Computer Security Symposium 2017
23 - 25 October 2017

亜種マルウェア自動検出のためのデータ圧縮と深層学習

長野 雄太[†1]　宇田 隆哉[†1]

概要：近年，マルウェアがコンピュータユーザの脅威となりセキュリティ上の重大な課題の１つとなっている．マルウェアの亜種はツールキットの台頭により簡単に大量に作成されるため，マルウェアの増加の原因となっている．また，既存手法により自動的に検出することは困難になってきている．課題解決のため，本論文では畳み込みニューラルネットワークを用いることにより既存手法に頼ることなく亜種マルウェアを自動的に検出する手法を提案する．シンプルな構成の畳み込みニューラルネットワークにマルウェアの Hexdump を学習させることにより，マルウェアと良性なソフトウェアを２値分類し，亜種マルウェアをマルウェアファミリーへ多クラスに分類する．

キーワード：マルウェア検出，機械学習，深層学習，畳み込みニューラルネットワーク，データ圧縮

1. はじめに

1.1 背景

マルウェアとは，コンピュータ上で悪意のある動作を行うソフトウェアやコードの総称であり，セキュリティ上の重大な課題の 1 つである．マルウェアの活動により機密情報の漏洩や金銭の奪取などのユーザへの直接的な被害や，ユーザが気づかぬうちに感染した端末がボットとなり悪意のある攻撃者に利用されてしまう被害などがコンピュータユーザに対してもたらされており脅威となっている．また，マルウェアを作成できるツールキットの台頭により，既知のマルウェアとはバイナリの異なる亜種のマルウェアが簡単に大量に作成される．McAfee から報告された 2017 年 6 月のレポートでは，多数の新たなマルウェアのサンプルが検出されたと記されている[1][2]．マルウェアの感染経路には，悪意のあるウェブサイトからダウンロードされたファイルや電子メールに添付されたファイルをユーザが実行することによる感染などがあり，このような疑わしいファイルをユーザが実行してしまう前に自動的に検出できる手法が研究者により提案されている．

マルウェアを自動的に検出するために，多くのアンチウイルスソフトウェアが始めの検出手法として既知のマルウェアのハッシュのリストと対象のファイルのハッシュを比較することで検出するパターンマッチング方式を採用している．しかしながら，マルウェアの亜種のハッシュがリストに登録されておらず，この方式により正しく検出されない場合があり問題となっている．

パターンマッチング方式により判定出来ないが，それでも疑わしいファイルは，クラウド環境上に構築されたサンドボックスに送信される．そして，サンドボックス上で自動的に動的解析され，悪意の有無について判定される．しかしながら，近年，マルウェアがサンドボックス環境上であるかどうかを区別し，サンドボックス上であるとみなされると活動を止めてしまう機能を持つようになってきている．この機能のため，サンドボックス上での動的解析が正しく行えない問題がある．また，Yokoyama らの研究では，サンドボックスと人間の管理下にある PC のハードウェアや履歴などの複数の特徴を機械学習の分類器に学習させ，その分類器により，サンドボックスであるかどうかの区別を非常に高い精度で行えることが指摘されている[3]．

1.2 提案

本研究では，前述の問題を抱えた背景を踏まえて，既存手法のパターンマッチング方式やサンドボックス上での動的解析を用いずに，深層学習を用いることで自動的に亜種マルウェアを検出する手法を提案する．具体的には，実行ファイルを配列に変換しラベル付けを行い，畳み込みニューラルネットワークを用いた深層学習の分類器に配列を入力し学習させる．その学習済の分類器を用いることで，マルウェアと良性なソフトウェアを判別し，また，マルウェアをマルウェアファミリーに分類する．これにより，マルウェアの亜種を正確に自動的に検出できると考えられる．この際に，実行ファイルをそのまま配列に変換するのではなく，その前にサイズ削減のために，実行ファイルを N-gram と頻度分析を組み合わせた手法などにより圧縮し，圧縮済のデータを配列に変換する．圧縮することにより，畳み込みニューラルネットワークに入力される配列サイズが巨大にならず，学習と分類のために割かれるコストの高いリソースを用意する必要がなくなり，現実的にシステムとして実装可能になると考えられる．

本論文では，2 章にて本手法で用いられる要となる技術である畳み込みニューラルネットワークについて説明し，次に，３章にて関連研究について示し，4 章にて提案手法の詳細について説明する．最後に，考察と今後の課題について述べる．

2. 畳み込みニューラルネットワーク

本論文で提案される手法で使用される要となる技術である深層学習では畳み込みニューラルネットワークをベース

†1 東京工科大学
Tokyo University of Technology

とした分類の手法が用いられる．この章ではその畳み込みニューラルネットワークについて説明される．近年，深層学習は物体認識や音声処理，自然言語処理などの幅広い分野で使用されている[4]．特に画像の分類などの物体認識のために深層学習を用いる場合には，畳み込みニューラルネットワークをベースとした手法が用いられることが多く，コンペティションなどでも非常に高い認識精度を誇っている．深層学習における畳み込みニューラルネットワークはいくつかのレイヤを持っており，それに畳み込み層とプーリング層が含まれていることが特徴である．このレイヤで行われる演算が入力されるデータに対して適用されることで特徴量が抽出され，ネットワークが持っている重みなどのパラメータが更新されていく．学習が完了した後に，更新された重みなどのパラメータを持ったネットワークによって未知のデータが分類される．

機械学習では，人が設計した特徴量の抽出の手法を用いることで画像などのデータをベクトル化しているが，深層学習ではその特徴量の抽出も機械が自動的に行うため，生のデータを学習させることが可能であり，分類のタスクに応じて人が適切な特徴量を考える必要がない．

多くの深層学習のためのモジュールでは学習後に重みなどのパラメータを保存することが出来るため，この保存済みのモデルを分類の際に使うことで学習にかかるリソースを用意する必要がない．そのため，一度，高性能なコンピュータ上で学習を完了してしまえば，その学習済みのモデルを限られたリソースの端末上で読み込むことで分類が可能となる．この技術がスマートフォンのカメラを使ったアプリで物体認識をする場合などに活用されている．

3. 関連研究

3.1 Malware Images

マルウェアの実行ファイルを 2 次元配列に変換し，機械学習の分類器に学習させることで自動的に分類する技術に関する研究について記す[5][6]. L.Nataraj らはマルウェアの実行ファイルをグレースケールの画像に変換し，その画像と機械学習を組み合わせることでマルウェアを自動的に分類する研究を行った．具体的には，マルウェアの実行ファイルを 2 Byte ずつ読み込み，幅が固定された 2 次元配列へと変換する．これによりファイルサイズによって高さが可変するグレースケール（0:黒，255:白）の画像へと変換される．グレースケールの画像から GIST という画像のテクスチャ特徴を計算するための手法を使って特徴量を抽出し，その特徴量を k 近傍法（k=3）の分類器に学習させ，学習済の分類器に未知のファイルをマルウェアファミリーに分類させた．25 種類のマルウェアファミリーを持つ 9458 サンプルのマルウェアのデータセットを，評価手法として 10 分割交差検証を用いて実験を行い 98%の精度を実現したと報告されている．

グレースケールの画像の視覚的類似性にも着目しており，多くのマルウェアファミリーのマルウェアの画像はレイアウトやテクスチャなどが視覚的に類似したと報告している．論文では，Fakerean のファミリーに属する 3 つのサンプルの画像と Dontovo.A のファミリーに属する 3 つのサンプルの画像が示されており視覚的類似性が実際にあることが考えられる．

3.2 Sandprint

Yokoyama らによるサンドボックス環境を自動的に検出する技術に関する研究について記す[3]．この研究では，サンドボックス環境と PC ユーザの環境とを機械学習により 2 値に分類することで，システムがサンドボックスであるかどうかを最大で 100%の確率で判断出来ると示している．評価実験では，Amnpardaz Sandbox や Anubis などの 20 種類の実際のマルウェア解析サービスで提供されている 152 台のサンドボックスやコンピュータユーザが使用している 50 台の PC から，CPU コア数，RAM 容量，PS/2 マウスの有無，ディスプレイ解像度，ログイン履歴，IE のアクセス履歴，システムの稼働時間，ファイルイメージ名の変更の有無，クリップボードの中身とシステム製造元の値の合計 10 種類の特徴を取得し，これらの特徴を SVM に学習させ，サンドボックス環境であるかユーザの PC の環境であるかについて 2 値分類した．その結果，10 種類のすべての特徴を用いた場合には 100%の精度で分類出来たと報告している．

この研究における手法が悪意のある攻撃者によって実際にマルウェアのシステムに組み込まれ脅威となった場合には，サンドボックスを用いて動的解析する既存の手法ではマルウェアを自動的に検出することは困難になると考えられる．

4. 提案手法

パターンマッチング方式やサンドボックス環境での動的解析などの既存手法を用いたマルウェア検出における問題を解決するために，深層学習を用いた分類による亜種マルウェアの自動検出手法を提案する．この章では提案手法の詳細について記す．図 1 に提案手法の概要図を示す．提案手法は大きく 3 つのステップに分かれる．はじめに，実行ファイルを圧縮し配列に変換する．その後，作成された配列にラベルを付けて畳み込みニューラルネットワークへ入力し学習を行う．最後に，学習済みの畳み込みニューラルネットワークが未知のファイルをマルウェアか良性なソフトウェアに分類する．そして，マルウェアである場合には，マルウェアファミリーに多クラス分類する．これにより，マルウェアと良性なソフトウェアを区別することが出来る．なおかつ，そのマルウェアの種類を特定出来る．

3 章にて紹介した L.Nataraj らの研究により同じマルウェアファミリーに属するマルウェアのサンプルを 2 次元配列

に変換し，それをグレースケールの画像として表示させた場合に，それらは視覚的類似性を持っているという特徴があると報告されている．また，特徴量を抽出した際にもそれらは機械学習により高い精度で分類されることが報告されている．このことから，畳み込みニューラルネットワークを用いる本手法でも，同じマルウェアファミリーに属するマルウェアから作成された 2 次元配列は類似性の高い特徴量が抽出され高い精度で分類可能であることが予想される．

深層学習では，学習の際にネットワークのパラメータを調整する．そして，学習後にはそのパラメータをモデルとして保存することが出来る．そしてその保存されたパラメータを読み込み，未知のデータを分類することが可能である．この際に，再学習が必要でないためスマートフォンなどの限られたリソースでも実行可能であると考えられる．端末上で実行可能であることから，学習済のパラメータを端末上に保存した後であればネットワーク接続不可の環境でも動作が可能であり，疑わしいファイルをクラウド環境に送信する必要もないため端末の通信トラフィックの増加を防ぐことが可能であると考えられる．

また，人が設計した特徴量抽出のための手法を使用する必要がないため, 生のデータを学習させること可能であり，より信頼性の高い分類が可能になると考えられる．

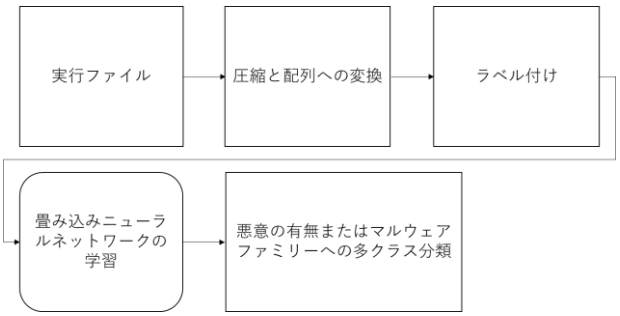

図 1　提案手法の概要図

4.1 ファイルの圧縮と配列への変換

提案手法の始めのステップであるファイルの圧縮と配列への変換について説明する．深層学習を行う際には，ネットワークに入力する前に画像やテキストを数値化し，それを数次元の配列に変換し，入力することで学習と分類を行う．本手法においては，実行ファイルを配列へと変換し入力する．

本提案手法では深層学習に畳み込みニューラルネットワークを用いるが，畳み込みニューラルネットワークにデータを入力する前に，データをそのまま入力するのではなく圧縮したデータを入力する．データ圧縮がなぜ必要なのかについて解説する．深層学習では入力されるデータのサイズが大きければ大きいほど学習にさかれるリソースが増えてしまい学習が困難となるため実行ファイルをそのままの形で入力することは現実的ではない．また，深層学習する際には入力される全てのサンプルのデータのサイズは揃っていなければならない．しかしながら，実行ファイルのサンプルのサイズは揃っていないため，何らかの方法により揃える必要がある．入力されるサンプルの最大のデータサイズに合わせて，サイズの小さいデータの末尾を 0 パディングすることで揃える場合には，データサイズが無駄に大きくなってしまう．そこで，データを先頭から途中まで切り同じ大きさに揃える方法を検討し実験を行った. 4.2 節にて解説される構成の畳み込みニューラルネットワークによりマルウェアと良性なソフトウェアの 2 値分類を行った．データセットは, MWS データセット 2016 に含まれる CCC データセット 2012[7]の 1800 個のマルウェアと Windows に標準インストールされているものや Download.com[8]からダウンロードされた 1800 個の良性なソフトウェアの合計 3600 個の実行ファイルをサンプルとして使用した．データを先頭から 1024Byte までに切ることでサイズを揃えた．サンプルの内, 2520 個を学習用に使い, 1080 個を評価用に使った．学習のエポック数は 12 回である．学習と評価を 10 回繰り返し，評価用のサンプルを使用して評価の値を取り最終的に平均値を算出した．この際に．学習用と評価用のサンプルは毎回ランダムに選別した．評価の結果を表 1 に示す．F 値が高いことから高い精度で分類できていると考えられる．しかしながら，F 値は 100%ではなく，誤分類されているものは長いコードのマルウェアと思われる．今回使用したサンプルでは，長いコードのマルウェアは非常に少なかったため影響がごく僅かではあったが，この分類手法を攻撃者が知っている場合には, 故意に 1024Byte 以上の間隔をあけて攻撃コードを作成することで，検出を逃れることが可能となってしまう．これらの問題を解決するために，提案手法によりデータを圧縮することで，悪意のある部分を残したままサイズを揃えることが可能になると考えられる．本提案手法では 3 つの圧縮手法を用いることを提案する．

表 1　10 回の評価の平均値

Accuracy	F-measure	Precision	Recall
0.998981481	0.998981481	0.998981481	0.998981481

4.1.1 Trigram と頻度分析による圧縮

はじめに, Trigram と頻度分析を用いることで実行ファイルのサイズを削減する手法について説明する．Unigram や Bigram では同じ組み合わせが多数存在し頻度分析がうまくいかないと予想される. また, 10-gram のような大きな組み合わせでは組み合わせ数が増えてしまい頻度分析の計算に時間がかかってしまう．また，比較する Byte 数が長ければ長いほど，その間に意味のない命令を挟んだ亜種を作成

しやすくなり，N-gram による検出を逃れることが可能となる。これらの理由を踏まえて提案手法では Trigram を採用した．図 2 に実行ファイルから Trigram を取得するまでの流れを示す．実行ファイルは 2 進数表現になっており，これを 2byte ずつに区切ることにより 16 進数表現に変換する．複数の良性なソフトウェアの実行ファイルと複数のマルウェアの実行ファイルの 16 進数表現から Trigram を取得し，その Trigram がどれくらいの頻度で実行ファイル中に登場するのか数える．そして，マルウェアと良性なソフトウェアの両方の実行ファイルの中に高頻度で登場する Trigram を実行ファイルから削除する．これにより悪意のない部分が実行ファイル中からなくなり，悪意のある情報を保ったまま全体のサイズが縮小されると考えられる．

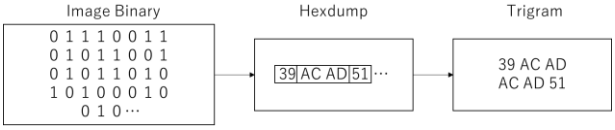

図 2　実行ファイルから Trigram を取得する流れ

4.1.2 ファイル圧縮方式による圧縮

圧縮前と展開・圧縮後のファイルが等しくデータの劣化を防ぎつつもファイルサイズを小さくするために，一般的には可逆圧縮が用いられる．本提案手法では，Julian により開発された bzip2 と呼ばれるファイル圧縮方式[9]を採用し，これにより実行ファイルを圧縮しファイルサイズを縮小する．

4.1.3 面積平均法による圧縮

画像ファイルのファイルサイズを小さくする際に，画像をリサイズし解像度を下げることによりそれを実現する方法があり，この方法として面積平均法（平均画素法）が広く用いられている．提案手法では，まず実行ファイルを後述の方法で 2 次元の配列に変換し，その配列に対して面積平均法を使って配列のサイズを小さくする．図 3 に面積平均法を用いたファイルサイズの削減の流れを示す．図 3 は縦横の長さが 4 の面積を縦横の長さが 3 の面積に縮小する方法を示している．はじめに，元の面積を縮小後の長さの 3 倍に拡大し，おわりに，拡大された面積を元の面積の長さの 4 で分割し平均を計算する．この手法により大きさの異なる複数の 2 次元配列を同じ大きさの 2 次元配列に変換することが可能である．

4.1.4 配列への変換

実行ファイルを畳み込みニューラルネットワークに入力するために配列に変換する．本手法では 2 次元配列に変換される．図 4 に変換の流れを示す．

まず，変換される実行ファイルが 2 進数表現のままである場合には，2Byte ずつに区切って 16 進数表現に変換する．これにより実行ファイルを 0 から 255 の連続した値にすることが出来る．次に，2 次元配列の幅（列数）と高さ（行数）の最大値を決定し，そのサイズに合わせて 2 次元配列を作成する．ファイルのサイズが小さいために幅や高さの最大値に届かない場合には，末尾に 0 を挿入する．

最後に，作成された 2 次元配列の全ての値を 255 で割り浮動小数点数を求める．深層学習の多くのモジュールでは float32 型の値が扱われる．

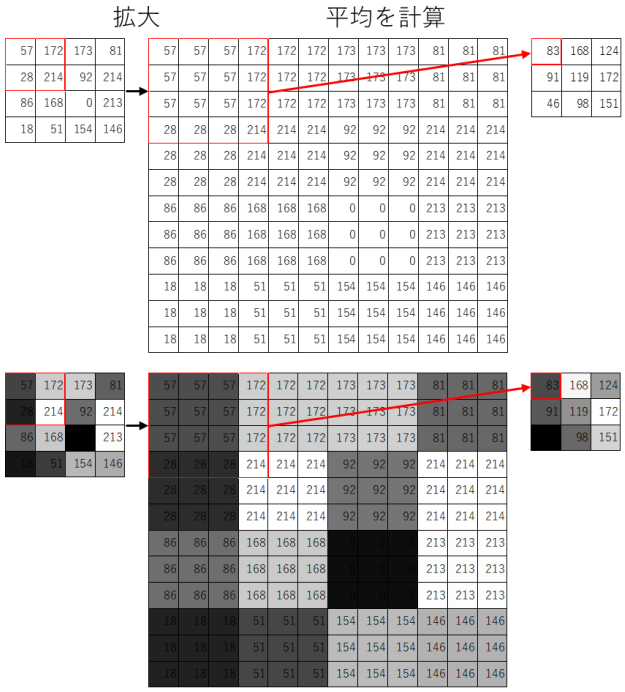

図 3　面積平均法による縮小

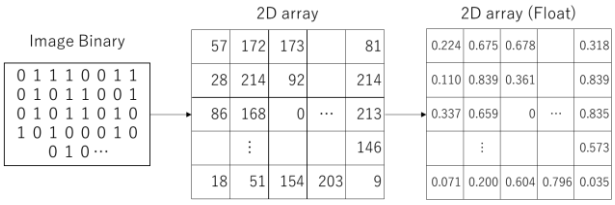

図 4　配列への変換

4.2 ラベル付けと深層学習

作成された配列の内で学習のために用いるデータに対してラベル付けを行う．マルウェアか良性なソフトウェアの 2 値に分類したい場合には，それぞれのラベルを学習データに付与する．マルウェアをマルウェアファミリーに多クラス分類したい場合には，マルウェアのデータにマルウェアファミリーのラベルを学習データに付与する．

次に，ラベル付けされた学習用のデータを畳み込みニューラルネットワークに学習させる．提案手法では 2 つの畳み込み層と 2 つのプーリング層と 2 つの全結合層の構成が

シンプルな畳み込みニューラルネットワークを用いる．活性化関数には ReLU を用いる．重みパラメータを更新するために Adam による最適化を行う．過学習の抑制のために全結合層の後に Dropout レイヤを使用する．図 5 に構成を示す．

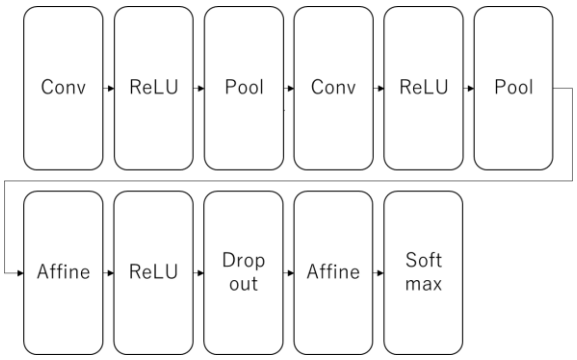

図 5　畳み込みニューラルネットワークの構成

4.3 学習済みのモデルによる未知のファイルの分類

畳み込みニューラルネットワークの学習が完了した後に，学習済みのモデルを使って未知のファイルを分類する．マルウェアと良性なソフトウェアの 2 値のラベルを付けて学習を行った畳み込みニューラルネットワークを使うことで，それぞれに分類することが出来る．マルウェアファミリーのラベルを付けて学習を行った畳み込みニューラルネットワークを使うことで，マルウェアをマルウェアファミリーに分類することが出来る．

端末上で分類を行いたい場合には，事前に高スペックなマシンで学習を完了させておき，学習済みのモデルを端末上に保存しておく．この学習済みのモデルを使うことでスマートフォンや市販のコンピュータなどの限られたリソース上でも分類が可能となり，ネットワーク接続のない環境でも亜種のマルウェアを自動的に検出可能となると考えられる．

5. 今後の課題とまとめ

本論文では，亜種マルウェアを自動検出するために，深層学習による分類を用いる手法を提案した．深層学習する前に入力される配列のサイズを巨大にならないようにするために，データの圧縮方法を提案した．しかしながら，データの圧縮が分類の精度に影響する可能性が考えられる．そのため，今後の課題として，データの圧縮が分類の精度にどのように影響するのかについて評価実験を行い明らかにしたい．

謝辞　本研究を行うにあたり，データセットの提供のために MWS 実行委員会の協力をいただきました．ここに感謝します．

参考文献

[1] “McAfee Labs 脅威レポート: 2017 年 6 月”. https://www.mcafee.com/jp/resources/misc/infographic-threats-report-jun-2017.pdf, (参照 2017-08-28).
[2] “McAfee Labs Quarterly Threat Report June 2017”. https://www.mcafee.com/us/resources/reports/rp-quarterly-threats-jun-2017.pdf, (参照 2017-08-28).
[3] A. Yokoyama, K. Ishii, R. Tanabe, Y. Papa, K. Yoshioka, T. Matsumoto, T. Kasama, D. Inoue, M. Brengel, M. Backes and C. Rossow. “SandPrint: Fingerprinting Malware Sandboxes to Provide Intelligence for Sandbox Evasion”. LNCS. 2016, vol. 9854, RAID 2016, p. 165-187.
[4] 斎藤康毅．“ゼロから作る Deep Learning ――Python で学ぶディープラーニングの理論と実装”．オライリージャパン．2016.
[5] L. Nataraj, S. Karthikeyan, G. Jacob and B. S. Manjunath. “Malware images: visualization and automatic classification”. ACM. 2011, VizSec ’11, No.4.
[6] L. Nataraj, V. Yegneswaran, P. Porras and J. Zhang. “A comparative assessment of malware classification using binary texture analysis and dynamic analysis”. ACM. 2011, AISec '11, p. 21-30.
[7] 高田雄太，寺田真敏，村上純一，笠間貴弘，吉岡克成，畑田充弘．“マルウェア対策のための研究用データセット～MWS Datasets 2016～”．情報処理学会，2016，2016-CSEC-74，pp 1-8.
[8] “Windows PC Software - Free Downloads and Reviews”. http://download.cnet.com/windows/, (参照 2017-08-28).
[9] “bzip2 : Home”. http://www.bzip.org/, (参照 2017-08-28).
[10] “Keras Documentation”. https://keras.io/ja/, (参照 2017-08-28).
[11] “Chainer: A flexible framework for neural networks”. https://chainer.org/, (参照 2017-08-28).
[12] “scikit-learn: machine learning in Python — scikit-learn 0.19.0 documentation”. http://scikit-learn.org/, (参照 2017-08-28).
[13] 新井悠，岩村誠，川古谷裕平，青木一史，星澤裕二．“アナライジング・マルウェア――フリーツールを使った感染事案対処”．オライリージャパン．2010.
[14] 岡谷貴之．“深層学習”．講談社．2015.