

CNNを用いた画像認識における 中間層の情報の応用

木下研究室 201503772 大塚和真

研究背景

- ◆ CNN (Convolutional Neural Network) は特に画像認識の分野で用いられ、その有効性の高さから日々応用範囲が拡大している。
- ◆ CNNの高い性能を上げられる理由の一つが中間層にある。
- ◆ しかし、中間層における情報表現を活用する手法は確立されていない。

目的

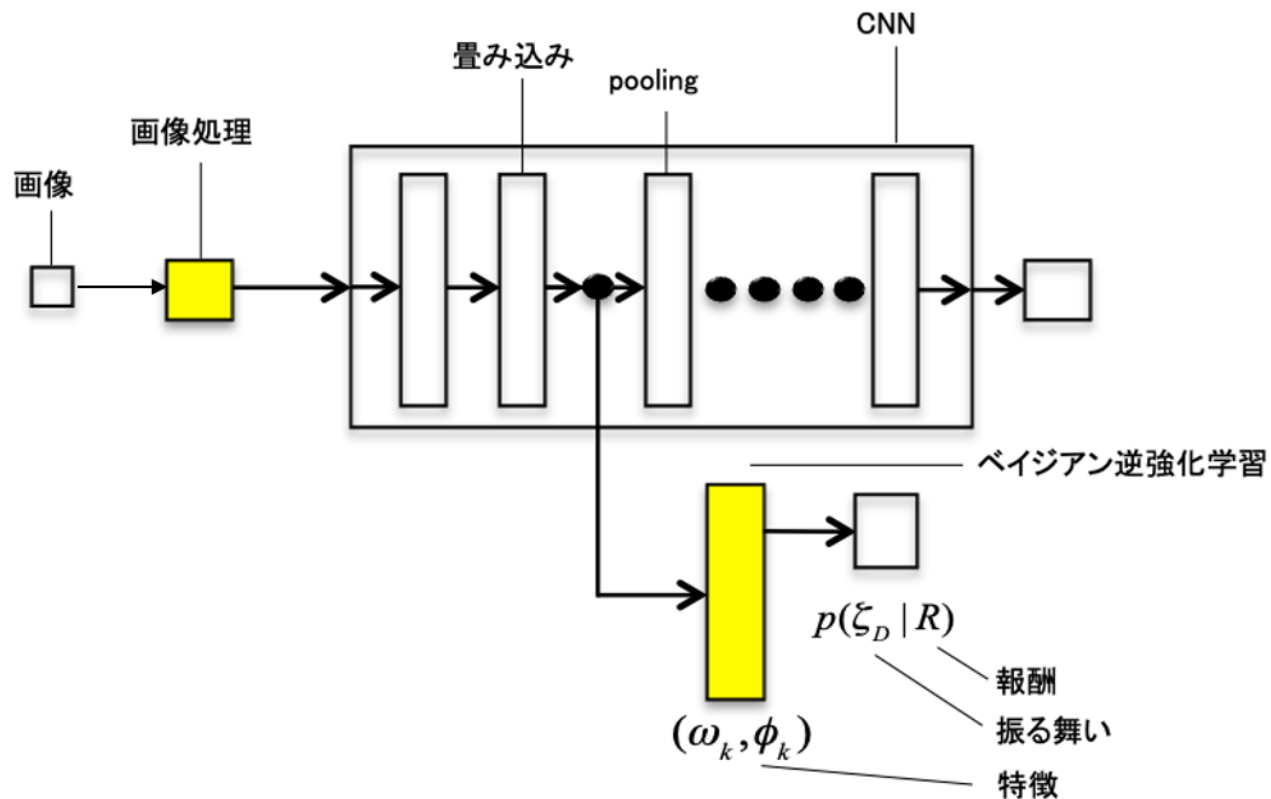
- CNNの中間層から得られる情報をもとに、画像に固有の特徴量を抽出する。

※本研究では、スケーリング(拡大・縮小)画像や、アフィン変換(変換前後で並行性を保つ変換)画像ではなく、回転画像を用いる。

オリジナリティ

- 回転画像を複数枚用意し、
Keras(機械学習のライブラリ)のCNNを用いて中間層を抜き出す。
- 抜き出した中間層のデータをベイジアン逆強化学習によって学習し、
特徴量を分析する。

提案手法



ベイジアン逆強化学習による画像の不変量学習モデル

ベイジアン逆強化学習

ベイジアン逆強化学習とは逆教科学習をベイズの枠組みで定式化したものであり、以下の式で表す。

$$p(R|Z_d) = \frac{p(Z_d | R)p(R)}{p(Z_d)}$$

事後確率分布 : $p(R|\xi_d)$

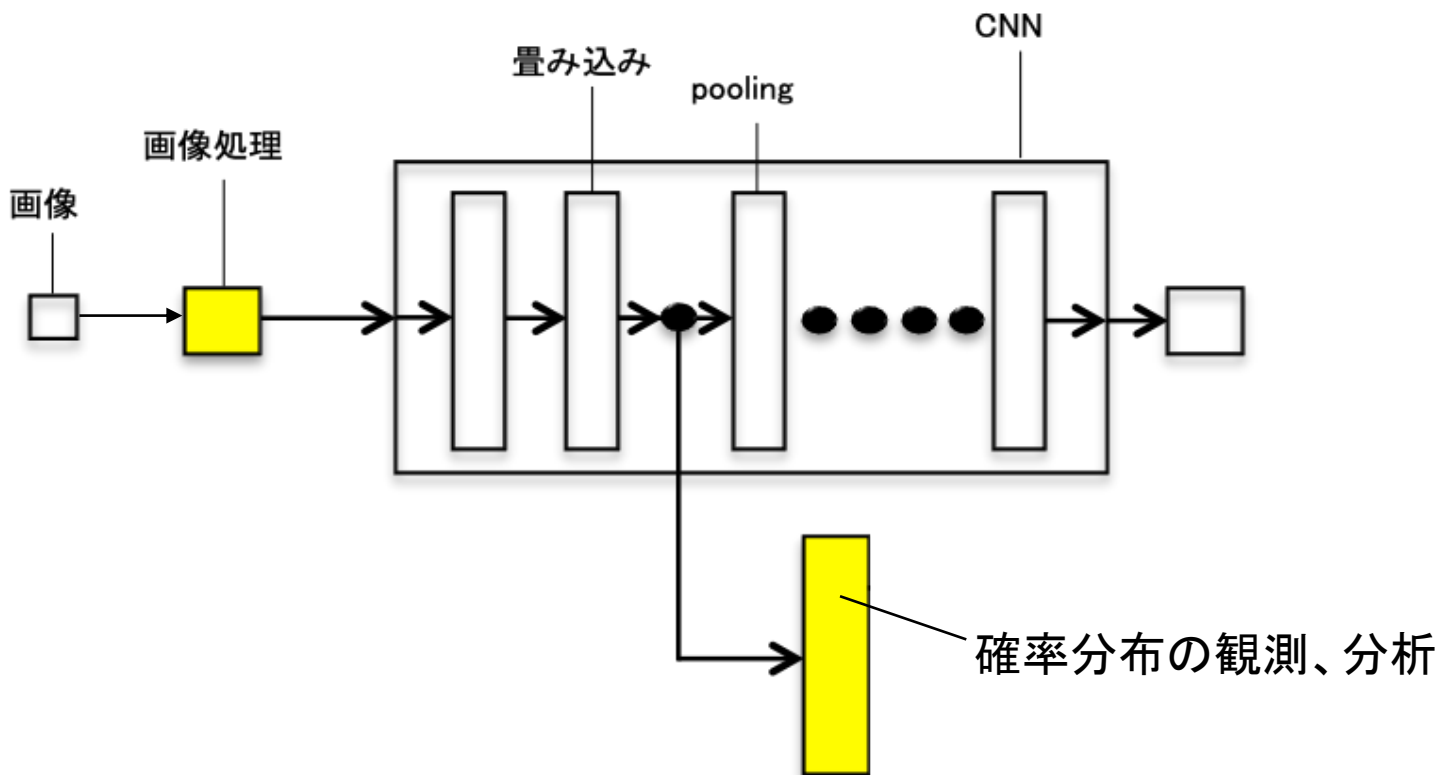
報酬 : $R = f(w, \phi)$

特徴 : ϕ

特徴量 : W

逆強化学習は、エキスパートの振る舞いから報酬を推定することができ、それによって表現しにくい報酬を求めることができる。

本研究で実施するモデル



- ベイジアン逆強化学習においては、特徴量をモデル化する必要がある。
- 今回は画像を変形した時にどのような不変な量がそこにあるのか、確率的なパラメータを知っておく必要がある。

Keras

■ Kerasのインストール

- Kerasのver2.2.4を導入

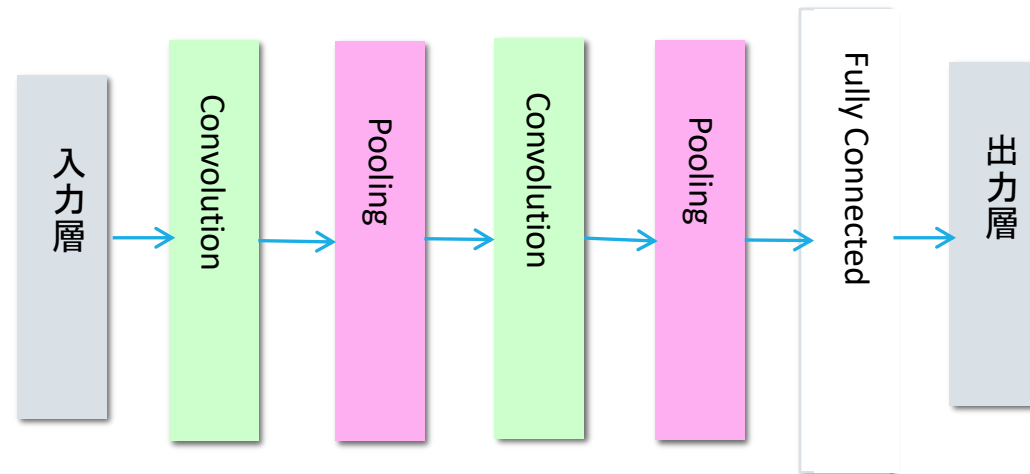
機械学習のライブラリであり、Python(言語)で書かれたTensorFlow(さまざまな機械学習の分野で使用するためオープンソフトウェアライブラリ)などで実行可能なニューラルネットワークライブラリである。

→比較的短いソースコードで実装することが可能。

■ CNNはこのKerasに入っている。

KerasのCNN(畳み込みニューラルネットワーク)

- CNNはニューラルネットワークの中間層の部分に畳み込み層(Convolution層)とプーリング層(Pooling層)を組み込んで構成されるニューラルネットワーク。



- 畳み込み層(Convolution層)は画像内にある小領域を設けて特徴量のまとまりを作るフィルタのような処理を実施する。
- Pooling層で特徴量を圧縮することで、画像の移動や変形に対して影響を受けにくくし、計算量を下げる。

提案手法

1. Jupyter Notebook上で画像処理ライブラリであるOpenCVを使い、 0° ~ 30° まで 1° ずつ回転させた画像を用意する。
2. その後KerasのCNNを用いて中間層を抜き出す。
3. 抜き出した中間層データの特徴量の確率分布を観測、分析する。
 - 学習させた中間層データの確率分布の不変量から画像の特徴量を分析する。

CNNソースコード

■ KerasのCNNのモデル概要

- conv2d:2次元の畳み込み層
- max_pooling2D:2次元ののマックスプーリング層
- Dropout:過学習の防止モジュール
- Flatten:入力を平滑化モジュール
- Activation:活性化関数モジュール

■ 実装環境

- Anaconda4.5.12
- Python 3.6
- Keras 2.2.4

■ KerasのCNNで学習後に中間層の出力を取り出すコード

```
From keras.models import 学習済みのモデル名
middle_model=
Model(入力レイヤの名前=model.input,
出力レイヤの名前=model.get_layer(layer_name).outputs)
Middle_output=middle_model.predict(入力データ)
```

■ 出典

Keras Documentation(<https://keras.io/ja/>)
[keras]中間層の出力
(<https://qiita.com/sasayabaku/items/75496baff597fd5fe79>)

Keras動作確認

■ Kerasで中間層の出力

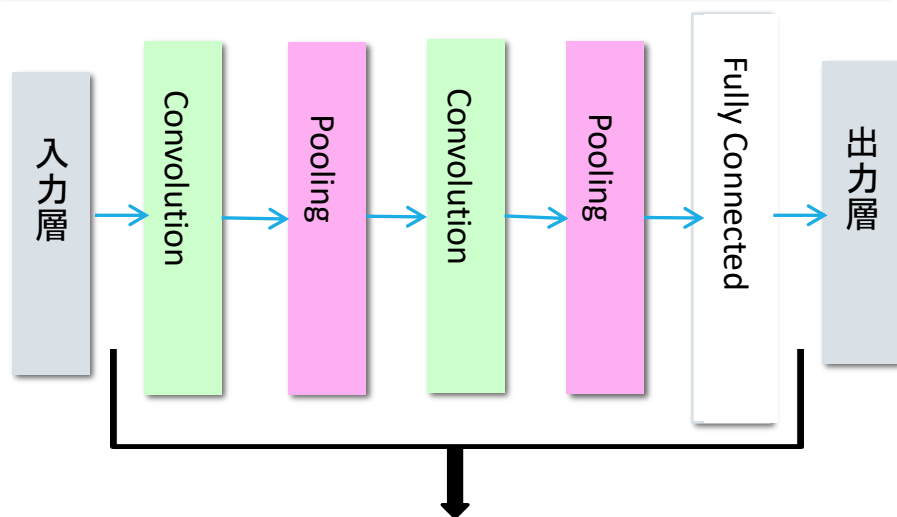
- ・Kerasで用意されているCIFAR-10という画像のデータセットとサンプルコードを用いて、`cnn_model.json`という名前のCNNを構築する。

ここで畳み込み層、プーリング層を何層組むのを決められる。

Keras動作確認

- 以下のコードでCNNのモデルを読み込む。

```
from keras.models import model_from_json  
  
json_string = open('cnn_model.json').read()  
model = model_from_json(json_string)
```



- model.summaryで中間層のモデルの要約を出力する。

```
model.summary()
```

Layer (type)	Output Shape	Param #
conv2d_13 (Conv2D)	(None, 32, 32, 32)	896
activation_19 (Activation)	(None, 32, 32, 32)	0
conv2d_14 (Conv2D)	(None, 30, 30, 32)	9248
activation_20 (Activation)	(None, 30, 30, 32)	0
max_pooling2d_7 (MaxPooling2D)	(None, 15, 15, 32)	0
dropout_10 (Dropout)	(None, 15, 15, 32)	0
conv2d_15 (Conv2D)	(None, 15, 15, 64)	18496
activation_21 (Activation)	(None, 15, 15, 64)	0
conv2d_16 (Conv2D)	(None, 13, 13, 64)	36928
activation_22 (Activation)	(None, 13, 13, 64)	0
max_pooling2d_8 (MaxPooling2D)	(None, 6, 6, 64)	0
dropout_11 (Dropout)	(None, 6, 6, 64)	0
flatten_4 (Flatten)	(None, 2304)	0
dense_7 (Dense)	(None, 512)	1180160
activation_23 (Activation)	(None, 512)	0
dropout_12 (Dropout)	(None, 512)	0
dense_8 (Dense)	(None, 10)	5130
activation_24 (Activation)	(None, 10)	0

Total params: 1,250,858
Trainable params: 1,250,858
Non-trainable params: 0

実験結果

OpenCVによる画像回転

▪ OpenCV ver3.4.2 : 動画・画像処理ライブラリ

- Jupyter Notebook上でlena.jpgという画像を 0° ~ 30° まで 1° ずつ回転させた画像を出力する。
- cv2.getRotationMatrix2Dという関数により、画像中心に対して θ° 回転することが可能。
- Imread('読み込み画像')
Imwrite('保存名',表示名)

```
import cv2
import matplotlib.pyplot as plt
import numpy as np
from scipy import ndimage
# 画像読み込み
I = cv2.imread('lena.jpg')

rIntr = 1
rs = 0
re = 30

Ir = []

for r in range(rs, re+1, rIntr):
    center = tuple(np.array(I.shape[0:2])/2)
    rotMat = cv2.getRotationMatrix2D(center, r, 1.0)
    Irot = cv2.warpAffine(I, rotMat, I.shape[0:2], flags=cv2.INTER_LINEAR)
    Ir.append(Irot)

cols = 4
rows = int(np.ceil(len(Ir) / float(cols)))
fig, axes = plt.subplots(nrows=rows, ncols=cols, figsize=(3*cols,3*rows))

for idx, I in enumerate(Ir):
    r = idx // cols
    c = idx % cols

    title = 'rotate=%d' % (rIntr*idx)

    axes[r,c].imshow(cv2.cvtColor(I, cv2.COLOR_BGR2RGB))
    axes[r,c].set_title(title)
    axes[r,c].set(adjustable='box-forced', aspect='equal')
    axes[r,c].get_xaxis().set_visible(False)
    axes[r,c].get_yaxis().set_visible(False)

for i in range(idx+1, rows*cols):
    r = i // cols
    c = i % cols
    fig.delaxes(axes[r,c])

# 画像保存
I = cv2.imwrite('lenafile/lena1.jpg',img_rot)

# 画像表示
cv2.imshow('img_rot', img_rot)
```

実験結果

lena.jpgを 0° ~ 30° まで回転させた結果



結論と考察

- Kerasにより、CNNの中間層データを抽出し、画像の変換による不変量を分析するモデルを提案した。また、学習に使う画像の準備、実際にKerasを動かす、サンプルコードによるCNNの構築をした。
- 本研究で用いる学習データに回転画像を提案したが、スケーリングやアフィン変換、さらに回転画像の背景の消去など、用いる画像によって得られる特徴量は変わると推測する。また、CNNを構築する際に畳み込み層、プーリング層の組み方を変えることで画像から得られる特徴量は変わると推測する。

今後の課題

- 準備した回転画像からCNNを構築し、そこから中間層データを得る。
- データの確率分布の不変量から、画像の特徴量を分析する。
- ベイジアン逆強化学習への応用。