Interface 増刊
ボード・コンピュータ・シリーズ

人工知能

ラズベリー・パイからはじめる身の回りAI実験

人工知能を作る

小池 誠，鎌田 智也 他 著

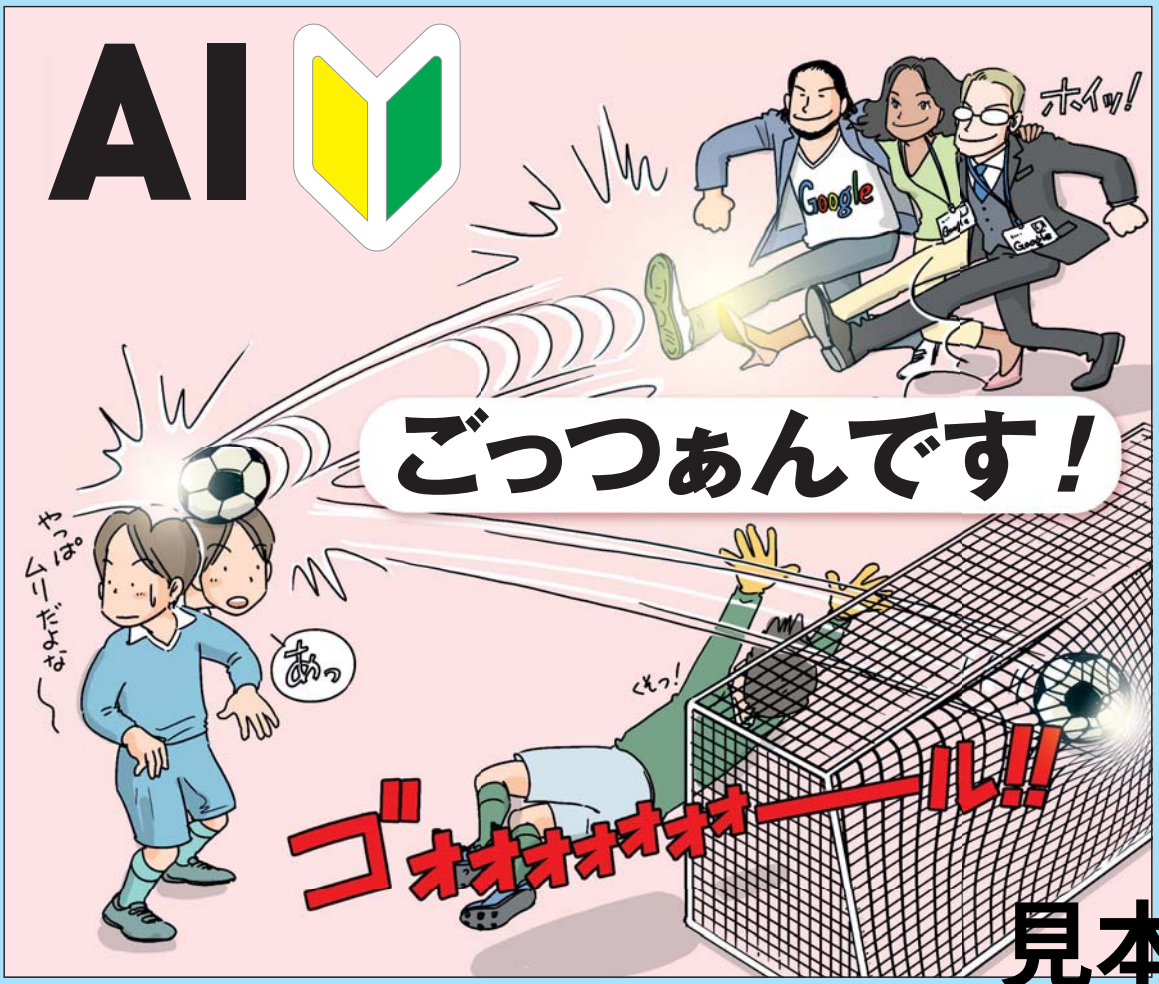
見本

ご購入はこちら．
http://shop.cqpub.co.jp/hanbai/books/MIF/MIFZ201804.htm

CQ出版社

第1部

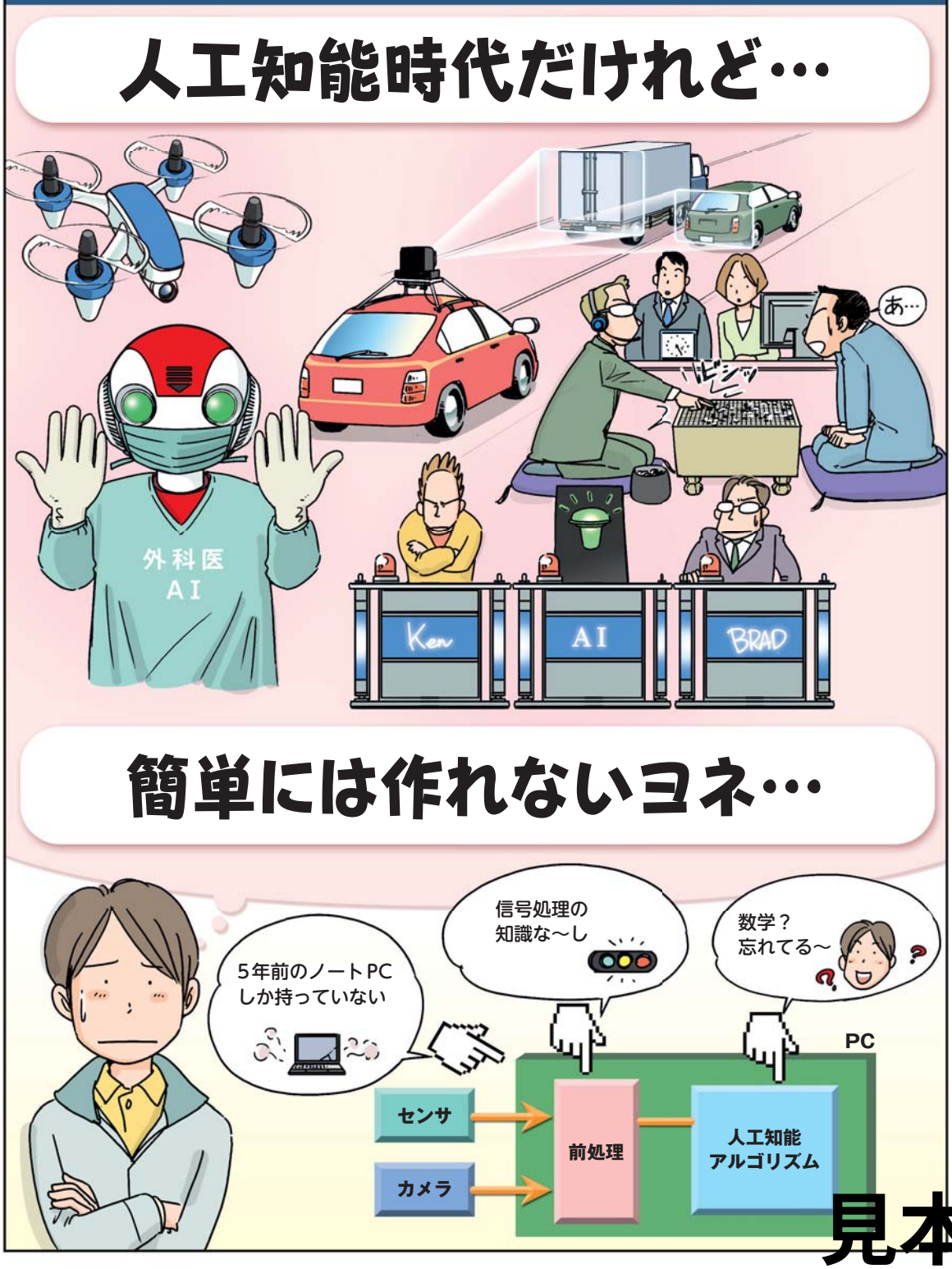
ディープ・ラーニングでラズパイ人工知能を作る

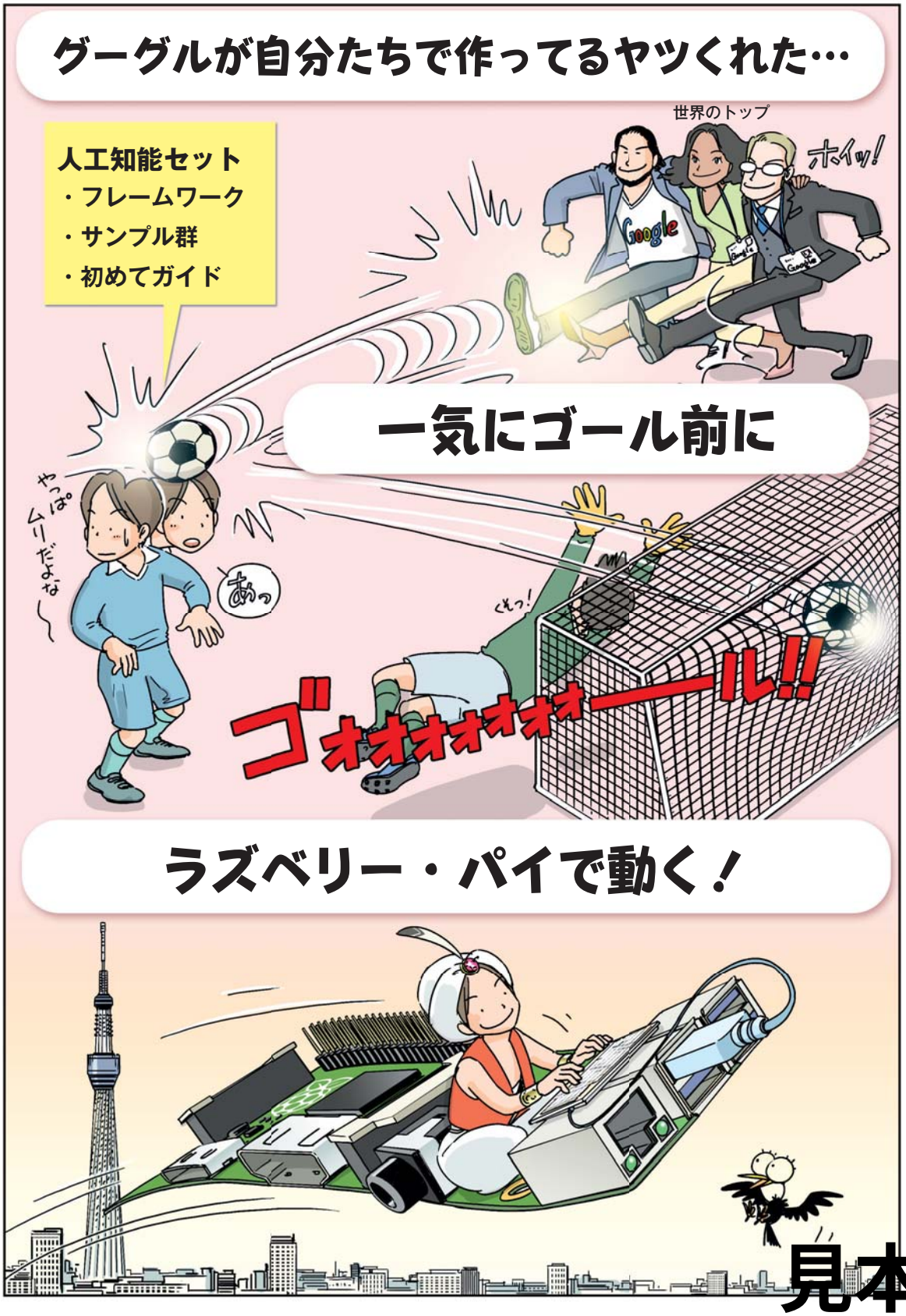


本書で紹介する手順等は執筆時点のものです．ソフトウェアのバージョンUP等によって手順が変わる可能性がありますので，適宜読みかえてください

グーグルが大サービス！ 手のひら人工知能が自宅で作れる時代

編集部




グーグルが自分たちで作ってるヤツくれた…
人工知能セット
・フレームワーク
・サンプル群
・初めてガイド
世界のトップ
ホイッ!
Google
一気にゴール前に
やっぱムリだよな〜
あっ
くそっ!
ゴオオオオオオール!!
ラズベリー・パイで動く!

第1章 ビギナから使える 世界トップのAIライブラリON!

ラズパイからOK! Google人工知能で広がる世界

足立 悠，小池 誠，佐藤 聖

GoogleはAIのトップランナー

人工知能（AI）に関するニュースを新聞やテレビで目にする機会が増えています．皆さんも既にその恩恵に授かっているのですが，そのAI技術を支えている企業の1つがGoogle社です．

● そもそも今のAIブームの発端はGoogleの画像認識

Google社が猫を認識するAIを開発した，というニュースが2012年に発表されました．人間が機械に「これ（対象）は猫だ」と正解を教えることなく，機械が自分で学習し対象が猫だと理解したという内容は衝撃的で業界を賑わせました．ニュース・ソースはGoogle社のブログです(1)．

このニュースを発端に，AI技術が社会的に認知され，一大旋風を巻き起こしました注1．

● GoogleのトップAI技術①…囲碁ソフト

2015年10月，Google社が開発した囲碁ソフト「AlphaGo」注2が人間のプロ囲碁棋士に初めて勝利したニュースが話題となりました．従来の囲碁ソフトは，囲碁のルールと棋譜（対局者が行った手をの記録）をもとに，次の打つ手を考えていました．それに対してAlphaGoは，実際に対局し，勝った時と負けた時それぞれの結果と結果に対する状況（攻撃の手や守りの手など）をディープ・ラーニングを使って学習しています．

AlphaGoの登場を皮切りに，各社でディープ・ラーニングを搭載した囲碁ソフトの開発が進められています．

● GoogleのトップAI技術②…多国語翻訳サービス

2016年9月，Google社は従来の翻訳サービスGoogle翻訳を，ディープ・ラーニングを搭載した「Google Neural Machine Translation（GNMT）」に切り替えたことを発表しました．

ある新聞では，TOEIC 700点取得者ほどの能力を持っていると記載されていました．対象言語は英語，中国語，日本語，フランス語，ドイツ語，スペイン語，ポルトガル語，韓国語，トルコ語です．ディープ・ラーニングの搭載によって，文脈を考慮し文章全体として意味の通る翻訳が可能になりました．

その後，Microsoft社もディープラーニングを搭載した「Microsoft Translator」を発表するなど，各種翻訳サービスへのディープ・ラーニングの実装が進められています．

● GoogleのトップAI技術③…検索エンジン

Google検索エンジンで何らかのキーワードについて検索すると，結果をランク付けして表示します．ウェブ・ページの質を重要度とし，高い順にランク付けしています．質とは例えば，文章の流れとして筋が通っている，文章がオリジナルである，文法が間違っていないなど挙げられます．

重要度の算出には従来，PageRankと呼ばれるアルゴリズムを適用していましたが，2015年にディープ・ラーニングを搭載した「RankBrain」と呼ばれるアルゴリズムを導入しました．検索エンジンにディープ・ラーニングを搭載したことにより，ウェブ・ページの質を判断する基準を機械で自動的に作成できるようになりました．

Google社はこの検索エンジンを他社にも提供しています．

● GoogleのトップAI技術④…自動運転

人間は車を運転する際，交通状況を判断してアクセルを踏む，ブレーキをかけるなどの動作を行います．Google社を始め各自動車関連のメーカは，人間のこの動作を機械で制御し自動化する「自動運転技術」の開発を進めています．自動運転技術では，GPSによる位置データ，車体に搭載された各種センサ・データ，画像データなどのさまざまな情報を機械にディープラーニングで学習させています．その結果，車線変

注1：DeepMindというライブラリを利用している．
注2：https://deepmind.com/research/alphago/

や追い越し，歩行者を避けるなどを自動で判断し実行します．

Google社は本格的な自動運転技術開発を始めた最初の企業です．今後は子会社のWaymo（ウェイモ）注3を中心に，FCA（フィアット・クライスラー・オートモービルズ）社，ホンダ社など各自動車メーカと共同で自動運転の実用化に向け技術開発を進めていくようです．

そんなGoogleが提供するオープンソースAIライブラリTensorFlow

● 商用/非商用を問わず使用できる

Googleは，TensorFlowという人工知能向けライブラリ（フレームワーク）を提供しています．オープンソースのライブラリであり，商用/非商用を問わず使用できます（Apache2.0ライセンス記載）注4．

● 開発言語はPythonとC++

TensorFlowのコア部分はC++で実装されていますので，ユーザはC++を使って学習とモデル作成を行うことができます．

Pythonの場合は多くのアルゴリズムが既に関数として用意されているため，複雑な演算を意識することなく簡易に実装できます注5．Pythonはスクリプト言語であり，これからプログラミングを始める方向けの言語です．Pythonは初めてという方も，他の言語に慣れていれば実装は可能です．

動作環境

● ほとんどのコンピュータで動く

TensorFlowはLinuxとOSXをサポートしており，`pip install`で入手できます．2016年11月29日にバージョンのr0.12がリリースされ，64ビット版Windowsへも直接インストールできるようになりました．

32ビット版WindowsユーザはLinux仮想環境を構築するDockerを利用するなどの手段で，TensorFlowを使って開発できます．環境の構築については後の章で紹介します．

● ラズベリー・パイなどのボードでも動く

TensorFlowにはラズベリー・パイ用のインストーラも用意されています．第2章以降のキュウリ仕分け機の事例では，ラズベリー・パイ3にTensorFlowを搭載し，キュウリのサイズや曲がり具合で階級を判断する学習済みモデルを動かしていました．

個人で試せる

ハードウェア（組み込み機器）とTensorFlowを組み合わせることで何ができるのか，いくつかの事例と今後の可能性について紹介します．

● 判定に客観性を持たせる…キュウリの自動仕分け

キュウリの仕分け作業では基本的に，キュウリ1本1本の特徴を目視で確認し，人手でランク別に分類しているようです．しかし人手での作業は非常に時間がかかるため，TensorFlowを使ってキュウリを等級/階級別に自動分類しています．

ここではデータの学習にPC，データの判定にラズベリー・パイ3，仕分けマシンの制御にArudino Microを使っています．

学習のための正解データ（ある特徴を持つキュウリの等級/階級はXXである）として，人手で仕分けしたキュウリの画像を使っています．

学習モデルの構築は，TensorFlow公式サイトのチュートリアル「Deep MNIST for Experts」がベースとなっています注6．このチュートリアルでは，手書き数字（0～9）の画像データセットMNIST（Mixed National Institute of Standards and Technology）database注7を使って，畳み込みニューラル・ネットワーク・アルゴリズムによるモデル学習を行います．

あだち・はるか

● 自動で分別してくれるゴミ箱!

TensorFlowとラズベリー・パイを利用して身近な問題をスマートに解決した例として，自動分別ゴミ箱があります．

サンフランシスコで開催されたハッカソンで作られたこのゴミ箱は，ラズベリー・パイのカメラを使って，捨てられたゴミがリサイクル可能なものかどうかを判断して，自動で分別してくれるというものです(2)．

広がる世界!
こんな装置が作れるかも

筆者はカメラ画像からキュウリの等級/階級を判断

注3：https://waymo.com/

注4：もともとGoogle製品の機能は「DistBelief」と呼ばれるGoogle社の社内基盤を使って開発されていました．実用性は認められ検証されているものの，DistBeliefはGoogle社内の環境に依存したものだったため，外部に公開することが不可能でした．そこで社外に公開でき，そしてDistBeliefよりも処理速度が速く（2倍と言われる），拡張性に富んだTensorFlowが開発されました．

注5：Pythonの APIは数千用意されています．https://www.tensorflow.org/versions/r0.11/api_docs/index.html．APIの詳細はAppendix2で解説します．

注6：https://www.tensorflow.org/versions/r0.11/tutorials/mnist/pros/index.html

注7：http://yann.lecun.com/exdb/mnist/

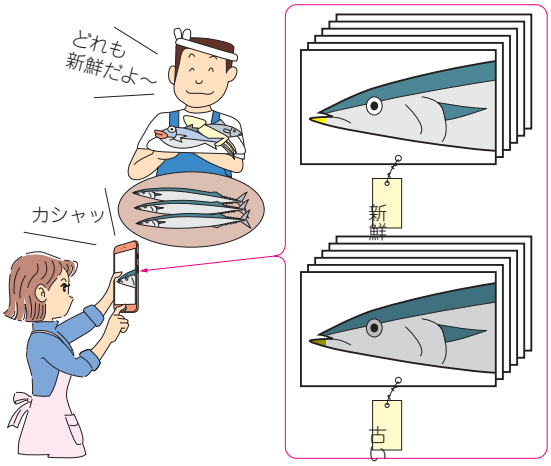


図1　魚屋さんのノウハウを吸収! さんまの鮮度判定器が作れる

するシステムを作りました（第2章以降で紹介）．同じ手法でいろいろなアプリケーションが作れます．

● さんまの鮮度判定器

スーパーでさんまを買うときに，少しでも鮮度のよいものを選びたいですよね．さんまの鮮度は下あごの色や目の色などから判断可能だと言われています．

そこで，さんまの写真をたくさんとってTensorFlowで学習することで，さんまの鮮度判定器が作れるかもしれません．スマホアプリとして実装すれば，賢い買い物ができるかもしれません（**図1**）．

● 観葉植物の自動管理装置

観葉植物の水やりを自動化できるかもしれません．TensorFlowで植物のしおれ具合を判断し，必要な分量の水を，ラズベリー・パイを使ってかん水することもできるでしょう（**図2**）．さらに，葉焼けやカビ，低温障害なども自動で判断して，アラームを鳴らすなんてこともできそうです．

● 酔っぱらい判断装置

顔画像を使って酔っぱらい判断装置が作れそうです．

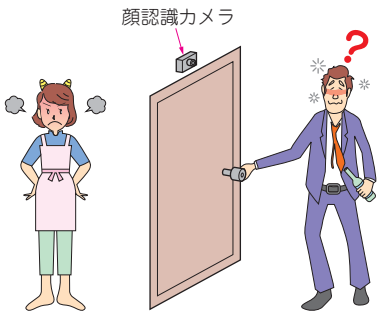


図3　度を越して酔っているときはドアを開けない装置が作れる

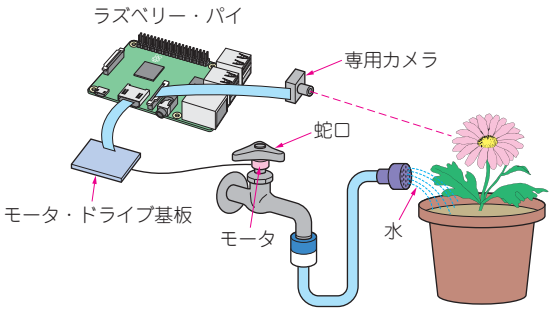


図2　植物のしおれ具合を判断し給水してくれる装置が作れる

す．玄関にカメラを設置すれば，旦那が酔っぱらって帰ってきたときは，玄関のカギを開けないなんてこともできるかもしれません（**図3**）．アイディア次第でまだまだいろいろな装置が作れそうですね．

こいけ・まこと

● 顔認識ドアホン

ラズベリー・パイにTensorFlowライブラリを組み込めば，どんな工作ができるかアイディアを出してみましょう．例えば，状況に応じて対処を選択できる顔認識ドアホンが作れるかもしれません（**図4**）．

- 知り合いがドアの前に立つとドアホンのボタンを押す前に玄関先に○○さんが来たとスマホに通知
- 知らない人なら呼び鈴が鳴らずに留守録で対応
- 家族ならドアロックを自動解錠

実現方法としては，あらかじめ家族の写真やドアホンに写る画像データからPCで教師データを作成します．ドアホンのカメラ映像をラズベリー・パイで受信して，顔の画像と教師データとを比較して，ドアの前に立つ人が誰かを顔認識します．

知らない人の画像を近所で共有できるようにしたら，防犯に役立ちますね．

ラズベリー・パイのような小型ボードで実現できれば，夜間に自動車が所有者を見つけて，所有者を自動車まで誘導するのにライトで足下を照らしたり，所有

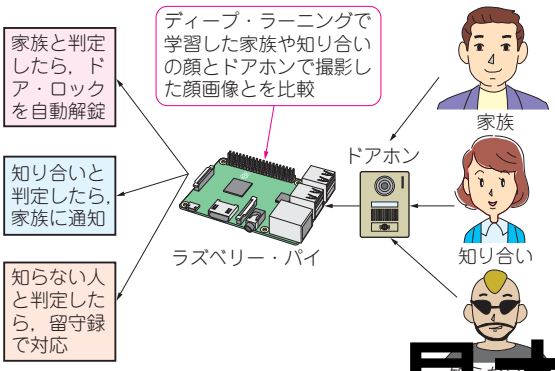


図4　顔認識ドアホンがあれば不審者を寄せ付けない

◆参考文献◆
(1) Using large-scale brain simulations for machine learning and A.I., 2012, Google.
https://googleblog.blogspot.jp/2012/06/using-large-scale-brain-simulations-for.html

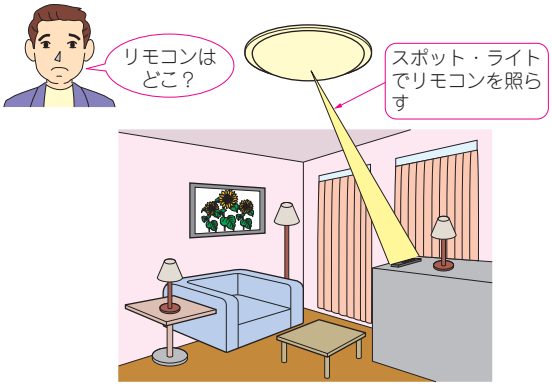


図5　なくした物を探してくれるシーリング・ライトのイメージ

者以外が自動車に近づいたときには所有者のスマホに通知したりするなどの使い方ができそうです．

● なくした物を探してくれるシーリング・ライト

部屋でなくした物を探せるシーリング・ライトが作れると思います．**図5**のようにシーリング・ライトに音声認識と画像認識を組み込めば，「鍵はどこ？」などと探している物の名前を話すと，カメラで部屋の中から対象物を認識して，対象物の位置を赤のスポット・ライトで照らしてくれる工作ができると思います．

実現方法は，シーリング・ライトにカメラ，マイク，スポットライトを動かすサーボモータをつなげたラズベリー・パイを内蔵します．あらかじめ**図6**のように部屋の画像からテーブル，椅子，テレビ，ソファー，リモコン，植木鉢，床など部屋にある物の画像からPCで教師データを作成します．部屋の中をカメラ撮影して，教師データと照らし合わせて室内の物を位置を特定します．

別の部屋にあるシーリング・ライトと連携すれば，家中の物の位置情報を特定して捜し物を見つけ出すことができ，新たに加わった物を探したりするのにも応用できます．

もし，通常定位置にある物が地震により動いたら転倒を予測して通知することもできるでしょう．

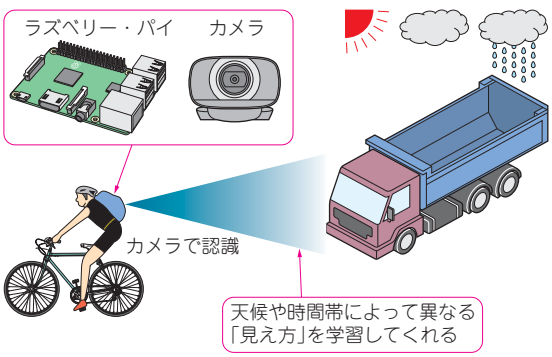


図7　自転車で後方の危険を察知するイメージ

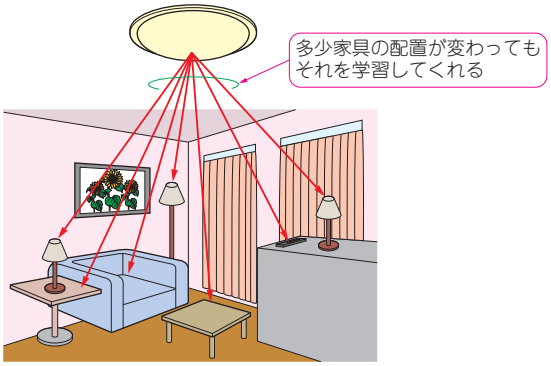


図6　あらかじめ部屋の画像から教師データを作成しておく

● 歩行者や自転車の背後の危険を察知

ラズベリー・パイにカメラを接続して，背後から車両や自転車の接近を教えてくれる危険予知の工作もできると思います（**図7**）．接近を検知したら，バイブレータやLEDフラッシュなどで注意を促します．

実現方法はリュックサックにカメラを取り付けたラズベリー・パイをセットして，後方より接近する車両や自転車を画像認識させます．日中/夕方/夜間などの時間差，晴れ/曇り/雨などの天気の違いで車両や自転車の見え方がかなり異なります．時間帯や天気とそのときの画像を教師データとして学習すれば，見え方の違いを克服できるようになります．

● 良い質問をしてくれる装置

良いアイディアがあっても忘れてしまうことがあります．そうした場合に，良い質問をしてくれる人がいると思い出しやすくなります．ディープ・ラーニングで日常の会話やメールの内容，ウェブ閲覧履歴を学習させ，対話的に話題に即した質問をしてくれると役立つかもしれません．思い出した情報をディープ・ラーニングの新たな学習データとして再学習に利用すれば，徐々に質問がうまくなるかもしれません．

実現方法は，音声認識で会話をテキスト・データに変換して，ディープ・ラーニングで自然言語処理して，会話に関連した質問を生成します．質問内容と回答を記録して体系化すれば，よくある質問だけでなく，意外な質問をしてくれるかもしれません．また，質疑応答の内容はアイディアのネタ帳として役立つかもしれません．

上記の応用として，アイディアの共起分析をすれば，アイディアを可視化でき，今まで試したことがないアイディアや，その組み合わせを発見できるかもしれません．

さとう・せい

(2) Auto-Trash sorts garbage automatically at the TechCrunch Disrupt Hackathon.
https://techcrunch.com/2016/09/13/auto-trash-sorts-garbage-automatically-at-the-techcrunch-disrupt-hackathon/

Appendix 1 オープンソースで初心者も独学OK!

Googleの人工知能ライブラリTensorFlowを勧める理由

佐藤 聖

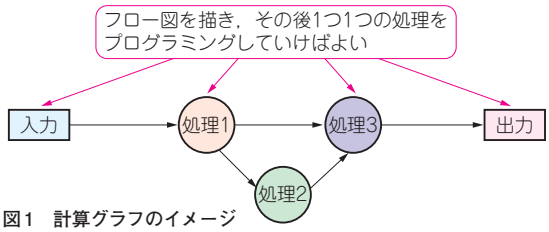


図1　計算グラフのイメージ

ディープ・ラーニングを理論からマスタし，自分でプログラムを書けるようになるには何年もかかります．最近では，いろいろな団体や企業からディープ・ラーニングを試すためのライブラリ/フレームワークが提供されています．各社特徴がありますが，ここではTensorFlowを利用することにしました．理由は下記の通りです．

● 理由1…トップランナーGoogleが作って実際に使っているくらいの品質

TensorFlowはGoogle社の機械学習研究機関のGoogle Brainチームに参加していた研究者やエンジニアによって開発されました．Googleが提供している画像認識や音声認識，翻訳などにTensorFlowが使われています．

● 理由2…初心者も独学で始められる

TensorFlowの公式サイト（https://www.tensorflow.org/）にはチュートリアルがあります．試しにチュートリアルに沿ってMNISTやCIFAR-10などを実行してみるとディープ・ラーニング等がどのように機能するのか体験できます．

紹介されているプログラムのアルゴリズムを理解しようとすると，調べないと分からない用語も出てきます．そのことがディープ・ラーニングや機械学習のとっかかりになります．

● 理由3…オープンソースで使いやすい

CaffeのBSD-2ライセンスやChainerのMITライセンスはとても緩いですが，TensorFlowはApache 2.0ライセンスであり，それ以上に緩いです．作者から利用者への特許権の許可を明確にしています．

詳細：http://d.hatena.ne.jp/nishiohirokazu/20140221/1392962370

図解：http://www.catch.jp/oss-license/2014/02/22/apache-license2-0/

● 理由4…処理の流れが視覚的にわかるように図示できる

TensorFlowのプログラムは，計算をグラフとして構築する部分と，グラフを実行する部分に分かれます．

計算をグラフで構築する部分は**図1**のようなデータ・フロー図をイメージすると理解しやすいです．プログラミング経験があれば，計算をフローとして「関数化」しているとイメージすると分かりやすいと思います．

データは入力から処理1に流れて計算を処理され，処理1の結果は処理2に流れて計算を処理し，処理1と処理2の結果を処理3に流して計算を処理します．そして処理3の結果を出力に流すフローを作ります．このようにフロー構造（グラフ）を最初に定義すればよく，データ・フロー図を描いてからコードを書いていくと理解しやすいと思います．

● 理由5…性能UPしたいときはすぐにGPUで動かせる

TensorFlowにはCPU版とGPU版があります．CPU版で開発したコードでも，GPU版をインストールすれば，ほぼそのまま動きます．公式サイトの「HOW TO」でサブタイトル「Using GPUs」（https://www.tensorflow.org/how_tos/using_gpu/）に詳しい説明があります．基本的にはGPUが搭載されていればCPUよりもGPUが優先されます．明示的にCPUだけで処理するには「tf.device('/cpu:0')」，GPUで処理するには「tf.device('/gpu:0')」を宣言すればよいだけです．「cpu:」や「gpu:」の後ろに付く数字はCPUやGPUのID番号です．CPUやGPUが1つならいずれも「0」です．CPUやGPUが複数搭載されていても明示的にID番号を指定しないとデフォルトでID番号「0」で処理されます．複数タスクを複数CPUやGPUに分散して実行するには，タスクにCPUやGPUを明示的に割り当てる必要があります．

このように簡単にGPUが利用できるのでグループでプログラム開発するときにハードウェアの違いをほとんど意識せずに済みます．

さとう・せい

ラズパイ×Google人工知能…キュウリ自動選別コンピュータ

小池 誠

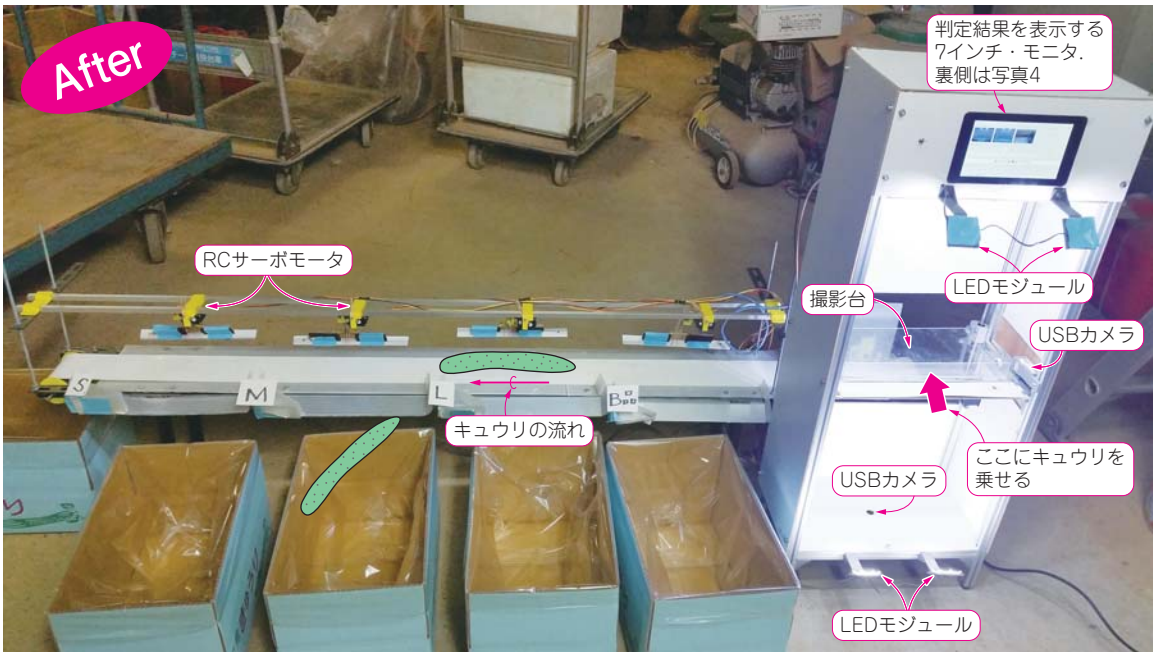


写真2　今回ラズパイ×Google人工知能ライブラリで実現したキュウリ自動選別コンピュータ

写真1
現在の仕分け作業
…手作業で8時間
もかかってます

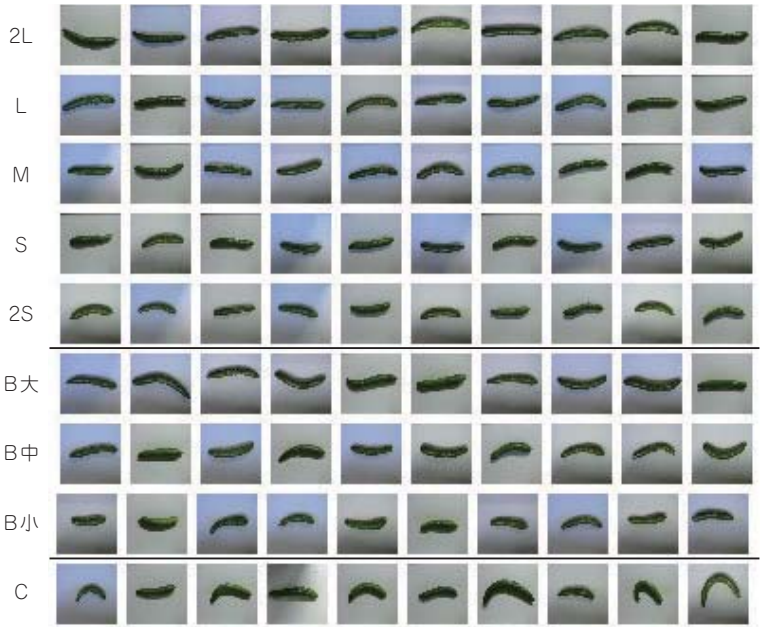


写真3　ぜひとも人工知能コンピュータを投入したい…農家のキュウリの仕分け

表1　キュウリの仕分け基準の例

等　級	秀　品					B品			C品
階　級	2L	L	M	S	2S	大	中	小	−

（a）等級・階級表

	秀品	B品	C品
曲がり具合	真っ直ぐ←→曲がっている		
太　さ	均一←→不均一（先細り，先太り）		
色　艶	艶がある←→艶がない		
傷	ない←→ある		

（b）等級の仕分け基準の例

	2L	L	M	S	2S
長さ	約25cm以上	約23～25cm	約21～23cm	約19～21cm	約17～19cm

（c）階級の仕分け基準の例

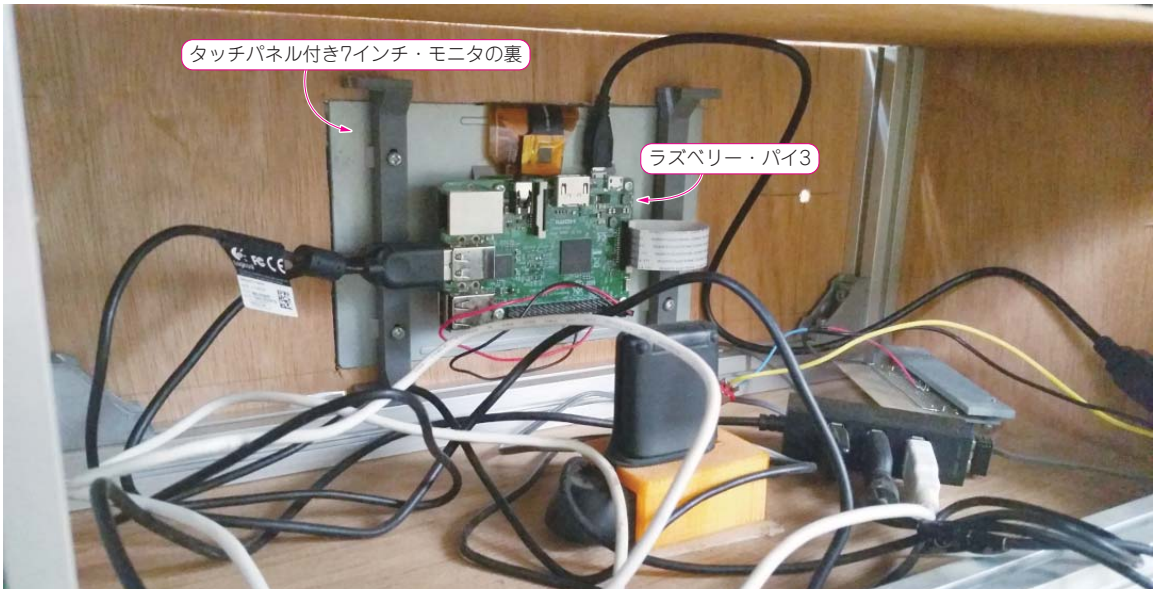


写真4　人工知能によるキュウリの等級/階級判定はラズベリー・パイ3で行う

ここでは，私がラズベリー・パイとGoogleのオープンソース・ライブラリを組み合わせて作ったオリジナル人工知能コンピュータを紹介します．これを使って，現在手作業で行っているキュウリの仕分け（**写真1**）のスマート化（**写真2**）に挑戦しました．

人工知能（ディープ・ラーニング）に注目したきっかけ

● 農家がやっていること

野菜の仕分け（選果）作業はご存知で[illegible]が，一般的に農家が収穫した野菜は，卸売市場[illegible]出荷され[illegible]

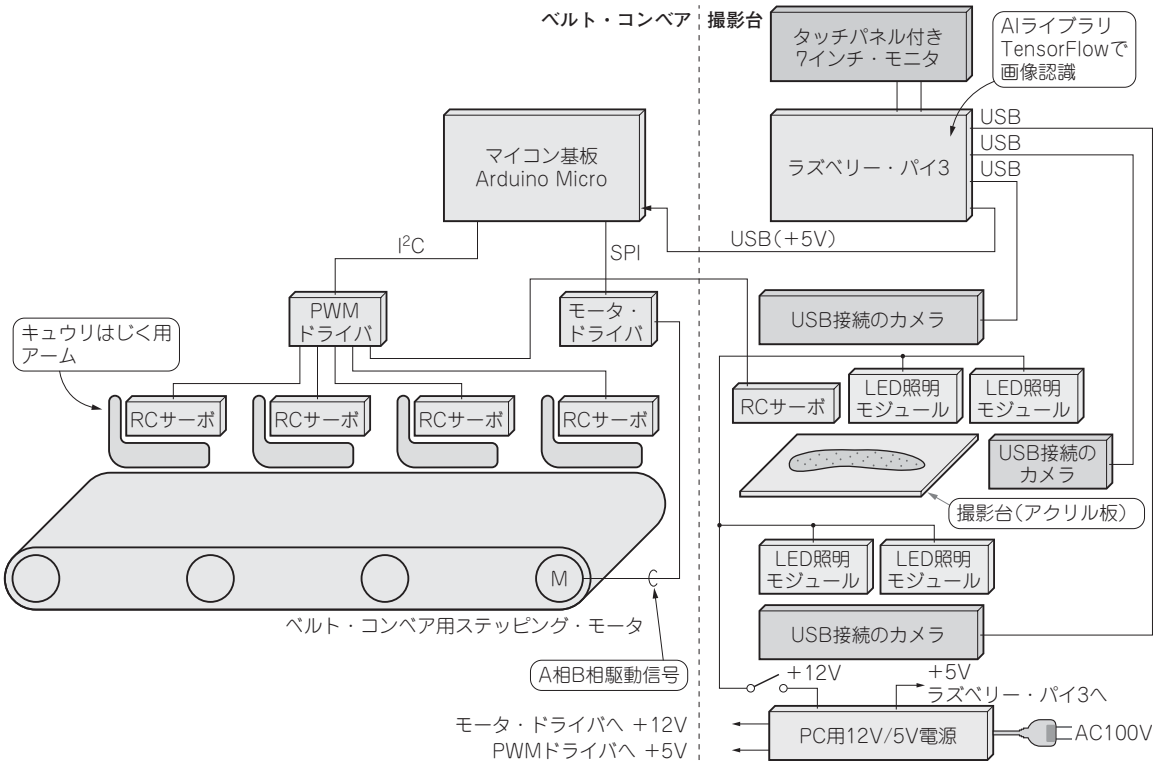


図1　ラズパイ×Google人工知能…キュウリ自動選別コンピュータのハードウェア構成

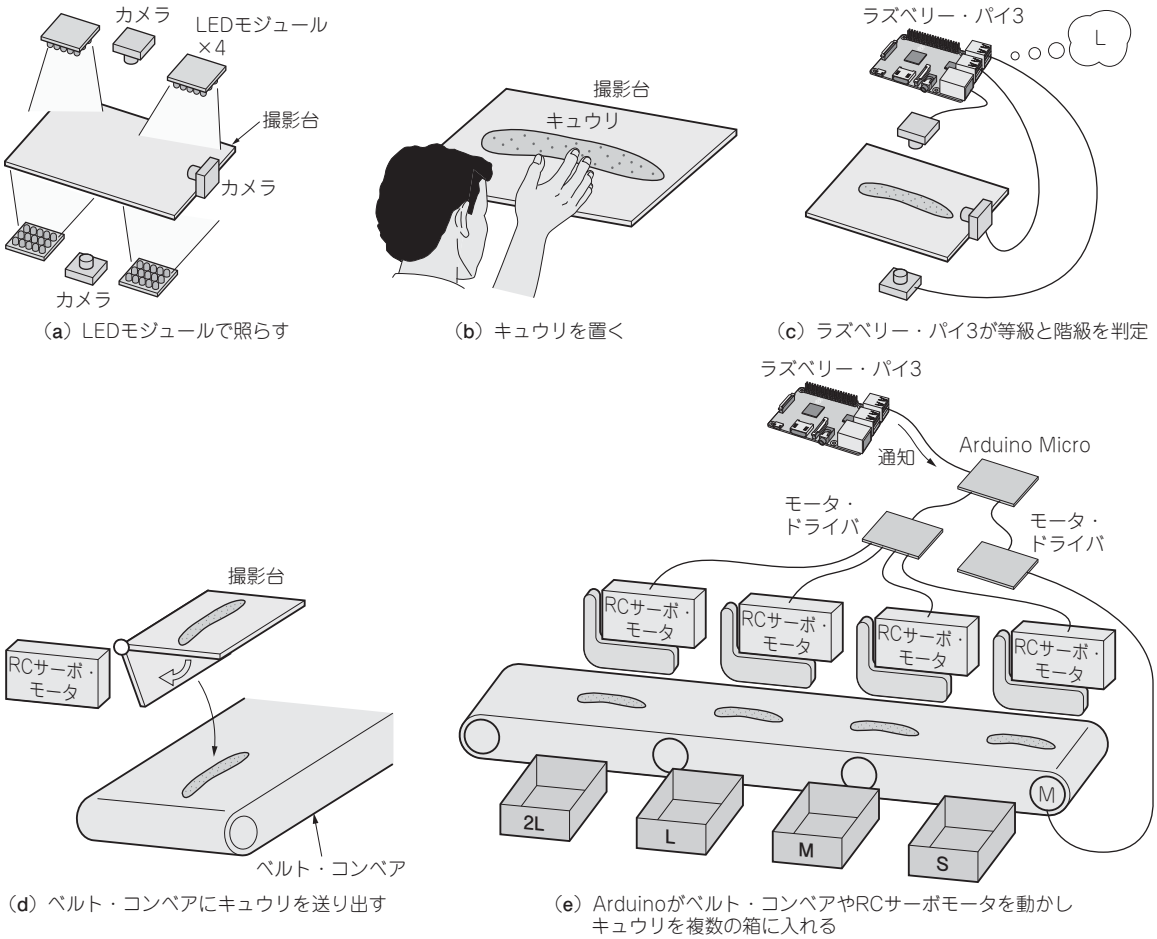
表2　AIキュウリ自動選別コンピュータに使った部品

品　名	型　式	個　数	用　途	入手先
ラズベリー・パイ3 Model B	−	1	サイズ判定	秋月電子通商
7インチ・モニタ　タッチパネル付き	899-7466	1	UI	秋月電子通商
USB接続のウェブ・カメラ(上部，下部)	Logicool C270	2	キュウリ画像の撮影	Amazon
USB接続のウェブ・カメラ(側部)	ELECOM UCAM-C0220FE	1	キュウリ画像の撮影	Amazon
LED照明モジュール	AE-LEDLAMP-7X5	4	照明	秋月電子通商
スイッチ	1MS1-T1-B1-M2-Q-N	1	LED照明のON/OFF	秋月電子通商
USBケーブル(Aオス　microBオス 1.5m)	−	1	ラズパイ-Arduino通信用	秋月電子通商
RCサーボモータ	SG-5010	1	アクリル版の上下駆動部	秋月電子通商

(a) 撮影台

品　名	型　式	個　数	用　途	入手先
Arduino Micro	−	1	ベルト・コンベアの制御	秋月電子通商
PC用ATX電源(MAX330W)	−	1	システム全体の電源	PCショップ
L6470ステッピング・モータ・ドライバ・キット	−	1	ベルト・コンベア駆動用	ストロベリー・リナックス
42mm ステッピング・モータ 2.8V 2相 400ステップ	ST-42BYH1684-1613	1	ベルト・コンベア駆動用	ストロベリー・リナックス
PCA9685搭載16チャネル PWM/サーボ ドライバー (I²C接続)	−(Adafruit製)	1	キュウリ選別アーム駆動	スイッチサイエンス
RCサーボモータ	SG-5010	4	キュウリ選別アーム駆動	秋月電子通商

(b) ベルト・コンベア



（a）LEDモジュールで照らす　（b）キュウリを置く　（c）ラズベリー・パイ3が等級と階級を判定

（d）ベルト・コンベアにキュウリを送り出す　（e）Arduinoがベルト・コンベアやRCサーボモータを動かしキュウリを複数の箱に入れる

図2　キュウリ自動選別コンピュータの動作フロー

前に，病気や傷があるものが除かれたり，大きさ，形，色などにより等級・階級別に仕分けされます．

これは，不良品の市場出荷を防ぐためであり，加えて，品質，サイズを揃えることで買い手が安心して生産物を選ぶことができるようになり，より高い値段で買ってもらえるためです．

私が生産しているキュウリも出荷前には仕分け作業を行っており，収穫したキュウリを9種類の等級および階級に選別しています．**写真3**が実際の等級・階級別に並べたキュウリです（**写真4**が判定を行う装置）．

等級は，曲がり具合，太さ，色艶，傷の有無など，生産物の品質により3種類に分けています．

階級は，長さにより，秀品は5種類，B品は3種類に分けています（**表1**）．

● キュウリ仕分けに手作業で8時間…なんとかしたい!

この仕分け作業ですが，私たちのような小規模農家では，ほとんど人間の手で行っています（**写真1**）．繁忙期には1日8時間もかかる大変な作業なので，なんとか自動化できないかと考えていました．そこで注目したのがディープ・ラーニングです．取り分け画像認識の分野においては，人間の精度を超えたとも言われているディープ・ラーニングを使えば，キュウリの仕分けもできるのではないかと考えたわけです．

ということで，今回はディープ・ラーニングを使ってキュウリの仕分けにチャレンジしてみました．

ハードウェア

写真2が今回作成した自動選別コンピュータです．ハードウェアの構成を**図1**に示します．右上表示パネルの裏にはラズベリー・パイ3が入っており（**写真4**），キュウリの階級/等級を判定します．使用した部品は**表2**の通りです．制作に当たっては，気軽に買えるようなものを目標としているため，できるだけ

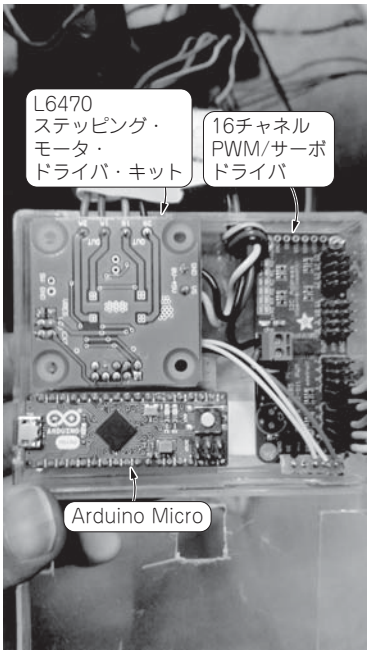


写真5　モータ類の制御にはマイコン基板Arduinoを使った

写真6　その1：ベルト・コンベア用ステッピング・モータ

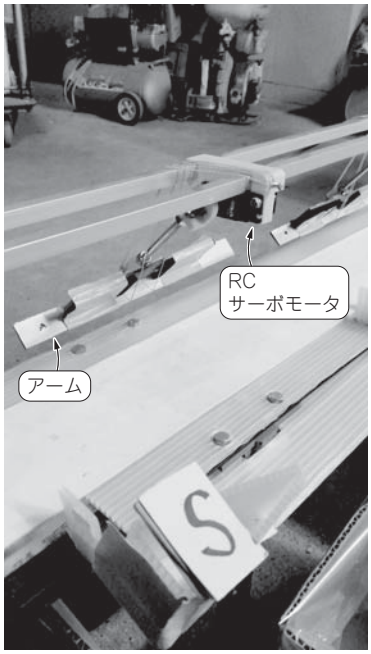


写真7　その2：キュウリを箱に送り出すアーム用RCサーボモータ

ト，安価，部品調達が簡単なものを選びました．

● 動作のフロー

基本動作を**図2**に示します[注1]．

- 撮影台をLEDモジュールで照らす［**図2**(a)］
- 人間が撮影台にキュウリを置く［**図2**(b)］
- USB接続のUSBカメラで撮影（3カ所）
- ラズベリー・パイ3でキュウリ画像を取り込む
- ラズベリー・パイ3上の人工知能で等級と階級を識別［**図2**(c)］
- 識別結果をベルト・コンベア側のマイコン基板Arduino（**写真5**）に通知
- ArduinoがRCサーボモータを動かしキュウリをベルト・コンベアに送り出す（落とす）［**図2**(d)］
- Arduinoがステッピング・モータ（**写真6**）を回しベルト・コンベアを動かす［**図2**(e)］
- そして指定の位置までキュウリを運搬
- Arduinoが指定の位置でアーム（**写真7**）を動かしキュウリを箱に移す［**図2**(e)，**写真8**］

ソフトウェア

自動選別コンピュータとしての動作は，上記の通りです．キュウリの等級/階級判定には，ディープ・ラーニングを利用しています．今回はGoogleが提供しているTensorFlowという機械学習用ライブラリを，Python言語で利用しています．詳細はこの後の章にて順に説明します．

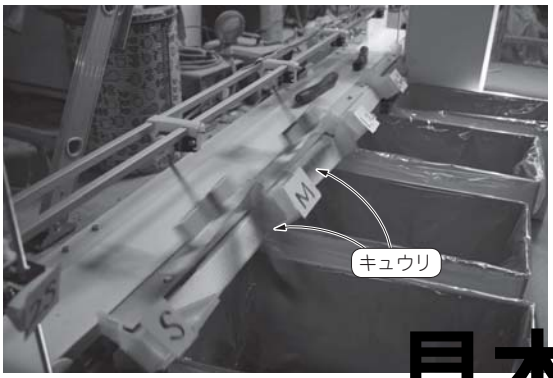


写真8　箱に移されるキュウリ

注1：動画をYouTubeにアップしてあります．https://www.youtube.com/watch?v=Nho2yyCdb3A

表3　キュウリ自動選別コンピュータの制作に使用したコンピュータ

項目＼マシン	デスクトップPC	ノートPC	ラズベリー・パイ3
型名	Magnate EM	ThinkPad X200	–
メーカ名	ドスパラ	Lenovo	ラズベリーパイ財団
CPU	Core i5 3470	Core 2 Duo P8400	BCM2837
クロック	3.2GHz	2.26GHz	1.2GHz
GPU	未使用	未使用	–
RAM［バイト］	24G	1G	1G
HDD［バイト］	1T	160G	–
OS	Ubuntu 14.04	Ubuntu 14.04	Raspbian（2016-5-27）
TensorFlow	0.10.0	0.10.0	0.10.0

● 開発環境/動作環境

キュウリ学習用のデスクトップPCの仕様を**表3**に示します．デスクトップPCのOSはUbuntu 14.04です．このOS上にTensorFlowをインストールしています．

キュウリ識別用のノートPCの仕様を**表3**に示します．ノートPCのOSもUbuntu 14.04です．このOS上にTensorFlowをインストールしています．

特集ではキュウリの識別プログラムをラズベリー・パイ3で動くようにしています．ラズベリー・パイ3上のOSはRaspbianです．このOS上で，TensorFlowのC++ライブラリをビルドして使用しています．

● 学習や判定のアプリケーション・プログラム

TensorFlowが用意してくれているディープ・ラーニングのAPIはPythonとC++です．今回は学習，判定のプログラムをPythonで記述しています．

こいけ・まこと

Appendix 2 AIの定番言語Pythonから無料で使える

Google人工知能ライブラリTensorFlowの正体

佐藤 聖

位置づけ

● Google社の人工知能群の一つ

TensorFlowはGoogleが提供する人工知能ライブラリの1つです．Googleグループでは他にも人工知能ライブラリを持っています．例えば以下があり，**図1**のような関係になります．

- TensorFlowと連携して言語構文を解析するSyntaxNet
- 囲碁を通じての人工知能研究を目的としたAlphaGo（現在ライブラリ非公開）
- 人間並みの制御ツールとしてのDQN

● 用途は研究からビジネスまで

TensorFlowは人工知能を開発するのに便利な機能が多数用意されています．オフィスで一般的に利用される会計処理やBIツール[注1]などにも使えます．アルゴリズム次第で人工知能だけでなく一般的な数学や統計などの計算処理ができるので，汎用性の高い人工知能ライブラリと言えます．

つまり，Googleの人工知能＝TensorFlowではないですし，TensorFlow＝ディープ・ラーニングでもないのです．ディープ・ラーニングは処理負荷が高くなりがちです．もし，処理負荷がより低いサポート・ベクタ・マシンやニューラル・ネットワークで成果が得られるなら，あえてディープ・ラーニングを選択しないケースも考えられます．TensorFlowなら比較的簡単なコーディングでさまざまなアルゴリズムを高速に処理することができるので魅力です．

用意されているAPI

● 数がすごい…Pythonの場合は3780個

TensorFlowライブラリには多種多様なAPIが用意されています．Pythonで利用できるAPIは**表1**のように43カテゴリがあり，公式サイトのガイドには3780のAPIが定義されていました[注2]．

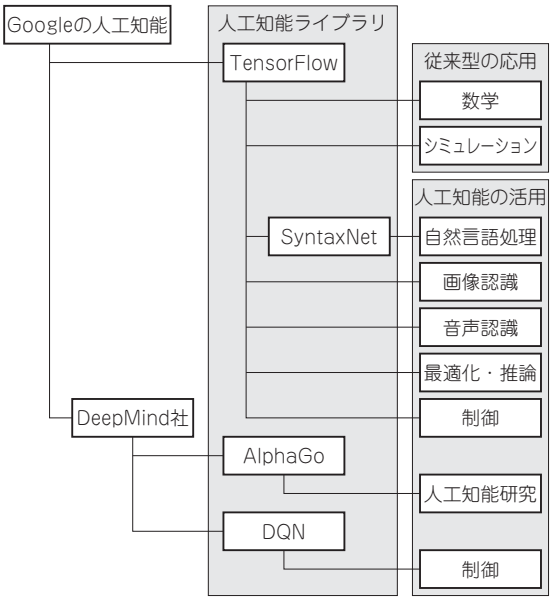


図1　TensorFlowはGoogleの人工知能ライブラリの一つ

● APIの一例

表2は数学関連のAPIの例です．基本的にはそれぞれの引き数に値を渡せばよく，表計算ソフトの関数のような手軽さです．

表3はニューラル・ネットワーク関連のAPIの例です．これ以外にも多数のAPIがあり，組み合わせによってディープ・ラーニングのアルゴリズムも作成できます．APIの詳細な説明は公式サイトのガイドにあり，APIの処理内容，引き数や戻り値の説明，サンプル・コード，エラー発生ケースが掲載されているので参考にできます．

さとう・せい

注1：ビジネス・インテリジェンス・ツール．企業が持つ大量のデータを収集・分析する．

注2：この中で3割近くはGoogleではなく，TensorFlowコミュニティの個人や企業から寄与されたAPIを含んでいます．寄与ガイドライン（https://github.com/tensorflow/tensorflow/blob/master/CONTRIBUTING.md）なるものがあり，APIを寄与すればTensorFlowの機能拡張に貢献することもできます．

表1　Pythonで利用できるAPI

カテゴリ	主な機能
グラフの作成	コア・グラフのデータ構造，テンソル・タイプ，ユーティリティ関数，グラフ・コレクション，新しい操作の定義，TensorFlow構築ライブラリ，その他の関数とクラス（TensorFlowデバイスの仕様を表示，新しいDeviceSpecオブジェクトを作成，DeviceSpecの文字列表現を返すなど）
アサーションとブール・チェック	アサーションとブール・チェック関連関数
定数，シーケンス，ランダム値	定数値テンソル，シーケンス，ランダム・テンソル
変数	変数，ヘルパ関数，保存，復元，共有，シャーディングのための変数パーティショナ，スパース変数の更新，メタグラフのエクスポートとインポート
テンソル変換	キャスティング，図形，形状，分離，接合，偽量子化
数学	算術演算子，基本的な数学関数，行列演算関数，複素数関数，フーリエ変換関数，削減，スキャン，セグメンテーション，シーケンスの比較と索引付け，その他の関数とクラス（要素ごとに$x * y$を返す，要素の負の数値を計算，要素ごとに$x - y$を返すなど）
文字列	ハッシング，接合，分割，変換
ヒストグラム	ヒストグラム関連関数
制御フロー	制御フロー操作，論理演算子，比較演算子，デバッグ操作
高次関数	高次演算子
テンソル配列操作	テンソル配列操作のメソッドとクラス
テンソル・ハンドル操作	テンソル・ハンドル操作
イメージ	エンコードとデコード，リサイズ，トリミング，反転，回転，転置，色空間間の変換，画像調整，バウンディング・ボックスの操作
スパース・テンソル	スパース・テンソル表現，変換，操作，削減，数学的な操作
入力と読み込み	プレースホルダ，読み込み，変換，キュー，条件付きアキュムレータ，ファイル・システムへの対処，入力パイプライン
データIO	データIOに関するPython関数
ニューラル・ネットワーク	アクティベーション関数，畳み込み，プール，形態学的フィルタリング，正規化，損失，分類，埋め込み，リカレント・ニューラル・ネットワーク，コネクショニスト時系列分類法，評価，候補サンプル，その他の関数とクラス（バッチ正規化，4-D入力とフィルタ・テンソル与えられた2次元の深さ方向の畳み込みを計算など）
ニューラル・ネットワークRNNセル	全てのRNNセルのベース・インターフェース，TensorFlowのコアRNNメソッドで使用するためのRNNセル，分割RNNセル状態を格納するクラス，RNNセル・ラッパ
グラフの実行	セッション管理，エラー・クラスと便利な関数
トレーニング	最適化，勾配計算，勾配クリッピング，学習率の低下，移動平均，コーディネータとキュー・ランナ，分散実行，要約操作，イベント・ファイルへのサマリの追加，トレーニング・ユーティリティ，状態保存でミニバッチへのシーケンス分割入力，オンライン・データ・リサンプリング，バケット，その他の関数とクラス（集計勾配を同期させてオプティマイザに渡す，var_list変数のために損失の勾配を計算，チーフが終了する前に実行するクリーンアップ操作を返すなど）
Pythonラップ関数	スクリプト言語演算子
要約操作	要約生成，ユーティリティ，その他の関数とクラス（指定されたメッセージの内容を現在のメッセージにコピー，初期化されていない必須フィールドを検索など）
テスト	単体テスト，ユーティリティ，勾配チェック，その他の関数とクラス（TensorFlowベンチマークのヘルパを提供する抽象クラス，ベンチマーク報告，与えられたセッションで操作またはテンソルを実行など）
ベイズ・フロー・エントロピー	操作（負のカルバック・ライブラ情報量や変分下限の出現率の見積もり，モンテカルロまたはシャノンのエントロピーの決定論的計算，Renyiダイバージェンスに現れる比率のモンテカルロ推定，Renyi比率に適した指数関数的に減衰するテンソルなど）
ベイズ・フロー・モンテカルロ	操作（期待値のモンテカルロ推定）
ベイズ・フロー確率グラフ	確率的計算グラフ・ヘルパ関数
ベイズ・フロー確率的テンソル	確率的テンソル・クラス，確率的テンソル値のタイプ，その他の関数とクラス（観測値を持つ確率的テンソルなど）
ベイズ・フロー変分推論	操作（カルバック・ライブラ情報量の計算，計算を制御する定数，変分StochasticTensorを事前分布に関連付けなど）
可変次数	可変次数レイヤ構築の関数
統計的分布	ベース・クラス，スカラ分布，多変量分布，変換分布，混合モデル，先行するコンジュゲートによる通常の尤度，カルバック・ライブラ・ダイバージェンス
ランダム変数変換	全単射
FFmpeg	FFmpegを使用したオーディオのエンコードとデコード
フレームワーク	引き数範囲，変数
グラフ・エディタ	モジュール・ユーティリティ，モジュール選択，モジュール・サブグラフ，モジュール・リルート，モジュール編集，モジュール変換，モジュール・マッチ，有用なエイリアス
統合	操作（常微分方程式のシステムを統合など）
レイヤ	ニューラル・ネットワーク・レイヤ構築のための高レベル操作，正則化，初期化，最適化，要約，特徴列
学習	推定，グラフ・アクション，入力処理
モニタ	操作（通常使用されるモニタのデフォルト・セットを返す，モニタの基本クラス，トレーニングの開始時に呼び出す，トレーニング／評価の終了時にコールバックなど），その他の関数とクラス（モニタをSessionRunHookにラップ，モニタをフックでラップなど）
損失	関数とクラス（トレーニング手順に絶対差分損失を追加，損失のコレクションに外部定義された損失を追加，loss_collectionからの損失リストを取得，総損失を示す値をテンソルに返す，ヒンジ損失の損失テンソルを返すなど）
RNN	RNNセル追加，その他の関数とクラス（双方向GridLstmセル，LSTMセルのパラメータを初期化，レイヤ正規化および再帰的ドロップアウトを伴うLSTMセル，双方向リカレント・ニューラル・ネットワークを作成など）
メトリクス	メトリック操作，操作の設定
ユーティリティ	ユーティリティ関数（TensorProtoを作成，テンソルからnumpy ndarrayを作成，グラフで使用される操作の[illegible]収集など）
グラフ要素のコピー	関数とクラス（1つのグラフからの操作オリジナル・インスタンスを与える，1つのグラフからの変数インスタンスを与える，1つかのグラフから操作インスタンスを与えるなど）

コラム　TensorFlowでもよく出てくる…ディープ・ラーニングがスゴイ理由　佐藤 聖

● 従来は特徴量を手作業で抽出していた

人間は犬や猫の種類が違っても意識せずに犬らしさや猫らしさを読み取り，犬か猫かを判別します．逆に犬らしい特徴を備えた猫は犬と誤認します．ディープ・ラーニング以前は，犬と猫の顔を画像認識させるために研究者や技術者が犬と猫の顔から特徴量を計算してニューラル・ネットワークの変数として設定しました．この作業は機械学習の知識に加えて犬や猫の違いを識別するために動物の知識も必要になります．やっかいなことに画像認識対象を犬や猫ではなく猿とシマウマに変える必要が生じたら猿とシマウマの画像から特徴量計算と変数の設定を手作業で行う必要がありました．

● 人間が考えつかないことを考えてくれる

ディープ・ラーニングでは，入力となる画像データを各層のニューラル・ネットワークで学習を繰り返すことで特徴量が自動計算されます．ディープ・ラーニングの自動計算で特徴が明らかになり，人間が特徴と気づいていない部分が特徴だったり，反対に人間が特徴と考えられていた部分が実は特徴と言えないことが発見されたりすることもあるでしょう．

ディープ・ラーニングは人間と同じ認識機能を提供するだけでなく，人間の認識機能を補完することもできるため，汎用人工知能の実現に寄与するものと期待されています．

表2　APIの具体例1…数学関連（抜粋）

カテゴリ	機能説明	API
算術演算子	要素ごとにx＋yを返す	`tf.add(x, y, name=None)`
基本的な数学関数	xに最も近い要素単位の整数を返す	`tf.rint(x, name=None)`
行列演算関数	指定された対角値を持つ対角テンソルを返す	`tf.diag(diagonal, name=None)`
複素数関数	2つの実数を複素数に変換する	`tf.complex(real, imag, name=None)`
フーリエ変換関数	最も内側に1次元の離散フーリエ変換を計算する	`tf.fft(input, name=None)`
削減	テンソルの次元間の要素の合計を計算する	`tf.reduce_sum(input_tensor, axis=None, keep_dims=False, name=None, reduction_indices=None)`
スキャン	軸に沿ったテンソルxの累積合計を計算する	`tf.cumsum(x, axis=0, exclusive=False, reverse=False, name=None)`
セグメンテーション	テンソルのセグメントに沿った合計を計算する	`tf.segment_sum(data, segment_ids, name=None)`
シーケンスの比較と索引付け	テンソルの軸上で最小の値を持つインデックスを返す	`tf.argmin(input, axis=None, name=None, dimension=None)`

表3　APIの具体例2…ニューラル・ネットワーク関連（抜粋）

カテゴリ	機能説明	API
アクティベーション関数	整流された線形を計算する	`tf.nn.relu(features, name=None)`
畳み込み	N-D畳み込みの合計を計算する	`tf.nn.convolution(input, filter, padding, strides=None, dilation_rate=None, name=None, data_format=None)`
プール	入力の平均プーリングを実行する	`tf.nn.avg_pool(value, ksize, strides, padding, data_format='NHWC', name=None)`
形態学的フィルタリング	4-D入力と3-Dフィルタのテンソルのグレー・スケール拡張を計算する	`tf.nn.dilation2d(input, filter, strides, rates, padding, name=None)`
正規化	L2ノルムを使用して次元dimに沿って正規化する	`tf.nn.l2_normalize(x, dim, epsilon=1e-12, name=None)`
損失	log_inputの与えられたlogポアソン損失を計算する	`tf.nn.log_poisson_loss(log_input, targets, compute_full_loss=False, name=None)`
埋め込み	埋め込みテンソルのリストでIDを検索する	`tf.nn.embedding_lookup(params, ids, partition_strategy='mod', name=None, validate_indices=True, max_norm=None)`
リカレント・ニューラル・ネットワーク	RNNセルで指定されたリカレント・ニューラル・ネットワークを作成する	`tf.nn.dynamic_rnn(cell, inputs, sequence_length=None, initial_state=None, dtype=None, parallel_iterations=None, swap_memory=False, time_major=False, s...None)`
評価	最後の次元のためにk個の最大エントリ値とインデックスを検索する	`tf.nn.top_k(input, k=1, sorted=True, name=N...`

CQ出版社

Interface 増刊 人工知能を作る

AI
人工知能

人工知能を作る

CQ出版社
〒112-8619 東京都文京区千石4-29-14
TEL.(03)5395-2141(営業部)
定価:本体2,600円+税

見本

雑誌　01620-4
Ⓛ-2018.7.30