

画像識別モデルの 実利用

アジェンダ

- 効率的な学習方法
- 異なるドメインの学習結果を利用する
- ImageNetによる事前学習
- 今回の事前学習で利用するモデル
- ファインチューニング
- ハンズオン

効率的な学習方法

教師あり学習において、目的とするタスクでの教師データが少ない場合に、別の目的で学習した学習済みモデルを再利用する転移学習についてご説明します。

異なるドメインの学習結果を利用する

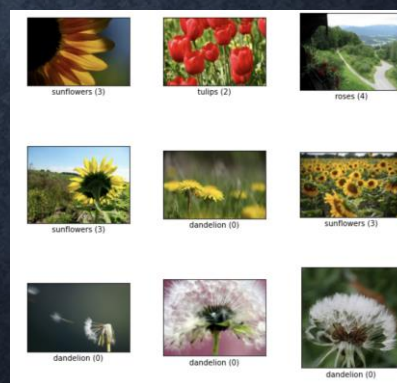
ImageNet(大量のデータ)



学習済みモデル



対象タスク(少ないデータ)



利用?

異なるドメインの学習結果を利用する

異なるドメインのデータで精度の高い学習済みモデルがあるとした場合・・・

- そのモデルの構造は似たタスクでも有効ではないか？
- 学習済みモデルを別タスクでそのまま利用できるのではないか？
- 事前に学習した情報から始めた方が学習が効率的になるのではないか？

ImageNetによる事前学習

ImageNetは1400万件以上の写真のデータセット。様々なAI/MLモデルの評価基準になっており、学習済みモデルも多く公開されている。



事前学習で利用するモデル

ImageNetを1000分類で分類した教師データを利用。ResNetにより学習。以下はサンプル。

No.	Index	Label
1	0	tench, Tinca tinca
2	1	goldfish, Carassius auratus
3	2	great white shark, white shark, man-eater, man-eating shark, Carcharodon carcharias
4	3	tiger shark, Galeocerdo cuvieri
5	4	hammerhead, hammerhead shark

事前学習で利用するモデル

ImageNet学習済みモデルの概要(ResNet抜粋)。

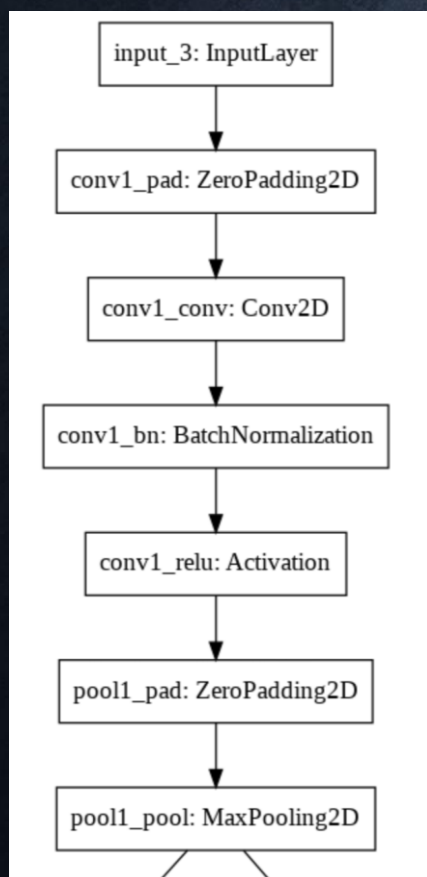
No.	Model	Size	Top-1 Acc	Params
1	ResNet50	98MB	0.749	25,636,712
2	ResNet101	171MB	0.764	44,707,176
3	ResNet152	232MB	0.766	60,419,944

ハンズオンではResNet50での例を説明。その他のモデルでの例はリンク先を参照のこと。

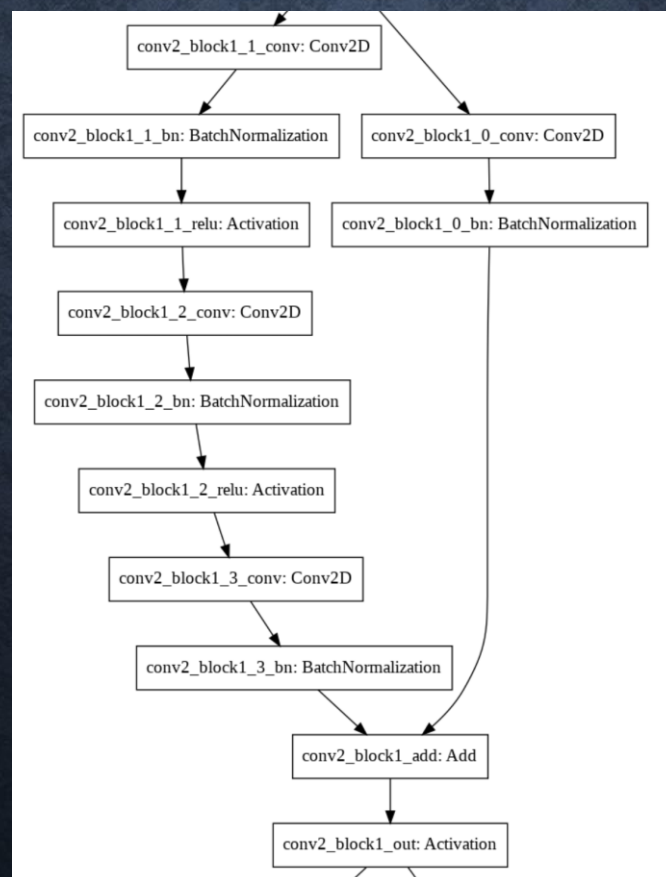
事前学習で利用するモデル

ResNetの構造の概要。

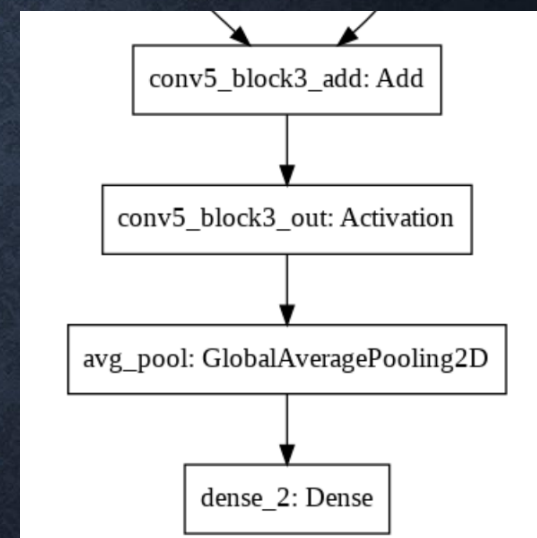
入力層



中間層(繰り返し)

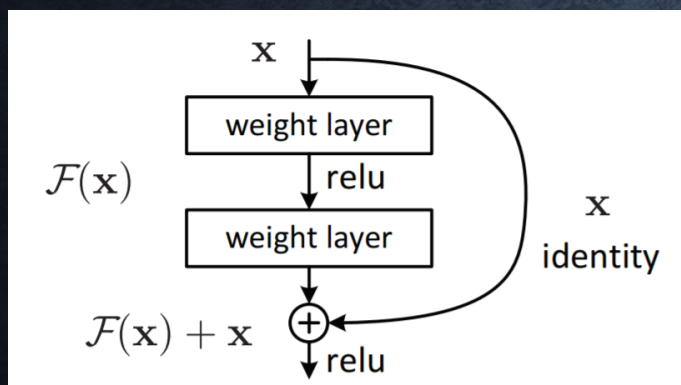


出力層



ResNet: SkipConnection

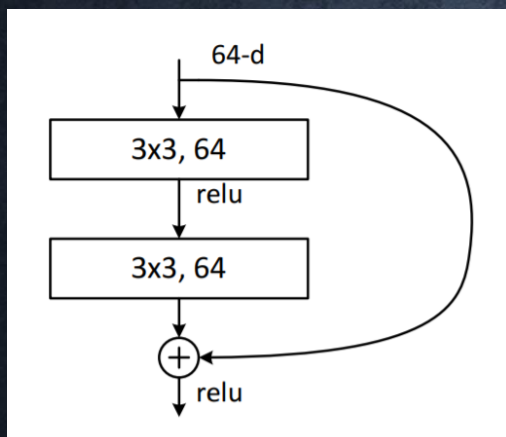
中間層部分



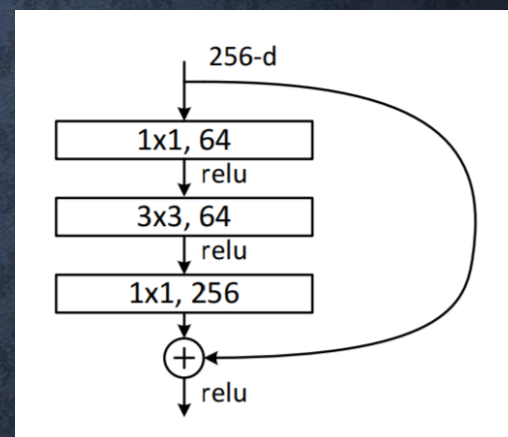
- 深い層の積み重ねでも学習可能に
 - 勾配消失の回避
 - 勾配爆発の回避
- 中間層の部分出力: $H(x)$
- 残差ブロック: $H(x) = F(x) + x$
- 学習部分: $F(x)$

ResNet: Bottleneck構造

Plainアーキテクチャ



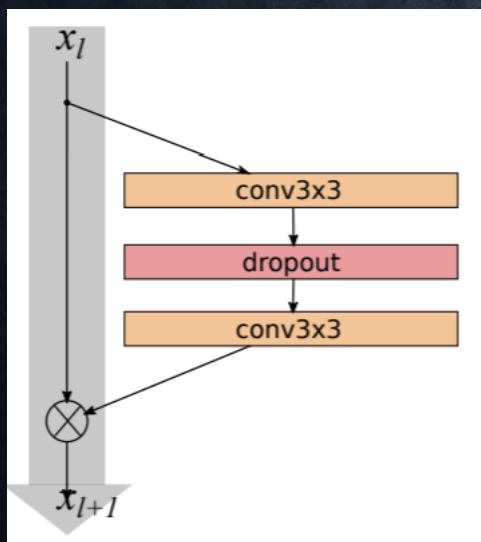
Bottleneckアーキテクチャ



- 同一計算コストで1層多い構造
- 途中の層で3x3の畳込みを行う

WideResnet: 構造

group name	output size	block type = $B(3,3)$
conv1	32×32	$[3 \times 3, 16]$
conv2	32×32	$\begin{bmatrix} 3 \times 3, 16 \times k \\ 3 \times 3, 16 \times k \end{bmatrix} \times N$
conv3	16×16	$\begin{bmatrix} 3 \times 3, 32 \times k \\ 3 \times 3, 32 \times k \end{bmatrix} \times N$
conv4	8×8	$\begin{bmatrix} 3 \times 3, 64 \times k \\ 3 \times 3, 64 \times k \end{bmatrix} \times N$
avg-pool	1×1	$[8 \times 8]$



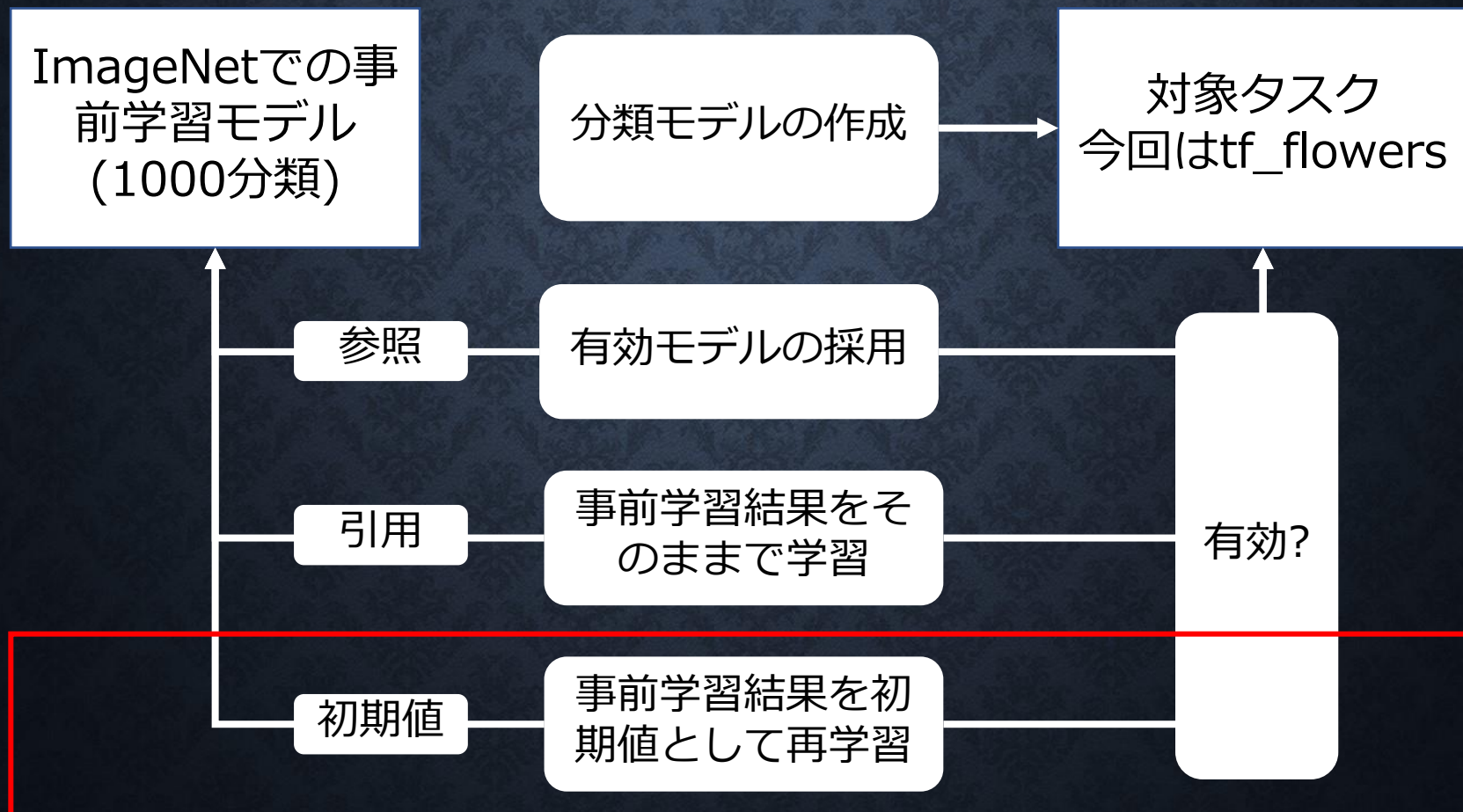
- ResNetにおけるフィルタ数をK倍
 - 畳込みチャンネル数が増加
 - 高速・高精度の学習が可能に
 - GPUの特性に合った動作
- ResNetに比べ層数を浅くした
- DropoutをResidualブロックに導入

事前学習で利用するモデル

Wide ResNet。

- フィルタ数を k 倍したResNet。
- パラメータを増やす方法として、層を深くするのではなく、各層を広く(Wide)した。
- ハンズオンではResNet-50とResNet-50 x 3($k=3$ のWide ResNet)の実装例を解説。

ファインチューニング



ハンズオン

Google Colaboratoryによるハンズオン。

- `transfer-learning.ipynb`
- `wide-resnet.ipynb`