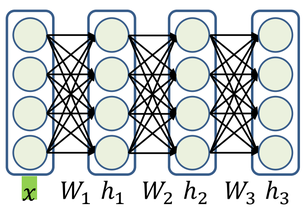
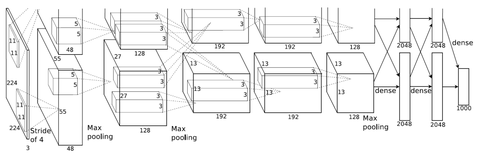
深層学習理論チーム
Deep Learning Theory Team

鈴木 大慈
Taiji Suzuki
RIKEN

チームの目標:
- 深層学習の理論的理解の促進
- 理論に基づいた新しい手法の開発
- 関連する機械学習問題の解法と最適化手法の開発

深層学習理論



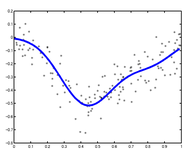


様々な応用で高い精度 → なぜか？

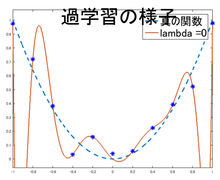
■ReLU深層ニューラルネットワークの適応能力

[Suzuki: Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality. ICLR2019]

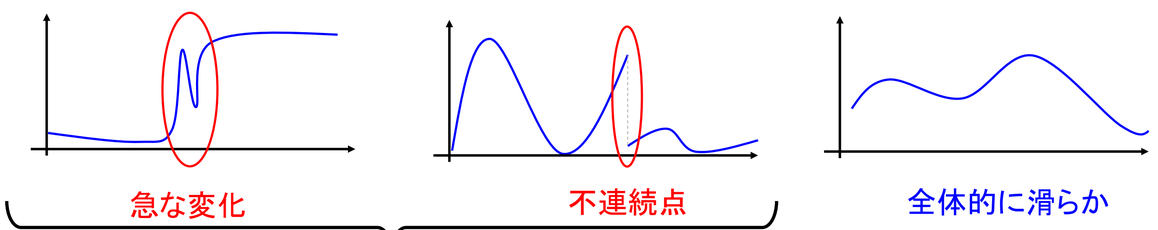
理論のポイント:深層学習の高い<u>適応能力</u>



教師有り学習
「<u>データに合うようモデルをフィット</u>」
しかし<u>過学習</u>も防がないといけない



機械学習では様々な形状をした複雑な関数が現れる



このような様々な関数形状にフィットする学習をするのは<u>難しい</u>.
- 急な変化に対応させようとすると必要以上にモデルが複雑に→過学習
- 滑らかな部分を重視すると急な変化に対応できない→ 悪い当てはまり

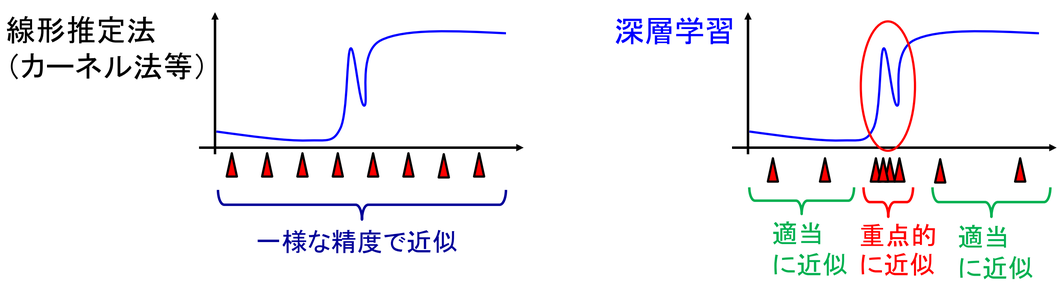
「適応力」が重要

定理

深層学習はBesov空間($B^s_{p,q}$)の元を推定するのに最適レートを達成する.
(複雑な関数形状に適応的にフィットすることができる)

線形推定法 (浅い学習)	深層学習
例:<u>カーネルリッジ回帰</u>, Sieve法, Nadaraya-Watson推定量...	
$\hat{f}(x) = K_{x,X}(K_{X,X} + \lambda \mathrm{I})^{-1} Y$ (カーネルリッジ回帰)	
推定誤差 $n^{-\frac{2s-2(1/p-1/2)_+}{2s+1-2(1/p-1/2)_+}}$	$\gg$ $n^{-\frac{2s}{2s+1}}$
推定誤差が大きい	推定誤差が小さい

カーネル法などの“浅い”手法は適応力が低く,最適レートを達成できない.
(n: サンプルサイズ, p:滑らかさの空間的一様性, s: 滑らかさ)

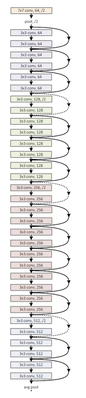


- 深層学習の高い特徴抽出能力を理論的に証明.
- 論文ではさらに, <u>次元の呪い</u>を回避できることも理論的に証明.
- 現在, GANや判別への理論展開が進展.

■関数勾配ブースティング法によるResNetの学習

[Nitanda&Suzuki: Functional Gradient Boosting based on Residual Network Perception. ICML2019]

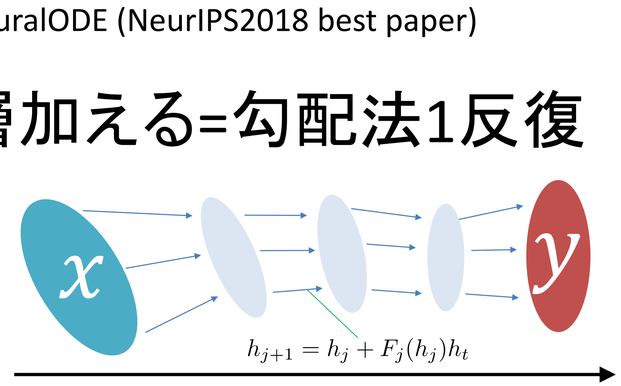
- **Residual Network**
  スキップコネクションを持つ巨大な深層ニューラルネットワーク.
  画像認識タスク等でSOTA.
- **勾配ブースティング (XGBoost, LightGBM)**
  予測器についての関数勾配によるブースティング法 (アンサンブル学習法).
  データマイニング系のコンペティションで最も有用とされるモデル.



両者の関連性に着目し, 新しいブースティング法を開発

ResNetの各層は特徴の最適化の一反復, 微分方程式の離散化とみなせる.
→特徴量のアンサンブルとして利用可能.
c.f., NeuralODE (NeurIPS2018 best paper)

1層加える=勾配法1反復



$$h_{j+1} = h_j + F_j(h_j)h_t \;\wr\!\!\!\wr\; \frac{\mathrm{d}h_t}{\mathrm{d}t} = \underbrace{\nabla L(h_t)}_{(=F_t)} \circ h_t$$

層を重ねるごとに目的関数を減少
(関数空間での無限次元勾配法)

中〜大規模データでの多値識別問題. 以下の手法と比較.
Random Feature + SVM, Random Forest, Gradient Boosting (LightGBM)

	METHOD	LETTER	USPS	IJCNN1	MNIST	COVTYPE	SUSY
提案手法	RESFGB (LOGISTIC)	**0.975 (0.0016)**	**0.954 (0.0006)**	0.987 (0.0011)	0.985 (0.0007)	0.968 (0.0017)	**0.804 (0.0000)**
提案手法	RESFGB (SMOOTH HINGE)	**0.975 (0.0012)**	0.950 (0.0022)	**0.988 (0.0018)**	**0.987 (0.0010)**	0.965 (0.0058)	**0.804 (0.0004)**
	SUPPORT VECTOR MACHINE	0.959 (0.0062)	0.948 (0.0023)	0.977 (0.0015)	0.969 (0.0041)	0.824 (0.0059)	0.754 (0.0534)
	RANDOM FOREST	0.964 (0.0012)	0.939 (0.0018)	0.980 (0.0005)	0.972 (0.0005)	0.948 (0.0005)	0.802 (0.0004)
	GRADIENT BOOSTING	0.964 (0.0011)	0.938 (0.0039)	0.982 (0.0010)	0.981 (0.0004)	**0.972 (0.0005)**	**0.804 (0.0005)**

- **SOTAとされるLightGBM以上の精度を確認.**
  幾つかのデータでは数反復で収束. 通常の勾配ブースティングより**効率的な最適化**.
- **理論保証もあり.**

■部分観測モデルにおける確率的最適化

[Murata&Suzuki: Sample Efficient Stochastic Gradient Iterative Hard Thresholding Method for Stochastic Sparse Linear Regression with Limited Attribute Observation. NeurIPS2019]

モデル: $y = \theta_*^\top x + \xi$

目標: $\min_{\|\theta\|_0 \le s'} \mathrm{E}_{(x,y)}[(y - \theta^\top x)^2]$ (スパース回帰)

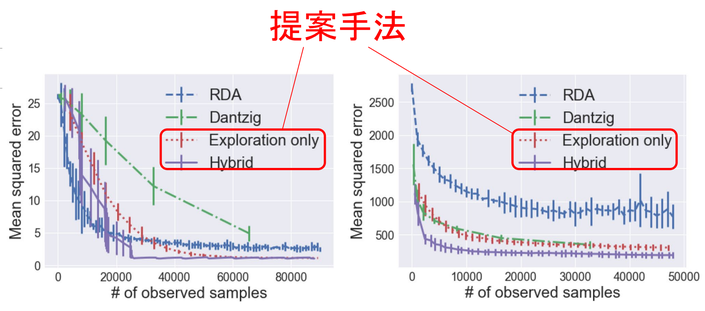
設定:
- (x, y)を一つずつ逐次的に観測 (オンライン学習)
- <u>xはその一部分(s'個)しか観測できない.</u>

応用例
- 医療診断調査→ 各患者ごと全ての項目について検査できない.
- 顧客アンケート

貢献: <u>**最適な推定精度**</u>を達成する<u>**確率的最適化手法**</u>を提案
- 観測数の次元への依存性を軽減
- 観測数の目標精度への依存性を軽減

既存手法との比較

	Sample complexity	# of observed attrs per example	Additional assumptions
Dantzig [2]	$\widetilde{O}\left(\frac{ds_*^2}{s'}\left(\sigma + \frac{d}{s'}\right)^2 \frac{1}{\varepsilon}\right)$	1	restricted isometry condition
RDA1 [1]	$O\left(d^2 \frac{\sigma^2}{\varepsilon^2}\right)$	$s_* + 2$	linear independence of features
RDA2 [1]	$O\left(\frac{d^6}{s'^6} + d\frac{\sigma^2}{\varepsilon^2}\right)$	s_*	linear independence of features
RDA3 [1]	$O\left(\frac{d^6}{s'^6} + d\frac{\sigma^2}{\varepsilon^2}\right)$	s_*	compatibility
Exploration	$\widetilde{O}\left(\frac{ds_*^2}{s'} + \frac{ds_*\sigma^2}{s'\varepsilon}\right)$	$O(s_*)$	restricted smoothness & restricted strong convexity
Hybrid	$\widetilde{O}\left(\frac{ds_*^2}{s'} + \left(\frac{s_*}{r_{\min}^2} \wedge \frac{ds_*}{s'}\right)\frac{\sigma^2}{\varepsilon}\right)$	$O(s_*)$	restricted smoothness & restricted strong convexity



難しさ:どの変数が重要かわからない.
- 探索と深化をほどよくバランス
- 重要な変数を探索し特定しつつ, パラメータを推定