

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	3
第2章 複製・競合を考慮した動径基底関数ネットワーク	4
§ 2.1 競合動径基底関数ネットワーク	4
§ 2.2 ターミナルアトラクタ	7
§ 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 最小化手法の長所・短所	19
第4章 提案手法	21
§ 4.1 RC-RBFN の各パラメータの学習方法	21
§ 4.2 Python の高速化ライブラリ	23
§ 4.3 機械学習への組み込み	26
第5章 実験結果並びに考察	30
§ 5.1 実験の概要	30
§ 5.2 実験結果と考察	31
第6章 おわりに	34
謝辞	35
参考文献	36

図一覧

4.1	テキストデータのフォーマット	22
4.2	システムのフロントページ	22
4.3	解釈の出力例	25
4.4	ユーザー辞書のフォーマット (csv)	25
4.5	提案システム	29
4.6	IPL への活用	29
5.1	ベクトル化の結果	32
5.2	出力されたタイトル	32
5.3	出力された 3D グラフ	33

表一覧

5.1 アンケート内容	31
5.2 アンケート結果	34

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
出力ニューロンの番号	i
出力ニューロン数	N
第 i ニューロンにおける微小領域の番号	k
第 i ニューロンにおける第 k 番目の微小領域	B_{ik}
内的自然増加率	α_{ik}^j
第 j ニューロンと第 h ニューロンの競合効果	γ_{ik}^{jh}
微小領域 B_{ik} におけるシナプス結合荷重の大きさ	w_{ik}^j
微小領域 B_{ik} におけるシナプス結合荷重の大きさ	w_{ik}^j
Positional Encoding における位置エンベディングの次元数	i
Positional Encoding における埋め込みベクトルの次元数	d_{model}
Siamese Network における埋め込み表現の次元	n
Siamese Network におけるラベルの数	k
UMAP における他の点 x_i の近傍に x_j が属する強さ	$v_{j i}$
UMAP における他の点 y_i の近傍に y_j が属する強さ	$w_{j i}$
UMAP における他の点 x_j が属する強さ	$v_{j i}$
UMAP における点 x_i と x_j の距離	r_{ij}
UMAP における点 y_i と y_j の距離	d_{ij}
UMAP における点 x_i に対して, k 近傍の集合	K_i
UMAP における点の疎密に対応するための変数	σ_i
k-menas における n 個の個体	$\vec{x}_i = (x_{i1}, \dots, x_{iD})$
k-menas における n 個の個体の集合	x
k-menas における K 個の重なるの無いクラス	$X_k, k = 1, \dots, K$
k-menas におけるクラスタの中心	\vec{c}_k
k-menas における $X_k^{(t)}$ に属する個体の数	$n_k^{(t)}$
k-menas における $X_k^{(t)}$ の $(K + 1)$ 回目のクラスタの中心	$\vec{c}_k^{(t+1)}$
シルエット分析における各データのサンプル	$x^{(i)}$
シルエット分析における $x^{(i)}$ が属するクラスタ	C_{in}
シルエット分析における $x^{(i)}$ に最も近いクラスタ	C_{near}

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) 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。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。

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。

§ 1.2 本研究の目的

本研究では、粒子群最適化 (Particle Swarm Optimization: PSO) の探索過程における応答曲面の効率的な作成手法を提案し、その実現のために RC-RBFN を活用することを目的とする。PSO は、群れを形成する複数の粒子が相互に情報を共有しながら、探索空間を効果的に探索するアルゴリズムであり工学的設計、機械学習モデルの最適化、金融工学など多岐にわたる分野で活用されている。しかしながら、PSO の性能は応答曲面の構築手法に大きく依存することが知られており、応答曲面が精度不足や過剰な計算負荷を抱える場合、探索の効率や精度に深刻な影響を及ぼす。

応答曲面の作成には通常、関数近似技術が用いられる。中でも、RBFN はその高い近似性能と設計のシンプルさから多くの応用で採用されているが、学習におけるニューロン数の最適化が課題として挙げられている。ニューロン数が過剰である場合には計算負荷が増大し、さらに過学習のリスクが生じる。一方、ニューロン数が不足している場合には応答曲面の精度が劣化し、探索空間全体を適切に表現できなくなる。

これらの課題に対処するため、RC-RBFN が提案されており、動的な基底関数の複製と競合メカニズムにより最適な基底関数の数を自動的に調整することが可能となっている。RC-RBFN は、データの特性に応じて必要な基底関数を生成し、同時に冗長な基底関数を削除することで学習の効率化とモデルの精度向上を両立している。本研究ではこの RC-RBFN の特性を PSO の応答曲面作成に応用することで、以下の具体的な課題解決を目指す。

応答曲面作成の計算効率の向上

RC-RBFN は基底関数の冗長性を排除しつつ、データ特性に応じた柔軟な応答局面の構築を可能とする。この特性を活用することで PSO における応答曲面作成に伴う計算コストを削減する。

高精度な応答局面の構築

RC-RBFN の動的調整機能を活かして、従来の方法では難しかった高次元空間における正確な応答曲面の構築を実現する。これにより PSO の探索精度の向上を図る。

探索性能の向上

高精度かつ効率的な応答曲面に基づく PSO は解空間全体の探索を効果的にを行い、従来の手法では到達が困難であった最適解やグローバル最適解への到達可能性を高める。

さらに本研究では、RC-RBFN を PSO に統合する際の設計指針を提示し、従来の PSO では対応が難しかった問題、特に非線形性が強く複雑な構造を持つ最適化問題への応用可能

性を検証する。最終的に本研究が目指すのは、RC-RBFN を応答曲面作成に応用することで PSO 全体の性能を抜本的に改善し、従来法では達成困難であった計算負荷削減と探索精度向上を両立することである。この取り組みにより RC-RBFN の応用範囲を拡大し、PSO をはじめとする最適化アルゴリズムの進化に寄与することを期待している。

§ 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。

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

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。

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。

$$\frac{dw_j}{dt} = \alpha_j w_j + g_j w_j + f_j \quad (2.1)$$

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

$$\alpha_j = \int \eta(x) \xi_j(x) dx \quad (2.2)$$

で定義される。また、NGF の量 g_j は次の方程式に従う。

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

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．

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

ここでは，競合係数 γ_{jh} を

$$\gamma_{jh} = \frac{\theta_h}{\epsilon_j} = \int \xi_j(x) \xi_h(x) dx \quad (2.5)$$

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

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

であり，正定関数 $V(\mathbf{w})$ として

$$V(\mathbf{w}) = \frac{1}{2} \int \{\eta(x) - s(x)\}^2 dx \quad (2.7)$$

を定義する．ここで， $\mathbf{w} \equiv [w^1, w^2, \dots, w^M]$ であり，

$$s(x) = \sum_{j=1}^M w_j \xi_j(x) \quad (2.8)$$

である．この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 $\xi_j(x)$ と，シナプス結合荷重 w_j との積の総和であるので， $s(x)$ を神経伝達物質放出量と呼ぶこととする．正定関数 $V(\mathbf{w})$ はシナプス後発火頻度 $\eta(x)$ と神経伝達物質放出量 $s(x)$ の差を表す指標である．シナプス後発火頻度 $\eta(x)$ が時間に依存しないと仮定すると，その時間変化が

$$\begin{aligned}\frac{dV(\mathbf{w})}{dt} &= \sum_{j=1}^M \frac{\partial V(\mathbf{w})}{\partial w_j} \frac{dw_j}{dt} \\ &= - \sum_{j=1}^M \left[\int \eta(x) \xi_j(x) dx - \sum_{h=1}^M \int \xi_j(x) \xi_h(x) dx w_h \right] \frac{dw_j}{dt}\end{aligned}$$

$$\begin{aligned}
&= - \sum_{j=1}^M w_j \left(\alpha_j - \sum_{h=1}^M \gamma_{jh} w_h \right)^2 \\
&\leq 0
\end{aligned} \tag{2.9}$$

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。

これらから、シナプス後発火頻度 $\eta(x)$ を入力 x に対する望ましい出力、シナプス前発火頻度 $\xi_j(x)$ 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いうこととする。

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 による関数近似は二乗誤差関数

$$E(\mathbf{w}) = \frac{1}{2} \sum_j^M \{ \eta(\mathbf{x}_j) - s(\mathbf{x}_j) \}^2 \tag{2.10}$$

を減少させることにより実現される。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

$$\begin{aligned}
\frac{dw_j}{dt} &= -\Delta \frac{\partial E(\mathbf{w})}{\partial w_j} \\
&= \Delta \sum_x \{ \eta(x) - s(x) \} \xi_j w_j(x),
\end{aligned} \tag{2.11}$$

$$\begin{aligned}
\frac{dm_j}{dt} &= -\Delta \frac{\partial E(\mathbf{w})}{\partial m_j} \\
&= \Delta \sum_x \{ \eta(x) - s(x) \} w_j \xi_j(x) \frac{(x - m_j)}{\sigma_j^2},
\end{aligned} \tag{2.12}$$

$$\begin{aligned}
\frac{d\sigma_j}{dt} &= -\Delta \frac{\partial E(\mathbf{w})}{\partial \sigma_j} \\
&= \Delta \sum_x \{ \eta(x) - s(x) \} w_j \xi_j(x) \frac{(x - m_j)^2}{\sigma_j^3}
\end{aligned} \tag{2.13}$$

が得られる。 Δ は適当な正の定数である。このような学習則に従う RBFN を CRBFN とする。

§ 2.2 ターミナルアトラクタ

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はリプシッツ条件，すなわち初期値問題に対する解の一意性を保証する条件を満たす。

リプシッツ条件

$f(t, x)$ を，閉区間 $\Omega = (t, x) || t - t_0| \leq a, |x - x_0| \leq b$ 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，

$$|f(t, x) - f(t, y)| \leq L|x - y| \quad (2.14)$$

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 [?]．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．

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

$$\frac{dV(\mathbf{w})}{dt} = -\frac{V(\mathbf{w}^0)^R V(\mathbf{w})^{\frac{1}{r}}}{Rt^*} \quad (2.15)$$

で定義する．ここで， r は任意の奇数であり， $R = \frac{(r-1)}{r}$ である． \mathbf{w}^0 は \mathbf{w} の初期値である．このような定義が可能となったのは，シナプス可塑性方程式に対する 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

$$\frac{dw_j}{dt} = \Delta(\alpha_j - \sum_{h=1}^M \gamma_{jh} w_h) w_j \quad (2.16)$$

とすることができる．このとき，Lyapunov 関数の時間変化は

$$\frac{dV(\mathbf{w})}{dt} = -\Delta \sum_{j=1}^M w_j (\alpha_j - \sum_{h=1}^M \gamma_{jh} w_h)^2 \quad (2.17)$$

となる．ここで， Δ は式 (2.15) と式 (2.17) から求めることができ，

$$\Delta = \frac{1}{\sum_{j=1}^M w_j (\alpha_j - \sum_{h=1}^M \gamma_{jh} w_h)^2} \times \frac{V(\mathbf{w}_0)^R V(\mathbf{w})^{\frac{1}{r}}}{Rt^*} \quad (2.18)$$

と決定される．以上のことから，望ましい時刻 t^* で平衡解へ収束する修正されたシナプス可塑性方程式を

$$\frac{dw_j}{dt} = \frac{(\alpha_j - \sum_{h=1}^M \gamma_{jh} w_h) w_j}{\sum_{j=1}^M w_j (\alpha_j - \sum_{h=1}^M \gamma_{jh} w_h)^2} \times \frac{V(\mathbf{w}_0)^R V(\mathbf{w})^{\frac{1}{r}}}{R t^*} \quad (2.19)$$

で定義することができる．

§ 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．これら従来の研究には，しきい値などを考え動径基底関数の削除と追加を行うものもある．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，環境の変化に適應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．ここで， $\mathbf{m} \equiv [m_1, m_2, \dots, m_M]$, $\Sigma \equiv [\sigma_1, \sigma_2, \dots, \sigma_M]$ である．シナプス結合荷重 w^j を

$$y_j^2 = w_j \quad (2.20)$$

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

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

となる．(2.21) は積分条件

$$\frac{\partial}{\partial y_h} \frac{dy_j}{dt} = \frac{\partial}{\partial y_j} \frac{dy_h}{dt} \quad (2.22)$$

を満たすため，ポテンシャル

$$U'(\mathbf{y}, \phi) = - \sum_{j=1}^M \int_a^{y_j} \left(\frac{\alpha_j}{2} - \sum_{h=1}^M \frac{\gamma_{jh}}{2} y_h^2 \right) y_j' dy_j' \quad (2.23)$$

を考えることができる．ただし $\mathbf{y} \equiv [y_1, y_2, \dots, y_M]$ 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 $\{\mathbf{m}_j, \Sigma_j\}$ である．そして ϕ で集合 $\{\phi_1, \phi_2, \dots, \phi_M\}$ 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，

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

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

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

と書き直すことができる．この結果，

$$E(\mathbf{w}, \phi) = 4U(\mathbf{w}, \phi) + \frac{1}{2} \sum_i^N \eta^2(x_i) \quad (2.26)$$

であることが示される．ただし， $\mathbf{x}_i \equiv [x_1, x_2, \dots, x_N]$ 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．

今，(2.20) 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

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

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

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

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

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

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

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

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

$$\eta(x) = 3N(-1.5, 1) + 2N(1, 0.5) \quad (2.31)$$

で与えることとする． $N(m, \Sigma)$ は平均 m ，分散 Σ の正規密度関数を表す．この教師信号を動径基底関数一つ（シナプス結合荷重 $w = 1$ ，パラメータ $\Sigma = 0.2$ ）だけ用いて近似することを考える．この場合，近似しようとしている非線形関数 $\eta(x)$ 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ため，累積 2 乗誤差関数 $E(\mathbf{w})$ を 0 にすること自体が不可能である．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

$$p_{\beta'}(m|w, \Sigma) = \frac{p_{\beta'}(w)}{\int_m p_{\beta'}(w) dm} \quad (2.32)$$

は導出することができる．

よって，シナプス結合荷重 $w = 1$ ，パラメータ $\Sigma = 0.2$ 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（パラメータ $w = 1$ ， $\Sigma = 0.2$ ）をもつ動径基底関数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 (2.13) 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 $\eta(x)$ 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 $\Sigma_i \rightarrow 0$ 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 0 とできる場合がある．ここで，0 は零行列を表す．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，前章で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 Σ_j ）が新たに追加される動径基底関数を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} に動径基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

$$p_{\beta'}(\mathbf{x}_i | \mathbf{m}_{j[i]}, \phi'_j, \phi''_j) = Z_{\beta'}^{-1}(\mathbf{m}_j, \phi'_j, \phi''_j) \exp\{-\beta' E(\mathbf{x}_i, \mathbf{m}_{j[i]}, \phi'_j, \phi''_j)\} \quad (2.33)$$

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

$$Z_{\beta'}(\mathbf{m}_j) = \sum_{i=1}^N \exp\{-\beta' E(\mathbf{x}_i, \mathbf{m}_{j[i]})\} \quad (2.34)$$

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

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

が一定となるという二つの制約のもとで，エントロピー

$$\langle E(\mathbf{m}_j) \rangle_{\beta'} = \sum_{i=1}^N p_{\beta'}(\mathbf{x}_i | \mathbf{m}_{j[i]}) E(\mathbf{x}_i, \mathbf{m}_{j[i]}) \quad (2.36)$$

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

$$F_{\beta'}(\mathbf{m}_j) = -\frac{1}{\beta'} \log Z_{\beta'}(\mathbf{m}_j) \quad (2.37)$$

で定義すれば，

$$S_{\beta'}(\mathbf{m}_j) = -F_{\beta'}(\mathbf{m}_j) + \beta' \langle E(\mathbf{m}_j) \rangle_{\beta'} \quad (2.38)$$

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， $m_{j[i]} = \mathbf{x}_i (i = 1, 2, \dots, N)$ 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．

4. 2 複製する位置の決定法

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

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

$$\begin{aligned}
&= \sum_{i=1}^N p_{\beta'}(\mathbf{x}_i | \mathbf{m}_{j[i]}) \Delta m_{j[i]} \\
&= \langle \Delta m_j \rangle_{\beta'}
\end{aligned} \tag{2.39}$$

で与えることとする。ここで、

$$\Delta m_{j[i]} = -\epsilon \frac{\partial E(\mathbf{x}_i, \mathbf{m}_{j[i]})}{\partial m_{j[i]}} \tag{2.40}$$

である。

特に $\beta' = 0$ であり、初期の状態が $m_{j[i]} = m_j (\forall i)$ である場合は

$$\Delta_0 m_j = \Delta m_j \tag{2.41}$$

であることが示される。この場合は、RC-RBFN のパラメータ m_j の更新則が従来の RBFN のパラメータ 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 $\Sigma_j \rightarrow 0$ にすると、 $\Delta_0 m_j = 0$ とするパラメータ $m_{j[i]}$ は

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

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 $\beta' \rightarrow \infty$ にすると、 $\Delta_\infty m_j^k = 0$ とするパラメータ $m_{j[i]}$ は

$$\sum_{i=1}^N \exp\{-\beta' E(\mathbf{x}_i, \mathbf{m}_{j[i]})\} \xi(\mathbf{x}_i, \mathbf{m}_{j[i]})(\mathbf{x}_i - \mathbf{m}_{j[i]}) \{ \eta(\mathbf{x}_i) - s(\mathbf{x}_i, \mathbf{m}_{j[i]}) \} = 0 \tag{2.43}$$

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 m_j の更新則 $\Delta_{\beta'} m_j$ では、 $\Delta_0 m_j$ で従来の RBFN のパラメータ m_j の更新則を実現し、更に、 $\Delta_\infty m_j^k$ とすることで、すべての入力ベクトル \mathbf{x}_i の第 k 要素 $x_i^k (\forall i)$ を含む任意の値を安定な収束点とすることができる。ここで、提案手法とゆう度解析との関係について示す。まず、次のような自由エネルギー

$$F_{\beta'} = -\frac{1}{\beta'} \log Z_{\beta'} \tag{2.44}$$

を考える。ただし、分配関数は

$$Z_{\beta'} = \sum_{i=1}^N \sum_{j=1}^M \exp\{-\beta' E(\mathbf{x}_i, \mathbf{m}_{j[i]})\} \tag{2.45}$$

で与えられる定数である。このとき、式 (2.47) は

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

$$= -\frac{1}{\beta'} \log \sum_{i=1}^N p_{\beta'}(\mathbf{x}_i, m_{j[i]}) \quad (2.46)$$

に変形することができる．ここで，確率密度関数 $p_{\beta'}(\mathbf{x}_i|\mathbf{m}_{j[i]})$ は

$$p_{\beta'}(\mathbf{x}_i|\mathbf{m}_{j[i]}) = Z_{\beta'}^{-1} \exp\{-\beta' E(\mathbf{x}_i, \mathbf{m}_{j[i]})\} \quad (2.47)$$

である．つまり，自由エネルギー $F_{\beta'}(\mathbf{m}_j)$ のパラメータ m_j^k に関する最小化は，対数ゆう度関数に関する最大化に等価であることを示すことができる．

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．

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、機械学習は学术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。

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

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 $h(x) = a^T x$ (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

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

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

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

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 $\pi_k \geq 0 (k = 1, \dots, K)$ かつ $\sum_{k=1}^K \pi_k = 1$ 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

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

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 $a \in A$ (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

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

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ことで，

$$\pi^*(s) = \operatorname{argmax}_{a \in A} \sum_{s' \in S} P(s'|s, a) U^{\pi^*}(s') \quad (3.6)$$

を満たすはずである．このとき， U^π の定義から

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

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プローチの一つである．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 $\mathbf{x} = (x_1, x_2, \dots, x_n)$ 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 $\text{grad}f(\mathbf{x}^{(k)})$ ）の方向に沿って、 \mathbf{x} を次のように更新する。

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

ここで、 $\text{grad}f(\mathbf{x})$ 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い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

$$\mathbf{Ax} = \mathbf{b} \quad (3.9)$$

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

$$\mathbf{u}^T \mathbf{Av} = 0 \quad (3.10)$$

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

$$\begin{aligned} \langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{A}} &= \langle \mathbf{A}^T \mathbf{u}, \mathbf{v} \rangle \\ &= \langle \mathbf{Au}, \mathbf{v} \rangle \\ &= \langle \mathbf{u}, \mathbf{Av} \rangle \\ &= \mathbf{u}^T \mathbf{Av} \end{aligned} \quad (3.11)$$

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，

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

と書ける．ただし係数は，

$$\begin{aligned} \mathbf{A}x^* &= \sum_{i=1}^n \alpha_i \mathbf{A}p_i \\ &= \mathbf{b} \end{aligned} \quad (3.13)$$

$$\begin{aligned} \mathbf{p}_k^T \mathbf{A}x^* &= \sum_{i=1}^n \alpha_i \mathbf{p}_k^T \mathbf{A}p_i \\ &= \mathbf{p}_k^T \mathbf{b} \end{aligned} \quad (3.14)$$

$$\begin{aligned} \alpha_k &= \frac{\mathbf{p}_k^T \mathbf{b}}{\mathbf{p}_k^T \mathbf{A}p_k} \\ &= \frac{\langle \mathbf{p}_k, \mathbf{b} \rangle}{\langle \mathbf{p}_k, \mathbf{p}_k \rangle_{\mathbf{A}}} \end{aligned} \quad (3.15)$$

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

$$c_1x_1 + c_2x_2 \quad (3.16)$$

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 最小化手法の長所・短所

前節で述べた通り、目的関数の最適化を行う手法は多岐にわたって研究されている。もちろん、それぞれの手法には長所と短所があり、本節では前節で紹介した手法の長所・短所を述べたのち、一部のパラメータに関しては反復法によって最適化させていくのではなく、一律で与える方法が提案されていることを付記する。

最急降下法

最急降下法は、アルゴリズムがシンプルで実装が容易である点が最大のメリットである。この手法は、凸関数に対して適切なステップサイズを選べば収束が保証されるため、基礎的な最適化問題の解法として広く利用されている。また、関数の連続性があれば適用可能で、さまざまな最適化問題に応用できる点も利点だと考えられる。しかし一方で、条件数が多い問題では、最急降下法はジグザグな経路をたどる傾向があり、収束に時間がかかるという欠点がある。したがって、収束速度が遅くなることが実用上の課題となることが多い。

共役勾配法

共役勾配法は、最急降下法と比べて高速に収束することが主なメリットである。特に二次関数の場合には、反復回数に変数の次元数以内で厳密解に到達する特性がある。そのため、大規模な線形システムや二次計画問題の解法として非常に効率的である。また、ヘッセ行列を直接計算せずに適用できるため、メモリ使用量が少なく済むという実用的な利点もある。ただし、欠点としては丸め誤差や計算誤差の影響を受けやすく、特に大規模問題で反復が進むと精度が低下する可能性があるという点である。また、関数が必ずしも二次形式でない場合や、条件数が極端に大きい場合には収束が保証されない場合もある。

線形計画法

線形計画法の主なメリットは、問題が線形である場合に最適解を効率よく得られることである。特にシンプレックス法や内点法などの手法は、実際の問題において非常に高い精度で解を見つけることができる。また、実行可能解が存在する場合に収束が保証されている点も重要な利点である。一方でデメリットとしては、問題が非線形の場合には直接適用できないことが挙げられる。また、大規模な問題では計算負荷が高くなり、メモリや計算資源が制約となることがある。特にシンプレックス法では、最悪の場合に指数的な計算時間がかかる可能性があり、大規模な問題では内点法の方が適している場合がある。

一撃アルゴリズムの提案

これらの手法のほかに、目的の値を一回の計算によって導出する一撃アルゴリズムも提案されている。例えば、中山らは、RBFN に用いられるガウス関数の半径 r を一律に与える式を以下のように定めている。ただし、中山らの半径式は次元が増加した場合には近似最適解を得ることが困難になることが、北山らによって検証されている。

$$r = \frac{d_{max}}{\sqrt[n]{nm}} \quad (3.17)$$

n は変数の数であり, d_{max} はサンプル点間の最大距離である. また, m はサンプル点数を表している. ここで, 水準数を K とし, サンプル点を等間隔 Δd に配置した全数組合せ実験計画を考える. このとき, サンプル点間の最大距離は

$$d_{max} = \sqrt{n}(K-1)\Delta d \quad (3.18)$$

となる. また, 総サンプル点数 m と水準数 K の関係は

$$m = K^n \quad (3.19)$$

で与えられるため, d_{max} に代入すると

$$\begin{aligned} r &= \frac{d_{max}}{\sqrt[n]{nm}} \\ &= \frac{\sqrt{n}(K-1)\Delta d}{\sqrt[n]{nK^n}} \\ &= \frac{\sqrt{n}}{\sqrt[n]{n}} \left(1 - \frac{1}{K}\right) \Delta d \end{aligned} \quad (3.20)$$

が得られる. ここで, r と Δd の比を考えると, 上式は次のように変形できる.

$$\frac{r}{\Delta d} = n^{\frac{n-2}{2n}} \left(1 - \frac{1}{K}\right) \quad (3.21)$$

この式において $K \rightarrow \infty$ とする. これは関数空間内に無限のサンプル点を等間隔に配置するということを意味しており, 理想的なサンプル点配置である. このとき, $n=1$, $n=2$ のときはある値 (すなわち 1) に収束するが, $n \leq 3$ となったとき, n の値に応じて個別の値に収束することがわかる. すなわち, 中山らによって提案された半径式を用いた場合に次元が増加するにつれて近似最適解を得ることが困難となった理由の一つとして, 次元の増加とともに r と Δd の比が個別の値に収束するということが指摘されている.

そこで, 北山らは近似最適解を得るための半径式の十分条件として以下の 4 つを挙げている.

1. 設計変数の数, すなわち次元を考慮できること.
2. サンプル点の数を考慮できること.
3. サンプル点間の距離を考慮できること.
4. サンプル点数の増加に伴い, 次元の大きさに依存せず, ある同じ値に収束すること.

これらの項目を満足する式として, 以下の半径式が提案されている.

$$r = \frac{d_{max}}{\sqrt{n} \sqrt[n]{m}} \quad (3.22)$$

提案手法

§ 4.1 RC-RBFN の各パラメータの学習方法

本研究の提案手法は、大別すると以下のような工程からなる。

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に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。

ここで、スクレピングを行う際に時間がかかってしまう問題を解決するために、python のモジュール `threads` を用いてマルチスレッドによるスクレピングを行う。コンピュータの性能によって並列にする数を増やすとかかる時間が長くなる。そのため、並列にする数を6つにして実行を行った。

`threads`

スレッドベースの並列タスク実行を助けるモジュールである。`threads` を利用することで、一つの python プログラムの中で複数の処理を同時に実行するマルチスレッド処理を実現できる。`threads` モジュールには `Threads` クラスが定義されており、この `Threads` をスーパークラスとして新しいスレッドを定義する。`run ()` メソッドの中にそのスレッドが実行する処理を書き、`start ()` メソッドを呼び出すことでスレッドが起動し、並列で処理が進む。

`join ()` メソッドを使用すれば、スレッドの終了を待つ制御することもできる。`Lock` や `Semaphore`, `Event` などの動機機構も利用でき、スレッド間でデータを安全に共有

特許本文のテキスト	特許番号
<p>\\n\\n\\n 本発明は、外灯機器が切れたときの不点原因箇所の探査に使用する不点探査装置及び、</p> <p>\\n\\n\\n 本発明は、押出成形体の製造方法に関し、さらに詳しくは高剛性であり、屈形金型に、</p> <p>\\n\\n\\n 本発明は、複数のビットにより構成されるビット列を暗号化する暗号化装置に関する。</p> <p>\\n\\n\\n 本発明は、既設の鉄塔を支える基礎を改修する工法、及び改修する構造、及びそれに</p> <p>\\n\\n\\n 本発明の実施形態は、送電用鉄塔などの送電系統において使用される塔上開閉装置の、</p>	<p>patent/IP596564682/ja</p> <p>patent/IP2012126139A/ja</p> <p>patent/IP2013167729A/ja</p> <p>patent/IP500273581/ja</p> <p>patent/IP2013198381A/ja</p>
⋮	リンク
<p>\\n\\n\\n 本発明は、電力需要者や電力供給者等が電力の消費や発電によって創出された二酸化...</p> <p>\\n\\n\\n 特許法第30条第2項適用 令和4年9月13日に、富山県立富山工業高等学校（富山県富山...</p> <p>\\n\\n\\n 本発明は、電力需要者や電力供給者等が電力の消費や発電によって創出された二酸化...</p> <p>\\n\\n\\n 本発明は、電力需要者や電力供給者等が電力の消費や発電によって創出された二酸化...</p> <p>\\n\\n\\n 本発明は、基準価格算出装置及び基準価格算出方法に関する。\\n\\n\\n\\n\\n...</p>	<p>patent/IP724665981/ja</p> <p>patent/IP732664181/ja</p> <p>patent/IP733681681/ja</p> <p>patent/IP736949481/ja</p> <p>patent/IP741034981/ja</p>

図 4.1: テキストデータのフォーマット

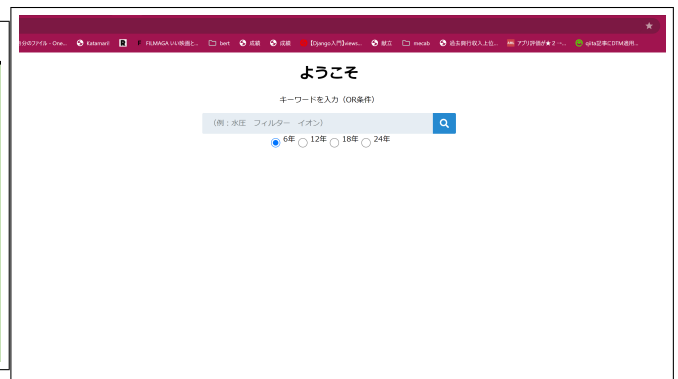


図 4.2: システムのフロントページ

することも可能である。ファイル IO やネットワークアクセス時の待ち時間を隠蔽したり、応答性を向上させたり、並列処理による速度向上が図れるなど、threads は即効性の高い並列プログラミングを実現できる。

提案手法では、対象期間を1年ごとに分割し、それぞれの期間を個別のスレッドに割り当てて並列処理を行う。具体的には、6つのスレッドを作成し、各スレッドが1年間のデータをスクレイピングする。つまりスレッド1は1年分、スレッド2は次の1年分となる。そして、各スレッド内で1年ごとにデータ収集を行う。こうすることで期間ごとの並列処理が可能となり、スクレイピングの速度や効率を向上させることができる。最後に、すべてのスレッドが実行完了後、収集したデータを統合することで、対象期間全体のデータセットを取得する。

データの分類

収集したテキストデータをもとに Sentence-BERT を用いてそれぞれをベクトルに表現する。この時、Sentence-BERT によって出力されるベクトルを pickle を用いて保存しておく。それらのベクトルを UMAP を用いて15次元および2次元のベクトルに圧縮する。この際、UMAP に設定するパラメータについて説明する [?].

パラメータの設定

n_neighbors

n_neighbors パラメータは、各データポイントの埋め込みにおいて考量する近隣点の数を指定する。この値が大きいほどデータ全体の構造が強調され、小さいほど局所的な構造が強調される。小さな n_neighbors 値 (5-20) は、小規模なクラスターや微細な構造の検出に適していおる。一方大きな n_neighbors 値 (50-200) は、データ全体の構造や大規模なクラスターを強調するときに用いられる。

min_dist

min_dist パラメータは、UMAP によって生成される低次元埋め込み空間内のデータ点間の最小距離を制御する。小さな min_dist 値 (0.0-0.3) はデータが密集し

たクラスタリングを得る際に適用する．中程度の `min_dist` 値 (0.3-0.7) は、クラスタ間のバランスが取れた埋め込みを得る際に適用する．大きな `min_dist` 値 (0.7以上) は、クラスタが広がり、隣接するクラスターとの距離を最大化する場合に適用する．`min_dist` パラメータは低次元空間内のデータ点は位置をコントロールし、クラスタの密度やスペースを調整する役割がある．

n_components

`n_components` パラメータは、UMAP によって生成される埋め込み次元の次元数を指定する．`n_components=2` や `n_components=3` の低次元埋め込みは、データの分布を直観的に把握しやすいため、結果の可視化を目的とする場合に選択される．一方で `n_components` が 3 以上の値を設定した場合、特定のアルゴリズムでの利用や、次元削減後のデータをほかの分析タスクに利用されることを目的とする．`n_components` パラメータの値は解析目的やデータ利用法に応じて適切に定める必要がある．

metric

`metric` パラメータは、データ間の類似度や距離を算出するための手法を指定することができる．これにより、データ空間の幾何学的性質が定義される．数値データの場合、ユークリッド距離やマンハッタン距離などの標準的な手法を指定するのが一般的である．一方テキストデータの場合には、コサイン距離やハミング距離などのテキスト向けの手法が利用される．データの型や構造に応じた適切な手法を `metric` パラメータに設定することで、UMAP のパフォーマンスが最大化される．

15 次元のベクトルはクラスタリングを行う際と、クラスターの解釈を行う際に用いる．2 次元のベクトルはクラスタリングを行ったデータをプロットする際に用いる．プロットする際に 2 次元ベクトルを用いる理由は、3 次元ベクトルやそれ以上の次元数のベクトルと比較して、2 次元のベクトル空間上にプロットされた各データ点間の距離感や密集具合を人間の知覚として把握しやすいためである．また、データ間の類似性を可視化する上でも、2 次元空間上では各クラスター内でのデータ点のまとまり方を把握しやすく、データセット全体の構造を俯瞰しやすい．

§ 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。

システムのグラフ表示およびクラスターの選択

実際に作成された散布図を画像データとして保存する。散布図の生成には `pyvis` を用いる。その際画像の中のクラスターはそれぞれ別の色でプロットして、それらのクラスターの番号を補足情報として画像に追加する。またそれぞれの点がでかすぎて画像が見づらくなることを加味し、点のサイズをあらかじめ設定しておく。さらに、クラスターの数が多くなると色の違いによるクラスターの判別が難しくなるため、それぞれのクラスターの点の形を変えることで色だけでなく形でもクラスターを区別できるようにしている。そのあと画像データを html 上に表示する。また、各クラスターの内容とそれらのクラスター番号をそれぞれ箇条書きで表示する。画像のクラスターとその内容を照合することで、ユーザーが任意のクラスターを指定できるようにする実際の解釈の出力例を図 4.3 に示す。

データの前処理

共起語ネットワークを作成する際に、文章を分かち書きする必要がある。この時、特許には多数の専門用語や複合語が含まれるため、それらを抽出したうえで分かち書きを行う。`termextract` では専門用語の抽出を行うことはできるが、それらを用いて分かち書きを行うことはできない。そこで、今回用いた分かち書きのモジュールである `Janome` にユーザー辞書として専門用語や複合語を登録する。

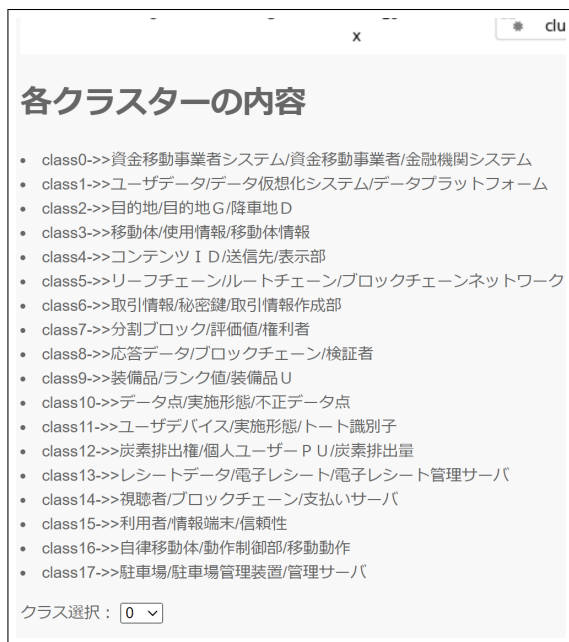


図 4.3: 解釈の出力例

ユーザー辞書のフォーマット

Janome は独自の単語や品詞情報を追加することができる。この機能を用いることで、特定の文脈や専門用語に適した分かち書きを行うことができる。csv 形式でユーザー辞書を与える必要がある。

< ユーザー辞書の形式 >

ユーザー辞書はカンマ区切り CSV ファイルで、「表層形, 左文脈 ID, 右文脈 ID, コスト, 品詞, 品詞細分類 1, 品詞細分類 2, 品詞細分類 3, 活用型, 活用形, 原型, 音読み, 発音」という形式で与える必要がある。今回は表層系に抽出された専門用語や複合語を入力し, 左文脈 ID および右文脈 ID は-1 に指定する。コストはすべて 1000 とし, 品詞には名詞, 品詞細分類 1 には固有名詞を設定する。品詞細分類 2, 品詞細分類 3, 活用形, 活用型, 音読み, 発音は未設定とし原型においては表層系と同じ文字列を入力する。実際に作成された CSV ファイルを図 4.4 に示す。

この時, 作成される辞書の量が多くなると Janome の分かち書きが正しく動作しないことがある。それらを解決するために辞書の量をあらかじめ削減する。削減する方法は事前に求めた専門用語や複合語の重要度が低いものを優先的に削除していく。

分かち書きを行ったのち, 共起語ネットワークを作成する。クラスターごとに共起語を分析する。共起語を分析する際, 一般的な用語が多く含まれてしまうことがあるため, 重要度が高い単語が優先的に含まれるようにする。事前に計算した単語の重要度を用いて, 共起語の中に重要ではない単語が含まれているものを除外する。

共起語ネットワーク

本研究では, 単語間の共起関係を分析するために Simpson 係数を用いる。Simpson 係数は Jaccard 係数や Dice 係数と比較して, 差集合の要素数による影響をより小さく抑えるこ

A	B	C	D	E	F	G	H	I	J	K	L	M
系統連系	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	系統連系	*	*
水素電池	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	水素電池	*	*
交流電力	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	交流電力	*	*
水素ガス	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	水素ガス	*	*
発電電力	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	発電電力	*	*
変圧器	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	変圧器	*	*
負荷運転	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	負荷運転	*	*
格出力	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	格出力	*	*
負荷率	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	負荷率	*	*
太陽光パネル	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	太陽光パネル	*	*
太陽光発電カーブ	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	太陽光発電	*	*
許容負荷	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	許容負荷	*	*
PCC出力	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	PCC出力	*	*
許容範囲	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	許容範囲	*	*
変圧器バンク	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	変圧器バンク	*	*
電力系統	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	電力系統	*	*
連系ステーション	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	連系ステーション	*	*
連系ユニット	-1	-1	1000 名詞	固有名詞	*	*	*	*	*	連系ユニット	*	*

図 4.4: ユーザー辞書のフォーマット (csv)

とができる。ただし、Simpson 係数は一方の集合が他方の真部分集合である場合に 1 となる。そこで、今回、Simpson 係数が 1 となる場合には、それらがもともと一つの単語であったとみなして分析対象から除外している。また、片方の集合の要素数が極端に少ないと係数値が大きくなる傾向があることから、お互いの集合数に 5000 以上の開きがある場合も分析対象から除いている。さらに、共起関係を求める際に、一般的な用語が多く出現する傾向があるため上記で求めた、単語の重要度を用いて重要度が高いものを優先に分析を行う。上記の分析を用いて作成された共起語ネットワークを 3D グラフおよび、2D グラフによって可視化を行う。

Three.js を用いた 3D グラフ作成

3D グラフの作成には Three.js のモジュールである 3D Force-Directed Graph を用いる。3D Force-Directed Graph グラフでは Json ファイルの形式でデータを与えることができる。先ほど作成された共起語ネットワークを、ノードの名前と、矢印の元のノードおよび矢印の先のノードを "nodes" および "links" として与えることでグラフの描画に必要な情報を受け渡す。受け渡された情報をもとにグラフを作成する。ここで、3D Force-Directed Graph の初期設定ではグラフのノードは球体になっており、テキストの表示を行うには適しているとは言えない。そこでノードそのものをテキストにする。さらに、3D グラフにおけるノード間の線には共起元の単語から共起先の単語への矢印を描画しているが、ノード間の距離が広いと矢印による識別が難しくなる。そこで矢印の方向方向に向かって流動的なアニメーションを追加している。他にも、読み込まれたグラフのカメラ操作はグラフの中心を軸に 360 度回転することができるが、ノードの場所によってはあまり詳細に表示できない場合がある。そのためノードをクリックすることでそのノードを中心とする回転に変更でき、そのノードを中心としてノードの付近を見渡すことができる。

pyvis を用いた 2D グラフ作成

2D グラフの作成には pyvis を用いる。pyvis では共起元のワードと共起先のワードをノードとして追加し、それらの関係について定義することでグラフを描画することができる。また、初期設定では、ノード間の線は矢印にはなっていない。そこでグラフを描画する際の設定 directed という項目の設定を変更する必要がある。また、2D グラフでは、共起関係の強さによって矢印の太さを変更している。共起関係が強いものは太く、弱いものは細くなるようにしている。このことで、一目で単語同士の共起関係の強さを把握することができる。さらに、ノードをクリックすることで、そのノードに向かっているノードとそのノードから向いているノードを強調表示することができる。

§ 4.3 機械学習への組み込み

4 章で示した各手法を統合した課題解決のための提案手法全体の流れの説明を行う。また、これまでに説明した技術のそれぞれがどの部分に組み込まれているかについて整理しながら、flask を用いて作成した提案手法を組み込んだシステム全体の流れを説明する。提案システム全体のフロー図を図 4.5 に示す。

flask アプリケーションの作成

ユーザーからの入力や、結果の出力を行うためのアプリケーションを flask を用いて作成した。flask とは python で Web アプリケーションを作成するための軽量なフレームワークであり、flask の最大の特徴は軽量性とシンプルさである。設定やコードが少なく、簡単にアプリケーションの開発を行うことができる。加えて、flask は拡張性に優れている。必要に応じて様々なライブラリを使用することで、機能を拡張することができる。また、URL の経路設定や経路処理、テンプレート、データベースなど重要な機能を柔軟に組み合わせることができるため、目的に合ったアプリケーションを作成することができる。

提案手法全体の流れ

Step 1: キーワードの入力・取得年数の選択

フロントページにてユーザーからのキーワードの入力を取得する。一つの単語だけでなく複数の単語でも検索できるようにすることで広い範囲の検索を可能にする。複数の単語を入力したい場合は単語と単語の間にスペースを空けて入力することで、複数の単語の入力を行うことができる。またここで取得したい年数を指定することができる。初期設定は 6 年間となっており、2017 年から 2023 年までのデータを取得することができる。それ以外にも 12 年、18 年、24 年と選択することができる。

このようにユーザーが選択できるようにすることで、データの取得数が不足、または、過剰となることを回避することができる。このことにより、結果の質的向上を目指している。また取得するデーターが多いと、スクレイピングに時間がかかってしまうため、それらの回避も行うことができる。ユーザーの用途や目的によって年数を選択することができる。

ユーザーが入力した情報をもとにリアルタイムでスクレイピングを行う。ただし、検索を行ってからスクレイピングを行っている間、画面がそのままであると待機状態が不明瞭となるため、スクレイピングを行っている際は、処理時間専用の画面を用意している。入力されたキーワードをもとにスクレイピングを行ったデータは次のフローへ送信される。

Step 2: 特許の俯瞰とクラスターの選択

Step1 で取得したデーターをもとにしたクラスタリングの結果をプロットする。これらのプロットの結果を見ることで特許全体を俯瞰することができる。また、それらのなかでの分野のまとまりについても見るることができる。各クラスターについてはそれぞれ違う色の点で描画しており、それらのまとまりをより視覚的にわかりやすいようにしている。ここで、色だけの差別化では、色による違いが判らない場合があるため、それぞれの点の形を変更することで、それぞれのクラスターの区別を行っている。

さらに、クラスターの解釈を図の中に組み込んでしまうと図と文字が重なって見づらくなるため、クラスターの解釈については図の外に記述してある。ユーザーは図の中クラスターの番号と解釈におけるクラスターの番号を照らし合わせることでそれぞ

れのクラスターの解釈を確認することができる。クラスターの解釈をもとにユーザーは自身の興味のあるクラスターを選択することができる。クラスターの選択は自身の選択したクラスターの番号を選択し、送信ボタンを押すことで、システムにその情報が送信される。

以上のことにより、ユーザーは自分が知りたい技術分野や、特許の散らばり具合から、密になっている部分や疎になっている部分に対して分析の対象を選択することができる。例えば特定分野が集中的にクラスターになっている部分や、逆に分散している部分からそれらのクラスターを分析対象とすることができる。さらに、散布図での出力により、特許全体やそれぞれのクラスターからなる技術のトレンドなどを一目で把握することができると思う。

Step 3: 選択されたクラスターにおける共起語ネットワークの作成

Step2 にて選択されたクラスターにおける共起語ネットワークを作成する。Simpson 係数を用いて共起関係の分析を行う。そこで計算された係数値をもとに 3D グラフおよび 2D グラフによる可視化を行う。ここで、グラフを表示するときのグラフの大きさを設定できるようにしている。ユーザーによって分析の目的やニーズによって異なるケースが考えられる。広い範囲を分析することで広域的な分析を行いたい人、また狭い範囲で狭域的な分析を行いたい人など、様々なケースが考えられるので、出力されるグラフの大きさをユーザーが指定することができるようにしている。このことにより、ユーザーは自分の分析目的に合わせて適切なグラフの大きさを設定することが可能であり、広い範囲を見渡すことで全体像を把握したり、狭い範囲で詳細な関係性を分析したりすることが可能である。

また、選択されたクラスターに含まれている特許の特許番号を一覧に表示する。特許をスクレピングする際に取得された特許番号と各ベクトルが紐づけられているためクラスターの中に含まれるベクトル情報からクラスターに含まれる特許番号を取得する。さらにその特許番号をクリックすることで、元の特許における GooglePatents のページが表示されるようになっている。このことで、実際の特許にもアクセスすることが可能となる。

以上の操作の結果もとめられた共起元の単語と共起先の単語およびそれらの Simpson 係数をデータフレームに保存し pickle データの形式で保存する。ここで、pickle を用いるのは、csv 形式で保存してしまうと、

Step 4: 共起語ネットワークの可視化

Step3 で作成された共起元の単語と共起先の単語および Simpson 係数の値を用いて言共起語ネットワークを 2D および 3D グラフによって可視化する。グラフの作成には Three.js のモジュールである 3D Force-Directed Graph を用いる。

Json ファイルで共起元の単語と共起先の単語を json ファイル形式で与える。与えたファイルをもとに 3D グラフを描画する。描画されたグラフはマウスをドラッグすることで回転でき、異なる視点からの観察が可能である。またホイールを回転させることでグラフの拡大、縮小を行うことができる。さらに、単語をクリックすることで

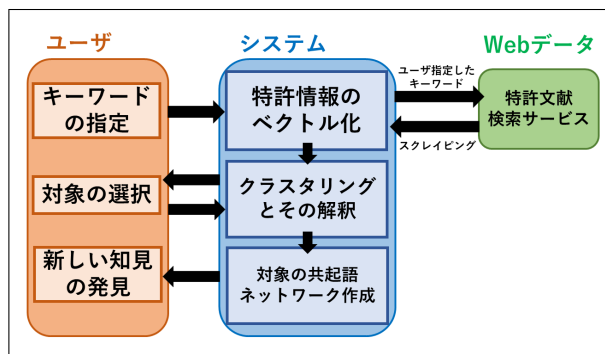


図 4.5: 提案システム

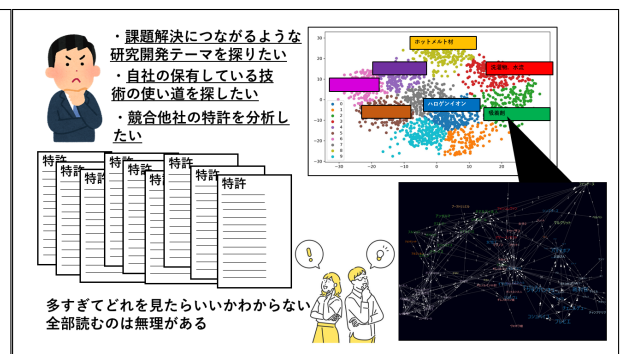


図 4.6: IPL への活用

その単語を中心とした回転に変更することができる。加えてグラフの矢印の向きは線の途中に描画してあるが見つらい部分があるため、流動的なアニメーションを追加することでわかりやすくしてある。上記で説明したシステムの実際の IPL への活用例を図 4.6 に示す。

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い。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5.1に示す。表5.1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

表 5.1: アンケート内容

システムの操作性はわかりやすいか	システムの機能は理解しやすいか
レイアウトは親切か	デザインは見やすいか
ストレスなく利用することができたか	クラスターの内容を理解することができたか
共起語ネットワークによる出力は適していたか	3Dグラフによる提示は適切であるか
効率的な特許探索を行えると思ったか	新しい知見を発見できそうか

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とが把握的た。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ト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

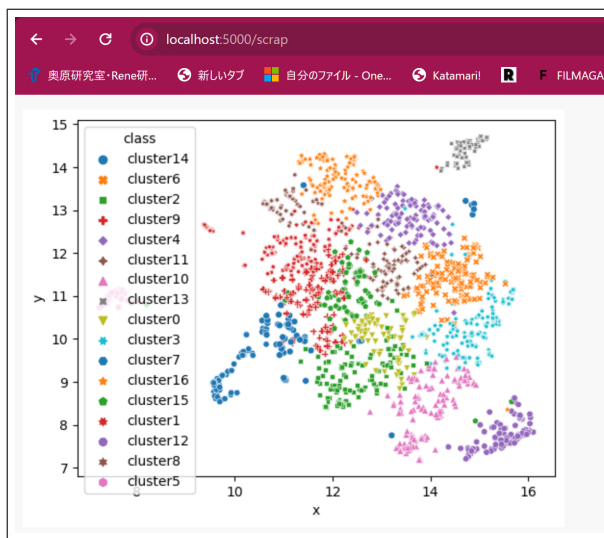


図 5.1: ベクトル化の結果



図 5.2: 出力されたタイトル

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長いこと、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わから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思う。

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

表 5.2: アンケート結果

	解答者A	解答者B	解答者C	解答者D	解答者E
システムの操作性はわかりやすいか	4	4	5	4	4
システムの機能は理解しやすいか	3	5	4	5	4
レイアウトは適切か	4	4	5	4	5
デザインは見やすいか	5	4	4	5	5
ストレスなく利用することができたか	2	2	3	2	2
クラスターの提示は適切であるか	4	4	2	3	4
共起語ネットワークは適切であるか	3	4	5	2	5
3Dグラフによる出力は適切であるか	3	4	4	5	5
効率的な特許探索を行えそうか	5	4	5	4	4
新しい知見を発見できそうか	4	5	5	4	4
入力してもらったキーワード	・ スマホ ・ キーホルダー	・ アジ ・ 餌	・ ネットワーク ・ アローダイアグラム	・ 音楽 ・ 楽曲 ・ ゲーム	・ アメリカ ・ インド ・ ドイツ

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、従来よりも多くの情報を見ることができ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られる。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．

2024 年 2 月

平井 遥斗

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>

