Vol 57, No 1—February 2023

ANALOG DEVICES
想像を超える可能性を
AHEAD OF WHAT'S POSSIBLE™

AnalogDialogue

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

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

AI（人工知能）の分野では技術の急速な進化が続いています。実際、AIを利用することによって、従来は実現不可能であったり、実装が非常に困難であったりしたアプリケーションが具現化されるようになりました。本連載では、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ます。特に、畳み込みニューラル・ネットワーク（CNN：Convolutional Neural Network）について詳しく解説することにします。CNNは、複雑なデータの中から特徴を抽出するために利用される強力なツールです。例えば、音声信号や画像を対象とした複雑なパターン認識などを実現できます。今回（Part 1）は、従来の線形計画法との比較を交えながらCNNの基本について詳しく解説します。なお、Part 2ではCNNのモデルのトレーニング方法について説明します。Part 3では、AI専用のマイクロコントローラを使用して、モデルのテストを行う方法について検討します。

CNNとは何か？

ニューラル・ネットワークとは、ニューロンから成る人間の神経回路網を模したシステム／構造のことです。AIがデータについてより深く理解し、複雑な問題を解決できるようにするために使用されます。ニューラル・ネットワークにはいくつもの種類があります。それらのうち、本連載ではCNNに焦点を絞ることにします。CNNは、ディープ・ラーニングで使用される人工的なニューラル・ネットワークの一種です。その主な用途としては、入力データに含まれる対象物のパターンの認識、分類が挙げられます。CNNは、入力層、いくつかの畳み込み層、出力層で構成されます。なかでも、最も重要なのは畳み込み層です。この層では、固有の重み付けとフィルタが使用されます。それによって、入力データから特徴を抽出できるようにします。取り扱うデータの種類は、画像、音声、テキストなど様々です。CNNでは特徴を抽出するプロセスをベースとして、データに含まれるパターンを特定します。その結果、より効果的で効率の高いアプリケーションを構築することが可能になります。

従来の制御工学における線形計画法

CNNについて深く理解するために、まずは線形計画法に触れておくことにします。従来の制御工学では、次のようなことが行われます。まず、1つまたは複数のセンサーによってデータを取得します。次に、それらのデータに処理を適用し、ルールに従って応答を返します。その上で、結果を表示したり転送したりといったことが実行されます。例として、温度を制御する装置を考えます。その場合、温度センサーから出力されるデータをマイクロコントローラ・ユニット（MCU）によって取得します。例えば、秒単位で温度を把握するといった具合です。センサーから取得したデータは、閉ループ制御システムの入力となります。そのループでは、設定温度との比較が行われます。これは、MCUによる線形実行（linear execution）の一例です。この手法では、事前に設定された値と測定した値に基づいて最終的な結果を得ます。それに対し、AIを利用するシステムでは、確率が重要な役割を果たします。

複雑なパターンと信号処理

アプリケーションの中には、パターン認識を担うシステムによって入力データを解釈し、その結果に基づいて動作するというものが数多く存在します。パターン認識は様々なデータ構造に適用することができますが、本稿では1次元と2次元のデータ構造に限定して話を進めることにします。1次元のデータの一般的な例としては、音声信号、心電図（ECG）、光電式容積脈波（PPG）、振動などが挙げられます。2次元のデータとしては、一般的な画像、熱画像、ウォーターフォール図などがあります。

上記のようなデータに対するパターン認識は、どのようにすれば実現できるでしょうか。実は、MCUで実行される従来のコードによって必要な変換処理を行うのは非常に困難です。例として、画像に含まれる対象物を認識したいケースを考えます。ここでは、猫を対象物にすることにしましょう。その場合、画像が記録されたタイミングが問題になることはありません。過去に記録されたものでも、カメラ（センサー）で取得されたばかりのものでも、特に違いはありません。いずれにせよ、解析用のソフトウェアは、とがった耳、三角形の鼻、ひげなど、猫において特徴的であると考えられるパターンをルールに基づいて検索します。画像においてこれらの特徴が認識されたら、ソフトウェアは猫が見つかったという情報を報告します。ただ、このプロセスについてはいくつかの疑問が生じます。例えば、対象とする画像に猫の後ろ姿が含まれていた場合、パターン認識用のシステムはどのような判断を下すのでしょうか。あるいは、ひげがなかったり、事故で脚を失ったりしている場合にはどうなるのでしょうか。このようなことが起きる頻度は低そうですが、パターン認識用のコードは、発生し得るあらゆるケースを網羅した大量のルールを用いて検索を実施する必要があります。上記の例のようなシンプルなパターン認識においても、ソフトウェアで使用すべきルールは非常に大規模なものになってしまいます。

機械学習で従来のルールを置き換える

AIによる機械学習の背景には、規模を抑えつつ人間の学習方法を模倣するという考え方が存在します。if、thenから成る大量のルールを定式化するのではなく、パターン認識用の汎用マシンのモデルを構築するということです。2つの手法の主な違いは次の点にあります。すなわち、一連のルールを使用する方法とは対照的に、AIでは明確な結果を出しません。つまり、「画像内に猫が存在することを認識しました」という報告は行われないということです。そうではなく、「画像に猫が写っている確率は97.5%です。ヒョウ（2.1%）または虎（0.4%）である可能性もあります」という結果をリポートします。つまり、実際のアプリケーションでは、パターン認識のプロセスの最後に別途判断を実施する必要があります。そのためには閾値を用いた判定が行われます。

もう1つの違いとしては、AIで使われるパターン認識用のマシンは固定化されたルールを備えていないということが挙げられます。その代わりに、パターン認識用のマシンのトレーニングを実施します。この学習プロセスでは、ニューラル・ネットワークに対して猫の画像を大量に提示します。その結果として、同ネットワークは最終的には画像内に猫がいるか否かを独力で認識できるようになります。重要なのは、その後に行われる認識が、トレーニングの際に既に見知った画像に限定されることなく行われるという点です。なお、この種のニューラル・ネットワークは、MCU内にマッピングされる必要があります。

パターン認識用のマシン内で行われる処理

先述したように、ニューロン・ネットワークは、人間の脳内の神経回路網を模して構築されています。ニューロン・ネットワークの各ニューロンは、複数の入力と1つの出力を備えています。それらのニューロンは、通常は入力部の線形変換とそれに続く非線形関数を備えています。前者の線形変換では、入力に対してある数（重みw）を掛け、定数（バイアスb）を加算するという演算処理が行われます。後者の非線形関数は活性化関数（Activation Function）とも呼ばれ、固定化された処理を実行します[1]。この関数は、ネットワークにおける唯一の非線形成分として、人工のニューロンが発火（fire）する値の範囲を定義する役割を果たします。ニューロンの機能は、数学的には次式のように表すことができます。

$$[出力] = f(w*x+b) \quad (1)$$

ここで、fは活性化関数、wは重み、xは入力データ、bはバイアスです。なお、データとしては、個々のスカラーやベクトル、あるいは行列形式のものが使われる可能性があります。図1に示したニューロンは、3つの入力とReLU[2]と呼ばれる活性化関数を備えています。ネットワーク内の各ニューロンは、必ずいずれかの層に配置されます。

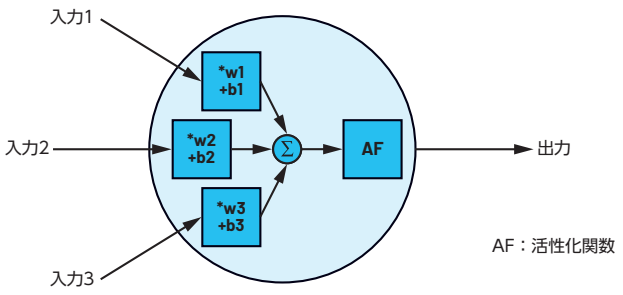

図1. 3つの入力と
1つの出力を備えるニューロン

[1] シグモイド関数、tanh関数、ReLU関数などがよく使用されます。

[2] ReLUは、Rectified Linear Unit（正規化線形ユニット）の略です。この関数に負の値を入力すると出力はゼロになります。入力値がゼロよりも大きい場合には入力値と同じ値が出力されます。

先述したように、CNNは入力データに含まれる対象物のパターン認識と分類に使用されます。また、CNNは、1つの入力層、いくつかの隠れ層（hidden layer）、1つの出力層から成ります。ここで、図2に示した小規模のニューラル・ネットワークをご覧ください。このネットワークは、3つの入力を備える入力層、5つのニューロンから成る1つの隠れ層、4つの出力を備える1つの出力層によって構成されています。入力層、隠れ層の各ニューロンの出力は、いずれも次の層のすべての入力に接続されています。このような小規模なネットワークであっても、それについて表現するための式には、32のバイアスと32の重みが含まれることになります。なお、このネットワークは説明のために用意したものです。これによって意味のあるタスクを処理することはできません。

CNNの一種に、CIFARというものがあります。このニューラル・ネットワークは、画像認識のタスクで広く使用されています。CIFARは、主に畳み込み層とプーリング層という2種類の層で構成されます。これらの層は、どちらもニューラル・ネットワークのトレーニングで効果的に活用されます。畳み込み層では、畳み込みという数学演算を使用することにより、ピクセル値の配列内に含まれるパターンを識別します。図3に示すように、畳み込みは隠れ層で行われます。このプロセスは、必要なレベルの精度が得られるまで何度も繰り返されます。なお、畳み込み演算の出力値は、比較する2つの入力値（この場合は画像とフィルタ）が近い場合には必ず非常に大きくなります。この層は、フィルタ行列、フィルタ・カーネル、または単にフィルタと呼ばれます。そして、この畳み込み層の演算結果はプーリング層に引き渡されます。プーリング層は、特徴マップ（feature map）を生成する役割を果たします。特徴マップとは、入力データの重要な特徴を表現したものです。これは、もう1つのフィルタ行列だと考えることができます。トレーニングを実施後のネットワークが運用される際には、それらの特徴マップが入力データと比較されます。特徴マップは対象物のクラス固有の特徴を保持しており、それが入力画像と比較されます。ニューロンの出力は、内容がよく似ている場合だけトリガされます。CIFARでは、これら2つの手法を組み合わせることで、画像内の様々な対象物を高い精度で認識／分類します。

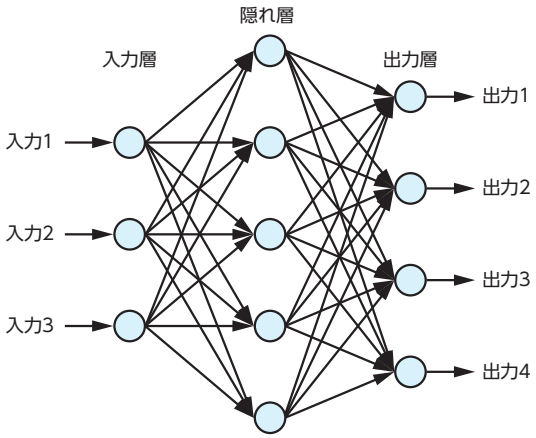

図2. 小規模のニューラル・ネットワーク

CIFARのトレーニングでよく使用されるデータ・セットに、CIFAR-10というものがあります。これには、32×32ピクセルのカラー画像が6万枚含まれています。各カラー画像は、ウェブ・ページ、ニュースグループ、個人の画像コレクションといった様々なソースから収集されたものです。CIFAR-10は、10のクラスから成ります。各クラスには6000枚の画像が含まれており、トレーニング用、テスト用、検証用のセットとして均等に分割されています。このような構成であることから、コンピュータ・ビジョンの新たなアーキテクチャやその他の機械学習モデルのテストに最適です。

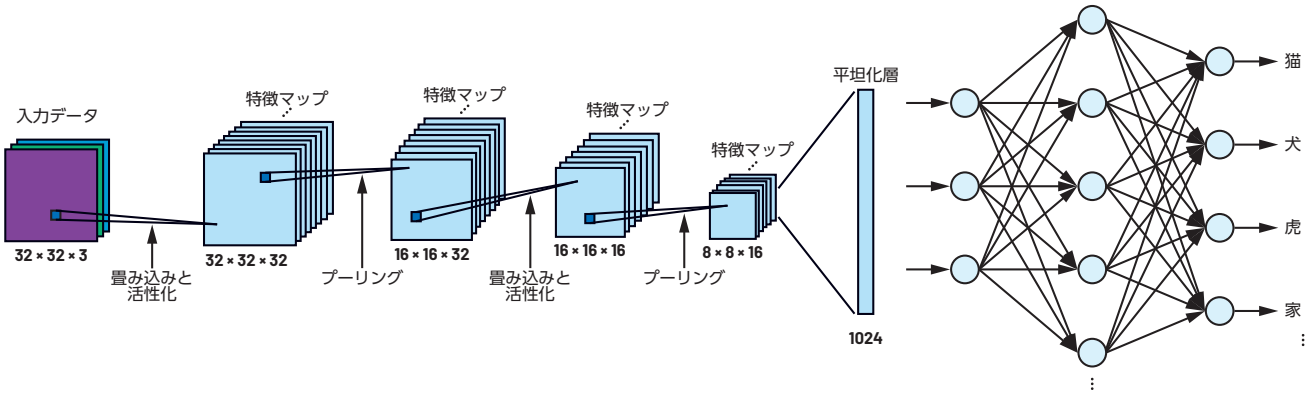

図3. CIFARのモデル。
CIFAR-10のデータ・セットでトレーニングされる場合の例を示しています。

CNNと他のニューラル・ネットワークの主な違いとしては、データの処理方法が挙げられます。CNNでは、入力データが各種の特性を備えているかどうか、フィルタリングによって順次精査されます。直列に接続された畳み込み層の数が多くなるほど、認識できる詳細度は向上します。このプロセスは、最初の畳み込みの後、エッジや点など対象物の単純な性質をターゲットとして開始されます。2回目の畳み込みが完了したら、角、円形、矩形など詳細な構造へとターゲットが変更されます。3回目の畳み込みが終わった後に得られた特徴は、画像内の対象物に似た複雑なパターンを表します。また、得られた特徴は、通常は対象物のクラスに固有のものとなります。これは、冒頭に挙げた例で言えば猫のひげや耳に相当します。図4では特徴マップを可視化していますが、これはアプリケーション自体に必要なものではありません。ただ、畳み込みについて理解するのには役に立つでしょう。

CIFARのような小規模のネットワークでも、各層には数百ものニューロンが存在します。そして多くの層が直列に接続されています。必要な重みとバイアスの数は、ネットワークの複雑さとサイズが大きくなると急増します。図3に示したCIFAR-10の例では、既に20万個のパラメータが存在することになります。また、それらのパラメータには、トレーニングのプロセス中に決定された値のセットが必要になります。特徴マップをプーリング層で更に処理することにより、重要な情報を維持したまま、トレーニングを必要とするパラメータの数を減らすことができます。

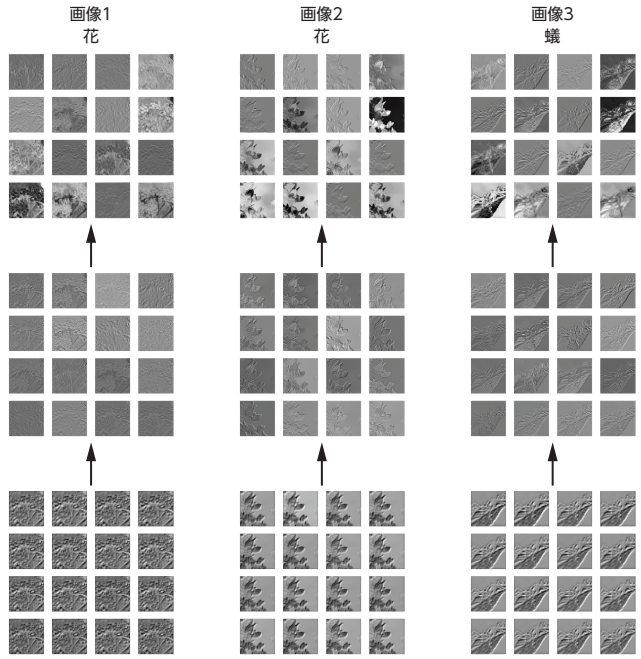

図4. CNNの特徴マップ

先述したように、CNNで各畳み込みが実施された後は、多くの場合、プーリング（文献によってはサブサンプリングと呼ばれることもあります）が行われます。この処理は、データの次元を減らすことに寄与します。図4の特徴マップを見ると、大きな領域には意味のある情報がほとんど含まれていないか、全く含まれていないことがわかります。なぜなら、対象物は画像全体を構成しているのではなく、そのごく一部を構成しているにすぎないからです。画像の残りの部分は、この特徴マップでは使用されないので、分類には関与しません。プーリング層については、プーリングの種類（最大値プーリングまたは平均値プーリング）とウィンドウ行列のサイズが指定されています。ウィンドウ行列は、プーリングのプロセスにおいて、入力データに対してステップ的に機能することになります。例えば、最大値プーリングでは、ウィンドウ内のデータの最大値が取得されます。それ以外の値はすべて破棄されます。このようにして、データの数は連続的に減少していきます。最終的には、畳み込みの処理に伴って、各対象物のクラスに固有の性質が抽出されます。

但し、畳み込みとプーリングの処理を繰り返した結果は、多数の2次元行列になります。分類という目標を達成するためには、2次元のデータを長い1次元のベクトルに変換する必要があります。この変換処理は、平坦化層と呼ばれる層で実行されます。この層の下流には、1つまたは2つの全結合層が配置されます。最後の2つの層に含まれるニューロンは、図2に示したのと同様の構造を成しています。本稿で例にとったニューラル・ネットワークの場合、最後の層は区分すべきクラスの数と同じ数の出力を備えています。また、最後の層では、データの正規化も行われます。その際に、猫が97.5%、ヒョウが2.1%、虎が0.4%といった確率分布が生成されます。

以上で、本稿で例にとったニューラル・ネットワークのモデルを俯瞰できたことになります。但し、カーネル行列とフィルタ行列の重みと内容についてはまだ触れていません。これらについては、モデルを機能させるために実施するトレーニングを通じて決定する必要があります。これについては、Part 2（次回）で説明します。それに続くPart 3では、ニューラル・ネットワーク（猫の認識に使用する例）を実現するためのハードウェアについて説明します。そのハートウェアにおいては、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。

©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。

AD5702-0-2/23