

機械学習関連メモ

Ryoichi Matsumoto

2023 年 8 月 11 日

目次

1	はじめに	1
2	Deep Learning	2
2.1	Deep Learning の歴史	2
2.2	エッジ AI	4
2.3	Deep Learning のモデル構造	6
2.4	Deep Learning の学習テクニック	8
3	Section	10

表目次

1	FLOPs vs Weights	5
---	------------------	---

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
5	モデルごとの FLOPs と Weights	6
6	LeNet アーキテクチャ	7
7	AlexNet アーキテクチャ	7
8	ZF Net アーキテクチャ	8

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。

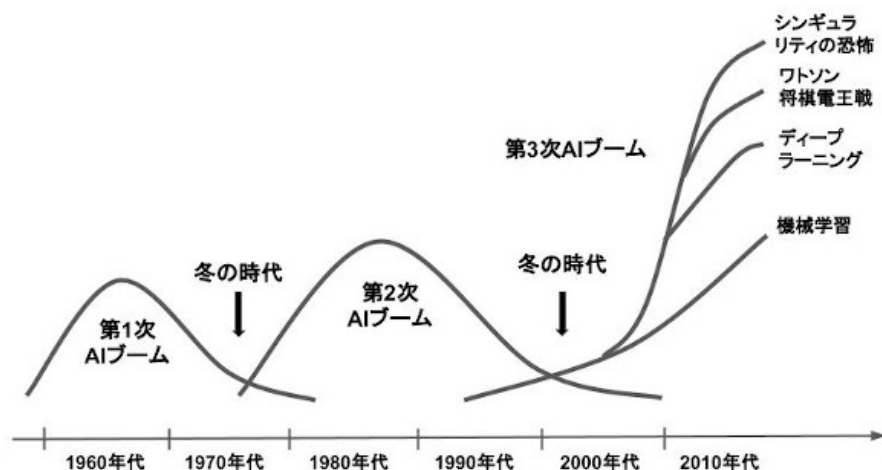


図 1 AI のブームと冬の時代

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) [6] 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% [7] を下回るアルゴリズムが登場した^{*1} (図 2, 図 3)。

^{*1} 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

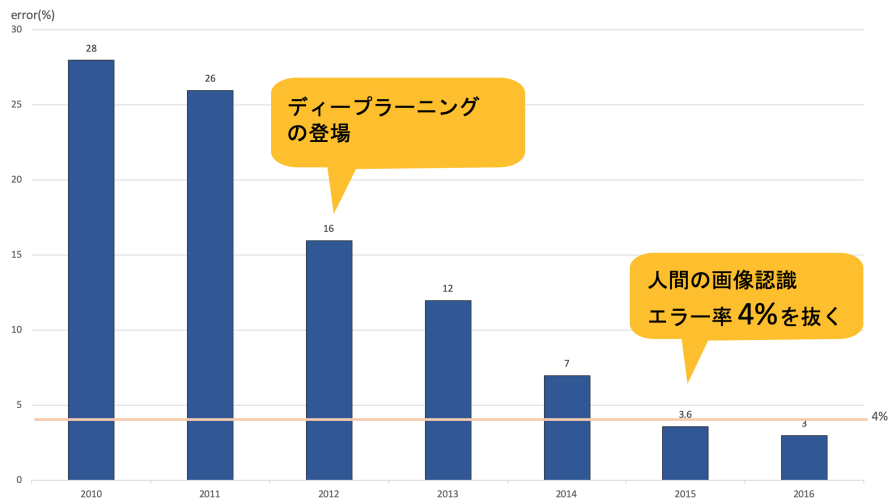


図2 ILSVRC の歴代優勝モデル (2010 年～2016 年)

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

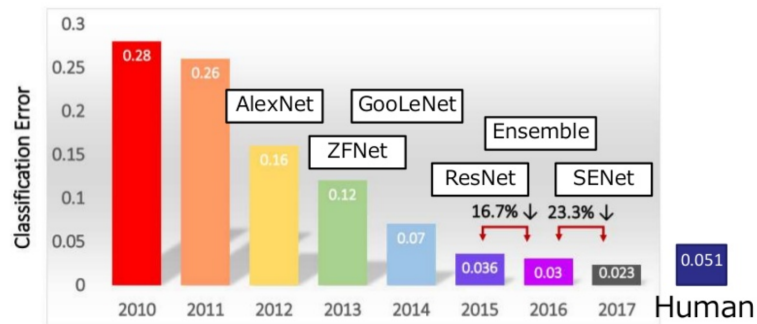
文字認識

2012 年以降の ILSVRC の優勝モデルは，図 4 に示す通り [8] *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

11

図 4 2012 年以降の優勝モデル

以降、様々なアーキテクチャが考案され、物体検出・音声認識・異常検知・画像/音声の生成など、多くの用途で活用が進んでいる。

2.2 エッジ AI

非常に多くの分野で活用されるようになった DeepLearning 技術だが、モデルの性能向上に伴い計算に必要な資源も増加している。

表 1 FLOPs vs Weights

Model Name	FLOPs	Weights
Xception	8357403496	22801424
VGG16	15470264320	138357544
VGG19	19632062464	143667240
ResNet50	3857973248	25530472
ResNet101	7570194432	44496488
ResNet152	11282415616	60117096
InceptionV3	5713216096	23800136
InceptionResNetV2	13155794016	55782920
MobileNet	568740352	4210088
MobileNetV2	300774272	3470760
DenseNet121	2834161664	7895208
DenseNet169	3359843328	13991080
DenseNet201	4291365888	19784872
NASNetMobile	563638816	5253240
NASNetLarge	23783414658	88556482
EfficientNetB0	388121280	5246532
EfficientNetB1	690912160	7732136
EfficientNetB2	998832224	9042426
EfficientNetB3	1836129536	12145936
EfficientNetB4	4413319168	19216416
EfficientNetB5	10306979360	30217048
EfficientNetB6	19136716704	42816272
EfficientNetB7	37868782912	66037240
EfficientNetV2B0	719342144	7079096
EfficientNetV2B1	1198640192	8069980
EfficientNetV2B2	1700157088	10013798
EfficientNetV2B3	3015809440	14249190
EfficientNetV2S	8375340032	21304616
EfficientNetV2M	24609813184	53847324
EfficientNetV2L	56127521408	118002696
ConvNeXtTiny	25491808974	27011848
ConvNeXtSmall	50982829662	48625672
ConvNeXtBase	90635773534	85772648
ConvNeXtLarge	203929679454	191474728
ConvNeXtXLarge	362540942942	339054312

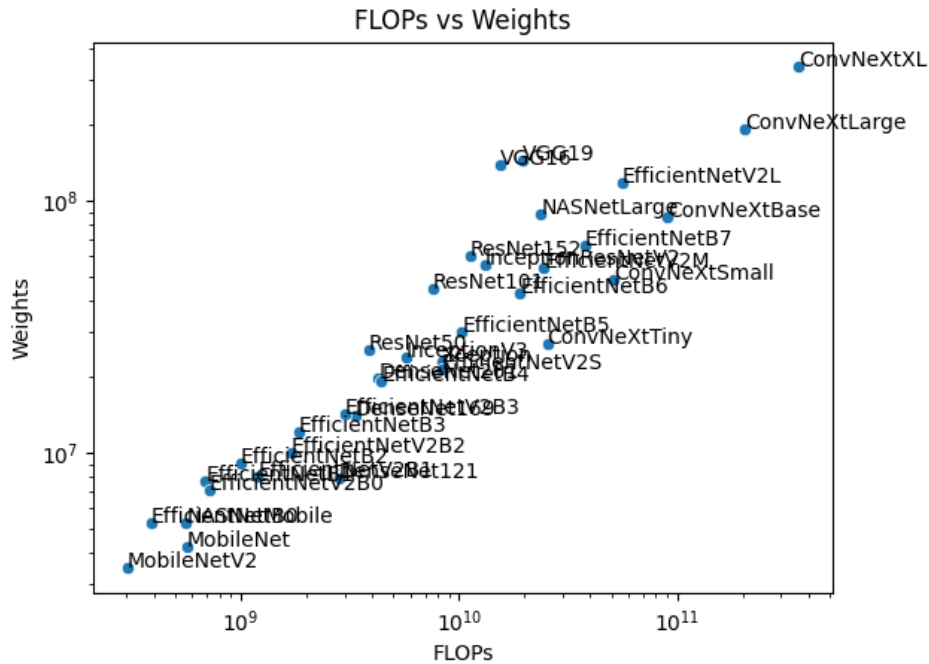


図5 モデルごとの FLOPs と Weights

計算機の資源、特に GPU やスーパーコンピュータの性能も向上しているが、スマートフォンやスマートカメラなど計算資源が限られているエッジデバイスでは、実行可能なモデルも制限される。

市場規模

業界団体 TinyML

企業

2.3 Deep Learning のモデル構造

2.3.1 LeNet

LeNet は Deep Learning の根幹ともいえる CNN アーキテクチャで、図 6 に示す構造を持つ [9] [10]。誤差逆伝播法による学習で、MNIST^{*3}などの文字認識で高精度を実現した。

^{*3} THE MNIST DATABASE of handwritten digits, <http://yann.lecun.com/exdb/mnist/>

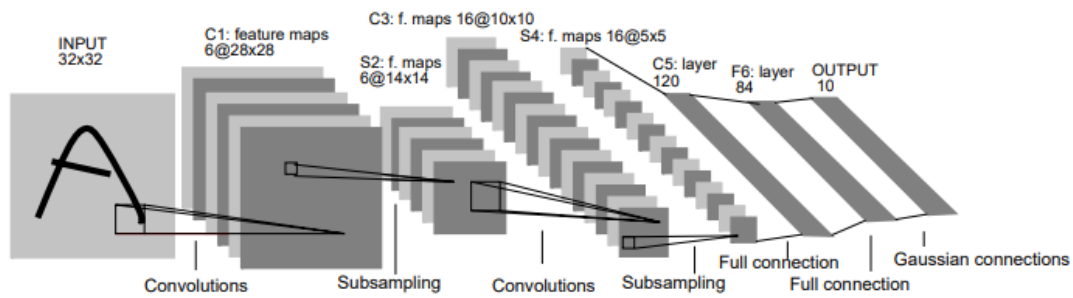


図 6 LeNet アーキテクチャ

2.3.2 AlexNet

2012 年の ILSVRC 優勝モデルで、図 7 に示す構造を持つ [11]。下記特徴を持つ。

- Rectified Linear Units(ReLU) による非線形変換導入で勾配消失問題の解決と学習の高速化。
- Local Response Normalization (LRN) による正規化で学習の安定化と高速化。当時, Local Contrast Normalization(LCN) と比較してより適切な正規化法と主張していたが、後の VGG Net で効果が認められなかったことや、Batch Normalization の登場により近年のモデルでは利用されていない。
- Overlapping pooling によるプーリング層でのオーバーラップにより過学習を提言。
- 全結合層の Dropout による学習時のニューロン無効化によりモデルの汎化性能を向上。
- Momentum SGD による最適化で学習を安定化。
- 重み減衰による正則化で過学習を抑制。
- 学習率減衰で学習の収束度を向上。

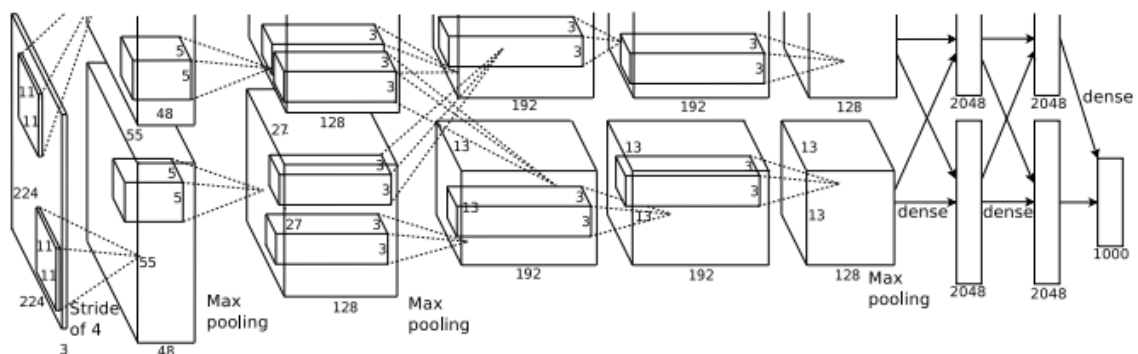


図 7 AlexNet アーキテクチャ

2.3.3 ZFNet

2013 年の ILSVRC 優勝モデルで図 8 に示す構造を持つ [12]. 下記特徴を持つ.

- Alex Net に対して, 第一層目のカーネルサイズを 11x11 から 7x7 に変更することで, 高周波及び低周波の情報量を抑制
- Alex Net に対して, 第一層目のストライドを 4 から 2 に変更することで, 第二層目に生じるエイリアシングを削減

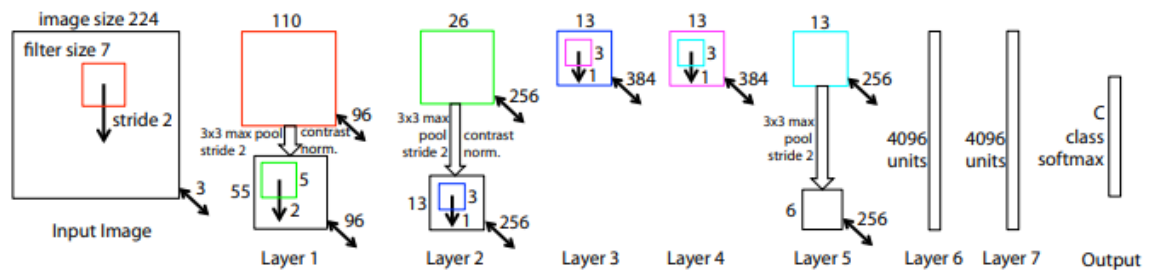


図 8 ZF Net アーキテクチャ

2.3.4 GoogLeNet

T.B.D

2.3.5 ResNet

T.B.D

2.3.6 SNet

T.B.D

2.3.7 YOLO(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.

2.4 Deep Learning の学習テクニック

2.4.1 Batch Normalization

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バ

ラメータ学習法, モデル選択法等が用いられる [13] 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 [14] [15].

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 [16]. 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 Θ を, loss を最小化するように最適化する.

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

$x_{1...N}$ 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 $x_{1...m}$ 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, Θ_1, Θ_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. Θ_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 Θ_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, Θ_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 γ, β を用いて,

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

3 Section

T.B.D

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
- [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] [サーベイ論文] 畳み込みニューラルネットワークの研究動向, http://mprg.jp/data/MPRG/F_group/F188_uchida2017.pdf
- [6] IMAGENET Large Scale Visual Recognition Challenge (ILSVRC) 2017 Overview, http://image-net.org/challenges/talks_2017/ILSVRC2017_overview.pdf
- [7] ImageNet Large Scale Visual Recognition Challenge, <https://arxiv.org/abs/1409.0575>
- [8] 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>
- [9] GradientBased Learning Applied to Document Recognition, http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf
- [10] Understanding and Implementing LeNet-5 CNN Architecture (Deep Learning), <https://towardsdatascience.com/understanding-and-implementing-lenet-5-cnn-architecture-deep-learning-a2d531ebc342>
- [11] ImageNet Classification with Deep Convolutional Neural Networks, <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- [12] Visualizing and Understanding Convolutional Networks, <https://arxiv.org/abs/1311.2901>
- [13] 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/
- [14] 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
- [15] Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, <https://arxiv.org/abs/1502.03167>

- [16] Optimization: Stochastic Gradient Descent, <http://deeplearning.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent/>
- [17] 【物体検出手法の歴史 : YOLO の紹介】, <https://qiita.com/mdo4nt6n/items/68dcda71e90321574a2b>
- [18] Object Detection YOLO v1 , v2, v3, <https://medium.com/@venkatakrishna.jonnalagadda/object-detection-yolo-v1-v2-v3-c3d5eca2312a>
- [19] You Only Look Once: Unified, Real-Time Object Detection, <https://arxiv.org/abs/1506.02640>
- [20] YOLO9000: Better, Faster, Stronger, <https://arxiv.org/abs/1612.08242>
- [21] YOLOv3: An Incremental Improvement, <https://arxiv.org/abs/1804.02767>