

人工知能（AI）時代に、 企業は、何をすべきか？

武藤佳恭 慶應義塾大学環境情報学部教授

私自身が経験してきたAIを通して、人工知能時代に、企業は何をすべきかを、本稿で記述します。最後にAI時代に解決すべき課題にも言及します。そもそも、人工知能（AI: artificial intelligence）に関わったのは、1983年8月 就職で米国に移住して、non-tenure-track¹⁾の教員としてフロリダ州タンパ市にある南フロリダ大学のコンピュータサイエンス学科で働き始めたからです。当時、日本の大学でも習ったことがない、AIの授業をいきなり担当させられました。私の指導教授の相磯秀夫先生等が始めた通産省大型プロジェクトの第五世代コンピュータ（AIコンピュータの実現が目標）が世界的にも有名になり、日本人はAIを勉強していると思われたようです。1983年、AIでは、A*アルゴリズム²⁾（A starと発音します）やDijkstraアルゴリズム³⁾などの様々な発見的探索手法や、Prolog言語を使った演繹手法が主流でした。演繹手法（deductive method）とは、簡単に説明すると、3段論法のことです。3段論法とは、AならばBでかつ、BならばCであれば、AならばCであると結論でき

ます。有名な例題を次に示します。

ソクラテスは人間である。すべての人間は死ぬ。この2つの事実（ルール）から、ソクラテスが死ぬかどうかを3段論法で導き出します。Prologでは、この2つのルールを次のように2行で表現します。man(socrates). は、ソクラテスは人間であると言う表現です。また、mortal(X) :- man(X).は、Xが人間であるならば、Xは死ぬ運命にある。Xは変数と呼ばれます。":-"の記号は、右辺の項が正しければ、左辺が成立することを示す"矢印"のことです。つまりB:-Aは、AならばB（ $A \rightarrow B$ ）ということになります。

man(socrates).

mortal(X) :- man(X).

ここで、次のような質問をシステムにすると、true.の答えを返してきます。つまりソクラテスは死ぬ運命にあるということです。

mortal(socrates).

true.

下記サイトから、簡単にprologプログラムが動作できる無料のwebサービスがあります。<https://swish.swi-prolog.org/example/>

examples.swinb

Run!ボタンをクリックすると、mortal(X).を実行し、X=socratesと答えが返ってきます。また、mortal (socrates) .と書き換えて、Run!ボタンをクリックするとtrueの答えが返ってきます。様々な例題があるので、チャレンジして遊んでみてください。

1983年当時、Prologをさらに飛躍的に進化させた自動推論エンジン (otter)を試して驚愕しました。otterはアルゴンヌ国立研究所 (Argonne National Laboratory) で開発されました。アルゴンヌ国立研究所は、3千人以上の研究者を抱える巨大な米国政府の研究機関です。otterを使えば、一流の論理設計者でも解けない難しい論理設計問題を簡単に解くことができます。当時 (1983年頃) は無料でotterを入手できました。1985年以降には、otterは高価なソフトウェアになりました。現在は、オープンソースソフトウェアとして無料で入手できます。otter automated reasoningの3つのキーワードでgoogle検索してください。

現在でも、演繹手法の最高峰はotterであると思います。その理由は、与えられたルールから、新たなルールを生み出すことができるからです。つまり、otterは人間の特性であるはずの創造性・創作力を持っています。そのルール生成は、熱力学に基づいて

いて、複雑な動きをしており、私もさっぱり分かりません。また、その熱力学のルール生成に関しては、ほとんど説明がありません。otterのソースは、公開されているので、覗いてみてください。

理解できたら、教えてください。2018年に卒業生の安藤君と共著で近代科学社からotterの本を出版するので、是非、読んでみてください。otterのプログラムは短いですが、少し難しいかもしれません。

otterで次のパズル問題を解いてみます。ある島に3人 (A,B,C)が住んでいます。3人のそれぞれが、嘘つき、正直者、スパイのいずれかです。嘘つきは、常にうそをつきます。正直者は、常に、正直です。スパイは、うそをついたり、正直になったりします。ある島の3人 (A,B,C)はすべて違う種類の人間 (嘘つき、正直者、スパイ) です。次の情報を、貴方は得ました。

1. "Cは嘘つきだ"とAが言いました。
2. "Aは正直者だ"とBが言いました。
3. Cは"自分がスパイだ"と言いました。

この3つの情報から、A, B, Cの誰が嘘つきで、正直者で、スパイなのか、解りますか？

次のプログラム 1 をotterで実行すると、次の答えを生成します。

P(N(B)). P(L(C)). P(T(A)).

otterによると、Bがスパイ、Cが嘘つき、

Aが正直者となります。

set(hyper_res).

list(usable).

-P(T(x)) | -P(Says(x,y)) | P(y).
-P(L(x)) | -P(Says(x,y)) | -P(y).
P(T(x)) | P(L(x)) | P(N(x)).
-P(T(x)) | P(L(x)).
-P(L(x)) | P(T(x)).
-P(T(A)) | -P(N(B)) | P(L(C)).
-P(L(A)) | -P(N(B)) | P(T(C)).
-P(L(A)) | -P(T(B)) | P(N(C)).
-P(T(A)) | -P(L(B)) | P(N(C)).
-P(N(A)) | -P(L(B)) | P(T(C)).
-P(N(A)) | -P(T(B)) | P(L(C)).

end_of_list.

list(sos).

P(Says(A,L(C))).
P(Says(B,T(A))).
P(Says(C,N(C))).

end_of_list.

プログラム 1 : normal.in 3人 (A,B,C)
のうち、誰がスパイか？

otterは、Ubuntuでは、簡単にインストールできます。次のコマンド(sudo apt install otter)をUbuntu上で実行してください。Ubuntuのインストールに関しては、アンサンブル機械学習やIoTデバイス設計・実装の本を参照してください。

```
$ sudo apt install otter
```

先ほどのプログラム1を次のコマンドでダウンロードします。

```
$ wget http://web.sfc.keio.ac.jp/~takefuji/  
normal.in
```

実行するには、

```
$ otter <normal.in
```

...

```
** KEPT (pick-wt=6): 51 [hyper,49,1,13]  
P(N(B))|P(T(A)).
```

```
given clause #17: (wt=6) 50 [hyper,48,4]  
P(N(B))|P(L(C)).
```

```
** KEPT (pick-wt=3): 52 [hyper,50,2,14,22]  
P(N(B)).
```

人工知能の話に戻りますが、演繹法と対照的な手法が帰納法 (inductive method)です。帰納法の代表が統計手法です。今、流行りのアンサンブル機械学習(ensemble machine learning)やディープラーニング (deep learning) も、統計手法に過ぎません。演繹法では、ルールが正しければ、導き出された結論は100%正しいこととなります。帰納法では、導き出された結果が100%正しいことになることはありません。帰納法のこの弱点は、人間が持っている特性に似ているのかもしれませんが、人間は、難しい問題に直面すると、判断に揺らぎが生じ、間違った判断をすることがあります。帰納法

の精度は、機械学習の手法によって決まってくる。

一番簡単な機械学習法で、比較的 성능が良いアルゴリズムは、アンサンブル機械学習です。通常のCPUマシンでアンサンブル学習は動作します。データが数値データだけである場合は、まず、アンサンブル機械学習を機械学習では、試すべきです。一方、データに画像が多い場合は、画像処理のためのディープラーニングを用います。ディープラーニングは、GPUマシンで機械学習させます。

アンサンブル機械学習やディープラーニングは2つとも統計手法と述べましたが、歴史的にはまったく違う分野から成長・進歩してきています。アンサンブル学習は、最新の統計手法から発展してきています。アンサンブル学習には、アダブースト (Adaboost)、ランダムフォレスト (RandomForest)、エキストラツリー (ExtraTree)、エキストラツリーズ (ExtraTrees)、グラディエントブースティング (GradientBoosting)、バグギング (Bagging)、多数決分類器 (VotingClassifier) などがあります。

ディープラーニングは、ニューラルネットワークから発展してきました。ニューラルネットワークとは、人工ニューラルネットワーク (artificial neural network、ANN)

とも呼ばれ、脳機能に見られるいくつかの単純な特性を計算機上のシミュレーションによって表現することを目指した数学モデルのことです。ディープラーニングのディープとは、ニューラルネットワークの層が深いことを意味します。これと真逆で、ニューラルネットワークの層が浅い場合は、ファンクショナル・リンク・ネット (functional-link net) と呼ばれます。詳しくは、1992年のIEEE Computer, 25, 5, pp.76-79の論文をご覧ください。下記サイトから、pdfファイルで読むことができます。

<http://neuro.sfc.keio.ac.jp/publications/pdf/paocomputer.pdf>

ディープラーニングは、画像処理に非常に向いています。ディープラーニングは、先ほど述べたように、GPUマシンでないと時間がかかりすぎます。ディープラーニングは、まさに、GPUとオープンソースの2つの登場によって、短期間に飛躍的に進歩・成長し、人工知能処理が比較的簡単に実行できるようになりました。

皆さんの使っているパソコンは、恐らく、CPUマシンです。CPUマシンでは、**Central Processing Unit**と呼ばれるモジュールが使われています。同時に計算できるコア (計算ユニット) は、せいぜい2個か4個です。GPUマシンのGPUは**Graphics Processing**

Unitの略で、GTX1080 Ti GPUカードではCUDAコア数が3584個あります。GTX1080 Tiは、1枚10万円程で市場から購入できます。

2018年に登場した、最新のGTX 2080 Ti GPUカードは、5376個のCUDAコア数になります。CUDA (Compute Unified Device Architecture) コアとは、簡単に言うと、並列処理できるストリーム演算のための演算ユニット数と思えば良いかもしれません。大まかな計算スピードを議論する時は、コア数のルートを計算します。3584 \approx 3600なので、通常のパソコンに比べて、 $\sqrt{3600}$ =60倍程度の演算能力が期待できます。

GPU分野では、NVIDIA社が一人勝ちの状態です。NVIDIA社は、最初からAIの会社ではなく、ゲーム専用のビデオカードの会社でした。ゲーム専用のビデオカードで使われているGPUがAI処理に使えることに気づき、NVIDIA社のGPUカード専用のオープンソースソフトウェアを開発しました。また、無料でそれらを提供しています。

NVIDIA社は、オープンソースのディープラーニング基本ツール (CUDA toolkit) を提供しています。NVIDIAのCUDA toolkitをインストールしたあとは、ニューラルネットワークのフレームワークをインストールします。良く使われているディープラーニングフレームワークは、すべてオープン

ソースです。比較的簡単にフレームワークを使うと、所望のシステムが構築できます。現在、AIシステム開発では、Python言語を使うのが主流です。ディープラーニングフレームワークには、

1. Googleが開発したTensorflow、
2. Keras、
3. PyTorch、
4. Chainer (日本製)、
5. Caffe などがあります。

最先端のイメージ処理では、darknetのYOLO (You only look once) の物体認識・検知が有名です。オープンソースのYOLOはC言語で記述されています。

例えば、動物園の写真をYOLOに処理させると、1秒以内に、次の写真を生成します。



写真の原画：<https://www.zoo.org.au/melbourne/wild-encounters/close-up-encounters>

YOLOは、次の結果を生成します。人 (person) が3人、1頭のキリン (giraffe)、1つのbottle (哺乳瓶) があることと、それぞれの位置がわかります。%はその物体の認識率を示しています。物体の位置に関しては、次の駐車場の例で説明します。

bottle: 25%

174, 270, 198, 297

person: 77%

34, 77, 131, 230

person: 35%

124, 96, 217, 303

person: 84%

50, 130, 194, 378

giraffe: 88%

174, 0, 619, 264

このように、画像やビデオから、複数の物体認識や状況認識・状態認識が可能になってきています。

次に、駐車場の例を示します。GPUマシンを使って、駐車場の原画をYOLOに与えると、直ちにその結果と車の位置 (x1, y1, x2, y2) を示してくれます。carやtruckの位置は、発見した物体ボックスのleft top (x1, y1) と right bottom (x2, y2) を表していて、写真での (x, y) 座標を示しています。写真の左上が原点座標 (0,0) になります。このように、物体認識した車の位置を把握できるの

で、全自動の駐車場管理システムや駐車位置へのナビゲーションが可能になります。



駐車場Cの原画 : <http://www.c-kumiai.net/service/warehouse.html>

car: 31%

513, 235, 556, 266

...

truck: 65%

154, 207, 308, 364

更に、YOLOを使って面白い実験をしたので、ここで紹介します。人間にはシミュラクラ現象があります。Wikipediaによれば、シミュラクラ現象 (シミュラクラげんしょう、英: Simulacra) とは、人間の目には3つの点が集まった図形を人の顔と見るようにプログラムされているという脳の働きのことです。人間のこの錯覚現象を、YOLOで試して見ました。

simulacra door dogの3つのキーワードを使ってgoogle検索し、使えそうな画像を見つけ出します。次の画像を発見したので、YOLOに与えてみました。



<https://soraneews24.com/2014/12/07/dog-seems-to-appear-from-plank-of-wood-in-either-coolest-pet-trick-or-silliest-astral-projection/>

YOLOの結果を次に表示します。ドアの節目をdogと判断したようです。GPUマシンでなくても、普通のパソコンであれば20秒から30秒ぐらいで結果がでます。GPUマシンでは、1秒以内です。



YOLOの結果

有名なシミュラクラ現象の板切れをYOLOで試してみました。この板切れの節目を、先ほどと同様、YOLOが犬と錯覚しました。

YOLOが出力する詳細結果を、次に示しま

す。このYOLOでは、32層のニューラルネットワーク構成を使っています。昔の単純なsigmoid関数のニューロンモデルだけでなく、YOLOでは、畳み込み層:conv（空間的な情報を維持）、マックスプーリング層:max（入力データをより扱いやすい形に変形するための情報圧縮）、ルート層:route（細かい粒度の特徴抽出）、reorg層:reorg（後層の特徴マップのサイズを一致させるため）などあります。

| layer | filters | size | input | output |
|--------|---------|-------|--------------|-----------------|
| 0conv | 32 | 3x3/1 | 608x608x 3 | -> 608x608x 32 |
| 1max | | 2x2/2 | 608x608x 32 | -> 304x304x 32 |
| 2conv | 64 | 3x3/1 | 304x304x 32 | -> 304x304x 64 |
| 3max | | 2x2/2 | 304x304x 64 | -> 152x152x 64 |
| 4conv | 128 | 3x3/1 | 152x152x 64 | -> 152x152x 128 |
| 5conv | 64 | 1x1/1 | 152x152x 128 | -> 152x152x 64 |
| 6conv | 128 | 3x3/1 | 152x152x 64 | -> 152x152x 128 |
| 7max | | 2x2/2 | 152x152x 128 | -> 76x76x 128 |
| 8conv | 256 | 3x3/1 | 76x76x 128 | -> 76x76x 256 |
| 9conv | 128 | 1x1/1 | 76x76x 256 | -> 76x76x 128 |
| 10conv | 256 | 3x3/1 | 76x76x 128 | -> 76x76x 256 |
| 11max | | 2x2/2 | 76x76x 256 | -> 38x38x 256 |
| 12conv | 512 | 3x3/1 | 38x38x 256 | -> 38x38x 512 |
| 13conv | 256 | 1x1/1 | 38x38x 512 | -> 38x38x 256 |
| 14conv | 512 | 3x3/1 | 38x38x 256 | -> 38x38x 512 |
| 15conv | 256 | 1x1/1 | 38x38x 512 | -> 38x38x 256 |
| 16conv | 512 | 3x3/1 | 38x38x 256 | -> 38x38x 512 |
| 17max | | 2x2/2 | 38x38x 512 | -> 19x19x 512 |
| 18conv | 1024 | 3x3/1 | 19x19x 512 | -> 19x19x 1024 |
| 19conv | 512 | 1x1/1 | 19x19x 1024 | -> 19x19x 512 |
| 20conv | 1024 | 3x3/1 | 19x19x 512 | -> 19x19x 1024 |
| 21conv | 512 | 1x1/1 | 19x19x 1024 | -> 19x19x 512 |
| 22conv | 1024 | 3x3/1 | 19x19x 512 | -> 19x19x 1024 |
| 23conv | 1024 | 3x3/1 | 19x19x 1024 | -> 19x19x 1024 |

```
24conv 1024 3x3/1 19x19x 1024 -> 19x19x 1024
25route 16
26conv 64 1x1/1 38x38x 512 -> 38x38x 64
27reorg /2 38x38x 64 -> 19x19x 256
28route 27 24
29conv 1024 3x3/1 19x19x 1280 -> 19x19x 1024
30conv 425 1x1/1 19x19x 1024 -> 19x19x 425
31detection
mask_scale: Using default '1.000000'
Loading weights from yolo.weights...Done!
dogT1.jpg: Predicted in 0.017147 seconds.
dog: 58%
71, 12, 339, 318
```



<http://www.ohmidog.com/tag/simulacra/>

GPU並列処理マシンの組み立て方は、GPU parallel computing for machine learning in Pythonの本を2017年に出版しま

した。この本は、amazon.co.jpからkindleで読むことができます。GTX 1080 Ti GPUカードを使って、25万円～30万円ほどで、GPU並列処理マシンを自作構築できます。この事実は、"個人で、スーパーコンピュータを自分で組み立てたり、活用することが可能になった"ということです。今や、スーパーコンピュータを個人で持てる時代になったわけです。GPUマシンを使うために必要なソフトウェアは、すべて無料のオープンソースソフトウェアを利用することが出来ます。

そもそも、機械学習とは何であるかを説明します。ここで、入力をX、出力をyとした場合、機械学習とは、 $y=f(X)$ の関数fを求めることになります。この関数fを比較的簡単に計算できる、優秀なアルゴリズムがオープンソースで登場しました。アンサンブル機械学習の本では、アイスクリームのデータを使って説明しています。アイスクリームの売り上げを予測するに当たり、その日の最高気温 (x1) とアイスクリーム店頭を通過する通行人数 (x2) のデータを使います。

従来の手法では、専門家がデータを眺めながら、専門知識を使って重回帰モデルを作成し、データを使って重回帰式のパラメータ (temp, street, c) を決定します。

$$y = \text{temp} * x1 + \text{street} * x2 + c$$

一方、機械学習では、モデルはデータから自動的に作成します。機械学習では、データさえあれば、モデル構築に専門知識は要りません。様々な分野には、それぞれの専門職のモデルビルダーがありますが、将来、それらの仕事はなくなるでしょう。そもそも、複雑なシステムでは入力パラメータの数 (n) が数十や数百ある場合は、専門家の直感を駆使しても重回帰モデルを構築するのは不可能です。予測したい出力Yは、次式のようになります。

$$Y = f(x1, x2, \dots, xn) \quad \text{ここで、} Y = y1, y2, \dots, ym$$

アンサンブル機械学習とは簡単に言えば、従来のいくつかの機械学習の“良いところ取り”をした計算手法です。GPUマシンでなくても、従来のCPUパソコンでも高速に機械学習し、予測することができます。scikit-learnと呼ばれる、オープンソースのアンサンブル機械学習が有名です。scikit-learnのインストールに関しては、アンサンブル機械学習を参照してください。

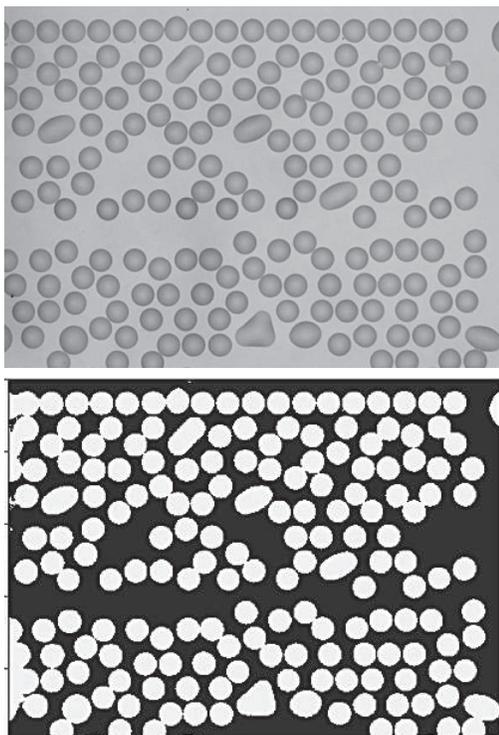
現在、私が取り組んでいる機械学習プロジェクトは、8つあります。

1. モリタホールディングのモリタ・宮田と共同で人工知能搭載のスプリンクラーを構築中です。危ない炎を発見し、

初期消火を実現するのが目的です。

2. 東亜利根と共同で、人工知能搭載の掘削ボーリングマシンを構築中です。ボーリングマシンのパラメータから、SPT（硬さ）や層を予測して、最適なマシン制御をします。
3. 大同工業と共同で、人工知能を使って、チェーンの寿命を判定するのが目的です。
4. 前兆から、犯罪を予測しようとする新しいプロジェクトです。（警視庁）
5. ABC店舗と共同で、不動産サブリースの問題を人工知能で解決中です。
6. 環境リサーチと共同で、オリンピックに向けて、人間よりも早くトイレの匂いを検知しに問い問題を解決できる、人工知能搭載の人工鼻を構築中です。
7. 環境リサーチと共同でアスベストの自動検知システムを構築中です。
8. 大勇フリーズと共同で、液体窒素を使った水道管内での氷形成予測システムを構築中です。

人工知能を使えば、物体数を数えたりすることが簡単にできます。下記プログラムで実行すると次の結果が得られます。結果は、185個の物体数であることがわかります。Python言語では、比較的プログラム（11行）が短くなります。



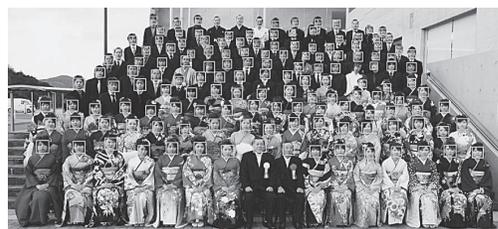
物体の図：count.jpg

```
from skimage import io, filters
from scipy import ndimage
import matplotlib.pyplot as plt
from skimage import measure
im = io.imread('count.jpg', as_grayscale=True)
val = filters.threshold_otsu(im)
drops = ndimage.binary_fill_holes(im <
val)
labels = measure.label(drops)
plt.imshow(drops, cmap='gray')
print(labels.max())
plt.show()
```

物体数を数えるPythonプログラム

淡路島の成人式の参加者数を調べてみましょう。結果は148人ですが、一人の男性の顔が斜めに写っていて判定に失敗しています。実際は149人です。認識できなかった斜めの人でも学習しておけば、ミスなくカウントすることができるようになります。下記サイトから人数を認識するPythonプログラムをダウンロードして、実行してください。詳しくは、アンサンブル機械学習の本を参照してください。

<http://web.sfc.keio.ac.jp/~takefuji/face.py>



淡路島の成人式の集合写真

私が最初に解決したニューラルネットワークの問題は、1989年のグラフの平面埋め込み問題です。私がケースウエスタンリザーブ大学にいた時、どうやったら有名な研究者になれるか、大学内の著名な先生に

直接聞きに行きました。この大学にはノーベル賞級の研究者がたくさんいました。その先生は、"ScienceかNatureに出版しなさい"と言われました。コンピュータ分野の私にはそれらの雑誌にまったく馴染みがありませんでした。Scienceに出そうと思いましたが、コンピュータの論文はほとんどありません。そこで考えたのが、次の作戦です。コンピュータの分野にはノーベル賞はありませんが、それに代わって、チューリング賞があります。つまり、ニューラルネットワークに向けた問題を見つけ、チューリング賞をもらった人達の研究成果と比較することを思いつきました。それが、グラフの平面埋め込み問題です。19本の線しか平面に埋め込めなかった従来手法に対して、我々のニューラルネットワーク手法は20本の平面埋め込みに成功しました。これが、世界

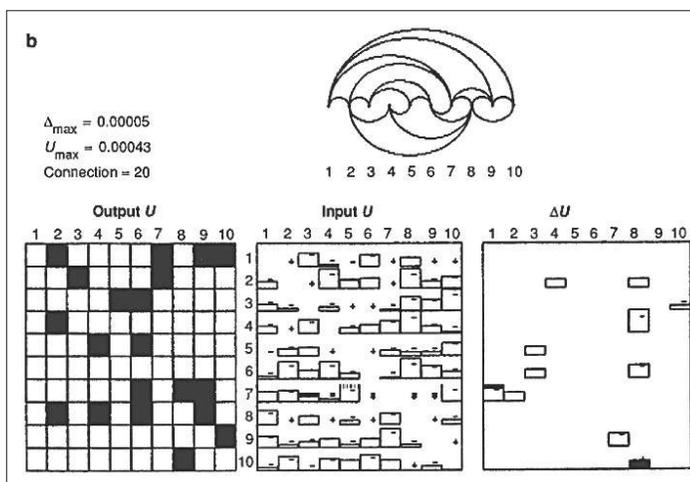
で最初のニューラルネットワークのアプリケーションとしてScienceに、1989年9月15日に出版されました。詳しくは、下記論文を参照してください。

<http://neuro.sfc.keio.ac.jp/publications/pdf/science.pdf>

この当時は、ニューラルネットワークのニューロン数を増やすと、従来のコンピュータでは、時間がかかりすぎます。したがって、当時は、小規模のニューラルネットワークしか、実行できませんでした。現在では、ニューロンの数を気にすることはなくなりました。

次にチャレンジしたのは、MRIなどの画像診断でした。オハイオ州クリーブランド市には、全米でも最大規模の医療機関（クリーブランドクリニック）があります。Scienceの出版効果があり、クリーブランドクリニックから共同研究の

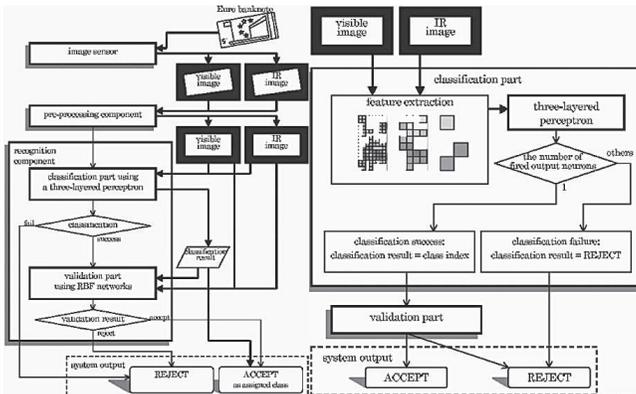
依頼が来ました。共同研究では、MRI画像のクラスタリングに成功し、腫瘍部分を識別できるようになりました。その成果は、IEEE Trans. on Medical Imaging, 11, 2, 215-220, 1992で出版されました。詳しくは、下記サイトを参照してください。
<http://neuro.sfc.keio.ac.jp/>



publications/pdf/sundar.pdf

実際の人工知能の製品を世の中に出したのは、トヨコム（東洋通信機）との共同研究で、紙幣鑑別機（BV-6000）です。丁度、EURO紙幣が流通する2002年1月1日以前に、私の研究室には、流通前のEURO紙幣がありました。ニューラルネットワークを使って、偽札と本物紙幣の識別、紙幣の分類が主な目的です。2012年の時点で世界10カ国12000台が導入されました。論文は翌年の2003年に発表しました。

<http://neuro.sfc.keio.ac.jp/publications/pdf/banknote.pdf>



BV-6000：紙幣鑑別機

何年前か忘れてましたが、EUROの新札でも紙幣鑑別機がエラーを出すというクレームが私のところに来ました。良く聞いてみると、当初、EURO紙幣の印刷は品質管理されていましたが、何年前から、EUROに所属する国ごとに印刷されるようになったそう

です。国によって、品質管理が一定でなく、ある国のEURO紙幣は偽札として判別されていたようでした。その偽札に判別された紙幣を含めて再学習するように企業に指示し、無事、問題は解決しました。

これからどのようなAI社会になるのか、AIを専門とする私でも想像を超えています。現役の教授の間に、超並列計算できるGPUマシン（スーパーコンピュータ）が、個人で所有できるとは思っていませんでした。世界では、AIがチェス世界チャンピオンを打ち負かし、将棋のプロ棋士に勝利をおさめ、最近では、世界囲碁チャンピオンに勝

利しています。

私の研究室では、卒業生の山田君が活躍し、MIT、Stanford、CMUなどの名立たる大学を撃破し、一流企業のGoogle、IBMや

有名な米国AIベンチャーを倒して、AI最大級のコンペ（NIPS 2017）のクイズボールで優勝しました。クイズボールとは早押し物知りクイズです。歴代の米国クイズ王6人にも圧勝しました。最終結果は、6人のクイズ王の140ポイントに対して、山田君のAIシス

テムが365ポイントです。多くの学会参加者は、AIシステムに対して6人のクイズ王が勝つと思っていたらしく、勝負が決まった瞬間、会場はシーンとなったようです。クイズボールの成果は、次の論文に詳しく記述してあります。

<http://neuro.sfc.keio.ac.jp/publications/pdf/quiz.pdf>



6人のクイズ王vs AIの最終戦 (AIの圧勝)

将棋や囲碁などのゲームでは、強化学習 (reinforcement learning) が台頭し、人間が思いつかない新しい手筋を発見しています。強化学習システム同士で戦わせ、切磋琢磨できるので、人間の何百年・何千年分の学習が短期間で実現できます。ゲームでは、AIシステムは神レベルに近づいています。神レベルが何かは、分かりませんが…

2017年のAI将棋大会で優勝したのが強化学習を使ったelmoです。短いコード (2500行程度) で書かれたUSIプロトコル準拠の思考エンジンで、すべてオープンソースで公

開されています。実用的な、強化学習を勉強するのであれば、このソースコードが短くて一番良いでしょう。

<https://github.com/mk-takizawa/elmo>

chess、囲碁、将棋などのゲームでの強化学習の論文は、次の論文が参考になるかもしれません。

<https://arxiv.org/pdf/1712.01815.pdf>

強化学習では、2つの重要なモジュール (agentとenvironment) があります。environmentは、状態 (state) と報奨 (reward) をagentに伝達します。agentは状態と報奨から次のactionを決定し、そのactionをenvironmentに伝えます。これらの伝達 (state, reward, action) がagentとenvironmentの間で続いていき、強化学習が進行していきます。

このようなAI時代の状況で、“企業は何をすればよいか？”という問いに対して、私の答えは、次のようになります。まず、現在の企業が行っている多く仕事は、AIに置き換わっていきます。単純な事務仕事だけでなく、現在の総合職が行っている仕事も含まれます。このようなAI時代を踏まえて、企業はしっかり準備しておく必要があります。

1. 例えば、営業職です。営業の仕事は、自分たちのサービスや製品を、客に届ける仕事です。何処に客がいるのか、また、どのようなサービスや製品をどの客に

- 届ければ成約し易いのか？ そのようなマッチング問題は、AIの得意とするところであり、データからその答えを発見することが可能です。また、新たな客や新たな市場を構築するにあたり、データさえあれば、AIはデータから新たな客や市場を開拓できます。どのようなデータを用意すれば良いのか、また、どのようなデータを採取すれば良いのか、データの準備が一番重要になります。
2. 開発・研究分野の仕事でも、AIに置き換わる可能性があります。強化学習の特徴は、専門家でも思いつかない新しい手筋を発見できる可能性があります。つまり、強化学習のagentとenvironmentさえ用意できれば、AIは優秀な研究者にもなります。
 3. 経営者ですら、AIに置き換わる可能性があります。AIは、データからAIシステム自らを教育し、経営者として成長できる可能性があります。現実の経営者は、自分で経験するしかありませんが、AIの場合、複数の学習経験を統合できる可能性があります。経営者の一番重要な仕事は、データからの迅速な判断です。
 4. 芸術家やアーティストでも、AIに置き換わる可能性があります。AIは、様々なアーティストの特徴を取り出して、新たなアーティストを合成できます。
 5. 人事の仕事も、AIに置き換わる可能性があります。どのような人材が企業に向くのか、採用した人材を生かすにはどのような人事をしたら良いのか？ AIは、データを用いて人事が抱える様々な課題を解決できる可能性があります。
 6. 人間の得意なコミュニケーション力を磨いている人ですら、AIに置き換わる可能性があります。つまり、微妙な表情、目線、その他、ありとあらゆる身体のセンシング情報を得ることで、AIはその人の状態・感情などを知ることが出来る可能性があります。
- このようなAI時代では、遅かれ早かれ、社会の多くの仕事が失われるので、子供へのキャリア・アドバイスは極めて難しいこととなります。また、AI時代には、子どもにどのような教育をしたらよいのか？ AIが人間に代わって多くの仕事を行うので、その反面、人間の余暇時間が増えます。その膨大な余暇時間をどう過ごしていくかも、重要な課題です。人間は、何のために生きているのか、どのように生きればよいのか、根本のところを問われているのが、AI時代に生きる人間なのです。
- 1) 米国では、すべての大学教員はtenureを目指します。tenureとは終身雇用のことです。
筆者は、ケースウエスタンリザーブ大学（ノー

ベル賞15個)の電気工学応用物理学科から36歳のときにtenureを獲得しました。tenure獲得の条件は、研究(論文)・教育(学生の評価)・研究資金獲得の3つが重要な要素になります。米国では、大学での定年は差別であると言う裁判があちこちの州で起こり、現在、多くの大学で定年制が廃止されています。つまり、終身雇用教授は自分で辞めたり、死なない限り、優秀な卒業生が大学に残れない状況が現在でも続いています。その反面、それらの優秀人材の流動によって、シリコンバレーでのベンチャー企業の活躍があります。オバマ前大統領は、米国大学

でのこのような停滞状況を改善しようと試みましたが、任期中にその目的は果たせず、そのチャレンジは失敗に終わりました。

- 2) A*アルゴリズムを直感的に理解するために、次のサイトのgif動画を見てください。

https://en.wikipedia.org/wiki/A*_search_algorithm

- 3) Dijkstraアルゴリズムは、次のサイトのgif動画を確認してください。

https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

プロフィール.....

たけふじ・よしやす 1955年生まれ。慶應義塾大学環境情報学部教授。専門分野は、ニューラルコンピューティング、電子おもちゃ、セキュリティ、温度差発電、横波スピーカー。1983年慶應義塾大学大学院博士課程電気工修了。工学博士。慶應義塾大学にて博士号取得後、米国サウスフロリダ大学コンピュータサイエンス学科客員助教授、サウスキャロライナ大学コンピュータサイエンス学科助教授として勤務。1991年、ケースウェスタンリザーブ大学にて、36歳で終身雇用(tenured)契約を取得。慶應義塾大学環境情報学部助教授を経て1997年より現職。主な著書に「体験する!! オープンソースハードウェア: NanoPi NEO, Arduino他で楽しむIoT設計」(近代科学社、2017年)、「超実践 アンサンブル機械学習」(近代科学社、2016年)、「武藤博士の発明の極意: いかにしてアイデアを形にするか」(近代科学社、2013年)他多数。