AI×デザイン
―オープンな開発環境がつくる未来―
Artificial Intelligence & Design

ソーシャルデザイン学科
井　上　貢　一
Koichi INOUE

1．はじめに
本稿の目的は、AIの開発にデザイナーが関わることを前提に、AIの歴史と現状、またその開発環境のトレンドを明らかにしつつ、次世代を担うデザイナーへ向けた提言を行うことにある。

M.マクルーハン流に言えば、デザインの対象となるものはいずれも「人間の拡張」である。第三次ブームと言われる現在のAIは、人間の視聴覚センサーの拡張として、またパターン認識を行う脳領域の拡張として位置付けられるが、それは人間には検知できない情報（例：位置情報）や、ネットワークを介して得られる膨大なデータを背景に、目前の情報を瞬時に、しかも人間よりも高い精度で識別して行動できるレベルに達している。

これからのプロダクトの多くが、クラウドとの接続を前提として、状況認識と自らの行動を自動化しつつある。それは人の所有物としての「道具」の域を超えて、人間をネットワークに接続させてその行動をコントロールする存在でもある。

AIの技術の背景にあるのはデータサイエンスと情報通信技術である。今後デザイナーは多くの現場で、データサイエンティストやITエンジニアと協働することになるであろう。そして、このとき大きな問題になるのが、技術者とデザイナーとの間で「言葉が通じない」ということである。技術者の主たる関心事は製品開発とその性能向上にあり、未来に生じるであろう問題を想像することは後回しになりがちである。人と社会の理想とテクノロジーをうまく融合するには、デザイナーが開発の段階から加わり、デザイン思考にもとづく未来を設計することが必要だ。AIの開発は、もはや他人事ではない。危機感と未来感をもってそれに取り組む必要がある。

2．AIの基本概念
Artificial Intelligenceという言葉は、1956年に開催された世界初の人工知能の国際会議「ダートマス会議」にその起源があるといわれている。

何をもってAIと呼ぶか、現在でも明確な定義はないが、AIに期待される能力には主に、認識・探索・推論の3つがあって、現在のAIというのは、大半が「パターン認識マシン」と同義である。

AIには議論の混乱を避けるために分類上いくつかの区分がある。以下にその用語を概説する。

2.1．汎用型AIと特化型AI
汎用型AIは、人間と同様に様々な問題に対応できる能力を持つもので、現時点では未来のAIと言える。一方特化型AIは、自動運転、画像認識、音声認識など、機能が限定されたもので、現在私たちの身近にあるAIの大半はこれである。

2.2．強いAIと弱いAI
強いAIとは、SF映画のロボットのように物事を認識して仕事をこなすことを想定している。一方、弱いAIは、人間のような意識を備えたものではなく、人間の情報処理機能の一部を代替する機械的な存在である。現在あるAIは、その意味では弱いAIと言える。

2.3．記号処理的人工知能と非記号処理的人工知能
記号処理的人工知能とは、記号処理言語で構築されるもので、処理プロセスはトップダウン的なものになる。記号が厳密に定義されるため限定的なフレーム（文脈）でしか動かない。一方、非記号処理的人工知能は、ニューラルネットワークに代表される機械学習によって構築されるもので、処理プロセスはボトムアップ的になる。記号は厳密な定義を受けないためフレームにも柔軟である。現在の主流は非記号処理的人工知能である。

3．AIの歴史

AIは、過去二回の「ブーム」と「冬の時代」を経て、現在は第三次ブームの時代の只中にあると言われている。一般にブームというものには山と谷があるが、AIのそれは階段状で、コンピュータの進化を背景に、その都度技術がコモディティー化するかたちで推移している。以下、時代ごとの話題を通してその歴史を概観する。

3.1．黎明期　1940年頃～

ダートマス会議以前という意味でのこの黎明期、AIのアイデアとそれを実現するコンピュータが登場している。

1）チューリングマシン

チューリングマシンとは、計算機を数学的に実現するために、1936年に数学者A.M.チューリングが考案したもので、「紙テープ」と「0,1を書き込むヘッド」を備えた仮想的な機械である。「0を書き込んで右に1コマ移動」「紙テープの文字を読んで、左に1コマ戻る」などの命令を並べたプログラムで、計算機能を仮想的に実現する。コンピュータのアイデアの原型といわれる。

2）人工ニューロン

1943年、W.S.マカロックとW.J.ピッツが形式ニューロンというアイデアを発表した。入力信号に重みを乗じた値の総和が一定の閾値を超えると他のニューロンに信号を出力するという人間の脳神経系のモデルで、ニューラルネットワーク（後述）のアイデアの原型とも言える。

3）ENIAC

1946年、世界初の電子式コンピューターENIACが開発された。現在のAIのハード基盤も要は電子計算機。その究極は加算と乗算にすぎない。

4）チューリングテスト

その名のとおりA.M.チューリングが1950年の論文で提案した「人と人工的知性」を見分けるためのテストで、「人がテレタイプを介して対話したときに、相手が人間か機械か区別できないものであれば、それをAIとみなす」とするテストである。人工知能とは何かを説明する際のひとつの指針となっている。

3.2．第一次ブーム　1960年頃～

1）LISP（LISt Processor）

LISPは1958年にJ.マッカーシーが考案したプログラミング言語で、記号やリストと呼ばれる可変長のデータの列を扱うことができ、記号処理系のプログラムの記述・開発に適している。

2）ELIZA

ELIZAはJ.ワイゼンバウムが1964から開発を手がけた会話模倣システム。概念辞書を用いた単純なパターンマッチ技術で、人工無脳（chatbot）と呼ばれる会話ボットの原型となった。

3）エキスパートシステム

記号処理的な推論型のAIで、有識者の知識（推論手順、条件式）をプログラムしたもの。ルールベースのAIと言われるが、人間（有識者）の知識を超えるものではなかった。

3.3．第二次ブーム　1980年頃～

コンピュータの性能向上と低価格化が進み、AIワークステーション、ナレッジエンジニアが登場したこの時期、ITベンダーは続々とエキスパートシステムを導入し、実績をPRした。専用言語LISPから汎用言語Cへの移行期でもある。

1）第5世代コンピュータ

1982年、日本の国家プロジェクトとしてスタートした非ノイマン型のアーキテクチャーで並列処理を行う推論マシン構想。残念ながら一般市場向けの画期的な応用は実現しなかった。

2）バックプロパゲーション（誤差逆伝播法）

D.E.ラメルハートらが1986年に発表したニューラルネットワークの学習アルゴリズムのひとつで、このアルゴリズムが多層ニューラルネットワークにおける機械学習（後述）の可能性を広げた。

3）日本人工知能学会

1986年、日本にAIを専門とする学会が誕生した。

3.4．第三次ブーム　2000年頃～

コンピュータの処理速度と記憶容量の向上、クラウドコンピューティング環境の充実、また様々な分野で「機械が人間に勝利する」という現象がおこり、危機感とともにその話題性が高まっているのが現在、第三次ブームの時代である。

1）IBM DeepBlue

1997年、チェスのプログラムDeepBlueが人間の世界チャンピオンに勝利した。アルゴリズムは「力ずく探索」と呼ばれる従来型のものだったが、その勝利は、第三次ブームの火付け役となった。

2）IBM Watson

2011年、米国のクイズ番組Jeopardy!で、IBMのWatsonが人間のチャンピオンに勝利した。アルゴリズムは従来の探索型自然言語処理だったが、人間に勝利した事実は大きな話題となった。

3）Boston Dynamics BigDOG

Boston Dynamicsは、1992年にM.レイバートが、MITをスピンアウトして設立したロボット開発企業で、米国防高等研究計画局（DARPA）の支援の下で開発した四足歩行ロボット、ビッグドッグ、スポット、また人型ロボットのアトラスなどの姿勢制御技術が高く評価されている。

4）Bonanza / Ponanza

Bonanzaはコンピュータ将棋ソフト。保木邦仁が作成したフリーウェアで、プロの棋譜を機械学習に取り入れたことで2006年のコンピューター将棋選手権で優勝。一方Ponanzaは山本一成による後継的ソフトで、自己対戦の結果で強化学習させた結果、2013年の第2回電王戦で初めて現役の棋士に勝利した。情報処理学会は2015年に「すでにコンピュータ将棋の実力は現時点でトッププロに追い付いている」として、コンピュータ将棋プロジェクトの「終了宣言」を出した。この分野では、シンギュラリティー（特異点越え）がすでに起きていることになる。

5）AlphaGo

2015年、Google DeepMindによって開発された囲碁プログラムAlphaGoが世界のトップ棋士に勝利した。多層ニューラルネットワークを利用した強化学習による勝利は、Ponanzaの話題とともに、機械学習ブームの火付け役となった。

第三次ブームにおける技術的な話題の中心は、機械学習、ニューラルネットワーク、ディープラーニングで、現代のAIの開発には、その概念理解が必須となる。次節以降で順に概説したい。

4．AI開発のキーワード

4.1．機械学習

機械学習（ML）とは、データサイエンス、あるいは非記号型の人工知能研究における手法の一つで、大量のデータを利用した反復的な学習によって、コンピュータ上にパターン認知や推論のためのモデルを構築する技術である。人間がコンピュータに与えるのは、学習するためのルールと、学習素材としてのデータセット。例えば、ヒト、犬、猫の画像と、その画像がヒトか、犬か、猫かという「正解ラベル」である。こうしたデータを大量に与えることで機械は徐々に識別能力を高めていく。このときコンピュータの中に出来上がる「入力と出力の関係づけ」のことを「モデル」という。

機械学習には大きく3つのタイプがある。以下それぞれについて概説する。

1）教師あり学習

教師あり学習とは、ヒト・犬・猫の事例のように問題（説明変数）と答え（目的変数）をセットにして学習させるタイプのものである。回帰、分類、ニューラルネットワークなどの手法があり、迷惑メールの判別や気象予測などを含む様々なパターン認識や予測に利用されている。

2）教師なし学習

教師なし学習とは、与えられたデータから背後にある規則性を発見する手法で、クラスタリングや主成分分析などがそれにあたる。教師なし学習は、正解・不正解が存在しないのが最大の特徴で、その答えを自動的に導き出すことを目的としている。与えられた基準にしたがってデータの分布を学習するもので、おすすめメニューや商品を紹介するレコメンド機能、また電子メールの自動分類などに使われている。

3）強化学習

強化学習では「行動」を入力として、「正解」の代わりに「報酬」を与える。とるべき「行動」に無数の選択肢があり、明確な「答え」をあらかじめ与えることが難しい場合、「報酬」の大きな行動に高いスコア、「報酬」の少ない行動に低いスコアを与えていくことで、どう振る舞えば最大

の報酬が得られるかを学習する。囲碁・将棋のAIに利用されているのが、この強化学習である。

4.2. ニューラルネットワーク

ニューラルネットワーク（NN）とは、人間の脳内にあるニューロン（神経細胞）とその回路網を数式モデルで再現したもので、シナプスの結合強度を学習によって調整することで、入力データを識別する能力を獲得させようとするものである。

図1は1個の人工ニューロンを図式化したものである。n個の入力信号（x_1～x_n）それぞれが重み（w）の影響を受けつつ、細胞に接続され、その総和が閾値（θ）を超えると、細胞が発火して次の細胞へと信号を送り出す。一般に出力（y）は0または1。関係式は以下のようになる。

$$y = f(w_1 \cdot x_1 + w_2 \cdot x_2 + \cdots + w_n \cdot x_n - \theta)$$

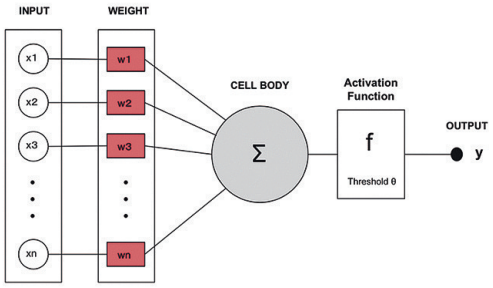

図1．人工ニューロン（単純パーセプトロン）

ニューラルネットワークは、この人工ニューロンを配列した入力層・中間層・出力層を全結合することで最終的な出力を得る。各ニューロンに入る複数の刺激に対する重み（w）を学習によって最適化するアルゴリズムが、先述のバックプロパゲーション（誤差逆伝播法）である。以下がその流れの概略である。

1）ネットワークに学習サンプルを与える。
2）ネットワークの出力の誤差を求め、それを用いて、各出力ニューロンについての誤差を計算。
3）期待された出力と実際の出力の差（局所誤差）が小さくなるよう各ニューロンの重みを調整。
4）より大きな重みで接続された前段のニューロンに対して、局所誤差の原因があると判定し、前段のニューロン、さらにその前のニューロンについて、順次同様の処理を行う。

4.3. ディープラーニング

ディープラーニング（DL：深層学習）はトロント大学のG.E.ヒントンが2006年の論文で用いたディープ・ネットワークという言葉に由来する。ニューラルネットワークにおける中間層の数を数十～百段階程度まで増やして、多段階の神経接続で出力を得ようとするものである。タイプとしては「教師あり機械学習」の一種。現在の主流となっている技術である。図2にその図解を示す。

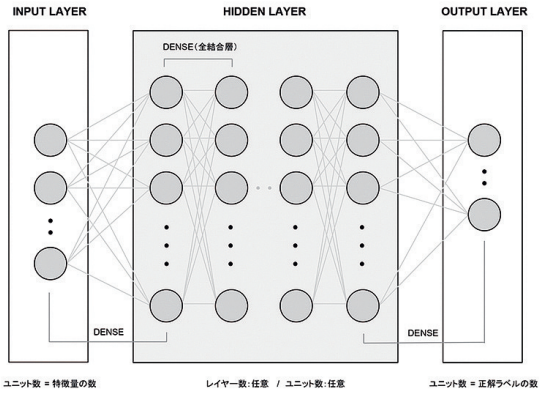

図2．多層ニューラルネットワークのイメージ

入力層には説明変数に対応するユニットを設定し（例えば28×28pxの画像の場合784個）、出力層には判定したいカテゴリーに対応するユニットを設定する（例えば0～9の数字であれば10個）。

問題は中間にある隠れ層の数と各層のユニット数だが、それを決める公式はなく「任意」である。層が多いほど学習の可能性は広がるが、学習は難しく、良い結果が得られるとは限らない。また、一般にモデルのパラメータを増やすと「過学習」が生じると言われる。つまり、訓練データに忠実になりすぎて現実のデータで判定を誤る。要するに頭が固くなる。計算負荷を考え、なるべくコンパクトにモデルを実現すべく調整が必要である。

ここで最も重要なのは学習用のデータである。機械学習には大量のデータセットが必要で、その量と質がAIの性能を左右する。Web上にあふれる大量の音声や画像はもちろん、様々なサービスを通して得られるユーザーの情報が、現在のAI開発にとって最も価値のある存在となっている。

5．開発環境

デザイナーにとってAIの開発はハードルが高いものに思えるが、現在は誰もが簡単に体験できる環境が整っている。以下それらを紹介したい。

5.1．ハードウェア

AI開発に特別なハードウエアは必要ないが、大量のデータを用いた学習モデルの構築には、高速演算を実現するハードウエア（チップ）を利用することも多い。以下、その事例を紹介する。

1）GPU：Graphics Processing Unit

GPUは3DCG用のチップだが、数十から数千のコアを使った並列処理が得意で、大量のデータを扱うディープラーニングにとって、非常に有用な存在である。現在Amazon、IBM、Microsoftなど、大半のクラウド（後述）が、そのトップ企業であるNVIDIAのGPUを採用している。

2）FPGA：Field-Programmable Gate Array

FPGAは、汎用のCPUと高速なGPUの中間的な存在で、購入者が構成を設定できる。FGPAの特徴は省電力で、データセンターのサーバーに適正が高いことから、最近ではMicrosoftのBing検索やAzure翻訳、Amazon AWS、IBMクラウド、Baiduなどで採用されている。

3）ASIC：Application Specific Integrated Circuit

ASICは特定用途向けの集積回路で、量産によって単価を下げられる特徴がある。

4）TPU：Tensor Processing Unit

TPUはGoogleが開発したチップで、一般的なGPUより15～30倍高速である。TPUは前述のASICのひとつで、Google画像検索、Googleフォト、Google翻訳などでも使われている。

5.2．クラウドプラットフォーム

一般に上述のようなハードウエア環境を準備するにはそれなりの経費がかかるが、現在ではクラウドプラットフォームと呼ばれるサービスの利用でそれを代替できる。これはプログラムの実行やデータの保存を可能とする作業環境をWeb上に実現したもので、著名なものにAmazon Web Services（AWS）[*1]、Google Cloud AI[*2]、IBM Watson[*3]、Microsoft Cognitive Services（Azure）[*4]などがある。図3はGoogleTrends[*5]で見た話題性の推移で、AWSの人気が高いことがわかる。

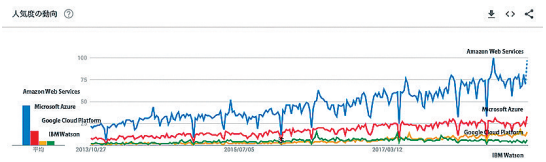

図3．クラウドプラットフォームの話題性の推移（過去5年間）

5.3．開発環境　Google Colaboratory

Google Colaboratory[*6]とは、AI開発の促進を目的としたGoogleのプロジェクトで、アカウントさえあれば、誰でも自由にクラウド上でJupyter Notebook（後述）を利用したPythonのプログラミングが可能になる。ラインタイム環境では、Python 2系/3系を選択でき、またGPUやTPUの利用を選択することもできる。

記述したファイル（ipynb）はGoogleDriveに保存される他、GitHubリポジトリ上への公開も可能となっており、図4のとおり、発表と同時に急速に話題の的になっている。

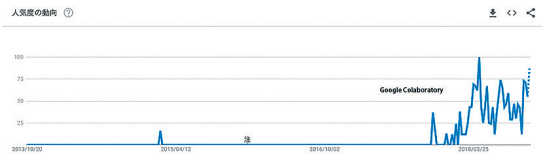

図4．Google Colaboratoryの話題性の推移（過去5年間）

5.4．開発環境　Jupyter Notebook

Jupyter notebook[*7]はPythonの対話型実行環境をノートブック形式で利用できるように拡張した環境で、ソースコードをブロックに区切って逐次的に処理を進めることができる。またプログラムの実行結果や、開発メモをMD（MarkDown）形式で記述できるため、作業の振り返りや、開発メンバーとの情報共有に適している。図5のとおり、その環境は現在の開発のトレンドとなっている。

図5．Jupyter Notebookの話題性の推移（過去5年間）

6. 開発言語/ライブラリ/サンプル

6.1. 言語　Python

Pythonはオランダのグイド・ヴァンロッサムがBBCのコメディ番組『空飛ぶモンティ・パイソン』にちなんで名付けたインタープリタ型の汎用プログラミング言語である。初版は1991年。オブジェクト指向、命令型、手続き型、関数型など、複数のプログラミングパラダイムに対応して、様々な分野のソフトウエア開発に使われている。広範な標準ライブラリとサードパーティのモジュールの充実により、現在のAI開発においては事実上標準の言語となっている。Google社においてもC++、javaと並ぶ3大言語のひとつに位置づけられており、図6のとおり、その話題性は過去5年間で確実に高まっている。

Pythonには、現役で2系と3系のバージョンがあるが、これらには互換性がないため、それぞれに仮装環境を用意する必要があるが、先述のGoogle Colaboratoryでは、それを選択できるようになっており、またローカルマシンにおける開発でも、Anaconda[*8]という統合環境がオープンソースで提供されているため、初心者でも環境の構築でつまづくことは少ない。

図6. プログラミング言語Pythonの話題性の推移　(過去5年間)

6.2. ライブラリ

ライブラリとは、汎用性の高い複数のプログラムを再利用可能な形にまとめたものである。Pythonによる統計処理にはNumPy(数値計算)、Pandas(データ分析)、matplotlib（グラフ描画）といった定番のライブラリがあるが、さらに今日ではWeb上にはAI（機械学習）用のライブラリが多数公開されている。表1はその代表的なもので、大半がPythonを主要言語としている。最も著名なTensorFlowも含め、いずれもオープンソースであり、誰でも自由に利用できる。

表1. 代表的なAIライブラリ（すべてオープンソース）

ライブラリ	言語	開発・サポート	ライセンス	発表
Caffe	C++	Berkeley, facebook	BSD	2013
Caffe2	Python, C++	Berkeley, facebook	BSD	2017
Chainer	Python	Preferred Networks	MIT	2015
Cognitive Toolkit	Python	Microsoft	MIT	2016
Deeplearning4j	Java, C	Adam Gibson	Apache	2014
Keras	Python	François Chollet	MIT	2015
MXNet	Python, C++	Amazon	Apache	2016
PaddlePaddle	Python	Baidu	Apache	2016
Pytorch	Python	Ronan collobert	BSD	2017
Scikit-learn	Python, C++	David Cournapeau	BSD	2007
TensorFlow	Python, C, Java	Google Brain	Apache	2016
Theano	Python	University of Montreal	BSD	2007

6.3. データサンプル

AIの開発を学ぶには、大量のデータサンプルが必要になる。以下、著名なものを紹介する。

1）Iris[*9]

植物学者R.Fisherによる「あやめ」のデータで、統計ソフトの練習用として最も有名。Iris setosa、Iris virginica、Iris versicolor　3種について、がく片の長さ、がく片の幅、花弁の長さ、花弁の幅の4つの計測データが、各50、計150件含まれる。

2）The Boston Housing Dataset[*10]

米国ボストン市郊外における地域別の住宅価格のデータセットで、犯罪発生数、住居の平均部屋数、幹線道路へのアクセスしやすさなど、14項目の情報が506件分記録されている。

3）Wine Quality[*11]

ポルトガルワインの一種Vinho Verdeを測定したデータで、赤ワインと白ワインの2種類のデータ群それぞれに、アルコール、フラバノイド、色彩強度など13種類の成分データと鑑定士による味覚評価が含まれる。

4）MNIST[*12]

手書き数字の画像データ(28×28px)と正解ラベルのペアを学習用60,000個、評価用10,000個、セットにしたデータサンプルである（次節の事例で活用）。

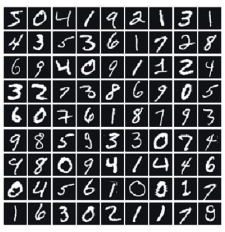

図7. MNISTの画像(一部)

5）MegaFace and MF2[*13]

顔認証の機械学習のサンプルで、約70万人分の顔写真（同一人につき複数画像）470万枚ほどのデータを含む。ただしサイズは160GBを超える。

7. サンプルプログラム

AIのプログラムというと、膨大なコードの記述がイメージされがちだが、実際には多くの手順がライブラリーによってラップされているため、想像以上にシンプルなものになる。

図8は先述の手書き数字サンプルMNISTを用いて、0～9の手書き数字認識モデルをニューラルネットワークの技術で機械学習させるプログラムである。以下、コードを順に概説する。

1）データのインポート

ライブラリからデータを読み込む。xは入力画像、yは正解ラベル。trainが学習用、testが評価用。

2）入力画像データのスケール変換

2次元画像（28×28px）を1次元（784px）に変形。また輝度（0～255）を最小値0、最大値1になるよう正規化している。

3）正解ラベルの対応づけ

0～9の正解ラベルをカテゴリーデータに変形

4）学習モデルの構築

中間層の追加が簡単なSequentialモデルを選択し、入力層はデータ仕様から784、中間層は64と設定した。ニューロンの出力を判定する活性化関数にはrelu（ランプ関数）*14を使用。出力層には正解数10、判定にはSoftmax関数*15を設定。

5）モデルに訓練過程を設定

model.compileで訓練過程を設定する。最適化の手法として、確率的勾配降下法(SGD)、損失関数に多値分類(categorical_crossentropy)を選択。

6）機械学習の実行

model.fitで学習開始。epochsは学習反復回数で、ここでは10回反復。lossは正解とのズレで、0に近いほど正解に近い。またaccuracyは正確性で、100%に近いほど正解に近い。この例では10回の反復計算を20秒程度で終えている。

7）評価用のデータで正解率を検証

評価用データに対する正解率をmodel.evaluateで計算。ここでは94.4%正解するモデルができた。

8）混同行列でミスの発生状況を確認

100件のデータを用いて、実際にどこにミスが生じるかを検証。出力された表は、混同行列といい、

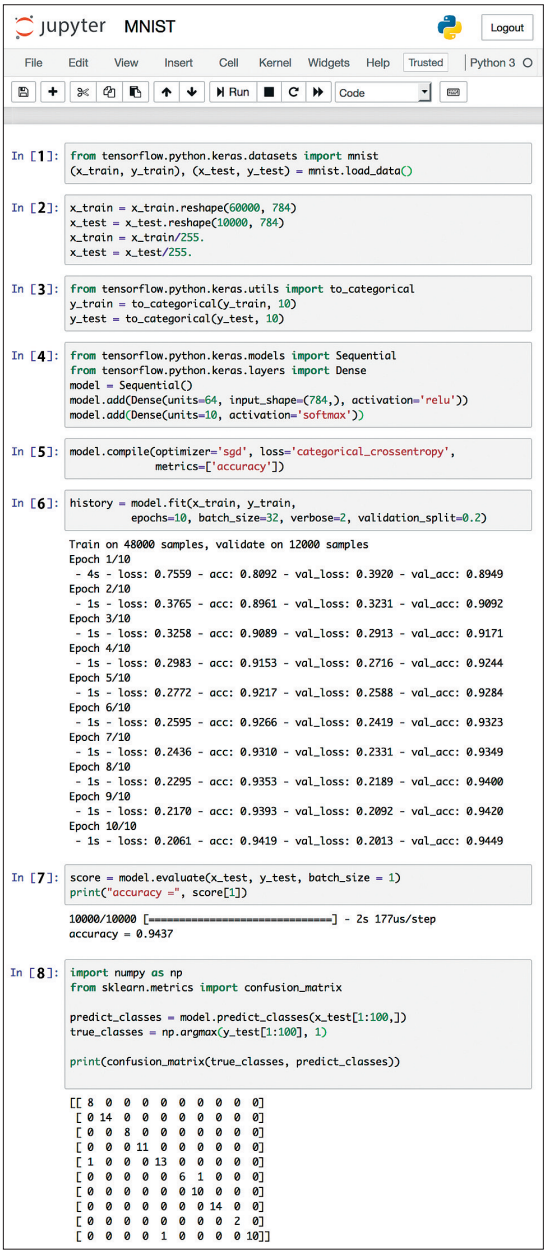

図8．MNIST手書き文字認識のサンプルプログラム

行が予測された数字で、列が実際の正解に対応する。対角線上にあるのが正しく分類されたもので、対角線外のセルは、誤って予測されたものである。表からは、サンプル100件中3件、4と0、5と6、9と4を誤認するエラーがあったことがわかる。

8. 最後に

開発環境、開発言語、ライブラリ、データサンプル、いずれもオープンなものであり、誰もがこれを自由に利用することができる。環境がクラウドにあるということは、PCを持つ必要もなく、必要なのはネットワーク上のアカウントだけである。

デザイナーにとってはすでに十分な環境が整っている。要は知っているか否か、危機感があるか否かだけが問題だといえる。

現代は、AI，ロボット，IoT等をキーワードとする第4次産業革命の只中にある。過去3度の産業革命では、大量の失業者を新たな産業が吸収してきたが、現在の革命はこれまでとは違う。それ自体が人工的な労働者を生み出すわけであるから、革命後の新たな産業に吸収できるのは高度なスキルを持ったわずかな人材のみである。

人手不足が叫ばれてはいるが、それは「高度なスキルを持った人材の不足」と「ロボットよりも安く使える低賃金労働者の不足」とに二極化していて、後者の問題は、ロボットの低価格化によってやがて消滅する。ロボットは24時間働いても文句を言わないし、ルーチンワークのスピードと正確性においてはヒトを上回る能力を発揮する。この先数十年の間に大量の失業者が生まれることは、ほぼ間違いない。

人の仕事が機械に代替される順序は、1）足、2）脳、3）腕、4）顔(表情)、5）手・指の順であると言われるが[*16]、現代は「脳」の代替が急速に進んでいる時代だと言える。AIにおけるパターン認識機能は「ソフトウエア」であるがゆえに開発に要する費用が少なく、モデルが完成すれば瞬時に複製・拡散される。その意味では、この過渡期において「ある日突然」の影響を受けるのは、「パターン認識」や「マニュアル化可能なサービス」に特化した知的作業を生業としている人たちであろう。

結果として、知的作業の代替は、職業訓練のみならず、資格・受験産業、教育機関にも影響する。AIの能力はすでにTOEIC試験で900点レベル、私大模試でも偏差値57[*17]を実現している。教育現場は、まさにこうした状況をふまえた改革を迫られている。第3次産業革命前の社会に求められた能力については、削除するか、あるいは優先順位を下げる必要が生じるであろう。今頃になってプログラミング教育が叫ばれているが、それは第3次産業革命時代に対応済みとなるべき課題であった。しかし残念ながら、一部の成功事例を除いて、この国のIT教育は完全に周回遅れの状況にある[*18]。教育改革は喫緊の課題である。

クリエイティブといわれる業種においてもその代替は始まっている。定型パターンに単語をあてはめれば済む「天気予報」や「新聞記事」の執筆はすでに実用段階にあり、また作曲、作画、小説の執筆（サポート）にも実績がある。さらに、AIが物理現象の観察から「仮説形成」を行った例もある[*19]。良質なデータさえあれば「研究論文」も書けるのだ。人間の強みと思われてきた創造性もすでにAIの能力のひとつとなっている。

さて、代替の最後は「手・指」であるが、ここに至るにはまだ時間がかかると予想される。ロボットは「ハードウエア」であるがゆえに、その開発・製造・輸送に多額の費用と時間がかかるということと、ホムンクルス[*20]のイメージを見れば明らかなように、人間の手・指の感覚と運動には多くの脳領域が割り当てられており、その人工的な再現には、さらなる技術的な進化が必要だと予想されるからである。しかしいずれにせよ、そう遠くない未来に、現在我々が従事する仕事の多くがAIとロボットに代替されることになる。

もはや我々は、パターン化、マニュアル化できるようなスキル、すなわち、点数で評価できるような能力の開発にエネルギーを注ぐべきではない。

ヒトにしか感じることのできない身体的な感覚、他者との共感、問題意識と好奇心、それらの能力を伸ばすことに取り組むべきであろう。

理想の未来と現状とのギャップを埋める方法を模索するのがデザイナーの仕事だとすれば、未来感なしに理想を描くことはできず、理想が描けなければデザインはできない。

AIがもたらす未来には脅威論も楽観論もある

が、確実に言えることは、この流れは止めることができないということである。すでに自動化可能な作業の多くが人から機械へと代替されており、その失業対策として、産業の効率化によって得た富を再配分する「ベーシックインカム」の社会実験も世界各国で始まっているが、これを実現するには、その財源をAIとロボットが稼ぎ出す必要がある。

社会を構成する全員がAIに関する知見と関心を共有し、その活用能力をもって生産性の向上を図らねばならないし、同時にその過程で未来に生じるであろう問題を予見して、「共感」を前提とした社会のルールづくりが必要になるであろう。

前節でみてきたとおり、AIの開発はインターネット革命を引き起こしたオープンな思想の上に進められている。その開発現場を見れば明らかなように、プロジェクトの多くが国境を超えてアイデアを共有し、関係者全員で議論を繰り返しつつ、プロトタイプの修正とテストを繰り返している。政治がどうあれ、すでに世界はつながっている。

理想を共有して、現実とのギャップを埋める。そのためには、技術をオープンにし、情報を共有できる環境を整えることが重要である。

現代に生きる我々は、労働の対価として賃金を得て生活するということを常識と感じているが、そもそもこのような生き方をするようになったのは、ここ数百年の話に過ぎない。貨幣は我々が無意識化した典型的な共同幻想のひとつに過ぎず、その価値は所与のものではないのだ。労働を美徳とする価値観もまた、何世代にも渡る試練を耐え抜くために必要な呪縛に過ぎなかったのかもしれない。

真偽は定かでないが、古代ローマでは「奴隷が働いてローマ人は議論に明け暮れる自由な身分であった」という話がある。本来AIもロボットも、人間が自由を獲得するための便利な機械として考案されたのではなかったか。「失業」を「解放」と考えれば、未来の見え方は変わる。「奴隷的な仕事」から解放されて「新たな価値を創造する暮らし」へシフトする。これを理想と考えて、未来を見据えた能力の開発を早期に行うことが、我々にとって必要なことなのではないだろうか。

ほぼ無尽蔵にあるといっていい太陽エネルギーをうまく活用すれば、食料、生活に必要な物資と環境、そしてAIやロボット自身を、持続可能な方法で生産し続けることができるはずである。

註

1) https://aws.amazon.com/jp/
2) https://cloud.google.com/products/ai/
3) https://www.ibm.com/watson/
4) https://azure.microsoft.com/ja-jp/services/cognitive-services/
5) https://trends.google.co.jp/trends/
6) https://colab.research.google.com/
7) http://jupyter.org/
8) https://www.anaconda.com/
9) http://archive.ics.uci.edu/ml/datasets/Iris
10) http://lib.stat.cmu.edu/datasets/boston
11) https://archive.ics.uci.edu/ml/datasets/wine+quality
12) http://yann.lecun.com/exdb/mnist/
12) http://megaface.cs.washington.edu/
14) ReLU（ランプ関数）：活性化関数のひとつ。入力値が0以下のとき0になり、1より大きいとき入力をそのまま出力する。
15) Softmax関数：出力層の各ユニットに判定結果を%値として出力する。一般に最もパーセンテージの高いものを答えとして採用する。
16) 鈴木貴博，仕事消滅，講談社，2017，p.76
17) 新井紀子，AI vs 教科書が読めない子どもたち，東洋経済，2018
18) 小川博，初中等教育におけるプログラミング教育と一地方での実践，芸術工学会誌 No.77，2018，pp.50-51
19) Michael Schmidt，Hod Lipson，Distilling Free-Form Natural Laws from Experimental Data，2009，Science Vol.324　コンピュータが、揺れる振り子の動きから、運動の法則を導出
20) 脳の中の小人：脳外科医ペンフィールドによるヒトの大脳皮質上の運動野・体性感覚野と体部位との対応関係図。

参考文献

・日本経済新聞社編，AI 2045，日経プレミア，2018，，
・松原　仁，AIに心は宿るのか，集英社，2018，，
・井上智洋，人工知能と経済の未来，文芸春秋，2016，
・田中潤　松本健太郎，誤解だらけの人工知能，光文社，2018
・日経ビッグデータ編，Googleに学ぶディープラーニング，日経BP，2017，
・太田満久他，TensorFlow開発入門，翔泳社，2018，
・吉川隼人，機械学習と深層学習，リックテレコム，2017，
・掌田津耶乃，データ分析ツールJupyter入門，秀和システム，2018，