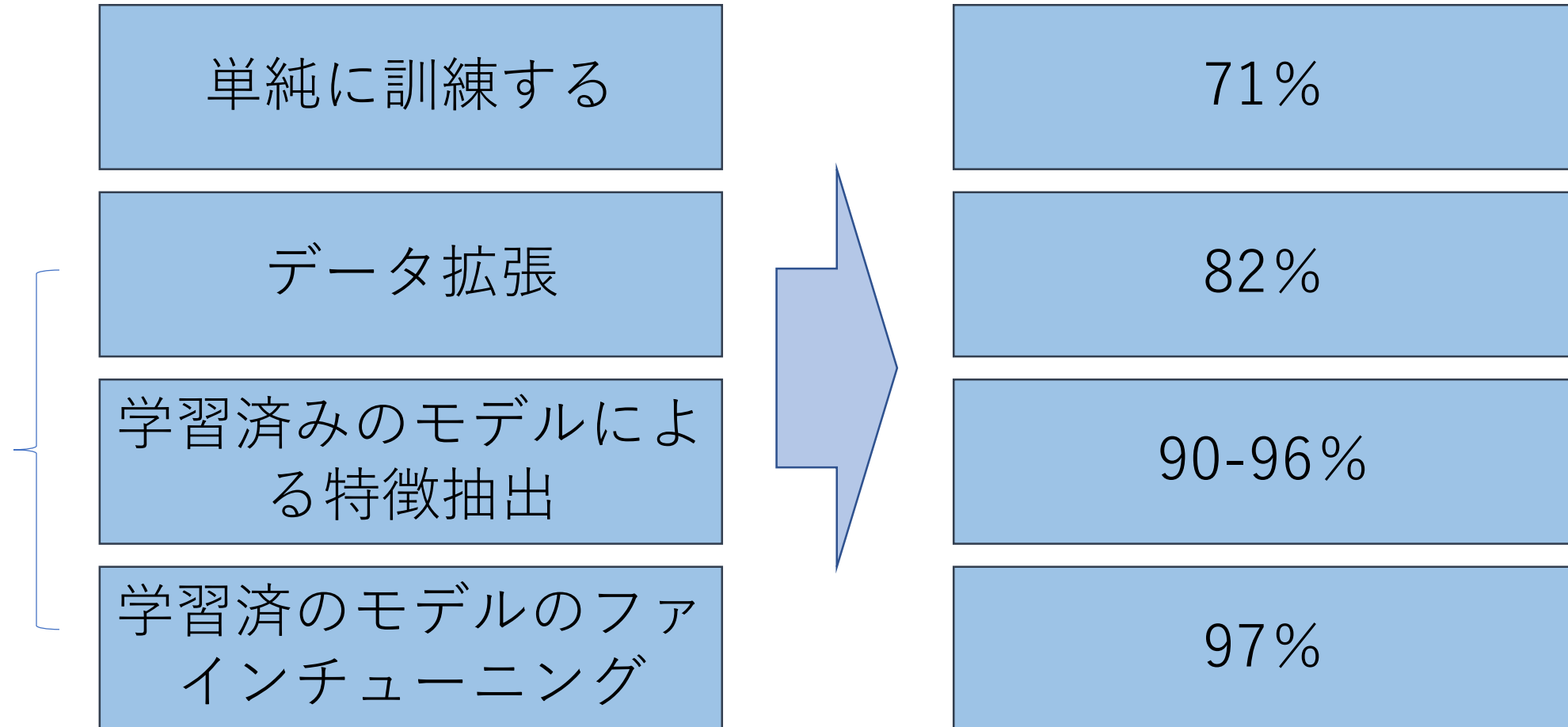


# PythonとKerasによるディープ ラーニング

19nd302h マブン

## 5.2 小さなデータセットでCNNを一から訓練する

正解率を上げるために、三つの戦略があります





## 5.2.1 小さなデータセットとディープラーニング

ディープラーニングによる学習精度を上げるためには大量のデータが必要と言われていますが、学習済みのモデルを流用することで少ないデータから効率よく・高精度な学習をさせることが可能な場合があります。

さらに、ディープラーニングのモデルはそもそも別の目的に使用できる可能性が非常に高いモデルです。

## 5.2.2データのダウンロード

- ここで使用する Dogs vs.cats データセットは、Keras でパッケージ化されていません。このデータセットは、Kaggle の 2013 年後半のコンピュータービジョンコンペで提供されたものです。

## 5.2.3 ネットワークの構築

ここでは少し大きな画像を扱っており、問題もより複雑です。そのため、CNNもより大きくなり、Conv2D層とMaxPooling2D層が一組増えています。これには2つの理由があります。

- ネットワークのキャパシティの増強
- Flatten層に到達したときに特徴マップが大きくなりすぎよう、特徴マップのサイズを削除することです。この場合、入力のサイズ $150 \times 150$ で始まり、特徴マップのサイズはFlatten層に渡される前に $7 \times 7$ になります。

## 5.2.4 データの前準備

データをCNNに供給するには、浮動小数点数型のテンソルとして適切に処理しておく必要があります。データはJPEGファイルとして保存されています。このため、データをCNNに渡すための手順は下にあります。

- ①画像ファイルを読み込み
- ②JPEGファイルの内容をRGBのピクセルリッドにデコードする
- ③これらのピクセルを浮動小数点数型のテンソルに変換する
- ④ピクセル値 (0~255) の尺度を取り直し、 $[0,1]$  の範囲の値にする

## 5.2.5 データ拡張

- データ拡張は、既存の訓練サンプルからさらに訓練データを生成するというアプローチをとります。ランダムな変換をいくつか実行していかにもそれっぽく見える画像を生成することで、散布を水増やしするのです。
- 目標として、モデルが全く同じ画像を2回にわたって学習したりしないようにすることです。そうすれば、モデルがデータの性質をさらに学習し、うまく汎化するようになります。