

土井 晃一, 佐川 浩彦, 田中 英彦

東京大学 工学部

文京区本郷 7-3-1

内容梗概

本論文ではニューラルネットワークを用いた隠喩理解の方式を提案する。一つの隠喩には多くの意味があり、状況、文脈に応じて有力な意味に絞り込むことが重要である。

本方式では入力文は命題に分解され、単語間の意味変化は結合係数の変化として扱われ、混合理論に基づき意味が選択される。その結果ネットワークの構造が簡単になり、収束も早くなる。学習済みのネットワークを用いた実験では良好な結果が得られている。

1 はじめに

自然言語理解を計算機上で行なうためには、単語のニュアンスの問題を含んだ言外の意味の解析、状況理解、文脈理解、話者の認識の仕方の解析などが必要となる。我々は隠喩理解という範囲内でこれらを取り扱う。隠喩理解によって単語の新しい意味や新単語の意味の解析が可能になる。

隠喩理解では連想網の実現が普通の文の理解よりも重要である。本論文では、隠喩理解における連想網の実現をニューラルネットワーク [1] で行なう方法を提案し、さらに計算機上での実験結果について述べる。

言内の意味だけではなく、言外の意味を発見しないと会話の理解はできない [2]。

ここで言う文脈理解とは、外の世界の状況とは無関係に、前後の文章あるいは全体の文章によって文の解釈が変わる場合のことをいう。文脈理解の中には、話し手と聞き手の心の状態、共有知識、先行する発話などが入ってくる。文脈理解も自然言語理解には不可欠である。

一方、その場の状況理解とは、文章の中の世界だけではなく外の世界とのつながりを持つことの理解を指し、話題と直接関係ないことが発話されたときそれが話題とどう関係するのかを調べることである。話者をとりまく世界に関する情報はすべて状況理解と関係する。世界に関する記述は、特別の指摘のない限り、話し手の発話をそのまま信用して取り込むことにする。

新しい状況に対する話者の認識の仕方の解析とは、現実の世界を話者がどの様に理解したかを示すという意味である。話者が未知の対象について陳述しようとする場合、既知の表現では表現しきれないから、その話者に固有の表現が発話されることになる。そうするとその対象に対する話者の認識の仕方が表れることになる。このように隠喩理解を行うことにより、人間の心理的な側面に関する研究を行うことができるようになる。

隠喩理解を行うことにより、既存単語の新しい意味、新単語の意味の解析が可能になる。例えば「プログラムが走る」という例を挙げると「走る」とい

土井 晃一, 佐川 浩彦, 田中 英彦

東京大学 工学部

文京区本郷 7-3-1

内容梗概

本論文ではニューラルネットワークを用いた隠喩理解の方式を提案する。一つの隠喩には多くの意味があり、状況、文脈に応じて有力な意味に絞り込むことが重要である。

本方式では入力文は命題に分解され、単語間の意味変化は結合係数の変化として扱われ、混合理論に基づき意味が選択される。その結果ネットワークの構造が簡単になり、収束も早くなる。学習済みのネットワークを用いた実験では良好な結果が得られている。

1 はじめに

自然言語理解を計算機上で行なうためには、単語のニュアンスの問題を含んだ言外の意味の解析、状況理解、文脈理解、話者の認識の仕方の解析などが必要となる。我々は隠喩理解という範囲内でこれらを取り扱う。隠喩理解によって単語の新しい意味や新単語の意味の解析が可能になる。

隠喩理解では連想網の実現が普通の文の理解よりも重要である。本論文では、隠喩理解における連想網の実現をニューラルネットワーク [1] で行なう方法を提案し、さらに計算機上での実験結果について述べる。

言内の意味だけではなく、言外の意味を発見しないと会話の理解はできない [2]。

ここで言う文脈理解とは、外の世界の状況とは無関係に、前後の文章あるいは全体の文章によって文の解釈が変わる場合のことをいう。文脈理解の中には、話し手と聞き手の心の状態、共有知識、先行する発話などが入ってくる。文脈理解も自然言語理解には不可欠である。

一方、その場の状況理解とは、文章の中の世界だけではなく外の世界とのつながりを持つことの理解を指し、話題と直接関係ないことが発話されたときそれが話題とどう関係するのかを調べることである。話者をとりまく世界に関する情報はすべて状況理解と関係する。世界に関する記述は、特別の指摘のない限り、話し手の発話をそのまま信用して取り込むことにする。

新しい状況に対する話者の認識の仕方の解析とは、現実の世界を話者がどの様に理解したかを示すという意味である。話者が未知の対象について陳述しようとする場合、既知の表現では表現しきれないから、その話者に固有の表現が発話されることになる。そうするとその対象に対する話者の認識の仕方が表れることになる。このように隠喩理解を行うことにより、人間の心理的な側面に関する研究を行うことができるようになる。

隠喩理解を行うことにより、既存単語の新しい意味、新単語の意味の解析が可能になる。例えば「プログラムが走る」という例を挙げると「走る」とい

う単語に物理的に物体が走るという意味が従来あったが、これに「実行される」という新しい意味が付け加えられたことになる。つまり隠喩理解は新知識獲得の一手法でもある。

また単語のニュアンスの問題を扱うことを目的として機械翻訳にも活用できる。新聞、雑誌など毎日多量に生産される文献は人間の手を介して翻訳するには量的に無理があるので、翻訳の質向上に機械翻訳の出番となるが、各国語の文化的背景を持った隠喩は頻繁に出現するものであり、また新しい隠喩は次々に誕生し、単語の意味もどんどん変わっていく、このような状況下では隠喩理解は不可欠の要素になる。また現在の研究に役立つ点としては、言語処理の分野でシソーラスを作る際に単語の意味を確定するのに必要となることが挙げられる。

隠喩にはいわゆる「死んだ隠喩」と「生きた隠喩」とがある。前者は定型句化してその隠喩の持つイメージがほとんど残っていないものをいう。それに対して後者はまだ定着していない隠喩で、時間が経つと廃語になる可能性を持っているものをいう。前者の例として「机の脚」、後者の例として「ホーナー効果」が挙げられる。「死んだ隠喩」に対しては定型句化した扱いが可能であるが、「生きた隠喩」に対しては特別な扱いが必要となる。先に述べたように新聞、雑誌などの機械翻訳の際には、「生きた隠喩」が頻繁に現れ、その場で使い捨てにされる隠喩も多数出現する。このような隠喩は人間が処理するにはあまりにも量が多すぎる。どんな隠喩が現れても自動的に処理できなければこのような機械翻訳は実現できない。本論文では「生きた隠喩」を計算機上で理解する方法について議論する。

隠喩理解について今までなされた研究は、文献[3][4][5]等がある。

文献[3]では”That man is a baby.”という例を挙げ、is や a 等も含んだ文成分各々を一ノードにおいて、収束時に活性化値の大きいノードが文の主要な意味を表すとした。しかしこのように文の全ての要素をネットワークの上に載せようとする、ネットワー

クが複雑になり、収束も遅くなる。

また文献[4]の研究は”John gave Mary cold”という例を挙げ、この文が”John gave Mary money”と構文的に似ていることを利用してこの文の理解をlispを使って逐次的な方法で試みた。しかしこの方法では、「人間は狼である」という隠喩例では逐次方式のために検索する範囲が広くなり過ぎてうまく扱えない。

文献[5]では視点表現を導入し、さらに属性値集合の情報量を使って概念に属性を与えることによって意味を算出する。

我々は文を命題に分解して、命題単位でニューラルネットワークの各ノードに入力する。これによりネットワークの簡略化、検索範囲の狭小化、高速化ができる。また概念と属性の間にも間接的にも接続があれば意味の算出ができる。文の命題への分解は必ずしも完全でなくても良く、自立語の組みを生成し、このニューラルネットワークで意味を算出し、命題分解にフィードバックをかけることによりさらに完全な命題分解ができることになる。

我々の隠喩理解モデルは相互作用説に基づいている[6]。相互作用説によると、隠喩ではたとえる語とたとえられる語とがお互いに影響しあい、意味を変化させる。

例えば「人間は狼である」という文では、「人間」も「狼」もその互いの意味が変化し、「残酷である」あるいは「孤独である」といった意味を帯びようになる。

ニューラルネットワークを使うことにより、このような単語の意味の変化を扱うことができる。さらに意味の変化を学習させることもできる。

我々は混合理論[7]に基づいた隠喩理解モデルを提案する。混合理論によると、

- (1) 多義個所に至ると、聞きては複数の解釈を算出する。
- (2) その中から、文脈を利用して最適の解釈を選ぼうとする。
- (3) 文が終るまでに多義性が解消しなかった時にも、

一つを選びそれに固執する。

(4) 選んだ解釈が後続の文脈に合わない時には、前の節の表層構造を想起し直して、新しい解釈を算出しようとする。

となる。

すなわち(1)の各々の解釈を別のニューロンに割り当て、ニューロンの活性値の大小により、優先順位をつけ、またさらに前の文章によって特定のニューロンのバイアス値を上げることにより、混合理論を実現する。

以下本論文の構成は第二章で我々の提案する隠喩理解システムの全体構成について述べ、第三章で命題分解の結果、必要となるノードについて述べ、第四章でニューラルネットワーク部の全体構成とその機能について述べ、第五章でSunNetを使った実験について述べ、第六章で隠喩理解ルーチンの動作について述べる。

2 隠喩理解システムの全体構成

我々の隠喩理解モデルは図1で示す通り、七つの部分からなる。このモデルは、スペルベルの象徴解釈のモデルに基づいている[8]。「人は石だ」という例で説明する。まず入力部でパズし、「人」と「石」は内部形式である"man"と"stone"に置き換えられる。

次に命題分解ルーチンで、パズされた文章を命題に分解し、(man, stone)という一つの組みにまとめられる。この組みが後にニューラルネットワークの入力として使われる。各々の命題は隠喩検出ルーチンで隠喩的要素を持つかどうか調べられる[9]。

隠喩検出ルーチンで隠喩的要素を持つと判定された命題は隠喩理解ルーチンに入り、その真の意味を探索される。真の意味の探索にはニューラルネットワークを使い、入力として命題の二つの要素である"man"と"stone"が使われる。出力としては入力によって活性化された"insensible"というノードが選択される。

"insensible"という意味が算出されたので、隠喩理解ルーチンは出力部に"man","stone"という一組み

の命題を出力する。出力部はこれを文章の形にして「人は無感動だ」という出力を出す。

3 命題分解ルーチン

入力部から入ってきた文はここで命題に分解される。ここで扱う文は単文に限るものとし、複文、重文は扱わない。石に関連した隠喩を例にとりて説明する。入力文と例えられたもの、その含意を付録に表1に示す。入力文は文献[10]を参照した。例えられたものが「石」である隠喩を選び、さらに例えるものが人である例を扱う。出典は直喩であるがここでは隠喩形に直した。さらに類義語は可能な限り一つの語で表し、同じノードに割り当てられることとした。

これらの入力で使われるノードは

概念層

人、石、鉄、気持ち身体、都会、断層、坂道、沈黙
計9

属性層

つまらない、感情の無い、重い、動かない、関係の無い、弾き返される、確かな、無知な、転げ落ちる、固い、邪魔な、はずんだ、意識の無い、黙って、退屈な、無言の、
蹴飛ばす、生きる、なかなか、起き上がる、見る、冷然と、尻目にかける、前、立つ、背を見せる、背に負う、感じる、熱心に、落ち込む、迂回する、座り込む、立ちすくむ、かけおける、眠りに落ちる、急に、生き返る、捨てる、食べる、別れる

計39

となる。これだけ広い範囲の文が扱える。

このように命題分解ルーチンで文法的問題を吸収する。隠喩理解ルーチンにはこのうち付属語部分を取り除いた、自立語部分だけを命題の組として送り出す。これらの例では「人」と「石」と自立語がニューラル・ネットワークの入力となることになる。

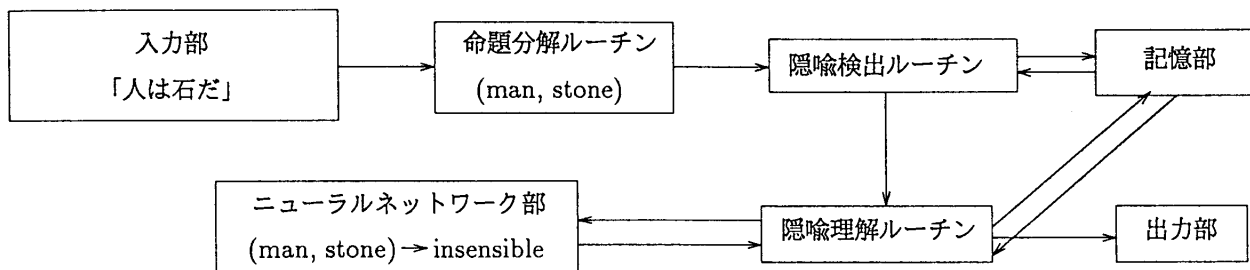


図 1: 隠喩理解システムの全体構成

「例えられたもの」が何であるかの判定も本当は必要であるが、これは指示代名詞の指す内容を求めることと本質的に同じことであるのでここでは指示対象は分かっているものとする。

4 ニューラルネットワーク部

我々は三層からなるニューラルネットワークによって隠喩理解を行なう [11]。そこでは連想の係数をニューラルネットワークの二つの層の間の重み (0 から 1 の値をとる) として扱う。ネットワーク全体は図 2 で示す。この図では伝搬は一方方向で示されているが、実際には双方向になるように二本のリンクが張ってある。図 2 では五つのノードにだけ単語が割り当てられていて、あとは何も書いてないが、新しい単語が入ってくると、その単語に新しいノードが割り当てられ、リンクの重みとバイアス値が新たに計算されることになる。

第一層を「概念層」とし、各々のニューロンは概念を表す。第一層内の二つのニューロンを入力として使う。「人は石だ」という例を取ると、「stone」と「man」が入力として使われる。

第二層は「属性層」である。各々のニューロンは属性を表す。第一層から第二層への接続はある概念とある属性の連想の係数を表す。

まず入力の man, stone に対応するニューロンは 1.0 に設定され、他のニューロンは 0.0 に設定される。ニューロン間の重み、バイアスは学習された値をロードする。

出力は属性層の活性化値を見て、もっとも大きい

ものに対応する属性を一つ選択する。後に解釈が文脈に合わない時には、次に活性化値が大きいノードに対応する属性を解釈として選択する。

最後に第三層は「感情層」である。「概念層」、「属性層」の伝搬を反映して、その発話でどのような感情が想起されるかを表す。この層を見ることにより、言外の意味が算出される。

入力に命題の組を入れ、文章の構成要素をすべて入れない理由はネットワークの簡略化、収束の早さ、検索範囲の狭小化による。従って文の構成要素を全て入力に与えて文全体を認識させるよりは、命題の組だけを与えることによって単語からどのような概念を連想されるかを調べるのにニューラル・ネットワークを用いた方がよい。

このように概念層、属性層をエネルギーが往復することによって、意味を探索する方法は文献 [12] の隠喩理解の理論とも相通じるところがある。グループ μ は、隠喩は二つの提喩に分離できると説明した。提喩とは全体と部分の関係から成り立つ比喩で、全体で部分を部分で全体を表す。グループ μ は全体と部分の分割の仕方を次の二つとした。

(II) 木=枝 かつ 葉 かつ 幹 . . .

(Σ) 木=ポプラ または 柏 または 柳 . . .

つまり Π 分解は全体をその構成要素に分解する分解法であり、 Σ 分解は全体をその各々の種類に分解す

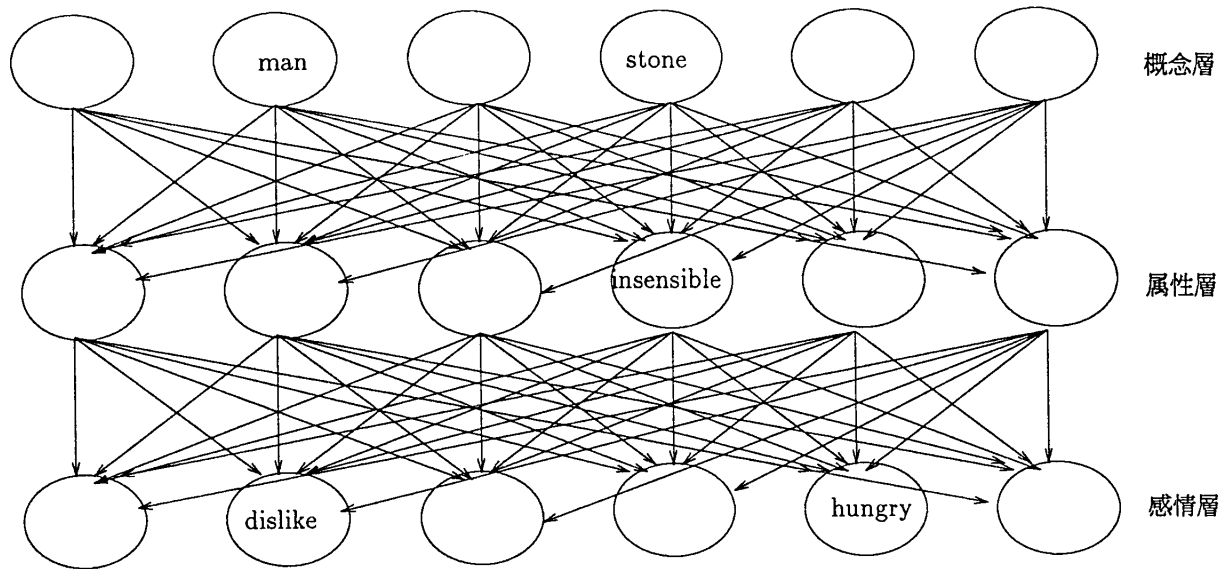


図 2: ネットワークの全体構成

る。これを「人は石だ」という例に当てはめてみると「石」を類の提喩によって「感情のないもの」と呼ぶ（一般化の提喩）。次に「感情のないもの」という類を逆に種の提喩で「人」と呼ぶ（特殊化の提喩）。この二つの動作によって隠喩の理解は行われるとした。ただしここに出て来る提喩は常識的な提喩ではないという難点を持つ。このように概念層と属性層の間でエネルギーのやり取りをする状態は、二つの提喩による人間の隠喩理解の方法のシミュレートとなる。

5 SunNet 上での実験

我々は計算機上での実験を SunNet を用いて行なった [13]。

SunNet のプログラミングは input, forward, activation, target, backward, delta, learn, learnbias によって組み立てられる。我々のシミュレーションではこのうち input, forward, activation を用いた。以下簡単にこれらの機能について説明する。

input はインプットバッファの内容を概念層にロードする。

forward は結合を通じてニューロンの活性化を行なう。a[i] を結合を通じて送られる i 番目のニューロ

ンの活性化値とし、w[i][j] を概念層と属性層との間のニューロン間の重みの値とすると、forward は全ての j に対して j 番目の属性層の活性化値 net[j] を

$$\Delta net[j] = \sum_i a[i]w[i][j]$$

の様に増加させる。

activation は属性層内の活性化値を計算する。a[i] を活性化値とし、net[i] を入力値とし、bias[i] を i 番目のバイアス値であるとし、f() を活性化関数とすると、activation は全ての i に対して、

$$a[i] = f(net[i] + bias[i])$$

$$logistic : f(x) = \frac{1}{1 + e^{-x}}$$

実験では logistic 関数を活性化関数として用いた。

我々の実験モデルは学習の終わった二層からなるネットワークによって行なった。

5.1 実験環境

入力の方法としては、自立語を逐次的に入れていく方法と、ある程度ためてから入れる方法の二通り

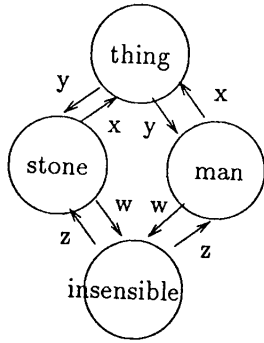


図 3: 「人は石だ」の隠喩理解(その一)

の方法が考えられるが、ここではまずある程度ためてから入力する方法を採用した。

まずシミュレーションの前に基本的な測定を行なった。図 3 で示すようなネットワークを組む。

図 3 は図 2 の一部を切り出してきたもので、 w, x, y, z の値を同じ値にして、0.1 から 0.1 刻みで 0.7 まで動かしてみた。バイアス値は全て 0.0 に設定した。入力値は最初だけ man と stone のノードだけ 1.0 にして、その後はネットワークの状態にまかせた。活性化関数は linear 関数を使うと収束しないので logistic 関数を選んだ。

結果は重みを大きくするほど、活性化値の収束値も大きくなり、両者の関係は比例関係にある。また重みを大きくするほど収束は遅くなった。thing と insensible の二つのノードは全く同じ振舞いをする。

5.2 活性化値の抑制

次に man と stone のノードの活性化値の収束値を抑制することを考える。これらのノードを抑制するためには thing と insensible からの重みを負にすれば良い。y と z の値を $-x$ にして、w と x の値を 0.3, 0.7, 1.0, 2.0, 3.0, 4.0 と動かしてみた。

thing と insensible の二つのノードが、man と stone のノードよりも活性化値は上がった。そして前者の二つのノードの活性化値は x の値が 1.0、つまり入力値と等しい場合最大になった。重みの値を大きくすればするほど、収束は遅くなる。結局入力値と重み

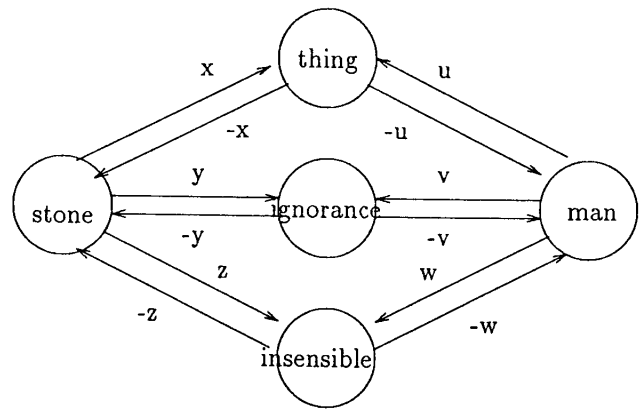


図 4: 「人は石だ」の隠喩理解(その二)

の値を等しくした時がもっとも良くなる。このことによって連想が実現されたことになる。

5.3 活性化値の選択的抑制

今度は図 3 の x と y の値を変えてみる。w と z の値はそれぞれ 1.0 と -1.0 に固定する。そして x の値を 0.3 から 4.0 まで変えてみた。y の値は $-x$ にしておく。

thing と insensible の活性化値に差異が現れる。最大の活性化値の値を持つノードを隠喩の意味と解釈できる。

5.4 属性ノードの数を増やした場合

次に入力のノードを二ノードにしたまま出力のノード数を三にした場合を実験してみる。図 4 で示すようなネットワークを組む。

図 4 の man と stone の初期値を 1.0 に設定し、x, y, z の値を全て同じ値にして 0.1 から 0.1 刻みで 1.0 まで動かす。収束は早く、高々 6 サイクルで収束する。その動的変化を図 5 に示す。

次に x, z, u, v, w の値は全て 0.7 に固定し、y の値だけを 0.1 から 0.1 刻みで 0.7 まで動かした。その結果を図 6 に示す。

insensible と thing に差異は現れず、ignorance の活性化値が 0.7 で insensible, thing と交わる。隠喩理解ルーチンが属性層の各ニューロンの活性化値を調べ、活性化値の一番大きいニューロンをその隠喩の

意味であるとするば、man と ignorance の結び付きが小さいときには thing か insensible を選択し、man と ignorance の結び付きが thing と insensible より大きくなると ignorance を選択することになる。前者の場合は「人は石だ」から「人は無感動だ」または「人はものだ」という意味を算出し、後者の場合は「人は無知だ」という意味を算出することができる。ニューロン間の重みは初期値を一般的な連想関係から生成し、さらに文脈、状況によって変化することになる。

今度は x を 0.5、 z, u, v, w を 0.7 に固定して y の値を 0.1 から 0.1 刻みで 0.7 まで動かす。その結果を図 7 に示す。

y の値が小さい間は、活性化値が大きい順に thing, insensible, ignorance であるが、0.5 で insensible と ignorance が逆転し、さらに 0.7 で ignorance と thing が逆転する。結合値の大きい順に活性化値が大きくなる。隠喩検出ルーチンは man と ignorance の結び付きにより、値の小さい方から ignorance, insensible, thing の順に意味を選択することになる。

次に x, u を 0.7 に z, w を 0.5 に固定して $y=v$ で v の値を 0.1 から 0.1 刻みで 1.0 まで動かす。その結果を図 8 に示す。

v の値が小さいうちは大きい順に thing, insensible, ignorance になるが 0.5 で insensible と ignorance が逆転し、さらに 0.7 で ignorance と thing が逆転する。

今度は u, x の値を 0.7 に、 w, z の値を 0.1 に固定して $y=v$ の関係を満たしながら y の値を 0.1 から 0.1 刻みで 1.0 まで動かす。その結果を図 9 に示す。

y の値が 0.1 のとき ignorance と insensible の値は等しく、 y の値が 0.7 になると ignorance と thing の値が逆転する。

次に x, u の値を 0.7 を z, w の値を 0.3 に固定して、 $y=v$ の関係を満たしながら y の値を 0.1 から 0.1 刻みで 1.0 まで動かす。その結果を図 10 に示す。

y の値が 0.3 未満では thing, insensible, ignorance の順であるが、 $y=0.3$ で insensible と ignorance が逆転し、 $y=0.7$ で thing と ignorance が逆転する。

以下の実験は重みの値を二つ同時に動かした場合である。 u, w, x, z の値を 0.7 に固定して、 $y=v$ の関係を満たしながら y の値を 0.1 から 0.1 刻みで 1.0 まで動かす。その結果を図 11 に示す。

次に x, u の値を 0.7 を z, w の値を 0.9 に固定して、 $y=v$ の関係を満たしながら y の値を 0.1 から 0.1 刻みで 1.0 まで動かす。その結果を図 12 に示す。

y の値が 0.7 未満では thing, insensible, ignorance の順であるが、 $y=0.7$ で insensible と ignorance が逆転し、 $y=0.9$ で thing と ignorance が逆転する。

以上の実験から insensible と各ニューロンの結び付きが小さくても十分動くことが確認された。ニューロン間の重みは広い範囲にわたって値をとることができる。次に u, w, x の値を 0.7 に z の値を 0.2 に固定して、 $y=v$ の関係を満たしながら y の値を 0.1 から 0.1 刻みで 1.0 まで動かす。その結果を図 13 に示す。

y の値が 0.4 未満では thing, insensible, ignorance の順であるが、 $y=0.4$ で insensible と ignorance が逆転し、 $y=0.7$ で thing と ignorance が逆転する。以上三つの実験から、重みの値を同時に二つ動かしても選択的な動作を行なえることがしめせた。

6 おわりに

本論文ではニューラルネットワークを用いた隠喩理解の方法を提案した。文脈理解の方法として心理学の混合理論、具体的な隠喩理解の方法として言語学の相互作用説、システムの全体構成としては文化人類学の象徴解釈モデルを利用した。ニューラルネットワークを連想網として使う方法について述べ、実際に SunNet 上でネットワークを組み連想網が実現することを示した。その結果ニューロン間の重みをいろいろに変えることにより、第二層の活性化値の収束値をより大きな結合に対してより大きな活性化値に変化させることができることがあきらかとなった。また命題分解を行なうことにより、ネットワークの構成が簡単になり、収束も早くなる。

ネットワークの学習には収束が早く、しかもバイアス値、重みが大きく変るような学習方法が必要で

ある。今後はあらかじめ重みが解っているリンクを利用して未知のリンクの重みを決定するネットワークの学習の方法、隠喩理解ルーチンの構築をしていく予定である。

参考文献

- [1] 麻生 英樹, ニューラルネットワーク情報処理, 産業図書, 1988
- [2] 安井 稔, 言外の意味, 研究社, 1978
- [3] 竹内 晴彦, メタファー理解の試み, 情報処理学会第33年全国大会 3L-1 pp. 1201-1202, 1986
- [4] J.H.Martin, "A Computational Theory of Metaphor", Report No. UCB/CSD 88/465 University of California Berkeley California 94720, 1988
- [5] 岩山 真, 徳永 健伸, 田中 穂積, 比喩を含む言語理解における視点の役割, 情報処理学会自然言語処理研究会報告 73-7, 1989
- [6] M.Black, "Metaphor", *Proceedings of the Aristotelian Society*.55 pp.273-294. Harrison & Sons Ltd. London, 1954
- [7] H.H.Clark and E.V.Clark, 藤永 保 他訳, 心理言語学, 新曜社, 1977
- [8] 菅野 盾樹, メタファーの記号論, 勁草書房, 1985
- [9] 土井 晃一, 田中 英彦, スパルベルの象徴解釈モデルに基づく隠喩の検出, 情報処理学会論文誌 Vol.30, No.10 掲載予定
- [10] 中村 明, 比喩表現辞典, 角川書店, 1977
- [11] 津田 一郎, 野村 浩郷, カオス神経回路網による意味概念空間の構成と利用について - ニューラルネットワークによるシソーラス -, 情報処理学会自然言語処理研究会報告 71-5, 1989
- [12] グループμ, 佐々木 健一, 樋口 桂子訳, 一般修辞学, 大修館書店, 1981

- [13] Y.Miyata, "SunNet Version 5.2: A tool for constructing, running, and looking into a PDP network in a Sun Graphics Window, ICS Report 8708, Institute for Cognitive Science, Univ. of California at San Diego, La Jolla, CA. 1987

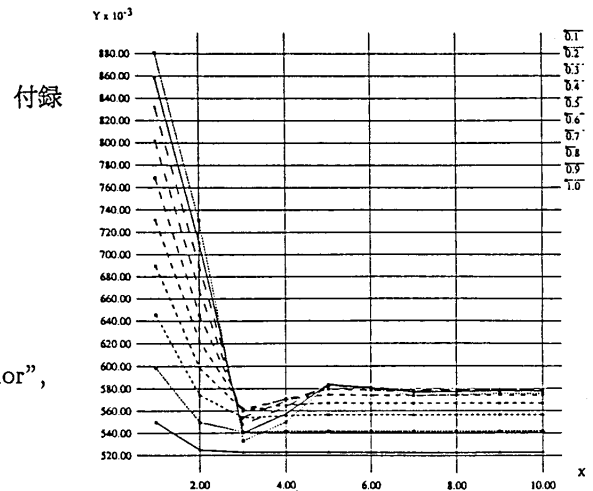


図 5: 活性値の動的変化

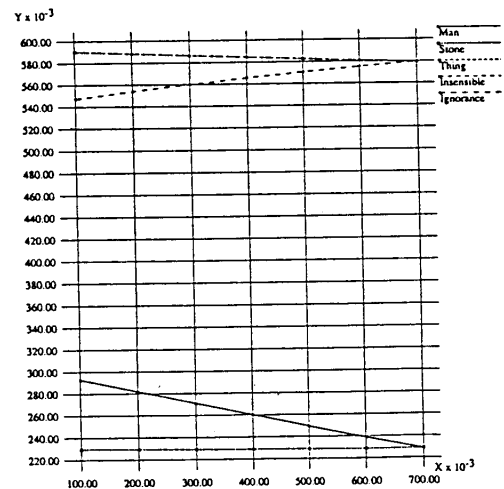


図 6: 収束値その一

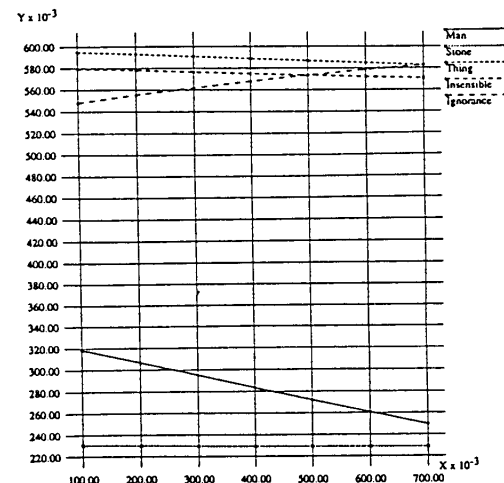


図 7: 収束値その二

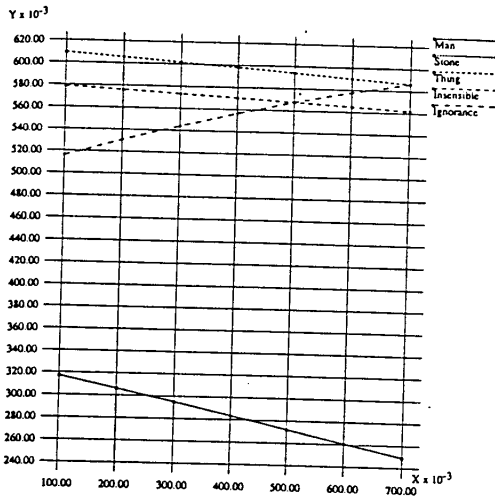


図 8: 収束値その三

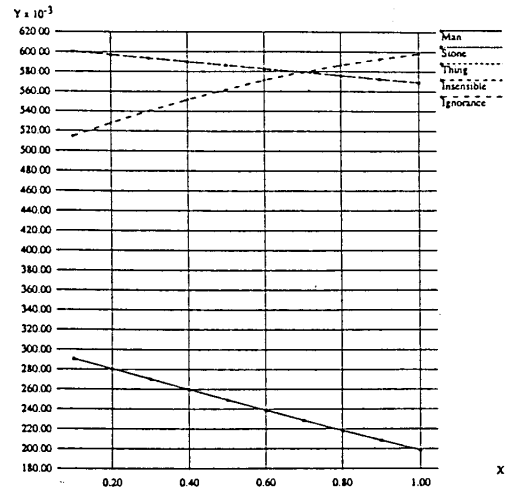


図 11: 収束値その六

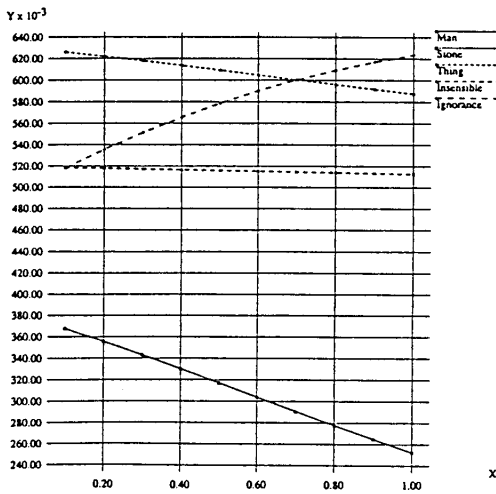


図 9: 収束値その四

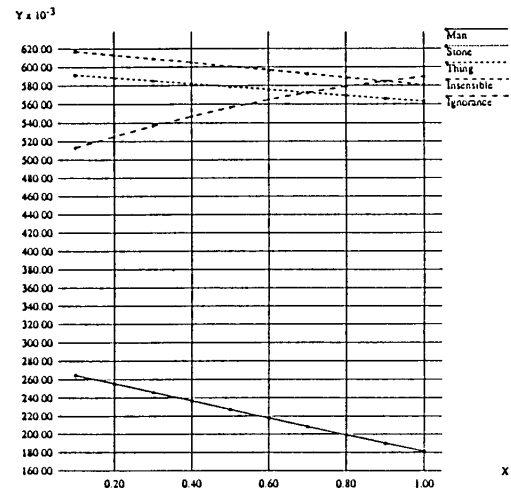


図 12: 収束値その七

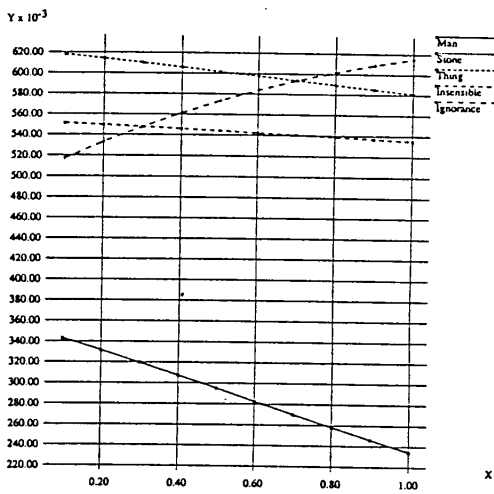


図 10: 収束値その五

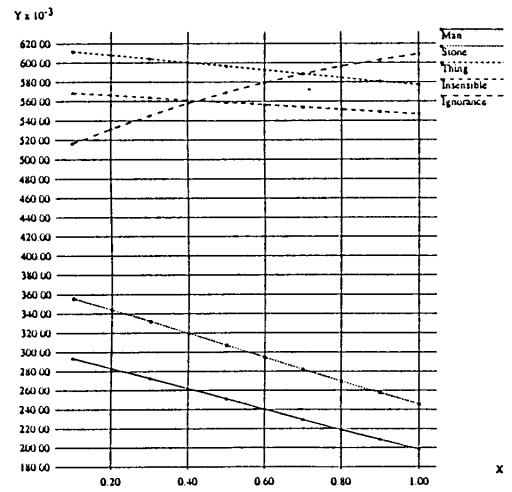


図 13: 収束値その八

表 1: 石に関連する隠喩

番号	入力文	命題の組み	その含意
1	「石は蹴飛ばされて生きてゆく」	(stone,man,kick,live)	つまらないもの
2	「非情な石」	(stone,man,inanimate)	非情な
3	「人は石だ」	(stone,man)	感情の無い
4	「石はなかなか起き上がらなかった」	(stone,man,not_easy,rise)	重い
5	「石を見ています」	(stone,man,look)	動かない
6	「石を冷然と尻目にかけて」	(stone,man,coldly,disregard)	関係のない
7	「石を前にして立った」	(stone,man,front,stand)	隙間もない、動かない
8	「石は背を見せて、動かなかった」	(stone,man,turn_back,not_move)	動かない
9	「石は立ち上がった」	(stone,man,stand)	動じない
10	「石か鉄を背に負うような心持」	(stone,man,shoulder,feeling)	重い
11	「石には弾き返されるばかりだった」	(stone,man,flick)	弾き返される
12	「石は確かなものに感じられた」	(stone,man,sure,feel)	確かな
13	「石は熱心に爺さんを見つめている」	(stone,man,eagerly,look)	動かない
14	「僧侶は石だ」	(stone,man)	無知な
15	「石が転げ落ちた」	(stone,man,fall_down)	転げ落ちる
16	「石は堅い身体だった」	(stone,man,hard,body)	固い
17	「石は身を固くした」	(stone,man,hard,body)	固い
18	「石が都会の断層の中に落ちこんでいく」	(stone,man,city,gap,fall_down)	重い
19	「石を迂回した」	(stone,man,detour)	邪魔な
20	「石が座り込んでしまった」	(stone,man,sit)	固い
21	「石が立ちすくんでしまった」	(stone,man,petrify)	動かない
22	「石は坂道をかけおりた」	(stone,man,slope,run_down)	はずんだ
23	「石は眠りに落ちた」	(stone,man,sleep)	動かない、意識の無い
24	「石の沈黙から急に生き返って来る」	(stone,man,silent,rapid,revive)	動かない
25	「石は黙っていた」	(stone,man,silent)	黙って
26	「石を喰って黙りこくった」	(stone,man,eat,silent)	重い、感情の無い
27	「捨てられた石」	(stone,man,junk)	退屈な
28	「石は彼女と別れてきた」	(stone,man,apart)	無言の