

感性情報を扱うための Prolog と ニューラルネットワークの結合 -Neuro-Prolog/II の作成-

今中 武 曾我真人 上原邦昭 豊田順一
大阪大学産業科学研究所

本稿では、論理的情報とユーザの主観によって決まる感覚的情報を1つのシステム内で同時に利用できるようにしたシステム Neuro-Prolog/II について述べる。Neuro-Prolog/II は、Prolog インタプリタ C-Prolog とニューラルネットワークシミュレータ SunNet の結合システムであり、Prolog とニューラルネットワークは、それぞれ論理的情報、感覚的情報を扱うために用いられる。両者の結合は、N規則という新たな概念を導入して行った。N規則は、Prolog と同様にヘッド部とボディ部に分かれており、それぞれが出力層、入力層に割り当てられたニューラルネットワークで実行される。Neuro-Prolog/II では、Prolog の規則とN規則が自由に呼び出し合えるようになっている。

An Integration of Prolog and Neural Networks to Deal with Sensibility in Logic Programs

Takeshi IMANAKA, Masato SOGA, Kuniaki UEHARA
and Jun'ichi TOYODA

The Institute of Scientific and Industrial Research
Osaka University, 8-1 Mihogaoka, Ibaraki 567, Osaka, Japan

We have developed Neuro-Prolog/II which integrates a Prolog interpreter C-Prolog and a neural network simulator SunNet. By using Neuro-Prolog/II, users can handle both definite information like logical rules and indefinite information like sensibility concurrently in a single system. In Neuro-Prolog/II, the definite information is handled in Prolog and indefinite information is handled in neural networks. The two systems, Prolog and neural network, communicate with each other by execution of P-rules and N-rules. A P-rule is a conventional logical rule in Prolog. An N-rule is newly proposed for Neuro-Prolog/II to activate its corresponding neural network. Since, both P rules and N rules can activate each other in Neuro-Prolog/II, problems containing both logical information and sensibility can be handled by Neuro-Prolog/II.

1. まえがき

近年、人工知能の分野では、論理的に記述困難な問題を扱うための研究が行われている。特に、「好き」、「美しい」などの言葉で表される人間特有の感覚に関する情報(以降では感覚的情報と呼ぶ)を扱う試みがなされている[1,5]。しかしながら、Prologなど、従来の人工知能システム作成に用いられてきた言語では、「ある」、「なし」や真、偽といった2値で表現可能な情報を、明確に記述された規則と事実を用いて処理しているために、感覚的情報を上手く扱うことができないという欠点がある。これは、感覚的情報には、どの程度「好き」、「美しい」かなどを表す量(以降では感覚量と呼ぶ)が存在し、かつ感覚量を扱う関数の表現が困難になるためである。さらに、感覚量の計算には個人差が生じ、感覚量を入力に持つ関数を、あらかじめ固定的に決定したり、規則や事実を用いて明確に表現することも困難である。

一方、ニューラルネットワークには、①入力変数の多い複雑な関数でもそれを近似する能力がある、②入出力パターンを用いたネットワークの設定のみを行えば、入出力関係をあらかじめ明確な形式で定義しなくてもよい③ニューラルネットワークの設定を、動的に修正できるなどの利点がある。したがって、感覚的情報の持つ上記の特徴を上手く扱うことができる。しかしながら、ニューラルネットワークはPrologに比べ計算過程の説明が困難であるといった欠点があり、個々の知識を組み上げて1つの結論を得るような論理的情報の処理には不適當である。

以上のことから、論理的情報はPrologで扱い、論理的に扱うことが本質的に不可能に近い感覚的情報などはニューラルネットワークで扱うことが合理的である。たとえば、「年収の多すぎる社員には残業をさせない」といったような規則はif-thenの形式をしており、論理的に処理を行うべき情報であるが、ある社員の年収が多すぎるかどうかの部分は主観的判断に基づく感覚的情報である。したがって、「年収の多すぎる社員には、残業をさせない」というような規則ではある社員の年収が多すぎるかどうかをニューラルネットワークで計算し、年収が多すぎると判断された場合にPrologでこの規則を実行するメカニズムが合理的である。

我々はこのような考えに従い、ニューラルネットワークとPrologの結合システムNeuro-Prolog/IIを開発した。

Neuro-Prolog/IIは、PrologインタプリタC-PrologとニューラルネットワークシミュレータSunNetを結合したシステムで、現在ワークステーションSPARKstation330上で稼動中である。Neuro-Prolog/IIでは、ニューラルネットワークを利用して非論理的な感覚的情報を扱い、Prologを用いて論理的情報を扱っている。両者の結合は、[4]で示したニューラルネットワーク操作用語を用いて、Prolog上でメタインタプリタを作成して行った。メタインタプリタは従来のPrologと同様に後ろ向き推論を行い、推論途中で感覚的情報の処理が必要となれば、ニューラルネットワークを用いて計算を行う。このときニューラルネットワークの計算結果によっては、推論途中で発火される規則が異なることもあり、Neuro-Prolog/IIにおけるPrologとニューラルネットワークの結合は密なものとなっている。以下、2章でNeuro-Prolog/II作成の動機について述べ、3,4章でPrologとニューラルネットワークの結合方法と、Neuro-Prolog/II上でのプログラム実行方法を示す。また、5章では、Neuro-Prolog/IIの有効性を示すために作成した簡単な実験プログラムを示す。実験プログラムは、レンタカー会社の接客を想定しており、客の好みや目的に合うレンタカーを検索するシステムである。実験プログラムでは、客の好みを感覚的情報として扱い、レンタカーが客の目的に合致しているかなどの部分を論理的情報として扱っている。

2. Neuro-Prolog/II作成の動機

2.1 ニューラルネットワークと感覚的情報処理

本報告では、「ある」、「なし」や真、偽といった2値で表現可能なものを論理情報と呼び、「好き」、「豪華」といったように、どれくらい「好き」、「豪華」かなどの程度を伴い、かつ程度が主観的に決まるものを感覚的情報と呼ぶ。また、「好き」に対してどれくらい「好き」かの程度を感覚量と呼び、連続値[0,1]で表すことにする。特に、感覚量を計算する際には

(1) 個人差がある

- (2) 複数の要因から感覚量を計算する関数は非線形で複雑である
- (3) 感覚量の計算過程について人間から獲得できる情報は、入出力例のみである

といった特徴を考慮する必要がある。たとえば、自動車の内装に対していただく感覚「豪華さ」について考えると、ある人にとっては本皮シートの有無が「豪華さ」の評価に大きく影響するが、別の人にとっては少ししか影響しなかったりする。このように、主観的評価による感覚的情報には個人差が生じる。また、「豪華さ」を評価する場合、あまりにも色々装備されていれば、かえって煩雑で安っぽい印象を与え、「豪華さ」の評価が下がることがある。言い換えると、「豪華さ」の評価値を大きくするように作用していた要因が、他の要因の属性値によっては評価値を小さくするように作用することがある。したがって、感覚量の決定における各要因間の関係は非線形であり、要因から出力を求める関数は複雑になるため、予め固定的に定義することは殆ど不可能である。さらに、感覚量は「かなり好き」、「とても豪華だ」のように主観的評価の結果を言うことはできるが、その理由及び程度を決めた根拠は説明できない場合が多い。無理に根拠や評価過程を説明しようとしても、イ)各要因の重みは他の要因の値によって変化する、ロ)要因の属性値も重みも連続値となり、他の要因との関わりを簡単には示せないなどの点を考慮すれば、殆ど不可能である。以上のように、感覚量の計算には上記の(1)、(2)、(3)の特徴を考慮しなければならない。

以上のような感覚量の計算における3つの特徴は、ニューラルネットワークが持つ以下の3つの特徴と非常に整合性の良いものである。

- 1) バックプロパゲーション機能で、設定を動的に変化させることができる
- 2) 入力変数が多く、かつ複雑に関わり合った関数でも近似できる
- 3) 入出力例のみから、入出力関係を学習することができる

1)により、個人差に合わせて、ニューラルネットワークの設定を変えることができる。2)、3)により、入出力例のみから感覚量を計算する複雑な非線形関数を近似することができる。

また、ニューラルネットワークの導入手法には感覚量を出力ユニットに、感覚量を計算する際に要因となる属性を入力ユニットに割り当てる方法を用いた。たとえば、車の内装の「豪華さ」を本皮シート、パワーウィンドウなどの装備の有無から求めるニューラルネットワークは「本皮シート」、「パワーウィンドウ」の有無が入力ユニットに、「豪華さ」が出力ユニットに割り当てられる。この方法の有効性は、我々がすでに[2,3]で示した通りである。[2,3]では、例として計算機画面上に表示される色のRGB値と「派手な」などの色のイメージを対応付け、ニューラルネットワークを用いれば、人間の感覚量の計算を近似できることを確かめている。

2.2 Prolog とニューラルネットワークの結合

従来より、Prologを用いて論理的に扱える情報を処理できることが、よく知られている。Prologの規則は、ボディ部分にヘッドの真偽を導くための要因となる条件が記述される。ボディ部の条件は1つでも偽になればヘッドを偽とするAND関係、1つでも真であればヘッドを真とするOR関係のみの記述が許されており、各条件は独立に真偽が評価できる。したがって、独立に評価できる要因から論理的に導かれる問題のみが扱われる。しかしながら、現実問題では各要因を独立に評価できない感覚的情報が入り混じって用いられる。たとえば、「内装が豪華で、4人乗りであれば高級車である」といった規則を表現する場合、「内装が豪華か」どうかは、内装を構成する多くの要因が複雑に関わり合い、AND、OR関係では割り切れない。また、4人乗れるかどうかは車の規格から論理的に求められる。さらに、両者は高級車であるかどうかを決定する上で論理積を表すAND関係である。以上のように、現実問題を考えた場合、AND、OR関係から論理的に求められる情報と感覚的情報は入り混じって扱われる。したがって、論理情報の処理機構と感覚的情報の処理機構を1つのシステム内に結合する必要がある。

3. Neuro-Prolog/IIの結合方式

3.1 結合方式の概要

Neuro-Prolog/IIのシステム構成を図1に示す。Neuro-Prolog/IIのメタインタプリタで解釈、実行されるのは、従来のPrologプログラムとニューラルネットワークにより計算される規則が入り混じったものである。ニューラルネットワークにより計算される規則は、Prologの規則と区別するためにオペレータ":~"を用いて記述される。以降では、Prologの規則とオペレータ":~"で定義された規則をそれぞれ、P規則、N規則と呼ぶことにする。Neuro-Prolog/IIでは、N規則ごとに1つのニューラルネットワークが用いられ、推論中にN規則が起動されると、N規則に対応づけたニューラルネットワークが起動される。すなわち、複数のニューラルネットワークから適切なニューラルネットワークが推論により動的に選択、実行されることになる。また、図1に示すように、各ニューラルネットワークには設定手続きプログラムが付加されており、各ニューラルネットワークの設定はこのプログラムを起動して行なわれる。

3.2 N規則

Neuro-Prolog/IIでは、感覚的情報を扱う部分がN規則で記述される。N規則はP規則と同様、ヘッドとボディを持っており、

ヘッド:~ボディ、

の形式をしている。ヘッドと単一化できるゴールが実行されると、P規則と同様にボディがサブゴールと見なされて実行される。ボディ部に複数の述語が存在する場合、P規則ではサブゴールのうち1つでも偽が見つければ、規則の実行をやめるAND関係、1つでも真であれば規則の実行を成功させるOR関係がある。一方、N規則のボディ部では、サブゴールをすべて実行して各サブゴールの評価値を連続値[0,1]の範囲で求め、各評価値を入力と見なしてニューラルネットワークを起動する。ニューラルネットワークを起動した結果、得られる出力はN規則のヘッドに評価値として連続値[0,1]の範囲で割り当てられる。このとき、ニューラルネットワークの入力となるボディ部にP規則で定義された述語がある場合、真偽はそれぞれ値"1","0"として評価している。また、N規則で評価値の求められたヘッドがP規則のボディ部に条件として記述されている場合、評価値がしきい値0.5以上の場合は真、0.5未満の場合は偽と見なしている。

次に、N規則の別の使い方としてボディ部に直接評価値を記述したものを示す。これは、P規則の事実に対応するものである。たとえば、P規則では車1がスポーティであることが真であれば、

スポーティ(車1):-true.

のように記述する。N規則の場合も感覚的に評価して、車1が[0,1]の範囲で0.7ぐらいスポーティであるという事実を同様の形式で、

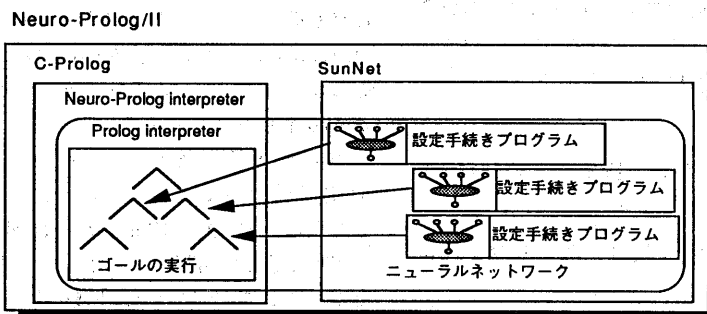


図1 Neuro-Prolog/IIのシステム構成

スポーティ(車1):~0.7.

と記述する。

さらに、Neuro-Prolog/II では述語と評価値の関係を扱う述語 "value" を新たに導入している。述語 "value" は (a) 利用者が定義した場合と (b) ゴールとして実行した場合で意味が異なる。(a) の場合は、第2引数の数値を第1引数の述語に対する評価値にする。(b) の場合は、第1引数の述語の評価値を第2引数に求める。たとえば、以下のデータベースがあるものとする。

試験結果(英語, 80).

試験結果(国語, 70).

試験結果(数学, 90).

この場合に、"value" 述語を用いて

value(点数(X), H) :- 試験結果(X, T), H is T/100.

を定義すれば、述語 "点数(X)" の評価値にニューラルネットワークの入力として用いることのできる範囲の連続値 [0, 1] が割り当てられる。この用い方は、上記の (a) の場合である。

また、豪華さを求めるN規則がすでに定義されている時に、述語 "value" を用いて、2つの物X、Yに対して、Xの方がより豪華かどうかを求める述語を以下のように定義できる。

より豪華(X, Y) :- value(豪華さ(X), H1),

value(豪華さ(Y), H2), H1 > H2.

このP規則では、ボディ部の述語 "value" によって、X、Yの豪華さがそれぞれH1、H2に求められ、H1、H2の大小比較を行うことによって、より豪華かどうかを定義している。この用い方は、上記の (b) の場合である。

4. Neuro-Prolog/IIにおけるプログラムの実行

4.1 ニューラルネットワークの構成

Neuro-Prolog/II は、Prolog インタプリタと同様にプログラムを内部データベースに格納してから実行する。この際に、Neuro-Prolog/II のメタインタプリタは各N規則の実行に必要なニューラルネットワークを構成する。構成されるニューラルネットワークは、N規則のボディにある2種類のオペレータ ":", ";" によって異なる。オペレータ ":" は、N規則のボディにおける各属性が独立な場合を表わす。ここで、独立とは出力値を決める際に各入力属性の重みが互いに干渉しない場合である。たとえば、「文章の見やすさ」を決定する属性を「内容の平易さ」、「印刷の美しさ」とする場合、「内容の平易さ」が変化することによって「印刷の美しさ」の重要性が大きくなったり、小さくなったりはしない。このように、独立属性として定義された入力ユニットどうしはオペレータ ":" を用いて

文章の見やすさ :- 内容の平易さ ; 印刷の美しさ.

と定義し、この記述に対しては共通の中間層を持たないように図2(a)の形式でニューラルネットワークが構成される。図2(a)の形式をしたニューラルネットワークでは、各入力値の加法的評価値が出力値に割り当てられ、独立な属性から感覺量を計算する際に適している。

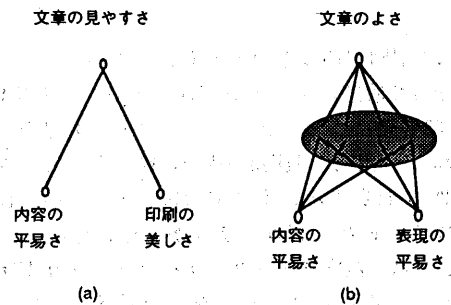


図2 ニューラルネットワークの構成

一方、オペレータ ";" は、各入力属性が非独立な場合を表わす。各入力属性が非独立とは、各入力属性の出力値決定に関わる度合いが互いに干渉する場合である。たとえば、「文章のよさ」を決定する属性を「内容の平易さ」、「表現の平易さ」から判断する場合、一般には「表現の平易さ」の値は大きい方が、「文章のよさ」の値は大

きくなる。しかしながら、「内容の平易さ」の値が小さい、すなわち内容が難しい場合に「表現の平易さ」があまり大きいと、難しい内容を、あまりにも幼稚に表現することは好ましくないため、「文章のよさ」の評価値は小さくなることがある。このように非独立な属性は図2(b)のような3層のニューラルネットワークで構成する。

4.2 ニューラルネットワークの設定

4.2.1 設定方法に関するオプション

Neuro-Prolog/II は、N規則の実行に用いるニューラルネットワークの設定について、分割設定オプションと実行時設定オプションの2つを提供している。Neuro-Prolog/II 上でプログラムを起動する際に分割設定オプションを指定すると、プログラム実行前にすべてのN規則が検索され、それぞれに対応するニューラルネットワークを設定するための処理が行われる。設定順序は、呼び出し関係にある複数のN規則がある場合、呼び出される規則のニューラルネットワークから先に設定が行われる。たとえば、Neuro-Prolog/II 上で開発された応用プログラム中に

(規則1) a:~b,c.

(規則2) b:~e.

(規則3) c:~f,g.

なるN規則があるものとする。この時、プログラムを分割設定オプションで起動すると、規則2、規則3の順にニューラルネットワークの設定が行われる。規則2、規則3の設定が終了すると、次に規則1を設定する。すべてのN規則について設定が終了すると、プログラムが実行される。この方法では、設定フェーズと実行フェーズが明確に区別されるために、システムの見通しが良くなるといった利点がある。しかしながら、プログラム中のすべてのN規則に対して設定を行うために、推論時に実行されないN規則に対するニューラルネットワークについても設定することになり、非効率な面がある。

実行時設定オプションは、実行時に必要に応じてニューラルネットワークを設定するものである。この方法では、N規則が起動されるたびに規

則に対応するニューラルネットワークが設定済みであるかどうか調べ、設定されていない場合は、設定を行う方法である。実行時設定オプションを用いると、実行に必要なニューラルネットワークのみを設定できるために、分割設定オプションに比べて効率的である。しかしながら、プログラムの実行途中で、幾度か実行が中断されるといった欠点もある。Neuro-Prolog/II では以上の2通りの設定手法のいずれもプログラムの実行時にオプション指定できるようになっている。

4.2.2 設定手続きプログラム

Neuro-Prolog/II を用いて応用プログラムを作成した場合、利用者の感覚量の計算を近似するために、利用者から入出力例を獲得し、ニューラルネットワークに設定しなければならない。しかしながら、ニューラルネットワークの入出力になっている述語の評価値を、利用者へ直接入力させることは、利用者がプログラムの内容を知っているわけではないので非現実的である。したがって、Neuro-Prolog/II では利用者へ理解しやすい形式で入力させ、さらにその値を述語の評価値に変換するための設定手続きプログラムという新たな概念を導入した。設定手続きプログラムは、ニューラルネットワークごとに定義するProlog プログラムであり、ニューラルネットワーク設定時に呼び出されるものである。設定手続きプログラムは、各ニューラルネットワークが計算する感覚量に合わせて、利用者からの入出力例の獲得方法を規定することができる。たとえば、「文章のよさ」を求めるN規則に対する設定においては、利用者へ評価値[0,1]を入力させるのではなく、{非常によい, よい, まあよい, どちらでもない, 少し悪い, 悪い, 非常に悪い}といった選択肢を用いて入力させるなどの方法を採用することができる。

設定手続きプログラムを作成する際には、パターンファイルを操作する組み込み述語が利用可能になっている。パターンファイルとは、ニューラルネットワークごとに設定用の入出力例を保存するもので、SunNetにより提供されているものである[8]。組み込み述語には、入出力例の追加、削除、更新のための述語を用意した。これらの述語については、[3,4]で述べているために詳しい説明は省略し、図3に4.1節の「文章のよさ」を計算するニューラルネットワークに付加

した設定手続きプログラムの例を示す。図3の設定手続きプログラム中では、"value" 述語の実行によって、別のニューラルネットワークが起動される場合があるが、分割設定オプションの場合、前節で述べた設定順序となっているために、未設定のニューラルネットワークが起動されることはない。また、実行時設定オプションの場合、"value" 述語実行途中において、未設定のニューラルネットワークが起動されれば、そのニューラルネットワークから先に設定を行う。

```

set_up-
  value (内容の平易さ,V1),
  value (表現の平易さ,V2),
  get_adv(V1,W1),
  get_adv(V2,W2),
  write(内容の平易さは),
  write(W1), nl,
  write(表現の平易さは),
  write(W2), nl,
  write('のときに、この文章はどのような
文章であると思うか?'),nl,
  write('とてもよい,良い,どちらでもない,
悪い,ひどく悪い'),
  nl,get_value(V3,W3),
  add_pattern ([V1,V2],[V3]),
  write('とてもよい,良い,どちらでもない,
悪い,ひどく悪い'),nl,
  active_net(learn).
get_adv(X,とてもよい):-0.8<X,1.
get_adv(X,良い):-0.6<X,1.

```

図3 設定手続きプログラムの例

4.3 プログラムの実行

P規則とN規則が入り混じって実行される場合について説明する。呼び出し関係にある2つの述語の定義においてP規則,N規則が組み合わせられている以下の8通りを考える。

- (1) a:-b. (2) a:-b. (3) a:~b. (4) a:~b.
 b:-c. b:~c. b:-c. b:~c.
 (5) a:-b. (6) a:-b. (7) a:~b. (8) a:~b.
 a:-c. a:~c. a:-c. a:~c.

(1)はP規則のみの組み合わせであり、従来のProlog 処理系と同様の動作をする。(2)は、a を実行すればbがサブゴールとして呼び出され、b をヘッドに持つN規則を実行するためのニューラルネットワークが起動される。ニューラルネットワークの計算はcの評価値を入力として実行さ

れ、出力値がbの評価値と見なされる。bの評価値がしきい値0.5を越えたならば、bが真であると見なされ、aの評価値はP規則の導出により真となる。(3)は、a を実行すれば、aの評価値を計算するニューラルネットワークの入力であるbの評価値を求める。bの評価値はP規則で導かれ、真であれば"1"、偽であれば"0"となる。(4)は、N規則が順に実行され、ニューラルネットワークの計算が階層的になっている場合である。この場合、cの評価値を入力としてbの評価値が、bの評価値を入力としてaの評価値がそれぞれ別々のニューラルネットワークで計算される。(5)はaの定義をP規則のみで行なったものであり、従来のPrologと同様の意味を持つ。(6)はaが論理的にbで定義されるが定義できない場合には、ニューラルネットワークで計算することを示す。(7)はbを入力とするニューラルネットワークでaの評価値を計算した結果、しきい値を越えないならば、論理的に真偽で定義することを示す。(8)はaの評価値がbを入力とするニューラルネットワーク、もしくはcを入力とするニューラルネットワークで定義できることを表わしている。

以上のP規則、N規則の組み合わせのうち、(1),(5)については従来のPrologと同じであるために、まず(2),(3),(4)について用例を示す。

- (規則1) 高級車(X) :- 内装が豪華(X),
 (規則2) 内装が豪華(X) :- 内装の豪華さ(X).
 (規則3) 内装の豪華さ(X) :- フル装備(X),
 内装の程度(X),
 (規則4) フル装備(X):- パワーステアリング(X),
 パワーウィンドウ(X),

規則1と規則2、規則3と規則4、規則2と規則3がそれぞれ上記の組み合わせ(2),(3),(4)に対応する。この用例は、ある車"X"が高級車かどうかを求めるプログラムであり、高級車を内装が豪華かなどの条件で定義し、内装が豪華かどうか、内装の豪華さについてはN規則で定義している。特に、規則2についてはニューラルネットワーク設定時に出力値を"1","0"の2値に設定することによって、「内装の豪華さ」がある特定の値より大きいときには、「内装が豪華」を真とするように設定する

しており、4人でスキーに行く客に対して、利用者に合う車をレンタカー会社の社員に替わってデータベースから検索するものである。以下にプログラムの一部分を示す。

利用者に合う車(X):- 割安(X),

スキーに行ける(X),

4人乗り(X),好みに合う(X).

スキーに行ける(X):- 4輪駆動(X),

スキーを運べる(X).

スキーに行ける(X):- 前輪駆動(X),チェーン(X),

スキーを運べる(X).

スキーを運べる(X):- キャリア付き(X).

スキーを運べる(X):- バンタイプ(X).

割安(X):- 車格(X),値段(X).

好みに合う(X):- 内装(X),外装(X),エンジン(X).

内装(X):- パワステアリング(X),

ステレオ(X), エアコン(X).

外装(X):- アルミホイール(X),

フォグランプ(X).

エンジン(X):- 排気量(X),出力(X).

検索の際には、分割設定オプションを用いた。実験プログラム中では、スキーに行けること、4人乗れることを論理表現で扱い、残りの条件は、割安で好みに合うといった感覚的情報で表現している。割安かどうかは車格と値段から、好みに合う度合は内装、外装、エンジンに対する評価から求める。以上のプログラムを実際に起動したところを図4に示す。図4の左右のウィンドウは、それぞれ別々の利用者に合わせて設定した場合を示しており、同じ検索プログラムでも、各利用者の感性における個人差をネットワークが計算しているために、別の車が検索されている。また、計算過程を利用者に示し、利用者個人の感覚を反映していない部分を指摘させれば、その部分を計算するN規則の追加設定を行うことが可能である。その結果、利用者の要求に合うようにニューラルネットワークの設定を修正することができる。

6. むすび

本論文ではPrologとニューラルネットワークの結合システムNeuro-Prolog/IIについて述べた。Neuro-Prolog/IIの開発によって、論理情報と感覚的情報の入り混じった問題を扱うことができるようになった。Neuro-Prolog/IIでは、主観によって決められる「好みに合う度合」などの程度を表す値を感覚量として扱い、Prologの述語に評価値として割り当てた。したがって、従来のPrologでは述語の評価値が真偽の2値になっているのに対し、Neuro-Prolog/IIでは感覚量を扱う部分は述語の評価値が連続値[0, 1]になる。Neuro-Prolog/IIと同様にPrologの述語に真偽のみでなく、連続値[0, 1]を割り当てて推論を実行するものに、ファジィ推論の枠組みがある。ファジィ推論では、推論実行時に用いられる操作がmin関数など、予め定めた固定的な関数であり、個人差を含む感覚量の扱いは困難である[†]。さらに、固定的な関数を予め個人に合わせて定義することもできるが、人手によって関数化するために入力となる属性が3つ以上になると、出力と合わせて4変数以上の式となり、関数の決定自身が非常に困難な作業となる。すなわち、ファジィ推論は、データや推論時に用いる知識のあいまいさを確信度として数学的に扱うことを目的としたもので、感覚的情報の処理に関しては、Neuro-Prolog/IIの方が優れている。

また、要因となる属性値の線形和で出力値を表わせることが分かっている場合に有効な方法として、多変量解析的アプローチがある[7]。この場合は、Neuro-Prolog/IIにおいてN規則のボディ部が";"のみから記述される場合と同等である。多変量解析的アプローチを用いれば、要因となる各属性の重みに対する数学的根拠付けなどの点で優れているが

- 1) システム内に多変量解析的アプローチの計算メカニズムを、完全に実現することは多大な労力が必要となる
- 2) 感覚的情報の処理では、評価値が属性値の線形和で求められる場合が少ない

[†] min関数はデフォルトとなっており、利用者が自由に変更できるが、いずれにせよ予め定めた固定的な関数である

などの点を考慮して、Neuro-Prolog/II では導入していない。

最後に、Neuro-Prolog/II の開発に伴い副産物として得られた利点を示す。Neuro-Prolog/II は、N規則によってニューラルネットワークを起動するために、複数のニューラルネットワークから適切なものを推論メカニズムで動的に選択し、起動することができる。これにより、関連する属性のみを入力に持つコンパクトなニューラルネットワークを複数個準備することができる。このように、ニューラルネットワークをコンパクトにすることは、設定に用いる入出力例の数を少なくすることに役立つ。たとえば、図5において、(a) のような4入力ニューラルネットワークを (b) のように、2入力ニューラルネットワーク3つに展開できれば、入出力が $\{0, 0.5, 1\}$ の3値と少ない場合でも、考え得るすべての入出力例は、(a) で 3^4 、(b) で $3^2 \times 3$ 通りとなり、扱うべき問題の空間が小さくなる。現実問題においても階層化可能な問題は数多くあり[9]、たとえば、車の例ではバンパーの色とハンドルの好みは、完全に独立要因であり、別々のニューラルネットワークで扱うべきである。以上のように、Neuro-Prolog/II はニューラルネットワークの設定に用いる入出力例の減少といった面からも有効である。

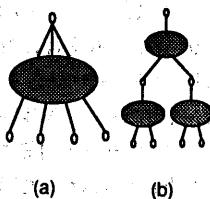


図5 ネットワークの階層化

参考文献

- [1] 原島博: 知的画像符号化と知的通信, テレビジョン学会誌, Vol.42, No.6, pp.519-525(1988).
- [2] 今中, 東坂, 曾我, 上原, 豊田: ニューラルネットワークを利用したイメージを表す自然語入力からの色出力システム, 情報処理学会全国大会, 3D-4, pp.404-405(1989).
- [3] Imanaka, T., Soga, M., Uehara, K. and Toyoda, J.: An Integration of Prolog and Neural Networks to Deal with Sensibility in Logic Program, ICSI '90, IEEE Computer Society (to be appeared).
- [4] 今中, 上原, 豊田: Prolog とニューラルネットワークの結合システム Neuro-Prolog, 情報処理学会, 知識工学と人工知能研究会, AI66-4, Vol.89, No.78(1989).
- [5] 片寄, 今井, 井口: 音楽における感性情報抽出の試み, 人工知能学会誌, Vol.3, No.6, pp.748-754(1988).
- [6] 金井, 石塚: Prolog-ELF: ファジ理論を組み込んだ Prolog, 情報処理学会論文誌, Vol.27, No.4, pp.411-416(1986).
- [7] 栗田, 下垣, 加藤: 主観的類似度に適応した画像検索, 情報処理学会論文誌, Vol.31, No.2, pp.227-237(1990).
- [8] Miyata, Y.: A Tool for Constructing, Running, and Looking into a PDP Network in a Sun Graphic Window, ICS Report 8708(1987).
- [9] 小橋康章: 決定を支援する, 認知科学選書 18, 東京大学出版会(1988).
- [10] 馬野元秀: ファジ集合の概念を用いた FS-Prolog について, 信学技法, COMP86-81, pp.87-95(1986).