

ニューラルネットの 積分表現理論

第2回 産総研 人工知能セミナー

早稲田大学 先進理工学研究科
博士後期課程(学振研究員)

園田 翔

園田 翔

2010-2012
修士(工学)



- 専攻:機械学習(村田昇教授)
- **ニューラルネット積分表現理論**
- 鉄鋼データ解析, ブログ解析



2012-2013
会社員
Panasonic

- オートモーティブ
& インダストリアルシステムズ社
- 商品開発・制御ソフト設計



2013-
博士(工学)

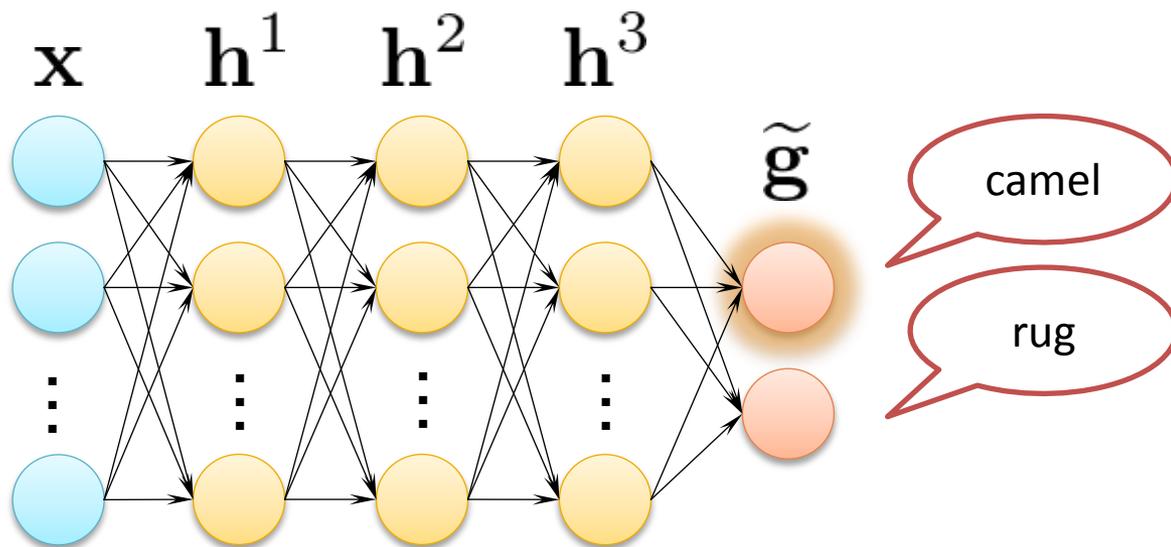
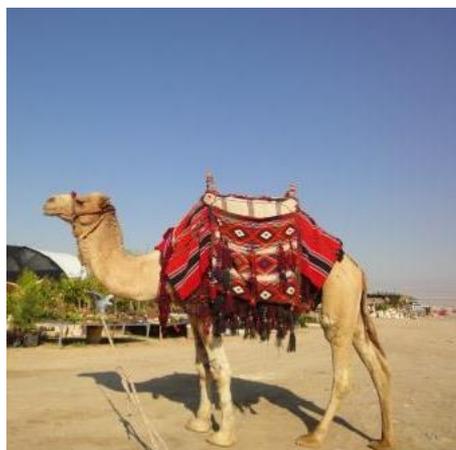


- 専攻:機械学習(村田昇教授)
- **深層学習のリッジレット解析**
- 自動運転, 脳波解析, 鉄鋼データ解析



深層構造の中では何が起きているのか？

$$g(\mathbf{x}) = \tilde{g}(h^3(h^2(h^1(\mathbf{x}))))$$



画素値
 $\mathbf{x} \in \mathbb{R}^m$

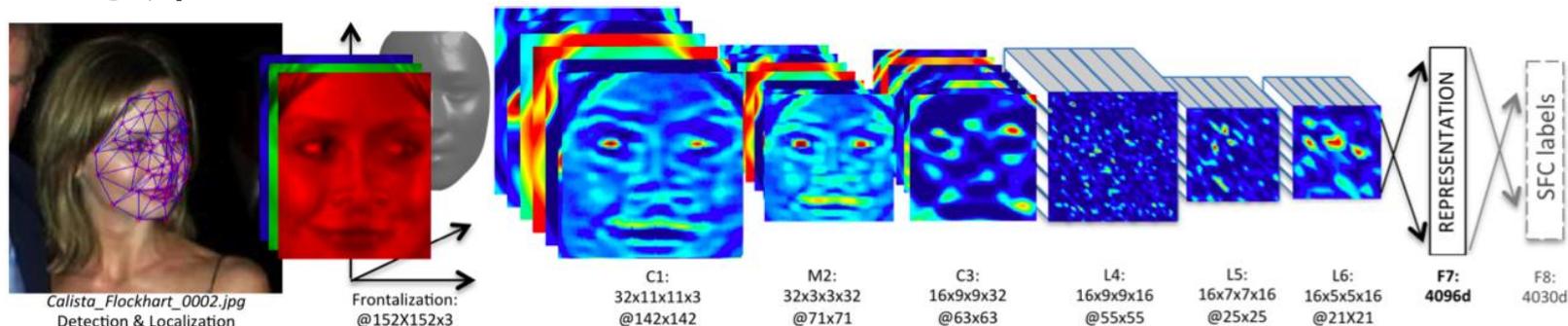
情報変換

ラベルの確率値
 $\mathbf{y} \in \mathbb{R}^k$

パラメタ数 ~10億
正答率 95+%

深い方が効率的 vs. 浅くても万能

典型的な深いモデル



[Y.Taigman+14]

徐々に抽象化する

従来の浅いモデル



MNIST

急速にコードする

各層の Complexity が低いので、
トータルのパラメタ数は少ない。

[Ba&Caruana14]
深層学習後の内部表現を
浅い構造で再学習して、
モデルを圧縮すればいい。

(バックプロパゲーション)

最小二乗法ではよく分からない

$$\operatorname{argmin}_{\{\mathbf{a}_j, b_j, c_j\}_j} \mathbb{E} \left[\left| f(\mathbf{X}_n) - \sum_j c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{X}_n - b_j) \right|^2 \right]$$

損失コスト



パラメータ空間

ニューラルネットの積分表現

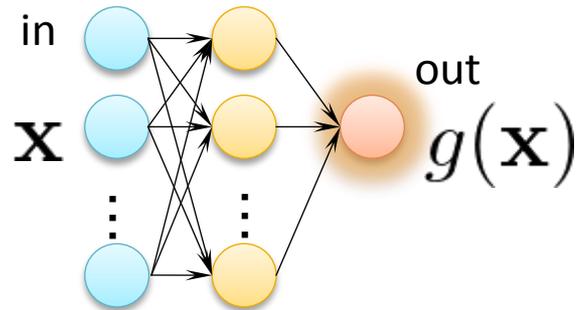
[Murata 96]

[Candes 98]

(浅い)ニューラルネット

$$g_J(\mathbf{x}) = \frac{1}{J} \sum_j c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

連続化



離散化



積分表現

$$g(\mathbf{x}) = \int_{\mathbb{R}^{m+1}} T(\mathbf{a}, b) \eta(\mathbf{a} \cdot \mathbf{x} - b) \frac{d\mathbf{a}db}{\|\mathbf{a}\|}$$
$$= \mathcal{R}_\eta^\dagger T(\mathbf{x}) \quad \text{双対リッジレット変換}$$

※当然, 10億 < 無限大

ニューラルネットのしくみ

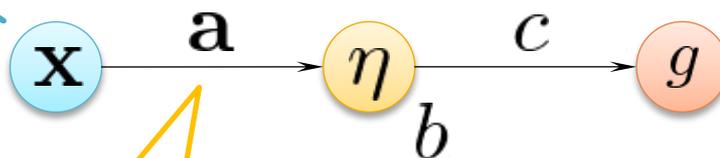
(中間層素子1個)

最小のニューラルネット

$$g(\mathbf{x}) = c \cdot \eta(\mathbf{a} \cdot \mathbf{x} - b),$$

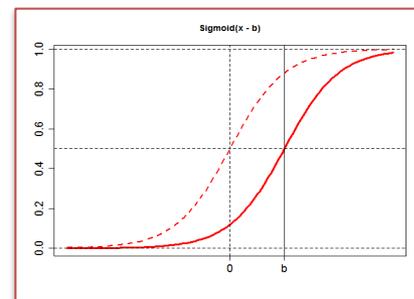
where $\eta(z) = \tanh(z), \exp(-z^2), \dots$

1. 入力 x



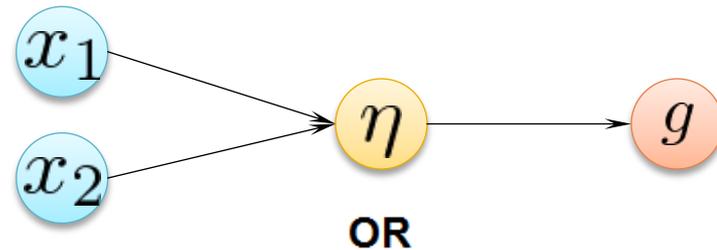
2. 入力の内積をとる

3. 閾値を超えたら発火

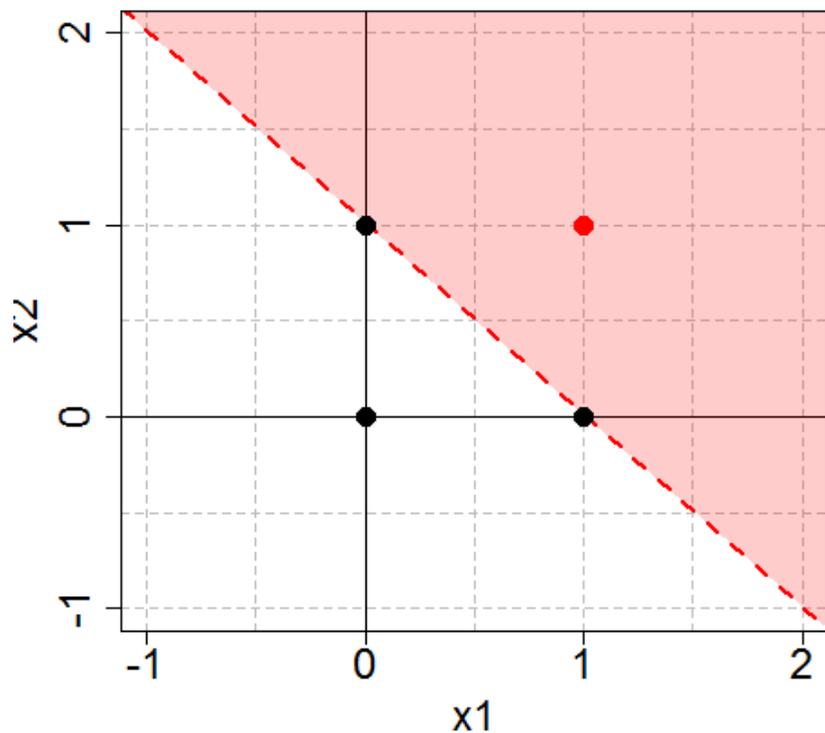


NNによる識別 = 入力空間の分割

$$g(\mathbf{x}) = c \cdot \eta(\mathbf{a} \cdot \mathbf{x} - b)$$

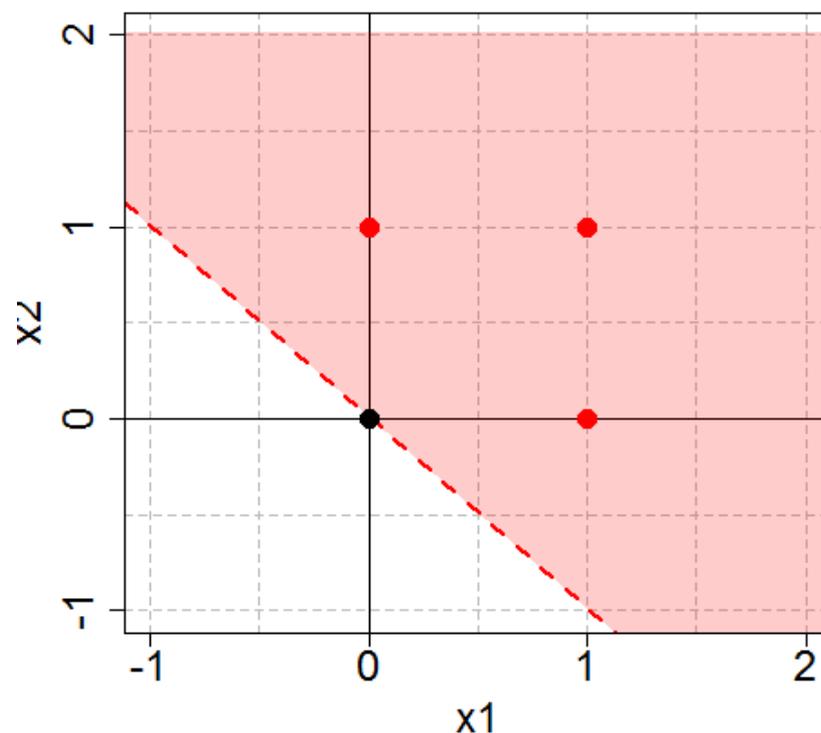


AND



$$(\mathbf{a}, b) = ((1, 1), 1)$$

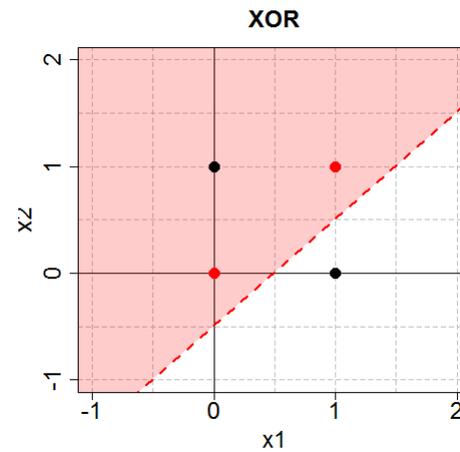
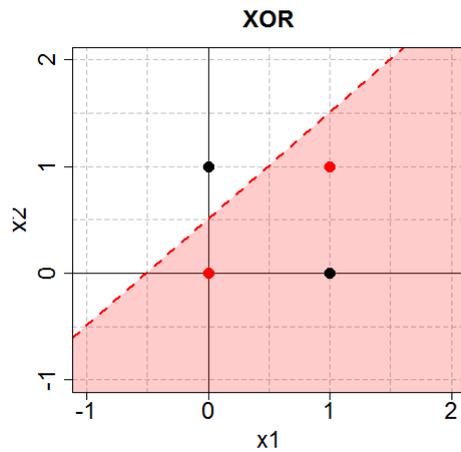
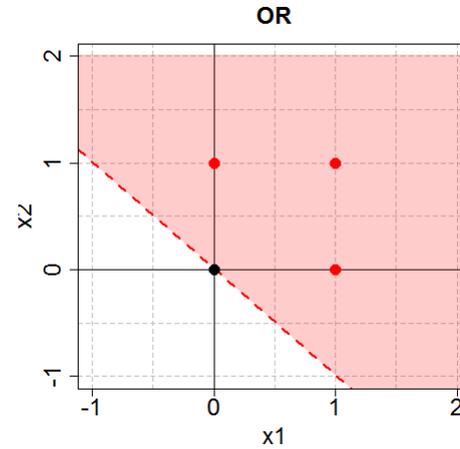
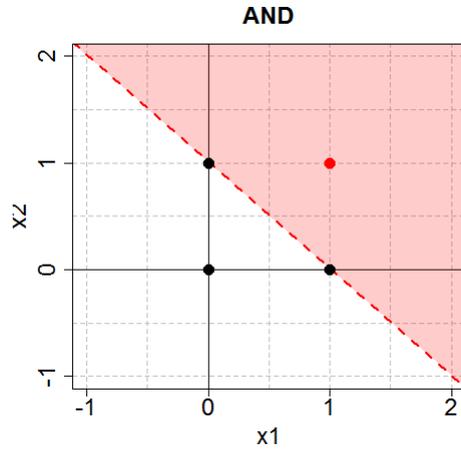
OR



$$(\mathbf{a}, b) = ((1, 1), 0)$$

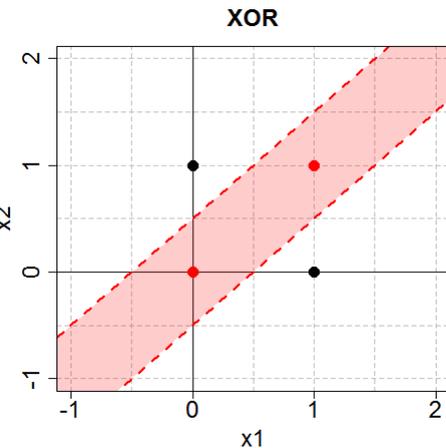
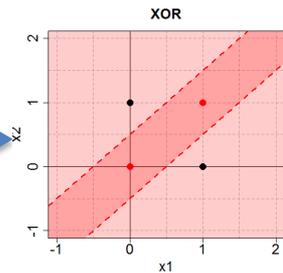
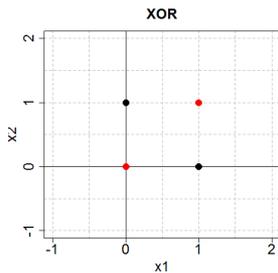
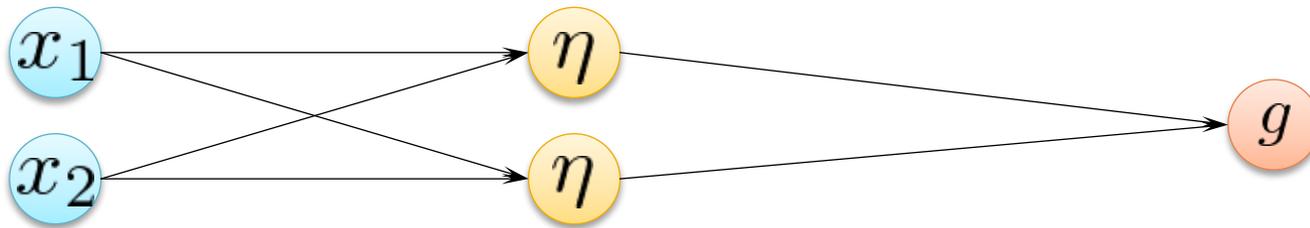
最小構成では識別できない例

[Minsky+69]



中間層素子を増やせば識別可能

$$g(\mathbf{x}) = \sum_{j=1}^2 c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$



$$(\mathbf{a}_1, b_1) = ((-1, 1), 0.5)$$

$$(\mathbf{a}_2, b_2) = ((1, -1), 0.5)$$

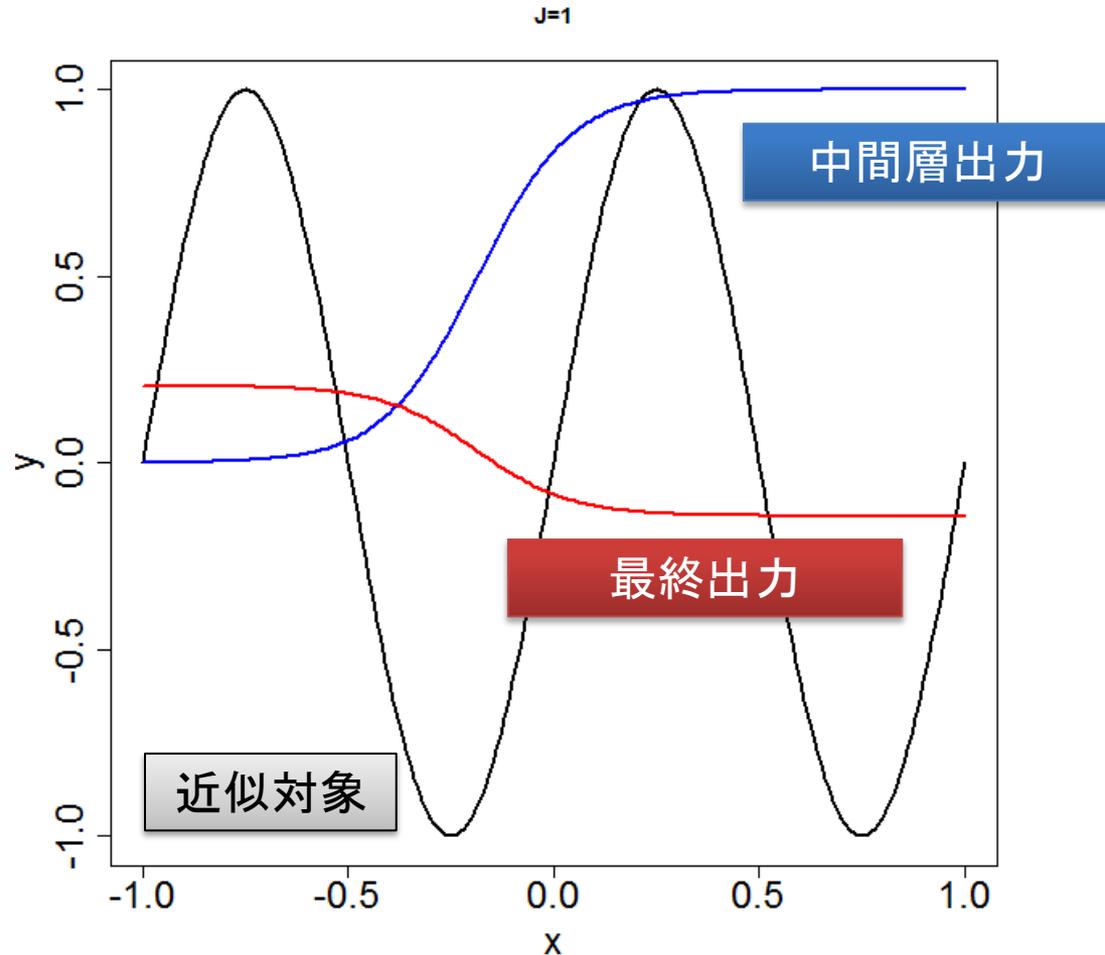
$$\mathbf{c} = (1, 1)$$

NNによる関数近似(1/10)

- 例) $y = \sin 2\pi x$ の学習例

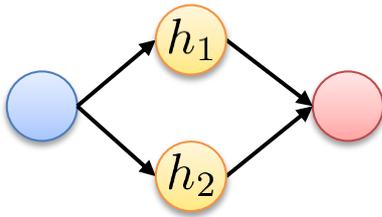


$$g(\mathbf{x}) = \sum_{j=1}^1 \underbrace{c_j}_{\text{係数}} \cdot \underbrace{\eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)}_{\text{中間層素子 = 基底関数}}$$

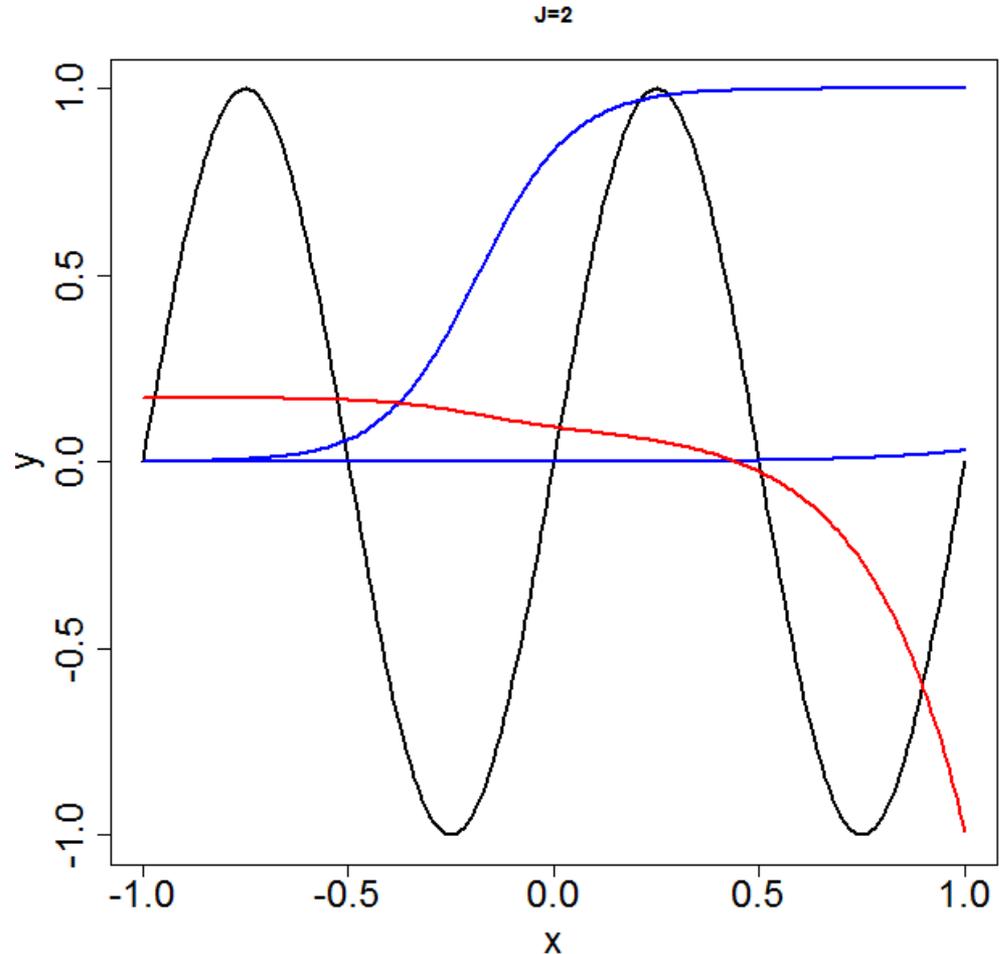


NNによる関数近似(2/10)

- 例) $y = \sin 2\pi x$ の学習例

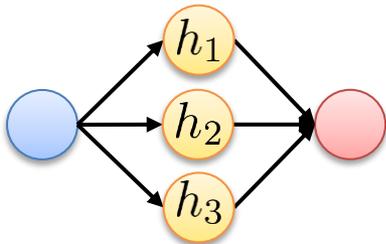


$$g(\mathbf{x}) = \sum_{j=1}^2 c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

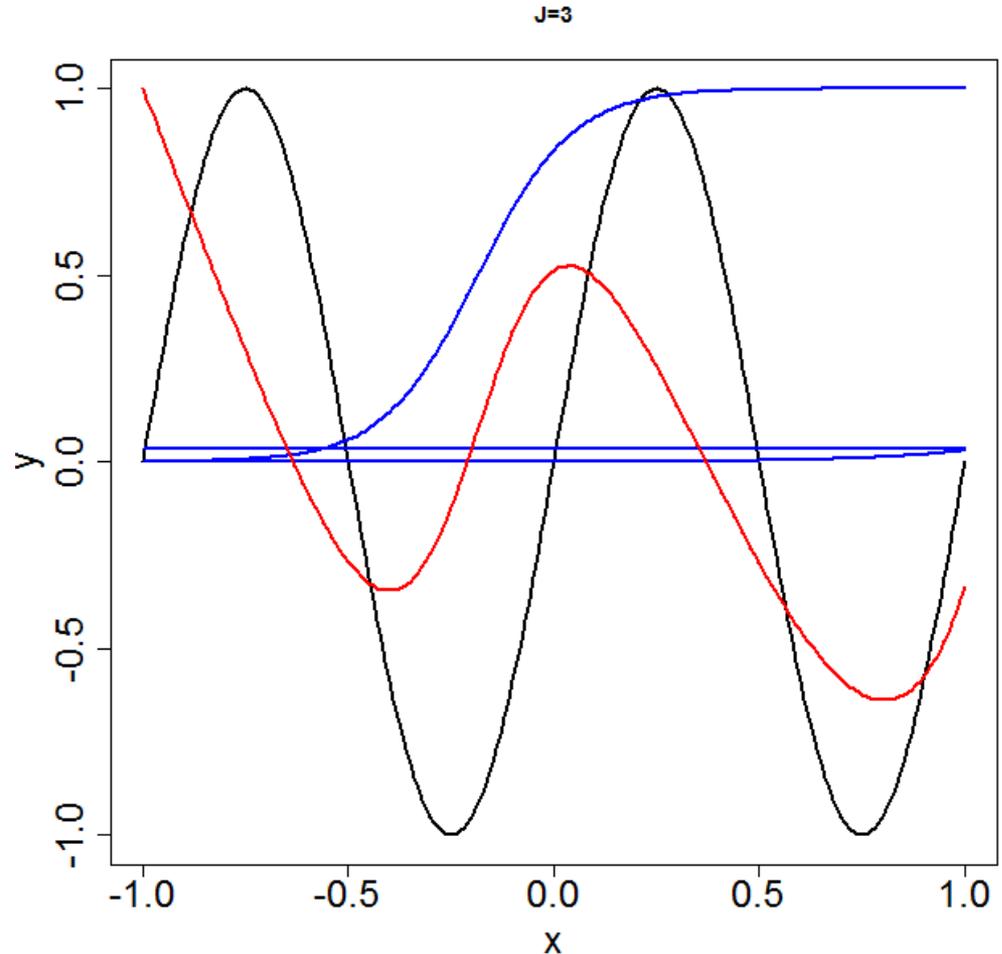


NNによる関数近似(3/10)

- 例) $y = \sin 2\pi x$ の学習例

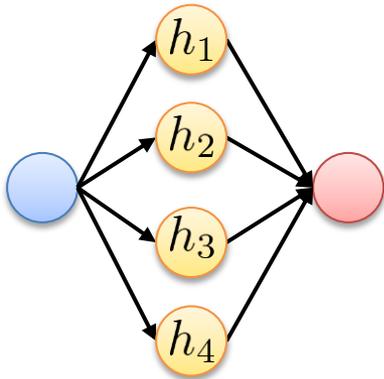


$$g(\mathbf{x}) = \sum_{j=1}^3 c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

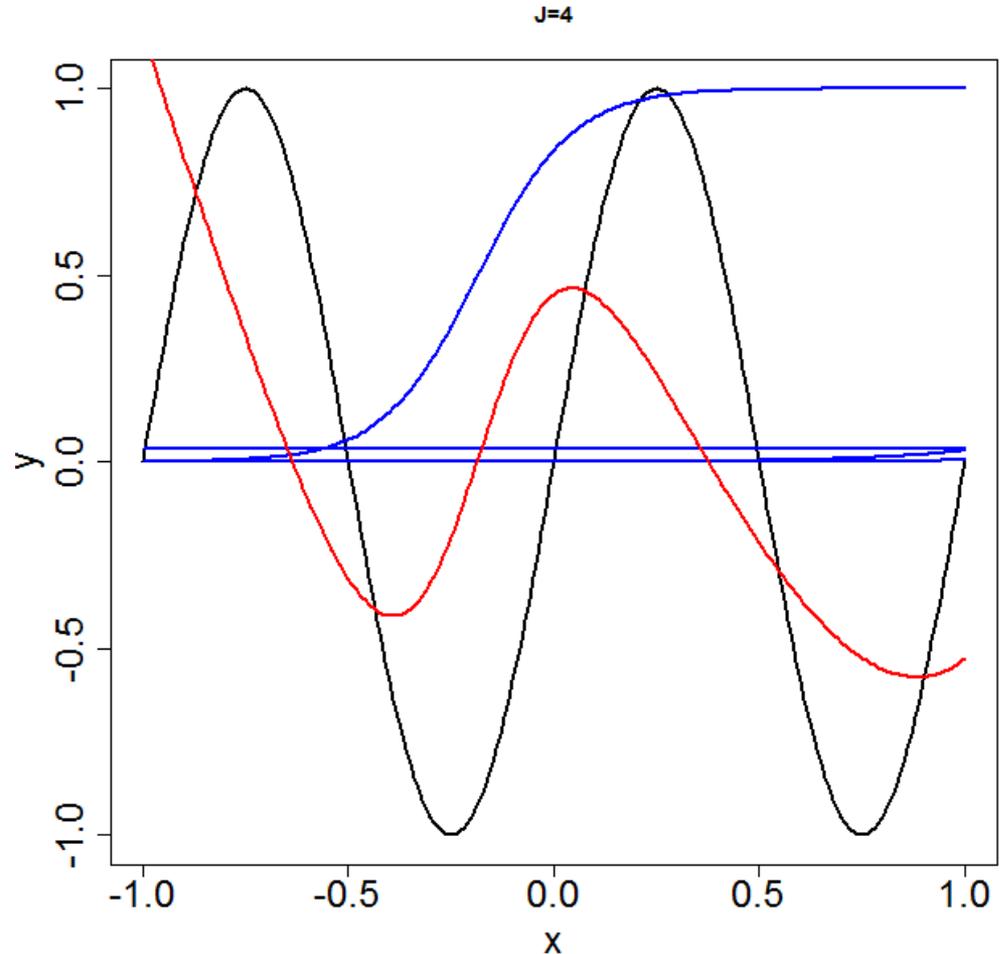


NNによる関数近似(4/10)

- 例) $y = \sin 2\pi x$ の学習例

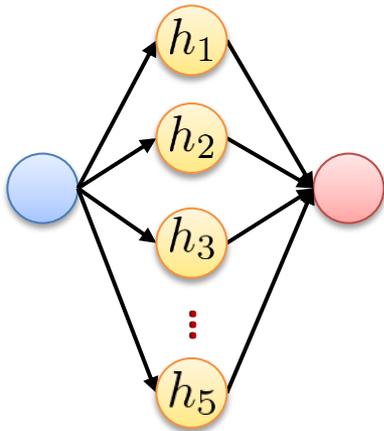


$$g(\mathbf{x}) = \sum_{j=1}^4 c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

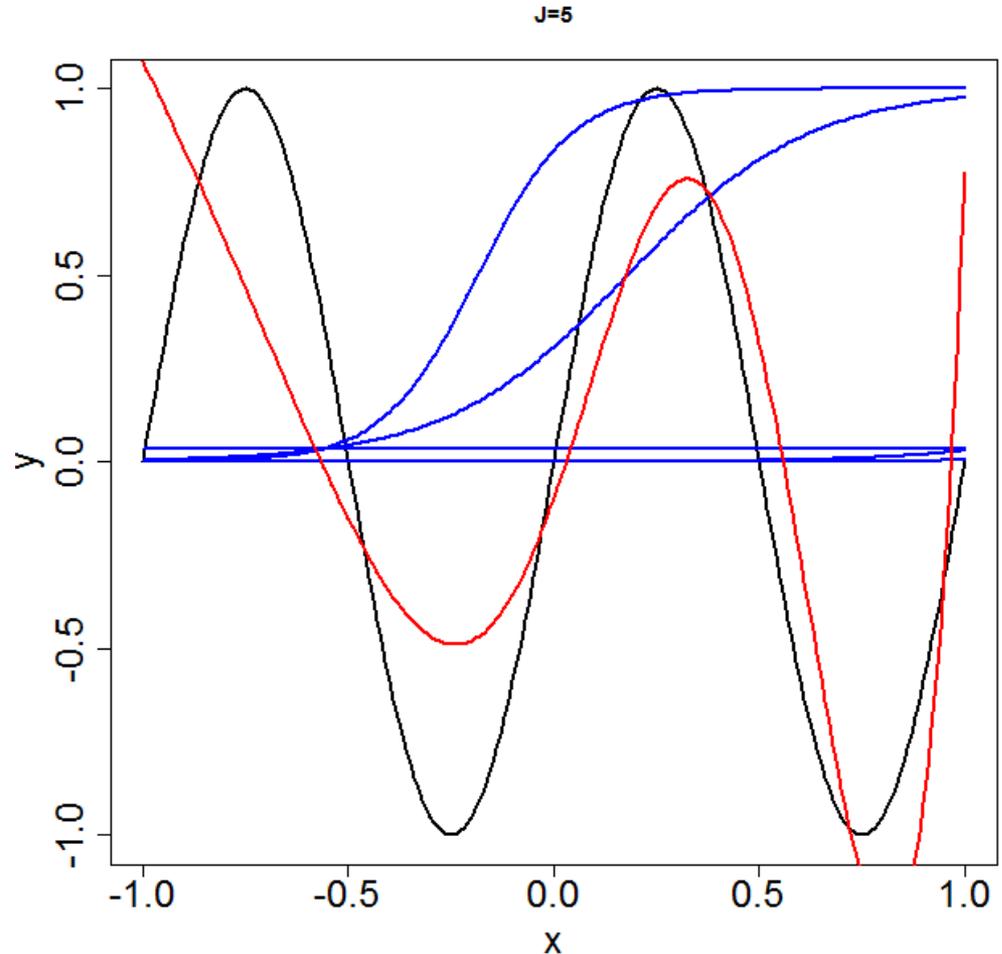


NNによる関数近似(5/10)

- 例) $y = \sin 2\pi x$ の学習例

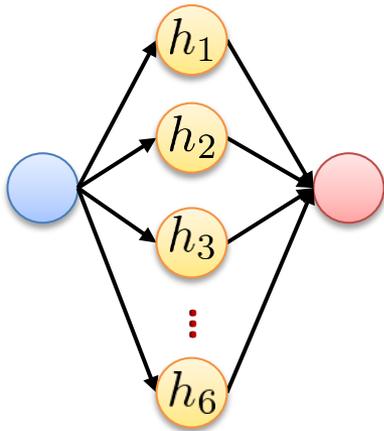


$$g(\mathbf{x}) = \sum_{j=1}^5 c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

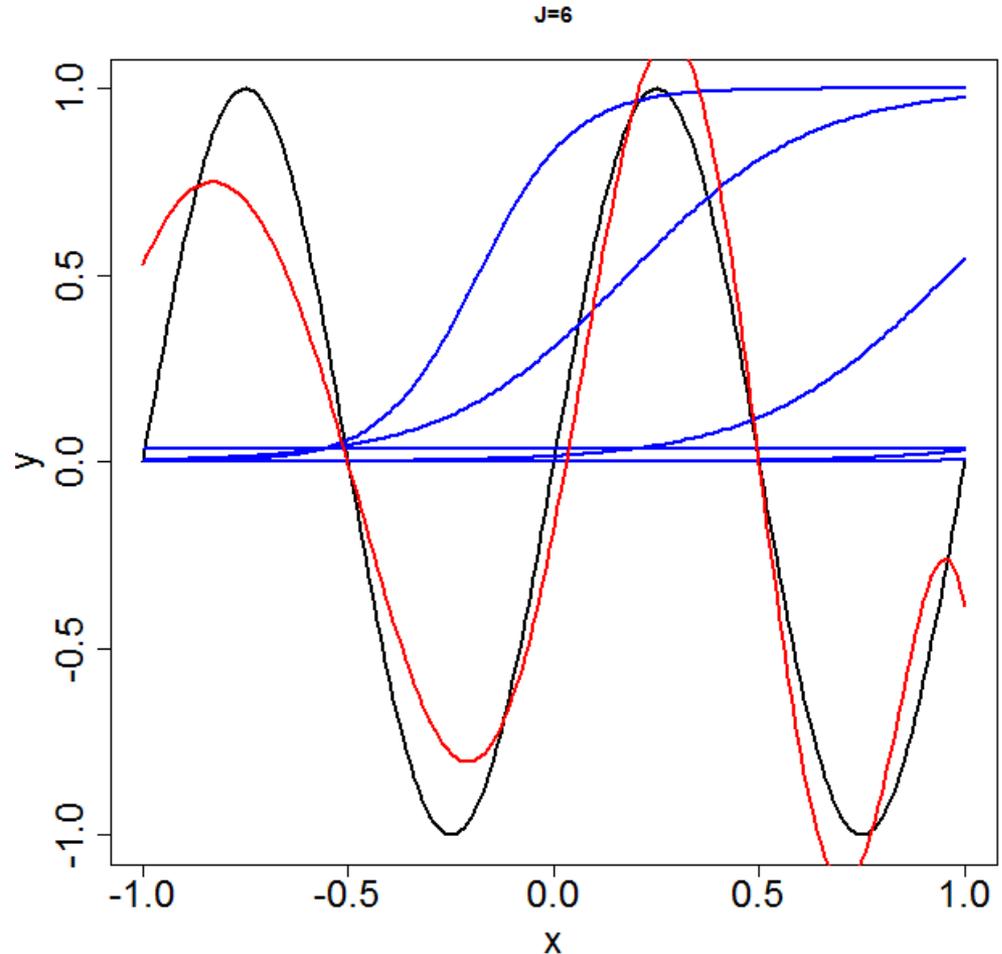


NNによる関数近似(6/10)

- 例) $y = \sin 2\pi x$ の学習例

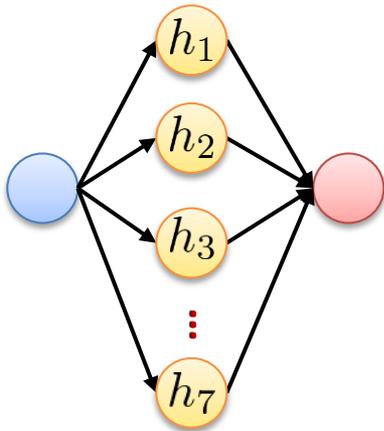


$$g(\mathbf{x}) = \sum_{j=1}^6 c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

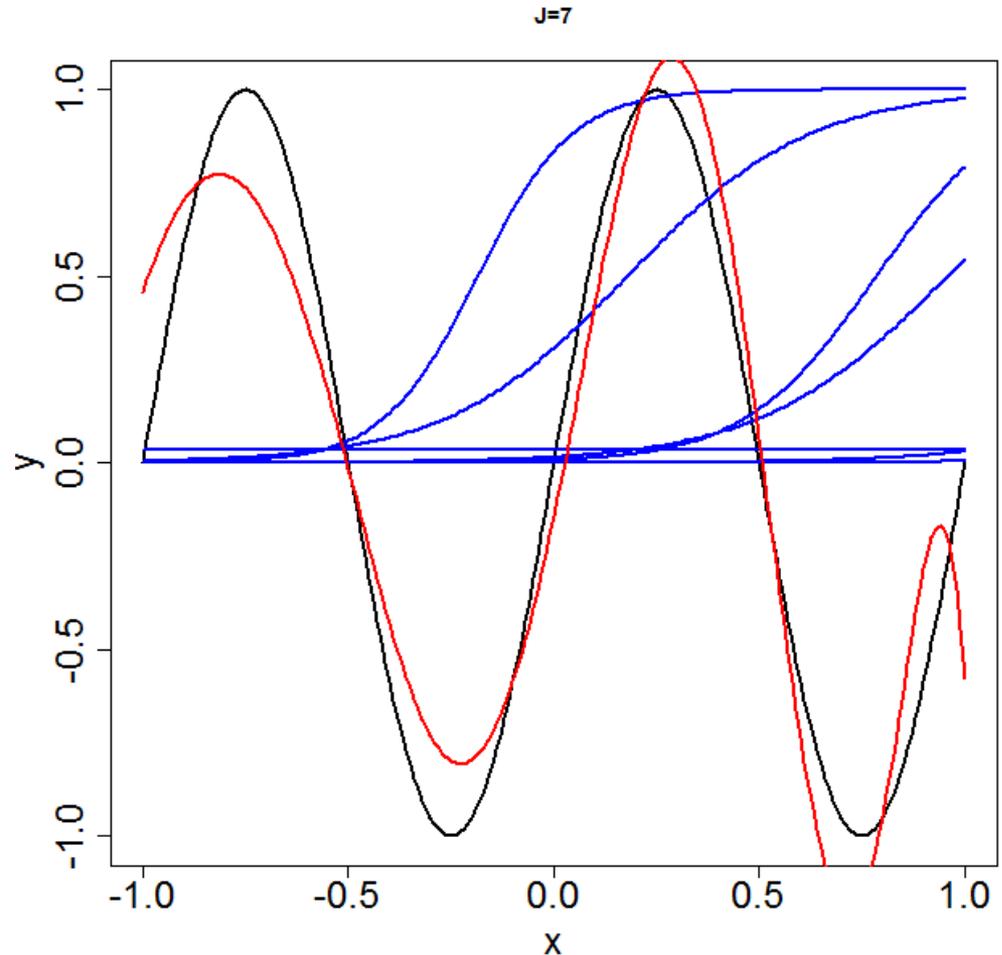


NNによる関数近似(7/10)

- 例) $y = \sin 2\pi x$ の学習例

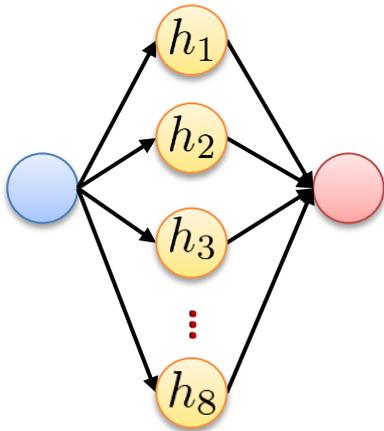


$$g(\mathbf{x}) = \sum_{j=1}^7 c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

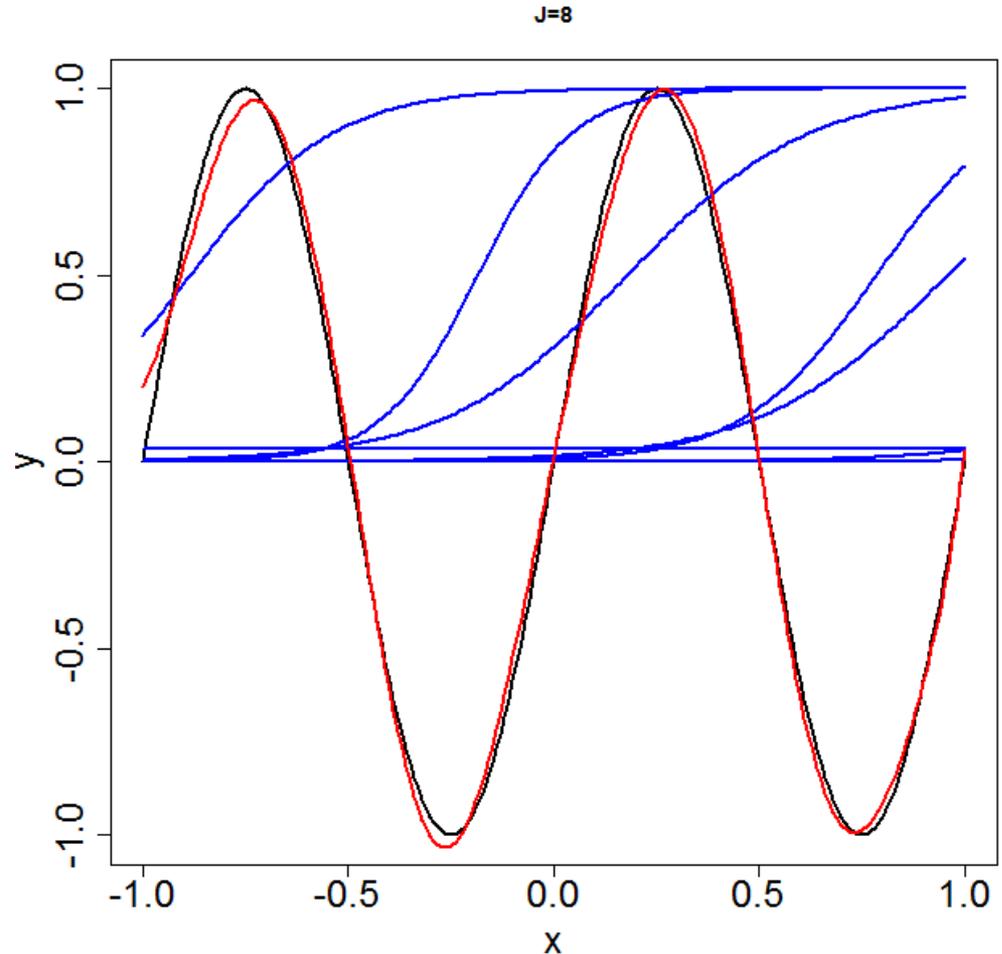


NNによる関数近似(8/10)

- 例) $y = \sin 2\pi x$ の学習例

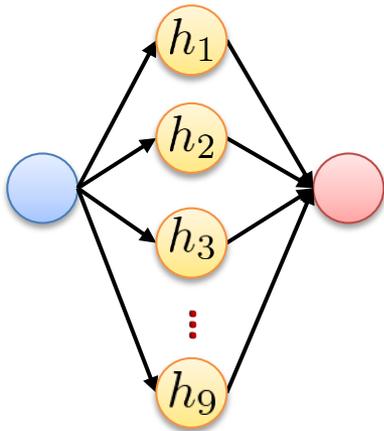


$$g(\mathbf{x}) = \sum_{j=1}^8 c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

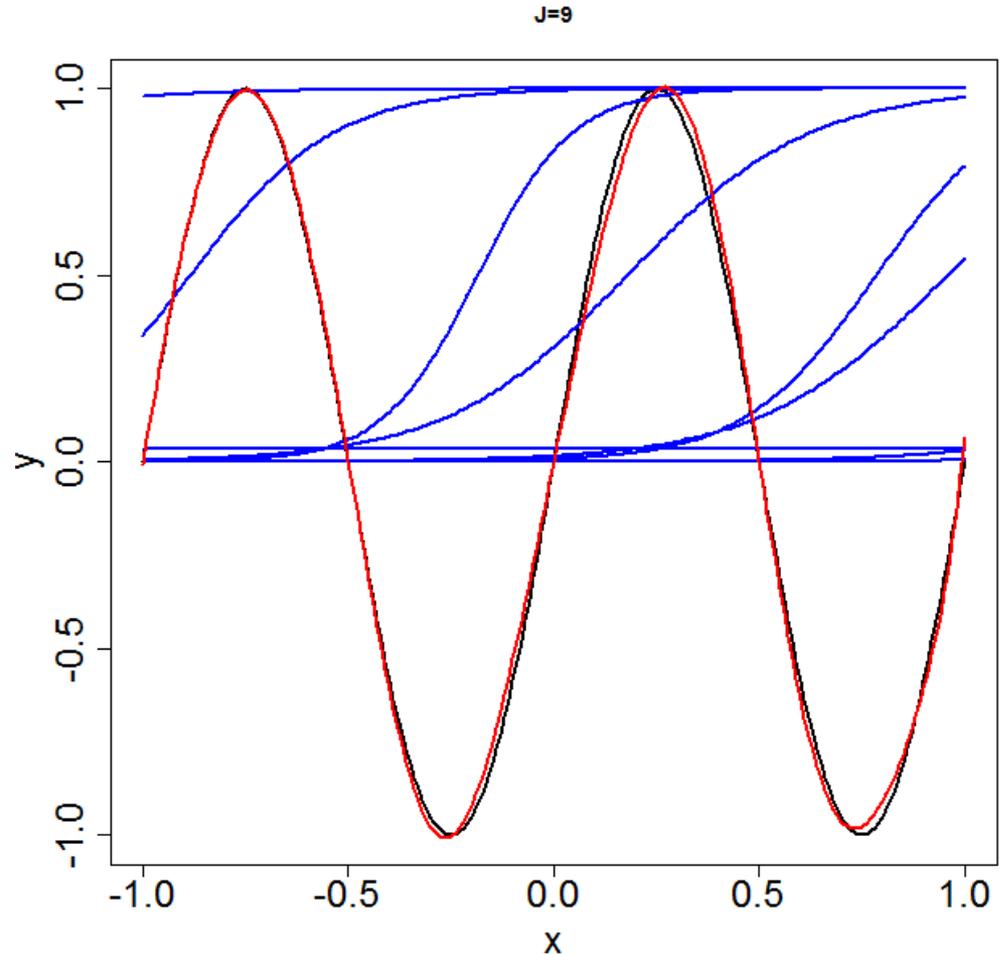


NNによる関数近似(9/10)

- 例) $y = \sin 2\pi x$ の学習例

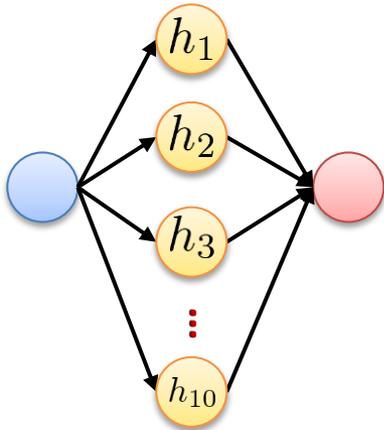


$$g(\mathbf{x}) = \sum_{j=1}^9 c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

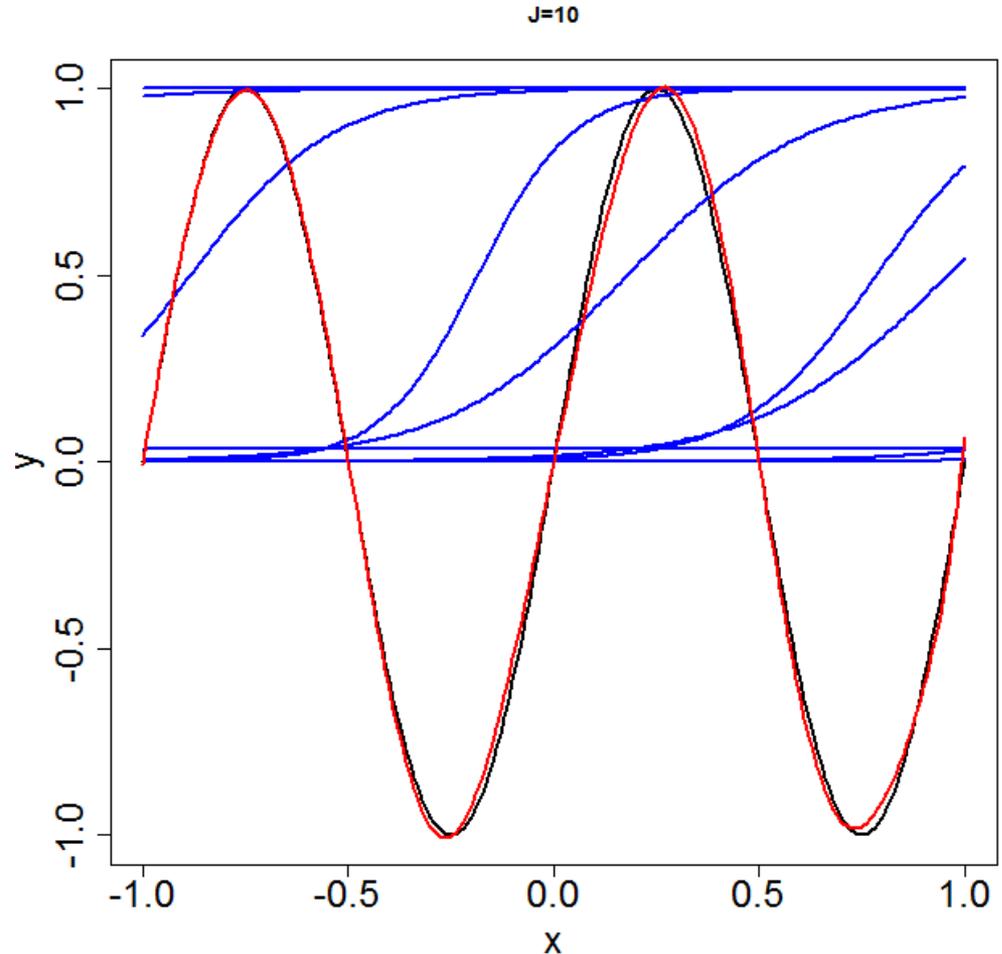


NNによる関数近似(10/10)

- 例) $y = \sin 2\pi x$ の学習例

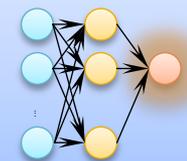


$$g(\mathbf{x}) = \sum_{j=1}^{10} c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$



中間層素子が無限にあれば、任意の関数を近似できる。

$$g(\mathbf{x}) = \sum_{j=1}^{\infty} c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$



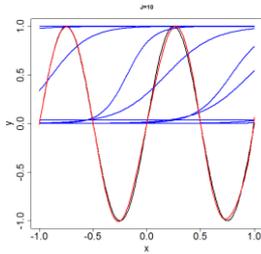
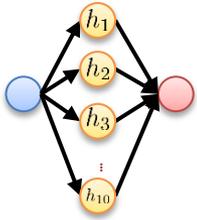
年	著者	発火関数	空間	位相	証明のテクニック
1987	Hecht-Nielsen	対象毎に構成	$C(\mathbb{R}^m)$	$C(K)$	Kolmogorov
1988	Gallant + White	Cos	$L_2(K)$	L_2	Fourier series
	Irie + Miyake	L_1	$L_2(\mathbb{R}^m)$	L_2	Fourier trans.
1989	Carroll + Dickinson	Conti. Sigmoid	$L_2(K)$	L_2	Radon trans.
	Cybenko	Conti. Sigmoid	$C(K)$	$C(K)$	Hahn-Banach
	Funahashi	Mono. Bound.	$C(K)$	$C(K)$	Irie + Miyake
	Hornik et al.	Mono. Bound.	$C(K)$	$C(K)$	Stone-Weierstrass
1993	Barron	Sigmoid	$L_2(\mathbb{R}^m)$	L_2	Fourier trans.
	Mhaskar + Micchelli	Poly. growth	$C(K)$	$C(K)$	B-spline
1996	Murata	Sigmoid	$L_p(\mathbb{R}^m)$	L_p	(Ridgelet trans.)
1998	Candes	Schwartz	$L_2(\mathbb{R}^m)$	L_2	Ridgelet trans.

(SVM, Bayesian 時代の到来とともに下火になる)

ニューラルネットの連続化

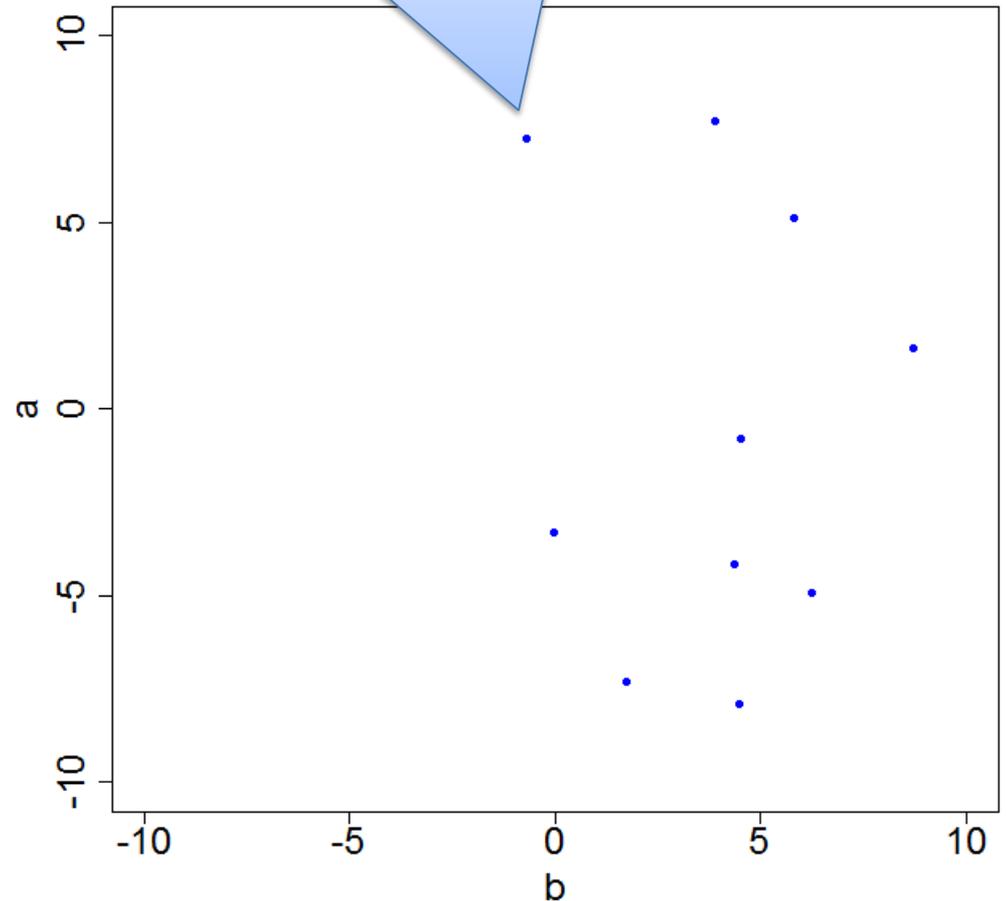
学習後の中間層パラメータをプロット

- 中間層素子10個



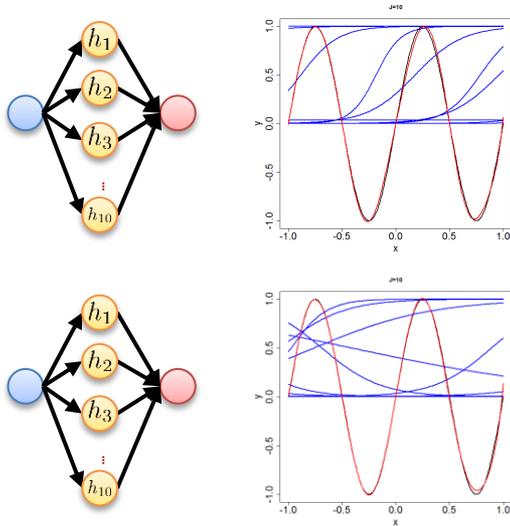
各点が中間層素子
(=基底関数)に対応する。

$$g(\mathbf{x}) = \sum_{j=1}^{10} c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$



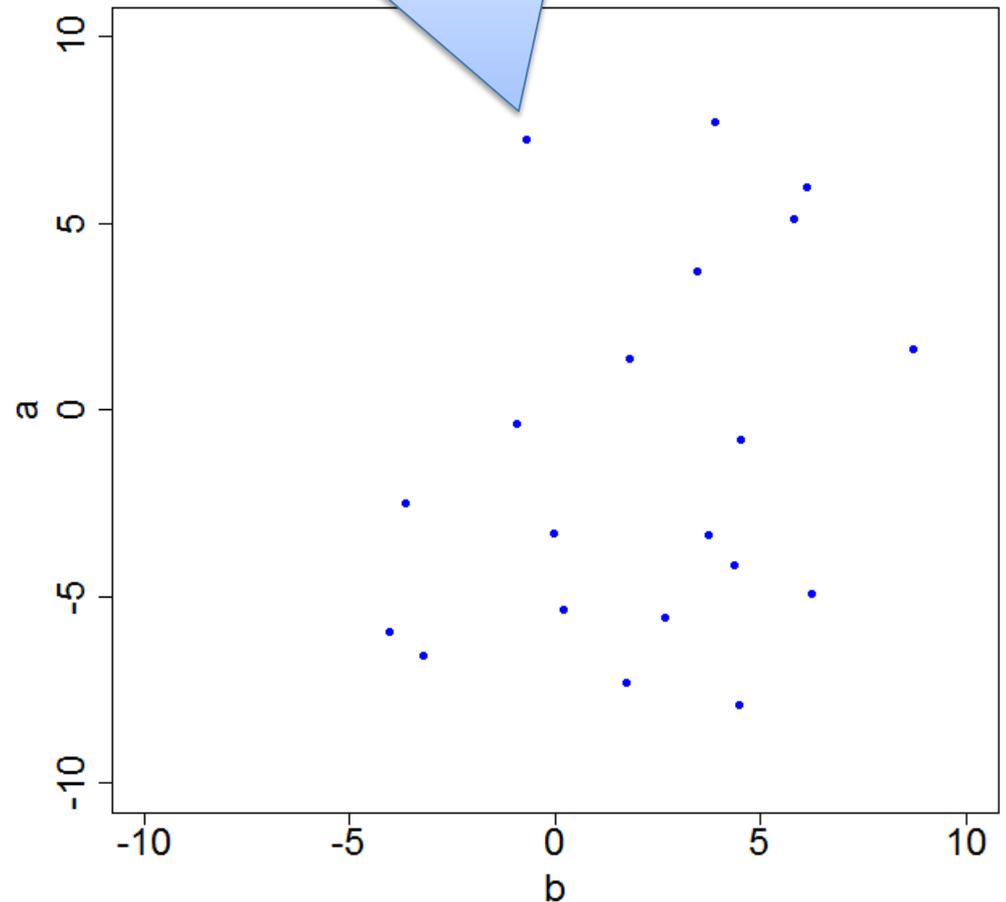
学習後の中間層パラメータをプロット

- 中間層素子10個



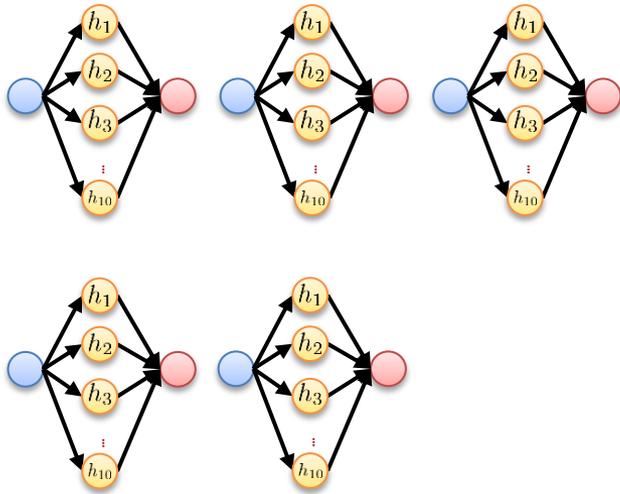
$$g(\mathbf{x}) = \sum_{j=1}^{10} c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

ニューラルネット2個分
= 20点



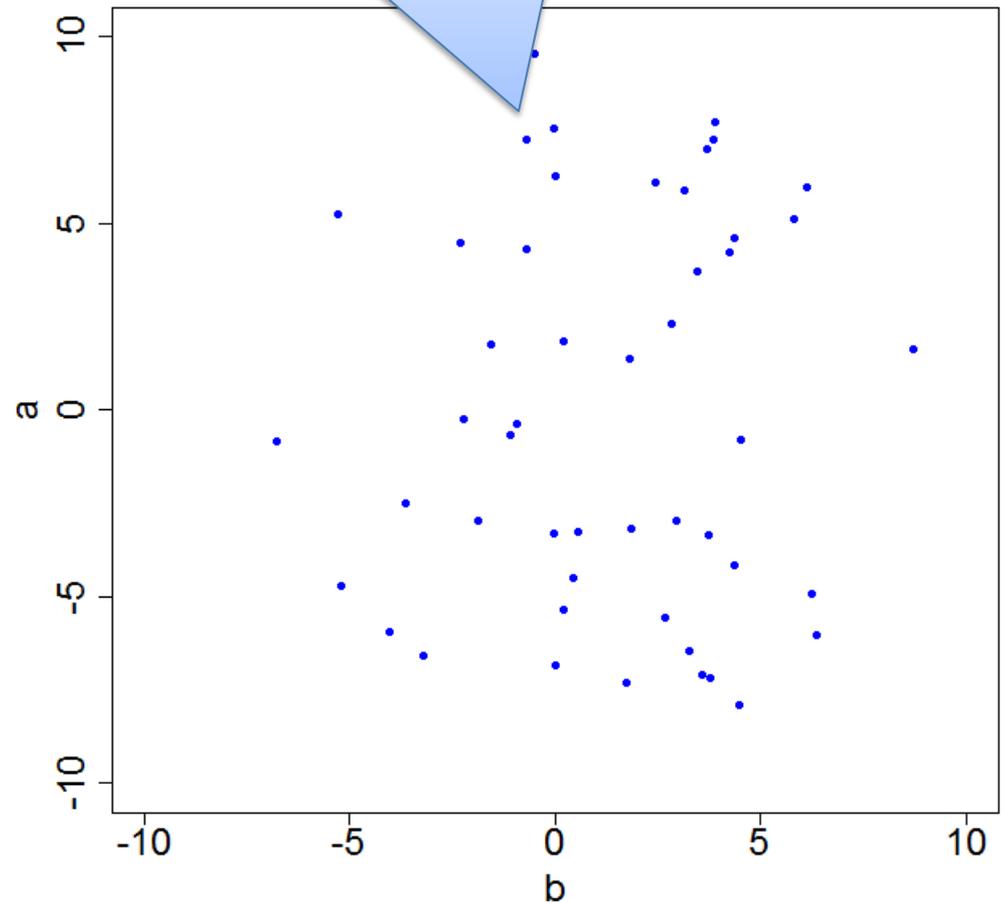
学習後の中間層パラメータをプロット

- 中間層素子10個



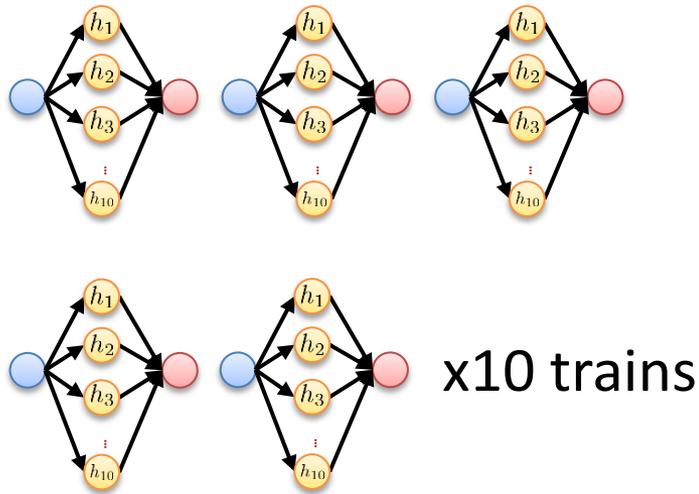
$$g(\mathbf{x}) = \sum_{j=1}^{10} c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

ニューラルネット5個分
= 50点



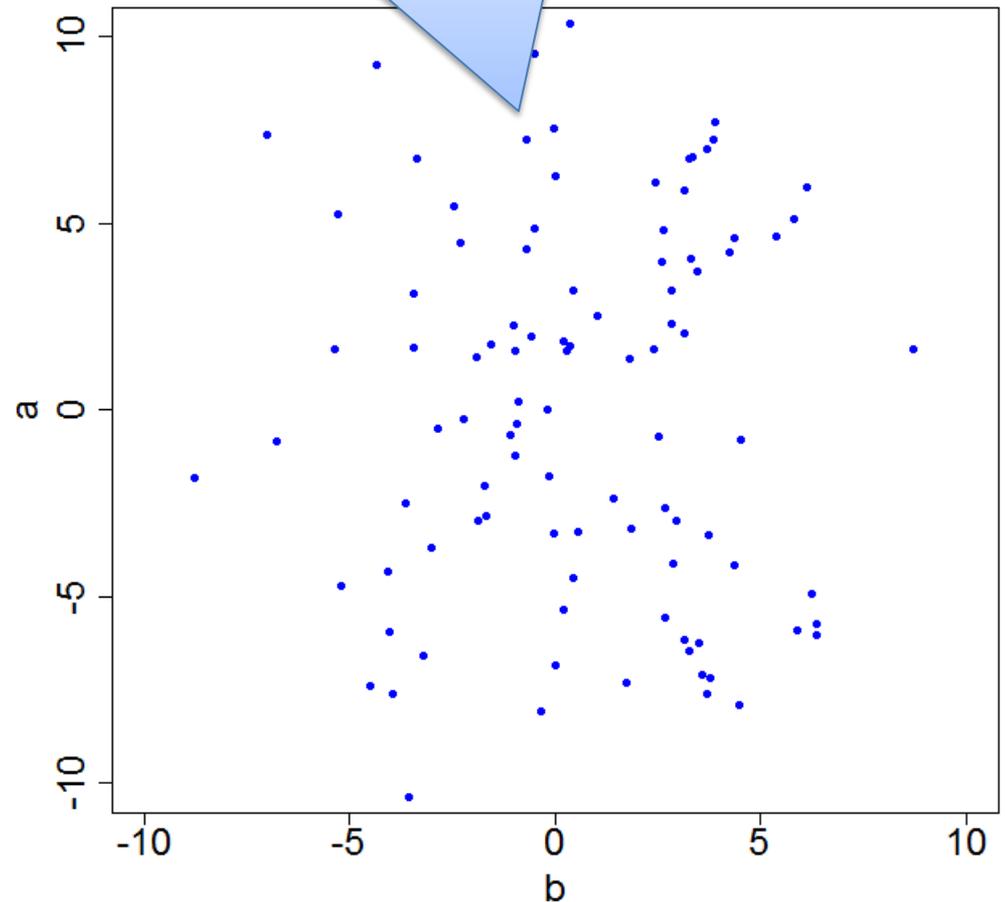
学習後の中間層パラメータをプロット

- 中間層素子10個



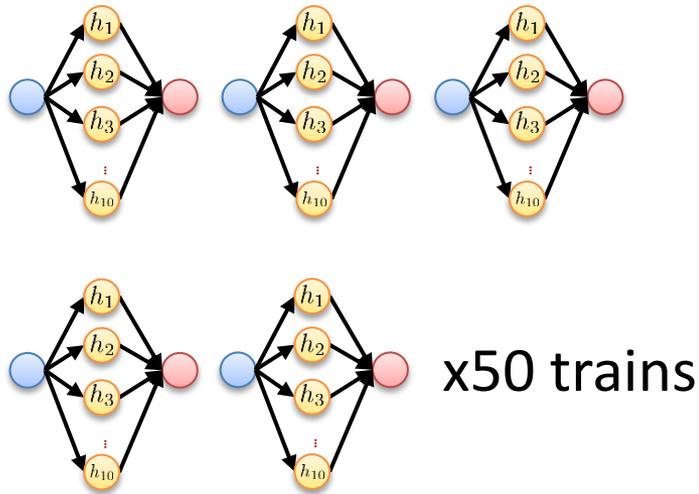
$$g(\mathbf{x}) = \sum_{j=1}^{10} c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

ニューラルネット10個分
= 100点

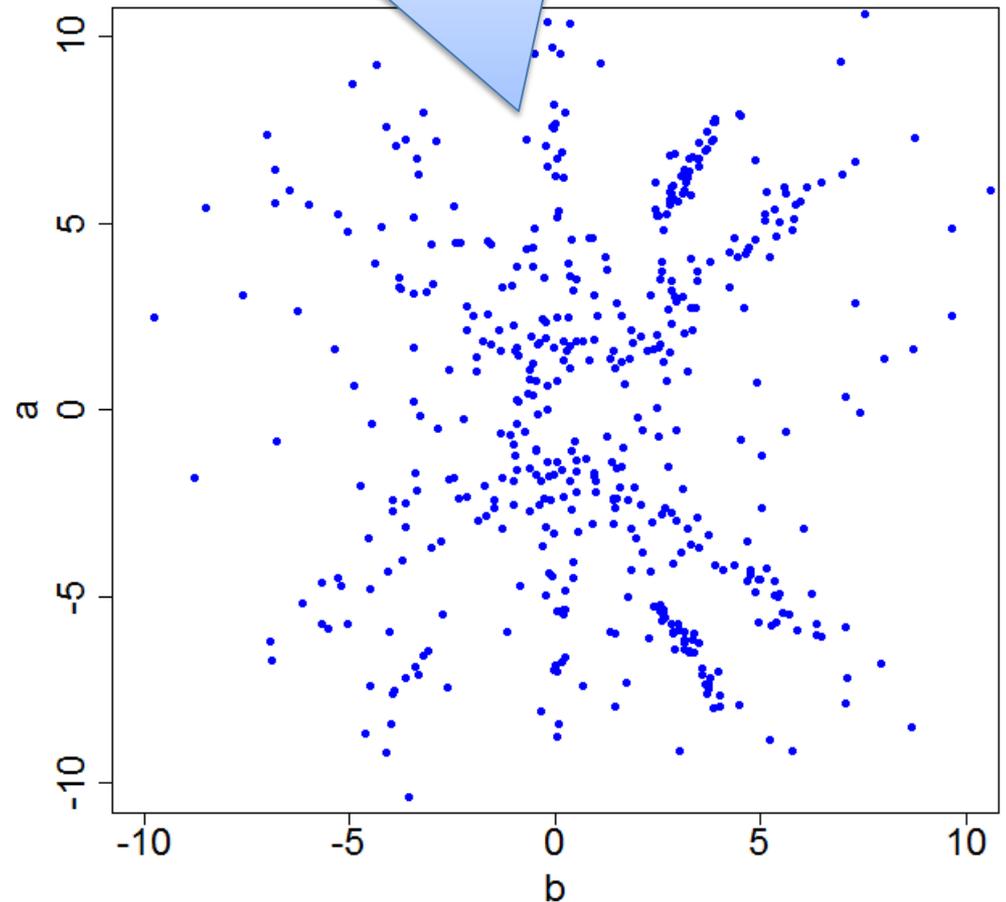


学習後の中間層パラメータをプロット

- 中間層素子10個



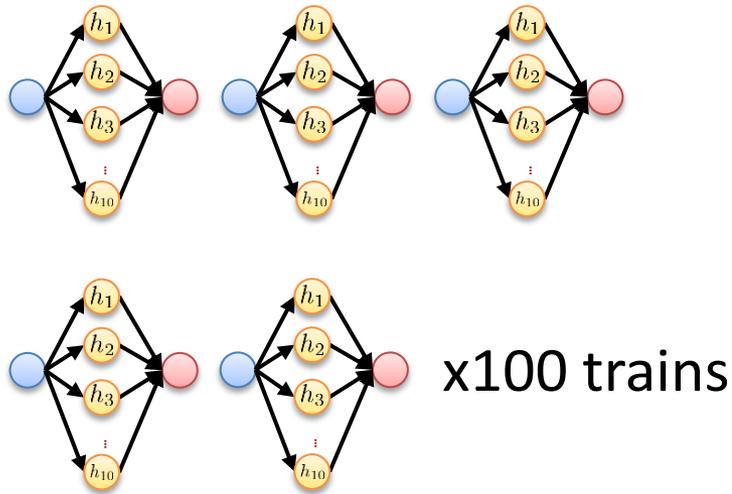
ニューラルネット50個分
= 500点



$$g(\mathbf{x}) = \sum_{j=1}^{10} c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

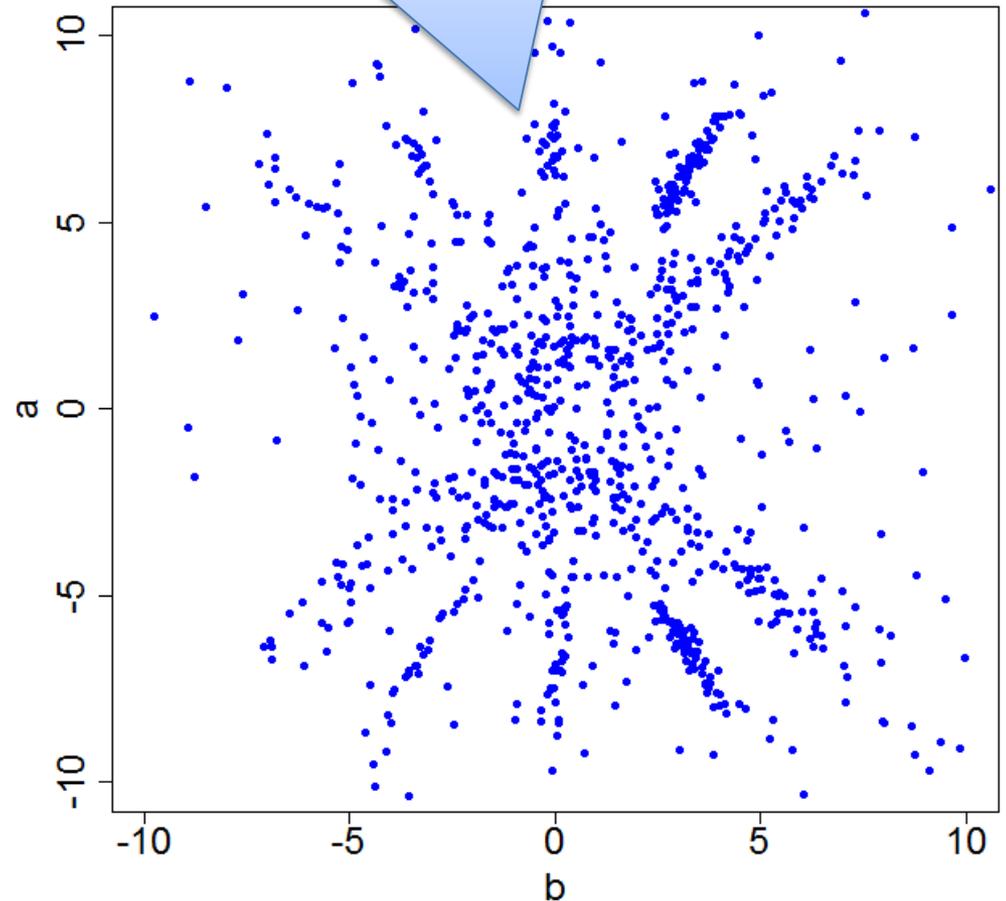
学習後の中間層パラメータをプロット

- 中間層素子10個



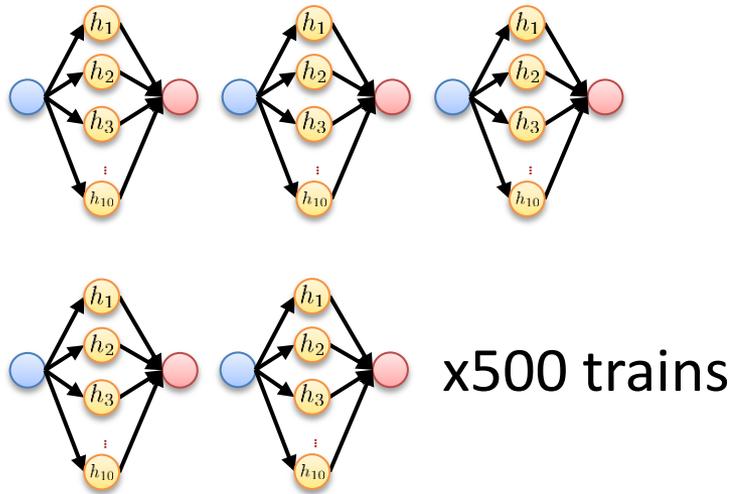
$$g(\mathbf{x}) = \sum_{j=1}^{10} c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

ニューラルネット100個分
= 1,000点

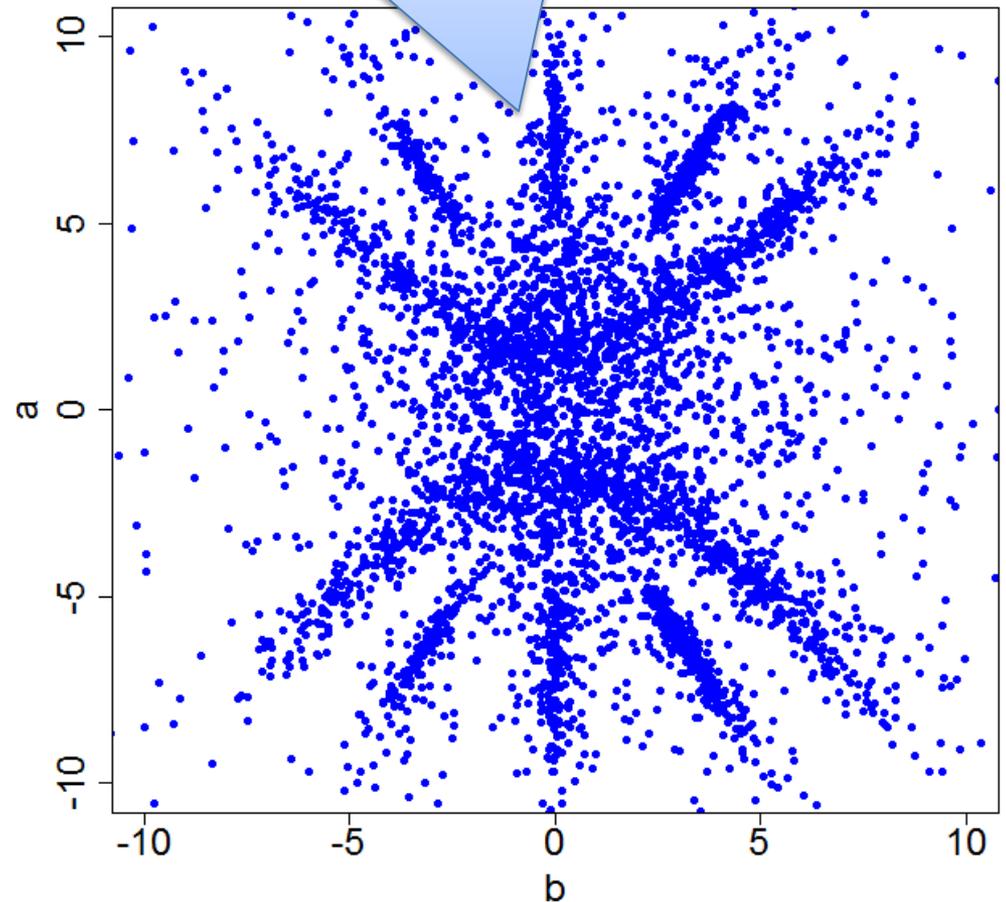


学習後の中間層パラメータをプロット

- 中間層素子10個



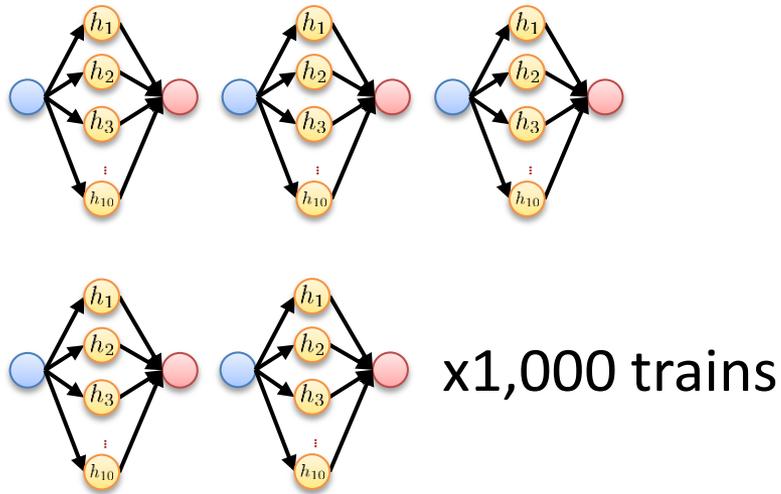
ニューラルネット500個分
= 5,000点



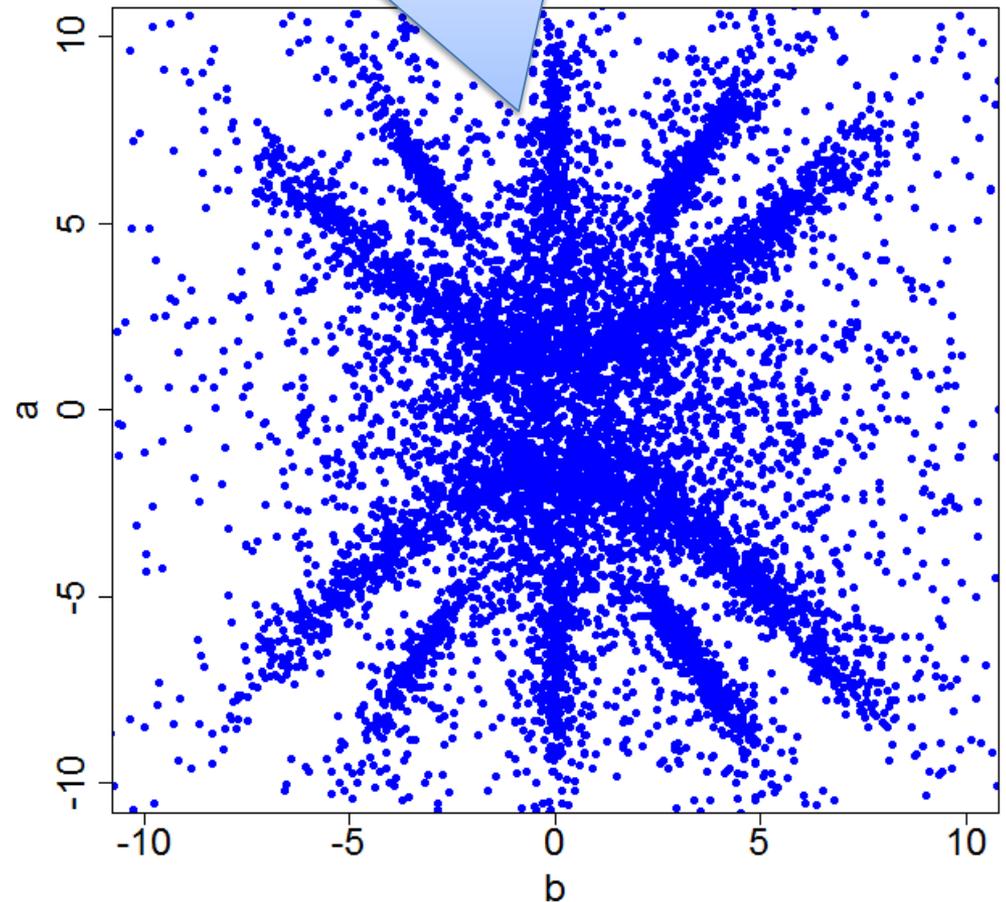
$$g(\mathbf{x}) = \sum_{j=1}^{10} c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

学習後の中間層パラメータをプロット

- 中間層素子10個



ニューラルネット1,000個分
= 10,000点

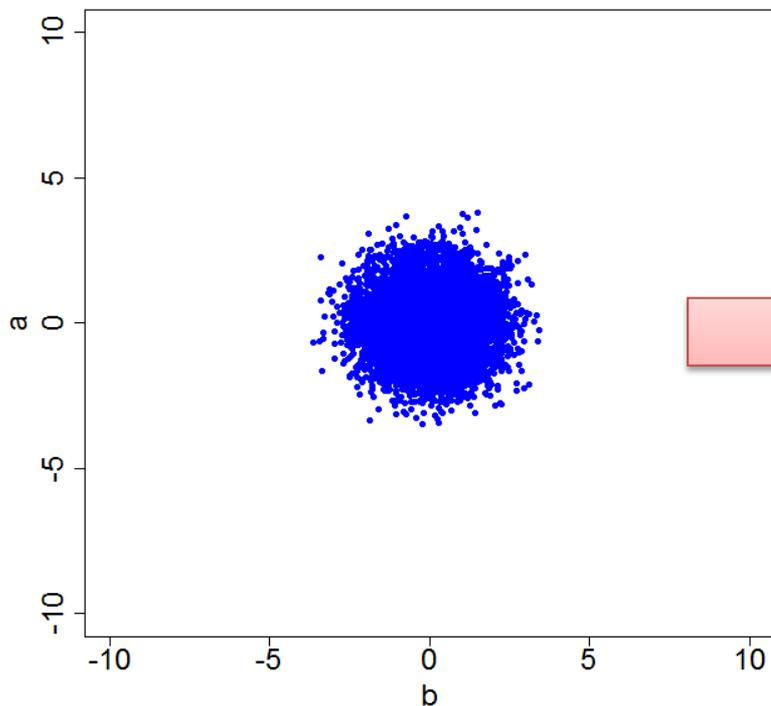


$$g(\mathbf{x}) = \sum_{j=1}^{10} c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

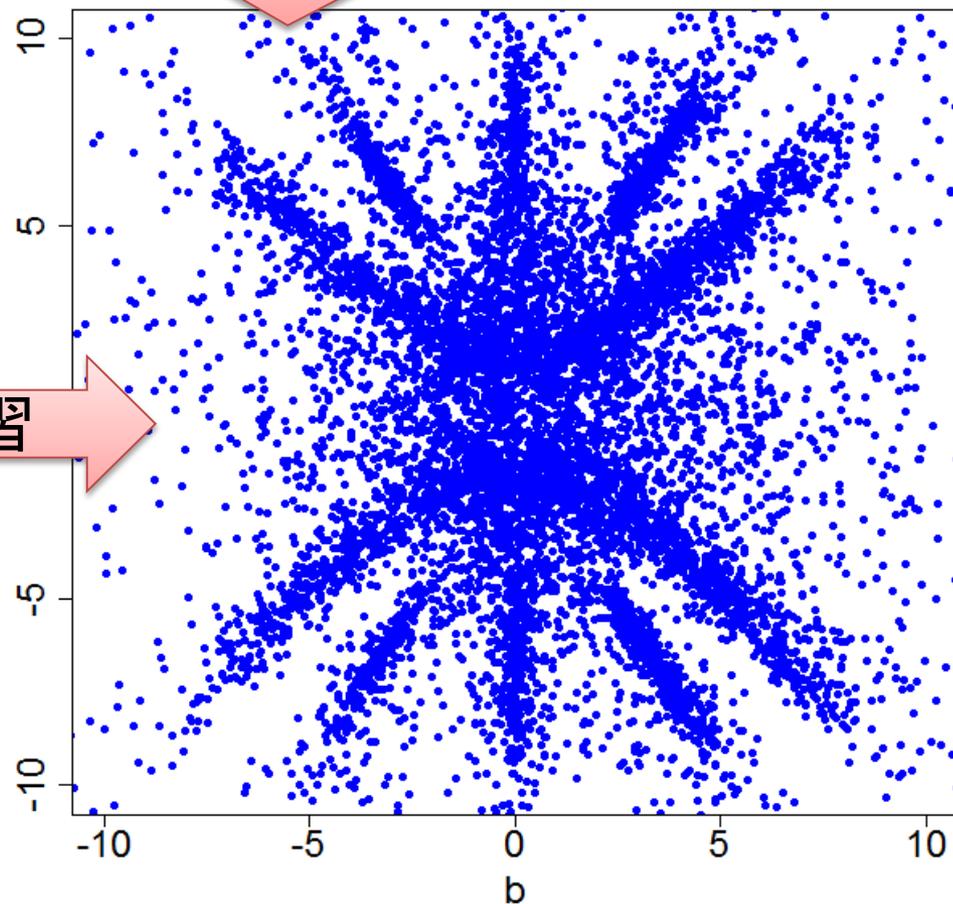
正規乱数初期化 vs. 学習後

使われやすさにバラつきがある

$$a, b \sim \mathcal{N}(0, 1)$$

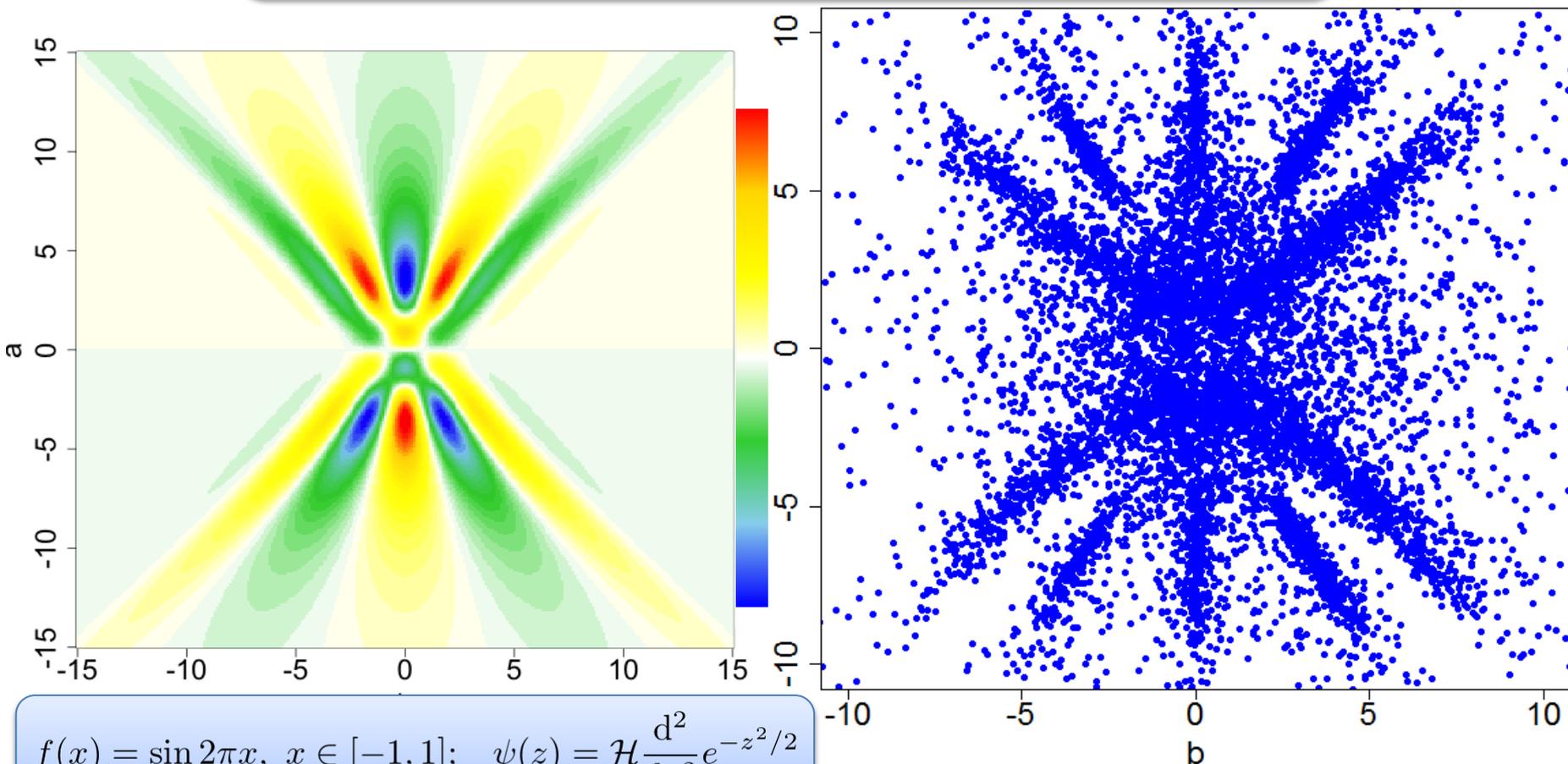


学習



パラメタ数無限大 → リッジレット変換

$$\mathcal{R}_\psi f(a, b) := \int_{\mathbb{R}} f(x) \overline{\psi(ax - b)} dx$$



$$f(x) = \sin 2\pi x, \quad x \in [-1, 1]; \quad \psi(z) = \mathcal{H} \frac{d^2}{dz^2} e^{-z^2/2}$$

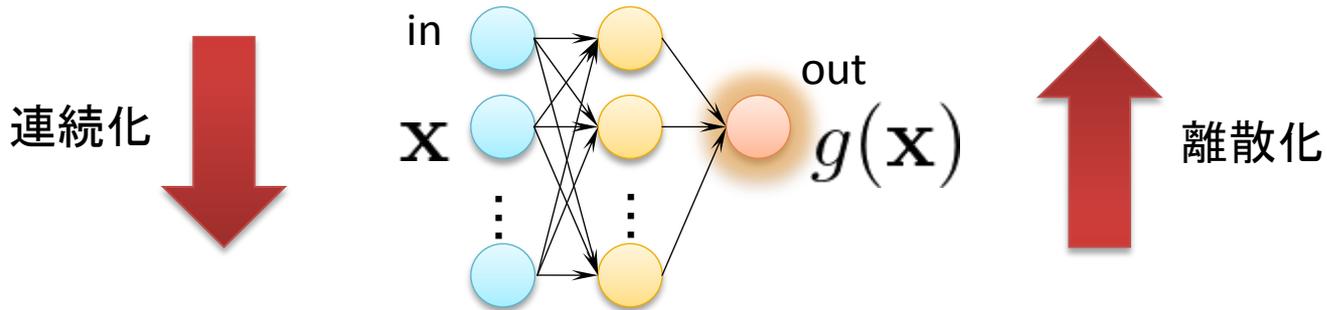
cf. [Sonoda&Murata2015, arXiv: 1505.03654]

積分表現理論 (リッジレット解析)

(再掲)ニューラルネットの積分表現

(浅い)ニューラルネット

$$g_J(\mathbf{x}) = \frac{1}{J} \sum_j c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$



積分表現

$$g(\mathbf{x}) = \int_{\mathbb{R}^{m+1}} \mathsf{T}(\mathbf{a}, b) \eta(\mathbf{a} \cdot \mathbf{x} - b) \frac{d\mathbf{a}db}{\|\mathbf{a}\|}$$
$$= \mathcal{R}_\eta^\dagger \mathsf{T}(\mathbf{x}) \quad \text{双対リッジレット変換}$$

リッジレット変換

[Murata 96, Candes 98,
Rubin98, Kostadinova14]
[Sonoda&Murata15,preprint]

$f \in L^1(\mathbb{R}^m)$ 近似対象

$\psi \in \mathcal{S}(\mathbb{R})$ リッジレット

ユークリッド表示

$$\mathcal{R}_\psi f(\mathbf{a}, b) := \int_{\mathbb{R}^m} f(\mathbf{x}) \overline{\psi(\mathbf{a} \cdot \mathbf{x} - b)} \|\mathbf{a}\| d\mathbf{x}$$

極座標表示

$$\mathcal{R}_\psi f\left(\frac{\mathbf{u}}{\alpha}, \frac{\beta}{\alpha}\right) := \int_{\mathbb{R}} \mathbf{R}f(\mathbf{u}, p) \overline{\psi\left(\frac{p - \beta}{\alpha}\right)} \frac{1}{\alpha} dp,$$

ラドン領域での
ウェーブレット変換

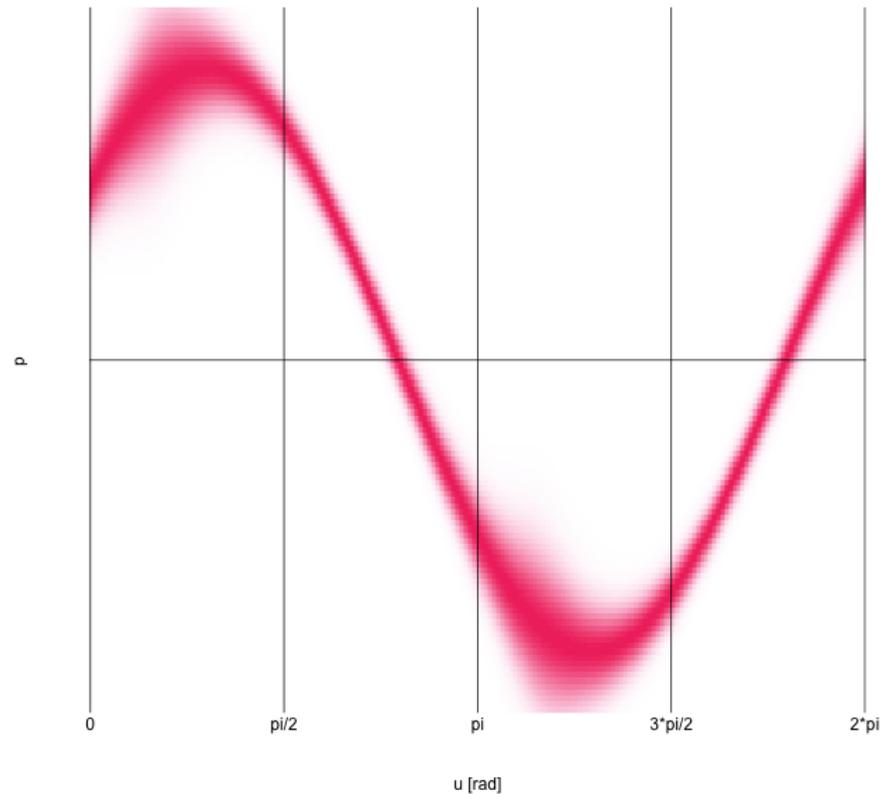
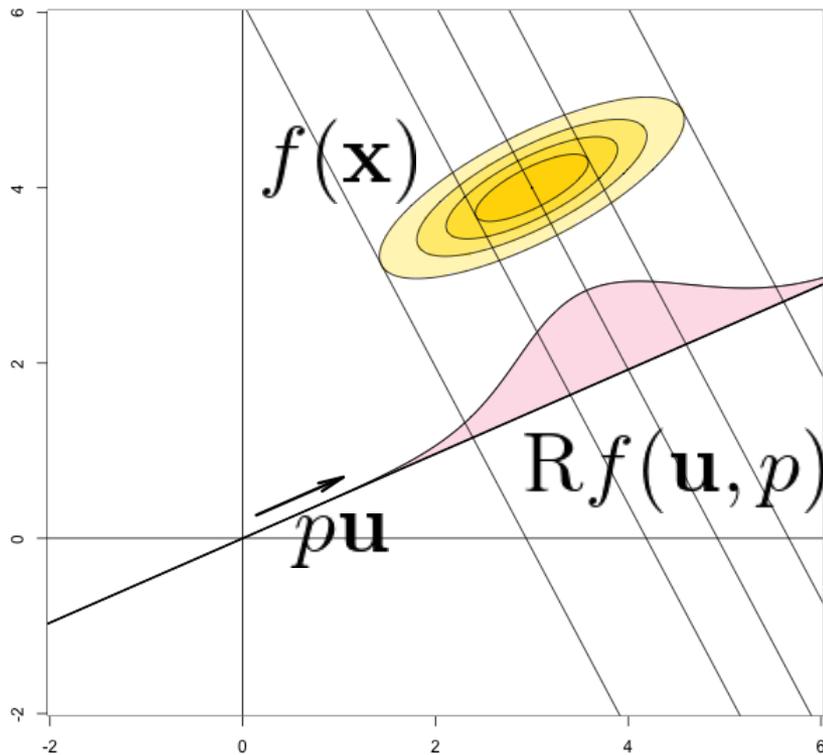
ただし、ラドン変換は

$$\mathbf{R}f(\mathbf{u}, p) := \int_{(\mathbb{R}\mathbf{u})^\perp} f(p\mathbf{u} + \mathbf{y}) d\mathbf{y}, \quad (\mathbf{u}, p) \in \mathbb{S}^{m-1} \times \mathbb{R}$$

ラドン変換 (CTスキャン, 周辺化)

$$Rf(\mathbf{u}, p) := \int_{(\mathbb{R}\mathbf{u})^\perp} f(p\mathbf{u} + \mathbf{y}) d\mathbf{y}, \quad (\mathbf{u}, p) \in \mathbb{S}^{m-1} \times \mathbb{R}$$

theta - phi = -4.28571



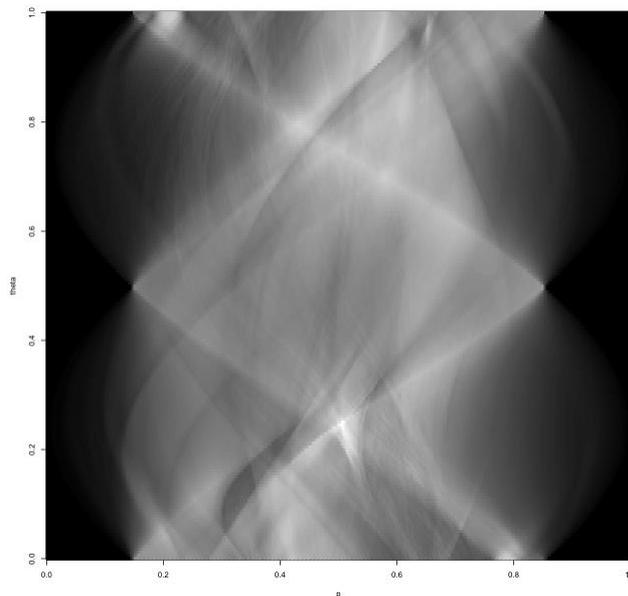
反転公式には逆投影フィルタが必要

$$\widehat{\Lambda\phi}(\zeta) = |\zeta|\widehat{\phi}(\zeta) \quad \text{逆投影フィルタ}$$

$f(\mathbf{x})$

$Rf(\mathbf{u}, p)$

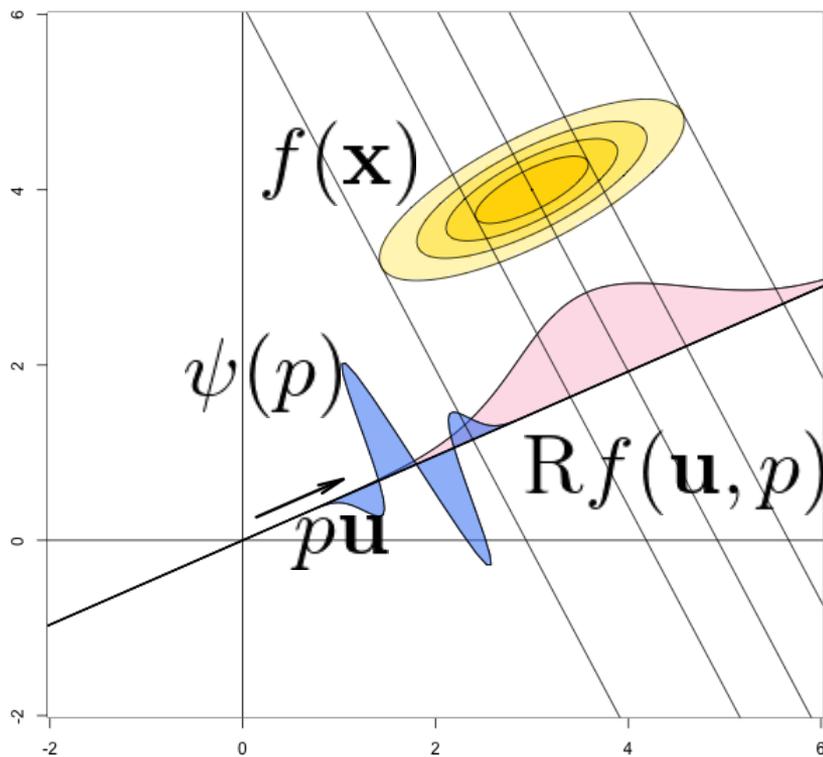
$R^\dagger \Lambda Rf = f$



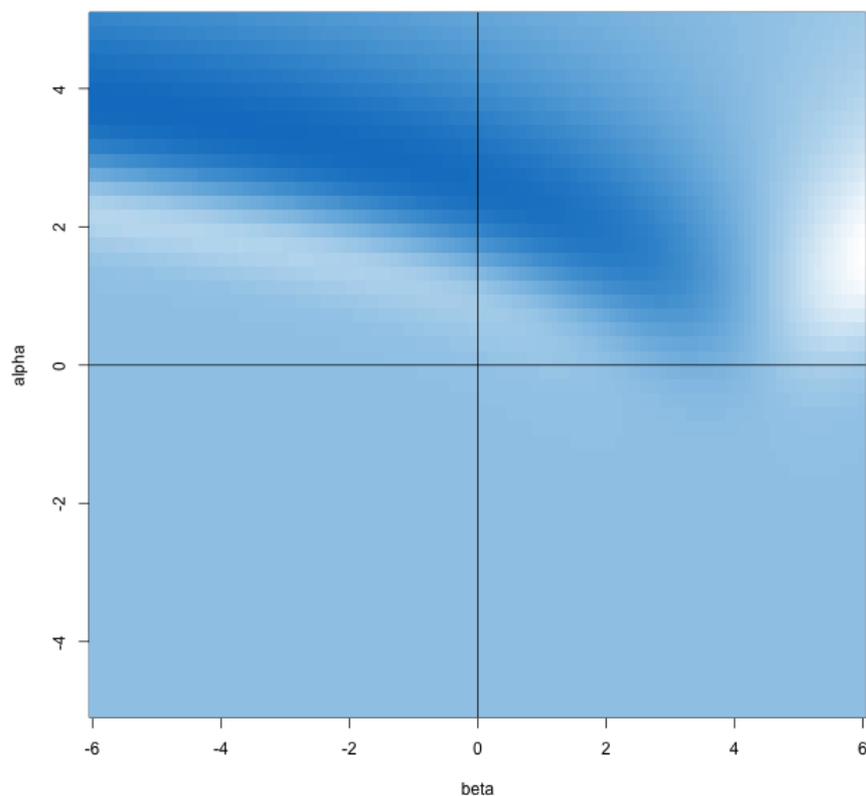
ラドン領域でのウェーブレット変換

$$\mathcal{R}_\psi f(\mathbf{u}, \alpha, \beta) := \int_{\mathbb{R}} \mathcal{R}f(\mathbf{u}, p) \psi\left(\frac{p - \beta}{\alpha}\right) \frac{1}{\alpha} dp,$$

theta - phi = -4.28571



theta = 25.7143, k = 4



双対リッジレット変換

$$T : \mathbb{S}^{m-1} \times \mathbb{H} \rightarrow \mathbb{C}$$

$\eta \in \mathcal{S}(\mathbb{R})$ (活性化関数)

ReLUを含む条件に拡張できる。

[Sonoda&Murata15]

$$\begin{aligned} \mathcal{R}_\eta^\dagger T(\mathbf{x}) &:= \int_{\mathbb{R}^{m+1}} T(\mathbf{a}, b) \eta(\mathbf{a} \cdot \mathbf{x} - b) \frac{d\mathbf{a}db}{\|\mathbf{a}\|} \\ &= \int_{\mathbb{S}^{m-1}} \left[\int_{\mathbb{H}} T(\mathbf{u}, \alpha, \beta) \eta\left(\frac{\mathbf{u} \cdot \mathbf{x} - \beta}{\alpha}\right) \frac{d\beta d\alpha}{\alpha^{m+1}} \right] d\mathbf{u} \end{aligned}$$

$$\forall \mathbf{x} \in \mathbb{R}^m$$

双対ウェーブレット変換
の双対ラドン変換

許容条件のもとで再生公式

$$\int_{\mathbb{R}} \frac{\overline{\widehat{\psi}(\zeta)} \widehat{\eta}(\zeta)}{|\zeta|^m} d\zeta = 1 \quad \Rightarrow \quad \mathcal{R}_\eta^\dagger \mathcal{R}_\psi f = f, \quad \text{a.e.}$$

許容条件の解釈

許容条件 \Rightarrow 再生公式

$$\int_{\mathbb{R}} \frac{\overline{\widehat{\psi}(\zeta)} \widehat{\eta}(\zeta)}{|\zeta|^m} d\zeta = 1 \quad \Rightarrow \quad \mathcal{R}_{\eta}^{\dagger} \mathcal{R}_{\psi} f = f, \quad \text{a.e.}$$

ラドン変換の反転公式

$$\mathbf{R}^{\dagger} \Lambda^{m-1} \mathbf{R} f = f$$

$$\widehat{\Lambda} \phi(\zeta) = |\zeta| \widehat{\phi}(\zeta) \quad (\text{逆投影フィルタ})$$

リッジレット変換の関係式

$$\mathcal{R}_{\eta}^{\dagger} \mathcal{R}_{\psi} f = \mathbf{R}^{\dagger} \mathcal{W}_{\eta}^{\dagger} \mathcal{W}_{\psi} \mathbf{R} f$$

つまり, 許容条件 \Rightarrow 逆投影フィルタ

$$\mathcal{W}_{\eta}^{\dagger} \mathcal{W}_{\psi} = \Lambda^{m-1}$$

積分表現とリッジレット解析の対応

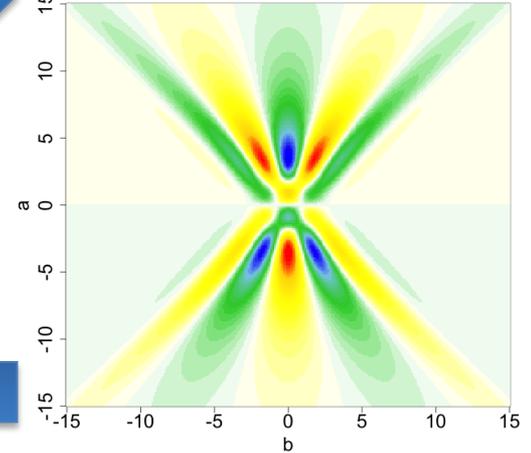
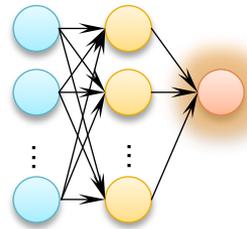
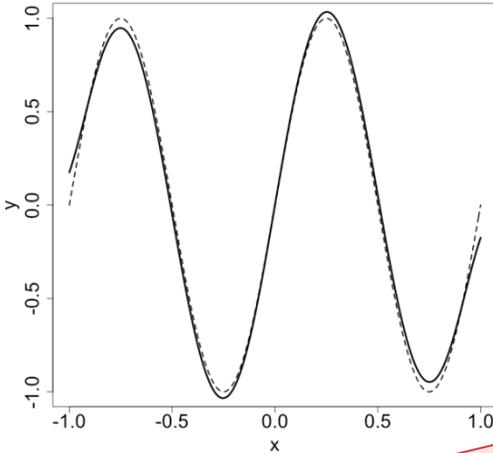
近似対象

$$f(\mathbf{x}) \in L^1(\mathbb{R}^m)$$

$$\mathcal{R}_\psi f \left(\frac{\mathbf{u}}{\alpha}, \frac{\beta}{\alpha} \right) := \int_{\mathbb{R}} \mathcal{R}f(\mathbf{u}, p) \psi \left(\frac{p - \beta}{\alpha} \right) \frac{1}{\alpha} dp,$$

中間層パラメタ = リッジレット変換

$$T(\mathbf{a}, b) \in \mathcal{S}'(\mathbb{R}^m \times \mathbb{H})$$



積分表現 = 双対リッジレット変換

$$\mathcal{R}_\eta^\dagger T(\mathbf{x}) := \int_{\mathbb{R}^{m+1}} T(\mathbf{a}, b) \eta(\mathbf{a} \cdot \mathbf{x} - b) \frac{d\mathbf{a} db}{\|\mathbf{a}\|}$$

再生公式
= 万能近似能力

離散化

連続化

ニューラルネット

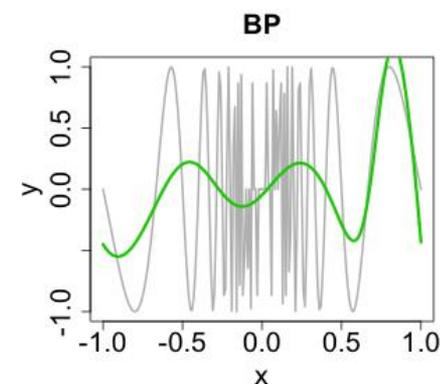
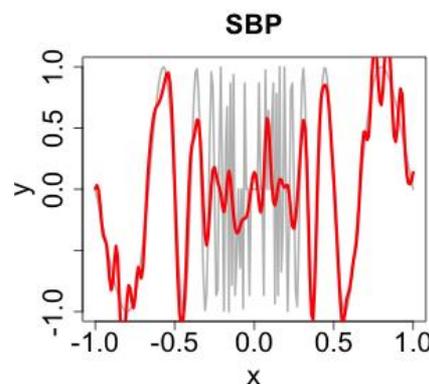
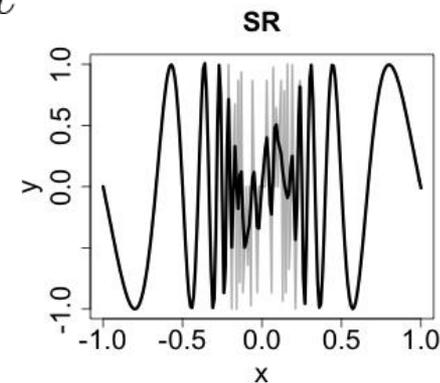
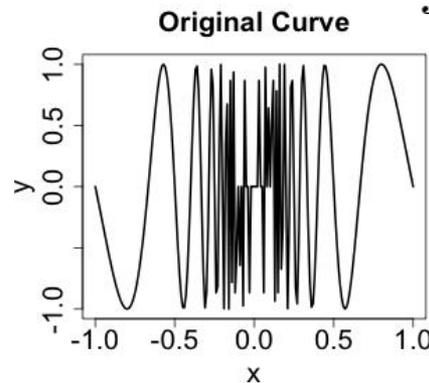
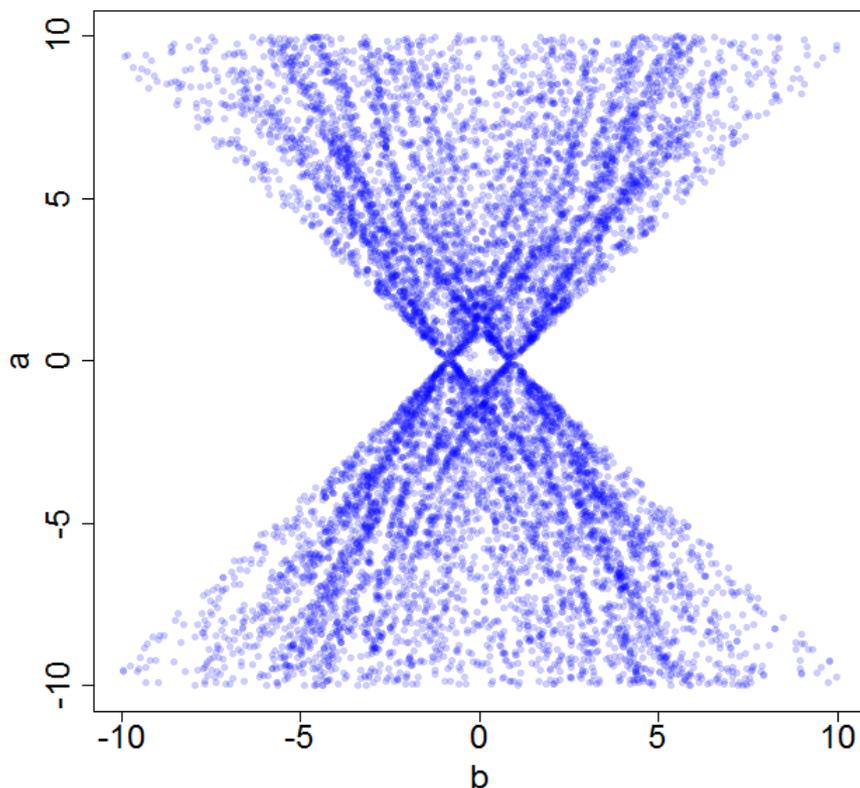
$$g_J(\mathbf{x}) = \frac{1}{J} \sum_j^J c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

応用例：オラクル分布

[Sonoda&Murata14]

リッジレット変換を離散化して学習する方法
(ランダム初期化より有利)

$$f(x) = \sin \frac{2\pi}{x}, \quad x \in [-1, 1]$$



$$g_J(\mathbf{x}) = \frac{1}{J} \sum_j c_j \cdot \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)$$

離散化の考え方

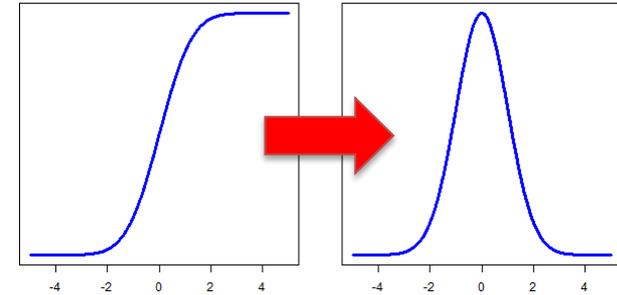
再生公式を(適当な方法で)数値積分する

$$\begin{aligned}\mathcal{R}_\eta^\dagger \mathcal{R}_\psi f(\mathbf{x}) &= \int \left[\int f(\mathbf{x}') \overline{\psi(\mathbf{a} \cdot \mathbf{x}' - b)} d\mathbf{x}' \right] \eta(\mathbf{a} \cdot \mathbf{x} - b) d\mathbf{a} db \\ &\approx \sum_j \left[\sum_n f(\mathbf{x}_n) \overline{\psi(\mathbf{a}_j \cdot \mathbf{x}_n - b_j)} \right] \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j)\end{aligned}$$

リッジレット関数の求め方

活性化関数 η の微分が急減少関数になる場合

$$\exists k \text{ s.t. } \eta^{(k)} \in \mathcal{S}(\mathbb{R})$$



次の内積が非零になる関数 $\psi_0 \in \mathcal{S}(\mathbb{R})$ を探す

$$\int \overline{\widehat{\psi_0}(\zeta)} \widehat{\eta^{(k)}}(\zeta) d\zeta \neq 0,$$

このとき以下は許容条件 (の定数倍) を満たす

$$\psi := \Lambda^m \psi_0^{(k)}, \quad \eta = \eta$$

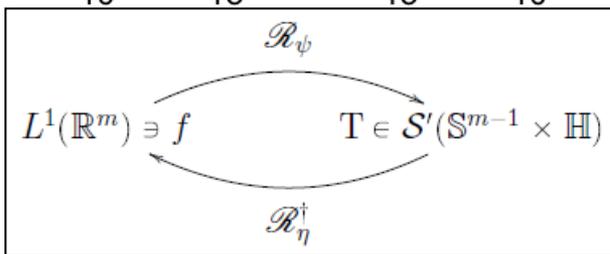
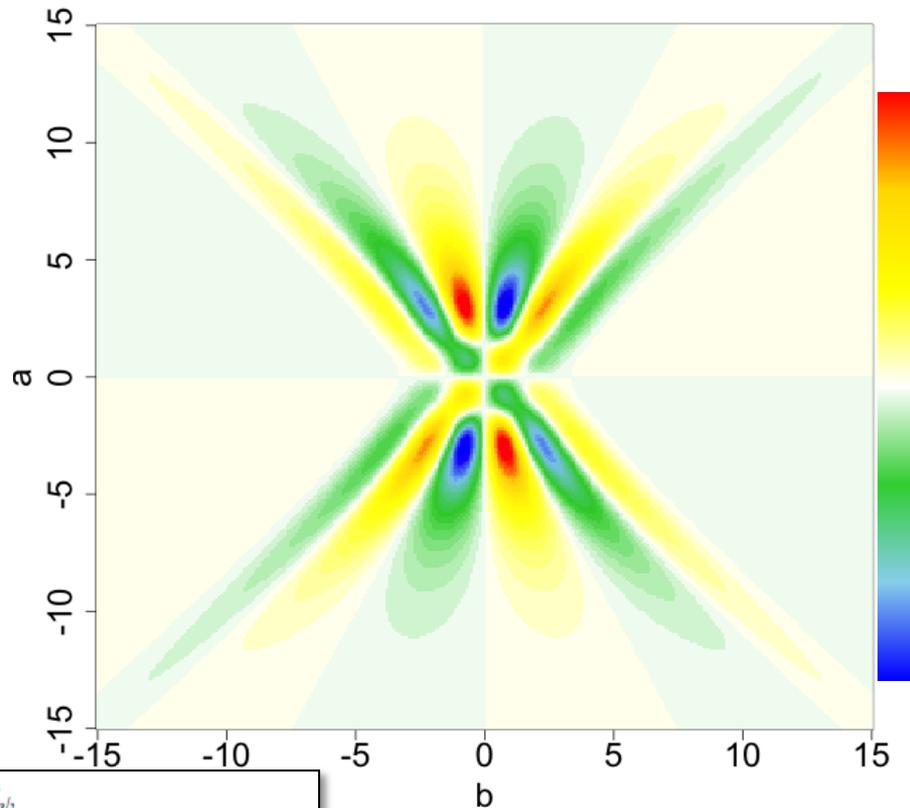
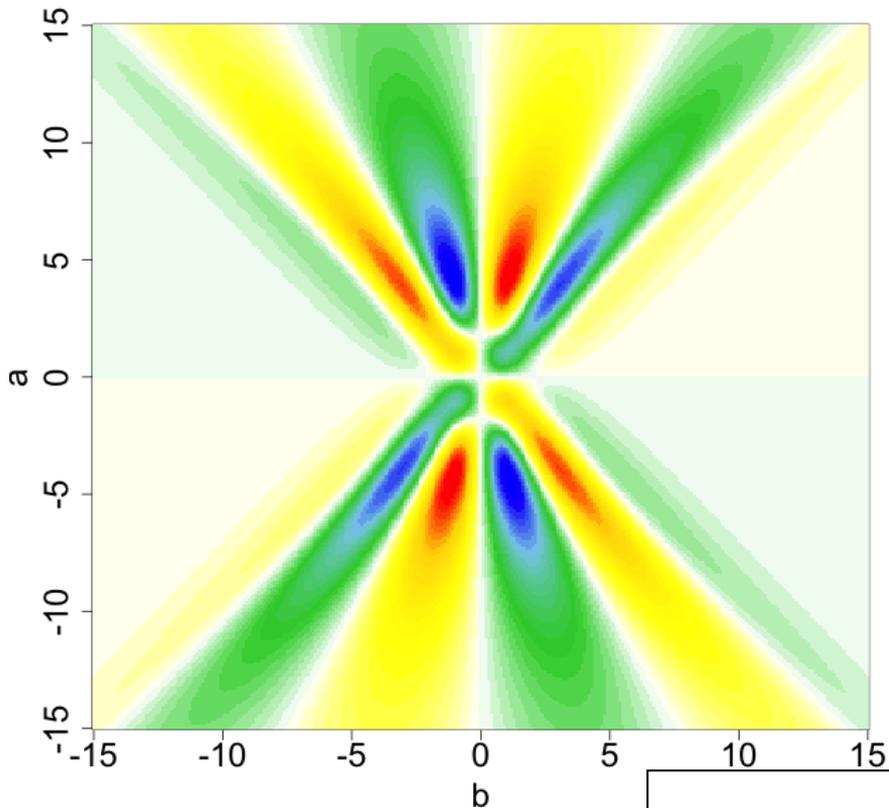
$$\therefore \int \frac{\overline{\widehat{\psi}(\zeta)} \widehat{\eta}(\zeta)}{|\zeta|^m} d\zeta \propto \int \frac{|\zeta|^m \cdot \overline{\widehat{\psi_0}(\zeta)} \widehat{\eta^{(k)}}(\zeta)}{|\zeta|^m} d\zeta \neq 0$$

許容リッジレットは無数にある

(バックプロパゲーションの最適解が複数あることに対応する)

$$\mathcal{R}_\psi f, \psi(z) = \mathcal{H} \frac{d}{dz} e^{-z^2}$$

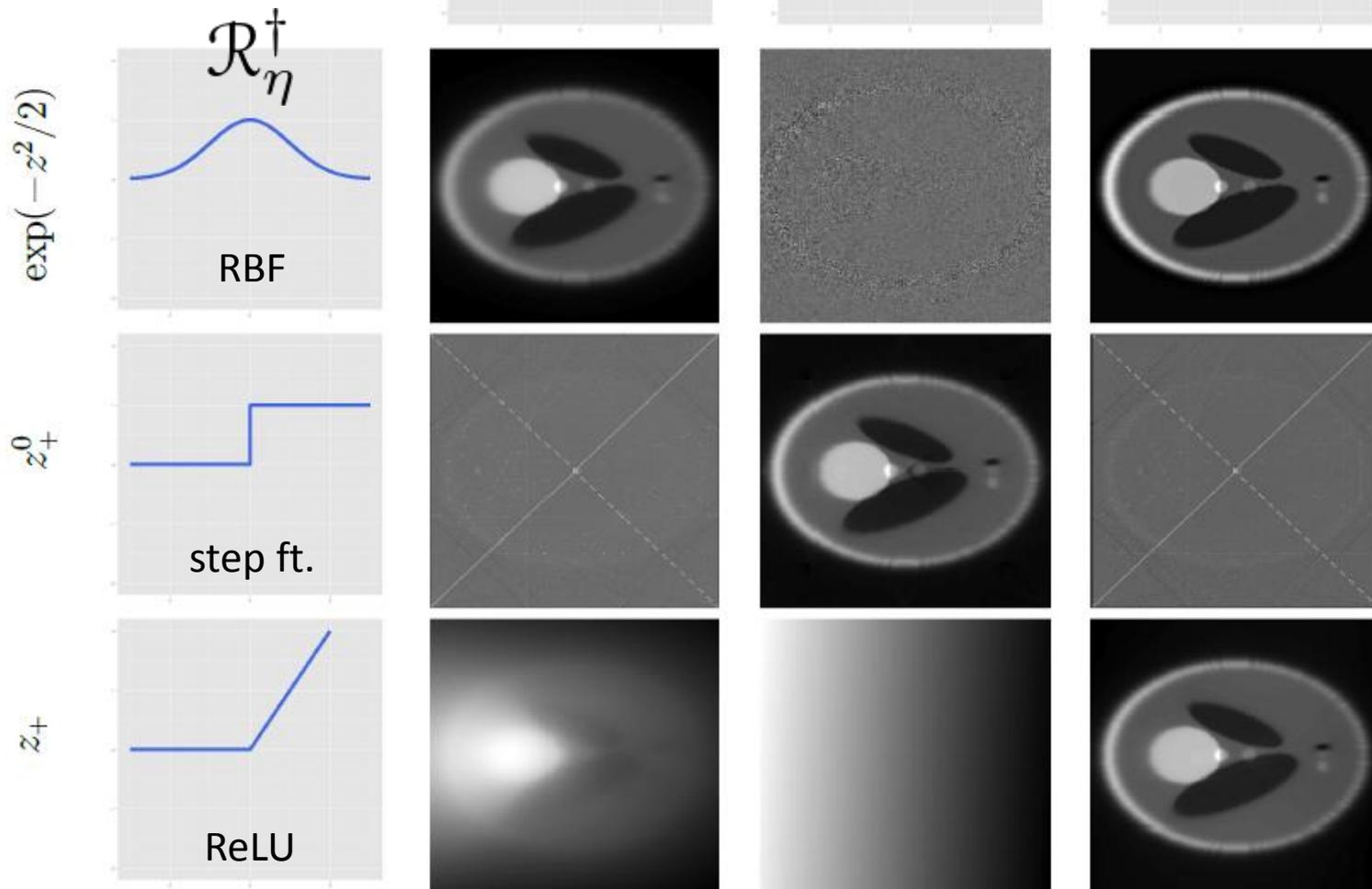
$$\mathcal{R}_\psi f, \psi(z) = \mathcal{H} \frac{d^3}{d^3 z} e^{-z^2}$$



[Sonoda&Murata15, preprint]

二次元信号の近似例

許容条件を満たす組合せ
であれば近似できる



最良のリッジレットは？

離散化に伴う近似誤差評価 [Murata96; Kurkova12]

$$\|f - [\mathcal{R}_\eta^\dagger \mathcal{R}_\psi f]_J\|_2 \leq \frac{1}{J} \|\eta\|_\infty \cdot \|\mathcal{R}_\psi f\|_1$$

(中間層素子J個で離散化)

最良のリッジレットを求める問題

$$\underset{\psi \in \mathcal{S}(\mathbb{R})}{\text{minimize}} \|\mathcal{R}_\psi f\|_1 \text{ s.t. } (\psi, \eta) \text{ admissible.}$$

リッジレット変換 = ニューラルネットの係数 なので,

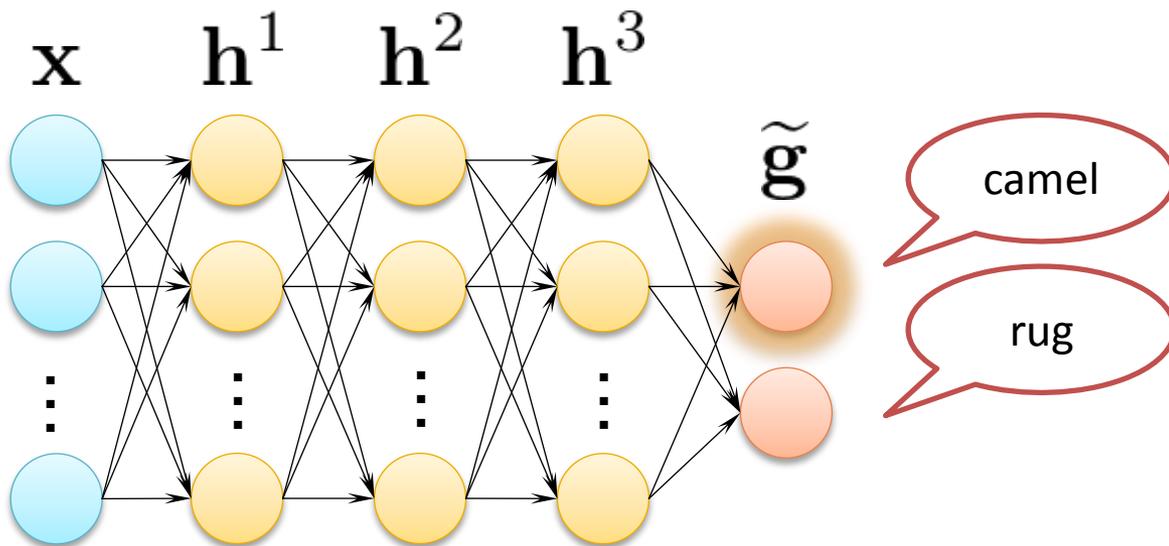
L1正則化付き最小二乗法

$$\underset{(\mathbf{a}_j, b_j, c_j)}{\text{minimize}} \left\| f(\mathbf{x}) - \sum_{j=1}^J c_j \eta(\mathbf{a}_j \cdot \mathbf{x} - b_j) \right\|_2 + \lambda \|\mathbf{c}\|_1$$

深層理論について

深層構造の中では何が起きているのか？

$$g(\mathbf{x}) = \tilde{g}(h^3(h^2(h^1(\mathbf{x}))))$$



画素値
 $\mathbf{x} \in \mathbb{R}^m$

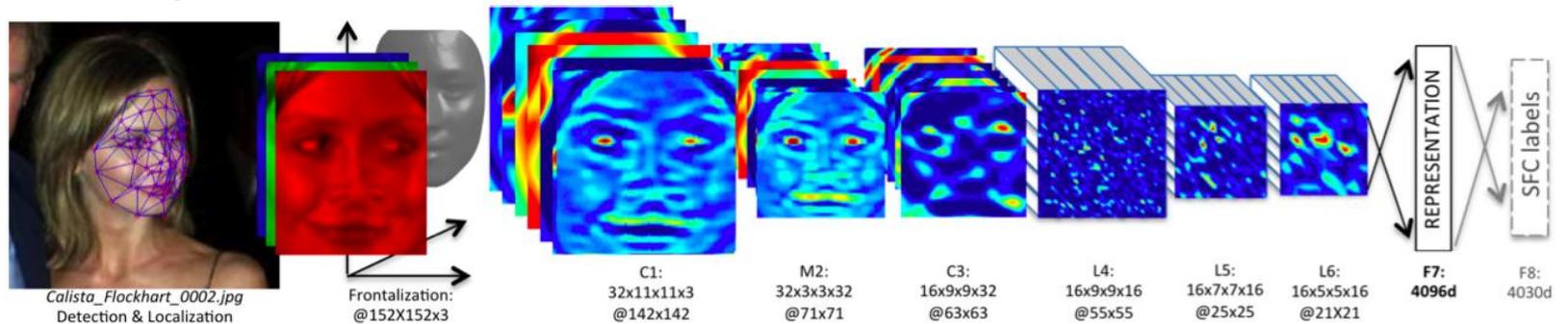
情報変換

ラベルの確率値
 $\mathbf{y} \in \mathbb{R}^k$

パラメータ数
~10億

深い方が効率的 vs. 浅くても万能

典型的な深いモデル



[Y.Taigman+14]

徐々に抽象化する

従来の浅いモデル



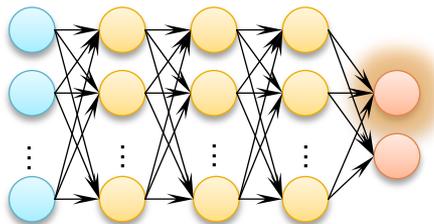
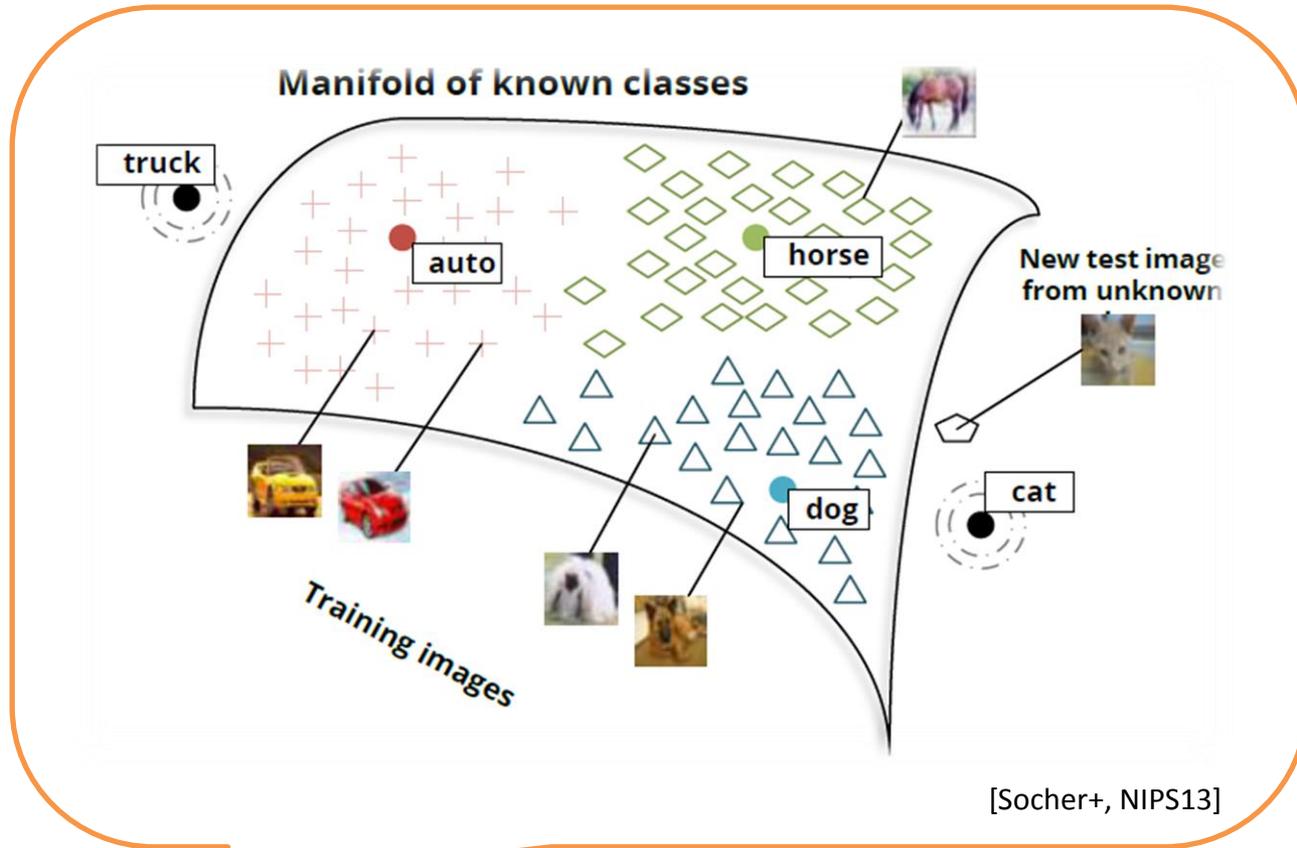
MNIST

急速にコードする

各層の Complexity が低いので、
トータルのパラメタ数は少ない。

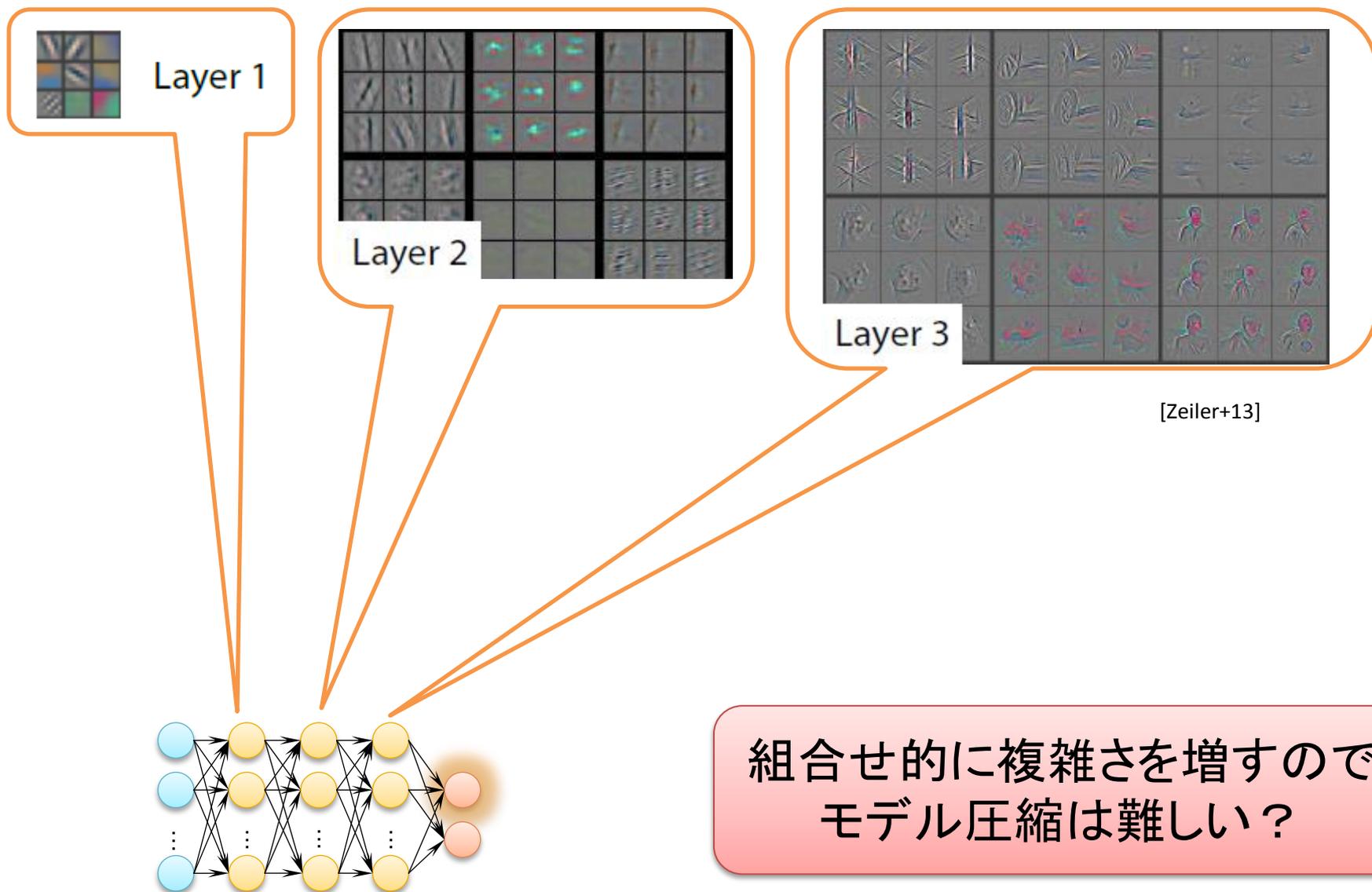
深層学習後の内部表現を
浅い構造で再学習して、
モデルを圧縮すればいい。

識別しやすいように再配置している説



再配置の経路は関係ないので
モデル圧縮可能と考えられる

意味の階層を構築している説



組合せ的に複雑さを増すので
モデル圧縮は難しい？

諸説

階層化すると効率的である

知識の階層 [Hinton89]

タスクの難しさ Type-1,2 [Clarck97]

回路的評価 [Hastad]

モデル圧縮, VC評価 [Utgoff+02]

教師なしの解析

良い事前分布になっている [Erhan&Bengio+10]

KPCAの繰返し [Montavon+10][Cho12]

逆拡散 [Gangulli+15]

Sample complexity評価 [Arora+14]

ランダム初期化+Deep構造の解析

線形deep で解析[Saxe&Gangulli14]

Deep GP は× [Duvenaud&Ghahramani+14]

ReLU+圧縮センシングは○ [Girye+15]

浅くて十分

Deep特徴量を再学習 [Ba+14]

ReLUの効果

maxout > relu > tanh [Goodfellow+13]

Dropoutの効果

モデルの幾何平均 [Baldi]

MAP [Srivastava+13]

ConvNet は変換不変性で説明できる

視覚野 → Derived kernel [Smale+09]

Scattering 変換 [Bruna&Mallat13]

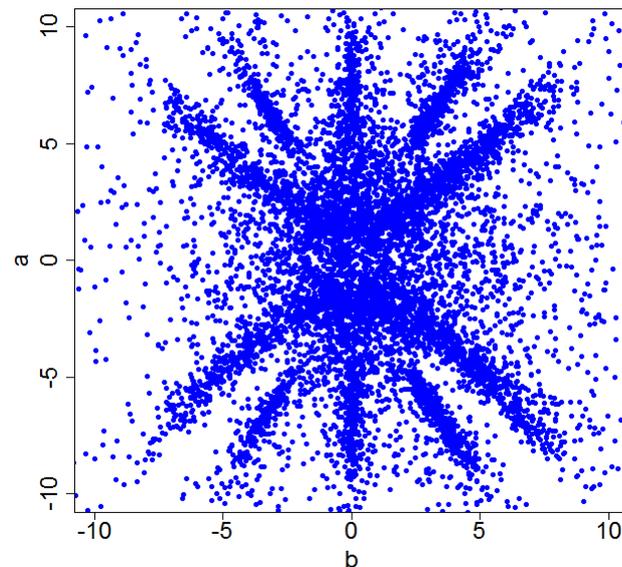
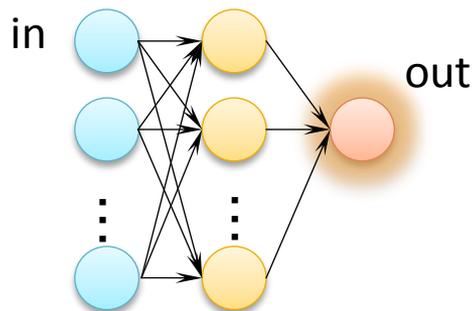
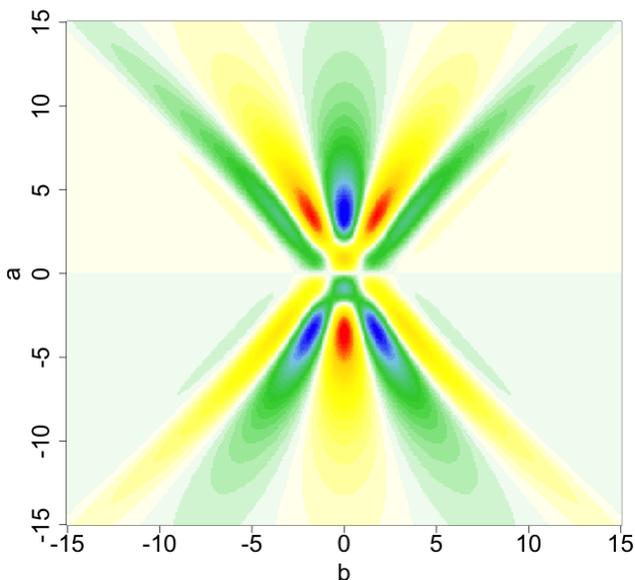
深層化=不変群の分解 [Rosasco&Poggio+14]

画像生成モデル+Bayes = CNN[Patel+15]

積分表現にすると中間層が見える

パラメータ数10億といえども, リッジレット変換の離散化

$$g(\mathbf{x}) = \int_{\mathbb{R}^{m+1}} T(\mathbf{a}, b) \eta(\mathbf{a} \cdot \mathbf{x} - b) \frac{d\mathbf{a}db}{\|\mathbf{a}\|}$$



中間層の情報表現は, **係数Tのみ**が担う。

$$T := \mathcal{R}_\psi f \Rightarrow g = f$$

まとめ

深層構造がなぜ、優れた性能を発揮するのか調べたい

(万能であるはずの)浅いモデルとの本質的な違いは何か？

積分表現理論(リッジレット解析)から切り込む

ニューラルネットの積分表現 = 双対リッジレット変換

リッジレット変換 = 中間層パラメータの極限分布

= ラドン変換とウェーブレット変換の合成

再生公式の離散化 = 学習に応用できる(オラクル分布)

本日のメッセージ

パラメータ数が10億といえども、リッジレット変換の離散化