卒業論文

ターミナルアトラクタを組み込んだ
複製・競合メカニズムによる効率的な機械学習

Building Efficient Machine Learning
with Reproduction and Competition Mechanisms Incorporating
Terminal Attractors

富山県立大学 工学部 情報システム工学科
2120014 小澤 翔太
指導教員 奥原 浩之 教授
提出年月: 令和7年(2025年)2月

目 次

図一覧 **ii**

表一覧 **iii**

記号一覧 **iv**

第 1 章　はじめに **1**
§ 1.1 本研究の背景 1
§ 1.2 本研究の目的 2
§ 1.3 本論文の概要 2

第 2 章　複製・競合を考慮した動径基底関数ネットワーク **3**
§ 2.1 競合動径基底関数ネットワーク 3
§ 2.2 ターミナルアトラクタ 6
§ 2.3 基底関数の複製 8

第 3 章　機械学習と最適化の関係 **14**
§ 3.1 代表的な機械学習の手法について 14
§ 3.2 エネルギー関数の最小化手法 16
§ 3.3 処理高速化のための手法 18

第 4 章　提案手法 **20**
§ 4.1 新たに提案されるターミナルアトラクタの定義式 20
§ 4.2 Python の高速化ライブラリ 21
§ 4.3 関数近似器としての活用 22

第 5 章　実験結果並びに考察 **24**
§ 5.1 実験の概要 24
§ 5.2 実験結果と考察 25

第 6 章　おわりに **28**

謝辞 **29**

参考文献 **30**

図一覧

2.1 初期状態の基底関数 6
2.2 競合の結果 6
2.3 ターミナルアトラクタなし 8
2.4 ターミナルアトラクタあり 8

4.1 テキストデータのフォーマット 21
4.2 システムのフロントページ 21

5.1 ベクトル化の結果 26
5.2 出力されたタイトル 26
5.3 出力された 3D グラフ 27

表一覧

記号一覧

以下に本論文において用いられる用語と記号の対応表を示す.

用語	記号
入力ニューロンの番号	j
入力ニューロン数	M
入力ベクトルの番号	i
入力ベクトルの数	N
第 j ニューロンの内的自然増加率	α_j
第 j ニューロンと第 h ニューロンの競合効果	γ_{jh}
第 j ニューロンのシナプス結合荷重の大きさ	w_j

第1章

はじめに

§ 1.1　本研究の背景

近年，人工知能や機械学習の発展に伴い，複雑なデータの解析や予測を可能にするさまざまな手法が提案されている．その中でも関数近似問題やパターン識別に適したニューラルネットワークの一つとして注目されているのが，動径基底関数ネットワーク（Radial Basis Function Network: RBFN）[1] である．RBFN は，最初に有限個の入出力データを補完する方法として提案されたもので，3 層構造を持つニューラルネットワークである．その設計において，RBFN は多層パーセプトロンと同様に任意の非線形関数の近似が可能であるという特長を持つ．

RBFN の最大の特徴の一つは，基底関数としてガウス関数を採用している点である．これにより，階層型ニューラルネットワークと比較して，ニューロンごとの局所的な学習が可能となり，収束の速さやモデルの柔軟性において優れた特性を示す．具体的には，各ニューロンが局所的なデータ領域に焦点を当てるため，ノイズに対する頑健性が向上し，学習が効率的に進むという利点がある．RBFN は様々な応用領域においてその有用性が示されている．例えば，非線形関数の線形和を用いてがけ崩れ発生限界雨量線を設定する研究 [4] や，複数の波源から発生した信号の到来方向を予測するためのツール [5] として活用されている．これらの応用例は，RBFN が非線形関数の近似器として非常に強力なツールであることを示している．

しかしながら，RBFN にはいくつかの課題が存在する．その一つが，未知の非線形関数を近似する際に必要となるニューロン数が事前には分からないという問題である．特に多くの冗長なニューロンを含む場合，学習プロセスが遅延し，さらに過学習のリスクが増大することが知られている．

これらの問題を解決するために，適者生存型学習則に基づいたシナプス可塑性方程式を適用した競合動径基底関数ネットワーク（Competitive RBFN: CRBFN）[2] が提案されている．CRBFN は，ニューロン間に競合を発生させることで学習に必要なニューロンのみが選択され，不要な冗長ニューロンを自然に排除することが可能である．この特性により，モデルの簡潔化と学習効率の向上が実現される．一方で，CRBFN は基底関数を新たに追加する機能を備えていないために，ニューロンの数が不足している場合には十分な関数近似が困難になるという欠点がある．

これを受けて，CRBFN に基底関数を複製する機能を付加した，複製・競合動径基底関数ネットワーク（Reproductive CRBFN: RC-RBFN）[3] が提案されている．RC-RBFN は新しいデータや状況に対応するために必要な基底関数を動的に複製し，さらに競合を活用し

て不要な基底関数を排除することで，モデルの柔軟性と効率を向上させている．この機構により，基底関数の過剰複製を抑制しつつ関数近似の精度向上を図ることが可能となっている．

§ 1.2　本研究の目的

既存の RBFN による関数近似手法では，基底関数の数が適切に設定されていない場合に，近似精度が著しく低下する，または過学習を引き起こすという問題が指摘されてきた．この問題に対処するために，RC-RBFN が提案され，基底関数の動的な複製機能と競合による冗長な基底関数の削除を組み合わせることで，モデルの柔軟性や効率性を大幅に向上させる仕組みが実現された．RC-RBFN は基底関数の数をデータや状況に応じて動的に調整できる点で，従来の固定された基底関数の数を前提とした手法に比べて高い適応性を持ち，特に多様なデータ構造や非線形関数の近似において優れた性能を発揮することが期待されている．

しかし一方で，RC-RBFN やその基盤である CRBFN は，目的関数の次元数が増加するに伴い，計算量が指数関数的に増大するいわゆる「次元の呪い」の影響を受けやすいという課題を抱えている．この問題は特に高次元データを扱う場合に顕著であり，計算負荷が非常に大きくなることで，現実的な時間内で学習を終了させることが困難になる．

このような背景を踏まえ，従来の RC-RBFN を基盤としながら，次元の呪いに起因する計算負荷の増大を軽減するための新たな手法の提案を目的とする．次元の呪いを回避することにより，計算効率を大幅に改善しつつ，さらなる汎化性能を持つ関数近似モデルの開発が可能となることを目指す．本研究では特に，RC-RBFN の計算負荷の主要因となっている高次元データにおける多重積分処理部分に着目し，このアルゴリズムを改良することで効率的かつ精度の高い学習を実現する方法を検討する．

§ 1.3　本論文の概要

本論文は次のように構成される．

第 1 章 本研究の背景と目的について説明する．

第 2 章 適者生存型学習則を適用した CRBFN と環境の変化に対応できる RC-RBFN についてまとめる．

第 3 章 現在の機械学習の手法についてまとめる．

第 4 章 提案手法について説明する．

第 5 章 実際の事例を設けて，第 4 章で述べた手法で，IP ランドスケープ実施の支援を行い，システムの評価を行う．

第 6 章 本論文における前章までの内容をまとめつつ，本研究で実現できたことと今後の展望について述べる．

第2章

複製・競合を考慮した動径基底関数ネットワーク

§ 2.1　競合動径基底関数ネットワーク

シナプス可塑性方程式

ニューラルネットワークは大きく分けて，素子であるニューロン，それらを結合するシナプス，そして動作規則により構成される．なかでも，記憶にもっとも関係した情報処理は，シナプスにおいて行われているとされる．記憶には種々のものが考えられるが，本研究では短期記憶と長期記憶に着目し，短期記憶はニューロンの発火頻度，長期記憶は細胞膜の特性の変化により生じるものとする．シナプスの可塑性を記述する方程式は，これらの要因を含んだものとなっていなければならない．さらに，成長や活動に必要な神経成長因子（Nerve Growth Factor: NGF）はシナプス間隔の微小な領域において競合が発生することによりシナプスに摂取される．第 j ニューロンの微小領域におけるシナプス前終末発火頻度を ξ_j とし，これが作用するニューロンの細胞膜におけるシナプス後発火頻度を η とする．また，第 j ニューロンへの入力数を $i(i = 1, ..., N)$ とする．

そこで，発火頻度や膜の特性変化を生じる物質の時間変化と，微小な領域での競合を考慮したシナプス結合荷重の大きさの時間変化は以下の方程式に従うものとする．

$$\frac{dw_j}{dt} = \alpha_j w_j + g_j w_j + f_j \tag{2.1}$$

ここで，$w_j \geq 0$ である．また，g_j は微小領域に供給される NGF のうち，その領域に付着している第 j ニューロンのシナプスが入手できる量であり，f_j は NGF と環境因子に依存するゆらぎである．α_j は内的自然増加率である．内的自然増加率は脳の可塑性を表す Hebb 則 [7] を意味する．内的自然増加率 α_j は

$$\alpha_j = \sum_{i=1}^{N} \eta(\mathbf{x}_i)\xi_j(\mathbf{x}_i) \tag{2.2}$$

で定義される．また，NGF の量 g_j は次の方程式に従う．

$$\begin{aligned}\frac{dg_j}{dt} &= \epsilon_j(G - g_j) - (\theta_j w_j + \sum_{h \neq j} \theta_h w_h) \\ &= \epsilon_j(G - g_j) - \sum_{h} \theta_h w_h \end{aligned} \tag{2.3}$$

G は微小領域への NGF の供給速度であり，膜の特性により決定される変数である．ϵ_j, θ_h は正の定数である．NGF の量の時間変化もシナプスの興奮性，抑制性によらず，そのシナプス間感度の大きさに依存する．また，領域へ付着するシナプスが入手し得る NGF の量の時間変化に対し，NGF の供給速度の時間変化が無視できるとして G を定数とみなす．シナプス間感度の時間変化は発火頻度に依存するため，シナプス間感度の時間変化に対し，NGF の量の時間変化は無視できるものとする．そこで，第 j ニューロンと第 h ニューロンが同時に NGF を消費することによる競合の効果 γ_{jh} を導入すると以下のシナプス可塑性方程式が導かれる．

$$\begin{aligned}\frac{dw_j}{dt} &= (G + \alpha_j - \frac{1}{\epsilon_j}\sum_h \theta_h w_h)w_j + f_j \\ &= (G + \alpha_j - \sum_h \gamma_{jh} w_h)w_j + f_j \end{aligned} \tag{2.4}$$

ここでは，競合係数 γ_{jh} を

$$\gamma_{jh} = \frac{\theta_h}{\epsilon_j} = \sum_{i=1}^{N} \xi_j(\mathbf{x}_i)\xi_h(\mathbf{x}_i) \tag{2.5}$$

で定義する．

簡単のため，NGF の摂取量が一定 $(G = 0)$ で揺らぎのない場合 $(f_j = 0)$ を考える．このとき，シナプス可塑性方程式は

$$\frac{dw_j}{dt} = (\alpha_j - \sum_h \gamma_{jh} w_h)w_j \tag{2.6}$$

と表される．

適者生存型学習則を適用した RBFN

ここで，シナプス後発火頻度 $\eta(\mathbf{x}_i)$ を入力 x に対する望ましい出力，シナプス前発火頻度 $\xi_j(\mathbf{x}_i)$ を動径基底関数であるとみなすことで，シナプス結合荷重 w_j は競合を行いながら RBFN と同様に望ましい出力と動径基底関数の二乗誤差関数を減少させることが示される．本論文では，(2.6) 式のシナプス可塑性方程式を適者生存型学習則とする．このようなモデル式は一般 Lotka-Volttera 競争系と呼ばれ，競争関係にある生物の個体数の変動を表す数理モデルの一種である．

次に，この適者生存型学習則を RBFN に適用する．RBFN では未知の非線形関数を近似するためにあらかじめ必要なニューロン数が不明であり，冗長なニューロンを必要とする．そこで，適者生存型学習則によって近似に不要な冗長なニューロンを削除する機能を備えた RBFN として CRBFN がある．CRBFN ではシナプス結合荷重間に競合を生じさせ，学習に必要なニューロンのみが自然に生き残り，学習の効率化を図ることができる．RBFN における動径基底関数として一般的にガウス関数が用いられることが多く，本研究でもガウス関数を用いる．このとき，動径基底関数 $\xi_j(\mathbf{x}_i)$ は

$$\xi_j(\mathbf{x}_i) = exp\{-\frac{1}{2}(\mathbf{x}_i - \mathbf{m}_j)^{\mathrm{T}}\mathbf{\Sigma}_j^{-1}(\mathbf{x}_i - \mathbf{m}_j)\} \tag{2.7}$$

で定義される．T は行列の転置を表す．パラメータ m_j，$\mathbf{\Sigma}_j^{-1}$ はそれぞれ j 番目の動径基底関数の中心位置と分散共分散行列であり，$\mathbf{m} \equiv [\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_M]$，$\mathbf{\Sigma} \equiv [\mathbf{\Sigma}_1, \mathbf{\Sigma}_2, ..., \mathbf{\Sigma}_M]$ である．

RBFN による関数近似は二乗誤差関数

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\{\eta(\mathbf{x}_i) - s(\mathbf{x}_i)\}^2 \tag{2.8}$$

を減少させることにより実現される．ここで，$\mathbf{w} \equiv [w_1, w_2, ..., w_M]$ であり，

$$s(\mathbf{x}_i) = \sum_{j=1}^{M} w_j\xi_j(\mathbf{x}_i) \tag{2.9}$$

である．(2.9) 式の右辺は微小領域に付着しているすべてのシナプス前終末のシナプス前発火頻度 $\xi_j(\mathbf{x}_i)$ と，シナプス結合荷重 w_j との積の総和であるので，$s(\mathbf{x}_i)$ は神経伝達物質放出量となる．

つまり，RBFN が学習により獲得しなければならないのは，第 j ニューロンのシナプス結合荷重 w_j，パラメータ m_j ならびにパラメータ Σ_j である．学習アルゴリズムに最急降下法を適用することで

$$\begin{aligned}
\frac{dw_j}{dt} &= -\Delta\frac{\partial E(\mathbf{w})}{\partial w_j} \\
&= \Delta\sum_{i=1}^{N}\{\eta(\mathbf{x}_i) - s(\mathbf{x}_i)\}w_j\xi_j(\mathbf{x}_i), && (2.10)\\
\frac{d\mathbf{m}_j}{dt} &= -\Delta\frac{\partial E(\mathbf{w})}{\partial \mathbf{m}_j} \\
&= \Delta\sum_{i=1}^{N}\{\eta(\mathbf{x}_i) - s(\mathbf{x}_i)\}w_j\xi_j(\mathbf{x}_i)\frac{(\mathbf{x}_i - \mathbf{m}_j)}{\mathbf{\Sigma}_j{}^2}, && (2.11)\\
\frac{d\mathbf{\Sigma}_j}{dt} &= -\Delta\frac{\partial E(\mathbf{w})}{\partial \mathbf{\Sigma}_j} \\
&= \Delta\sum_{i=1}^{N}\{\eta(\mathbf{x}_i) - s(\mathbf{x}_i)\}w_j\xi_j(\mathbf{x}_i)\frac{(\mathbf{x}_i - \mathbf{m}_j)^2}{\mathbf{\Sigma}_j{}^3} && (2.12)
\end{aligned}$$

が得られる．Δ は学習率であり，適当な正の定数である．このような適者生存型学習則に従う RBFN を CRBFN と呼ぶ．図 2.1，図 2.2 にシナプス結合荷重の競合の様子を示す．初期状態では二つあった基底関数が CRBFN によって学習が行われることで競合が生じ，学習終了時には一つに減少していることがわかる．

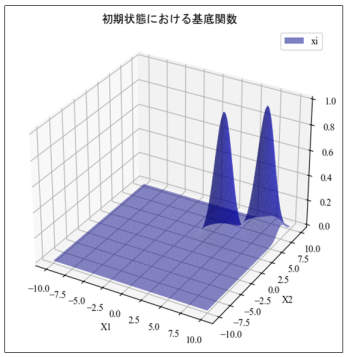

図 2.1: 初期状態の基底関数

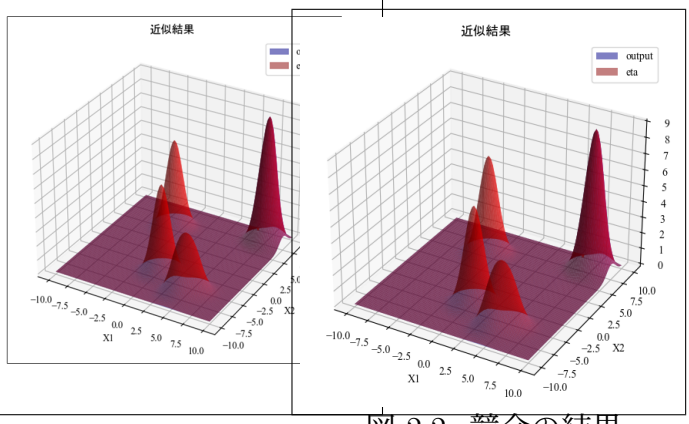

図 2.2: 競合の結果

§ 2.2　ターミナルアトラクタ

2.1 節ではシナプス可塑性方程式のような非線形ニューラルネットワークについて述べた．本節では，ある動的なシステム系が特別な条件を満たす場合に収束時間を指定することが可能になるターミナルアトラクタについて述べる．

まず，微分方程式における初期値問題に対する解の一意性を保証する条件であるリプシッツ条件について説明する．

リプシッツ条件

$f(t,x)$ を，閉区間 $\Omega = (t,x)||t-t_0| \leqq a, |x-x_0| \leqq b$ で定義された関数とする．ある定数 L があり，Ω 内の 2 点 $(t,x),(t,y)$ において，

$$|f(t,x) - f(t,y)| \leqq L|x-y| \tag{2.13}$$

となるとき，f はリプシッツ条件を満たす．

このリプシッツ条件が満たされると，それぞれの初期値問題に対して一意な解が存在し，その解の軌道は漸近的に平衡点に近づく．すなわち，軌道は平衡点に近づくだけで，有限時間内で平衡点に到達できない．これは CRBFN に置き換えると，競合に負けたシナプス結合荷重は有限時間で 0 に到達することができないということである．

そこで，Zak はこのリプシッツ条件を破るという考えに基づいて，解の一意性を破ることにより，有限時間内でニューラルネットワークが平衡点に収束することを示した [9] [10]．このような安定な平衡点をターミナルアトラクタと呼ぶ．このターミナルアトラクタを CRBFN に適用することで，収束時間の上限値を指定できるようにした．ここでは，時刻 t^* で平衡解へ収束できるように修正されたシナプス可塑性方程式を導出する．

いま，正定関数 $V(\mathbf{w})$ として

$$V(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\{\eta(\mathbf{x}_i) - s(\mathbf{x}_i)\}^2 \tag{2.14}$$

を定義する．正定関数 $V(\mathbf{w})$ はシナプス後発火頻度 $\eta(\mathbf{x}_i)$ と神経伝達物質放出量 $s(\mathbf{x}_i)$ の差を表す指標である．シナプス後発火頻度 $\eta(\mathbf{x}_i)$ が時間に依存しないと仮定すると，その時間変化が

$$
\begin{aligned}
\frac{dV(\mathbf{w})}{dt} &= \sum_{j=1}^{M} \frac{\partial V(\mathbf{w})}{\partial w_j} \frac{dw_j}{dt} \\
&= -\sum_{j=1}^{M} [\sum_{i=1}^{N} \eta(\mathbf{x}_i)\xi_j(\mathbf{x}_i) - \sum_{h=1}^{M}\sum_{i=1}^{N} \xi_j(\mathbf{x}_i)\xi_h(\mathbf{x}_i) w_h] \frac{dw_j}{dt} \\
&= -\sum_{j=1}^{M} w_j(\alpha_j - \sum_{h=1}^{M} \gamma_{jh} w_h)^2 \\
&\leq 0
\end{aligned}
\tag{2.15}
$$

となるため，正定関数 $V(\mathbf{w})$ が Lyapunov 関数となることがわかる．さらに，$V(\mathbf{w}) > 0$ であり $\frac{dV(\mathbf{w})}{dt} \leq 0$ であることから，このシステムは漸近安定であるということもわかる．

ここで，重み w_j の時間微分を

$$
\frac{dw_j}{dt} = -\Delta \frac{V(\mathbf{w})^c}{\sum_{j=1}^{M} (\frac{\partial V(\mathbf{w})}{\partial w_j})^2} \frac{\partial V(\mathbf{w})}{\partial w_j} \tag{2.16}
$$

とおくと，$V(\mathbf{w})$ の時間微分は

$$
\begin{aligned}
\frac{dV(\mathbf{w})}{dt} &= \sum_{j=1}^{M} \{ \frac{\partial V(\mathbf{w})}{\partial w_j} \frac{dw_j}{dt} \} \\
&= -\Delta V(\mathbf{w})^c \leq 0
\end{aligned}
\tag{2.17}
$$

となる．ただし，c は $0 < c < 1$ を満たす．$V(\mathbf{w})$ は時間とともに単調減少し，平衡点は漸近安定となることがわかる．このときの収束時間 t^* は，

$$
\begin{aligned}
t^* &= \int_0^{t^*} dt = \int_{V(\mathbf{w}_0)}^{V(\mathbf{w}_{t^*})} dV(\mathbf{w}) \frac{dt}{dV(\mathbf{w})} \\
&= \frac{V(\mathbf{w}_0)^{1-c} - V(\mathbf{w}_{t^*})^{1-c}}{\Delta(1-c)} \\
&\leq \frac{V(\mathbf{w}_0)^{1-c}}{\Delta(1-c)}
\end{aligned}
\tag{2.18}
$$

で与えられ，有限時間内で収束することがわかる．$V(\mathbf{w}_0)$ は重みの初期値を用いて計算した Lyapunov 関数 $V(\mathbf{w})$ の初期値で，$V(\mathbf{w}_{t^*})$ は平衡点での $V(\mathbf{w})$ の値である．$V(\mathbf{w}_{t^*}) = 0$ の場合，(2.18) 式の等号が成立する．そこで学習率 Δ を，

$$
\Delta = \frac{t^*(1-c)}{V(\mathbf{w}_0)^{1-c}} \tag{2.19}
$$

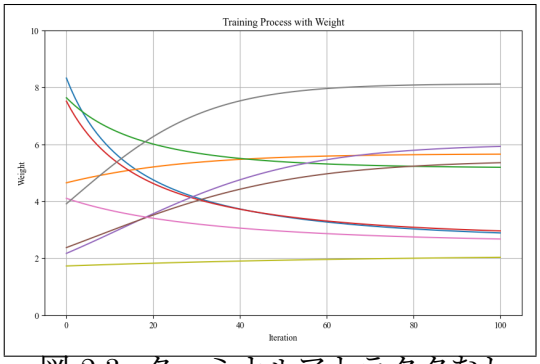

図 2.3: ターミナルアトラクタなし

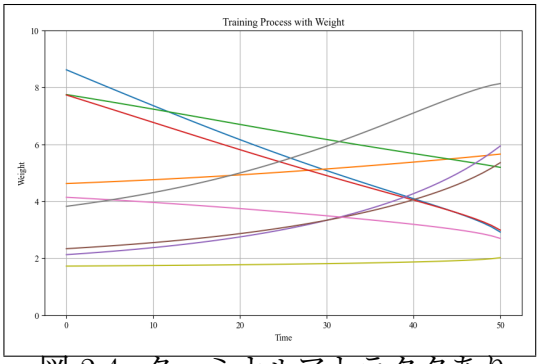

図 2.4: ターミナルアトラクタあり

とすると収束時間を指定できる．このターミナルアトラクタの概念を CRBFN における重みの学習に適用することで，初期状態において基底関数の数が多い場合でも従来の更新則よりも速く学習を終了させることが可能となる．

図 2.3，図 2.4 にターミナルアトラクタ適用前後の重み w の学習過程を示す．適用前の学習回数は 100 回であるのに対し，適用後は学習回数が 50 回の時点で適用前の学習終了時と同様の結果を得ることができており，収束時間を指定できていることがわかる．

§ 2.3　基底関数の複製

2.1 節で冗長なニューロンを削除する手法を提示してきた．これに対して，ニューラルネットワークでの関数近似に必要な数のニューロンが存在しない場合は，関数近似をすること自体が不可能となる．そこで，新たに必要なニューロンを追加する手法が提案されている [11]．ところが，このような手法では基準となるしきい値の決定自体が困難であることが予想できる．また，教師信号が動的に変化する環境では削除する手法と追加する手法を組み合わせて学習を行わなければならない．当然それぞれにとって良い手法を，ただそのまま組み合わせただけでは，動径基底関数の数が振動するなどして望ましい結果が得られるとは限らない．

そこで新しい動径基底関数を追加する手法がある．この手法は必要な動径基底関数を効率的に追加することができる．そして，先に提案された CRBFN にこの手法を組み合わせたニューラルネットワークが RC-RBFN である．この RC-RBFN は，環境の変化に適応する能力を備えたものとなっている．

まず，パラメータが従う確率密度関数の導出を行う．ここでは，CRBFN のシナプス結合荷重と平均ベクトル，標準偏差が学習終了時にとる同時確率密度 $p(\mathbf{w}, \mathbf{m}, \Sigma)$ を導出する．シナプス結合荷重 w^j を

$$y_j^2 = w_j \tag{2.20}$$

と変数変換する．y_j の定義域は任意の実数である．このとき，(2.6) 式は

$$\frac{dy_j}{dt} = (\frac{\alpha_j}{2} - \sum_{h=1}^{M} \frac{\gamma_{jh}}{2} y_h^2) y_j \tag{2.21}$$

となる．(2.21) は積分条件

$$\frac{\partial}{\partial y_h}\frac{dy_j}{dt} = \frac{\partial}{\partial y_j}\frac{dy_h}{dt} \tag{2.22}$$

を満たすため，ポテンシャル

$$U'(\mathbf{y},\phi) = -\sum_{j=1}^{M}\int_a^{y_j}(\frac{\alpha_j}{2} - \sum_{h=1}^{M}\frac{\gamma_{jh}}{2})y'_j dy'_j \tag{2.23}$$

を考えることができる．ただし $\mathbf{y} \equiv [y_1, y_2, ..., y_M]$ であり，パラメータ ϕ_j は平均ベクトルと共分散行列の集合 $\{\mathbf{m}_j, \mathbf{\Sigma}_j\}$ である．そして ϕ で集合 $\{\phi_1, \phi_2, ..., \phi_M\}$ を表す．変数 y_j の時間変化はポテンシャル $U'(\mathbf{y}, \phi_j)$ から，

$$\frac{dy_j}{dt} = -\frac{\partial U'(\mathbf{y},\phi)}{\partial y_j} \tag{2.24}$$

で導くことができ，関係式 (2.20) からポテンシャル $U'(\mathbf{y},\phi)$ は

$$U(\mathbf{w},\phi) = -\sum_{j=1}^{M}\{\frac{\alpha_j}{4} - \sum_{h\neq j}^{M}\frac{\gamma_{jh}}{4}w_j w_h - \frac{\gamma_{jj}}{8}w_j^2\} \tag{2.25}$$

と書き直すことができる．この結果，

$$E(\mathbf{w},\phi) = 4U(\mathbf{w},\phi) + \frac{1}{2}\sum_{i=1}^{N}\eta^2(\mathbf{x}_i) \tag{2.26}$$

であることが示される．ただし，$\mathbf{x}_i \equiv [x_1, x_2, ..., x_N]$ である．よって，累積 2 乗誤差関数 $E(\mathbf{w},\phi)$ の最小化はポテンシャル $U(\mathbf{w},\phi)$ の最小化と等価であることがわかる．

いま，(2.21) 式に従う y_j はポテンシャル $U'(\mathbf{w},\phi)$ の最急降下方向に更新される．その結果，ひとたび極小解に収束すると，そこから逃れることができなくなる．そこで，極小解から脱出させるための手法として，y_j の更新則を

$$y_j(t+\Delta t) = y_j(t) - \frac{\partial U(\mathbf{w},\phi)}{\partial y_j}\delta t + \sqrt{Q\Delta t}n_j(t) \tag{2.27}$$

のようにノイズを考慮し離散近似した見本過程で与えることが考えられる．ただし，$n_j(t)$ は独立な確率変数であり，平均 0，分散 1 の正規分布 $N(0, 1)$ に従う．Q は任意の正の定数である．このとき，学習終了時に CRBFN のシナプス結合荷重と平均ベクトル，標準偏差が満たす同時確率密度 $p_\beta(\mathbf{w},\phi)$ は

$$p_\beta(\mathbf{w},\phi) = Z_\beta^{-1}exp\{-\beta U(\mathbf{w},\phi)\} \tag{2.28}$$

で得ることができる．ここで $\beta = 2/Q$ である．Z_β は分配関数であり

$$Z_\beta = \int_{\mathbf{w}}\int_{\phi} exp\{-\beta V(\mathbf{w},\phi)\}d\mathbf{w}d\phi \tag{2.29}$$

で定義される．また，式 (21) はポテンシャル $V(\mathbf{w})$ と累積 2 乗誤差関数 $E(\mathbf{w})$ の関係式 (2.26) より

$$p_{\beta'}(\mathbf{w}) = Z_{\beta'}^{-1} exp\{-\beta' E(\mathbf{w})\} \tag{2.30}$$

と書き直すことができる．ここで，$\beta' = (2Q)^{-1}$ である．また，$Z_{\beta'}$ は分配関数である．

以上のようにして，パラメータが従う確率密度関数が導出できたことにより，与えられた条件のもとで累積 2 乗誤差関数 $E(\mathbf{w})$ を最小とするパラメータの値が検出できることを示す．ここでは，教師信号 $\eta(x)$ を一つの動径基底関数では累積 2 乗誤差関数 $E(\mathbf{w})$ を 0 にすることが不可能であるような非線形関数で与えることとする．

しかし，この動径基底関数のパラメータ m が従う条件付き確率密度関数

$$p_{\beta'}(m|w, \Sigma) = \frac{p_{\beta'}(w)}{\int_m p_{\beta'}(w) dm} \tag{2.31}$$

は導出することができる．

よって，あるパラメータをもつ動径基底関数が与えられた条件のもとで累積 2 乗誤差関数 $E(\mathbf{w}, \phi)$ を最小とするためには，パラメータ m を条件付き確率 $p_{\beta'}(m|w, \Sigma)$ を最大とする値に定めればよいことがわかる．また，もし同じ形質をもつ動径基底関数を一つ追加することができるなら，条件付き確率 $p_{\beta'}(m|w, \Sigma)$ を極大とするパラメータ m へ配置することが最も累積 2 乗誤差関数 $E(\mathbf{w}, \phi)$ を小さくできることもわかる．

一般に，式 (2.12) に従いパラメータ m_j を更新し続けると極小解にとらわれ，累積 2 乗誤差関数 $E(\mathbf{w}, \phi)$ の値を 0 にすることができないことがある．または，近似しようとしている非線形関数 $\eta(\mathbf{x}_i)$ の複雑さに対し，必要とされる動径基底関数が十分に存在していないときには，2 乗誤差関数 $E(\mathbf{w}, \phi)$ の値を 0 にすること自体が不可能である．

ところで，確率的な要素や未知の教師信号などが存在しないものとするなら，すべての入力ベクトル $\mathbf{x}_i$ ごとに動径基底関数を作成し，シナプス結合荷重が $w_i = \eta(\mathbf{x}_i)$ かつパラメータ $\mathbf{\Sigma}_i \to 0$ であるときに，パラメータ $\mathbf{m}_i$ が $\mathbf{x}_i$ となることで近似的に $\mathbf{0}$ とできる場合がある．もちろん，多くの問題ではすべての入力ベクトルについて動径基底関数を用意しなくても，このようなことが可能であるものと思われる．

そこで，累積 2 乗誤差関数 $E(\mathbf{w}, \phi)$ の値がある正数 $\epsilon > 0$ より大きな値に収束し，学習が収束したと判断されるときに，新たに必要な動径基底関数を追加する．この手法では，先ほど導出した確率密度関数を利用しているため，学習が収束した時点で得られている動径基底関数の一部の形質（シナプス結合荷重 w_j，パラメータ $\mathbf{\Sigma}_j$）が新たに追加される動径基底関数がもつパラメータに引き継がれている．そのため，効率的に最も累積 2 乗誤差関数を小さくするパラメータ $\mathbf{m}$ に動径底関数を追加していくことができる．なおかつ，最悪の場合にはすべての入力ベクトル $\mathbf{x}_i$ をパラメータ $\mathbf{m}_i$ とする動径基底関数を作成することができる．

ところで，累積 2 乗誤差関数 $E(\mathbf{w}, \phi)$ の最小化は各入力ベクトル $\mathbf{x}_i$ ごとに 2 乗誤差関数 $E(\mathbf{x}_i, \mathbf{w}, \phi)$ を最小化することに等価である．そこで，各入力ベクトル $\mathbf{x}_i$ に依存した平均ベクトル $\mathbf{m}_{j[i]}$ を考える．そして，学習収束の時点で得られている第 j 番目の動径基底関数に着目すると，入力ベクトル $\mathbf{x}_i$ の条件付き確率密度関数は

$$p_{\beta'}(\mathbf{x}_i|\mathbf{m}_{j[i]}, \phi'_j, \phi''_j) = Z_{\beta'}^{-1}(\mathbf{m}_j, \phi'_j, \phi''_j) exp\{-\beta' E(\mathbf{x}_i, \mathbf{m}_{j[i]}, \phi'_j, \phi''_j)\} \tag{2.32}$$

と導出できる．ここで，パラメータ ϕ'_j は着目した第 j 番目の動径基底関数のシナプス結合荷重 w_j と共分散行列 Σ_j の集合であり，パラメータ ϕ''_j は着目した第 j 番目の動径基底関数以外のシナプス結合荷重，共分散行列並びに平均ベクトルの集合である．以後は記法の簡便のため，パラメータ ϕ'_j とパラメータ ϕ''_j は省略する．また，分配関数は

$$Z_{\beta'}(\mathbf{m}_j) = \sum_{i=1}^{N} exp\{-\beta' E(\mathbf{x}_i, \mathbf{m}_{j[i]})\} \tag{2.33}$$

で定義される．

条件付き確率密度関数 $p_{\beta'}(\mathbf{x}_i|\mathbf{m}_{j[i]})$ は，確率の正規化と 2 乗誤差関数 $E(\mathbf{x}_i, \mathbf{m}_{j[i]})$ の条件付き期待値

$$\langle E(\mathbf{m}_j)\rangle_{\beta'} = \sum_{i=1}^{N} p_{\beta'}(\mathbf{x}_i|\mathbf{m}_{j[i]}) E(\mathbf{x}_i, \mathbf{m}_{j[i]}) \tag{2.34}$$

が一定となるという二つの制約のもとで，エントロピー

$$\langle E(\mathbf{m}_j)\rangle_{\beta'} = \sum_{i=1}^{N} p_{\beta'}(\mathbf{x}_i|\mathbf{m}_{j[i]}) E(\mathbf{x}_i, \mathbf{m}_{j[i]}) \tag{2.35}$$

を最大にする確率密度関数として導出できる．ここで，記号

$\langle \cdots \rangle_{\beta'}$ は $p_{\beta'}(\mathbf{x}_i|\mathbf{m}_{j[i]})$ を掛けて $\mathbf{x}_i$ に関する和をとる演算を表すものとする．このとき，自由エネルギーを

$$F_{\beta'}(\mathbf{m}_j) = -\frac{1}{\beta'} \log Z_{\beta'}(\mathbf{m}_j) \tag{2.36}$$

で定義すれば,

$$S_{\beta'}(\mathbf{m}_j) = -F_{\beta'}(\mathbf{m}_j) + \beta' \langle E(\mathbf{m}_j)\rangle_{\beta'} \tag{2.37}$$

と表すことができる．この式はエントロピー $S_{\beta'}(\mathbf{m}_j)$ を最大化する条件付き確率密度関数 $p_{\beta'}(\mathbf{x}_i|\mathbf{m}_{j[i]})$ は，自由エネルギー $F_{\beta'}(\mathbf{m}_j)$ を最小化するものであることを示している．このような自由エネルギーは，データのクラスタリングのための手法であるメルティング [8] においても同様に定義されている．メルティングとは，$m_{j[i]} = \mathbf{x}_i$ かつ β' が ∞ である初期状態から，徐々に β' を 0 へ近づけていきながら，パラメータ $m_{j[i]}$ を自由エネルギー $F_{\beta'}(\mathbf{m}_j)$ の最急降下方向に更新していくものである．その結果，パラメータ $m_{j[i]}$ は徐々に同じ値をとりはじめ，最終的に一つの値 $m_{j[i]} = m_j (\forall i)$ に収束する．

そこで，RC-RBFN ではパラメータ m_j の更新則を式 (2.13) の Δm_j の代わりに

$$\begin{aligned}\Delta \mathbf{m}_{\beta'} &= -\epsilon \sum_{i=1}^{N} \frac{\partial F_{\beta'}(\mathbf{m}_j)}{\partial \mathbf{m}_{j[i]}} \\ &= -\epsilon \sum_{i=1}^{N} \frac{\partial F_{\beta'}(\mathbf{m}_j)}{\partial Z_{\beta'}(\mathbf{m}_j)} \frac{\partial Z_{\beta'}(\mathbf{m}_j)}{\partial \mathbf{m}_{j[i]}}\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{N} p_{\beta'}(\mathbf{x}_i|\mathbf{m}_{j[i]})\Delta\mathbf{m}_{j[i]} \\
&= \langle \Delta\mathbf{m}_j \rangle_{\beta'}
\end{aligned} \tag{2.38}
$$

で与えることとする．ここで，

$$
\Delta m_{j[i]} = -\epsilon \frac{\partial E(\mathbf{x}_i, \mathbf{m}_{j[i]})}{\partial \mathbf{m}_{j[i]}} \tag{2.39}
$$

である．

特に $\beta' = 0$ であり，初期の状態が $\mathbf{m}_{j[i]} = \mathbf{m}_j(\forall i)$ である場合は

$$
\Delta_0 \mathbf{m}_j = \Delta \mathbf{m}_j \tag{2.40}
$$

であることが示される．この場合は，RC-RBFN のパラメータ $\mathbf{m}_j$ の更新則が従来の RBFN のパラメータ $\mathbf{m}_j$ の更新則そのものとなっていることがわかる．このとき，$\beta' = 0$ で固定したままパラメータを $\mathbf{\Sigma}_j \to 0$ にすると，$\Delta_0 \mathbf{m}_j = 0$ とするパラメータ $\mathbf{m}_{j[i]}$ は

$$
\sum_{i=1}^{N} \xi(\mathbf{x}_i, \mathbf{m}_{j[i]})(\mathbf{x}_i - \mathbf{m}_{j[i]})\{\eta(x_i) - s(\mathbf{x}_i, \mathbf{m}_{j[i]})\} = 0 \tag{2.41}
$$

を満たし，$\mathbf{m}_{j[i]} = \mathbf{x}_i(\forall i)$ であることがわかる．つまり，教師入力信号がパラメータ $\mathbf{m}_j$ の収束点として検出されることとなる．逆に，パラメータ Σ_j を固定したまま $\beta' \to \infty$ にすると，$\Delta_\infty \mathbf{m}_j = 0$ とするパラメータ $\mathbf{m}_{j[i]}$ は

$$
\sum_{i=1}^{N} exp\{-\beta' E(\mathbf{x}_i, \mathbf{m}_{j[i]})\}\xi(\mathbf{x}_i, \mathbf{m}_{j[i]})(\mathbf{x}_i - \mathbf{m}_{j[i]})\{\eta(\mathbf{x}_i) - s(\mathbf{x}_i, \mathbf{m}_{j[i]})\} = 0 \tag{2.42}
$$

を満たし，$\mathbf{x}_i(\forall i)$ を含む任意の値となることがわかる．これらの結果から，提案する RC-RBFN のパラメータ $\mathbf{m}_j$ の更新則 $\Delta_{\beta'}\mathbf{m}_j$ では，$\Delta_0 \mathbf{m}_j$ で従来の RBFN のパラメータ $\mathbf{m}_j$ の更新則を実現する．さらに，$\Delta_\infty \mathbf{m}_j$ とすることで，すべての入力ベクトル $\mathbf{x}_i$ の任意の値を安定な収束点とすることができる．

以上のことから，動径基底関数の複製を考慮した RC-RBFN の学習則を次のように提案する．

RC-RBFN の学習則

1. シナプス結合荷重 w_j を式 (8) のシナプス可塑性方程式により更新，パラメータ m_j を式 (32) の $\Delta_0 m_j$ により更新，パラメータ Σ_j は式 (7) により更新する．

2. 累積 2 乗誤差関数が $E(\mathbf{w}, \phi) \neq 0$ となったら学習終了．ある正数 $\epsilon > 0$ より大きな値に収束したなら STEP 3. へいく．

3. 学習収束時に得られているすべての動径基底関数について，β' を 0 から徐々に大きくしていきながら，式 (32) に従いパラメータ m_j を更新する．

4. 分岐により $\Delta_{\beta'} m_j = 0$ となる点が増えたとき，第 j 動径基底関数を第 p 動径基底関数として複製する．そのとき，シナプス結合荷重 w_p，パラメータ Σ_p 並びにパラメータ m_p は形質としてもとの第 j 動径基底関数のものを引き継ぎ，パラメータ m_p は新たに増えた点とする．STEP 1. へ戻る．

第3章

機械学習と最適化の関係

§ 3.1　代表的な機械学習の手法について

現在，機械学習は学術界だけでなく産業界においても幅広く用いられ，人工知能技術のコア技術として重要な役割をはたしている．機械学習はもともと人と同様の知能的機能を実現させるために研究開発が進められてきた分野であるが,「データから学ぶ」という過程とデータ解析・統計学の方法論がうまくマッチし，現在では狭い意味での人工知能としての使い方にとどまらず幅広いデータ科学の方法論として発展している．本節では機械学習の概要として機械学習の代表的な問題設定および機械学習を実行するためのツールについて述べる．

教師あり学習

教師あり学習は機械学習の中でも特に基本的な問題である．教師あり学習において目指すべき目標点は，ある入力 $x \in X$(画像やテキストなど) に対して，そのラベル $y \in Y$(画像に写っている物体や人など) を予測することにある．訓練データとして n 個の入出力データの組 $(z_i)_{i=1}^n = (x_i, y_i)_{i=1}^n$ が得られているとして，X から Y への関数 $h(x)$ を考える．損失関数は正解ラベル y と $h(x)$ の差を評価する関数として 2 乗誤差評価関数 $E(y, h(x))$ が用いられることが多い．

$$E(y, h(x)) = \frac{1}{n}\sum_{i=1}^{n}(y_i - h(x_i))^2 \tag{3.1}$$

教師あり学習の基本的な問題として回帰がある．回帰問題においては $Y = \mathbb{R}$ で $X = \mathbb{R}^p$ のときに $h(x)$ として線形関数 $h(x) = a^{\mathrm{T}}x$(ただし，$a \in \mathbb{R}^p$) を用いれば線形回帰になる．このときの訓練誤差の最小化

$$\min_{a \in \mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^{\mathrm{T}}a)^2 \tag{3.2}$$

は最小二乗法と呼ばれている．

教師なし学習

教師なし学習は教師あり学習と違い，入力に対するラベルが付いていない．このような問題は，分類に代表されるクラスタリングや，データの低次元への圧縮として考

えられる．クラスタリングの場合，観測データからその裏にある真の分布を推定することで実現されることが多い．例えば，混合ガウス分布の密度関数

$$p(x|\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K) = \sum_{k=1}^K \pi_k \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} exp(-\frac{1}{2}||x-\mu_k||^2_{\Sigma_k^{-1}}) \tag{3.3}$$

をデータにあてはめることでソフトクラスタリングが実現できる．ここで，$\{\pi_k\}_{k=1}^K$ は混合を表した変数で $\pi_k \leq 0 (k=1,...,K)$ かつ $\Sigma_{k=1}^K \pi_k = 1$ を満たす．$\{\mu_k\}_{k=1}^K$ は各クラスタの平均を表し，$\{\Sigma_k\}_{k=1}^K$ は各クラスタの分散共分散行列を表す．データ $(x_i)_{i=1}^n$ へのあてはめは基本的に負の対数尤度最小化

$$\min_{(\pi_k,\mu_k,\Sigma_k)_{k=1}^K} \sum_{i=1}^n -\log\left(p(x_i|\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K)\right) \tag{3.4}$$

で実現可能となる．

半教師あり学習

半教師あり学習は，教師あり学習と教師なし学習の中間に位置する問題として広く研究されている分野である．この学習手法は，データセットの一部に教師ラベル（正解ラベル）が与えられ，残りのデータには教師ラベルが得られていないという状況を前提としている．このような状況で，ラベル付きデータとラベルなしデータの両方を活用して判別機や予測モデルを構成することを目的とする．半教師あり学習の特徴は，ラベルなしデータを有効活用することでデータ全体の構造や分布を理解し，それを学習プロセスに活かす点にある．ラベル付きデータのみを用いた場合，データのサンプル数が少ない場合や分布が偏っている場合にモデルの性能が限定されることがある．しかし，ラベルなしデータを活用することで，データの全体像を推定しやすくなり，結果としてモデルの性能を向上させることが可能となる．これは特に，ラベル付け作業が時間的・費用的に困難な場合に有効であり，現実の多くの応用分野で利用されている．例えば，画像認識や自然言語処理の分野では，大量のデータが容易に得られる一方で，それに対して正確なラベルを付与することは非常に手間がかかり，無駄が生じてしまう．このような場合，半教師あり学習は少数のラベル付きデータから効率的に情報を抽出し，大量のラベルなしデータを活用してモデルを補強する手法として注目されている．

強化学習

強化学習は環境に合わせて自ら最適な行動を学ぶ学習問題である．例えばゲームを解く AI やロボットの動作学習などに用いられる．実際に，AlphaGo は強化学習を要素技術として用いている [5]．強化学習は能動的にデータを取得しながら学習するという点で，上記の学習方法とは異なっている．その応用範囲の広さから深層学習による応用研究が盛んに行われている．強化学習の基本的な手法である価値反復法について説明する．強化学習では状態 $s \in S$(ゲームの局面など) と行動 $a \in \mathbb{A}$(その局面に対して行う行動) が基本要素になる．$P(s'|s,a)$ を状態 s で行動 a を取ったときに，次の時刻

に状態 s' に移る遷移確率を表し，$R(s)$ で状態 s に到達したときに得られる報酬を表す．例えば迷路を解く問題であれば，$R(s)$ としてゴールすることで高い報酬が得られる関数に設定する．ある方策 π(各状態で次に起こす行動を決める関数) に従って行動するときの報酬の総和の期待値を

$$U^{\pi}(s) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi, s_0 = s] \tag{3.5}$$

と定義する．これは，状態 s から初めて未来に得られる報酬の総和であり，$\gamma < 1$ は時間割引を表す係数である．強化学習が目標とするのは報酬 $U^{\pi}(s)$ を最大にする方策 π を求めることにある．もし，$U^{\pi}(s)$ が分かっているのなら，最適な方策は期待報酬を最大化する行動を取るので，

$$\pi^*(s) = \underset{a \in \mathbb{A}}{\mathrm{argmax}} \Sigma_{s' \in S} P(s'|s,a) U^{\pi^*}(s) \tag{3.6}$$

を満たすはずである．このとき，U^{π} の定義から

$$U^{\pi^*}(s) = R(s) + \gamma \max_{a \in \mathbb{A}} \sum_{s' \in S} P(s'|s,a) U^{\pi^*}(s') \tag{3.7}$$

が満たされる．これを Bellman 方程式という．しかし，実際は π^* も $U^{\pi^*}(s)$ も分からないので，状態空間を遷移しながら Bellman 方程式に従って各遷移ごとに $U^{\pi}(s)$ を更新し収束するまで続ける．このような方法を価値反復法と呼ぶ．価値反復法は $P(s'|s,a)$ を知っている必要があるが，実際はこれも推定する必要がある．これも考慮した学習方法が Q-学習である．また，深層学習を用いた Q-学習の方式として Deep Q-network が提案されている．

§ 3.2　エネルギー関数の最小化手法

機械学習はポテンシャルやエネルギー関数を最小化するための探索手法によって種類分けされる場合がある．エネルギー関数とは，学習問題における目標を数学的に定式化したものであり，モデルが最適な状態に達するための基準を示すスカラー値の関数である．例えば，回帰問題では二乗誤差，分類問題では交差エントロピーがエネルギー関数として用いられることが多い．

最急降下法

最急降下法は，関数の傾き（一階微分）のみを利用して，関数の最小値を探索するための連続最適化問題における勾配法のアルゴリズムの一つである．この手法はシンプルかつ直感的であり，様々な最適化問題に応用可能であるため，広く利用されている．最急降下法では，まず目的関数 $f(\mathbf{x})$ が与えられる．この関数は n 次元ベクトル $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ を引数として定義されるスカラー値の関数であり，その極小値を求めることを目指す．

勾配法に基づくアルゴリズムでは，反復法を用いて解を段階的に更新することによって，最適解に近づけていく．最急降下法では，現在の解の位置 $\mathbf{x}^{(k)}$ が与えられたとき，その勾配（すなわち $\mathrm{grad} f(\mathbf{x}^{(k)})$）の方向に沿って，$\mathbf{x}$ を次のように更新する．

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha \mathrm{grad} f(\mathbf{x}^{(k)}) \tag{3.8}$$

ここで，$\mathrm{grad} f(\mathbf{x})$ は関数 f の勾配ベクトルを表し，各変数 x_i についての偏微分の結果を並べたものである．α は学習率と呼ばれるパラメータであり，更新量を制御する重要な役割を持つ．通常，α は小さな正の定数として設定されるが，この値の選び方はアルゴリズムの性能に大きな影響を及ぼす．

たとえば，学習率 α が大きすぎる場合，更新ステップが急激すぎて目的関数の値が発散し，最適解に到達できない可能性がある．逆に，α が小さすぎる場合，更新ステップが非常に緩やかになり，収束が遅くなる．そのため，適切な学習率を選定することは最急降下法を効果的に活用するための重要な課題である．

具体例として，$f(x) = x^2$ という単純な二次関数を考える．この関数の最小値は $x = 0$ で達成されるが，$\alpha > 1$ のような大きな値を選ぶと，反復計算の結果として解が発散し，かえって悪い方向へ進む可能性がある．一方で，$\alpha = 0.1$ などの適切な値を選べば，反復ごとに解が最小値へと収束していく様子が観察される．

共役勾配法

共役勾配法は対称正定値行列を係数とする連立一次方程式を解くためのアルゴリズムである．反復法として利用され，直接法では大きすぎて取り扱えない大規模な疎行列を解くために利用される．そのような問題は偏微分方程式などを数値的に解く際に常に現れる．また，共役勾配法は，エネルギー最小化などの最適化問題を解くために用いることもできる．

対称正定値行列 $\mathbf{A}$ を係数とする n 元連立一次方程式

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.9}$$

の解を $\mathbf{x}^*$ とする．非零ベクトル $\mathbf{u}, \mathbf{v}$ が

$$\mathbf{u}^{\mathrm{T}}\mathbf{A}\mathbf{v} = 0 \tag{3.10}$$

を満たすとき，$\mathbf{u}, \mathbf{v}$ は $\mathbf{A}$ に関して共役であるという．$\mathbf{A}$ は対称正定値なので，左辺から内積

$$\begin{aligned} \langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{A}} &= \langle \mathbf{A}^{\mathrm{T}}\mathbf{u}, \mathbf{v} \rangle \\ &= \mathbf{u}^{\mathrm{T}}\mathbf{A}\mathbf{v} \end{aligned} \tag{3.11}$$

を定義することができる．この内積に関して 2 つのベクトルが直交するなら，それらのベクトルは互いに共役である．この関係は対称で，$\mathbf{u}$ が $\mathbf{v}$ に対して共役であれば，$\mathbf{v}$ も $\mathbf{u}$ に対して共役である．

$\{\mathbf{p}_k\}$ を n 個の互いに共役なベクトル列とする．$\mathbf{p}_k$ は基底 $\mathbf{R}^n$ を構成するので，$\mathbf{A}x = \mathbf{b}$ の解 $\mathbf{x}^*$ をこの基底で展開すると，

$$\mathbf{x}^* = \sum_{i=1}^{n} \alpha_i \mathbf{p}_i \tag{3.12}$$

と書ける．ただし係数は，

$$
\begin{aligned}
\mathbf{A}x^* &= \sum_{i=1}^{n} \alpha_i \mathbf{A} p_i \\
&= \mathbf{b} && (3.13) \\
\mathbf{p}_k^{\mathrm{T}} \mathbf{A} x^* &= \sum_{i=1}^{n} \alpha_i \mathbf{p}_k^{\mathrm{T}} \mathbf{A} p_i \\
&= \mathbf{p}_k^{\mathrm{T}} \mathbf{b} && (3.14) \\
\alpha_k &= \frac{\mathbf{p}_k^{\mathrm{T}} \mathbf{b}}{\mathbf{p}_k^{\mathrm{T}} \mathbf{A} p_k} \\
&= \frac{\langle \mathbf{p}_k, \mathbf{b} \rangle}{\langle \mathbf{p}_k, \mathbf{p}_k \rangle_{\mathbf{A}}} && (3.15)
\end{aligned}
$$

で与えられる．以上より，まず n 個の共役な方向を見つけ，それから係数 α_k を計算すればよいことがわかる．

線形計画法

線形計画法は，数理計画法の一分野であり工学，経済学，経営学，物流など幅広い分野で利用されている最適化手法である．この方法は，いくつかの 1 次不等式および 1 次等式を満たす変数の値の中から，ある 1 次式（目的関数）の最大値または最小値を求めることを目的とする．線形計画法において対象となる最適化問題を線形計画問題という．

数学的には，線形計画問題は目的関数と制約条件がすべて線形で表現される最適化問題に分類される．例えば，以下のような問題設定を考える．2 変数 $x_1(\geq 0)$ および $x_2(\geq 0)$ に対して，与えられた定数係数 a_{ij}，b_i，c_j を用いて，不等式制約条件

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 &\leq b_1 \\
a_{21}x_1 + a_{22}x_2 &\leq b_2
\end{aligned}
$$

を満たす解を求める．この制約条件のもとで，次式で表される目的関数

$$
c_1 x_1 + c_2 x_2 \tag{3.16}
$$

の最大値または最小値を求める問題が，典型的な線形計画問題となる．

§ 3.3　処理高速化のための手法

本研究のプログラムの開発に用いた Python は，高い可読性と豊富なライブラリを持つ汎用プログラミング言語であり，学術研究やプロトタイピングに適した選択肢である．しかし，Python はインタプリタ型の言語であり，コードを実行時に逐次的に解釈するためにコ

ンパイル型の言語であるCやC++と比較すると，処理速度が遅くなる場合がある．特に，大規模な数値計算や複雑なアルゴリズムを多用する場合にはこの性能差が顕著となる．

この欠点を補うため，近年ではPythonの実行速度を向上させるための高速化ライブラリが数多く開発されている．例えば，数値計算を効率化するためのNumPyや，PythonコードをJust-In-Timeコンパイルする Numbaが挙げられる．さらに，PythonコードをC言語に変換してコンパイルするCythonがある．

以下にNumbaを使用した場合と未使用の場合の比較結果を示す.

Numba

NumbaはPythonコードの高速化を助けるためのモジュールである．Numbaを利用することで，PythonコードをJust-In-Time(JIT)コンパイルし，CPUやGPU上でネイティブマシンコードとして実行することが可能になる．Numbaモジュールには，JITデコレーターが提供されており，このデコレーターを関数やメソッドに付加することでその関数がJITコンパイルされる．JITコンパイルによって，Pythonの動的型付けによるオーバーヘッドが排除され，実行速度が大幅に向上する．特に，ループを多用する数値計算やデータ処理タスクで顕著な効果を発揮する．

JITデコレーターには「nopython=True」オプションを指定することもでき，このオプションを設定することで完全なネイティブコードが生成され，さらなる性能向上が期待できる．また，GPUを活用した並列処理を可能にするcuda.jitデコレーターや，並列化を自動的に行う@jit(parallel=True)といった高度な機能も提供されている．

Numbaを使用することで，PythonコードのパフォーマンスをCやFortranと同等の水準に引き上げることができるため，科学計算や機械学習，シミュレーション，データ分析など計算負荷の高いアプリケーションにおいて非常に有用である．従来の手法ではボトルネックとなっていた処理の高速化を容易に実現できる点が，Numbaの大きな特徴である．

また，並列分散処理を簡単に実現するRayライブラリも広く利用されている．

Ray

RayはPythonベースのオープンソースの並列および分散計算フレームワークで，効率的で拡張性のあるアプリケーションを構築するために設計されている．RayはシンプルなAPIを提供しつつ，高速な並列処理を可能にし，クラスタ環境でもシームレスに動作する．機械学習やデータ処理，シミュレーションなどのさまざまな用途に適している．

第4章

提案手法

§ 4.1　新たに提案されるターミナルアトラクタの定義式

本研究では，2.2 節で述べた Zak によるターミナルアトラクタではなく，奥原によって提案されたターミナルアトラクタ [2] を用いて基底関数の重みの学習を行う．そこで，本節では奥原によって提案されたターミナルアトラクタについて述べる．

まず，望ましい時刻 t^* で収束するシナプス結合荷重の時間変化を Lyapunov 関数を用いて規定する．そこで，Lyapunov 関数の時間変化を

$$\frac{dV(\mathbf{w})}{dt} = -\frac{V(\mathbf{w}_0)^R V(\mathbf{w})^{\frac{1}{r}}}{Rt^*} \tag{4.1}$$

で定義する．ここで，r は任意の奇数であり，$R = (r-1)/r$ である．このような定義が可能になったのは，Lyapunov 関数が導出され，望ましい出力が動径基底関数を定数倍して足し合わせることで実現できる特別の場合を考えているからである．シナプス可塑性方程式は

$$\frac{dw_j}{dt} = \Delta(\alpha_j - \sum_{h=1}^{M} \gamma_{jh} w_h) w_j \tag{4.2}$$

とすることができる．このとき，Lyapunov 関数の時間変化は

$$\frac{dV(\mathbf{w})}{dt} = -\Delta \sum_{j=1}^{M} w_j (\alpha_j - \sum_{h=1}^{M} \gamma_{jh} w_h)^2 \tag{4.3}$$

となる．ここで，Δ は (4.1) 式と (4.3) 式から決定することが可能で，

$$\Delta = \frac{1}{\sum_{j=1}^{M} w_j (\alpha_j - \sum_{h=1}^{M} \gamma_{jh} w_h)^2} \frac{V(\mathbf{w}_0)^R V(\mathbf{w})^{\frac{1}{r}}}{Rt^*} \tag{4.4}$$

と導出される．以上のことから望ましい時刻 t^* で平衡解へ収束するシナプス可塑性方程式を

$$\frac{dw_j}{dt} = \frac{(\alpha_j - \sum_{h=1}^{M} \gamma_{jh} w_h) w_j}{\sum_{j=1}^{M} w_j (\alpha_j - \sum_{h=1}^{M} \gamma_{jh} w_h)^2} \frac{V(\mathbf{w}_0)^R V(\mathbf{w})^{\frac{1}{r}}}{Rt^*} \tag{4.5}$$

で定義することができる．

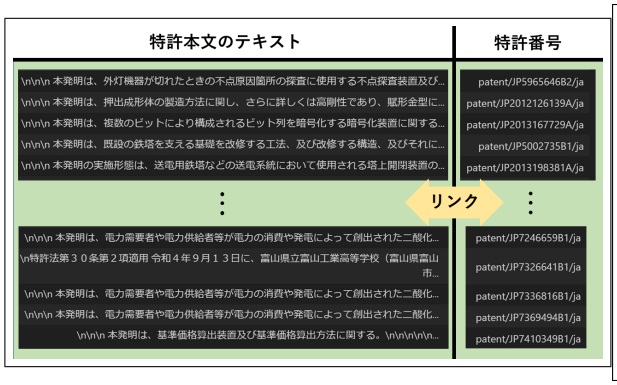

図 4.1: テキストデータのフォーマット

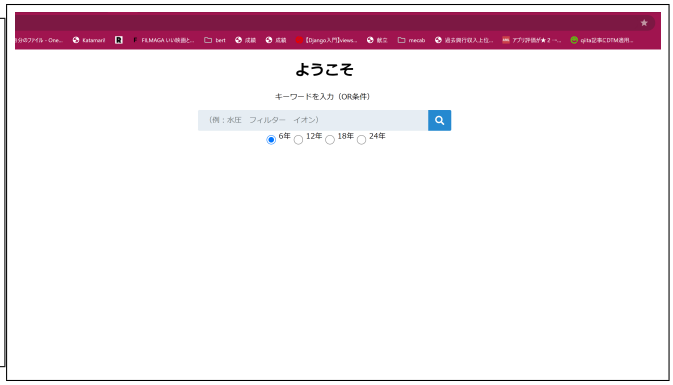

図 4.2: システムのフロントページ

1. 目的関数に従うサンプルデータ群を取得する．
2. 基底関数の中心位置，標準偏差の学習を行う．
3. 学習が進んで累積二乗誤差関数の変化量が十分小さくなったとき，基底関数を複製する．
4. ターミナルアトラクタを適用して重みの学習を行う．
5. 累積二乗誤差関数が収束判定以下の値であれば学習終了，収束判定以上であれば 2. に戻る．

本研究では，RC-RBFN を用いて基底関数の各パラメータと重みを学習することによって，与えられたデータ群が従う目的関数を近似する．近似に必要な各パラメータの学習方法は重みがターミナルアトラクタを適用した適者生存型学習則，基底関数の中心位置を複製アルゴリズム，基底関数の分散共分散行列を一撃アルゴリズムである．まず，分散共分散行列を一撃アルゴリズムによって導出する．次に中心位置を学習し，初期状態の基底関数を目的関数が位置する場所に配置する．そして，適者生存型学習則によって近似に不適な基底関数の重みを 0 にし，適した基底関数の重みは目的関数との誤差が小さくなるように学習する．

§ 4.2　Python の高速化ライブラリ

本研究のプログラムの開発に用いた Python は，高い可読性と豊富なライブラリを持つ汎用プログラミング言語であり，学術研究やプロトタイピングに適した選択肢である．しかし，Python はインタプリタ型の言語であり，コードを実行時に逐次的に解釈するためにコンパイル型の言語である C や C++と比較すると，処理速度が遅くなる場合がある．特に，大規模な数値計算や複雑なアルゴリズムを多用する場合にはこの性能差が顕著となる．

この欠点を補うため，近年では Python の実行速度を向上させるための高速化ライブラリが数多く開発されている．例えば，数値計算を効率化するための NumPy や，Python コードを Just-In-Time コンパイルする Numba が挙げられる．さらに，Python コードを C 言語に変換してコンパイルする Cython や，並列処理を簡単に実現する multiprocessing ライブラリも広く利用されている．

本研究においては，これらの高速化ライブラリの 1 つの Numba を使用することで計算コストを低減し，効率的なプログラム開発を実現した．これにより，Python の持つ可読性や

開発の柔軟性を維持しつつ，処理速度の課題を克服することが可能となった．
以下に Numba を使用した場合と未使用の場合の比較結果を示す．

Numba

Numba は Python コードの高速化を助けるためのモジュールである．Numba を利用することで，Python コードを Just-In-Time(JIT) コンパイルし，CPU や GPU 上でネイティブマシンコードとして実行することが可能になる．Numba モジュールには，JIT デコレーターが提供されており，このデコレーターを関数やメソッドに付加することでその関数が JIT コンパイルされる．JIT コンパイルによって，Python の動的型付けによるオーバーヘッドが排除され，実行速度が大幅に向上する．特に，ループを多用する数値計算やデータ処理タスクで顕著な効果を発揮する．
JIT デコレーターには「nopython=True」オプションを指定することもでき，このオプションを設定することで完全なネイティブコードが生成され，さらなる性能向上が期待できる．また，GPU を活用した並列処理を可能にする cuda.jit デコレーターや，並列化を自動的に行う@jit(parallel=True) といった高度な機能も提供されている．
Numba を使用することで，Python コードのパフォーマンスを C や Fortran と同等の水準に引き上げることができるため，科学計算や機械学習，シミュレーション，データ分析など計算負荷の高いアプリケーションにおいて非常に有用である．従来の手法ではボトルネックとなっていた処理の高速化を容易に実現できる点が，Numba の大きな特徴である．

§ 4.3 関数近似器としての活用

関数近似器の性能評価を行うにあたり，ベンチマーク関数を用いる．この節では，ベンチマーク関数の概要とその利用意義について述べる．
ベンチマーク関数とは，最適化アルゴリズムや関数近似手法の性能を比較・評価するために設計された，特定の数理モデルである．これらの関数は，実験環境において目的関数として機能し，アルゴリズムや手法がどの程度正確に近似・最適化できるかを定量的に測定する指標を提供する．関数近似や最適化の分野において，ベンチマーク関数の活用は非常に重要であり，特に新しいアルゴリズムやモデルの開発段階において，その性能を客観的に比較するための標準的な手段として広く用いられている．
ベンチマーク関数には、以下のような特徴がある：

1. 数式的に明確な構造：明確な数式で定義され，解析的な解が求められるものが多いため結果の正確性を容易に評価できる．

2. 多様な難易度：ベンチマーク関数は線形性，非線形性，周期性，急峻な変化，高次元性，ノイズの有無など，さまざまな特性を持つ関数を含む．これにより、関数近似器がさまざまな状況にどれだけ対応できるかをテストすることが可能となる。

3. 汎用性：多次元に拡張可能なものが多く，高次元問題への適用にも利用可能である．この特性により，現実の複雑な問題を想定したシナリオにおいても性能を測定することが可能である．例えば，次元数を変更することで，モデルの性能が次元の増加に伴いどのように変化するかを観察することができる．

本研究では，これらの特性を活かして関数近似器の性能を客観的かつ多角的に評価することを目的としている．そのため，異なる構造や難易度を持つ複数のベンチマーク関数を選定し，それぞれの関数に対して近似精度や学習速度などの指標を計測する．これにより，関数近似器が持つ特性や適用可能な範囲を明確にし，さらに改良すべき点を浮き彫りにすることが期待される．

具体的には，以下のようなベンチマーク関数を用いる予定である：※画像挿入予定

Sphere 関数

Sphere 関数は非常に単純で解析的に理解しやすい構造を持つベンチマーク関数であり，アルゴリズムの基礎的性能を測定するために広く使用されている．この関数の定義は次の通りである：

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2 \tag{4.6}$$

ここで，$\mathbf{x} = (x_1, x_2, ..., x_n)$ は入力ベクトルであり，次元数 n に応じて計算が行われる．Sphere 関数のグローバル最適解は，入力ベクトルのすべての要素が 0 の場合 (すなわち $\mathbf{x} = \mathbf{0}$) であり，そのときの関数値は $f(\mathbf{x}) = 0$ である．この関数は，連続的かつ単調増加する性質を持つため，アルゴリズムが最適解に収束する能力や学習速度を測定する際に適している．また，高次元空間でもその性質が変わらないため，次元数の増加に伴うアルゴリズムの性能変化を観察するのに用いられる．さらに，シンプルな構造を持つことから，複雑な関数と組み合わせて実験を行う際の基準としても利用される．

Ackley 関数

Ackley 関数は、探索空間の性質が滑らかな領域と急激に変化する領域を含んでいるため、グローバル最適解を見つける難易度が高いことで知られている。この関数は、多様なアルゴリズムの適応力やグローバル探索能力を評価するために使用される。Ackley 関数の数式は以下のように表される：

$$f(\mathbf{x}) = a - a\exp(-b\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n} cos(cx_i)) + e \tag{4.7}$$

ここで $a = 20$，$b = 0.2$，$c = 2\pi$ とするのが一般的であり，$\mathbf{x} = (x_1, x_2, ..., x_n)$ は入力ベクトルである．Ackley 関数の特徴として，中心付近に幅広い滑らかな平坦な領域があり，その外側に急激な変化を持つ環状の構造がある．

第5章

実験結果並びに考察

§ 5.1　実験の概要

本研究における提案手法においてIPランドスケープ実施への支援が行えているかに注目して評価実験を行う．IPランドスケープの取り組みとして，技術の特徴を生かした有望用途の探索を行うことを目的とする．今回の評価実験では，IPランドスケープの一環として特許情報の探索およびその中から知見を発見することを目的として検証を行う．実際には，自社の保有している技術の使い道を探すという題材をもとに検証を行う．

そのため,「ブロックチェーン技術を活用した決済システムの特許分析」という事例を設けて実験を行う．実際にシステムの入力欄に「ブロックチェーン」「決済システム」という単語を検索欄に入れ検索年数を6年にして実行を行った．

UMAPに設定するパラメータはn_neighborsの値はあまり大きなクラスターにしてしまうとそれぞれの要素の数が多くなってしまい大まかな分類になってしまうことを踏まえ「25」に設定し，min_distの値は出力されるクラスターの密度やスペースの具合を加味し「0.1」，metricは今回用いるデータがテキストを定量化したデータであるため「cosine」に設定して実験を行った．また，3Dグラフを描画するときに指定できる大きさの設定は表示する共起関係の数であり，小は1000，中は2000，大は3000個の共起関係を表示している．

さらに，実際にシステムを使用してもらい，アンケートに答えてもらう．アンケートの項目は全部で10個あり，その10個には必ず答えてもらう．以上の項目を5段階評価のリッカート尺度による評価を行ってもらう．今回のアンケートでは5段階評価のうち，1を「まったく満足していない」，2を「あまり満足していない」，3を「どちらでもない」，4を「やや満足している」，5を「非常に満足している」といったようなアンケートを行った．

また実際のシステム利用時の大剣を直接フィードバックできるように，アンケートと同時にコメントを入力できる欄を設けて置き，実際に入力したキーワードなどを自由にコメントができるようにする．実際のアンケート内容を表??に示す．表??を見てわかるように，アンケートの半分についてはシステムの使用感についての質問を設定している．システムの使用感についての質問から客観的なシステムの使用感に関する質問を行っている．残りの半分はそれぞれの機能についての質問を行っている．システムから出力されたものが適切であるかに関する質問を行う．

調査の対象は同研究室の学部4年生，3年生の合計5人に実際に開発したシステムを使用してもらい，アンケートを答えてもらった．実験では，利用者に実際にキーワードを考えてもらい，それを検索欄に入力することを行った．また，取得する年数に関しては，まずは6年を指定してもらい，得られた結果が少ない場合は徐々に年数を上げていくというこ

とを行った．

この評価を通じて，本手法が IP ランドスケープの支援に役立つ実践的支援機能を果たしているかどうかの確認を行う．

§ 5.2　実験結果と考察

まず，事例を設けての実験についての結果と考察を行う．この時 1588 個の特許をスクレイピングすることができた．実際に出力された散布図は図 5.1 となり，クラスターは 17 個となった．それぞれのクラスターに対応したタイトルは図 5.2 のような出力となった．

クラスターにおいて 3D グラフからブロックチェーンの使い道を検討した．クラスター 4 を選択した際に出力された 3D グラフを図 5.3 に示す．出力された 3D グラフから「コンサート」や「グッズ」などから「ファン通貨」という単語につながりがあるこが把握的た．この関係性から，アーティストのファン特有の通貨をブロックチェーン技術を用いて作り出し，ファンのコミュニティー内でその通貨を発行することが考えられる．通貨を獲得するには，アーティストのコンサートなどに行ったり，それらの情報を外部に発信したときなどがあげられる．ファンがこれらの活動を行うことで，通貨を獲得することができる．この通貨を用いることでファンコミュニティ独自の決済システムを採用することが可能となる．

ファンはこの通貨を使用してアーティスのグッズやコンサートチケットなどを購入することができる．また，通貨の利用により，ファン同士の交流やコミュニティの活性化を促進することも期待することができる．このようなファンコミュニティ独自の通貨システムは，アーティストとファンの絆を深めるだけでなく，ファンの忠誠心や参加意欲を高める効果も期待できる．さらに，ブロックチェーンを用いることで，通貨の取引履歴や所持数などの透明性や信頼性を確保することもできる．

したがって，出力されたグラフの結果から，アーティスト特有の通貨を作り出し，ファンコミュニティ内で利用することで，独自の決済システムを実現するということを考えることができる．

最後にアンケート調査における結果と考察を行う．

一個目に,「システムの操作性はわかりやすいか」という質問を行った．結果として，全体的に好印象な評価を得ることができた．この結果から．システムの操作性は容易であることが考えられる．システム全体的に直観的に操作できるということが考えられる．

二個目に,「システムの機能は理解しやすいか」という質問を行った．結果として，好印象な評価が四人であったが，残りの一人に関してはどちらでもないという意見であった．この結果からシステムを初めて使う人でもある程度すぐにシステムの機能を理解することができるということがわかる．また，もう少し画面に出力されているものがどういうものなのかを説明することで，よりシステムの機能を理解してもらうことができると考えられる．

三個目に,「レイアウトは適切か」という質問を行った．結果として，全体的に好印象な評価を得ることができた．この結果から．グラフやボタン，テキストなどの表示位置が適切であると考えることができる．

四個目に,「デザインは見やすいか」という質問を行った．結果として，全体的に好印象な評価を得ることができた．この結果から，本システムの画面全体を通してのデザインが

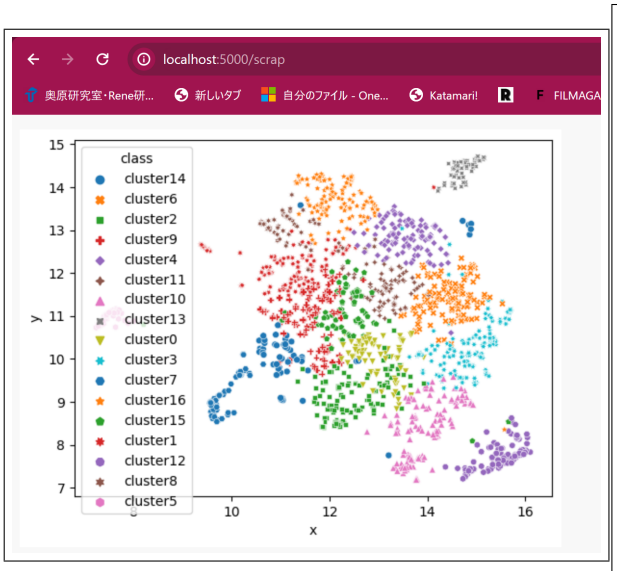

図 5.1: ベクトル化の結果

図 5.2: 出力されたタイトル

見やすいということが考えられる．画面に表示する情報は必要最低限にしているためであると考えることができる．一方で，二個目の質問で考察したように，システム機能の説明を付け加えることを考えると，デザインの構成を考える必要がある．

五個目に,「ストレスなく利用することができたか」という質問を行った．結果として，全体的にあまり好印象な結果を得ることができなかった．この結果から，システムの利用においてはストレスを感じるということが考えられる．その理由として，システム全体の処理時間の遅さがあげられる．システム全体の処理時間が遅いことで，ユーザーは待っている時間がいこと，またロード画面が静止画であるため，いつまで待てばいいのかがわかないことなどが考得られる．この解決策として，マルチプロセスや分散処理を用いたスクレイピングの更なる高速化や，分かち書きの高速化などがあげられる．また，3D グラフにおける描画処理も遅いため 3D グラフの描画手法についても検討が必要である．さらに，ロード画面に進捗バーなどを追加することで，処理が長くなってもあまりストレスなく利用することができると考える．

六個目に,「クラスターの提示は適切であるか」という質問を行った．結果として，肯定的な意見が三件，否定的な意見が一件，どちらでもないが一件となった．この結果から，入力するキーワードによって，出力されるクラスターが異なり，キーワードによってはあまり，適していないクラスターが出力されていることが考えられる．この理由として，今回用いたクラスタリング手法である k-means では外れ値による影響が多く，データによっては，適していないクラスターが含まれる可能性がある．そこで，外れ値に強いクラスタリング手法を用いることで，これらの問題は解決すと考えられる．

七個目に,「共起語ネットワークは適切であるか」という質問を行った．結果として，肯定的な意見が三件，否定的な意見が一件，どちらでもないが一件となった．この結果から，入力するキーワードの違いや，取得されるデータの違いによって，共起語ネットワークの精度が異なることがあげられる．今回用いた simpson 係数でしきい値を設定したが，このしきい値が場合によってあまり適していないものであるということが考えられる．そこで，すべての場合において適するようなしきい値に変更することで解決できると考えられる．

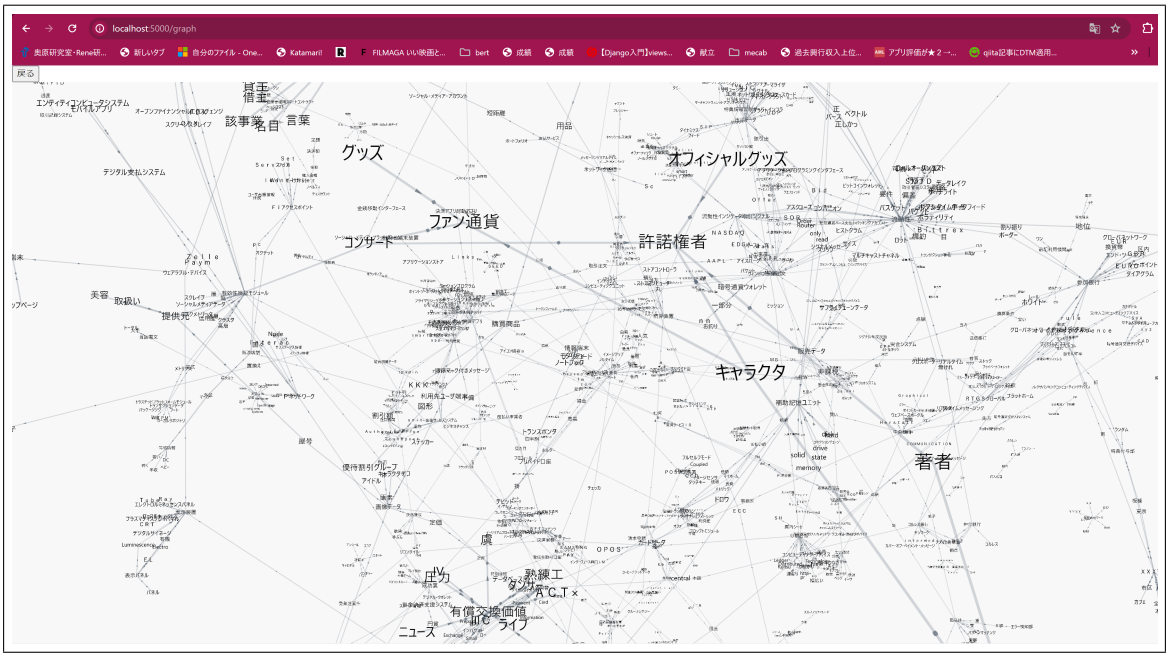

図 5.3: 出力された 3D グラフ

八個目に,「3D グラフによる出力は適切であるか」という質問を行った．全体的に好印象な評価を得ることができた．この結果から，3D グラフによる共起語ネットワークの可視化は有用であるということがわかる．3D グラフで出力することで，よりインタラクティブなグラフになることが考えられる．

九個目に,「効率的な特許探索を行えそうか」という質問を行った．結果として，全体的に好印象な評価を得ることができた．この結果から，システムを用いずに行う特許探索よりも，システムを用いた特許探索の方が効率的であるということができる．特許全体を羅列するだけではなく，散布図による可視化や，共起語ネットワークによる可視化を行うことで，効率的な特許探索を行うことができると考える．

十個目に,「新しい知見を発見できそうか」という質問を行った．結果として，全体的に好印象な評価を得ることができた．この結果から，実際にシステムを利用することで，新しい知見を発見できると考えることができる．

また，自由記述では,「選択できる年数を増やした方がいい」という意見があり，入力されるキーワードによって，取得される特許の数が違い 24 年では十分な数の特許を取得することができなかったことが考えられる．そこで，もう少し取得する年数を増やすか，それらのキーワードが含まれる特許が多く含まれる年からのスクレイピングなどがあげられる．

第6章

おわりに

本研究では，莫大な量の特許群を分析することで，IP ランドスケープ実施の支援を行うシステムの開発を行った．既存の特許プラットフォームでは，膨大な特許文献データを一気に集積し，特許全体をビッグデータとして分析を行うことは容易ではない．本システムでは，大量の特許文を効率的に収集し，特許情報を整理整頓し，そのうえでデータマイニングと機械学習の手法を駆使し，特許群から有用な知的財産情報を抽出，解析することを目的とした．このシステムを活用することで IP ランドスケープの調査や技術トレンド分析など，大規模な特許情報を活用した様々な業務支援を行った．

本研究で提案したシステムの特徴をまとめる．一つ目の特徴は，莫大な特許文章群をベクトル表現に変化し，そのベクトル空間上で潜在的なクラスタリングを行ったことである．現在までに蓄積された膨大な特許文章は，技術の進歩や新たな発明に伴い年々増加している．こうした文章群を一つの統一されたベクトル空間に投影することができれば，特許技術の全体像や内在する構造を可視化し，俯瞰的な解釈が可能になると考える．これらにより，従来になりマクロな視点から特許技術の全体を捉え，新たな知見の発見につなげることができることを確認した．

二つ目の特徴は，共起関係の分析による共起語ネットワークを作成しそれらを 3D グラフおよび 2D グラフによって可視化を行ったことである．2D グラフでは従来どおり共起語間の関係を平面上で表現することができる．2D グラフだけでなく 3D グラフによる描写によって，従来よりもより多くの情報を見ることができまた空間的な表現を行うことができる．これらのことにより，いままでの分析では得られなかった新たな知見をを得られることである．

今後の課題として，実行時間の短縮があげられる．本研究ではスクレイピングによる処理をマルチスレッドを用いることで高速化を図った．しかし，まだまだ処理の時間がかかっており更なる高速化が可能だと考えれらる．そこでマルチプロセスや GPU を用いた並列処理，他にも複数台のコンピュータを用いた分散処理などの手法が有効だと考えられる．さらに分かち書きの処理の高速化もあげられる．本手法で用いた分かち書きのモジュールである Janome はユーザー辞書の登録が容易であるのに対してデータの量が増えると処理時間が長くなるという問題もある．そこで近年開発された Vibrato のような高速な分かち書きシステムを用いることで高速に分かち書きを処理することができ使い勝手がよいシステムになると考える．以上の点を今後改善・検討することで，本手法の実用性と性能を一層向上させることができると考える．処理速度の向上こそが大規模データセットの分析では不可欠な要件であるといえる．

謝辞

本研究を遂行するにあたり，多大なご指導と終始懇切丁寧なご鞭撻を賜った富山県立大学工学部電子・情報工学科情報基盤工学講座の奥原浩之教授，António Oliveira Nzinga René 講師に深甚な謝意を表します．最後になりましたが，多大な協力をしていただいた研究室の同輩諸氏に感謝致します．

2025 年 2 月

小澤 翔太

参考文献

[1] M.J.D. Powell, “Radial basis functions for multivariable interpolation: a review. ”, *Algorithms for approximation*, 143-167，1987.

[2] 奥原 浩之, 尾崎 俊治,“ 適者生存型学習則を適用した競合動径基底関数ネットワーク ”, 電子情報通信学会論文誌, pp. 3191-3199, 1997.

[3] 奥原 浩之, 佐々木 浩二, 尾崎 俊治,“ 環境の変化に適応できる複製・競合動径基底関数ネットワーク ”, 電子情報通信学会論文誌, pp. 941-951, 1999.

[4] 倉本 和正, 鉄賀 博己, 東 寛和, 荒川 雅生, 中山 弘隆, 古川 浩平,“ RBF ネットワークを用いた非線形がけ崩れ発生限界雨量線の設定に関する研究 ”, 土木学会論文集, pp. 117-132, 2001.

[5] 持田 英史, 飯國 洋二,“ RBF ネットワークを用いた到来波方向の適応推定 ”, 電子情報通信学会論文誌, pp. 1205-1214, 2004.

[6] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al., “Mastering the game of go with deep neural networks and tree search ”, *Nature*, 529[7587]，2016.

[7] 高橋 直矢，池谷 裕二，松木 則夫，Hebb 則，脳科学辞典，`https://bsd.neuroinf.jp/wiki/`ヘブ則

[8] Y. Wong，“Clustering Data by Melting”，*Neural Computation*，vol. 5, no. 1, pp. 89-104，1993.

[9] Zak, Michail，“Terminal attractors for addressable memory in neural networks.”，*Physics Letters A*， 133 (1988): 18-22.

[10] Zak, Michail，“Terminal attractors in neural networks.”，*Neural Networks*，2 (1989): 259-274.

[11] Z. Wang，C.D. Massimo，M.T. Tham，and A.J.Morris，“A procedure for determining the topology for multilayer feedforward networks”，*Neural Networks*，vol.7，pp.291-300，1994.