

パーセプトロン

浅川伸一 <asakawa@twcu.ac.jp>

1 パーセプトロン

パーセプトロン perceptron とはローゼンブラット (Rosenblatt, 1958) によって提案された図 1 のような 3 層の階層型ネットワークモデルである。パーセプトロンはマッカロック・ピッツの形式ニューロンを用いて学習則にヘップ則を使ったモデルで、単純な認識能力を獲得することができる。パーセプトロ

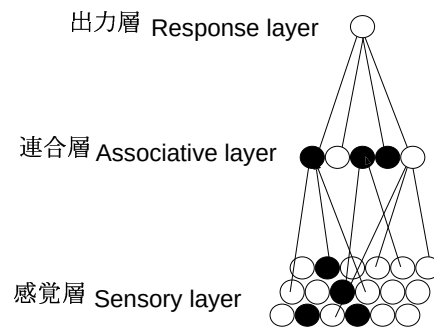


図 1: パーセプトロンの模式図

ンは 3 層の階層型ネットワークでそれぞれ、S(sensory layer), A(associative layer), R(response layer) と呼ばれる層からなっている。 $S \rightarrow A \rightarrow R$ のうちパーセプトロンの本質的な部分は $A \rightarrow R$ の間の学習にある。最下層の入力層は外界からの刺激入力を変換している。中間層では入力情報の変換が行われ、最上位層である出力層で認識に到ることになる。

パーセプトロンとは、入力パターンに現われる P^+ と P^- という 2 つのパターンを識別する問題を考えたとき、 P^+ が入力されたとき 1 を、 P^- のとき 0 を出力する機械である。あるいは、出力ユニットが 1 つしかない場合を考えれば、パーセプトロンは入力層に現われるパターンを 1 と 0 とに 2 分する機械であるといえる。出力層の i 番目のユニットへの入力 u_i は

$$u_i = \sum_j w_{ij} x_j - h_i = \mathbf{w}_i \cdot \mathbf{x}_i - h_i, \quad (1)$$

と表現される。ここで中間層の j 番目のユニットの出力 y_j とこのユニット

との結合係数を w_{ij} 、しきい値を h_i とした。このユニットの出力 y_i は、

$$y_i = \lceil u_i \rceil \quad \begin{cases} 1 & \text{if } u_i \geq 0, \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

と表される。すなわち、活性値 u_i は、式 (2) のような関数によって、1 または 0 の出力に変換されることを意味する。活性値が 0 より大きければ 1 を出力し、それ以外の場合は 0 となる。

1.1 パーセプトロンの学習

パーセプトロンの学習は連合層から反応層への結合係数の変化として表現される。最初の入力がなされる感覚層から連合層への結合係数は一定で変化しない。すなわちパーセプトロンは 3 層のネットワークではあるが、学習を考える際には、連合層と出力層の 2 層間の学習だけ考えれば良い。

パーセプトロンが特定のパターンに対してだけ出力するようになるためには、学習によってネットワーク内部の結合係数を変化させる必要がある。その際、パーセプトロンには種々の入力パターンが与えられ、それぞれが検出すべきパターンであるか否かが教師信号として与えられる。具体的には、入力信号を 2 分する問題を学習する場合、検出すべきパターンの時に教師信号として 1 が、それ以外のパターンの場合には、教師信号として 0 が与えられる。

パーセプトロンの学習は中間層 (連合層) から出力層への結合荷重の変化として表現され、入力層から中間層への結合荷重については考慮されないことに注意が必要である。パターン c に対する教師信号を t_c と書くことにするとパーセプトロンの学習、すなわち結合係数の更新式は

$$\Delta w_{ji} = \eta \delta x_{c,j} = \eta (t_{c,j} - y_{c,j}) x_{c,j}. \quad (3)$$

と表すことができる。ここで η は学習係数と呼ばれる定数である。パーセプトロンの出力と教師信号と差分 $\delta = t_{c,j} - y_{c,j}$ のことを誤差信号と呼ぶ。上式は w_{ji} の更新式として

$$w_{ji}(n+1) = w_{ji}(n) + \eta \Delta w_{ji}(n) = w_{ji}(n) + \eta (t_{c,j} - y_{c,j}) x_{c,j}. \quad (4)$$

のような漸化式として表現されることがある。ベクトル表現すれば

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta (\mathbf{t}_c - \mathbf{y}_c) \mathbf{x}_c. \quad (5)$$

となる。ここで、 $\mathbf{w}(n)$ は n 回目の学習終了時点での結合係数を入力層のユニットだけ並べて $(w_1, w_2, \dots, w_N)$ としたものである。 $\mathbf{t}_c$ はパターン c に対応する教師信号を表し、 $\mathbf{y}_c$ はパターン c に対するパーセプトロンの出力を表す。 $\mathbf{x}_c$ は入力信号である。

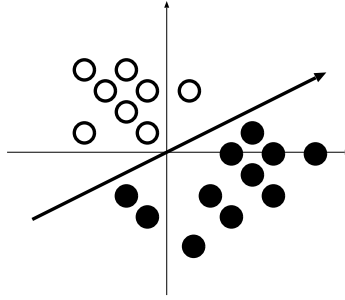


図 2: パーセプトロンの入力の幾何学的表現と判別直線

入力ユニット数 2、出力ユニット数 1 の 2 層で構成される単純なネットワークを考えれば、このネットワークへの入力全体は 2 次元平面で表現できる。この平面のことを入力空間と呼ぶことがある。入力ユニットが n 個あれば n 次元空間が入力空間になる。入力層のユニット数が 2 であれば平面である。パーセプトロンにおける学習とはこの入力空間を 2 分割するような領域に分割する判別直線を見つけることである。 n 次元の入力空間におけるパーセプトロンによるパターン分類とは、出力ユニットが 1 つの場合、 n 次元の入力パターン空間を $\mathbf{w}\mathbf{y} = \theta$ を満たす境界 ($n-1$ 次の超平面) によって 2 つの部分空間に分割することである。

いま、しきい値が 0、すなわち入力データを 2 群に分ける直線が原点を通る場合を考えることにしよう。すなわち、図 2 で判別直線より上の白丸には 1 を、判別直線より下の領域にある黒丸には 0 を出力するような学習を考える。このとき $w_1x_1 + w_2x_2 > 0$ であれば 1 を出力し $w_1x_1 + w_2x_2 \leq 0$ ならば 0 を出力することは、ベクトル (x_1, x_2) と ベクトル (w_1, w_2) との内積の正負の判断をしているのと同義である。なぜならベクトルの内積とは 2 通りの表現があって

$$\mathbf{w} \cdot \mathbf{x} = w_1x_1 + w_2x_2 = |\mathbf{w}| |\mathbf{x}| \cos \theta \quad (6)$$

だからである。上式の最右辺の正負は 2 つのベクトルのなす角 $\cos \theta$ のみによって決まる。 $\cos \theta$ が正となる範囲は 2 つのベクトルのなす角が $-\pi/2$ から $\pi/2$ までであるから 図 3 のように斜線をつけた領域にある全てのベクトルは、2 群を判別する直線の法線ベクトル (2 つの結合係数を要素とするベクトル) との内積が正となる。入力信号と結合係数ベクトルのなす角 θ が $-\pi/2 < \theta < \pi/2$ の範囲のとき $\cos \theta > 0$ となり、そうでなければ負になる。

このパーセプトロンに学習すべきデータが一つ入って来たと仮定しよう。図 4 は 1 と出力すべき ($y = 1$ すなわち法線ベクトルとの内積が正であるべき) データを、誤って 0 と出力してしまったという事態である。このとき、式 (5) 内の $(t - y)$ は $1 - 0 = 1$ になるので結合荷重の更新式、すなわち法

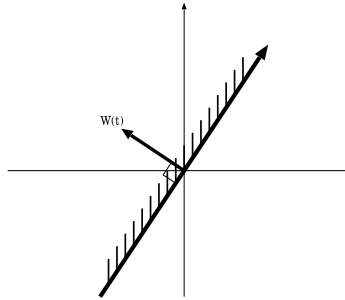


図 3: 2 群を判別するとき斜線の領域で正になるとは、法線ベクトルとの内積が正であることを意味する

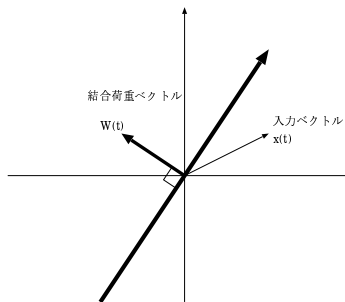


図 4: 学習すべきデータ

線ベクトルの更新式は

$$\mathbf{w}(n) + \eta(t_c - y_c) \mathbf{x}_c = \mathbf{w}(n) + \eta \mathbf{x}_c. \quad (7)$$

となってベクトルの足し算になる (図 5)。これによって判別直線が回転し (図 6)、今度は法線ベクトル $\mathbf{w}$ とデータ $\mathbf{x}$ とのなす角が 90 度以内になる。0 と出力すべきデータを、誤って 1 と出力してしまった場合はベクトルの引き算になる。

最後に判別直線が原点を通らない場合、すなわちしきい値が 0 ではない場合を説明しよう (図 7)。この場合は、原点を通る直線を並行移動したことになるので、原点を通る判別直線では 1 と答えるべき領域 (図 7 中の白マル) に入ってしまった点を底あげして 0 と答えるべき領域 (図 7 中の黒マル) の領域にするために行われると考えることができる。

パーセプトロンによるパターン分類とは、出力ユニットが 1 つの場合、入力パターン空間 (n 個の入力があれば n 次元空間) を $\mathbf{w}\mathbf{y} = \theta$ を満たす境界 ($n - 1$ 次の超平面) によって 2 つの部分空間に分割することであると言える。逆に言えば、入力パターン空間が一本の判別直線によって分割できないようなパターンをパーセプトロンは学習することができないことを意味して

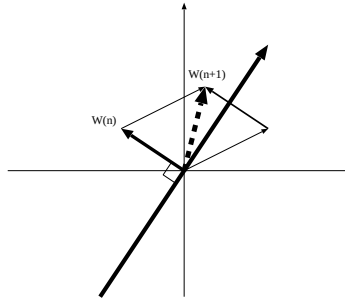


図 5: パーセプトロンの学習則による結合荷重ベクトル w の更新

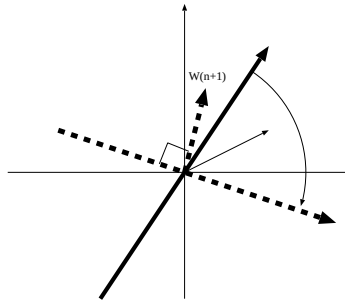


図 6: 判別直線の回転

いる。このことを線形分離可能性 (linear separability) という。

以上をまとめると、出力層と入力層との間の結合係数を入力データ空間におけるベクトルと考えれば、パーセプトロンの出力は結合係数ベクトルと入力データとの内積が 0 より大きければ 1 を、小さければ 0 を出力する機械であるみなすことができる。また、パーセプトロンの学習は、入力信号ベクトルと結合係数ベクトルの内積の大小によって結合係数ベクトルを回転させることだと言うこともできる。このようにして訓練されたパーセプトロンでは、入力データ空間上で線形分離可能な問題ならば、必ず学習が可能であることが重要である。逆に言えば、線形分離不可能な問題でも、適切に次元を設定することで線形分離可能な空間に写像するような中間層表現を得ることができれば、パーセプトロンの持つ限界を越えることができるというアイデアに結び付く。

ミンスキーとパパート (Minsky & Papert, 1988) はパーセプトロンのベクトル表示について悲観的な考え方を持っているが、ここでは理解のしやすさを優先してベクトル表示による説明を用いた。

パーセプトロンのような階層型のネットワークは、中間層のユニットを基底関数とみなすことで、関数近似法の一つと考えことができる。このことか

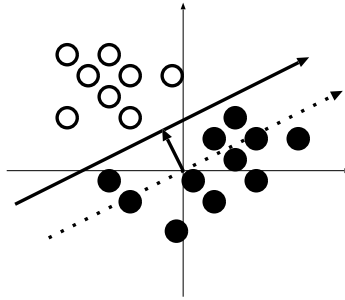


図 7: しきい値の意味

ら、他の数学的な関数近似手法との関連¹が指摘できる。一般にこれらの数学的手法は入力変数の次元数 n (この場合入力層のユニット数) が増えれば、一定の近似誤差内で関数を近似するためには、必要なパラメータ数 (この場合中間層のユニット数) が n 乗のオーダーで増加することが知られており、このことは次元の呪い (curse of dimensionality) と呼ばれている。多層パーセプトロンによる関数近似では、この次元の呪いを避けることができることが指摘されている。

1.2 パーセプトロンの情報処理能力

ミンスキーとパパート (Minsky & Papert, 1988) によるパーセプトロンの能力を示した定理を幾つか紹介しておく。

パーセプトロンの収束定理: 解くべき問題が線形分離可能であるならば、パーセプトロンの学習は有限回で停止し、そのとき得られた結合係数はすべての入力パターンに対して正解を与えるものになっている。

パーセプトロンの循環定理: 解くべき問題が線形分離可能でないとき、パーセプトロンは一般に収束しないが w_{ji} の大きさは有界であり有限個の結合状態を循環する。

パーセプトロンの群不変定理: ある変換群のもとで幾何学的な性質を判断するパーセプトロンに、その変換群のもとで同値な w_{ji} を使うことで同値な値に対して同じ出力を得ることができる。

その他、パーセプトロンに代表される階層型のネットワークの能力についての研究が多く研究者によって進められてきた (概説は (上坂吉則, 1993) など)。たとえば、

¹たとえばテーラー展開、フーリエ級数展開、スプライン関数など。

論理関数の計算に関する完全性: それぞれのユニットがしきいユニットである中間層のユニットを必要なだけたくさん使えば、3 層のネットワークによって任意の 2 値論理関数を実現できる。

連続関数のシミュレータとしての完全性 中間層のユニットを必要なだけたくさんつかえば、ユニット間の結合 (およびジグモイド状関数のしきい値) を適切に設定することによって、任意の連続な関数 $f([0, 1]^n \rightarrow (0, 1)^m)$ を任意の精度で近似することができる (船橋賢一, 2000)。

いかに複雑な関数であっても絶対可積分であれば 2 層のパーセプトロンで実現できることが証明されている。

などがある。

1.3 パーセプトロンの限界

パーセプトロンの学習 (1.1 節) でみたとおり、パーセプトロンは線形分離可能な問題でなければ解くことができない。パーセプトロンでは、入力層から中間層への結合についてはランダムであり学習が定義されていないからである。ネットワークの出力と教師信号と間の誤差を中間層以下のより下層へ伝播させることができれば、線形分離不可能な問題を正しく学習できるかも知れない訳である。

ミンスキーとパパート (Minsky & Papert, 1988) は図形の連結性を識別する簡単な問題も有限直径パーセプトロン (diameter-limited perceptron) では解けないことも例示している。たとえば、図 8 のような問題である。 x_{00} の

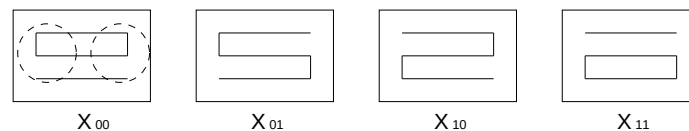


図 8: 図形の連結性を認識する問題

2 つの丸 (点線) で描かれている場所を見ている (受容野の位置と考えればよい) 2 つのユニット (A_1, A_2) があつたとする。このパーセプトロンが連結図形 (X_{10}, X_{01}) と非連結図形 (X_{00}, X_{11}) とを $W_1 A_1 + W_2 A_2 + \theta > 0$ によって識別できないことが分かる。

ミンスキーとパパートは、この他にも結合係数が非常に大きくなってしまふ問題、実質的な問題を解くためには膨大な数の中間層のユニット数が必要となる問題なども指摘している。

2 実習：パーセプトロンによる線分の認識

パーセプトロンに次のような 5×5 の受容野を考えて、線分を学習させることを考える。

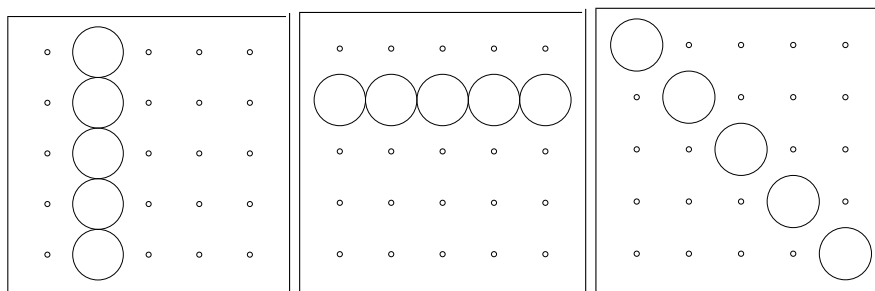


図 9: 学習パターン

縦線 5 本、横線 5 本、斜線 2 本の計 12 パターンをパーセプトロンに学習させることを考えよう。入力データは 5×5 の正方格子の線分とし、出力および教師信号の次元は 12 次元のベクトルとした。各々の線分に対して一つの出力層ユニットが応答するように訓練した。

結果を可視化するために出力層から入力層への結合係数の大きさを円で表したグラフを用いる。正の結合係数を○で、負の結合係数を●で表し、結合係数の大きさを円の直径で表したグラフをヒントンググラフという。

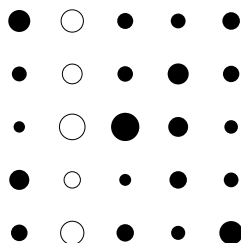


図 10: 学習結果のヒントンググラフ

結果はグラフから明らかである。左から 2 列目の縦線に応答する出力層ユニットから入力層ユニットへの結合係数は、左から 2 列目のみ正の結合で、その他の領域では負になっている。このことからこのユニットは左から 2 列目の縦線を検出する特性を獲得したといえる。

ではパーセプトロンに 5 本の縦線全てに応答するように、すなわち縦線概念を獲得するようにさせることは可能だろうか。

文献

- 上坂吉則 (1993). **ニューロコンピューティングの数学的基礎**. 近代数学社.
- 船橋賢一 (2000). 多層パーセプトロン. In 甘利 & 外山 (Eds.), **脳科学大辞典**. 朝倉書店.
- Minsky, M. & Papert, S. (1988). *Perceptrons, Expanded Edition* (2 Ed.). Cambridge, MA: MIT Press. パーセプトロン, 中野馨, 坂口豊訳, パーソナルメディア, 1993.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65, 386–408. In J. A. Anderson and E. Rosenfeld (Eds.) *Neurocomputing* (1988), MIT Press.