

機械学習とは何か？【Part 2】 畳み込みニューラル・ネットワーク のトレーニング

著者：Ole Dreessen、フィールド・アプリケーション担当スタッフ・エンジニア

概要

この連載では、AIを利用するシステムにおける機械学習（マシン・ラーニング）の重要性について説明しています。主なテーマとして、パターン認識や対象物の分類に使用される畳み込みニューラル・ネットワーク（CNN：Convolutional Neural Network）の特性やアプリケーションに焦点を絞っています。前回の記事「[機械学習とは何か？【Part 1】畳み込みニューラル・ネットワークの基本](#)」では、マイクロコントローラにおける従来の線形プログラミングとCNNの違い／優位性について説明しました。また、CIFARと呼ばれるCNNについての解説も行いました。これを使用すれば、猫、家、自転車など、画像に含まれる対象物の分類を行ったり、単純な音声のパターン認識を実施したりすることができます。今回（Part 2）は、そうしたアプリケーションを実現するためのCNNのトレーニング方法について解説します。

CNNのトレーニングのプロセス

Part 1 で取り上げたCIFARネットワークは、ニューロンから成る様々な層で構成されています。図1の例では、32ピクセル×32

ピクセルの画像データをネットワークの各層で処理していきます。CNNにおける最初のステップは、識別の対象となる物に固有の特徴や構造を検出することです。その検出にはフィルタ行列が使用されます。設計者がCIFARなどのニューラル・ネットワークのモデルを構築した時点では、フィルタ行列の詳細はまだ決まっています。つまり、その段階のネットワークでは、パターンや対象物を検出することはできないということです。

必要な処理を実現するためには、行列のすべてのパラメータと要素を決定します。その上で、対象物を検出する精度を最大化したり、損失関数を最小化したりする必要があります。このプロセスは、ニューラル・ネットワークのトレーニングと呼ばれています。Part 1 で紹介した一般的なアプリケーションでは、開発とテストの段階を通じてネットワークのトレーニングを1回実施していました。それにより、ネットワークは使用可能な状態になっており、それ以上パラメータを調整する必要はありませんでした。システムが既に把握済みの対象物の分類を行うためには、それ以上のトレーニングは必要ないということです。トレーニングが必要になるのは、システムにとって全く新たな対象物を分類する必要がある場合だけです。

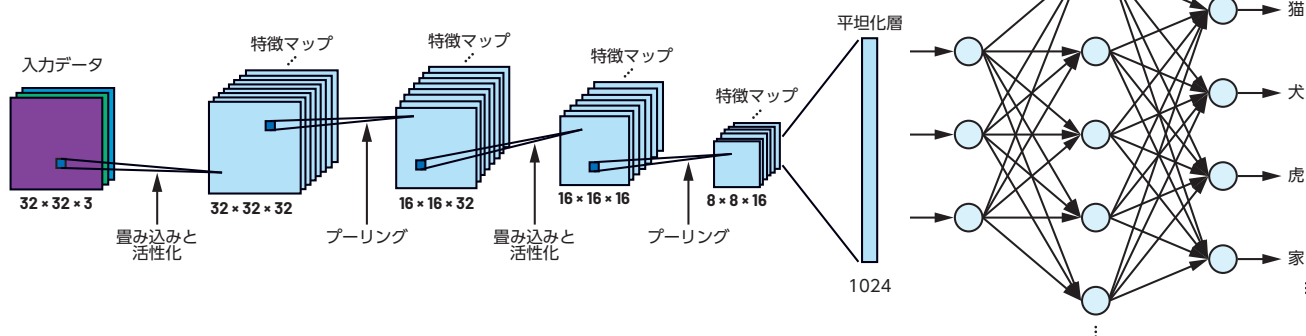


図1. CNN（CIFAR）のアーキテクチャ

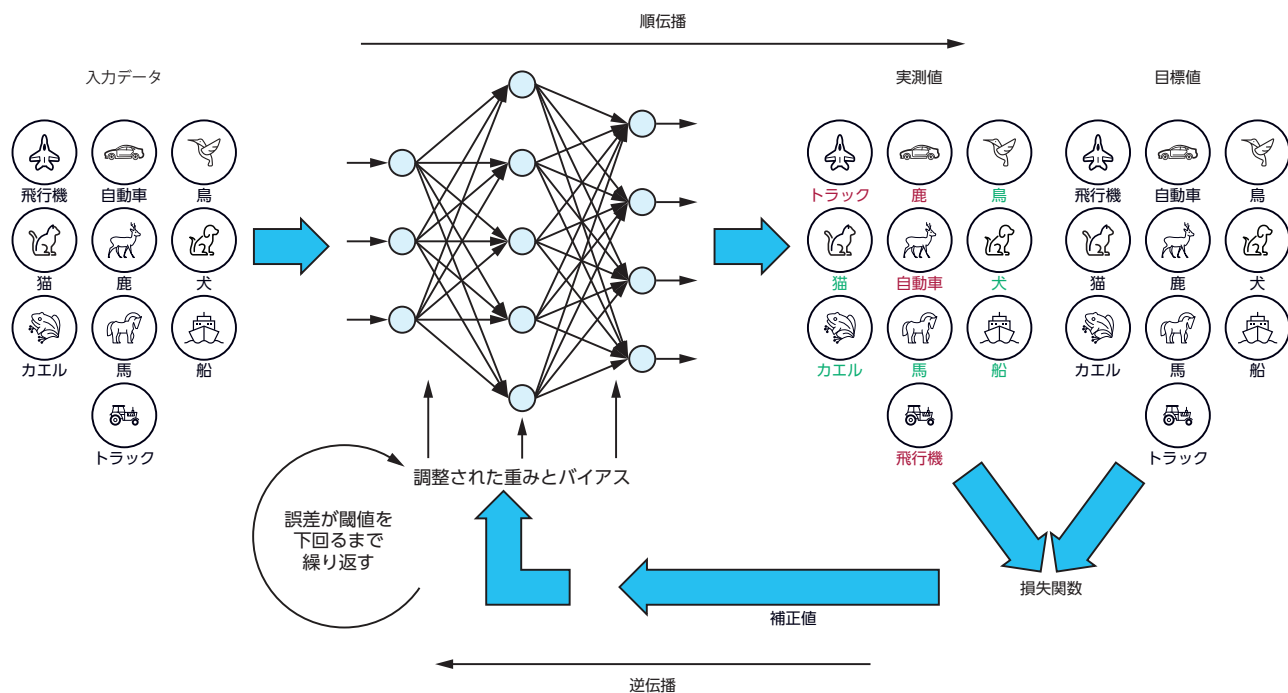


図2. 順伝播と逆伝播から成る
トレーニング用のループ

ネットワークのトレーニングを実施するには、そのためのデータが必要になります。また、トレーニングを実施した後は、同様のデータ・セットを使用してネットワークの精度をテストしなければなりません。例えば、CIFAR-10のデータ・セットには、10のクラス（飛行機、自動車、鳥、猫、鹿、犬、カエル、馬、船、トラック）に対応する画像のデータが含まれています。CNNのトレーニングを実施するには、事前にそれらの画像に名前を付ける必要があります。これは、AIをベースとするアプリケーションを開発する際、最も煩雑な作業だと言えるかもしれません。

本稿では、誤差逆伝播法（backpropagation：バックプロパゲーション）の原理に基づくトレーニングのプロセスについて説明します。ネットワークに多数の画像を読み取らせるのですが、その際には目標値も同時に与えます。本稿の例では、対象物のクラスに対応する値を目標値として与えることにします。そのようにすることで、画像を読み取らせるたびに、対象物のクラスに対応する目標値と実際の値（実測値）が一致するようにフィルタ行列が最適化されます。このプロセスが完了したら、ネットワークは、トレーニング中に読み取ったわけではない画像に含まれる対象物も検出できるようになります。

トレーニングの過不足

ニューラル・ネットワークのモデルを構築する際には、その複雑さをどの程度に設定すべきなのかという疑問が生じることがよくあります。つまり、層の数やフィルタ行列の大きさは、何を根拠にして決めればよいのかわからないということです。この疑問に答えるのは容易ではなく、トレーニングの過不足という概念について考えなければなりません。つまり、ネットワークの過学習と学習不足について考慮することが重要になります。過学習は、パラメータが多すぎてモデルが過度に複雑になった結果として生じます。これについては、トレーニング用のデータの損失とテスト用のデータの損失を比較する作業を実施するとよいでしょう。

それにより、トレーニング用のデータに対する予測モデルの適合度が低すぎるのか、高すぎるのかという判断が行えるようになります。例えば、トレーニング中の損失は少ないのに、ネットワークに読み取らせたことがないテスト用のデータを使用すると損失が過度に増大したとします。その場合、ネットワークはパターン認識の処理を一般化できていないだけでなく、トレーニング用のデータを記憶している状態にあるということが強く示唆されます。このような状態は、主にネットワークがパラメータの記憶域を持ちすぎているか、畳み込み層が多すぎる場合に発生します。対処策としては、ネットワークのサイズを小さくすることが必要になります。

損失関数とトレーニング用のアルゴリズム

ネットワークにおける学習は2つのステップで実施します（図2）。1つ目のステップでは、ネットワークに画像を読み取らせます。すると、それらの画像がニューロンのネットワークで処理され、出力ベクトルが生成されます。出力ベクトルの最大値は、検出した対象物のクラス（犬、猫など）を表します。トレーニングの段階では、必ずしも正しい結果が得られる必要はありません。このステップを順伝播（フィードフォワード：Feedforward）と呼びます。

出力で生じる目標値と実測値の差を損失と呼びます。つまり、[損失] = [目標値] - [実測値] です。また、この損失を表す関数が損失関数です。損失関数には、ネットワークのすべての要素とパラメータが含まれています。ニューラル・ネットワークの学習プロセスでは、損失関数が最小になるようにそれらのパラメータの値を決めることが目標になります。この最小化の処理は、出力で生じる偏差を、ネットワークのすべての構成要素を通してネットワークの開始層まで逆方向に送るプロセスによって達成することができます。この処理は、逆伝搬法（バックプロパゲーション：Backpropagation）と呼ばれています。

トレーニングのプロセスでは、フィルタ行列のパラメータをステップバイステップで決定していきます。このようにすることで1つのループが出来上がります。順伝播と逆伝播から成るプロセスは、損失の値があらかじめ決められた値を下回るまで繰り返されます。

最適化のアルゴリズム、勾配、勾配降下法

図3は、トレーニングのプロセスについて説明するためのものです。これは、2つのパラメータXとYだけで構成される損失関数を表しています。Z軸は損失に対応しています。この関数は説明のために用意したものであり、具体的な意味は持っていません。この3次元のプロットを見ると、この関数には全体的な最小値（大域的最小値）に加えて局所的な最小値（局所的最小値）が存在することがわかります。

重みとバイアスは、数値を最適化するための数多くのアルゴリズムを使用することで決定することができます。最も簡単なアルゴリズムは勾配降下法（Gradient Descent Method）です。勾配降下法は、次のような考え方に基づいています。すなわち、ランダムに選択した損失関数中の出発点から、勾配を使用したステップバイステップのプロセスによって大域的最小値に至る経路を見つけ出すというものです。ここでいう勾配とは、数学演算子として、ある物理量の変化量を表すものです。これは、損失関数の各点においてベクトルを与えます。このベクトルは勾配ベクトルと呼ばれており、関数値の変化が最大になる方向を表します。また、このベクトルの大きさは変化量に対応しています。図3の関数において、勾配ベクトルは右下のどこかの点（赤い矢印）において最小値の方向を指し示しています。表面が平坦であることから、ベクトルの値（大きさ）は小さくなります。この状況は、遠くにあるピークの近くでは異なったものになります。図3の例では、ベクトル（緑の矢印）は急勾配で下の方向を指し示しています。また、起伏が大きいことから値も大きくなります。

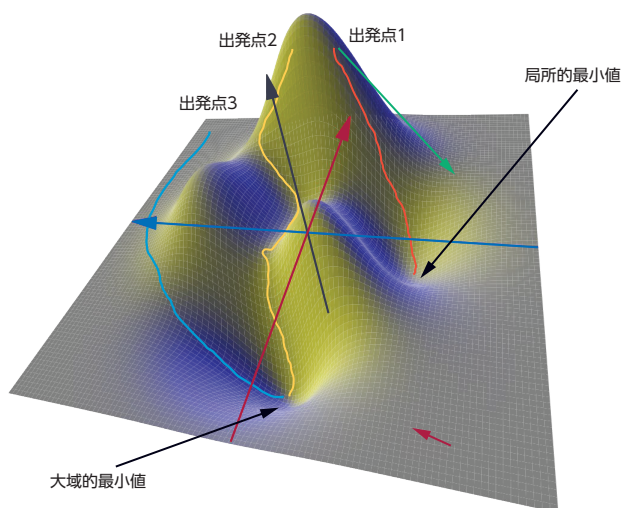


図3. 勾配降下法における目標までの様々な経路

勾配降下法において、降下が最も急勾配の谷に至る経路は、任意に選んだ点から出発して反復的に求めることができます。つまり、最適化のアルゴリズムによって出発点の勾配を計算し、最も急降下する方向へ少し前進するといった具合です。その中間点で勾配を再計算して、谷への経路を引き続き進みます。このようにすることで、出発点から谷の中の点までの経路が形成されます。ここで問題になるのは、出発点はあらかじめ定められているわけではなく、ランダムに選択しなければならないという点です。注意深い方であれば、図3の2次元マップの場合、関数プロットの左側のどこかに出発点を置くことを考えるでしょう。そうすれば、（例えば青色の）経路の終点は確実に大域的最小値になります。他の2つの経路（黄色とオレンジ色）ははるかに長くなるか、または終点が局所的最小値になってしまいます。実用的な最適化のアルゴリズムでは、2つのパラメータだけでなく、何十万ものパラメータを最適化する必要があります。そのため、出発点を正しく選択できるか否かは偶然にすぎないということにすぐに気づきます。この手法は、実際には役に立たないように感じられるかもしれませんが。なぜなら、選択する出発点によっては、経路（つまりトレーニングの時間）が非常に長くなってしまいうからです。また、目標点が大域的最小値にならない可能性もあります。その場合、ネットワークの精度が低下してしまいます。

最適化に向けては、上記の2つの問題を解決するために、過去数年間にわたって数多くのアルゴリズムが開発されました。勾配降下法に代わる手法の例としては、確率勾配降下法、モーメンタム法、AdaGrad、RMSProp、Adamなどが挙げられます。各アルゴリズムにはそれぞれ長所と短所があります。したがって、実際にどのアルゴリズムを使用するのは、ネットワークの開発者が決定することになります。

トレーニング用のデータ

先述したように、トレーニングのプロセスでは、自動車や船といった正しいクラスの情報を付加した画像をネットワークに読み取らせませす。本稿の例では、一般的に使われているCIFAR-10のデータ・セットを使用しました。もちろん、実際のアプリケーションでは、猫、犬、自動車などが対象になるとは限りません。AIによって認識すべきものはそれらとは異なる可能性があります。例えば、製造工程でネジの品質を検出するアプリケーションを開発する必要があったとします。その場合、ネットワークのトレーニングはネジの良品と不良品のデータを使用して実施しなければなりません。そのようなデータ・セットの作成には、非常に多くの手間と時間がかかる可能性があります。多くの場合、このステップはAIをベースとするアプリケーションの開発において最もコストのかかる作業になるでしょう。データ・セットを用意できたら、それらをトレーニング用のデータとテスト用のデータに分けます。トレーニング用のデータは、先述したような形で使用します。一方、テスト用のデータは、開発プロセスの最後に、トレーニング済みのネットワークの機能を確認するために使用します。

まとめ

今回は、CNNの基本、設計、機能について詳しく説明しました。今回説明した内容を踏まえれば、機能を実現するために必要な重みとバイアスを決定することも可能になります。その結果、本稿で例にとったネットワークは適切に動作すると想定できます。次回 (Part 3) は、このニューラル・ネットワークをハードウェア

によって実現し、猫を認識するテストを実施する例を示します。そのハードウェアとしては、アナログ・デバイセズが開発した「MAX78000」を使用する予定です。同製品は、ハードウェア・ベースのCNN用アクセラレータを搭載したAI専用のマイクロコントローラです。



著者について

Ole Dreessenは、アナログ・デバイセズでフィールド・アプリケーションを担当するスタッフ・エンジニアです。2014年に入社しました。それ以前は、Avnet Memec、Macnicaで通信技術や高性能のマイクロプロセッサのサポート業務に従事。マイクロコントローラとセキュリティに関する広範な専門知識を有しています。カンファレンスやイベントなどにおけるプレゼンテーションの経験も豊富です。余暇にはChaos Computer Clubの熱心なメンバーとして活動。リバース・エンジニアリングや組み込みセキュリティなどに関する取り組みを行っています。



想像を超える可能性を
AHEAD OF WHAT'S POSSIBLE™

アナログ・デバイセズ株式会社

お住いの地域の本社、販売代理店などの情報は、analog.com/jp/contact をご覧ください。

オンラインサポートコミュニティEngineerZoneでは、アナログ・デバイセズのエキスパートへの質問、FAQの閲覧ができます。

©2023 Analog Devices, Inc. All rights reserved.
本紙記載の商標および登録商標は、各社の所有に属します。
Ahead of What's Possibleはアナログ・デバイセズの商標です。

AD5703-0-3/23

VISIT [ANALOG.COM/JP](https://analog.com/jp)