学習における特異点構造の分析について

On the analysis of singularity structure in learning

甲南大学大学院 自然科学研究科 知能情報学専攻　　鷲野朋広 *1
Tomohiro Washino
Graduate School of Natural Science, Konan University

甲南大学 知能情報学部 知能情報学科　　高橋正 *2
Tadashi Takahashi
Department of Intelligence and Informatics, Konan University

Abstract

3 層パーセプトロンである学習モデルの中間ユニット数が 2 から 1 に変化する特異領域の近くでは学習が停滞するプラトーが起きる [1]. Guo らは特異領域の近くの学習のダイナミクスを 5 つのパターンに分けて考察している [2]. 学習モデルの座標系を変換することにより, 変化の速いパラメータを固定して変化の遅いパラメータの軌道を調べる [3]. 甘利らによって特異領域の近くの学習の安定性やダイナミクスが求められている [1], [4]. 第 1 に学習モデルとして Mathmatica においてニューラルネットワークを作成する. 第 2 に特異領域の近くにある真の分布が学習モデルによって実現される場合, 学習モデルの初期値を変えることによって臨界直線に影響を受けて変化するダイナミクスについて考察する. 第 3 に特異領域にある真の分布が学習モデルによって実現される場合, 学習損失が減少するが, 汎化損失が増加する過学習が起き, 真の分布が学習モデルによって実現されない場合, 学習の初期に汎化損失が学習損失より減少して小さくなり, 増加して大きくなる過剰般化が起きる. 過学習, 過剰汎化についてパラメータの変化を調べ, 学習・汎化損失曲面上のダイナミクスの考察をする.

Abstract

Let a statistical model be a three-layer neural network. plateau phenomena have been observed in singular regions where two hidden neurons can be rewritten with only one hidden neuron. Guo classify dynamics of learning near singularity into five patterns. We introduce coordinate transformation of parameters of the statistic model. Let us fix variables moving quickly and search trajectories of learning of variables moving slowly. Amari calucalated stability and dynamics of learning near singular regions. Firstly we constract a neural network as the statistical model by using Mathmatica. Secondly we change an initial value of the statistic model and consider that the dynamics of learning changing under the influence of a critical line in case the true distribution near the singular regions is realizable by the statistical model. Thirdly over-fitting phenomena that a training loss decreases but a validation loss increases have been observed in case the true distribution on the singularity is realizable by the statistical model. In the early process of learning, over-generalization phenomena that the validation loss decreases and becames smaller and increases and becames bigger than training loss have been observed in case the true distribution is outside of the statistic model, We examine the evolution of parameter and consider the dynamics of learning on the training loss surface and validation loss surface.

*1 〒 658-8501 神戸市東灘区岡本 8-9-1　E-mail: d1523001@s.konan-u.ac.jp
*2 〒 658-8501 神戸市東灘区岡本 8-9-1　E-mail: takahasi@konan-u.ac.jp

1　定義

定義 1 (入力, 雑音, 訓練データ, テストデータ)
$\mathbb{R}^1$ 上の一様分布に従う確率変数 X を入力とする. 平均 0, 分散 σ の $\mathbb{R}^1$ 上の正規分布に従う確率変数 Z を雑音とする. $\theta_{\mathbf{0}} = (w^*_{11}, w^*_{12}, w^*_{21}, w^*_{22}, w^*_{31}, w^*_{32}) \in \mathbb{R}^6$ に対して, 次で定まる $\mathbb{R}^1$ 上の確率変数 Y を訓練データ, テストデータとする.

$$Y := f(x, \theta_0) + Z = w^*_{31} \tanh(w^*_{11}x + w^*_{21}) + w^*_{32} \tanh(w^*_{12}x + w^*_{22}) + Z.$$

定義 2 (関数近似モデル)
パラメータ $\theta = (\mathbf{w_1}, \mathbf{w_2}, w_{31}, w_{32}) \in \mathbb{R}^6$, $\mathbb{R}^1$ への関数 $f(x, \theta)$ に対して, 次で定まる $\mathbb{R}^1$ 上の確率変数 Y を関数近似モデルという.

$$\begin{aligned} Y := f(x, \theta) + Z &= w_{31}\phi(\mathbf{x}, \mathbf{w_1}) + w_{32}\phi(\mathbf{x}, \mathbf{w_2}) + Z = w_{31} \tanh({\mathbf{w_1}}^{\mathrm{T}}\mathbf{x}) + w_{32} \tanh({\mathbf{w_2}}^{\mathrm{T}}\mathbf{x}) + Z \\ &= w_{31} \tanh(w_{11}x + w_{21}) + w_{32} \tanh(w_{12}x + w_{22}) + Z. \end{aligned}$$

ここで, $\mathbf{w_1} = (w_{11}, w_{12})$, $\mathbf{w_2} = (w_{21}, w_{22})$, $\mathbf{x} = (x, 1)$ とする.

定義 3 (学習モデル, 真の分布)
関数近似モデル Y が従う条件付き確率を次で定め, 学習モデルという.

$$p(y|x, \theta) := \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{|y - f(x, \theta)|^2}{2\sigma^2}\right).$$

出力 Y が従う条件付き確率を次で定め, 真の分布という.

$$q(y|x) := \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{|y - f(x, \theta_0)|^2}{2\sigma^2}\right).$$

定義 4 (Overlap singularity, Elimination singularity)
次で定めるパラメータ領域を *Overlap singularity* という *[1].*

$$R_0 := \{\theta \in \mathbb{R}^6 | \mathbf{w_1} = \mathbf{w_2}\}.$$

次で定めるパラメータ領域を *Elimination singularity* という *[1].*

$$R_1 := \{\theta \in \mathbb{R}^6 | w_{31} = 0\} \cup \{\theta \in \mathbb{R}^6 | w_{32} = 0\}.$$

関数近似モデルは Overlap singularity では $f(x, \theta) = (w_{31} + w_{32})\phi(\mathbf{x}, \mathbf{w_1})$ と表され, Elimination singularity では $f(x, \theta) = w_{32}\phi(\mathbf{x}, \mathbf{w_2})$, $w_{31}\phi(\mathbf{x}, \mathbf{w_1})$ と表される.

座標系 $\theta = (\mathbf{w_1}, \mathbf{w_2}, w_{31}, w_{32})$ から座標系 $\xi = (\mathbf{a}, b, \mathbf{v}, w)$ への次の座標変換を考える [1].

$$\begin{aligned} \mathbf{a} &= \mathbf{w_2} - \mathbf{w_1}, \quad b = \frac{w_{31} - w_{32}}{w_{31} + w_{32}}, \\ \mathbf{v} &= \frac{w_{31}\mathbf{w_1} + w_{32}\mathbf{w_2}}{w_{31} + w_{32}}, \quad w = w_{31} + w_{32}. \end{aligned}$$

このとき座標系 θ は座標系 ξ を用いて次で表される.

$$\begin{aligned} \mathbf{w_1} &= \mathbf{v} + \frac{1}{2}\mathbf{a}(b - 1), \quad \mathbf{w_2} = \mathbf{v} + \frac{1}{2}\mathbf{a}(b + 1), \\ w_{31} &= \frac{1}{2}w(1 + b), \quad w_{32} = \frac{1}{2}w(1 - b). \end{aligned}$$

$y = f(x, \theta_0) + Z$ に対して, 誤差関数 $l(y, \mathbf{x}, \theta) := \frac{1}{2}(y - f(\mathbf{x}, \theta))^2$ と定める. このとき学習係数 η に対して, 次で定めて学習を更新する.

$$\theta(t+1) - \theta(t) := -\eta \frac{\partial l(y_t, \mathbf{x_t}, \theta_t)}{\partial \theta}.$$

定義 5 (座標系 θ の学習方程式)
座標系 $\theta = (\mathbf{w_1}, \mathbf{w_2}, w_{31}, w_{32})$ に対して, 学習方程式を次で定める.

$$\dot{\theta}(t) := -\eta \left\langle \frac{\partial l(y, \mathbf{x}, \theta)}{\partial \theta} \right\rangle.$$

ここで $\left\langle \frac{\partial l(y,\mathbf{x},\theta)}{\partial \theta} \right\rangle = \int \frac{\partial l(y,\mathbf{x},\theta)}{\partial \theta} q(y|\mathbf{x}) dy d\mathbf{x}$ とする.

誤差 $e(y, \mathbf{x}, \xi) := y - f(\mathbf{x}, \xi)$ に対して, 誤差関数 $l(\xi)$ の勾配について次が成り立つ [1].

$$\begin{aligned}
l_{\mathbf{v}}(\xi) &= w \left\langle e(y, \mathbf{x}, \xi) \frac{\partial \phi(\mathbf{x}, \mathbf{v})}{\partial \mathbf{v}} \right\rangle + \frac{1}{8} w (1 - z^2) Q(\mathbf{v}, \mathbf{a}) + O(\mathbf{a}^3), \\
l_w(\xi) &= \langle e(y, \mathbf{x}, \xi) \phi(\mathbf{x}, \mathbf{v}) \rangle + \frac{1}{8}(1 - z^2) \left\langle e(y, \mathbf{x}, \xi) \mathbf{a}^{\mathrm{T}} \frac{\partial^2 \phi(\mathbf{x}, \mathbf{v})}{\partial \mathbf{v} \partial \mathbf{v}^{\mathrm{T}}} \mathbf{a} \right\rangle + O(\mathbf{a}^3), \\
l_{\mathbf{a}}(\xi) &= \frac{1}{4} w (1 - z^2) \left\langle e(y, \mathbf{x}, \xi) \mathbf{a} \frac{\partial^2 \phi(\mathbf{x}, \mathbf{v})}{\partial \mathbf{v} \partial \mathbf{v}^{\mathrm{T}}} \right\rangle + \frac{1}{24} w z (1 - z^2) \left\langle e(y, \mathbf{x}, \xi) \frac{\partial D(x, \mathbf{v}, \mathbf{a})}{\partial \mathbf{a}} \right\rangle + O(\mathbf{a}^3), \\
l_b(\xi) &= -\frac{1}{4} w z \left\langle e(y, \mathbf{x}, \xi) \mathbf{a}^{\mathrm{T}} \frac{\partial^2 \phi(\mathbf{x}, \mathbf{v})}{\partial \mathbf{v} \partial \mathbf{v}^{\mathrm{T}}} \mathbf{a} \right\rangle + O(\mathbf{a}^3).
\end{aligned}$$

ここで $Q(v, \mathbf{a}) := \left\langle e(y, \mathbf{x}, \xi) \frac{\partial}{\partial \mathbf{v}} (\mathbf{a}^{\mathrm{T}} \frac{\partial^2 \phi(\mathbf{x}, \mathbf{v})}{\partial \mathbf{v} \partial \mathbf{v}^{\mathrm{T}}} \mathbf{a}) \right\rangle$, $D(\mathbf{x}, \mathbf{v}, \mathbf{a}) := \sum_{i,\, j,\, k} \frac{\partial^3 \phi(\mathbf{x}, \mathbf{v})}{\partial v_i \partial v_j \partial v_k} a_i a_j a_k$ と定める.
$\mathbf{a} \approx \mathbf{0}$ のとき $l_{\mathbf{v}}$, l_w は $O(1)$ のオーダーよりパラメータ $(\mathbf{v}, w)$ の変化は速く, $l_{\mathbf{v}}(\xi) = l_w(\xi) = 0$ である平衡安定状態になる. 一方で $l_{\mathbf{a}}$ は $O(\mathbf{a})$ のオーダー, l_b は $O(\mathbf{a}^2)$ のオーダーよりパラメータ $(\mathbf{a}, b)$ は遅く変化する [3].

定義 6 (座標系 ξ の学習方程式)
座標系 $\xi = (\mathbf{a}, b, \mathbf{v}, w)$ に対して学習方程式を次で定める.

$$\dot{\xi} := -\eta \frac{\partial \xi}{\partial \theta^{\mathrm{T}}} \left(\frac{\partial \xi}{\partial \theta^{\mathrm{T}}} \right)^{\mathrm{T}} \left\langle \frac{\partial l(y, \mathbf{x}, \xi)}{\partial \xi} \right\rangle.$$

このとき学習方程式について次が成り立つ [1].

$$\begin{aligned}
\dot{\mathbf{v}} &= \frac{b^2 + 1}{2} l_{\mathbf{v}} + \frac{b^2 + 1}{2w^2} \mathbf{a}\mathbf{a}^{\mathrm{T}} l_{\mathbf{v}} + \frac{b}{w} \mathbf{a} l_w - b l_{\mathbf{a}} - \frac{b^2 + 1}{w^2} \mathbf{a} l_b, \\
\dot{w} &= \frac{b}{w} \mathbf{a}^{\mathrm{T}} l_{\mathbf{v}} + 2 l_w - \frac{2b}{w} l_b, \\
\dot{\mathbf{a}} &= -b l_{\mathbf{v}} + 2 l_{\mathbf{a}}, \\
\dot{b} &= -\frac{b^2 + 1}{w^2} \mathbf{a}^{\mathrm{T}} l_{\mathbf{v}} - \frac{2b}{w} l_w + \frac{2(b^2 + 1)}{w^2} l_b.
\end{aligned}$$

ここでパラメータ $(\mathbf{v}, w)$ を最適解 $(\mathbf{v}^*, w^*)$ に固定して, パラメータ $(\mathbf{a}, b)$ の変化を分析する.
$\xi^* = (\mathbf{v}^*, w^*, \mathbf{0}, b)$ に対して, $H(\mathbf{v}^*, w^*) := \frac{1}{4} w^* \left\langle e(y, \mathbf{x}, \xi) \frac{\partial^2 \phi(\mathbf{x}, \mathbf{v})}{\partial \mathbf{v} \partial \mathbf{v}^{\mathrm{T}}} \right\rangle \Big|_{\xi = \xi^*}$ と定める.

定理 7 (特異領域の近くの学習の安定性)
誤差関数 $l(y, \mathbf{x}, \xi)$ に対して, $\left\langle \frac{\partial^2 l(y,\mathbf{x},\xi)}{\partial \xi \partial \xi^{\mathrm{T}}} \right\rangle \Big|_{\xi=\xi^*} = (1-b^2)H(\mathbf{v}^*, w^*)$ が成り立ち, 特異領域の近くの学習の安定性について次が成り立つ.
真の分布が特異領域にあるとき R_0 は安定である. 真の分布が特異点上にないとき R_0 の安定性は $H(\mathbf{v}^*, w^*)$ の固有値によって次の *3* つの場合に分けられる *[4].*
(1) 固有値が正と負の値を持つ場合*:* R_0 は不安定.
(2) 固有値が *2* つとも負の場合*:* R_0 の中で $b^2 < 1$ を満たす部分が安定, $b^2 > 1$ を満たす部分が不安定.
(3) 固有値が *2* つとも正の場合*:* R_0 の中で $b^2 > 1$ を満たす部分が安定, $b^2 < 1$ を満たす部分が不安定.

$\tilde{\xi} = (\mathbf{v}^*, w^*, \mathbf{a}, b)$ として, 誤差関数 $l(\tilde{\xi})$ の勾配について次が成り立つ [1].

$$
\begin{aligned}
l_{\mathbf{v}}(\tilde{\xi}) &= \frac{1}{8} w^* (1-z^2) Q(\mathbf{v}^*, \mathbf{a}) + O(\mathbf{a}^3), \\
l_w(\tilde{\xi}) &= \frac{1}{2} \frac{1-z^2}{w^*} \mathbf{a}^{\mathrm{T}} H(\mathbf{v}^*, w^*) \mathbf{a} + O(\mathbf{a}^3), \\
l_{\mathbf{a}}(\tilde{\xi}) &= (1-z^2) H(\mathbf{v}^*, w^*) \mathbf{a} + \frac{1}{24} w^* z (1-z^2) \left\langle e(y, \mathbf{x}, \xi) \frac{\partial D(x, \mathbf{v}, \mathbf{a})}{\partial \mathbf{a}} \right\rangle \Big|_{\xi=\tilde{\xi}} + O(\mathbf{a}^3), \\
l_b(\tilde{\xi}) &= -b \mathbf{a}^{\mathrm{T}} H(\mathbf{v}^*, w^*) \mathbf{a} + O(\mathbf{a}^3).
\end{aligned}
$$

$\mathbf{a} \approx \mathbf{0}$ のとき $l_{\mathbf{a}}(\tilde{\xi})$ は $O(\mathbf{a})$ のオーダー, $l_b(\tilde{\xi})$, $l_{\mathbf{v}}(\tilde{\xi})$, $l_w(\tilde{\xi})$ は $O(\mathbf{a}^2)$ のオーダーより $O(\mathbf{a}^n)$ 高次の項を無視することによって R_0 の近傍での学習方程式について次が成り立つ [1].

$$
\begin{aligned}
\dot{\mathbf{a}} &= 2(1-b^2) H(\mathbf{v}^*, w^*) \mathbf{a}, \\
\dot{b} &= -\frac{b(1-b^2)}{w^{*2}} \mathbf{a}^{\mathrm{T}} H(\mathbf{v}^*, w^*) \mathbf{a} - \frac{2b(b^2+1)}{w^{*2}} \mathbf{a}^{\mathrm{T}} H(\mathbf{v}^*, w^*) \mathbf{a}.
\end{aligned}
$$

定理 8 (特異領域の近くの学習のダイナミクス)
エネルギー関数 $h(\mathbf{a}) := \frac{1}{2} \mathbf{a}^{\mathrm{T}} \mathbf{a}$ に対して, 特異領域の近くの学習のダイナミクスについて次が成り立つ *[1].*
(1) R_0 の近傍では $\dot{h} = \mathbf{a}^{\mathrm{T}} \dot{\mathbf{a}} = \frac{2w^{*2}(b^2-1)}{b(b^2+3)} \dot{b}$ が成り立ち, 学習方程式のダイナミクスは次の式で表される.

$$
h(\mathbf{a}) = \frac{2w^{*2}}{3} \log \frac{(b^2+3)^2}{|b|} + C.
$$

(2) $R_0 \cap R_1$ の近傍では $\dot{h} = \frac{w^{*2}(b^2-1)}{b(b^2+1)} \dot{b}$ が成り立ち, 学習方程式のダイナミクスは次の式で表される.

$$
h(\mathbf{a}) = w^{*2} \log \left(|b| + \frac{1}{|b|} \right) + C.
$$

定義 9 (特異領域の近くの学習のダイナミクスの分類)
学習を始めるパラメータの初期値により, 特異領域の近くの学習のダイナミクスを次の *5* つの場合に分ける *[2].*
(1) Overlap singularity: 特異領域 R_0 で学習が停滞して真の分布まで到達しない.
(2) Cross elimination singurarity: 特異領域 R_1 を横断するとき学習が停滞して真の分布に近づく.
(3) Fast convergence: 特異領域を通過せずに真の分布に速く収束する場合.
(4) Near elimination singularity: 特異領域 R_1 に近づき学習が停滞して真の分布に近づく.
(5) Output weight 0: 特異領域 R_1 で停滞して真の分布まで到達しない.

2　特異領域の近くにおける学習のダイナミクス

2.1　Mathmatica におけるニューラルネットワークの作成

変数 $F1$, $F2$, 定数 $elem0$, $elem1$, $elem2$, $elem3$ を次のように Mathmatica に入力して定める.

$F1[a\_] := NetInsertSharedArrays[NetChain[LinearLayer[1,"Weights"->a,"Biases"->None]],"Linear1"]$,
$F2[b\_] := NetInsertSharedArrays[NetChain[LinearLayer[1,"Weights"->b,"Biases"->None]],"Linear2"]$,
$elem0 := ElementwiseLayer[\#*(1/2)\&]$, $elem1 := ElementwiseLayer[\#*(-1)\&]$,
$elem2[v\_] := ElementwiseLayer[\#*(v)\&]$, $elem3[w\_] := ElementwiseLayer[\#*(w)\&]$.

初めに $w_{11}x = \left(v + \frac{1}{2}(b-1)a\right)x$ より, 次のように入力して図 1 の左側に $net11$ を出力させる.

$$net11[a\_, b\_, v\_] := NetGraph[elem0, elem1, F1[a], F2[b], elem2[v], TotalLayer[],$$
$$NetPort["Input"]->1, 1->3->4, 3->2, 4, 2, 5->6]$$

$w_{31}\tanh(x) = \frac{1}{2}w(b+1)\tanh(x)$ より, 次のように入力して図 1 の右側に $net12$ を出力させる.

$$net12[a\_, b\_, w\_] := NetGraph[Tanh, elem0, elem3[w], F2[b], TotalLayer[],$$
$$NetPort["Input"]->1, 1->2->3->4, 3, 4->5]$$

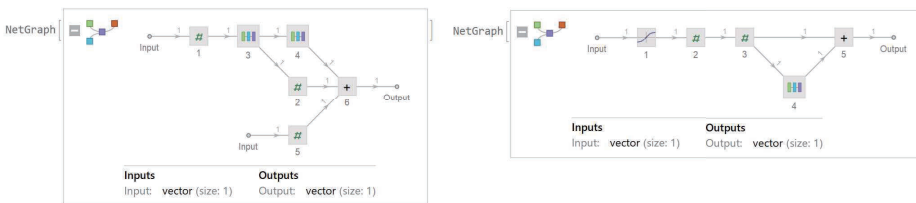

図 1: $net11$, $net12$

また $w_{12}x = \left(v + \frac{1}{2}(b+1)a\right)x$, $w_{32}\tanh(x) = \frac{1}{2}w(-b+1)\tanh(x)$ より, $net21$, $net22$ を同様に定めて図 2 に出力させる.

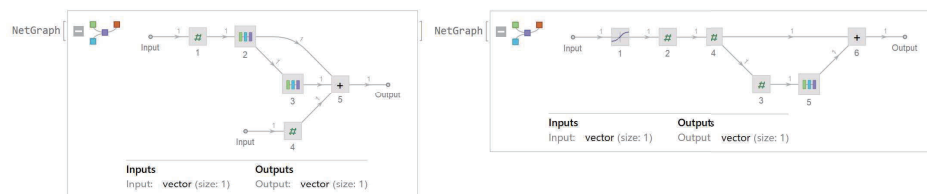

図 2: $net21$, $net22$

次に $w_{31}\tanh(w_{11}x) = \frac{1}{2}w(b+1)\tanh[\left(v + \frac{1}{2}(b-1)a\right)x]$ より, 次のように入力して図 3 の左側に $net1$ を出力させる.

$$net1[a\_, b\_, v\_, w\_] := NetGraph[net11[a, b, v], net12[a, b, w], NetPort["Input"]->1, 1->2]$$

また $w_{32}\tanh(w_{12}x) = \frac{1}{2}w(-b+1)\tanh[(v+\frac{1}{2}(b+1)a)\,x]$ より, $net2$ を同様に定める.

最後に $w_{31}\tanh(w_{11}x) + w_{32}\tanh(w_{12}x)$ より, 次のように入力して図 3 の右側に $parameterNet$ を出力させる.

$parameterNet[a\_, b\_, v\_, w\_] := NetGraph[net1[a, b, v, w], net2[a, b, v, w], TotalLayer[],$
$NetPort["Input"] -> 1, NetPort["Input"] -> 2,1, 2 -> 3 -> NetPort["Output1"], "Input" -> enc]$

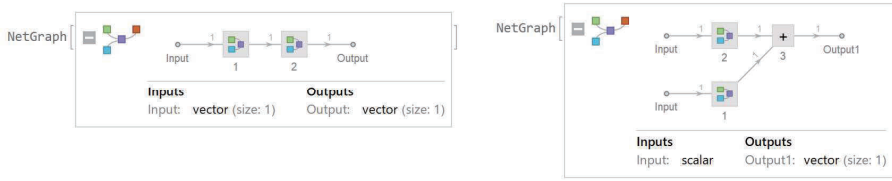

図 3: $net1$, $parameterNet$

損失関数を対数尤度比関数として, 次のように入力して図 4 に $trainingNet$ を出力させる.

$gaussianLikelihood[y\_, \mu\_] := PDF[NormalDistribution[\mu, 1], y]$
$trainingNet[a\_, b\_, v\_, w\_] := NetGraph[<|"params" -> parameterNet[a, b, v, w], "lhood" ->$
$ThreadingLayer[gaussianLikelihood], "neglog" -> ElementwiseLayer[-Log[\#]\&]|>,$
$NetPort["Output"], NetPort["params", "Output1"] -> "lhood", "lhood" -> "neglog" -> NetPort["Loss"]]$

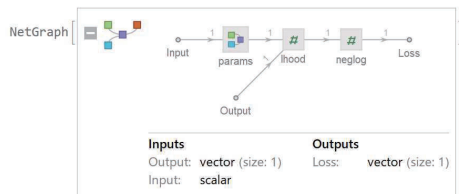

図 4: $trainingNet$(対数尤度比)

訓練データとテストデータに対して, 次のように入力して学習損失関数 G, 汎化損失関数 H を定める.

$G[a\_, b\_] := Mean[trainingNet[a, b, v0, w0][<|"Input" -> dataX, "Output" -> enc[dataY]|>]]$
$H[a\_, b\_] := Mean[trainingNet[a, b, v0, w0][<|"Input" -> testX, "Output" -> enc[testY]|>]]$

2.2 特異領域の近くにおける学習のダイナミクス

例 1 (訓練データ, 真の分布)

$-3 \leq x \leq 3$ 上の入力 X, $\sigma = 0.05$ の雑音 Z に対して, 訓練データを $0.25\tanh(0.2x) + 0.25\tanh(0.4x) + Z$, 真の分布を $q(y|x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{|y-(0.25\tanh(0.2x)+0.25\tanh(0.4x))|^2}{2\sigma^2}\right)$ と定める.

$a = 0.2$, $b = 0$, $v = 0.3$, $w = 0.5$ より真の分布が特異領域上の近くにあるとき, 初期値によって学習のダイナミクスを 5 つの場合に分けて考察する.

2.2.1 Overlap singularity のダイナミクス

学習モデルの初期値を $a = 0.15$, $b = -1.8$, $v = 0.3$, $w = 0.5$ とする. 真の分布が学習モデルで実現され, 損失関数を対数尤度比関数として, 次のように入力してニューラルネットワークを 200 回学習させる.

$$results1[a\_, b\_] := NetTrain[trainingNet[a, b, v, w], <|"Input" -> dataX, "Output" -> enc[dataY]|>, "RoundWeightsHistories", "TrainedNet", "RoundLossList", LossFunction -> "Loss", Method -> "ADAM", "InitialLearningRate" -> 0.1, BatchSize -> 30, MaxTrainingRounds -> 200]$$

パラメータ a, b の配列を作成し, 学習回数に対する変化を図 5 の左側に臨界直線に対する変化を図 5 の右側に表す.

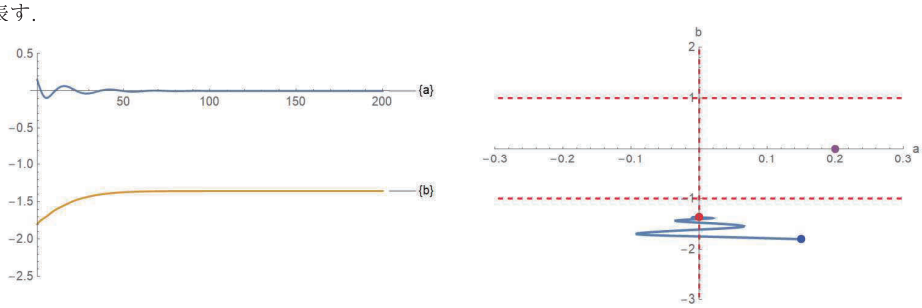

図 5: パラメータ a, b の変化

学習損失の配列を作成し, 学習損失の変化を図 6 の左側に学習損失曲面上のダイナミクスを図 6 の右側に出力する.

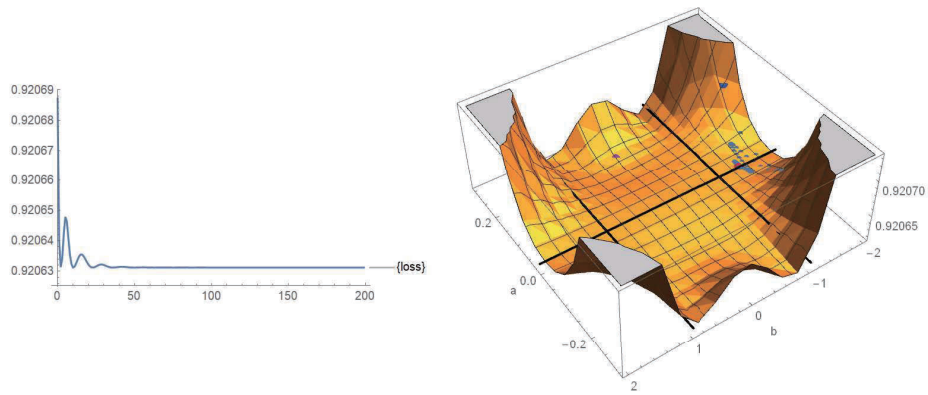

図 6: 学習損失, 学習損失曲面上のダイナミクス

臨界直線 $a = 0$ で停滞してプラトーが起こり, 真の分布に到達しない.

2.2.2 Cross elimination singularity のダイナミクス

学習モデルの初期値を $a = 0.15$, $b = -1.5$, $v = 0.3$, $w = 0.5$ とする．真の分布が学習モデルで実現され，損失関数を対数尤度比関数として，次のように入力してニューラルネットワークを 100 回学習させる．

$$results2[a\_, b\_] := NetTrain[trainingNet[a, b, v, w], <|"Input" -> dataX, "Output" -> enc[dataY]|>, "RoundWeightsHistories", "TrainedNet", "RoundLossList", LossFunction -> "Loss", Method -> "ADAM", "InitialLearningRate" -> 0.1, BatchSize -> 30, MaxTrainingRounds -> 100]$$

パラメータ a, b の配列を作成し，学習回数に対する変化を図 7 の左側に臨界直線に対する変化を図 7 の右側に表す．

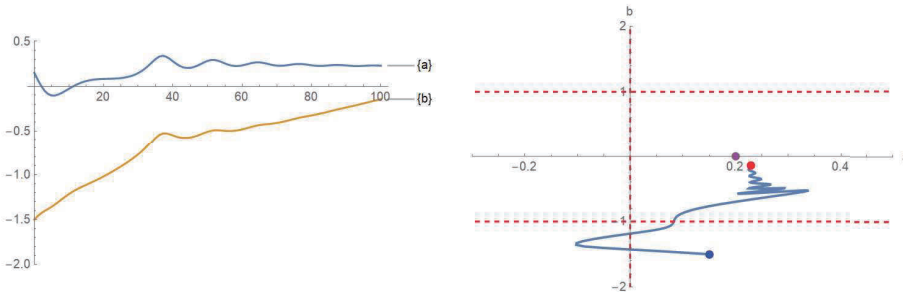

図 7: パラメータ a, b の変化

学習損失の配列を作成し，学習損失の変化を図 8 の左側に学習損失曲面上のダイナミクスを図 8 の右側に出力する．

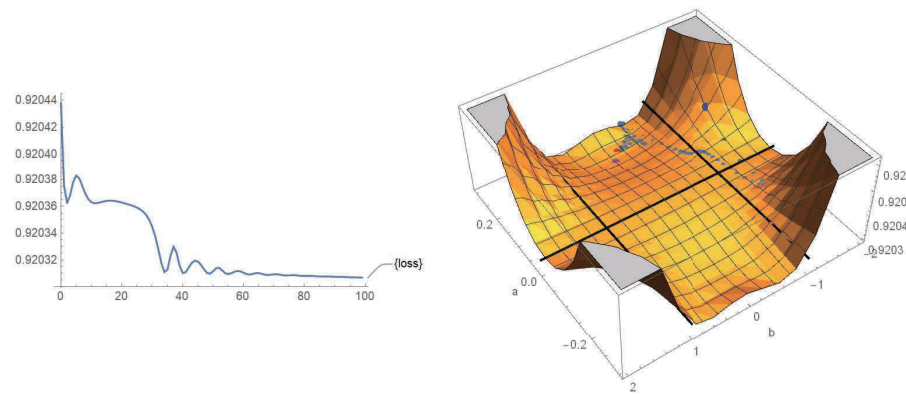

図 8: 学習損失, 学習損失曲面上のダイナミクス

臨界直線 $b = -1$ を横断するとき停滞してプラトーが起こり，真の分布に到達する．

2.2.3 First convergence のダイナミクス

学習モデルの初期値を $a = 0.05$, $b = 0$, $v = 0.3$, $w = 0.5$ とする．真の分布が学習モデルで実現され，損失関数を対数尤度比関数として，次のように入力してニューラルネットワークを 5 回学習させる．

$$results3[a\_, b\_] := NetTrain[trainingNet[a, b, v, w], <|"Input" -> dataX, "Output" -> enc[dataY]|>, "RoundWeightsHistories", "TrainedNet", "RoundLossList", LossFunction -> "Loss", Method -> "ADAM", "InitialLearningRate" -> 0.1, BatchSize -> 30, MaxTrainingRounds -> 5]$$

パラメータ a, b の配列を作成し，学習回数に対する変化を図 9 の左側に臨界直線に対する変化を図 9 の右側に表す．

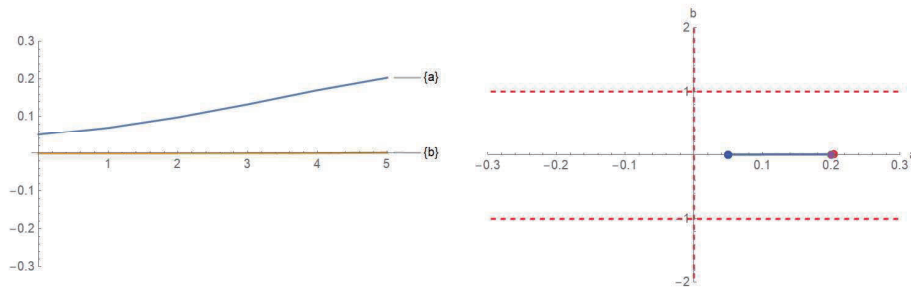

図 9: パラメータ a, b の変化

学習損失の配列を作成し，学習損失の変化を図 10 の左側に学習損失曲面上のダイナミクスを図 10 の右側に出力する．

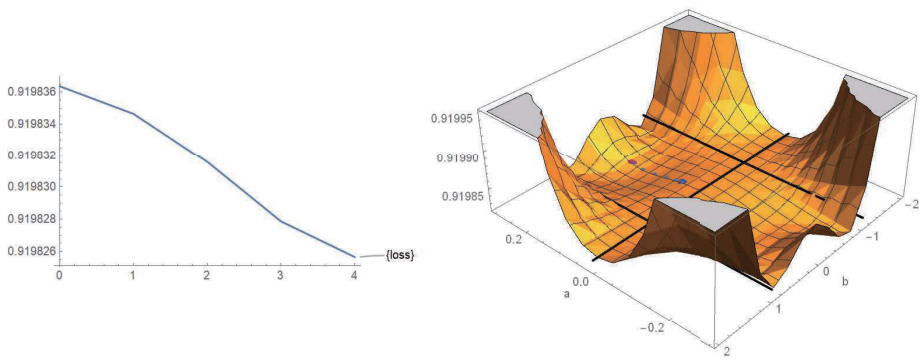

図 10: 学習損失, 学習損失曲面上のダイナミクス

真の分布に速く収束する．

2.2.4 Near elimination singularity のダイナミクス

学習モデルの初期値を $a = 0.7$, $b = -0.6$, $v = 0.3$, $w = 0.5$ とする．真の分布が学習モデルで実現され，損失関数を対数尤度比関数として，次のように入力してニューラルネットワークを 200 回学習させる．

$$results4[a\_, b\_] := NetTrain[trainingNet[a, b, v, w], <|"Input" -> dataX, "Output" -> enc[dataY]|>, "RoundWeightsHistories", "TrainedNet", "RoundLossList", LossFunction -> "Loss", Method -> "ADAM", "InitialLearningRate" -> 0.1, BatchSize -> 30, MaxTrainingRounds -> 200]$$

パラメータ a, b の配列を作成し，学習回数に対する変化を図 11 の左側に臨界直線に対する変化を図 11 の右側に表す．

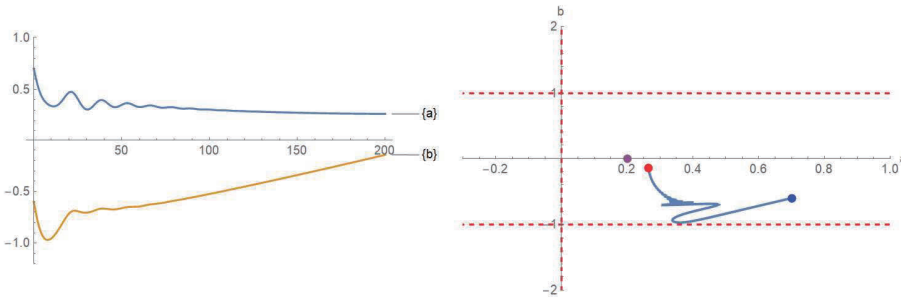

図 11: パラメータ a, b の変化

学習損失の配列を作成し，学習損失の変化を図 12 の左側に学習損失曲面上のダイナミクスを図 12 の右側に出力する．

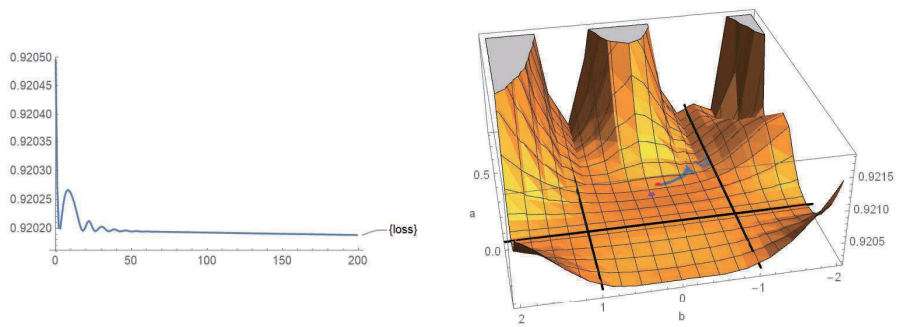

図 12: 学習損失，学習損失曲面上のダイナミクス

臨界直線 $b = -1$ に近づき，真の分布に到達する．

2.2.5 Output weight 0 のダイナミクス

学習モデルの初期値を $a = 0.7$, $b = -0.8$, $v = 0.3$, $w = 0.5$ とする．真の分布が学習モデルで実現され，損失関数を対数尤度比関数として，次のように入力してニューラルネットワークを 100 回学習させる．

$$results5[a\_, b\_] := NetTrain[trainingNet[a, b, v, w], <|"Input" -> dataX, "Output" -> enc[dataY]|>, "RoundWeightsHistories", "TrainedNet", "RoundLossList", LossFunction -> "Loss", Method -> "ADAM", "InitialLearningRate" -> 0.1, BatchSize -> 30, MaxTrainingRounds -> 100]$$

パラメータ a, b の配列を作成し，学習回数に対する変化を図 13 の左側に臨界直線に対する変化を図 13 の右側に表す．

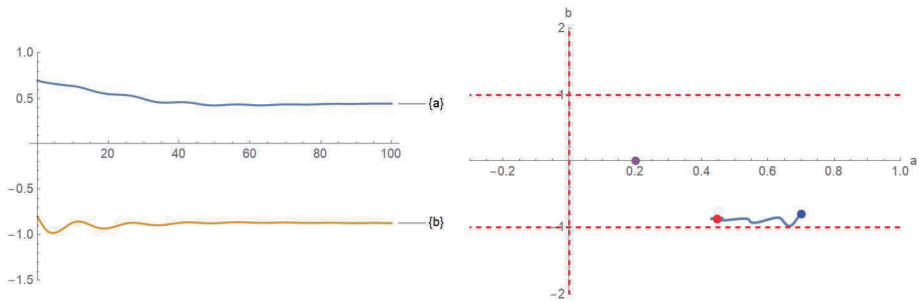

図 13: パラメータ a, b の変化

学習損失の配列を作成し，学習損失の変化を図 14 の左側に学習損失曲面上のダイナミクスを図 14 の右側に出力する．

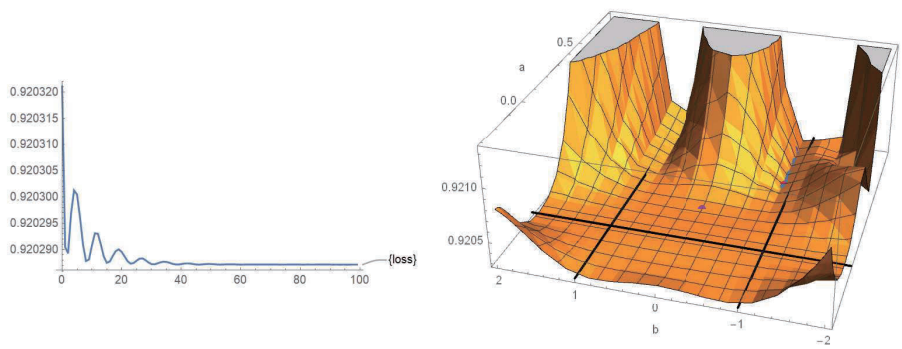

図 14: 学習損失, 学習損失曲面上のダイナミクス

臨界直線 $b = -1$ で停滞してプラトーが起こり，真の分布に到達しない．

3　過学習・過剰汎化をおこす学習のダイナミクス

3.1　Mathmatica におけるニューラルネットワークの作成

変数 $F1$, $F2$, 定数 $elem0$, $elem1$, $elem2$, $elem3$, $elem4$ を次のように Mathmatica に入力して定める.

$F1[a\_] := NetInsertSharedArrays[NetChain[LinearLayer[1,"Weights"->a,"Biases"->None]],"Linear1"]$,
$F2[b\_] := NetInsertSharedArrays[NetChain[LinearLayer[1,"Weights"->b,"Biases"->None]],"Linear2"]$,
$elem0 := ElementwiseLayer[\#*(1/2)\&]$, $elem1 := ElementwiseLayer[\#*(-1)\&]$,
$elem2[v\_,d\_] := ElementwiseLayer[\#*(v)+d\&]$, $elem3[w\_] := ElementwiseLayer[\#*(w)\&]$,
$elem4[c\_] := ElementwiseLayer[\#+(c)\&]$.

初めに $\mathbf{a}=(a,\,c)$, $\mathbf{v}=(v,\,d)$ とする.
$\mathbf{w_1}^{\mathrm{T}}\mathbf{x}=\left(\mathbf{v}+\frac{1}{2}(b-1)\mathbf{a}\right)^{\mathrm{T}}\mathbf{x}$ より, 次のように入力して図 15 の左側に $net11$ を出力させる.

$$net11[a\_,b\_,c\_,d\_,v\_] := NetGraph[elem0, elem1, F1[a], F2[b], elem2[v,d], elem4[c], TotalLayer[],$$
$$NetPort["Input"]->3, 3->6->1, 1->4, 1->2, 4, 2, 5->7]$$

$w_{31}\tanh(x)=\frac{1}{2}w(b+1)\tanh(x)$ より, 次のように入力して図 15 の右側に $net12$ を出力させる.

$$net12[a\_,b\_,c\_,d\_,w\_] := NetGraph[Tanh, elem0, elem3[w], F2[b], TotalLayer[],$$
$$NetPort["Input"]->1, 1->2->3->4, 3, 4->5]$$

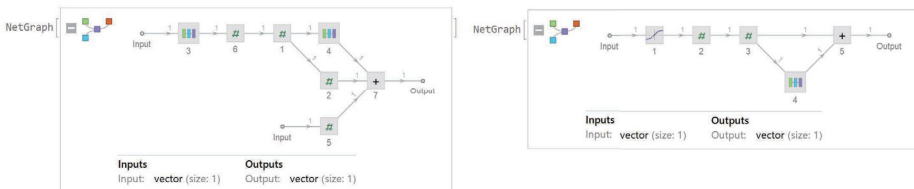

図 15: $net11$, $net12$

また $\mathbf{w_2}^{\mathrm{T}}\mathbf{x}=\left(\mathbf{v}+\frac{1}{2}(b+1)\mathbf{a}\right)^{\mathrm{T}}\mathbf{x}$, $w_{32}\tanh(x)=\frac{1}{2}w(-b+1)\tanh(x)$ より, $net21$, $net22$ を同様に定めて図 16 に出力させる.

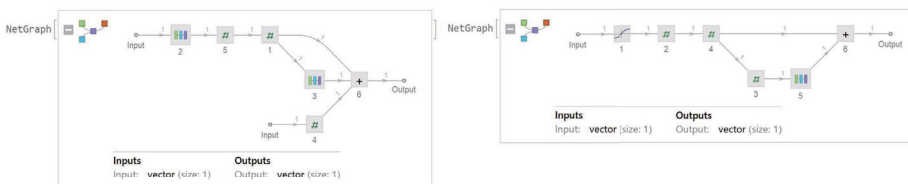

図 16: $net21$, $net22$

次に $w_{31}\tanh(\mathbf{w_1}^{\mathrm{T}}\mathbf{x})=\frac{1}{2}w(b+1)\tanh[\left(\mathbf{v}+\frac{1}{2}(b-1)\mathbf{a}\right)^{\mathrm{T}}\mathbf{x}]$ より, 次のように入力して図 17 の左側に $net1$ を出力させる.

$$net1[a\_,b\_,c\_,d\_,v\_,w\_] := NetGraph[net11[a,b,c,d,v], net12[a,b,c,d,w], NetPort["Input"]->1, 1->2]$$

また $w_{32}\tanh(\mathbf{w_2}^{\mathrm{T}}\mathbf{x}) = \frac{1}{2}w(-b+1)\tanh[\left(\mathbf{v}+\frac{1}{2}(b+1)\mathbf{a}\right)^{\mathrm{T}}\mathbf{x}]$ より, $net2$ を同様に定める.

最後に $w_{31}\tanh(\mathbf{w_1}^{\mathrm{T}}\mathbf{x}) + w_{32}\tanh(\mathbf{w_2}^{\mathrm{T}}\mathbf{x})$ より, 次のように入力して図 17 の右側に $parameterNet$ を出力させる.

$parameterNet[a\_, b\_, c\_, d\_, v\_, w\_] := NetGraph[net1[a, b, c, d, v, w], net2[a, b, c, d, v, w], TotalLayer[],$
$NetPort[\text{"}Input\text{"}] -> 1, NetPort[\text{"}Input\text{"}] -> 2,1,2 -> 3 -> NetPort[\text{"}Output1\text{"}], \text{"}Input\text{"} -> enc]$

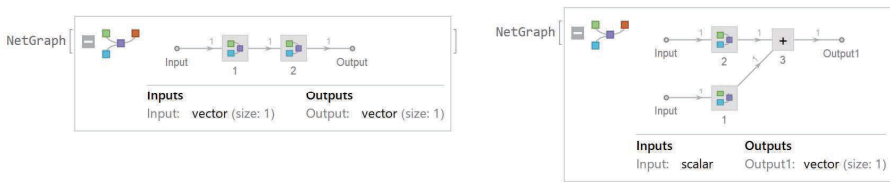

図 17: $net1$, $parameterNet$

損失関数を 2 乗誤差関数として, 次のように入力して図 18 に $trainingNet$ を出力させる.

$trainingNet[a\_, b\_, c\_, d\_, v\_, w\_] := NetGraph[<|\text{"}params\text{"} -> parameterNet[a, b, c, d, v, w], \text{"}loss\text{"} ->$
$MeanSquaredLossLayer[]|>, NetPort[\text{"}Output\text{"}], NetPort[\text{"}params\text{"}, \text{"}Output1\text{"}] -> \text{"}loss\text{"}]$

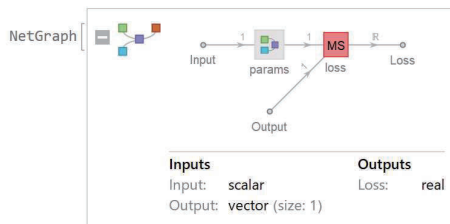

図 18: $trainingNet$(2 乗誤差)

訓練データとテストデータに対して, 次のように入力して学習損失関数 G, 汎化損失関数 H を定める.

$G[a\_, b\_] := Mean[trainingNet[a, b, c0, d0, v0, w0][<|\text{"}Input\text{"} -> dataX, \text{"}Output\text{"} -> enc[dataY]|>]]$
$H[a\_, b\_] := Mean[trainingNet[a, b, c0, d0, v0, w0][<|\text{"}Input\text{"} -> testX, \text{"}Output\text{"} -> enc[testY]|>]]$

3.2 過学習をおこす学習のダイナミクス

例 2 (訓練データ, テストデータ, 真の分布)

$-3 \leq x \leq 3$ 上の入力 $X, \sigma = 0.05$ の雑音 Z に対して, 訓練・テストデータを $0.25\tanh(3x)+0.25\tanh(3x)+Z$, 真の分布を $q(y|x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{|y(0.25\tanh(3x)+0.25\tanh(3x))|^2}{2\sigma^2}\right)$ と定める.

訓練データを図 19 の左側に, テストデータを図 19 の右側に出力する.

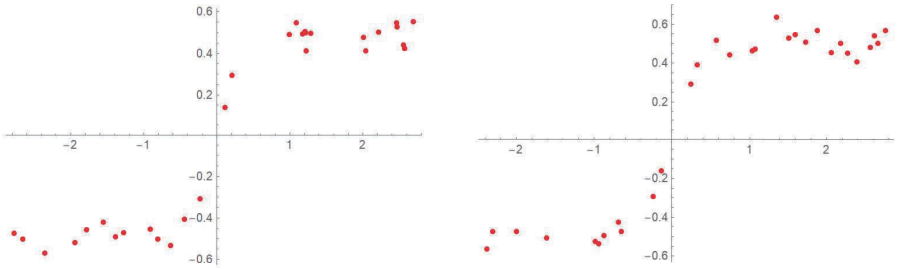

図 19: 訓練データ, テストデータ

$a = 0, b = 0, c = 0, d = 0, v = 3, w = 0.5$ より真の分布が特異領域上にあるとき, 過学習を起こす場合の学習のダイナミクスを考察する.

学習モデルの初期値を $a = 0.2, b = 0, c = 0, d = 0, v = 3, w = 0.5$ とすると, 真の分布が学習モデルによって実現される. 損失関数を 2 乗誤差関数, 検証集合をテストデータとして, 次のように入力してニューラルネットを 300 回学習させる.

$$results1[a\_, b\_] := NetTrain[trainingNet[a, b, 0, 0, v1, w1], <|"Input" -> dataX, "Output" -> enc[dataY]|>, All, ValidationSet -><|"Input" -> testX, "Output" -> enc[testY]|>, LossFunction -> "Loss", Method -> "ADAM", "InitialLearningRate" -> 0.1, BatchSize -> 30, MaxTrainingRounds -> 300]$$

学習・汎化損失の配列を作成し, 学習損失の変化を図 20 の左側に汎化損失の変化を図 20 の右側に出力する.

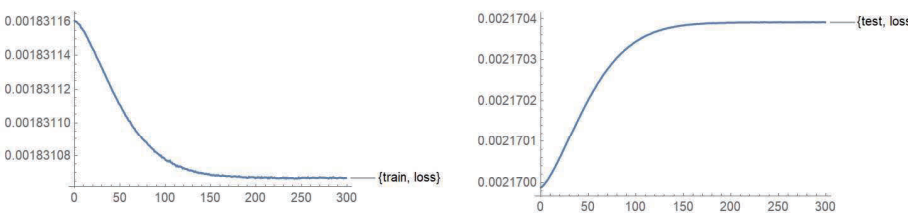

図 20: 学習損失・汎化損失

学習損失が減少するが汎化損失が増加して, 過学習が起きている.

学習・汎化損失の変化を図 21 の左側に，訓練・テストデータの上に学習後のニューラルネットの出力を図 21 の右側に出力する．

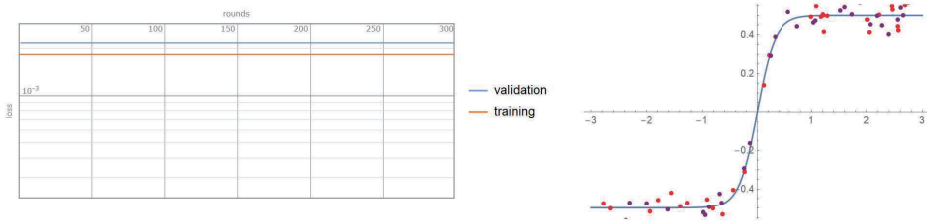

図 21: 学習・汎化損失, 学習後のニューラルネット

次のように入力してニューラルネットワークを 300 回学習させる．

$$results2[a\_, b\_] := NetTrain[trainingNet[a, b, 0, 0, v1, w1], <|"Input" -> dataX, "Output" -> enc[dataY]|>,$$
$$"RoundWeightsHistories", "TrainedNet", "RoundLossList", LossFunction -> "Loss",$$
$$Method -> "ADAM", "InitialLearningRate" -> 0.1, BatchSize -> 30, MaxTrainingRounds -> 300]$$

パラメータ a, b の配列を作成し, 学習回数に対する変化を図 22 の左側に臨界直線に対する変化を図 22 の右側に表す．

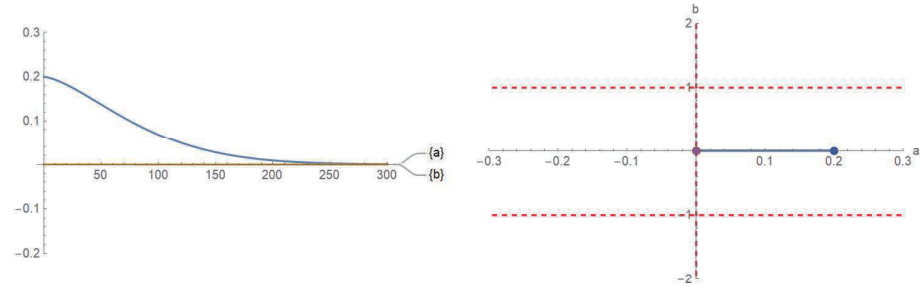

図 22: パラメータ a, b の変化

臨界直線 $a = 0$, $b = \pm 1$ に対して, 学習損失曲面上のダイナミクスを図 23 の左側に汎化損失曲面上のダイナミクスを図 23 の右側に出力する．

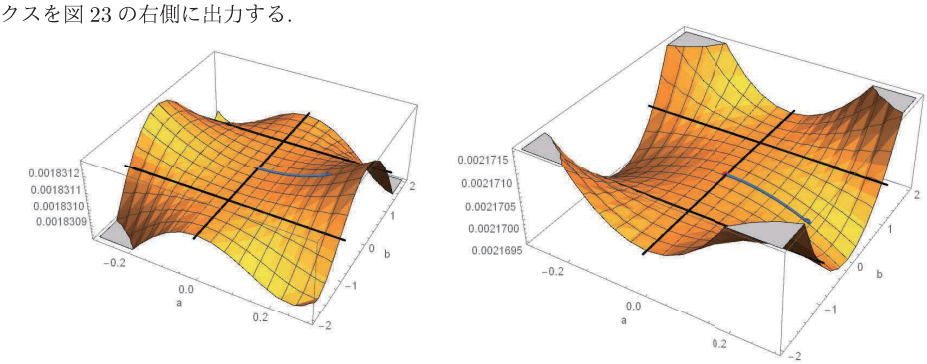

図 23: 学習・汎化損失曲面上のダイナミクス

3.3 過剰般化をおこす学習のダイナミクス

例 3 (訓練データ, テストデータ, 真の分布)

$-3 \leq x \leq 3$ 上の入力 X, $\sigma = 0.05$ の雑音 Z に対して, 訓練データを $0.5\tanh(4.8x)+0.4\tanh(5x-10)+Z$ と定め, $2.1 \leq x \leq 3$ 上の入力 X, 雑音 Z に対して, テストデータを $0.2\tanh(5x-10)+0.2\tanh(5x-10)+0.5+Z$ と定める. 真の分布を $q(y|x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{|y-(0.5\tanh(4.8x)+0.4\tanh(5x-10))|^2}{2\sigma^2}\right)$ で定める.

訓練データを図 24 の左側に, テストデータを図 24 の右側に出力する.

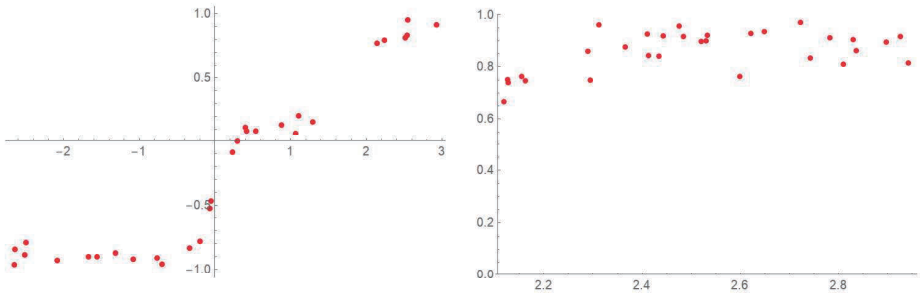

図 24: 訓練データ, テストデータ

$a = 0.2$, $b = 0.11$, $c = -10$, $d = -4.44$, $v = 4.88$, $w = 0.9$ より真の分布が特異領域上の近くにあり. 過剰汎化を起こす場合の学習のダイナミクスを考察する.

学習モデルの初期値を $a = 0.2$, $b = 0.11$, $c = -18$, $d = -4.44$, $v = 4.88$, $w = 0.9$ とすると, $c = -18$ と固定するため, 真の分布が学習モデルによって実現されない. 損失関数を 2 乗誤差関数, 検証集合をテストデータとして, 次のように入力してニューラルネットを 20 回学習させる.

$results1[a\_, b\_] := NetTrain[trainingNet[a, b, -18, -4.44, v1, w1], <|"Input" -> dataX, "Output" -> enc[dataY]|>, All, ValidationSet -><|"Input" -> testX, "Output" -> enc[testY]|>, LossFunction -> "Loss", Method -> "ADAM", "InitialLearningRate" -> 0.1, BatchSize -> 30, MaxTrainingRounds -> 20]$

学習・汎化損失の配列を作成し, 学習損失の変化を図 25 の左側に汎化損失の変化を図 25 の右側に出力する.

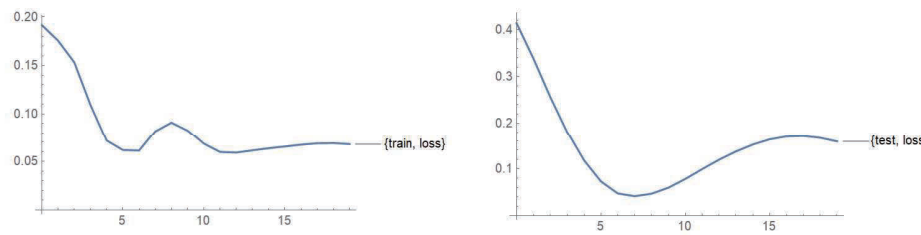

図 25: 学習損失・汎化損失

学習・汎化損失の変化を図 26 に出力させる.

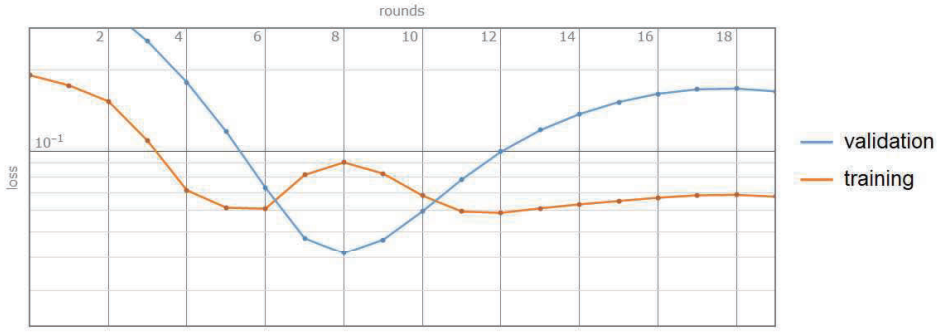

図 26: 学習・汎化損失

汎化損失が減少して学習損失よりも小さくなり,増加して学習損失よりも大きくなる過剰般化が起きている.
テストデータの上にニューラルネットの出力について学習前を図 27 の左側に学習後を図 27 の右側に表す.

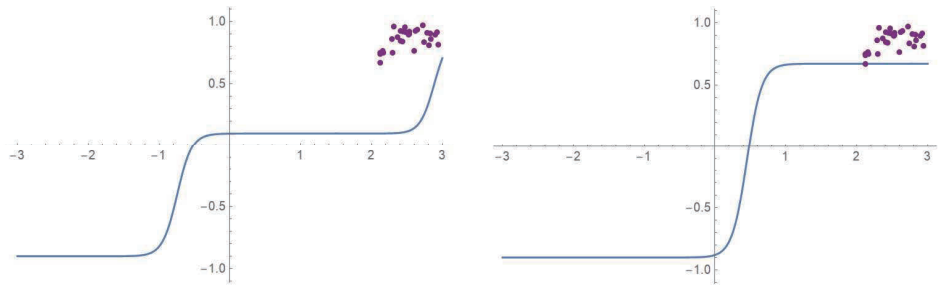

図 27: 学習前・学習後のニューラルネット

次のように入力してニューラルネットワークを 70 回学習させる.

$results2[a\_, b\_] := NetTrain[trainingNet[a, b, -18, -4.44, v1, w1], <|"Input" -> dataX, "Output" -> enc[dataY]|>, "RoundWeightsHistories", "TrainedNet", "RoundLossList", LossFunction -> "Loss", Method -> "ADAM", "InitialLearningRate" -> 0.1, BatchSize -> 30, MaxTrainingRounds -> 70]$

パラメータ a, b の配列を作成し,学習回数に対する変化を図 28 の左側に臨界直線に対する変化を図 28 の右側に表す.

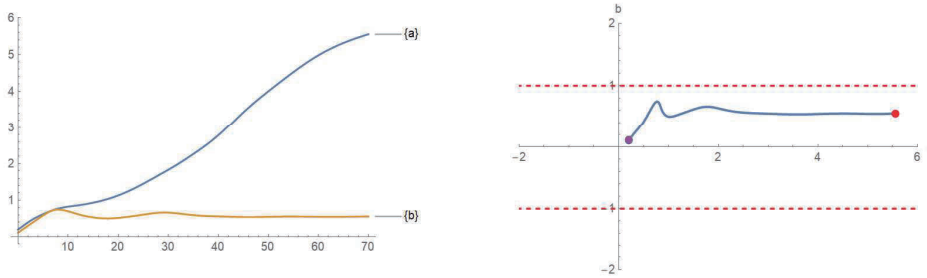

図 28: パラメータ a, b の変化

テストデータを検証集合として次のように入力してニューラルネットワークを 70 回学習させる.

$$results3[a\_, b\_] := NetTrain[trainingNet[a, b, -18, -4.44, v1, w1], <|"Input" -> dataX, "Output" -> enc[dataY]|>, All, ValidationSet -><|"Input" -> testX, "Output" -> enc[testY]|>, LossFunction -> "Loss", Method -> "ADAM", "InitialLearningRate" -> 0.1, BatchSize -> 30, MaxTrainingRounds -> 70]$$

学習・汎化損失の変化を図 29 の左側に, テストデータの上に学習後のニューラルネットの出力を図 29 の右側に出力する.

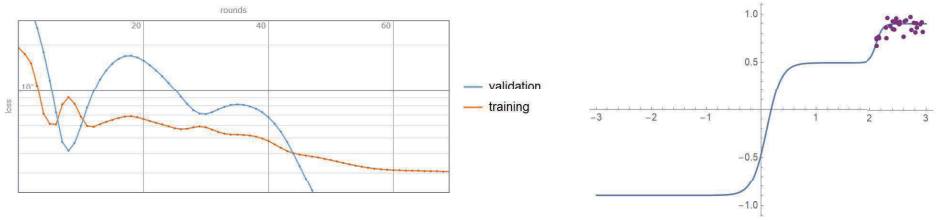

図 29: 学習・汎化損失, 学習後のニューラルネット

臨界直線を $b = \pm 1$ に対して, 学習損失曲面上のダイナミクスを図 30 の左側に汎化損失曲面上のダイナミクスを図 30 の右側に出力する.

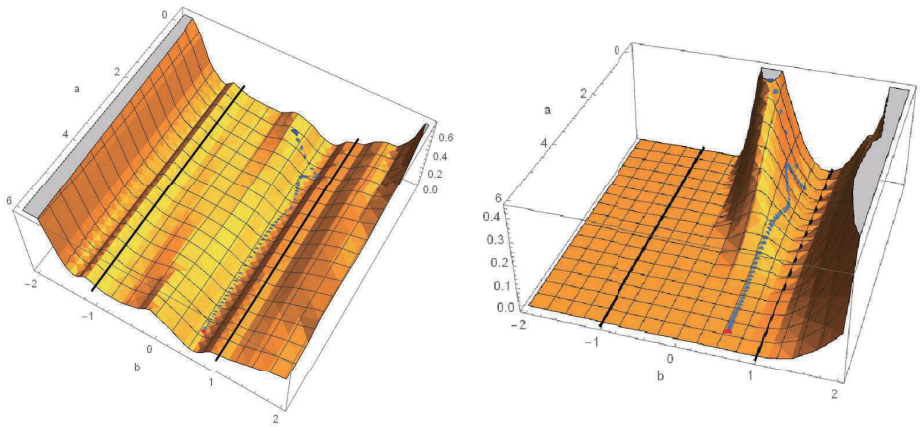

図 30: 学習・汎化損失曲面上のダイナミクス

参 考 文 献

[1] H. Wei, J. Zhang, F. Cousseau, T. Ozeki, and S. Amari, “Dynamics of learning near singularities in layered networks,” Neural Computation, vol. 20, no. 34, pp. 813–843, 2008.

[2] W. Guo, H. Wei , Y. Ong, J. R. Hervas, J. Zhao, H. Wang, K. Zhang, “Numerical Analysis near Singularities in RBF Networks,” Journal of Machine Learning Research, vol. 19, no. 1, pp. 1-39, 2018.

[3] F. Cousseau, T. Ozeki, and S. Amari, “Dynamics of Learning in Multilayer Perceptrons Near Singularities,” IEEE Transactions on Neural Networks, vol. 19, no. 8, pp. 1313–1328, 2008.

[4] K. Fukumizu and S. Amari, “Local minima and plateaus in hierarchical structures of multilayer perceptrons,” Neural Networks, vol. 13, no. 3, pp. 317–327, 2000.