パターン認識とニューラルネットワーク

栗田多喜夫
脳神経情報研究部門
産業技術総合研究所
E-mail: takio-kurita@aist.go.jp

1　パターン認識とは

パターン認識 (pattern recognition) は、認識対象がいくつかの概念に分類できる時、観測されたパターンをそれらの概念のうちのひとつに対応させる処理である。この概念をクラス (class) あるいは類 (category) と呼ぶ。例えば、数字の認識は、入力パターンを 10 種類の数字のいずれかに対応させることが目的である。

パターンを認識する機械を実現するためには、まず、認識対象から何らかの特徴量を計測 (抽出) するための方法を考えなければならない。一般には、そのような特徴量は 1 種類だけではなく、複数の特徴量を計測し、それらを同時に認識に用いることが多い。例えば、文字認識の場合には、スキャナ等で取り込んだ画像そのものを特徴とみなすこともあるが、文字の識別に必要な本質的な特徴のみを抽出するのが一般的である。特徴量としては、例えば、文字線の傾き、曲率、面積などが用いられたりする。そのような特徴量は、通常、まとめて特徴ベクトル (feature vector) $\boldsymbol{x} = (x_1, x_2, \cdots, x_M)^T$ として表される。ここで、$\boldsymbol{x}^T$ は、$\boldsymbol{x}$ の転置を表す。また、M は、特徴量の個数である。一般に、特徴ベクトルによって張られる空間を特徴空間 (feature space) と呼ぶ。この場合、各パターンは、特徴空間上の 1 点として表される。認識対象のクラスの総数を K とし、各クラスを $C_1, C_2, \ldots, C_K$ と表すことにする。もし、特徴ベクトルの選び方が適切ならば、同じクラスの特徴ベクトルは互いに似ており、異なるクラスの特徴ベクトルは互いに違っていると考えられるので、特徴空間上では、各クラスごとにまとまった塊となるはずである。このような塊をクラスタ (cluster) と呼ぶ。

パターン認識における最も基本的な課題は、未知の認識対象を計測して得られた特徴ベクトル $\boldsymbol{x}$ からその対象がどのクラスに属するかを判定する識別方法を開発することである。そのためには、まず、クラスの帰属が既知の学習用のサンプル集合から特徴ベクトルとクラスとの確率的な対応関係を知識として学習することが必要である。このような学習は、教師あり学習と呼ばれている。次に、学習された特徴ベクトルとクラスとの対応関係に関する確率的知識を利用して、与えられた未知の認識対象の特徴からその認識対象がどのクラスに属していたかを推定 (決定) する方式が必要となる。その際、間違って識別する確率 (誤識別率) をできるだけ小さくすることが望ましい。このような識別の問題は、認識対象に依存しないこともあって、パターン認識の初期において、統計的決定理論 (statistical decision theory) を援用したベイズ識別理論 [3] として理論的に研究され、様々な実際的な方式が考えられている。

特徴ベクトルとクラスとの確率的な対応関係が完全にわかっている理想的な場合には、未知の認識対象を間違って他のクラスに識別する確率 (誤識別率) をできるだけ小さくするような理論的に

最適な識別方式（ベイズ識別方式）が知られている。2 章では、まず、その識別方式について概説する。しかし、実際の問題では、特徴ベクトルとクラスとの確率的な対応関係が完全わかっていることは稀であり、実際にパターン認識装置を設計する場合には、そのような確率的な関係をデータから推定（学習）する必要がある。3 章では、そのための確率密度分布の推定法について概説する。このような推定法とベイズ識別方式とを組み合わせることにより、一応、実際的なパターン認識器が実現できるようになる。

次に、4 章では、近年、パターン認識にも盛んに利用されるようになったニューラルネットについて、特に、統計的パターン認識の観点から述べる。

2　ベイズ決定理論

パターン認識では、未知の認識対象を間違って他のクラスに識別する確率 (誤識別率) をできるだけ小さくするような識別方式が最も望ましい。ここでは、まず、特徴ベクトルとクラスとの確率的な対応関係が完全にわかっている場合について、そのような識別方式がどうすれば実現できるかについて述べる。そのための理論的に最適な識別方式は、ベイズ識別方式と呼ばれている。

2.1　ベイズ決定方式

識別したい K 個のクラスを $C = \{C_k\}_{k=1}^{K}$ で表し、認識対象を計測して得られた特徴ベクトルの空間（特徴空間）を $X = \{\boldsymbol{x} \in R^M\}$ で表す。

識別対象がクラス C_k に属している確率 $P(C_k)$ は、事前確率 (*prior* probability) あるいは先見確率と呼ばれている。識別対象が K 個のクラスのどれかに属しているとすると、$\sum_{k=1}^{K} P(C_k) = 1$ が満たされる。また、あるクラス C_k に属する対象を計測した時、特徴ベクトル $\boldsymbol{x}$ が観測される確率密度分布を $p(\boldsymbol{x}|C_k)$ で表す。この時、当然、$\int p(\boldsymbol{x}|C_k)d\boldsymbol{x} = 1$ が満たされる。これらの確率がわかれば、特徴ベクトルとクラスとの確率的な関係はすべて計算できる。例えば、パターン認識で非常に重要な事後確率 (*posterior* probability)、つまり、ある対象から特徴ベクトル $\boldsymbol{x}$ が観測された時、それがクラス C_k に属している確率 $P(C_j|\boldsymbol{x})$ は、ベイズの公式 (Bayes theorem) から、

$$P(C_k|\boldsymbol{x}) = \frac{P(C_k)p(\boldsymbol{x}|C_k)}{p(\boldsymbol{x})}, \qquad \sum_{k=1}^{K} P(C_k|\boldsymbol{x}) = 1 \tag{1}$$

のように計算できる。ここで、

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P(C_k)p(\boldsymbol{x}|C_k), \qquad \int p(\boldsymbol{x})\, d\boldsymbol{x} = 1 \tag{2}$$

は $\boldsymbol{x}$ の確率密度分布である。

このように特徴ベクトルとクラスの関係が確率統計的知識として事前に完全にわかる場合には、識別の問題は以下のように統計的決定理論の枠組で完全に定式化される。

特徴ベクトル $\boldsymbol{x}$ に基づき対象がどのクラスに属するかを決定する関数（決定関数）を $d(\boldsymbol{x})$ で表し、クラス C_k の対象をクラス C_j に決定したときの損失 (loss) を $r(C_j|C_k)$ で表すと、損失の期待値（平均損失）は、

$$R[d] = \sum_{k=1}^{K} \int r(d(\boldsymbol{x})|C_k)P(C_k|\boldsymbol{x})p(\boldsymbol{x})\, d\boldsymbol{x} \tag{3}$$

となり、決定関数の汎関数となる。これを最小とする決定関数 $d(\boldsymbol{x})$ を求めるのが統計的 (ベイズ) 決定理論である。

特に 0-1 損失、つまり、誤った識別に対して均等な損失を与える場合には、損失関数は、

$$r(C_k|C_j) = 1 - \delta_{jk} \tag{4}$$

で与えられ、これを最小とする最適な識別関数は、

$$d(\boldsymbol{x}) = C_k \qquad \text{if} \qquad P(C_k|\boldsymbol{x}) = \max_j P(C_j|\boldsymbol{x}) \tag{5}$$

となる。これは、事後確率が最大となるクラスに決定する識別方式であり、ベイズ識別方式と呼ばれている。この識別関数によって達成される最小誤識別率は、

$$P_e^* = 1 - \int \max_j P(C_j|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} \tag{6}$$

で与えられる。

また、識別したいクラスが 2 つ ($K = 2$) の場合には、さらに簡単になり、最適な識別方式は、

$$\text{If} \;\; P(C_1|\boldsymbol{x}) \geq P(C_2|\boldsymbol{x}), \;\; \text{then} \;\; \boldsymbol{x} \in C_1, \;\; \text{else} \;\; \boldsymbol{x} \in C_2 \tag{7}$$

のように事後確率の大小を比較して識別すればよい。これは、尤度比検定

$$\text{If} \;\; \text{Ł} = \frac{p(\boldsymbol{x}|C_1)}{p(\boldsymbol{x}|C_2)} \geq \theta, \;\; \text{then} \;\; \boldsymbol{x} \in C_1, \;\; \text{else} \;\; \boldsymbol{x} \in C_2 \tag{8}$$

と等価となる。ただし、閾値 θ は、$\theta = P(C_2)/P(C_1)$ である。

実際のパターン認識の応用では、K 個 のクラスのどのクラスにも識別できないような対象が含まれることがある。そのような対象は、識別できない対象のクラスとして区別できれば簡単である。今、そのようなクラスを D とし、対象を K 個のクラス $C = \{C_k\}_{k=1}^K$ および識別できない対象のクラス D のいずれかに決定する識別関数 $\hat{d}(\boldsymbol{x})$ を考えてみよう。また、損失関数も識別できなかった場合の損失も考えて、$\hat{r}(C_j|C_k) = 1 - \delta_{jk}$ および $\hat{r}(D|C_k) = d$ とする。この場合には、最適な識別関数は、

$$\begin{aligned} \hat{d}(\boldsymbol{x}) &= C_k \qquad \text{if} \qquad P(C_k|\boldsymbol{x}) = \max_j P(C_j|\boldsymbol{x}) \quad \text{and} \quad P(C_k|\boldsymbol{x}) > 1 - d \\ \hat{d}(\boldsymbol{x}) &= D \qquad \text{if} \qquad \text{each} \;\; P(C_j|\boldsymbol{x}) \leq 1 - d \end{aligned} \tag{9}$$

となる [2]。

2.2 正規分布の場合

次に、クラス C_k に属する対象を計測して特徴ベクトル $\boldsymbol{x}$ が観測される確率密度分布が $p(\boldsymbol{x}|C_k)$ が、平均 $\boldsymbol{\mu}_k$、共分散行列 Σ_k の多変量正規分布

$$p(\boldsymbol{x}|C_k) = \frac{1}{(\sqrt{2\pi})^M \sqrt{|\Sigma_k|}} \exp\Big\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \Sigma_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)\Big\} \tag{10}$$

に従う場合について、具体的に、最適な識別関数を求めてみよう。ただし、記号 $\boldsymbol{x}^T$、$|\Sigma_k|$、および Σ_k^{-1} は、それぞれ、$\boldsymbol{x}$ の転置、行列 Σ_k の行列式、および行列 Σ_k の逆行列である。

ベイズ識別方式では、事後確率を最大とするクラスに決定する識別方式が最適であるが、事後確率の大小の比較のためには、対数を取って考えても結果は変わらない。また、$p(\boldsymbol{x})$ の項は、各クラスで共通であるため、それを無視すると、事後確率の対数は実質的に $\boldsymbol{x}$ の 2 次関数

$$g_k(\boldsymbol{x}) = \log P(C_k) - \frac{1}{2}\{(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\Sigma_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k) + \log|\Sigma_k|\} \tag{11}$$

を考えれば良いことになり、この値が最大のクラスに識別すれば良いことになる。このような関数 $g_k(\boldsymbol{x})$ を 2 次の識別関数と呼ぶ。

各クラスの分散共分散行列が等しい場合 ($\Sigma_k = \Sigma$) には、さらに簡単になり、

$$g_k(\boldsymbol{x}) = \boldsymbol{\mu}_k^T\Sigma^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_k^T\Sigma^{-1}\boldsymbol{\mu}_k + \log P(C_k) \tag{12}$$

のように $\boldsymbol{x}$ に関して 1 次の関数となる。これは、線形識別関数と呼ばれている。線形識別関数は、形の簡単さもあり、実際の応用で広く利用されている。

さらに、クラスが 2 つ ($K = 2$) で各クラスの共分散行列が等しい場合 ($\Sigma_1 = \Sigma_2 = \Sigma$) には、

$$\phi(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x}) = (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T\Sigma^{-1}\boldsymbol{x} - \frac{1}{2}(\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2) + \log\frac{P(C_1)}{P(C_2)} \tag{13}$$

のようにとなる。

また、特徴量が統計的に独立でそれぞれの分散が等しいく、しかも等方的な ($\Sigma_k = \sigma^2 I$) 場合には、事後確率の対数は、実質的に、

$$g_k(\boldsymbol{x}) = -\frac{||\boldsymbol{x}-\boldsymbol{\mu}_k||^2}{2\sigma^2} + \log P(C_k) \tag{14}$$

のようになる。これは、先見確率 $P(C_k)$ が等しい場合には、特徴ベクトル $\boldsymbol{x}$ と各クラスの平均ベクトル $\boldsymbol{\mu}_k$ との距離が最も近いクラスに決定する識別方式となる。つまり、各クラスの平均ベクトル $\boldsymbol{\mu}_k$ をテンプレートと考えると、特徴ベクトル $\boldsymbol{x}$ と各クラスのテンプレートとのマッチングにより識別する方式となる。

3 確率密度分布の推定

ベイズ決定理論は、期待損失最小の意味で最適な識別方式を与えるが、そのためには、事前に各クラスと特徴ベクトルとの間の確率分布構造が完全にわかっていなければならない。しかし、実際の応用では、背後の確率的構造があらかじめ完全にわかっていることは稀で、それらをデータから推定する必要があり、推定結果を利用してベイズ決定方式に基づく識別器を設計することになる。

確率密度分布の推定には、大きくわけて 3 つの方法がある [1]。第 1 の方法は、比較的少数のパラメータをもつパラメトリックモデル (parametric models) を用いて確率密度分布を表現し、そのモデルをデータに当てはめ、データに尤もよく合うパラメータを推定する方法である。この方法は、比較的簡単で、最もよく利用される方法であるが、モデルが真の分布を表現しきれない場合には問題がある。第 2 の方法は、特定の関数型を仮定しないで、データに依存して分布の形を決めるノンパラメトリックモデル (non-parametric models) を用いる方法である。この方法は、逆に、パラメータの数がデータとともに増大し、扱い難くなってしまうおそれがある。第 3 の方法は、これらの手法の中間的なもので、複雑な分布を表現するためにパラメータの数を系統的に増やせるようにすることで、パラメトリックモデルよりも一般的な関数型を表現するセミパラメトリック (semi-parametric) な手法である。その代表的なモデルとしては、混合分布モデル (mixture

distribution models) がある。また、階層型ニューラルネットワークもセミパラメトリックモデルの一種と考えることができる。

以下、それぞれについて、その代表的な方法について解説するが、ここで指摘しておきたいのは、有限のデータから確率密度分布を推定する問題はそれほど簡単では無いということである。特に、高次元の空間での有限のデータからの推定はかなり難しい。

3.1 パラメトリックモデル

パラメトリックモデルを用いて確率密度分布を推定するには、まず、確率密度を調整可能ないくつかのパラメータを用いてを表現する必要がある。正規分布は、最も簡単で、しかも、最も広く用いられているパラメトリックモデルのひとつである。次に、データからパラメータを推定する必要があるが、そのための代表的な手法には、最尤法 (maximum likelihood) とベイズ推定 (Bayesian inference) がある。これらの手法は、ほぼ同様の推定結果を与えるが、概念的には、かなり異なった手法である。

3.1.1 最尤法

最尤法では、学習データから導かれる尤度 (likelihood) を最大とするようなパラメータを探索する。求めたい確率密度関数が P 個のパラメータ $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_P)$ を用いて $p(\boldsymbol{x}; \boldsymbol{\theta})$ のように表されているとする。N 個の学習用データの集合 $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ が与えられた時、これらのデータが確率分布 $p(\boldsymbol{x}; \boldsymbol{\theta})$ の独立なサンプルである尤もらしさ (尤度) は、

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i; \boldsymbol{\theta}) \tag{15}$$

で定義される。

最尤法では、この尤度を最大とするようなパラメータ $\boldsymbol{\theta}$ を求める。実際には、尤度の対数 (対数尤度) をとって、

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\boldsymbol{x}_i; \boldsymbol{\theta}) \tag{16}$$

を最大とするパラメータを求めることが多い。

一般には、最適なパラメータ (最尤解) は、数値計算法を用いて求められるが、多変量正規分布モデルの場合には、最尤解を解析的に求めることができ、

$$\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i \tag{17}$$

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T \tag{18}$$

となる。これは、平均ベクトル $\boldsymbol{\mu}$ の最尤推定 $\hat{\boldsymbol{\mu}}$ は、サンプル平均ベクトルとなり、共分散行列 Σ の最尤推定 $\hat{\Sigma}$ は、ベクトルの積 $(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T$ のサンプル平均となることを表しており、直観とも非常に良く合う結果である。

3.1.2 ベイズ推定

上記の最尤法では、パラメータ $\boldsymbol{\theta}$ を未知の定数として扱い、データから最も尤もらしいパラメータを一つ推定したが、ベイズ推定では、パラメータ $\boldsymbol{\theta}$ を仮に確率変数とみなして、パラメータの値の確信度を確率密度分布を用いて表現する。そして、データ X を観測する前にパラメータが取るであろう値の確率密度分布 $p(\boldsymbol{\theta})$ を事前確率として表現し、データが観測された後にパラメータが取るであろう値の確率密度分布 (事後確率密度分布) $p(\boldsymbol{\theta}|X)$ を推定する。一般に、データを観測する前には、パラメータがどんな値を取るかに関する情報が得られないので、パラメータの取るであろう値の確率密度分布 $p(\boldsymbol{\theta})$ は、広がった分布となる。データが観測されると事後確率密度分布 $p(\boldsymbol{\theta}|X)$ は、データと整合性の良いパラメータほど大きな値を持つような分布となる。つまり、事後確率分布は事前確率分布よりも狭い分布となる。このようなデータを観測することにより確率分布が先鋭化される現象は、ベイズ学習 (Bayesian learning) と呼ばれている。

今、N 個の学習用データの集合 $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ から事後確率密度分布 $p(\boldsymbol{\theta}|X)$ が計算できるとすると、学習用データと同じ分布から特徴ベクトル $\boldsymbol{x}$ が得られる確率密度分布は、

$$p(\boldsymbol{x}|X) = \int p(\boldsymbol{x}, \boldsymbol{\theta}|X) d\boldsymbol{\theta} \tag{19}$$

のように計算できる。ここで、条件付き確率密度分布の定義から

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{\theta}|X) &= p(\boldsymbol{x}|\boldsymbol{\theta}, X) p(\boldsymbol{\theta}|X) \\ &= p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|X) \end{aligned} \tag{20}$$

である。この時、$\boldsymbol{x}$ はパラメータのみに依存し、データ X に依存しない、つまり、$\boldsymbol{x}$ の確率密度分布は、パラメトリックモデルとして表現できるという仮定で議論しているので、$p(\boldsymbol{x}|\boldsymbol{\theta}, X) = p(\boldsymbol{x}|\boldsymbol{\theta})$ のように簡単化した。これを、式 (19) に代入すると、

$$p(\boldsymbol{x}|X) = \int p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|X) d\boldsymbol{\theta} \tag{21}$$

のように書ける。つまり、ベイズ推定では、パラメータ $\boldsymbol{\theta}$ の特定の値を決める代わりに、すべての可能な値を考え、$p(\boldsymbol{\theta}|X)$ を重みとした重み付き平均により $\boldsymbol{x}$ の確率密度分布を推定する。

学習データ $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ が同じ分布からの独立なサンプルと仮定すると、

$$p(X|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i; \boldsymbol{\theta}) \tag{22}$$

のようになる。これは、式 (15) の尤度の定義と同じである。ベイズの定理を用いると、

$$p(\boldsymbol{\theta}|X) = \frac{p(X|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(X)} = \frac{p(\boldsymbol{\theta})}{p(X)} \prod_{i=1}^{N} p(\boldsymbol{x}_i; \boldsymbol{\theta}) \tag{23}$$

のように計算できる。ここで、分母の正規化項は、

$$p(X) = \int p(\boldsymbol{\theta}) \prod_{i=1}^{N} p(\boldsymbol{x}_i; \boldsymbol{\theta}) d\boldsymbol{\theta} \tag{24}$$

である。

以上のように形式的には、$p(\boldsymbol{x}|X)$ を計算することができるが、一般には、このための積分計算はかなり複雑で、解析的な結果は正規分布などの一部の分布関数属についてのみ可能である。

3.2 ノンパラメトリックな方法

この場合、ノンパラメトリックという用語は、推定したい確率密度関数の形がデータに依存して決まり、予め指定されないという意味で用いる。その意味で、ヒストグラムは、最も簡単なノンパラメトリックな手法のひとつである。しかし、ヒストグラムによって推定された密度関数は、滑らかではない。また、高次元への拡張が難しい等の問題がある。ここでは、もう少し凝った手法として、核関数に基づく方法 (kernel-based methods) および K-NN 法 (K-nearest-neighours methods) について紹介する。

3.2.1 ノンパラメトリックな確率密度関数の推定

ノンパラメトリックな確率密度関数の推定のための基本的な考え方は、直観的には比較的単純である。

今、あるベクトル $\boldsymbol{x}$ が未知の確率密度関数 $p(\boldsymbol{x})$ からのサンプルであるとすると、このベクトル $\boldsymbol{x}$ がある領域 R の内側に入る確率 P は、

$$P = \int_R p(\boldsymbol{x}')d\boldsymbol{x}' \tag{25}$$

で与えられる。確率密度関数 $p(\boldsymbol{x})$ が連続で、領域 R 内でほとんど変化しない場合には、確率 P は、

$$P = \int_R p(\boldsymbol{x}')d\boldsymbol{x}' \approx p(\boldsymbol{x})V \tag{26}$$

と近似できる。ただし、$\boldsymbol{x}$ は領域 R 内の点であり、V は領域 R の体積である。

次に、独立な N 個のサンプルが与えられた場合を考えよう。この場合、N このうちの K 個が領域 R に入る確率は、二項分布の定義から

$$\Pr(K) = \binom{N}{K} P^K (1-P)^{N-K} \tag{27}$$

で与えられる。また、K の期待値は、

$$E[K] = NP \tag{28}$$

となる。二項分布は平均付近で鋭いピークを持つので、比 $\frac{K}{N}$ は確率 P の良い推定値であると考えられる。

これらの結果から、確率密度関数は、

$$p(\boldsymbol{x}) \approx \frac{K}{NV} \tag{29}$$

のように推定できることがわかる。

ただし、このような近似が成り立つためには、次の様な相反する要請を満足するように領域 R を選ばなければならない。まず、領域 R 内で確率密度関数 $p(\boldsymbol{x})$ があまり変化しないためには、領域 R は十分小さくなければならない。一方、二項分布が鋭いピークを持つためには、領域 R に入るサンプルの数が十分多くなければならないので、領域 R はある程度大きくなければならない。つまり、このような近似を成り立たせるためには適切な大きさの領域 R を選ぶ必要がある。

3.2.2 核関数に基づく方法

核関数に基づく方法では、領域 R の体積 V を固定して、データから K を決定する。今、領域 R として、点 $\boldsymbol{x}$ を中心とする辺の長さが h の超立方体 (hyper cube) を考えよう。この時、領域 R の体積は、

$$V = h^M \tag{30}$$

となる。原点を中心とする辺の長さが 1 の超立方体は、核関数

$$H(\boldsymbol{u}) = \begin{cases} 1 & |u_j| < 1/2 \quad j = 1, \ldots, M \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

を用いて表すことができる。従って、$H(\frac{(\boldsymbol{x}-\boldsymbol{x}_i)}{h})$ はデータ点 $\boldsymbol{x}_i$ が $\boldsymbol{x}$ を中心とする一辺 h の超立方体の内側にある時にのみ 1 となり、それ以外の場合は 0 となる。このような核関数 $H(\boldsymbol{u})$ は、Parzen の窓関数 (Parzen window) と呼ばれている。この核関数を用いると、N 個のデータのうち領域 R 内に入るデータの個数 K は、

$$K = \sum_{i=1}^{N} H\left(\frac{(\boldsymbol{x} - \boldsymbol{x}_i)}{h}\right) \tag{32}$$

のように表せる。求めたい確率密度分布 $p(\boldsymbol{x})$ は、式 (30) および式 (32) を式 (29) に代入することにより、

$$\tilde{p}(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{h^M} H\left(\frac{(\boldsymbol{x} - \boldsymbol{x}_i)}{h}\right) \tag{33}$$

により推定できる。ただし、Parzen の窓関数を用いた推定法では推定された密度分布は滑らかでは無い。これを滑らかにするためには、核関数として滑らかなものを利用する必要がある。滑らかな核関数として、一般に、多変量正規分布に基づく核関数が良く用いられる。この場合には、求めたい確率密度分布 $\tilde{p}(\boldsymbol{x})$ は、

$$\tilde{p}(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{(2\pi h^2)^{M/2}} \exp\left\{-\frac{||\boldsymbol{x} - \boldsymbol{x}_i||^2}{2h^2}\right\} \tag{34}$$

のように推定される。

このような核関数に基づく方法では、領域の大きさを h 変更することにより推定される密度分布の滑らかさが制御できる。もし推定される密度分布が滑らかさを大きくしすぎると、バイアスが大きくなり良い推定結果が得られなくなる。一方、滑らかさが十分で無い場合には、密度分布が個々の学習データに強く依存するようになり、推定結果の分散が大きくなってしまう。従って、良い推定結果を得るためには、滑らかさのパラメータを適切な値に決めることが重要となる。

学習データに対する尤度は、モデルの良さを測る基準であるが、滑らかさの値が小さいほど尤度の値が大きくなってしまうので、滑らかさのパラメータを決めるための基準としては適当ではない。つまり、滑らかさを制御するためには尤度以外の評価基準が必要となる。滑らかさのパラメータを決定する目的は、未知の真の確率密度分布 $p(\boldsymbol{x})$ に出来るだけ近い確率密度のモデルを $\tilde{p}(\boldsymbol{x})$ を求めることである。そのためには、二つの確率密度分布間の距離尺度が必要となるが、一般には、Kullback-Leibler の距離尺度

$$L = -\int p(\boldsymbol{x}) \log \frac{\tilde{p}(\boldsymbol{x})}{p(\boldsymbol{x})} d\boldsymbol{x} \tag{35}$$

を用いることが多い。これは、真の分布がわからないため実際には計算できないが、これを近似的に計算することにより、滑らかさのパラメータを決定する方法が考えられている。

3.2.3 K-NN 法

核関数に基づく方法では、領域 R の体積 V を固定して、データから K を決定するが、K-NN 法では、逆に K を固定して、V を決定することにより密度分布を推定する。点 $\boldsymbol{x}$ を中心とする超球を考え、その超球内にちょうど K 個のデータ点が含まれるまで超球の半径を大きくして行く。ちょうど K 個のデータ点が含まれるようになった超球の体積を $V(\boldsymbol{x})$ とすると、密度分布は式 (29) から、

$$\tilde{p}(\boldsymbol{x}) = \frac{K}{NV(\boldsymbol{x})} \tag{36}$$

のように推定できる。

K-NN 法では、超球に含まれるデータ点の個数 K を大きくすると、推定される密度分布は次第に滑らかになる。核関数に基づく方法の場合と同様に、滑らかさを大きくしすぎると、バイアスが大きくなり良い推定結果が得られなくなり、滑らかさが不十分な場合には、密度分布が個々の学習データに強く依存するようになり、推定結果の分散が大きくなってしまう。ここでも、滑らかさのパラメータを適切な値に決めることが重要となる。

パターン認識において確率密度分布を推定するのは、識別 (classifier) を構成するためであるが、K-NN 法を利用して各クラスの条件付き確率密度分布 $p(\boldsymbol{x}|C_k)$ を推定すると、以下のような簡単な識別器が構成できる。

今、学習データとして、クラス C_k から N_k 個の特徴ベクトルが得られたとする。また、全学習データ数は $N = \sum_{k=1}^{L} N_k$ とする。点 $\boldsymbol{x}$ を中心とする超球を考え、その中にちょうど K 個の学習データを含むまで超球の半径を大きくして行った時の超球の体積を $V(\boldsymbol{x})$ とする。また、その超球内には、クラス C_k のデータが K_k 個含まれているとする。この時、クラス C_{k} の条件付確率密度関数は、

$$\tilde{p}(\boldsymbol{x}|C_k) = \frac{K_k}{N_k V(\boldsymbol{x})} \tag{37}$$

のように推定できる。同様に、$\boldsymbol{x}$ の確率密度関数は、

$$\tilde{p}(\boldsymbol{x}) = \frac{K}{NV(\boldsymbol{x})} \tag{38}$$

となる。また、事前確率は、

$$\tilde{P}(C_k) = \frac{N_k}{N} \tag{39}$$

のように推定できるので、Bayes の定理より、事後確率は、

$$\tilde{P}(C_k|\boldsymbol{x}) = \frac{\tilde{P}(C_k)\tilde{p}(\boldsymbol{x}|C_k)}{\tilde{p}(\boldsymbol{x})} = \frac{K_k}{K} \tag{40}$$

となる。従って、誤り確率を最小とするような識別のためには、この値が最大となるクラスに識別すればよい。このような識別方法は、K-NN 識別規則 (K-NN classification rule) と呼ばれている。

3.3 セミパラメトリックな手法

これまで確率密度分布の推定方法として、パラメトリックモデルに基づく方法とノンパラメトリックな方法について述べた。パラメトリックモデルに基づく方法は、新しいデータに対する確率密度の計算が比較的簡単であるが、真の分布と仮定したモデルが異なる場合には必ずしも良い推定結果が得られるとは限らない。一方、ノンパラメトリックな手法は、真の確率密度分布がどんな関

数系であっても推定できるが、新しいデータに対して確率密度を評価するための計算量が学習用のデータ数が増えるとどんどん増大してしまう。セミパラメトリックな手法は、パラメトリックモデルに基づく方法とノンパラメトリックな方法の中間的な手法であり、これらの手法の良い点を取り入れ、欠点を改善するような手法である。以下では、セミパラメトリックな方法の代表例として、混合分布モデル (mixture model) に基づく方法ついて述べる。

3.3.1 混合分布モデル (Mixture Model)

確率密度分布 $p(\boldsymbol{x})$ が、O 個の確率密度分布 $\{p(\boldsymbol{x}|j); j = 1, \ldots, O\}$ の重み付き線形結合

$$p(\boldsymbol{x}) = \sum_{j=1}^{O} \omega_j p(\boldsymbol{x}|j) \tag{41}$$

によってモデル化できるとする。このような分布は、混合分布 (mixture distribution) と呼ばれている。また、重み係数 ω_j は、混合パラメータ (mixinig parameter) と呼ばれており、条件

$$\sum_{j=1}^{O} \omega_j = 1, \quad 0 \leq \omega_j \leq 1 \tag{42}$$

を満たすものとする。同様に、各確率密度分布 $p(\boldsymbol{x}|j)$ は、

$$\int p(\boldsymbol{x}|j) d\boldsymbol{x} = 1 \tag{43}$$

を満たすものとする。

以下では、確率密度分布として、平均 $\boldsymbol{\mu}_j$、共分散行列 $\Sigma_j = \sigma_j^2 I$ の正規分布

$$p(\boldsymbol{x}|j) = \frac{1}{(2\pi\sigma_j^2)^{d/2}} \exp\left\{ -\frac{||\boldsymbol{x} - \boldsymbol{\mu}_j||^2}{2\sigma_j^2} \right\} \tag{44}$$

の場合を例にパラメータの推定法について説明する。

3.3.2 パラメータの最尤法

各確率密度分布が式 (44) の正規分布に従う場合の混合分布では、パラメータとして、重み係数 ω_j、 各確率密度の平均 $\boldsymbol{\mu}_j$ および分散 σ_j を推定する必要がある。学習用の N 個 のデータ $\{\boldsymbol{x}_n|n = 1, \ldots, N\}$ から最尤法でこれらのパラメータを推定することを考える。与えられた学習用のデータに対する尤度は、

$$L = \prod_{n=1}^{N} p(\boldsymbol{x}_n) = \prod_{n=1}^{N} \left\{ \sum_{j=1}^{O} \omega_j p(\boldsymbol{x}_n|j) \right\} \tag{45}$$

となる。従って、対数尤度 l は、

$$l = \log L = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n) = \sum_{n=1}^{N} \log \left\{ \sum_{j=1}^{O} \omega_j p(\boldsymbol{x}_n|j) \right\} \tag{46}$$

のようなる。

対数尤度 l を最大とするようなパラメータは、非線形最適化手法を用いて求めることができる。ただし、パラメータの選び方によっては、対数尤度が無限大になってしまうので、それを避けるための工夫が必要となる。対数尤度 l はパラメータに関して微分可能な連続関数であるので、パラメータ $\boldsymbol{\mu}_j$ および σ_j で偏微分すると

$$\frac{\partial l}{\partial \boldsymbol{\mu}_j} = \sum_{n=1}^{N} \frac{\omega_j p(\boldsymbol{x}_n|j)}{p(\boldsymbol{x}_n)} \frac{(\boldsymbol{x}_n - \boldsymbol{\mu}_j)}{2\sigma_j^2} = \sum_{n=1}^{N} P(j|\boldsymbol{x}_n) \frac{(\boldsymbol{x}_n - \boldsymbol{\mu}_j)}{2\sigma_j^2} \tag{47}$$

$$\frac{\partial l}{\partial \sigma_j} = \sum_{n=1}^{N} \frac{\omega_j p(\boldsymbol{x}_n|j)}{p(\boldsymbol{x}_n)} \left\{ -\frac{d}{\sigma_j} + \frac{||\boldsymbol{x}_n - \boldsymbol{\mu}_j||^2}{\sigma_j^3} \right\} = \sum_{n=1}^{N} P(j|\boldsymbol{x}_n) \left\{ -\frac{d}{\sigma_j} + \frac{||\boldsymbol{x}_n - \boldsymbol{\mu}_j||^2}{\sigma_j^3} \right\} \tag{48}$$

となる。ただし、

$$P(j|\boldsymbol{x}_n) = \frac{\omega_j p(\boldsymbol{x}_n|j)}{p(\boldsymbol{x}_n)} \tag{49}$$

である。一方、混合パラメータ ω_j は、条件 (42) を満たす必要がある。補助パラメータ γ_j を用いて

$$\omega_j = \frac{\exp(\gamma_j)}{\sum_{k=1}^{O} \exp(\gamma_k)} \tag{50}$$

のように定義すると、混合パラメータ ω_j は条件を満たすようになる。これは softmax 関数と呼ばれている。対数尤度 l を補助パラメータ γ_j で変微分すると、

$$\frac{\partial l}{\partial \gamma_j} = \sum_{k=1}^{O} \frac{\partial l}{\partial \omega_k} \frac{\partial \omega_k}{\partial \gamma_j} = \sum_{n=1}^{N} \{P(j|\boldsymbol{x}_n) - \omega_j\} \tag{51}$$

となる。対数尤度の微分に関するこれらの結果を利用して、尤度を最大とするパラメータ (最尤解) を非線形最適化手法により求めることができる。

また、対数尤度の微分を 0 とおくことにより、最尤解に関して、

$$\hat{\omega}(j) = \frac{1}{N} \sum_{n=1}^{N} P(j|\boldsymbol{x}_n) \tag{52}$$

$$\hat{\boldsymbol{\mu}_j} = \frac{\sum_{n=1}^{N} P(j|\boldsymbol{x}_n) \boldsymbol{x}_n}{\sum_{n=1}^{N} P(j|\boldsymbol{x}_n)} \tag{53}$$

$$\hat{\sigma}_j^2 = \frac{1}{d} \frac{\sum_{n=1}^{N} P(j|\boldsymbol{x}_n) ||\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_j||^2}{\sum_{n=1}^{N} P(j|\boldsymbol{x}_n)} \tag{54}$$

のような関係が成り立つことがわかる。これは、最尤解が各要素への帰属度を表す事後確率 $P(j|\boldsymbol{x}_n)$ を重みとして計算されることを示している。

3.3.3 EM アルゴリズム

EM アルゴリズムは、混合分布モデルのパラメータの推定にも利用できる不完全データからの学習アルゴリズムであり、最急降下法と同様に解を逐次改良することにより次第に最適な解に近づけて行く手法である。直感的にわかりやすく、しかも初期段階での収束性が速いことが知られている。その基本的な考え方は、統計学では古くから知られてたが、EM アルゴリズムの一般的な定式化は Dempster 等によって行われた [56]。また、最近、Amari は EM アルゴリズムの幾何学的な意味を明らかにし、なぜアルゴリズムがうまく働くかの直感的なイメージを与えた [57]。

平均 $\boldsymbol{\mu}_j$、共分散行列 $\Sigma_j = \sigma_j^2 I$ の正規分布の混合分布の場合、もしデータがどの正規分布から生成されたのかが全て分かっていれば非常に簡単になる。しかし、実際には、どの正規分布から生成されたのかも含めて推定する必要がある。EM アルゴリズムを適用するためには、どの正規分布から生成されたのか番号 z を含めたもの $(\boldsymbol{x}, z)$ を完全データとみなし、$\boldsymbol{x}$ を不完全データとみなす。この場合、完全データ $(\boldsymbol{x}, z)$ の分布は、

$$f(\boldsymbol{x}, z) = \omega_z p(\boldsymbol{x}|z) \tag{55}$$

のように書ける。また、この分布に従う N 個の完全データに対する対数尤度は、

$$\hat{l} = \sum_{n=1}^{N} \log f(\boldsymbol{x}_n, z_n) = \sum_{n=1}^{N} \log\{\omega_{z_n} p(\boldsymbol{x}_n|z_n)\} \tag{56}$$

となる。

EM アルゴリズムでは、パラメータの適当な初期値 $\boldsymbol{\theta}^{(0)}$ から初めて、E ステップ (Expectation step) と M ステップ (Maximization step) と呼ばれる二つの手続きを繰り返すことにより、パラメータの値を逐次更新する。今、繰り返し回数 t でのパラメータの推定値を $\boldsymbol{\theta}^{(t)}$ とすると、E ステップと M ステップでは、それぞれ、

E ステップ: 完全データの対数尤度 $\hat{l}$ のデータ $\boldsymbol{x}$ とパラメータ $\boldsymbol{\theta}^{(t)}$ に関する条件付き期待値

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = E[\log f(\boldsymbol{x}, z)|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})] \tag{57}$$

を計算する

M ステップ: $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ を最大とするパラメータを求め新しい推定値 $\boldsymbol{\theta}^{(t+1)}$ とする

のような計算を行う。このような E ステップと M ステップの繰り返しで得られるパラメータは尤度を単調に増加させることが知られている。

平均 $\boldsymbol{\mu}_j$、共分散行列 $\Sigma_j = \sigma_j^2 I$ の正規分布の混合分布の場合には、$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ を最大とするパラメータは陽に求まり、EM アルゴリズムの繰り返し計算は、

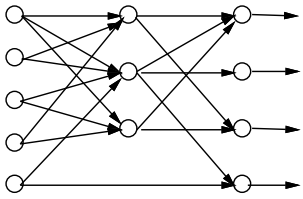
$$\hat{\omega}_j^{(t+1)} = \frac{1}{N} \sum_{n=1}^{N} P(j|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \tag{58}$$

$$\hat{\boldsymbol{\mu}}_j^{(t+1)} = \frac{\sum_{n=1}^{N} P(j|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \boldsymbol{x}_n}{\sum_{n=1}^{N} P(j|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)})} \tag{59}$$

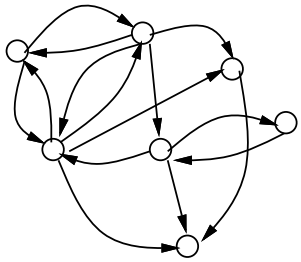
$$\hat{\sigma}_j^{2\ (t+1)} = \frac{1}{d} \frac{\sum_{n=1}^{N} P(j|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) ||\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_j||^2}{\sum_{n=1}^{N} P(j|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)})} \tag{60}$$

のようになる。これは、先に導出した式 (54) と全く同じであり、各要素への帰属度を表す事後確率の現時点での推定値 $P(j|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)})$ を重みとして、パラメータを更新することを繰り返せば良いことになる。

このような EM アルゴリズムは各繰り返しのステップで尤度を単調に増加させることから、他の同様な反復アルゴリズムに比べより数値的に安定な挙動を示すことが知られている。また、逆行列の計算が必要なく、Newton 法等の非線形最適化手法に比べて簡単である。しかも、多くの実例では他の手法に比べて良い解に収束することが知られており、繰り返しの初期の段階では Newton 法と同定度に速い。ただし、解の近くでは収束が遅くなるので、注意が必要である。また、大域的な収束は保証されていないので、初期値の選び方を工夫しなければならない場合もある。



(a) 階層的ネットワーク



(b) 相互結合ネットワーク

図 1: ネットワークの種類

4　階層型ニューラルネット

近年、パターン認識の分野でもニューラルネットが盛んに利用されるようになってきた。それは、ひとつには多層パーセプトロンの学習法として考案された誤差逆伝播法が、良い識別性能を持つ識別器を学習データから比較的簡単に構成できる効果的な学習法であることが認識されるようになったためである。

ニューラルネットはその構造から、図 1 (a) のような階層的なネットワークと図 1 (b) のような相互結合のある非階層的なネットワークに分類して考えることができる。階層型ネットワークは、図 1 (a) のように、ユニットが複数の階層をなすようにならび、入力層から出力層へ向かう一方向の結合のみが許されるネットワークである。現在、最もよく使われている多層パーセプトロンは、このタイプのネットワークの代表例である。多層パーセプトロン以外では、Radial Basis Function (RBF) ネットワークなども階層型ニューラルネットと考えることができる。階層型のネットワークでは、通常、各ユニットの出力がそのユニットへの入力のみによって決まる。そのため、静的ネットワークと呼ばれることもある。一方、相互結合ネットワークは、図 1 (b) のように、任意のふたつのユニット間に双方向の結合を許すようなネットワークである。その代表例は、Hopfield のネットワーク [8, 9] と Boltzmann Machine [10, 11, 12] である。これらのネットワークは、ユニットの入出力関係が微分方程式により記述されるので、動的ネットワークと呼ばれることもある。

以下では、多層パーセプトロンと Radial Basis Function (RBF) ネットワークに関連する話題について、特に、統計的パターン認識の観点から概説する。

4.1　多層パーセプトロン

パーセプトロン [5] の拡張としての多層パーセプトロンは、Rumelhart らが誤差逆伝搬学習法 [6, 7] と呼ばれるネットワークのパラメータを推定するためのアルゴリズムを提案して以来、パターン認識や制御などのさまざまな問題に適用され、いくつかの問題で実際に役立つようになってき

た。また、理論的には、ネットワークの能力、学習アルゴリズムの高速化、多変量データ解析との関係、汎化能力の高いネットワークを構成するための方法などに関する多くの知見が得られている。

以下では、まず、多層パーセプトロンの原形である単純パーセプトロンについて簡単に説明し、その拡張としての多層パーセプトロンがどんなモデルかを説明し、その能力に関する理論的な結果について紹介する。

4.1.1 単純パーセプトロン

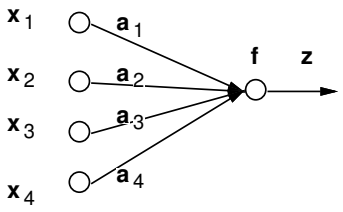


図 2: 単純パーセプトロンの例

パーセプトロンは 1985 年に Rosenblatt が提案した学習する識別機械である。図 2 にその例を示す。以下では多層パーセプトロンと区別するためにこのネットワークを単純パーセプトロンと呼ぶものとする。単純パーセプトロンでは、入力 $\boldsymbol{x} = (x_1, \ldots, x_I)^T$ に対する出力 z は

$$\begin{aligned} z &= f(\eta) \\ \eta &= \sum_{i=1}^{I} a_i x_i + a_0 = \boldsymbol{a}^T \tilde{\boldsymbol{a}} \end{aligned} \tag{61}$$

のように計算される。ここで、a_i は、i 番目の入力から出力への結合荷重であり、a_0 はバイアスである。これらをまとめて、$\boldsymbol{a} = (a_0, a_1, \ldots, a_I)^T$ のように表すものとする。また、入力特徴ベクトルに定数項を加えたベクトルを $\tilde{\boldsymbol{x}} = (1, x_1, \ldots, x_I)^T$ と表す。出力ユニットの入出力関数 f は、Rosenblatt のオリジナルなモデルではしきい関数

$$f(\eta) = \begin{cases} 1 & \text{if } \eta \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{62}$$

が用いられた。この他の入出力関数としてはロジスティック関数

$$f(\eta) = \frac{1}{1 + \exp(-\eta)} \tag{63}$$

や線形関数

$$f(\eta) = \eta \tag{64}$$

がよく使われる。多変量データ解析的用語を用いれば、出力ユニットの入出力関数が線形関数の単純パーセプトロンは、線形重回帰モデルであり、ロジスティック関数の単純パーセプトロンは、ロジスティック回帰モデルである。つまり、単純パーセプトロンは、多変量データ解析と密接に関係している。

4.1.2 単純パーセプトロンの学習

単純パーセプトロンの結合荷重 (パラメータ) を推定するための学習アルゴリズムとしていくつかの方法が提案されているが、Rosenblatt らの方法は、ネットワークにあるパターンを分類させてみて間違っていたら結合荷重を修正する誤り訂正型の方法であった。しかし、この学習規則は、線形分離可能でない場合、すなわち、誤識別 0 にする線形識別関数が存在しない場合には、誤り訂正の手続きを無限に繰り返しても解に到達できない可能性がある。また、学習を途中で打ち切った場合に得られるパラメータが最適であるという保証がない。

これに対して、出力ユニットの入出力関数として線形関数を用い、ネットワークの出力と教師信号と平均 2 乗誤差を最小にするような結合荷重を推定する場合には、平均 2 乗誤差の意味で最適なパラメータを求めることができる。

今、P 個の学習用のデータを $\{\boldsymbol{x}_p, u_p | p = 1, \ldots, P\}$ とする。ここで、$\boldsymbol{x}_p$ が入力ベクトルで、その入力ベクトルに対する望みの出力 (教師信号) が u_p である。この時、この学習用のデータに対する 2 乗誤差は、

$$\varepsilon_{emp}^2 = \sum_{p=1}^{P} (u_p - z_p)^2 = \sum_{p=1}^{P} \varepsilon_{emp}^2(p) \tag{65}$$

となる。最適なパラメータを求めるために、パラメータ (結合荷重) $\boldsymbol{a}$ を逐次更新することにより次第に最適なパラメータに近似させる最急降下法を用いることにすると、2 乗誤差 ε_{emp}^2 のパラメータに関する偏微分を計算する必要がある。2 乗誤差 ε_{emp}^2 のパラメータに関する偏微分は、

$$\frac{\partial \varepsilon_{emp}^2}{\partial a_i} = \sum_{p=1}^{P} -2\delta_p x_{pi} \tag{66}$$

となる。ただし、$x_{p0} = 1$ とする。また、$\delta_p = (u_p - z_p)$ である。従って、最急降下法によるパラメータの更新式は、

$$a_i \Leftarrow a_i + \alpha(\sum_{p=1}^{P} \delta_p x_{pi}) \tag{67}$$

のようになる。これは、Widrow-Hoff の学習規則 (Widrow-Hoff learning rule) と呼ばれている。また、教師信号 u_p とネットワークの出力 z_p の誤差 δ_p に応じてパラメータを修正するため、デルタルール (delta rule) と呼ばれることもある。

Widrow-Hoff の学習規則では、最急降下法を用いて逐次近似によりパラメータを推定するが、最適な解を行列計算により陽に求めることも可能である。

今、学習用データの入力ベクトルを並べた $N \times (M+1)$ 次元の行列を $X = (\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_P)^T$ とし、教師信号を並べた N 次元のベクトルを $\boldsymbol{u} = (u_1, \ldots, u_P)^T$ とする。これらを用いると 2 乗誤差は、

$$\varepsilon_{emp}^2 = \sum_{p=1}^{P} (u_p - z_p)^2 = ||\boldsymbol{u} - X\boldsymbol{a}||^2 \tag{68}$$

のように書ける。これをパラメータ bma で偏微分し、0 と置くと、

$$\frac{\partial \varepsilon_{emp}^2}{\partial \boldsymbol{a}} = X^T(\boldsymbol{u} - X\boldsymbol{a}) = 0 \tag{69}$$

となる。従って、$(X^T X)$ が正則ならば、最適なパラメータは、

$$\boldsymbol{a} = (X^T X)^{-1} X^T \boldsymbol{u} \tag{70}$$

となる。これは、重回帰分析 (multiple regression analysis) と呼ばれる最も基本的な多変量データ解析と等価である。重回帰分析では、$\boldsymbol{x}$ を説明変数 (explanatory variable)、 u を目的変数 (criterion variable) と呼ぶ。

入出力関数としてロジスティック関数を用い、最尤法によりパラメータを推定する場合には、ロジスティック回帰と呼ばれる手法と等価となる。この場合には、ロジスティック回帰のためのパラメータ推定アルゴリズムとして知られているフィッシャーのスコアリングアルゴリズムを学習に利用することも可能である。

4.1.3 多層パーセプトロン

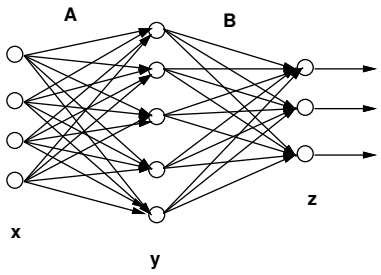


図 3: 多層パーセプトロンの例

多層パーセプトロンは、図 3 のように、単純パーセプトロンを層状に繋ぎ合わせたネットワークである。例えば、中間層が 1 層のネットワークでは、入力信号 $\boldsymbol{x}$ に対する出力信号 $\boldsymbol{z} = (z_1, \ldots, z_K)^T$ は、

$$\begin{aligned} \zeta_j &= \sum_{i=1}^{I} a_{ij} x_i + a_{0j} \\ y_j &= f_{hidden}(\zeta_j) \\ \eta_k &= \sum_{j=1}^{J} b_{jk} y_j + b_{0k} \\ z_k &= f_{out}(\eta_k) \end{aligned} \tag{71}$$

のように計算される。ただし、a_{ij} は、i 番目の入力から中間層の j 番目のユニットへの結合荷重であり、b_{jk} は、中間層の j 番目のユニットから出力層の k 番目のユニットへの結合荷重である。a_{0j} および b_{0k} は、それぞれ、中間層の j 番目のユニットおよび出力層の k 番目のユニットのバイアスである。また、f_{hidden} および f_{out} は、それぞれ、中間層のユニットの入出力関数および出力層のユニットの入出力関数である。中間層のユニットの入出力関数としては、普通、ロジスティック関数が使われる。出力層のユニットの入出力関数は、利用目的に応じて決められる。例えば、関数近似のためにネットワークを利用する場合には線形関数が使われ、パターン認識に利用する場合にはロジスティック関数とすることが多い。

このような多層パーセプトロンの能力、つまり、どのような関数が表現可能かに関して、非常に

強力な結果が得られている [16]。それは、中間層のユニットの入出力関数が

$$\sigma(t) = \begin{cases} 1 & \text{as } t \to +\infty \\ 0 & \text{as } t \to -\infty \end{cases} \tag{72}$$

のような性質を持つ非線形の連続な単調増加関数（シグモイド関数）であり、出力層の入出力関数が線形関数のとき、中間層が 1 層の多層パーセプトロンによって、任意の連続関数が近似可能であるというものである。もちろん、任意の連続関数を近似するためには中間層のユニットの数を非常に多くする必要があるかもしれない。この結果は、多層パーセプトロンを入出力関係を学習するために使うには、理論的には、中間層が 1 層のみのネットワークで十分であることを示している。

4.1.4 誤差逆伝搬学習法

多層パーセプトロンは任意の連続関数を近似するのに十分な表現能力をもっているが、そうしたネットワークに望みの情報処理をさせるためにはユニット間の結合荷重を適切なものに設定しなければならない。ユニットの数が増えると結合荷重の数も増え、それらをいちいち設定することは難しい。一般には、それらは利用可能なデータからの学習によって求めらる。そのためのアルゴリズムとしては、最急降下法に基づく誤差逆伝搬学習法 [6, 7] が有名である。

ここでは、中間層のユニットの入出力関数がロジスティック関数で、出力層のユニットの入出力関数が線形の中間層が 1 層のみネットワークに対する誤差逆伝搬学習法について説明する。

今、学習用のデータを $\{\boldsymbol{x}_p, \boldsymbol{u}_p\}$ とする。また、学習のための評価基準として 2 乗誤差

$$\varepsilon^2_{emp} = \sum_{p=1}^{P} ||\boldsymbol{u}_p - \boldsymbol{z}_p||^2 = \sum_{p=1}^{P} \varepsilon^2_{emp}(p) \tag{73}$$

を用いるとする。2 乗誤差 ε^2_{emp} の結合荷重に関する偏微分を計算すると、

$$\frac{\partial \varepsilon^2_{emp}}{\partial a_{ij}} = \sum_{p=1}^{P} \frac{\partial \varepsilon^2_{emp}(p)}{\partial a_{ij}} \tag{74}$$

$$\frac{\partial \varepsilon^2_{emp}}{\partial b_{jk}} = \sum_{p=1}^{P} \frac{\partial \varepsilon^2_{emp}(p)}{\partial b_{jk}} \tag{75}$$

となる。ただし、

$$\frac{\partial \varepsilon^2_{emp}(p)}{\partial a_{ij}} = -2\gamma_{pj}\nu_{pj}x_{pi} \tag{76}$$

$$\frac{\partial \varepsilon^2_{emp}(p)}{\partial b_{jk}} = -2\delta_{pk}y_{pj} \tag{77}$$

$$\nu_{pj} = y_{pj}(1 - y_{pj}) \tag{78}$$

$$\gamma_{pj} = \sum_{k=1}^{K} \delta_{pk} b_{jk} \tag{79}$$

$$\delta_{pk} = u_{pk} - z_{pk} \tag{80}$$

である。また、$x_{p0} = 1$ および $y_{p0} = 1$ としている。従って、最急降下法による結合荷重の更新式は

$$a_{ij} \Leftarrow a_{ij} - \alpha \frac{\partial \varepsilon^2_{emp}}{\partial a_{ij}} \tag{81}$$

$$b_{jk} \Leftarrow b_{jk} - \alpha \frac{\partial \varepsilon_{emp}^2}{\partial b_{jk}} \tag{82}$$

のようになる。ただし、α は学習率と呼ばれる正のパラメータである。このアルゴリズムは、教師信号とネットワークの出力との誤差 δ を結合荷重 b_{jk} を通して逆向きに伝搬して γ を計算していると解釈できるので誤差逆伝搬法と名付けられている。

上記のアルゴリズムは学習データ集合全体を見て結合荷重を修正しているが、学習データ毎に $\frac{\partial \varepsilon_{emp}^2(p)}{\partial a_{ij}}$ および $\frac{\partial \varepsilon_{emp}^2(p)}{\partial b_{jk}}$ により結合荷重を更新することも可能であり、実際にはこちらの方法を使うことが多い。

このような最急降下法を用いた学習法では、学習率をどのように決めるかによってアルゴリズムの収束の速さが影響を受けるので、学習率を適切な値に設定するための方法がいくつかの提案されている（例えば、[13]）。また、学習の高速化に関しては、多くの方法が提案されている。例えば、Quick Prop [34] は、多くのヒューリスティックを組み合わせて、学習を高速化している。

4.1.5 最尤推定としての定式化

上記の誤差逆伝搬学習法では、2 乗誤差を最小とするような結合荷重を最急降下法により求めた。ここでは、これを最尤推定の観点から見てみることにする。

今、教師信号 $\boldsymbol{u}_p$ とネットワークの出力 $\boldsymbol{z}_p$ との誤差 $\boldsymbol{e}_p = \boldsymbol{u}_p - \boldsymbol{z}_p$ が、互いに独立な正規分布 $N(\boldsymbol{0}, \sigma^2 I)$ に従うとする。この時、学習データ集合に対する教師信号とネットワーク出力の誤差の尤度は、

$$L = \prod_{p=1}^{P} (2\pi\sigma^2)^{\frac{K}{2}} \exp\{-\frac{1}{2\sigma^2} \boldsymbol{e}_p^T \boldsymbol{e}\} \tag{83}$$

となる。従って、その対数（対数尤度）は、

$$\begin{aligned} l &= -\frac{KP}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{p=1}^{P} \boldsymbol{e}_p^T \boldsymbol{e}_p \\ &= -\frac{KP}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \varepsilon_{emp}^2 \end{aligned} \tag{84}$$

となる。これを最大にすることは、第 2 項の ε_{emp}^2 を最小化することと同値になるので、上記の誤差逆伝搬学習法は最急降下法によりネットワークのパラメータ（結合荷重）を最尤推定しているとみなすことができる。

次に、ネットワークに対するフィッシャー情報行列を具体的に計算してみる。ここで対数尤度の微分に関する関係

$$0 = \mathrm{E}(\frac{\partial l_p}{\partial z_{pk}}) = \frac{1}{2\sigma^2} \{\mathrm{E}(t_{pk}) - z_{pk}\} \tag{85}$$

を用いて整理するとフィッシャーの情報行列の要素は、

$$\begin{aligned} F_{a_{lm} a_{ij}} &= -\mathrm{E}\{\frac{\partial^2 l}{\partial a_{lm} \partial a_{ij}}\} \\ &= \frac{1}{\sigma^2} \{\sum_{p=1}^{P} x_{pi} \nu_{pj} \chi_{jm} \nu_{pm} x_{pl}\} \\ F_{a_{lm} b_{jk}} &= -\mathrm{E}\{\frac{\partial^2 l}{\partial a_{lm} \partial b_{jk}}\} \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{\sigma^2}\{\sum_{p=1}^{P} y_{pj} b_{km} \nu_{pm} x_{pl}\} \\
F_{b_{mn}a_{ij}} &= -\mathrm{E}\{\frac{\partial^2 l}{\partial b_{mn}\partial a_{ij}}\} \\
&= \frac{1}{\sigma^2}\{\sum_{p=1}^{P} y_{pm} b_{nj} \nu_{pj} x_{pi}\} \\
F_{b_{mn}b_{jk}} &= -\mathrm{E}\{\frac{\partial^2 l}{\partial b_{mn}\partial b_{jk}}\} \\
&= \begin{cases} \frac{1}{\sigma^2}\{\sum_{p=1}^{P} y_{pj} y_{pm}\} & \text{if } n = k \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

となる。ただし、

$$\chi_{jm} = \sum_{k=1}^{K} b_{jk} b_{mk} \tag{86}$$

である。つまり、フィッシャー情報行列の結合荷重 a_{ij} に関する部分は入力ベクトル $\{\boldsymbol{x}_p\}$ の重みつき相関、結合荷重 b_{jk} に関する部分は中間層の出力 $\{\boldsymbol{y}_p\}$ の相関、結合荷重 a_{ij} と b_{jk} に関する部分は $\{\boldsymbol{x}_p\}$ と $\{\boldsymbol{y}_p\}$ の重みつき相関と関係している。

このフィッシャー情報行列 F を利用してネットワークの結合荷重を学習するアルゴリズムを構成することができる。今、ネットワークの結合荷重をまとめて $\boldsymbol{\theta} = (a_{01}, \ldots, a_{IJ}, b_{01}, \ldots, b_{JK})^T$ と書くとする。このとき、結合荷重の修正ベクトル $\delta\boldsymbol{\theta}$ をフィッシャー情報行列に関する線形方程式

$$F\delta\boldsymbol{\theta} = \nabla l \tag{87}$$

から求めて、結合荷重を

$$\boldsymbol{\theta} \Leftarrow \boldsymbol{\theta} + \delta\boldsymbol{\theta} \tag{88}$$

のように更新する。ただし、ここで ∇l は対数尤度 l の結合荷重に関する勾配を表す。これは最尤推定のためのフィッシャーのスコアリング法として知られているアルゴリズムをニューラルネットの学習へ応用したものである [17, 18]。

この学習アルゴリズムではフィッシャーの情報行列を完全な形で用いているが、大きなネットワークに対してそれを計算するのはかなり大変であり、また、式 (87) の線形方程式を解くのはさらに大変である。そのため、実用的にはなんらかの方法でこれをさぼる必要がある。また、並列計算機を利用することを考えると、各ユニットに関連する局所的な情報のみから学習できることが望ましい。そこで、フィッシャー情報行列のブロック対角線分のみを用いて残りの部分を無視したアルゴリズムを考えてみる [17, 18]。この場合には、中間層の j 番目のユニットへ入る結合荷重 $\boldsymbol{a}_j = (a_{0j}, \ldots, a_{Ij})^T$ の更新式は、

$$\boldsymbol{a}_j = (X^T W_{\boldsymbol{a}_j} X)^{-1} X^T W_{\boldsymbol{a}_j} (\boldsymbol{\zeta}_j + W_{\boldsymbol{a}_j} \boldsymbol{\delta}_{\boldsymbol{a}_j}) \tag{89}$$

となり、出力層の k 番目のユニットへ入る結合荷重 $\boldsymbol{b}_k = (b_{0k}, \ldots, b_{Jk})^T$ の更新式は、

$$\boldsymbol{b}_k = (Y^T Y)^{-1} Y^T (\boldsymbol{\eta}_k + \boldsymbol{\delta}_{\boldsymbol{b}_k}) \tag{90}$$

となる。ただし、

$$X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_P]^T$$

$$
\begin{aligned}
Y &= [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_P]^T \\
W_{\boldsymbol{a}_j} &= \mathrm{diag}(\nu_{pj}\chi_{jj}\nu_{pj}) \\
\boldsymbol{\zeta}_j &= (\zeta_{1j}, \ldots, \zeta_{Pj})^T \\
\boldsymbol{\eta}_k &= (\eta_{1k}, \ldots, \eta_{Pk})^T \\
\boldsymbol{\delta}_{\boldsymbol{a}_j} &= (\gamma_{1j}\nu_{1j}, \ldots, \gamma_{Pj}\nu_{Pj})^T \\
\boldsymbol{\delta}_{\boldsymbol{b}_k} &= (\delta_{1k}, \ldots, \delta_{Pk})^T
\end{aligned}
$$

である。これらは、各ユニット毎にそのユニットに入る結合荷重を重み付き最小２乗法を繰り返すことにより更新していると解釈できる。

4.2 多層パーセプトロンと非線形回帰

4.1.1 節で述べたように、単純パーセプトロンと多変量データ解析、特に、回帰分析とは密接な関係がある。従って、その拡張としての多層パーセプトロンも多変量データ解析と密接に関係していると考えられる。

ここでは、多層パーセプトロンが究極的に何を学習しようとしているのかを見るために、非線形回帰との関係について考察する。

多層パーセプトロンは、入力と望みの出力（教師信号）の対からなる学習用データ $\{\boldsymbol{x}_p, \boldsymbol{u}_p | p = 1, \ldots, P\}$ に基づいて、入力 $\boldsymbol{x} \in R^I$ から望みの出力 $\boldsymbol{u}$ を推定するような非線形の変換

$$
\hat{\boldsymbol{u}} = \Phi(\boldsymbol{x}) \tag{91}
$$

をユニット間の結合重荷を調節することによって構成するための手段であると考えることができる。出来上がったネットワークを関数関係の近似に利用する場合には、望みの出力は実数ベクトル $\boldsymbol{u} \in R^K$ とし、パターン認識に使う場合には、２値ベクトル $\boldsymbol{u} \in [0,1]^K$ とするのが一般的である。ユニット間の結合荷重を決定するための評価基準としては、普通、２乗誤差

$$
\varepsilon^2_{emp} = \sum_{p=1}^{P} ||\boldsymbol{u}_p - \Phi(\boldsymbol{x}_p)||^2 \tag{92}
$$

が使われている。つまり、多層パーセプトロンは、たくさんの単純なユニットを結合したネットワークを用いて非線形の回帰を行っているとみなすことができる。

今、中間層のユニットの個数を任意に多く用いることができ、従って、任意の連続関数が実現でき、また、学習サンプル $\{\boldsymbol{x}, \boldsymbol{u}\}$ が確率密度分布 $p(\boldsymbol{x}, \boldsymbol{u})$ で表される母集団から無数に得られるような理想的な場合を考えてみる。この場合には、平均２乗誤差

$$
\varepsilon^2(\Phi) = \int ||\boldsymbol{u}_p - \Phi(\boldsymbol{x}_p)||^2 p(\boldsymbol{x}, \boldsymbol{u}) d\boldsymbol{x} d\boldsymbol{u} \tag{93}
$$

を最小とするような最適な変換 Φ は、変分法を用いて陽に求められ、

$$
\hat{\boldsymbol{u}} = \Phi_{opt}(\boldsymbol{x}) = \int \boldsymbol{u} p(\boldsymbol{u}|\boldsymbol{x}) d\boldsymbol{u} \tag{94}
$$

となる [14, 15]。これは、入力 $\boldsymbol{x}$ をそのもとでの $\boldsymbol{u}$ の条件つき平均に写像することを示しており、非線形回帰として自然な結論となっている。例えば、この写像により達成される最小平均２乗誤差は、$\hat{\boldsymbol{u}}$ と $\boldsymbol{x}$ の相関係数 ρ を用いて

$$
\varepsilon^2_{opt} = \sigma^2_u (1 - \rho^2) \tag{95}
$$

のような線形回帰で見なれた関係が成り立つことが確かめられる。この変換 Φ 入力 $\boldsymbol{x}$ と 望みの出力 $\boldsymbol{u}$ との確率的な関係が完全にわかるような理想的な場合の結果であるが、実際のニューラルネットでは、有限個の学習サンプルからの学習によって、ネットワークの制約のもとでこの写像が近似的に実現されていると考えることができる。

多層パーセプトロンを入力パターン $\boldsymbol{x}$ を K 個のクラス $\{C_k|k=1,\ldots,K\}$ に識別する問題に応用する場合には、クラス C_k に対応して k 番目の要素のみが 1 で残りの要素が全て 0 の 2 値ベクトルを教師信号 $\boldsymbol{u}$ とするのが普通である。この場合には、上述の最適な写像は、

$$\hat{\boldsymbol{u}} = \Phi_{opt}(\boldsymbol{x}) = \begin{bmatrix} p(C_1|\boldsymbol{x}) \\ \vdots \\ p(C_K|\boldsymbol{x}) \end{bmatrix} \tag{96}$$

のようにベイズ事後確率を要素とするベクトルとなる。従って、この場合には、ニューラルネットは、有限個の学習サンプルからネットワークの制約のもとで事後確率を近似しているとみなすことができる。さらに、麻生ら (麻生 89, 麻生 90) は、パターン認識のための多層パーセプトロンと非線形判別分析との関係について考察している。同様な考察は、Webb ら (Webb90,Lowe91) にも報告されている。

4.3 汎化性

一般に、ネットワークが小さすぎると、その表現能力は低くなるので、課題を十分に学習させることが出来なくなり、明らかに、未知データに対しても有効に働くような汎化性を確保することは難しくなる。一方、4.1.3 節で述べたように、多層パーセプトロンは理論的には中間層のユニット数を多くすると任意の関数を近似できる能力を持つので、原理的には中間層のユニット数を多くすると学習データに対して平均 2 乗誤差のほとんどないネットワークを構成すること可能であるが、中間層のユニット数を多くすることによって学習データに対していくら近似を良くしても、必ずしもそのネットワークが未知データに対しても良い近似を与えるとは限らない。

実際問題に多層パーセプトロンを応用する場合には学習データに対してだけでなく未知のデータに対しても有効に働くような汎化能力の高いネットワークを構成することが非常に重要であるので、多層パーセプトロンの汎化能力に関する研究もさかんに行われている。以下では、代表的な方法について紹介する。

4.3.1 情報量基準による汎化能力の評価

統計では汎化性の問題は古くから議論されており、特に有名なのは情報量基準と jackknife 法 [23, 24] あるいは bootstrap 法 [25, 26, 27] などの resampling 手法である。情報量基準としては、赤池の AIC(An Information Theoretical Criterion)[28, 29] や Rissanen の MDL(Minimum Description Length)[30, 31] が有名である。多層パーセプトロンのの汎化能力を評価するためにこれらの手法を利用することが考えられるが、resampling 手法を用いるためには、resampling したデータに対して何度もネットワークの結合荷重を学習する必要があるので、その評価にはかなりの計算時間が必要となる。一方、情報量基準を用いて評価する方法は、学習は一回のみでよく、比較的簡便な評価が可能となる [32]。

4.1.5 節のように、多層パーセプトロンの結合係数の学習を最尤推定とみなすと、学習が収束したネットワークに対する対数尤度を最大対数尤度とみなして AIC や MDL などの情報量基準を近

似的に計算することにより、汎化能力を比較することが可能となる。

AIC は赤池により最大対数尤度と期待平均対数尤度の間の偏りの解析的評価から導出されたもので、最尤推定するモデルの自由度を N とすると、

$$AIC = -2(\text{最大対数尤度}) + 2N \tag{97}$$

のように定義される。一方、MDL は Rissanen により符号化における記述長最小化 (Minimal Discription Length) 原理として導出されたもので、

$$MDL = -(\text{最大対数尤度}) + \frac{N}{2}\log P \tag{98}$$

のように定義される。これらの評価を用いると、学習データに対する当てはまりに大きな差があると第 1 項に大きな差があらわれ当てはまりの良いネットワークが選ばれ、第 1 項に大きな差が無い場合には第 2 項が作用して自由度の小さいネットワークが選択される。

従って、汎化能力の高いネットワークを設計するためには、予め中間層のユニット数の異なるネットワークの候補をいくつか用意し、各ネットワークの結合荷重を学習用のデータに対して十分に学習させ、そのパラメータを用いて式 (84) から対数尤度を計算し、AIC あるいは MDL の小さいネットワークを選択すればよい。

ただし一般のニューラルネットワークの学習では、学習の結果得られたパラメータは最尤推定量の近似値であるため AIC の導出における仮定 (主に漸近正規性) を満足できない可能性がある。そのため、情報量基準の改良に関する報告もいくつかなされており、例えば最尤推定量を仮定しないで情報量基準を導出し、クロスバリデーションによってペナルティ項を決定するもの (和田、川人 (1991)) がある。またブートストラップ法を用いて最尤法以外の推定でも評価可能な EIC(北川ら (1993)) もニューラルネットワークの学習においては有効であると考えられる。

4.3.2 VC 次元

AIC が尤度と期待平均尤度との差から導出されたのに対し、サンプルについての経験損失と真の分布についての期待損失の差を PAC 学習の枠組から評価して得られた VC(Vapnik-Chervonenkis) 次元を用いて評価する方法がある (Vapnik(1982))。

ここで仮にモデルの VC 次元が d であるとする。これから汎化性の評価として Vapnik の一様収束の理論からサンプルから得られる経験損失 I_{emp} と真の期待損失 I の推定の差が VC 次元 d によって書ける。

$$I < I_{emp} + \sqrt{\frac{d(\log 2m/d + 1) - \log \eta}{m}}$$

(η は PAC 学習における最大許容誤差、m はサンプル数)

この VC 次元により階層型ニューラルネットワークの汎化性の評価も行われている (Baum,Haussler(1989))

また Vapnik はモデルを構造化し、経験損失ではなく上式の右辺を最小化するような基準によってモデル選択を行う方法 (Structual Risk Minimization) を提唱している。

しかしモデルの VC 次元を厳密に求めることは特殊な場合を除いて実際にはそれほど容易ではないため、簡単な近似や推定値を用いることになる。また VC 次元は学習法やモデルによらない評価である一方、非常に粗い最悪評価になっているためより実際的にはよりタイトな評価である実効 VC 次元などの拡張が有効である。

4.3.3 Optimal Brain Surgeon

汎化性を確保するためには、課題の複雑さと利用可能なサンプルの数にマッチするようにネットワークの大きさを決めることが重要である。そこで、学習結果に応じてネットワークの大きさを変化させる方法もいくつか提案されている。

小さなネットワークからはじめて、徐々にネットワークを大きくするような方法としては、例えば、Fahlman らの Cascade Correlation[35]、the Group Method of Data Handinig (GMDH) に基づくもの [36, 37, 38] などがある。

一方、大きなネットワークから不要な (冗長な) 結合荷重を徐々に取り除いて、問題のサイズにマッチしたネットワークを構成する方法は、Pruning と呼ばれ、いくつかの方法が提案されている。そのための最も直観的な方法は、結合荷重の絶対値が小さい結合荷重を削除する方法である。しかし、結合荷重の絶対値が小さいからといって必ずしもネットワークの評価基準 (2 乗誤差) への貢献が小さいとは限らない。Cun らは、それを改善するために Hessian 行列の対角要素に基づいて結合荷重の貢献度を見積り、貢献度の小さい結合荷重を削除する Optimal Brain Damage (OBD) と呼ばれる方法を提案している [39]。さらに、Hassibi らは、以下のような Hessian 行列の全ての要素を考慮に入れて貢献度を見積もる方法を提案している [40]。

学習が収束した後のネットワークの結合荷重を $\boldsymbol{\theta}$ とし、それを $\delta\boldsymbol{\theta}$ だけ変化させた時、2 乗誤差がどれだけ増加するか考えてみよう。この時、学習は収束しているので Taylor 展開の 1 次の項は 0 となり、2 次以上の項のみが残り、2 乗誤差の増加量は、高次の項を無視すると、

$$\delta\varepsilon^2_{emp} = \varepsilon^2_{emp}(\boldsymbol{\theta}+\delta\boldsymbol{\theta}) - \varepsilon^2_{emp}(\boldsymbol{\theta}) \approx \frac{1}{2}\delta\boldsymbol{\theta}^T H \delta\boldsymbol{\theta} \tag{99}$$

のように近似できる。ただし、$H = \frac{\partial^2 \varepsilon^2_{emp}}{\partial \boldsymbol{\theta}^2}$ は、Hessian 行列である。

今、一つの結合荷重 θ_q を 0、つまり、

$$\boldsymbol{e}_q^T \delta\boldsymbol{\theta} + \theta_q = 0 \tag{100}$$

とした時、2 乗誤差の増加量を最小とする結合荷重の変化量 $\delta\boldsymbol{\theta}$ を求めてみよう。ここで、$\boldsymbol{e}_q$ は、q 番目要素のみ 1 で残りがすべて 0 のベクトルである。この問題は Lagrange 乗数を λ として、

$$Q = \frac{1}{2}\delta\boldsymbol{\theta}^T H \delta\boldsymbol{\theta} + \lambda(\boldsymbol{e}_q^T \delta\boldsymbol{\theta} + \theta_q) \tag{101}$$

を最小とすることと等価になり、その解は、

$$\delta\boldsymbol{\theta} = -\frac{\theta_q}{[H^{-1}]_{qq}} H^{-1}\boldsymbol{e}_q \tag{102}$$

となる。ただし、$[H^{-1}]_{qq} = \boldsymbol{e}_q^T H^{-1} \boldsymbol{e}_q$ は、H^{-1} の q 行 q 列の要素を表す。また、その時の 2 乗誤差の増加量は、

$$Q_q = \frac{1}{2}\frac{\theta_q^2}{[H^{-1}]_{qq}} \tag{103}$$

となる。これは、結合荷重 θ_q を 0 とした時、それ以外の結合荷重をうまく調整して達成できる 2 乗誤差の増加量の最小値である。従って、この値が小さくなるような、つまり、その結合荷重を削除しても 2 乗誤差の増加の少ない結合荷重を削除すればよいことになる。

この方法では、Hessian 行列の逆行列を計算する必要があるが、Hessian 行列を 4.1.5 節で示したフィッシャーの情報行列で代用して、計算を簡単化することも可能である。

こうした方法以外には、線形回帰分析の変数選択法や主成分分析を用いて中間層の不要なユニットを取り除いて小さなネットワークを構成する方法 [22] なども提案されている。

4.3.4 Weight Decay

汎化能力の高いネットワークを構成するために、正則化の考え方を取り入れて、不要な結合荷重を陽に取り除かないで、学習の評価基準に複雑すぎる解の探索を抑制するような項を追加する方法もいくつか提案されている。例えば、Hanson ら [21] は、不要な結合荷重が大きくなり過ぎないようなペナルティ項を加えることにより、学習が進むにつれて不要な結合荷重が 0 に近付くようにする Weight Decay 法と呼ばれる方法を提案している。具体的には、2 乗誤差基準にネットワークの結合荷重の 2 乗和の項を加えた評価基準

$$Q = \varepsilon_{emp}^2 + \frac{\lambda}{2}\left\{\sum_{i=0}^{I}\sum_{j=1}^{J} a_{ij}^2 + \sum_{j=0}^{J}\sum_{k=1}^{K} b_{jk}^2\right\} \tag{104}$$

を最小化するような結合荷重を求める。ここで、λ は、2 乗誤差に対するペナルティ項の影響を制御するパラメータである。この基準に基づく最急降下法による結合係数の更新式は、

$$a_{ij} \Leftarrow a_{ij} - \alpha\frac{\partial\varepsilon_{emp}^2}{\partial a_{ij}} - \alpha\lambda a_{ij} \tag{105}$$

$$b_{jk} \Leftarrow b_{jk} - \alpha\frac{\partial\varepsilon_{emp}^2}{\partial b_{jk}} - \alpha\lambda b_{jk} \tag{106}$$

となる。この第 3 項は、常に結合荷重の絶対値が小さくなる方向に働き、学習が進むに連れて不要な結合荷重が 0 に近付く。

別の形のペナルティ項を用いる同様な試みは、[41, 42, 43] などにも報告されている。また、情報量基準の漸近的近似としてペナルティ項を定義し実験的評価を行った例 (渡辺 (1993)) やベイズ推定の立場からペナルティ項の係数 λ の評価を行った例 (Mackey(1992)) などもある。

4.3.5 ノイズの付加による汎化能力の向上

最後に、学習用のデータやネットワークにノイズを付加することにより、学習結果の汎化性を向上させようとする試みについて紹介する。例えば、赤穂 (1992) では、正則化の観点から観測データを補間するような多数の学習用データを生成することにより汎化能力の高いットワークを構成する方法が提案されている。また、栗田ら (1993) は、中間層の各ユニットの入力 ζ_j に、平均 0 分散 σ の正規ノイズを付加した場合の誤差逆伝搬学習アルゴリズムの平均的な振舞について解析している。中間層の各ユニットにノイズを付加することによって、2 乗誤差は平均的に

$$\mathrm{E}\{\tilde{\varepsilon}_{emp}^2\} \approx \varepsilon_{emp}^2 + \frac{1}{2}\sum_{j=0}^{J}\sum_{k=1}^{K}(b_{jk}\nu_j\sigma)^2 \tag{107}$$

のように増加する。全体としてこの 2 乗誤差を小さくするためには、第 2 項も小さくする必要があり、b_{jk} と ν_j の絶対値をともに小さくする必要がある。ν_j を小さくするためには、中間層の各ユニットの出力 y_j が 0 か 1 に近い値をとる必要があるので、結局、中間層の各ユニットの出力は 0 か 1 の確定的な値をとり、中間層から出力層への結合荷重の絶対値は小さくなるように学習が進むと期待できる。これにより、ネットワークが構造化され、汎化能力も向上すると期待できる。Murray ら (1993) は、ネットワークの結合荷重にノイズを加える場合について、同様な考察を行っている。

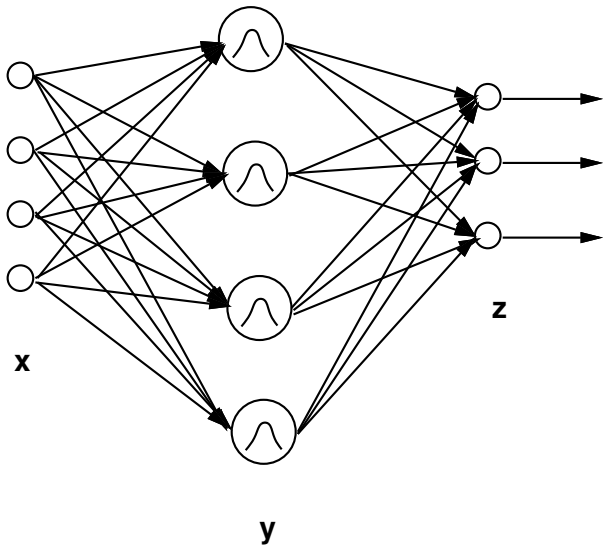


図 4: RBF ネットワークの例

4.4 RBF ネットワーク

最後に、Radial Basis Function (RBF) ネットワークについて簡単に紹介しておく。Radial Basis Function (RBF) ネットワークは図 4 に示すように、中間層の基底関数の出力を線形結合することによってネットワークの出力を計算するようなネットワークである [33]。中間層の基底関数としては、普通、

$$y_j = \exp\left[-\frac{(\boldsymbol{x}-\boldsymbol{a}_j)^T(\boldsymbol{x}-\boldsymbol{a}_j)}{2\sigma_j^2}\right] \tag{108}$$

のようなガウス関数が使われる。ここで、y_j は中間層の j 番目のユニットの出力である。また、$\boldsymbol{a}_j$ は j 番目のユニットのパラメータベクトル（ユニット j の中心）であり、σ_j^2 は j 番目のユニットの正規化パラメータである。従って、中間層の基底関数は入力がその中心に近い場合にのみ大きな出力を出す。出力層の k 番目のユニットの出力 z_k は

$$z_k = \boldsymbol{b}_k^T \boldsymbol{y} \tag{109}$$

のように中間層の出力の線形結合により計算される。ここで、$\boldsymbol{b}_k$ は中間層から出力層の k 番目のユニットへの結合荷重である。

RBF ネットワークは、多層パーセプトロンと同様に、中間層のユニット数が多ければ任意の連続関数を近似する能力を持つ。また、中間層の入出力関数がシグモイド関数である多層パーセプトロンでは中間層の出力が入力空間の無限に大きな領域で大きな値を持つが、RBF ネットワークでは入力空間の局所的な領域でのみ大きな値を持つ。なお、RBF ネットワークは、3.2.2 節の核関数に基づく確率密度関数の推定方法と密接に関連している。

4.4.1 RBF ネットワークでの学習

RBF ネットワークのパラメータを学習するためには、いくつかのアプローチがある。良く使われるのは、まず、中間層での学習と出力層での学習を別々に行い、その結果を初期値として全体のパラメータを微調整する方法である。初期値を決めるための中間層での学習にはクラスタリング手法（unsupervised な手法）が用いられる。クラスタリング手法としては、K-means 法を用いることが多い。クラスタリングの結果から、各クラスターにひとつのユニットが割り当て、クラスターの中心をそのユニットの中心とし、正規化パラメータはクラスターの広がり具合から推定する。出力層での学習は、最小２乗法によって陽に求めたり、あるいは、最急降下法によって２乗誤差を最小とするよう求めたりする。

参考文献

[1] C.M.Bishop, *Neural Networks for Pattern Recognition*, Oxford Univ. Press, 1995.

[2] B.D.Ripley, *Pattern Recognition and Neural Networks*, Cambridge Univ. Press, 1996.

[3] C.K.Chow, “An optimum character recognition system using decision functions,” IRE Trans., Vol.EC-6, pp.247–254, 1957.

[4] 麻生 (1988):“ニューラルネットワーク情報処理,” 産業図書, .

[5] Minsky,M. and Papert,S.(1969): “*Perceptrons*,” MIT Press, .

[6] Rumelhart,D.E.,Hinton, G.E. and Williams,R.J.(1986): “Learning representaions by back-propagating errors,” Nature, Vol.323-9, pp.533-536.

[7] Rumelhart,D.E., Hinton,G.E., and Williams,R.J.,(1986): “Learning internal represenations by error propagation,” in *Parallel Distributed Processing* Volume 1, J.L.McCleland, D.E.Rumelhard, and The PDP Ressearch group, MIT Press.

[8] Hopfield,J.J.,(1982): “Neural networks and phisical systems with emargent collective computational abilities,” Proc. of the National Academy of Sciences USA 79, pp.2254-2258.

[9] Hopfield,J.J.(1984): “Neurons with graded responce have collective computational properties like those of two-state neurons,” Proc. of the National Academy of Sciences USA 81, pp.3088-3092.

[10] Farlman,S.E.,Hinton,G.E., and Sejnowski,T.J.,(1983): “Massively parallel architectures for AI: NETL, Thistle, and Boltzman Machines,” Proc. of the National Conf. on Artificial Intelligence AAAI-83, pp.109-113.

[11] Hinton,G.E., Sejnowski,T.J. and Ackley,D.H.,(1984): “Boltzmann Machines : Constraint satisfaction networks that learn,” Tech. Rep. CMU-CS-84-119.

[12] Hinton,G.E. and Sejnowski,T.J.,(1986): “Learning and relearning in Boltzmann Machines,” in *Parallel Distributed Preocessing* Volume 1, J.L.McCleland, D.E.Rumelhard, and The PDP Ressearch group, MIT Press.

[13] Hush,D.R. and Horne,B.G.,(1993): “Progress in supervised neural networks,” IEEE Signal Processing Magazine, pp.8-39.

[14] Otsu,N.(1982): “Optimal linear and nonlinear solutions for least-square discriminant feature extraction,” Proc. 6th ICPR, pp.557-560.

[15] Otsu,N. Kurita,T. and Asoh,H.(1988): “A unified study of multivariate analysis methods by nonlinear extensions and underlying probabilistic structures,” in *Recent Developments of Clustering and Data Analysis*, E.Diday *et al.* eds., Academic Press.

[16] Cybenko,G.,(1989): “Approximation by superpositions of a sigmoidal function,” Mathematics of Control, Signals, and Systems, Vol.2, No.4, pp.303-314.

[17] 栗田,(1991): “階層型ニューラルネットのパラメータの最尤推定について,” 電子情報通信学会, ニューラルコンピューティング研究会資料, NC91-36.

[18] Kurita,T.(1992): “Iterative weighted least squares algorithms for neural networks classifiers,” Proc. of the Third Workshop on Algorithmic Learning Theory, Tokyo, Oct. 20-22.

[19] Baum,E.B. and Haussle,D.,(1989): “What size net gives valid generalization ?,” Neural Computation, Vol.1, pp.151-160.

[20] Akaho,S.,(1992): “Regularization learning of neural networks for generalization,” Proc. of the Third Workshop on Algorithmic Learning Theory, Tokyo, Oct. 20-22.

[21] Hanson,S.J. and Pratt,L.Y.,(1989): “ Comparing biases for minimal network construction with back-propagation,” In D.S.Touretzky, ed. *Advances in Neural Information Processing Systems 1*, pp.177-185, Morgan kaufmann.

[22] Sekita,I., Kurita,T., Asoh,H., and Chiu,D.(1993): “Reconfigurating feedforward networks with fewer hidden nodes,” Proc. of SPIE conf. on Adaptive and Learning Systems II. (to be appear).

[23] Miller,R.G.,(1974): “The jacknife -a review,” Biometrika, Vol.61, No.1, pp.1-15.

[24] Stone,M.,(1974): “Cross-validatory choice and assessment of statistic al predictions,” Journal of Royal Statistical Society, Vol.B36, pp.111-147.

[25] Efron,B.,(1979): “Bootstrap methods: anothoe look at the jackknife,” The Annals of Statistics, Vol.7, No.1, pp.1-26.

[26] Efron,B.,(1983): “Estimating the error rate of a prediction rule: imp rovements in cross-validation,” Journal of Amerian Statistical Association, Vol .78, pp.316-331.

[27] Efron,B.,(1985): “The bootstrap method for assessing statistical accu racy,” Behaviormetrika, Vol.17, pp.1-35.

[28] Akaike,H.,(1974): “A new look at the statistical model identification,” IEEE Trans. on Automatic Control, vol.AC-19, No.6, pp.716-723.

[29] 坂本, 石黒, 北川,(1983): “情報量統計学,” 共立出版.

[30] Rissanen,J.,(1983): “A universal prior for integers and estimation by minimum description length,” The Annals of Statistics, Vol.11, NO.2, pp.416-431.

[31] Rissanen,J.,(1986): “Stochastic complexity and modeling,” The Annals of Statistics, Vol.14, No.3, pp.1080-1100.

[32] 栗田,(1990): “情報量基準による３層ニューラルネットの隠れ層のユニット数の決定法,” 電子情報通信学会論文誌, Vol.J73-D-II, No.11, pp.1872-1878.

[33] Possio,T. and Girosi,F.,(1990): “Networks for approximation and learning,” Proc. of the IEEE, Vol.78, No.9, pp.1481-1497.

[34] Fahlman,S.E.,(1988): “An empirical study of learning speed in back-propagation networks,” Tech. Report, CMU-CS-88-162.

[35] Fahlman,S.E. and Lebiere,C.,(1990): “The cascade-correlation learning architecture,” in D.Touretzky, editor, Advances in Neural Information Precessing Systems 2, pp.524-532, Morgan Kaufmann.

[36] Ivakhnenko,A.C.,(1971): “Polynomial theory of complex systems,” *IEEE Trans. on Systems, Man, and Cybernetics*, Vol.SMC-1, No.4, pp.364-378.

[37] Tenorio,M.F. and Lee,G.,(1990): “Self-organizeing Network for optimum supervised learning,” *IEEE Trans. on Neural Networks*, Vol.1, No.1.

[38] Iba,H., Kurita,T., deGaris,H. and Sato,T.,(1993): “System identification using structured genetic algorithms,” *Proc. of 5h Inter. Joint Conf. on Genetic Algorithms.*

[39] Cun,Y.L., Denker,J.S. and Solla,S.A.,(1990): “Optimal brain damage,” in D.Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pp.598-605.

[40] Hassibi,B. and Stork,D.G.,(1993): “Second order derivatives for network pruning: Optimal brain surgeon,” S.J.Handon, J.D.Cowan, and C.L.Giles, editers, *Advances in Neural Information Processing Systems 5*, Morgan Kaufman, pp.164-171.

[41] Ishikawa,M.,(1989): “A structural learning algorithm with forgetting of link weights,” *Porc. Inter. Joint. Conf. on Neural Networks.*

[42] 石川,(1990): “忘却を用いたコネクショニストモデルの構造学習アルゴリズム,” 人工知能学会誌, Vol.5, No.5, pp.595-603.

[43] Nowlan,S.J. and Hinton,G.E.,(1992): “Simplifying neural networks by soft weight-sharing,” *Neural Computation*, Vol.4, No.4, pp.473-493.

[44] 栗田, 麻生, 梅山, 赤穂, 細美, (1993):“独立なノイズの付加による多層パーセプトロンの構造化学習,” 電子情報通信学会技術報告, NC93-21, pp.71-77.

[45] Murray,A.F. and Edwards,P.J.,(1993): “Synaptic weight noise during multilayer perceptron training: falst tolerance and trainig improvements,” *IEEE Trans. on Neural Networks*, Vol.4, NO.4, pp.722-725.

[46] Asoh,H. and Otsu,N.,(1989): “Nonlinear data analysis and multilayer perceptrons,” *Proc. of Inter. Joint Conf. on Neural Networks*,” Vol.II, pp.411-415.

[47] Asoh,H. and Otsu,N.,(1990): “An approximation of nonlinear discriminant analysis by multilayer neural networks,” *Proc. of Inter. Joint Conf. on Neural Network*, Vol.III, pp.211-216.

[48] Webb,A.R. and Lowe,D.,(1990): “The optimised internal representaion of multilayer classifier networks performs nonlinear discriminant analysis,” *Neural Networks*, Vol.3, pp.367-375.

[49] Lowe,D. and Webb,A.R.,(1991): “Optimized feature extraction and the Bayes decision in feed-forward classifier networks,” *IEEE Trans. on Pattern Analsysis and Machine Intelligence*, VOl.PAMI-13, NO.4, pp.355-364.

[50] Vapnik,V.,(1982):”Estimation of Dipendences Based on Empirical Data” , Springer-Verlag.

[51] Vapnik,V.,(1992):”Principles of Risk Minimization for Learning Theory”,Advanced in Neural Information Processing System 4,Morgan Kaufmann, pp.831-839.

[52] Jordan,M.,Jacobs,R.,(1993): ”Hierarchical mixtures of experts and the EM algorithm,” Proc. of IJCNN’93 NAGOYA.

[53] Mackey,D.J.C.,(1992) “A Practical Bayesian Framework for Backpropagation Networks,” Neural Computation,4,pp.448-472.

[54] 甘利 他 ,(1993): “ニューラルネットの新展開,” サイエンス社.

[55] 麻生,(1992): “期待損失最小化学習のための基準の比較,” 電総研研究速報

[56] Dempster,A., Laird,N. and Rubin,D., (1977): “Maximum likelihood from incomple data via the EM algorithm,” J. Roy. Statist. Soc. B, Vol.39, pp.1-38.

[57] Amari,S., (1994): “Information geometory of the EM and em algorithms for neural networks,” Technical Report METR 94-04, University of Tokyo. (xxxxxx to appear in Neural Networks **********)