

Deep Learningの実装と現状

中山浩太郎 岩澤有祐 黒滝紘生 松尾 豊

東京大学

Deep Learning

1970年代に登場したパーセプトロンに起源を持つ Multi Layer Perceptron (以降 MLP) などのニューラルネットワークは、線形分離不能な問題に適用できる機械学習のアルゴリズムとして、幅広い分野で利活用されている。ニューラルネットワークは、理論的には層を重ねることで表現する能力が高まり、より複雑な問題に適用できるはずだと考えられていたが、実際には層を重ねていくと、学習が収束せず、認識精度は向上するどころか大幅に下がることが指摘されていた。そのため、歴史的には3層程度のレイヤ数の低い MLP が現実的なネットワーク構成として採用されてきた。

ニューラルネットワークの研究は、長らく顕著なブレイクスルーがなかったため、研究分野としては最近は大きな着目を集めてきたとは言いがたい状況であったが、2006年に Hinton ら¹⁾ が3層以上多層にレイヤを重ねても、レイヤごとに事前に学習を行うことで、深い層を持つネットワークでも学習ができることを証明し、「Deep Learning」として再び注目されるようになった。

Deep Learning のアルゴリズムには、Deep Berief Networks (以降 DBN) や Stacked denoising Autoencoders (以降 SdA) のように教師なし学習による事前学習 (Pre-training)^{☆1} を行った後に、一般的な MLP と同様にラベルを利用した教師あり学習 (Fine-Tuning)^{☆2} を行うものがある。この方式のアルゴリズムにおいては、

☆1 事前学習では、入力層に近い層から順番に入力データを復元する表現を隠れ層との結合を調整することで学習する。

☆2 Fine-Tuning では、出力層において教師データとの誤差を最小化するように学習する。

事前学習を行うことで、データに内在する重要な特徴表現を学習していることが従来研究によって明らかにされている²⁾。たとえば、Le らの研究³⁾では、深い階層を持つネットワークに対し、大量の画像を利用して事前学習を行うことで、下位の層 (入力に近い層) ではエッジ情報などプリミティブな特徴表現が学習されたことを確認し、より上層では猫や人といった抽象度の高い特徴表現が学習できていたことを確認している。

一方、Maxout や CNN のように事前学習を明示的に行わないアルゴリズムも存在する。そのため、事前学習によって特徴抽出を行うものを総称して Deep Learning である、と一様にまとめることは難しい。しかし、下層 (入力層に近い層) で、入力情報の特性を学習し、特徴的な表現を抽出することで、深い層を持つニューラルネットワークの学習が可能になっている点は多くのアルゴリズムで共通している。そのため、本稿では、下層で特徴的な表現を学習する仕組みを持つことで、深いレイヤ構造のニューラルネットワークの学習を可能にしているアルゴリズムを総称して Deep Learning と呼ぶこととする。

以下、Deep Learning の実装およびその周辺の情報について俯瞰する。

Deep Learning の実装

ここ数年で Deep Learning に関する実装が精力的に整備され、すでに多くのライブラリや計算モデルが公開されている。本章では、その中でも比較的コミュニティが大きく、活発に開発が進められているものとして、Caffe, Torch7, Pylearn2 を中心に解説する。Caffe は

実行速度やコミュニティの面でリードしており、Torch7 はモデルの構築や利用方法が容易であるという特徴を持つ。また、Pylearn2 は独自のモデルを詳細なレベルで構築・利用することやライブラリをカスタマイズすることが可能であり、研究者用の機能が充実しているという特徴を持っている。以下、各ライブラリについて詳細に説明する。

□ Caffe

概要

Caffe^{☆3} は、Berkley Vision and Learning Center (BVLC) によって開発された、Deep Learning のライブラリの1つである。Caffe の大きな特長として、高速な動作・ネットワーク構造の記述の容易さ・コミュニティが充実していることなどが挙げられる。

Caffe の中核部分は、プログラミング言語 C++ によって記述されている。具体的には、学習アルゴリズムやレイヤの計算処理などが該当する。Caffe の特長である高速な実行速度も、C++ によって支えられている部分が多い。また、これに加えて NVIDIA 社が開発した、高速に畳み込み計算を行うことができるライブラリ「CuDNN」を利用することで、さらに高速にネットワークの学習処理を行うことが可能である。Caffe では設定ファイルを変更するのみで CuDNN を利用できる。

Caffe では、利用者の利便性を向上させるために、C++ から Caffe の各種処理を呼び出す方法以外に、2つの利用方法を提供している。まず、1つ目の利用方法は、Python と Matlab 用のプログラミングインタフェースである。これらのインタフェースを利用することにより、C++ よりも簡単な記述で、各種機能にアクセスすることができる。また、2つ目の利用方法は、prototxt 形式と呼ばれる、テキストベースの構造化設定ファイルである。この設定ファイルを利用することで、プログラミングを一切行わなくても、新しいネットワーク構造を記述できる。prototxt の設定ファイルは、ネットワーク構成そのものの定義ファイルと、学習率など学習方法を定義するファイルの2つに分かれているが、

これらの定義をそれぞれ独立させることで、実験設定の再利用性を向上させている。

Caffe の主な機能

Caffe には、Deep Learning を構成するための、さまざまなレイヤが実装されている。レイヤには、シグモイド関数、ReLU 関数、畳み込み、ソフトマックスなどが含まれており、Dropout のオプションも用意されている。また、学習アルゴリズムとしては、SGD (確率的最急降下法)、AdaGrad などを利用することができる。

Caffe のチュートリアルには、ネットワークの定義および学習結果のサンプルがついている。これらは「Model Zoo」と呼ばれ、画像認識のためのネットワークとして使うことができる。また、チュートリアルには、著名な画像データセットである、MNIST・CIFAR10・ImageNet の画像を扱う方法が示されており、Caffe による Deep Learning 研究をすぐに始めることが可能である。

また Caffe には、Python ベースのユーティリティツールが付属している。たとえば、prototxt によるネットワーク定義ファイルから、ネットワークの接続図を画像ファイルとして出力することができる。本機能を利用することで、複雑な構成のネットワークであっても全体像が容易に把握できるようになる。また、Caffe を利用して学習したネットワークについて、内部的にどのような学習が行われたのかを解析するために、重みフィルタを可視化して画像に出力することも可能である。図-1 は、ImageNet の画像識別タスクのために学習された重みフィルタの例である。

Caffe では、CPU を使った計算と GPU を使った計算を容易に切り替えることが可能である。具体的には、学習方法の設定ファイルに記述するパラメータを1つ修正するだけで処理を切り替えることが可能である。これは、多くの Deep Learning 研究について言えることだが、開発中は GPU を搭載していない開発用の計算機を利用し、大規模な解析を行う際には高性能の GPU を搭載したサーバを利用することがある。そのため、GPU と CPU を使った処理を容易に切り替えることができるのは、Deep Learning の実装として重要な要素の1つであると言える。

☆3 <http://caffe.berkeleyvision.org/>

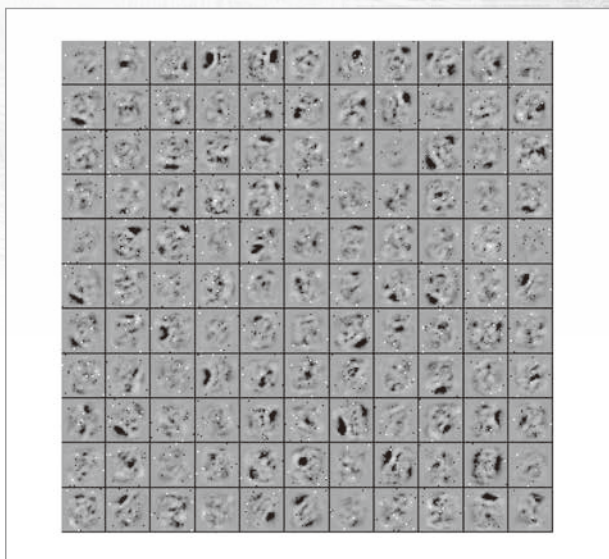


図-1 Caffe で学習された MNIST の重みフィルタ (各セルが隠れ層のユニットに対応する。セルの中央あたりに曲線のパターンなどを学習していることが読み取れる)

Caffe を利用した研究・プロジェクト

Caffe は、オープンソースのプロジェクトであり、そのコミュニティは非常に活発に研究開発を続けている。Caffe による成果が共有されることも比較的多い。画像認識のコンテストである ILSVRC 2014 でトップに立った、GoogLeNet モデルや、物体認識に使える R-CNN の実装、ILSVRC 2012 で使われた ImageNet の分類モデル、AlexNet などが提供されている。また、他の Caffe 利用者が作ったモデルを、Github (gist) を介して共有することができる。この形式で共有されたモデルの例として、「Network in Network」モデルなどがある。

□ Torch7

概要

Torch7^{☆4} は、軽量スクリプト言語「Lua」で記述された科学技術計算ライブラリである。Pylearn2 や Caffe が基本的に Deep Learning やそれにかかわる基本技術を実装したライブラリであるのに対して、Torch7 はニューラルネットワークに強みを持つものの、どちらかという科学技術のための実装プラットフォームとしての色合いが強い。そのため、ニューラルネットワーク

を含む多様な機械学習や信号処理など幅広い実装が提供されており、さまざまな場面で利用することができるという特徴がある。もともとは NEC Laboratories America で開発・実装されていたものであったが、最近では Facebook が自社内での人工知能プロジェクトに関する Torch7 上の実装を公開するなど、さまざまな企業や研究室で利用・開発が行われている。

Torch7 は、基本的には、1) C で書かれた高速な数値計算機能、2) 上位機能を利用するための Lua のインタフェース (torch.Tensor)、3) Lua で記述された拡張パッケージの 3 つの要素で構成されている。C 言語で書かれた数値計算機能とそれを利用する lua インタフェースが提供されているため、既存のモデルを利用したり新しいモデルを作成する際には、基本的には C 言語の最適化など複雑な部分を考えずに拡張部分のみを lua を利用して記述することができる。

Torch7 の主な機能

先ほど述べたとおり Torch7 は機械学習に関する機能を広く網羅しているため、たとえばグラフィカルモデルや SVM などを利用することもできるが、ここではニューラルネットワークの実装に焦点を当てて特徴を述べる。Torch 自体が提供する数値計算機能の上で、さまざまなニューラルネットワーク (NN) に関する拡張パッケージが公開、提供されている。表-1 は、Torch7 で提供されているニューラルネットワークの実装に関する代表的なパッケージをまとめたものである。ただしこれらは筆者らが選択した代表的なものであり、ここに挙げることができなかったパッケージも存在していることには留意されたい。Torch7 では、これらの機能を個別に利用したり、組み合わせたりすることで柔軟にモデルを作成、利用することができる。一部のパッケージは標準ではインストールされていないが、torch をインストールした際に標準搭載されている「LuaRocks」と呼ばれるスクリプトを利用することで多くの場合は簡単にインストールすることが可能である。

標準パッケージとして提供されている「nn」と「optim」では、ニューラルネットワークの構築と最適化に関する機能が網羅的に提供されている。「nn」は基本的な MLP のネットワークを実装するためモジュールを提供するパッケージであり、畳み込みや隠れユニットなどの

☆4 <http://torch.ch/>

名称	概要 (提供するモジュール, 機能)
nn	基本レイヤ, 基本活性化関数, 誤差関数
optim	目的関数の最適化用のモジュール
unsup	教師なし学習に関する機能
dp	モデルの実行を補助する周辺機能
cunn	モデルを GPU で実行するための機能
nnx	nn の実験的な拡張機能
nnglyph	nn で作ったモデルの可視化機能

表-1 Torch7のNN関連パッケージ

基本的なレイヤ, シグモイド関数や ReLu などの活性化関数, Dropout などの拡張機能を nn.Sequence と呼ばれる「箱」に入れていくことで複雑なネットワークをシンプルに記述することができる。また, 単純な1つの出力を持つようなニューラルネットワークのほかにも, nn.Parallel や nn.Concat と呼ばれるモジュールを利用することで, 出力を複数持つような複雑なネットワークを比較的簡単に記述できる点も特徴的である(図-2を参照)。
 ‘optim’はさまざまな最適化関数を実装したパッケージであり, MiniBatch-SGDをはじめとして, RMSProp, Adam などの最新の手法が実装されている。
 ‘optim’を利用することで, nnで記述したモデルをさまざまな手法で最適化することができる。
 ‘unsup’は, 教師なし学習に関するモジュールを提供するパッケージである(‘unsup’は unsupervised の頭をとったもの)。
 Autoencoder などのモジュールが nn と併用可能な形で提供されているほか, k-means や PCA などの手法も提供されている。
 ‘dp’は nn で提供されている機能に加えて, Mnist や Cifar10 などの標準的なデータセットを読み込む機能や実験結果をファイルに保存する機能など, 実際にモデルを動かして評価を行う際に便利な周辺機能がオブジェクト指向のデザインパターンで提供されている。また, ‘cunn’や ‘cutorch’を利用することで, その他の記述部分を大きく変更することなく GPU を利用することができる。

Torch を利用した研究・プロジェクト

Torch を利用した代表的な研究・プロジェクトとしては, Facebook が公開した fbcunn に関するものがある。また, Neural Turing Machine や変分オートエンコーダなどのデモが公開されるなど, 最新の研究事例が精力的に実装されている^{☆5}。そのほか, ‘loadcaffe’ という

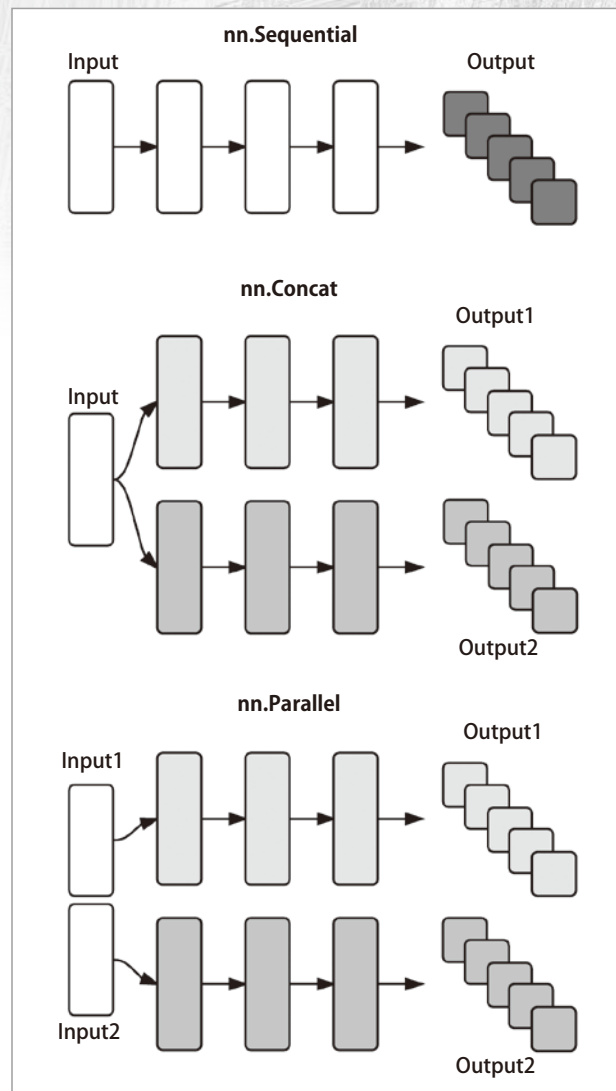


図-2 Torch7での複雑なNN実装(複数の下層レイヤからの入力を束ねるネットワークなどが簡単に記述できる)

パッケージを利用することで caffe 上で提供されているモデルを Torch7 上で利用することができたり, iOS や Android 上で利用できるように修正された torch を利用することでスマートフォン上で利用できたりと, 利用可能性を拡張するプロジェクトが多くなされている。

□ Pylearn2/Theano

概要

Pylearn2^{☆6}は, Python で開発された Deep Learning のライブラリとしては, 最も有名なものの1つであ

☆5 <https://github.com/torch/torch7/wiki/Cheatsheet>
 ☆6 <http://deeplearning.net/software/pylearn2/>

り、Montreal 大学の Yoshua Bengio が所属する LISA Lab が開発主体となって活発に開発が進められている。

図-3 に Pylearn2 の構造を示す。

Pylearn2 は内部的に Theano という完成度の高い汎用的な数値計算ライブラリを利用している。Theano は汎用的な数値計算のための機能を多数提供しているが、テンソルの計算や自動微分といった、ニューラルネットワークの計算やモデルを簡単に記述できる機能が特に充実しているという特徴を持っている。この結果、Pylearn2 上でのモデル開発が効率化され、SdA, Maxout, Dropout, CNN や DBN といった幅広い種類のアルゴリズムが実装・提供されている。

また、Theano の重要な機能の1つとして、GPU コードの自動生成がある。Theano では設定を変更するだけで GPU 用の C コードを自動的に生成する機能があり、Python 側のコードを一切変更せずに CPU モードと GPU モードを切り替えできる。そのため、Theano をベースにした Pylearn2 も設定を変更するだけで GPU を利用した高速な解析 (CPU と比較して数倍から数十倍程度高速) が可能となる。また、GPU を利用しない場合であっても、各種の計算処理に対して最適化された C のコードを生成して利用するため、ある程度高速に動作する。

Pylearn2 の主な機能

Theano が数値計算機能や GPU コードの生成といった基礎的な部分を担当するのに対し、Pylearn2 は機械学習や Deep Learning, ニューラルネットワークなどのより高位の処理を担当する。以下、Pylearn2 の主な機能について説明する。

Pylearn2 では、YAML 形式の設定ファイルでモデルや学習方法を記述するだけで学習を実行できるため、Python コードを書かなくても利用を始めることが可能である。具体的には、データセットやモデル (ネットワークの構成)、学習アルゴリズムなどを記述した設定ファイルを用意・利用し、訓練用のスクリプトを実行することで利用可能である。内部的には設定ファイルから Python オブジェクトをシリアライズする仕様になっているため、設定ファイルの定義自身に、多少冗長な面はあるものの、コードを記述せずに学習やテストができる

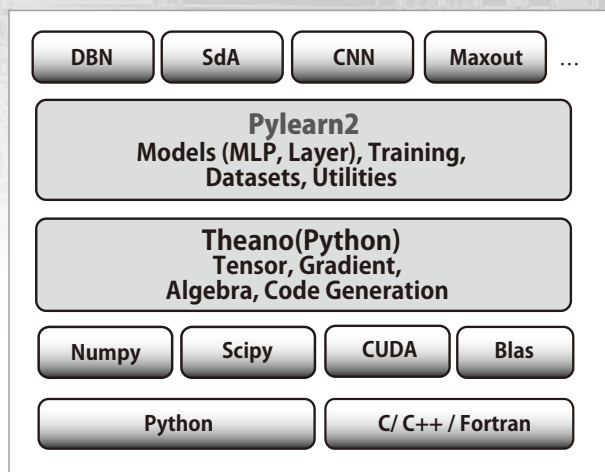


図-3 Pylearn2 の構造 (内部的に GPU 用のコードや CPU 用のコードを自動生成する)

のは特筆すべき特徴の1つである。この仕様は、自分でモデルを作成した場合にも設定ファイルで簡単に参照できるという利点がある。

また、Pylearn2 には訓練を効率的に行うためのツールや、可視化のスクリプトなどが用意されている。具体的には、学習したネットワークを可視化するためのスクリプトや、学習結果のサマリーを出力する機能などが提供されている。

さらに、学習アルゴリズムとしてよく利用される SGD (Stochastic Gradient Descending) については、学習率 (Learning Rate) を自動調整する機能や、前回のパラメータ修正値を利用する Momentum 機能などが用意されており、効率的にモデルを訓練することが可能となっている。

Pylearn2 を利用した研究・プロジェクト

Pylearn2 を利用した研究は数多く存在するが、代表的なものでは Maxout Networks や Dropout に関する論文がある。これらの研究の成果を再現する設定 (YAML ファイル) が Pylearn2 のリポジトリ内に存在しており、簡単に利用することが可能になっている。

□ その他の実装

ConvNetJS

ConvNetJS^{☆7} は、JavaScript で記述された Deep

☆7 <http://cs.stanford.edu/people/karpathy/convnetjs/>

Learning ライブラリであり、画像認識や Deep Q Learning などのデモが公開されている。ConvNetJS の大きな特徴は、JavaScript で実装されているため、環境整備をする必要なく、一般的なブラウザ上で実行できる点である。しかし、その一方で、大量のデータを扱うようなアプリケーションに対しては実用的な実行速度を得ることは難しいという問題がある。これは、そもそも JavaScript は Web アプリケーションの非同期処理に向いているものの、高度な数値計算用などに向いていないことに起因する。

図-4 に ConvNetJS で実装された Deep Q Learning のデモを示す。

また、JavaScript で記述されたソフトウェア全般に言えることだが、通常の数値処理に関するコーディングは冗長になる傾向にあることには注意したい。

Keras

Keras^{☆8} は Pylearn2 と同様、Python と Theano をベースにした実装だが、設計思想は Torch7 に似ており、非常にシンプルな記述で複雑なモデルを記述できる上に、訓練に便利な機能が用意されている。また、畳み込み、Dropout、Relu などのレイヤを組み合わせ（積み重ね）て、複雑なネットワークを構築できるほか、RNN や LSTM などのモデルやサンプルも提供されている。Pylearn2 と比較した場合、対応しているデータセットや研究開発に便利な機能としては制限があるものの、非常に活発に開発が進められており、これからの躍進が期待できる。

GeSdA

上述の通り、Deep Learning に関する各種ライブラリが開発されているが、ここで筆者らの研究グループが開発している「GeSdA (GPU empowered Stacked denoising Autoencoder)」を紹介する。GeSdA は、完成度の面では上述のライブラリ群と比較できるレベルにはまだ達していないが、GPU (CUDA) 上で高速に動作する SdA の実装であり、多様なアプリケーション

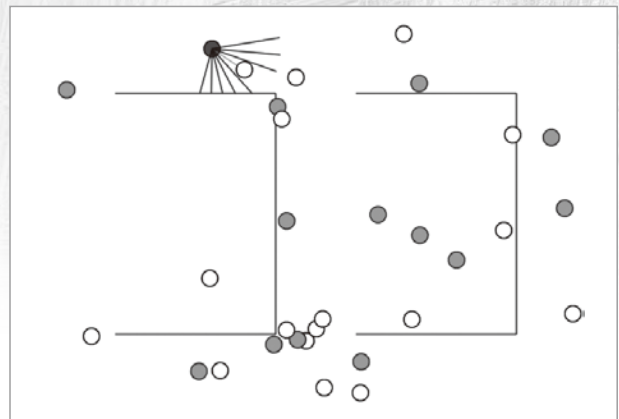


図-4 ConvNetJS : Reinforcement Learning with Deep Q Learning (強化学習によってエージェントがグレーの玉を避けつつ白の玉を取得する行動を学習する)

に適用可能な汎用性の高さが特徴である。

□ 概要

GeSdA は、Deep Learning のアルゴリズムの代表例の1つである SdA に特化したライブラリであり、SdA の事前学習 (Pre-training) の各種処理 (エンコード・デコード・誤差計算・パラメータ更新) を GPU に最適化させることで処理を高速化させている。具体的には、GPU 内にキャッシュする情報を工夫し、並列可能な部分を並列化して動作させることで高速な計算を実現している。これは、GPU 内部では、スレッドと呼ばれる処理単位が「共有メモリ」(Shared Memory) と呼ばれる高速なメモリを共有することを利用できる点と、ホストとデバイス間の制御フローの往來を最小化することで高速化できることを利用した結果である。具体的には、CPU (Numpy + Blas) と比較した場合、5 倍から 10 倍程度高速になり、Theano で実装した SdA と比較して約 2 倍程度高速に動作することが分かっている。

□ GeSdA の機能

GeSdA には、高速に動作するほかにいくつか特徴的な機能があるが、その中でも主要なものを以下に列挙する。

まず、1つ目の特徴は、簡潔かつ汎用性の高い YAML ファイルによるモデルの記述が可能な点である。これは、Pylearn2 の YAML や Caffe の Prototxt は冗長な記述が必要になることが多いという問題を解決す

☆8 <http://keras.io/>

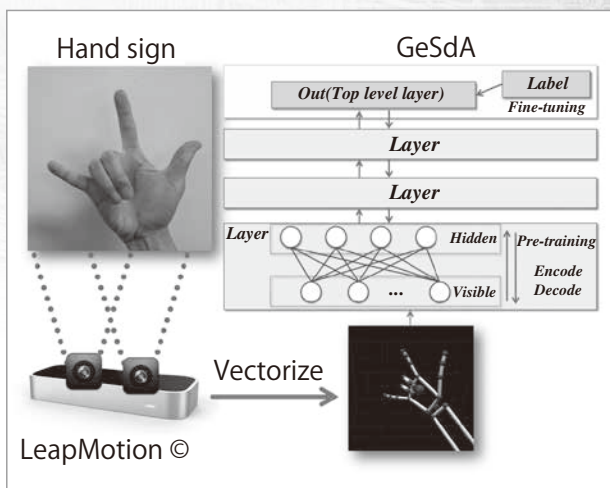


図-5 DeepMotion の概要 (SdA の事前学習によって各ジェスチャや指の動作の特徴を学習する)

ることを目指したものである。2つ目の特徴は、標準でグリッドサーチの機能を備えている点である。Deep Learning の研究において技術的に困難な点の1つとして、学習率やレイヤ数、ノード数といったハイパーパラメータについて、最適な組合せを見つけることが難しいことが挙げられる。GeSdAでは、これらのハイパーパラメータの組合せを自動的に検証し、分類精度を高めるための組合せを自動的に発見することができる。特に重要な点は、レイヤ数も1つのパラメータとして指定できる点である。また、グリッドサーチで並列処理するプロセス数を制御することが可能であり、指定したプロセス数を超えないように自動的に並列計算を行う。3つ目の特徴は、LibSVM 形式などの汎用的なデータの読み込みに対応している点である。汎用的なデータセットへ対応することにより、多様なアプリケーションへの適用が容易になった。

□ アプリケーション例：Deep Motion

筆者らの研究グループでは、単にライブラリを開発するだけでなく、SdA や GeSdA の有効性を示すためにアプリケーション開発を進めている。ここではその一例として、ジェスチャ認識システム「DeepMotion」を紹介する。DeepMotion の概要図を図-5 に示す。

DeepMotionでは、センサから指の位置や方向といった情報を受け取り、多層の SdA で学習し、ジェスチャ認識へ利用している。DeepMotion の評価実験では、

ライブラリ	主な特徴	利用方法
Caffe	<ul style="list-style-type: none"> • 高速な動作 • モデルや実装を共有するコミュニティが充実 • 画像処理のモデルが充実 	<ul style="list-style-type: none"> • Protobuf ファイルでモデル作成 • Python/Matlab インタフェース • C++ 言語でのモデル作成
Torch7	<ul style="list-style-type: none"> • 高度なパッケージ管理機能 • シンプルかつ強力なモデル設計機能 	<ul style="list-style-type: none"> • Lua 言語でのモデル作成
Pylearn2	<ul style="list-style-type: none"> • 高いカスタマイズ性 • モデル作成に Theano の自動微分が利用可能 	<ul style="list-style-type: none"> • YAML ファイルでモデル作成 • Python でモデル作成

表-2 ライブラリー一覧

GeSdA のグリッドサーチ機能を利用して各種パラメータで学習したところ、従来の MLP や SVM に比べてエラー率が大幅に減少するパラメータの組合せを発見できることを確認した。また、SVM や従来の MLP などでは難しかった微量の差を特徴量として検知し、認識精度が向上している点も確認した。

まとめ

Deep Learning に関する技術開発は活発に進められており、Caffe, Torch7, Pylearn2 といったライブラリが開発されてきた。表-2 に本稿で紹介した主なライブラリの特徴を列挙する。Caffe は実行速度や充実したコミュニティの面で評価ができる。また Torch7 は、パッケージ管理機能が充実しており、多くのパッケージを追加することができるほか、非常にシンプルな表現で複雑なネットワークを構築できるといった特徴を持っている。Pylearn2 は自動微分や高度な行列処理が可能な Theano をベースに開発されており、新しいアルゴリズムの構築とその利用が簡単にできる点が特徴である。また、筆者らの研究グループで開発している GPU 上で高速に SdA の計算をする「GeSdA」は、グリッドサーチや YAML によるモデル構築などが可能であり、各種アプリケーションへの適用が容易である。さらに、最近では Keras や ConvNetJS など、新しいライブラリも登場し、より活発に研究開発が進められている印象を受ける。

また、各種の IT 企業も非常に活発に Deep Learning に関する研究開発へ投資し、その勢力圏は非常にめまぐるしく変化している。たとえば、Facebook は CNN 研究の第一人者 Yann LeCun を招聘し、パリに Deep Learning に関する研究所を設立した。さらに、Google は Geoffrey Hinton や、Caffe の作者である Yangqing Jia を招聘しているほか、先端の Deep Learning 研究において高いプレゼンスを発揮し続けている。Baidu はシリコンバレーに Deep Learning 研究所を設立し、機械学習の研究で著名なスタンフォード大学の Andrew Ng を招聘している。

これらの企業は、人材を囲い込むだけでなく、積極的に Torch7 や Caffe のコミュニティに参加していることも多い。たとえば、前述のとおり、Facebook は Torch7 の上で動作する fbconv というライブラリを公開し、Nvidia は各種のライブラリと組み合わせて使うことで畳み込み計算を高速化する CuDNN を公開している。また、Google は Caffe のコミュニティへ学習済みのネットワークやモデルを提供し、DeepDream などのインパクトのある成果に関するコードを公開するなど、積極的に研究コミュニティへ貢献している。このように、民間企業・大学含めて非常に活発に Deep Learning 研究が進められており、さらなる発展が期待できる分野であると言える。

参考文献

- 1) Hinton, G. E. and Salakhutdinov, R. R. : Reducing the Dimensionality of, *Science*, Vol.313, No.7, pp.504-507 (2006).
- 2) Bengio, Y., Courville, A. and Vincent, P. : Representation Learning : A Review and New Perspectives, No.1993, pp.1-34 (2012).
- 3) Le, Q. V., Ranzato, M. A., Devin, M., Corrado, G. S. and Ng, A. Y. : Building High-level Features Using Large Scale Unsupervised Learning, *Proc. of International Conference on Machine Learning (ICML)* (2012).

(2015 年 7 月 31 日受付)

■ 中山浩太郎 (正会員) nakayama@weblab.t.u-tokyo.ac.jp

東京大学工学系研究科特任講師。2007 年に大阪大学大学院情報科学研究科博士号取得後、大阪大学情報科学研究科特任研究員、東京大学知の構造化センター助教、講師を経て、現在に至る。専門分野は AI, Web, 大規模データ解析。

■ 岩澤有祐 iwazawa@weblab.t.u-tokyo.ac.jp

上智大学理工学研究科修士課程卒業。2014 年より東京大学博士後期課程在学。学術振興会特別研究員 (DC1)。専門分野は、人工知能、人間行動センシング、人間行動認識。

■ 黒滝敏生 (正会員) kurotaki@weblab.t.u-tokyo.ac.jp

東京大学工学系研究科修士課程在学中。東京大学工学部卒業。専門分野は、人工知能。

■ 松尾 豊 (正会員) matsuo@weblab.t.u-tokyo.ac.jp

東京大学工学系研究科特任准教授。2002 年同大学院にて博士号 (工学) 取得。その後、産業技術総合研究所研究員、スタンフォード大学客員研究員を経て現在に至る。専門分野は、人工知能、Web マイニング、ビッグデータ分析。人工知能学会からは論文賞 (2002 年)、創立 20 周年記念事業賞 (2006 年)、現場イノベーション賞 (2011 年)、功労賞 (2013 年) の各賞を受賞。人工知能学会 学生編集委員、編集委員を経て、2010 年から副編集委員長、2012 年から編集委員長・理事。2014 年より倫理委員長。