5.4 CNNが学習した内容を可視化する

17T4015S

内間けんじ

5.4.0 CNNが学習した内容の可視化

ディープラーニングモデルはブラックボックスとよく言われるがCNNはその限りではない。

CNNによって学習された表現「視覚概念の表現」であるため可視化に非常に適している。

5.4.1 中間層の出力の可視化

中間層の活性化の可視化は、入力がどのように変換されるかを理解しCNN個々のフィルタの意味を把握するのに役立つ。

中間層の活性化は、特定の入力をもとにCNNのさまざまな畳み込み層とプーリング層によって出力される特徴マップを表示することで可視化できる。

5.4.1 中間層の出力の可視化

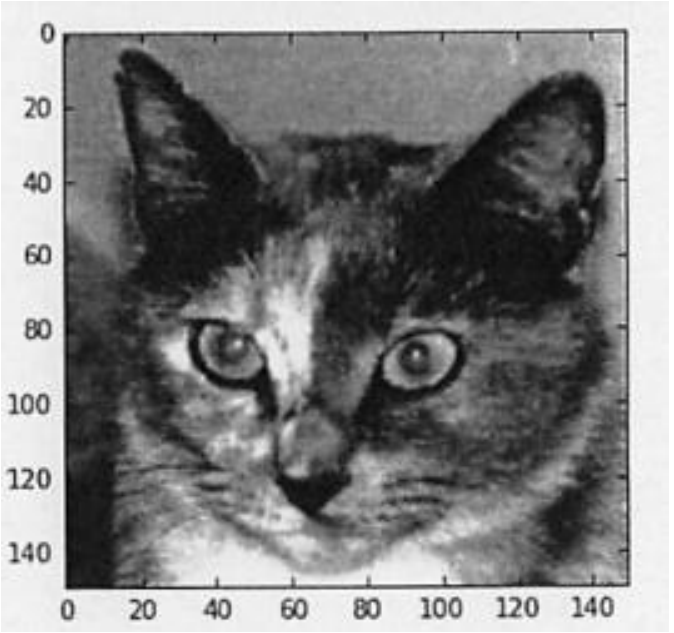
5.2節で利用したモデルを使う

Layer(type)	OutPutShape	Param
Conv2d_5(Conv2D)	(None, 148, 148, 32)	896
Maxpooling2d_5(MaxPooling2D)	(None, 74, 74, 32)	0
Conv2d_6(Conv2D)	(None, 72, 72, 64)	18496
Maxpooling2d_6(MaxPooling2D)	(None, 36, 36, 64)	0
Conv2d_7(Conv2D)	(None, 34, 34, 128)	73856
Maxpooling2d_7(MaxPooling2D)	(None, 17, 17, 128)	0
Conv2d_8(Conv2D)	(None, 15, 15, 128)	147583
Maxpooling2d_8(MaxPooling2D)	(None, 7, 7, 128)	0
Flatten_2(Flatten)	(None, 6272)	0
Dropout_1(Dropout)	(None, 6272)	0
dense_3(Dense)	(None, 512)	3211776
dense_4(Dense)	(None, 1)	513

5.4.1 中間層の出力の可視化

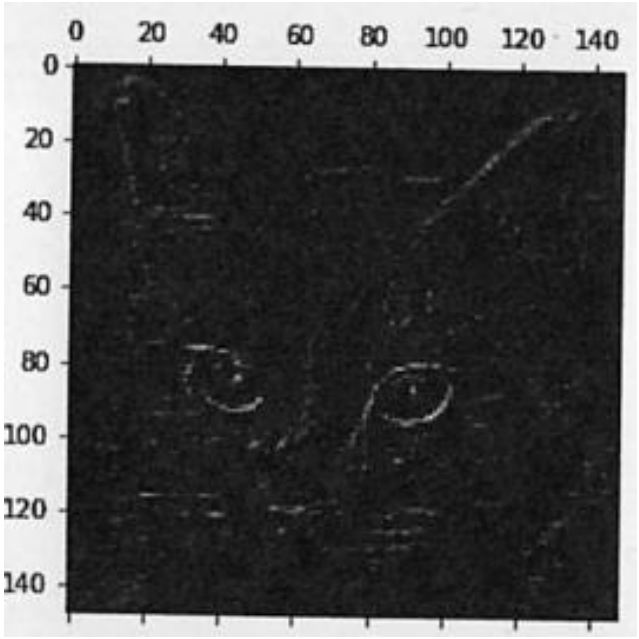
CNNが学習してない猫の画像を入力とし、元のモデルにおける活性化の値を調べる。

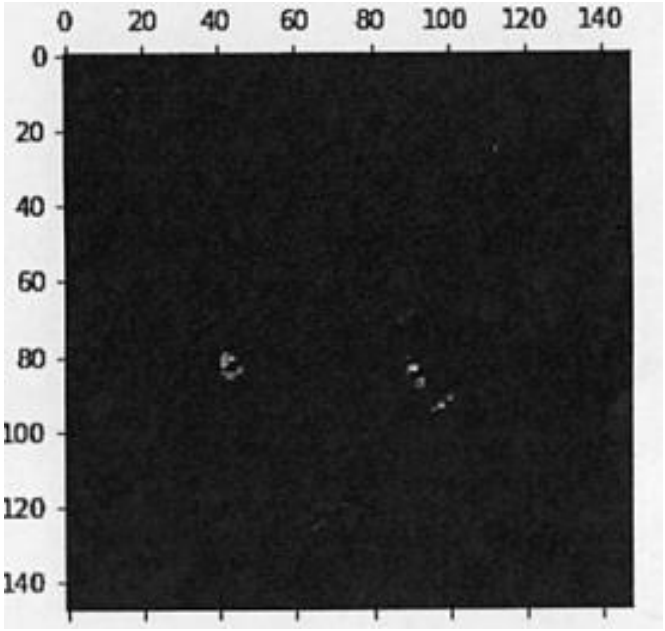
このとき、最初の畳み込み層の活性化についての特徴マップは148*148の32チャネルであった。



5.4.1 中間層の出力の可視化

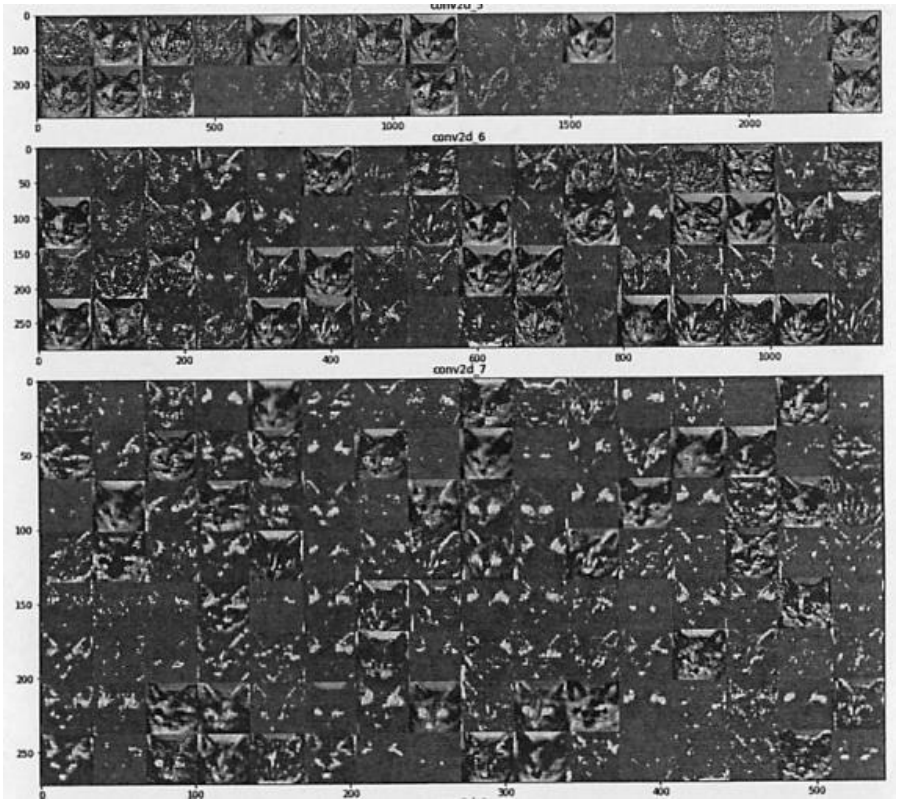
例として3番目のチャネル(左図)をプロットすると対角エッジ検出器のようである。30番目のチャネル(右図)は明るい緑ドット検出器のようである

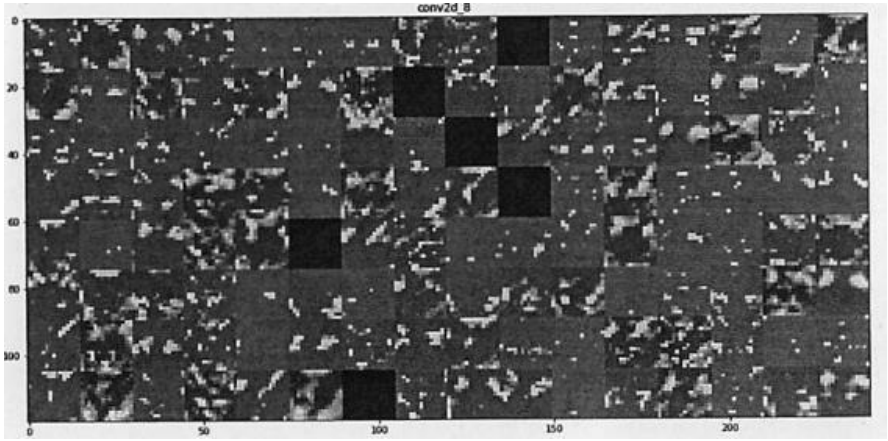




5.4.1 中間層の出力の可視化

すべての層の活性化チャネル





5.4.2 CNNのフィルタの可視化

フィルタの可視化は、各フィルタが受け入れる視覚パターンや視覚概念がどのようなものであるかを理解するのに役立つ。

空の入力画像から始めCNNの入力画像の値に勾配降下法を適用することで特定のフィルタの応答を最大化することで視覚化をおこなう

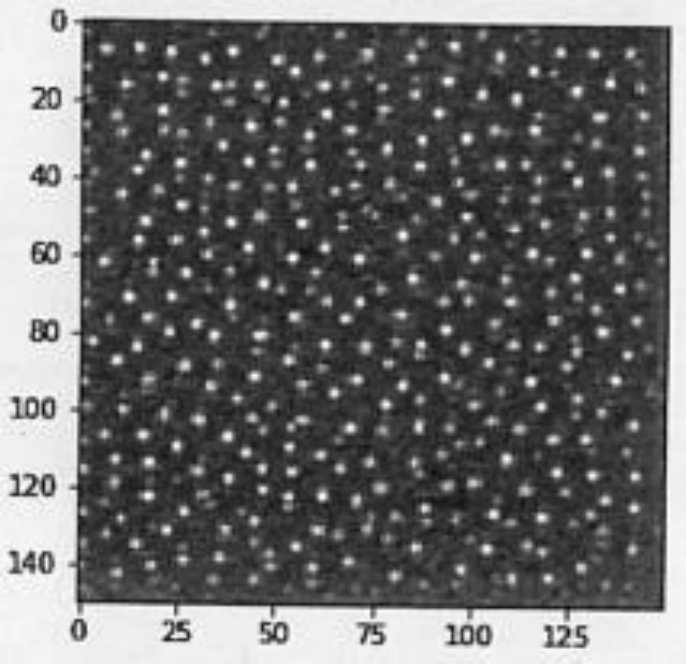
5.4.2 CNNのフィルタの可視化

仕様モデル：VGG16

Layer(type)	Output Shape	Param	Layer(type)	Output Shape	Param
input_1	(None, 150, 150, 3)	0	block4_conv1	(None, 18, 18, 512)	1180160
block1_conv1	(None, 150, 150, 64)	1792	block4_conv2	(None, 18, 18, 512)	2350808
block1_conv2	(None, 150, 150, 64)	36928	block4_conv3	(None, 18, 18, 512)	2350808
block1_pool	(None, 75, 75, 64)	0	block4_pool	(None, 9, 9, 512)	0
block2_conv1	(None, 75, 75, 128)	73856	block5_conv1	(None, 9, 9, 512)	2350808
block2_conv2	(None, 75, 75, 128)	147584	block5_conv2	(None, 9, 9, 512)	2350808
block2_pool	(None, 37, 37, 128)	0	block5_conv3	(None, 9, 9, 512)	2350808
block3_conv1	(None, 37, 37, 256)	295168	block5_pool	(None, 4, 4, 512)	0
block3_conv2	(None, 37, 37, 256)	590080			
block3_conv3	(None, 37, 37, 256)	590080			
block3_pool	(None, 18, 18, 256)	0			

5.4.2 CNNのフィルタの可視化

例として、Block3_conv1のフィルタ0を見ると水玉模様のパターンに応答しているように見える



5.4.3 活性化をヒートマップで可視化

CAMという手法を用いることで入力画像からクラス活性化のヒートマップをつくり画像のどの部分が**CNN**の最終的な分類の決め手になったかを理解できる。

クラス活性化のヒートマップは特定の出力クラスに関連付けられたスコアからなる二次元グリッドであり、入力画像の位置ごとに目的のクラスにとってどれぐらい重要であるかを計算する。

CAM(Class Activation Map)

畳み込み層の出力特徴マップと入力画像をもとに特徴マップの各チャネルに重み付けを行う。

このとき重み付はそのチャネルに関するクラスの勾配に基づいて行われる。

つまり、「入力画像によって様々なチャネルがどれくらい重要か」を表す空間マップを「そのクラスにとって各チャネルがどれくらい重要か」を表す値で重み付することで、「入力画像によってそのクラスがどれくらい強く活性化されるか」を表す入力マップが得られる。

5.4.3 活性化をヒートマップで可視化

次の画像をVGG16モデルで読み込めるように前処理し
学習済みのネットワークを適用する

アフリカゾウのテスト画像

```
>>print('Predict: ', decode_predictions(preds, top3)[0])
Predict: [('n02504458', 'African_elephant', 0.90942144)
          ('n01871265', 'tusker', 0.08618243)
          ('n02504013', 'Indian_elephant', 0.0043545929)]
```

5.4.3 活性化をヒートマップで可視化

Grad-CAMアルゴリズムを設定しヒートマップを
取得、元の画像にスーパーインポーズする。