

第2章 帰納的機械学習手法の天気予報への応用

2.1 はじめに

今日の天気予報において数値予報が重要な位置を占めていることはいうまでもない。しかし、特に目先の24時間程度の子測を考へるとき、予測を行う時点での実況というものも重要な資料であり、この実況と数値予報がずれていた場合は、予報を組み立てる上で、実況とモデル結果との間にどのような折り返しを付けるかということが予報者の重要な役割となってくる。

従来、このような作業は予報者の経験と知識に基づいてなされてきた。今日でも基本的にはその在り方は変わっていないが、時間的・空間的により細かな予報が求められるにつれて、この作業を人間だけの力で行うのは物理的に困難になってきている。加えて、「より細かな予報」というのは熟練した予報者にとっても「未経験の分野」である場合があり、十分な経験が蓄積されるまでには時間がかかるといった状況もある。

予報者が「経験を積む」というのは、「予測のための資料」と「実際に現れた天気」との組み合わせを多数の事例について体験し、「資料」と「天気」との関係についての論理的なつながりを見出していくことにほかならない（「論理的」といっても、常に気象学的あるいは物理学的な説明ができるわけではないが、その場合でも「経験則」という形で法則化することは可能である）。従って、この論理的なつながりを発見するための機械的な方法があれば、「経験を積む」という行為を自動化することができる可能性がある。

人工知能の分野で研究されている「機械学習」は、人間の学習行為をコンピュータによって再現しようとするもので、その手法も対象も極めて多岐にわたっている。その中には、上に述べたような、多数の事例からの帰納的な学習により問題解決を図る方法も含まれており、それらの手法によって「経験則」の発見とその経験則に基づく処理（たとえば天気の子測）の自動化（および高速化）を図ることが可能になると期待されている。

本章では、それらの帰納的機械学習手法を天気予報に適用した例について報告する。これらの技術を天気予報に応用するにあたっては、応用事例の報告がまだまだ少ないこともあり、それぞれの技術について「どのように使うのか」「どの程度使えるのか」といった点から調べて、どの技術が使いやすいのかを見極めることが、第一の課題である。そのため、本章では、各手法の概要と応用例を並列的に記述している。とりあげた手法は、単純ベイズ法・ニューラルネットワーク・ID3・エントロピーネットである。予測対象は特定の地点または地域での降水の有無で、予測時間は24時間以内を想定している。予測に用いる資料は、L-ADESSによって配信されている数値モデルGPVと実況観測値である。本報告の研究は1991年度～1995年度の間に行われたものであるため、数値モデルの資料としては、ASMおよびJSMのGPVが用いられている。

本報告の対象となる研究期間(1991-95年度)の後に、ニューラルネットワークの応用についていくつかの進展が見られた。これらについては、その概略を付録B「ニューラルネットワーク応用の最近の進展(1996年以降)」に記した。

2.2 単純ベイズ法

2.2.1 手法の概要

おおよそ我々の経験的知識は「AならばB」という因果関係で記述されることが多い。これをもう少し正確に(確率論的に)表現すれば、何もデータが無いときのBの確率より、Aという情報が与えられた時のBの条件付き確率の方が大きい、ということである。さて、ここに「AならばB」と「XならばB」という2つの知識があった場合、「AとXがともにある」「AはあるがXはない」「AがなくてXがある」といったそれぞれのケースについて、Bの確率がどのように振る舞うかを判断するのは、直観的には難しい。これらを「Bの条件付き確率」という形で統一的に表現し、条件が2つ

以上組み合わせた時に、その確率がどのように振る舞うかを客観的に表わすのがベイズの法則である。本節でとりあげる単純ベイズ法とは、予測対象の生起確率（予測資料の元での条件付き確率）をベイズの法則によって求める手法である（安西,1989を参照）。

予測のための資料を a 、予想対象を x とし、ある事象 s の起こる確率を $p(s)$ で表すことにすると、何も資料が無いときは x の起こる確率は $p(x)$ と書ける（いわば気候学的な確率である）。ここで資料 a が与えられると x の起こる確率は条件付き確率となり、 $p(x|a)$ と表される。経験を積むというのは、つきつめれば、過去の事例の生起確率に基づいて a が与えられたときの条件付き確率 $p(x|a)$ を推定することである。たとえば、 x が「雨が降る」という現象、 a が「低気圧が近くにある」という資料だとすると、低気圧と雨の関係についての知識がある人（経験を積んでいる人）は、 $p(x|a) < p(x)$ である（すなわち、低気圧が近づくと雨の降る確率は高くなる）ということがすぐにわかるはずである。一般に、予測を行うということは、与えられた資料のもとでの予測対象の条件付き確率を求めることである。

さて、ここで $p(x)/p(\bar{x})$ という変数を考える（ここで $p(\bar{x})$ は x が起こらない確率で $1-p(x)$ に等しい）。この変数は資料 a が存在する場合は条件付き確率の比の形になり、 $p(x|a)/p(\bar{x}|a)$ と書ける。この変数はベイズの法則を用いて次のように変形することができる。

$$\frac{p(x|a)}{p(\bar{x}|a)} = \frac{p(a|x)}{p(a|\bar{x})} \times \frac{p(x)}{p(\bar{x})}$$

つまり、求める条件付き確率の比は、元の確率の比と $p(a|x)/p(a|\bar{x})$ との積で表すことができ、従って $p(a|x)/p(a|\bar{x})$ の値が1より大きければ、「 a が存在すると x の確率は高くなる」という事がわかり、逆に1より小さければ、「 a が存在するとき x の確率は低くなる」ことがわかる。

資料が複数（例えば a 、 b の2つ）ある場合は、それらの資料が x, \bar{x} の下で互いに独立である場合に限り次のように表せる。

$$\frac{p(x|a,b)}{p(\bar{x}|a,b)} = \frac{p(a|x)}{p(a|\bar{x})} \times \frac{p(b|x)}{p(b|\bar{x})} \times \frac{p(x)}{p(\bar{x})}$$

つまり個々の資料について $p(a|x)/p(a|\bar{x})$ 、 $p(b|x)/p(b|\bar{x})$ などという値を計算しておけばそれらの積と $p(x)/p(\bar{x})$ を掛け合わせた値を使って、与えられた資料の下での x の確率を求めることができる。もし確率値が不要で、与えられた資料から x が気候値に比べて「起こり易い」か「起こりにくい」かを判断するだけでよい場合は、個々の資料の $p(a|x)/p(a|\bar{x})$ 、 $p(b|x)/p(b|\bar{x})$ などの値を掛け合わせたものが1より大きいかどうかを調べればよい。また資料の一部が入手できなかった場合は、それについては $p(a|x)/p(a|\bar{x})$ を1とすれば他に影響を及ぼすことなく取り扱えるし、資料 a が現れていないことが明らかな場合には（言い換えれば資料 \bar{a} が現れている場合には）、 $p(a|x)/p(a|\bar{x})$ の代わりに $p(\bar{a}|x)/p(\bar{a}|\bar{x})$ を使えばよい。

ここで用いる確率の値は、事例数を数えることによって推定する。ある期間における事象 a の事例数を $n(a)$ とし、 a と b が同時に現れる事例の数は $n(a,b)$ のように表すことにすると、たとえば $p(a|x)$ は次のように推定される。

$$p(a|x) = \frac{n(a,x)}{n(x)}$$

この手法の場合、「経験を積む」ということは貯えられる事例数が増える（従って、推定される確率値の信頼性が増す）ということと同じである。またここでは、事例の計数は新しい事例が与えられるたびに行うことができるので、この手法は逐次学習の一手法であるといえる。

この手法は、複数の資料が x, \bar{x} の下で互いに独立である、という仮定をおいて問題を単純化していることから「単純ベイズ(Naive Bayes)法」と呼ばれており、現実の問題への適用に際しては、独立性の仮定が満たされていない（現実には、これが満たされることは、ほとんどない）ということが、しばしば問題にされる。しかし、実際には、手法

の前提となる独立性の仮定が満たされていない場合、多くの場合に有効な結果をもたらすことが知られており、その理由についての調査も行われている (Domingos and Pazzani,1997)。

2.2.2 降水の有無の予測に適用した場合

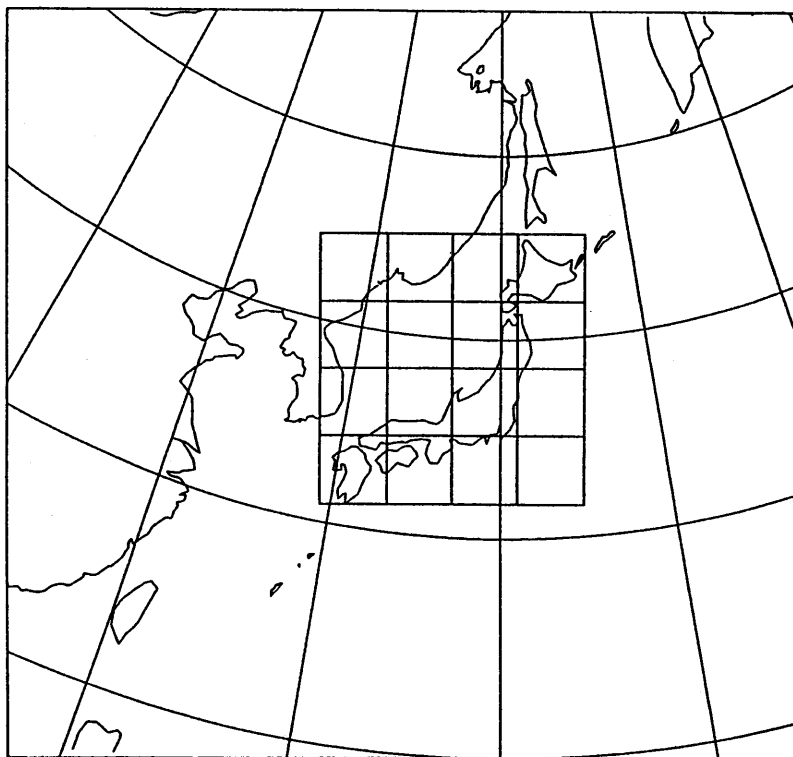
L-ADESSのデータを用いて単純ベイズ法のテストを行った。予想の対象 x としては「ある特定の1時間のうちに東京(大手町)のアメダスポイントで1mm以上の降水があること」とした。資料はL-ADESSで配信されるASM広域・JSM狭域のGPVをもとに、第1表に掲げたものを用いることにした。この方法では離散的な量を扱う方が簡単なので、低気圧等の位置についてはASM広域GPVの配信領域の中に第1図のようなマス目(1マスにASM広域の格子点が4つ入る)を作り、どのマスに入ったかで表現することにした。また、風向は8方位とし、その他の量は適当なしきい値で離散化した。

数値予報の初期時刻から0,6,12,18,24時間後の予想値を資料とし、valid timeの前1時間に東京で1mm以上の降水があったかどうかを調べて、事例数をカウントした。予想時間による区別は行っていない。具体的にはプログラムは以下のような手順で動作する。

1. 1日2回、9時35分と21時35分(日本時)に自動起動し、それぞれ前日の00z, 12zを初期値とする数値予報GPVを読む。
2. 数値予報の初期時刻から0,6,12,18,24時間後の各々について以下の事を行う。まず、数値予報GPVとその時点での事例数に基づく条件付き確率の推定値から予想を行って結果を出力する。その後アメダスデータを読んで事例数を更新する。

第1表 単純ベイズ法で入力として用いた資料

- | |
|----------------------|
| 1. 低気圧の位置 |
| 2. 850mbの卓越風向 |
| 3. 850mbの温度集中帯の位置と強さ |
| 4. 850mbの温度集中帯の走向 |
| 5. 850mbのリッジの傾き |
| 6. 700mb上昇流の位置と強さ |
| 7. 500mbの正渦の位置と強さ |
| 8. 500mbのリッジの傾き |
| 9. 東京付近の900mbの風向 |
| 10. 東京付近の850mbの風向 |
| 11. 東京付近の700mbの風向 |
| 12. 東京付近の500mbの風向 |
| 13. 東京付近の900mbのT-Td |
| 14. 東京付近の850mbのT-Td |
| 15. 東京付近の700mbのT-Td |
| 16. 東京付近の500mbのT-Td |
| 17. 東京付近の900mbの温度移流 |
| 18. 東京付近の850mbの温度移流 |
| 19. 東京付近の700mbの温度移流 |
| 20. 東京付近の500mbの温度移流 |
| 21. 東京付近のRsfc |



第1図 単純ベイズ法において、低気圧等の位置を離散化するためのマス目

第2図はこのシステムを1991年9月中旬から動かし、1ヶ月ごとに計算したスキルスコアの時系列である。予想については実際のシステムの出力は確率の形だが、ここでは前に述べた $p(a|x)/p(a|\bar{x})$ の積が1より大きければ降水あり、1以下なら降水無しとして、2x2の分割表を作ってスコアを計算した。

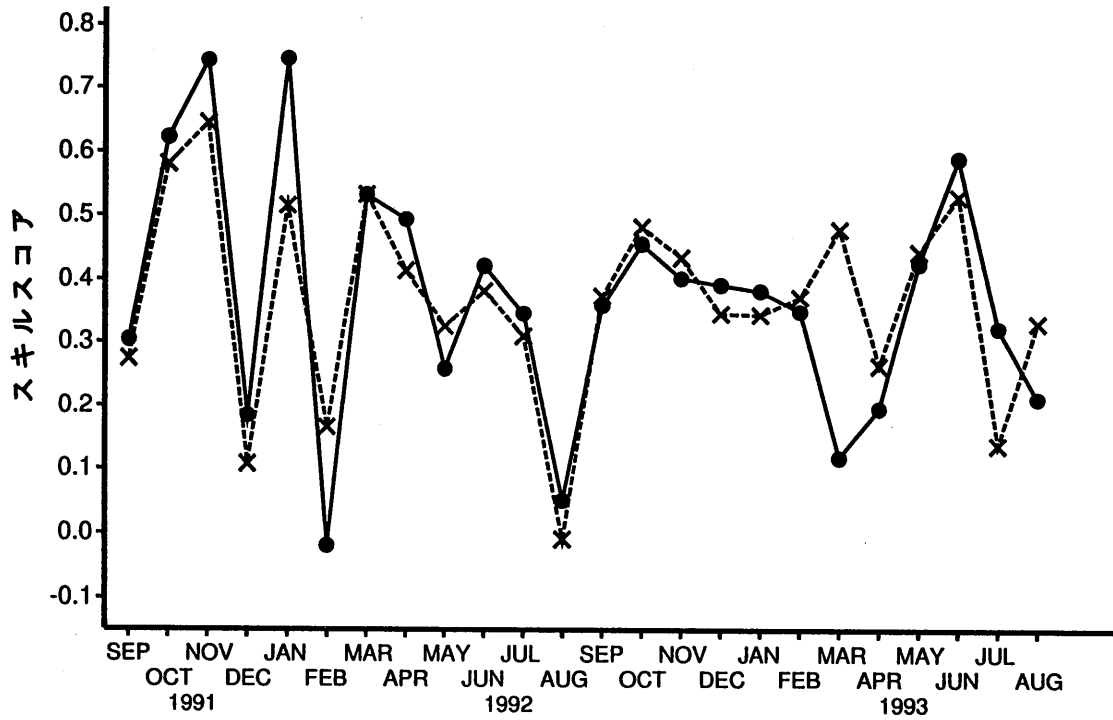
事例数を数えるだけの極めて単純な方法であるにもかかわらず、短期間の中に十分なスコアが得られている。試みに、10日ごとに蓄積した事例数を消去して、学習を再スタートさせた場合のスコアの時系列が第2図の破線である。このようにしても、得られるスコアの値はほとんど変わらず、この手法は極めて短期間のデータで予測が可能になることを示している。

本来、このシステムは確率を与えるものであるから、その値が確率値として適当であるか、ということも重要である。第3図は、横軸にシステムが与えた予測確率値、縦軸に実際の降水の出現頻度をプロットしたものである。一見してわかる通り、予測値は実際の出現頻度よりも高い値に偏っていることがわかる。これは、手法の前提となる資料相互の独立性が満たされていないことから来る、確率の過大評価と考えられる。ちなみに、第3図の破線は、10日ごとに蓄積した事例を消去した場合のプロットである。こちらの方が偏りが少なく、長期間の学習によって個々の確率が安定して推定されることが、かえって確率の過大評価に結び付いているようにみえる。

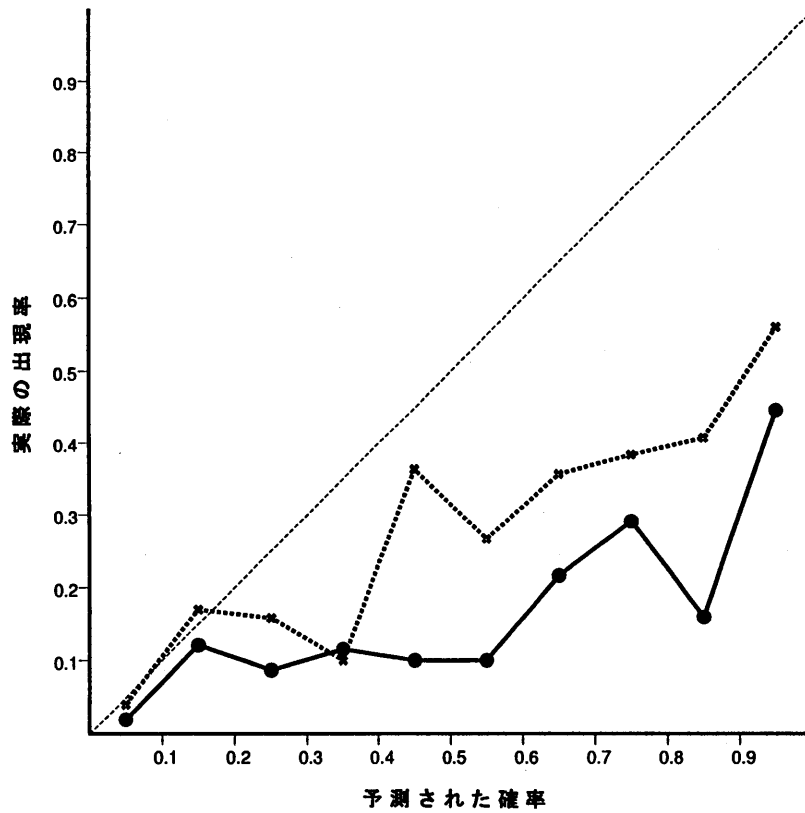
この事からすると、予測を行う時点に近い数日~数十日のデータのみを使うという方法が良いのかもしれない。これは、現象の出現確率が、ここで用いられているデータだけでは表現できないような背景の場の影響を受けている、ということかもしれない。ただし、いずれにせよ、10日ごとに学習を0からやり直すというのは、実用上は問題があるので、なんらかの方法で学習の履歴を監視し、古いものから消去する、あるいは、学習の際に最近の事例に重みのかかった学習を行なう、という方法を考案する必要がある。

2.2.3 考察と将来の展望

単純ベイズ法は、予測対象の条件付き生起確率を直接推定するものであり、予測資料相互の独立性以外には仮定が



第2図 単純ベイズ法のスキルスコア(実線)。1991年9月に学習を開始し、1ヶ月単位でスコアを計算した。破線は10日ごとにそれまで学習したことを消去した場合のスコア



第3図 単純ベイズ法による降水確率(横軸)と実際の降水の出現率(縦軸)との関係(実線)。破線は10日ごとにそれまで学習したことを消去した場合

必要でないため、極めて堅牢な手法である。また、予測に有効である可能性のある資料を用意して事例数を数えるだけなので、システム構築も極めて簡単であるし、予測対象にたいして情報を持たない資料については $p(a|x)/p(a|\bar{x})$ が 1 に近づくことで自動的に排除されるので、重回帰法のように、予測変数の数が手法の性能に影響を与えることもない。

また、ある特定の予測対象に対して、定性的な経験則しか持ち合わせていないような場合でも、事例をある程度蓄積するだけで確率予報を行うことができるという点は、極めて有用である。

ただし、この方法には幾つかの弱点がある。その一つは、予測資料を離散化する必要があるという点である。離散化は、細かく行うほど多様な事例についてきめ細かな予測が可能になる反面、一つのカテゴリーに入る事例数が減少して確率の推定精度が下がるという問題が生ずる。また、確率の推定のためにはカテゴリーごとに事例数を保存しておく必要があるため、細かい離散化を行うとそれだけ多くの計算機資源を消費することにもなる。

もう一つの弱点は、この手法が仮定している独立性が、現実にはほとんど満たされていないという点にある。このため、得られた生起確率が、0.5 を越える場合には過大に、0.5 を下回る場合には過小になる傾向があり、確率の値そのものとしてはやや信頼性に欠けるきらいがある。

ただ、この点については、近年 Friedman et al. (1997) によって提案された TAN (Tree Augmented Naive Bayes) を用いることにより、互いに独立でない予測資料を用いた場合でも、より適切な (かつ精度の高い) 予測が可能になる見込みが出てきた。TAN についても天気予報への応用を試みる必要がある。

単純ベイズ法は、現在のところ、天気予報の現場では全く利用されていない。実用上最大の難点は、天気予報に用いられる資料のほとんどが連続量であるために、離散化しなければ手法を適用することができない、という点にある。データの条件付き確率分布を、ガウス分布で近似するなど、連続量をそのまま扱うための改良を試みる余地はあると思われる。

2.3 ニューラルネットワーク

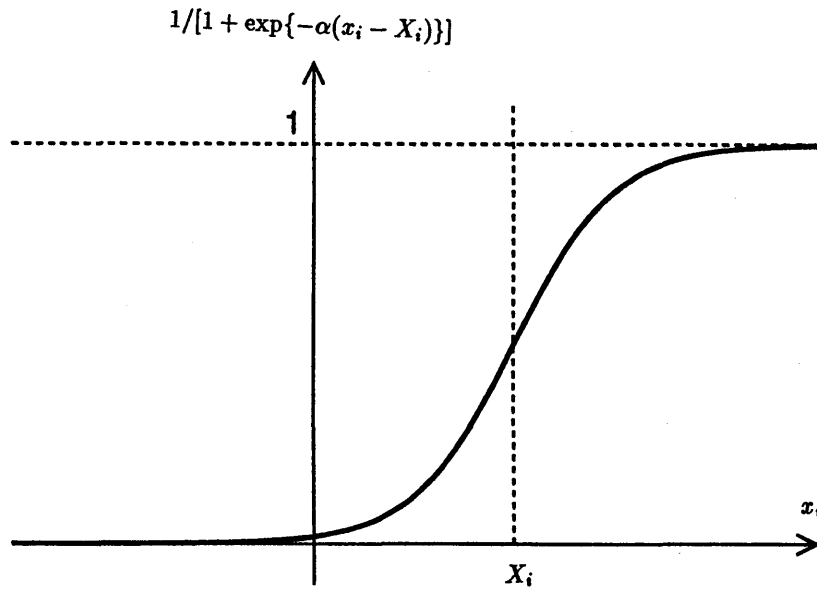
2.3.1 手法の概要

ニューラルネットワークは、もともとは脳の中の信号処理過程のモデルとして発想されたものである。しかし、ニューラルネットワーク研究の中から生み出されたある種のネットワークモデルは、未知の関数をきわめて柔軟に近似できる非線型のパラメトリック関数としての性質を持っていたために、脳神経学的な興味とは別に、様々な分野に応用されるようになった。

天気への予測への応用という観点から見たニューラルネットワークの最大の魅力は、入出力ともに連続量を扱える、という点にある。気象データの多くが連続量で表わされ、予測対象もまた、本質的には連続量であることを考えると、このことは実用上大きな意味を持っている (ニューラルネットワークの実用的な側面については、久間・中山,1992, 小泉,1997a等を参照。また、天気への予測への応用については、柳野,1998abも参考になる)。

本節では、最も広く使われている 3 層フィードフォワード型 (以下、FF型と略記) のニューラルネットワークを用い、バックプロパゲーション (以下、BPと略記) アルゴリズムによって学習を行うモデルについて記述する。

ここに n_I 個の要素からなる入力ベクトルの集合 I と n_O 個の要素からなる出力ベクトルの集合 O があり、 I に属するベクトル x にはそれぞれ対応する出力ベクトル y が一つ存在するものとする。 x と y との間一般的な関係が不明であるとき、その関係のある関数 ϕ で近似することを考える (すなわち、 $y = \phi(x)$ となるような ϕ を求める)。例えば、比較的簡単に得られる ϕ として、 n_O 個の重回帰式の集合を考えることができる。このとき個々の重回帰式は $n_I + 1$ 個のパラメータを持つので、 ϕ は $n_O(n_I + 1)$ 個のパラメータによって決まることになる。この場合はパラメータの決定は最小二乗法により、行列計算によって行われるのが普通である。



第4図 シグモイド関数の形

重回帰法は線形近似であるが、 x と y が複雑な対応関係を示す場合は非線形関数による近似が望ましいと思われることも多い。近年、FF型のニューラルネットワークが、適応性の高い非線形関数として広く利用されるようになってきた。

FF型ニューロでは次のような操作によって入力-出力関係が定義されている。まず、 n_I 個の入力変数の線形結合によって中間変数 u を作る ($u_j = w_{0j} + \sum_i w_{ij}x_i$)。ここで x_i は入力ベクトル x の i 番目の要素、 w はパラメタである。 u は任意の個数定義できるので、ここでは個数を n_M 個とし、 j 番目の変数を u_j と表すこととする。

次に、 u_j を非線形の応答関数 f によって v_j に変換する ($v_j = f(u_j)$)。 f は通常シグモイド関数を持ちいる。本研究でも $f(x) = 1 / (1 + \exp(-x))$ とした ($f(x)$ の形を第4図に示す)。

最後に v_j の線形結合により出力変数 o を作る ($o_k = W_{0k} + \sum_j W_{jk}v_j$)。ただし W はパラメタで、 o_k は n_O 個の出力変数の k 番目のものを表す。この o を f で変換した値 ($f(o)$) を出力とする場合もある (本節では $f(o)$ を出力としている。以下の文章では $f(o)$ を o と書く)。

上のように定義された ϕ は w と W というパラメタによって決まる非線形関数の組である。この形の関数は、**パラメタを適切に与えれば**、任意の連続関数を表すことができることが知られている。ただ、そのための**適切なパラメタ**を求める方法は知られておらず、実用的には、関数が与える出力値と望ましい出力値との差を小さくするように少しずつパラメタを変えていく方法が採られている。この逐次的なパラメタ調節の方法の一つがBP法であり、手順は以下の通りである。

1. ϕ の出力 o と真の値 y とのずれを評価する関数 E を定義する。本研究では $E = \sum_s \sum_k (o_k - y_k)^2 / 2$ とした。ただし o_k は o の k 番目の要素、 \sum_s は与えられた x と y の組合せ全体についての総和、 \sum_k は k についての総和である。
2. E を各パラメタで微分する。 E を w_{ij} で微分した値を α_{ij} 、 W_{ij} で微分した値を β_{ij} とする。
3. w_{ij} を $w_{ij} - \eta\alpha_{ij}$ で、 W_{ij} を $W_{ij} - \eta\beta_{ij}$ で置き換える。ただし η は任意の (通常は微少な) 正定数である。
4. このように修正された ϕ を用いて出力 o を再計算し、はじめに戻る。 o と y の差が充分小さくなったら調節終了とする。

2.3.2 降水分布予測への応用

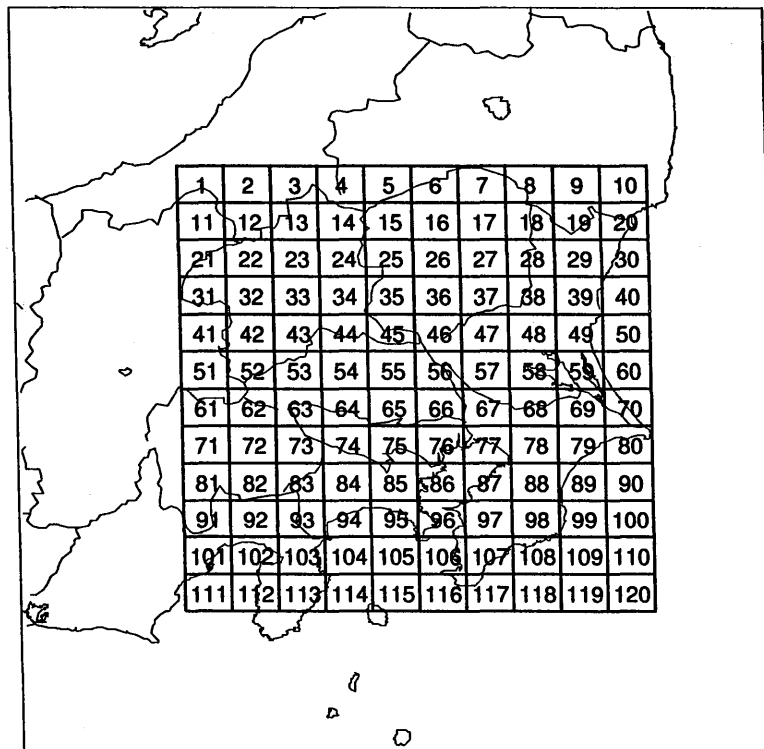
2.3.2.1 ニューラルネットワークに用いるデータおよび学習手順

上に述べたアルゴリズムを用いて、降水の有無の分布予測を試みた。ここで用いたニューラルネットワークは200個の中間層ニューロンと120個の出力層ニューロンを持ち、第5図に示したメッシュ一つ一つにおける降水の有無の予測を0から1の間の数値で出力するようになっている（降水なしが0，降水ありが1）。メッシュの大きさは約20km×20kmである。実際の降水の有無は、3時間積算したレーダーアメダス解析雨量が最低レベルの雨量以上となったかどうかで決め、ニューラルネットワークに与える真値としては、各メッシュに占める降水領域の割合を用いた。

ニューラルネットワークへの入力値はL-ADESSで配信される数値モデルの結果及び各種実況観測値である。ニューラルネットワークへの入力値の種類を第2表に示した。値を取得する格子点の位置は、第6図に示した通りである。実況観測値からは、格子点に内挿したアメダスの風の東西成分・南北成分および収束値、格子点に内挿したアメダスの気温とJSMの850hPaの気温との差、レーダーアメダス解析雨量値、降水短時間予報による予想雨量値、レーダー合成図によるエコー強度値、SDUSのLR-FAX画像による静止気象衛星ひまわりの赤外輝度温度値を用いた。

エコー強度値については第7図に示した領域ごとにエコー強度の階級値(0-15)を平均して入力値とした。エコー強度を平均する領域の広さは降水域のおおまかな動きを捉えるために50km四方程度に設定している。また、遠い西方のエコーほど予測対象領域の降水との関係が曖昧になっていくので、領域を広くしてある。

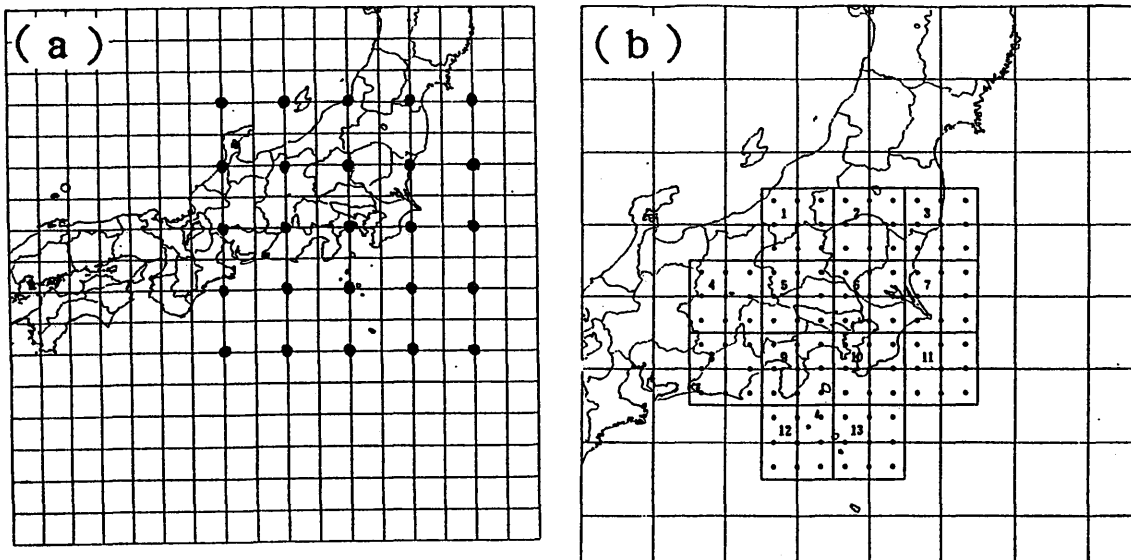
輝度温度値については第8図に示した各領域で階級値(0-63)を平均して入力値としている。SDUSの画像データは、雲の画像に緯経線や海岸線が重ねられた状態で配信されてくるが、これらの緯経線や海岸線のデータは一定の値でなく、各点で雲とのコントラストがはっきり付くような値になっており、取り除くことが非常に困難である。このため、平均をとる領域を設定するにあたっては緯経線や海岸線をなるべく避けることが必要となり、第8図のような不規則な配列となった。



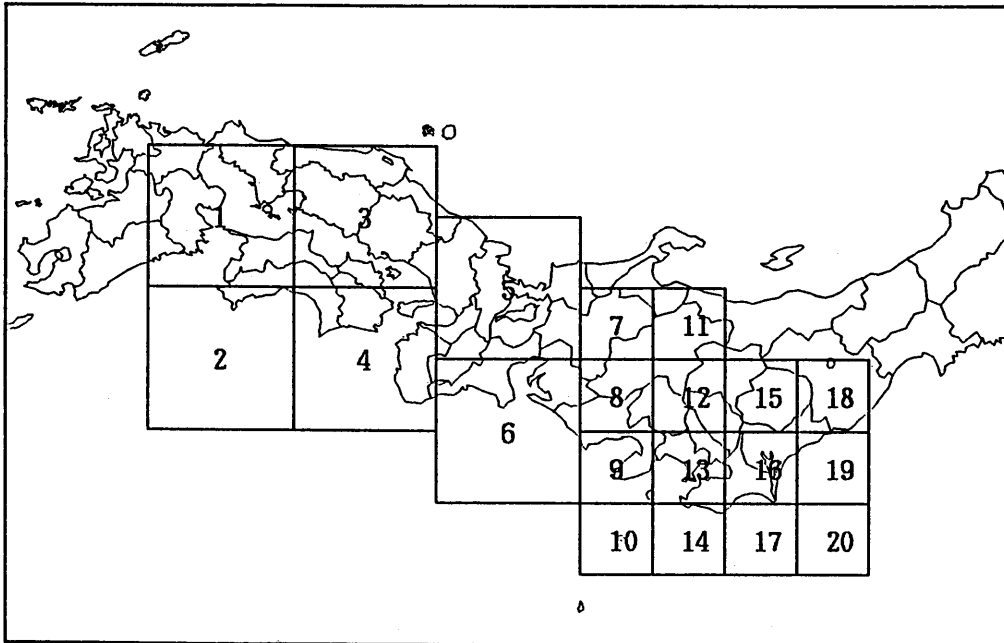
第5図 ニューラルネットワークの予測対象領域。番号の付いているメッシュそれぞれについて降水の有無を予測する。各メッシュの大きさは約20km四方

第2表 ニューラルネットワークへの入力データ

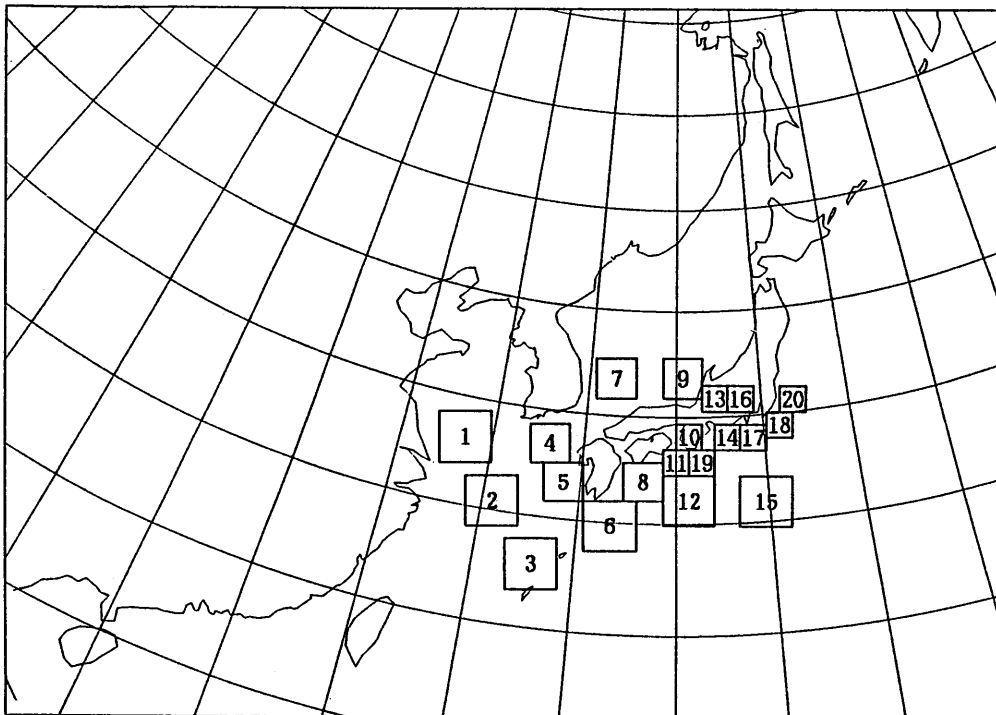
データの種別	内容	データの数
数値モデル (ASM)	Qベクトルの発散	50
	相当層厚の傾度	50
	相当層厚による相対湿度	50
数値モデル (JSM)	全雲量	26
	中層雲量	26
	相当層厚による相対湿度	26
	SSI	26
	500hPaの温度移流	26
	700hPaの上昇流	26
	850hPaの風(東西成分・南北成分)	52
	900hPaの水蒸気流束の発散	26
	降水量(3時間積算値)	26
	実況観測値	アメダスの風(東西成分・南北成分)
アメダスの風の発散		42
アメダスの気温(JSMの850hPaの気温との差)		42
レーダーエコー強度の領域平均値(全国)		20
レーダーアメダス解析雨量(関東)		30
降水短時間予報(3時間積算)		30
GMSのLR-FAXデータの領域平均値		20
その他	季節変化項(元日から数えた通年日のsinとcos)	2
	定数項	1
		計 681



第6図 ニューラルネットワークの入力として用いられる数値モデルGPVの格子点の位置。(a) ASM狭域データ, (b) JSMデータ。JSMは図中のマス内の9点を平均して用いた



第 7 図 ニューラルネットワークの入力として用いられるレーダーエコー強度のメッシュ



第 8 図 ニューラルネットワークの入力として用いられるSDUSのH画像のメッシュ。SDUSの画像データには海岸線と緯経線が含まれているので、できるだけそれらを避けるように設定した

学習の手順については、地方官署のワークステーション上で使用することを想定し、次のような逐次学習方式を採用した。

1. ニューラルネットワークへの入力データは、個々のデータが入るたびに更新しておく。たとえば、数値モデルのデータについては、0530Zおよび1730Zに、その時点の最新の予想値を使って入力データを作って保存しておく。実況観測値については3時間毎に入力データを作成する。入力データは常に1日分を保存しておくようにする。
2. 3時間ごとに、レーダー・アメダス解析雨量のデータが入った時点で検証用のデータ（真の出力値）を作成する。
3. 作成した「真の出力値」と、保存してある入力値を使ってネットワークのBP学習を行う。パラメタ値の変更は一度だけおこなう。
4. 学習の終わったネットワークを使い、最新の入力データによって18時間先までの予測を行う。

2.3.2.2 ニューラルネットワークによる予測例

第9図にこのニューラルネットワークによる予測例を示す。予測の最初の段階では、予測を行う時点の降水パターンからの情報によって、実況との連続性の良い予測パターンとなっている。予測対象時刻までのラグが大きくなるに連れて、実況パターンの影響は小さくなり、JSMの予測の影響が強くなってきている。この例の場合は、実際の降水が持続的なものであったために、むしろ実況パターンの影響を持続させた方が良かったかも知れない。しかし一般的には予測の前半では実況との連続性を重視し、次第にモデルの予測の重みを増していくという方法が妥当と思われるので、この例のようなニューラルネットワークの振る舞いは予測資料として妥当なものといえる。

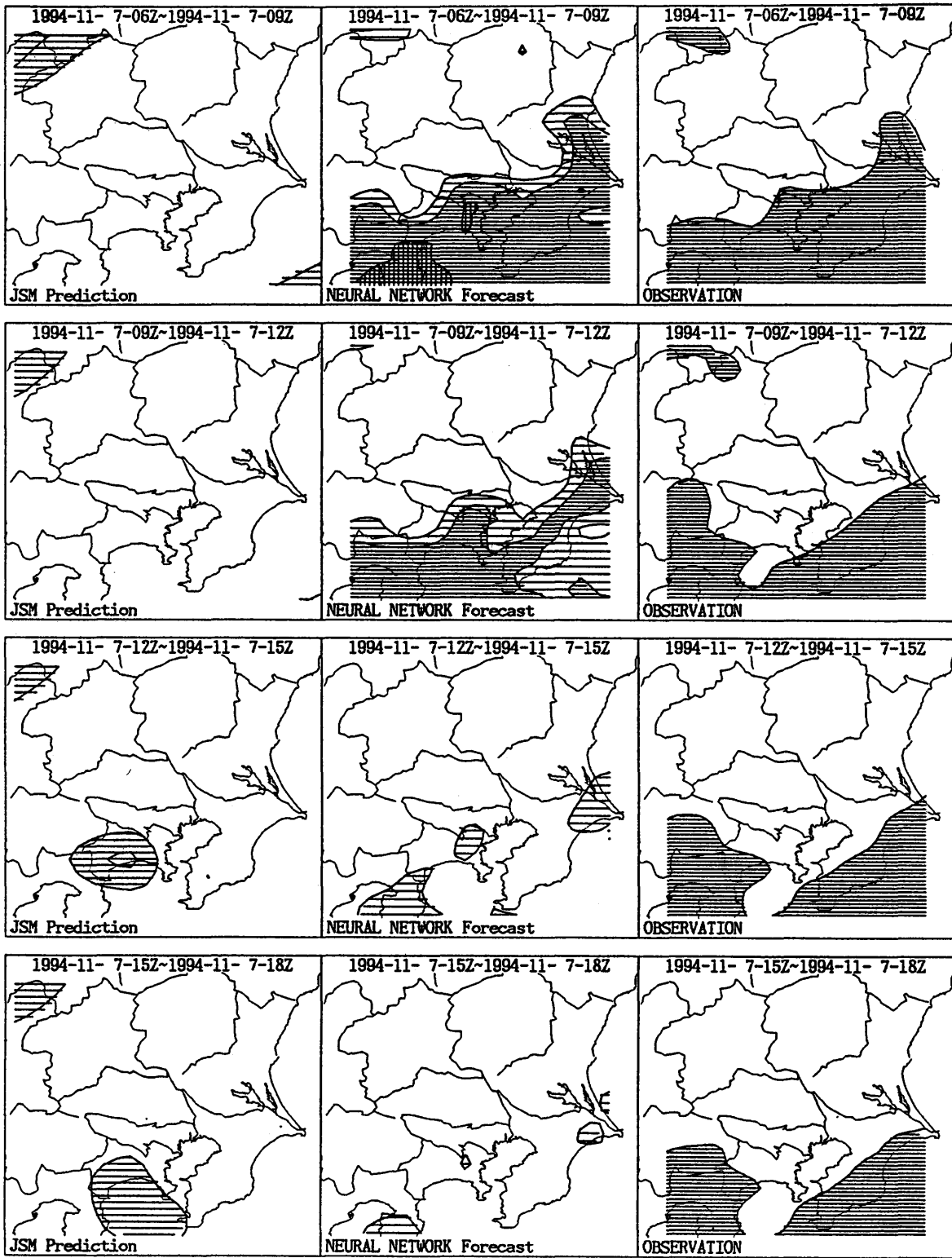
このニューラルネットワークの日々の予測結果を見ると、一般的な傾向としては実況を重視しすぎるきらいがあり、予測の後半まで実況パターンを持続させるような予測をおこなって、かえって予報をはずすという場合が多く見られた。ニューラルネットワークは任意の入力-出力関係を学習させることができるという利点を持っているが、学習の途中では学習の順序（何をまず学ぶか）をコントロールすることができない（最初に乱数で与えるパラメタの初期値できまってしまう）。ここで試みているような逐次学習の場合は、ゆっくりとした学習が長期間継続しているようなものであるから、ネットワークが「何から学びはじめるか」ということは、日々の予報精度に直接影響を与える大きな問題である。

学習の順序をコントロールする方法としては、例えば、第一推定値としてJSMの降水パターンを与え、そこからの修正量をニューラルネットワークに予測させるという方法も考えられるが、この場合は「誤差の予測」を行うことになるので予測自体がかなり困難になる可能性もある。ニューラルネットワークのパラメタの初期値として、乱数でなく何か意味のある値を与えることが最も本質的な解決であると考えられるが、その「意味のある値」をどのように作るかが問題となる。現時点では、後述するエントロピーネットが、ニューラルネットワークに意味のある初期値を与える一つの手法として期待されている。

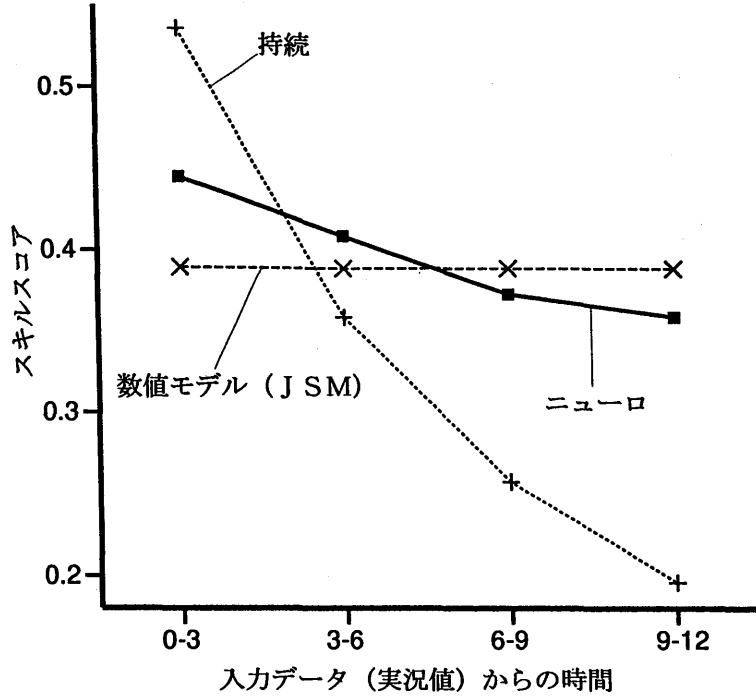
2.3.2.3 ニューラルネットワークの精度

ニューラルネットワークの予測精度を客観的に評価するため、1994年3月17日～10月7日の期間の予測についてスキルスコアを計算した。第10図には入力した実況値の時刻と予測対象時刻とのラグによるスキルスコアの変化を示してある。このスキルスコアはニューラルネットワークの予測値が50%以上であった時を降水ありの予想として計算したものである。JSMの降水量をそのまま利用する場合と比べて、6時間程度先までは実況値による改善が行われているが、それより先ではかえって予報を悪くしている。これは前節でも触れたとおり、予測開始時の降水パターンの影響が先の時間の予測にまで強く残りすぎていることが大きな原因と考えられる。

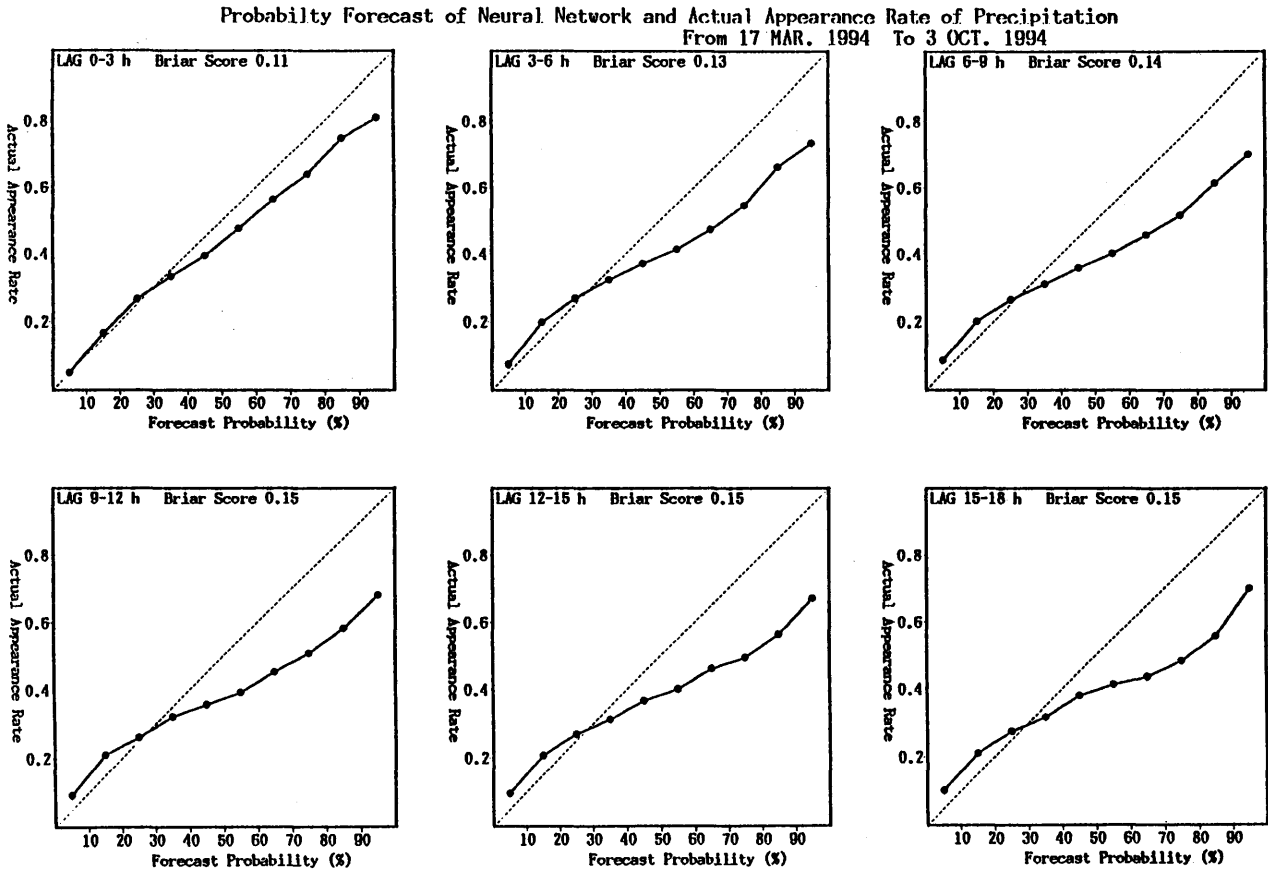
一方このシステムの出力値は便宜的に降水確率とみなすことができるので、確率値として妥当であるかどうかを調べた。第11図はニューラルネットワークの出力した値と、実際の出現率との対応を示したものである。3時間以降の予測においては、予測値が大きい領域で実際の出現率が予測値よりも低くなる傾向がみられるが、ニューラルネット



第 9 図 左:JSMの降水域(3時間積算降水量が0.4mm以上の領域)。中央:ニューラルネットワークによる降水確率(30%, 50%, 80%の等値線が引いてある)。右:レーダーアメダス解析雨量で降水が観測された領域。1994年11月7日00Zを初期値とする数値モデルを用い、11月7日06Zの実況値とあわせてニューラルネットワークへの入力とした



第10図 ニューラルネットワークのスキルスコア。持続予報およびJSMの予想降水量のスコアを合わせて示した。スコアは1994年3月17日～10月7日の期間の予測について計算した



第11図 ニューラルネットワークによる降水確率と実際の降水出現率との関係

ワークの出力値を降水確率とみなして利用することも可能であることが示唆される。

2.3.3 逐次法の限界

このニューラルネットの精度は、3-6時間後の予測については、一応実用になるものであったが、それより先の時間についてはJSMの降水予測と同等以下であり、ニューラルネットが何ら積極的な役割を果たしていないことを示している。

一般にニューラルネットワークのBP学習を行うときは、一定量の入力-出力データのペアからなる「教師データ」を与え、そのデータについての誤差が十分小さくなるまで繰り返して学習（パラメタ調節）するのが普通である。しかし、ここでは日々与えられる新しいデータを、一度学習した後に捨てるという「逐次学習」の方法を採用している。

逐次学習には、大量の「教師データ」を保存しておく必要がなく、一回の学習に必要な計算時間が短くて済むという利点があり、特に小規模のコンピュータ上での運用に適していると考えられる。

しかし、この方法には次のような問題がある。

1. ニューラルネットワークのアルゴリズムは「誤差の極小化」を行うものであり、そのように最適化された状態で本来の機能を発揮するものと考えられている。しかし、逐次学習では、最適化の途中の状態のネットワークを使っていることになるので、期待される性能を発揮する保証がない。
2. 逐次学習はいわば「開かれた教師データ」の学習と考えることができる。「開かれた」とはいつても、無限に新しいデータが与えられるわけではなく、大きく見れば一年を単位とするデータセットを少しずつ変化を付けながら繰り返し学習していると見ることもできる。とすると、この場合は一回の繰り返しの一年かかっているわけで、本来それほど効率的でないBPアルゴリズムでは、誤差が十分減少するまでには相当の時間がかかることが予想される。仮に200回の繰り返して実用的なネットワークが得られるとすると、200年の歳月が必要になるわけで、これでは全く実用にならない。

この調査が行われた時点から今日に至るわずか数年の間にも、計算機に関する環境は大きく変動してきており、今や、数GBのハードディスクと高速のCPUが個人でも購入可能になっている。従って、上に述べたような、計算時間やディスク容量の節約という発想はもはや時代遅れになってしまっており、少なくともニューラルネットワークの学習に関しては、一般的に繰り返し学習を採用すべきであると考ええる。

しかしながら、「システムを逐次的に変更する」という考え方そのものは、天気予報の分野では重要な意味を持つと考えられるので、この事自体は別に探求する必要がある（付録A参照）。

2.3.4 蓄積された知識の抽出について

ニューラルネットワークの問題点の一つは、それがどのような性質を持っているのか、つまり、一つの入力変数が出力に対してどのような影響を及ぼすのか、ということがわかりにくい点にある。重回帰式の場合は、どの予測変数が、被予測変数に対してどの程度の影響を与えるか、という点について、回帰係数が一定の情報を与えてくれる。しかし、ニューラルネットワークの場合、一つの入力変数が出力変数に与える影響は非常に複雑である。このために、ニューラルネットワークの予測値が異常であった場合でも、その原因を把握することは困難であり、このことが、天気予報の現場でニューラルネットワークを使いにくいものになっている。

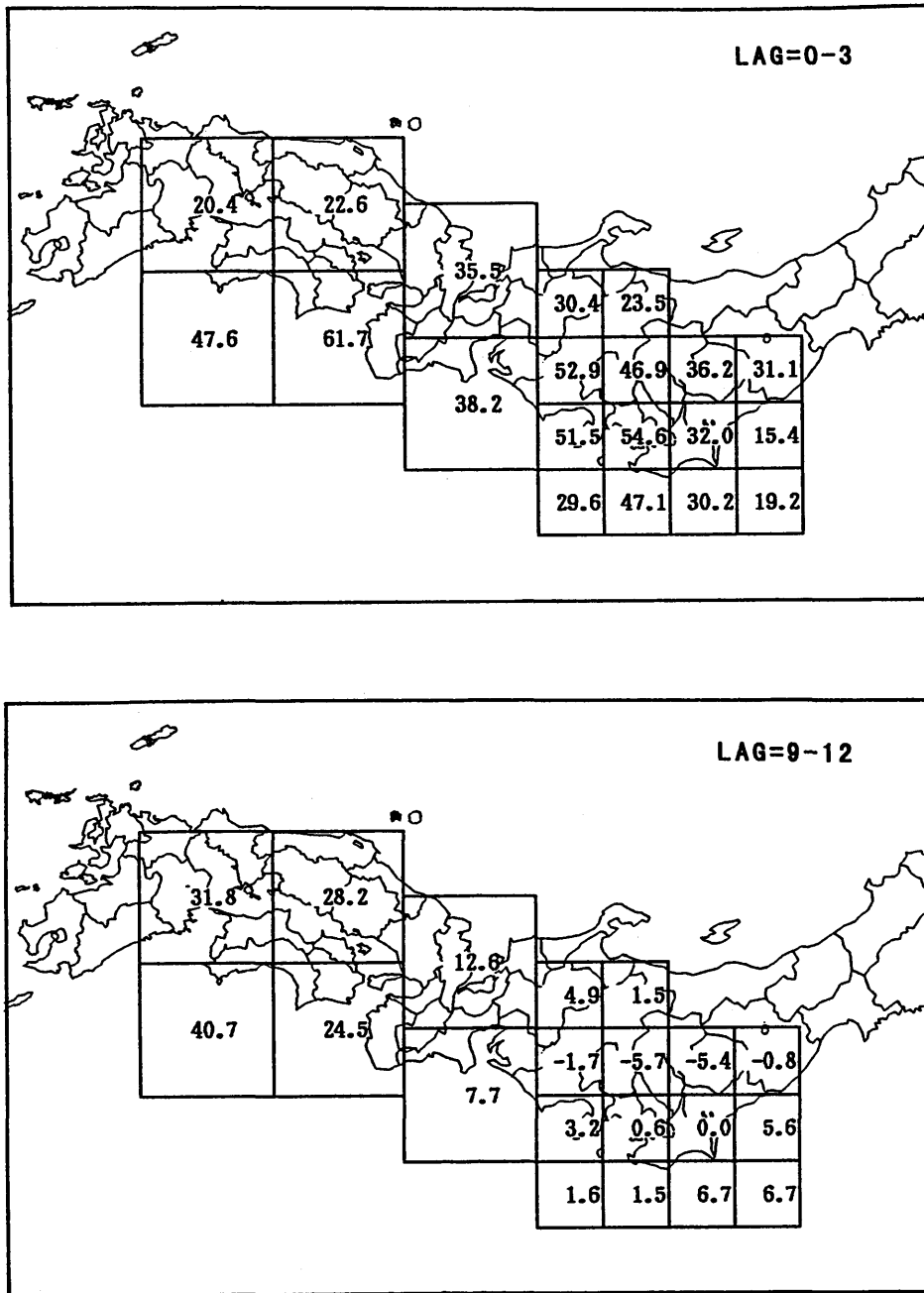
ニューラルネットワークの性質を解析する方法についてはSarle(1998)にまとめられているが、一口に言えば、うまい方法は無い、ということになる。特に、入力変数が相互に相関を持つ場合には個々の入力変数に対するネットワークの振る舞いを正確に理解するのは全く不可能と言って良い。

比較的単純な方法としてLee et al.(1990)が採用した線形応答係数というものがある。ニューラルネットワークの中間層及び出力層で用いられている応答関数を線形関数(最も単純な形は $f(x)=x$)で置き換えると、全体は線型重回帰式になる。そうして得られた線形重回帰式の係数を、出力に対するその入力の相対的な重みとみなすのである。こ

の係数を便宜的に線形応答係数と呼ぶ。

この量を、入力変数の相対的な重要度の指標として用いるには、いくつかの問題がある。第1には、線形重回帰式の回帰係数を予測変数の相対的な重要度の指標として用いるときと同様に、予測変数が正規化されていなければならない、ということである。特に、予測変数間で分散が異なっているような場合には、この量は意味を持たない。また、予測変数同士の間に関連がある場合、この量の絶対値は意味を持たず、相対的な大小関係のみが意味を持つようになる。第2に、ニューラルネットワークは本来非線形関数であるものを、無理に線形化しているため、値の大小が重要度を正確に反映しているわけではない、ということがある。オーダーが一桁小さければ相対的に重要でないと言えるが、たとえば10.0と20.0の場合どちらが重要かを論ずることはできない。

本節のニューラルネットワークの場合、入力変数の正規化は行わなかったため、異なる要素間の重要度の比較を行うことはできない。また、隣接する格子点値相互には強い相関があり、同一要素の中でも正確な比較は困難である。



第12図 レーダーエコー強度の線型応答係数。上は予測対象時間までのラグが0-3時間の場合。下はラグが9-12時間の場合

線形応答係数の計算例を掲げる。第12図は、レーダーエコー強度の線形応答係数を、予測時間毎に示したものである。予測時間が長くなるにつれて、相対的に重要な領域が西に移っており、ニューラルネットワークが遠い先の予測のために、より西の方のエコーに着目するという、きわめて常識的な知識を取得していることが見て取れる。しかし、係数の値そのものは厳密な評価に堪えるものではないため、この値を使ってこれ以上の詳しい解析を行うことは困難である。

ニューラルネットワークからの知識抽出のためには、ネットワークの設計段階からの適切な処置が必要である。入力変数の正規化は必須であるし、もしかしたら、EOF解析などによって相互に相関を持たないような入力変数に変換することも必要かもしれない。また、線形応答係数は、相対的重要度の推定値としては、あまりにも曖昧である。一定量（できれば数年分）のサンプルについて、一つの入力変数を少し動かした場合の出力の変化をひとつひとつ調べるという方法が、最も有効であると考えられる。

2.3.5 考察

ニューラルネットワークは、その非線形性によって、従来の重回帰式では表現できなかった入出力関係を表現できるという点で、優れた方法であることは、いまや周知の事実となっている。

しかし、どのようにして最適なパラメタを求めるか、という点については、必ずしも明快な解答があるわけではない。特に、ネットワークの規模が大きくなった場合には、効率的なアルゴリズムを選択することが必須となる。

また、ネットワークの構造（中間層をいくつ設けるか、中間層ニューロンをいくつにするか）や、BPアルゴリズムの学習定数には任意性があるため、これらを最適に設定するにはどうしたら良いか、という難問もある。更に、ニューラルネットワークにおいても、パラメタ数がサンプル数に比べて多い場合に重回帰式と同じ様な「過適応(overfitting)」が起こることが知られており、これをどのように回避するか、ということも問題になる。

このように、ニューラルネットワークそれ自体は強力なツールではあっても、最適なモデルを得るための手順は必ずしも自動化されているわけではない（むしろ、多くの面でまだ試行錯誤の途上にある）という点には注意する必要がある。「ニューラルネットワークを使っているから良い結果がでるはず」という思い込みは危険である。ニューラルネットワークに関しては、「既に確立した技術を応用すればよい」といった安易な状況にはないことを改めて強調しておきたい。

2.4 ID3による分岐木の自動生成

単純ベイズ法もニューラルネットワークも、与えられた全資料を一度に取り込んで一つの結論を出すものである。しかし、一般に人間の思考過程には、パターン認識のように全体を一度に見渡して判断を下すようなもの他に、論理的に順を追って結論にたどり着くというものもある。人間の予報者が天気予報を行う場合でも、天気図などの個々の資料はパターン認識で一瞬にして把握するにせよ、それぞれの資料の相互の関りについては、順序立った検討が行われることが普通である。

こうした段階的な思考過程を客観的に記述する手段として分岐木がある。地方官署などでは、特定の現象（たとえば雷など）の発生を予測するための道具として、分岐木の形の「ワークシート」を作成している所がある。

このように、分岐木は人間の思考過程を記述する一つの方法としても使えるが、これをデータから自動的に作成する方法もあり、こうすればデータに内在している知識を、人間にわかりやすい形で表現することが可能になる。

いくつかの判断材料と正しい判断結果からなるデータの集合が与えられたときに、そこから効率的な判断規則を、**木構造の形で帰納的に与える方法**がいくつか知られているが、本節では、それらの中で最も基本的な方法であるID-3 (Quinlan,1986 ただし本稿では安西,1989 の紹介による) について述べる。

2.4.1 ID3アルゴリズムの概要

与えられたデータの集合から、個々の判断材料（「テスト」と呼ばれる）の有効度を以下のように計算する。N個のサンプルからなるデータの集合があるとし、最終的な判断結果は○か×かのような2値であたえられるとする。また、個々のテストもyesかnoかのような2分割のみとする（もちろん判断結果が3値以上であったり、2より大きい分割を行うテストが含まれていたりしても本質的には変わらない）。このN個のサンプルのうち、最終結果が○となるものが N_o 個、×となるものが N_x 個とすると、このデータの集合の情報量 E は次の式のように定義される。

$$E = - \sum_{i=0}^1 \frac{N_i}{N} \log_2 \frac{N_i}{N}$$

この定義式からも推察できるように、サンプルの中で最終結果がランダムであれば E は大きく、最終結果がそろっていれば（即ち、 N_i/N が0か1に近い値であれば） E は小さくなる。

このサンプルの内、あるテストでyesであったものが N_y 個、noであったものが N_n 個とする。このテストの結果によって、N個のサンプルをyesのグループとnoのグループの2つに分割すると、yesとなったグループの情報量 E_y は上と同様に

$$E_y = - \sum_{i=0}^1 \frac{N_{yi}}{N_y} \log_2 \frac{N_{yi}}{N_y}$$

と計算できる。ただし、この N_y 個のサンプルのうち、最終結果が○となるものの数を N_{y0} 個、×となるものの数を N_{y1} 個としている。同じように、テスト結果がnoとなったグループの情報量 E_n も計算することができ、この E_y と E_n から、分割した後の情報量の総和 \hat{E} を次のように求めることができる。

$$\hat{E} = \frac{N_y}{N} E_y + \frac{N_n}{N} E_n$$

一般に \hat{E} は E と同じか、より小さくなる（分割することで、より結果のそろったグループを生じる）ので、情報量の降下量 $E - \hat{E}$ によってそのテストの持つ有効度を測ることができる。つまり、 $E - \hat{E}$ が大きいほど、ランダムな状態から秩序のある（最終結果のそろった）状態へ変化しているということになる。

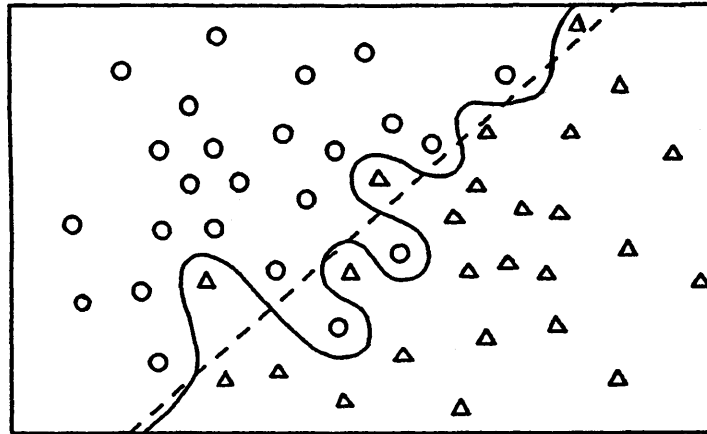
全てのテストのうち、最も有効度の高いテストによってデータを分割し、分割された各々のグループについて上と同様にテストの有効度を計算し、最も有効度の高いテストで更に分割を行う。以下、この操作を繰り返すことによって、与えられたデータから自動的に一つの分岐木が得られる。

2.4.2 枝切りの問題

上の手順は、分割が不可能になるまで繰り返すことができる。しかし多くの場合、最後まで分割を繰り返すと、個々のグループに含まれるサンプルの数が少なくなってきて、意味のある分割かどうかわからなくなってしまふ。そこで、どこまでで分割を止めるかという、分岐木の「枝切り」の問題が出てくる。これについては、いくつかの提案がなされているようだが、ここでは伊藤（1992）の提案したMDL原理による方法を用いた。

与えられたデータに対して適合する複数のモデルがあるときに、どのモデルが最も適当かを判断することは、統計的手法において重要であり、かつ、困難な問題である。たとえば、重回帰式で予測を行う場合に、どの予測因子を選択するか、予測因子の数を幾つにしたら良いか、といったことを決定するのは容易ではない。こうした問題の解決のために、モデルの適合性の客観的な尺度として、情報量基準というものが提案されている。MDL原理はそうした情報量基準の一つである。MDLはMinimum Description Length（最小記述長）の略で、いくつかのモデルの中で記述長が最小になるモデルを最も良いと判断する原理である（Rissanen,1978, 山西・韓,1992）。

記述長とはどういうものかを簡単に説明する。第13図のような○と△のデータの分布を記述する場合次のような方法が考えられる。



第13図 2次元的なデータ分布の記述の例

1. 1つ1つのデータの座標と種類を記述する。この場合の記述の量が膨大なものになることはいうまでもない。
2. 第13図の破線のような境界線の記述を与えて、境界線の右側では△といったおおまかな記述を行い、更に例外の事例についての記述を行う。
3. 第13図の実線のような境界線の記述を与えて、境界線の右側では△といった記述を行う。この場合は例外の記述は不要であるが、境界線の記述が長くなる。

これにMDL原理を適用するとすれば、境界線の記述長・大域的な分布の記述長・例外の記述長をそれぞれ定量的に評価し、その総和が最も小さくなるものを求める。この場合は、直観的には上の2に近いものが得られると思われる。

分岐木にMDLを適用する場合も同様に、分岐木自体の記述長と分岐された末端における例外事例の記述長の和を計算して、最小となるものを求めれば良い。ただ、分岐木の記述長については様々な定義が考えられる上、一般論としては実現可能なあらゆる大きさの分岐木について総当たりに記述長を計算しなくては行けないので、現実にはかなり困難な作業であるといわざるを得ない（たとえば中莖他, 1992, 鈴木・圓川, 1995など）。

本節で用いた伊藤(1992)の方法は、最終結果が0か1かの2値で、途中の分岐もすべて2分割であるような特別の場合について、末端の枝1つについて下の式の L という量を計算し、これを末端の枝全てについて総和を取ると、それが分岐木全体の記述長となるというものである。 L の定義式のうち、第1項が分岐木自体の記述長(末端の枝1本当りの値)、第2項以下が例外事例の記述長に相当する。

$$L = 2 + 2\log_2\left(\frac{1}{s}\right) - n_0\log_2\left(\frac{n_0}{n}\right) - n_1\log_2\left(\frac{n_1}{n}\right)$$

ただし、 $s = n_0 n_1 / n^2$ 。 n はその枝に落ちてきたデータの数、 n_0 はそのうち最終結果が0となるものの数、 n_1 は最終結果が1となるものの数である。

この方法の場合、末端の枝についてだけ考えれば良いので、ある枝の L の値が L_0 、これを更に2つに分割した場合の2つの枝の L の値をそれぞれ L_1 、 L_2 とすると、 $L_0 > L_1 + L_2$ なら分割を行い、そうでなければ分割を行わないというやり方で最適な分岐木を求めることができる。

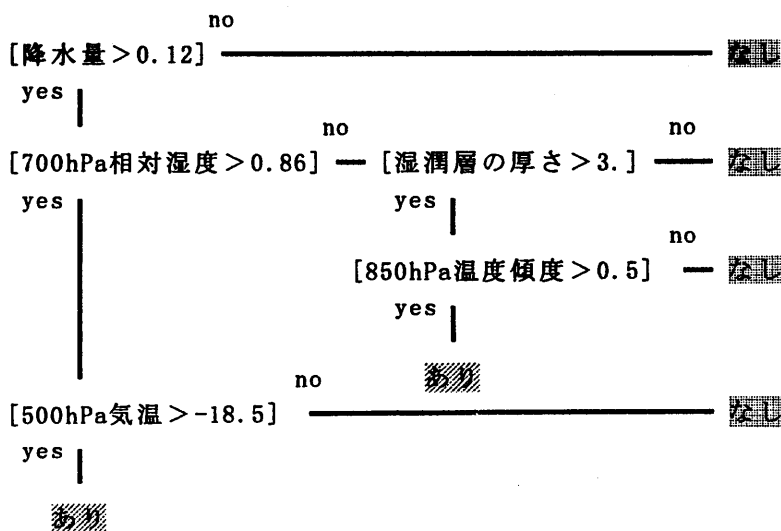
2.4.3 JSMデータから作った降水の有無を予測する分岐木

ここでは、JSMの24時間予想値を用いて東京で1mm/3h以上の降水があるかどうかを判断する分岐木を作成する例について説明する。この例では第3表のデータを予測資料として用いた。分岐のためのしきい値は、事前の調査によって求めた値(第3表に記載)を与えた。

1991年11月から1992年10月までの一年分のデータを用いて作成した分岐木が第14図である。大元にはJSM降水量に

第3表 ID3による分岐木作成のために用意した予測資料および分岐のためのしきい値

要素	事前に求めたしきい値
500hPa 相対湿度	0.900
700hPa 相対湿度	0.860
850hPa 相対湿度	0.820
900hPa 相対湿度	0.885
700hPa ω	-8.5
700hPa 湿数	2.0
850hPa 湿数	2.5
900hPa 湿数	2.5
湿潤層の厚さ (DWL)	3.0
700hPa 温度移流	2.0
850hPa 温度移流	3.5
900hPa 温度移流	3.5
850hPa 相当温位移流	10.5
900hPa 相当温位移流	16.0
500hPa 気温	-18.5
850hPa 気温	8.5
850hPa 相当温位	295.5
850hPa 渦度	2.0
850hPa 渦度移流	9.5
900hPa 渦度移流	10.0
850hPa 温度傾度	0.5
900hPa 温度傾度	0.5
1 時間降水量	0.12
気温差 (地上-900hPa)	6.0
気温差 (850-700hPa)	7.5
相当温位差 (850-900hPa)	1.0
温位差 (700-850hPa)	8.5



第14図 JSMの24時間予想値から、東京における 1 mm/ 3 h以上の降水の有無を判別する分岐木

よる分岐がおかれ、これがしきい値（ここでは0.12mm）より小さい場合には、「降水なし」とされる。JSM降水量がしきい値より大きい場合には、他の資料（700mbの湿度など）を参照してさらに細かく分岐していく。

当時のJSMの降水量は、どちらかといえば「見逃しを減らす」というチューニングになっていたようで、このため逆にJSM降水量が一定値より小さければ、かなりの高確率で「雨なし」と判断することができた。ここでの分岐木はJSM降水量のこうした性質を反映したものとなっている。

2.4.4 考察と最近の話題

ID3の最大の利点は、作られた予測モデルが、人間にわかりやすい形をしているということにある。また、自動的に作られた分岐木に、人間の専門家が手を入れるということも比較的容易である。従って、専門家の知識と機械学習の融合という点から見れば、この方法は本章でとりあげている各手法の中でも最もすぐれたものといえることができる。

上で得られた分岐木を例として考えると、この分岐木は、降水量によって最初の分岐が行われる形となっている。これはJSMの降水量が「降水なし」を分離するための尺度として優れているためであるが、おそらくこうした判断順序は、ベテランの予報者の判断過程とは合致しないであろう。もしこの形が専門家のやり方にあわないのであれば、作成された分岐木を元に、判断順序の入れ替えを行うなどの修正をすればよい。そのような修正を通して、人間の知識を客観的予報技術に取り込んでいくことができる。

ただし、このやり方が成功するためには、機械的に分岐木を作成する際、尺度の候補を十分に用意しておく必要がある（人間が参照しそうな資料をすべて尺度の候補として用意するのが理想である）。おそらく多くの予報官にとっては、総観場のチェックがはじめるのが自然なやり方であろうから、第14図のようにモデルの降水量からみていくという手順は不自然に思われることと思う。しかしここで用意した尺度の候補（第3表）は対象地点に近いJSMの格子点の値のみで、総観場を記述するものが無いため、残念ながらそうした予報官の感覚に基づいた修正をおこなうことができない。

ID3の弱点の一つは、単純ベイズ法と同じく、予測資料の離散化が必要となる点にある。この点については、連続量をそのまま扱い、分岐のためのしきい値も自動的に設定するアルゴリズム（IndectsやC4.5）が提案されており、こうした手法の導入も検討の余地がある。

2.5 分岐木とニューラルネットワークの相互変換

2.5.1 エントロピーネットの概要と降水予測への応用

機械的に作成された分岐木を人間が修正する場合、分岐のためのしきい値をどのように設定するかが問題になる。人間は、分岐木の構造のような論理構造については適切な判断ができるが、しきい値のような定量的な設定は苦手だからである。また、修正を行わない場合でも、作成した分岐木を長期間そのまま使用していると次第に陳腐化していく（つまり、分岐木作成の段階で与えたデータには含まれなかったような現象に遭遇して適切な予報ができなくなる）おそれがある。

これらに対処するもっとも根本的な方法は「分岐木を初めから作り直すこと」である。実際、Quinlan(前節で説明したID3アルゴリズムの考案者)も、「陳腐化するような分岐木は価値がないので初めから作り直すべき」といっている。しかし、このためにはすべてのデータを保存し続けなければならない、あまり現実的とはいえない。

そこで、ここではSethi(1990)のエントロピーネットの手法をつかって分岐木をニューラルネットワークに変換し、逐次学習を行うことにした。ニューラルネットワークの節に書いたとおり、ニューラルネットワークは与えられたデータを学習してその入力-出力関係を再現する能力がある。一般的なニューラルネットワークの場合には学習した結果を人間にわかるような形で取り出すことが難しいという欠点があったが、ここで述べる方法を使えば、分岐木という人間にわかる形のものをニューロ化し、また必要に応じてニューロから分岐木への変換を行うことで、このような

ニューラルネットのブラックボックス化を避けることができる。

たとえば第14図の分岐木は次式のようなニューラルネットワークとして記述できる。

$$a_1 = \frac{1}{1 + \exp\{-\alpha(x_1 - 0.12)\}} \quad (1)$$

$$a_2 = \frac{1}{1 + \exp\{-\alpha(x_2 - 0.86)\}} \quad (2)$$

$$a_3 = \frac{1}{1 + \exp\{-\alpha(x_3 - 3.0)\}} \quad (3)$$

$$a_4 = \frac{1}{1 + \exp\{-\alpha(x_4 - 0.5)\}} \quad (4)$$

$$a_5 = \frac{1}{1 + \exp\{-\alpha(x_5 - 18.5)\}} \quad (5)$$

$$b_1 = \frac{1}{1 + \exp\{-\alpha(a_1 + a_2 + a_5 - 2.5)\}} \quad (6)$$

$$b_2 = \frac{1}{1 + \exp\{-\alpha(a_1 - a_2 + a_3 + a_4 - 2.5)\}} \quad (7)$$

$$y = \frac{1}{1 + \exp\{-\alpha(b_1 + b_2 - 0.5)\}} \quad (8)$$

$$\alpha = \text{const.} (\gg 1.) \quad (9)$$

ただし、 x_1, x_2, x_3, x_4, x_5 はそれぞれ降水量、700hPaの相対湿度、湿潤層の厚さ、850hPaの温度傾度、500hPaの気温である。また、 y は出力値で0.5以上ならば「降水あり」、0.5未満ならば「降水なし」を表す。

ここで、個々の式について簡単に説明する。 $1/[1 + \exp\{-\alpha x\}]$ の形の関数は前出のシグモイド関数で、 x が0より大きいと急速に1に近づき、0より小さいと急速に0に近づくという性質を持っている(α を正の大きな値にすることで0および1への接近を加速することができる)。従って(1)~(5)式は、 x_i がそれぞれのしきい値より大きいか小さいかで a_i が1に近い値か0に近い値をとることを表しており、分岐木の個々の分岐点での分岐に対応している。また、(6)式は、 a_1, a_2, a_5 のすべてが1(に近い値)の時のみ b_1 が1に近い値をとることを表しており、第14図の分岐木の、降水量からまっすぐ下に降りている枝(降水量、700hPa相対湿度、500hPa気温がすべてyes)に相当する。同様に(7)式は a_1, a_3, a_4 がすべて1(に近い値)で、かつ a_2 が0(に近い値)である時にのみ b_2 が1に近い値をとることを表している(第14図で700hPa相対湿度でnoに進んだ後、下に降りる枝に相当する)。さらに、(8)式は、 b_1, b_2 のいずれかが1(に近い値)の時 y が1に近い値になることを表しており、これは第14図の分岐木で「降水あり」となるのは2つの枝のいずれかに落ちた場合であることに対応している。

一般の3層ニューラルネットワークは

$$u_i^{(1)} = \sum_k w_{ki}^{(1)} x_k + w_{0i}^{(1)} \quad (10)$$

$$v_i^{(1)} = \frac{1}{1 + \exp\{-\alpha(u_i^{(1)})\}} \quad (11)$$

$$u_j^{(2)} = \sum_n w_{nj}^{(2)} v_n^{(1)} + w_{0j}^{(2)} \quad (12)$$

$$v_j^{(2)} = \frac{1}{1 + \exp\{-\alpha(u_j^{(2)})\}} \quad (13)$$

$$y = \frac{1}{1 + \exp\{-\alpha(\sum_j w_j^{(3)} v_j^{(2)})\}} \quad (14)$$

のように書くことができる(模式的には第15図(a)のように表せる)。(1)~(9)の式はこの式の中のパラメタ $w_j^{(k)}$ を特定の値に固定したものと考えることができる(第15図(b))ので、一般のニューラルネットワークに(1)~(9)式に合うような各パラメタの初期値を与え、後は通常のニューラルネットの学習アルゴリズムを用いて逐次的に学習させることにより、第14図の分岐木を新しいデータで逐次的に洗練することができるのである。また、学習の際に(10)~(14)式の特定のパラメタだけを調節することによって、たとえば分岐のしきい値だけを学習させたりすることもできる。

第14図の分岐木から作成したニューラルネットワークの、検証期間(1992年11月~1993年10月)におけるスキルスコアは、分岐木をニューラルネットに変換しただけの状態では0.48であったが、検証期間のデータを逐次学習(一つのデータについて予測を行った後でそのデータの学習を行う)した場合には0.51となった。ここでは分岐のしきい値のみの学習を行ったのだが、学習によるスコアの向上が見られた。

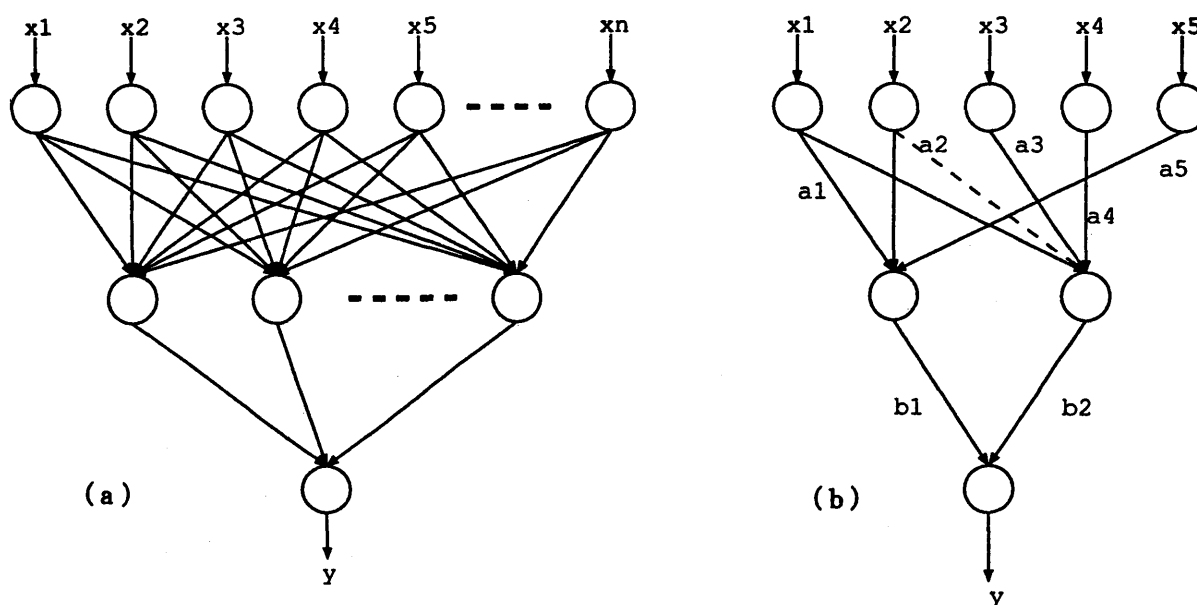
2.5.2 考察

2.5.2.1 「何を学習すべきか」という問題

分岐木から生成されたニューラルネットワークは、最初の時点では元の分岐木の構造をそのまま反映しており、各層(入力層・中間層・出力層)をつなぐリンクの中には値が0のもの(つまりリンクが存在していない部分)もある。

ニューラルネットワークの学習というのは、リンクの値の調整であるから、学習の過程でネットワークの構造は最初に分岐木の構造からは離れていく。問題は、この、分岐木構造からの逸脱をどの程度まで許容するか、ということである。

本節の応用例では、入力層-中間層リンクの中で、最初の段階で値を持っていたものだけについて値の変更を許した。これは分岐木の分岐のしきい値のみをニューロの学習によって調節したことと同じである。こうすれば元の分岐木の木構造自体は変更を受けない。しかし同時に、限定されたリンクだけを調節することによって学習が部分的なも



第15図 ニューラルネットワークの模式図。xは入力、yは出力を表わす。(a)一般的なネットワーク構造。(b)第14図の分岐木をニューラルネットワーク化した場合の構造

のになっていることもまた事実である。現に、最初の時点でリンクの存在しない箇所にも初期値0のリンクを仮定し、更に入力層—中間層以外の層についても学習を許すことにすれば、この例については、予測誤差を更に小さくすることが可能になる。この場合は、学習後に得られたネットワーク構造は元の分岐木とは全く異なるものになっている可能性が大きく、分岐木の形に逆変換できるかどうかもわからない。

このように、「構造が人間にとってわかりやすい」ということと「予測性能が高い」ということとは相容れない要請である場合がある。こうした場合に、それぞれの要請をどの程度まで満たすか、ということは、予測システム的设计思想にかかっている。「とにかく予測性能が高ければ良い」という設計方針もあるが、これはあまり得策とは思えない。なぜなら、どんなに性能の高いシステムであっても、それがブラックボックスになってしまうと、それ以上の改良の糸口が掴めなくなってしまうからである。

2.5.2.2 ニューラルネットワークの解釈法としての可能性

エントロピーネットの考え方は、ニューラルネットワークと分岐木が同じものを表現することができることを明らかにした点で、画期的であった。

このことを敷衍して、一般にはその構造の解釈が困難なニューラルネットワークを、分岐木の形に表現することで理解しやすいものにするという方向が考えられる。実際には、分岐木からニューラルネットへの変換が常に可能であるのに対して、ニューラルネットから分岐木への変換は常に可能であるとは限らず、また、可能な場合でも、作られた分岐木が解釈の困難な複雑なものになることがあるが、部分的にはあれ、ニューラルネットワークの解釈が可能になるということは大きな意味がある。

2.5.2.3 ニューラルネットワークの初期値としての可能性

一般に、FF型のニューラルネットワークのBP学習を行う場合、パラメタの初期値は乱数によって与えることが多い。乱数による初期値から始めるのは、人間の学習にたとえれば、何の知識も持たない赤ん坊の状態から学習を始めることに相当する。しかし実際には、人間の専門家は天気予報についての知識・経験を持っているのであるから、赤ん坊の状態から学習を始めるのは学習時間の浪費であり、効率的でない。

ニューラルネットワークの初期値の与え方としては、入力変数をあらかじめ主成分分析してその係数を与える（中間層ニューロンで表現される入力変数空間内の識別面を、あらかじめ入力変数の変動軸に直交するように設定する）などの工夫もあるが、本節で述べたエントロピーネットによる分岐木のニューロ化も、人間の知識をニューラルネットワークの初期値として与える手法として考えることができる。つまり、分岐木から変換したネットワークを初期値として学習を始めることによって、人間が既に取得している知識の上に機械学習による知識を積み上げることができる、という考えである。

そのような観点から見れば、本節で述べた例も、与えられた資料（第3表）と東京での降水の有無との関係を表すニューラルネットワークを作成するに当たって、既得の知識（分岐木）をもとにした初期値から学習を開始した、ということができる。本節で見たような小規模なネットワークの場合は、完全な乱数から学習を始めた場合とそれほど大きな違いはないが、もっと大規模なネットワーク（たとえば第3節で述べたようなもの）であれば、適切な初期値から学習を開始するということが、学習の効率化に大きく貢献するものと期待される。

ただし、大規模なニューラルネットワークに適切な初期値を与えるためには、人間の持っている複雑な知識を体系的に記述する必要がある、作られる分岐木も大規模複雑なものにならざるをえない。こうなると、そのような複雑な知識表現をどうやって行うのか、という問題が生じて、“データからの知識獲得”と“既得の知識の客観化”との間で、どちらを先に行うかという堂々巡りに陥ることになりかねない。

人工知能技術の応用全般に言えることだが、基本となる技術が未だ発展途上にあるため、我々としてはその都度その時点で利用可能な技術は何かを確認し、“開発のコスト”と“その技術から得られる利益”とのかねあいを考慮して

採用する技術の選択を行っていかねばならない。筆者の私見では、現時点では“データからの知識獲得”の方が“人間からの知識取材”よりもコストが低いと考えるが、こうした状況も日々変化しているのあり、また、開発者自身の知識・経験の量によっても変わってくる問題である点には注意する必要がある。

2.5.2.4 我々はどんな知識を持っているのか？

本節の応用例ではID3で作成した分岐木を元にしてテストをおこなったが、本来エントロピーネットの役割は、人間の経験的知識を分岐木の形で表現し、それをさらにニューロ化することによって洗練することにある。従って、元になる分岐木は人間が作り出したものであることが望ましい。

しかしながら、実際に予報の現場で分岐木形式のワークシートを作成しようとしたときに誰もが経験するとおり、「人間の経験的知識を分岐木の形に表現する」というのは、それほど容易ではない。これは、分岐木（ワークシート）の作成手法の問題ではなく、そもそも「分岐木」という形式が人間の知識・経験を表現するのに適したものなのかどうか、という問題ではないかと思われる。

一つには、天気予報で用いられる資料のほとんどが数値データであるにもかかわらず、予報者の持つ経験的知識は多くが天気図型に代表されるような記号的なものであり、しかもそれらは常に曖昧さを含んでいる。つまり、一般的に言って、人間の思考過程においては数値的なデータから曖昧さを含む記号的な情報を取り出してそれをもとに判断を行うことが多い。

分岐木は、人間に理解しやすいものではあるが、「曖昧さを含むことができない」という点では、人間の判断過程と決定的に食い違っている。その点で、ニューラルネットワークはファジネス（あいまいさ）を含んだものも扱うことができると考えられるので、「曖昧さを含む分岐木」を、ニューロによって表現することが試みられるべきなのかもしれない。

いずれにせよ、人間の持つ知識経験を適切に表現する手段については未だに十分満足できるものはない。人間の知識経験を表現方法にブレークスルーがおきるかどうかで、これからの客観的予報技術の在り方は大きく変わってくるであろう。先にも述べたとおり、現時点ではこの部分がボトルネックになっていて、人間が行っている天気予報全体の客観化を難しくしている。

2.6 まとめ—客観的予報技術が抱える課題

もともと、本研究が要請された背景には、天気予報が個々の予報者個人の技術レベルに依存する「職人芸」になっていることに対する問題意識から、熟練した専門家の持つ技術を客観的に記述して共有したいという要求があった。

しかし、この研究が行われている間に情勢は急速に変化し、今日では、客観的な手法による資料のみが重視され、予報者個人のスキルへの依存度は相対的に小さくなったように見える。この背景には客観的手法の最たるものであるところの数値モデルが急速に進歩したという事実があり、またそれ以外の技術にしても、計算機環境の改善によってより複雑高度な技術の利用が可能になったということがある。

数値モデルをはじめとする高度な客観的技術が発展していくことそのものは、それ自体きわめて望ましいことである。しかし、客観的な手法であれば無条件に主観的な（予報者個人のスキルに依存する）技術に優っているという、盲目的な信仰があるとすれば、それは問題であると言わざるを得ない。

本研究は、「熟練した予報者の持つスキル」の存在を前提としている。情勢は変化したが、この前提は間違っていないと思う。問題は、このスキルを誰の目にも明らかな形で表すにはどうしたら良いのか、ということである。エキスパートシステムや分岐木表現のような、専門家への取材に基づく方法が正攻法であることは事実であるが、これは取材方法・知識の表現方法がともに確立されておらず、実際の問題に適用するのは容易ではない。

本章では、データに内在している（従来の、数値モデルを含む客観的技術では取りこぼされた）情報を、機械学習

の技術によって取り出す試みについて述べた。専門家はこれらの、言わば「隠されている」情報を、その経験・知識によって取り出し利用していると考えられるので、その意味では、ここで述べた各種の手法はデータの側から、専門家のスキルに接近しようとする試みと言うこともできる。

このような機械学習の開発も、将来的には専門家との情報交換（機械学習で得られたモデルを専門家が修正する、あるいは、専門家の持つ定性的なモデルを機械学習によって定量的に構築する）を目指すべきであると考えられるので、やはり、人間の知識の客観的記述方法の開発が大きな課題であるということになる。

このことは、人工知能分野での基礎技術の進歩を待つという側面もあるが、天気予報という分野独自でやらなければならないことも多いと思う。たとえば「天気図型」ということ一つ取っても、テレビ天気予報などで天気図は多用されているにもかかわらず、「天気図型」の厳密な定式化や、天気図が持っている情報（我々はそこから何を読み取っているのか）についての体系的な記述はほとんどない。しかし専門家は一枚の地上天気図から多くの情報を取り出しているのである。たとえば、個々の専門家が自分の行っていることを客観的に記述するように努めてみるというだけでも有益な資料になる。そうしたことの積み重ねによって、天気予報に関する専門家の知識の体系的客観的な表現を目指すべきではないだろうか。

付録A. 逐次学習の必要性と可能性について

本章で述べてきた各種手法のうち、単純ベイズ法以外は、いずれも、本来は逐次学習を想定していない。ニューラルネットワークにしても、本章で試みたような逐次学習は学習効率の点で問題があり、実用に堪えるネットワークを得るためには、通常行われているように、固定した教師データについて繰り返し学習を行う必要があるという報告もある（小泉,1997）。

手法（予測モデル）が複雑になればなるほど、学習（パラメタの調節）には繰り返し(iteration)が必要になる。従って、そのためには固定したデータセットがどうしても必要であり、常に新しいデータに対して開かれているような逐次学習手法を構築することは困難になる。

しかしながら、天気予報を行うシステムにおいては、本来、逐次学習を行うことが好ましい。その理由は次のとおりである。システム構築の段階では、「起こりうるすべての事例」を用意することはできない。そのため、出来上がったシステムは、システム構築時には想定していなかったような現象に遭遇したときに、異常な振る舞いをする可能性がある。新しく与えられた事例をその都度学習していく逐次学習システムであれば、このような異常な挙動は、その事例がとりこまれるまでの一時的なもので済む。

非常に理想的な状況を想定すれば（計算機の能力・利用できるデータが共に無制限であるとすれば）次の2つの方法が考えられる。一つは、大規模複雑なモデルに十分大量なデータを与えて繰り返し学習させることで、「想定されるあらゆる場合に対応できるシステム」を構築すること、もう一つは、「事例に基づく推論」つまり、過去の類似事例検索による予測法を構築しておき、新しく与えられる事例を、その都度過去事例のデータベースに加えていく、というものである。

現時点では、この二つの方向のいずれが、より実現可能であるかを判断することは難しい。前者は例えば超大規模なニューラルネットワークに最低10年分くらいのデータを学習させる、といったことが考えられる。超大規模なニューラルネットワークが実用的な時間内に学習を終える（計算が収束する）ためには、相当高速の計算機が必要である上、データ量も巨大なものになるので、実現には今しばらくの時間が必要であるように思われる。一方、後者の「事例データベース」は、計算機の能力という点から見れば、前者よりも実現性が高いが、「類似事例の検索に基づく予測」という方法そのものがどれほど有効なのか、という点について疑問があるのと、データベースをどのように構築するかという明確な設計思想が存在していないという点で、実現には困難が伴うと思われる。

今日、天気予報ガイダンスで用いられているニューラルネットワークおよびカルマンフィルターは、固定されたデータセットによる学習をある程度行った後に、逐次的に与えられるデータによって、パラメタを更新していくという折衷的な方法を採用している。方法的には、これが現在実現可能な最も良いものだと考えられるが、ニューラルネットワークについては、本節のはじめに述べた通りに、本来、逐次学習が許されるものかどうかははっきりわかっていない（逐次学習をするということは、常に学習途中のネットワークを使うことになるが、学習途中のネットワークにどのような意味があるかが明らかでない）という問題があり、カルマンフィルターについては、基本となるモデルが線形重回帰式であるため、もともとあまり複雑な関係を表現できないという問題がある。折衷方式の改善のためには、基本となる統計モデルの見直しが最も本質的である。

付録B. ニューラルネットワーク応用の最近の進展(1996年以降)

B.1 共役勾配法の導入

第10図で見たとおり、本文中で取り上げたニューラルネットワークは、JSMの降水量をそのまま用いた場合と比べて、6-9時間より先については予測を改善しておらず、ニューラルネットワークを使った意味がないという結果になっている。

本文中でも考察したとおり、ニューラルネットが期待したほどの性能を発揮していない最も大きな原因は、学習(ニューラルネットワークのパラメタの調節)が十分に行われていない、という点にあると思われる。ここで採用した「バックプロパゲーション法の逐次学習」という方式は思ったほどうまく働いていない。

そこで、同じ問題(入出力およびネットワーク構造は同じとする)に対して、次のような方針でニューラルネットワークを構築し、その性能を調べた(詳細は、小泉,1999a, Koizumi,1999を参照)。

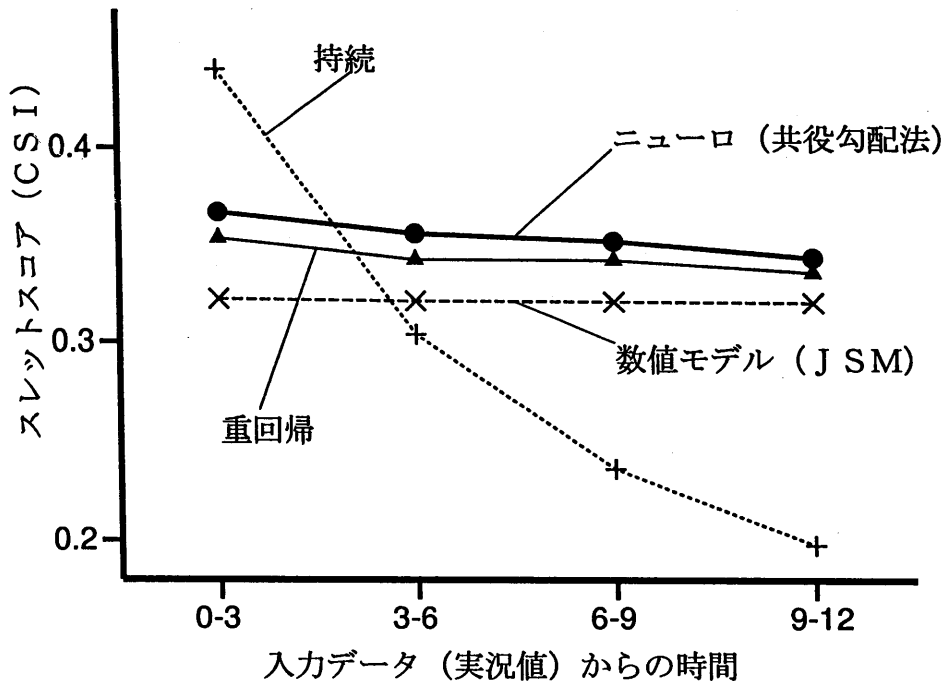
1. 一年分のデータを「教師データセット」として固定し、これの繰り返し学習を行うこととする。
2. バックプロパゲーション・アルゴリズムに代えて、共役勾配方向の直線探索というアルゴリズムを採用する。

ここで「共役勾配方向の直線探索」について少しだけ説明を加える(詳しくは専門書を参照されたい。また、小泉,1999aにも若干の解説がある)。本章第2節で紹介した通り、バックプロパゲーション法では、関数の最急降下方向に「少しだけ」パラメタを動かすのであり、パラメタを動かした結果の関数値については関知しない。これに対して「直線探索」では、与えられた方向(たとえば最急降下方向)に細かい探索を行なって極小点を探す。従って、バックプロパゲーション法は計算コストが低く、繰り返し1回あたりの計算時間が短いかわりに、最終的に極小点にたどりつくまでには繰り返し回数が多く必要となるのに対して、直線探索では、確実に極小点をたどりながらパラメタ空間全体における関数の極小点に向かうので、繰り返し計算の回数が少なくてすむかわりに計算コストは高く、繰り返し計算1回あたりの計算時間が長い。大規模ニューラルネットワークのような複雑な関数の場合は、後者の方が確実に極小点にたどりつけるという点で好ましい。(「最急降下方向」と「共役勾配方向」の違いについては煩雑になりすぎるので、ここでは触れない)

さて、本節のニューラルネットワークは、1993-1994年の2年間のデータで学習を行ない、1995年の1年分のデータで検証を行なった。付図1に、本節のニューラルネットワークの検証期間におけるスレットスコアを掲げた。スコアの計算にあたっては、出力値が0.5以上の場合に「降水あり」を予測したものとした。

このニューラルネットワークは、9-12時間先までJSMの予測を改善しているばかりでなく、同じ資料を用いてステップワイズ法で作成した線形重回帰式よりも高いスコアを示している。

このことから、少なくともここで扱ったような大規模なネットワークの場合には、共役勾配法の繰り返し学習という方法によらなければ、そのネットワークが持つ潜在的な能力を引き出すことは難しいことがわかる。逆に言えば、逐次学習を行なっているニューラルネットワークが芳しい成績を上げていない場合でも、それは必ずしもそのネット



付図1 共役勾配法で繰り返し学習を行なった場合のニューラルネットワークのCSI(スレットスコア)。持続予報, JSMの予想降水量のスコアの他, ニューロと同じ入力変数を用いて, 対象点ごとにステップワイズ法で作成した重回帰式のスコアも合わせて示した

ワークそのものの問題とは限らず, 学習アルゴリズムの選択の問題である可能性がある。

なお, 物理的な因果関係を考慮して作られている数値モデルの降水量予想値に対して, 単なる統計処理に過ぎないニューラルネットワークがより良い成績を示すのは不自然ではないか, という議論がある。これについては, 現時点では次のような回答を考えている。

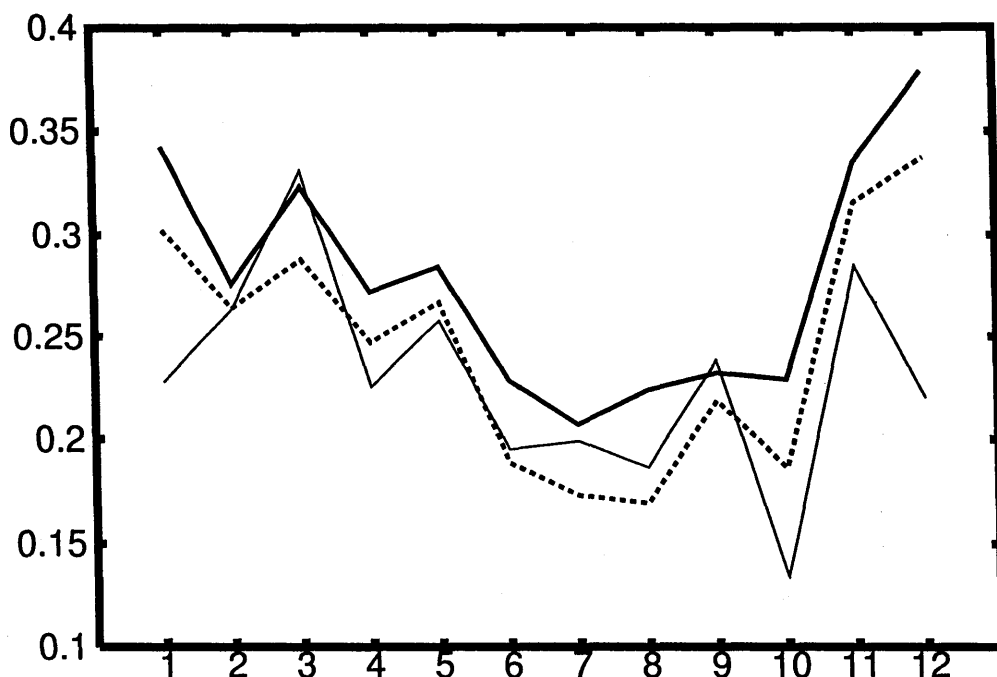
1. 地形による強制など, モデルで表現できないスケールの外的条件が, モデルで表現できるスケールの大気場と相互作用する場合は, モデル自体ではその相互作用を表現することはできないが, 統計処理を行なえば, **モデル内で表現されている大気場の情報だけからその相互作用について記述することが可能になる**。従って, 統計操作だけでもモデルの予想を改善することはありうる。
2. 付け加えれば, 本章で用いたニューラルネットワークはいずれも数値モデル出力だけでなく, 直前の実況観測データをも入力データとして用いており, 観測から得られた情報(主としてモデルの誤差に関する情報)を用いてモデルの予想を改善することができたものと思われる。

これらはいずれも仮説である。ニューラルネットワークを有効に使うためにも, ニューラルネットワークが具体的にどのように機能しているのか, ということについては, きちんと調査する必要がある。

B.2 降水量予測にかかわる諸問題

本稿では専ら「降水の有無」を予測する問題を扱ってきた。この問題は, 予測対象がカテゴリカルであるという点で扱いやすい面がある。しかし, 気象において予測の対象となるものの多くは連続量であり, その予測にはカテゴリ一予測の場合とは異なる困難が伴う。

筆者は最近の調査(小泉,1999b)において, 共役勾配法ニューラルネットを降水量予測に応用することを試みた。これまでのところ, 次のような結果を得ている(付図2参照)。



付図2 実況データから9-12時間後の降水量を予測した場合の月別スキルスコア。4層フィードフォワード線型出力ニューラルネットワーク(細実線), 3層ハイブリッド確率出力ニューラルネットワーク(太実線), チューニングを行なったステップワイズ重回帰(破線)のスコアを示した

1. 連続量の予測においては、ニューラルネットワークの出力層の応答関数をどのように設定するかが問題となる。降水量の場合は出力値を0以上に制約する、という意味では指数関数が適当とも思われるが、検証データにおける精度では良い結果が出なかった。精度の評価方法に依存する部分もあるが、降水量を適当な階級に分け、各階級に入る確率を予測する形にしたものが、最も良好な成績を示した。その際、階級の確率を $p = 1 / (1 + \exp(-x + \beta \log r))$ と表現する方法が有効であった(ただし、 p は降水量が r 以上である確率。 x は出力層への入力値。パラメータ β も学習によって調節する)。
2. 一般的に使われている3層(または4層)FF型のネットワークは、あまり精度が上がらなかった。3層FF構造に、入力層と出力層を直結するリンクを加えた構造(3層ハイブリッド構造)が有効であった。

これらの結果は、ニューラルネットワークの次のような性質に由来するものと思われる。

1. ニューラルネットワークは、直観的には、入力変数空間を中間層で分割し、その部分空間ごとに対応する出力値を与えるものと考えることができる。これは、入力変数の全値域にわたって一つの線型関係を与える重回帰式と大きく異なる点である。
2. 降水量は、0に突出した頻度のピークを持ち、値が大きくなるにつれて頻度が小さくなるという、偏った頻度分布を示す。このため、これに重回帰式をあてはめる場合は、回帰式の形が事例数の多い小雨のケースに支配されやすく、大雨のケースについては、小雨のケースから求めた線型関係を外挿するのに近い形となる。
3. これに対してニューラルネットワークの場合は、上記のように入力変数空間の分割が行なわれるために、小雨のケースと大雨のケースは別々に「関数のあてはめ」が行なわれる。
4. 一般に統計モデルをデータにあてはめる場合、事例数が多いほど頑健になる。これは、重回帰式もニューラルネットワークも同じである。従って、ニューラルネットワークの場合は、小雨を表現する部分については一般性が

高いが、大雨を表現する部分については一般性が低くなる、ということが起こっていると思われる。

5. 上に示した出力層の関数の形や、3層ハイブリッド構造は、本来独立でありうるはずの小雨事例と大雨事例とを人為的に結びつけるものである。これによって、非線型性を保持しつつ全体としての頑健性を補っているものと思われる。

以上のことは、さまざまな問題にニューラルネットワークを適用しようとするときに、対象となる問題の性質(ここでは予測対象の頻度分布)を把握して適切なネットワーク構造や応答関数を選択することが重要であることを示唆している。

参考文献

安西祐一郎, 1989: 認識と学習(岩波ソフトウェア講座16). 岩波書店

Domingos, P., M.Pazzani, 1997: On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning*, **29**, 103-130

Friedman, N., D. Geiger and M. Goldszmidt, 1997: Bayesian Network Classifiers. *Machine Learning*, **29**, 131-163

伊藤秀一, 1992: MDLのパターン認識への応用. 人工知能学会誌, **7**, 608-614

小泉耕, 1997a: ニューラルネットワーク. 天気, **44**, 51-55

小泉耕, 1997b: 大規模ニューラルネットワークは天気予報に使えるか?. 日本気象学会1997年度秋季大会予稿集

Koizumi, K., 1999: An Objective Method to Modify Numerical Model Forecasts with Newly Given Weather Data Using an Artificial Neural Network. *Weather and Forecasting*, **14**, 109-118

小泉耕, 1999a: ニューラルネットワークの構築法の改善について. 平成10年度全国予報技術検討会資料(気象研究所)

小泉耕, 1999b: ニューラルネットワークによる降水量予測(第2報). 日本気象学会1999年度春季大会予稿集

久間和生, 中山高(編著), 1992: ニューロコンピュータ工学. 工業調査会

Lee, J., R. C. Weger, S. K. Sengupta and R. M. Welch, 1990: A Neural Network Approach to Cloud Classification. *IEEE Transaction on Geoscience and Remote Sensing*, **28**, 846-855

中荃洋一郎, 古関義幸, 田中みどり, 1992: 確率モデルの学習方式と診断への応用. 人工知能学会誌, **7**, 862-869

Quinlan, J.R., 1986: Induction of Decision Trees. *Machine Learning*, **1**(1), 81-106

Rissanen, J., 1978: Modeling by Shortest Data Description. *Automatica*, **14**, 465-471

Sarle, W. S., 1998: How to measure importance of inputs? <ftp://ftp.sas.com/pub/neural/importance.html> より入手可能

Sethi, I.K., 1990: Entropy Nets: From Decision Trees to Neural Networks. *Proceedings of the IEEE*, **78**(10), 1605-1613

鈴木秀男, 圓川隆夫, 1995: MDL基準による判別木の生成. 人工知能学会誌, **10**, 572-579

西健司, 韓太舜, 1992: MDL入門:情報理論の立場から. 人工知能学会誌, **7**, 427-434

柳野健, 1998a: 気象の予測と研究のためのニューラルネットワークI. 天気, **45**, 865-872

柳野健, 1998b: 気象の予測と研究のためのニューラルネットワークII. 天気, **45**, 907-913