INFORMATION TECHNOLOGY CENTER
THE UNIVERSITY OF TOKYO

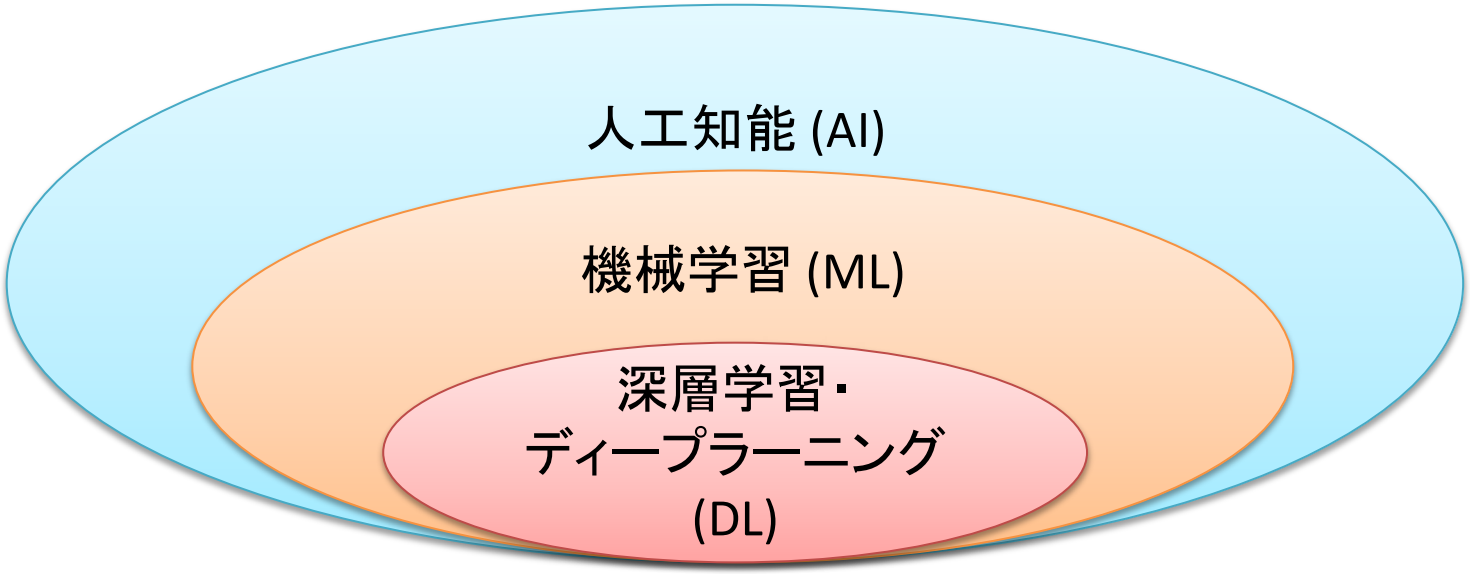
機械学習とその応用

スパコンを使った大規模・高速の学習に向けて

THE UNIVERSITY OF TOKYO

機械学習とは

- 機械学習 (ML: Machine Learning)
  - コンピュータで人間の学習に相当する仕組みを実現したもの
  - 人工知能 (AI: Artificial Intelligence)と同義で使われることが多いが、本来AIはもっと広い意味
- 深層学習・ディープラーニング (DL: Deep Learning)
  - 多数の層から成るニューラルネットワークを用いて行う機械学習の手法
  - DL技術の進歩が現在の人工知能ブームを支えている



<u>機械学習</u>:
明示的な指示を用いることなく、その代わりにパターンと推論に依存して、特定の課題を効率的に実行するためにコンピュータシステムが使用するアルゴリズムおよび統計モデルの科学研究である。(Wikipedia)

広がるAI応用とGPUの利用

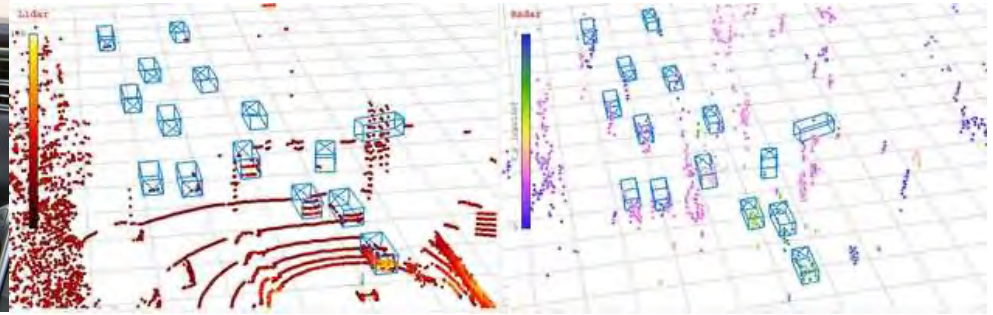
- 自動運転
  - 物体認識、レーン識別、自車位置把握、経路選択
- 自動翻訳
  - Google翻訳、DeepL
- 音声認識、自然言語処理
  - Amazon: Alexa
  - Apple: Siri
  - Google: Googleアシスタント
  - etc.

GPU

- 本来、画像処理のためのプロセッサ
- 良好な電力性能・コスト性能から、アクセラレータとしてスパコンに導入
- 行列演算が得意
  - 深層学習にも頻出

⇩

- 深層学習への適用で劇的に進化
- GPUに深層学習向けの機能も追加
  - テンソルコア
  - 半精度(FP16)などの低精度・超高速演算

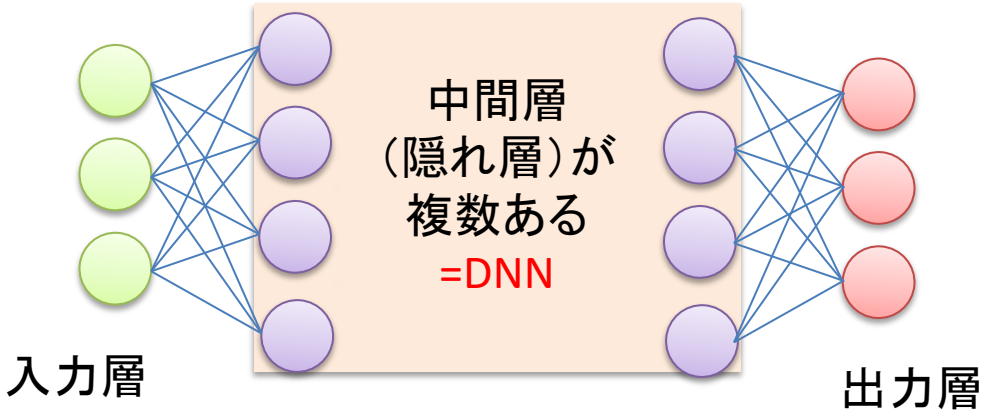


"Google Self-Driving Car" by smoothgroover22 is licensed under CC BY-SA 2.0

Automobile Italia/Flickr[CC BY 2.0]

ASTYX HIRES2019 DATASET CC-BY-NC-SA 4.0

ディープニューラルネットワークとその種類



"Data Set" by IBM Research is licensed under CC BY-ND 2.0

- 解きたい問題に合わせてネットワークの選択が必要
  - 畳み込み: CNN (Convolutional ...)
    - 分類、検出
    - 入力：猫の写真 =>出力：「ネコです」(ネコ 99.5%, ...)
  - 回帰型: RNN (Recurrent ...)
    - 時系列データ、（ひと昔前の）自動翻訳
  - オートエンコーダ
    - 次元削減、特徴抽出
  - 敵対的生成: GAN (Generative Adversarial Network)
    - 教師なし学習
    - 「それらしい」画像を生成

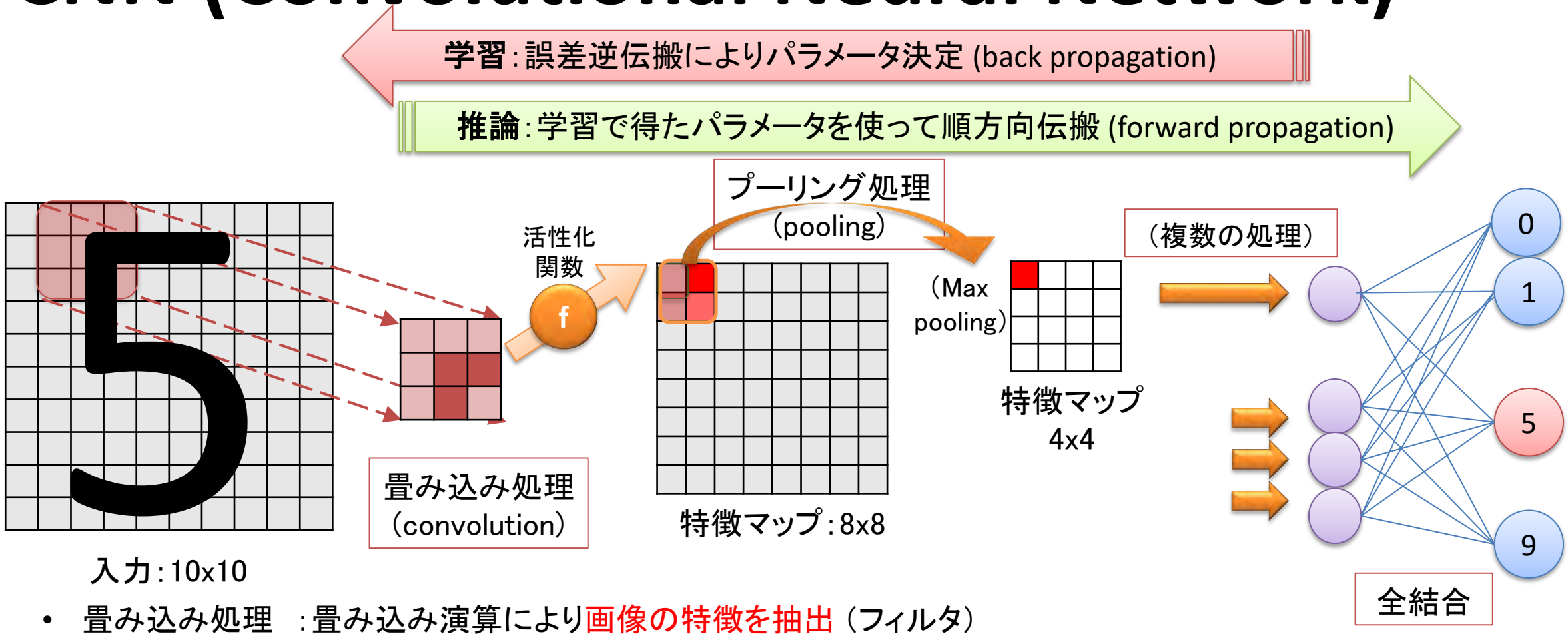
http://mattya.github.io/chainer-DCGAN/
Chainer-DCGANの開発者によるデモ
自分で試してみよう！

学習と推論

- 学習(training)：大量の学習データから特徴を抽出すること
  = ニューラルネットワークの結合の重み付けを最適に調整
  - 教師あり学習：正解のラベルをつけたデータ(教師データ)があらかじめ用意され、そこに近づけていく
  - 教師なし学習：教師データがない場合。クラスタリングによるグループ分け、異常検知など
  - 強化学習：報酬を与え一定の環境で試行錯誤させることで、報酬を最大化するように学習。AlphaGo Zeroなどのゲーム、ロボットの行動など 「パブロフの犬」
- 推論(inference): 学習済みモデルを用いて結果を導くこと

- 学習の方が桁違いに計算量が多く、時間がかかる

CNN (Convolutional Neural Network)



- 畳み込み処理　：畳み込み演算により画像の特徴を抽出（フィルタ）
- プーリング処理：物体の全体的な位置ずれを吸収

→物体の位置のずれや大きさの違いに影響されずに、画像全体の特徴を抽出する

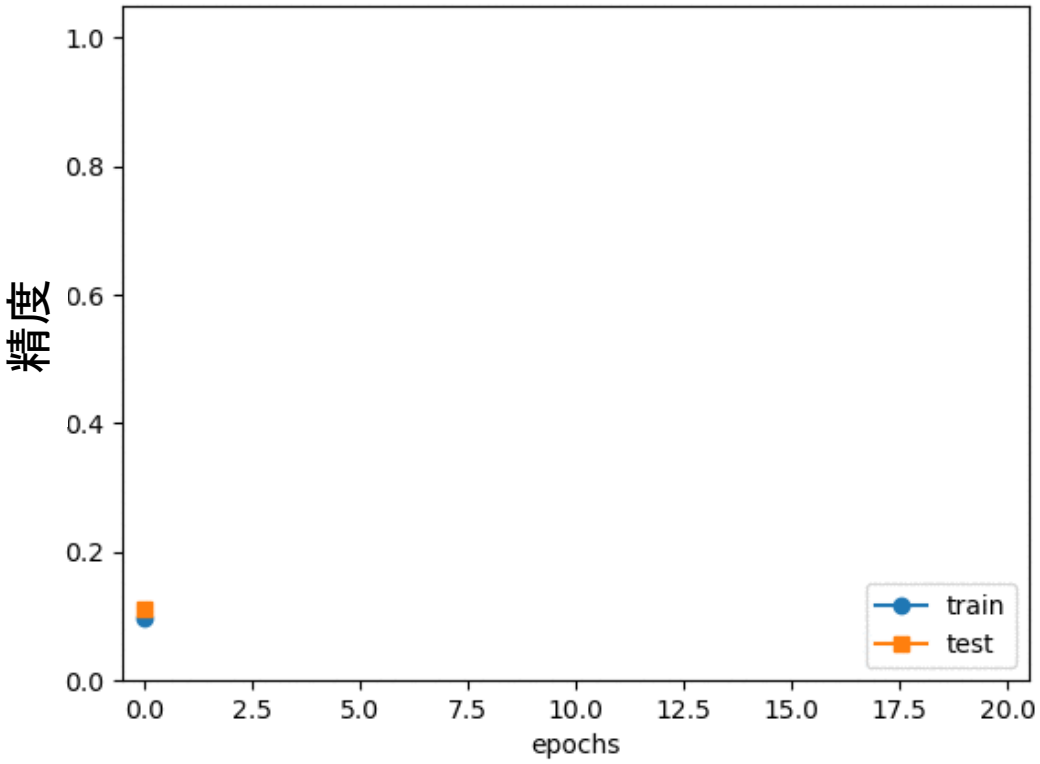
例題：手書き数字認識

- MNIST (Mixed National Institute of Standards and Technology database)
  - 学習用画像60,000枚、テスト画像10,000枚、正解ラベルつき
  - Deep Learningでまず最初にやる例題

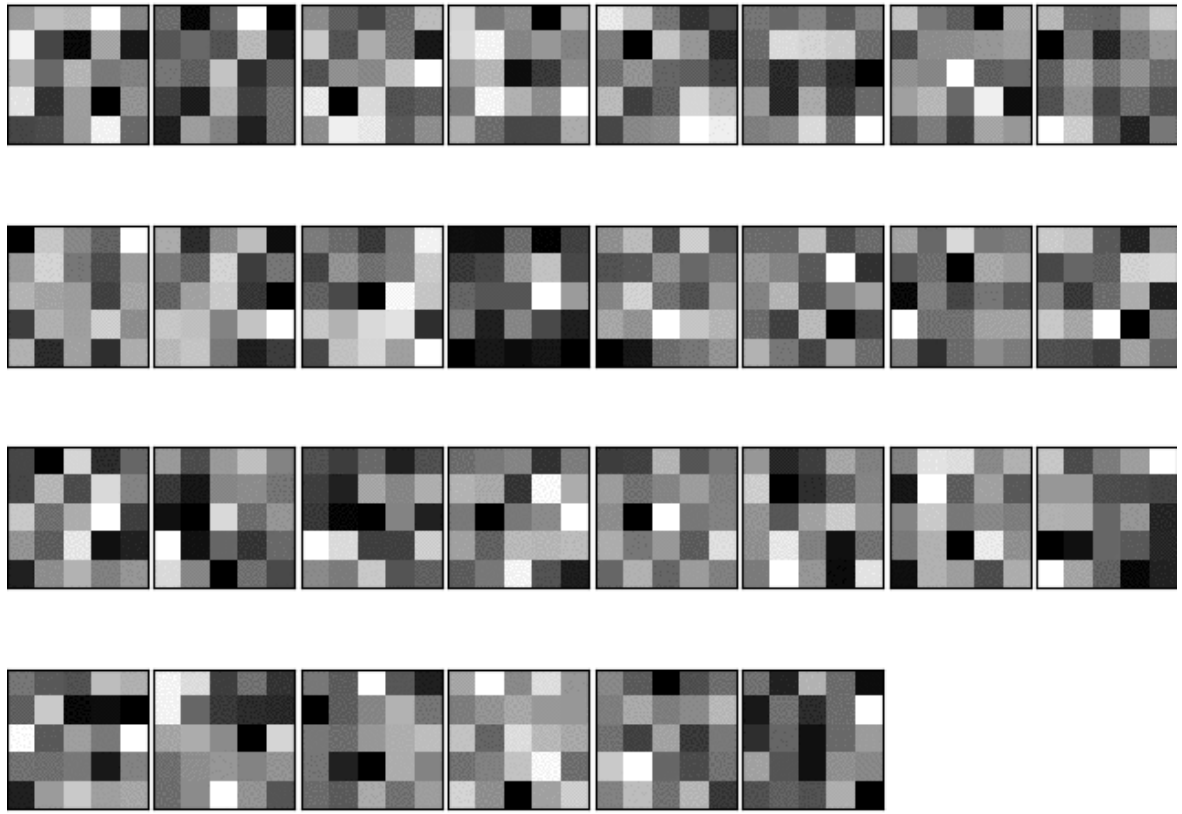
← 学習用画像の一部

単純なCNNを用いた場合

- 畳み込み層：1層、5x5のフィルタ
- 20回の学習で98.5%の精度



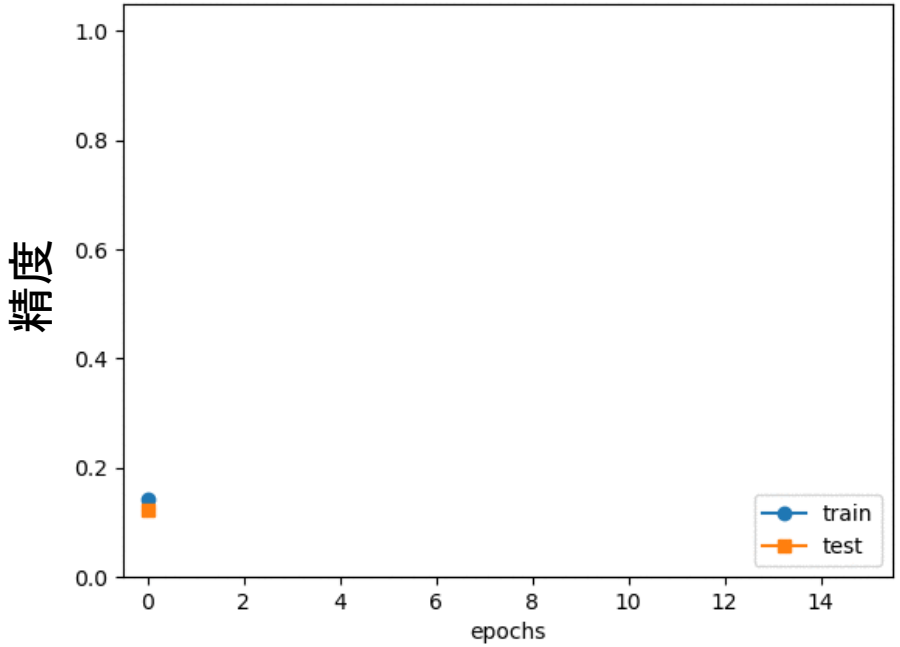
畳み込み層の重み



斎藤康毅: ゼロから作るDeep Learning, オライリー・ジャパン

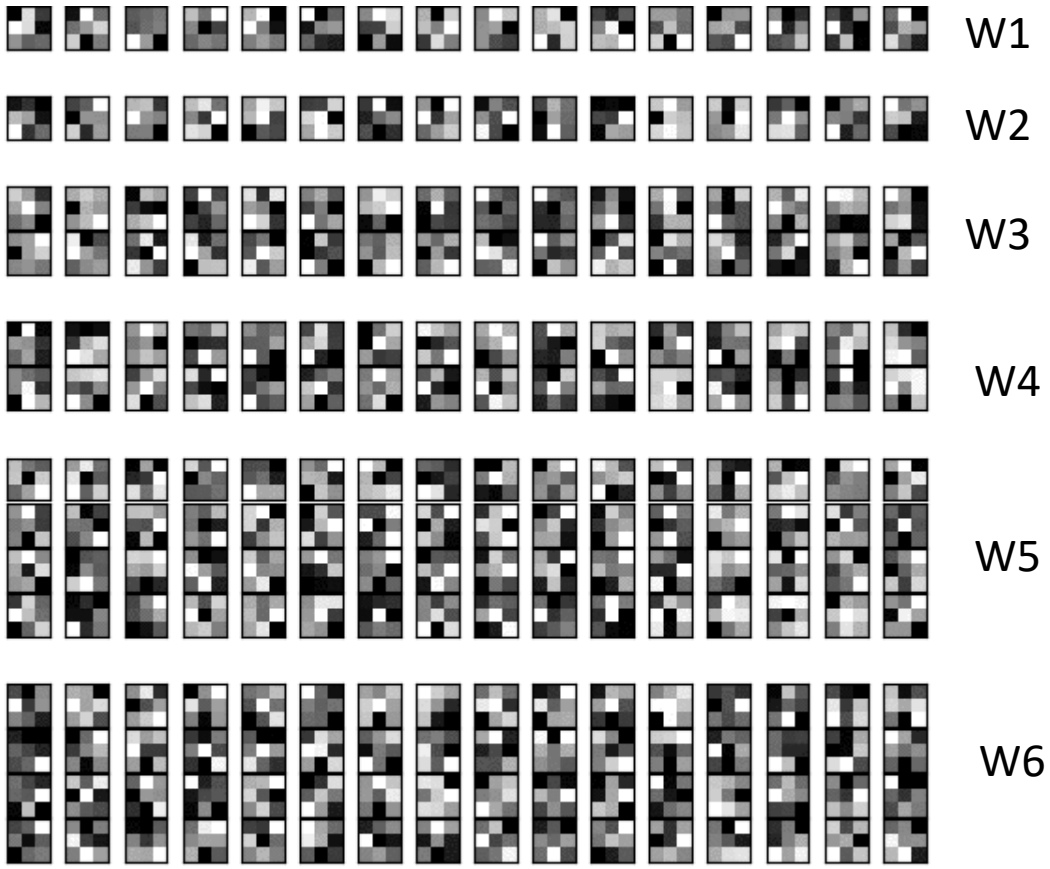
Deep Learning: よりDeepなCNN

- 畳み込み層: 6層、全結合層: 2層のディープニューラルネットワーク
  - 15回の学習で精度 99.2%



3x3のフィルタで構成

畳み込み層の重み



斎藤康毅: ゼロから作るDeep Learning, オライリー・ジャパン

機械学習を支えるソフトウェア

- Python
  - プログラミング言語、処理系
  - 対話的に使える（インタプリタ）
  - データ解析の機能が豊富
  - 機械学習の多くはPythonを使って実現されている
    - 実際に内部では高速なライブラリを使用

主な機械学習フレームワークと主な開発者

- TensorFlow
  - Google
- PyTorch
  - Facebook
- Chainer
  - プリファードネットワークス（日本）
  - いち早く分散学習を可能に（Chainer-MN）
  - すでに新規開発は停止、PyTorchに合流
- Neural Network Console
  - ソニー
- Horovod
  - Uber
  - 分散学習向け（TensorFlow, PyTorch等と組み合せ）

良い結果を得るためには

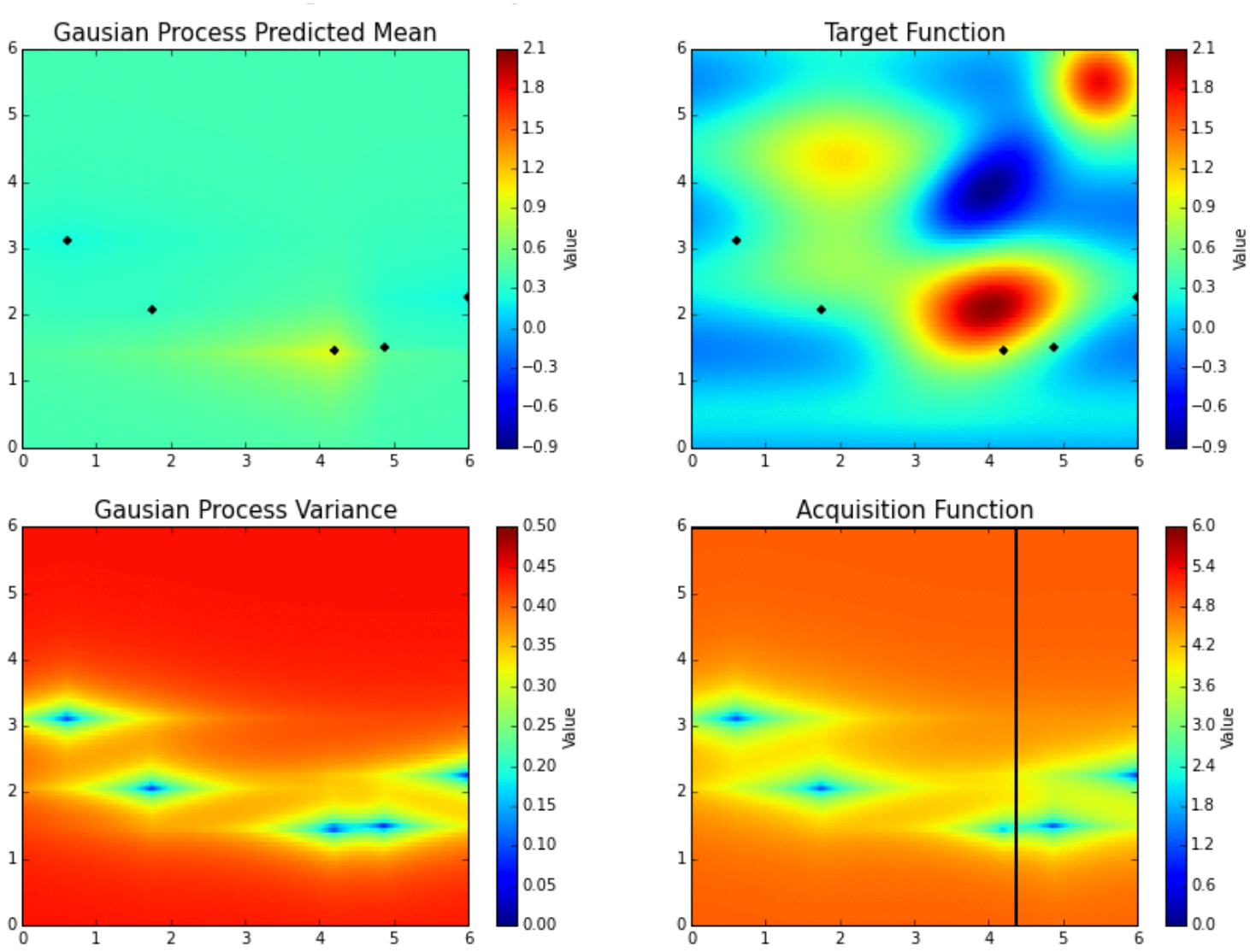
- ハイパーパラメータの最適化が重要
  - ハイパーパラメータ：ニューラルネットワークを構成する設定
  - 多数の項目が存在
  - 少し変えるだけでも精度が劇的に変化
- ハイパーパラメータ探索手法
  - グリッドサーチ：しらみつぶし、事実上不可能、無駄
  - ランダムサーチ：当てずっぽう、運任せ

  →賢く探す：ベイズ的最適化

- ベイズ的最適化
  - 前回までの探索結果を基に次に調べるパラメータを決定

    ➡比較的少ない試行回数でより優れたハイパーパラメータが選べる
    - 調べても無駄なところは省略
    - 可能性のあるところを探し出す
  - Bayesian Optimization for python
    - https://github.com/fmfn/BayesianOptimization
  - プリファードネットワークスOptuna
    - https://preferred.jp/ja/projects/optuna/
    - https://optuna.org

ベイズ的最適化の動作

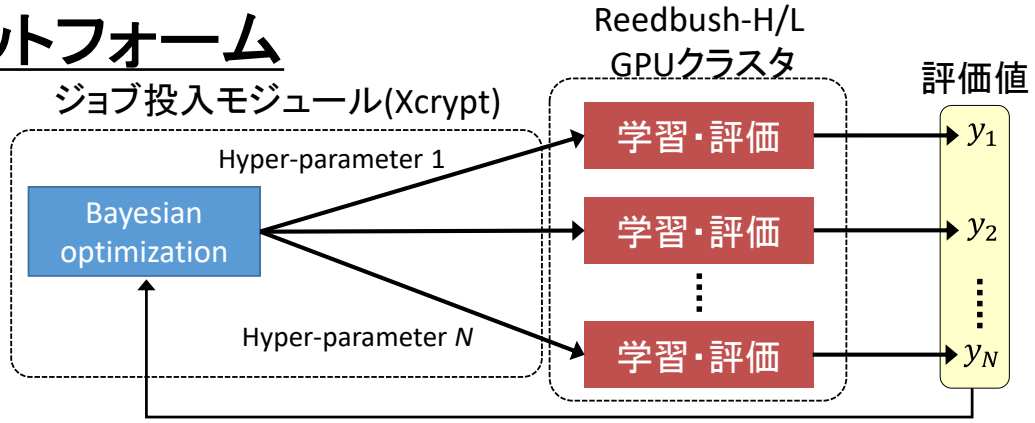


- 右上：目標関数
- 左上：予測した関数
- ステップを重ねるごとに適切に探索して目標に近づいている

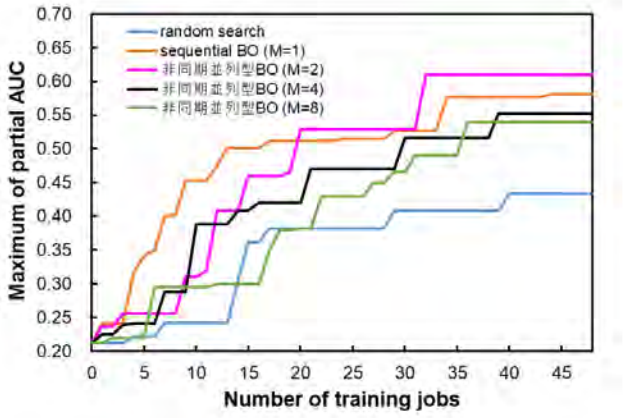
https://github.com/fmfn/BayesianOptimization/blob/master/examples/bayesian_optimization.gif

バイオインフォマティックス：医療画像処理
東大病院等

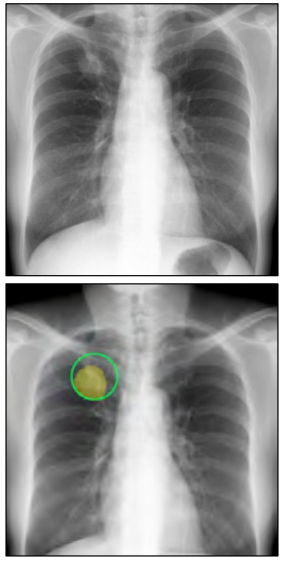
深層学習自動チューニングプラットフォーム



胸部X線写真の肺腫瘤検出



学習ジョブ数と評価値(partial AUC)の最大値との関係

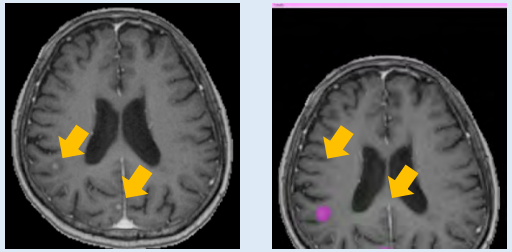


上：元画像、下：検出結果
（黄、緑丸：病変領域）

開発中のソフトウェア

頭部造影MR画像の転移性脳腫瘍検出

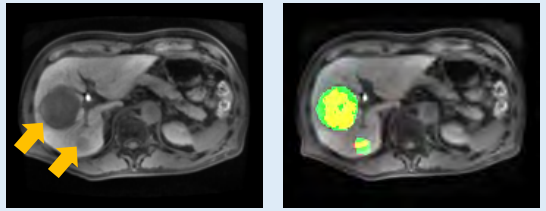
村田, JAMIT2018



検出結果例
左：元画像、右：検出結果(マゼンダ)

造影MR画像の肝結節性病変検出

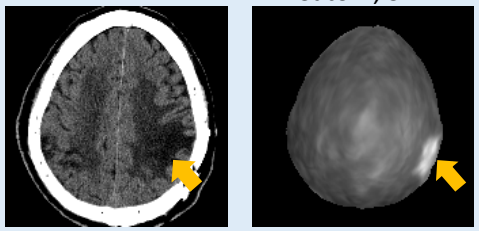
Takenaga T, CARS 2018



検出結果例
（肝細胞がん、左：元画像、右：検出結果）
●:検出、●:過検出、●:未検出

頭部救急CT画像の異常検知

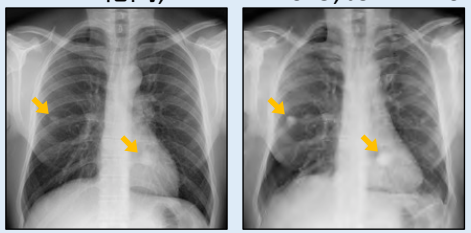
Sato D, SPIE MI 2018



脳梗塞症例
（左：元画像、右：異常度マップ）

胸部X線画像の異常強調

花岡, MAIAMI 2019, JSAIMI 2020

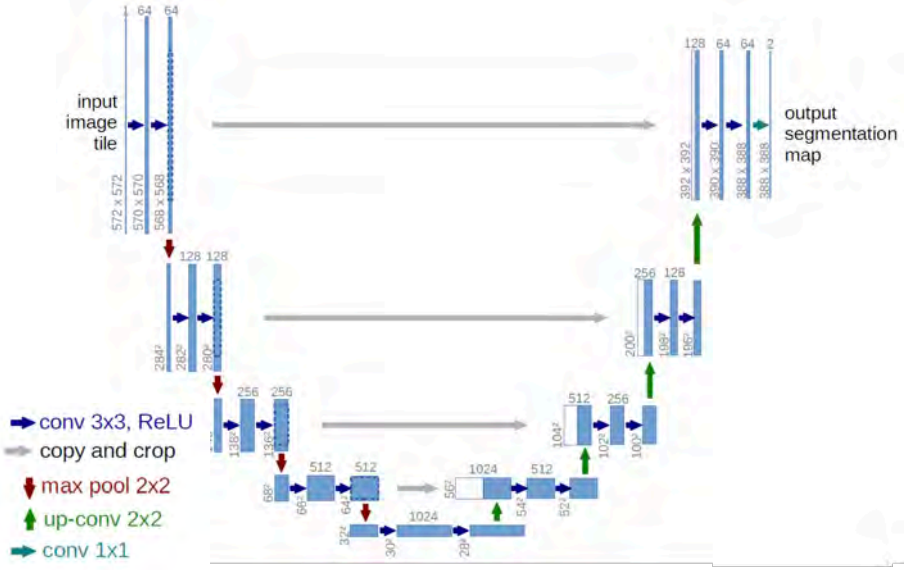


強調画像例
（左：元画像、右：強調結果、矢印：肺腫瘤）

Nomura Y, J Supercomput. 20 Jan 2020 (Epub ahead)

大規模化、高速化へ向けて

- ニューラルネットワーク全体を学習するためには、コンピュータのメモリに収まらないといけない
  - 多数の層を扱う深層学習では、膨大なパラメータを記憶する必要がある：DenseNet201では、なんと201層!!
  - 巨大なデータセット：例えばCT画像は1枚で数GB
  - 一方、データセンター向けのGPUでも32GB程度しかない
- 学習には長時間が必要
  - ImageNetの学習(100万画像)に1 GPUでは1週間以上かかる
- 分散学習（並列化）：大規模ネットワークの扱い＋高速化が可能に
  - データ並列
  - モデル並列



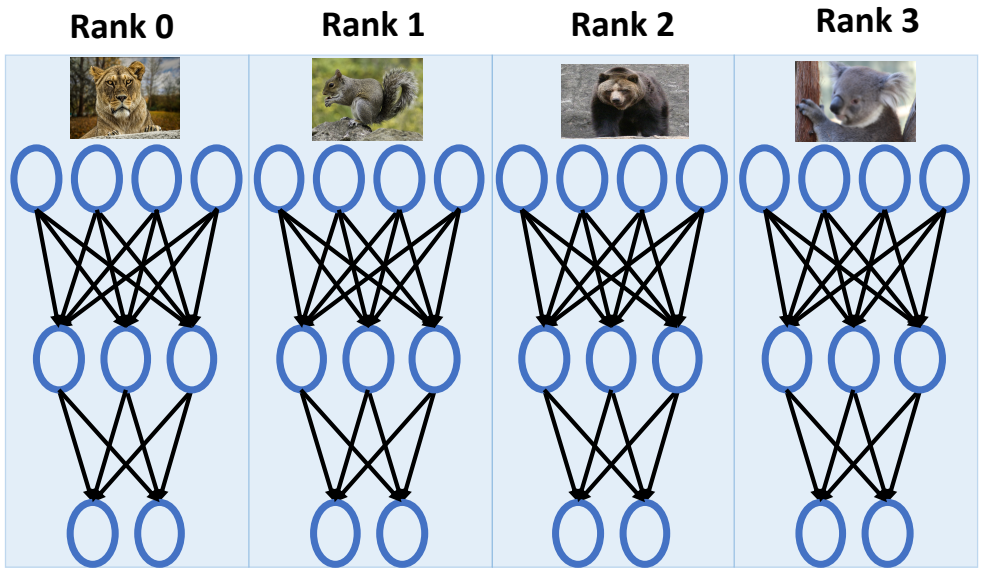
医用画像向けによく使われる U-Net

Olaf Ronneberger, et al., U-Net: Convolutional Networks for Biomedical Image Segmentation.", 2015

分散学習（並列化）の戦略

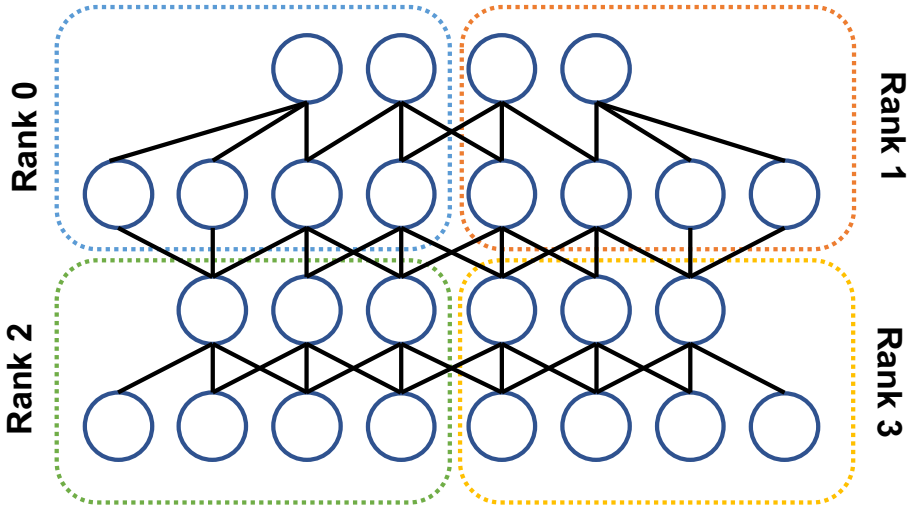
データ並列学習

- データを分割
  - 全ワーカーがモデルの複製を持つ
  1. 異なるデータを処理して個々にモデルを更新
  2. 互いのパラメータを持ち寄って平均を取る
  - 問題点：ワーカー数を増やすと精度がなかなか向上しない（ラージミニバッチ問題）

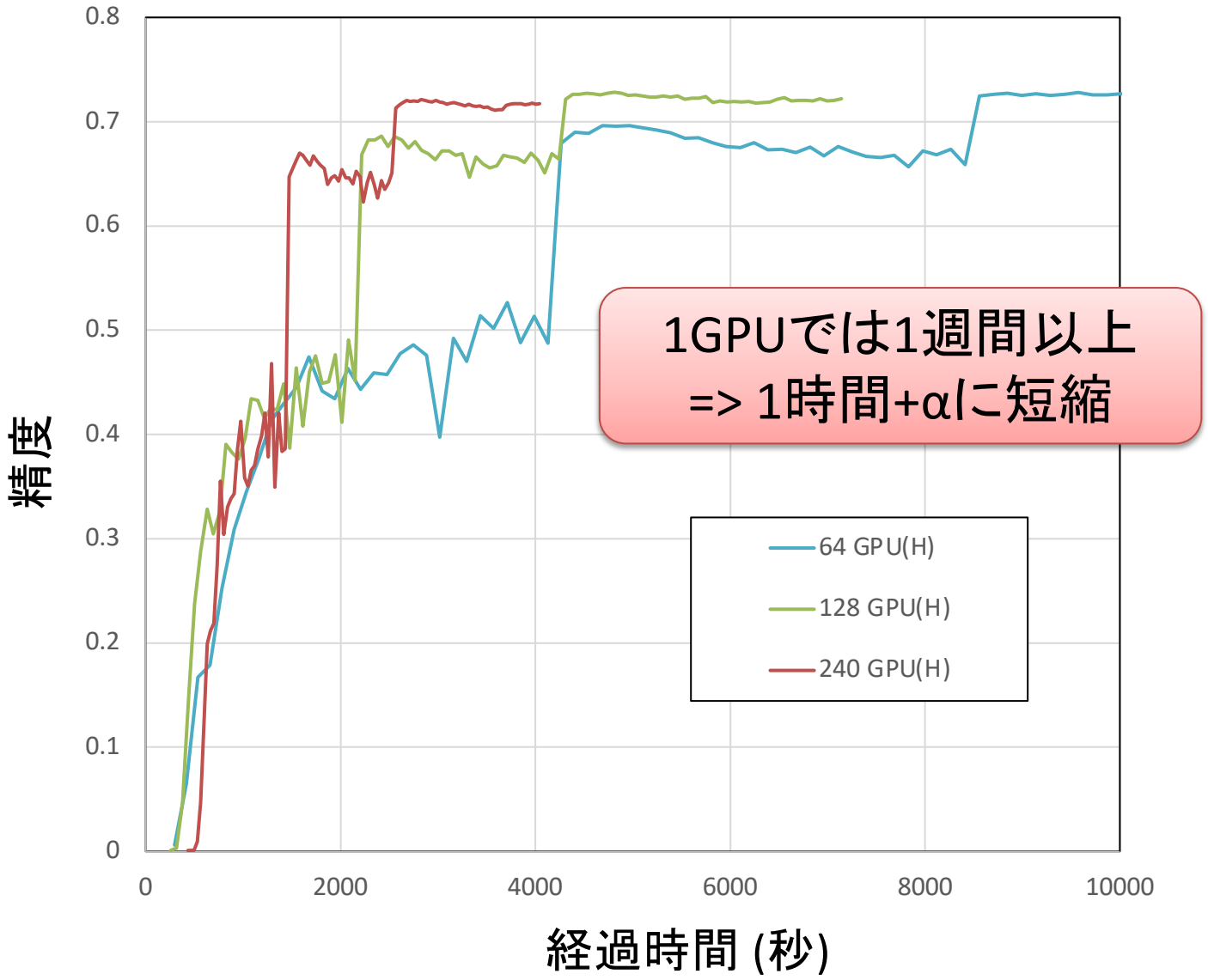


モデル並列学習

- モデル（ネットワーク）を分割
  - 1台では不可能な巨大なネットワークを実現
  - どうやって分割すれば効率がいいか？
    - 通信がなるべく少なくなるように
    - 並列性が活きるように、負荷が均等になるように
  - かなり難しい、熟練が必要➔ 自動化試みる研究が盛んに



実例：Reedbush-HでのImageNet学習（データ並列）



# GPUs	100 epoch実行時間	精度
32	6時間で終了せず	(72%)
64	3時間58分20秒	72.0%
128	1時間59分02秒	72.2%
240	1時間7分24秒	71.7%

- ResNet50
- 100エポック実行
- 64, 128, 240 GPU (RB-H)
- ChainerMN 1.0.0
  - OpenMPI 2.1.1, NCCLv2
  - Chainer 3.1.0
  - python 3.6.1, CUDA 8, cuDNN7, cupy 2.1.0

参考

- 総務省：令和元年版　情報通信白書
  - https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r01/index.html
  - 第1部　特集　進化するデジタル経済とその先にあるSociety 5.0
    第1章　ICTとデジタル経済はどのように進化してきたのか
    第3節　ICTの新たな潮流
    2　AIに関する動向https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r01/html/nd113200.html
- ゼロから作るDeep Learning
  - http://www.oreilly.co.jp/books/9784873117584/
- その他多数

まとめ

- 機械学習の概要
  - 応用分野
  - 分類
- 応用
  - 医用画像処理
- 大規模化、高速化へ向けて
  - 分散学習、データ並列とモデル並列

Thank you for watching