深層学習の原理を明らかにする理論の試み

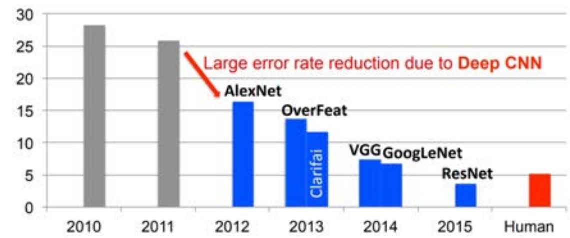
統計数理研究所 数理・推論研究系
学習推論グループ 助教

今泉允聡

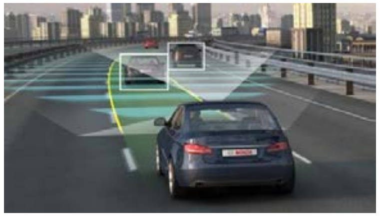
導入：深層学習の登場

基礎研究 ~2000 → **ブレイクスルー** 2012 → **実用化の進展** 2016~

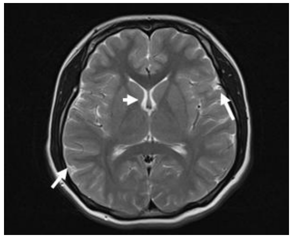
技術的課題 → 技術発展



高精度を発揮
画像識別精度: 75% → 96%



自動運転[1]



医療診断[2]

高精度：信頼できる

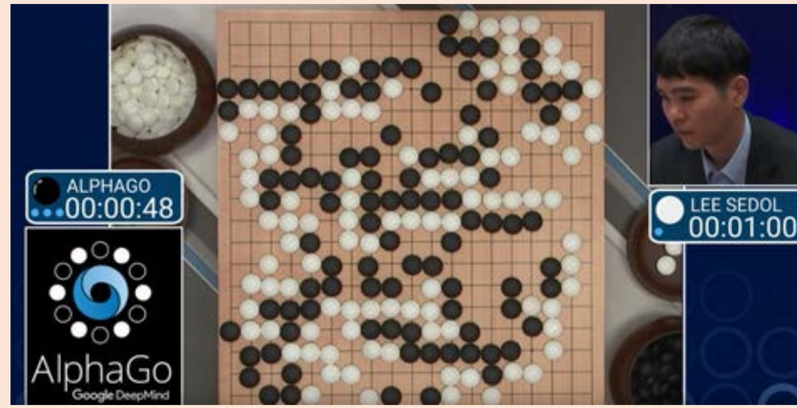
↓

社会の多様な領域へ応用

深層学習の成功例

AlphaGo (DeepMind)

- 囲碁で人間超え
  - 世界トップ棋士に勝利



BERT (Google)

- 言語テストで高得点
  - 人間: 91.2%
  - BERT: 93.2%

文章

In early 2012, NFL Commissioner Roger Goodell stated that the league planned to make the 50th Super Bowl "spectacular" and that it would be "an important game for us as a league".

文章に関する問い

Who was the NFL Commissioner in early 2012?
Ground Truth Answers: Roger Goodell | Roger Goodell | Goodell

いくつかのタスクで高い性能を発揮

理解の不在による壁

深層学習の運用にはまだ問題点が多い

膨大な計算コスト

うまい設定が
分からない
大量に試験しよう

計算がとても大変

ブラックボックスな挙動

失敗したが
原因は不明！

信頼できる
製品が作れない

実用化の進展には、**原理の理解**が必要

深層学習とは

深層学習の基本構造は関数

- 入力に対して、適切な出力を出すシステム

深層学習システムの中身

多層ニューラルネットワーク

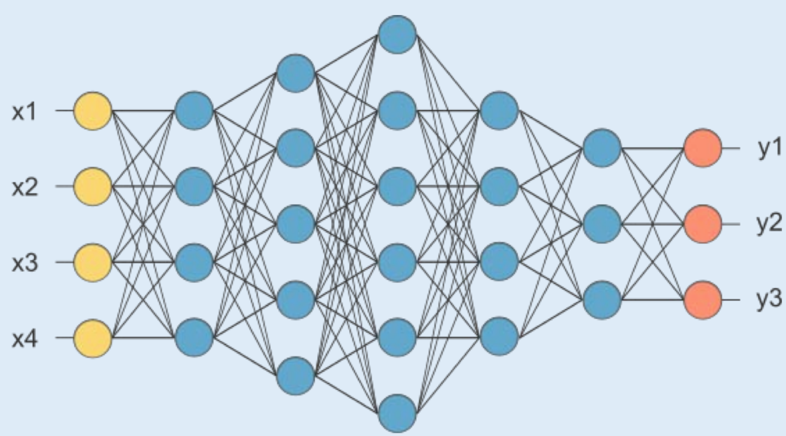
- 入力ベクトルを変換する関数のモデル

入力（例：画像）

変換

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

ベクトル x



多層ニューラルネットワーク

出力（例：情報）

これは
茶色い猫です

変換

$$y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

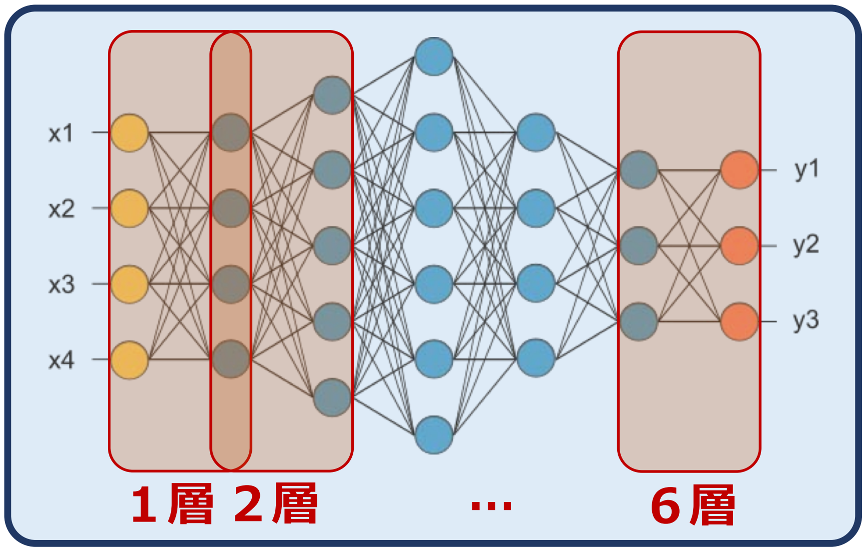
ベクトル y

深層学習システムの中身

ベクトルの変換を**層の数だけ**繰り返す

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

ベクトル x



$$y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

ベクトル y

ベクトルの変換

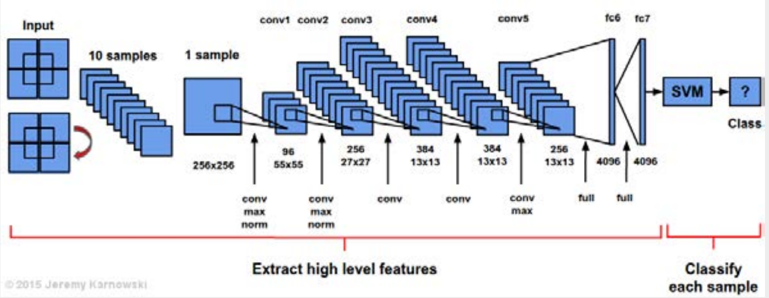
1層目	$z_1 = \eta(A_1 x + b_1)$
2層目	$z_2 = \eta(A_2 z_1 + b_2)$
⋮	⋮
6層目	$y = A_6 z_5 + b_6$

A: パラメタ (行列)
b: パラメタ (ベクトル)
η: 非線型変換

層を増やして巨大化するシステム

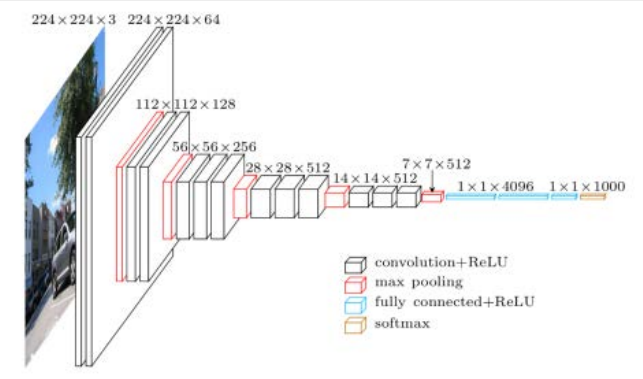
具体的な多層ニューラルネットワーク

AlexNet (トロント大)



層の数：８層
パラメタ数：６千万

VGG19 Net (オクスフォード大)



層の数：１９層
パラメタ数：１億

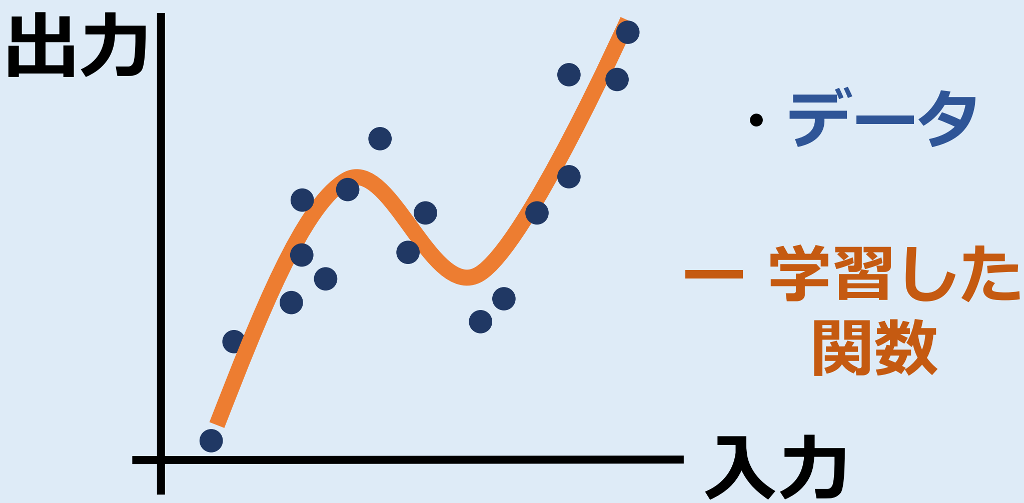
層の数が多い → パラメタも増える

膨大なパラメタはデータから学習

パラメタ：システムが機能するために必要

- データの構造を再現できるように学習

システムによる関数が
データに合うように学習



損失最小化

θ：パラメタ

(Y_i, X_i)：データ

$$\underbrace{\min_{\theta} \sum_i \left(Y_i - f_\theta(X_i)\right)^2}_{\text{損失}}$$

損失

＝システムによる関数とデータのズレ

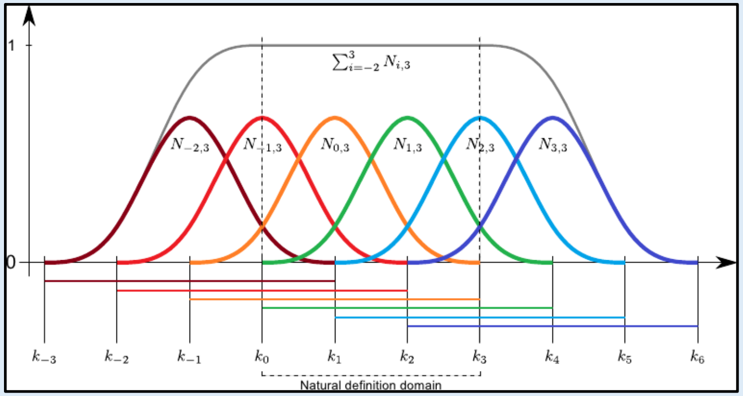
深層学習がもたらす謎

従来はどういう方法だった？

関数を表現する従来法はたくさんある

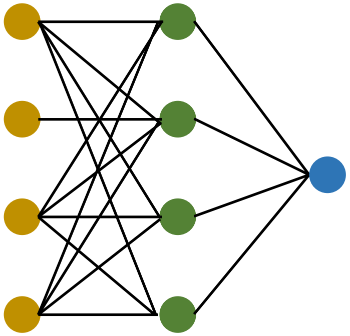
従来法

- フーリエ法
- スプライン法
- カーネル法
- 回帰木　　他多数

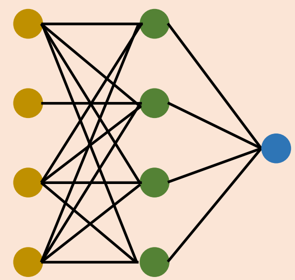


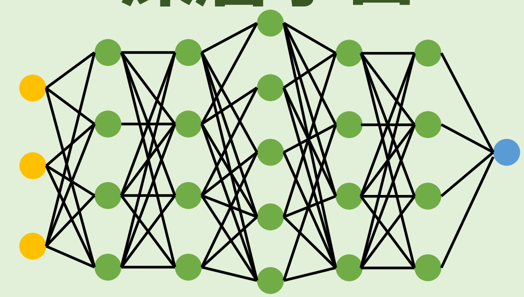
B-スプライン法による関数表現

- 従来法の特徴
  - データを1～2回変換する
    - 典型例：特徴写像 → 線形変換


従来法のネットワーク表現

従来法と深層学習の違い





	従来法	深層学習
層の数 (変換の回数)	１～２	３～１００以上
パラメタの数	数十 ～ 数千	数千 ～ 一億
性能	そこそこ	とても良い

層が多いなら、性能が良くなるのは当たり前？

謎(1/3): なぜ多層で性能が上がる？

理論　層数は少なくて良いと
数学的に証明されているよ

関数推定の最適性定理 (Stone (1980) など)
従来法（１～２層）で理論的に最適。

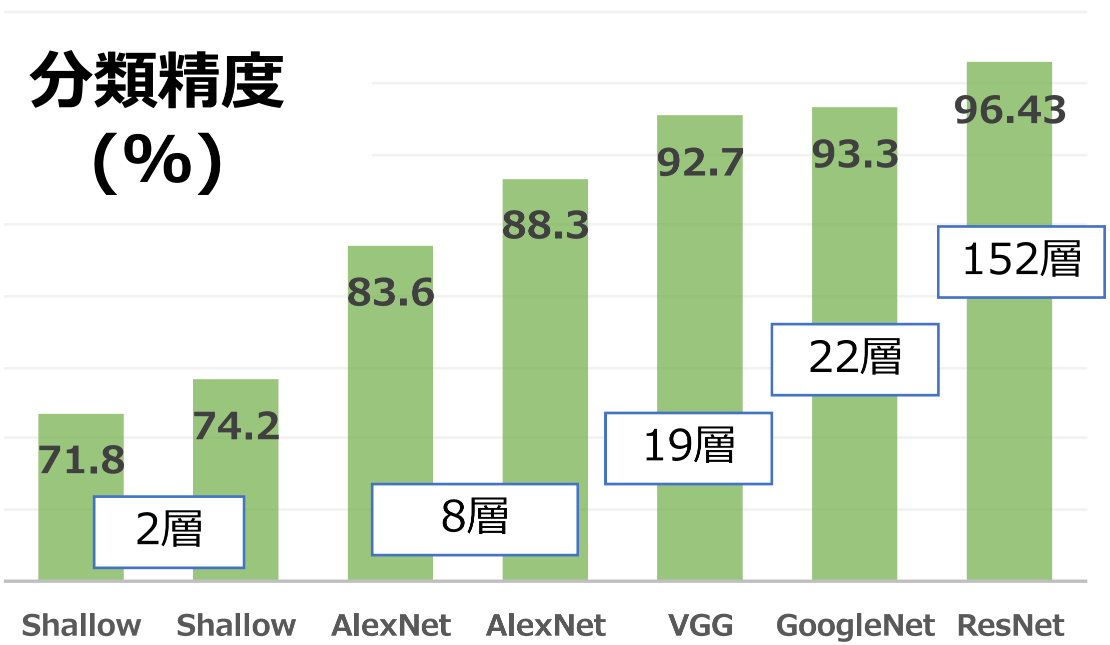

UC Berkley

普遍近似定理 (Cybenko (1989) など)
２層のニューラルネットワークで十分。


Prof. G. Cybenko

謎(1/3): なぜ多層で性能が上がる？

実際
多層にすると性能が向上するよ

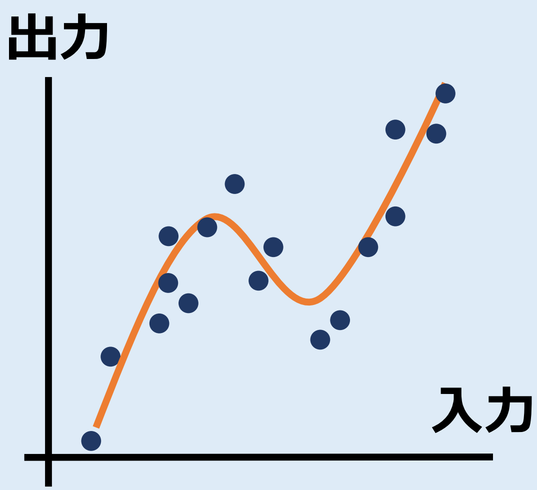


層が増えるほど高い精度を発揮

何層を使えば良い？
仕組みが分からないから
全部試すしかない…

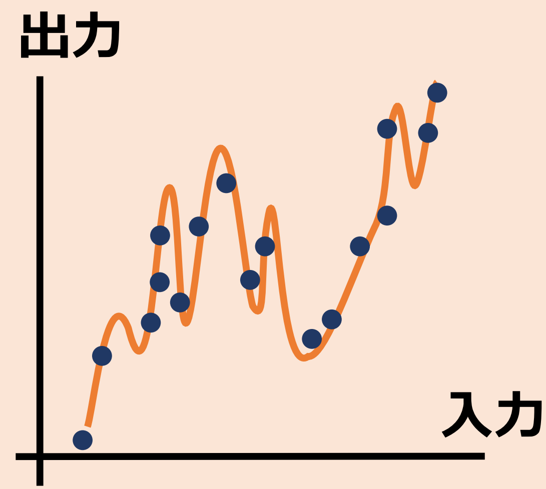
謎(2/3). 膨大なパラメタ数の謎

統計理論の(大)原則
大量のパラメタは精度を下げる！



適切な関数を学習

パラメタ数が
増えると…



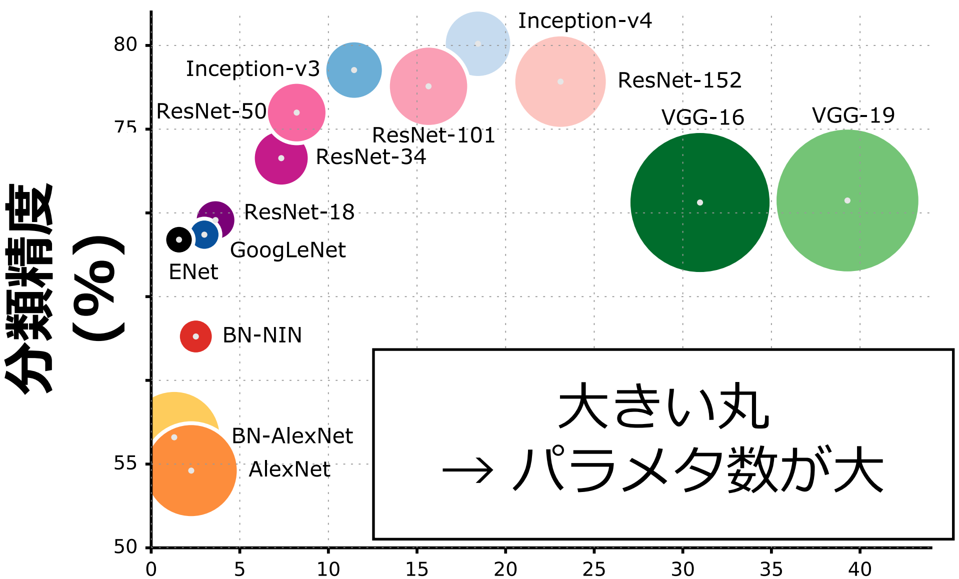
過適合した関数を学習
➡ **精度の低下**

- 既存の統計学はパラメタ数の削減に腐心…
  - 変数選択、スパース推定、正則化、適応化など

謎(2/3). 膨大なパラメタ数の謎

実際

大量パラメタでも精度は上がるよ



精度とパラメタ数の関係

Canziani+ (2017)

多ければ多い方が良い？
可能なら減らしたいけど
程度が分からない…

謎(3/3). なぜパラメタ学習ができる？

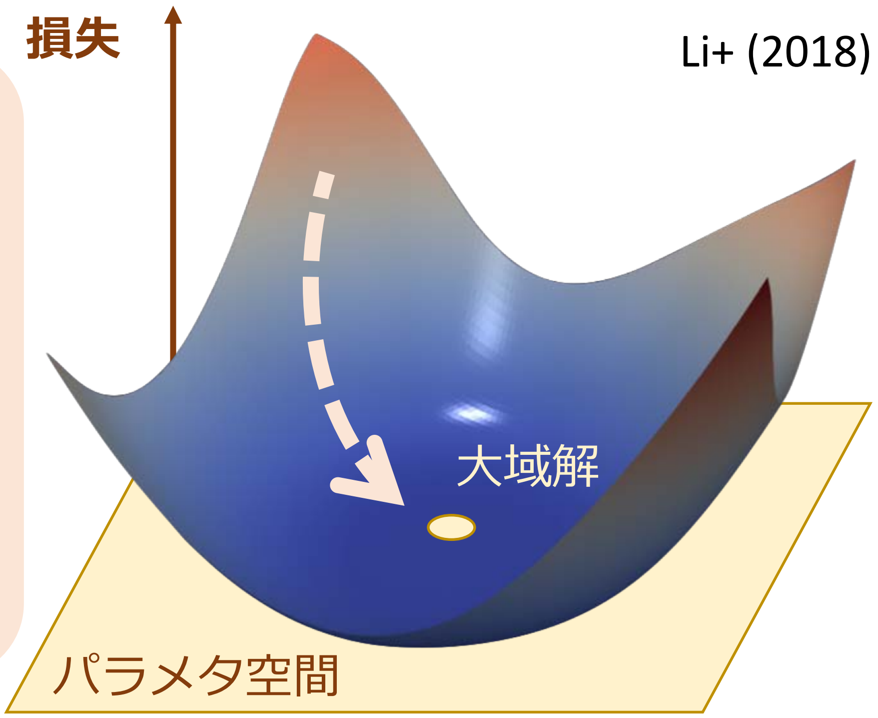
前提知識

パラメタ学習＝損失を最小にするパラメタ探し
＝**大域解**

従来法のパラメタ学習

・勾配 (損失の減少率) にしたがってパラメタを探索
・損失の斜面を下れば大域解が簡単に求まる

容易な学習が保証



勾配を下って大域解を発見する様子

謎(3/3). なぜパラメタ学習ができる？

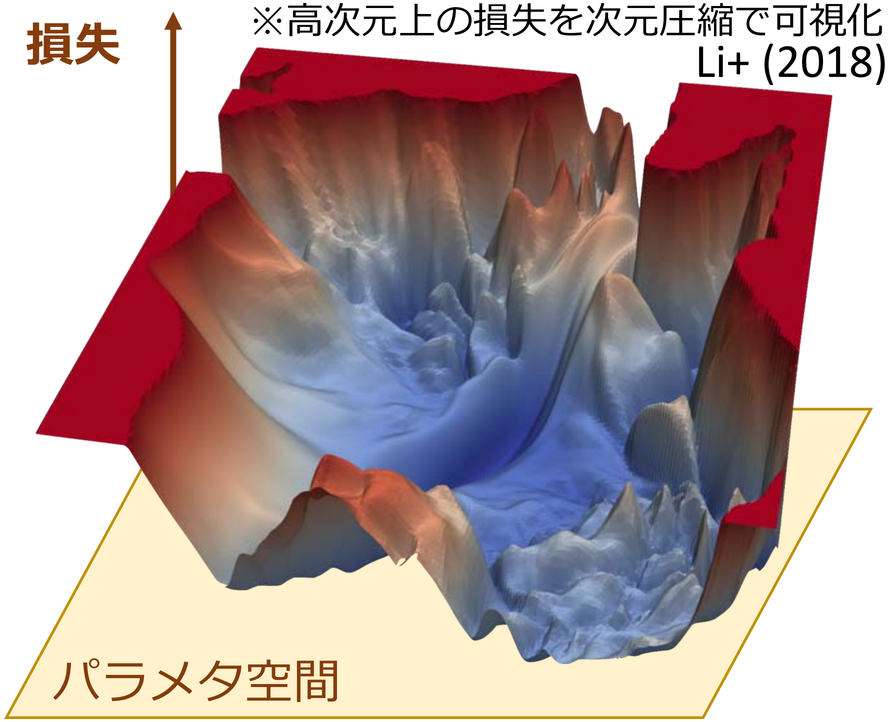
前提知識
パラメタ学習＝<u>損失を最小にするパラメタ</u>探し
＝**大域解**

<u>**深層学習**のパラメタ学習</u>

・多層 → 損失が複雑に
・パラメタが増えると
困難さは指数的に増加

何も保証されない

※高次元上の損失を次元圧縮で可視化
Li+ (2018)



深層学習の損失関数（非凸関数）

謎3/3. なぜパラメタ学習ができる？

理論　多層モデルのパラメタ学習は
難しくてできるはずがない

ギャップ

実際
でも最終的な精度は向上している

どういう時に上手くいくの？
学習の結果は信頼して良いの？

理論と実際のギャップ

1．多層の謎

なぜ層を増やすと性能が上がるのか？

2．大パラメタ数の謎

なぜ過適合による精度低下が無いのか？

3．パラメタ学習の謎

なぜ多層なのにパラメタ学習できるのか

原理究明のための理論の試み

理論と実際のギャップ

1.多層の謎

なぜ層を増やすと性能が上がるのか？

2.大パラメタ数の謎

なぜ過適合による精度低下が無いのか？

3.パラメタ学習の謎

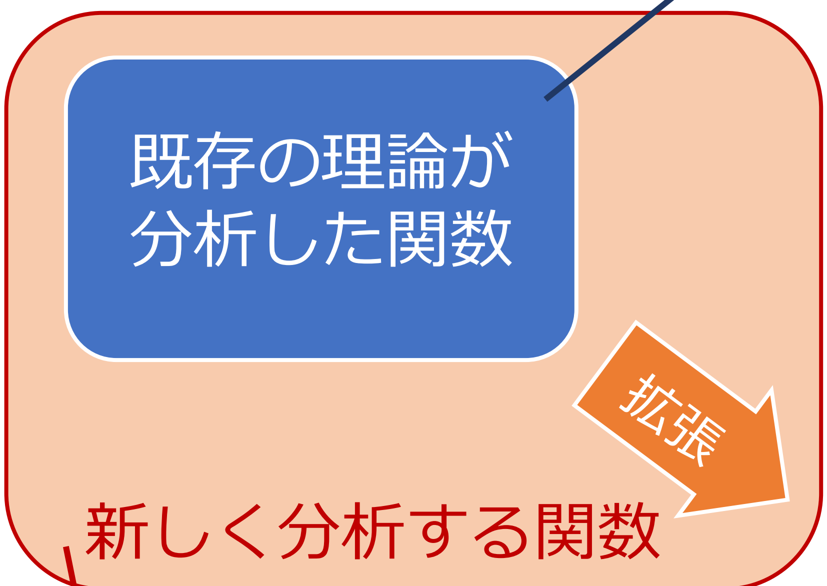
なぜ多層なのにパラメタ学習できるのか

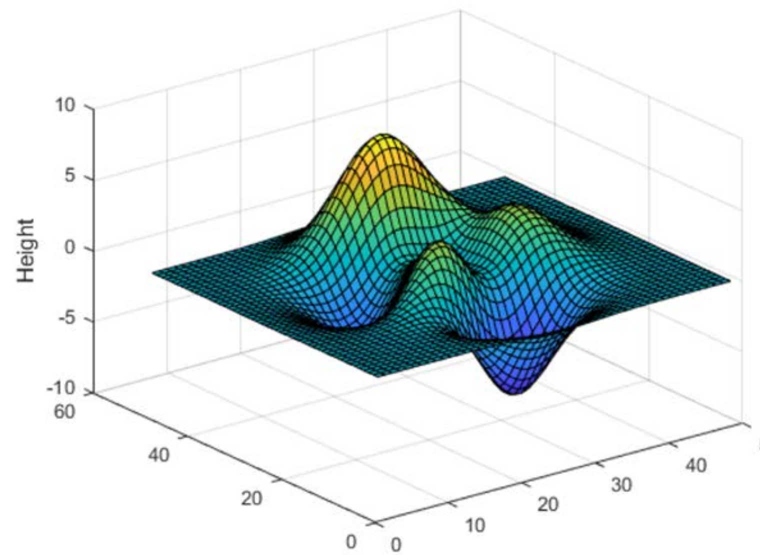
1. 多層構造が必要な関数

関心：表現したい関数と、多層構造との関係

- 考える関数の幅を広げる

<u>斉一的な性質を持つ関数</u>





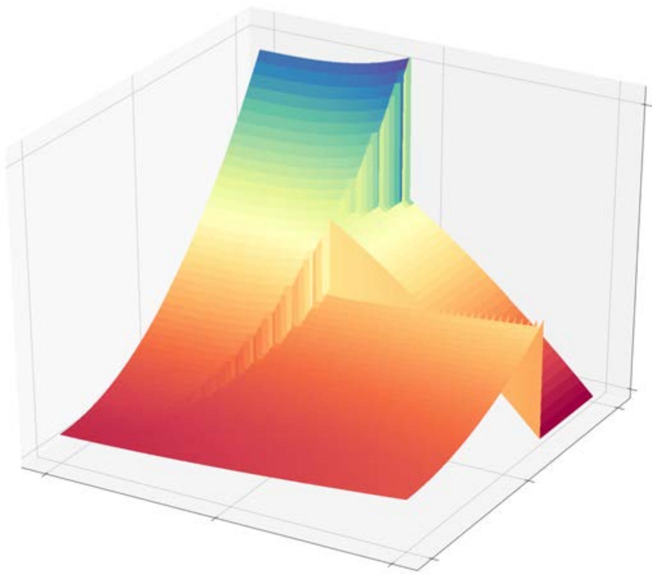
どこでも同じ性質 (例：滑らかさ)

<u>局所構造を持つ関数</u>

場所によって違う性質を持つ

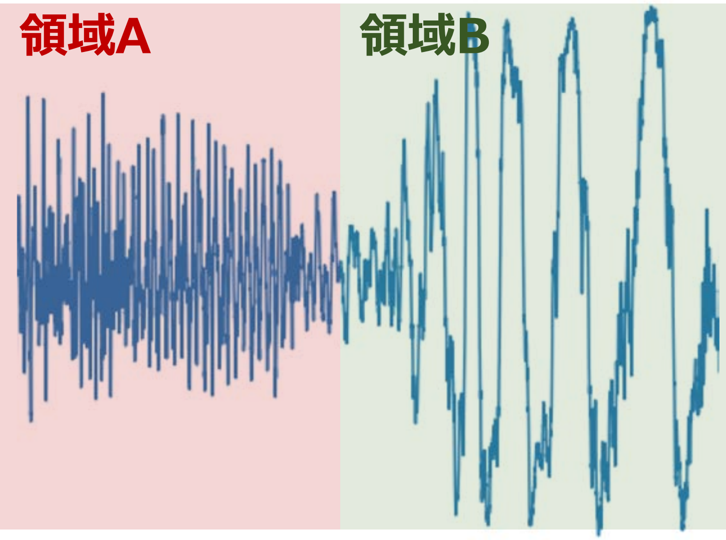
1. 多層構造が必要な関数

発見：多層は局所構造を持つ関数の表現に必要



分断された関数
(特異性を持つ関数)
例：相転移現象

Safran (2017), Imaizumi&Fukumizu (2019)

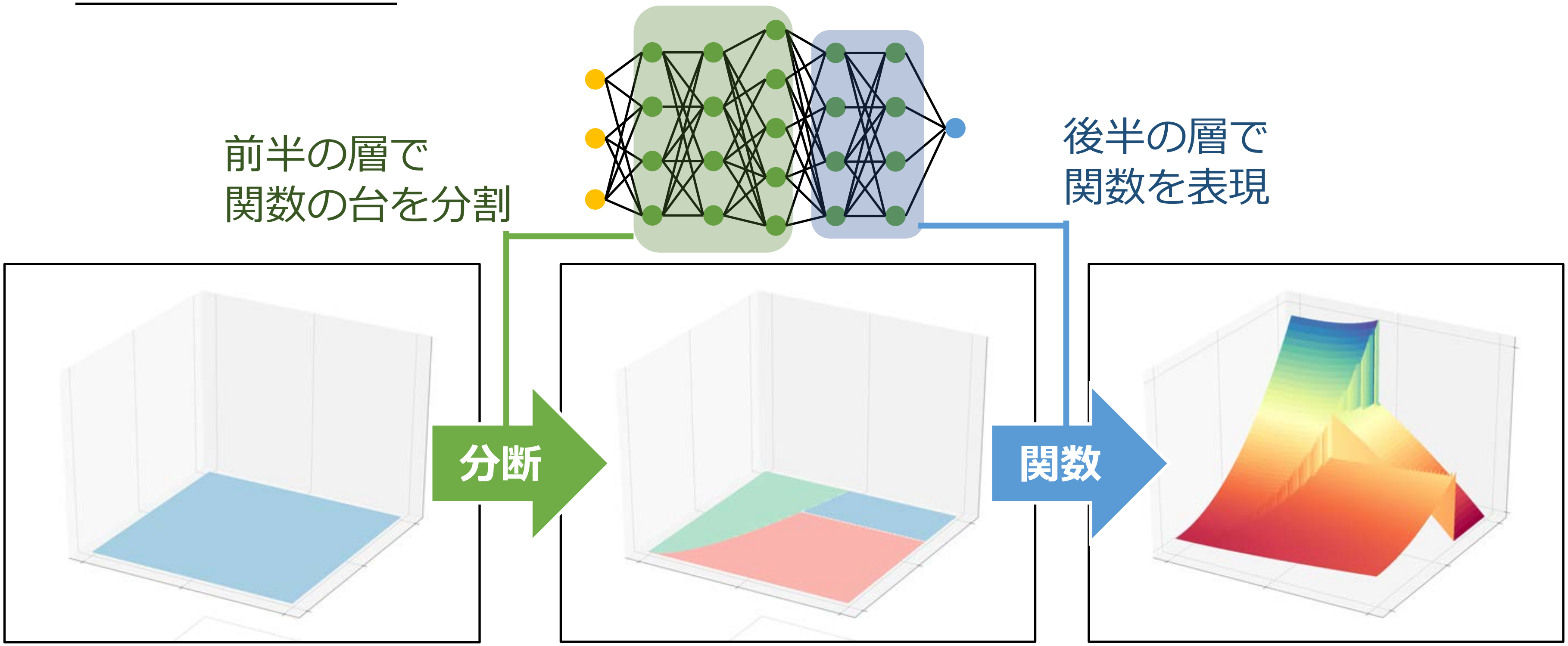


非斉一的に変動する関数
(Besov関数空間)
例：信号・音声

Suzuki (2019)

1. 多層構造が必要な関数

多層の役割： 局所的な性質の割り振り



理論的結果：局所構造のある関数を表現するには
深層学習が適していることを証明

理論と実際のギャップ

1.多層の謎

なぜ層を増やすと性能が上がるのか？

2.大パラメタ数の謎

なぜ過適合による精度低下が無いのか？

3.パラメタ学習の謎

なぜ多層なのにパラメタ学習できるのか

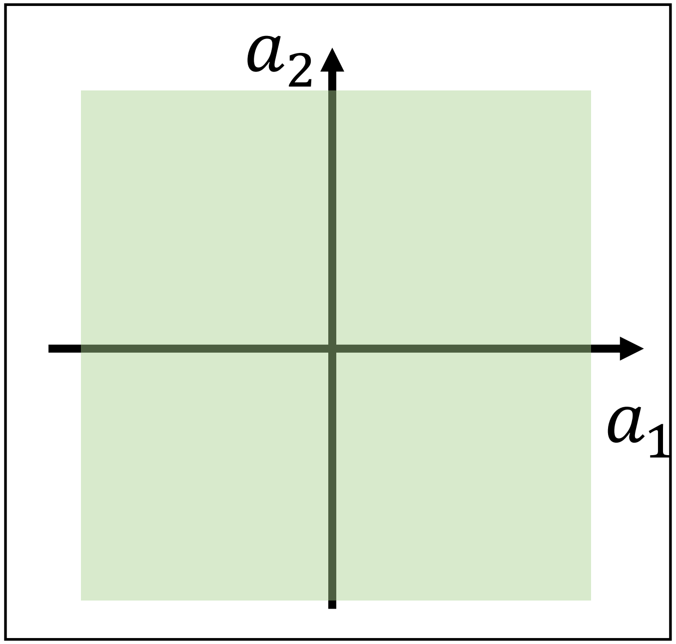
2. モデル自由度の再評価

疑問：なぜ多いパラメタ数は良くないのか？

既存の統計理論 (VC次元・R複雑性)

モデル自由度 ≈ パラメタ数
(不安定性)

不安定なモデル ➡ 過適合しやすい



既存理論でのモデル自由度
≈ パラメタが動く領域の大きさ

パラメタ a_1, a_2
が動く最大領域

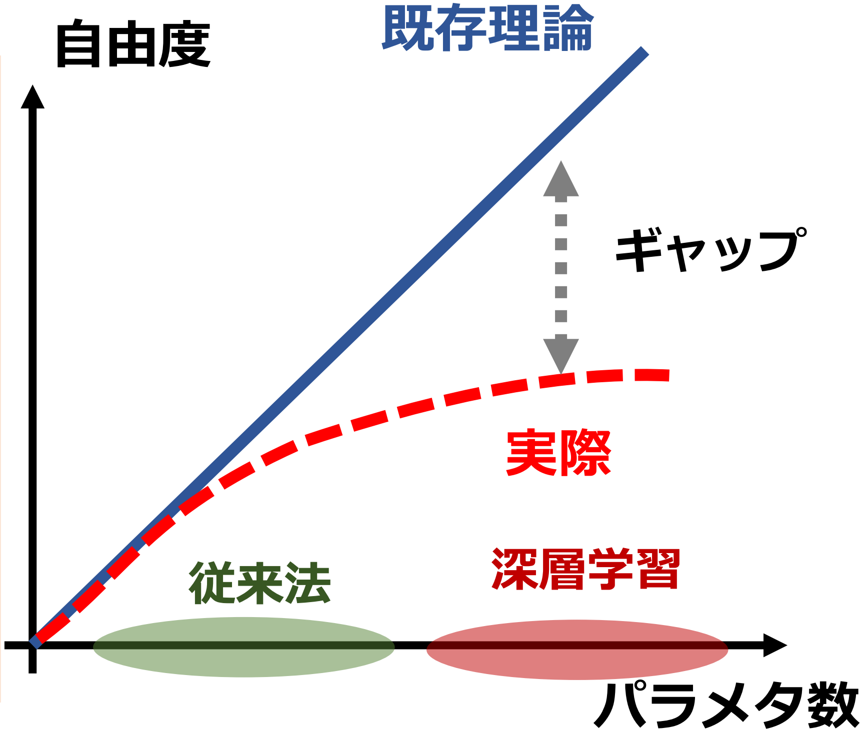
2. モデル自由度の再評価

既存の理論と深層学習の実際にはギャップ

<u>深層学習による**経験知**</u>

パラメタが多数あっても
モデル自由度は低いまま

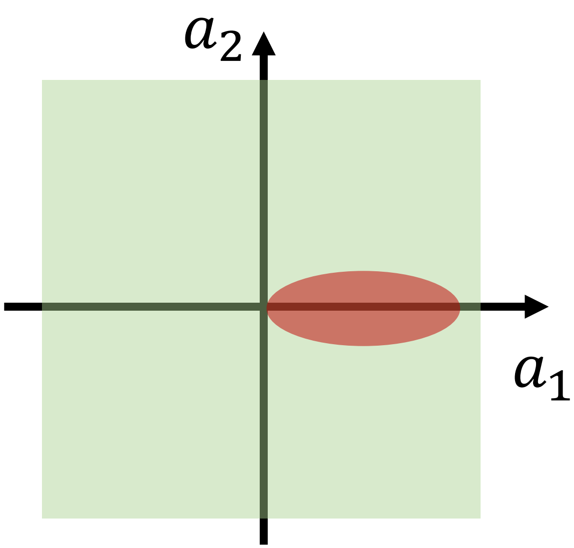
既存理論に則る方法(情報量規準など)が
のきなみ使えない状況



それでは "**実際の自由度**" は何で決まるのか？

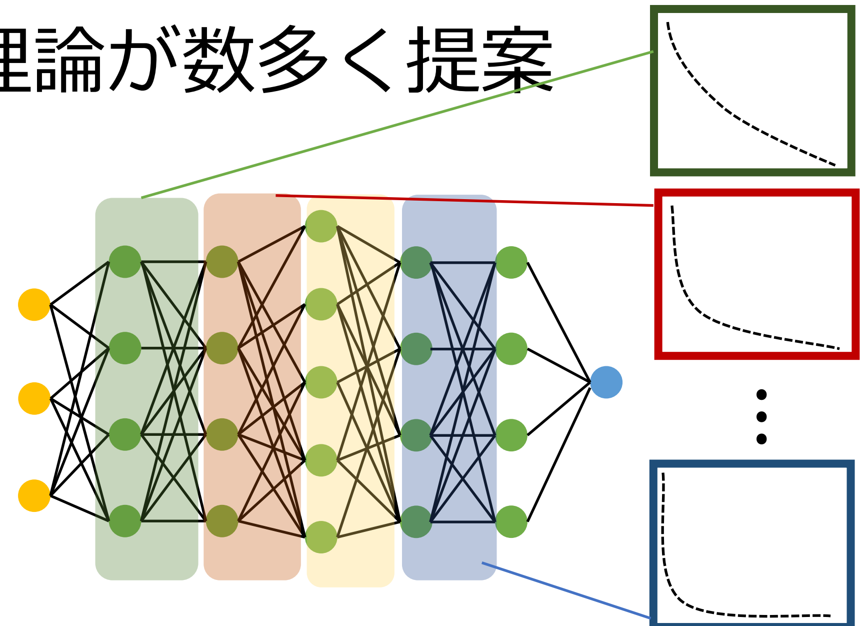
2.モデル自由度の再評価

新しいモデル自由度の理論が数多く提案



パラメタが実際に動く領域

Neyshabur+ (2015), Bartlett+ (2017)



各層のスペクトル
(パラメタ数に依存しない尺度)

Arora+ (2018)

理論的結果： 多様な尺度が提案
汎用的・統一的な理論は今後の課題

理論と実際のギャップ

1.多層の謎

なぜ層を増やすと性能が上がるのか？

2.大パラメタ数の謎

なぜ過適合による精度低下が無いのか？

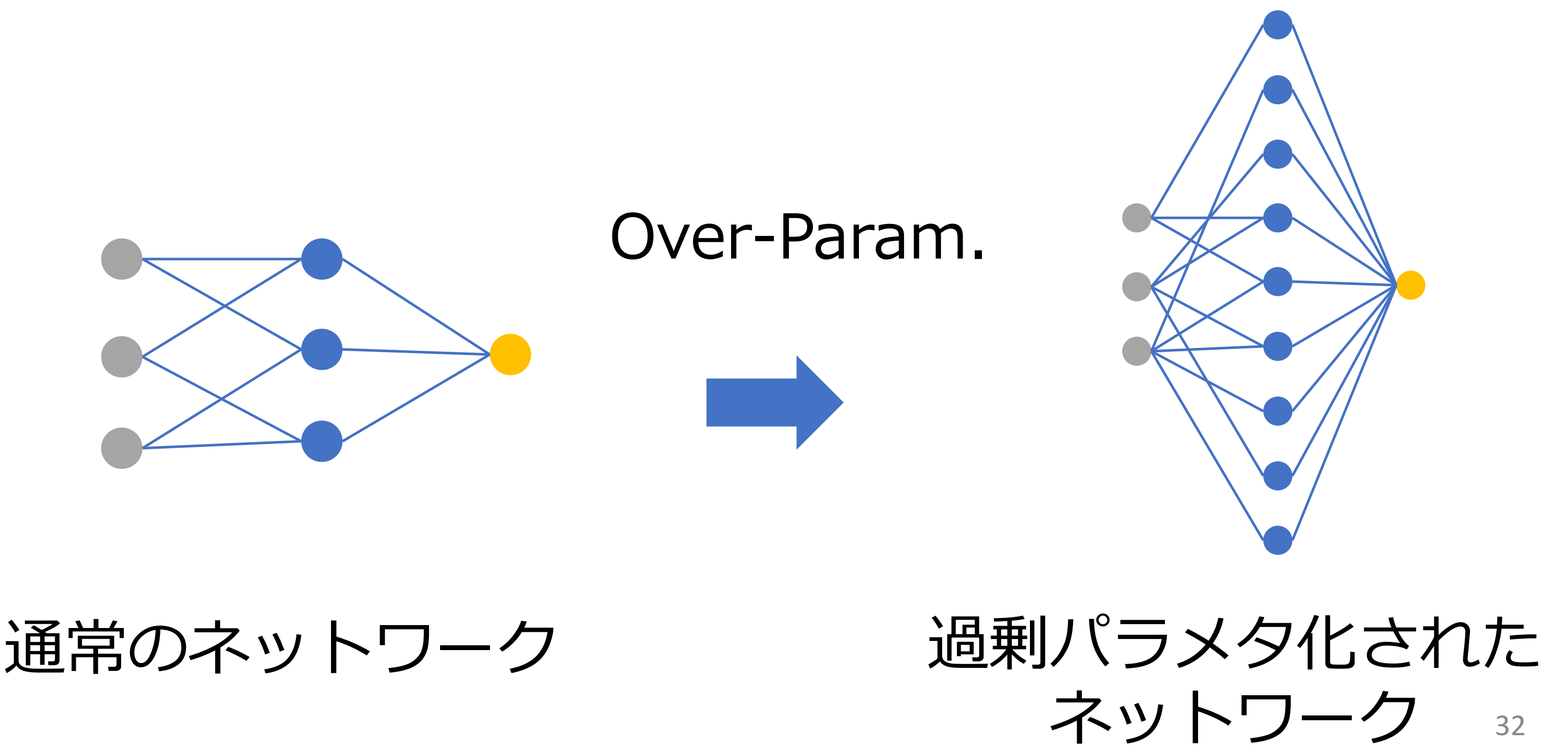
3.パラメタ学習の謎

なぜ多層なのにパラメタ学習できるのか

3.大域解を保証する試み

解決案：パラメタをさらに増やす

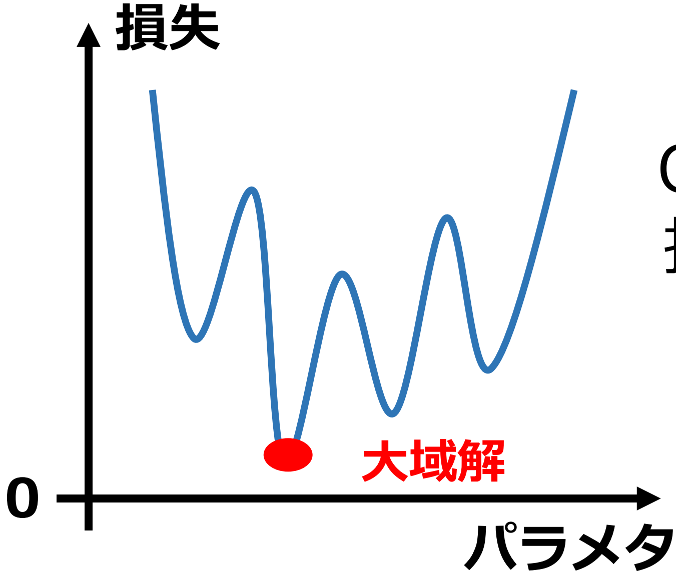
- 過剰パラメタ化（Over-Parametrization）



3.大域解を保証する試み

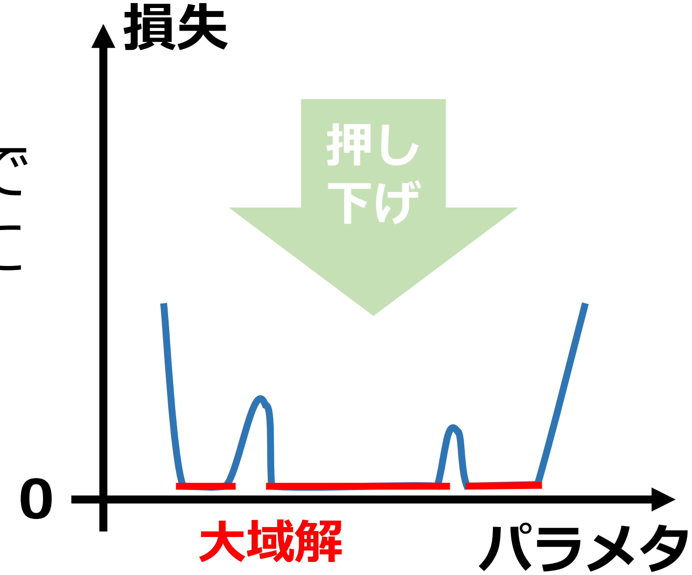
パラメタを増やして損失値を下げる

- 損失はマイナスにならない性質を使う



本来の損失

Over-Param.で
損失が全体的に
低下



ゼロに張り付く損失

Allen-Zhu+ (2019), Liang+(2018), Kawaguchi+ (2019)

損失が低下した結果、大域解への到達が容易に

3.大域解を保証する試み

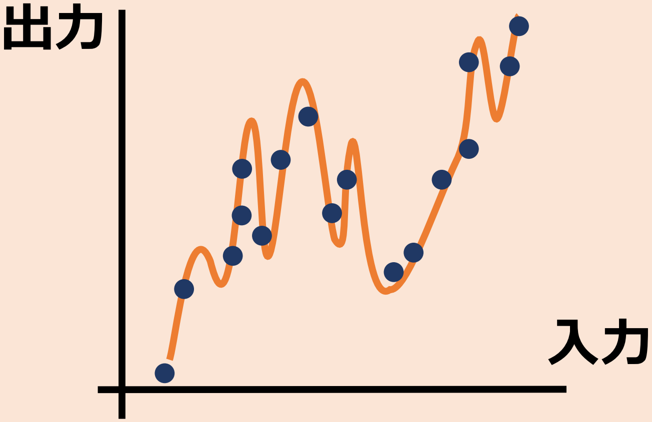
しかし、まだ非現実的な点が多い

<u>必要パラメタ数が**膨大**</u>

必要なパラメタ数
O(データ数^{30})

データ数が2倍になると
必要パラメタ数は10億倍

<u>**過適合**の問題</u>



統計理論と完全に突合

理論的結果：　解決への一つの方針が発見
しかし詳細は非現実的で、より研究が必要

理論的な試みのまとめ

問いに応える状況が個別に特定
- 網羅的な理論の構築は今後の課題

問い同士は深く関連 → 全てを明らかにする必要

深層学習と統計理論の今後

統計理論は何をするべき？

理論ができることは相対的に減る

性能確認したいけど
計算が大変！
理論で調べよう

昔：理論が保証を与えた

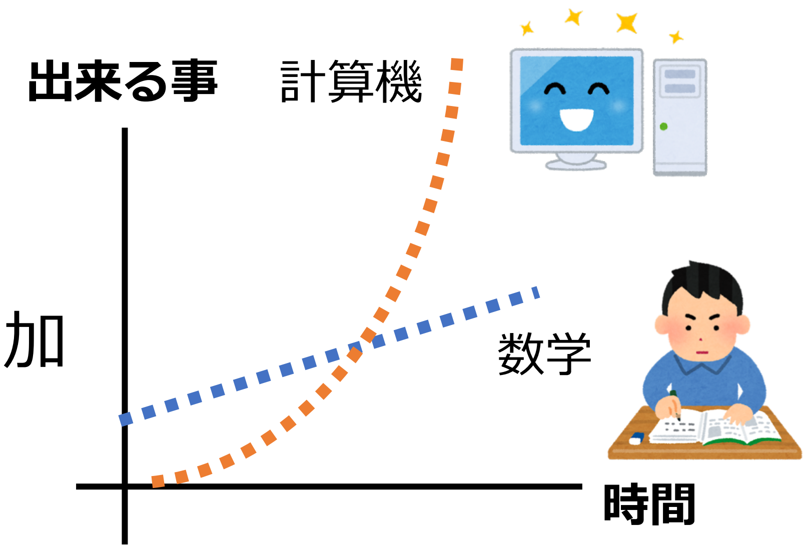
計算機の
発展

性能の確認は
計算機で十分！

今：計算機が保証を作れる

今後はより差が広がる

- ムーアの法則：
  計算機の性能は指数的に増加



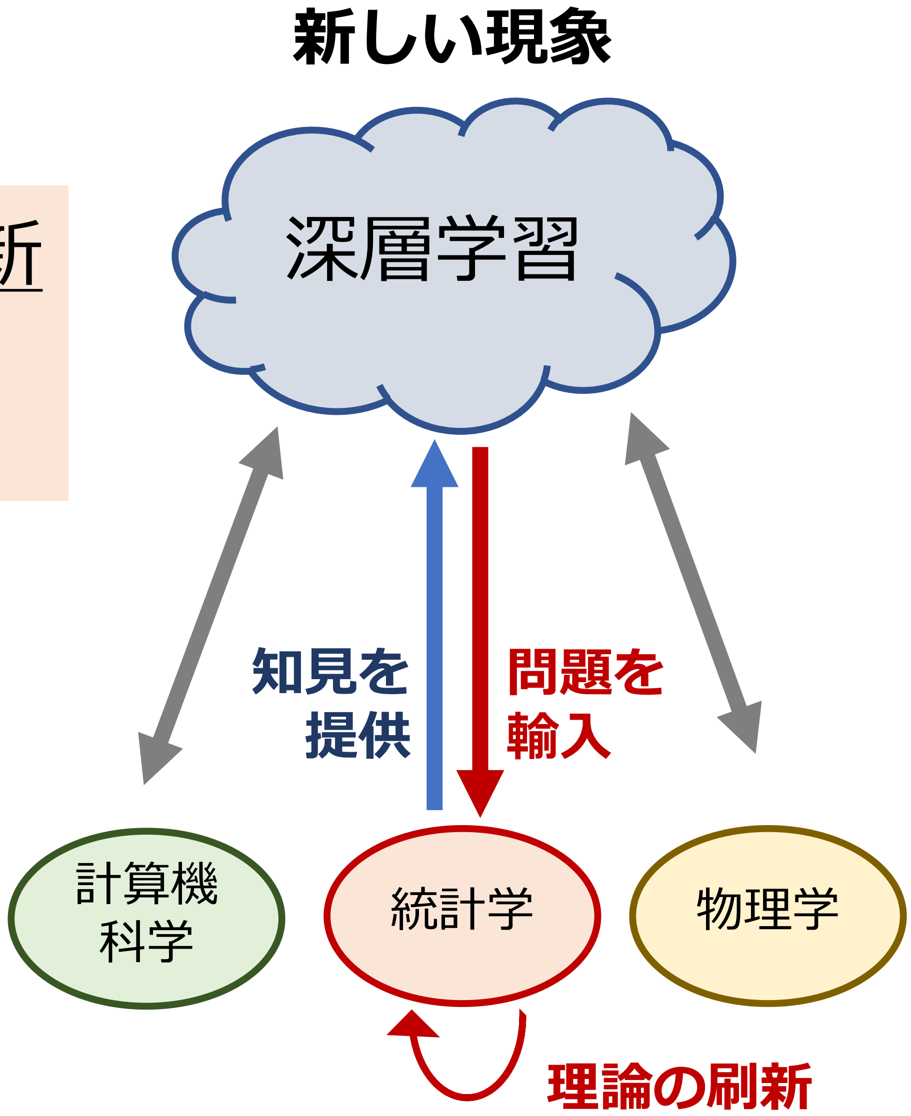
統計理論のすべきこと

新現象に基づく理論の更新

- 現象から問題を輸入
- 既存理論の再構築

現象に知見を提供

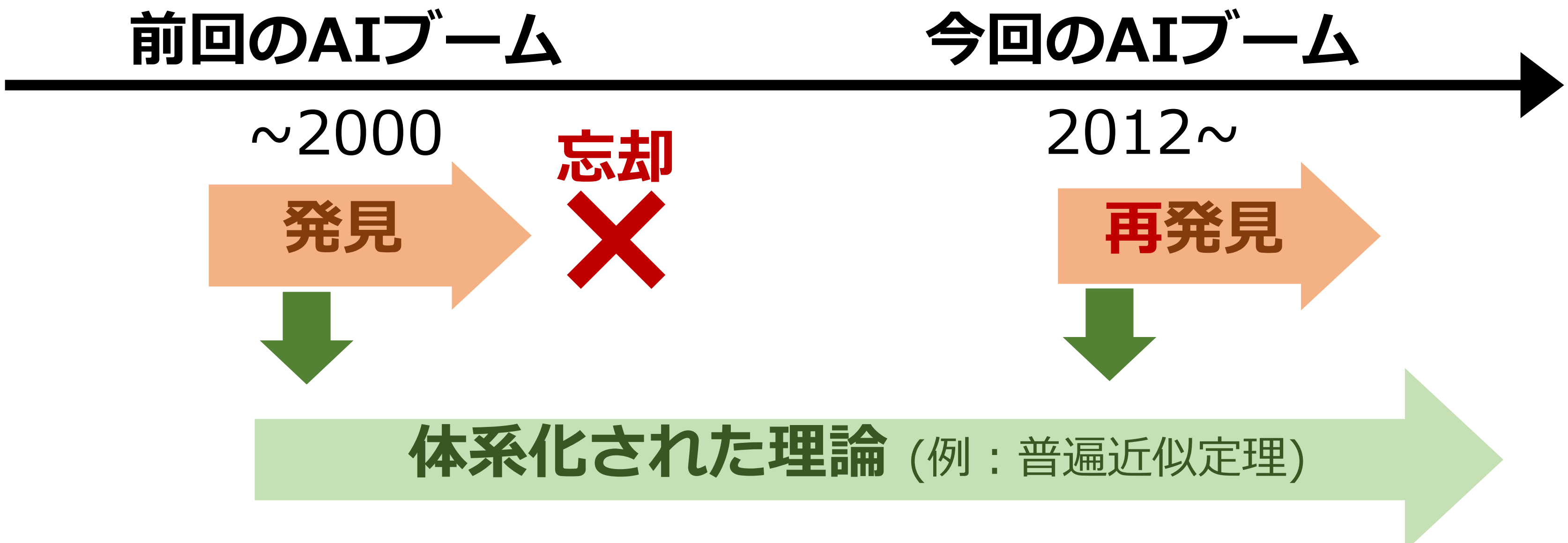
- 各領域間のシェア争い
- 題材が近いうちが好機



統計理論のすべきこと

発見を理論で体系化する

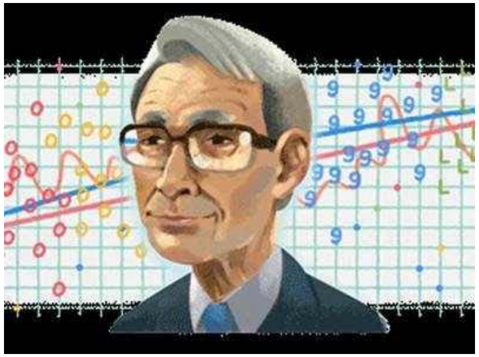
- 体系化されない知見は忘れられやすい



- 体系化 → 知見の継承

今後の展望

赤池弘次先生(元統数研所長)
From Googleトップページ



まとめ

- 新しいパラダイムへの対応

実用的な貢献のイメージ

1970年代

統計モデル → **理論的理解** → **AIC** など (赤池情報量規準)

2020年代

深層学習 → **理論的理解** → **?**

ご静聴ありがとうございました。

リファレンス

- 論文

  - Stone, C. J. (1980). Optimal rates of convergence for nonparametric estimators. The annals of Statistics.
  - Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems
  - Safran, I., & Shamir, O. (2017). Depth-width tradeoffs in approximating natural functions with neural networks. International Conference on Machine Learning.
  - Li, H., Xu, Z., Taylor, G., Studer, C., & Goldstein, T. (2018). Visualizing the loss landscape of neural nets. Advances in Neural Information Processing Systems.

リファレンス

- Imaizumi, M., & Fukumizu, K. (2019). Deep Neural Networks Learn Non-Smooth Functions Effectively. Artificial Intelligence and Statistics.
- Suzuki, T. (2018). Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: International Conference on Learning Representations.
- Neyshabur, B., Tomioka, R., & Srebro, N. (2015). Norm-based capacity control in neural networks. In Conference on Learning Theory.
- Bartlett, P. L., Foster, D. J., & Telgarsky, M. J. (2017). Spectrally-normalized margin bounds for neural networks. Advances in Neural Information Processing Systems.

リファレンス

- Arora, S., Ge, R., Neyshabur, B., & Zhang, Y. (2018). Stronger generalization bounds for deep nets via a compression approach. International Conference on Machine Learning.
- Allen-Zhu, Z., Li, Y., & Song, Z. (2018). A convergence theory for deep learning via over-parameterization. International Conference on Machine Learning.
- Liang, S., Sun, R., Lee, J. D., & Srikant, R. (2018). Adding one neuron can eliminate all bad local minima. Advances in Neural Information Processing Systems.
- Kawaguchi, K., & Kaelbling, L. P. (2019). Elimination of all bad local minima in deep learning. arXiv preprint.

おことわり

- 本スライドでは、シンプルな説明のため、厳密性をかなり思い切って犠牲にしています。正確な詳細を知りたい方は、リファレンスの論文の方を参照してください。

- “こういう解釈もあるよ！” というご意見がある方は、ご連絡をお願いします。是非議論しましょう。
    - imaizumi@ism.ac.jp