

# 機械学習関連メモ

Ryoichi Matsumoto

2020 年 7 月 12 日

## 目次

1	はじめに	1
2	Deep Learning	1
2.1	Deep Learning の歴史 . . . . .	1
2.2	Batch Normalization . . . . .	4
2.3	You Only Look Once . . . . .	6
3	Section	6
3.1	Subsection . . . . .	6
3.2	Subsection . . . . .	6

## 表目次

## 図目次

1	AI のブームと冬の時代 . . . . .	2
2	ILSVRC の歴代優勝モデル (2010 年～2016 年) . . . . .	3
3	人間の画像認識エラー率 . . . . .	3
4	2012 年以降の優勝モデル . . . . .	4

## 1 はじめに

機械学習に関して調べたことのメモ.

## 2 Deep Learning

### 2.1 Deep Learning の歴史

Deep Learning は AI を実現する手段のひとつであり, Deep Learning の歴史の前に AI 研究の歴史について概略を説明する. [1] [2] [3]

AIは図1に示すように、第一次ブームから第三次ブームがあり、これまでに2回、冬の時代が訪れている。第一次ブームは1950年代～1960年代で、商用コンピュータの登場によりAIの研究が進んだが、人間と同じ考え方を持たせるという理想への壁は高く、1970年代にブームが冷め、1回目の冬の時代が訪れた。第二次ブームは1980年代で、進化したコンピュータに知識を加えるアプローチが行われたが、コンピュータは知識の意味を理解するわけではないため、1990年代にブームが冷め、2回目の冬の時代が訪れた。第三次ブームは2000年代で、インターネットの発展により収集可能なデータの規模が劇的に増加し、AIの研究に活用ができるようになり、機械学習・Deep Learningの研究が加速し、2020年現在もなお、研究が進んでいる。

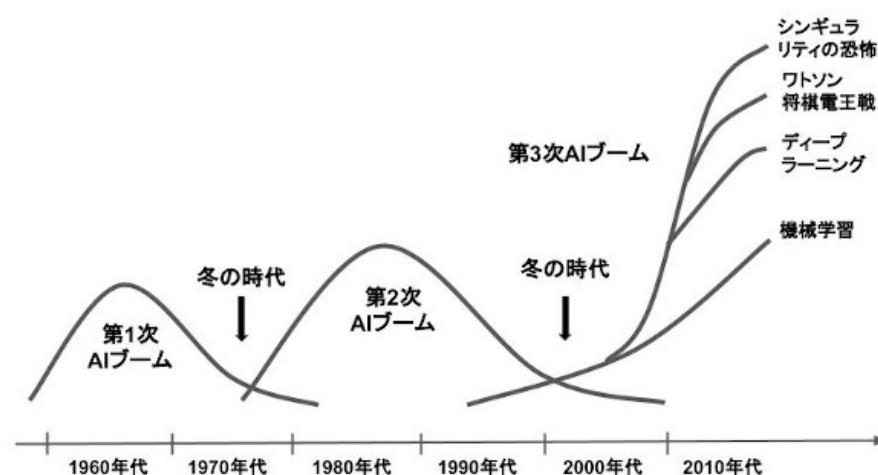


図1 AIのブームと冬の時代

Deep Learningが最初に脚光を浴びたのは、画像認識コンテスト ILSVRC(Imagenet Large Scale Visual Recognition Challenge)で、2012年に2011年の優勝モデルのエラー率を約10%下げて優勝した[4]。以降、Deep Learningを活用したアルゴリズムが台頭し、2015年には、人間のエラー率5.1% [5]を下回るアルゴリズムが登場した<sup>\*1</sup> (図2, 図3)。

<sup>\*1</sup> 図2の引用元のAINOWが何を参照して人間のエラー率を4%と定義したのかは不明

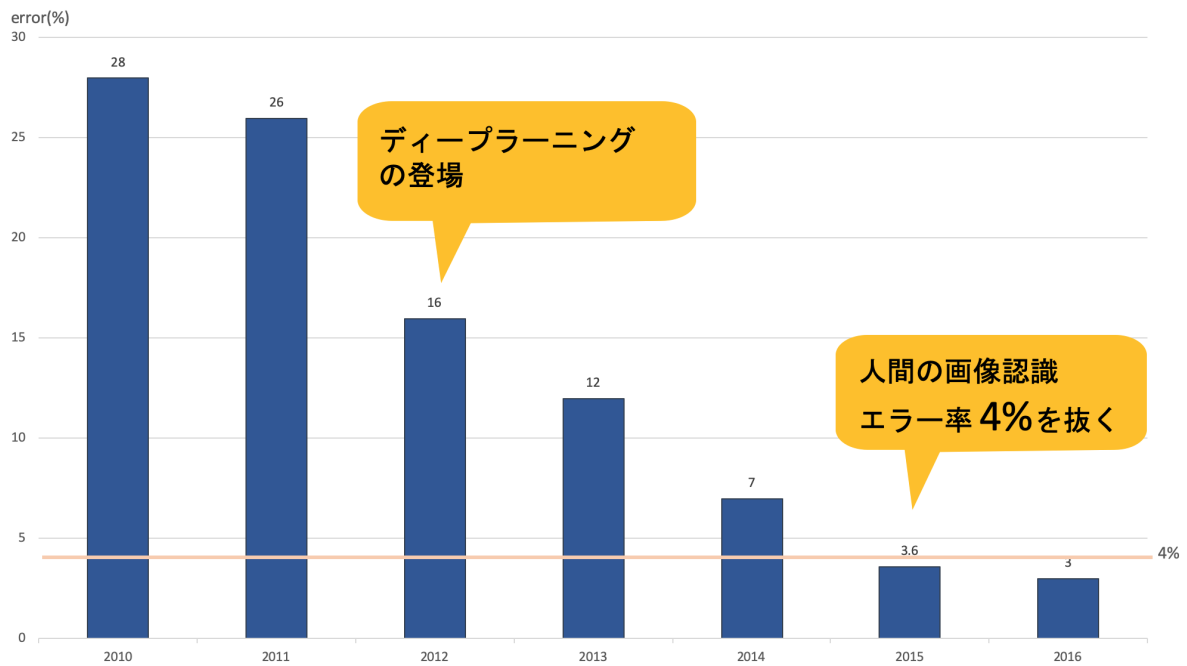


図2 ILSVRC の歴代優勝モデル (2010 年～2016 年)

Relative Confusion	A1	A2
Human succeeds, GoogLeNet succeeds	1352	219
Human succeeds, GoogLeNet fails	72	8
Human fails, GoogLeNet succeeds	46	24
Human fails, GoogLeNet fails	30	7
Total number of images	1500	258
Estimated GoogLeNet classification error	6.8%	5.8%
Estimated human classification error	5.1%	12.0%

**Table 9** Human classification results on the ILSVRC2012-2014 classification test set, for two expert annotators A1 and A2. We report top-5 classification error.

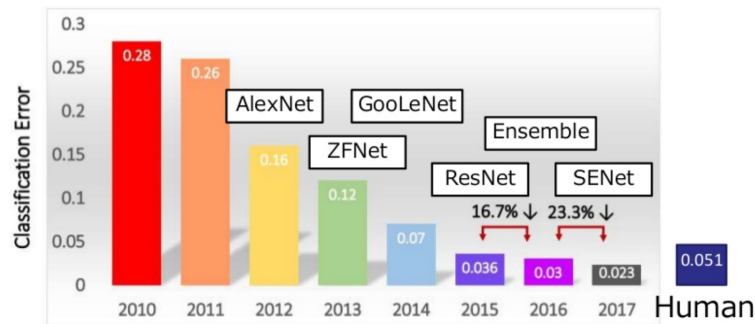
図3 人間の画像認識エラー率

2012 年以降の ILSVRC の優勝モデルは、図 4 に示す通り [6] \*2.

\*2 図中、2014 年の優勝モデルは GooLeNet となっているが、正しくは GoogLeNet

## ILSVRC

### ・クラス分類タスクのエラー率 (top5 error) の推移



DeNA [http://image-net.org/challenges/talks\\_2017/ILSVRC2017\\_overview.pdf](http://image-net.org/challenges/talks_2017/ILSVRC2017_overview.pdf)

11

図 4 2012 年以降の優勝モデル

#### 2.1.1 LeNet

#### 2.1.2 AlexNet

#### 2.1.3 XNet

#### 2.1.4 GoogLeNet

#### 2.1.5 ResNet

#### 2.1.6 SENet

### 2.2 Batch Normalization

#### 2.2.1 背景

訓練時とテスト時で入力（共変量）の分布が異なる状況は「共変量シフト」と呼ばれ、このような場合は良い学習結果が得られない。共変量シフト下においては、忘却型の学習アルゴリズムやパラメータ学習法、モデル選択法等が用いられる [7] が、Deep Learning においては、学習過程でパラメータ更新により各層の Activation の分布が変わる「内部共変量シフト (Internal Covariate Shift)」と呼ばれる問題がある [8] [9]。

#### 2.2.2 概説

Deep Learning は画像、音声など様々な分野で急速な技術進化をもたらした。Deep Network の学習方法として確率的勾配降下法 (Stochastic gradient descent ; SGD) が効果的であることが知られている [10]。SGD の派生として、Momentum, AdaGrad, RMSprop, Adam, AdaBound, RAdam などがある。SGD は Network のパラメータ  $\theta$  を、loss を最小化するように最適化する。

$$\Theta = \arg \min_{\Theta} \frac{1}{N} \sum_{i=1}^N l(x_i, \Theta) \quad (1)$$

$x_1 \dots x_N$  はデータ数  $N$  サンプルの学習データセットで、SGD での学習ステップは  $m$  サンプルのミニバッチ  $x_1 \dots x_m$  に対して行う。ミニバッチは、パラメータについての損失関数の勾配近似に使用され、式 (2) で導出する。

$$\frac{1}{m} \frac{\partial l(x_i, \Theta)}{\partial \Theta} \quad (2)$$

SGD では、学習率をはじめとするハイパーパラメータや、モデルパラメータの初期値の調整が重要である。各層の入力が一つ前の層のモデルパラメータの影響を受け、この影響がネットワークが深いほど大きくなるため、学習が複雑化する。学習によってモデルパラメータが更新され、各層の入力分布が変わり学習の系が変わることで、内部共変量シフトが引き起こされる。この内部共変量シフトを軽減することで学習を安定化する手法の一つが Batch Normalization である。

ネットワークの一部を切り出して、式 (3) の損失関数を考える。

$$l = F_2(F_1(u, \Theta_1), \Theta_2) \quad (3)$$

$F_1, F_2$  は任意の変換関数、 $\Theta_1, \Theta_2$  は損失  $l$  の最小化により学習されるパラメータである。 $\Theta_2$  の学習は、入力を  $x = F_1(u, \Theta_1)$  とおき、

$$l = F_2(x, \Theta_2) \quad (4)$$

とすると、パラメータ  $\Theta_2$  の更新

$$\Theta_2 \leftarrow \Theta_2 - \frac{\alpha}{m} \sum_{i=1}^m \frac{\partial F_2(x_i, \Theta_2)}{\partial \Theta_2} \quad (5)$$

は、入力  $x$  の独立したネットワーク  $F_2$  と等価である。したがって、学習効果を高める入力分布の性質は、学習データとテストデータで同じ分布を持たせることと同様に、サブネットワークの学習に対して有効である。つまり、学習時間経過によらず  $x$  の分布を一定に保つことは有効であり、 $\Theta_2$  の再調整を不要とすることができる。

そこで、各学習ステップで Activation の白色化を行うことを考える。学習後のバイアス  $b$  を加算し、学習データに対して計算された Activation の平均で減算した結果で正規化する入力  $u$  を持つ層は  $\hat{x} = x - E[x]$  で示される。ここで、 $x = u + b$ 、 $E[x] = \frac{1}{N} \sum_{i=1}^N x_i$  である。勾配降下ステップが  $E[x]$  の  $b$  についての依存がないとき、 $b$  の更新は  $b \leftarrow b + \Delta b$ 、 $\Delta b \propto -\partial l / \partial [\hat{x}]$  で示され、 $u + (b + \Delta b) - E[u + (b + \Delta b)] = u + b - E[u + b]$  となる。

$d$  次元の入力  $x = (x^{(1)} \dots x^{(d)})$  を持つネットワークの層について

Batch Normalization では Activation の出力  $x$  とペアのパラメータ  $\gamma, \beta$  を用いて、

$$y = \gamma x + \beta \quad (6)$$

## 2.3 You Only Look Once

You Only Look Once(YOLO) は物体検出手法の一つで, 「検出」と「識別」を同時に行うことで, 処理時間の遅延を解消することを特長とする.

## 3 Section

...

### 3.1 Subsection

...

### 3.2 Subsection

...

## 参考文献

- [1] AI(人工知能) の歴史 | 時系列で簡単解説, <https://ledge.ai/history-of-ai/>
- [2] 「Deep Learning」の進化で AI を活用した新しいビジネスが生まれる, [https://www.nttcom.co.jp/comware\\_plus/img/201705\\_DeepLearning.pdf](https://www.nttcom.co.jp/comware_plus/img/201705_DeepLearning.pdf)
- [3] 人工知能は人間を超えるか - ディープラーニングの先にあるもの, <https://www.ipa.go.jp/files/000050258.pdf>
- [4] 初心者でもわかるディープラーニング - 基礎知識から AI との違い、導入プロセスまで細かく解説, <https://ainow.ai/2019/08/06/174245/>
- [5] ImageNet Large Scale Visual Recognition Challenge, <https://arxiv.org/abs/1409.0575>
- [6] 畳み込みニューラルネットワークの高精度化と高速化, <https://www.slideshare.net/ren4yu/ss-145689425>
- [7] 共変量シフト下での教師付き学習, [https://www.jstage.jst.go.jp/article/jnns/13/3/13\\_3\\_111/\\_article/-char/ja/](https://www.jstage.jst.go.jp/article/jnns/13/3/13_3_111/_article/-char/ja/)
- [8] Batch Normalization : ニューラルネットワークの学習を加速させる汎用的で強力な手法, [https://deepage.net/deep\\_learning/2016/10/26/batch\\_normalization.html](https://deepage.net/deep_learning/2016/10/26/batch_normalization.html)
- [9] Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, <https://arxiv.org/abs/1502.03167>
- [10] Optimization: Stochastic Gradient Descent, <http://deeplearning.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent/>
- [11] 【物体検出手法の歴史 : YOLO の紹介】, <https://qiita.com/mdo4nt6n/items/68dcda71e90321574a2b>
- [12] Object Detection YOLO v1 , v2, v3, <https://medium.com/@venkatakrishna.jonnalagadda/object-detection-yolo-v1-v2-v3-c3d5eca2312a>

- [13] You Only Look Once: Unified, Real-Time Object Detection, <https://arxiv.org/abs/1506.02640>
- [14] YOLO9000: Better, Faster, Stronger, <https://arxiv.org/abs/1612.08242>
- [15] YOLOv3: An Incremental Improvement, <https://arxiv.org/abs/1804.02767>