産業技術短期大学誌　第 56 巻(2023)
Bull. Coll. Ind. Tech.
Vol.56, pp. 105 - 111 (2023)

教 育
ノート

機械工学と人工知能 (2)

樋口 善彦*

Mechanical Engineering and Artificial Intelligence (2)

Yoshihiko HIGUCHI*

Synopsis: The history of AI (Artificial Intelligence) dates back to the 'Dartmouth Conference' held in1965 when the first AI fad started. After the second fad in the 1980s and the AI winter in the 1990s-2000s, the third fad is in progress. On the back of big data acquired with the Internet, practical techniques are applied in every industry. However, the structures and mechanics are hard for many students to approach. In the present report, the broad outlines of AI including deep learning, and CNN (convolutional neural network) are explained and the specific methodology for the model construction is expounded.

(Received Sep. 12, 2022)

Key words: AI, image recognition, CNN, deep learning, pooling

1．緒　　言

人工知能（AI: Artificial Intelligence）という言葉は 1965 年のダートマス会議で初めて登場し，1960 年代から 1970 年代にかけて第 1 次 AI ブームが起こり大きな期待を集めた．しかし，当時の技術では「トイプロブレム」（Toy Problem）と呼ばれる迷路・パズルなどの単純なルールに基づくゲームしか解けずに期待外れに終わった．1980 年代以降，大型コンピュータの発達を背景に第 2 次 AI ブームが起こり，専門家の知識・技能をコンピュータ上で再現する「エキスパートシステム」の構築が進められた．しかし，このシステムは広範囲で膨大な知識を管理する規則（ルール）を人間が設定する必要があり，管理しきれないという問題に直面した．2010 年代以降からは第 3 次 AI ブームの時代となり実用的な技術が多数開発された．その背景には，インターネットの普及で学習に必要な「ビッグデータ」収集が容易になったこと，中央演算装置（CPU：Central Processing Unit）の演算能力が飛躍的に向上したこと，ルールを機械自ら学習する手法が発達したこと，がある．

しかし，残念なことに AI を高性能だがブラックボックスで個々人が触れる対象ではないという認識が一部でされている．そこで，本稿では AI の概略を解説し，具体的にどのような操作が行われているかを極力シンプルに説明する．

2．人工知能とは

AI は画像認識，音声認識，テキスト処理に応用され，現実社会の中で明確な効果を発揮している．画像認識では顔の表情から感情を判定し，手書き文字をテキスト化するなどの例があり，音声認識では会議録の自動生成，テキスト処理では自動翻訳，などがある．

画像認識 AI で実現されている技術の一つに「物体検出」（Object Detection）があるが，これは対象物を「バウンディングボックス」（Bounding Box）

＊　産業技術短期大学教授　博士（工学）　機械工学科

と呼ばれる四角い枠で囲み，物体の種類（カテゴリー）を判定する．また，「セマンティック・セグメンテーション」（Semantic Segmentation）は対象物の領域を画素（ピクセル）単位で検出する技術であり，画像から説明文章を自動生成（Image Caption Generator)する技術もある．「物体検出」と「セマンティック・セグメンテーション」を統合した「インスタンス・セグメンテーション」（Instance Segmentation）は，第一段階で画像内の対象物の位置をおおまかに検出し，第二段階で物体位置を画素単位で検出する．これにより，複数の物体が重なって写り込んだ画像，あるいは，同一カテゴリーの物体が複数存在する画像も物体位置を正確に把握することが可能である．また，AI 学習用の 1400 万枚の画像を収容した大規模データベースである ImageNet が構築され，それを利用した画像認識モデルのコンテスト ILSVRC(ImageNet Large Scale Visual Recognition Challenge) が 2010 年から開催された．これは画像に映っている物体の種類（カテゴリー）を当てるコンテストで，2012 年から 2017 年にかけてディープラーニング構造のモデルが上位の成績を収めた．

3．人工知能の要素技術

AI に関する代表的なキーワードに，機械学習，ディープラーニングがある．機械学習では多数のデータを機械が自ら学習してデータに潜む規則性等を見つけ出し，それをもとに情報を分類あるいは判定する．人間が設定した規則（ルール）に基づき判定する従来方式では，複雑な事象や大量データの中にある規則を見つけることが困難で実用的ではなかった．機械学習はこの労力を大幅に削減したが，そこではニューラルネットワークという分析手法を高度化したディープラーニング（深層学習）が使われているケースが多い．

機械学習は，教師あり学習(supervised learning)，教師なし学習(unsupervised learning)，強化学習(reinforcement learning)に分類される．教師あり学習は，正解ラベルが与えられているデータ群を使い，入力と正解との間の規則性を学習して，予測値を正解に近づけるものである．一方，教師なし学習は，正解ラベルがついていないデータ群を学習し，データ群を分類する．強化学習は，高い報酬（Reward）が得られる行動を試行錯誤（Trial & Error）で探索する手法である．

ニューラルネットワーク（Neural Network）は脳の神経回路を模擬した分析モデルで，入力層，中間層，出力層の 3 層から構成される（Fig.1）．中間層は隠れ層(hidden layer)と呼ばれ，入力層から来たデータに重み付けと変換を行って出力層へ渡す．出力層が出す予測と正解が比較され，その誤差が小さくなるように重みの値を調整する．この中間層を多層化したものが「ディープラーニング」であり，誤差がより小さくなるように重みが調整（チューニング）される(Fig.2)．

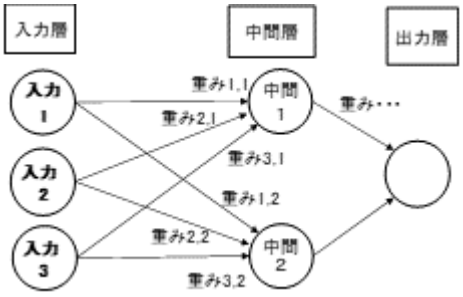


Fig.1　Neural network.

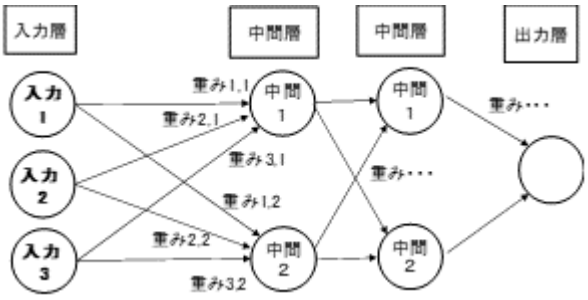


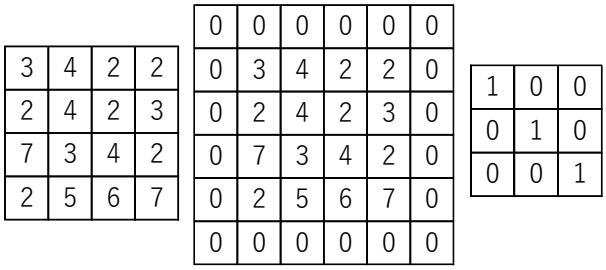
Fig.2　Deep neural network (2 hidden layers).

3.1　畳み込みニューラルネット

AI の予測能力を大幅に向上させた技術は 1989 年にヤン・ルカンらが提案した畳み込みニューラルネットワーク（CNN：Convolutional Neural Network）である．畳み込み層とプーリング層を重ねた構造であり，2012 年に隠れ層 8 層を実装した AlexNet（トロント大）画像認識コンテスト ILVRC で優勝し，2014 年は隠れ層を増やした GoogLeNet（Google），VGGNet（Oxford 大）が誤認識率を更に低減した．その翌年には隠れ層を 152 層にした

ResNet (Microsoft)が誤認識率 3.6%を達成し，人間の認識能力を凌駕するに至った．

畳み込み層で実施される計算の例を Fig.3 に示す．4×4 成分の(a)入力データの周りにゼロのデータを配置する(b)パディングを行い，これに対して 3×3 成分の(c)フィルタ（カーネルともいう）を作用させる例である．ここで，フィルタは対角成分が 1 で，非対角成分が 0 のものを使用している．



(a)input data　　(b)after pudding　　(c)filter

Fig.3　Change in data after operation.

次に，Fig.4 のようにパディング後のデータの左上の着色部分(3×3 成分)にフィルタを作用させると 0+3+4 =7 が得られ，それを 4×4 成分の特徴マップの左上部に代入する．そして，対象を右に 1 つずらし，フィルタを作用させ 0+4+2 =6 を対応する特徴マップの位置に代入する．さらに 1 つずらすと，0+2+3=5，もう一つずらすと，0+2+0=2 が得られ，それぞれを該当する位置に代入する．この一連の操作を下にずらしながら行うと，4×4 成分の特徴マップが出来上がる．

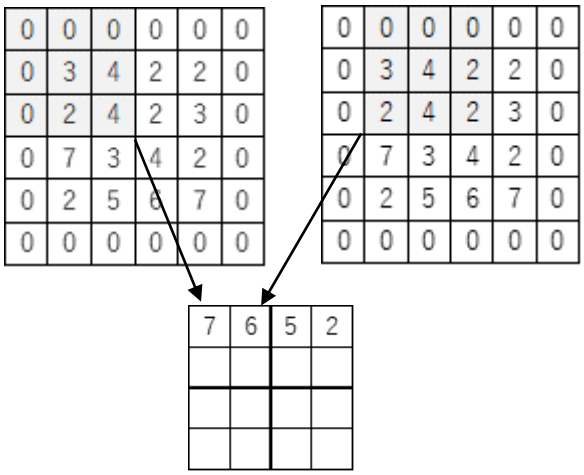
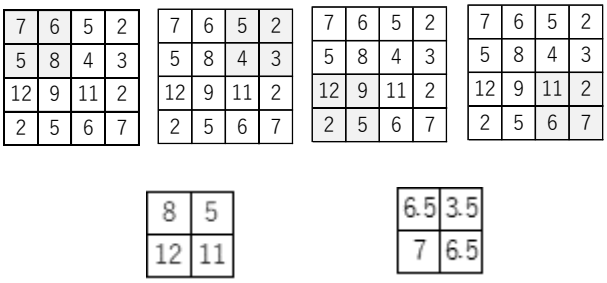


Fig.4 Convolution operation.

このように，入力データにフィルタを適用して特徴を抽出するのが「畳み込み」であり，空間的な配置情報を維持しながら特徴を抽出することができる．最初にパディングで画像の周りを値で埋めたのは，画像の縮小を防ぎ，画像端の情報を抽出するためである．なお，フィルタは抽出目的別に複数設定するため，1 つの入力データからフィルタの種類数と同じ数の特徴マップが得られる．

上述したように，畳み込みを続けると情報量は膨らみ続けることから，重要な特徴を残しながら情報量を圧縮するためにプーリング（Pooling）が行われる（Fig.5）．4×4 の特徴マップに対して，2×2 領域の特徴量として最大値を使うのがマックス・プーリングであり，左端の場合は (7,6,5,8)の最大値である 8 を代表値とする．その右隣は(5,2,4,3)の最大値である 5，次は(12,9,2,5)の最大値の 12，その次は(11,2,6,7)の最大値 11 を取り出して並べると Fig.5(a)の結果が得られる．最大値の代わりに平均値を使う場合はアベレージ・プーリングであり，Fig.5(b)に結果だけを示す．このように，プーリングという操作を行うことにより，特徴を残しながら情報量を圧縮することができる．



(a) max pooling　　(b) average pooling

Fig.7　Pooling operation.

3.2　ソフトマックス関数

以上の操作の後に，出力層で予測値を確率として表示するためにソフトマックス関数が用いられる．これは分類するクラス数を N，i 番目のクラスに関する計算値を x_i，i 番目のクラスの予測確率をy_iとすると，式（1）で表される．全クラスの出力y_iを全て足すと 1 となり，x_i が符号を含めたいかなる数値をとっても，個々のy_iは 0～1 の値をとる．

$$y_i = \frac{e^{x_i}}{\sum_{k=1}^{N} e^{x_k}} \qquad (i,k = 1,2,\cdots N) \tag{1}$$

例えば，手書き数字のクラスは0～9の10クラスあるため，それぞれの確率を表すためには Table 1 に示す One-Hot 形式を用いる．図中にソフトマックス関数の出力値の例を示すが，最も高いy_iの値を示すクラス名「4」がこのモデルの最も確からしい予測結果を表している．

Table 1　Softmax function in One-Hot formatting.

クラス名	0	1	2	3	4	5	6	7	8	9
確率 yi	0.03	0.07	0.02	0.01	0.59	0.19	0.02	0.04	0.02	0.01

3.3　誤差関数と重みの修正

予測と正解との差は誤差，エラー（Error），ロス（Loss）と呼ばれ，これが小さくなるようにモデル内のパラメータである重みを調整する作業が学習である．誤差を算出する誤差関数に使われる二乗和誤差関数と交差エントロピー誤差関数は，分類問題のクラスを N，正解の値をt_i，予測の値をy_i，とすると，それぞれ式(2)，(3)で表される．

$$L = \frac{1}{2}\sum_{i=1}^{N}(t_i - y_i)^2 \tag{2}$$

$$E = -\sum_{i=1}^{N} t_i \cdot log_e(y_i) \tag{3}$$

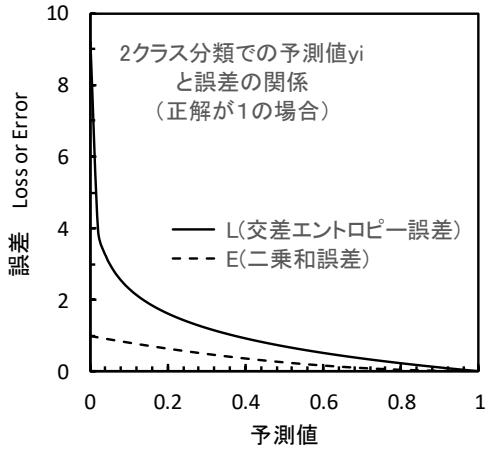
例えば，犬と猫を分類する場合は分類数 N が 2 であり，式(4)，(5)に書き換えることができる．

$$L = \frac{1}{2}\{(t_1 - y_1)^2 + (t_2 - y_2)^2\} \tag{4}$$

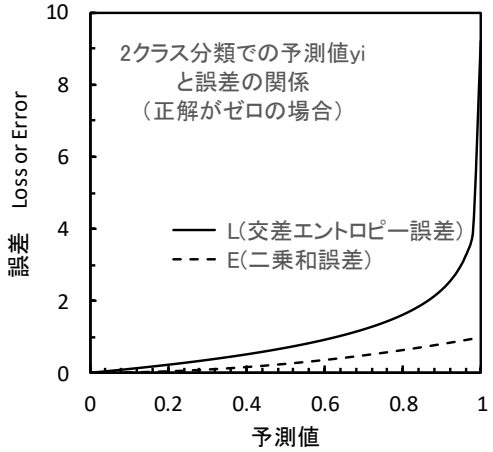
$$E = -\{t_1 \cdot log_e(y_1) + t_2 \cdot log_e(y_2)\} \tag{5}$$

ここで，正解が$t_1 = 1$（$t_2 = 1 - t_1 = 0$）の場合と$t_1 = 0$（$t_2 = 1 - t_2 = 1$）の場合について，予測 y_1（$y_2 = 1 - y_1$）の値と式(3)，(4)から求まる L，Eの関係を求め，それぞれ Fig.6(a),(b)に示す．正解が 1 の場合（$t_1 = 1$）でも，正解が 0 の場合（$t_1 = 0$）でも予測が正解に近づくほど，誤差はゼロに近づく．また，クロスエントロピーの方が正解から外れた場合の誤差の増大が顕著であり，これは誤差の感度が高いことを示している．

学習の最中に予測値が正解に近づくように重みを修正すると誤差が変化する．誤差を最小にする重みが存在すると考え，重み Wを横軸に誤差 Lを縦軸にとった Fig.7 を用いて修正の方法を説明する．初期の重みが W_0 の時の「誤差（Loss）の勾配（$\partial L/\partial W$）」はプラスであるから，誤差を小さくするには重み Wを減らす必要がある．また，初期の重みが W_2の時の誤差の勾配はマイナスであるから，重み W を増やす必要がある．



(a) correct solution=1



(a) correct solution=0

Fig.6 Relation between error and prediction.

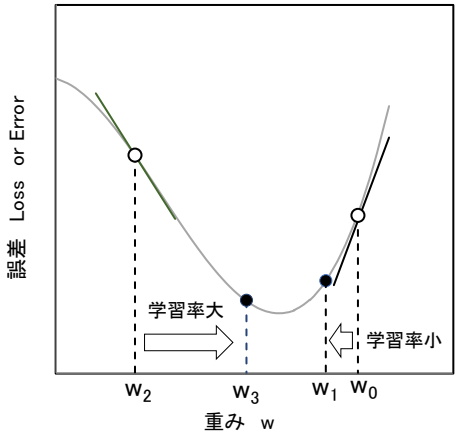


Fig.7 Gradient descent method.

勾配降下法(gradient descent method)では，以上の関係と修正量を調整する学習率αを用いて修正後の重み W' を式 (6) で求める．学習率を大きくすると，重みが大きく変化するため速く最小誤差に到達できる可能性があるが，最小誤差を跨いで行ったり来たりを繰り返してしまい収束しにくくなる可能性もある．そのため，学習率αを学習初期に大きくして，学習後半で小さくする方法により短時間で最適な重みに到達するやり方も考案されている．

$$W' = W - \alpha \cdot \frac{\partial L}{\partial W} \tag{6}$$

誤差の変化に大きな影響を及ぼすのは，通常は出力層に近い部分の重みであるため，重みの修正は出力層に近い部分が先に行われ，だんだんと入力層に近い部分へ進行する．この進め方は誤差逆伝搬(Backpropagation)と呼ばれている（Fig.8）．しかし，深層学習と呼ばれる DNN（Deep Neural Network）では中間層（隠れ層）を多数設定するため，出力層から遠い中間層では重みを変化させても誤差があまり変化しない．すなわち，誤差 (Loss) の勾配（$\partial L/\partial W$）がほとんどゼロになる勾配消失問題が発生しやすく，勾配降下法による重みの修正が不十分になり予測精度の向上が難しくなる．

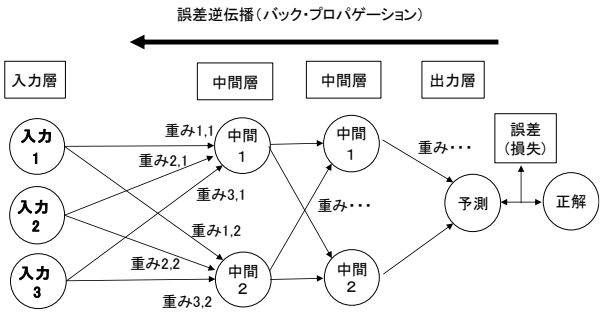


Fig.8 Gradient vanishing in backpropagation.

勾配消失問題の原因の一つに活性化関数（Activation Function）の性質がある．深層学習の中間層では，前の層から受け取った複数の値に重みをかけて足し合わせた後に，値を整えるために活性化関数を作用させて次の層に渡している．この活性化関数に，シグモイド関数（Sigmoid Function），Tanh 関数（双曲線関数），があり，それぞれ式（7），(8) で表される．入力値と出力値の関係を Fig.9 に示す．シグモイド関数の出力範囲は 0～1 であり，分類問題であるクラスに属する確率を表現するのに適しているが，入力値に対する出力値の傾きが比較的小さいために勾配消失問題が起こり易い．出力範囲が-1～1 の Tanh 関数は入力値に対する出力値の勾配がシグモイド関数より大きく勾配消失問題が緩和されるが，入力が大きい場合に傾きが小さくなり，問題は解消されない．それに対し，式 (9) で表される ReLU 関数（Rectified Linear Unit 関数，正規化線形関数）は負の入力値に対してゼロを，正の入力値ではそのまま出力する．そのため，正の入力値では傾きが小さくにならず，勾配消失問題が大きく軽減される．なお，入力値が負の場合に勾配がゼロにならない式(10)の Leaky ReLU 関数も利用されている．

$$f(x) = \frac{1}{1+e^{-x}} \quad Sigmoid\ \ Function \tag{7}$$

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad Tanh\ \ Function \tag{8}$$

$$f(x) = \max(0, x) \quad ReLU\ \ Function \tag{9}$$

$$f(x) = \max(0.001x, x) \quad Leaky\ \ ReLU\ \ Func. \tag{10}$$

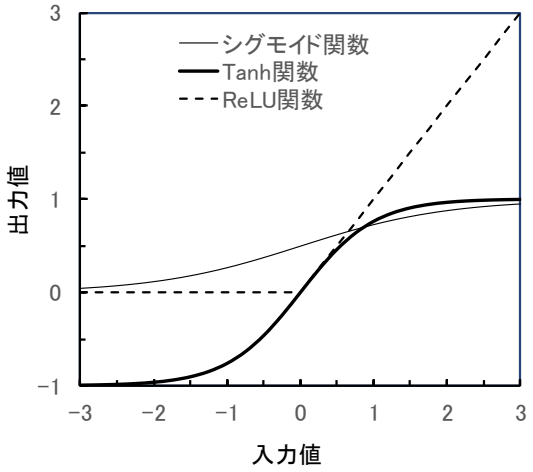


Fig.9 Functions of sigmoid, Tanh and Relu.

3.4 学　習

学習用データの全てを一度に使って重みを学習するバッチ学習は学習状況が安定し易いものの，計算負荷が高くなる．そのため，データを複数のグループに分割して損失関数を計算するミニバッチ学習を用いるのが一般的である．また，一度に使うデータ数が少ないため学習の停滞が起きにくいという利点もある．例えば学習データ全体が 20000 件で，200 件ずつ学習するときのバッチサイズは 200 である．この場合，全データを網羅するには

20000÷200=100 回学習する必要があり，この回数をイテレーション数と呼ぶ．全データを網羅した回数はエポック数であり，この数が多くなると学習モデルの精度が増加し，損失関数が減少する．ここで紹介した設定値や前述の学習率をハイパーパラメーターという．

全データを学習する回数であるエポック数を増やすと学習に使ったデータを再現する訓練精度(training accuracy)は増加するが，学習に使用していないデータを再現する予測精度(prediction accuracy)はあるエポック数で悪化する過学習(over-training, overlearning)が生じる（Fig.10）．これは重み修正を学習データに過度に適合させてしまい，学習に使わないデータに対する予測性能が低下する現象である．なお，Fig.11 に示すように過学習のタイミングは予測誤差が減少から増加に転ずる様子からも読み取ることができる．したがって，設定エポック数に達する前に学習を打ち切る早期終了（Early Stopping）を実施する場合がある．なお，過学習を抑制する手法にドロップアウト（Dropout）という手法がある．Fig.12 に示すように，通常では全節点（ノード，

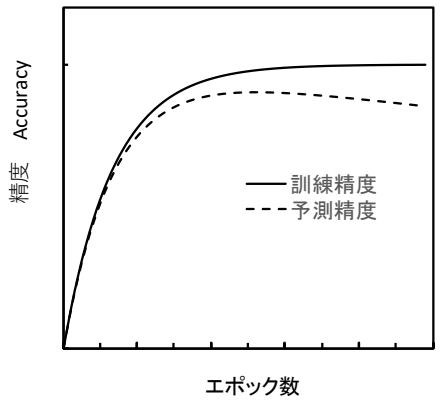


Fig.10 Relation between accuracy and epoch No.

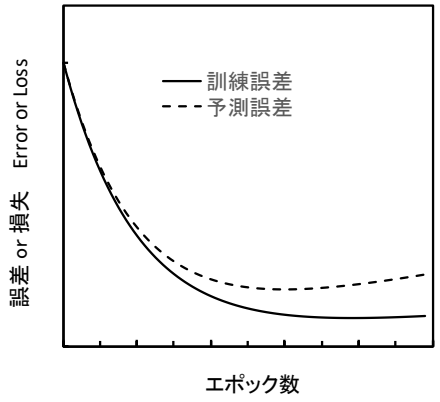
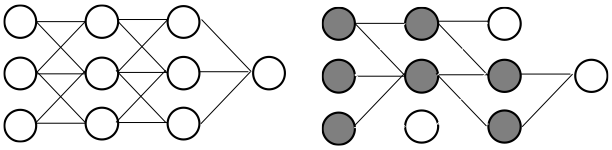


Fig.11 Relation between Loss and epoch No.

node）を接続して学習するが，ドロップアウトでは一定の確率でランダムに一部の節点を無視して学習することで，学習データに特化した重み計算になることを防いでいる．



(a) normal　　(b) dropout

Fig.12 Nodal connections.

４．ディープラーニングのモデル例

ディープラーニングのモデルの中身を Keras を使って設定した場合の例を Fig.13 で説明する．Keras とは，AI 用のフレームワークであるテンソルフロー(TensorFlow)のライブラリである．画像認識の場合，学習する画像サイズをあらかじめ 'in_shape' という変数に，(x_size,y_size,n)の形で定義しておく．ここで，x_size，y_size は画像の x 方向画素数，y 方向画素数で，n はカラーなら 3（3 原色より），白黒なら 1 である．畳み込みを表す 'Conv2D' の引数にフィルタ種類数(32)，フィルタ(カーネル)サイズ(3,3)，活性化関数の種類(relu)を設定している．フィルタは種類の数だけを設定し，Conv2D がフィルタの中身を自動生成する．マックスプーリングは MaxPooling2D(pool_size = (2, 2))のように，(2,2)のサイズを指定するだけである．ドロップアウト処理で指定するのはドロップアウト率であり，Dropout(0.5)であれば，毎回ランダムに選択された 50%の節点が学習から除外される．

```
model = Sequential()
model.add(Conv2D(32,kernel_size=(3,3),
        activation='relu', input_shape=in_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(out_size, activation='softmax'))
```

Fig.13 Example of model structure.

図中 Flatten は多次元データを 1 次元に変換する操作であり，層内の全てのニューロンが次の層の全ニューロンと接続する状態を表し，全結合層と呼ばれている．図中最後の Dense の中でソフトマックス関数が呼ばれており，ここで入力画像の判定結果が出力される．

AI モデルの構造を指定後，Fig.14 に示すように誤差（損失, loss）関数の種類を指定し，学習を実行可能な形式に変換する compile を実行する．図中では交差エントロピー(crossentropy)が指定されている．その次の行の fit で重みをフィット（調整）する学習が始まる．そこでは，学習や予測に使うデータ，バッチサイズ，エポック数などが指定され，変数 hist に学習中の誤差や精度の数値が格納される．学習した重みは 'model' の中に格納されており，model.predict()を呼び出すことにより判定結果を出力することができる．

```
model.compile( loss='categorical_crossentropy',
        optimizer=RMSprop(), metrics=['accuracy'])
hist = model.fit(X_train, y_train,  batch_size=128,
        epochs=12,verbose=1,
        validation_data=(X_test, y_test))
```

Fig.14 Compilation and execution of model.

５．結　　言

AI の基本的事項の理解のために，代表的なテクニックをわかりやすく解説した．ウェブ上で多く公開されている各種の例題はプログラミング言語 Python と TensorFlow を使って書かれていることが多いが，その内容を読み解きながら実践していくことがより深い理解につながると考えられる．

参考文献

1) Michael Wooldridge（神林靖訳）:“AI 技術史: 考える機械への道とディープラーニング”, 2022 年, インプレス.
2) “ゼロからわかる人工知能: 完全版: ついにはじまった AI 時代 社会や暮らしは急速に変化する”, 2022 年, ニュートンプレス.
3) 寺沢幹雄，福田收:“入門情報処理: データサイエンス, AI を学ぶための基礎”, 2022 年, オーム社.
4) 松田雄馬，露木宏志，千葉彌平:“AI・データサイエンスのための図解でわかる数学プログラミング: Google Colaboratory Jupyter Notebook 対応“, 2021 年, ソーテック社.
5) からあげ:“人気ブロガーからあげ先生のとにかく楽しい AI 自作教室”, 2021 年, 日経 BP.
6) 田口善弘:“はじめての機械学習: 中学数学でわかる AI のエッセンス”, 2021 年, 講談社.
7) 清水琢也，小川雄太郎:“AI エンジニアを目指す人のための機械学習入門: 実装しながらアルゴリズムの流れを学ぶ“, 2020 年, 技術評論社.