Convolutional Neural Networkの概要

2017年12月1日
乃村研究室 江見 圭祐

1 はじめに

本資料では，Convolutional Neural Network(以下，CNN) の各層における入出力の形状と畳み込み演算の仕組みについて説明する．

2 CNNの各層における入出力の形状

まず，CNN のモデルを可視化し，各層における入出力の形状を示す．Keras の Utils モジュールには，構築したネットワークの可視化機能が用意されている．なお，ネットワークの可視化には，Macports や Homebrew から graphviz，pip から pydot-ng をインストールする必要がある．

以下に fchollet/keras に用意されている mnist_cnn.py のネットワークを可視化するソースコードを添付する．また，図 1 に可視化したネットワークの図を示す．ここで，Dropout 層は入出力の形状が変わらないため，省略している．

```
 1 from __future__ import print_function
 2 import keras
 3 from keras.models import Sequential
 4 from keras.layers import Dense, Dropout, Flatten
 5 from keras.layers import Conv2D, MaxPooling2D
 6 from keras.utils import plot_model
 7 from keras import backend as K
 8
 9 batch_size = 128
10 num_classes = 10
11 epochs = 12
12 # input image dimensions
13 img_rows, img_cols = 28, 28
14
15 if K.image_data_format() == 'channels_first':
16     input_shape = (3, img_rows, img_cols)
17 else:
18     input_shape = (img_rows, img_cols, 3)
19
20 model = Sequential()
21 model.add(Conv2D(32, kernel_size=(3, 3),
```

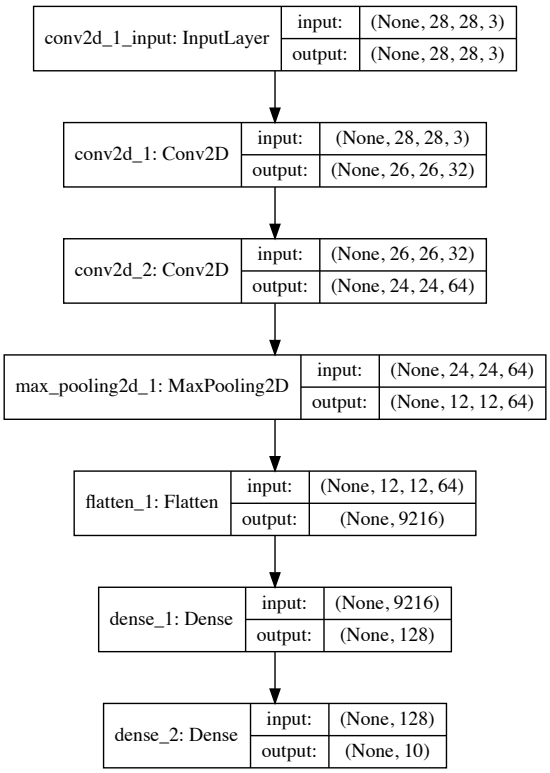

図 1: mnist_cnn.py のネットワーク

```
22                  activation='relu',
23                  input_shape=input_shape))
24 model.add(Conv2D(64, (3, 3), activation='relu'))
25 model.add(MaxPooling2D(pool_size=(2, 2)))
26 model.add(Flatten())
27 model.add(Dense(128, activation='relu'))
28 model.add(Dense(num_classes, activation='softmax'))
29 plot_model(model, to_file='cnn_model.pdf', show_shapes = True)
```

図 1 の最上部の InputLayer は，入力層を示す．ここでは，(None, 28, 28, 3) が入出力の形状となっている．入出力の 1 つ目の次元の値はバッチ数を表しており，ネットワークの構築時には，None として扱われる．2 つ目と 3 つ目の次元の値は，画像の縦横の画素数を表す．4 つ目の次元の値は，チャンネル数であり，ここでは RGB を表している．

次の Conv2D は，畳み込み層を表す．畳み込み層の出力は (None, 26, 26, 32) となっている．画像の縦横のノード数が減っているのは，畳み込み時にパディングが行われていないためである．ソースコードの 23 行目で，畳み込みのカーネルサイズが 3x3 となっているため，パディングしない限り，縦横の画素 2 つ分だけ畳み込み後の画像が小さくなる．また，3 つ目の次元の値は，畳み込み層のフィルタの数で

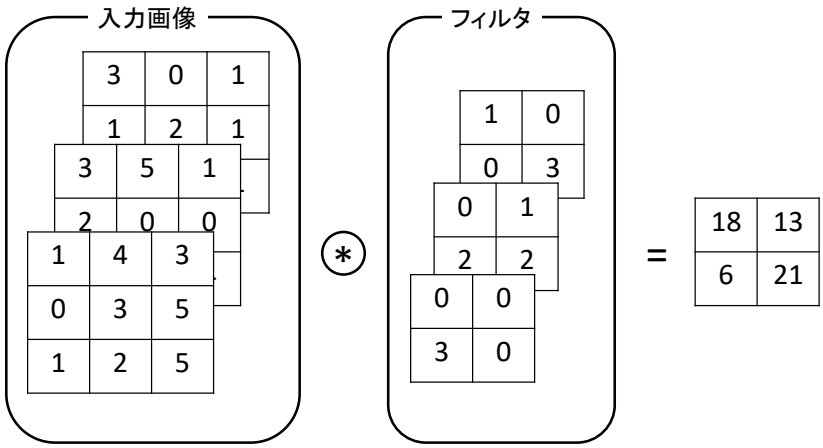

図 2: 畳み込み層の演算の様子

ある 32 となっている．つまり，RGB の画像に対し，各フィルタで畳み込んだ結果の 32 枚の画像が出力となる．同様にして，次の層でも畳み込みを行っている．

MaxPooling2D では，畳み込んだ画像から，特徴を抽出している．ソースコードの 27 行目から，プーリング層は 2x2 の範囲で特徴を抽出していることがわかる．このため，プーリング層の出力画像の縦横の大きさは，それぞれ入力の半分となっている．また，3 つ目の次元の値は 64 となっており，入力と変わらない．このことから，プーリング層は，与えられた画像から，特徴を抽出した画像を返すだけの役割を担っている．

その後，次の層では Flatten により，出力の形状を 1 次元 (バッチを含むと 2 次元) に圧縮し，全結合層を繋げている．

3　畳み込み層の演算

次に，畳み込み層で行っている演算の仕組みについて，図を用いて説明する．

図 2 に，入力として与えられた画像を単一の 2x2 フィルタで畳み込みを行う様子を示す．ここでは，RGB の 3 つのチャンネルをもつ 3x3 の画像を入力例として考える．入力画像がチャンネルを複数持つ場合，各フィルタもチャンネル数に対応する次元を持つ．図 2 の場合，入力画像は RGB の 3 つのチャンネルをもつため，フィルタも 3 つのチャンネルをもつ．畳み込み層では，入力画像とフィルタから，畳み込み演算を行い，画像を出力する．出力画像の各画素値は，チャンネルごとに畳み込んだ値の総和となる．たとえば，出力画像の $(1, 1)$ の値は，$(1*0+4*0+0*3+3*0)+(3*0+5*1+2*2+0*2)+(3*1+0*0+1*0+2*3)=18$ となる．畳み込み層のフィルタ数が増えると，その分だけ出力数のチャンネル数も増加する．

上記のようなチャンネルをもつ画像の畳み込み演算は，直方体のブロックを用

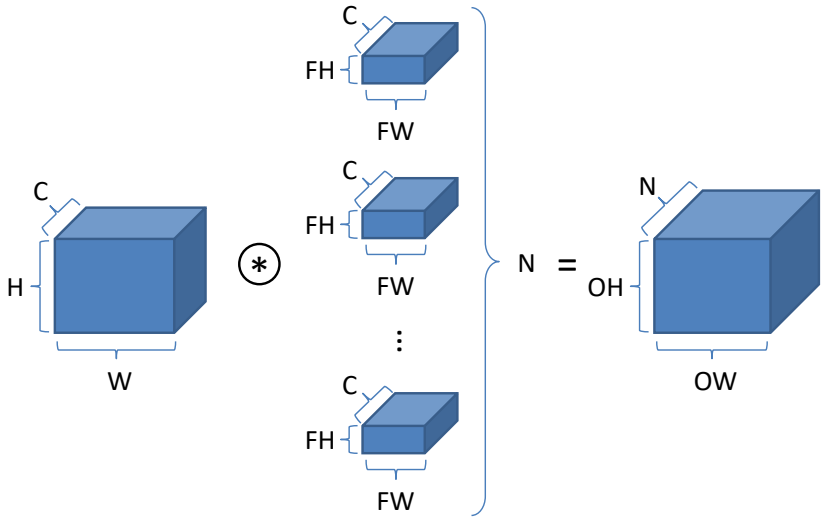

図 3: ブロックによる 3 次元の畳み込み演算の表現

いると図 3 のように表現できる．図 3 の左辺は，HxW の大きさの入力画像であり，チャンネル数は C で表現されている．一方，畳み込みを行うフィルタは，N 個用意されており，それぞれは FHxFW の大きさで，チャンネル数は C で表現される．この畳み込み演算の結果は，OHxOW の大きさで，チャンネル数は N の画像となる．ここで，OH と OW は，入力画像とフィルタのサイズによって決まる．パディングを行えば，OHxOW は HxW と等しい．

4 おわりに

本資料では，CNN の各層の入出力の形状と畳み込み演算の仕組みについて述べた．