Deep Learning with Python

備考

著者

Francois Chollet

掲載

F. Chollet, ``Deep Learning with Python'', 1st ed, Manning Publications Co., Greenwich, CT, USA, 2018.

5.2. Training a convnet from scratch on a small dataset

少ないデータで画像分類モデルを訓練しなければならないというのは一般的な状況であり、専門的なコンテクストでコンピュータビジョンを行う場合には、実際に遭遇する可能性が高いでしょう。 「少ない」サンプルとは、数百から数万枚の画像を意味します。 実際の例として、4,000枚の猫と犬の画像（2,000枚の猫と2,000枚の犬）を含むデータセットの中で、画像を犬と猫に分類することに焦点を当ててみましょう。 トレーニングに2,000枚、検証に1,000枚、テストに1,000枚の画像を使用します。

このセクションでは、この問題に取り組むための基本的な戦略をレビューします。 達成可能なベースラインを設定するために、正則化を行わずに、2,000個の訓練サンプルで小さなConvNetを素朴に訓練することから始めます。これにより、71%の分類精度が得られます。**この時点で、主な問題はオーバーフィットです。** 続いて、コンピュータビジョンにおけるオーバーフィッティングを軽減するための強力な手法であるデータオーグメンテーションを紹介します。 データオーグメンテーションを使うことで、ネットワークを改善して82%の精度を得ることができます。

次のセクションでは、**小さなデータセットにディープラーニングを適用するための2つの重要なテクニックをレビューします**：事前学習されたネットワークを使用した**特徴抽出**（これにより、90%から96%の精度が得られます）と事前学習されたネットワークの**微調整**（これにより、最終的な精度は97%になります）です。これら3つの戦略、すなわち、小さなモデルをゼロから訓練すること、事前訓練されたモデルを使って特徴抽出を行うこと、事前訓練されたモデルを微調整することは、小さなデータセットを使って画像分類を行う問題に取り組むための将来のツールボックスを構成します。

5.2.1. The relevance of deep learning for small-data problems

ディープラーニングは、多くのデータが利用可能な場合にのみ機能するということを時々耳にするでしょう。**ディープラーニングの基本的な特徴の1つは、手動の特徴工学を必要とせずに、学習データの中から興味深い特徴を見つけることができるということです。** これは、画像のように入力サンプルが非常に高次元である問題に特に当てはまります。

しかし、何が多くのサンプルを構成するかは、まず、訓練しようとしているネットワークの大きさと深さに関係しています。複雑な問題を数十個のサンプルで解くために ConvNet を訓練することはできませんが、モ

デルが小さく正則化されていてタスクが単純な場合は、数百個で十分な可能性があります。ConvNet は局所的な翻訳不変の特徴を学習するため、知覚問題のデータ効率が非常に高い。非常に小さな画像データセットでゼロから ConvNet を学習すると、データが比較的不足しているにもかかわらず、カスタムの特徴量を設計する必要がなく、妥当な結果を得ることができます。このセクションでは、この方法を実際に見てみましょう。

例えば、大規模なデータセットで学習した画像分類モデルや音声対テキストモデルを、わずかな変更を加えるだけで、大きく異なる問題に再利用することができます。具体的には、コンピュータビジョンの場合、多くの事前学習済みモデル（通常はImage-Netデータセットで学習したもの）が公開されており、ダウンロードして使用することができます。これを次のセクションで説明します。まずはデータを手に入れることから始めましょう。

5.2.3. Building your network

前の例ではMNISTのために小さな ConvNet を作ったので、そのような ConvNet に慣れているはずです。一般的な構造は同じものを再利用します： ConvNet は Conv2D（reluアクティベーション付き）と MaxPooling2D を交互に重ねたスタックになります。

5.3. Using a pretraind convnet

小規模な画像データセットでの深層学習のための一般的で非常に効果的なアプローチは、事前訓練されたネットワークを使用することです。**事前学習済みネットワークとは、大規模なデータセット、通常は大規模な画像分類タスクで事前に学習された保存されたネットワークのことである。** この元のデータセットが十分に大規模で一般的なものであれば、事前学習されたネットワークによって学習された特徴の空間階層は、視覚世界の一般的なモデルとして効果的に機能することができ、したがって、その特徴は、新しい問題が元のタスクとは全く異なるクラスを含んでいたとしても、多くの異なるコンピュータビジョンの問題に有用であることを証明することができる。 例えば、ImageNet（クラスの多くは動物と日常的な物）でネットワークを訓練し、この訓練されたネットワークを画像の中の家具のアイテムを識別するような遠隔地のものに再利用することができるかもしれない。このように、学習した特徴を異なる問題にまたがって移植できることは、多くの古い浅い学習アプローチと比較して、ディープラーニングの主な利点であり、小さなデータの問題に非常に効果的になります。

ここでは、ImageNetデータセット（140万枚のラベル付き画像と1,000種類のクラス）で学習された大規模なコンボネットを考えてみましょう。ImageNetには、異なる種類の猫や犬を含む多くの動物クラスが含まれているため、犬と猫の分類問題では十分な性能を発揮することが期待できます。

ここでは、2014年にKaren SimonyanとAndrew Zissermanによって開発されたVGG16アーキテクチャを使用します。このモデルは古いモデルであり、最新の技術とは程遠く、他の多くの最近のモデルよりもやや重くなっていますが、私がこのモデルを選んだのは、そのアーキテクチャがすでに皆さんがよく知っているものに似ていることと、新しい概念を導入することなく理解しやすいからです。 これは、VGG, ResNet, Inception, Inception-ResNet, Xceptionなどのキュートなモデル名に初めて出会うかもしれませんが、コンピュータビジョンのためのディープラーニングを続けていると、これらのモデルが頻繁に出てくるので、慣れるでしょう。

事前学習されたネットワークを使用するには、特徴抽出と微調整の2つの方法があります。この2つの方法について説明します。まずは特徴抽出から始めましょう。

5.3.1. Feature extractoin (特徴抽出)

特徴抽出は、前のネットワークで学習した表現を使用して、新しいサンプルから興味深い特徴を抽出します。これらの特徴は、新しい分類器を通して実行され、ゼロから訓練されます。

前に見たように、画像分類に使用される **ConvNet は2つの部分から構成されています：プーリングとコンボリューションの一連の層から始まり、密に接続された分類器で終わります。** 最初の部分はモデルの畳み込みベースと呼ばれています。ConvNet の場合、特徴抽出は、以前に訓練されたネットワークの畳み込みベースを利用して、新しいデータを実行し、その出力の上に新しい分類器を訓練することから成り立っています（図5.14参照）。

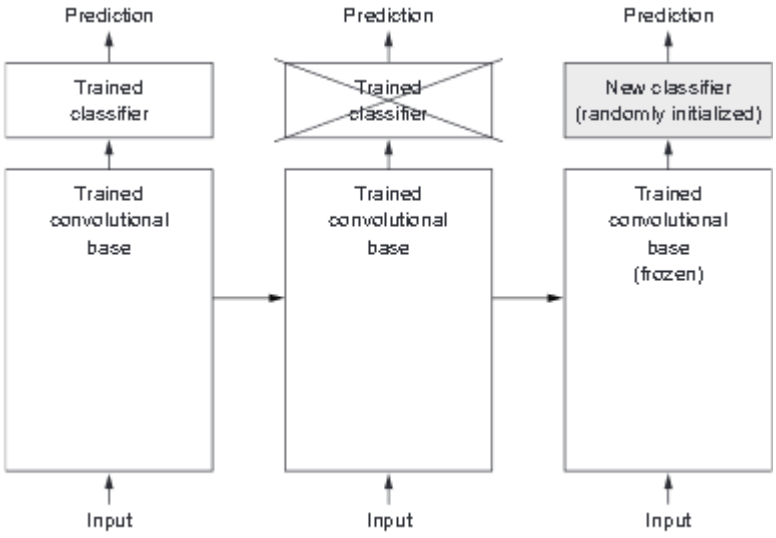

Figure 5.14 Swapping classifiers while keeping the same convolutional base

なぜ畳み込みベースだけを再利用するのか？密に接続された分類器を再利用することはできないのでしょうか？一般的には，そのようなことは避けるべきです．その理由は，**畳み込みベースによって学習された表現は，より汎用的である可能性が高いので，再利用可能性が高いからです．** しかし、**分類器によって学習された表現は、必然的にモデルが学習されたクラスのセットに固有のものとなります。 さらに、密に接続された層で見つかった表現は、入力画像の中でオブジェクトがどこにあるかに関する情報を一切含みません**：これらの層は空間の概念を取り除きますが、オブジェクトの位置は畳み込み特徴マップによって記述されます。**オブジェクトの位置が重要な問題では、密に接続された特徴量はほとんど意味がありません。**

特定の畳み込みレイヤーによって抽出される表現の一般性（したがって再利用可能性）のレベルは、モデル内のレイヤーの深さに依存することに注意してください。 モデルの中で初期のレイヤーは局所的で汎用性の高い特徴マップ（視覚的なエッジ、色、テクスチャなど）を抽出し、より上位のレイヤーはより抽象的な概念（「猫の耳」や「犬の目」など）を抽出します。そのため、新しいデータセットが元のモデルが訓練されたデータセットと大きく異なる場合は、畳み込みベース全体を使うよりも、モデルの最初の数層だけを使って特徴抽出を行った方が良いかもしれません。

この場合，ImageNet のクラスセットには複数の犬と猫のクラスが含まれているため，元のモデルの密に接続された層に含まれる情報を再利用することが有益である可能性が高い．しかし，新しい問題のクラス集合が元のモデルのクラス集合と重ならないというより一般的なケースをカバーするために，ここでは再利用しないことにします．ImageNet上で訓練されたVGG16ネットワークの畳み込みベースを使って、猫と犬の画像から興味深い特徴を抽出し、これらの特徴の上に犬と猫の分類器を訓練することで、これを実践してみましょう。

5.3.2. Fine-tuning

特徴抽出を補完するモデル再利用のために広く使われているもう一つの手法は，微調整である（図5.19参照）．微調整は，特徴抽出に使われた凍結モデルの最上位層のいくつかを凍結解除し，モデルの新たに追加された部分（この場合は完全に接続された分類器）とこれらの最上位層の両方を共同で学習することで構成されています． これは，再利用されるモデルのより抽象的な表現をわずかに調整して，目の前の問題により関連性のあるものにするために，微調整と呼ばれます．

先ほど，ランダムに初期化された分類器を訓練するためには，VGG16の畳み込みベースをフリーズさせる必要があると述べましたが，同様の理由で，畳み込みベースの最上層を微調整することはできません．同じ理由で、畳み込みベースの最上層を微調整することができるのは、最上層の分類器が既に訓練されている場合のみです。 分類器がまだ訓練されていない場合，訓練中にネットワークを伝搬する誤差信号が大きすぎて，微調整されているレイヤーによって以前に学習された表現が破壊されてしまいます． したがって，ネットワークを微調整するためのステップは以下の通りです．

1. 既に学習済みのベース・ネットワークの上にカスタム・ネットワークを追加します。
2. ベース・ネットワークをフリーズさせます。
3. 追加した部分を訓練します。
4. ベースネットワーク内のいくつかのレイヤーをアンフリーズします。
5. これらのレイヤーと追加したパーツの両方を共同で訓練します。

特徴抽出を行う際の最初の3つのステップはすでに完了しています。ステップ4に進みましょう: conv_baseを解凍し、その中の個々のレイヤーを凍結します。