

# 機械学習エンジニアコース Sprint

---

－ セマンティックセグメンテーション2 －



DIVE INTO CODE



# 大規模画像認識のコンペ

## ベンチマークデータセットを用いた 画像・動画画像認識コンペティション

画像認識手法や機械学習手法の研究論文を投稿する際は、標準的なベンチマークデータセットにおいて他手法と比較を行い、定量的に優位性を示すことが求められます。

ILSVRC (ImageNetを用いた大規模画像認識)

<http://image-net.org/challenges/LSVRC/2017/>

COCO challenge (Microsoft COCOを用いた物体領域分割、画像説明文生成)

<http://cocodataset.org/#home>

Places challenge (MIT Places を用いた 大規模シーン認識)

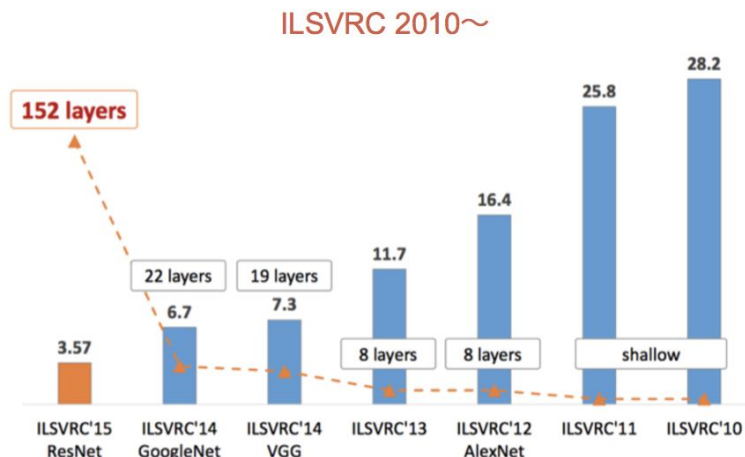
<http://places2.csail.mit.edu/demo.html>

VQA challenge (画像質問応答)

<http://vqa.cloudcv.org/>

The First LVIS Challenge (LVISを用いたインスタンスセグメンテーション)

<http://cocodataset.org/workshop/coco-mapillary-iccv-2019.html#vis-challenge>





# ILSVRC

# ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)

## ILSVRCとは？

**ImageNet** のデータの一部を用いた、2010年から2017年まで続いた大規模画像認識コンペティションのことです。スタンフォード大学をはじめ米国の複数の大学が共同で開催していました。

2012 年、この年、トロント大学の Hinton 教授らのグループが、8 層の畳込みニューラルネットワーク (CNN) を用いて 1,000 クラス識別を行い、2 位以下のチームに エラー率 10% 以上もの大差をつけて圧勝し、世界中の研究者に極めて大きな衝撃を与えました。

2015 年になると、Microsoft Research Asia が 152 層の CNN を用い、1,000 クラス識別においてエラー率 3.57% を達成しました。畳込み層をバイ

パスする結合を有する **residual network (ResNet)** と呼ばれる構造を備えたことによって、従来よりも更に深いモデルの学習が可能となりました。

ILSVRC のデータで学習されたネットワークはオープンソース上で共有されるようになり、他のデータセット上で **fine-tuning** を行う **転移学習** も現在ではこの分野内で定石の一つとして確立しています。

2017年以降は、ILSVRCはkaggleに引き継がれました。

ImageNet製作者Li Fei-Fei氏のスライド

[http://www.image-net.org/papers/ImageNet\\_2010.pdf](http://www.image-net.org/papers/ImageNet_2010.pdf)

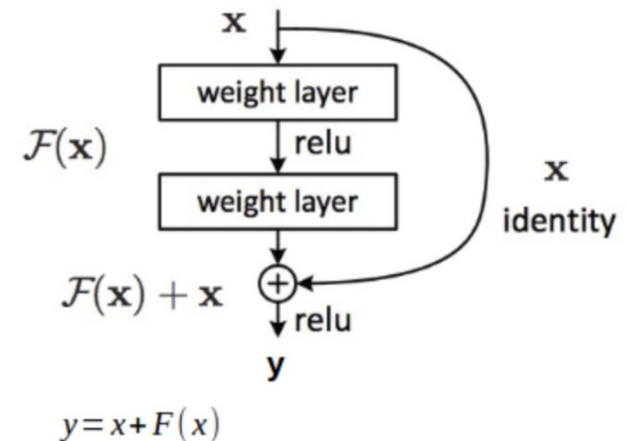


## Deep Residual Learning for Image Recognition (ResNet) (2015)

ResNetとは？

<https://arxiv.org/pdf/1512.03385.pdf>

ILSVRC 2015において、Kaiming Heらが提案した152層のNNをトレーニングするアーキテクチャで、「スキップコネクション」を導入し、大量のバッチノーマライゼーションを特徴とする。



$$\begin{aligned}\frac{\delta E}{\delta x} &= \frac{\delta E}{\delta y} * \frac{\delta y}{\delta x} = \frac{\delta E}{\delta y} * (1 + F'(x)) \\ &= \frac{\delta E}{\delta y} + \frac{\delta E}{\delta y} * F'(x)\end{aligned}$$



# Transfer Learning (転移学習)

転移学習とは？

<https://medium.com/@14prakash/transfer-learning-using-keras-d804b2e04ef8>

機械学習における研究課題の一つであり、ある問題を解決しそれを異なるが**類似した問題に適用する知識にフォーカスすること**です。

具体的には、pre-train（事前学習）されたネットワークの重みを、別のネットワークの重みの初期値とすることです。  
固定された特徴抽出器は、多くの問題解決に役立ちます。



# 転移学習

## どうして転移学習をするの？

非常に深いネットワークは学習コストが高くなります。

複雑なモデルは、高額なGPUが搭載されたマシンを何百個も  
使用し、数週間学習を行わなければなりません。

ResNetの学習時間は数週間かかる？

<https://www.slideshare.net/iwii/nips17-86470238>

<https://blog.nnabla.org/ja/news/imagenet-in-224s/>





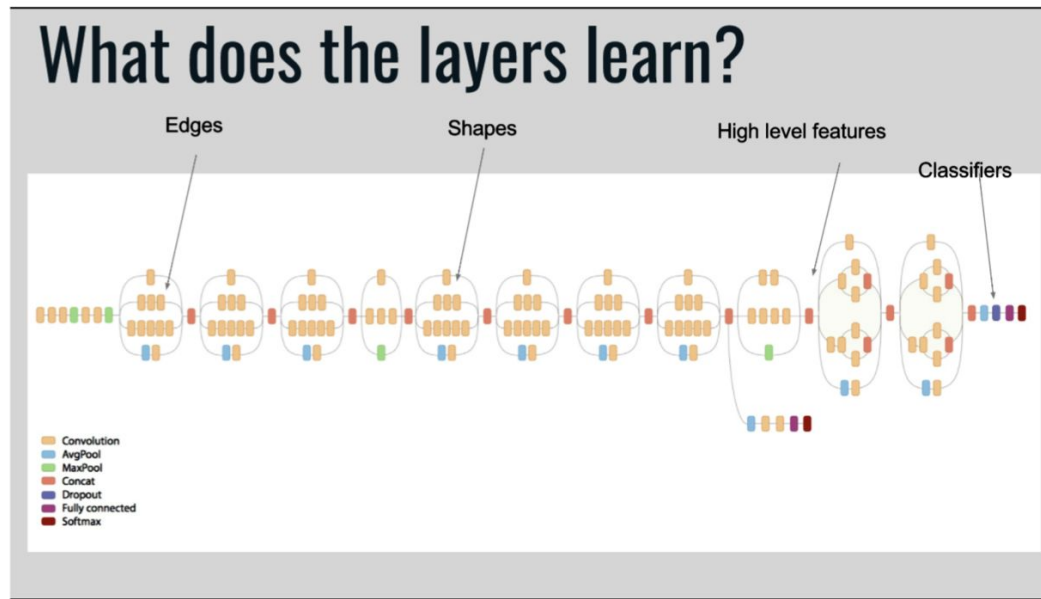
# 転移学習の重みの役割

## 転移学習の重みの役割は？

ネットワークの先頭レイヤは**edge**を検出し、中間レイヤは**shape**、後半レイヤは**高度な特徴**を検出しようとしています。

つまり、前方のレイヤはより一般的で、後方のレイヤはよりオリジナルデータセットに固有のものと言えます。

こうした検出器をもつ学習済みネットワークは、一般的に他のコンピュータビジョンの諸問題の解決に役立ちます。





# 学習済みの重み

## 事前学習されたネットワークはどこにあるの？

kerasの場合

ドキュメント：

<https://keras.io/ja/applications/>

github：

[https://github.com/keras-team/keras-applications/tree/master/keras\\_applications](https://github.com/keras-team/keras-applications/tree/master/keras_applications)

Keras Applications で用意されているImage Net画像の事前学習済み分類モデル

- [Xception](#)
- [VGG16](#)
- [VGG19](#)
- [ResNet50](#)
- [InceptionV3](#)
- [InceptionResNetV2](#)
- [MobileNet](#)
- [DenseNet](#)
- [NASNet](#)
- [MobileNetV2](#)

|  |   |  |
|--|---|--|
| Branch: master keras-applications / keras_applications / |   | Create new file Upload files Find file History |
| fchollet Prepare 1.0.8 release.                          |   | Latest commit 3b180cb 18 days ago              |
| ..   |   |  |
| __init__.py  | Prepare 1.0.8 release.  | 18 days ago                                    |
| densenet.py  | Revise docs for `include_top=False, pooling=None` (#48)       | 8 months ago                                   |
| imagenet_utils.py  | Update imagenet_utils.py                                      | 3 months ago                                   |
| inception_resnet_v2.py                                   | Add missing conference names of reference papers              | 3 months ago                                   |
| inception_v3.py  | Add missing conference names of reference papers              | 3 months ago                                   |
| mobilenet.py   | Custom image size in Mobilenets v1 and v2 (#101)              | 2 months ago                                   |
| mobilenet_v2.py  | Custom image size in Mobilenets v1 and v2 (#101)              | 2 months ago                                   |
| nasnet.py  | Add missing conference names of reference papers              | 3 months ago                                   |
| resnet.py  | Add ResNet, ResNetV2, and ResNeXt variants (#26)              | 7 months ago                                   |
| resnet50.py  | Add missing conference names of reference papers              | 3 months ago                                   |
| resnet_common.py   | Add ResNet, ResNetV2, and ResNeXt variants (#26)              | 7 months ago                                   |
| resnet_v2.py   | Add ResNet, ResNetV2, and ResNeXt variants (#26)              | 7 months ago                                   |
| resnext.py   | Add ResNet, ResNetV2, and ResNeXt variants (#26)              | 7 months ago                                   |
| vgg16.py   | Add missing conference names of reference papers              | 3 months ago                                   |
| vgg19.py   | Add missing conference names of reference papers              | 3 months ago                                   |
| xception.py  | Enable Xception, MobileNet, and V2 for `channels_first` (#96) | 3 months ago                                   |





# 転移学習

## 元のデータと新しいデータが異なるケースの対処は？

以下のようなケースが考えられます。

<https://cs231n.github.io/transfer-learning/>

### ①新しいデータセットが小さく、元のデータセットと似ている

# 最後の分類機 (FC layers) のみを訓練する

```
for layer in model.layers:
    layer.trainable = False
```

### ②新しいデータセットが大きく、元のデータセットと似ている

# 全ネットワークを訓練する。デフォルトでもTrueになっている

```
for layer in model.layers:
    layer.trainable = True
```

# もしedgeを検出する前半の重み (5レイヤー) がほしい場合は次のように書く

```
for layer in model.layers[:5]:
    layer.trainable = False
```



# 転移学習

## 元のデータと新しいデータが異なるケースの対処は？

以下のようなケースが考えられます。

<https://cs231n.github.io/transfer-learning/>

### ③新しいデータセットは小さいが元のデータセットと大きく異なる

# 転移学習が困難なパターン

# データが少ないので過学習を防ぐために上層だけを学習させたいが、似ていないデータを使って学習しているため、後半の特徴を使ってもうまく学習できないと考えられる。

# 後半ではなくどこか途中のレイヤに分類器を繋ぐとうまくいくかもしれない

### ④新しいデータセットが大きく、元のデータセットと大きく異なる

# ランダムな初期値を用いるか、pretrained modelの重みを初期値として用いる。一般に、後者の方が良いとされる。

# セマンティックセグメンテーション2 完