九州大学集中講義

深層学習および機械学習の数理

鈴木大慈
東京大学大学院情報理工学系研究科数理情報学専攻
理研AIP

2020年9月2日～4日

機械学習と人工知能の歴史

1946： ENIAC，高い計算能力
フォン・ノイマン「俺の次に頭の良い奴ができた」
1952： A. Samuelによるチェッカーズプログラム

統計的学習

ルールベース

1960年代前半：
ELIZA（イライザ），
擬似心理療法士

1980年代：
エキスパートシステム

人手による学習ルールの作りこみの限界
「膨大な数の例外」

Siriなどにつながる

1957：Perceptron，ニューラルネットワークの先駆け
第一次ニューラルネットワークブーム
1963：線形サポートベクトルマシン

線形モデルの限界

1980年代：多層パーセプトロン，誤差逆伝搬，
畳み込みネット
第二次ニューラルネットワークブーム
1992： 非線形サポートベクトルマシン
（カーネル法）

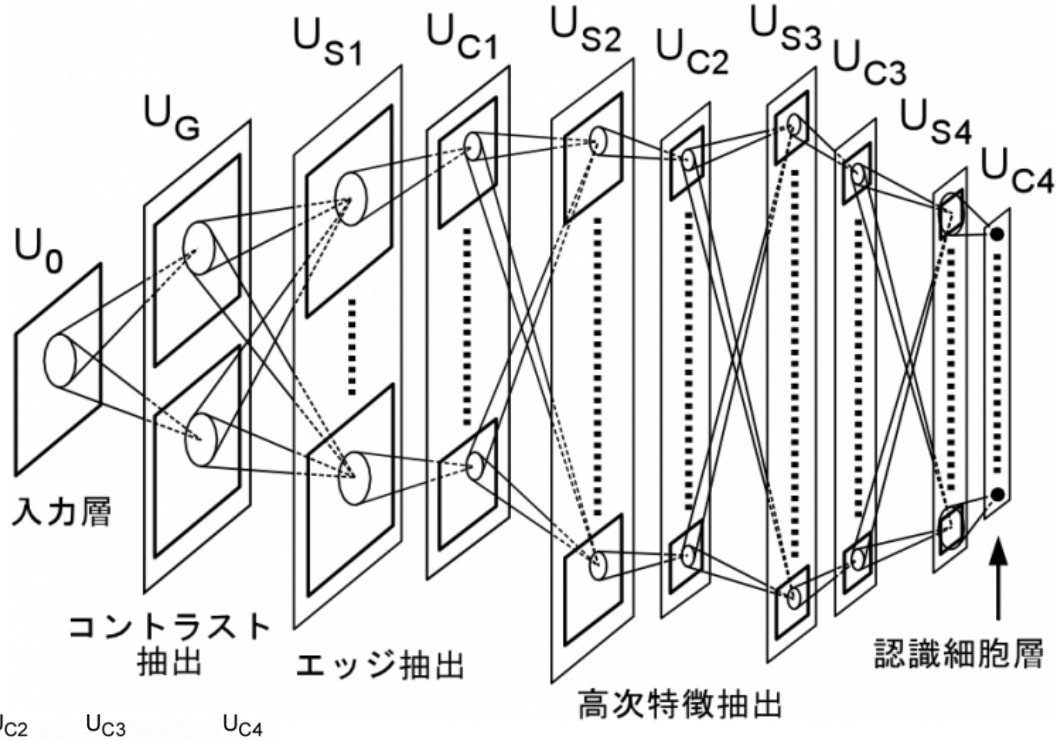
非凸性の問題

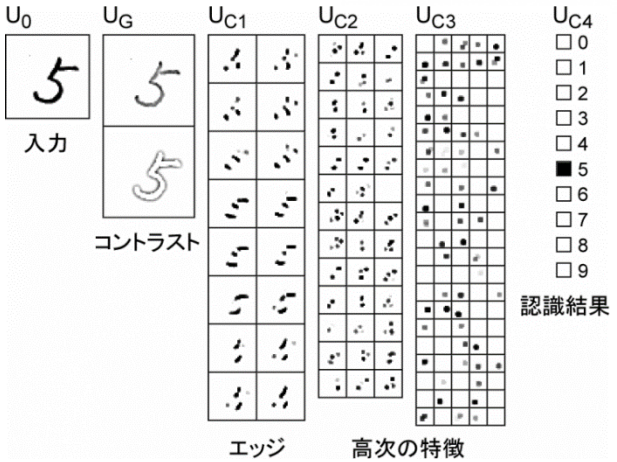
1996： スパース学習（Lasso）
2003： トピックモデル（LDA）
2012： Supervision（Alex-net）
第三次ニューラルネットワークブーム

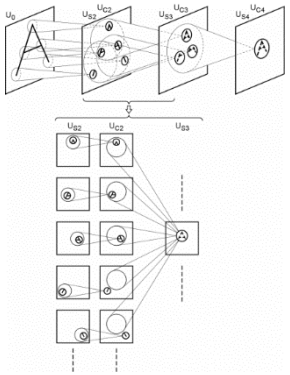
データの増加
＋計算機の強化

ネオコグニトロン

[福島,79]



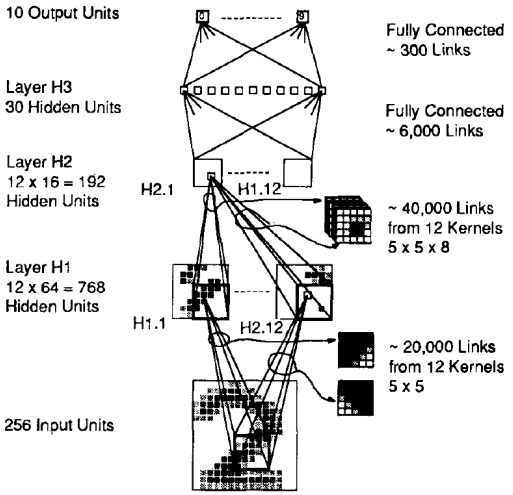




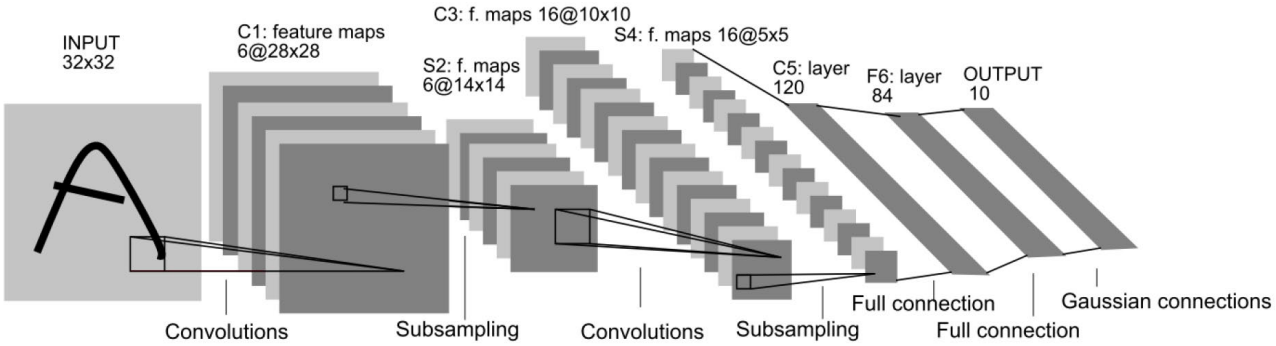
- 人間の脳を模倣
- 畳み込みネットの初期型
- 自己組織型学習
  →素子を足してゆく

LeNet

[LeCun+etal,89]



LeNet-5
[LeCun etal,98]



- 畳み込み＋プーリング：現在も使われている構造
- 誤差逆伝搬法でパラメータを更新
- 手書き文字認識データセット（MNIST）で９９％の精度を達成

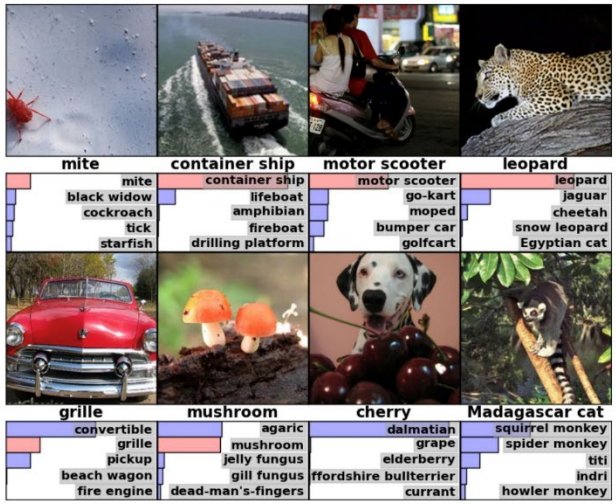
Deep Learning
深層学習

ImageNet

ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

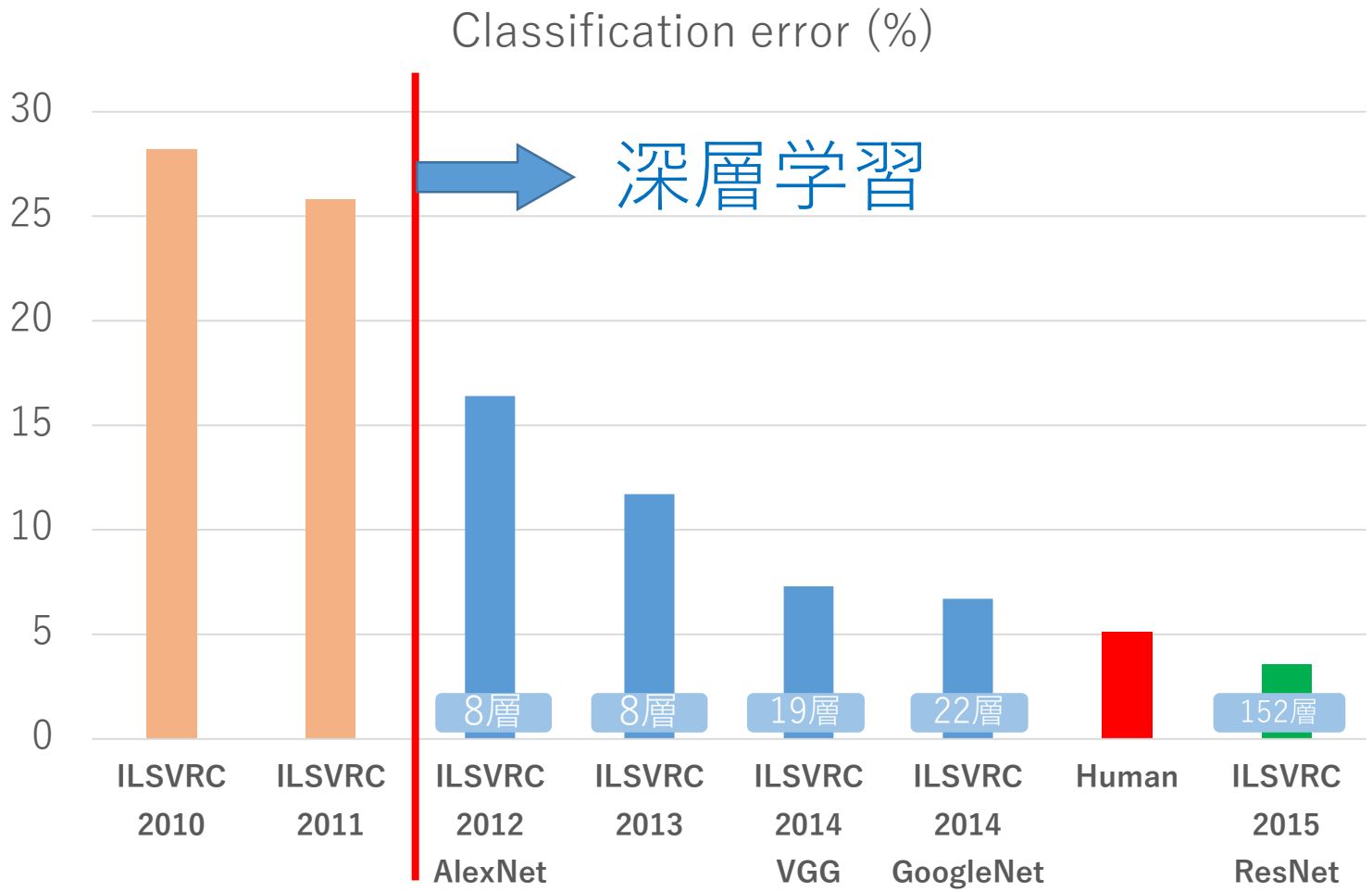


ImageNet: 1,000カテゴリ，約120万枚の訓練画像データ

ILSVRC (ImageNet Large Scale Visual Recognition Competition)

[J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei.
ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.]

ImageNetデータにおける識別精度の変遷

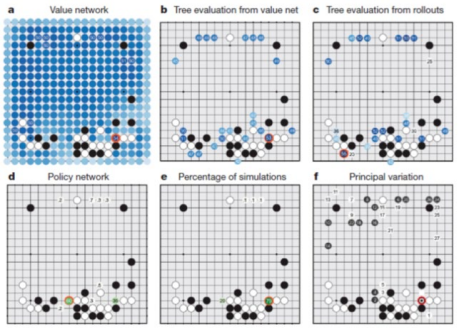


ImageNet: 21841クラス，14,197,122枚の訓練画像データ
そのうち1000クラスでコンペティション

深層学習の広がり

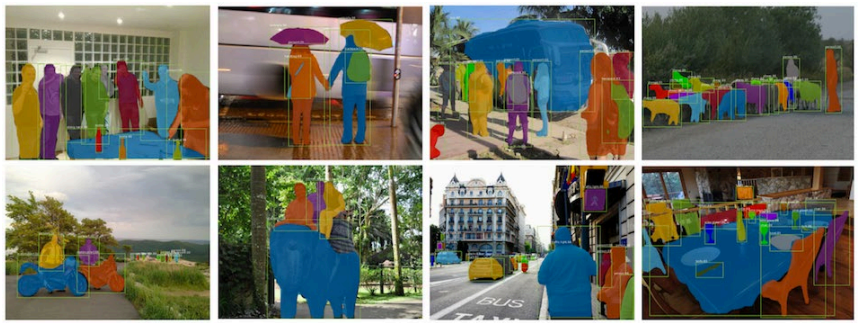
様々なタスクで高い精度

AlphaGo/Zero



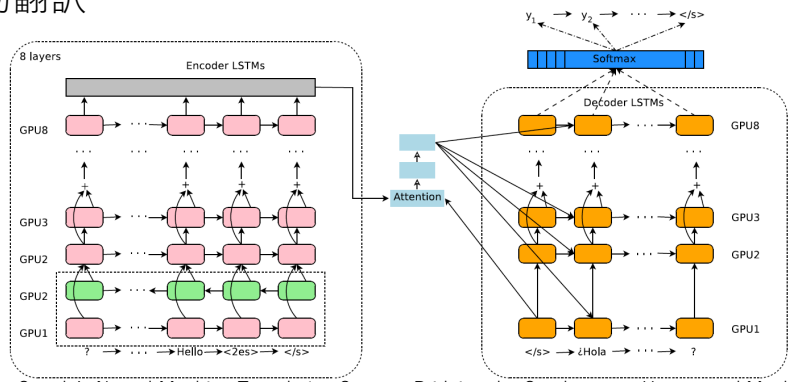
[Silver et al. (Google Deep Mind): Mastering the game of Go with deep neural networks and tree search, Nature, 529, 484—489, 2016]

画像認識



[He, Gkioxari, Dollár, Girshick: Mask R-CNN, ICCV2017]

自動翻訳



[Wu et al.: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. arXiv:1609.08144]

画像の生成

[Glow: Generative Flow with Invertible 1x1 Convolutions. Kingma and Dhariwal, 2018]

画像の変換

[Zhu, Park, Isola, and Efros: Unpaired image-to-image translation using cycle-consistent adversarial networks. ICCV2017.]

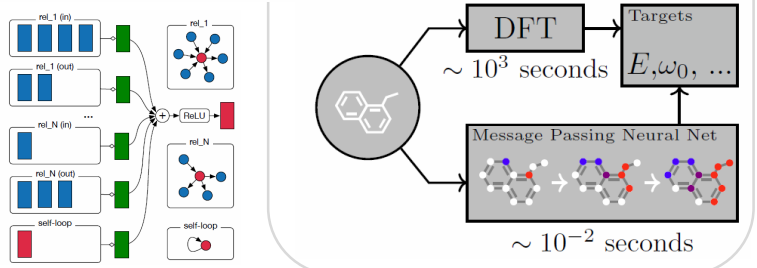
諸分野への波及

ロボット

[タオル畳み、サラダ盛り付け　「指動く」ロボット初公開, ITMedia:http://www.itmedia.co.jp/news/articles/1711/30/news089.html]

量子化学計算，分子の物性予測

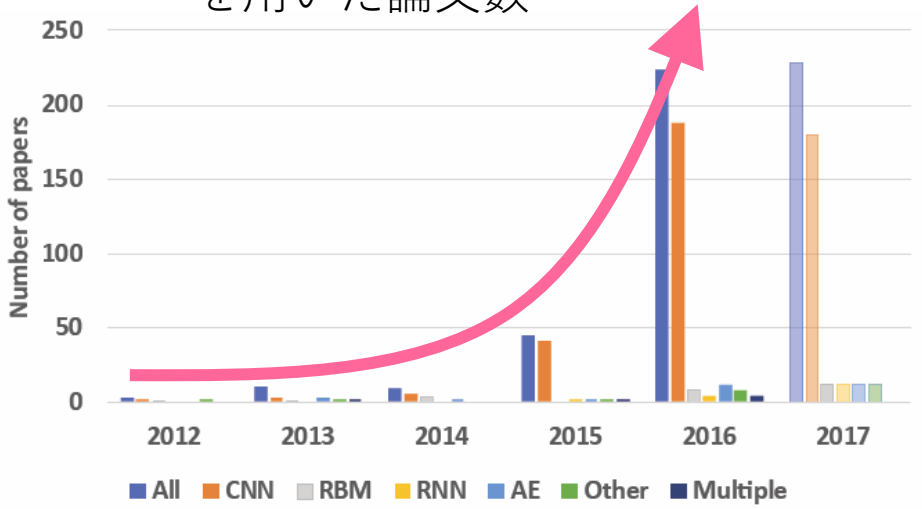
$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$



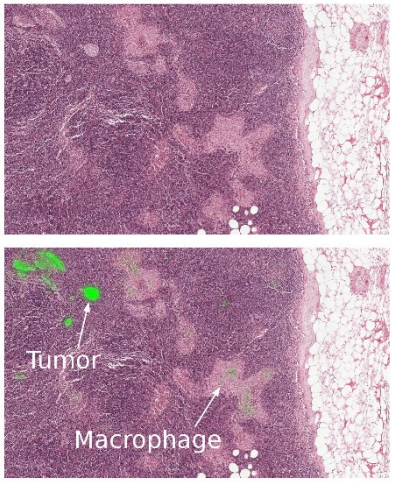
[Niepert, Ahmed&Kutzkov: Learning Convolutional Neural Networks for Graphs, 2016]
[Gilmer et al.: Neural Message Passing for Quantum Chemistry, 2017]
[Faber et al.:Machine learning prediction errors better than DFT accuracy, 2017.]

医療

医療分野における「深層学習」を用いた論文数



[Litjens, et al. (2017)]



- 人を超える精度
  （FROC73.3% -> 87.3%）
- 悪性腫瘍の場所も特定

[Detecting Cancer Metastases on Gigapixel Pathology Images: Liu et al., arXiv:1703.02442, 2017]

解決すべき問題点

なぜ深層学習はうまくいくのか？

- 「○○法が良い」という様々な仮説の氾濫.
- **世界的課題**

- 原理解明
- どうすれば“良い”学習が実現できるか？→新手法の開発

学会の問題意識

Ali Rahimi's talk at NIPS(NIPS 2017 Test-of-time award presentation)

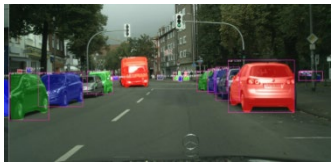
Ali Rahimi's talk at NIPS(NIPS 2017 Test-of-time award presentation)

Ali Rahimi’s talk at NIPS2017 (test of time award).
“Random features for large-scale kernel methods.”

“錬金術”という批判

民間の問題意識

- 中で何が行われているか分からないものは用いたくない.
- 企業の説明責任．深層学習のホワイトボックス化.



数学の必要性

深層学習の理論概観

応用

基礎

解釈可能性：
説明可能性，データの可視化，メンテナンスの容易化

各種テクニックの解析：
アーキテクチャの解析，損失関数の設計，最適化技法の解析

深層学習の原理解明：
表現理論，汎化誤差理論，最適化の収束理論

学習の本質解明：
“良い”学習手法の特徴付け，統一理論，深層学習を優越する方法論の提唱

理論を通して深層学習の不可思議な挙動を理解したい．

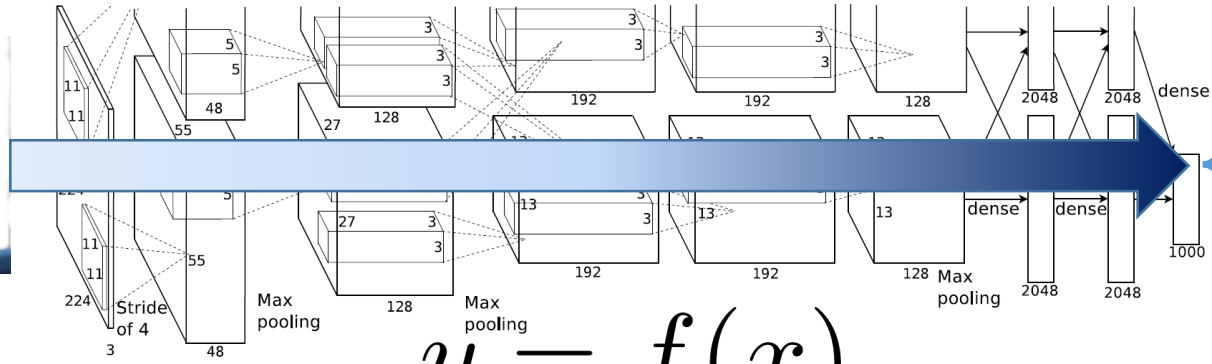
- 説明責任
- 可能性と限界の把握
- 学習手法設計の指針

応用から基礎まで広い範囲に“理論”は遍在する．

今日の範囲

教師あり学習

画像

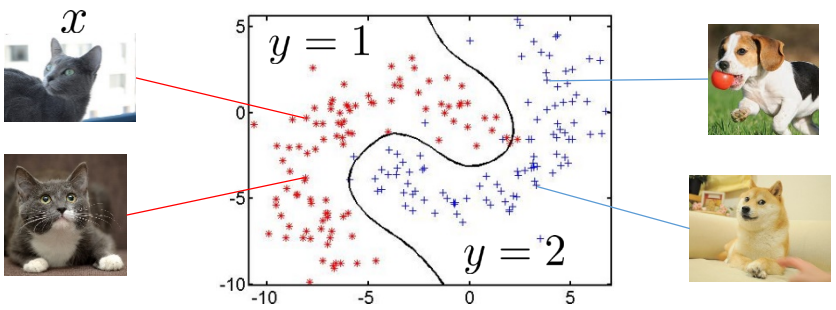


-猫　(y=1)
-犬　(y=2)
-人間 (y=3)

x

$y = f(x)$

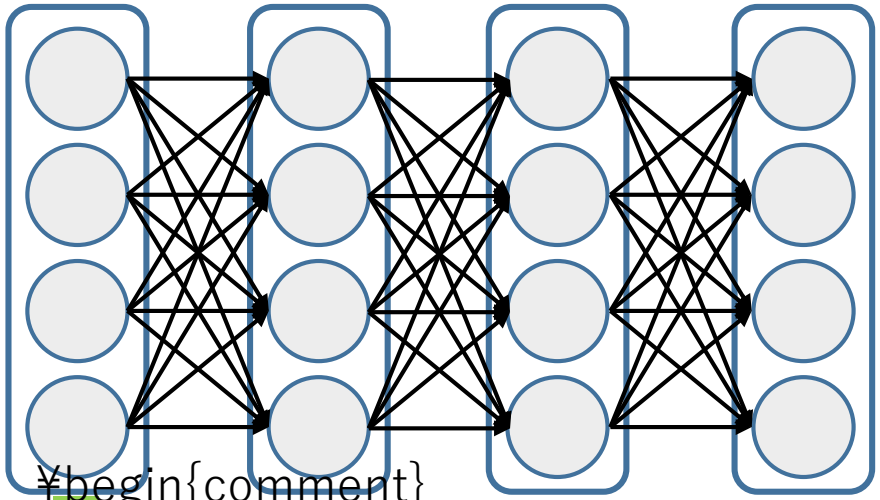
y



学習：「関数」をデータに当てはめる

モデル：関数の集合（例：深層NNの表せる関数の集合）

深層ニューラルネットの構造



¥begin{comment}

x W_1 h_1 W_2 h_2 W_3 h_3

基本的に「線形変換」と「非線形活性化関数」の繰り返し．

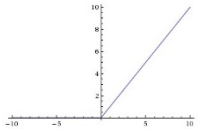
x → W_1x → $h_1(W_1x)$ → $W_2h_1(W_1x)$ → $h_2(W_2h_1(W_1x))$

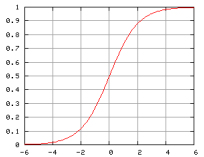
入力　線形変換　非線形変換　線形変換　非線形変換

$$h_1(u) = [h_{11}(u_1), h_{12}(u_2), \ldots, h_{1d}(u_d)]^T$$

活性化関数は通常要素ごとにかかる．Poolingのように要素ごとでない非線形変換もある．

- ☆ReLU (Rectified Linear Unit)：　$h(u) = \max\{u, 0\}$
- シグモイド関数：　$h(u) = \dfrac{1}{1+e^{-u}}$



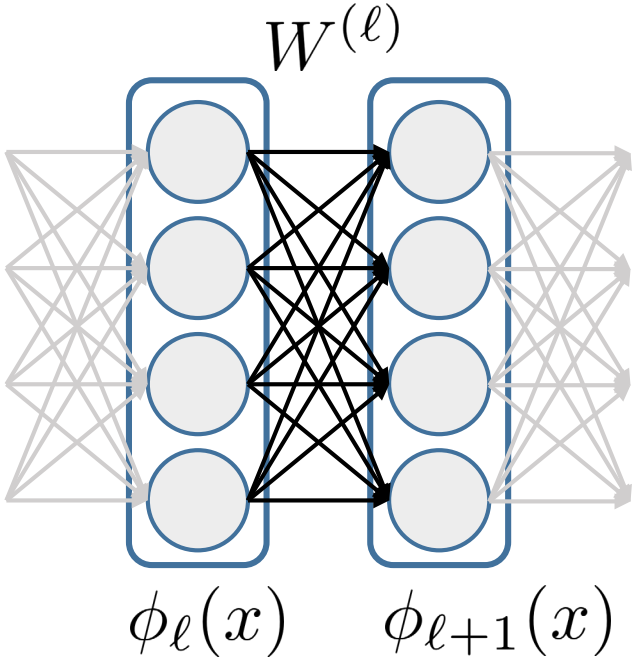


全結合モデル

- 第 ℓ 層

$$\phi_{\ell+1}(x) = \eta(W^{(\ell)}\phi_\ell(x) + b^{(\ell)})$$

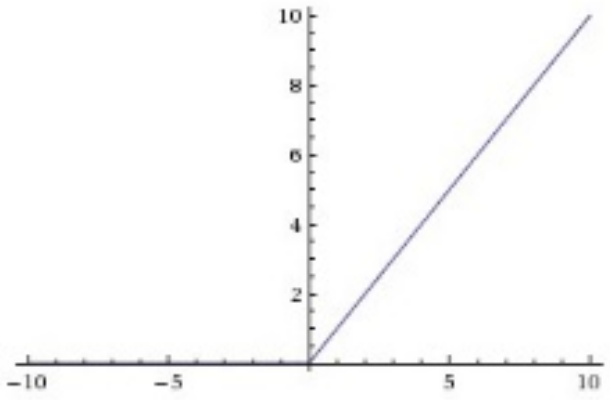
$$W^{(\ell)} \in \mathbb{R}^{m_{\ell+1} \times m_\ell} \quad b^{(\ell)} \in \mathbb{R}^{m_{\ell+1}}$$



活性化関数の例

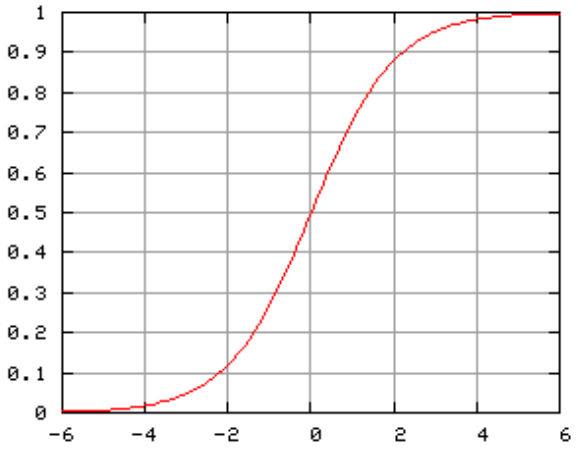
☆ReLU (Rectified Linear Unit)

$$\eta(u) = \max\{u, 0\}$$



シグモイド関数

$$\eta(u) = \frac{1}{1 + e^{-u}}$$



訓練誤差と汎化誤差

パラメータ θ：ネットワークの構造を表す変数

損失関数 $\ell(Y, f(X, \theta))$：パラメータ θ がデータをどれだけ説明しているか

予測誤差：損失の期待値

$$\mathrm{E}[\ell(Y, f(X, \theta))]$$

$$L(\theta)$$

本当に最小化したいもの.

訓練誤差：有限個のデータで代用

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i, \theta))$$

$$\widehat{L}(\theta)$$

代わりに最小化するもの.

(訓練データはテストデータと
同じ分布に従っていると仮定)

この二つには大きなギャップがある.
[過学習]

※クラスタリング等，教師なし学習も尤度を使ってこのように書ける.

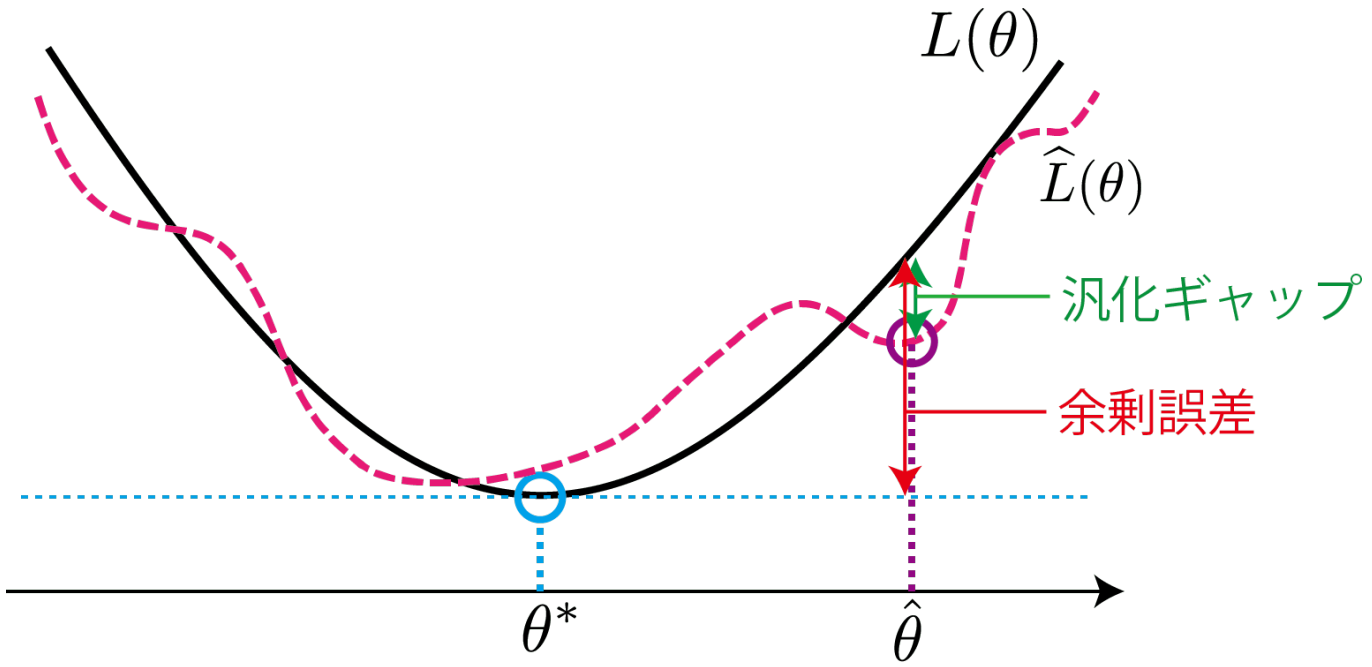
学習理論の設定

- 汎化ギャップ(汎化誤差)と余剰誤差
  Generalization gap　　Excess risk

汎化誤差: $L(\hat{\theta}) - \widehat{L}(\hat{\theta})$

余剰誤差: $L(\hat{\theta}) - \inf_{\theta} L(\theta)$

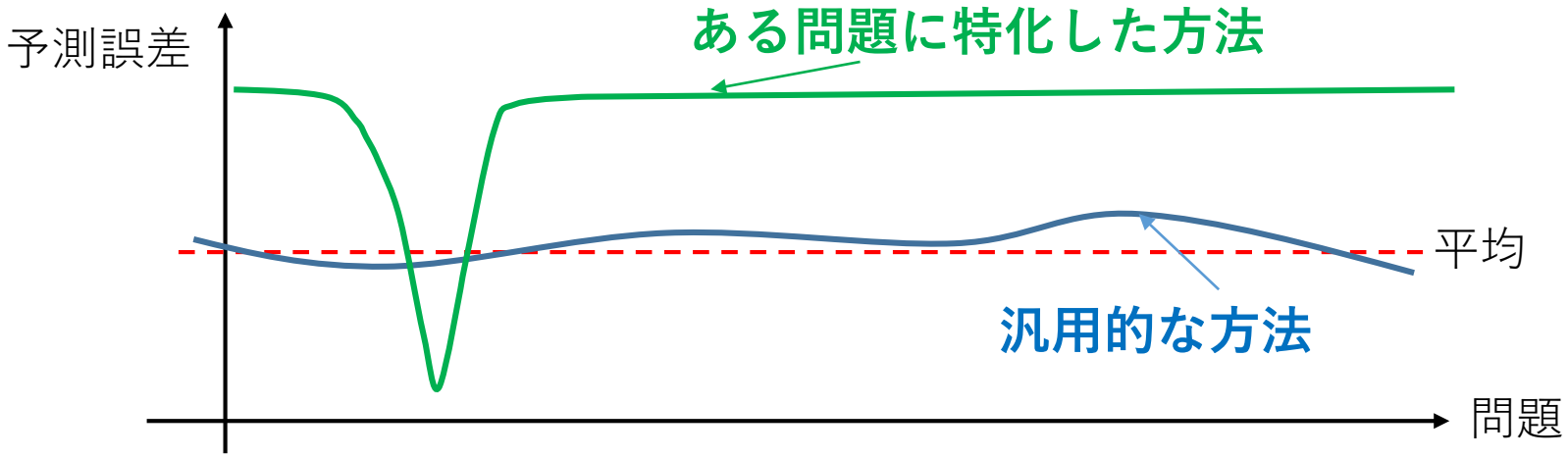
もしくは $L(\hat{\theta}) - \inf_{f} \mathbb{E}[\ell(Y, f(X))]$



学習機の複雑さと学習能力

- No free lunch theorem

「あらゆる問題で性能の良い汎用的学習機は実現不可能であり，ある問題に特殊化された手法に勝てない」



機械学習への教訓

「必要以上に複雑なモデルを当てはめると失敗する」

学習手法は「どこかを“贔屓”する必要がある」
→ モデリングの重要性 (オッカムの剃刀)

William of Ockham：1285-1347．スコラ学の神学者，哲学者．

No free lunch theorem: [D.H.Wolpert and W.G. Macready: 1995,1997][Y.C. Ho and D.L. Pepyne: 2002]

リスクの概念

- どこを贔屓するか？
  - モデルの外に真があればそもそも学習方法の優劣を論じることは難しい (どれもドングリの背比べ)
  - 仮にモデルに真が入っていた場合にどこが贔屓されるか

$\mathcal{P}$ ：真の分布のモデル　　$P^* \in \mathcal{P}$ ：真の分布

$D^n = (x_i, y_i)_{i=1}^n$ ：訓練データ　　$\hat{\theta}, \tilde{\theta}$: 推定量

■ ミニマックス最適性

$$\sup_{P^* \in \mathcal{P}} \mathbb{E}_{D^n \sim P^*}[L(\hat{\theta})] = \inf_{\tilde{\theta}:\text{Estimator}} \sup_{P^* \in \mathcal{P}} \mathbb{E}_{D^n \sim P^*}[L(\tilde{\theta})]$$

■ 許容性　つぎのような$\tilde{\theta}$が存在しない:

$$\mathbb{E}_{D^n \sim P^*}[L(\tilde{\theta})] \leq \mathbb{E}_{D^n \sim P^*}[L(\hat{\theta})] \quad (\forall P^* \in \mathcal{P})$$

$$\mathbb{E}_{D^n \sim P^*}[L(\tilde{\theta})] < \mathbb{E}_{D^n \sim P^*}[L(\hat{\theta})] \quad (\exists P^* \in \mathcal{P})$$

■ ベイズ最適性　　π_0: 事前分布

ベイズリスク $\int \mathbb{E}_{D^n \sim P^*}[L(\hat{\theta})] \mathrm{d}\pi_0(P^*)$ を最小にする推定法$\hat{\theta}$.

(→ ベイズ推定量)

損失関数最小化

経験損失（訓練誤差）

$$L(W) = \sum_{i=1}^{n} \ell(y_i, f(x_i, W))$$

$\ell(y, y') = (y - y')^2$　二乗損失（回帰）　$(y, y' \in \mathrm{R})$

$\ell(y, y') = -\sum_{k=1}^{K} y_k \log(y'_k)$　Cross-entropy損失（多値判別）
$(y_k \in \{0,1\}, y'_k \in [0,1]$, ともに和が1)

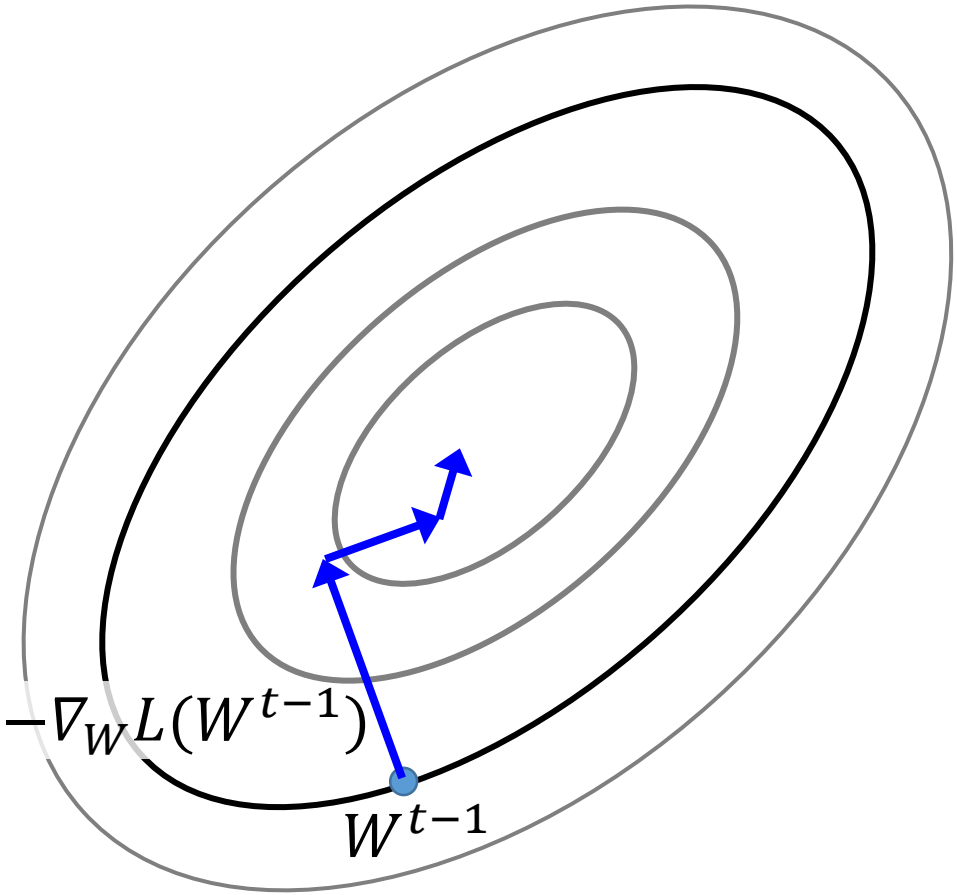
$$\min_W L(W)$$

$$W^t = W^{t-1} - \eta \nabla_W L(W^{t-1})$$

- 基本的には確率的勾配降下法 (SGD) で最適化を実行
- AdaGrad, Adam, Natural gradientといった方法で高速化

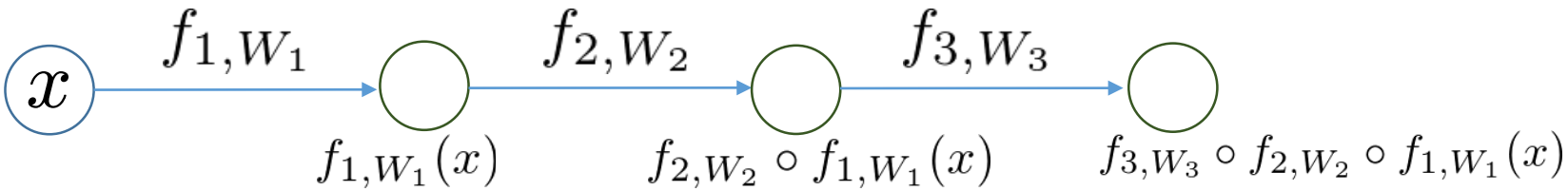
微分はどうやって求める？ → 誤差逆伝搬法

- 勾配降下法



$$W^t = W^{t-1} - \eta \nabla_W L(W^{t-1})$$

誤差逆伝搬法



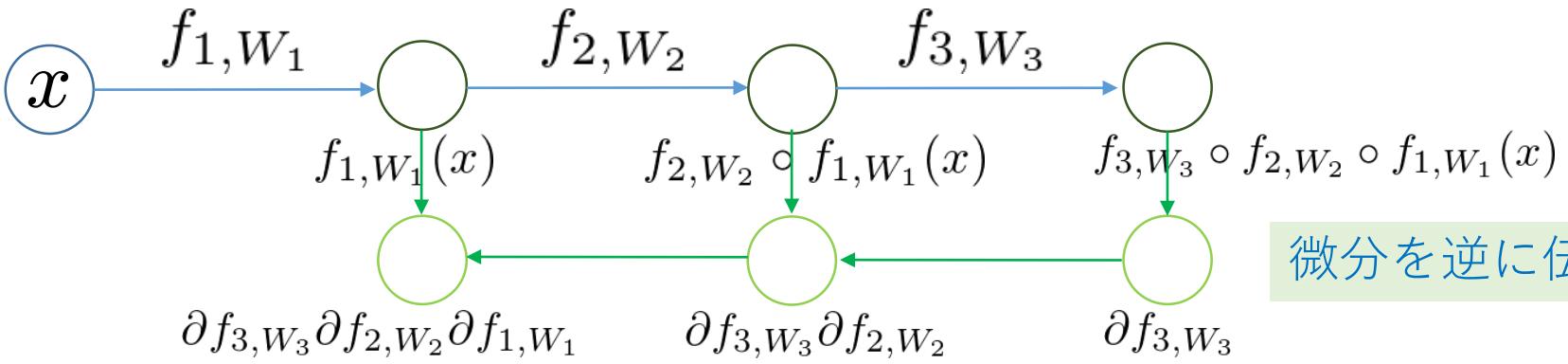
例： $f_{1,W_1}(x) = h(W_1 x)$

合成関数

$$f(x;W) = f_{3,W_3}(f_{2,W_2}(f_{1,W_1}(x)))$$
$$= f_{3,W_3} \circ f_{2,W_2} \circ f_{1,W_1}(x)$$

合成関数の微分

$$\frac{\partial f}{\partial W_1}(x) = \frac{\partial f_{3,W_3}}{\partial f_{2,W_2}} \frac{\partial f_{2,W_2}}{\partial f_{1,W_1}} \frac{\partial f_{1,W_1}}{\partial W_1}(x)$$



連鎖律を用いて微分を伝搬

$$\frac{\partial f}{\partial W_3}(x) = \frac{\partial f_{3,W_3}}{\partial W_3}\left(f_{2,W_2} \circ f_{3,W_3}(x)\right)$$

$$\frac{\partial f}{\partial W_2}(x) = \frac{\partial f_{3,W_3}}{\partial f_{2,W_2}} \frac{\partial f_{2,W_2}}{\partial W_2}(f_{3,W_3}(x))$$

$$\frac{\partial f}{\partial W_1}(x) = \frac{\partial f_{3,W_3}}{\partial f_{2,W_2}} \frac{\partial f_{2,W_2}}{\partial f_{1,W_1}} \frac{\partial f_{1,W_1}}{\partial W_1}(x)$$

パラメータによる微分と入力による微分は違うが，情報をシェアできる．

$f_{1,W}(x) = h(Wx)$ の場合　　$u = Wx$

$$\frac{\partial f_{1,W}}{\partial W_{ij}}(x) = \frac{\partial h}{\partial u_i}(u) x_j \qquad \frac{\partial f_{1,W_1}}{\partial x_j}(x) = \sum_i \frac{\partial h}{\partial u_i}(u) W_{ij}$$

確率的勾配降下法 (SGD)

(Stochastic Gradient Descent)

沢山データがあるときに強力

$$\min_W \underbrace{\frac{1}{n}\sum_{i=1}^{n}\ell(z_i, W)}_{\text{重い}}$$

大きな問題を分割して個別に処理

普通の勾配降下法：

$$W^t = W^{t-1} - \alpha\nabla L(W)$$

$$= W^{t-1} - \alpha\underbrace{\frac{1}{n}\sum_{i=1}^{n}\nabla\ell(z_i, W)}_{\text{全データの計算}}$$

確率的勾配降下法 (SGD)

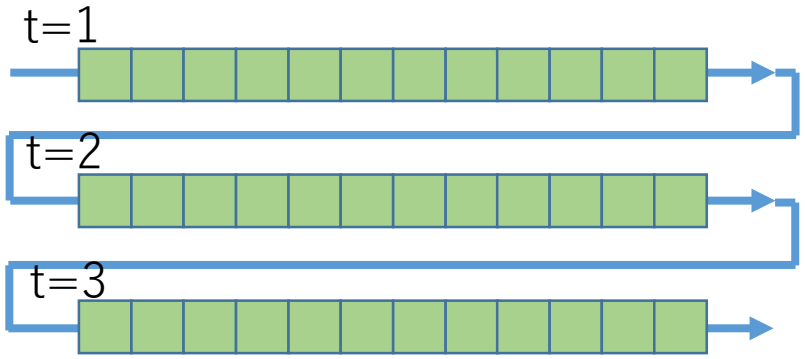
(Stochastic Gradient Descent)

沢山データがあるときに強力

$$\min_W \underbrace{\frac{1}{n}\sum_{i=1}^{n}}_{\text{重い}} \ell(z_i, W)$$

大きな問題を分割して個別に処理

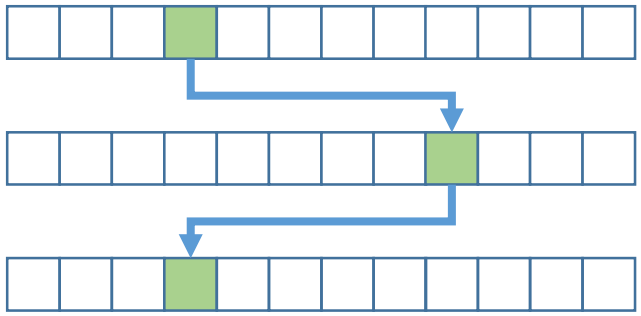
普通の勾配降下法：



$$W^t = W^{t-1} - \alpha \frac{1}{n}\sum_{i=1}^{n} \nabla \ell(z_i, W)$$

確率的勾配降下法：

毎回の更新でデータを一つ(または少量)しか見ない



$$W^t = W^{t-1} - \alpha \nabla \ell(z_i, W)$$

深層学習の理論

理論的課題

表現能力 (第一章)

どれだけ難しい問題まで学習できるようになるか？

汎化能力(第二章)

有限個のデータで学習した時，どれだけ正しく初見のデータを正解できるようになるか？

最適化能力(第三章)

最適な重みを高速に計算機で求めることが可能か？

要点のまとめ

- **表現能力**
  - 重要な関数クラス(Barron, Holder, Sobolev, Besov) はほぼ最適な効率性で近似できる.
  - 適応的な関数近似によりカーネル法を優越する.
- **汎化能力**
  - 重要な関数クラスの推定精度はミニマックス最適レートを達成できる→特徴抽出機能によりカーネル法を優越.
  - データサイズがパラメータ数より小さくても過学習しない→実質的統計的自由度が小さい.
  - 陰的正則化等により，自由度が小さく抑えられる→汎化する.
- **最適化能力**
  - 横幅を十分広くとれば大域的最適解が勾配法で求まる.
  - 初期パラメータのスケーリングによってNeural Tangent KernelとMean fieldの二つの状況に大きく分けられる.
  - ノイズ付加により平滑化・局所解からの脱出を実現.

第1章
深層学習の表現能力

Kolmogorovの加法定理

任意の連続関数は横幅固定の4層の“ニューラルネット”で表現できる.

定理 (Kolmogorov's superposition theorem)

- 定数 $\lambda_j > 0\ (j = 1, \ldots, d),\ \sum_{j=1}^{d} \lambda_j \leq 1,$
- $I = [0,1]$からIへの狭義単調増大連続関数 $\phi_q\ (q = 1, \ldots, 2d+1)$

が存在して，任意の連続関数$f \in C([0,1]^d)$が次のように表現できる:

$$f(x_1, \ldots, x_d) = \sum_{q=1}^{2d+1} g(\lambda_1 \phi_q(x_1) + \cdots + \lambda_d \phi_q(x_d))$$

なお，$g \in C([0,1])$はfにのみ依存した関数.

- **一変数関数の合成で多変数関数が作れてしまう.**
- 任意の連続関数は4層ニューラルネットの最終層だけを学習すればよいことになる．しかし，gの滑らかさはfおよび入力の次元dに強く依存し，最適な学習精度は達成できない.
- ヒルベルトの23の問題の13番目を一般化した定理

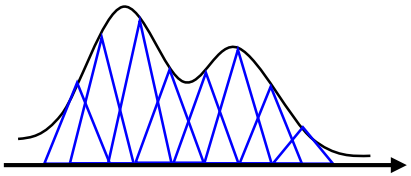
表現能力「万能近似能力」

二層ニューラルネットワークは
どんな関数も任意の精度で近似できる.

理論的にはデータが無限にあり，素子数が無限にあるニューラルネットワークを用いればどんな問題でも学習できる.

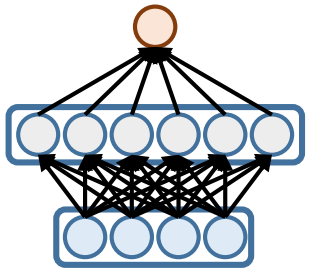
[Hecht-Nielsen,1987][Cybenko,1989]

「関数近似理論」

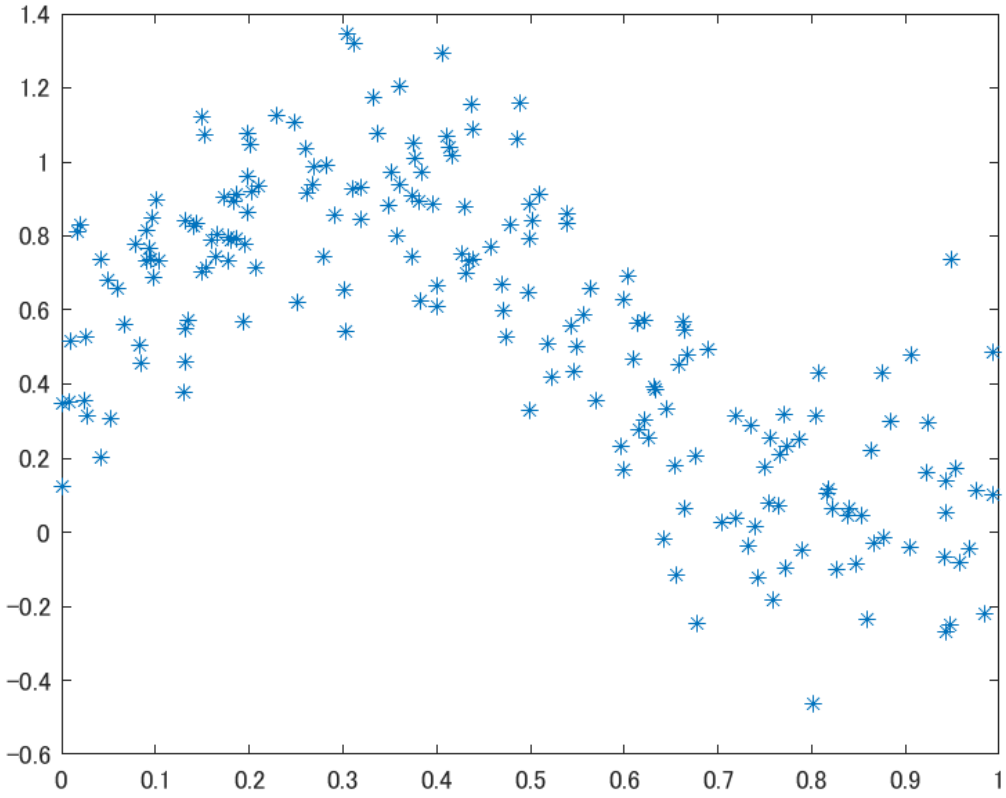


年		基底関数	空間
1987	Hecht-Nielsen	対象毎に構成	$C(R^d)$
1988	Gallant & White	Cos	$L_2(K)$
	Irie & Miyake	integrable	$L_2(R^d)$
1989	Carroll & Dickinson	Continuous sigmoidal	$L_2(K)$
	Cybenko	Continuous sigmoidal	$C(K)$
	Funahashi	Monotone & bounded	$C(K)$
1993	Mhaskar + Micchelli	Polynomial growth	$C(K)$
2015	Sonoda + Murata	**Unbounded**, admissible	$L_1(R^d)/L_2(R^d)$

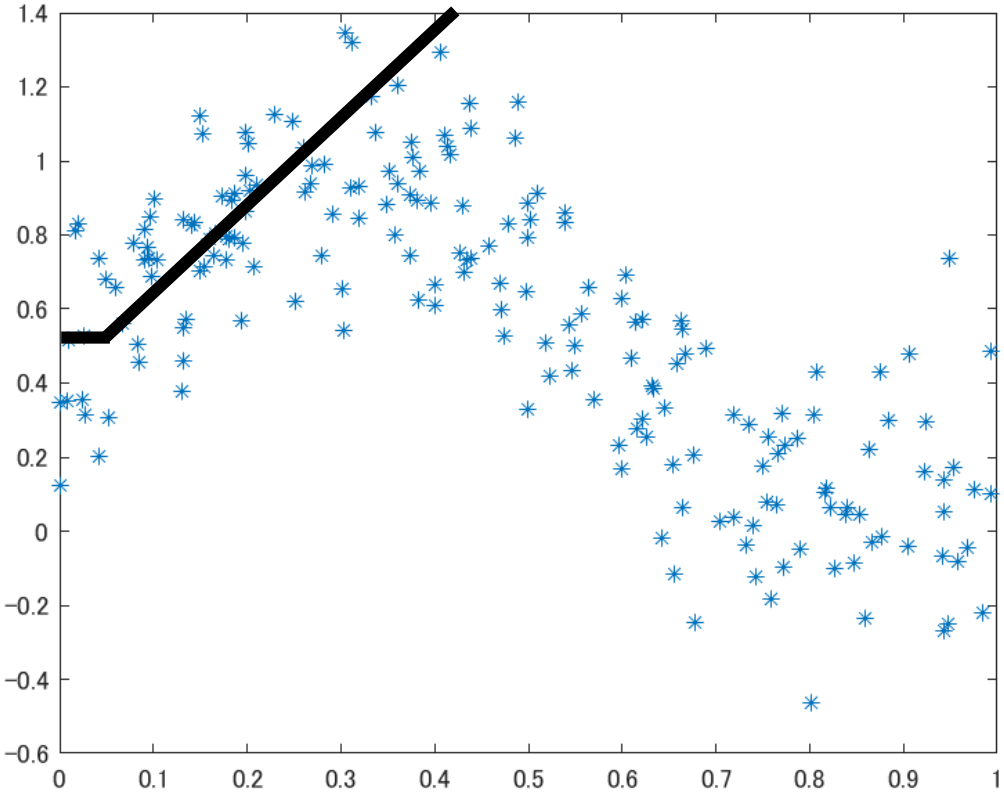
$$\hat{f}(x) = \sum_{j=1}^{m} v_j \eta(w_j^\top x + b_j)$$

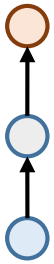


関数近似の様子



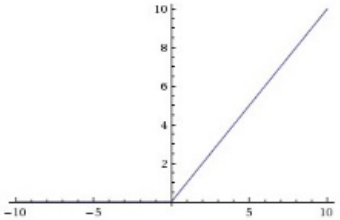
関数近似の様子





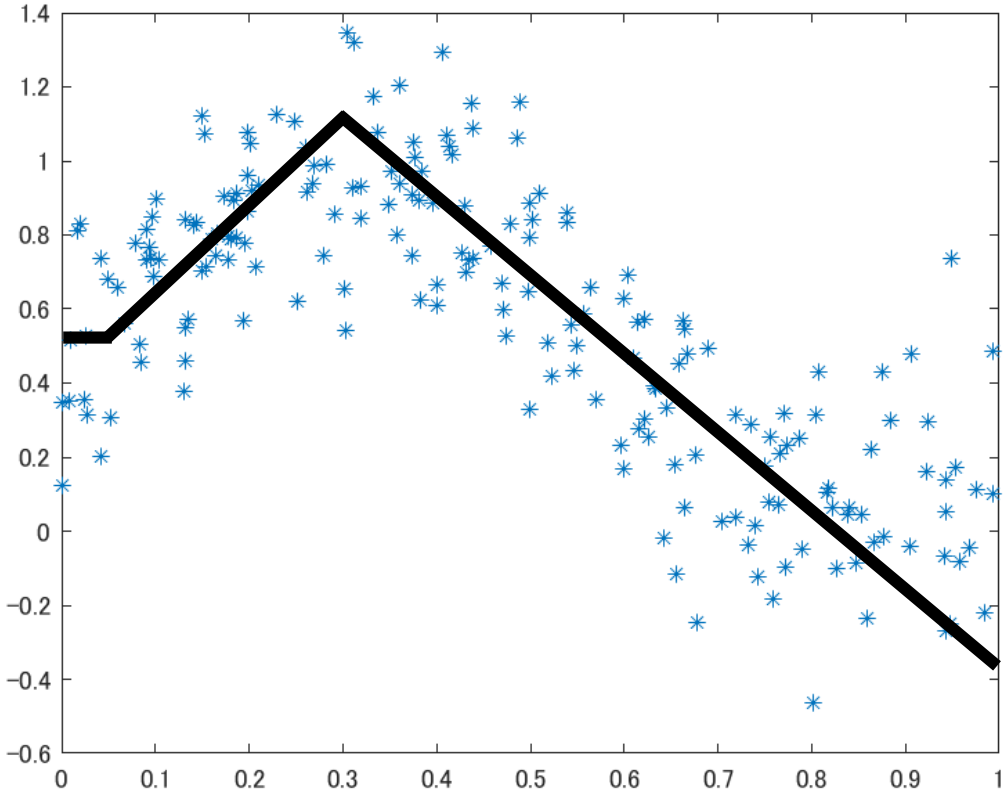
$a_1\eta(b_1x+c_1)$

ReLU活性化関数

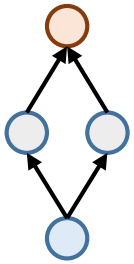


$\eta(u)=\max\{u,0\}$

関数近似の様子

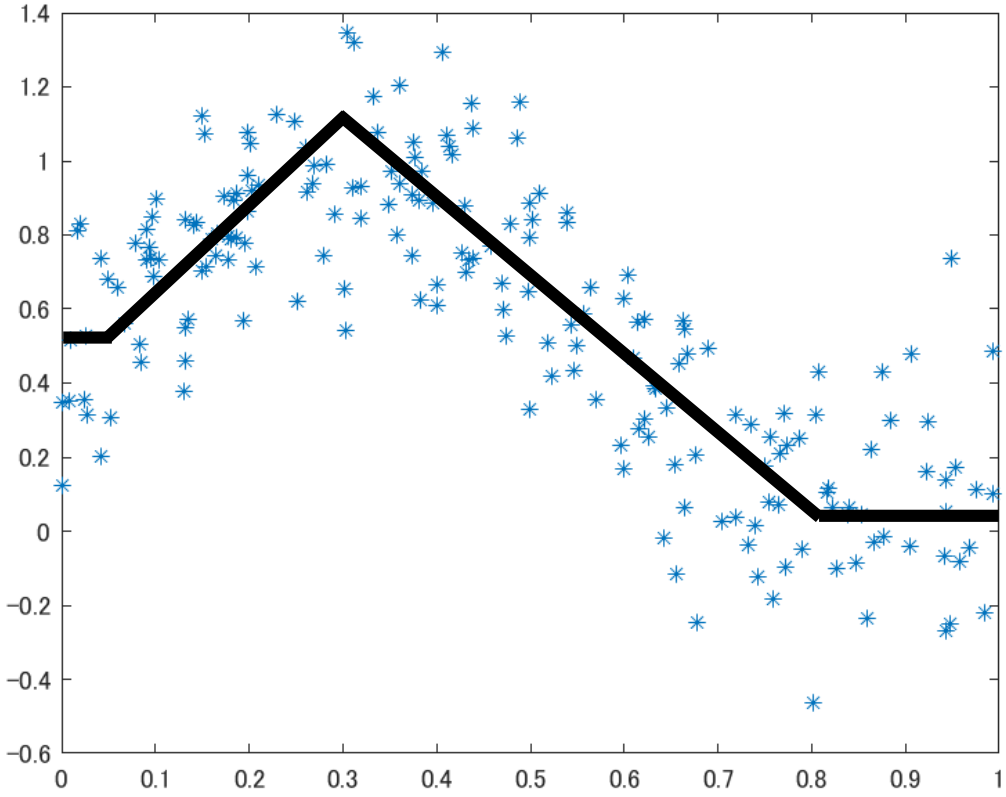


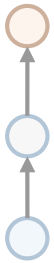
$a_1\eta(b_1x + c_1)$



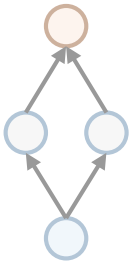
$a_1\eta(b_1x + c_1)$
$+ a_2\eta(b_2x + c_2)$

関数近似の様子

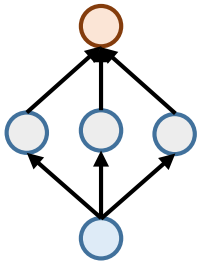




$a_1\eta(b_1x + c_1)$

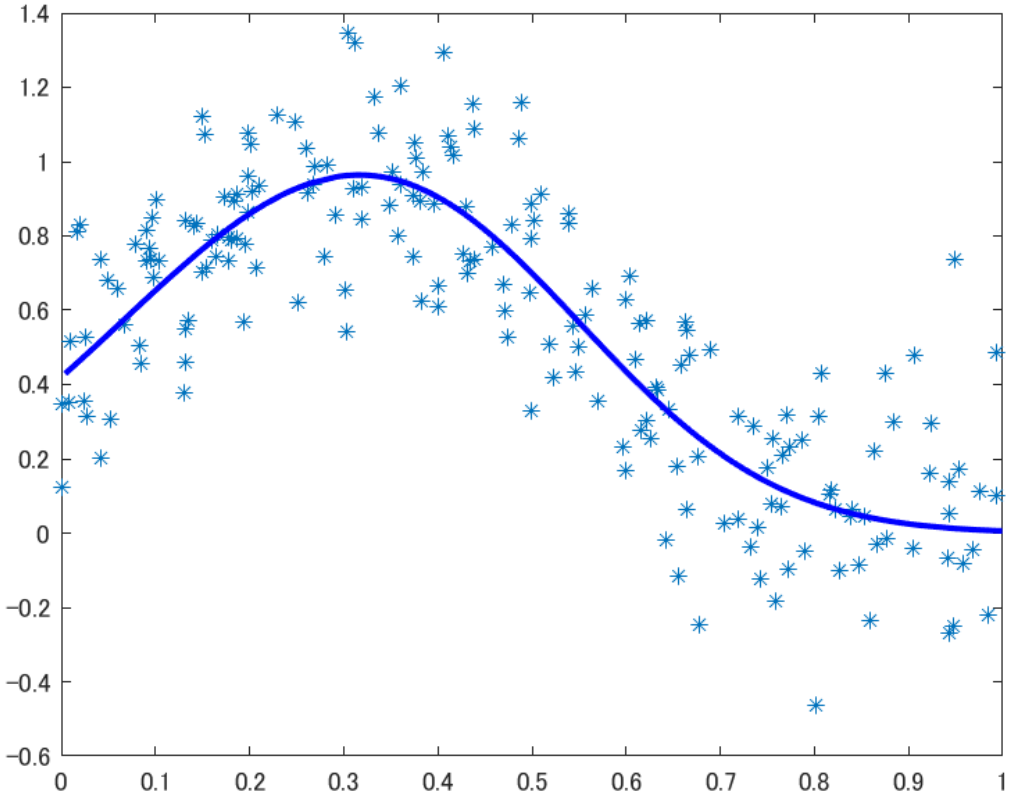


$a_1\eta(b_1x + c_1)$
$+ a_2\eta(b_2x + c_2)$

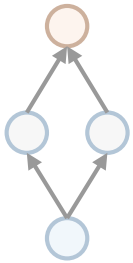


$$\sum_{j=1}^{3} a_j\eta(b_jx + c_j)$$

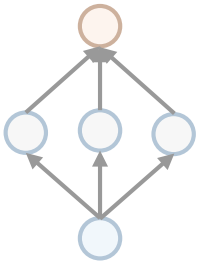
関数近似の様子



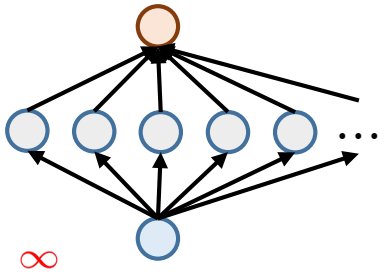
$a_1\eta(b_1x + c_1)$



$a_1\eta(b_1x + c_1) + a_2\eta(b_2x + c_2)$



$\sum_{j=1}^{3} a_j\eta(b_jx + c_j)$



$\sum_{j=1}^{\infty} a_j\eta(b_jx + c_j)$

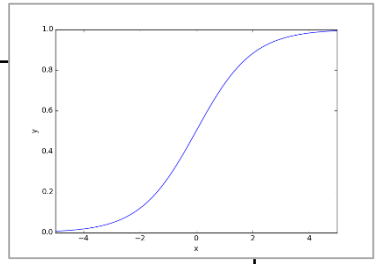
連続関数の近似

- Cybenkoの理論

[Cybenko: Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems,* 2(4): 303-314, 1989]

定義

活性化関数hがシグモイド的 $\Leftrightarrow$ $h(x) \to \begin{cases} 1 & (x \to \infty) \\ 0 & (x \to -\infty) \end{cases}$



定理

活性化関数hが連続なシグモイド的関数なら，任意の$f \in C([0,1]^d)$に対し，任意の$\epsilon > 0$において，ある$g(x) = \sum_{j=1}^{N} \alpha_i h(a_i x_i + b_i)$ 用いて，

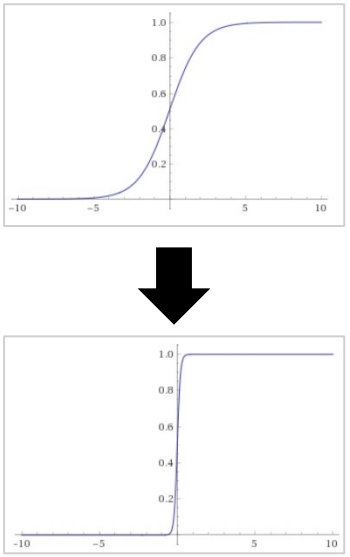
$$\sup_{x \in [0,1]^d} |f(x) - g(x)| \leq \epsilon$$

とできる.

証明の直感的概略

- シグモイド型の関数に対し，

$$h\left(a(\alpha^\top x+\beta)+\theta\right) \xrightarrow{a\to\infty} \begin{cases} 1 & (\alpha^\top x+\beta>0) \\ h(\theta) & (\alpha^\top x+\beta=0) \\ 0 & (\alpha^\top x+\beta<0) \end{cases}$$



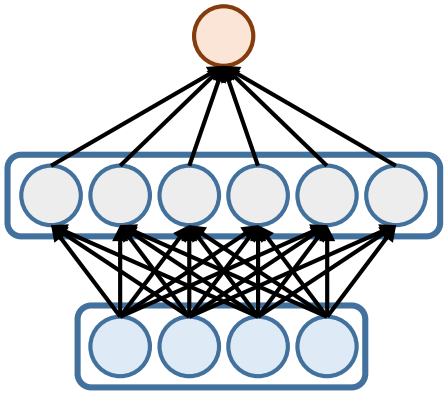
が成り立つ．つまり，スケールを適切に選べば，
階段関数をいくらでもよく近似できる．

- 階段関数を近似できれば，それらを足し引きすることで，
$\cos(\alpha^\top x+\beta)$ や $\sin(\alpha^\top x+\beta)$ をいくらでもよく近似できる．
- cos, sinが実現できるならFourier(逆)変換もできる．
- 任意の連続関数が近似できる．

積分表現

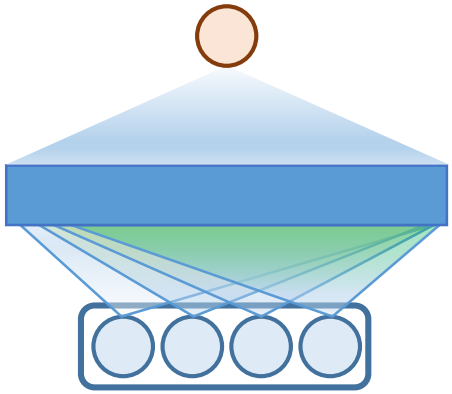
有限和近似（３層NN）

$$\hat{f}(x) = \sum_{j=1}^{m} v_j \eta(w_j^\top x + b_j) \quad \simeq$$



真の関数

$$f^o(x) = \int h^o(w,b)\eta(w^\top x + b)\mathrm{d}w\mathrm{d}b$$



(Sonoda & Murata, 2015)

- Ridgelet変換による解析（Fourier変換の親戚）
- ３層NNはridgelet変換で双対空間（中間層）に行ってから戻ってくる（出力層）イメージ

積分表現の概略（Ridgelet変換）

ある$\psi: \mathbb{R} \to \mathbb{R}$が存在して，以下の「許容条件」が成り立つとする：

$$K_{\psi,\eta} := (2\pi)^{d-1} \int_{\mathbb{R}\setminus\{0\}} \frac{\bar{\hat{\psi}}(\xi)\hat{\eta}(\xi)}{|\xi|^d} \mathrm{d}\xi < \infty$$

（$\hat{\psi}, \hat{\eta}$はFourie変換）

$$\mathscr{R}_\psi f(a,b) = \int_{\mathbb{R}^d} f(x)\overline{\psi(x^\top a - b)}\|a\|\mathrm{d}x$$

（Ridgelet変換）

$$\mathscr{R}_\eta^\dagger T(x) = \int_{a\in\mathbb{R}^d}\int_{\mathbb{R}} T(a,b)\eta(a^\top x - b)\|a\|^{-1}\mathrm{d}b\mathrm{d}a$$

（双対Ridgelet変換）

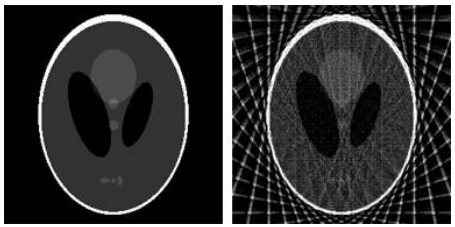
定理

$f, \hat{f} \in L^1(\mathbb{R}^d)$の時，許容条件を満たす$\eta, \psi$に対し以下がほとんどいたるところの$x \in \mathbb{R}^d$に対して成り立つ:

$$(\mathscr{R}_\eta^\dagger \mathscr{R}_\psi f)(x) = K_{\psi,\eta} f(x).$$

なお，連続点においては等式が常に成り立つ.

つまり，$f(x) = \int \frac{T(a,b)}{K_{\psi,\eta}\|a\|}\eta(a^\top x - b)\mathrm{d}a\mathrm{d}b$ と書ける.



Ridgelet変換
= Radon変換
+Wavelet変換

カーネル法との関係

リッジ回帰

- カーネル法のアイディア：
  - 機械学習には「内積」が頻繁に現れる.
    → 内積を“工夫”すれば非線形解析ができるはず.

- 例：リッジ回帰（Tikhonov正則化）

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \|X\beta - Y\|_2^2 + \lambda_n \|\beta\|_2^2$$

変数変換:

- 正則化項のため，$\hat{\beta} \in \mathrm{Ker}(X)^{\perp}$．つまり，$\hat{\beta} \in \mathrm{Im}(X^{\top})$.
- ある $\hat{\alpha} \in \mathbb{R}^n$ が存在して，$\hat{\beta} = X^{\top}\hat{\alpha}$ と書ける.

$$\hat{\alpha} \leftarrow \arg\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \|XX^{\top}\alpha - Y\|_2^2 + \lambda_n \alpha^{\top}(XX^{\top})\alpha.$$

新しい入力xに対しては $y = x^{\top}X^{\top}\hat{\alpha}$ で予測.

- リッジ回帰の変数変換版

$$\hat{\alpha} \leftarrow \underset{\alpha \in \mathbb{R}^n}{\arg\min} \frac{1}{n} \|XX^\top \alpha - Y\|_2^2 + \lambda_n \alpha^\top (XX^\top) \alpha.$$

※ $(XX^\top)_{i,j} = x_i^\top x_j$ はx_iとx_jの内積.

- **カーネル法のアイディア**

  x の間の内積を他の関数で置き換える:

$$x_i^\top x_j \;\rightarrow\; k(x_i, x_j)$$

この $k : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ をカーネル関数と呼ぶ.

カーネル関数の満たすべき条件

- 対称性：$k(x, x') = k(x', x)$
- 正値性：$\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j k(x_i, x_j) \geq 0, \quad \forall(\{x_i\}_{i=1}^m, \{\alpha_i\}_{i=1}^m, m)$

この条件さえ満たしていればなんでも良い

カーネルリッジ回帰

カーネルリッジ回帰: $K = (k(x_i, x_j))_{i,j=1}^n$ として,

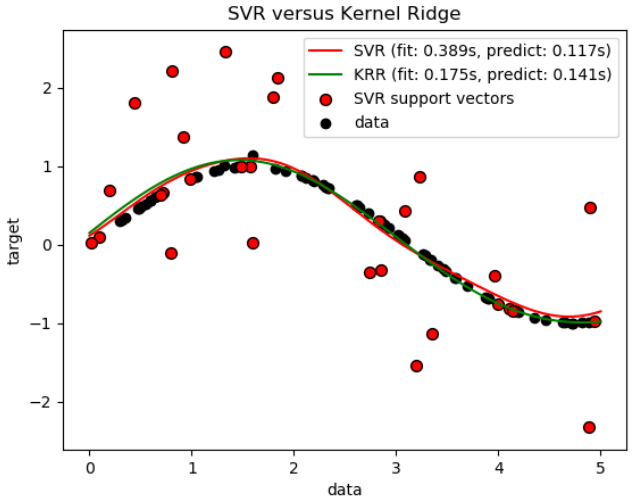
$$\hat{\alpha} \leftarrow \underset{\alpha \in \mathbb{R}^n}{\arg\min} \frac{1}{n} \|K\alpha - Y\|_2^2 + \lambda_n \alpha^\top K \alpha.$$

新しい入力 x に対しては,

$$y = \sum_{i=1}^{n} k(x, x_i)\hat{\alpha}_i$$

で予測.

線形代数で
非線形な回帰を実現.



[https://scikit-learn.org/stable/auto_examples/plot_kernel_ridge_regression.html]

カーネル関数の例

- ガウシアンカーネル

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

- 多項式カーネル

$$k(x, x') = \left(1 + x^\top x'\right)^p$$

- χ^2-カーネル

$$k(x, x') = \exp\left(-\gamma^2 \sum_{j=1}^{d} \frac{(x_j - x'_j)^2}{(x_j + x'_j)}\right)$$

- Matérn-kernel

$$k(x, x') = \int_{\mathbb{R}^d} e^{\mathrm{i}\lambda^\top (x - x')} \frac{1}{(1 + \|\lambda\|^2)^{\alpha + d/2}} \mathrm{d}\lambda$$

- グラフカーネル，時系列カーネル，...
(ユークリッド空間でなくてもカーネルが定義できればカーネル法が使える)

再生核ヒルベルト空間

カーネル関数 $k(x, x')$ ⇔ 再生核ヒルベルト空間 (RKHS) $\mathcal{H}_k$

定義

集合Ω上の再生核ヒルベルト空間（Reproducing kernel Hilbert space, RKHS）$\mathcal{H}$ とは，Ω 上の関数からなるヒルベルト空間であって，任意の$x \in \Omega$ に対し$\phi_x \in \mathcal{H}$が存在し，

$$f(x) = \langle \phi_x, f \rangle_{\mathcal{H}} \quad (f \in \mathcal{H})$$

を満たすものをいう．

- $k(x, y) := \langle \phi_x, \phi_y \rangle_{\mathcal{H}}$は正定値対称カーネル関数
- 逆に正定値対称カーネルが与えられたら対応するRKHSが一意に存在

定理 (Moore-Aronszajnの定理)

$k(x, y)$: 正定値対称カーネル (given)

Ω上の関数からなるヒルベルト空間$\mathcal{H}_k$で以下の条件を満たすものが一意に存在：

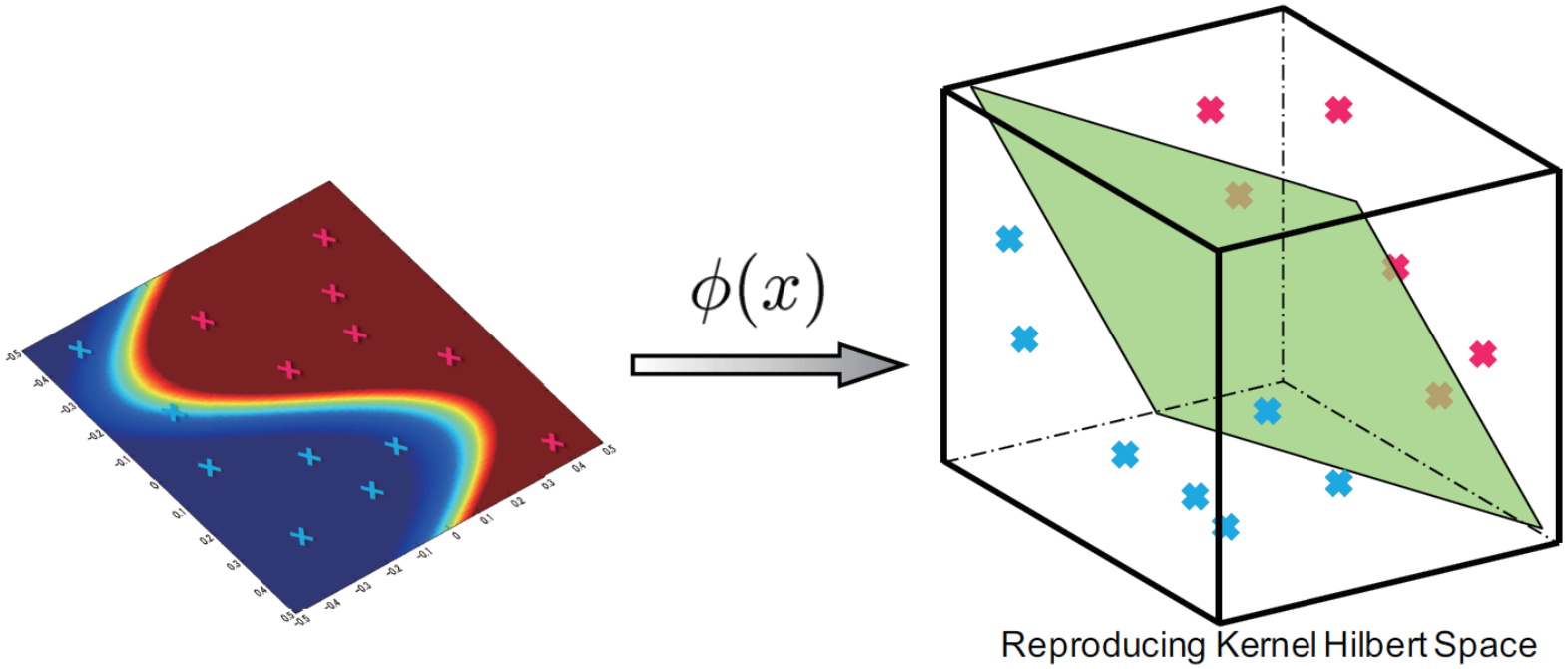
1. $k(x,\cdot) \in \mathcal{H}_k$
2. $f = \sum_{i=1}^{n} \alpha_i k(x_i,\cdot)$なる有限和は$\mathcal{H}_k$内で稠密
3. 再生成：

$$f(x) = \langle k(x,\cdot), f \rangle_{\mathcal{H}_k} \ (\forall x \in \Omega, \forall f \in \mathcal{H}_k).$$

$$f(x) = \left\langle \sum_{i=1}^{n} \alpha_i k(x_i,\cdot), k(x,\cdot) \right\rangle_{\mathcal{H}_k} = \sum_{i=1}^{n} \alpha_i k(x_i, x)$$

再生核ヒルベルト空間のイメージ

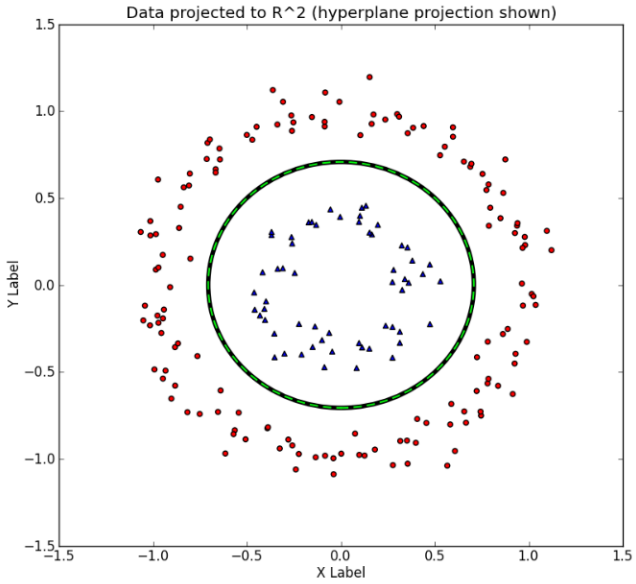
- 高次元(無限次元)特徴空間にϕで写像して推論を行う.
- 再生核ヒルベルト空間では線形な処理が元の空間では非線形処理になる.

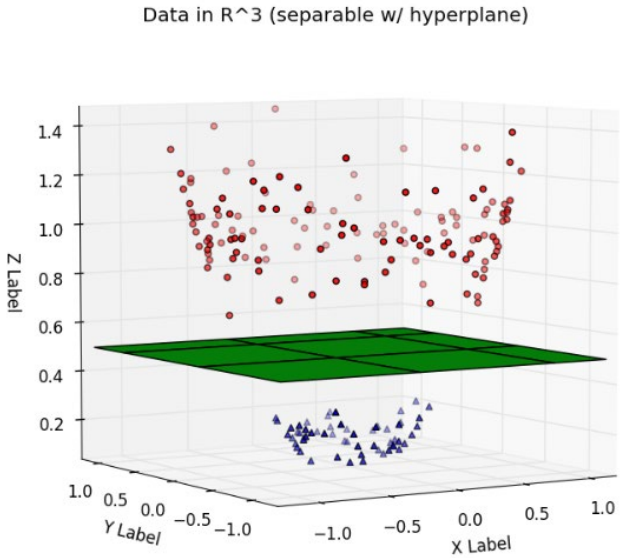


$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$$

多項式カーネル

非線形写像 ϕ_x





http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

カーネルリッジ回帰の再定式化

- 再生性: $f \in \mathcal{H}_k$ に対し

$$f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}_k}.$$

- カーネルリッジ回帰

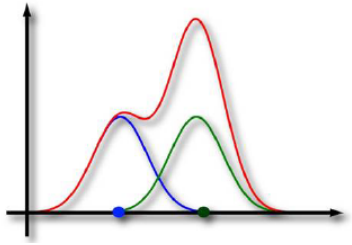
$$\hat{f} \leftarrow \min_{f \in \mathcal{H}_k} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + C\|f\|_{\mathcal{H}_k}^2$$

- 表現定理

$$\exists \alpha_i \in \mathbb{R} \quad \text{s.t.} \quad \hat{f}(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x),$$

$$\Rightarrow \quad \|\hat{f}\|_{\mathcal{H}_k} = \sqrt{\textstyle\sum_{i,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j)} = \sqrt{\boldsymbol{\alpha}^\top K \boldsymbol{\alpha}}.$$

さきほどのカーネルリッジ回帰の定式化と一致.



カーネル関数の分解とRKHSの表現

- カーネル関数は対象行列の対角化に対応する分解を許す.

定理

ある非負測度νに対して, $\int_\Omega k(x,x)\mathrm{d}\nu(x) < \infty$ が成り立つなら,

高々加算個の正実数列$(\mu_i)_{i\in I}$および$L_2(\nu)$内の正規直交基底$(e_i)_{i\in I}$が存在して,

$$k(x,x') = \sum_{i\in I} \mu_i e_i(x) e_i(x') \qquad (\forall x, x')$$

と分解できる (各点収束).

- このような分解は他にもいろいろとバージョンがある, e.g., Mercer展開.
  (詳細は[Steinwart&Scovel: *Constructive Approximation*, 35(3):363—417, 2012])
- さらに$\left(\sqrt{\mu_i}\, e_i\right)_{i\in I}$はRKHS内の正規直交基底になる.

$$\mathcal{H}_k = \left\{ f(x) = \sum_{i\in I} \alpha_i \sqrt{\mu_i} e_i(x) \mid \sum_{i\in I} \alpha_i^2 < \infty \right\}$$

$$\|f\|_{\mathcal{H}_k} = \sqrt{\textstyle\sum_{i\in I} \alpha_i^2}, \quad \langle f, f' \rangle_{\mathcal{H}_k} = \sum_{i\in I} \alpha_i \alpha_i'$$

カーネル回帰の再定式化

$$\mathcal{H}_k = \left\{ f(x) = \sum_{i \in I} \alpha_i \sqrt{\mu_i} e_i(x) \mid \sum_{i \in I} \alpha_i^2 < \infty \right\}$$

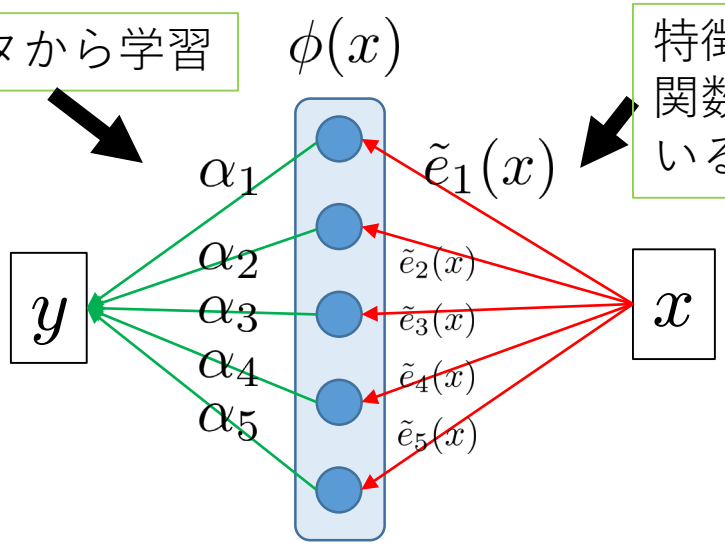
$$\|f\|_{\mathcal{H}_k} = \sqrt{\textstyle\sum_{i \in I} \alpha_i^2}, \quad \langle f, f' \rangle_{\mathcal{H}_k} = \sum_{i \in I} \alpha_i \alpha_i'$$

$$\min_{\alpha} \quad \sum_{i=1}^{n} \left(y_i - \sum_{j \in I} \alpha_j \tilde{e}_j(x_i) \right)^2 + \lambda_n \sum_{j \in I} \alpha_j^2$$

$$\tilde{e}_j(x) = \sqrt{\mu_j} e_j(x)$$

係数はデータから学習

特徴抽出層はカーネル関数によって定まっている.



カーネル法は
- 中間層が無限次元RKHS
- 第一層は固定
- 第二層を学習

であるニューラルネットに対応

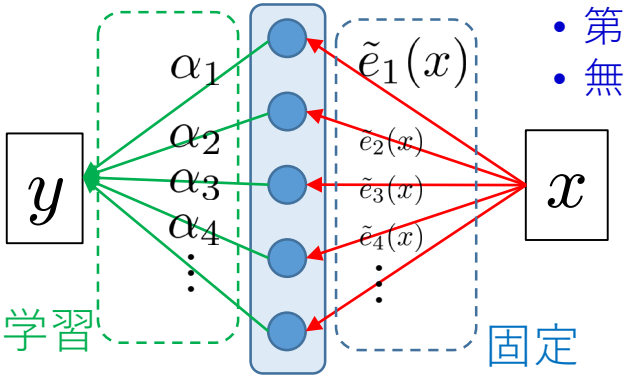
カーネル法

- 浅い手法の代表格.
- これも万能近似能力がある.

第1層目を固定した横幅無限の2層ニューラルネットワーク

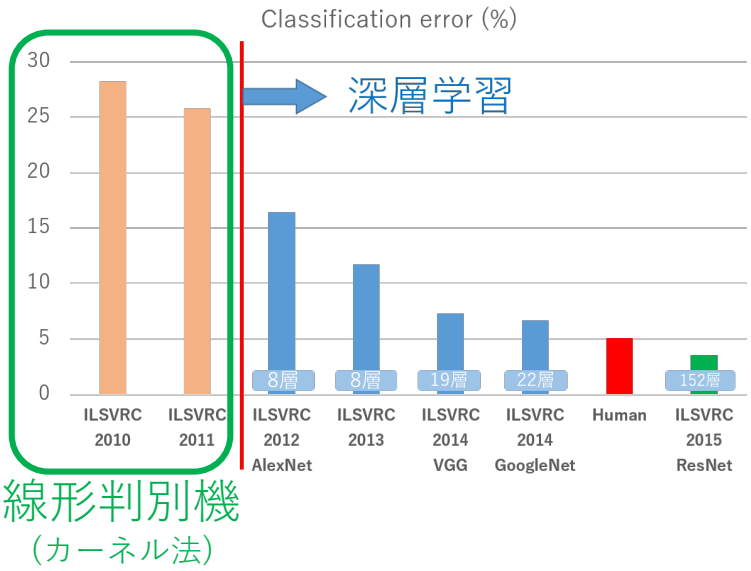
似た手法
- スプライン法
- 局所多項式回帰
- シリーズ推定量

(線形推定量と呼ばれるクラス)



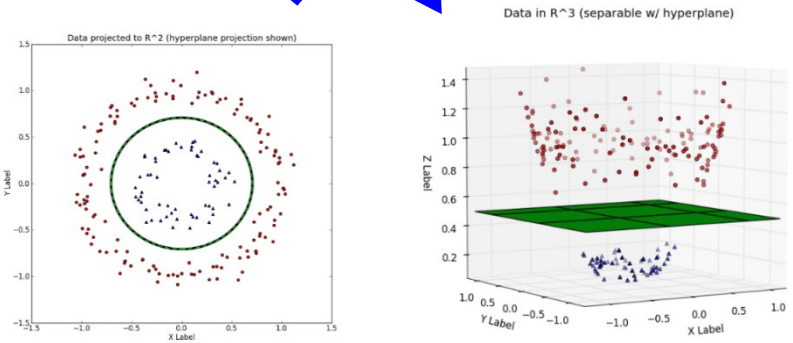
- 第一層目固定
- 無限個の素子

$$k(x,x') = \sum_{m=1}^{\infty} \tilde{e}_m(x)\tilde{e}_m(x') : \text{カーネル関数}$$



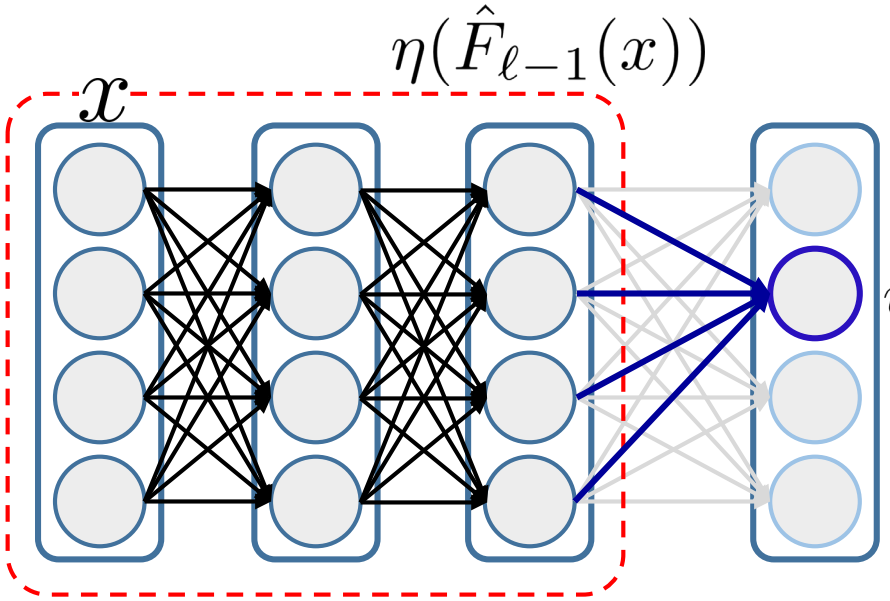
再生核ヒルベルト空間の理論

非線形写像 ϕ_x



深層NNとカーネル関数

深層NNは「カーネル関数をデータに合わせて学習する方法」と言える



$$\hat{k}_\ell(x, x') = \eta(\hat{F}_{\ell-1}(x))^\top \eta(\hat{F}_{\ell-1}(x'))$$

$\hat{k}_\ell$に対応した再生核ヒルベルト空間

$$\mathcal{H}_\ell = \{f(x) = w^\top \eta(\hat{F}_{\ell-1}(x)) \mid w \in \mathbb{R}^{m_\ell}\},$$

$$\|f\|_{\mathcal{H}_\ell} = \|w\|. \qquad \text{（より正確には同じ}f\text{を与える}w\text{の中でinfを取る）}$$

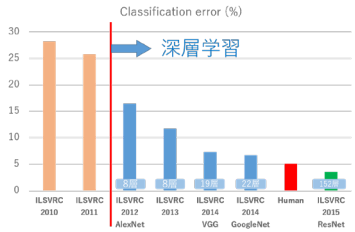
See Montavon et al. (2011); Bach (2015)

- 深層学習は各層でカーネル関数を逐次的に構築する学習方法であるとも言える.
- データから適応的にカーネル関数を学習する点が通常のカーネル法と異なる.

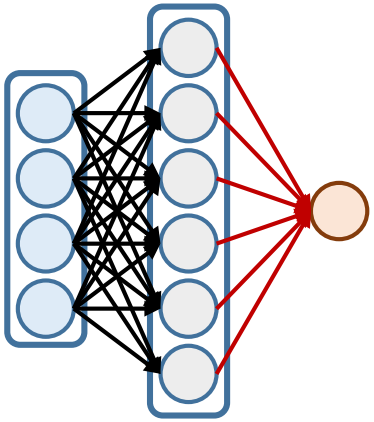
これまででわかったこと

- **[理論]** 万能近似能力という意味では浅層で十分.
- **[実際]** 実際は多層を使うことが多い.

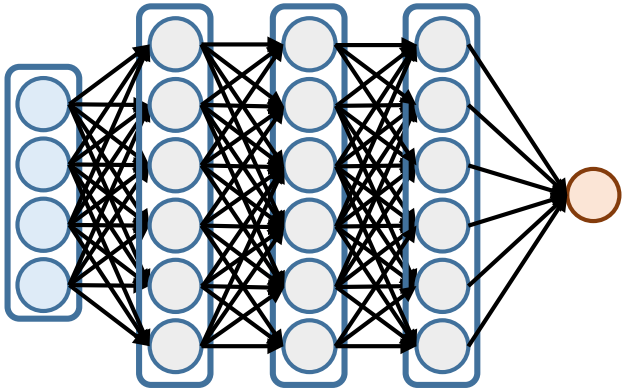
→　この差はどう埋める？



カーネル法
浅層



多層ニューラルネット
深層学習



→「表現力」を比べてみる.

万能近似理論より二層ニューラルネットでも中間層のユニット数を無限に増やせば任意の関数を任意の精度で近似できる.

歴史的には後にSVMの理論に繋がってゆく.

(例：Gaussian kernelの万能性)

Q：ではなぜ深い方が良いのか？

A：深さに対して**指数的に表現力が増大**するから.

表現力と層の数

NNの“表現力”：領域を何個の多面体に分けられるか？

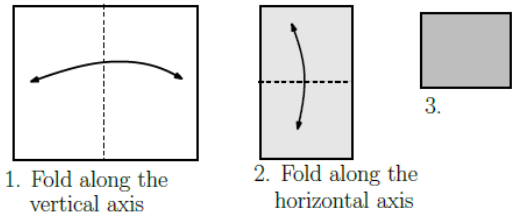
- 層の数に対して表現力は**指数的**に上がる.

$$\left(\frac{n}{n_0}\right)^{L-1}\sum_{j=0}^{n_0}\binom{n}{j}$$

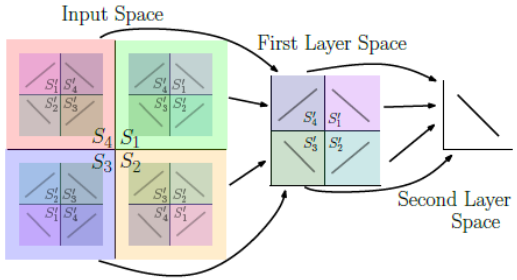
- 中間層のユニット数 (横幅) に対しては**多項式的**.

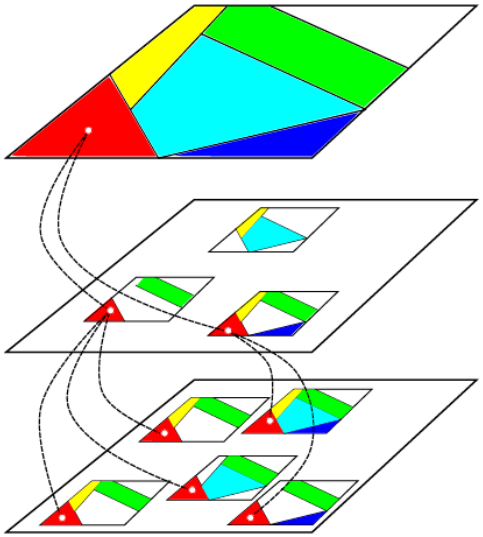
$$\sum_{j=0}^{n_0}\binom{n}{j}$$

L：層の数
n：中間層の横幅
n_0：入力の次元



(a)





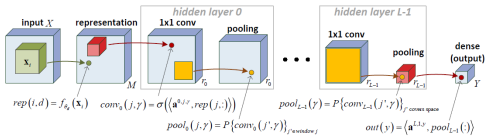
折り紙のイメージ

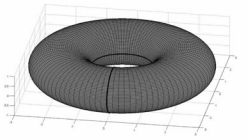
Montufar, Guido F., et al. "On the number of linear regions of deep neural networks." 2014.

多層で得する理由

他にも同様の結論を出している論文多数

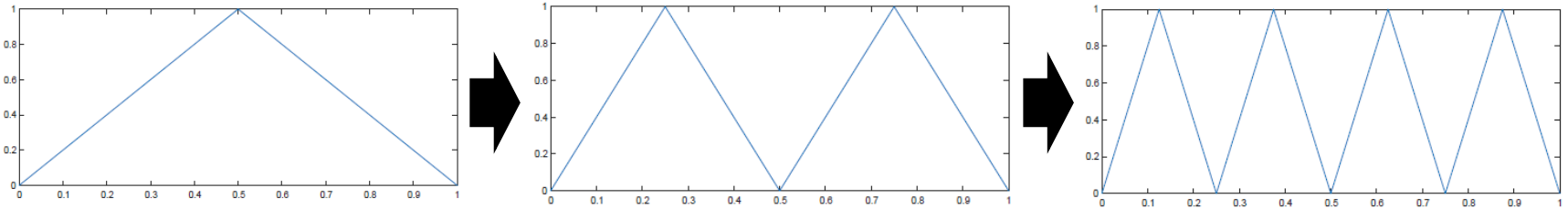
- **多項式展開，テンソル解析** [Cohen et al., 2016; Cohen & Shashua, 2016]
  単項式の次数
- **代数トポロジー** [Bianchini & Scarselli, 2014]
  ベッチ数(Pfaffian)
- **リーマン幾何 + 平均場理論** [Poole et al., 2016]
  埋め込み曲率





対称性の高い関数は，特に層を深くすることで得をする．

$$h(x) = \begin{cases} 2x & (0 \le x \le 1/2) \\ 2(1-x) & (1/2 \le x \le 1) \\ 0 & (\text{otherwise}). \end{cases}$$

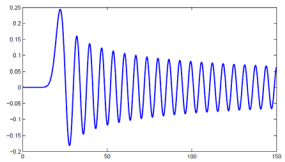


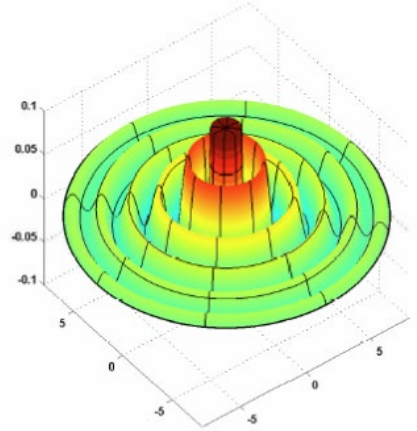
$h(x)$　　$h \circ h(x)$　　$h \circ h \circ h(x)$

多層が得する例

$$g(\|x\|^2) = g(x_1^2 + x_2^2 + \cdots + x_{d_x}^2)$$

gはBessel関数を元に構成



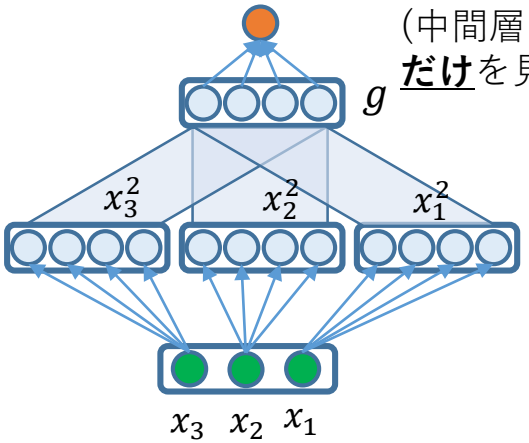


三層（中間層二層）：$O(\mathrm{poly}(d_x))$ノードで十分
二層（中間層一層）：$\Omega(\exp(d_x))$ノードが必要

(Eldan&Shamir, 2016)

三層

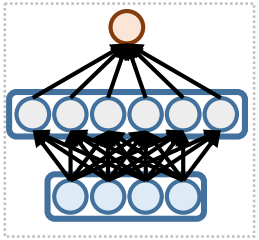
まず二乗和$x_1^2 + \cdots + x_{d_x}^2$を作ってからgを作用.

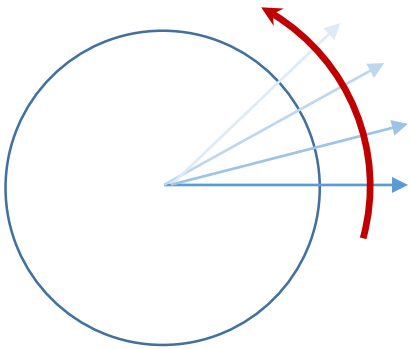


（中間層で**座標軸方向だけ**を見ればよい）

二層

全方向をケアする必要がある
（座標軸方向だけではダメ）

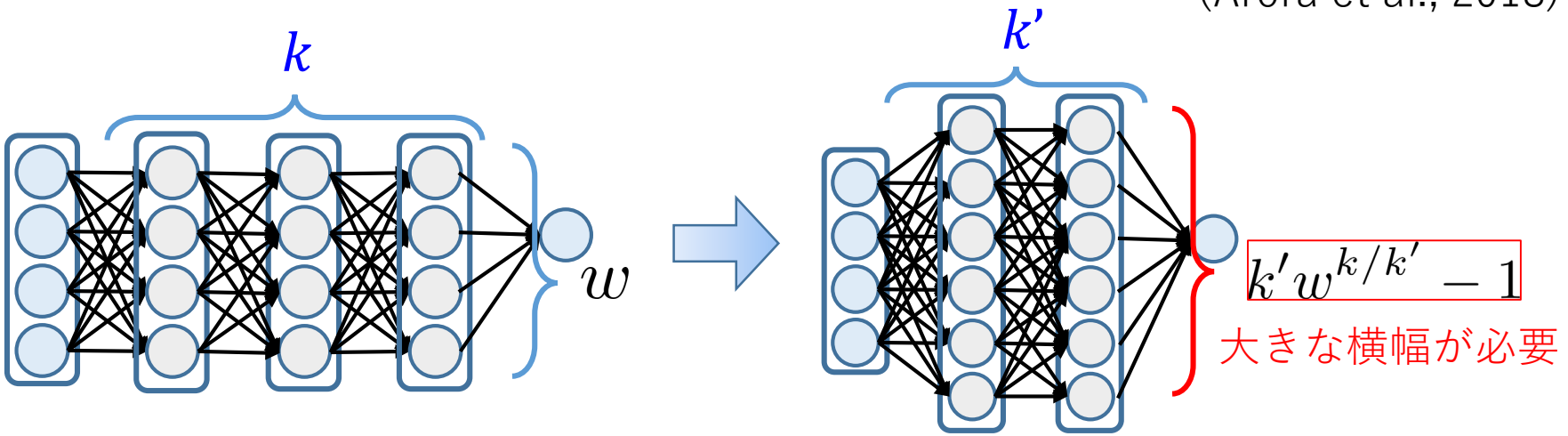




区分線形関数の表現

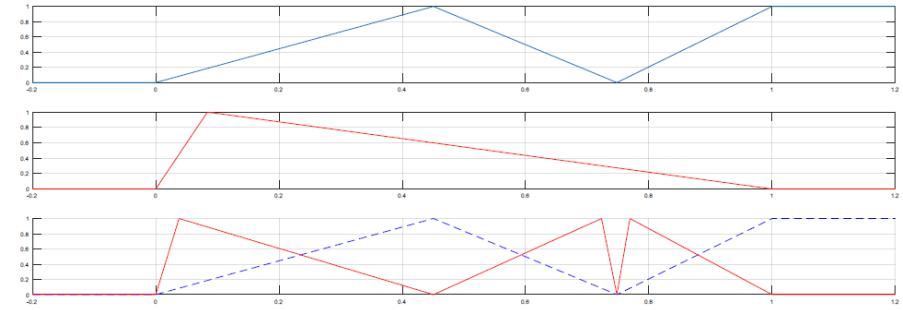
- 任意の区分線形関数($\mathrm{R}^d \to \mathrm{R}$)は深さ$\lceil \log_2(d+1) \rceil$のReLU-DNNで表現可能
- ある横幅w，縦幅kのReLU-DNNが存在して，それを縦幅$k'(<k)$のネットワークで表現するには横幅$k' w^{k/k'} - 1$が必要.

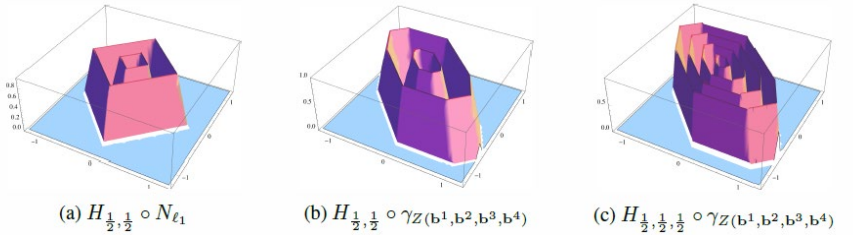
(Arora et al., 2018)



やはり層の深さに対し指数関数的に表現力が増加

上記のネットワークの例





有理関数の近似

- 有理関数をReLU-DNNで近似

$$p : [0,1]^d \to [-1,+1],\ q : [0,1]^d \to [2^{-k},+1] \quad \text{：r次多項式}$$

p/q をReLU-DNNで近似したい

あるReLU-DNNfが存在してノード数と近似誤差が次のように抑えられる：

ノード数

$$\mathrm{O}(\mathrm{poly}(k,r,d)\mathrm{poly}(\log(1/\epsilon)))$$

近似誤差

$$\sup_{x\in[0,1]^d}\left|f(x)-\frac{p(x)}{q(x)}\right|\le\epsilon$$

- ReLU-DNNを有理関数で近似

k-層で各層のノード数mの任意のReLU-DNNfに対しては，
次数と近似誤差が以下で抑えられる有理関数p/qが存在：

次数（分母qと分子pの次数の最大値）

$$\mathrm{O}(\log(k/\epsilon)^k m^k)$$

深さに対して指数的に増大

近似誤差

$$\sup_{x\in[0,1]^d}\left|f(x)-\frac{p(x)}{q(x)}\right|\le\epsilon$$

- ReLU-DNNを多項式で近似：　$\Omega(\mathrm{poly}(1/\epsilon))$ の次数が必要

→有理関数に比べて表現力が低い

有理関数の近似

- 有理関数をReLU-DNNで近似

$$p : [0,1]^d \to [-1,+1],\ q : [0,1]^d \to [2^{-k},+1] \quad : \text{r次多項式}$$

p/q をReLU-DNNで近似したい

あるReLU-DNN f が存在してノード数と近似誤差が次のように抑えられる：

ノード数

$$\mathrm{O}(\mathrm{poly}(k,r,d)\mathrm{poly}(\log(1/\epsilon)))$$

近似誤差

$$\sup_{x\in[0,1]^d}\left|f(x)-\frac{p(x)}{q(x)}\right|\le\epsilon$$

- ReLU-DNI

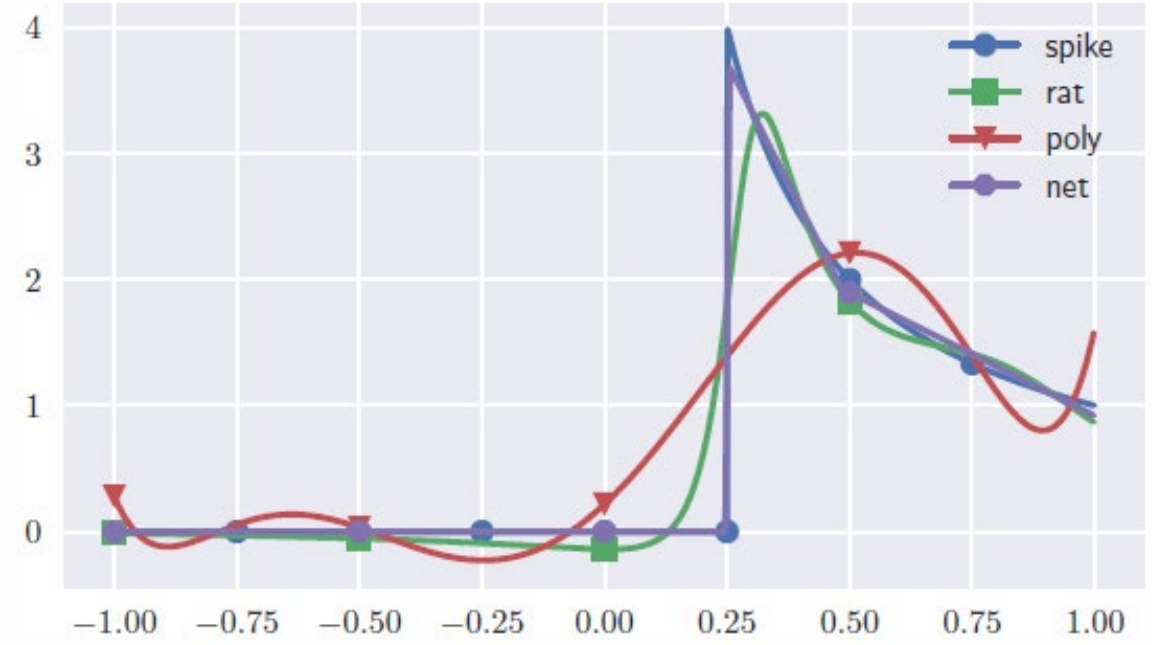
k-層で各

次数と近

次数（分母

O(lc

$$\left|f(x)-\frac{p(x)}{q(x)}\right|\le\epsilon$$



- ReLU-DNI

必要

→有理関数に比べて表現力が低い

統計的推定理論による比較

深層 vs 浅層 の統計理論

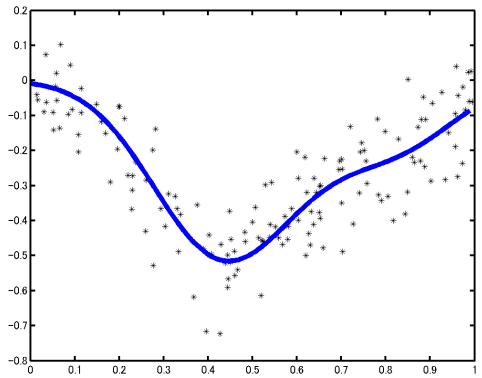
→「関数近似精度/推定精度」を比べてみる.

「多層」による特徴抽出と推定精度

ノンパラメトリック回帰の設定

$$y_i = f^{\circ}(x_i) + \xi_i \quad (i = 1, \ldots, n)$$

$\xi_i \sim N(0, \sigma^2)$は観測誤差



平均二乗誤差：

$$\mathbb{E}[\|\hat{f} - f^{\circ}\|^2_{L_2(P)}] < ?$$

※実はこれは二乗損失の平均余剰誤差になっている.

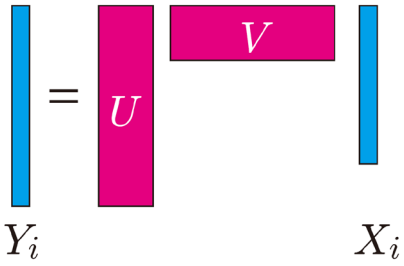
$$\mathbb{E}[L(\hat{\theta}) - \inf_f \mathbb{E}[\ell(Y, f(X))]]$$

なぜ深層学習が良いのか？

- 真の関数f^*の形状によって深層が有利になる

縮小ランク回帰

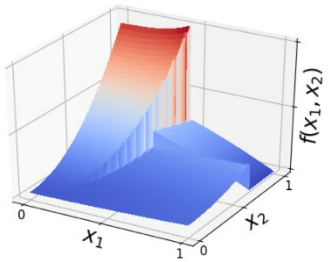
特徴空間の次元が低い状況は深層学習が得意



区分滑らかな関数

[Imaizumi&Fukumizu, 2019]

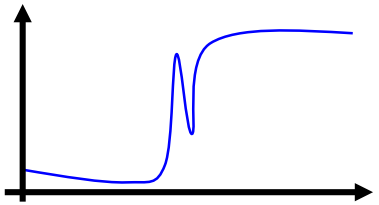
不連続な関数の推定は深層学習が得意



Besov空間

[Suzuki, 2019]

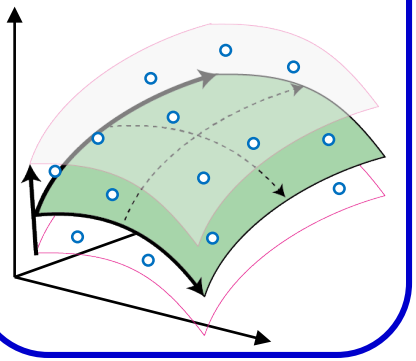
滑らかさが非一様な関数の推定は深層学習が得意



低次元データ

[Schmidt-Hieber, 2019] [Nakada&Imaizumi, 2019][Chen et al., 2019][Suzuki&Nitanda, 2019]

データが低次元部分空間上に分布していたら深層学習が有利



深層	$\frac{r(M+N)}{n}$	$n^{-\frac{2s}{2s+d}} \vee n^{-\frac{\alpha}{\alpha+D-1}}$	$n^{-\frac{2s}{2s+d}}$	$n^{-\frac{2s}{2s+D}}$
カーネル	$\frac{MN}{n}$	$\frac{1}{\sqrt{n}}$	$n^{-\frac{2s-2d(1/p-1/2)_+}{2s+d-2d(1/p-1/2)_+}}$	$n^{-\frac{2(s-D/p+d/2)}{2(s-D/p+d/2)+d}} \vee n^{-\frac{2s}{2s+D}}$

推定精度

深層学習の適応能力

深層学習には「**高い適応力**がある」

[Suzuki, ICLR2019]

明らかに犬

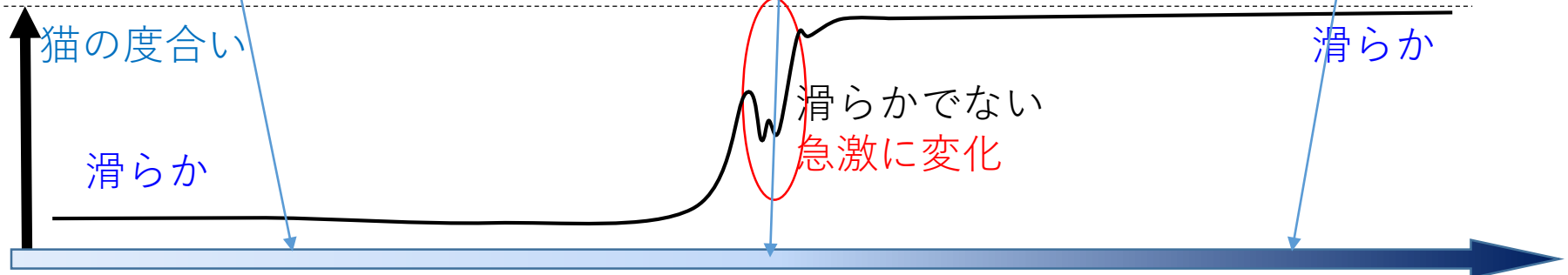
少し絵が変わっても
「犬」のまま

犬と猫の中間

少し絵が変わると
犬/猫のどちらかに偏る

明らかに猫

少し絵が変わっても
「猫」のまま



どこが重要でどこが重要でないかを見分けて，**重要な部分を重点的に学習**

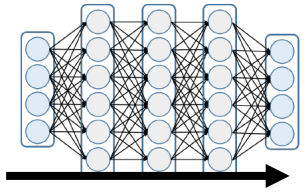
→ **多層だから可能**

深層学習はBesov空間($B^s_{p,q}$)の元を推定するのにミニマックス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)

「浅い」学習との比較

- 深層学習は場所によって解像度を変える適応力がある
  →学習効率が良い
- 浅い学習は様々な関数を表現できる基底を
  あらかじめ十分用意して“待ち構える”必要がある.
  →学習効率が悪い



[Suzuki, ICLR2019]

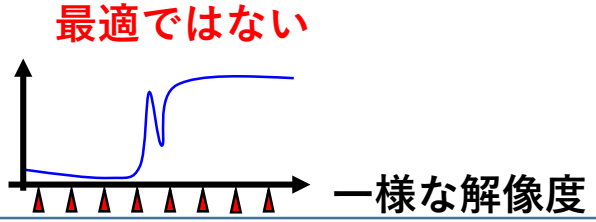
深層でない学習方法

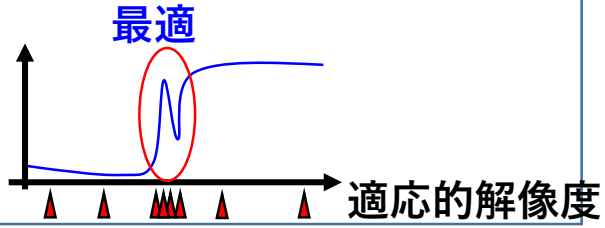
（カーネルリッジ回帰，KNN法，シーブ法など）

$$n^{-\frac{2s-2d(1/p-1/2)_+}{2s+d-2d(1/p-1/2)_+}} \gg n^{-\frac{2s}{2s+d}}$$

(n: sample size, p: uniformity of smoothness, s: smoothness)

深層学習





平均二乗誤差 $\mathrm{E}\left[\|\hat{f}-f^*\|^2\right]$ がサンプルサイズが増えるにつれ減少するレート

- ミニマックス最適性の意味で
- 理論上これ以上改善できない精度を達成できている.

各種関数クラスと
深層学習の近似/推定理論

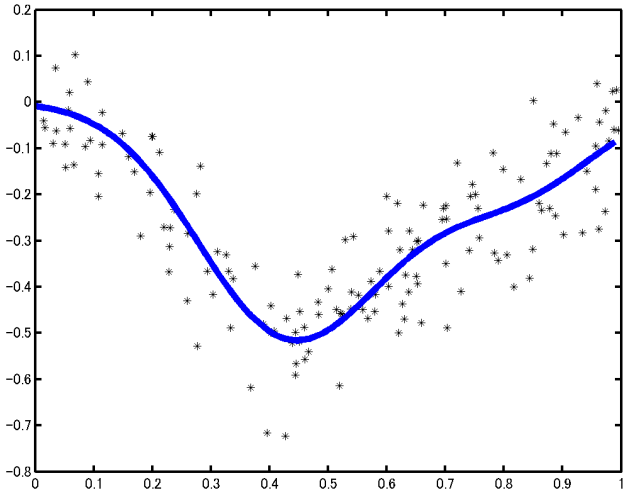
非線形回帰問題

非線形回帰モデル

$$y_i = f^{\mathrm{o}}(x_i) + \xi_i \qquad (i = 1, \ldots, n)$$

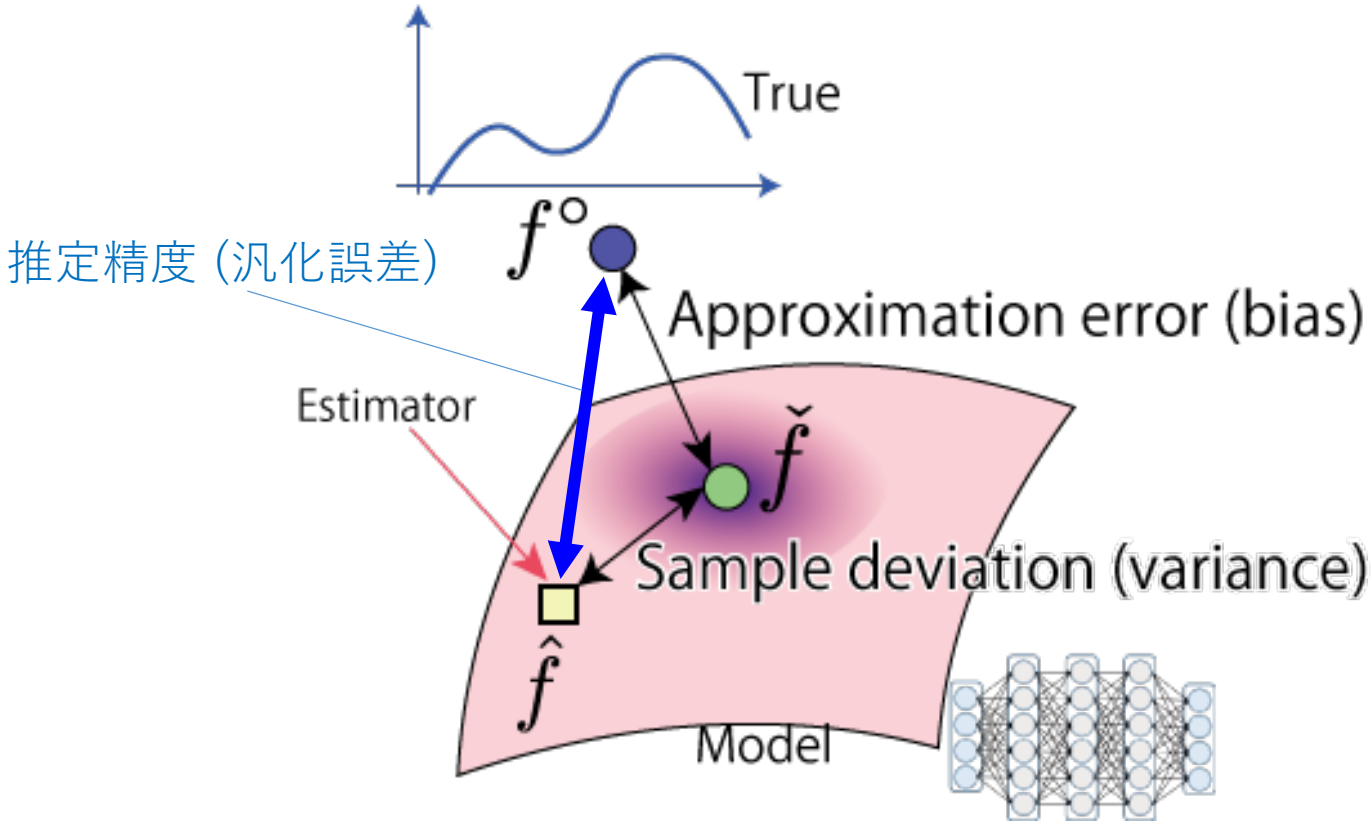
ただし，$\xi_i \sim N(0, \sigma^2)$かつ$x_i \sim P_X(X)$ (i.i.d.).

f^{o}をデータ$(x_i, y_i)_{i=1}^n$ から推定したい.



※ 以下の理論は判別問題でも展開可能

バイアスとバリアンスのトレードオフ



推定精度 = バイアス (モデル誤差) + バリアンス (分散)

重要な関数クラス

- Barronクラス
- Hölderクラス
- Sobolevクラス
- Besovクラス

真の関数が各クラスに含まれているときに
近似誤差はどれくらいになるか調べたい.

Hölder, Sobolev, Besov空間

$\Omega = [0,1]^d \subset \mathbb{R}^d$

- **Hölder space** $(\mathcal{C}^\beta(\Omega))$

$$\|f\|_{\mathcal{C}^\beta} = \max_{|\alpha|\le m} \left\|\partial^\alpha f\right\|_\infty + \max_{|\alpha|=m} \sup_{x\in\Omega} \frac{|\partial^\alpha f(x) - \partial^\alpha f(y)|}{|x-y|^{\beta-m}}$$

- **Sobolev space** $(W_p^k(\Omega))$

$$\|f\|_{W_p^k} = \left(\sum_{|\alpha|\le k} \|D^\alpha f\|_{L^p(\Omega)}^p \right)^{\frac{1}{p}}$$

- **Besov space** $(B_{p,q}^s(\Omega))$ $(0 < p, q \le \infty, 0 < s \le m)$

空間的非一様性

$$\omega_m(f,t)_p := \sup_{\|h\|\le t} \left\| \sum_{j=0}^m (-1)^{m-j} \binom{m}{j} f(\cdot + jh) \right\|_{L^p(\Omega)},$$

$$\|f\|_{B_{p,q}^s(\Omega)} = \|f\|_{L^p(\Omega)} + \left(\int_0^\infty [t^{-s}\omega_m(f,t)_p]^q \frac{\mathrm{d}t}{t} \right)^{1/q}.$$

滑らかさの度合い

- 直感：
  - 非整数回の微分も定義したい.
  - 整数回微分を“つなげる”→実補間
  - qはそのつなげ方を制御
  - sで関数の滑らかさを制御
  - pで滑らかさの空間的一様性を制御

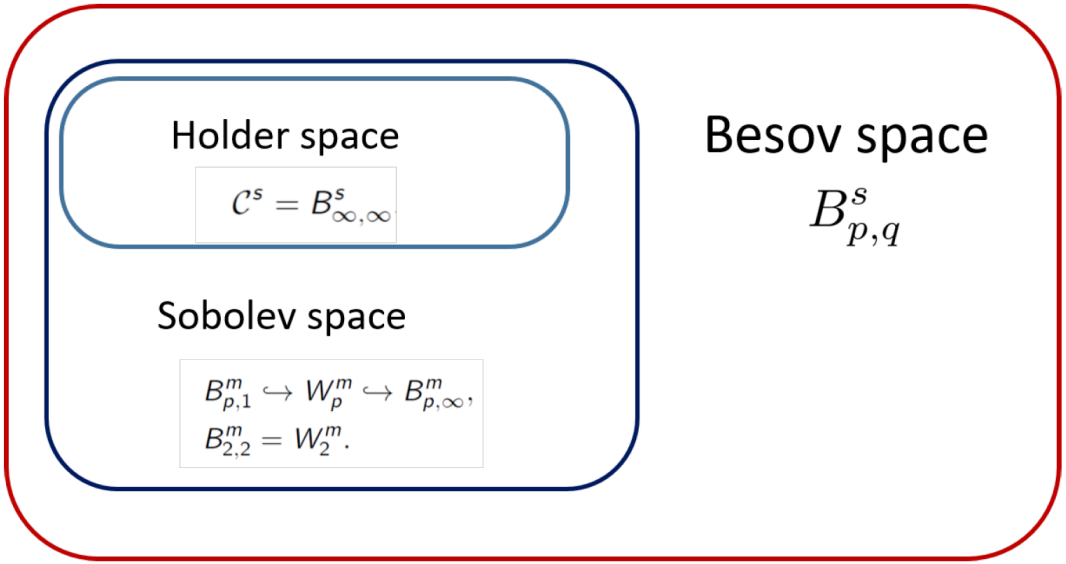
空間の間の関係

- For $m \in \mathbb{N}$,

$$B^m_{p,1} \hookrightarrow W^m_p \hookrightarrow B^m_{p,\infty},$$
$$B^m_{2,2} = W^m_2.$$

- For $0 < s < \infty$ and $s \notin \mathbb{N}$,

$$C^s = B^s_{\infty,\infty}.$$

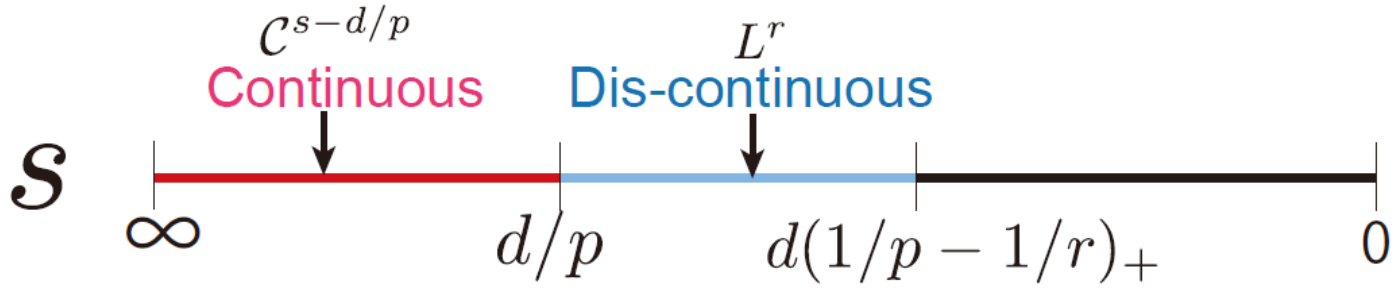


- 連続関数の領域： $s > d/p$

$$B^s_{p,q} \hookrightarrow C^0$$

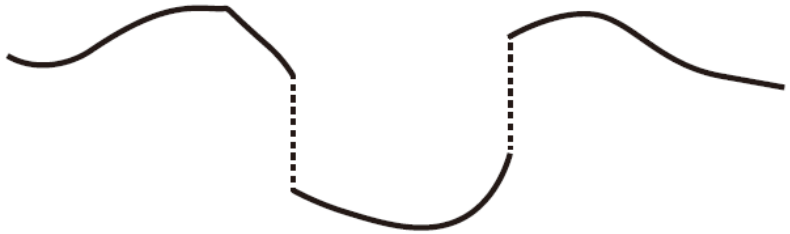
- L^r-可積分な領域：$s > d(1/p - 1/r)_+$

$$B^s_{p,q} \hookrightarrow L^r$$

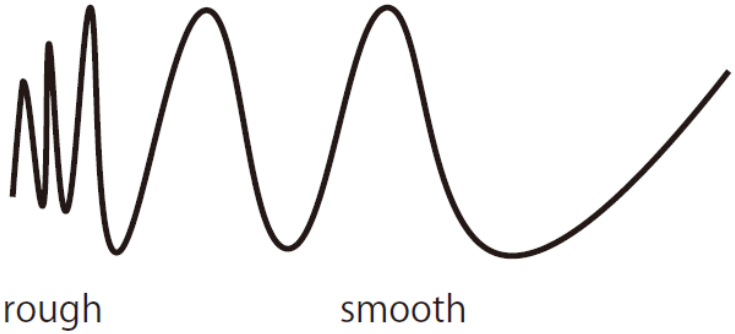


- 例： $B^1_{1,1}([0,1]) \subset \{\text{bounded total variation}\} \subset B^1_{1,\infty}([0,1])$

- 不連続な領域　$d/p > s$

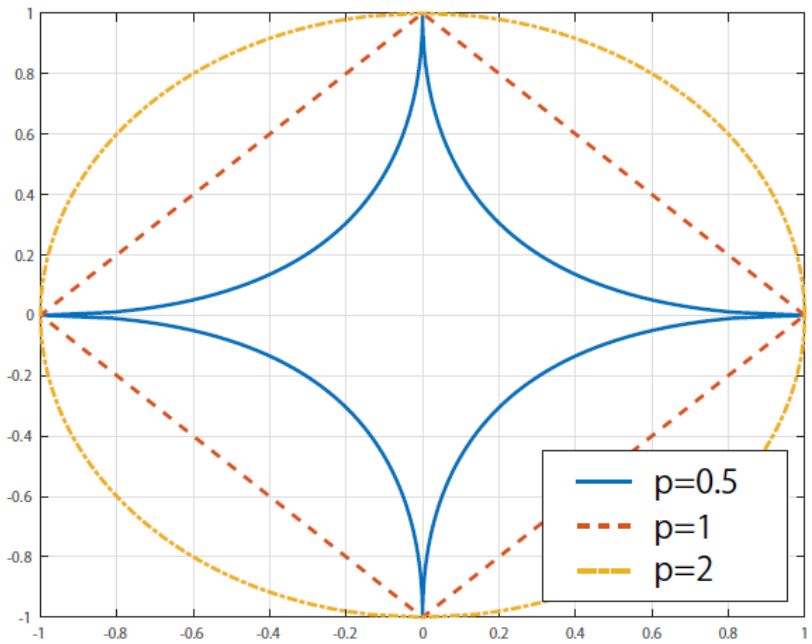


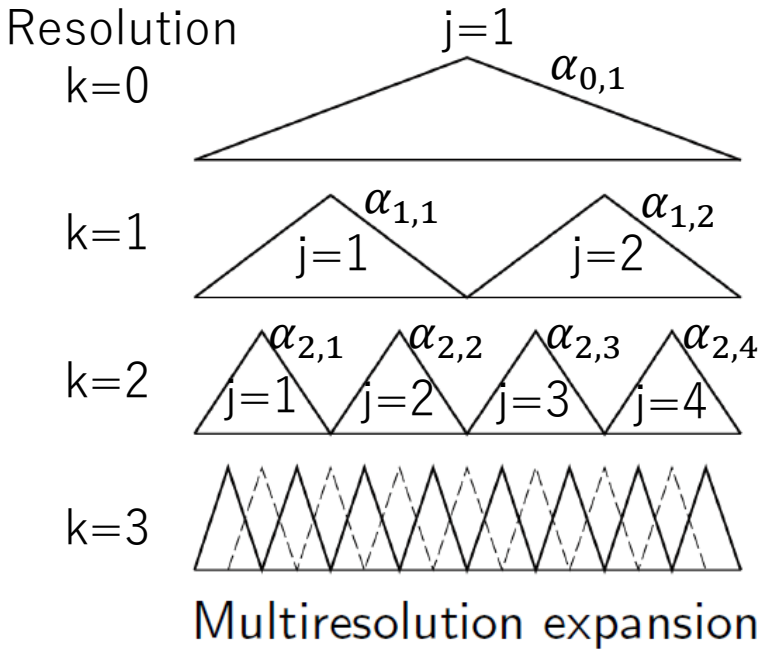
- 滑らかさが非一様的な場合：pが小さい状況



これらの性質にも関わらず深層学習は良い学習ができるか？

スパース性との関係





$$f = \sum_{k \in \mathbb{N}+} \sum_{j \in J(k)} \alpha_{k,j} \mathcal{N}_{k,j}^{(d)}$$

Wavelet基底による展開

$$\|f\|_{B^s_{p,q}} \simeq \left[\sum_{k=0}^{\infty} \{ 2^{sk} (2^{-kd} \sum_{j \in J(k)} |\alpha_{k,j}|^p)^{1/p} \}^q \right]^{1/q}$$

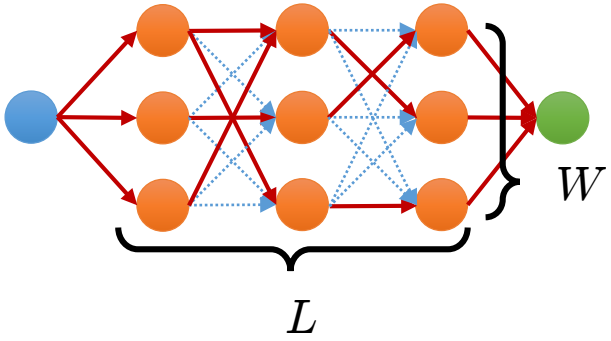
大体ℓ_p-ノルム

小さなp = スパースな係数
(非凸性)

→

空間的な滑らかさの
非一様性

深層学習のモデル



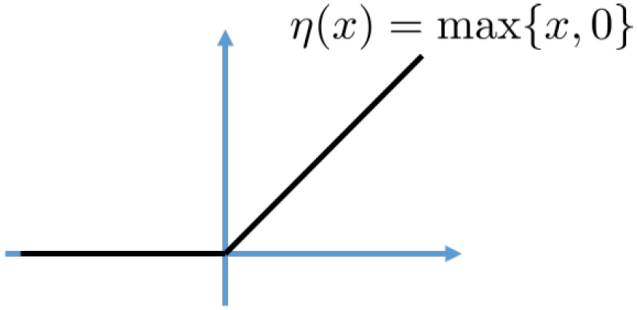
$$f(x) = (W^{(L)}\eta(\cdot) + b^{(L)}) \circ (W^{(L-1)}\eta(\cdot) + b^{(L-1)}) \circ \cdots \circ (W^{(1)}x + b^{(1)})$$

$\mathcal{F}(L, W, S, B)$

- 縦幅：L
- 横幅：W
- 枝の数：S
- 各パラメータの上限：B

の深層NNモデルの集合

- 活性化関数はReLUを仮定



関数近似能力

- $0 < p, q, r \leq \infty$と$0 < s < \infty$が以下を満たすとする:

$$s > d(1/p - 1/r)_+ \qquad (L^r\text{-可積分性})$$

- mを$s < \min\{m, m - 1 + 1/p\}$を満たす整数とする.

深層ニューラルネットワークの近似誤差

ある自然数Nと用いて深さL, 横幅W, 枝の数S, ノルム上界Bを以下のように定める：

$$L = O(\log(N)), \qquad W = O(N),$$
$$S = O(N\log(N)), \qquad B = O(N^{(d/p-s)_+}),$$

すると，深層NNは以下の誤差でBesov空間の元を近似できる：

大体パラメータ数

$$\sup_{f^o \in U(B^s_{p,q}([0,1]^d))} \inf_{\check{f} \in \mathcal{F}(L,W,S,B)} \left\| f^o - \check{f} \right\|_{L^r([0,1]^d)} \lesssim N^{-s/d}.$$

Pinkus (1999), Mhaskar (1996): $p = r$かつ$1 \leq p$, ReLU活性化関数ではない.
Petrushev (1998): $p = r = 2$, ReLU活性化関数ではない ($s \leq k + 1 + (d-1)/2$).

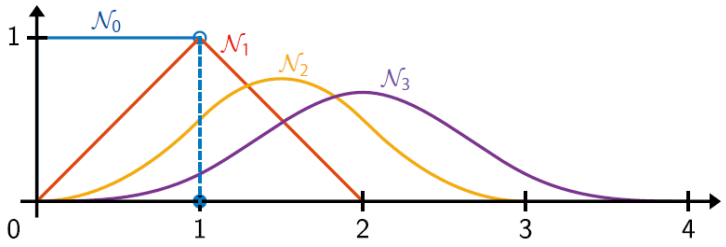
B-spline

$$\mathcal{N}(x) = \begin{cases} 1 & (x \in [0,1]), \\ 0 & (\text{otherwise}). \end{cases}$$

次数mのcardinal B-spline:

$$\mathcal{N}_m(x) = (\underbrace{\mathcal{N} * \mathcal{N} * \cdots * \mathcal{N}}_{m+1 \text{ times}})(x)$$

→ 区分多項式



$$\mathcal{N}_{k,j}^{(d)}(x_1, \ldots, x_d) = \prod_{i=1}^{d} \mathcal{N}_m(2^k x_i - j_i)$$

Cardinal B-splineによる展開 (DeVore & Popov, 1988)

- Atomic decomposition:

$f \in B^s_{p,q}$の必要十分条件:

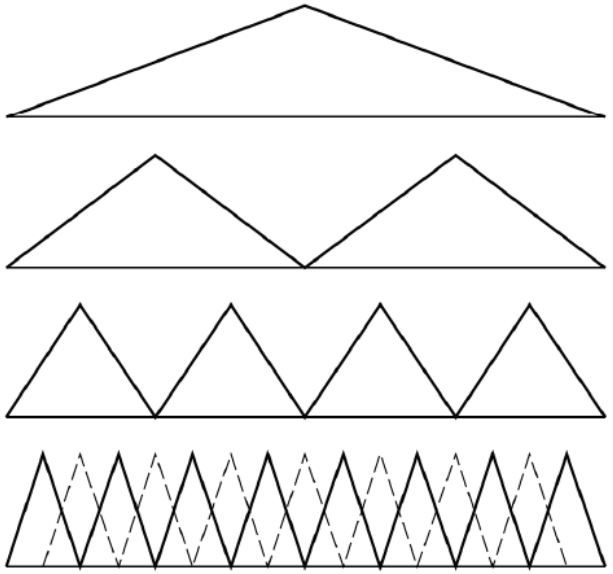
$$f = \sum_{k \in \mathbb{N}+} \sum_{j \in J(k)} \alpha_{k,j} \mathcal{N}^{(d)}_{k,j}$$

と分解できて
(ただし, $J(k) = \{j \in \mathbb{Z}^d \mid -m < j_i < 2^{k_i+1} + m\}$)

$$N(f) = \left[\sum_{k=0}^{\infty} \{2^{sk} (2^{-kd} \sum_{j \in J(k)} |\alpha_{k,j}|^p)^{1/p} \}^q \right]^{1/q} < \infty$$

- ノルムの同値性：

$$\|f\|_{B^s_{p,q}} \simeq N(f)$$



各B-spline基底をNNで近似

(see also Bolcskei, Grohs, Kutyniok, Petersen: Optimal Approximation with Sparsely Connected Deep Neural Networks. 2018)

基本戦略

- 真の関数f°が次のように展開できるとする:

$$f^\circ = \sum_{k=1}^{\infty} \sum_{j \in J(k)} \alpha_{k,j} \phi_{k,j}(x)$$

- 各基底関数$\phi_{k,j}$をReLU-NNでよく近似できるなら，f°も良く近似できる.

- Cardinal B-splineはReLU-NNでよく近似できる: $\log(1/\epsilon)$層で近似誤差ϵを達成する.

- B-splineに関する定理を深層学習に持ち込める.
  → Besov空間に限らない理論を展開可能

[Bölcskei et al.: arXiv:1705.01714]

比較

$s > d(1/p - 1/r)_+$ なる仮定のもとで

$$\inf_{\check{f} \in \mathcal{F}(L,W,S,B)} \sup_{f^o \in U(B^s_{p,q}([0,1]^d))} \|f^o - \check{f}\|_{L^r([0,1]^d)} \lesssim N^{-s/d}$$

- $p = q = \infty$の時，Yarotsky (2016) の結果に帰着 (Hölder空間)
- **適応的**な**非線形**近似が必要 (Dung, 2011)

線形近似 (Linear width)：

$$\begin{cases} N^{-s/d+(1/p-1/r)_+} & \begin{cases} \text{either} & (0 < p \le r \le 2), \\ \text{or} & (2 \le p \le r \le \infty), \\ \text{or} & (0 < r \le p \le \infty), \end{cases} \\ N^{-s/d+1/p-1/2} & (0 < p < 2 \end{cases}$$

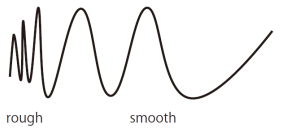
$p \neq r$が重要

非適応的近似（N-term approx., Ko

$$\begin{cases} N^{-s/d+(1/p-1/r)_+} & (1 < p < r \le 2,\ s > d(1/p - 1/r)), \\ N^{-s/d+1/p-1/2} & (1 < p < 2 < r \le \infty,\ s > d/p), \\ N^{-s/d} & (2 \le p < r \le \infty,\ s > d/2), \end{cases}$$

- 深層NNの適応能力
  特徴量の適切な抽出

Hölder空間では現れない性質

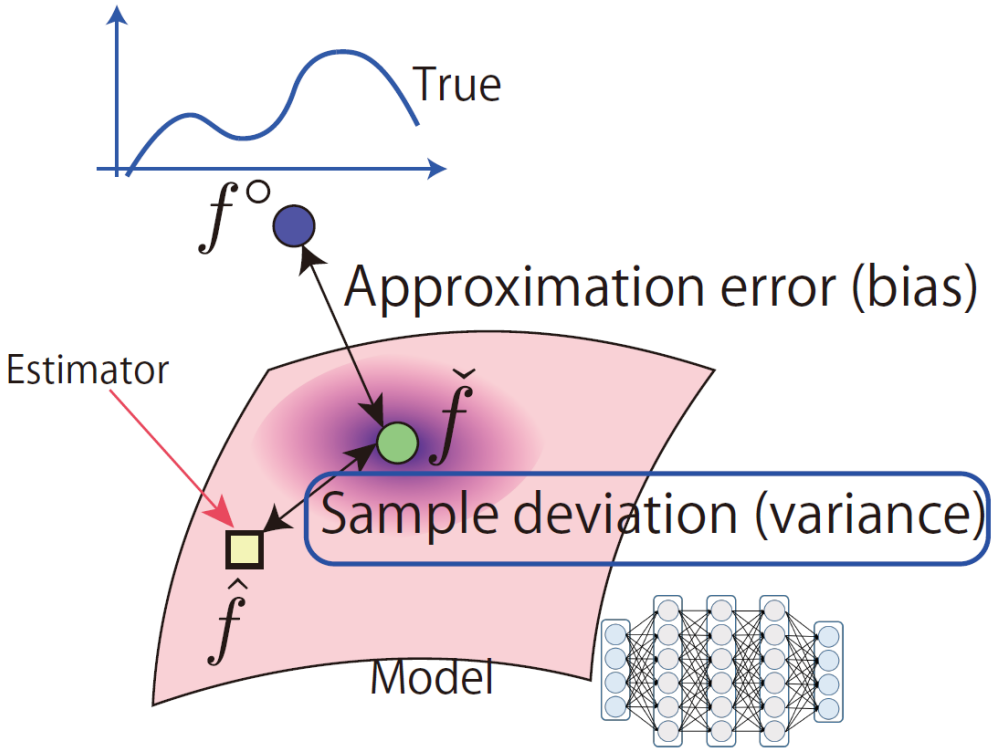


小さなp

関連研究

- Chui et al. (1994) and Bölcskei et al. (2017) dealt with a "smooth" activation with $\lim_{x\to\infty} \eta(x)/x^k \to 1$, $\lim_{x\to-\infty} \eta(x)/x^k = 0$ with $k \geq 2$ under $1 \leq p$. Mhaskar and Micchelli (1992) studied $s = k + 1$. Mhaskar (1993) studied $k \geq 2$ and $s = k + 1$, Mhaskar (1996) considered the Sobolev space W_p^m with a "bump" activation function (excluding ReLU).

バイアスとバリアンス分解



- これまで示したこと：バイアス（近似誤差）
- これから示すこと：経験誤差最小化のバリアンス

$$\hat{f} = \underset{f \in \mathcal{F}(L,W,S,B)}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - f(x_i))^2.$$

$$\mathrm{E}[\|f^{\circ} - \hat{f}\|^2_{L^2(P_X)}]$$
$$\lesssim \underbrace{\frac{S[L\log(BW) + \log(Ln)]}{n}}_{\text{Variance}} + \underbrace{\inf_{f \in \mathcal{F}(L,W,S,B)} \|f - f^{\circ}\|^2_{L^2(P_X)}}_{\text{Bias}}$$

深さ　横幅　スパース性（非零パラメータ数）　各パラメータの絶対値の上界

$$L = O(\log(N)), W = O(N), S = O(N\log(N)), B = O(N^{(d/p-s)_+})$$

なら

$$\text{Bias} = N^{-s/d}$$

（局所Rademacher complexityを用いて証明）

⇒ バイアスとバリアンスのトレードオフをバランスすればよい.

推定精度

- 最小二乗解 (訓練誤差最小化)

$$\hat{f} = \underset{\bar{f}: f \in \mathcal{F}(L,W,S,B)}{\arg\min} \sum_{i=1}^{n} (y_i - \bar{f}(x_i))^2$$

ただし, $\bar{f} = \min\{\max\{f, -F\}, F\}$ (clipping).

定理 (推定精度)

$\|f^\circ\|_{B^s_{p,q}} \leq 1, \|f^\circ\|_\infty \leq 1$ かつ $0 < p, q \leq \infty$, $s > d(1/p - 1/2)_+$ のとき, $N \asymp n^{\frac{d}{2s+d}}$ とすることで,

$$\|f^\circ - \hat{f}\|^2_{L^2(P_X)} \leq n^{-\frac{2s}{2s+d}} \log(n)^2.$$

$p = q = \infty$のとき, Schmidt-Hieber (2017) に帰着.

線形推定量との比較

- 線形推定法 (Donoho & Johnstone, 1994)

  $\hat{f}(x) = \sum_{i=1}^{n} y_i \varphi_i(x_1, \ldots, x_n; x) + b$ と書ける**任意の推定量**

  (**カーネルリッジ回帰**, Sieve法, Nadaraya-Watson推定量...)

  $\hat{f}(x) = K_{x,X}(K_{X,X} + \lambda \mathrm{I})^{-1} Y$ $\quad \hat{f} = \underset{f \in \mathcal{H}_k}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}_k}^2$

$$n^{-\frac{2s-2(1/p-1/2)_+}{2s+1-2(1/p-1/2)_+}}$$ (d=1の時)

- 深層学習

$$\bigvee$$

$$n^{-\frac{2s}{2s+1}}$$

$p < 2$で差が出る

pが小さい (p<2) と深層学習が優越
→ 空間的に滑らかさが非一様な時
(深層学習の適応性「基底の学習・非凸性」)

c.f., 不連続関数の例：Imaizumi&Fukumizu, 2018.

直感

$$\check{f}(x) = \sum_{j=1}^{N} \beta_j \varphi_j(x)$$

係数　基底

$n^{-\frac{2s-2(1/p-1/2)_+}{2s+1-2(1/p-1/2)_+}}$

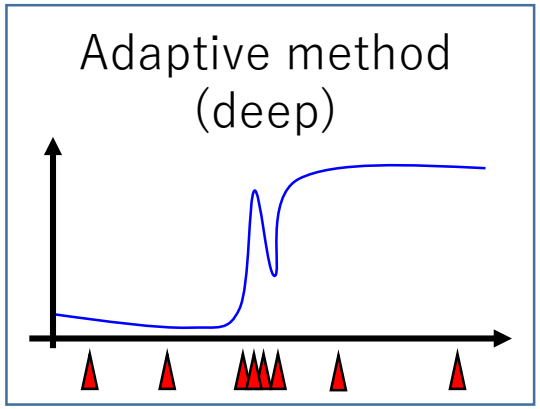
事前に設定: 非適応的手法
カーネルリッジ回帰, 最小二乗法, ...

推定する: 適応的手法
深層学習, スパース推定, Boosting, ...

$n^{-\frac{2s}{2s+1}}$

スパース推定との違い：

- スパース:
  あらかじめ用意した多数の基底の中から重要な基底を選択
- Deep:
  直接，基底を構築する



David Donoho: ガウス賞 (2018)
スパース推定，wavelet-shrinkage，圧縮センシング，...

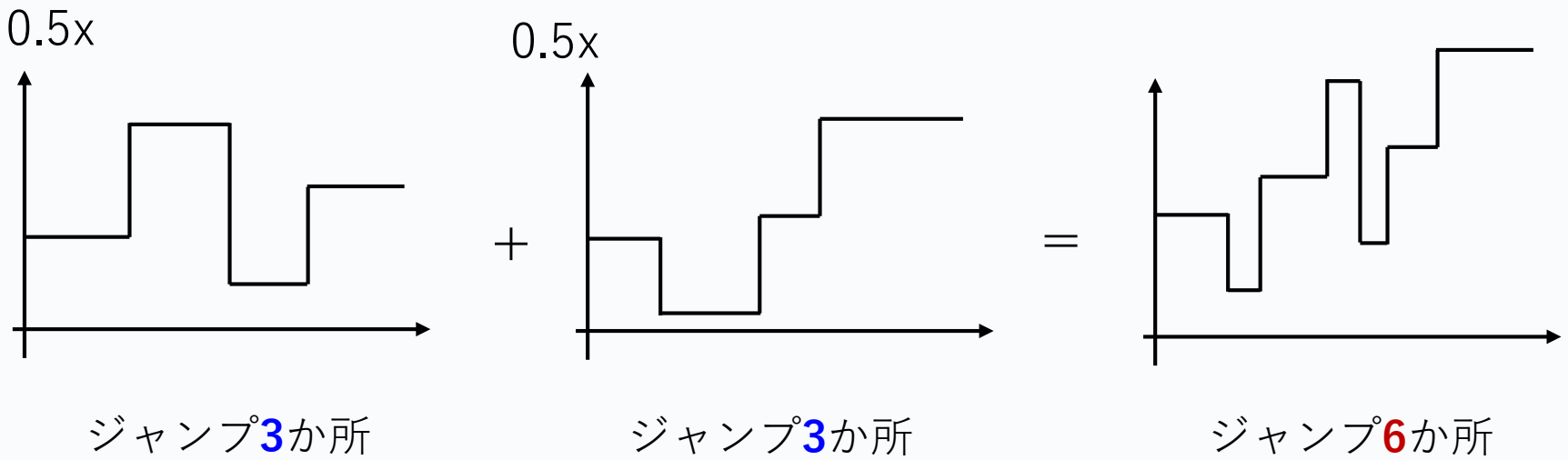
数学的に一般化

「滑らかさの非一様性」「不連続性」「データの低次元性」
凸結合を取って崩れる性質をもった関数の学習は深層学習が強い

→ 様々な性質を“凸性”で統一的に説明

例：ジャンプが３か所の区分定数関数

深層:$1/n$, カーネル: $1/\sqrt{n}$



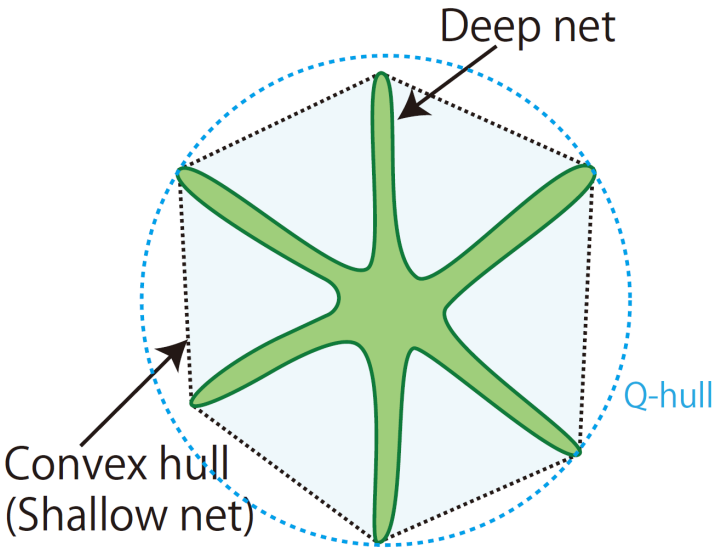
→ さらには「**スパース推定**」という観点からも説明できる.

[Satoshi Hayakawa and Taiji Suzuki: On the minimax optimality and superiority of deep neural network learning over sparse parameter spaces. arXiv:1905.09195.]

線形推定量の最悪誤差

線形推定量：$\hat{f}(x) = \sum_{i=1}^{n} y_i \varphi_i(x_1, \ldots, x_n; x) + b$ と書ける**任意の推定量**

例: カーネルリッジ回帰 $\hat{f}(x) = K_{x,X}(K_{X,X} + \lambda \mathrm{I})^{-1} Y$ （“浅い”学習法とみなす）



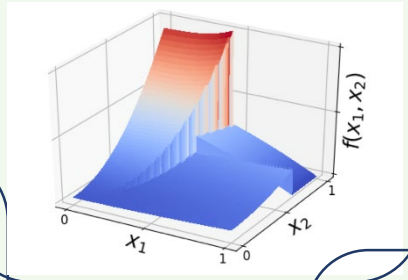
$$\inf_{\hat{f}:\text{Linear}} \sup_{f^o \in \mathcal{F}} \mathrm{E}[\|\hat{f} - f^o\|_{L_2(P)}^2] = \inf_{\hat{f}:\text{Linear}} \sup_{f^o \in \mathrm{conv}(\mathcal{F})} \mathrm{E}[\|\hat{f} - f^o\|_{L_2(P)}^2]$$

さらに条件を仮定すれば「Q-hull」まで拡張できる.

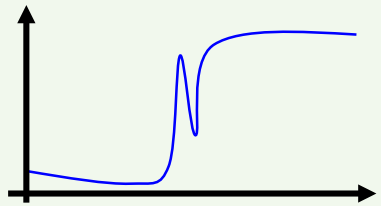
[Hayakawa&Suzuki: 2019][Donoho & Johnstone, 1994]

数学的一般化

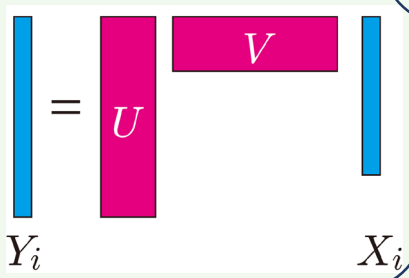
区分滑らかな関数



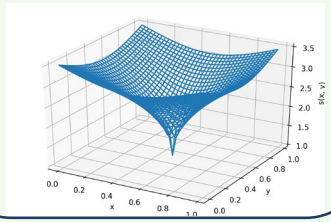
Besov空間



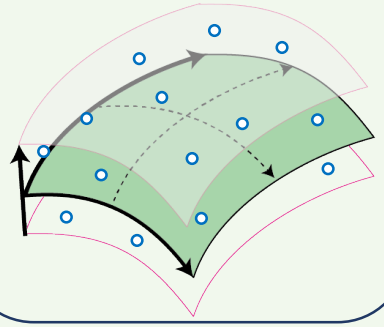
縮小ランク回帰



変動指数
Besov空間

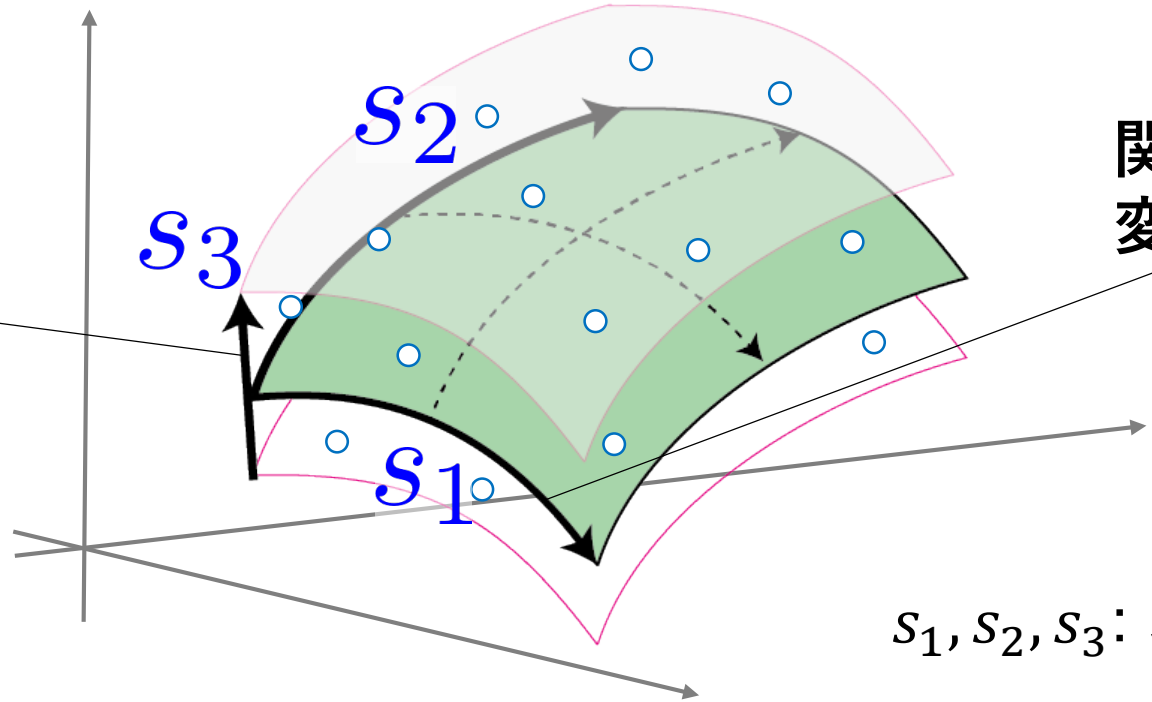


低次元データ



非凸性
スパース性

例 (1): 低次元データ構造



関数値
ほぼ一定

関数値が
変化する方向

s_1, s_2, s_3: smoothness

(non-smooth) $s_1, s_2 \ll s_3$ (smooth)

推定精度

深層学習

$$n^{-\frac{\tilde{s}}{\tilde{s}+1}}$$

$$\tilde{s} = \left(s_1^{-1} + s_2^{-1} + s_3^{-1}\right)^{-1}$$

浅い学習

$$n^{-\frac{s_1}{s_1+3}}$$

(次元の呪い)

例 (2): 縮小ランク回帰

- 縮小ランク回帰

$$Y_i = UVX_i + \xi_i \quad (i = 1, \ldots, n)$$

ただし, $Y_i \in \mathbb{R}^M, X_i \in \mathbb{R}^N$, $U \in \mathbb{R}^{M \times r}, V \in \mathbb{R}^{r \times N}$ $(r \ll M, N)$.

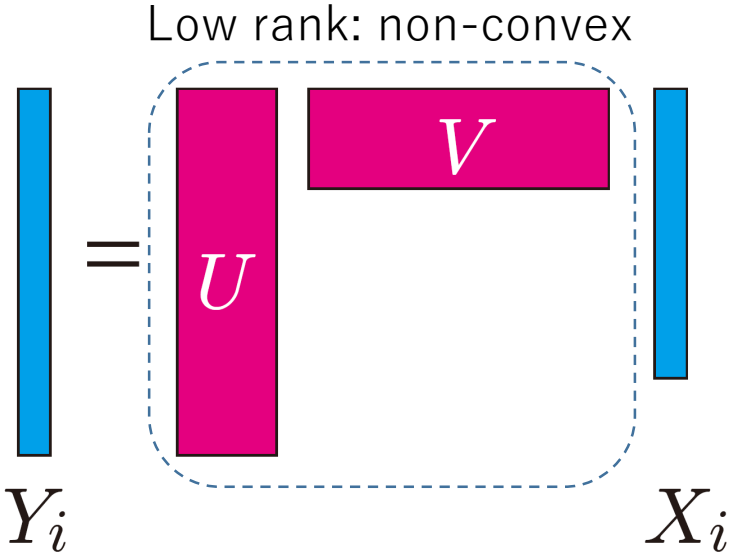
- Linear estimator $\hat{f}(x) = \sum_{i=1}^{n} Y_i \varphi(X_1, \ldots, X_n, x)$,
- Deep learning $\hat{f}(x) = \hat{U}\hat{V}x$.

推定精度の比較

$$\frac{r(M+N)}{n} \ll \frac{MN}{n}$$

Deep

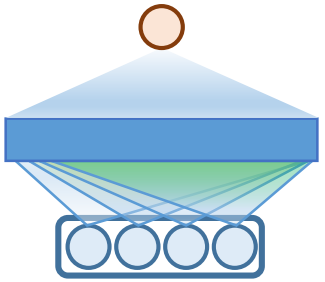
Shallow
(LSやRidge回帰等)



低ランク行列の凸包はフルランク行列

Barronクラスの汎化誤差

$$f(x) = \int_{\mathbb{R}^d} e^{iw^\top x} \tilde{f}(w) \mathrm{d}w \quad \text{(Fourier変換)}$$



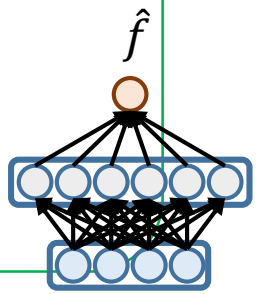
仮定： $\int_{\mathbb{R}^d} \|w\| |\tilde{f}(w)| \mathrm{d}w < \infty$

$(x_i, y_i)_{i=1}^n$: i.i.d. $\quad f(x) = \mathrm{E}[Y|X=x]$ $\quad$ (例：$y_i = f(x_i) + \epsilon_i$)

二層ニューラルネットワークの汎化誤差 (Barron 1991, 1993)

二層ニューラルネットワークのある種の正則化推定量$\hat{f}$が存在して次を満たす：

$$\mathrm{E}\left[\|\hat{f} - f\|^2_{L_2(P(X))}\right] \leq O\left(\sqrt{\frac{d\log(n)}{n}}\right)$$



活性化関数の条件： $\eta(-z) = 1 - \eta(z)$

$\|\eta'\|_\infty < \infty$

$\lim_{z \to -\infty} \sup \eta(z)/|z|^p < \infty$

(MDL, PAC-Bayes的解析)

線形推定量を用いると次元の呪い

Sobolevクラスの近似理論

$$\Pi_N := \left\{ \sum_{j=1}^{N} \alpha_j \eta(a_j^\top x + b_j) \mid \alpha_j \in \mathbb{R}, a_j \in \mathbb{R}^d, b_j \in \mathbb{R} \right\}$$

中間層の横幅がNの
二層ニューラルネットワーク

定理

ηがある開区間で無限回微分可能であり，その開区間のある点bにおいて

$$\frac{\partial^k \eta}{\partial x^k}(b) \neq 0 \ (\forall k \in \mathbb{Z}, k \geq 0)$$

とする．すると，$\forall f \in W_p^s([0,1]^d)$に対してある$g \in \Pi_N$が存在して，

$$\|f - g\|_p \lesssim N^{-\frac{s}{d}} \|f\|_{W_p^s}$$

（ノード数Nの中間層を用いた近似誤差）

[Mhaskar: Neural networks for optimal approximation of smooth and analytic functions. Neural Computation, 8(1):164–177, 1996]

- この近似誤差はN個の基底を用いた近似法の中で最適なオーダーを達成.
- シグモイド関数は条件を満たす．ReLUは満たさない.
- 滑らかな関数はより近似しやすい.

第2章
深層学習の汎化誤差

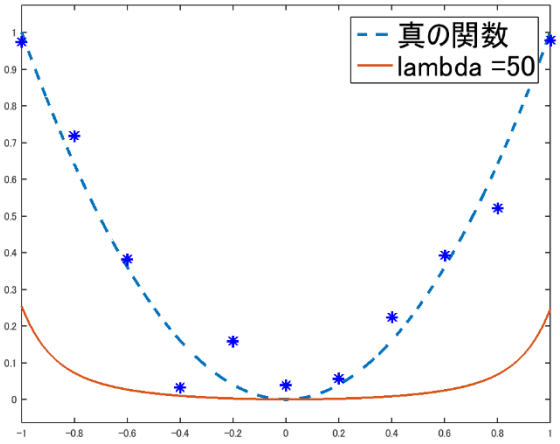
- これまでの議論は，実は問題に合わせて「適切なサイズのネットワーク」を用いた場合の議論であった．

- 実際は，かなりサイズの大きなネットワークを用いる．

過学習

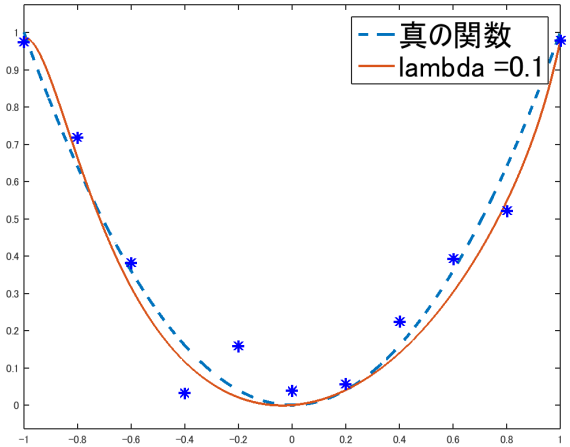
「なんでも表現できる方法」が最適とは限らない
少しのノイズにも鋭敏に反応してしまう

悪い学習結果



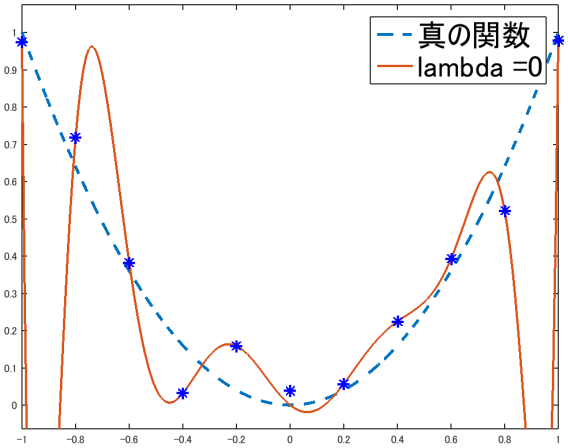
過小学習
説明力が低すぎる

良い学習結果



適切な学習
説明力が適切

悪い学習結果



過学習
説明力が高すぎる
（複雑すぎる）

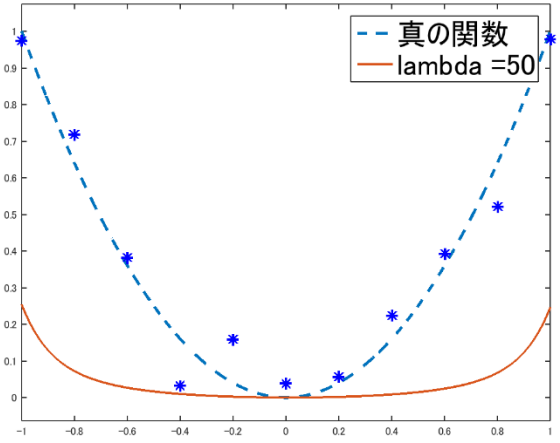
一見当てはまりが良いので危険

学習に用いるデータには誤りも含まれる

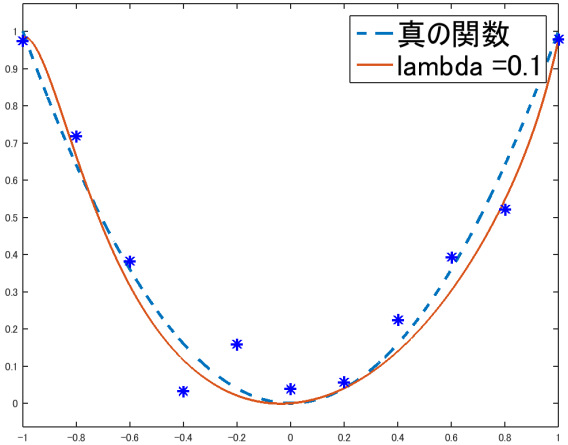
「過去問は解けるけれども本番の試験は解けない」という状況を回避したい

従来の学習理論

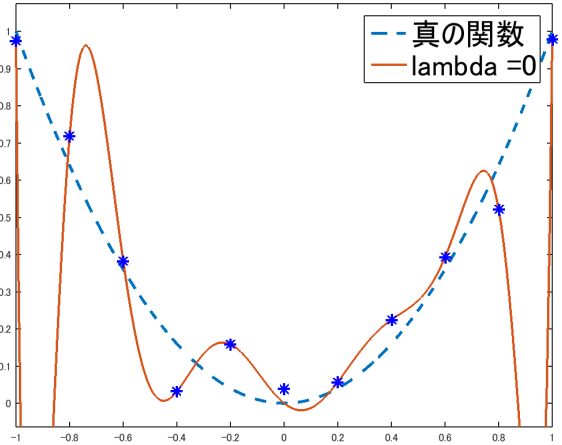
過小学習

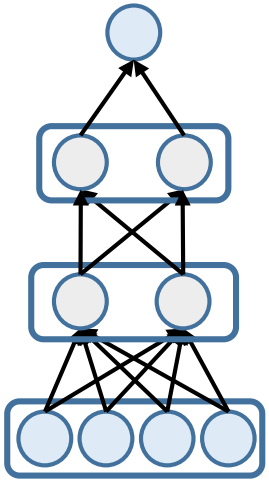


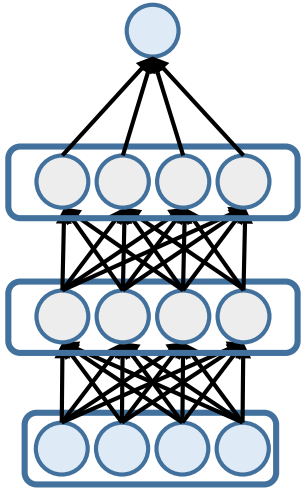
適切な学習

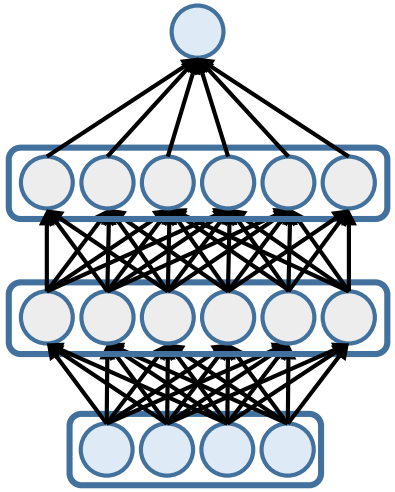


過学習







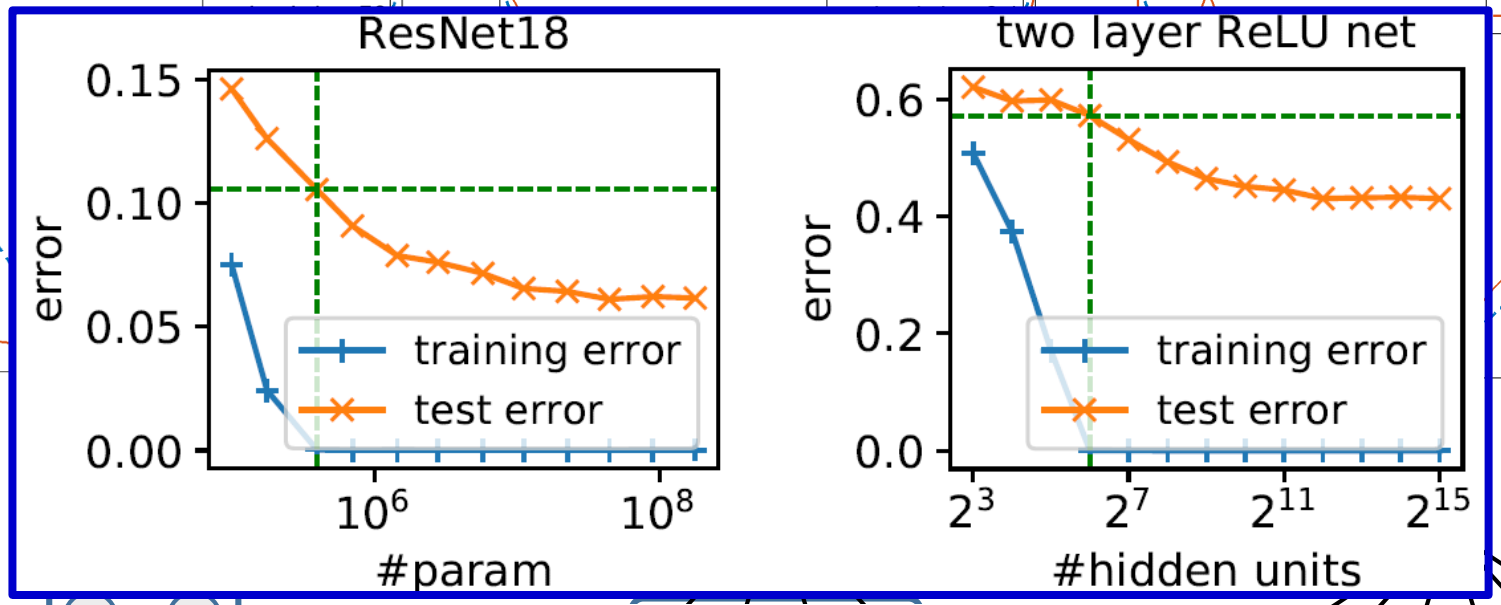


従来の学習理論

過小学習

実際は...

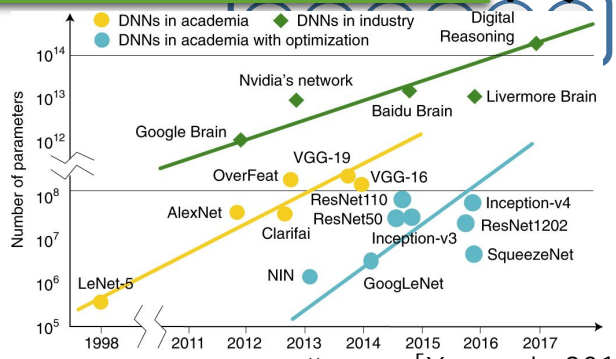
過学習



[Neyshabur et al., ICLR2019]

ネットワークのサイズを大きくしても過学習しない

データサイズ：120万
モデルパラメータサイズ：10億



[Xu et al., 2018]

深層ニューラルネットの冗長性

「Overparameterization」
パラメータサイズがデータサイズを超えている状況
での汎化性能を説明したい.

パラメータ数 ≫ データサイズ ≫ 実質的自由度

数十億　　数百万　　数十万

[仮説] 見かけの大きさ (パラメータ数) よりも
実質的な大きさ (自由度) はかなり小さいはず.

“実質的自由度”を調べる研究：
- ノルム型バウンド
- 圧縮型バウンド

「実質的自由度」として何が適切かを見つけることが理論上問題になる.

汎化誤差バウンド

深層学習の汎化誤差

$\psi(y_i, f(x_i))$：損失関数 (1-Lipschitz continuous w.r.t. f)

$$\hat{\Psi}(f) := \frac{1}{n}\sum_{i=1}^{n}\psi(y_i, f(x_i)) \qquad \Psi(f) := \mathbb{E}[\psi(Y, f(X))]$$

経験誤差（訓練誤差） **期待誤差（汎化誤差）**

- 汎化ギャップ

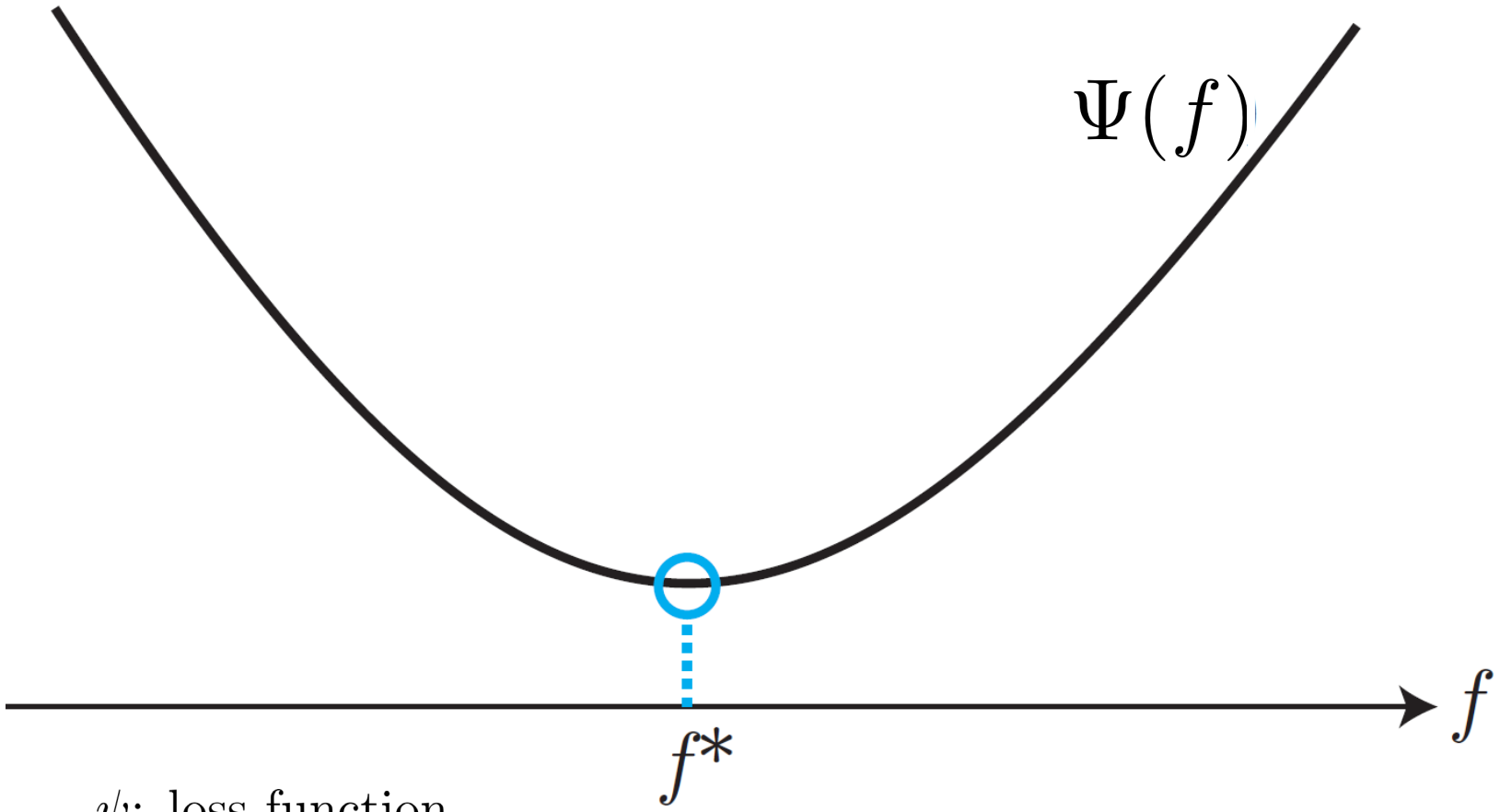
  任意の$\hat{f} \in \mathcal{F}$に対して成り立つ一様バウンドが欲しい:

$$\Psi(\hat{f}) \leq \hat{\Psi}(\hat{f}) + \underbrace{\bar{R}_n(\psi \circ \mathcal{F})}_{\text{Complexity}}$$

E.g., Rademacher 複雑度: $P(\epsilon_i = -1) = P(\epsilon_i = 1)$ (i.i.d.)

$$\bar{R}_n(\psi \circ \mathcal{F}) = \mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\epsilon_i\psi(y_i, f(x_i))\right]$$

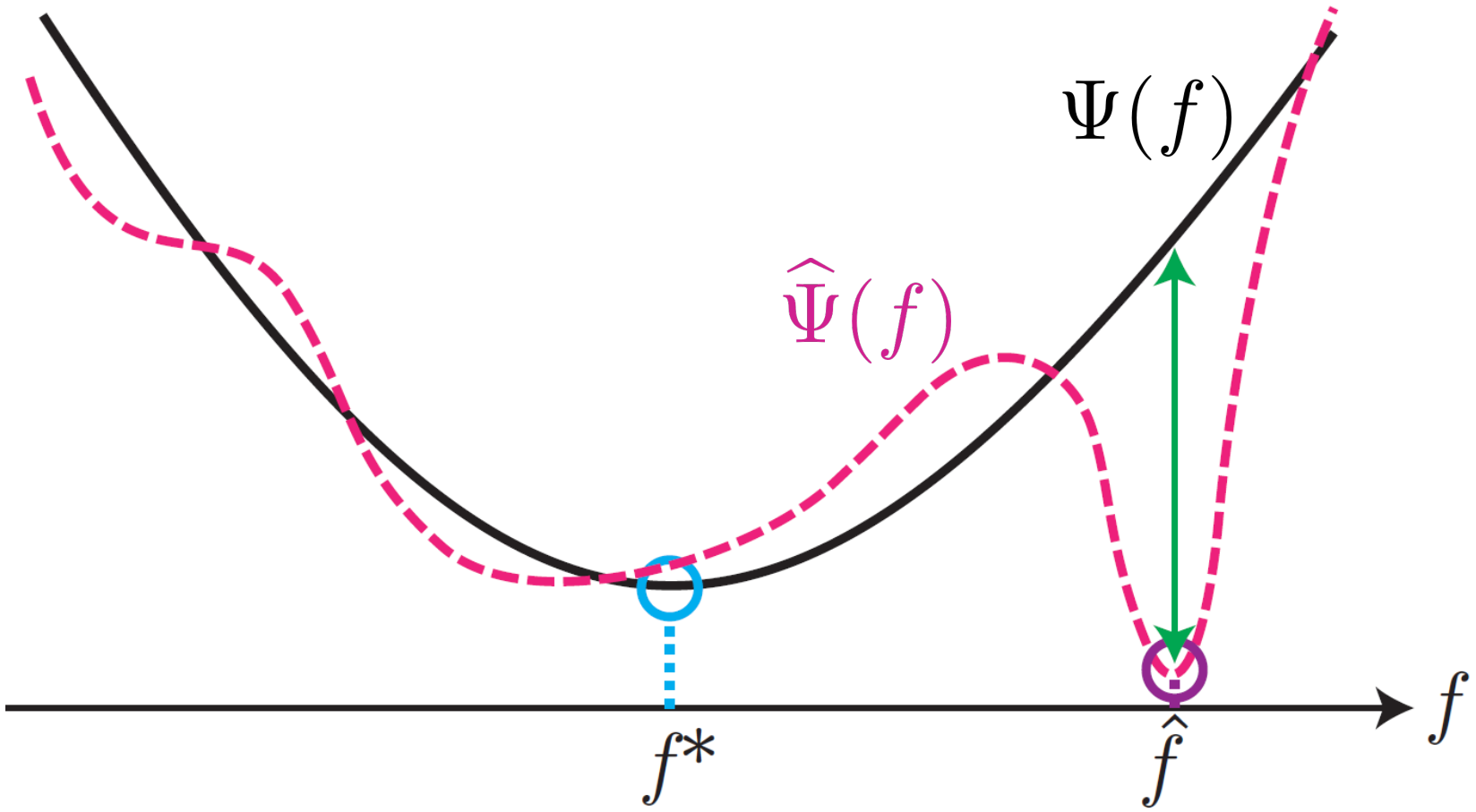
Uniform bound



ψ: loss function

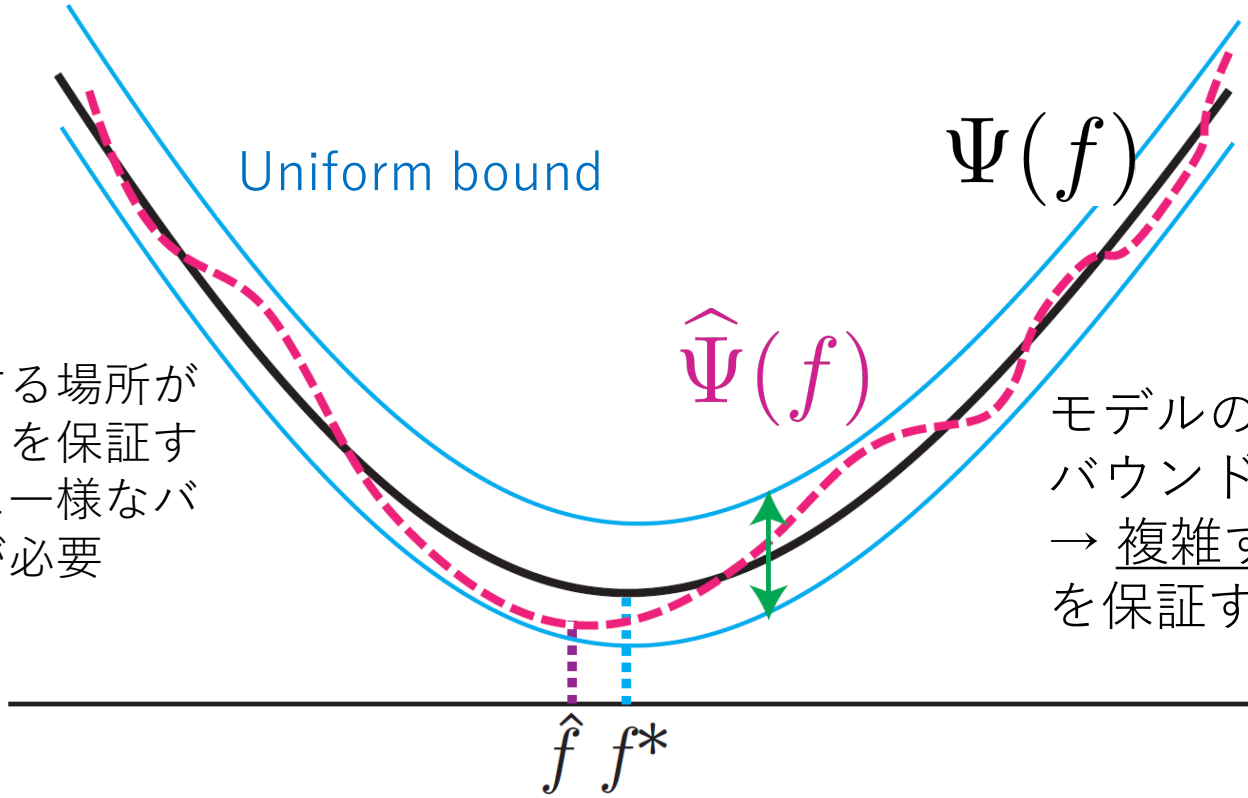
$$\widehat{\Psi}(f) := \frac{1}{n}\sum_{i=1}^{n}\psi(y_i, f(x_i)) \qquad \Psi(f) := \mathbb{E}[\psi(Y, f(X))]$$

Uniform bound



“運良くデータに強く当てはまる”場所があるかもしれない.
→ 過学習

Uniform bound



過学習する場所がないことを保証するために一様なバウンドが必要

モデルの「複雑さ」がバウンドに影響する
→ 複雑すぎないことを保証する必要がある.

Uniform bound

$$\Psi(\hat{f}) - \widehat{\Psi}(\hat{f}) \leq \sup_{f \in \mathcal{F}} \left\{ \Psi(f) - \widehat{\Psi}(f) \right\}$$

Rademacher complexity

$$\bar{R}_n = \mathbb{E}\left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \psi(y_i, f(x_i)) \right]$$

Dudley integral (covering num.)

$$\frac{1}{\sqrt{n}} \int_{1/n}^{\infty} \sqrt{\log(N(\epsilon, \|\cdot\|_\infty, \mathcal{F}))} \mathrm{d}\epsilon$$

深層学習の汎化誤差バウンド（抜粋）

ノルム型バウンド

Author	Rate	Bound type
Neyshabur et al. (2015)	$\frac{2^L \prod_{\ell=1}^L R_{\ell,F}}{\sqrt{n}}$	Norm base
Bartlett et al. (2017)	$\frac{\prod_{\ell=1}^L R_{\ell,2}}{\sqrt{n}} \left(\frac{R_{\ell,2\to 1}^{2/3}}{R_{\ell,2}^{2/3}} \right)^{3/2}$	Norm base
Neyshabur et al. (2017)	$\frac{\prod_{\ell=1}^L R_{\ell,2}}{\sqrt{n}} \sqrt{L^2 W \sum_{\ell=1}^L \frac{R_{\ell,F}^2}{R_{\ell,2}^2}}$	Norm base
Golowich et al. (2018)	$\prod_{\ell=1}^L R_{\ell,F} \min\left\{ \frac{1}{n^{1/4}}, \sqrt{\frac{L}{n}} \right\}$	Norm base
Li et al. (2018) Harvey et al. (2017)	$\frac{\prod_{\ell=1}^L R_{\ell,2} \sqrt{L^2 W^2}}{\sqrt{n}}$	VC-dim Naïve bound
Arora et al. (2018)	$\sqrt{\frac{L^2 \max_{1\le i\le n} \lvert \widehat{f}(x_i) \rvert^2 \sum_{\ell=1}^L \frac{1}{\mu_\ell^2 \mu_{\ell\to}^2}}{n}}$	Compression
Baykal et al. (2018)	$\sqrt{\frac{L^2 \max_{1\le i\le n} \lvert \widehat{f}(x_i) \rvert^2 \sum_{\ell=2}^L (\hat{\Delta}^{\ell\to})^2 \sum_{i=1}^W S_i^\ell}{n}}$	Compression
Suzuki et al. (2018)	$\sum_{\ell=2}^L \sqrt{\lambda_\ell} + \sqrt{\frac{\sum_{\ell=1}^L m_{\ell+1}^\sharp m_\ell^\sharp}{n}}$	Compression

圧縮型バウンド

L: depth
$W = \max_\ell m_\ell$: width

R_F: Frobenius norm, R_2: operator norm

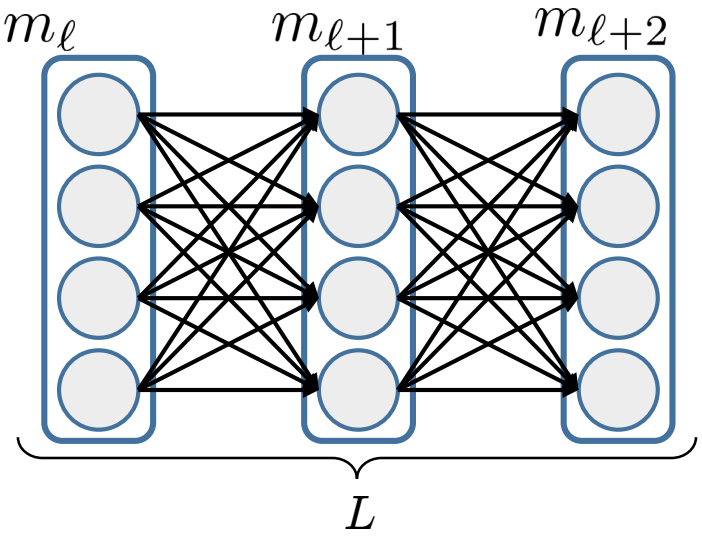
Naïve bound (VC-次元)

$$\Psi(\hat{f}) \leq \hat{\Psi}(\hat{f}) + \boxed{?}$$

VC-次元 [Harvey et al.2017]

$$\sqrt{L \frac{\sum_{\ell=1}^{L} m_\ell m_{\ell+1}}{n} \log(n)}$$

$$\left(\sqrt{L \frac{\#\text{ of parameters}}{n}} \right)$$



- m_ℓ: width of ℓ-th layer
- L: depth

☹ パラメータ数 $\sum_{\ell=1}^{L} m_\ell m_{\ell+1}$ がそのままバウンドに現れてしまう.

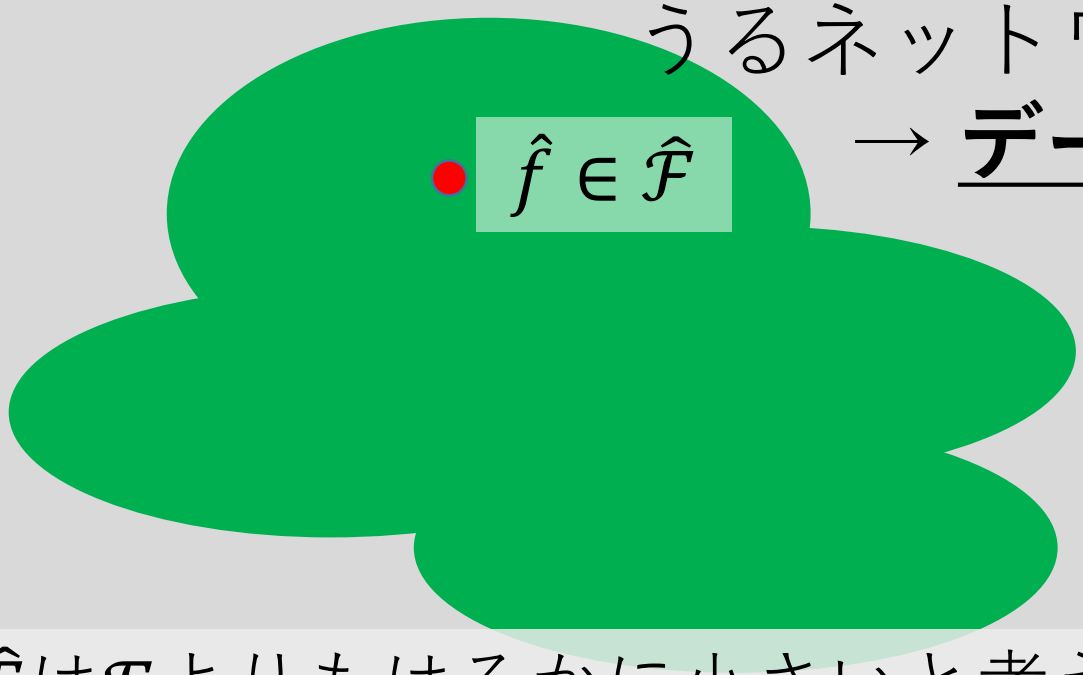
☹ パラメータ数≫データサイズの状況を説明できていない.

どうやって改善するか？

$\mathcal{F}$: ネットワーク全体

$\hat{\mathcal{F}}$: 学習済みモデルが入りうるネットワークの集合

→ **データ依存**



※$\hat{\mathcal{F}}$は$\mathcal{F}$よりもはるかに小さいと考えられる.

“典型的な学習済みネットワーク”の集合を解析する.

ノルム型バウンド

NNモデル: $f(x) = (W^{(L)}\eta(\cdot)) \circ (W^{(L-1)}\eta(\cdot)) \circ \cdots \circ (W^{(1)}x)$

- Bartlett et al. (2017): 正規化マージンバウンド

$$\frac{1}{\sqrt{n}} \prod_{\ell=1}^{L} R_{\ell,2} \left(\sum_{\ell=1}^{L} \frac{R_{\ell,2\to1}^{2/3}}{R_{\ell,2}^{2/3}} \right)^{3/2}$$

$$R_{\ell,2} := \|W^{(\ell)}\| = \sigma_1(W^{(\ell)}) \quad (\text{最大特異値}) \qquad R_{\ell,2\to1} := \sum_j \|W_{j,:}^{(\ell)}\|$$

- Golowich et al. (2018)

$$\prod_{\ell=1}^{L} R_{\ell,F} \min\left\{ \frac{1}{n^{1/4}}, \sqrt{\frac{L}{n}} \right\}$$

$R_{\ell,F} := \|W^{(\ell)}\|_{\mathrm{F}}$: Frobeniusノルム ($R_{\ell,2}$より大きい)

☺ 横幅に依存しない. → Overparametrizationの状況を説明！

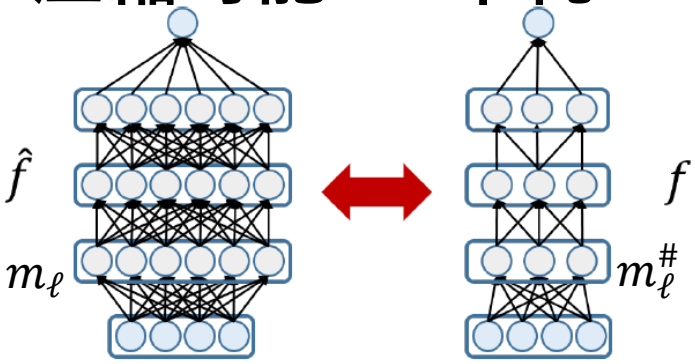
☹ 縦幅に指数的に依存する場合がある. (バウンドによる)

圧縮型バウンド

圧縮可能 ⇔ 単純

ネットワークが圧縮できるなら汎化する.

学習済みネット
(元ネットワーク)



圧縮したネットワーク

(プルーニング等により圧縮)

典型的な圧縮型バウンド:

$$\Psi(f^\sharp) \leq \widehat{\Psi}(\widehat{f}) + \hat{r} + \sqrt{\frac{\sum_{\ell=1}^{L} m_{\ell+1}^\sharp m_\ell^\sharp}{n}\log(n)}$$

圧縮したネットワークの横幅

Bias

Variance

圧縮したネットワーク

元のネットワーク

Bias-varianceトレードオフ

[Arora et al., 2018; Zhou et al., 2019; Baykal et al., 2019; Suzuki et al., 2018]

注：これらのバウンドは$\hat{f}$の汎化誤差は与えていない.

“圧縮していない”ネットワーク$\hat{f}$の汎化誤差も与えられる (次ページ).

非圧縮ネットの圧縮型バウンド

仮定: $\hat{f}$ がより小さなネットワーク$f^{\sharp}$に圧縮できるとする.
($\hat{f} \in \hat{\mathcal{F}}, f^{\sharp} \in \mathcal{F}^{\sharp}$; $\hat{\mathcal{F}}$は学習済みネットの集合, $\mathcal{F}^{\sharp}$ は圧縮したネットの集合)

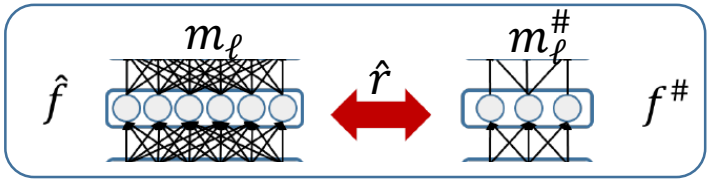
[Suzuki,Abe,Nishimura:ICLR2020]

非圧縮ネットの圧縮型バウンド:

$$\Psi(\widehat{f}) \leq \widehat{\Psi}(\widehat{f}) + \frac{1}{\sqrt{n}}\hat{r} + \sqrt{\frac{\sum_{\ell=1}^{L} m_{\ell+1}^{\sharp} m_{\ell}^{\sharp}}{n}\log(n)}$$

$\|\widehat{f} - f^{\sharp}\|_n^2 \leq \hat{r}^2 \quad (\text{a.s.})$:圧縮可能性は訓練データのみから判断.

(学習結果が圧縮可能なように学習するのが好ましい)



$$\Psi(f^{\sharp}) \lesssim \hat{\Psi}(\widehat{f}) + \hat{r} + \sqrt{\frac{\sum_{\ell=1}^{L} m_{\ell}^{\sharp} m_{\ell+1}^{\sharp}}{n}\log(n)}$$

(既存のバウンド)

非圧縮ネットの圧縮型バウンド

仮定: $\hat{f}$ がより小さなネットワーク$f^{\sharp}$に圧縮できるとする.
($\hat{f} \in \mathcal{F}, f^{\sharp} \in \mathcal{F}^{\sharp}$; $\mathcal{F}$は学習済みネットの集合, $\mathcal{F}^{\sharp}$ は圧縮したネットの集合)

[Suzuki,Abe,Nishimura:ICLR2020]

非圧縮ネットの圧縮型バウンド:

$$\Psi(\hat{f}) \leq \hat{\Psi}(\hat{f}) + \frac{1}{\sqrt{n}}\hat{r} + \sqrt{\frac{\sum_{\ell=1}^{L} m_{\ell+1}^{\sharp} m_{\ell}^{\sharp}}{n}\log(n)}$$

バリアンスを小さくできる

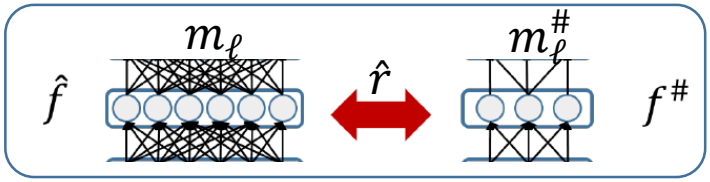
$$\|\hat{f} - f^{\sharp}\|_n^2 \leq \hat{r}^2 \quad \text{(a.s.)}$$

:圧縮可能性は訓練データのみから判断.

(学習結果が圧縮可能なように学習するのが好ましい)

元のネットワークを評価

改善される



$$\Psi(f^{\sharp}) \lesssim \hat{\Psi}(\hat{f}) + \hat{r} + \sqrt{\frac{\sum_{\ell=1}^{L} m_{\ell}^{\sharp} m_{\ell+1}^{\sharp}}{n}\log(n)}$$

(既存のバウンド)

より正確なステートメント

仮定: $\hat{f}$ がより小さなネットワーク$f^{\sharp}$に圧縮できるとする.
($\hat{f} \in \hat{\mathcal{F}}, f^{\sharp} \in \mathcal{F}^{\sharp}$; $\hat{\mathcal{F}}$は学習済みネットの集合, $\mathcal{F}^{\sharp}$ は圧縮したネットの集合)

- $\|\widehat{f} - f^{\sharp}\|_n^2 \leq \hat{r}^2 \quad (\text{a.s.})$
- $\dot{R}_r(\mathcal{G}') := \bar{R}_n(\{f \in \mathcal{G}' \mid \|f\|_{L_2(P_X)} \leq r\})$: 局所Rademacher複雑度
- $r_* := \inf\{r > 1/n \mid \dot{R}_r(\psi(\widehat{\mathcal{F}}) - \psi(\mathcal{F}^{\sharp})) \leq r^2\}$: 局所Rad.の不動点

$$\psi(\widehat{\mathcal{F}}) - \psi(\mathcal{F}^{\sharp}) := \{\psi(\widehat{f}) - \psi(f^{\sharp}) \mid \widehat{f} \in \widehat{\mathcal{F}}, f^{\sharp}\mathcal{F}^{\sharp}\}$$

Theorem (非圧縮ネットの圧縮型バウンド)

$$\Psi(\widehat{f}) \leq \hat{\Psi}(\widehat{f}) + \underbrace{\sqrt{\frac{t}{n}(\hat{r}^2 + r_*^2)} + \dot{R}_{\hat{r}+r_*}(\widehat{\mathcal{F}} - \mathcal{F}^{\sharp})}_{\textbf{Fast part (O(1/n))}\ \text{bias}} + \underbrace{\bar{R}_n(\mathcal{F}^{\sharp}) + \sqrt{\frac{2t}{n}}}_{\textbf{Main part (O(1/}\sqrt{n}\textbf{))}\ \text{variance}}$$

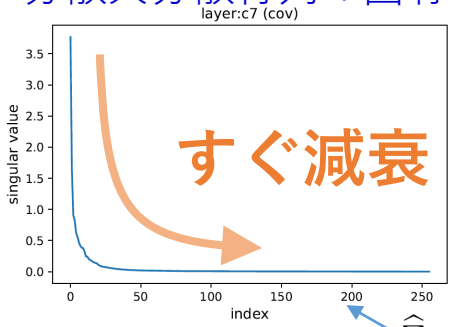
with probability at least $1 - e^{-t}$.

いつどれくらい圧縮できるか？

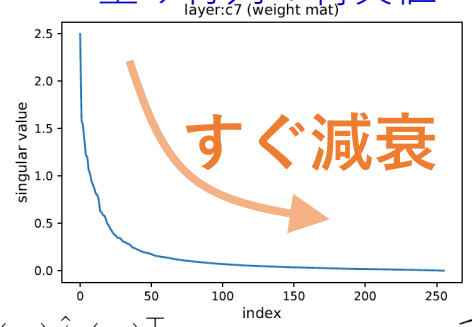
- 中間層の**分散共分散行列の固有値分布**
- 中間層の**重み行列の特異値分布**

が速く減衰するなら圧縮しやすい.

分散共分散行列の固有値

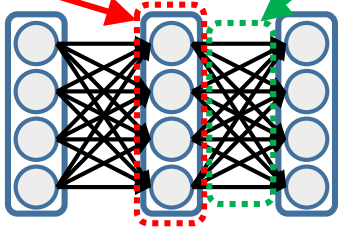


重み行列の特異値



$$\hat{\Sigma}_\ell = \frac{1}{n}\hat{\phi}_\ell(x_i)\hat{\phi}_\ell(x_i)^\top$$

$\hat{\phi}_\ell(x)$ $\widehat{W}(\ell)$



分散共分散行列も重み行列も
特異値が速く減衰
→**小さい統計的自由度**
(AIC, Mallows' Cp)

$$\Psi(\widehat{f}) \le \hat{\Psi}(\widehat{f}) + O\left(\left(L\frac{\sum_{\ell=1}^{L} m_\ell}{n}\log(n)\right)^{\frac{2\alpha}{1+2\alpha}} + \sqrt{L^{1+\delta}\left(\sum\right.}\right.$$

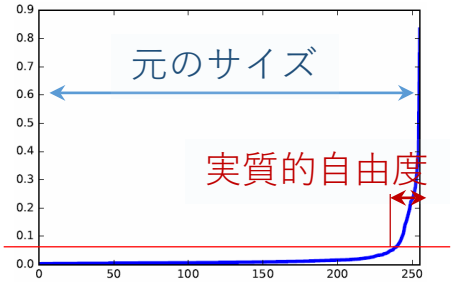
カーネル法の理論
(そもそもカーネルは無限次元モデル)

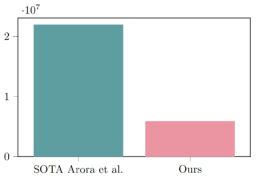
- 「**テンソル分解**」の援用によりCNNの詳細な評価も実現.

[Suzuki: Fast generalization error bound of deep learning from a kernel perspective. AISTATS2018]
[Li, Sun, Liu, Suzuki and Huang: Understanding of Generalization in Deep Learning via Tensor Methods. AISTATS2020]
[Suzuki, Abe, Nishimura: Compression based bound for non-compressed network: unified generalization error analysis of large compressible deep neural network, ICLR2020]
[Suzuki et al.: Spectral pruning: Compressing deep neural networks via spectral analysis and its generalization error. IJCAI-PRICAI 2020]

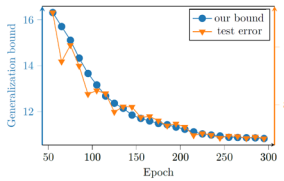
[実験的観察] 実際に学習したネットワークは圧縮しやすい.

	元サイズ	圧縮可能サイズ
Layer	Original	Our bound
1	1,728	1,013
4	147,456	84,499
6	589,824	270,216
9	1,179,648	50,768
12	2,359,296	4,583
15	2,359,296	3,886
	大	小





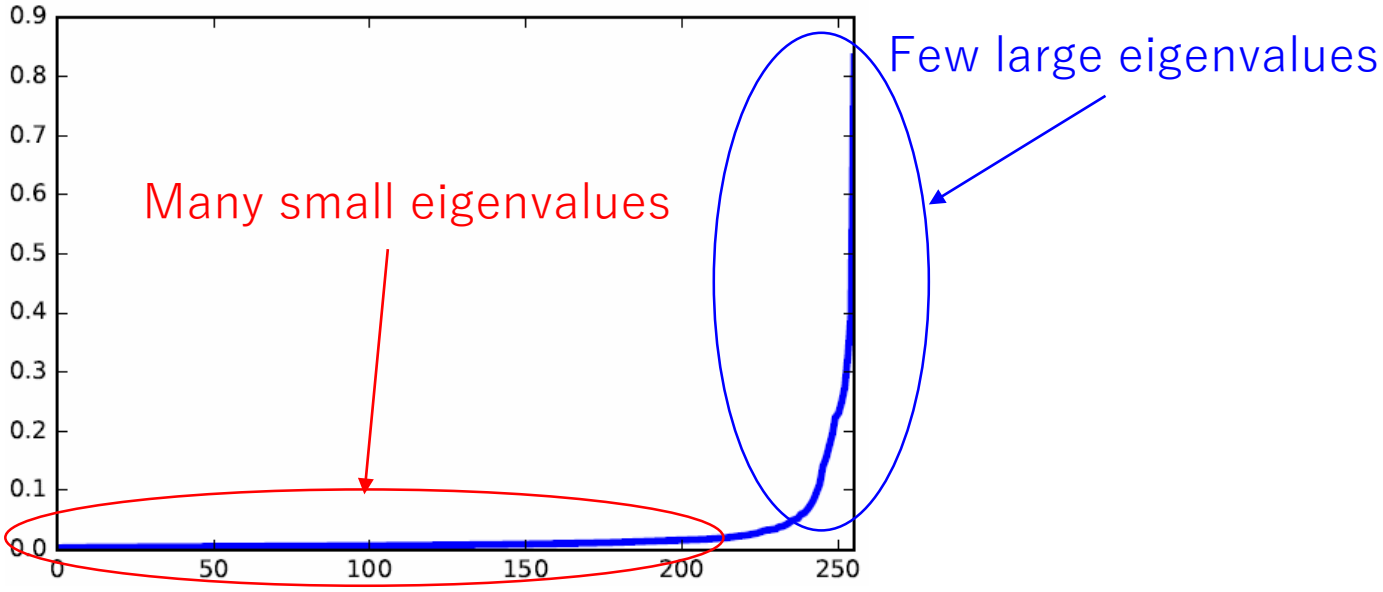
(a) Bound comparison



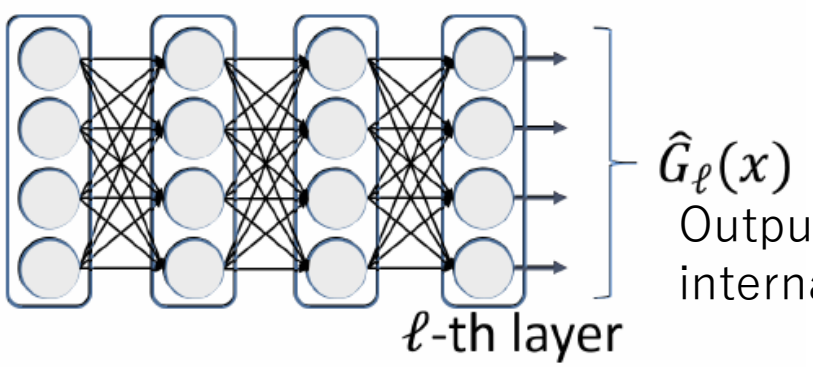
(b) Generalization bound

Near low rank covariance

Distribution of eigenvalues of the covariance matrix in an internal layer



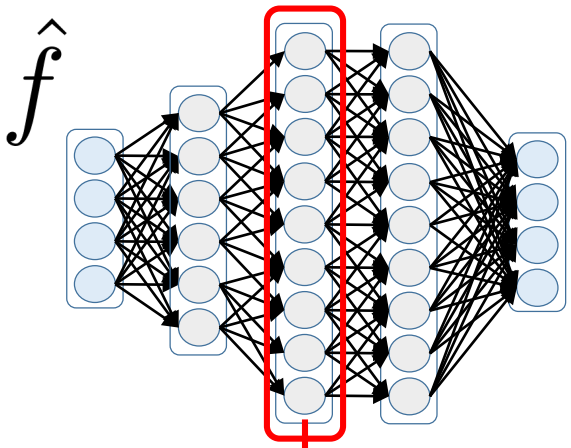
9-th layer in VGG-13 trained on CIFAR-10



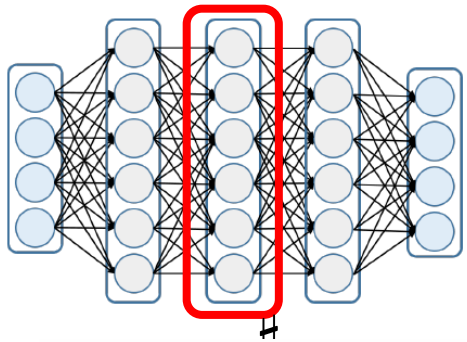
$$\Sigma_{i,j} = \mathrm{E}[\hat{G}_{\ell,i}(X)\hat{G}_{\ell,j}(X)]$$

$$\Sigma = USU^\top$$

Compression error

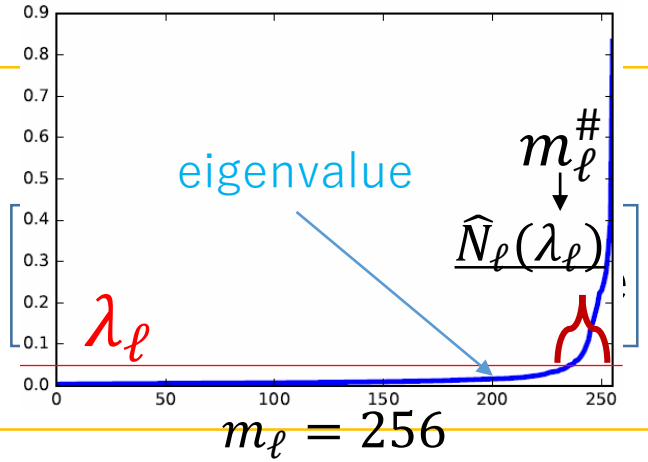


compression



Degree of freedom

$$\hat{N}_\ell(\lambda) = \sum_{j=1}^{m_\ell} \frac{\hat{\mu}_j^{(\ell)}}{\hat{\mu}_j^{(\ell)} + \lambda}$$



If the eigenvalue drops rapidly, then the network can be compressed into much smaller one.

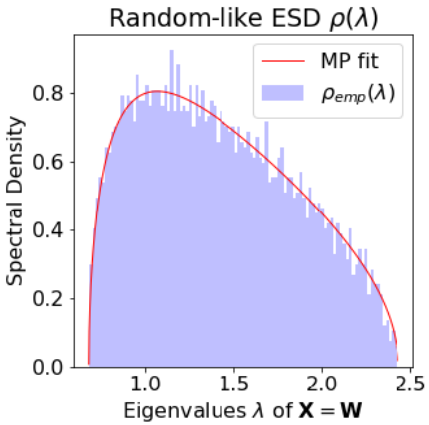
Then $\exists f^\#$ s.t. $\|f^\sharp - \hat{f}\|_n^2 \leq C \left(\sum_{\ell=2}^{L} \sqrt{\lambda_\ell} \right)^2$

$\|f\|_n^2 = \frac{1}{n} \sum_{i=1}^{n} f(x_i)^2$

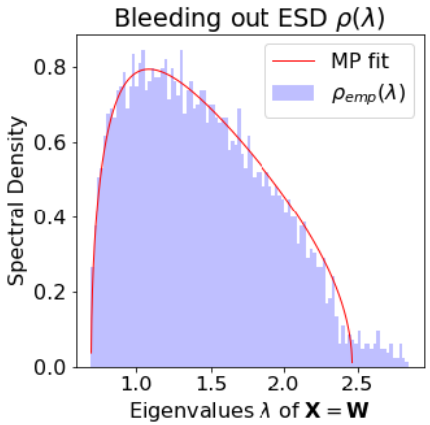
Singular value distribution

Singular value distribution of trained weight matrix

Random initialization

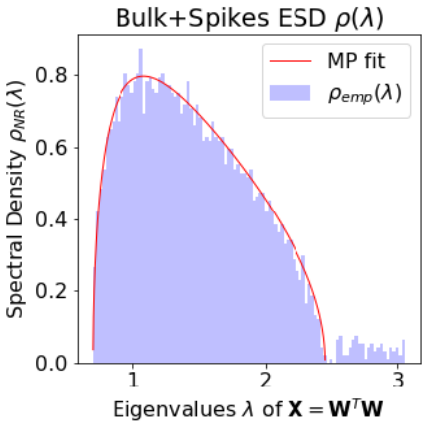


(a) RANDOM-LIKE.



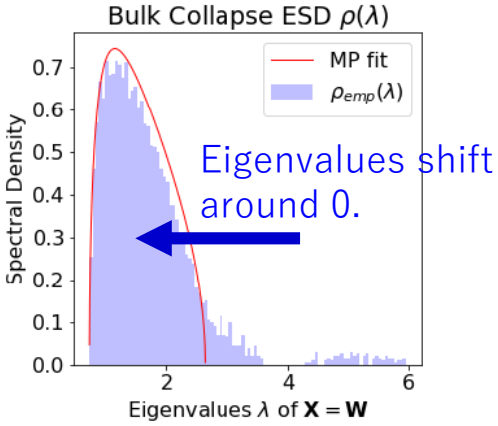
(b) BLEEDING-OUT.

After some epoch training



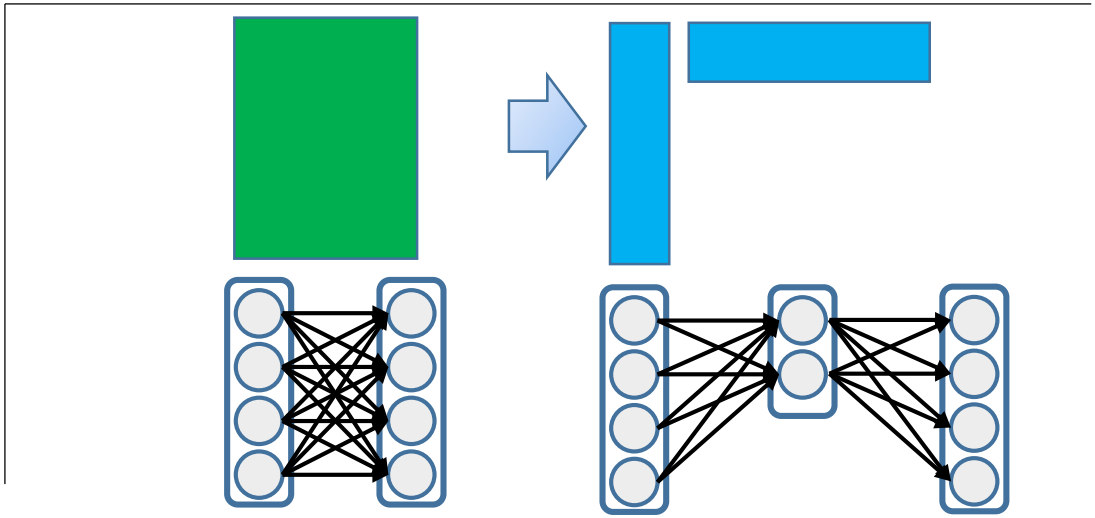
(c) BULK+SPIKES.

After several epoch training



(d) BULK-DECAY.

Over regularized



[Martin&Mahoney: Traditional and Heavy-Tailed Self Regularization in Neural Network Models. arXiv:1901.08276]

Near low rank weight and covariance

$$f(x) = (W^{(L)}\eta(\cdot)) \circ (W^{(L-1)}\eta(\cdot)) \circ \cdots \circ (W^{(1)}x)$$

- Near low rank weight matrix:
  - $\sigma_j(\widehat{W}^{(\ell)}) \lesssim j^{-\alpha}$
  - $\sigma_j(\widehat{\Sigma}^{(\ell)}) \lesssim j^{-\beta}$

Both of weight and covariance are near low rank

($\sigma_j(\cdot)$: j-th largest eigenvalue)

+ Other boundedness condition.

Theorem

$$\Psi(\widehat{f}) \leq \hat{\Psi}(\widehat{f}) + O\left(\left(L\frac{\sum_{\ell=1}^{L} m_\ell}{n}\log(n)\right)^{\frac{2\alpha}{1+2\alpha}} + \sqrt{L^{1+\delta}\frac{\left(\sum_{\ell=1}^{L} m_\ell\right)^{\frac{4/\beta}{4/\beta+2(1-1/2\alpha)}}}{n}\log(n)^3}\right)$$

where $\delta = \dfrac{\beta}{\frac{4\alpha}{(2\alpha-1)}+\beta}$.

Much smaller than the VC-bound:

$$\Psi(\widehat{f}) \leq \hat{\Psi}(\widehat{f}) + \sqrt{L\frac{\sum_{\ell=1}^{L} m_\ell m_{\ell+1}}{n}\log(n)}$$

Comparison with existing work

Comparison of intrinsic dimensionality between our degree of freedom and that in Arora et al. (2018). They are computed on VGG-19 network trained on CIFAR-10.

Layer	Original	Arora et al. (2018)	Our bound
1	1,728	1,645	567
4	147,456	644,654	75,240
6	589,824	3,457,882	394,200
9	1,179,648	36,920	10,395
12	2,359,296	22,735	756
15	2,359,296	26,584	882
		larger	smaller

Implicit number of parameters in each internal layer:

$$\hat{N}_\ell(\lambda_\ell)\hat{N}_{\ell+1}(\lambda_{\ell+1})k^2,$$

where k is the size of filter. We used $\lambda_\ell = 10^{-2}\mathrm{Tr}[\widehat{\Sigma}_{(\ell)}]$ (sufficiently small).

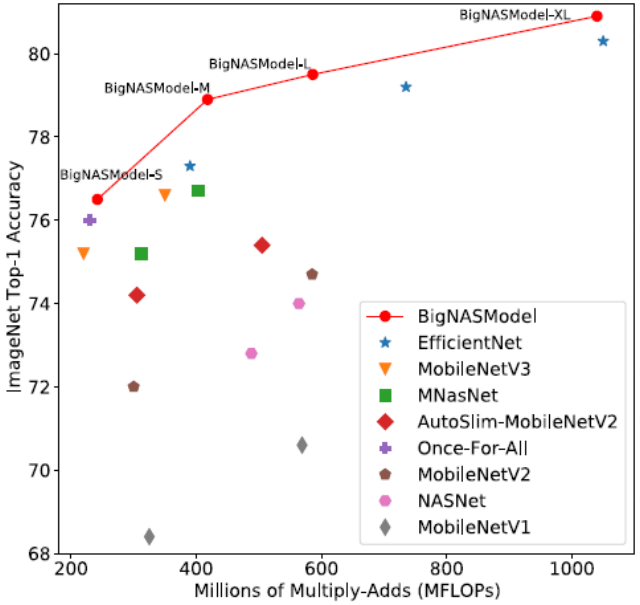
[S. Arora, R. Ge, B. Neyshabur, and Y. Zhang. Stronger generalization bounds for deep nets via a compression approach. ICML2018.]

BigNAS

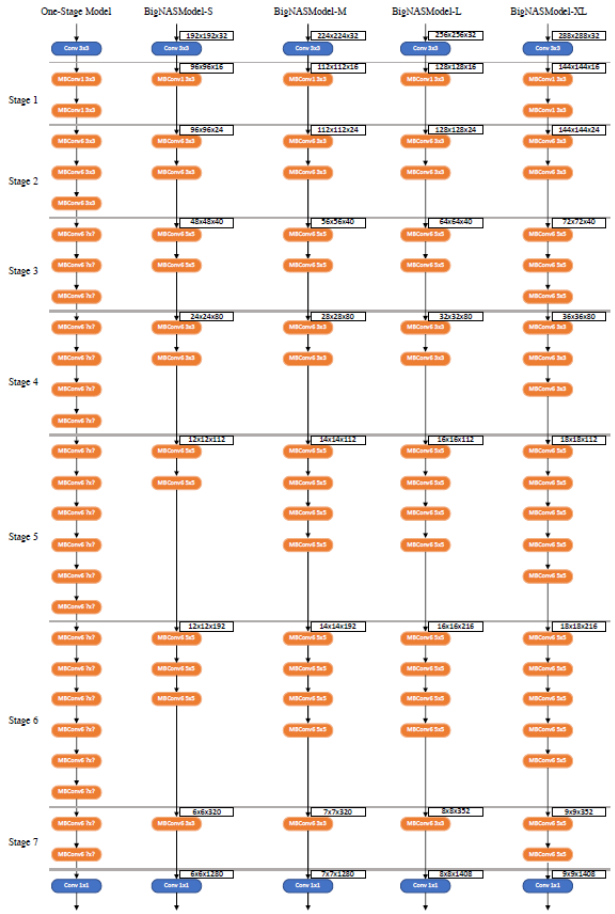
[Yu et al.: BigNAS: Scaling Up Neural Architecture Search with Big Single-Stage Models. ECCV2020]

(理論と関係あるNAS手法)

- 学習後のネットワークが圧縮できるように学習
- 大きなネットワークから小さなネットワークを生成できる
- EfficientNetを上回る効率性を実現



Group	Model Family	Params	FLOPs	Top-1
200M FLOPs	MobileNetV1 $_{0.5\times}$	1.3M	150M	63.3
	MobileNetV2 $_{0.75\times}$	2.6M	209M	69.8
	AutoSlim-MobileNetV2	4.1M	207M	73.0
	MobileNetV3 $_{1.0\times}$	5.4M	219M	75.2
	MNasNet $_{A1}$	3.9M	315M	75.2
	Once-For-All	4.4M	230M	76.0
	Once-For-All $_{\text{finetuned}}$	4.4M	230M	76.4
	BigNASModel-S	**4.5M**	**242M**	**76.5**
400M FLOPs	NASNet $_{B}$	5.3M	488M	72.8
	MobileNetV2 $_{1.3\times}$	5.3M	509M	74.4
	MobileNetV3 $_{1.25\times}$	8.1M	350M	76.6
	MNasNet $_{A3}$	5.2M	403M	76.7
	EfficientNet $_{B0}$	5.3M	390M	77.3
	BigNASModel-M	**5.5M**	**418M**	**78.9**
600M FLOPs	MobileNetV1 $_{1.0\times}$	4.2M	569M	70.9
	NASNet $_{A}$	5.3M	564M	64.0
	DARTS	4.9M	595M	73.1
	EfficientNet $_{B1}$	7.8M	734M	79.2
	BigNASModel-L	**6.4M**	**586M**	**79.5**
1000M FLOPs	EfficientNet $_{B2}$	9.2M	1050M	80.3
	BigNASModel-XL	**9.5M**	**1040M**	**80.9**

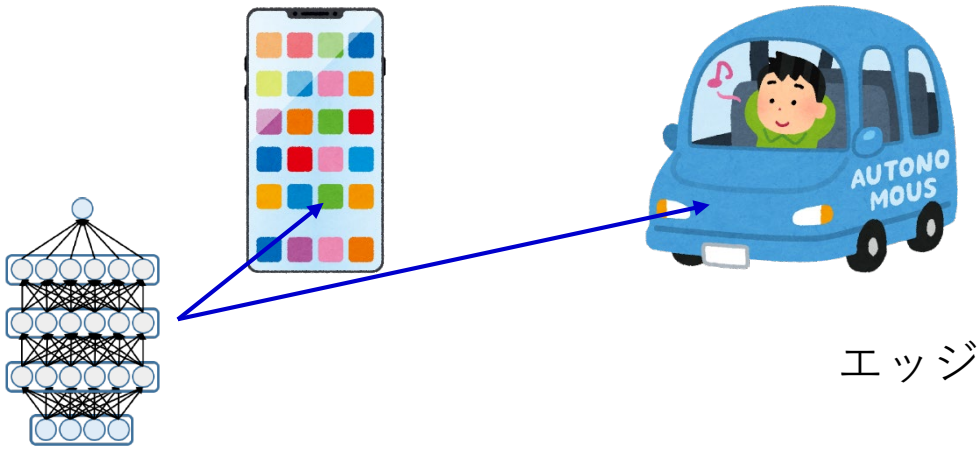


圧縮できるように学習するとスクラッチ学習より性能が向上することもある.

理論の副産物

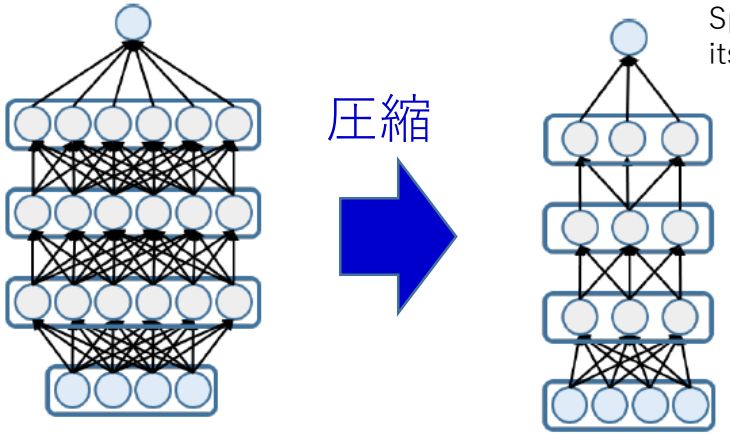
自由度の理論解析により，ネットワークのどこに着目すればどれだけ圧縮できるかがわかる．

深層ニューラルネットワークの圧縮技術への応用



エッジデバイスでの運用

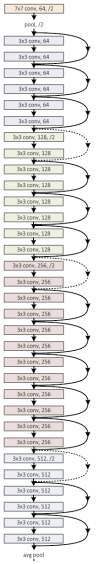
ニューラルネットワークの圧縮



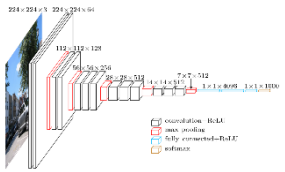
[Suzuki, Abe, Murata, Horiuchi, Ito, Wachi, Hirai, Yukishima, Nishimura: Spectral pruning: Compressing deep neural networks via spectral analysis and its generalization error. IJCAI-PRICAI 2020]

- メモリ消費量を減少
- 予測にかかる計算量を減少

→ 小型デバイスでの作動に有利
（自動運転など）



VGG-16ネットワークの圧縮



Model	Top-1	Top-5	# Param.	FLOPs
Original VGG	68.34%	88.44%	138.34M	30.94B
APoZ-2	70.15%	89.69%	51.24M	30.94B
ThiNet-Conv	69.80%	89.53%	131.44M	9.58B
ThiNet-GAP	67.34%	87.92%	8.32M	9.34B
Spec-Conv	**70.418%**	**90.094%**	131.44M	9.58B
Spec-GAP	**67.540%**	**88.270%**	8.32M	9.34B

提案手法：
従来手法より良い精度

94%の圧縮
（精度変わらず）

ResNet-50ネットワークの圧縮

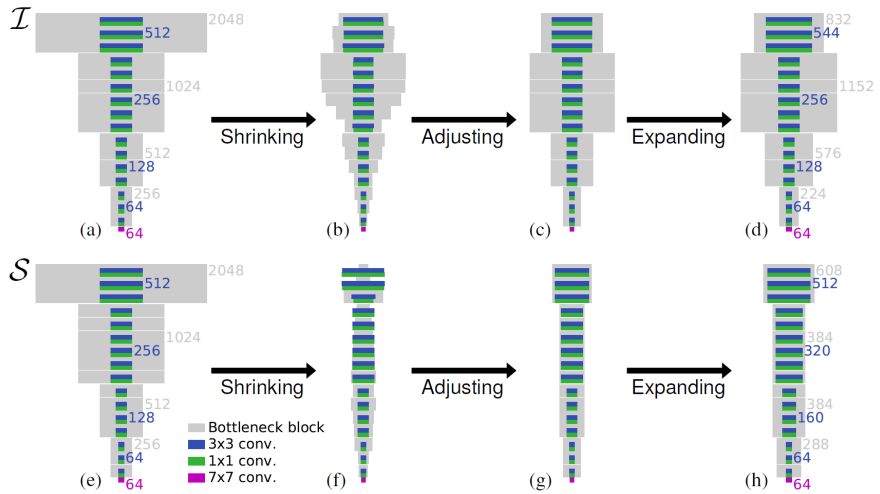
Model	Top-1	Top-5	# Param.	FLOPs
ResNet-50-1	72.89%	91.07%	25.56M	7.75G
ThiNet-70	72.04 %	90.67%	16.94M	4.88G
ThiNet-50	71.01 %	90.02%	12.38M	3.41G
NISP-50-A	72.68%	—	18.63M	5.63G
NISP-50-B	71.99%	—	14.57M	4.32G
Spec-ResA	72.99%	91.56%	12.38M	3.45G
ResNet-50-2	75.21%	92.21%	25.56M	7.75G
Sparse-reg wo/ ft	—	84.2%	19.78M	5.25G
Sparse-reg w/ ft	—	90.8%	19.78M	5.25G
Spec-ResB wo/ ft	66.12%	86.67%	20.69M	5.25G
Spec-ResB w/ ft	74.04%	91.77%	20.69M	5.25G

約半分に圧縮しても精度落ちず

転移学習のネットワーク構造決定

- ある閾値以上の固有値をカウント (e.g., 10^{-3}).
  → 縮小したネットワークのサイズとして使う.
- その後, スクラッチから学習 ($\mathcal{S}$) もしくはImageNet事前学習モデルをファインチューニングする ($\mathcal{I}$).

[Shinya, Simo-Serra, and Suzuki: Understanding the Effects of Pre-training for Object Detectors via Eigenspectrum. ICCV2019, Neural Architects Workshop]



Network size determination alg.

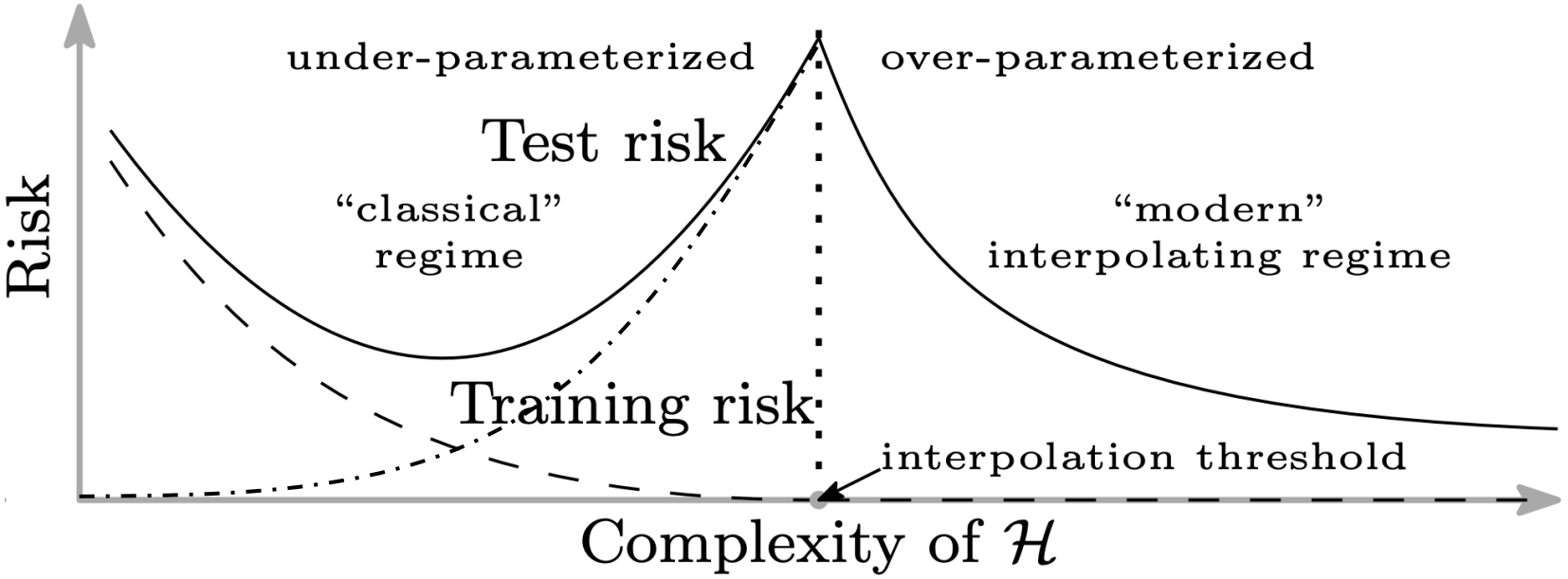
1: Set initial weights.
2: Train the whole network to find $\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}(\theta)$.
3: Calculate eigenspectra $S_{1:M}$.
4: Calculate intrinsic dimensionalities $d_{1:M}$ by $d_{1:M} = \mathrm{len}(S_{1:M} > T)$.
5: Determine new widths $O'_{1:M}$ by adjusting $d_{1:M}$.
6: Find the largest ω such that $c(\omega \cdot O'_{1:M}) \leq \zeta$.
7: **return** $\omega \cdot O'_{1:M}$.

Backbone	Normalization	Classification		COCO (2× schedule)					
		MACs	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-50 [35]	SyncBN	3.8 G	—	34.5	55.2	37.7	20.4	36.7	44.5
ResNet-50*	GN	4.09 G	25.5 M	35.5	55.6	38.5	21.3	37.5	45.3
ResiaxNet $\mathcal{S}$3-50 (MACs)	GN	4.06 G	18.6 M	35.4	55.4	38.6	21.5	37.3	45.2
ResiaxNet $\mathcal{I}$1-50 (MACs)	GN	4.05 G	21.7 M	35.5	55.5	38.6	21.4	37.3	**46.0**
ResiaxNet $\mathcal{I}$3-50 (MACs)	GN	4.07 G	22.0 M	35.4	55.6	38.4	21.3	37.8	45.5
ResiaxNet $\mathcal{I}$3-50 (params)*	GN	4.92 G	24.7 M	**35.8**	**55.9**	**38.9**	**21.8**	**38.0**	45.6
DetNet-59 [35]	SyncBN	4.8+ G	—	**36.3**	**56.5**	**39.3**	22.0	**38.4**	**46.9**
DetNet-59†	GN	5.00+ G	18.3+ M	36.2	56.0	**39.3**	22.1	38.3	46.0
DetiaxNet $\mathcal{I}$2-59 (MACs)	GN	4.94+ G	17.4+ M	36.2	56.0	**39.3**	**22.5**	38.1	46.0

Overparameterizeされた ネットワークの統計学

Double-descent (二重降下)

“新しい”バイアス-バリアンスのトレードオフ



- モデルがある複雑度 (サンプルサイズ) を超えた後，第二の降下が起きる.
- モデルサイズがデータより多いと推定量のバリアンスがむしろ減る.

※設定によるので注意が必要.

線形回帰における二重降下

- Hastie et al.: Surprises in High-Dimensional Ridgeless Least Squares Interpolation, arXiv:1903.08560.
- 線形回帰を考察

$$y_i = x_i^\top \beta + \epsilon_i$$

$$\mathbb{E}[\epsilon_i] = 0,\ \mathrm{Var}(\epsilon_i) = \sigma^2,\ \mathrm{Cov}(x_i) = \Sigma \qquad \|\beta\| = r$$

- 最小ノルム解：$\hat{\beta} = (X^\top X)^+ X^\top Y$

$$(X = [x_1, \ldots, x_n]^\top,\ Y = [y_1, \ldots, y_n]^\top)$$

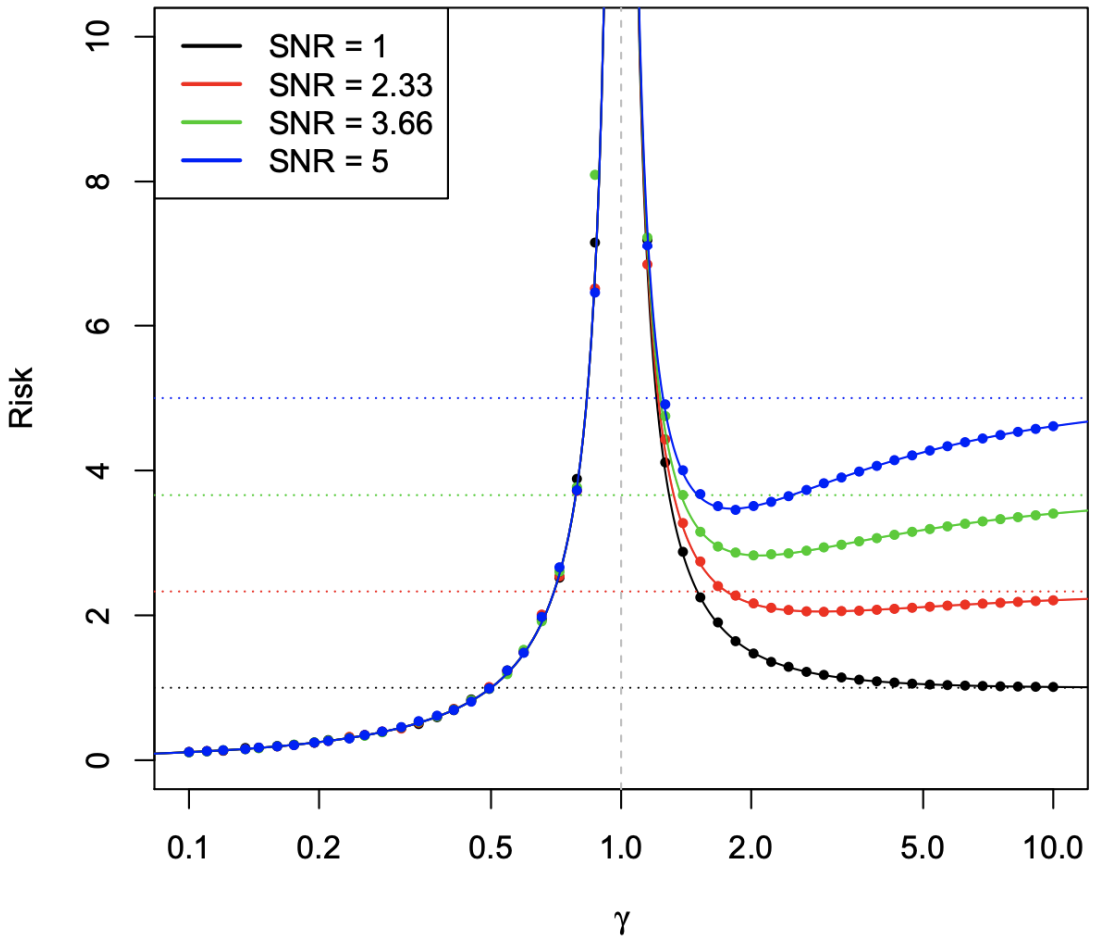
- 期待予測誤差：$\mathbb{E}_\epsilon[\|\hat{\beta} - \beta\|_\Sigma^2 | X]$

定理

期待予測誤差は以下の値に$n, d \to \infty$, かつ $d/n \to \gamma \in (0, \infty)$の極限で概収束する:

$$R(\gamma) = \begin{cases} \sigma^2 \frac{\gamma}{1-\gamma} & (\gamma < 1) \\ r^2 \left(1 - \frac{1}{\gamma}\right) + \sigma^2 \frac{1}{\gamma - 1} & (\gamma > 1) \end{cases}$$

バイアス　　バリアンス



直感：

- 次元(d)>サンプルサイズ(n)だとデータの張る部分空間は全体の一部
  →実質的自由度がdより低く，バリアンス小

注意：

- 次元が大きくなると真の関数も変化している設定.
- 単なる線形回帰なので第一層も学習する深層学習とは異なる.

2層NNの二重降下理論

[Mei, Song, and Andrea Montanari. "The generalization error of random features regression: Precise asymptotics and double descent curve." *arXiv preprint arXiv:1908.05355* (2019)]

Target: linear model

$$y_i = \langle x_i, \beta^* \rangle + \epsilon_i$$

Model: 2層NN

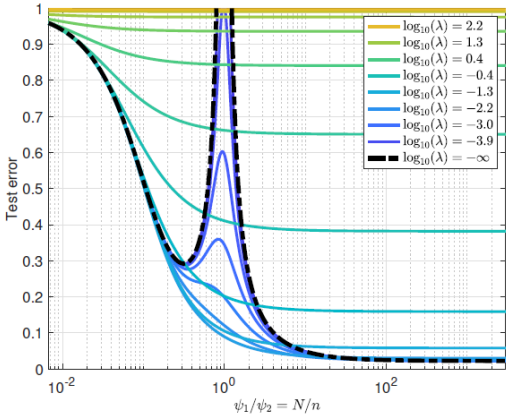
$$f(x) = \sum_{m=1}^{M} a_m \eta(\langle w_m, x \rangle / \sqrt{d})$$

Random featureモデル: $\hat{a} = \inf_{a \in \mathbb{R}^M} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{m=1}^{M} a_m \eta(\langle w_m, x \rangle / \sqrt{d}) \right)^2 + \frac{M}{d} \lambda \|a\|_2^2 \right\}$

(w_mはランダムに初期化して固定，二層目 (a_m) のみ学習)

$M, n, d \to \infty$で，$M/d \to \psi_1, n/d \to \psi_2$という状況.

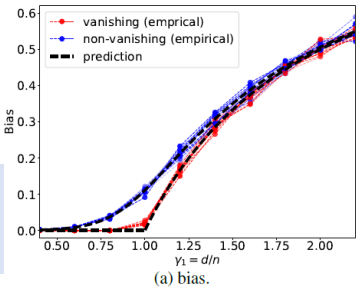
予測誤差が解析的に求まる．(長いので省略)
→ 横幅Mを広くとると二重降下が現れる

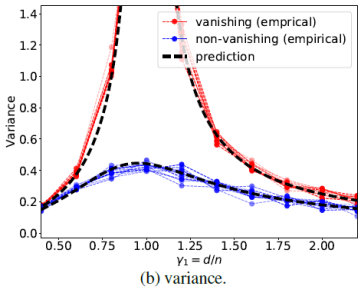


[Ba, Erdogdu, Suzuki, Wu, Zhang: Generalization of Two-layer Neural Networks: An Asymptotic Viewpoint. ICLR2020]

二層NNの学習．今度はa_mを固定してw_mを学習.
w_mの初期値が大きい状況 (NTK)→ 二重降下現れる
w_mの初期値が小さい状況 (平均場)→ 二重降下が弱い

小さな初期値から始めると真の関数を表す最小限の表現を獲得する
大きな初期値から始めるとカーネル法と似た状況で過学習しやすい





Implicit regularization（陰的正則化）

- ニューラルネットワークの学習では様々な「**陽的正則化**」を用いる：バッチノーマリゼーション，Dropout，Weight decay，…
- 実は深層学習の構造が自動的に生み出す「**陰的正則化**」も強く効いているという説.

例：線形ネットワーク

$$f_W := W^{(L)} W^{(L-1)} \dots W^{(1)} x$$

（L2正則化学習）

$$\widehat{W} = \operatorname*{argmin}_W \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \sum_{\ell=1}^{L} \|W^{(\ell)}\|_{\mathrm{F}}^2.$$

任意の局所最適解は低ランクになる：

$$\widehat{W}^{(L-1)} = \dots = \widehat{W}^{(1)} = \hat{u}\hat{v}^\top \quad (\text{rank } 1)$$

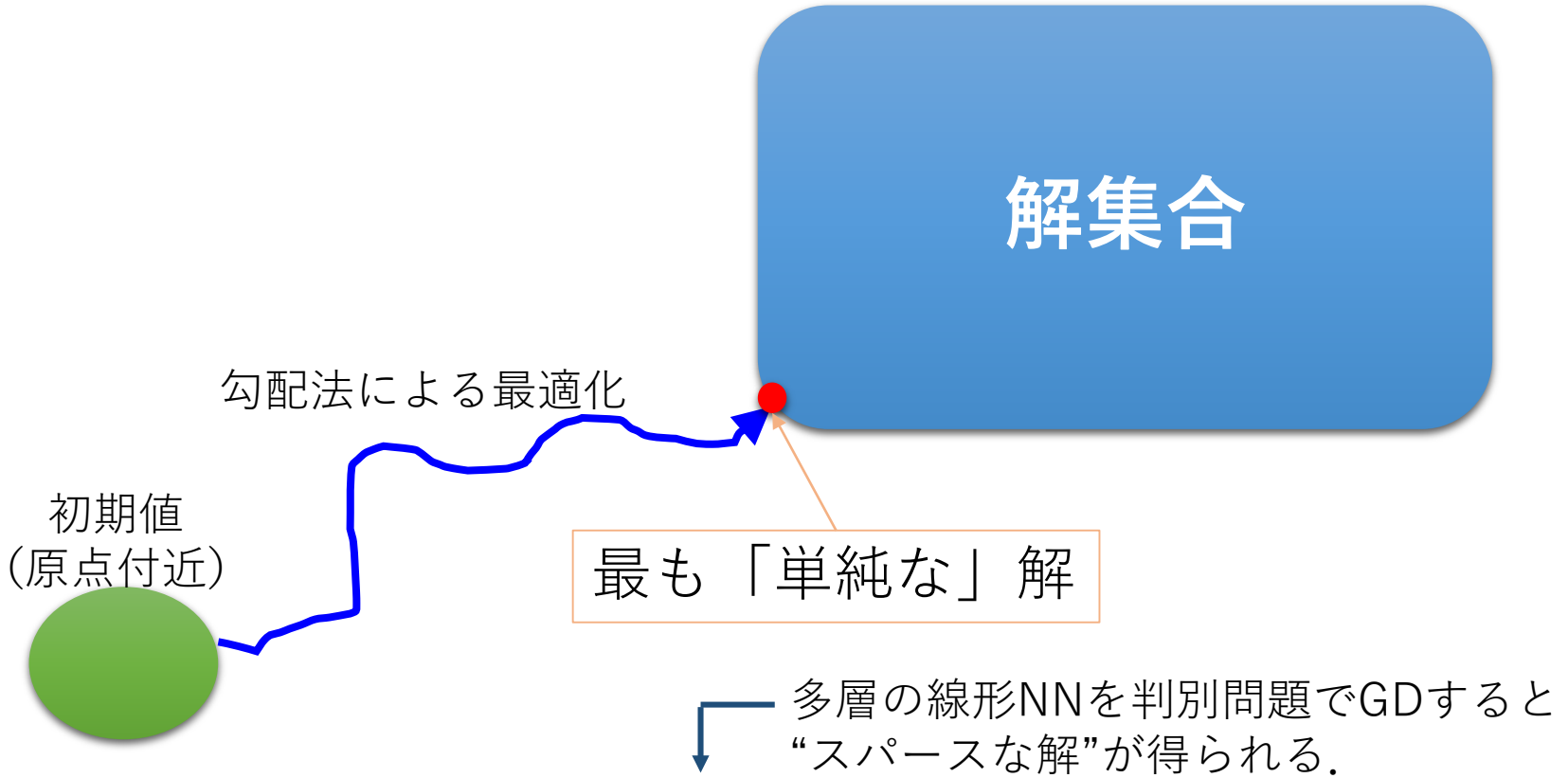
モデルの複雑さが大幅に削減されている.

（見た目のパラメータ数）$Lm^2 \rightarrow 2m$（実質的パラメータ数）

※ 非線形活性化関数がある場合は完全には解明されていない（論文は多数）

勾配法と陰的正則化

- 小さな初期値から勾配法を始めるとノルム最小化点に収束しやすい→陰的正則化

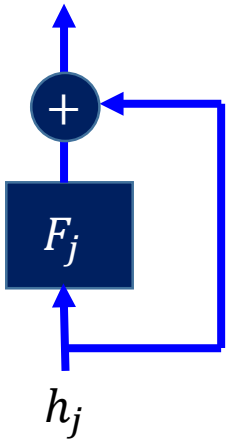


[Gunasekar et al.: Implicit Regularization in Matrix Factorization, NIPS2017]
[Soudry et al.: The implicit bias of gradient descent on separable data. JMLR2018]
[Gunasekar et al.: Implicit Bias of Gradient Descent on Linear Convolutional Networks, NIPS2018]
[Moroshko et al.: Implicit Bias in Deep Linear Classification: Initialization Scale vs Training Accuracy, arXiv:2007.06738]

ResNet

ResNetのODE解釈

ResNetの各層は特徴の最適化の一反復，常微分方程式の離散化とみなせる．

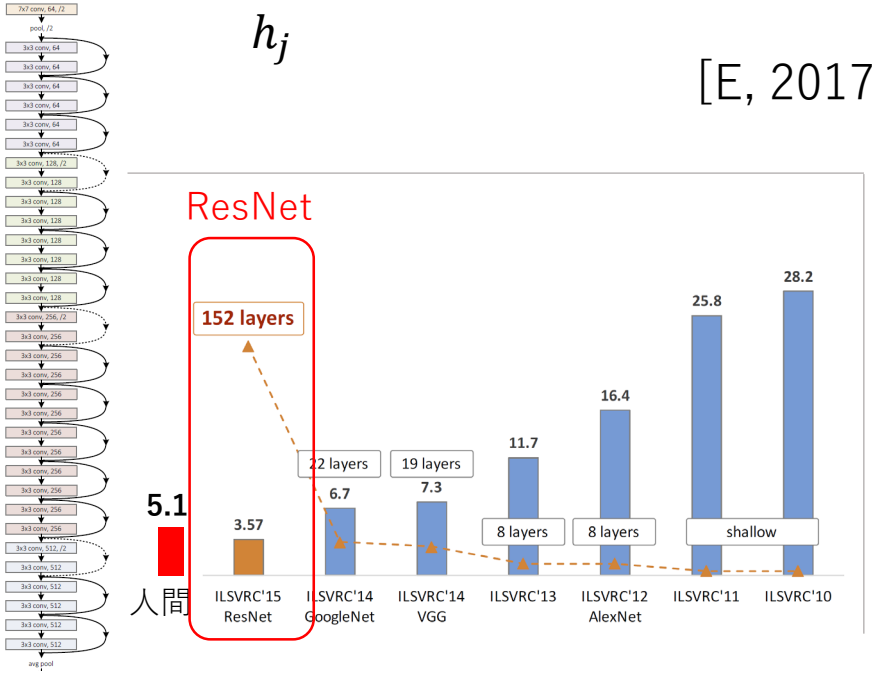


$$h_{j+1} = h_j + F_j(h_j)$$

$$\wr\wr$$

$$\frac{\mathrm{d}h_t}{\mathrm{d}t} = \underbrace{\nabla L(h_t)}_{(=F_t)} \circ h_t$$

[E, 2017][Sonoda & Murata, 2017][Li & Shi, 2017]



ResNetと常微分方程式をつなげることで常微分方程式の数値解法をネットワーク構造の決定に持ち込める．
→ PolyNet, FractalNet, RevNet, Linear-Multistep-ResNet, ...

[Lu et al.: Beyond Finite Layer Neural Networks: Bridging Deep Architectures and Numerical Differential Equations, ICML2018]

ODE-Net

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f_{\theta_t}(\mathbf{h}_t)$$

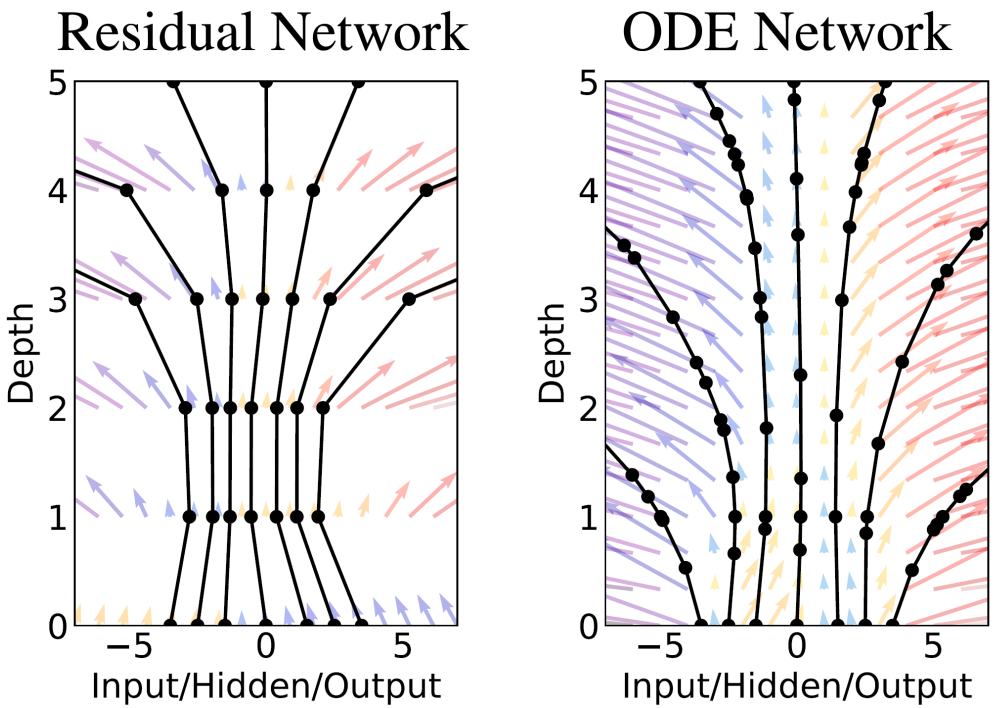
ResNet

連続化

$$\frac{\mathrm{d}\mathbf{h}(t)}{\mathrm{d}t} = f_\theta(\mathbf{h}(t), t)$$

ODE-Net

- 層を連続化することですべての層が一つのネットワークに集約される.
- ODEにすることで汎用ODEソルバーを適用できる.
  - ➢ 入力点に応じて離散化の粒度を変えられる. （層の概念がもはやない）

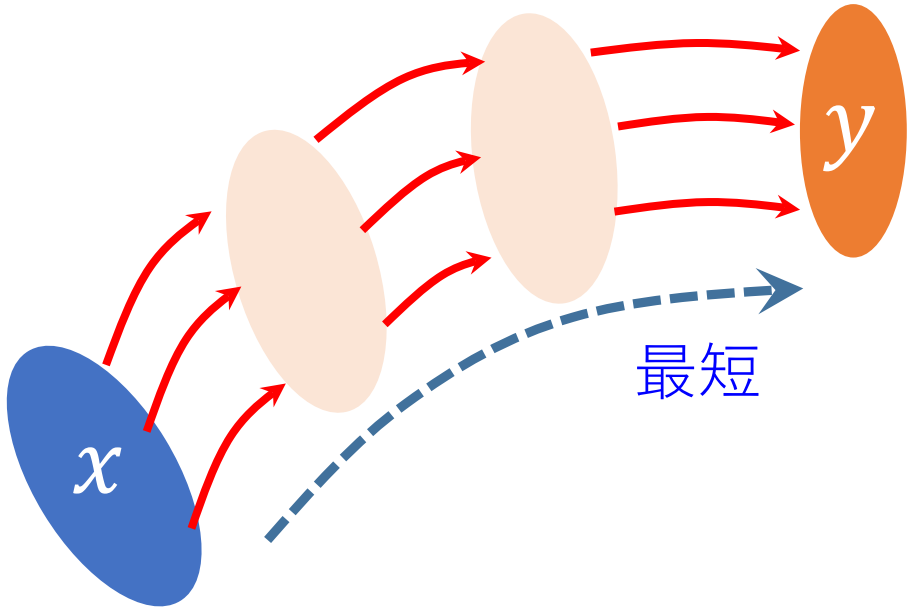


[Chen et al. Neural ordinary differential equations. NeurIPS2018 Best Paper.]

この方向性は現在「流行っている」
しかし，ODE的観点が汎化性能等の意味で本当に意味があり有用であるかは慎重な考察が必要である.

ResNetの平均場理論と最適制御

- ResNetは入力から出力へ“最短”で繋いでいる.
  → 最適制御理論による特徴付け (HJB方程式).



[E, Han, Li: A Mean-Field Optimal Control Formulation of Deep Learning. Research in the Mathematical Sciences, 6-10, 2019]
[Benning, Celledoni, Ehrhardt, Owren, Schönlieb: Deep learning as optimal control problems: Models and numerical methods. Journal of Computational Dynamics, 2019, 6(2) : 171-198.]

- 最適制御を用いたResNetの平均場における最適化理論研究もある.

  [Lu, Ma, Lu, Lu, and Ying: A mean-field analysis of deep resnet and beyond: Towards provable optimization via overparameterization from depth. ICML2020.]

ResNet型勾配ブースティング法

[Nitanda&Suzuki: Functional Gradient Boosting based on Residual Network Perception. ICML2018]
[Nitanda&Suzuki: Gradient Layer: Enhancing the Convergence of Adversarial Training for Generative Models. AISTATS2018]

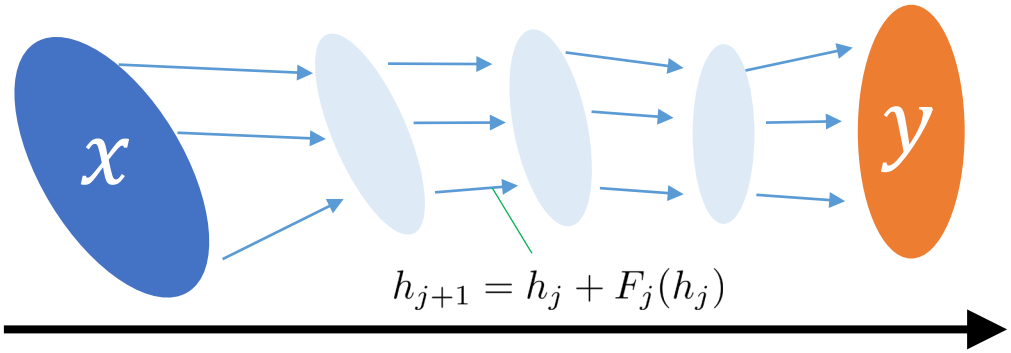
ResNetと勾配ブースティング法の類似性

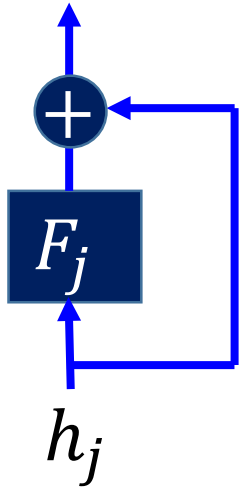
- **Residual Network**
  スキップコネクションを持つ巨大な深層ニューラルネットワーク.
  画像認識タスク等でSOTA.
- **勾配ブースティング (XGBoost, LightGBM)**
  予測器についての関数勾配によるブースティング法 (アンサンブル学習法).
  データマイニング系のコンペティションで最も有用とされるモデル.

ResFGB=両者の関連性に着目したブースティング法

[Veit, Wilber, &Belongie 2016, Littwin & Wolf 2016, Weinan 2017, Haber, Ruthotto, & Holtham 2017, Jastrzebski, Arpit, Ballas, Verma, Che, & Bengio 2017, Chang, Meng, Haber, Tung, and Begert 2017. etc]

1層加える=勾配法1反復





$$h_{j+1} = h_j + F_j(h_j)$$

$$\approx$$

$$\frac{\mathrm{d}h_t}{\mathrm{d}t} = \underbrace{\nabla L(h_t)}_{(=F_t)} \circ h_t$$

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)

ResFGB: ResNet型勾配ブースティング法

[Nitanda&Suzuki: Functional Gradient Boosting based on Residual Network Perception. ICML2018]

- ResNetの特性を備えた勾配ブースティング法

(識別問題の経験損失) $\min_{\phi,W} \mathcal{L}_n(\phi, W) = \widehat{\mathbb{E}}_{x,y}[l(W^\top \phi(x), y)]$.

特徴写像ϕについて$\mathcal{L}_n(\phi) = \min_W \mathcal{L}_n(\phi, W)$の関数勾配を用いた最適化.

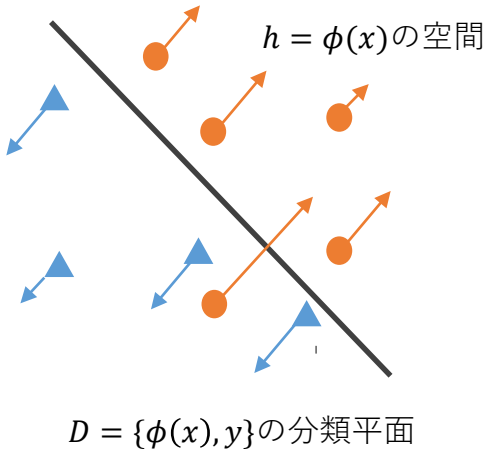
関数勾配$\nabla_\phi \mathcal{L}_n(\phi)$：訓練データの特徴 $\phi(x)$の線形分類可能性を向上させるための方向の群れ.

未知データに適用するには平滑化が必要.

→ カーネル$k(x, x') = \iota(\phi(x))^\top \iota(\phi(x'))$で畳み込む.
ιはNNで十分な降下方向を与えるよう学習.

$$T_k \nabla_\phi \mathcal{L}_n(\phi) = \widehat{\mathbb{E}}_{x,y}\left[\nabla_\phi \mathcal{L}_n(\phi)(x)\iota(\phi(x))^\top\right]\iota(\phi(\cdot))$$

残差ブロック：平滑化された関数勾配.
作り方から直交しない限り常に損失関数の降下方向.



ResFGBの概要

勾配ブースティングの一反復=ResNetの層追加.
関数空間での最適化の観点からマージン分布の最小化を示せる.
→ **新しい勾配ブースティングの理論に基づいた汎化保証付き深層ResNetの学習.**

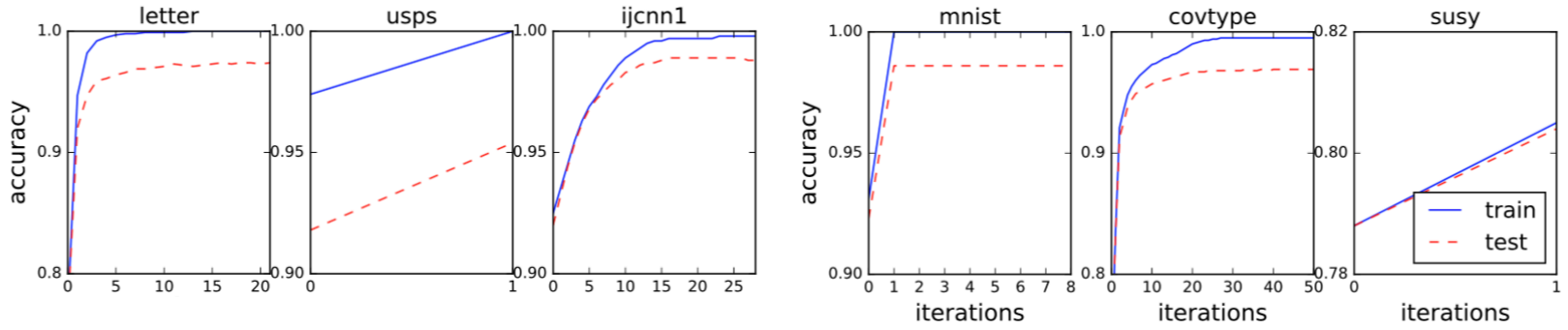
提案手法ResFGBの数値実験

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)

提案手法

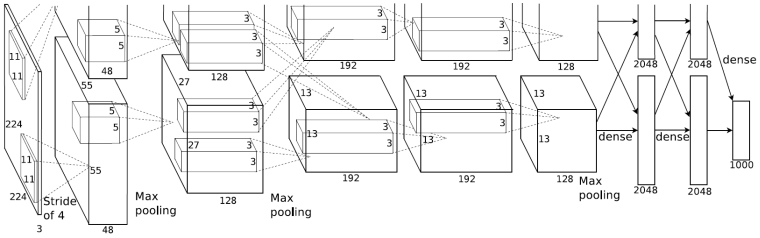
METHOD	LETTER	USPS	IJCNN1	MNIST	COVTYPE	SUSY
RESFGB (LOGISTIC)	**0.975 (0.0016)**	**0.954 (0.0006)**	0.987 (0.0011)	0.985 (0.0007)	0.968 (0.0017)	**0.804 (0.0000)**
RESFGB (SMOOTH HINGE)	**0.975 (0.0012)**	0.950 (0.0022)	**0.988 (0.0018)**	**0.987 (0.0010)**	0.965 (0.0058)	**0.804 (0.0004)**
SUPPORT 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)**

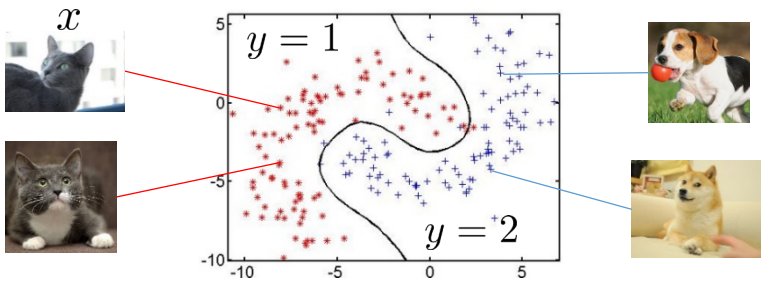
SOTAとされるLightGBM以上の精度を確認．
幾つかのデータでは数反復で収束．通常の勾配ブースティングより**効率的な最適化．**
↑関数勾配の平滑化の仕方の差異による．



第３章
深層学習の最適化

深層学習の“学習”





深層ニューラルネットワークをデータにフィットさせるとは？

$$L(W) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(W)$$

W: パラメータ

i番目のデータで正解していれば小さく，間違っていれば大きく

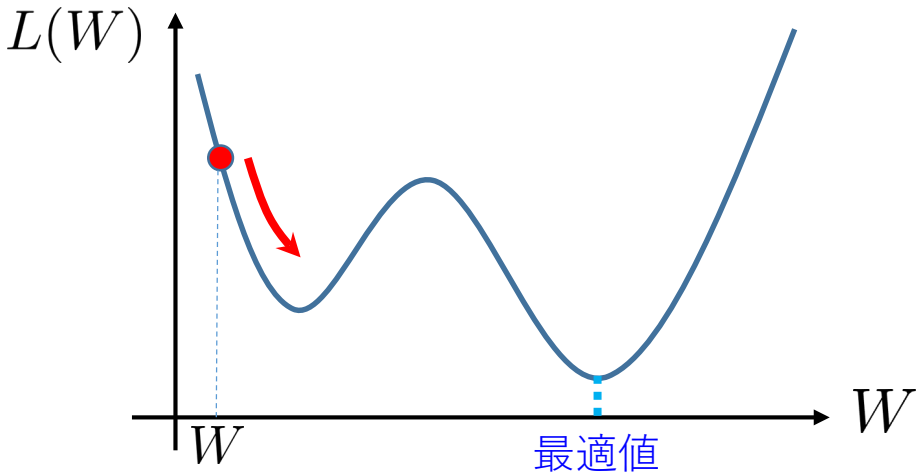
損失関数：データへの当てはまり度合い

損失関数最小化

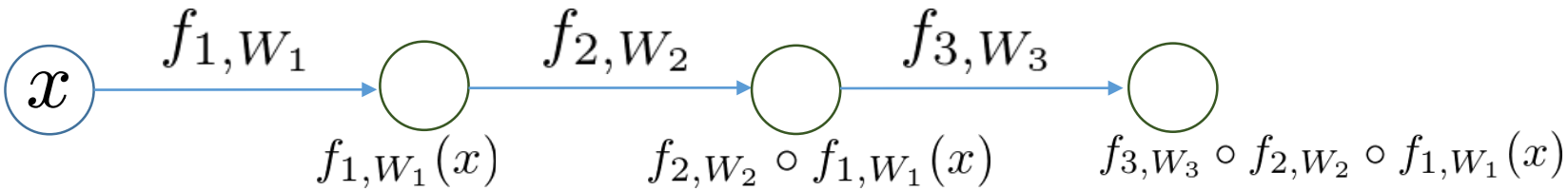
$$\min_W L(W)$$

(Wは数十億次元)

通常，**確率的勾配降下法**で最適化



誤差逆伝搬法



例： $f_{1,W_1}(x) = h(W_1 x)$

合成関数

$$f(x;W) = f_{3,W_3}(f_{2,W_2}(f_{1,W_1}(x)))$$
$$= f_{3,W_3} \circ f_{2,W_2} \circ f_{1,W_1}(x)$$

合成関数の微分

$$\frac{\partial f}{\partial W_1}(x) = \frac{\partial f_{3,W_3}}{\partial f_{2,W_2}} \frac{\partial f_{2,W_2}}{\partial f_{1,W_1}} \frac{\partial f_{1,W_1}}{\partial W_1}(x)$$

深層学習で主に使われる確率的最適化法

- **SGD**
- モーメンタム **SGD** (Nesterov の加速法と類似，一番よく利用されている)

$$
\begin{aligned}
g_t &= \nabla L(w_t) \\
\Delta w_t &= \theta \Delta w_{t-1} - (1-\theta)\eta g_t \\
w_{t+1} &= w_t + \Delta w_t
\end{aligned}
$$

- **Nesterov の加速法** (凸関数における加速法と同様，パラメータの設定は異なる)

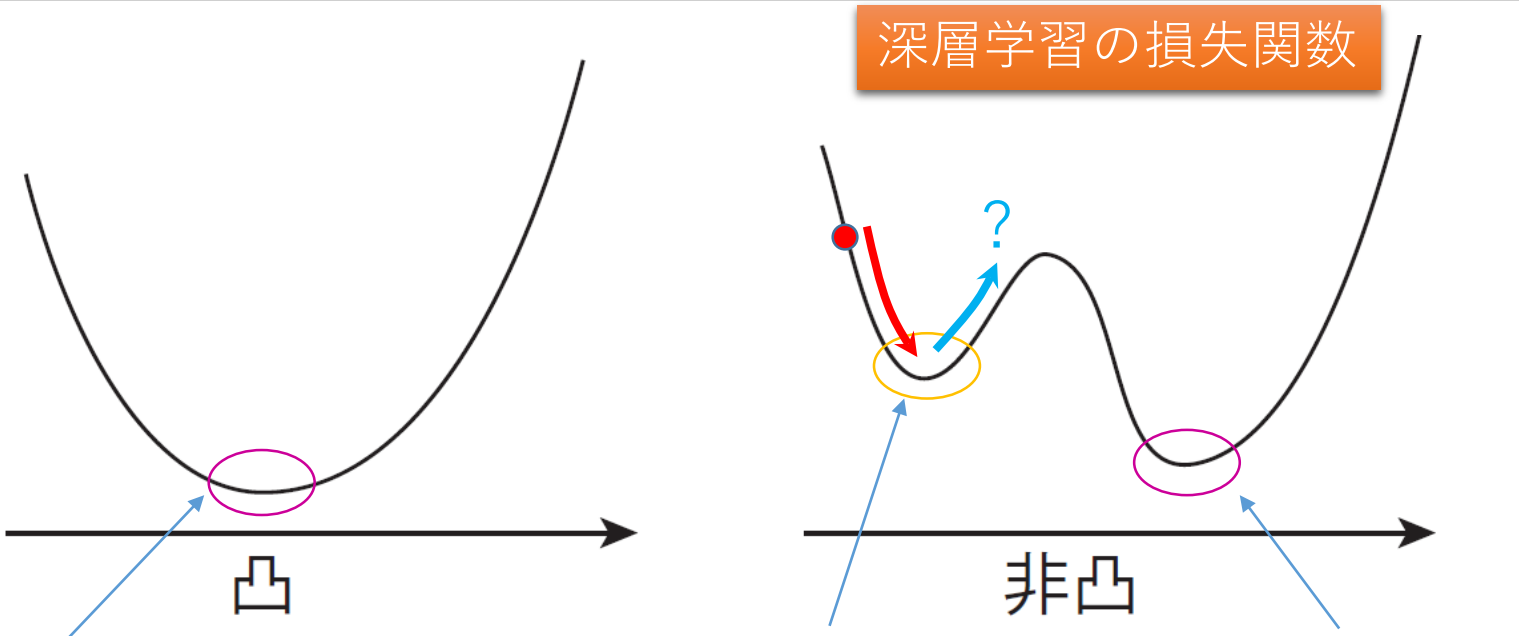
$$
g_t = \nabla L(w_t + \theta \Delta w_{t-1}), \Delta w_t = \theta \Delta w_{t-1} - (1-\theta)\eta g_t, w_{t+1} = w_t + \Delta w_t
$$

- **AdaGrad** (Duchi et al., 2011)
- **Adam** (Kingma and Ba, 2014) : AdaGrad と加速法を組み合わせたような方法．AdaGrad と違い，勾配のノルムを減衰させて次に伝える．モーメンタム SGD と並んでよく使われている．
- **RMSprop** (Hinton et al.) : AdaGrad において勾配のノルムを減衰させて和を取る方法．

問題点

目的関数が非凸関数

凸関数 $\theta f(x) + (1-\theta) f(y) \geq f(\theta x + (1-\theta) y) \quad (\forall x, y \in \mathbb{R}^p, \theta \in [0,1])$



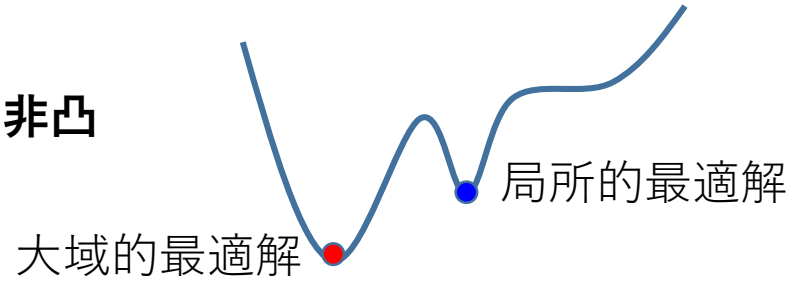
局所最適解や鞍点にはまる可能性あり

“**狭い**”ネットワークの学習はNP-完全:

- Judd (1988), Neural Network Design and the Complexity of Learning.
- Blum&Rivest (1992), Training a 3-node neural network is NP-complete.

大域的最適性

深層学習の目的関数は非凸



- 線形深層NNの局所的最適解は全て大域的最適解：
  Kawaguchi, 2016; Lu&Kawaguchi, 2017.

  ※ただし対象は線形NNのみ.

  → 臨界点が大域的最適解であることの条件も出されている
  (Yun, Sra&Jadbabaie, 2018)

- 低ランク行列補完の局所的最適解は全て大域的最適解：
  Ge, Lee&Ma, 2016; Bhojanapalli, Neyshabur&Srebro, 2016.

$$\min_{U \in \mathbb{R}^{M \times k}} \sum_{(i,j) \in E} (Y_{i,j} - (UU^\top)_{i,j})^2$$

Loss landscape

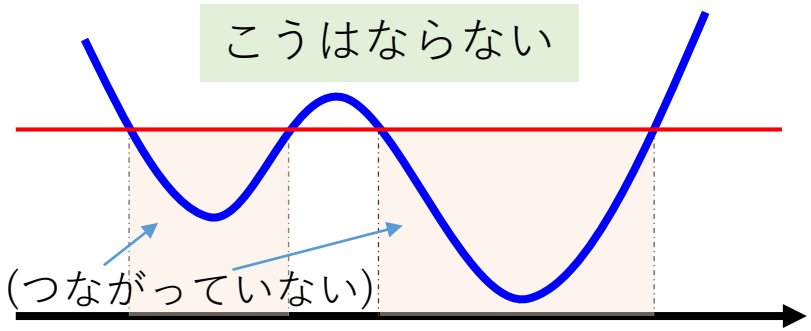
- 横幅の広いNNの訓練誤差には孤立した局所最適解がない．（局所最適解は大域的最適解とつながっている）

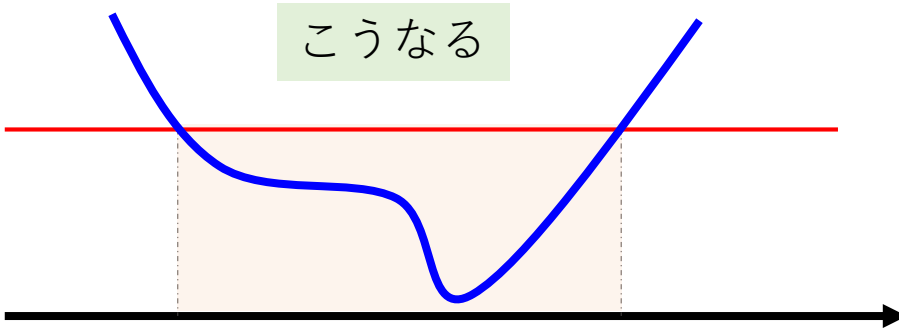
※とはいえ，勾配法で大域的最適解に到達可能かは別問題．

定理

n個の訓練データ$(x_i, y_i)_{i=1}^n$が与えられているとする．損失関数ℓは凸関数とする．
任意の連続な活性化関数について，横幅がデータサイズより広い（$M \geq n$）二層NN$f_{(a,W)}(x) = \sum_{m=1}^{M} a_m \, \eta(w_m^{\mathsf{T}} x)$に対する訓練誤差$\hat{L}(a, W) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_{(a,W)}(x_i))$の任意のレベルセットの弧状連結成分は大域的最適解を含む．言い換えると，任意の局所最適解は大域的最適解である．

[Venturi, Bandeira, Bruna: Spurious Valleys in One-hidden-layer Neural Network Optimization Landscapes. JMLR, 20:1-34, 2019.]





（参考）2層NN-非線形活性化関数-

二層目の重みを固定する設定

(Tian, 2017; Brutzkus and Globerson, 2017; Li and Yuan, 2017; Soltanolkotabi, 2017; Soltanolkotabi et al., 2017; Shalev-Shwartz et al., 2017; Brutzkus et al., 2018)

固定　こちらのみ動かす

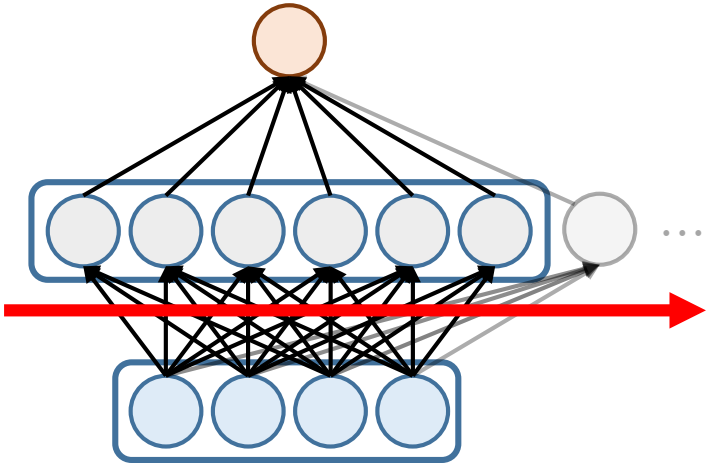
$$y = \sum_{j=1}^{k} v_j \eta(w_j^\top x + b_j)$$

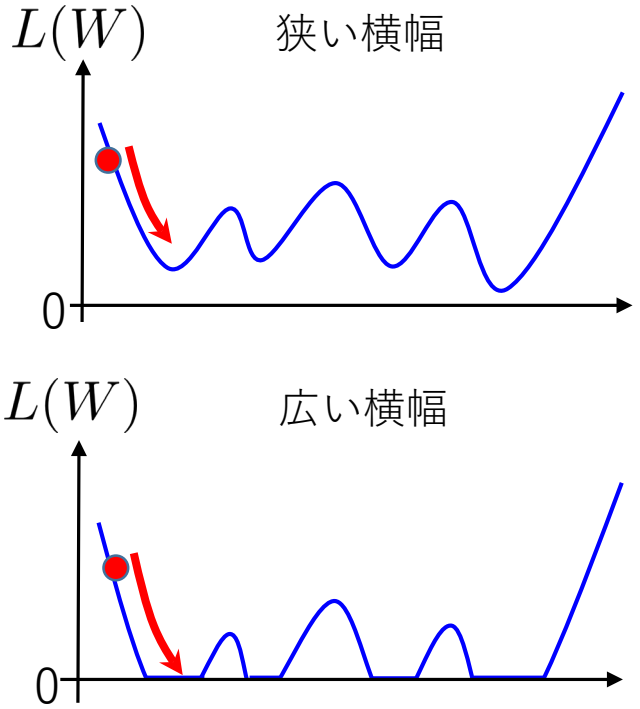
- Li and Yuan (2017): ReLU，入力はガウス分布を仮定
  - SGDは多項式時間で大域的最適解に収束
  - 学習のダイナミクスは2段階
    → 最適解の近傍へ近づく段階 + 近傍での凸最適化的段階
- Soltanolkotabi (2017): ReLU，入力はガウス分布を仮定
  - 過完備（横幅>サンプルサイズ）なら勾配法で最適解に線形収束
    (Soltanolkotabi et al. (2017)は二乗活性化関数でより強い帰結)
- Brutzkus et al. (2018): ReLU
  - 線形分離可能なデータなら過完備ネットワークで動かしたSGDは大域的最適解に有限回で収束し，過学習しない．
    （線形パーセプトロンの理論にかなり依存）

Li and Yuan (2017): Convergence Analysis of Two-layer Neural Networks with ReLU Activation.
Soltanolkotabi (2017): Learning ReLUs via Gradient Descent.
Brutzkus, Globerson, Malach and Shalev-Shwartz (2018): SGD learns over parameterized networks that provably generalized on linearly separable data.

オーバーパラメトライゼーション

横幅が広いと局所最適解が大域的最適解になる.





自由度が上がるため，初期値から最適解(完全フィット)へ到達しやすい.

- 二種類の解析手法
  - Neural Tangent Kernel
  - Mean-field analysis (平均場解析)

二つのスケーリング

$$f_W(x) = \sum_{j=1}^{M} a_j \eta(w_j^\top x)$$

- Neural Tangent Kernelのregime (lazy learning)
  - $a_j = O(1/\sqrt{M})$ [Jacot+ 2018][Du+ 2019][Arora+ 2019]
- 平均場解析のregime
  - $a_j = O(1/M)$ [Nitanda & Suzuki (2017), Chizat & Bach (2018), Mei, Montanari, & Nguyen (2018)]

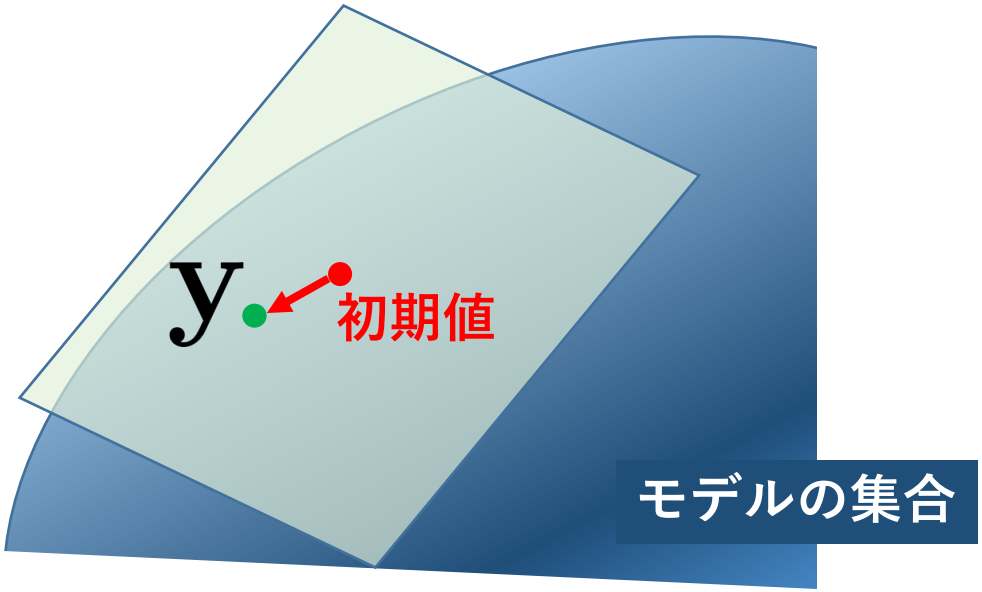
※NTKの$1/\sqrt{M}$自体はそこまで本質ではない，$1/M$より大きいことが重要.

初期化のスケーリングによって，初期値と比べて学習によって動く大きさの割合が変わる.
→ 学習のダイナミクス，汎化性能に影響
（解析の難しさも違う）

NTK

$$f_W(x) \simeq (W - W^{(0)})^\top \nabla_W f_{W^{(0)}}(x)$$

初期値のスケールが大きいので，初期値周りの線形近似でデータにフィットできてしまう．



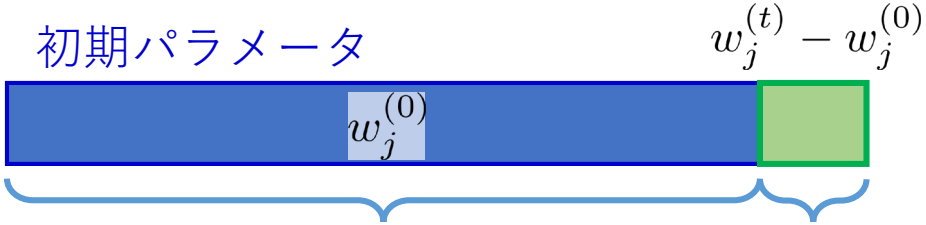
NTKと平均場の違い

$$f_W(x) = \sum_{j=1}^{M} a_j \eta(w_j^\top x)$$

η: ReLUとする．$a_j = O(1), w_j = O(1/\sqrt{M})$
または$w_j = O(1/M)$とスケール変換

- 各w_jが$O(1/M)$だけ動けば，全体としてO(1)の変化(データにフィットできる).
- 横幅は十分大きく取る: $M \gg n$ (overparameterization)

NTK：相対的変化小

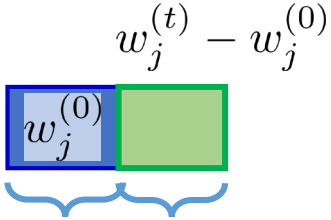


$$\eta(w_j^{(t)\top} x) - \eta(w_j^{(0)\top} x)$$
$$\simeq (w_j^{(t)} - w_j^{(0)})^\top x \eta'(w_j^{(0)\top} x)$$

NTKの特徴写像

テイラー展開により線形モデルとみなせる
→カーネル法の理論に帰着できる

平均場：相対的変化大



相対的な変化が大きいのでテイラー展開ができない.
→ 本質的に非凸最適化になる.

（原理的には展開しても良いが，
グラム行列の正定値性が保証されない）

Neural Tangent Kernel

連続時間ダイナミクスを考える.

Model： $f_W(x) = \sum_{j=1}^{M} a_j \eta(w_j{}^\top x)$

- a_j は固定
- w_j を学習

[Jacot, Gabriel&Hongler, NeurIPS2018]

$$\frac{\mathrm{d}w_j}{\mathrm{d}t} = -\nabla_{w_j}\hat{L}(f_W) \qquad \text{(Gradient descent, GD)}$$

$$= -\frac{1}{n}\sum_{i=1}^{n} \ell_i'(f_W(x_i)) a_j \nabla_{w_j}\eta(w_j^\top x_i)$$

$\nabla_{w_j}\eta(w_j^\top x_i) = x_i \eta'(w_j^\top x_i)$

(勾配法による更新)

$$\frac{\mathrm{d}f_W(x)}{\mathrm{d}t} = \sum_{j=1}^{M} a_j \nabla_{w_j}^\top \eta(w_j^\top x)\frac{\mathrm{d}w_j}{\mathrm{d}t}$$

$$= -\frac{1}{n}\sum_{i=1}^{n}\left(\underbrace{\sum_{j=1}^{M} a_j^2 \nabla_{w_j}^\top \eta(w_j^\top x)\nabla_{w_j}\eta(w_j^\top x_i)}_{k_W(x,x_i)}\right)\ell_i'(f_W(x_i))$$

$O(1/M)$：
特徴写像の内積の平均

$\ell_i'(f_W(x_i))$ residual (関数勾配)

Neural Tangent Kernel

目的関数の減少速度

$$\frac{\mathrm{d}\hat{L}(f_W)}{\mathrm{d}t} = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathrm{d}f_W(x_i)}{\mathrm{d}t} \ell_i'(f_W(x_i))$$

$$= -\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \ell_i'(f_W(x_i)) \underbrace{k_W(x_i, x_j)}_{(K_W)_{i,j}} \ell_j'(f_W(x_j))$$

$$= -\frac{1}{n^2}\|\nabla_f \hat{L}(f_W)\|_{K_W}^2$$

$$\leq -\lambda_{\min}\frac{1}{n^2}\|\nabla_f \hat{L}(f_W)\|^2 \quad (\lambda_{\min}: \text{グラム行列の}\underline{\textbf{最小固有値}})$$

Fact [Du et al., 2018; Allen-Zhu, Li & Song, 2018]

- <u>ランダム初期化</u>しておけば，$K_{W(0)} \succ \epsilon I$ が高確率で成立．
- 最適化の最中に最小固有値は正のまま $(\geq \epsilon/2)$．

→ **線形収束** $(\exp(-\boldsymbol{\lambda_{\min} t}))$

ランダム初期値とNTKの正定値性

$$K_{\infty,i,j} = \mathbb{E}_{w\sim N(0,I)}[x_i^\top x_j \eta'(w^\top x_i)\eta'(w^\top x_j)]$$

(横幅無限大のNTK)

補題

$$\|x_i\| = 1,\ \|x_i - x_j\| \geq \phi \quad \Rightarrow \quad K_\infty \succeq C\phi n^{-2}$$

Hoeffdingの不等式より

$$P\left(|K_{W^{(0)},i,j} - K_{\infty,i,j}| \leq \sqrt{\frac{\log(2/\delta')}{2M}}\right) \geq 1-\delta'$$

一様バウンドを取って

$$P\left(\|K_{W^{(0)}} - K_\infty\|_{\mathrm{F}}^2 \leq n^2 \frac{\log(2n^2/\delta)}{2M}\right) \geq 1-\delta$$

十分横幅Mが広ければ，ランダム初期化した$K_{W^{(0)}}$の正定値性が保証される．

Optimization in NTK regime

以下のように初期化する:

- $a_j \sim (\pm 1)\frac{1}{\sqrt{M}}$ ($+, -$ is generated evenly)
- $w_j \sim N(0, I)$

$$f_W(x) = \sum_{j=1}^{M} a_j \eta(w_j^\top x)$$

Theorem [Arora et al., 2019]

$M = \Omega(n^2 \log(n)/\lambda_{\min})$とすれば, 勾配法によって大域的最適解へ線形収束し, その汎化誤差は$\sqrt{\boldsymbol{y}^\top (K_{W(0)})^{-1}\boldsymbol{y}/n}$で抑えられる.

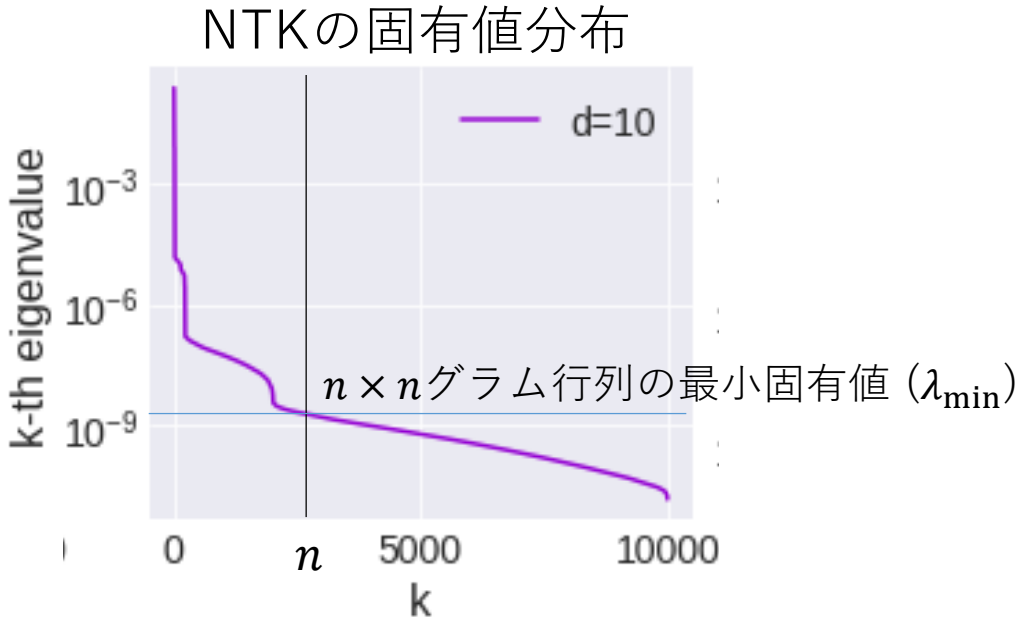
See also[Du et al., 2018; Allen-Zhu, Li & Song, 2018; Li & Liang, 2018]

- 訓練誤差0の解に線形収束する.
- 汎化誤差も一応抑えられている.

- データに完全にフィットさせてしまうので過学習の可能性あり.
- Early stoppingや正則化を入れれば過学習を防げる. (次ページ)

Spectral bias

- 最適化の観点からはoverparameterizationは有用に見える.
- 汎化誤差はどうであろうか?



- グラム行列の最小固有値は小さい $(1/\mathrm{poly}(n))$.
- 固有値の減少レートは多項式オーダー (理論+実験).

  → Spectral bias: 汎化の意味では好ましい.

Kernelによる平滑化という視点

- Frechet 微分 in $L_2(P_n)$: $\nabla_f \hat{L}(f)$

$$\nabla_f \hat{L}(f) = (\ell_i'(f(x_i)))_{i=1}^n$$

$$\hat{L}(f+h) = \hat{L}(f) + \langle \nabla_f \hat{L}(f), h \rangle_{L_2(P_n)} + o(\|h\|^2_{L_2(P_n)})$$

- **平滑化**積分作用素:

$$T_k f(x) := \int k(x, x') f(x') \mathrm{d}P_n(x')$$

$$T_{k_W} \phi_j = \mu_j \phi_j$$

- NTKにおける勾配は関数勾配を平滑化したもの:

$$\frac{\mathrm{d}f_W}{\mathrm{d}t} = -T_{k_W} \nabla_f \hat{L}(f_W)$$

勾配を平滑化！

$$(= -\tfrac{1}{n} \textstyle\sum_{i=1}^n k_W(\cdot, x_i) \ell_i'(f_W(x_i)))$$

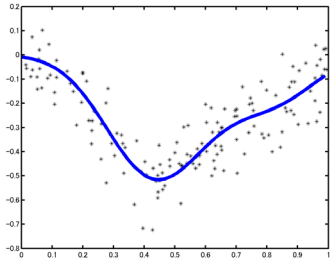
k_Wが高周波成分に小さな固有値を持てば，T_{k_W}は平滑化作用素として働く→ 帰納的バイアス (inductive bias).

NTK regimeでのSGD

モデル:

$$f_{a,W}(x) = \frac{1}{\sqrt{M}} \sum_{j=1}^{M} a_j \eta(w_j{}^\top x)$$

(We train both of first and second layers)



目的関数:

$$L(a, W) = \mathbb{E}[(Y - f_{a,W}(X))^2] + \frac{\lambda}{2}(\|a - a^{(0)}\|^2 + \|W - W^{(0)}\|_{\mathrm{F}}^2)$$

期待損失　　初期値からのずれ

$Y = f^*(X) + \epsilon$ （ノイズありの観測）

Averaged Stochastic Gradient Descent

for $t = 0$ **to** $T - 1$ **do**

Randomly draw a sample $(x_t, y_t) \sim \rho$

Perform SGD update for all $j \in \{1, \ldots, M\}$:

$$a_j^{(t+1)} = a_j^{(t)} - \alpha_t[\nabla_a \ell(y_t, f_{a^{(t)},W^{(t)}}(x_t)) + \lambda(a^{(t)} - a^{(0)})]$$

$$W_j^{(t+1)} = W_j^{(t)} - \alpha_t[\nabla_W \ell(y_t, f_{a^{(t)},W^{(t)}}(x_t)) + \lambda(W^{(t)} - W^{(0)})]$$

end for

Return $\bar{a}^{(T)} = \frac{1}{T}\sum_{t=0}^{T-1} a^{(t)}$, $\bar{W}^{(T)} = \frac{1}{T}\sum_{t=0}^{T-1} W^{(t)}$.

NTKにおける余剰誤差の速い収束

[Nitanda&Suzuki: Fast Convergence Rates of Averaged Stochastic Gradient Descent under Neural Tangent Kernel Regime, 2020.]

仮定：真の関数がNTKの作るRKHSに入っているとする.

NTK設定で適切な正則化を入れたSGDは“速い学習レート”を達成できる.

→ NTKによるsmoothingのおかげ.

Thm (速い収束レート)

f_T: T回更新後の解

NTKの固有値の減衰レート

$$\mathbb{E}[\|f_T - f^*\|_{L_2}^2] \leq \epsilon_M + O(T^{-\frac{2r\beta}{2r\beta+1}})$$

- ϵ_M: $M \to \infty$で0に収束する項
- $T^{-\frac{2r\beta}{2r\beta+1}}$: 速い学習レート ($O(1/\sqrt{T})$より速い)

→ $T^{-\frac{2r\beta}{2r\beta+1}}$ は**ミニマックス最適レート.**

(各種パラメータの意味は次ページに詳細)

仮定

2層NNのNTK:

$$k_\infty(x,x') = \mathbb{E}_{w^{(0)}}[\eta(w^{(0)\top}x)\eta(w^{(0)\top}x')] + \mathbb{E}_{w^{(0)}}[\eta'(w^{(0)\top}x)\eta'(w^{(0)\top}x')x^\top x]$$

横幅無限における積分作用素:

$$T_{k_\infty} f(x) = \int k_\infty(x,x') f(x') \mathrm{d}P_X$$

population

スペクトル分解: $T_{k_\infty}\phi_j = \mu_j\phi_j$, $k_\infty(x,x') = \sum_{j=1}^{\infty} \mu_j\phi_j(x)\phi_j(x')$

仮定

- $f^*(x) = \mathrm{E}[Y|X=x]$ が次のように書ける:
$$T_{k_\infty}^r h = f^*$$
for $h \in L_2(P_X)$, and $r \in [1/2,1]$.
- 固有値減衰条件:
$$\mu_j = \mathrm{O}(j^{-\beta}).$$

真の関数の平滑性

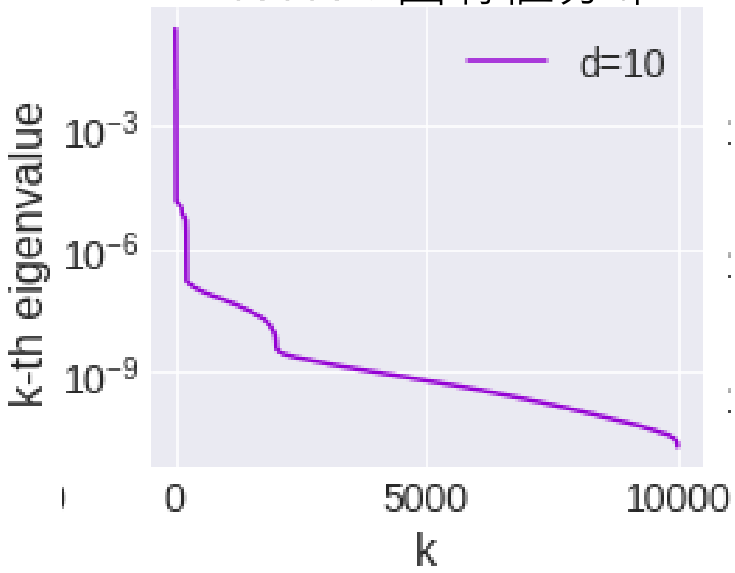
カーネル関数の“複雑さ”

カーネルリッジ回帰の解析における標準的な仮定; see, e.g., Dieuleveut et al. (2016); Caponnetto and De Vito (2007) (rの条件はやや強め).

$$k_\infty(x, x') = \sum_{m=1}^{\infty} \lambda_m \phi_m(x)\phi_m(x')$$

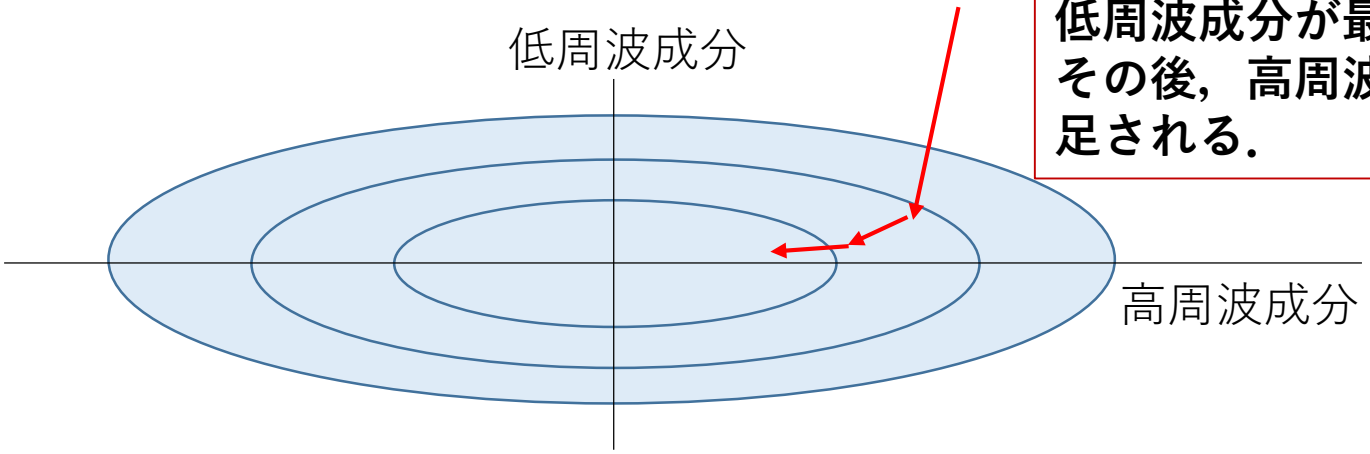
NTKの固有値固有関数分解
$(\phi_m)_{m=1}^{\infty}$: 固有関数. $L_2(P_X)$内の正規直交基底.

NTKの固有値分布



実際のNTKの固有値は多項式オーダーで減衰する.

[Bietti&Mairal (2019); Cao et al. (2019); Ronen et al. (2019)]



低周波成分が最初に補足される. その後, 高周波成分が徐々に補足される.

Beyond kernel

問題点：NTKは解析がしやすいが，結局カーネル法の範疇なので深層学習の“良さ”が現れない.

➢ NTKをはみ出す理論の試みがいくつかなされている.
（今後発展が予想される）

- Allen-Zhu&Li (2019,2020)

  Allen-Zhu&Li: What Can ResNet Learn Efficiently, Going Beyond Kernels? NIPS2019.
  Allen-Zhu&Li: Backward Feature Correction: How Deep Learning Performs Deep Learning. arXiv:2001.04413.
  (ResNet型ネットワークでカーネルを優越する状況)

- Li, Ma&Zhang (2019)

  Li, Ma&Zhang: Learning Over-Parametrized Two-Layer ReLU Neural Networks beyond NTK. arXiv:2007.04596.
  (テンソル分解の理論で深層学習がカーネルを優越することを示した)

- Bai&Lee (2020)

  Bai&Lee: Beyond Linearization: On Quadratic and Higher-Order Approximation of Wide Neural Networks. ICLR2020.
  (二次のテイラー展開まで使う)

平均場解析

- ニューラルネットワークの最適化をパラメータの分布最適化としてみなす.

$$f(x) = \frac{1}{M}\sum_{j=1}^{M} a_j \eta(w_j^\top x) \xrightarrow{M\to\infty} \int a\eta(w^\top x)\rho(a,w)\mathrm{d}a\mathrm{d}w \qquad f_\rho(x)$$

➡ (a,w)に関する確率密度ρによる平均とみなせる:

fの最適化 ⇔ ρの最適化

$$\frac{\partial \rho_t}{\partial t} = -\nabla\cdot(v_t\rho_t)$$

連続方程式

Wasserstein勾配流

[Atsushi Nitanda and Taiji Suzuki: Stochastic Particle Gradient Descent for Infinite Ensembles. arXiv:1712.05438.]

$$v_t(a,w) = -\frac{1}{n}\sum_{i=1}^{n} \nabla_{(a,w)}(a\eta(w^\top x_i))\ell'(y_i, f_{\rho_t}(x_i))$$ (各粒子は勾配降下方向へ移動)

MMDとの関係

MMD: Maximum Mean Discrepancy

[Gretton et al., NIPS2006]

$f_{\nu^*}(x)$ ：真の関数

$$\mathbb{E}_X[(f_\nu(X) - f_{\nu^*}(X))^2]$$
$$= \mathbb{E}_X\left[\int\int a\eta(w^\top X)a'\eta(w'^\top X)(\mathrm{d}\nu(a,w)\mathrm{d}\nu(a',w') - 2\mathrm{d}\nu(a,w)\mathrm{d}\nu^*(a',w') + \mathrm{d}\nu^*(a,w)\mathrm{d}\nu^*(a',w'))\right]$$
$$= \int\int \underbrace{\mathbb{E}_X\left[a\eta(w^\top X)a'\eta(w'^\top X)\right]}(\mathrm{d}\nu(a,w)\mathrm{d}\nu(a',w') - 2\mathrm{d}\nu(a,w)\mathrm{d}\nu^*(a',w') + \mathrm{d}\nu^*(a,w)\mathrm{d}\nu^*(a',w'))$$

$$k((a,w),(a',w')) = \langle\phi_k(a,w),\phi_k(a',w')\rangle_{\mathcal{H}_k}$$

↑正定値カーネルになっている.

$$= \langle\mathbb{E}_\nu[\phi_k(a,w)],\mathbb{E}_\nu[\phi_k(a,w)]\rangle_{\mathcal{H}_k} - 2\langle\mathbb{E}_\nu[\phi_k(a,w)],\mathbb{E}_{\nu^*}[\phi_k(a,w)]\rangle_{\mathcal{H}_k}$$
$$+ \langle\mathbb{E}_{\nu^*}[\phi_k(a,w)],\mathbb{E}_{\nu^*}[\phi_k(a,w)]\rangle_{\mathcal{H}_k}$$
$$= \|\mathbb{E}_\nu[\phi_k] - \mathbb{E}_{\nu^*}[\phi_k]\|^2_{\mathcal{H}_k} \text{ : MMD}$$

L2距離最小化⇔ MMD最小化

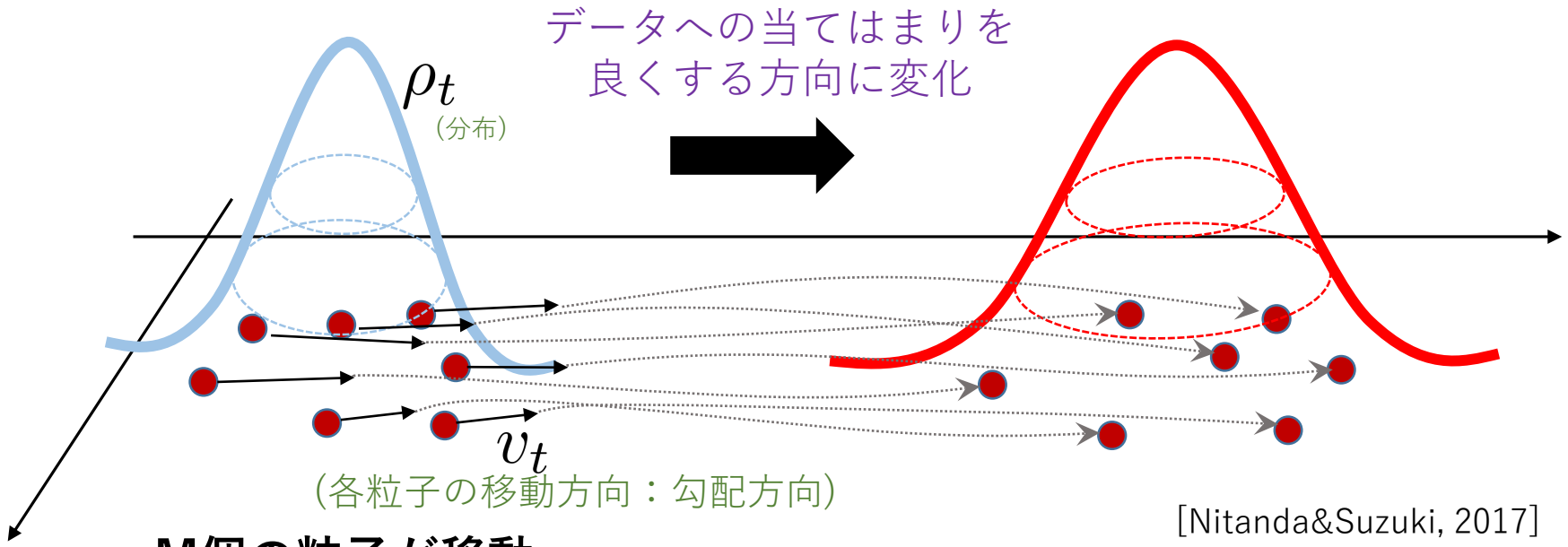
[Arbel et al. arXiv:1906.04370][Sonoda, arXiv:1902.00648]

粒子勾配降下法

$$f(x) = \frac{1}{M}\sum_{j=1}^{M} a_j \eta(w_j^\top x)$$

1つの粒子

- 各ニューロンのパラメータを一つの粒子とみなす.
- 各粒子が誤差を減らす方向に動くことで分布が最適化される.



[Nitanda&Suzuki, 2017]

$M \to \infty$の極限で，最適解への収束が成り立つ場合がある.

[Nitanda&Suzuki, 2017][Chizat&Bach, 2018][Chizat, 2019]

ノイズありのダイナミクス: McKean-Vlasov過程

[Mei, Montanari&Nguyen, 2018]

Wasserstein距離について

μ, ν:距離空間$(\mathcal{X}, c)$上の確率測度(通常$\mathcal{X}$はPoland空間)

$$W_p(\mu,\nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} c(x,y)^p \mathrm{d}\pi(x,y) \right)^{1/p}$$

$\Pi(\mu,\nu)$: 周辺分布がμ,νである$\mathcal{X}\times\mathcal{X}$上の同時分布の集合

周辺分布を固定した同時分布の中で最小化

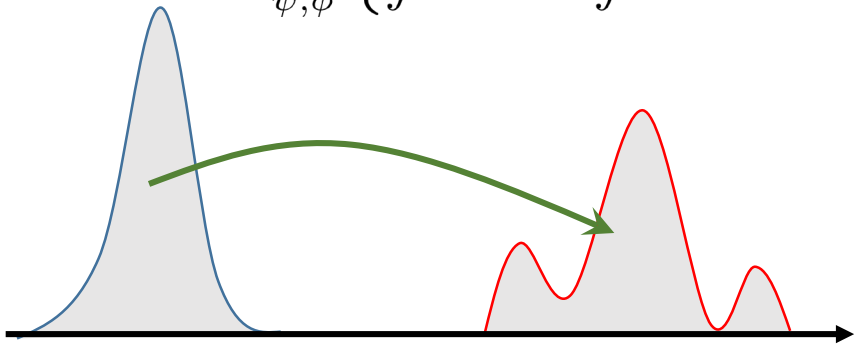
$$(\mathcal{X} = \mathbb{R}^d:\ c(x,y) = \|x-y\|)$$

- 分布のサポートがずれていてもwell-defined
- 底空間の距離が反映されている

  ※KL-divergenceは距離が反映されない.

(双対表現: Kantorovich双対)

$$\inf_{\pi\in\Pi(\mu,\nu)} \int c(x,y)^p \mathrm{d}\pi(x,y) = \sup_{\psi,\phi} \left\{ \int \psi \mathrm{d}\mu + \int \phi \mathrm{d}\nu \mid \psi(x)+\phi(y) \le c(x,y)^p \right\}$$



「輸送距離」とも言われる

W_2距離と粒子勾配降下法の関係

W_2距離による近接点アルゴリズムを考える：

（δは十分小さいとする）

$$f_\nu(x) = \int h(w,x)\mathrm{d}\nu(w)$$

$$\begin{aligned}
&L(\nu) + \frac{W_2^2(\mu,\nu)}{2\delta} \\
&= \mathbb{E}_X\left[\ell\left(\int h(w,X)\mathrm{d}\nu(w)\right)\right] + \frac{W_2^2(\mu,\nu)}{2\delta} \\
&\simeq \Big\langle \underbrace{\ell'\left(\int h(w,X)\mathrm{d}\mu(w)\right)}_{=:\Delta_\mu(X)}, \int h(v,X)\mathrm{d}(\nu-\mu)(v)\Big\rangle_{L^2(P)} + \inf_{\pi\in\Pi(\mu,\nu)}\int \frac{\|w-v\|^2}{2\delta}\mathrm{d}\pi(w,v) \\
&\simeq \inf_{\pi\in\Pi(\mu,\nu)}\int \underbrace{\left(\left\langle \Delta_\mu, h(v,\cdot)\right\rangle_{L^2(P)} + \frac{\|w-v\|^2}{2\delta}\right)}\mathrm{d}\pi(w,v) + \text{const.}
\end{aligned}$$

νについて最小化→各wごとにνの条件付分布を最小化 →Dirac測度 （∵局所的に強凸）

$$\begin{aligned}
v &= \arg\min_{v'}\left\{\langle\Delta_\mu, h(v',\cdot)\rangle_{L^2(P)} + \frac{\|w-v'\|^2}{2\delta}\right\} \\
&\simeq w - \delta\nabla_w\langle h(w,\cdot), \ell'(f_\mu)\rangle
\end{aligned}$$

：最急降下法

- 各粒子ごとにみると単純な最急降下法.
- 粒子勾配降下法はW_2距離を近接項とした近接点アルゴリズムの一次近似 → $\delta\to 0$の極限（連続時間）：**Wasserstein gradient flow**

連続の方程式と勾配流

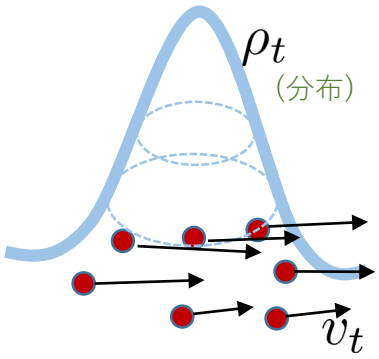
「連続の方程式」 $\frac{\partial \rho_t}{\partial t} = -\nabla \cdot (v_t \rho_t)$ の意味

$$\frac{\mathrm{d}}{\mathrm{d}t} \int f(w) \mathrm{d}\rho_t(w) = \int (\nabla f(w))^\top v_t(w) \mathrm{d}\rho_t(w)$$

$(= -\int f(w) \mathrm{d}[\nabla \cdot (v_t \rho_t)])$ $(\forall f$: コンパクトサポート，C^∞-級$)$

- 今，ρ_tは写像$T_t: R^d \to R^d$によるρ_0の押し出しであるとする：$\rho_t = T_{t\#}\rho_0$.
  つまり，$w \sim \rho_0$に対する$T_t(w)$の分布がρ_tであるとする.
- 写像T_tを生成するベクトル場を$\frac{\mathrm{d}T_t}{\mathrm{d}t}(w) = v_t(T_t(w))$とする.

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} \int f(w) \mathrm{d}\rho_t(w) &= \frac{\mathrm{d}}{\mathrm{d}t} \int f(T_t(w)) \mathrm{d}\rho_0(w) \\
&= \int \nabla f(T_t(w))^\top \frac{\mathrm{d}T_t(w)}{\mathrm{d}t} \mathrm{d}\rho_0(w) \\
&= \int \nabla f(T_t(w))^\top v_t(T_t(w)) \mathrm{d}\rho_0(w) \\
&= \int \nabla f(w)^\top v_t(w) \mathrm{d}\rho_t(w). \quad \text{(連続の方程式)}
\end{aligned}$$



$w_t = T_t(w)$ に対し，$v_t(w_t) = -\nabla_w \langle h(w, \cdot), \ell'(f_{\rho_t}) \rangle|_{w=w_t}$ としたのが前ページの更新式.

接ベクトル

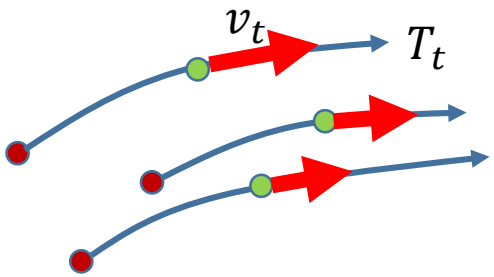
定理

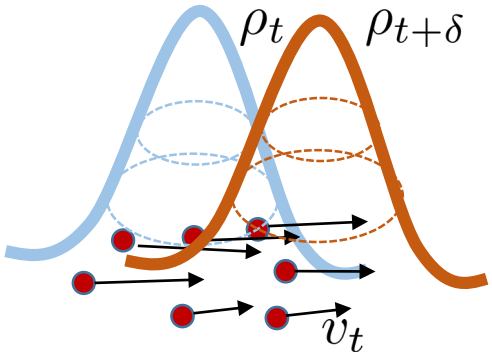
- $\rho_t = T_{t\#}\rho_0$
- $\frac{\mathrm{d}T_t}{\mathrm{d}t}(w) = v_t\big(T_t(w)\big)$
- あるϕ_tを用いて$v_t = \nabla\phi_t$と書けるとする.

この時，以下が成り立つ:

$$\lim_{\delta\to 0}\frac{W_2(\rho_{t+\delta}, (\mathrm{id}+\delta v_t)_{\#}\rho_t)}{\delta} = 0$$

詳細は以下を参照:
Ambrosio, Gigli, and Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Lectures in Mathematics. ETH Zürich. Birkhäuser Basel, 2008.





輸送写像

Brenierの定理

ρ_0, ρ_1が確率密度関数を持つ時，以下が成り立つ:

$$W_2^2(\rho_0, \rho_1) = \inf_{T:T_{\#}\rho_0=\rho_1} \mathbb{E}_{X\sim\rho_0}[\|X - T(X)\|^2]$$

- Infを達成する写像T^*が存在する.
- しかも，ある凸関数ψが存在して$T^*(x) \in \partial\psi(x)$と書ける.
- このT^*を最適輸送写像という.

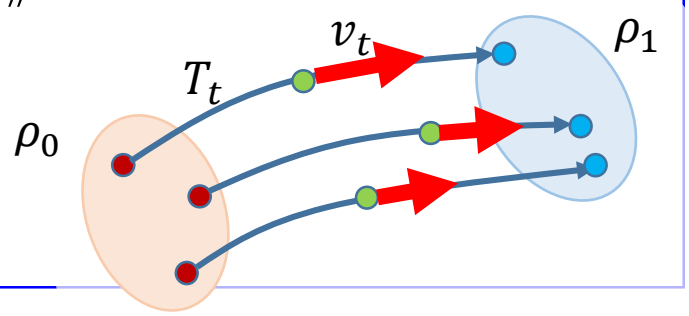
Benamou-Brenier formula (連続の方程式とW_2距離の関係):

同条件のもと

$$W_2^2(\rho_0, \rho_1) = \inf_{\{v_t\}_t} \int_0^1 \|v_t\|^2_{L_2(\rho_t)} \mathrm{d}t$$

ただし，infはρ_0からρ_1へ連続の方程式で“繋ぐ”全ての速度ベクトル場v_tに関して取る.

- $\rho_t = T_{t\#}\rho_0$
- $\frac{\mathrm{d}T_t}{\mathrm{d}t}(w) = v_t(T_t(w))$



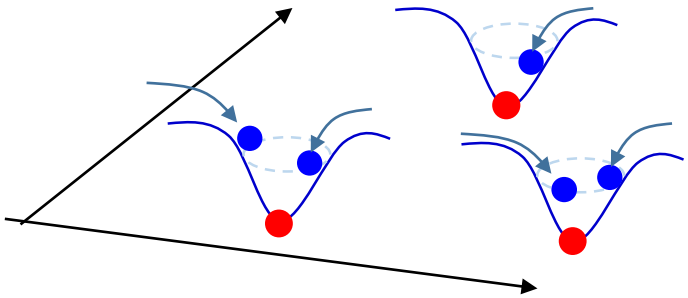
- Wasserstein勾配流はW_2距離を用いた近接点アルゴリズムで特徴づけられることが分かった.
- 目的関数がW2距離に関する凸性（displacement convexity）が成り立つなら，大域的最適解への収束が示せる（エントロピーなど）：

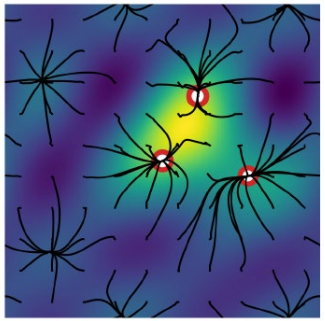
$$W_2(\rho_t, \rho^*) \leq e^{-\lambda t} W_2(\rho_0, \rho^*).$$

- しかし，NNの最適化では凸性は成り立たない.

  そのため，大域収束を示すことが難しい.
- もっとも，局所的には凸性が成り立ちうる.

例：スパースな最適解 [Chizat, 2019]





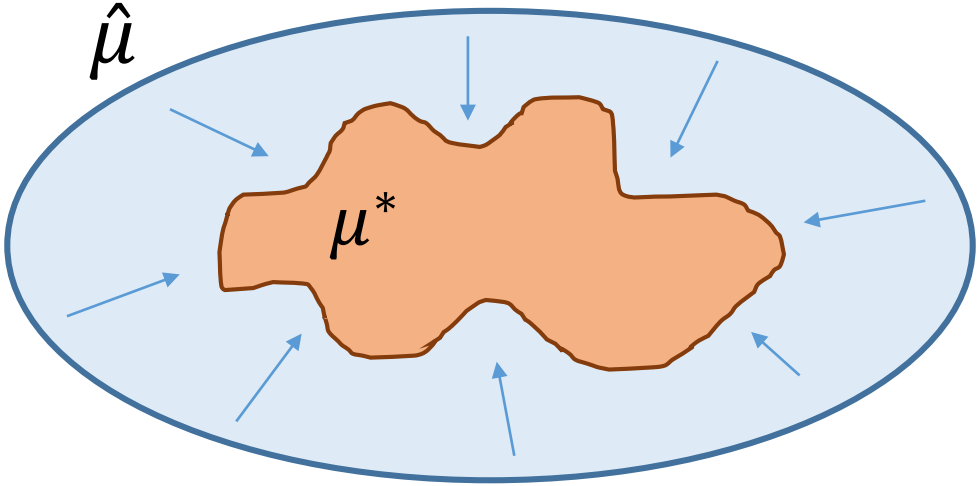
局所最適性条件

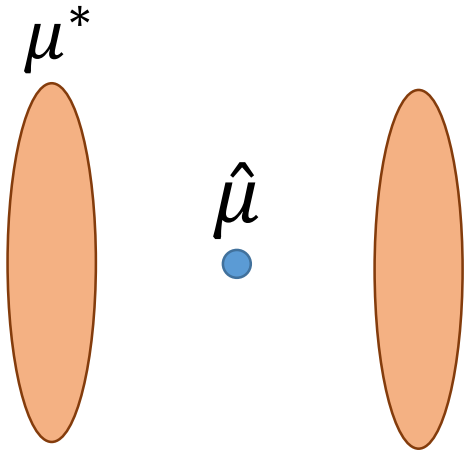
定理 (Nitanda&Suzuki, 2017)

ある解$\hat{\mu}$がコンパクトな台の確率密度関数を持つとする.
この時, あるμ^* s.t. $L(\mu^*) < L(\hat{\mu})$が存在して

- $\mathrm{supp}(\mu^*) \subseteq \mathrm{supp}(\hat{\mu})$かつ$\mu^*$は確率密度を持つ, or
- μ^*は密度を持たず$\mathrm{supp}(\mu^*)$は$\mathrm{supp}(\hat{\mu})$に内部に含まれる,

が満たされるとき, 降下方向が存在して粒子降下法によって目的関数値を減らすことができる.





このような場合は降下方向は無い

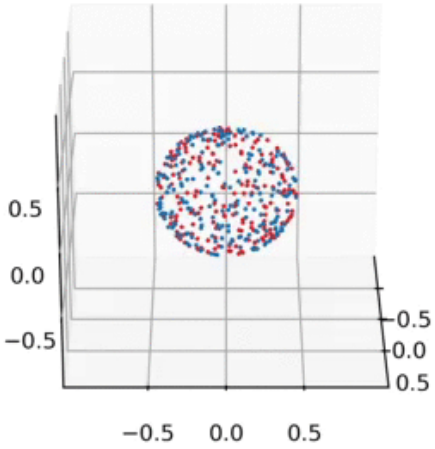
平均場解析と陰的正則化

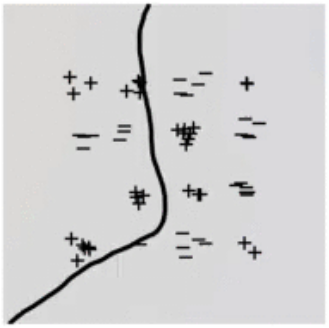
二値判別をexp-損失を用いて解く(ラベルノイズなしとする):

$$\min_{\rho} \sum_{i=1}^{n} \exp\left(-y_i f_\rho(x)\right) \qquad \text{ただし} \qquad f_\rho(x) = \int \eta(w^\top x) \mathrm{d}\rho(w)$$

符号付測度の中で最適化

平均場解析の設定で最適化する.
初期値が小さいので判別に必要なニューロンだけが「生えてくる」.





[Chizat&Bach:Implicit Bias of Gradient Descent for Wide Two-layer Neural Networks Trained with the Logistic Loss. COLT2020.]

→ **スパースな解：<u>陰的正則化</u>**

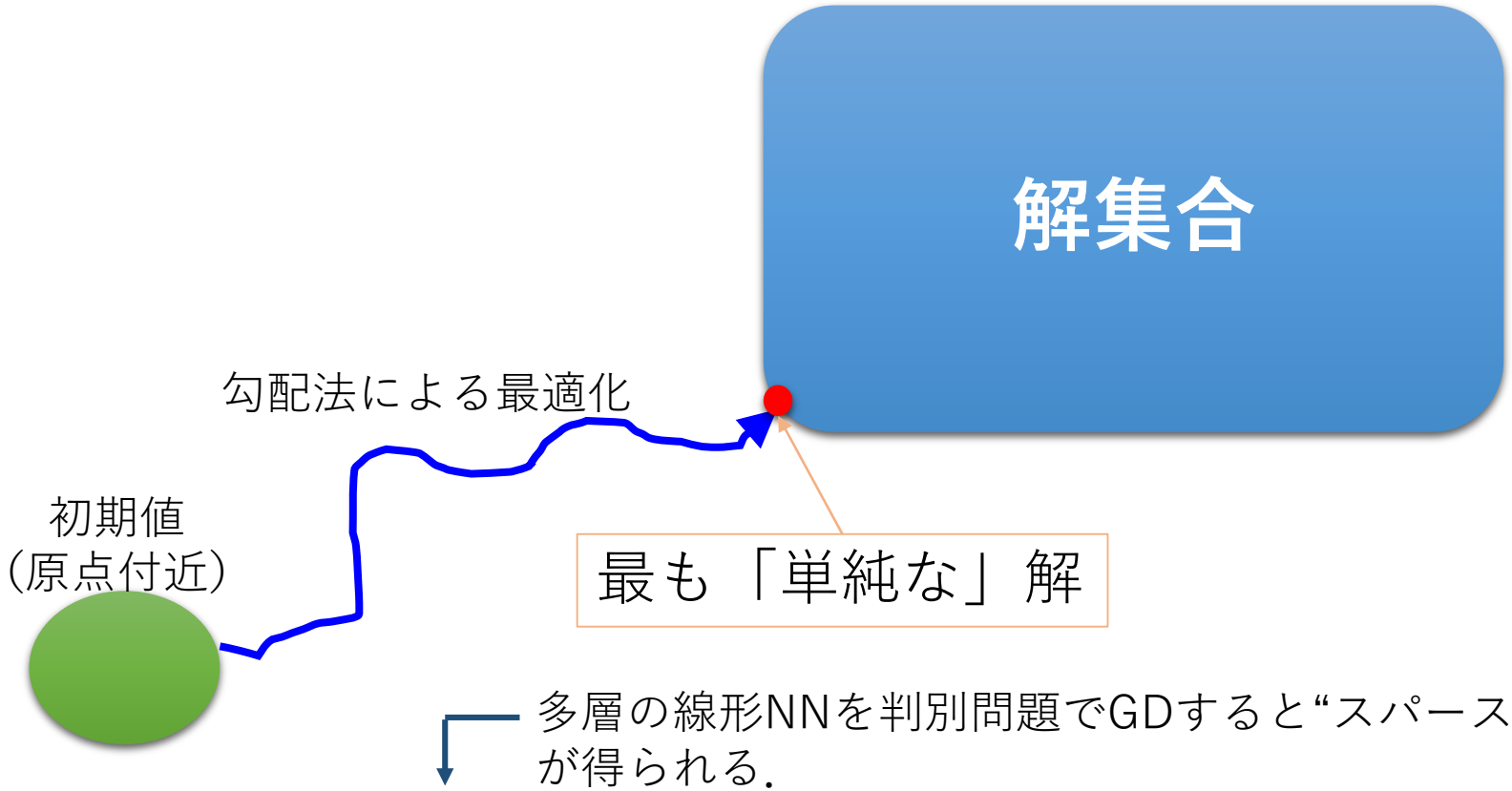
最適化の結果として「単純な」解が求まってしまう.

判別平面はL1-正則化解マージン最大化元に収束する:

$$\max_{\rho:\|\rho\|_{\mathcal{F}_1}} \min_{i\in\{1,\ldots,n\}} y_i f_\rho(x_i) \qquad \|\rho\|_{\mathcal{F}_1} = |\rho|(\mathbb{R}^d)$$

勾配法と陰的正則化（再掲）

- 小さな初期値から勾配法を始めるとノルム最小化点に収束しやすい→陰的正則化



[Gunasekar et al.: Implicit Regularization in Matrix Factorization, NIPS2017]
[Soudry et al.: The implicit bias of gradient descent on separable data. JMLR2018]
[Gunasekar et al.: Implicit Bias of Gradient Descent on Linear Convolutional Networks, NIPS2018]
[Moroshko et al.: Implicit Bias in Deep Linear Classification: Initialization Scale vs Training Accuracy, arXiv:2007.06738]

各regimeにおける陰的正則化

各regimeにおける陰的正則化の種類

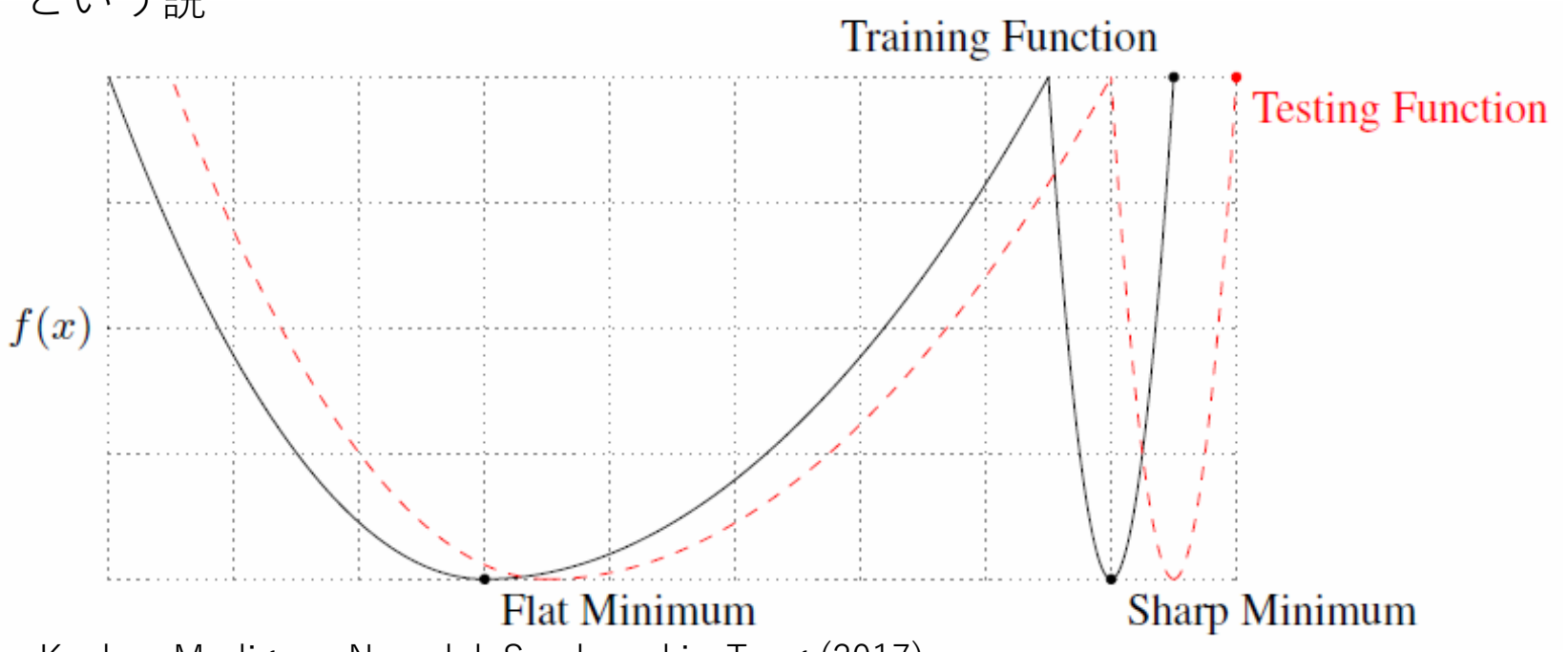
Regime	対応する正則化
NTK, カーネル法 with early stopping	L2-正則化
平均場理論	L1-正則化

- ニューラルネットワークの学習では様々な「**陽的正則化**」を用いる：バッチノーマリゼーション，Dropout，Weight decay，MixUp，…
- 一方で，深層学習の構造が自動的に生み出す「**陰的正則化**」も強く効いていると考えられる.
  → オーバーパラメタライズしても過学習しない.

ノイズあり勾配法と大域的最適性

Sharp minima vs flat minima

SGDは「フラットな局所最適解」に落ちやすい→良い汎化性能を示す
という説



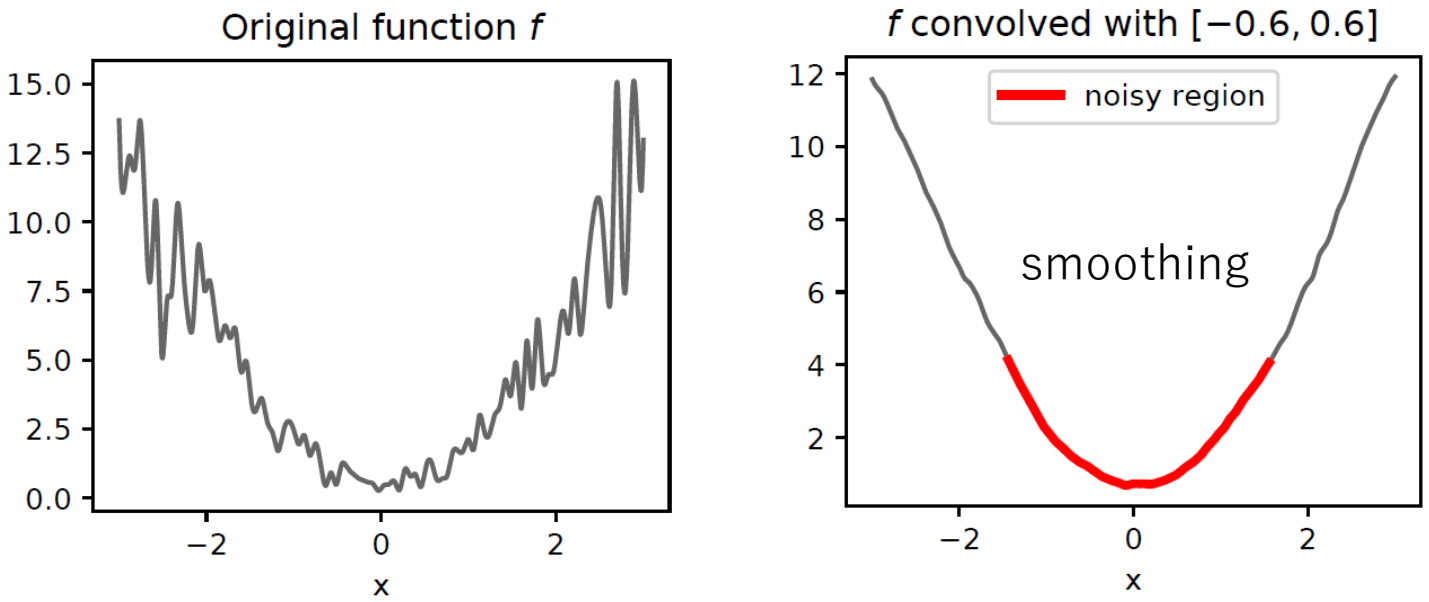
Keskar, Mudigere, Nocedal, Smelyanskiy, Tang (2017):
On large-batch training for deep learning: generalization gap and sharp minima.

$$\theta_t = \theta_{t-1} - \alpha_b \underbrace{\left(\frac{1}{b} \sum_{j=1}^{b} \nabla_\theta \ell(z_{i_j}; \theta) \right)}_{\cong \text{正規分布}}$$

→ ランダムウォークはフラットな領域にとどまりやすい

- 「フラット」という概念は座標系の取り方によるから意味がないという批判. (Dinh et al., 2017)
- PAC-Bayesによる解析 (Dziugaite, Roy, 2017)

ノイズによる平滑化効果



[Kleinberg, Li, and Yuan, ICML2018]

確率的勾配を用いる ⇒ 解にノイズを乗せている⇒ 目的関数の平滑化

$$x_t = x_{t-1} - \eta(\nabla L(x_{t-1}) + \xi_t) \qquad (y_t = x_t + \eta\xi_t)$$

$$\Rightarrow y_t = y_{t-1} - \eta\xi_{t-1} - \eta\nabla L(y_{t-1} - \eta\xi_{t-1})$$

$$\Rightarrow \mathbb{E}_{\xi_{t-1}}[y_t] = y_{t-1} - \eta\nabla\mathbb{E}_{\xi_{t-1}}[L(y_{t-1} - \eta\xi_{t-1})]$$

ノイズを加えて平滑化した目的関数 $\bar{L}(y_t) = \mathbb{E}_{\xi_t}[L(y_t - \eta\xi_t)]$ を最適化.

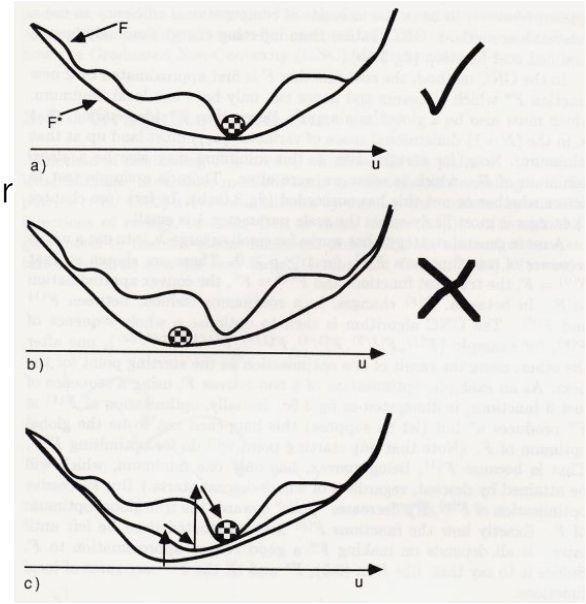
関連研究: Graduated optimization

- Graduated non-convexity

  Blake and Zisserman: *Visual reconstruction*, volume 2. MIT press Cambridge, 1987.

- Gaussian kernelとの畳み込み

Z. Wu. The effective energy transformation scheme as a special continuation approach to global optimization with application to molecular conformation. SIAM Journal on Optimization, 6(3):748-768, 1996.

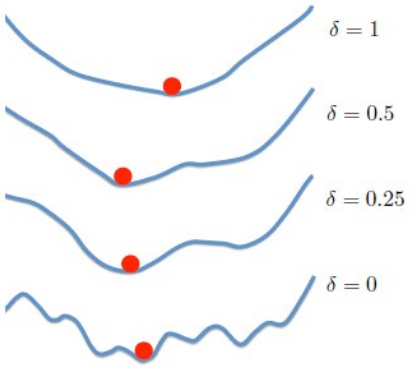


- Graduated optimization

Hazan, Levy, and Shalev-Shwartz: On graduated optimization for stochastic non-convex problems. *International conference on machine learning*, pp. 1833-1841, 2016.

σ-nice性の導入．多項式オーダーでの収束.

$$\hat{L}_\delta(x) = \mathrm{E}_{u \sim U(\mathrm{B}(\mathrm{R}^d))}[L(x + \delta u)]$$



Survey:
Mobahi and Fisher III. On the link between gaussian homotopy continuation and convex envelopes. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 43-56, 2015.

GLD/SGLD

- Stochastic Gradient Langevin Dynamics (SGLD)

$$\min_{x\in\mathbb{R}^d} L(x) = \min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell_i(x) \qquad \textbf{(非凸)}$$

β: 逆温度

$$\mathrm{d}X_t = -\nabla L(X_t)\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}B_t \qquad \text{(勾配Langevin動力学)}$$

定常分布： $\pi \propto \exp(-\beta L(X))$

離散化

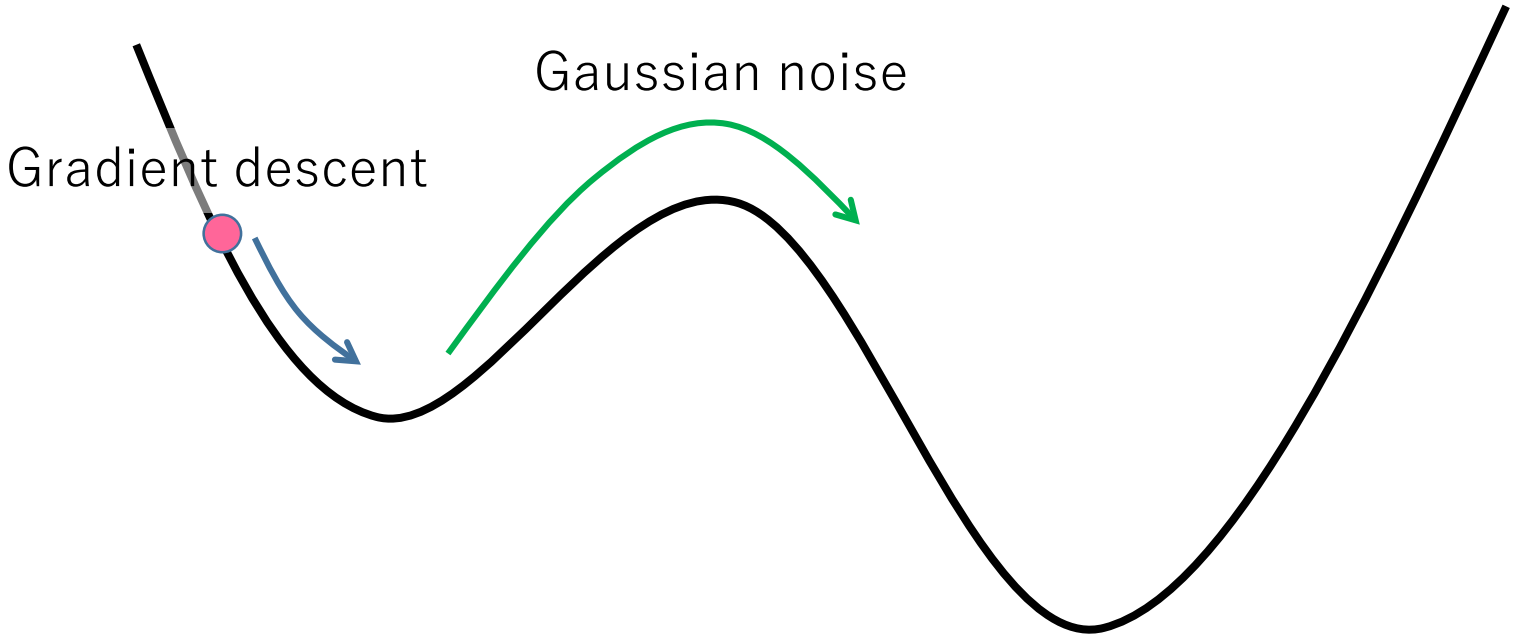
[Gelfand and Mitter (1991); Borkar and Mitter (1999); Welling and Teh (2011)]

GLD: $X_{t+1} = X_t - \eta\nabla L(X_t) + \sqrt{2\eta\beta^{-1}}\xi_t$ (Euler-Maruyama近似)

$\xi_t \sim \mathrm{N}(0, I)$

SGLD: $X_{t+1} = X_t - \eta\frac{1}{|I_B|}\sum_{i\in I_B}\nabla\ell_i(X_t) + \sqrt{2\eta\beta^{-1}}\xi_t$

確率的勾配


Gaussian noise
Gradient descent

収束定理（有限次元）

- f_i：有界，Lipschitz連続，滑らかな勾配

$$\|\ell_i\|_\infty \le A,\ \|\nabla \ell_i\|_\infty \le B,\ \|\nabla \ell_i(x) - \nabla \ell_i(y)\| \le M\|x-y\|$$

- **散逸条件:**

$$\langle \nabla L, w \rangle \ge m\|w\|^2 - b \quad (\forall w \in \mathbb{R}^d)$$

(+ その他細かい条件)

Thm [Raginsky, Rakhlin and Telgarsky, COLT2017]

$$\mathrm{E}[L(X_k)] - L(X^*) \le \tilde{\mathrm{O}}\left((\beta+d)k\eta^{5/4} + \frac{\beta+d}{\sqrt{\lambda^*}} \exp\left(-\tilde{\Omega}\left(\frac{\lambda^* k\eta}{\beta(d+\beta)}\right)\right) + \frac{d\log(\beta+1)}{\beta}\right)$$

- λ_* はスペクトルギャップと言われる量.
  → 次元dや逆温度パラメータβに対して指数関数的に依存.
- 逆温度パラメータが十分大きくて，更新を十分な回数回せば最適解付近に近づける.
- Xu et al. (NeurIPS2018) は収束レートを改善しているが，証明にいくつかの間違いあり.

散逸条件

対数Sobolev不等式

$\pi_\infty(\mathrm{d}x) \propto \exp(-\beta L(x))\mathrm{d}x$ ：連続時間ダイナミクスの定常分布

1. 散逸条件 (→Poincareの不等式)
2. 平滑性

➡ **対数Sobolev不等式**

$$\mathrm{d}\nu = f\ \mathrm{d}\pi_\infty \quad (\text{probability})$$

$$\int f\log(f)\mathrm{d}\pi_\infty \le 2c_{\mathrm{LS}}\int \frac{\|\nabla f\|^2}{f}\mathrm{d}\pi_\infty \quad (D(\nu||\pi_\infty) \le 2c_{\mathrm{LS}}I(\nu||\pi_\infty))$$

➡ **Geometric ergodicity**

ρ_t: X_tの周辺分布

$$D(\rho_t||\pi_\infty) \le \exp(-2t/c_{\mathrm{LS}})D(\rho_0||\pi_\infty)$$

定常分布へ線形収束

[Bakry, Gentil, and Ledoux: *Analysis and Geometry of Markov Diffusion Operators*. Springer, 2014. Th. 5.2.1]

連続の方程式再び

- 勾配ランジュバン動力学に対応する連続の方程式

$$\partial_t \rho_t = \nabla \cdot (\rho_t \nabla \log(\rho_t / \pi_\infty))$$

- 勾配ランジュバン動力学は相対エントロピー (KL-ダイバージェンス) をWasserstein勾配流で最適化していることに対応:

$$D(\rho||\pi_\infty) = \int \log\left(\frac{\mathrm{d}\rho}{\mathrm{d}\pi_\infty}\right) \mathrm{d}\rho$$

Remark:
通常の（相対でない）エントロピーはW_2-距離に関して凸 (displacement convexity)．つまり，W_2-距離に関する測地線上で凸関数になる．

無限次元への拡張

[Muzellec, Sato, Massias, Suzuki, arXiv:2003.00306][Suzuki, arXiv:2007.05824]

$$\min_{x \in \mathcal{H}} \quad L(x)$$

$\mathcal{H}$: Hilbert space

正則化

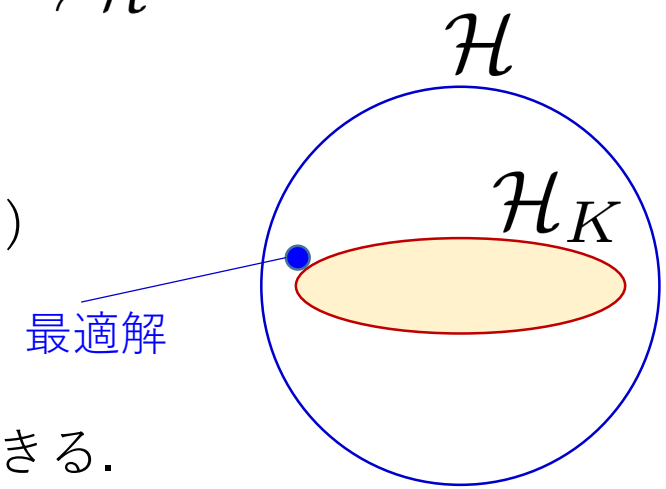
$$\min_{x \in \mathcal{H}} \quad L(x) + \lambda \|x\|^2_{\mathcal{H}_K}$$

$\mathcal{H}_K$:"smaller" Hilber space

$\mathcal{H}_K \hookrightarrow \mathcal{H}$

Ex.

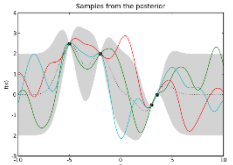
- $\mathcal{H}$: $L^2(\rho)$
- $\mathcal{H}_K$: 再生核ヒルベルト空間 (e.g. Sobolev空間)



暗黙の仮定: 大域的最適解は$\mathcal{H}_K$で十分に近似できる.

E.g., Bayesian optimization on infinite dimensional space

[Zimmermann and Toussaint. Bayesian functional optimization. AAAI, 2018]
[Vellanki, Rana, Gupta, de Celis Leal, Sutti, Height, and Venkatesh: Bayesian functional optimisation with shape prior. AAAI, 2019]



例: NNの学習

- 2層ニューラルネットワーク

Idea: 分布の学習 → **輸送写像の学習**

$$W : \mathbb{R}^d \to \mathbb{R}^d \qquad W \in L_2(\rho_0)$$

$$f_W(x) := \int_{\mathbb{R}^d} a(w)\sigma(W(w)^\top x)\mathrm{d}\rho_0(w)$$

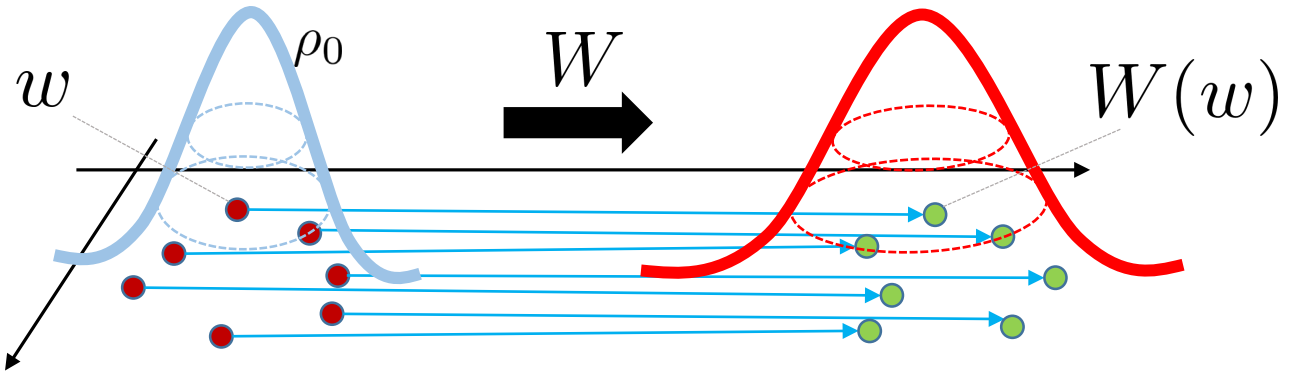
$$= \int_{\mathbb{R}^d} a(w)\sigma(w^\top x)\mathrm{d}W\sharp\rho_0(w)$$

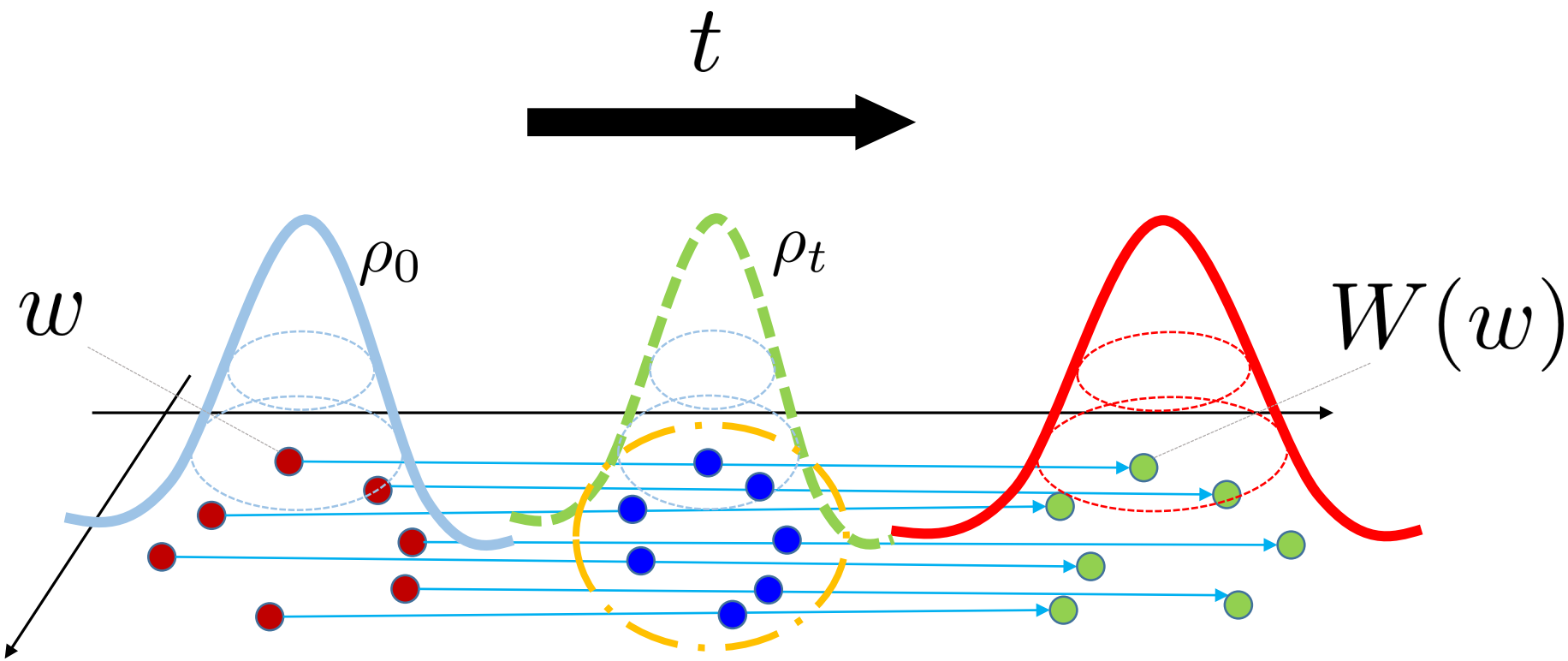
“Lift”

$$f_\rho(x) = \int_{\mathbb{R}^d} a(w)\sigma(w^\top x)\mathrm{d}\rho(w)$$

以前の表記

$$\min_{\rho} \; L(f_\rho) \longrightarrow \min_{W \in \mathcal{H}} \; L(f_{W\sharp\rho_0})$$



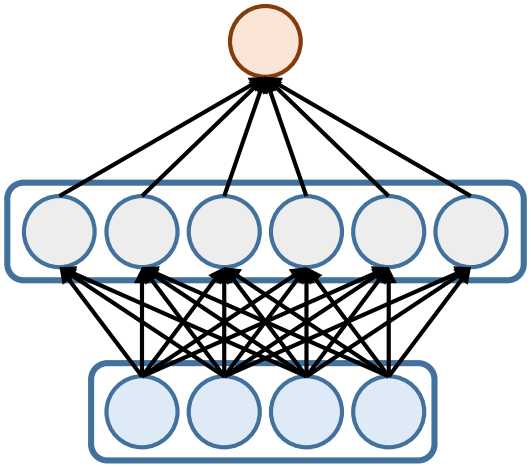

t
w
ρ_0
ρ_t
$W(w)$
$W_0 = \mathbb{I}$
W_t
W_∞

2層NNの学習: 直接表現

$$L(W) = \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \ell_i(f_W(x_i)) + \frac{\lambda_0}{2} \|W\|_{\mathrm{F}}^2$$

$$f_W(x) = \sum_{j=1}^{\infty} a_j \eta(w_j^\top x)$$

- $a_j \leq j^{-\gamma}$ for $\gamma > 1/2$
- η is a smooth activation, e.g., sigmoid.



NTKと違い，a_jはデータサイズにも横幅にも依存させずスケールを固定できる．
(TNKは$a_j = 1/\sqrt{M}$とする)

無限次元ランジュバン動力学

$$x = \sum_{j=1}^{\infty} x_j f_j \in \mathcal{H}$$

$$\min_{x \in \mathcal{H}} L(x) \Rightarrow \min_{x \in \mathcal{H}} \left\{ L(x) + \frac{\lambda}{2} \|x\|_{\mathcal{H}_K}^2 \right\}$$

$\mathcal{H}_K$: RKHS with kernel K.

$\mathcal{H}_K \hookrightarrow \mathcal{H}$

$$\mathrm{d}X_t = -\nabla \left(L(X_t) + \frac{\lambda}{2} \|X_t\|_{\mathcal{H}_K}^2 \right) \mathrm{d}t + \sqrt{\frac{2}{\beta}} \mathrm{d}\xi_t$$

ノルム: For $x = \sum_{j=1}^{\infty} x_j f_j \in \mathcal{H}$, we let $\|x\|_{\mathcal{H}_K}^2 = \sum_{j=1}^{\infty} \mu_j^{-1} x_j^2$ where $\mu_j \sim j^{-2}$.

Cylindrical Brownian motion: $\xi_t = \sum_{j=1}^{\infty} \xi_{j,t} f_j$

時間離散化:

$$X_{n+1} = S_\eta \left(X_n - \eta \nabla L(X_n) + \sqrt{2 \frac{\eta}{\beta}} \xi_n \right) \qquad \left(S_\eta := (I + \eta \lambda A)^{-1} \right)$$

(準陰的Eulerスキーム)

$$A = \mathrm{diag}(\mu_1^{-1}, \mu_2^{-1}, \dots)$$

$$\xi_n = \sum_{j=1}^{\infty} \gamma_{n,j} f_j \text{ where } \gamma_{n,j} \sim N(0,1) \text{ (i.i.d.).}$$

定常分布

$$\mathrm{d}X_t = -\nabla\left(L(X_t) + \frac{\lambda}{2}\|X_t\|^2_{\mathcal{H}_K}\right)\mathrm{d}t + \sqrt{\frac{2}{\beta}}\mathrm{d}\xi_t$$

$$\frac{\mathrm{d}\pi_\infty}{\mathrm{d}\mu_*}(x) \;\propto\; \exp\left(-\beta L(x)\right)$$

$\mu_* = N(0, C)$　(Hilbert空間上のガウス過程)

$$\text{where } C = (\beta\lambda)^{-1}\mathrm{diag}(\mu_0, \mu_1, \dots).$$

$$\pi_\infty(x) \;\propto\; \exp\left(-\beta L(x) - \frac{1}{2}x^\top C^{-1}x\right)$$　と解釈しても良い.

（無限次元）勾配ランジュバン動力学の定常分布は
ガウス過程事前分布を用いたベイズ事後分布に対応する.
→ 過学習を防ぎ汎化する

[Suzuki, arXiv:2007.05824]

無限次元の設定

ヒルベルト空間

$$\mathcal{H} = \left\{ \sum_{k=0}^{\infty} \alpha_k f_k \mid \sum_{k=0}^{\infty} \alpha_k^2 < \infty \right\}$$

$$\langle x, y \rangle = \sum_{k=0}^{\infty} \alpha_k \beta_k \quad \text{for } x = \sum_k \alpha_k f_k,\ y = \sum_k \beta_k f_k.$$

RKHS構造

$$\mathcal{H}_K = \left\{ \sum_{k=0}^{\infty} \alpha_k f_k \mid \sum_{k=0}^{\infty} \alpha_k^2 / \mu_k < \infty \right\}$$

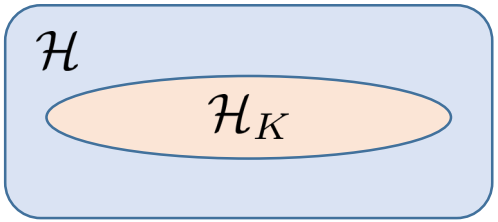
$$\langle x, y \rangle_{\mathcal{H}_K} = \sum_{k=0}^{\infty} \alpha_k \beta_k / \mu_k \quad \text{for } x = \sum_k \alpha_k f_k,\ y = \sum_k \beta_k f_k.$$

仮定（固有値の減少）

$$\mu_k \simeq k^{-2}$$

（あまり本質的ではない．$\mu_k \sim k^{-p}$ $(p > 1)$ としても良い．）

$$\min_{x \in \mathcal{H}} L(x) = \min_{x \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell_i(x) + \left(\frac{\lambda_0}{2} \|x\|^2 \right)$$



Assumption (1)

- It either holds:
  - (Strict Dissipativity) $\lambda > M\mu_0$, or (強):強凸
  - (Bounded gradients) $\|\nabla L(\cdot)\| \leq B$, for $B > 0$. (弱)

散逸条件:

For $A = -\frac{\lambda}{2}\nabla\|\cdot\|^2_{\mathcal{H}_K}$

$$\langle Ax - \nabla L(x), x\rangle \leq -m\|x\|^2 + c.$$

$$L(x) + \frac{\lambda}{2}\|x\|^2_{\mathcal{H}_K}$$

Assumption (2)

- Smoothness:

$$\|\nabla L(x) - \nabla L(y)\| \le M\|x - y\|$$

- Strong smoothness condition:

For $\alpha \in (1/4, 1)$, （これが無い場合はレートが遅くなる）

$$\|\nabla L(x) - \nabla L(y)\|_{-\alpha} \le M\|x - y\|$$

where $\|x\|_\varepsilon = \left(\sum_{k\ge 0}(\mu_k)^{2\varepsilon}|\langle x, f_k\rangle|^2\right)^{1/2}$.

(This is not standard, but, is satisfied in the previous examples)

- Third order smoothness:

Let $L_N = L(P_N x)$. There exists $\alpha' \in [0, 1)$ such that

$$\|D^3 L_N(x)\cdot(h,k)\|_{\alpha'} \le C_{\alpha'}\|h\|_0\|k\|_0,$$
$$\|D^3 L_N(x)\cdot(h,k)\|_0 \le C_{\alpha'}\|h\|_{-\alpha'}\|k\|_0.$$

誤差の解析

π_∞:定常分布

Thm (informal) [Muzellec, Sato, Massias, Suzuki, 2020]

上記の条件のもと，次が成り立つ：

$$L(X_n) - \int L(x)\mathrm{d}\pi_\infty(x) \lesssim \exp\left(-\Lambda_\eta^* n\eta\right) + \frac{c_\beta}{\Lambda_0^*}\eta^{1/2-\kappa}$$

(geometric ergodicity + time discretization)

ただし $\kappa > 0$ は任意の正の実数, $c_\beta = \sqrt{\beta}$ (有界な勾配), $c_\beta = 1$ (強散逸条件).

Remark: $$\int L(x)\mathrm{d}\pi_\infty(x) \simeq L(\tilde{x}) \quad \text{for} \quad \tilde{x} := \operatorname*{arg\,min}_{x\in\mathcal{H}} \left\{ L(x) + \frac{\lambda}{2}\|x\|_{\mathcal{H}_K}^2 \right\}$$

証明は以下の論文のテクニックを援用: Brehier 2014; Brehier&Kopec 2016; Mattingly et al., 2002; Goldys&Maslowski, 2006.

誤差の解析 (2)

π_∞: 定常分布

$$x^* := \arg\min_{x \in \mathcal{H}} L(x) \qquad \tilde{x} := \arg\min_{x \in \mathcal{H}} \left\{ L(x) + \frac{\lambda}{2} \|x\|^2_{\mathcal{H}_K} \right\}$$

Thm [Muzellec, Sato, Massias, Suzuki, arXiv:2003.00306 (2020)]

上記の条件のもと，次が成り立つ：

$$L(X_n) - L(x^*) \lesssim \exp\left(-\Lambda^*_\eta n\eta\right) + \frac{c_\beta}{\Lambda^*_0}\eta^{1/2-\kappa} \quad \text{(geometric ergodicity + time discretization)}$$

$$+ \frac{1}{\beta}\left(\sqrt{\frac{1}{\lambda}} + 1\right) + \lambda\left(\frac{\|\tilde{x}\|_{\mathcal{H}_K}}{\sqrt{\beta}} + \|\tilde{x}\|^2_{\mathcal{H}_K}\right)$$

$$+ L(\tilde{x}) - L(x^*) \quad \text{(bias of invariant measure)}$$

ただし $\kappa > 0$ は任意の正の実数, $c_\beta = \sqrt{\beta}$ (有界な勾配), $c_\beta = 1$ (強散逸条件).

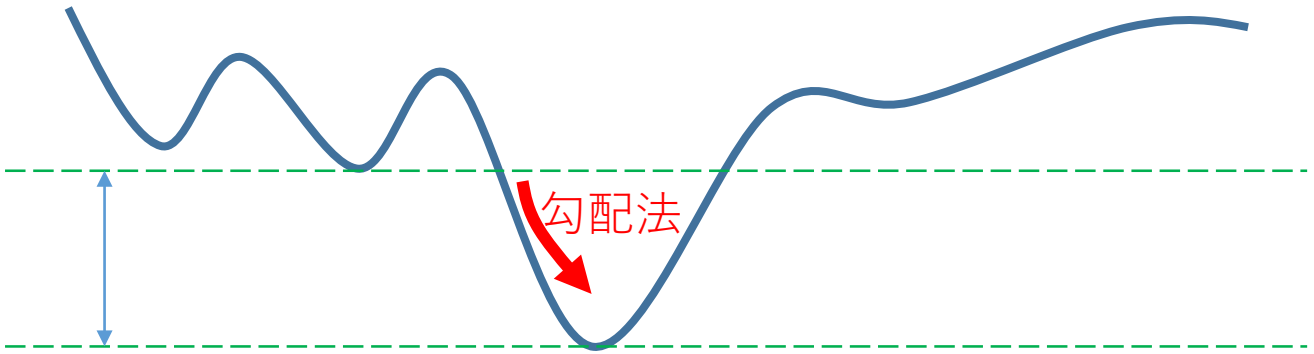
Λ^*_η: スペクトルギャップ，βに対して指数的依存がある.

証明は以下の論文のテクニックを援用: Brehier 2014; Brehier&Kopec 2016; Mattingly et al., 2002; Goldys&Maslowski, 2006.

- 深層学習の最適化への応用と汎化誤差解析：Suzuki, arXiv:2007.05824.

ノイズのコントロール

- 大域的最適解を得るためには$\beta \to \infty$が必要.
- スペクトルギャップはβに指数的に依存.
- 大域的最適解まわりで局所的に凸になっていて，離れた場所より目的関数値が真に小さければ途中で勾配法に切り替えても良い.
- 例えば2層NNでは訓練誤差の形状が局所的に強凸になることがある [Li and Yuan, 2017][Chizat, 2019]
  (各ニューロンが適度にばらけている場合はそうなる)



誤差の分解

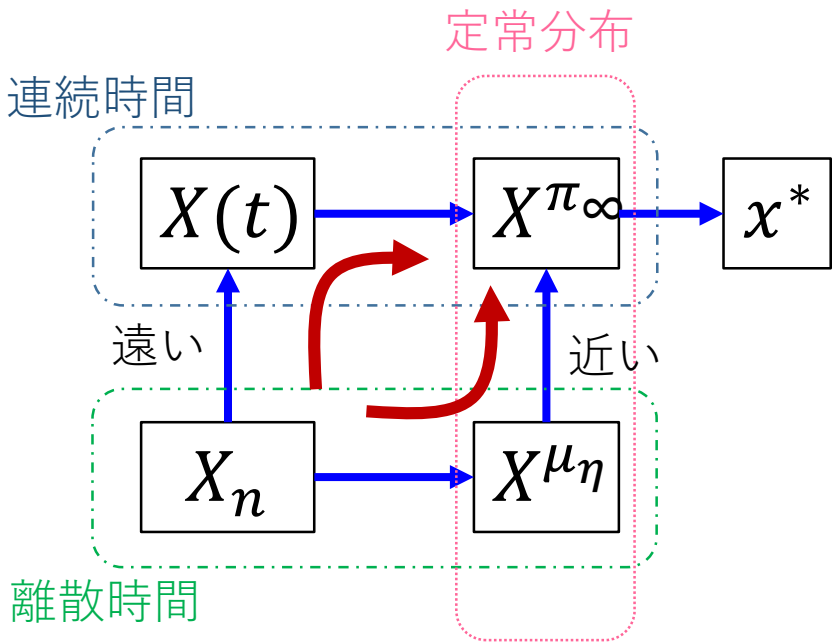
弱収束を示す:

$$|\mathbb{E}[\phi(X_n)] - \phi(x^*)| \leq ?$$

for a smooth function ϕ.

- Raginsky et al. (2017),
  Bréhier (2014), Bréhier and Kopec (2016):

$$\mathbb{E}[\phi(X_n) - \phi(X(n\eta))] + \mathbb{E}[\phi(X(n\eta)) - \phi(X^{\pi\infty})] + \mathbb{E}[\phi(X^{\pi\infty}) - \phi(x^*)]$$



- Xu et al. (2018):

$$\mathbb{E}[\phi(X_n) - \phi(X^{\mu_\eta})] + \mathbb{E}[\phi(X^{\mu_\eta}) - \phi(X^{\pi\infty})] + \mathbb{E}[\phi(X^{\pi\infty}) - \phi(x^*)]$$

レートが速い, 一方でより強い条件が必要
(Strong smoothness)

第一項のバウンド

$$\mathbb{E}[\phi(X_n) - \phi(x^*)] = \mathbb{E}[\phi(X_n) - \phi(X^{\mu_\eta})] + \mathbb{E}[\phi(X^{\mu_\eta}) - \phi(X^{\pi_\infty})] + \mathbb{E}[\phi(X^{\pi_\infty}) - \phi(x^*)]$$

補題 (離散時間ダイナミクスのGeometric ergodicity)

ある定常分布μ_ηがだた一つ存在して (極限分布),
geometric ergodicity (定常分布への線形収束) が成り立つ:

$$\mathbb{E}[\phi(X_n) - \phi(X^{\mu_\eta})] \leq C(1 + \|x_0\|) \exp\left(-\Lambda_\eta^* n\eta\right)$$

ただし，“スペクトルギャップ” Λ_η^* は以下のように与えられる,

(i) (Strict dissipative)

$$\Lambda_\eta^* = \frac{\frac{\lambda}{\mu_0} - M}{1 + \eta\frac{\lambda}{\mu_0}}$$

(ii) (Bounded gradient)

$$\Lambda_\eta^* = C \min\left(\frac{\lambda}{2\mu_0}, \frac{1}{2}\right)\delta$$

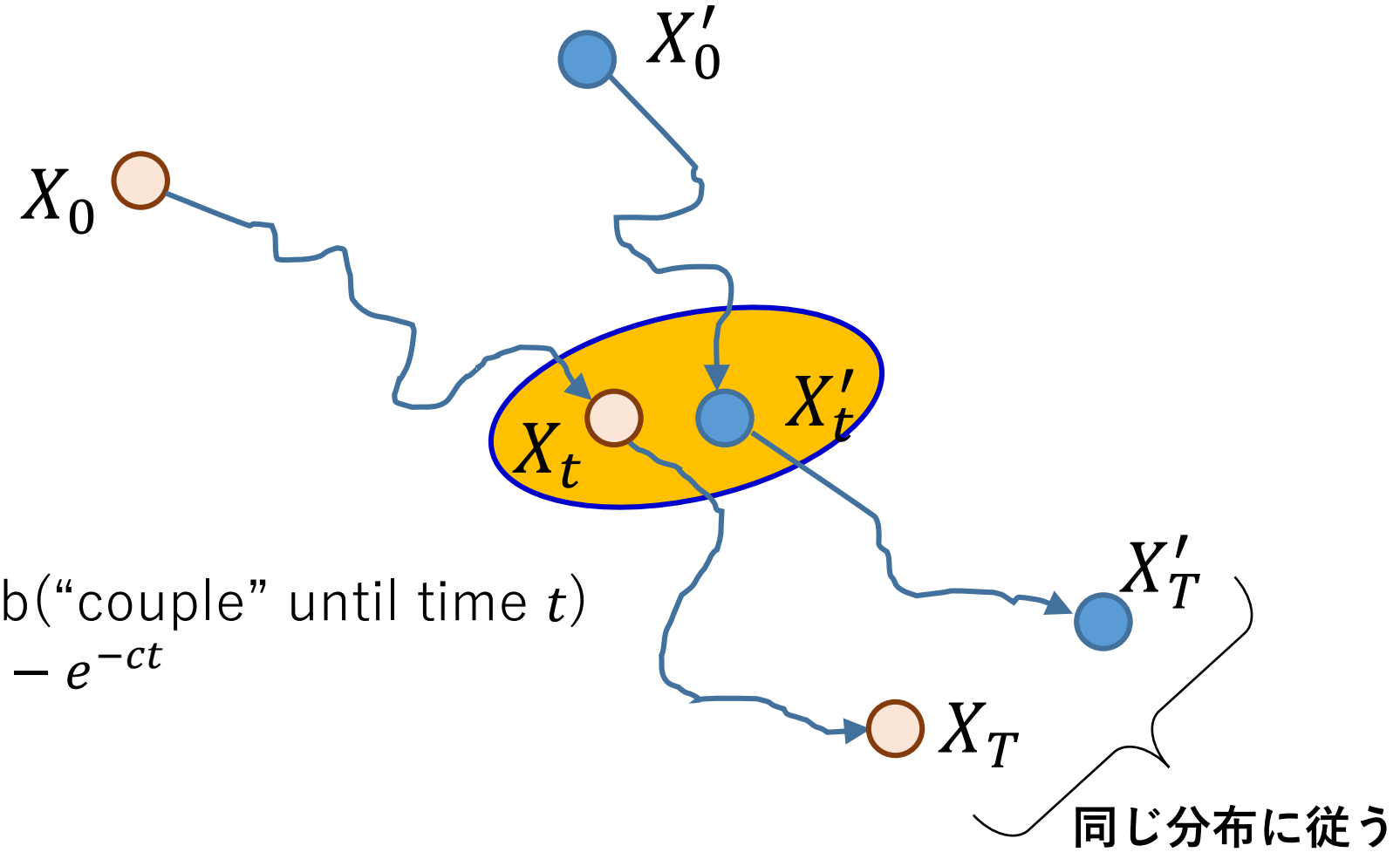
for $\quad \delta = \exp(-O(\beta))$

X^{μ_η}: r.v. obeying μ_η $\qquad$ $X_0 = x_0$ (constant)

- 有限次元の場合と違い，強平滑条件がないとおそらく成り立たない.
- **Coupling argument**: Lyapunov条件, majorization条件より
(Mattingly et al. (2002)とGoldys&Maslowski (2006)のテクニックを合わせる)

Geometric ergodicity

- Coupling argument



Prob(“couple” until time t)
$\geq 1 - e^{-ct}$

第二項のバウンド

$$\mathbb{E}[\phi(X_n) - \phi(x^*)] = \mathbb{E}[\phi(X_n) - \phi(X^{\mu_\eta})] + \mathbb{E}[\phi(X^{\mu_\eta}) - \phi(X^{\pi\infty})] + \mathbb{E}[\phi(X^{\pi\infty}) - \phi(x^*)]$$

X^{μ_η}: 離散時間ダイナミクスの定常分布
X^{π}: 連続時間ダイナミクスの定常分布（存在と一意性は保証されている）

補題（連続・離散時間ダイナミクスの定常分布の違い）

任意の$0 < \kappa < 1/2$に対し，ある定数Cが存在して，

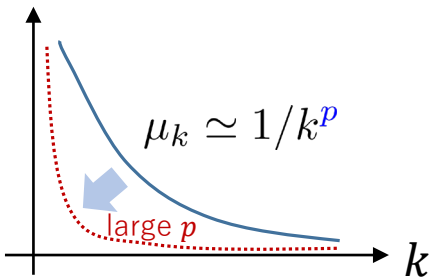
$$|\mathbb{E}[\phi(X^{\mu_\eta}) - \phi(X^{\pi})]| \leq C\|\phi\|_{0,2}\frac{c_\beta}{\Lambda_0^*}\eta^{1/2-\kappa}.$$

- $\|\phi\|_{0,2} = \max\{\|\phi\|_\infty, \|D\phi\|_\infty, \|D^2\phi\|_\infty\}$
- $c_\beta = \sqrt{\beta}$ for bounded gradient condition, and $\beta = 1$ otherwise

- **Malliavin解析**
- ステップサイズηを0に近づけると，離散時間ダイナミクスが連続時間ダイナミクスに近づく．
- βはΛ_0^*に影響している．

有限次元バージョンとの関係

$$\mathcal{H}_K = \left\{ \sum_{k=0}^{\infty} \alpha_k f_k \mid \sum_{k=0}^{\infty} \alpha_k^2 / \mu_k < \infty \right\}$$



- $\mu_k \simeq 1/k^2$ （我々の状況）

$$|\mathbb{E}[\phi(X_n) - \phi(X^\pi)]| \leq C \left[\exp\left(-\Lambda_\eta^* n\eta\right) + \frac{c_\beta}{\Lambda_0^*} \eta^{1/2-\kappa} \right] \quad \text{(optimal)}$$

- $\mu_k \simeq 1/k^p$ （予想）see [Andersson,Kruse&Larsson, 2016] for finite time horizon.

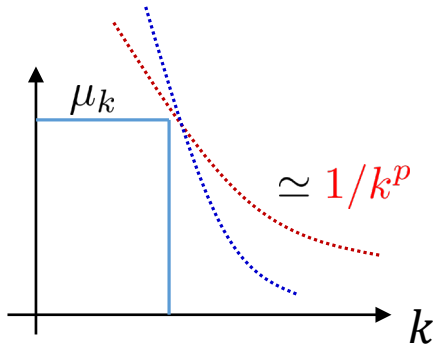
pが大きくなるほど関数クラスは“単純”になる.

$$|\mathbb{E}[\phi(X_n) - \phi(X^\pi)]| \leq C \left[\exp\left(-\Lambda_\eta^* n\eta\right) + \frac{c_\beta}{\Lambda_0^*} \eta^{\frac{p-1}{p}-\kappa} \right]$$

有限次元の解析は$p \to \infty$に対応（定数を無視すれば）:

$$|\mathbb{E}[\phi(X_n) - \phi(X^\pi)]| \leq C \left[\exp\left(-\Lambda_\eta^* n\eta\right) + \frac{c_\beta}{\Lambda_0^*} \eta \right]$$

[Xu et al. (2018)]



判別問題における速い収束

Assumption

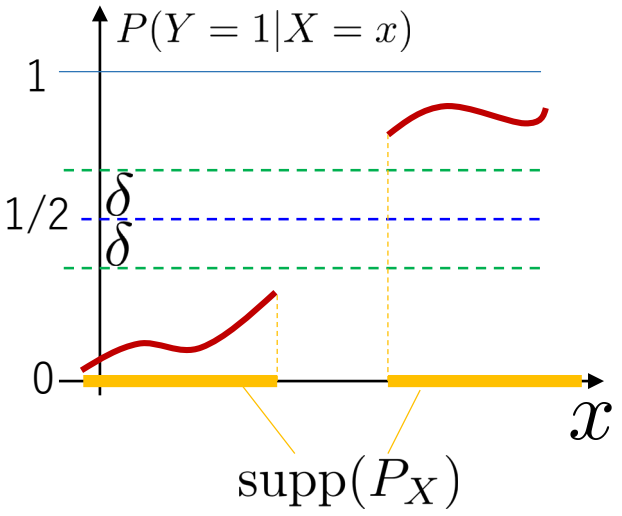
- **強低ノイズ条件:**

$$|P(Y=1|X)-1/2| \geq \delta \quad \text{(a.s.)}$$

- $\mathrm{supp}(P_X) \subset [0,1]^d$ and P_X has density p such that $p(x) \geq c_0 \quad (\forall x \in \mathrm{supp}(P_X))$.
- 活性化関数はなめらか:

$$\sigma \in \mathcal{C}^m(\mathbb{R}) \qquad \text{for } 2m > d$$

- 真の関数はモデルに入っているとする: $f^* = f_{W^*}$.



$$f_W(x) = \sum_{j=1}^{\infty} a_j \eta(w_j^\top x)$$

十分大きなnと$\beta \leq n$に対し,

$$\mathbb{E}[P_{\pi_k}(\{W_k \in \mathcal{H} \mid P_X[\mathrm{sign}(f_{W_k}(X)) = \mathrm{sign}(f^*(X))] \neq 0\})]$$

$$\lesssim \exp(-c\beta\delta^{2m/(2m-d)}) + \frac{\Xi_k}{\delta^{2m/(2m-d)}}$$

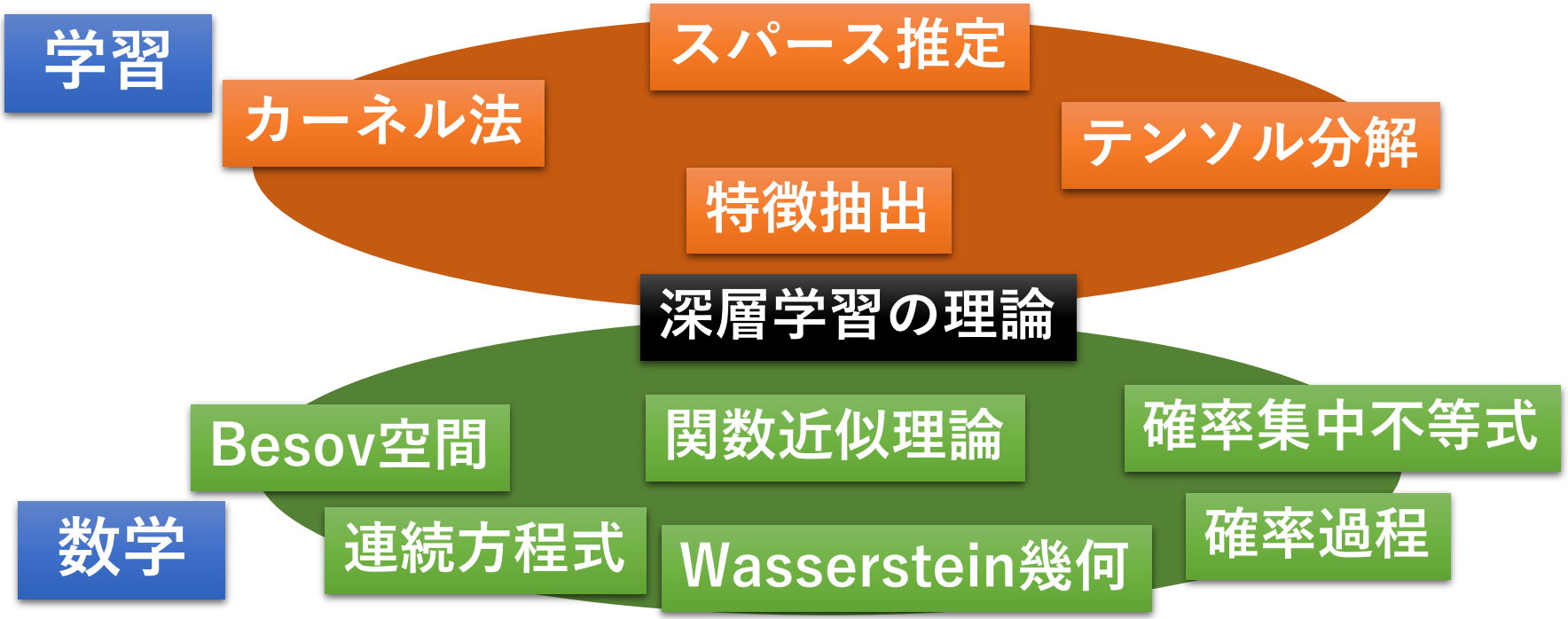
ベイズ最適な判別機が高い確率で求まる. (βは定数のままでも良い)

内容のアウトライン

- 深層学習の理論
  - 表現能力
  - 汎化能力
  - 最適化能力
- 表現力
  - 万能近似性
  - 層を深くすることで指数的に表現力増大
- 汎化能力
  - 適応的な学習手法を実現→真の関数に非凸性・スパース性があれば，カーネル法のような特徴写像を固定する方法に優越
  - 陰的正則化等でoverparametrizedな状況でも汎化
- 最適化理論
  - Overparametrizeされていれば大域的最適解を得る
  - Neural Tangent Kernelと平均場
  - スケールの仕方によって凸らしさが変わる→最適化の難しさと陰的正則化の種類が変わる（L2 vs L1）

まとめ

- 深層学習はなぜうまくいくのか？[世界的課題]
- 数学による深層学習の原理究明
  - 「表現能力」，「汎化能力」，「最適化」



理論により深層学習を“謎の技術”から“制御可能な技術”へ
深層学習を超える方法論の構築へ