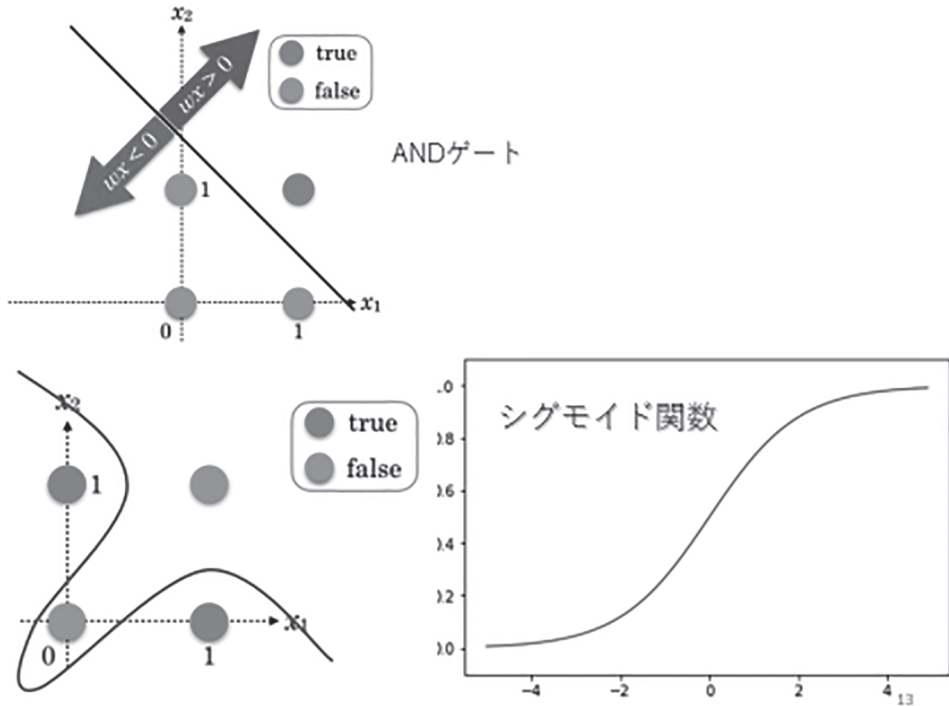
図4及び上述した通り、パーセプトロンはパターン認識を行うことができることから一時的なブームを引き起こしたが、線形分離可能な問題にしか対応する事ができない。つまり XOR ゲート (図5 (a)) を解くことができないことを指摘されたためにそのブームは終焉を迎える。

図5 (a) XOR ゲート (出典 : <http://hokuts.com/2015/12/04/ml3-mlp/>)



具体的には、パーセプトロンは、図5 (b) に示すように AND ゲートのような線形分離可能な場合は対処可能であるが、入力 x1, x2 のどちらかが1の時のみ出力 y に1を出力するような XOR ゲートの場合に対処することができない。この問題を解決するために生まれたものが次節の多層パーセプトロン (ニューラルネットワーク) である。

図5 (b) パーセプトロンの限界（出典：<http://hokuts.com/2015/12/04/ml3-mlp/>）



4. 第2次AIブーム（1980年代）

前節のパーセプトロンの限界を解決するために生まれたものが多層パーセプトロンである（図6）。多層パーセプトロンのアルゴリズムは、入力 x_1 , x_2 のORとNANDをANDでつなぐというものである。

図6 多層パーセプトロンのアルゴリズム

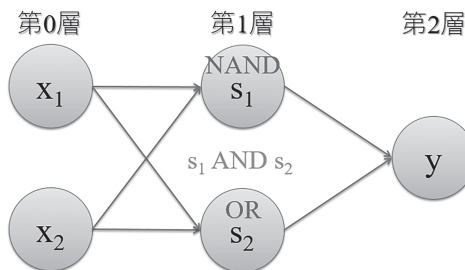
$$x_1 \text{ XOR } x_2 = (x_1 \text{ OR } x_2) \text{ AND } (x_1 \text{ NAND } x_2)$$



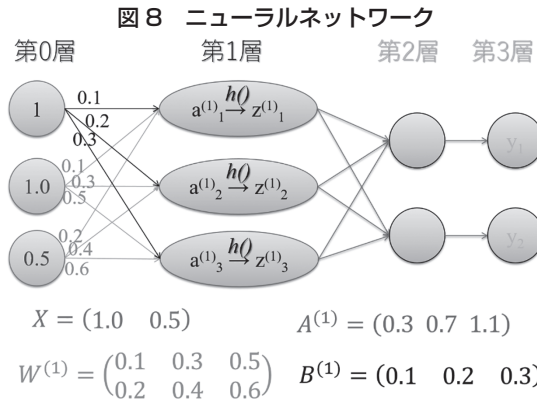
（出典：<http://hokuts.com/2015/12/04/ml3-mlp/>）

そして、この多層パーセプトロンにより XOR ゲートを再現すると図7に示す通りとなる。これがNNの原型となる。

図7 多層パーセプトロンのイメージ

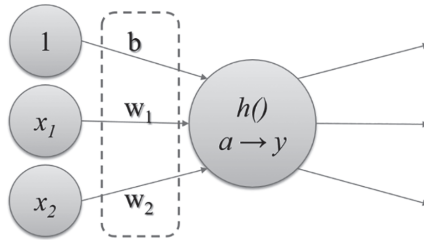


ここで、Perceptron と NN の違いについて整理する。図 8 に 4 層構造のニューラルネットワークを示す。



NN は、ニューロンのつながり方はパーセプトロンと同じであるが、入力値にバイアスを加えることと信号の伝達方法が異なる。NN はシグモイド関数で信号を変換したうえで次のニューロンに伝達する。つまり、図 9 に示す通り、重み付き入力信号の総和が a というノードとなり、それがシグモイド関数 $h(\cdot)$ によって y というノードに変換されたうえで、次のニューロンに信号が伝達される。

図 9 ニューラルネットワークにおける信号伝達



ここで、図 8 を用いて x_1, x_2 の信号を第 1 層で a にまとめ、 a をシグモイド関数にて変換するまでの流れについて述べる。まず、式 1 に X_1, X_2 の信号を第 1 層の a にまとめる式を示す。パーセプトロンと同様な式であるが、重み付き入力値の総和にバイアスを加えている点に違いがある。重み w_n は x_n の重要度を制御するものであるが、バイアスは重み付き入力値の総和の方向性を制御するものである。そして、もし出力を 0, 1 のみ限定する場合は、この重み付き入力値の総和が閾値を超えていれば 1 を、閾値以下であれば 0 を返す。

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

図 8 ではダミーデータも示しているが、このダミーデータにより図 8 の第 1 層の $a^{(1)}_1$ の式を、つまり、重み付き入力値の総和にバイアスを加える式を次の通り示す。

$$0.3(y) = 0.1(b) + 0.1(w_1) \cdot 1.0(x_1) + 0.2(w_2) \cdot 0.5(x_2) \quad (2)$$

次に, 式 2 で求めた 0.3 をシグモイド関数で 0~1 の実数値に変換する. シグモイド関数 $h(x)$ は次式に示す通りである.

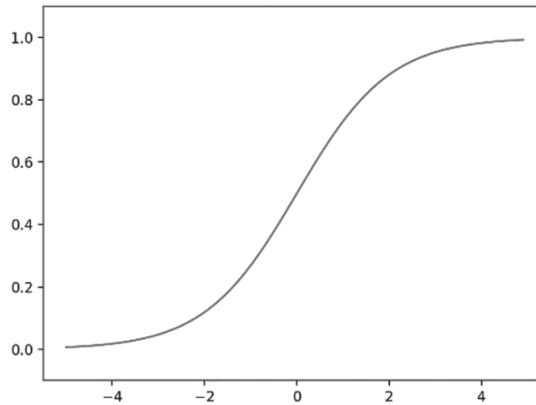
$$h(x) = \frac{1}{1 + \exp^{-x}} \quad (3)$$

$\exp(\)$ は指数関数であり, $e=2.71828\dots$ (自然対数の底, Napier の数) を用いる. $\exp(x)$ は ex と書き, これは「 e の x 乗」を意味する. つまり, $x=1$ の場合式 3 は次式に示す通りとなる.

$$h(1) = \frac{1}{1 + (2.71828)^{-1}} = 0.731\dots \quad (4)$$

NN でシグモイド関数を使う理由は, 図 10 に示すように, a を 0 から 1 の間の実数 (0.37 や 0.98 など) に変換するためである.

図 10 シグモイド関数を使う理由



最後に, NN によって手書き文字の予測を行うイメージを図 11 に示す. 図 11 は入出力層が 1 つずつ, 隠れ層が 2 のイメージである. NN や DL は画像処理に強いといわれているが, その理由は, 図 12 に示すように画像は輝度値と呼ばれるピクセルで構成されており, これらピクセルに色情報などを保持しているが, NN や DL における画像処理では図 11 に示すように, これら輝度値を入力値に設定して予測を行うことができるからである. 具体的には, 第 1 層で $a_1 \sim a_3$ にまとめた (重み付き入力値の総和にバイアスを足した) ものを, シグモイド関数を用いて 0~1 に変換する. これが z である. そして, 第 2 層ではこれら z が入力値となり, 第 1 層と同様の処理を行う. そして, 出力層を数字 0~9 までの 10 個のニューロン ($y_1 \sim y_{10}$) を準備して, $y_1 \sim y_{10}$ の中で一番スコアが高かったものが, 正解として予測された数字である.

図 11 NN による手書き数字の予測

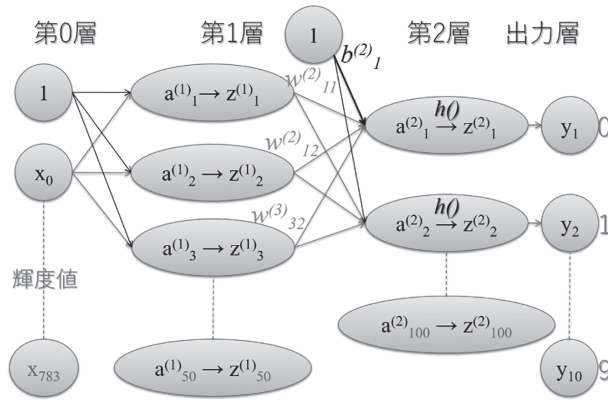
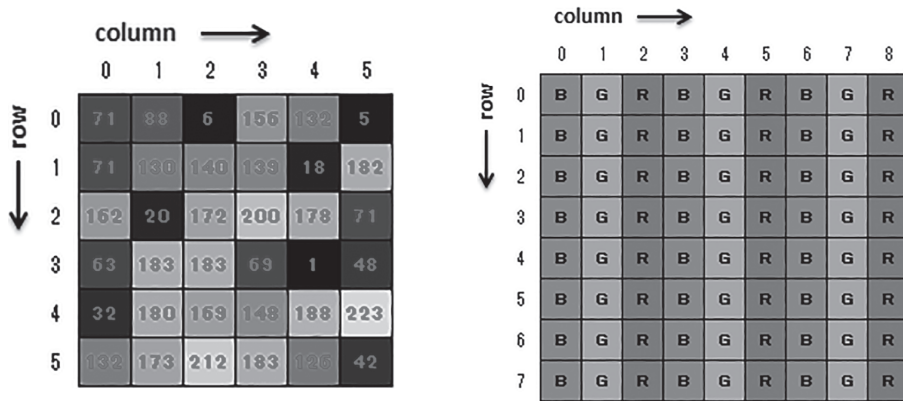


図 12 輝度値



(輝度値, グレースケールの例)

(輝度値, カラーの例)

(出典: https://imaging-solution.net/program/csharp/imagedata-class/get_image_value/)

5. 第3次 AI ブーム (2000 年代以降)

5.1 はじめに

2000 年以降が第 3 次 AI ブームと言われており, AI の代表的な技術として Deep Learning が登場した時代である. Deep Learning [7-10] は, 簡単に述べると図 11 に示した入出力層に, 隠れ層を 2 つ以上追加した 4 層構造以上のネットワークに「誤差逆伝搬法」を加えたものである.

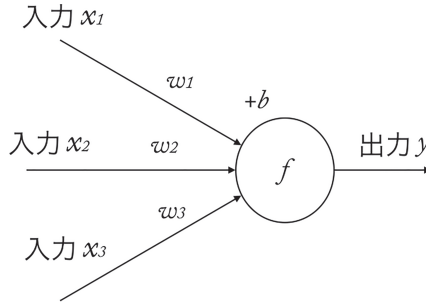
ここで一から整理すると, 入力層と出力層からなる単純なモデルのパーセプトロンの限界は, 図 5 あたりで述べたように, パーセプトロンでは 2 つのクラスを線形識別することしかできないことである. 現実世界での課題は, グラフ上に線を一本引いて 2 つに分類するだけで解決できるものは少ない.

5.2 隠れ層を増やすことで学習能力が向上

ニューロンを複数つなぐことで出力の数も複数に増やせることを図 11 で示したが、隠れ層自体を増やすことで学習能力が向上することが重要な点である。つまり、ニューラルネットワークの大事な性質として、隠れ層の数だけ任意の関数を任意の精度で近似することができることが重要な点である。

入力が複数の場合のイメージを図 13 に表しているが、それぞれの入力 x_1, x_2, x_3 に、重み w_1, w_2, w_3 をそれぞれ掛け合わせ、バイアス b を足して関数 f で処理すれば良いので、出力は、 $h(w_1x_1 + w_2x_2 + w_3x_3 + b)$ となるだけである。

図 13 複数入力のイメージ（出典：[11]）



ここで、 $w_1x_1 + w_2x_2 + w_3x_3$ は、次の行列 W, X の内積で求めることができるため、出力は $h(w^T x + b)$ と表すことができる。

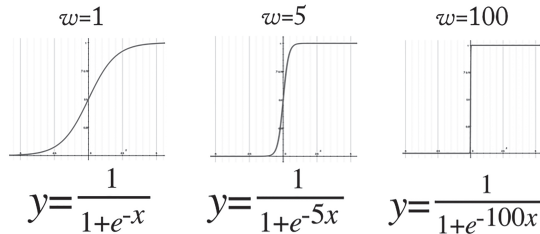
$$W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (5)$$

つまり、式 3, 4 で示したシグモイド関数 $f(x)$ は次式のように表すことができる [11]。

$$h(wx + b) = \frac{1}{1 + \exp^{-(wx+b)}} \quad (6)$$

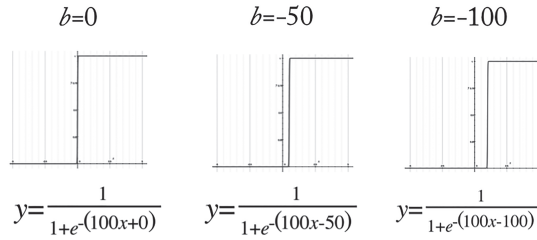
ここで、重み w を $w=1, w=5, w=100$ と変化させた場合 ($b=0$) のイメージを図 14 に示しているが、重み w を大きくするとステップ関数に近づくことが分かる。

図 14 重み w を変化させたイメージ（出典：[11]）



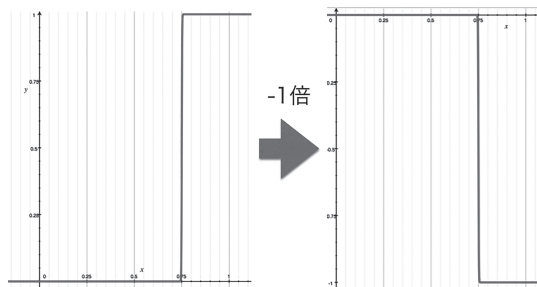
次に、 b の値を $b=0, b=-50, b=-100$ と変化させた場合 ($w=100$) のイメージを図 15 に示しているが、バイアスを変化させても段差の位置が変わるだけである。

図 15 バイアス b を変化させたイメージ (出典: [11])



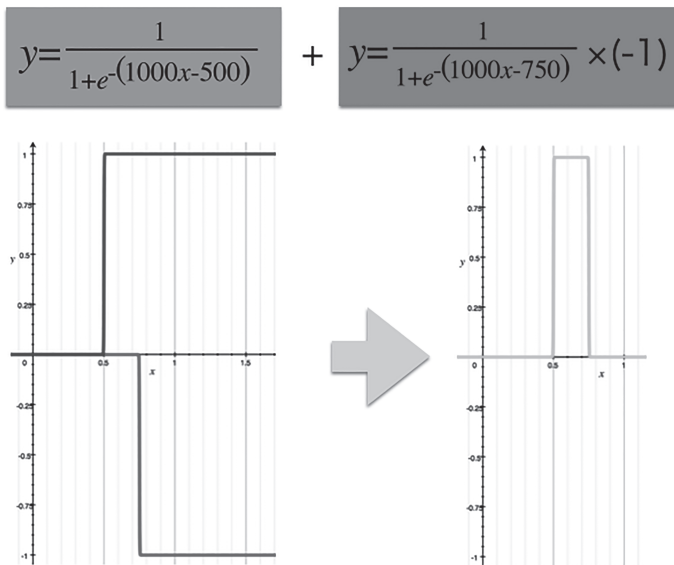
これら 2 つの性質を使うと任意のパルス波を生成することが出来るようになる。例として, $w=1000$, $b=-750$ のシグモイド関数を -1 倍すると図 16 に示す通りとなる。

図 16 $y = \frac{1}{1+exp^{-(1000x-750)}}$ を -1 倍したイメージ (出典: [11])



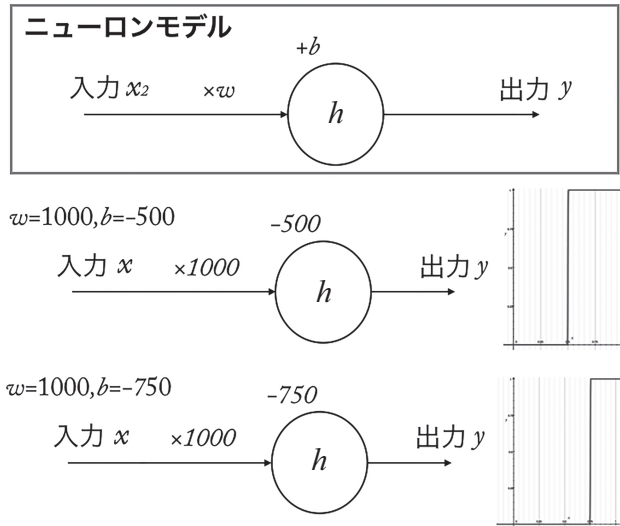
そして, $w=1000$, $b=-500$ のシグモイド関数と足し合わせたイメージを図 17 に示す。パルス波の幅や高さは, w , b の値を変えることで自由に調整することが可能である。

図 17 $w=1000$, $b=-500$ のシグモイド関数と足し合わせてパルス波を作成するイメージ (出典: [11])



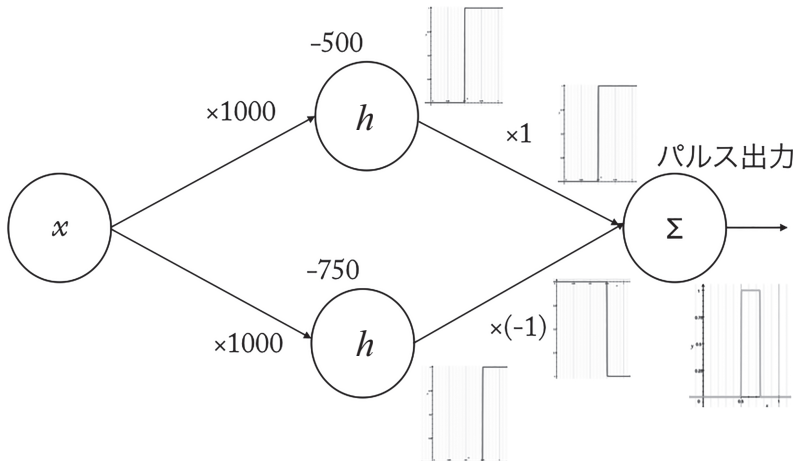
ここまでの内容を，ニューラルネットワークのモデルで表す．まず， $w=1000$ ， $b=-500$ 及び $w=1000$ ， $b=-750$ のシグモイド関数は，図 18 に示すニューロンモデルとなる．

図 18 $w=1000$ ， $b=-500$ 及び $w=1000$ ， $b=-750$ の場合のニューロンモデル（出典：[11]）



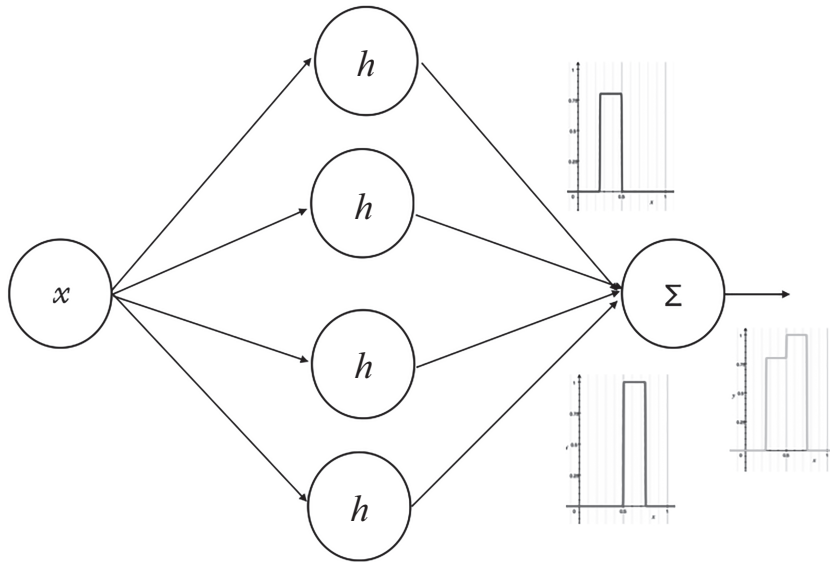
そして，これらを足し合わせてパルス波を生成したイメージ（ニューロンモデル）を図 19 に示す．

図 19 ニューロンモデル（出典：[11]）



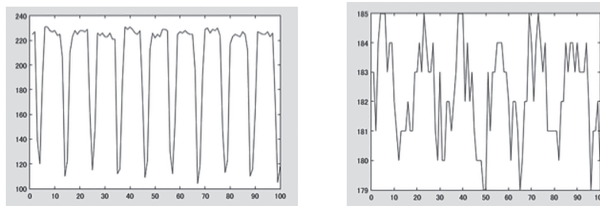
そして，生成したパルス波を足し合わせることで，任意の関数を近似することが可能となる．つまり，図 20 に示す通り中間層の隠れニューロンの数を増やすほど，より近似が正確になっていくということである．そして，実際の課題を解決するときは，手本となる関数を与えて，その関数に近くなるようにニューラルネットワークの重み (w) やバイアス (b) を調整することになる．

図 20 中間層の隠れニューロンの数を増やしたイメージ (出典 : [11])



この簡単なイメージについて述べる。図 21 に示すように、ロボットが柔らかいものと硬いものに触れている時で、センサから送られてくる電気信号の関数の形を記録しておくことで、ロボットが実際にものに触れた時に、実際にセンサから送られてくる電気信号の形がどちらに似ているかどうかで、ロボットは硬いものに触れたのか、柔らかいものに触れたのかを判定することが出来る。

図 21 関数の形 (センサから送られてくる信号) を使ったものを識別するイメージ (出典 : [12])
硬いものに触れたとき 柔らかいものに触れたとき

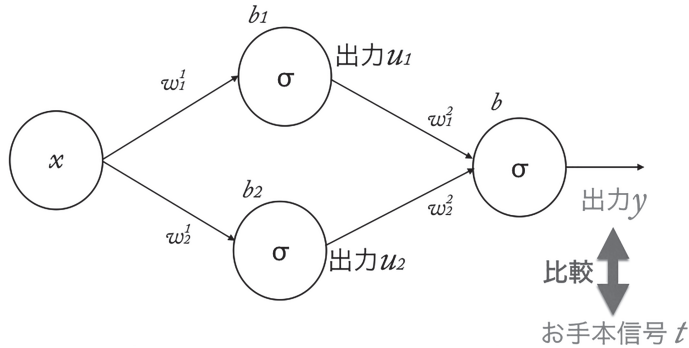


5.3 誤差逆伝搬法と最急降下法

しかし、従来のニューラルネットワークでは、ニューロンの集合体である隠れ層を増やすにつれ誤差も大きくなるため、結果として正しく学習できないという問題を抱えていた。そこで登場したのが「誤差逆伝搬法[8]」である。誤差逆伝搬法とは、逆伝播によって図 11 の重みとバイアスを修正していくアルゴリズムである。具体的には、図 22 に示すようにお手本となる信号 t と出力 y を次式 E により比較し、 E が 0 になるよう重み (w) やバイアス (b) を調整する。

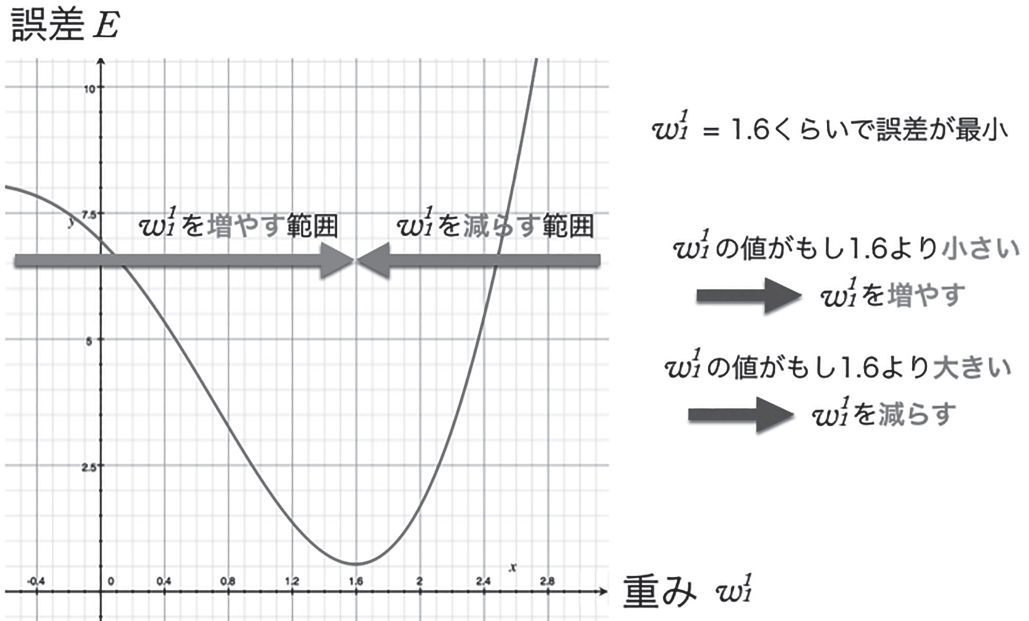
$$E = \frac{1}{2}(y - t)^2 \quad (7)$$

図 22 誤差逆伝搬法（出典：[12]）



誤差 E を縦軸，重み w_1^1 を横軸に取ったグラフが以下の状況であったとする．ここで， $w_1^1 = 1.6$ くらいで誤差が最小となる場合で， $w_1^1 < 1.6$ の場合は w_1^1 の値を増やし， $w_1^1 > 1.6$ の場合は w_1^1 の値を減らす形で誤差 E を調整する．

図 23 誤差 E を調整するイメージ（出典：[12]）



具体的には，図 24 に示す通り傾きに注目して w_1^1 の値を調整する．つまり，接線の傾き $\frac{\partial E}{\partial w_1^1} < 0$ であれば w_1^1 の値を増やし， $\frac{\partial E}{\partial w_1^1} > 0$ であれば w_1^1 の値を減らす．ここで，偏微分概念と連鎖率概念が重要となる．偏微分概念とは，ここで単に 1 つの変数に着目し，他は定数として扱うだけであり，今回の例では，重み w に着目してバイアスを固定するイメージである．

連鎖率は，例えば合成変数の微分では， y が u の関数で u が x の関数であるときに， y を x で微分するには次式のようになる．

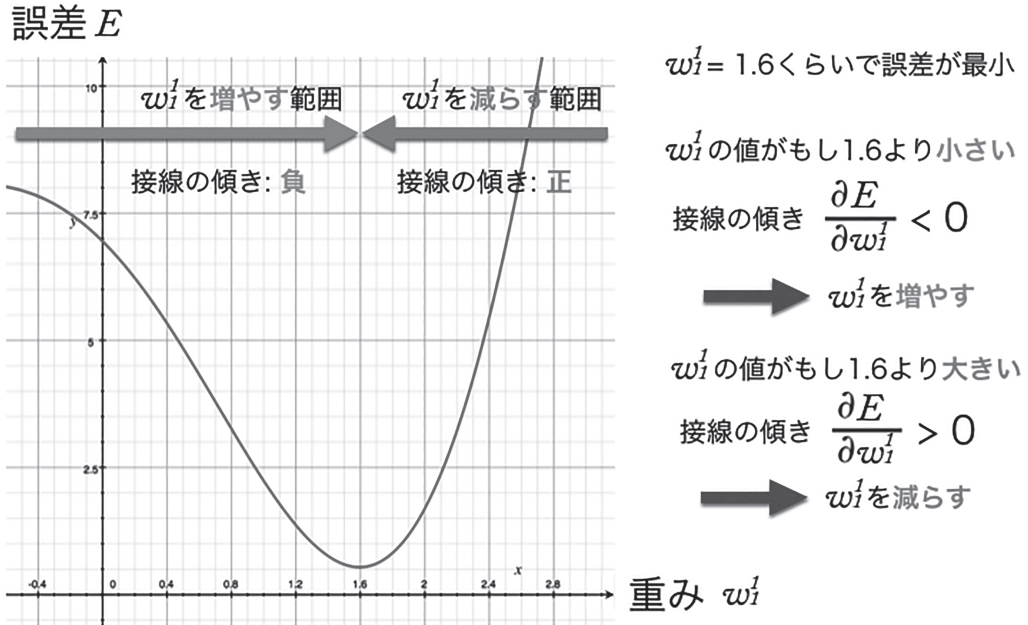
$$\frac{dy}{dx} = \frac{dy du}{du dx} \quad (8)$$

同様に、 z が u の関数で y が (x, y) の関数であるときに、 z を x 及び y で偏微分するには次式のようになる。

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial u} \frac{\partial u}{\partial x} \frac{\partial z}{\partial y} = \frac{\partial z}{\partial u} \frac{\partial u}{\partial y} \quad (9)$$

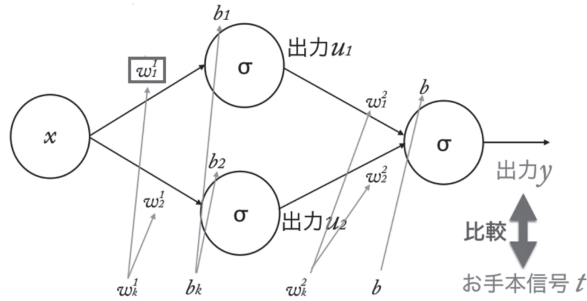
ここで、 ∂ (デル) は偏微分を表し、一番左の分数であれば、分子の z を分母の x で偏微分したことを表す。つまり、図 23 であれば、分子の誤差 E を分母の w_1^1 で偏微分して接線の傾きを求め、この傾きが < 0 であれば w_1^1 を増やし、 > 0 であれば w_1^1 の値を減らすということである。具体的なやり方は、 w_1^1 を反対方向に動かせばよいので、 w_1^1 を $-\alpha \frac{\partial E}{\partial w_1^1}$ で置き換えれば良いことが分かる。ただし、 α は正の値である必要がある。これが、機械学習で良く出てくる最急降下法と呼ばれるものである。

図 24 傾きに注目するイメージ (出典: [12])



本節では、図 24 の w_1^1 の誤差を調整するイメージについて述べたが、 w_k^2 や b_k や b も同様に調整する必要がある。

図 24 傾きに着目するイメージ (出典: [12])



ここで, w_k^1 , w_k^2 , b_k や b を一般化するには, 次式のように更新すれば良く, この更新を繰り返していくことで最終的に誤差を 0 にすることが出来る. 後はこれをコンピュータで何度も繰り返し, 重み w や, バイアス b を修正していくだけである.

$$w_k^1 \text{ を } w_k^1 - \alpha \frac{\partial E}{\partial w_k^1} \text{ に更新}$$

$$w_k^2 \text{ を } w_k^2 - \alpha \frac{\partial E}{\partial w_k^2} \text{ に更新}$$

$$b_k \text{ を } b_k - \alpha \frac{\partial E}{\partial b_k} \text{ に更新}$$

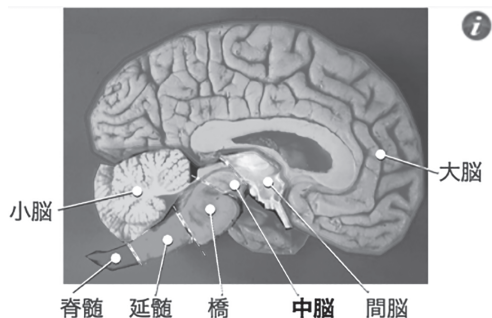
$$b \text{ を } b - \alpha \frac{\partial E}{\partial b} \text{ に更新}$$

(10)

6. 第 4 次 AI ブーム (予想未来)

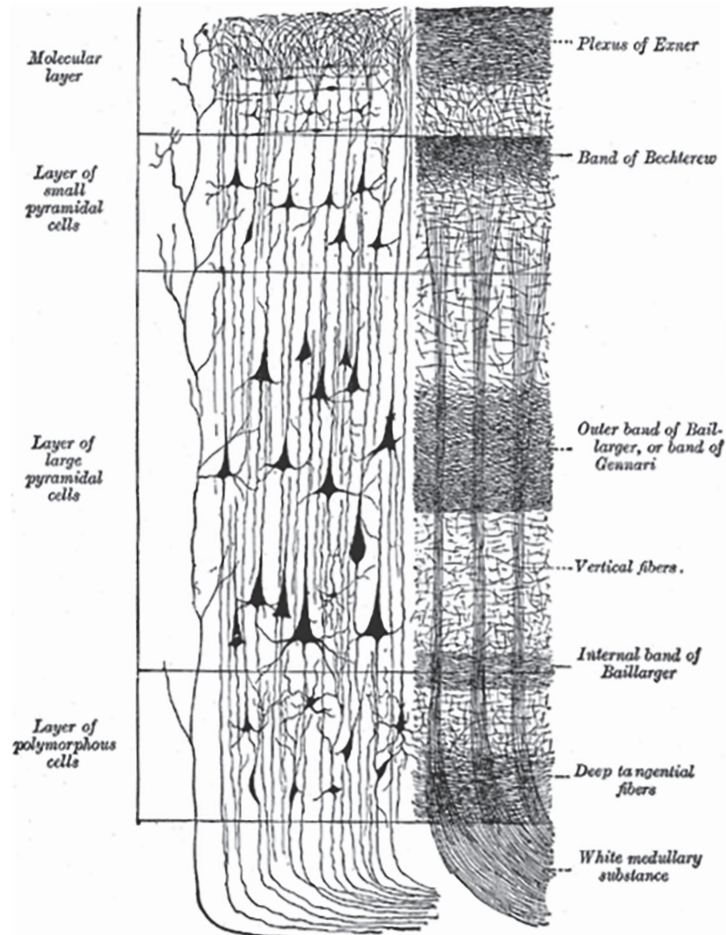
最後に, 著者の独断による視点で第 4 次 AI ブームの展望について簡単に述べたい. 第 3 次 AI ブームにおける Deep Learning は, あくまで人間の脳の表面的な構造を模倣したのみであり, 脳の本質的な部分を実装しているわけではない[13]ので, 現在の AI が感情や意思を持つことはできない. 中原[14]によると, 人間の意思決定に関わる主たる皮質下領域には, 図 25 及び図 26 に示す通り, 大脳皮質や, 大脳基底核 (すなわち線条体, 側座核, 黒質網様部, 淡蒼球), 中脳ドーパミン領域 (黒質緻密部, 腹側被蓋野) および扁桃体などが関係していると指摘している.

図 25 人間の脳の概要



(出典: [https://ja.wikipedia.org/wiki/%E6%A9%8B_\(%E8%84%B3\)](https://ja.wikipedia.org/wiki/%E6%A9%8B_(%E8%84%B3)))

図 26 大脳皮質



(出典 : <https://ja.wikipedia.org/wiki/%E5%A4%A7%E8%84%B3%E7%9A%AE%E8%B3%AA>)

つまり、大脳皮質などの神経回路の構造（軸索結合やシナプス結合選択性など）がモデル化され、このモデルによって、信号（ニューロン）がどのように連携したり、結合したり、流れたりすることで、我々人間が思考や感情などを形成しているのか、ということが脳科学的に、また情報学的に明らかにされて初めて、本当の意味での AI（Artificial Intelligence）が実現されるのだろうと考えられる。

しかし、現時点でもコンピュータは「記憶量」や「計算速度」において人間を超越している。量子コンピュータが汎用的になればこの差はより顕著となる。つまり、現在では、AI は人間のツールという位置付けであるが[15]、本当の意味での AI が実現された後は、AI 自身が感情を持ち意思決定を行えるようになるため、AI はツールの域を飛び越え、人間がその AI を制御することはできなくなるだろう。これがいわゆるシンギュラリティと呼ばれる概念である。ゆえに、我々人間は、この時代を迎える前に AI と人間とがどのように共存できるのかについて議論しなければならないと考えられる。この議論の前提条件として、人間と AI のアイデンティティについてそれぞれ再定義する必要があるだろうし、法制度なども整備しなければならないと考えられる。我々人間はこのシンギュラリティ時代に既に突入していることに危機感をもちながら対応していかなければならないと考える。

参考文献

1. CATALYST サイト : <http://ja.catalyst.red/articles/ai-infographic-01/>, 2018/6/9 閲覧.
2. Correspondence and Fusion of Artificial Intelligence and Brain Science 研究成果 Web サイト : <http://www.brain-ai.jp/jp/publications/>, 2018/5/28 閲覧.
3. 新学術領域研究「人工知能と脳科学の対照と融合」若手サマースクール : <http://www.brain-ai.jp/jp/news/detail/7971/>, 2018/5/28 閲覧.
4. A. M. Turing: Computing Machinery and Intelligence, *The Journal of the Mind Association*, Oxford University Press, (1950), vol. LIX, no. 236, pp. 433-60.
5. Warren S. McCulloch, Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* (Kluwer Academic Publishers), (1943), 5 (4): 115-133.
6. Rosenblatt, Frank. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, (1958), 65 (6): 386-408.
7. 浅川伸一ほか: 深層学習教科書 ディープラーニング G 検定 (ジェネラリスト) 公式テキスト, 翔泳社, 2018.
8. 斎藤康毅: 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, Oreilly Japan, 2016.
9. 我妻幸長: はじめてのディープラーニング -Python で学ぶニューラルネットワークとバックプロパゲーション-, SB Creative, 2018.
10. 岡谷貴之: 深層学習, MLP, 2015.
11. ニューラルネットワークの基礎を初心者向けに解説してみる : <https://www.yukisako.xyz/entry/neural-network>. (2019/7/11 閲覧)
12. 誤差逆伝播法を宇宙一わかりやすく解説してみる : <https://www.yukisako.xyz/entry/backpropagation>. (2019/7/12 閲覧)
13. 人工知能 (AI) の現状と未来 : <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/pdf/n4200000.pdf>. (2019/8/7 閲覧)
14. 強化学習と価値に基づく意思決定の解説 : <https://qiita.com/miyamotok0105/items/9ce8d4425768828628c9>. (2019/8/7 閲覧)
15. 「AI は単なるソフト, 人間の代わりにならない」機械学習の賢い使い方 : <https://japan.zdnet.com/article/35101730/>. (2019/8/7 閲覧)

