December 21, 2023

自己組織化論理

進藤　啓介

SHINDO Keisuke

kshindo999@gmail.com

概要

自己組織化論理は、確率によって順序回路を自律修正、自動生成することを目指した論理回路動作シミュレータの理論体系である。以下SOL(Self Organizing Logic)と呼称する。SOLは双方向ベイジアンネットワークを用いて論理回路を形成する。一般的な写像はこの双方向論理演算の逆伝搬を用いて記述可能であり、これにより階層的な順序回路を記述することが可能になる。これとは独立してネットワークの確率の伝搬を二項組の伝搬確率で表現する。逆方向伝搬確率はベイズの定理に基づいて定義する。この確率計算によってネットワークに対して厳密な確率論に基づいたフィードバックが可能になり、正確な論理演算の挿入も可能にする。さらに、このネットワークに対して仮定ベクトルと呼ぶ活性値を伝搬させる。この仮定ベクトルの要素比較でネットワークのノードを選択することにより、最小限の新規の連想リンクを生成する。これらの要素を用いれば、観測対象の無制限の規模の事象に対して、それと等価な順序回路の自己組織化生成を最速で行うシステムを構築可能と期待される。

Self-organizing logic is a theoretical system of logic circuit operation simulators that aims at autonomously modifying and automatically generating sequential circuits by probability. SOL forms logic circuits using bidirectional Bayesian networks. The general mapping can be described using the inverse propagation of this bidirectional logic operation, which makes it possible to describe hierarchical sequential circuits. Independently of this, the propagation of probabilities in the network is expressed in terms of binary pairs of propagation probabilities. The backward propagation probability is defined based on Bayes' theorem. This probability calculation allows for feedback based on rigorous probability theory for the network and also allows for the insertion of exact logic operations. In addition, an active state, called an assumption vector, is propagated to the network. By selecting nodes of the network by comparing elements of this assumption vector, we generate a minimal number of new associative links. Using these elements, we expect to be able to construct the fastest system for self-organizing generation of equivalent ordered circuits for events of unlimited scale to be observed.

keywords

machine learning,bidirectional baysian network,sequencial circuit

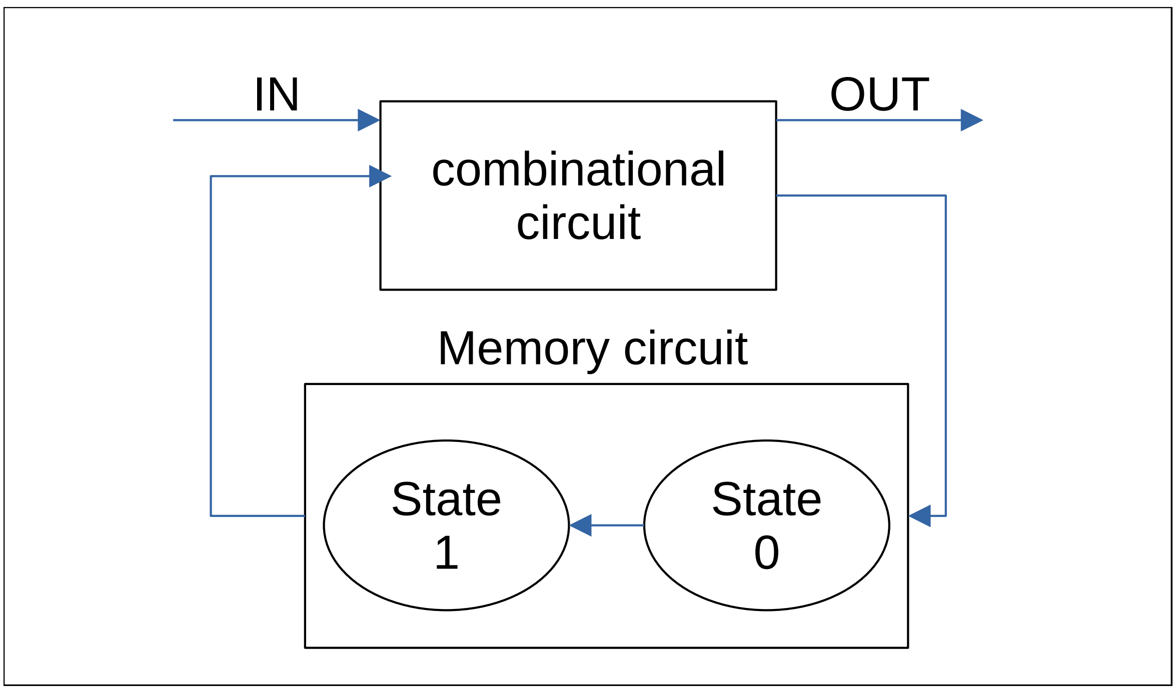
Contents

1.自己組織化論理の構成要素

1.1 双方向論理演算とリンクノード

1.1.1 古典的な順序回路

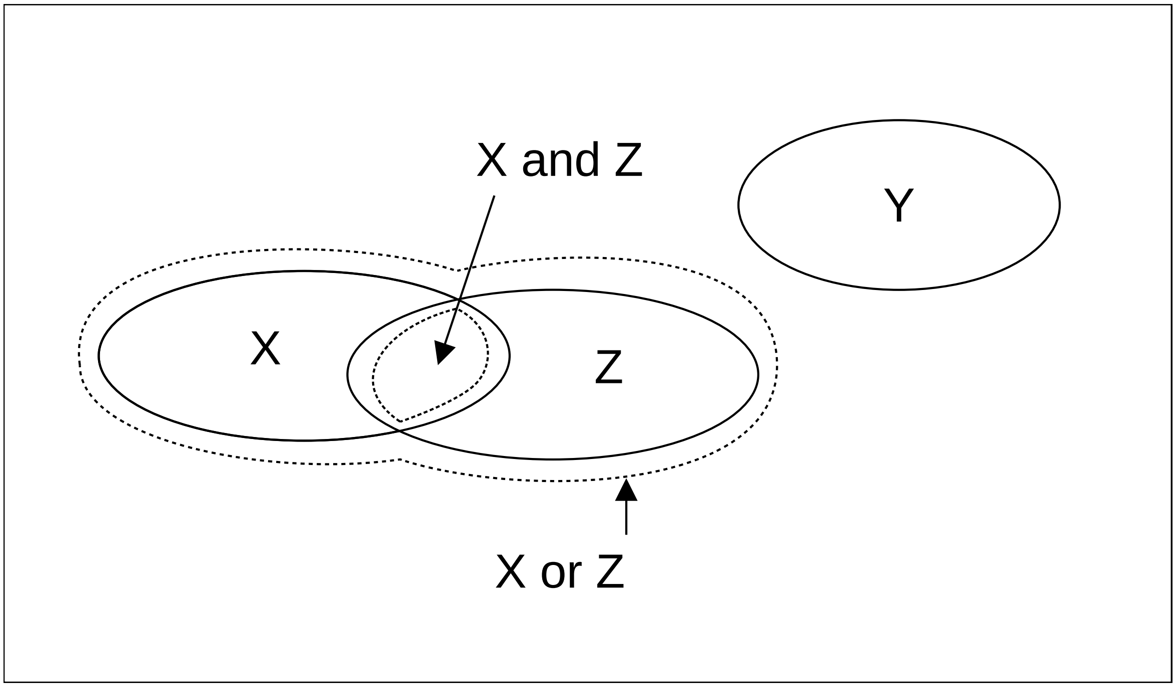
　通常の論理代数では、AND、OR、NOTの論理演算の要素をノードとして持ち、それらをリンクで結合して、基本的な論理動作を行う。順序回路では、これにフリップフロップFF、あるいはメモリなどの記憶機能が追加される。CPU、GPUなどはすべてこの順序回路に該当する。この順序回路は任意の入力列に対して理論上あらゆる出力列を得ることができる。



Sequencial circuit

1.1.2 論理回路の逆伝搬による写像

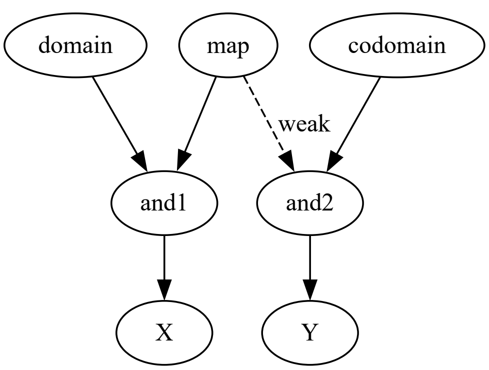
　論理演算は、ベン図などで表現されるように集合を互いに切り貼りする作用があるが、下の図のXとYのように重複のない排他的な部分集合間では論理演算が出来ない。それに対して、写像は、入力された集合に対して排他的な別の集合に到達することを可能にする。そのためには、写像の前後の集合を、それらを包含するさらに大きな集合であるMapノードの部分集合とみなす。入力ノードXと出力ノードYとは、Mapノードに対してそれぞれdomain,codomainノードとANDを取る結果と等しくなる。

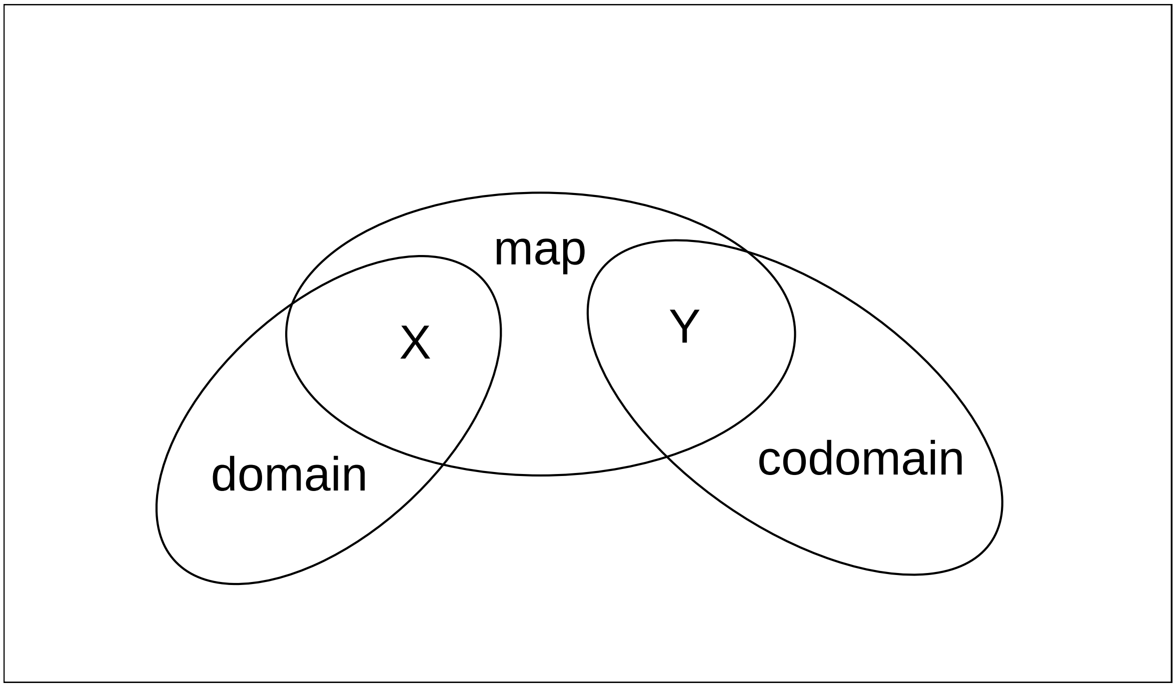


Basic venn diagram

　写像元Xから写像先Yへの伝搬は、Mapノードへの逆伝搬で実現する。排他的なノードの間の確率伝搬を定義するために、AND1からMapへの逆伝搬では、domainの補集合の部分を補うことで集合を拡大する。Mapノードで拡大された集合はcodomainと集合的に排他的ではなくなるため、codomainとのAND論理演算が可能になる。こうして集合的に重複のないXからYへの伝搬を可能にする。XORではなくてANDである理由は、入力ノードXの内部の全領域がMap内部に包含されるためである。

　この写像ノードと逆伝搬を用いた連想リンクによって、状態遷移の記述を可能にする。さらに、この作用がフリップフロップやメモリなどの記憶素子と同じ挙動を、より一般化して実現する。例えば、メモリの読み出しは、時空間が異なる集合からの写像と見なすことができる。

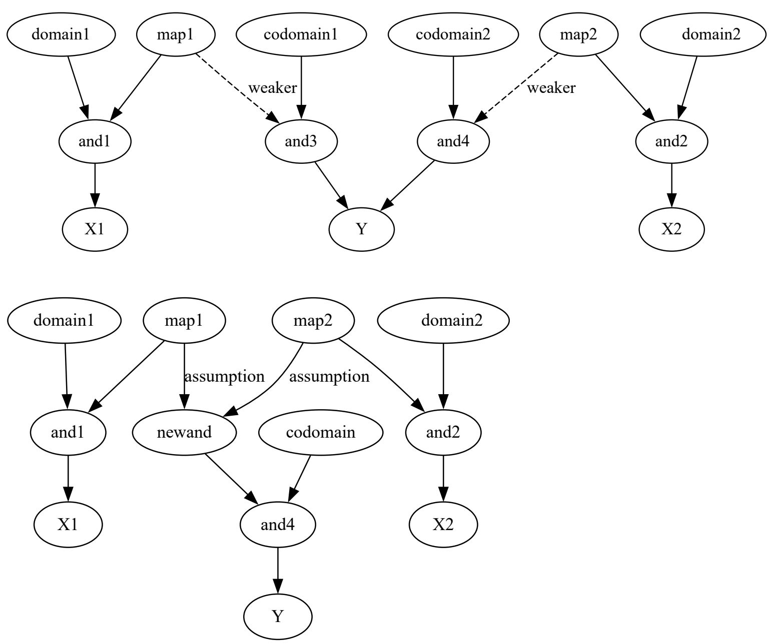




Map link by backward logic propagation

1.1.3 写像のリンクへのフィードバックと論理演算の挿入

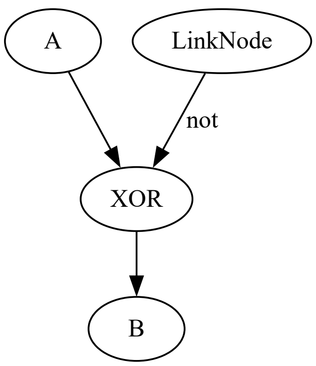
写像map1の写像先Yへのリンクは、観測条件によっては伝搬確率が確定していないことがある。この不確定なweakリンクが確定して確実にYに伝搬される条件を見出すために、別の写像map2との論理演算newandを暫定的に形成する。newandは論理演算のANDだけではなくて他にもOR,XORなど様々な種類の論理演算が考えられるが、weakリンクの伝搬確率と伝搬集合によって確定的に論理演算の種類が選択できる。



Feedback and create new logic node

1.1.4 因果関係とリンクノードへの逆伝搬

全てのリンクはフィードバックによって伝搬確率が変動する。変動したリンク$A \rightarrow B$自体の成立確率を利用して、リンク自体の成立条件に相当するXORノードとリンクノードを生成し逆伝搬する。これによって、写像前後A,Bの伝搬集合が統合され、リンクノード自体がリンクを表す集合となり、リンクの因果関係そのものを論理演算の対象とすることが可能になる。さらにこれを応用すると、状態変数の一致をリンクノードと見なすことで、ソフトウェアの条件文の制御に利用できる。順方向論理演算だけではこれは実現できない。



Link nodes indicate causal relationships

1.2 双方向二項確率ベイジアンネットワーク

1.2.1 確率伝搬

論理ノード間の接続リンクに、それぞれ伝搬確率を与える。伝搬する論理値は1,0の二値であり、伝搬確率を0から1までの値で示す。

リンクは２つの伝搬確率 P^{f11},P^{f00} のペアで構成され、さらに逆方向の伝播確率のペア P^{r11},P^{r00} も規定する。

AからBへの伝搬リンクにおいて、A、Bそれぞれの確率値P(A),P(B)は、２つの伝搬確率 P^{f11},P^{f00} によって関連付けられる。

$$P(B) = P(A)P^{f11} + (1 - P(A))(1 - P^{f00})$$

値の反転であるNOT演算は、P^{f11} , P^{f00} がそれぞれ０になる。NOTでない伝搬は、P^{f11} , P^{f00} がそれぞれ１になる。論理演算も確率計算として定義され、複数の論理演算の入力確率 P_1^{f11},P_2^{f11}, P_3^{f11} を統合する。以下はANDの例である。

$$P' = P_1^{f11} P_2^{f11} P_3^{f11} ...$$

これらの自明な確率計算を組み合わせることにより、任意の２つのノード間の論理演算の経路において、経路全体の伝搬確率が求まる。厳密な確率そのものを伝搬することで、しきい値関数などを一切使用しないという点が、ニューラルネットワークと異なる。

1.2.2 逆方向伝搬

リンクの逆方向伝搬の確率について説明する。たとえば、以下のようなANDノードCが生成されたとする。

$$C = A \cap B$$

逆に、ノードCが確率１に活性化された場合は、ノードA,Bの伝搬確率１での活性化が可能になる。これが逆方向伝搬である。

$$C \rightarrow B$$

逆方向伝搬により、２つの任意のノード間の伝搬が確実に可能になる。逆方向伝搬の確率 P_r^{11},P_r^{00} は、ベイズの定理を用いて算出される。P(A)をリンク元A、P(B)をリンク先Bの観測確率として、以下の式で求められる。

$$P_r^{11} = \frac{P(A)P^{11}}{P(B)}$$

$$P_r^{00} = \frac{(1-P(A))P^{00}}{1-P(B)}$$

つまり、双方向確率伝搬は、ベイズの定理が自然な形で組み込まれている。論理演算に対する逆方向伝搬は、ANDノードへの値１の逆伝搬のように値がすべて確定する場合はすべての入力リンクに対して伝搬する。ANDノードへの値０の逆伝搬は入力リンクの伝搬確率が確定しないが、伝搬する入力リンクを１つだけ選択すれば、選択したリンクだけは確率が確定する。

1.2.3 リンクノード伝搬

A→BのA,B双方に別のパスからそれぞれ活性値が伝搬されたとする。その際にA,Bの確率の一致や不一致に対してリンクノードLへの逆伝搬を定義できる。これがA,Bの等価性を示すリンクノードの挙動となる。

$$A xor \neg L = B$$

¬は集合のNOTを示す。確率P_Aと確率P_Bから、それらの間のリンクノードへの伝搬確率P_Lは、$P_L = P_{11} = P_{00}$と仮定して、以下の式で求められる。

$$P_L = \frac{P_A + P_B - 1}{2P_A - 1}$$

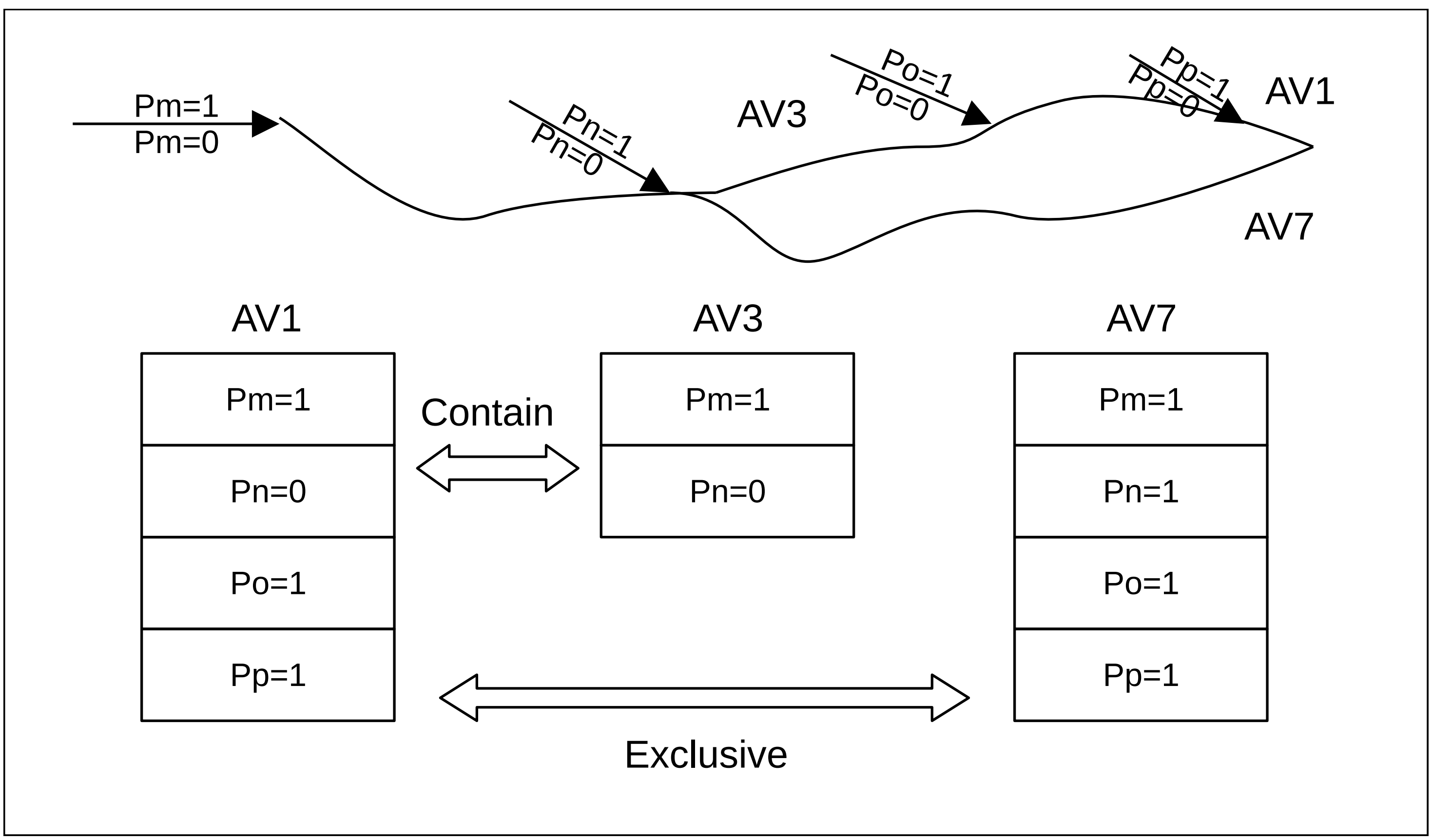
P_A, P_Bは確率が1と0の中間値であっても、($P_A = 0.5$以外で）値が一致していればP_Lで確率1が得られる。逆に、P_AとP_Bが値が1と0の組み合わせであるならば、リンクノードの伝搬確率は0にになる。

- $P_L = 1 \Leftrightarrow B = A$
- $P_L = 0 \Leftrightarrow B = notA$
- $P_L = 0.5 \Leftrightarrow$ AとBとは無関係

こうして、SOLではリンクをたどって到達できるノードであれば、いかなるノード同士でも伝搬確率を得ることができる。

1.3 仮定ベクトルと伝搬集合

SOLは、ネットワーク上に活性値を伝搬するが、活性値は伝搬するリンクの断面のすべてを集合として占めるわけではなく、起点となる複数のノードの値によって集合が分断されると考える。この伝搬集合はネットワーク上で通過する集合とは無関係であり、起点ノードの値の組み合わせによって決定される。そのため、複数の起点ノードの値の仮定の組み合わせを「仮定ベクトル」と呼ぶことにする。以下、仮定ベクトル(Assumption Vector)は必要に応じて略して「AV」と表記する。



Assumption vectors

1.3.1 仮定ベクトルの合成

複数の仮定ベクトルは、論理演算によって合成された結果、双方の仮定ベクトル要素が合成された仮定ベクトルとなる。

- 仮定ベクトル間で異なる仮定ベクトル要素はそのまま合成される。
- 仮定ベクトル間で同一の仮定ベクトル要素は１つに集約される。
- 集合的に包含関係の仮定ベクトル要素との合成は伝搬集合的に小さい要素に取って代わる。
- 集合的に排他的な仮定ベクトル要素との合成は伝搬集合自体が空集合になる。

1.3.2 仮定ベクトルによる伝搬集合比較

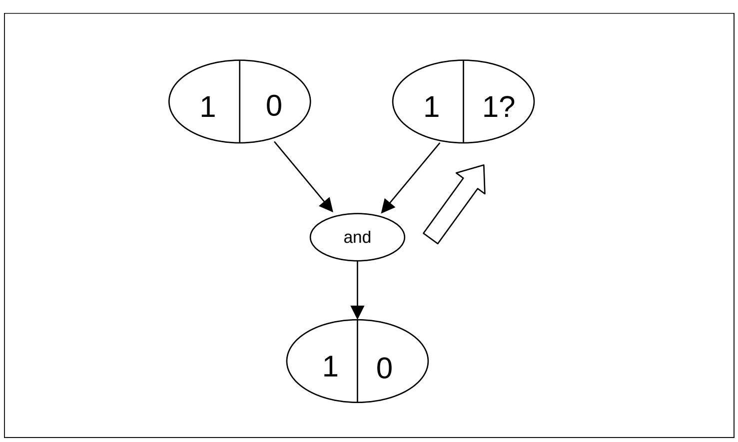
この仮定ベクトルを用いて、複数の活性値の伝搬集合としての大きさを比較することが可能になる。この比較は、写像を通過した伝搬集合としての比較であり、実際の集合とは無関係である。

- 仮定ベクトルの仮定集合要素がすべて同一であれば、実際の集合にかかわらず双方の伝搬集合は同一である。
- 仮定ベクトルの仮定集合要素の非同一部分が片方のみであれば、双方の伝搬集合は包含関係になる。
- 仮定ベクトルの仮定集合要素の非同一部分が双方にあれば、双方の伝搬集合は無関係になるので連想の対象にならない。
- 仮定ベクトルの仮定集合要素の間に一切重複がなければ、伝搬集合が消滅するので伝搬そのものを停止する。

仮定ベクトルを利用した厳密な伝搬集合の管理により、2つのノード間の正確な連想リンクを形成できる。SOLのリンク形成は、すべてこの仮定ベクトルに従って選択実行される。

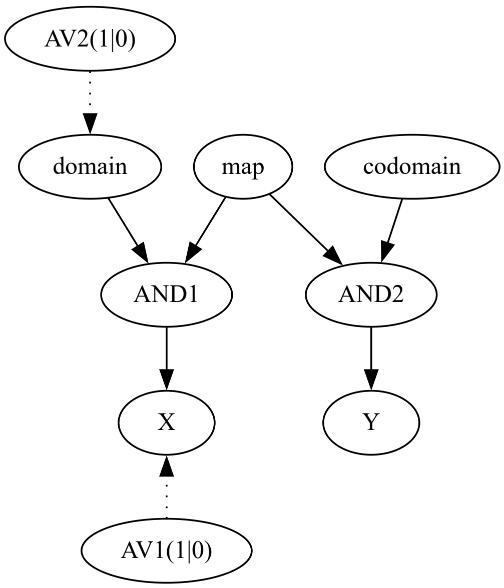
1.3.3 写像逆伝搬と伝搬集合拡大

Bool値などを示すノードX,Yについて、ノードXからノードMapへと逆伝搬する際に、集合を拡大する作用によって排他的なノードYへと伝搬する方法について述べる。



Propagate backward

上の図のように、ANDに対して出力から入力へと逆伝搬するということは、他の入力がすべて確定していたとしても、逆伝搬の結果の確率は確定しない。右の入力は0も1のどちらでも矛盾はない。



Using assumption vectors to propagate on map link

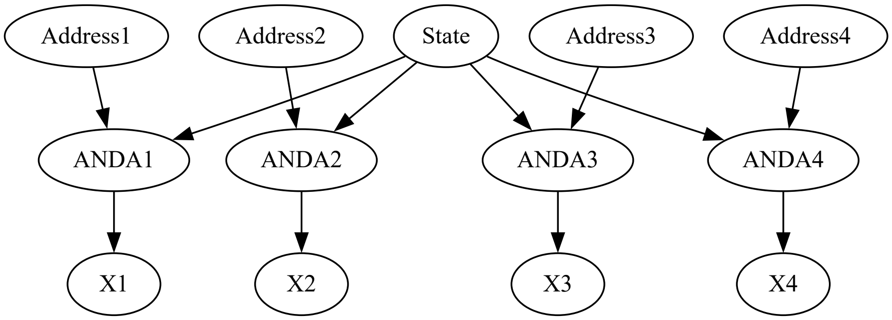
XからMapへの逆伝搬を行う。AND1からMapへの逆伝搬の際に、集合domainの入力が別にある場合は、domainの補集合の拡大を行うことができる。Xから伝搬されるAV1も、domainから伝搬されるAV2も、値が相補的な２つの伝搬集合のペアであるとする。この場合、Xの伝搬集合がdomainの伝搬集合に一致するか完全に含まれる場合に、Xからの集合を拡大してMapに伝搬する。伝搬集合が拡大するとは、domainの相補的な仮定ベクトルAVの要素を合成して伝搬集合を拡大するということであり、仮定ベクトルAVの要素が１つ減るということでもある。

仮定ベクトルの合成結果は伝搬集合が拡大されるので、確率は均一に１であるという可能性が高い。その理由は、Mapで合成された伝搬集合は、AV1に仮定要素を追加する前の伝搬集合と同一である可能性が高いため、その確率もまた仮定要素を追加する前の伝搬集合と同じく均一の値であると推定されるためである。

こうして、伝搬集合が拡大された状態でMapからAND2に伝搬され、codomainとANDを取る。この結果Yへと拡大された伝搬集合が伝搬される。ここで、mapノードからAND2ノードへのリンクは弱く確定していない状態であるが、それは上の集合の拡大の推定がまだ確定していないということでもある。逆に、弱いリンクがポジティブフィードバックによって強化されるということは、集合の拡大の確率も同時に確定に近づくということでもある。

1.3.4 一般的な状態の表現

写像を用いれば、巨大な状態を以下のように表現することができる。部分状態X1,X2などはさらなる状態を階層的に接続できる。それぞれの状態をAddress等で分割して示すだけではなく、連続な時系列や座標を使用することも可能で、さらに抽象的なノードを別の写像で連結することも可能になる。



Multi state using map links

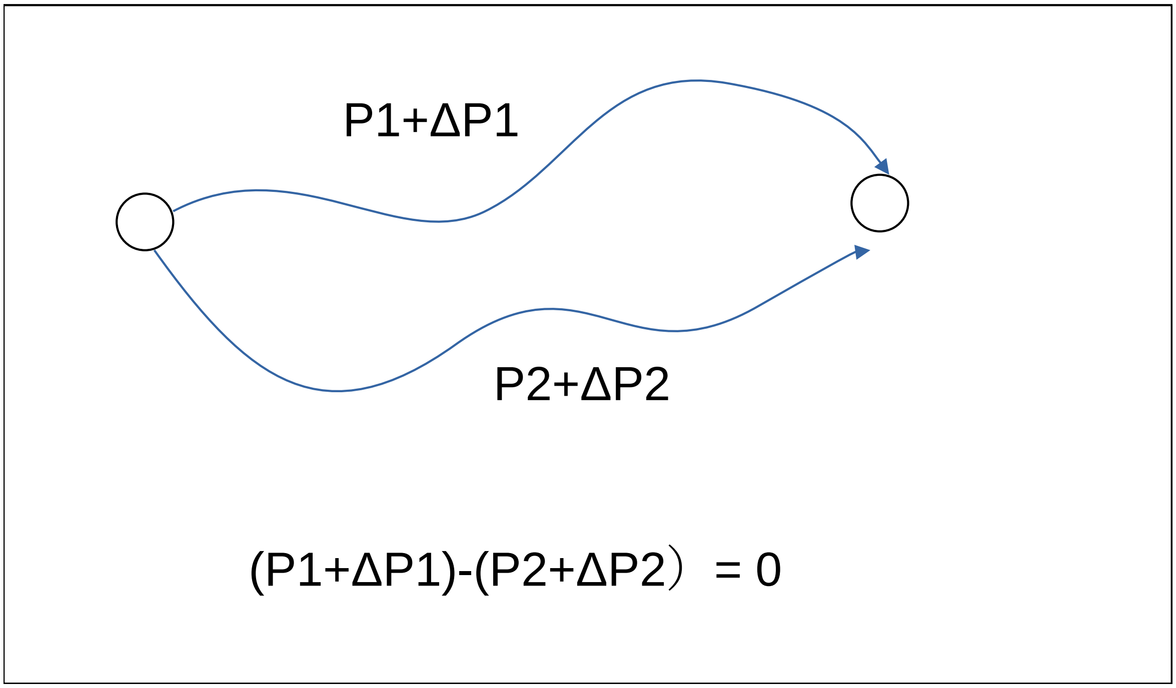
1.4 確率変動分配階層フィードバックアルゴリズム

SOLのネットワークを通過する伝搬確率と、外部観測関数で観測されたパスの伝搬確率との違いを解消して、観測された確率を再現すること。これがSOLにおける一般化された学習と考える。現存する通常のニューラルネットワークの学習と目的はほぼ同一だが、活性化関数のようなヒューリスティックスを用いずに、より厳密な方法で実現する。

1.4.1 確率伝搬と衝突とフィードバック

任意の共通の起点ノードから、ネットワークの複数のパスを通過して、もう一度別の同一のノードに到達することがある。例えば、過去に観測された結果のノードと、現在観測されている結果のノードとの衝突が典型例となる。

その時にそれらの仮定ベクトルの集合に重複があれば衝突とみなす。双方の確率が等しければポジティブフィードバックを行い、確率が異なればネガティブフィードバックを行う。フィードバックの量を正確に計算するために、この確率変動分配階層フィードバックで確率計算を行い、フィードバックすべきリンクとその量を特定する。リンクからのフィードバックからの作用は後述する。



Difference in probability of passing two passes

1.4.2 フィードバック値の算出

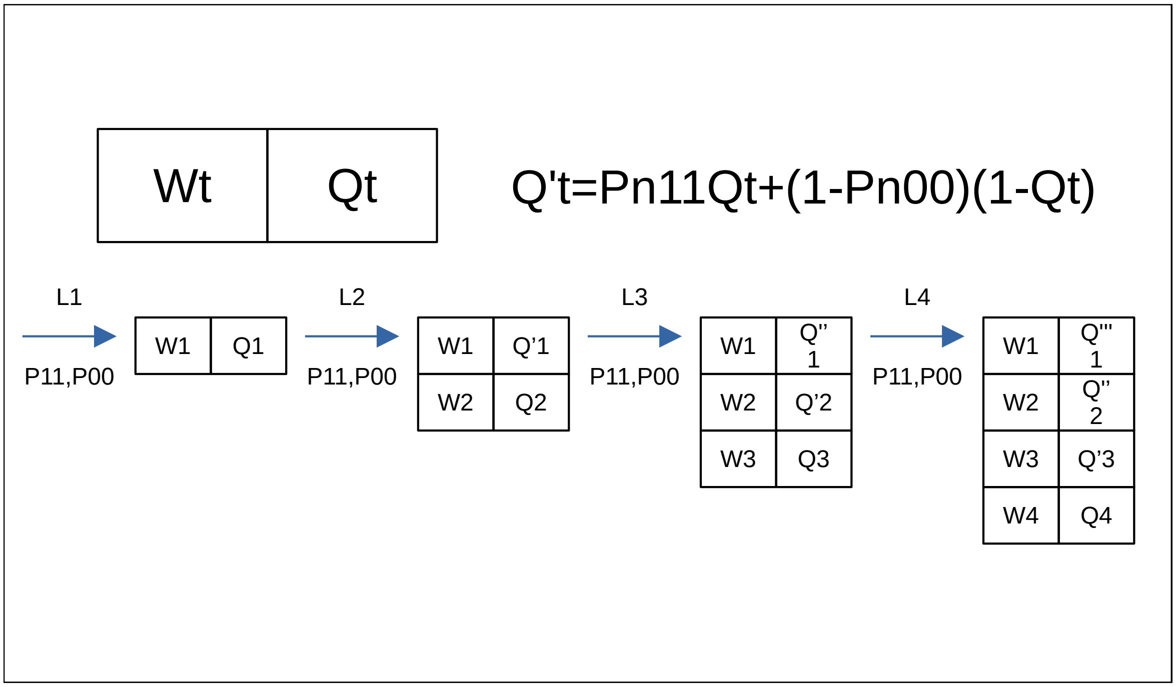
２つのパス全体の伝搬確率が観測されたときに、２つのパスの伝搬確率の違いが発生する。この違いを解消するために、それぞれのパスにフィードバックを行う。ここで問題になるのは、階層的なリンクでは、どのリンクにどの程度のフィードバックを作用させるかの判定が難しいという点である。この問題を、「確率変動分配階層フィードバック」アルゴリズムでは、より正確な手法で解決した。この手法は、理論上無制限の階層のリンクそれぞれに対して、確率論的に最適に近いフィードバックを、少ない計算オーダーで適切に与えることが可能になる。

基本的なコンセプトは、確率の変動の発生確率が、リンクの伝搬確率P_nと観測回数N_nに依存するところからスタートする。このリンクを複数通過した結果、全体伝搬確率P_{total}が観測されたとする。その全体伝搬確率の結果との相違が発生したら、その相違は、以下のようにそれぞれのリンクに対して、リンクごとのウェイトw_nに応じて分配されてフィードバックされる。リンクごとのウェイトw_nは以下の式で算出する（かなり厳密な導出に基づくがここでは省略する）。P_nはこれまで観測されたリンクの伝搬確率であり、N_nはこれまでフィードバックが適用された回数を示す。Q_nは、各リンクが伝搬で利用された確率であり、全体の伝搬によって算出される。

$$w_n = \frac{P_n(1-P_n)}{N_n}$$

$$P_n' = P_n + w_n \frac{\Delta P_{total}}{\sum_n Q_n w_n}$$

上記のW_nとQ_nとを用いて伝搬全体の確率を計算する。Q_nに対してはリンクの伝搬に従って個別のリンクの確率を適用して伝搬する。



Weight propagation

伝搬が終了したら、それぞれのリンクの伝搬確率を可能な限り0か1に近づけるように、かつ変動値の符号の整合性が取れるように、全体の観測確率の変動をW_n, Q_nに従って分配する。

フィードバックで補正された伝搬確率は、そのままではリンクの次回の伝搬確率にはならない。リンクのそれまでのフィードバック観測回数N_nが大きければ、それだけリンク伝搬確率のフィードバックの量が小さくなる。N_nはリンクの利用確率Q_nを用いて加算されてN_n'となる。

$$N_n' = N_n + Q_n$$

$$P_n' = \frac{N_n P_n + Q_n P_n'}{N_n'}$$

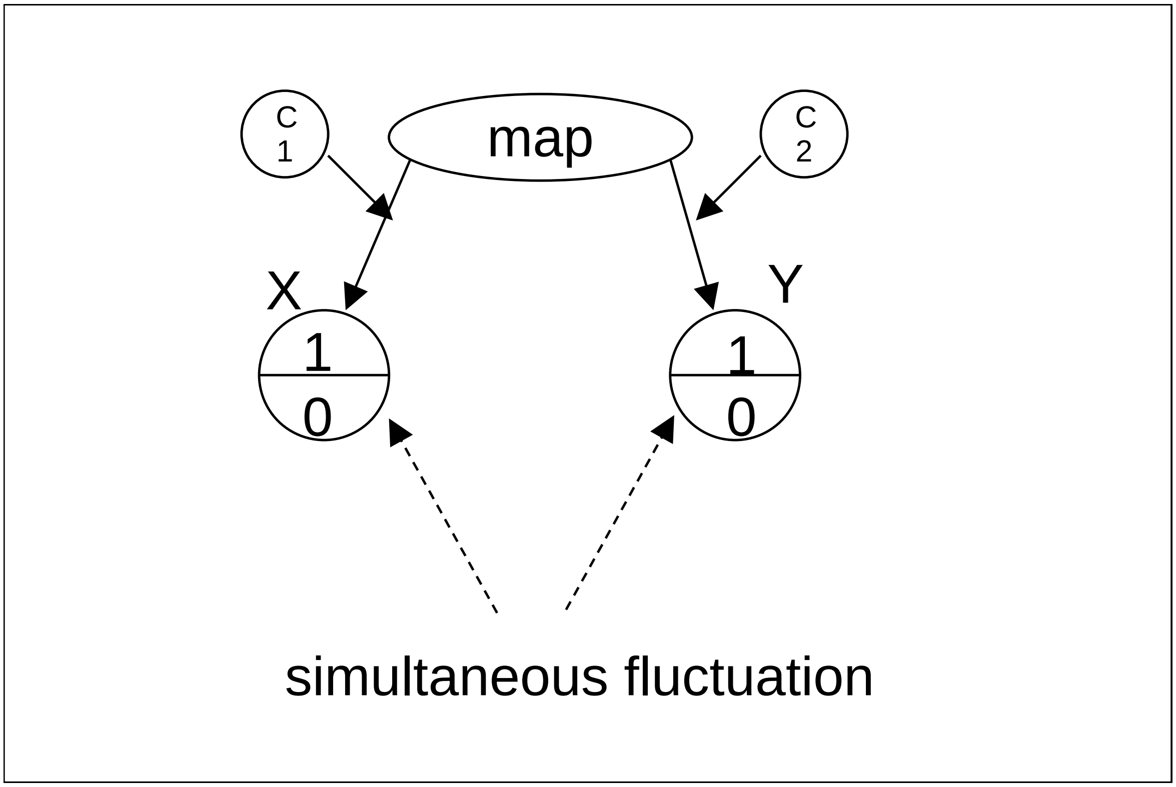
この確率変動分配階層フィードバックは、ニューラルネットワークのバックプロパゲーションと比較しても、深い階層のネットワークに対して、フィードバックをより正確に作用させることが可能になる。理由は、厳密な確率論の適用を行っているためである。

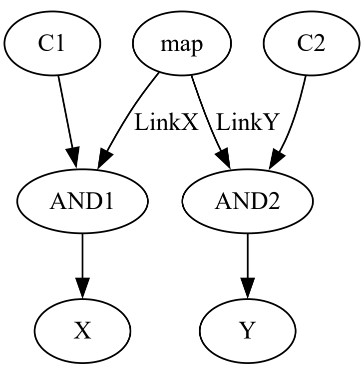
1.5 同時観測からの連想形成

同時観測は、集合的に重複のないノード間の連想を定義するための基本的な方法となる。あるノードの部分集合が、別のノードの部分集合と同時に値が確定したときに、それらを写像で連想するのが基本になる。同時というのは仮定ベクトルが一致あるいは包含関係になっているという意味である。

だがそれだけでは、無関係な事象が偶然一致したという可能性がある。そのために、フィードバックを用いて連想の成立確率を上げていく。

　そのため、２つの値の発生確率Pがある程度低く、かつ２つの値が同時に確定したという観測事実から写像連想を形成する。典型的にはノードXとノードYとの同時変動から連想を形成する。





Create association

　連想の不確定性は、MapからAND1、あるいはMapからAND2へのリンクの確率P^{11}とP^{00}で示される。リンクのN$^{\{11\},N}$$_{\{00\}}$\$はそのリンクへのフィードバックの回数を示す。

　X、Yのそれぞれの過去の観測確率をP_X, P_Yとする。$X=1$のとき常に$Y=1$である場合は、LinkYのパラメータの具体的な初期値は以下の式で決定される。特にP_Yが十分に小さい場合は強度が増す。

$$P_y^{11} = 1$$

$$P_y^{00} = 0.5$$

$$N_y^{11} = -log_2 P_Y$$

$$N_y^{00} = 0$$

$X = 0$のとき常に$Y = 0$である場合は、以下の式でパラメータが決定される。

$$P_y^{11} = 0.5$$

$$P_y^{00} = 1$$

$$N_y^{11} = 0$$

$$N_y^{00} = -log_2(1 - P_Y)$$

LinkXも同様に定義される。それ以外のリンクは確定値を持つとみなす。確定値のリンクはフィードバックの対象とならない。

$$P^{11} = 1$$

$$P^{00} = 1$$

$$N^{11} = \infty$$

$$N^{00} = \infty$$

連想形成後は、$X = 1$から$Y = 1$への伝搬と、$X = 0$から$Y = 0$への伝搬の、双方の伝搬確率をそれぞれ観測することでフィードバックを行う。この結果、連想リンクのN_y^{11}とN_y^{00}がさらに加算されていき、P_y^{11}側とP_y^{00}側の双方の確率が確定していく。

連想を形成する対象X、Yは、観測結果のノードだけではなく、フィードバックの結果である条件ノードや因果関係を示すリンクノードなども対象にできる。こうして、あらゆる部分集合の間で確率に基づいて連想を形成することができる。

1.6 確率的自律論理生成アルゴリズム

SOLは、「確率的自律論理生成アルゴリズム」を利用する。このアルゴリズムは、ノードとリンクとを自律的に生成、修正する。

ノードとリンクの考えられるあらゆる構成に対して、それを自律的に生成するためには、基本的には以下の機能で可能と考えられる。

1.6.1 「条件」ネガティブフィードバックからのリンク分割とAND、OR論理演算ノード形成

フィードバックの結果、リンクの確率が1から0.5、あるいは0から0.5に近づいた場合は、リンクがネガティブフィードバックされたと考える。つまり、現在の前提ベクトルを条件として、リンクの不確定な確率部分を論理演算を用いて分離する。

リンクの伝搬確率をP^{11},P^{00}の二項組で管理することで、ネガティブフィードバックが発生した伝搬確率の方向から、挿入すべきANDノード、ORノードを確定的に選択できる。P^{11}が1から0.5に近づいた場合はANDノード、P^{00}が1から0.5に近づいた場合はORノードである。XORノードはP^{11},P^{00}双方が0.5に近づいた時に選択されるが、P11,P00双方に対して同一の条件でフィードバックされた場合に限定される。挿入された論理演算に追加される条件リンクは、前提テンソルが等しいノードから選択される。

これが、SOLにおける論理演算の自律生成の基本であり、正確な論理演算の形成が可能になる。

1.6.2 「因果」ポジティブフィードバックからの因果関係を示すリンクノードの入力を逆活性化

あるランダムリンクを通過した２つの活性化パスの確率が一致する場合は、ランダムなリンクがポジティブフィードバックされる。その結果、そのランダムリンクに相当する「リンクノード」がポジティブフィードバックによって活性化されると見なす。

リンクノードの具体的な実体は、ランダムリンクに仮想的に追加挿入された排他的論理演算（XOR)ノードに入力される、未知の入力ノードである。

リンクノードへの逆伝搬は、リンク両端の伝搬確率P_A,P_Bの一致結果を伝搬する。具体的には伝搬確率P_Lは以下の式となる。

$$P_L = \frac{P_A + P_B - 1}{2P_A - 1}$$

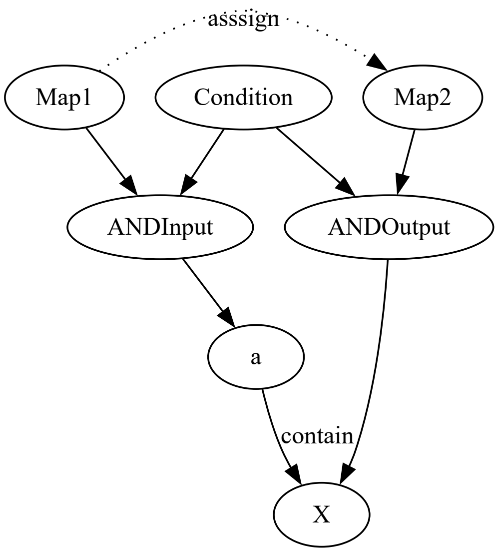
これは、異なる２つの集合間の因果関係の結果そのものを、確率計算の対象とするための基本的な方法になる。P_A, P_Bの確率が中間値であっても、0.5以外で一致すれば確定確率1になる。

因果関係は、数値等の一致判定からの条件判断制御、つまりソフトウェアのif条件文に応用できる。

1.6.3 「代入」　複数の写像を通過する際の集合的に等価なノード間の結合

順方向の論理演算によって伝搬を行うことで、ANDORなどの条件が追加されることで集合的には小さくなっていく。論理演算の逆方向に伝搬することで、条件が除去されて条件適用前の集合をそのまま伝搬することが可能になる。この結果、条件の追加と除去を通過した写像ノードの間で包含関係が

成立することがある。この整合性のとれた写像ノード間の伝搬を代入と呼ぶ。以下の例では値aは変数Xに含まれるので（$a \subset X$）、値aを含むMap1は変数Xを含むMap2の部分集合となり、Map1がMap2に代入される。



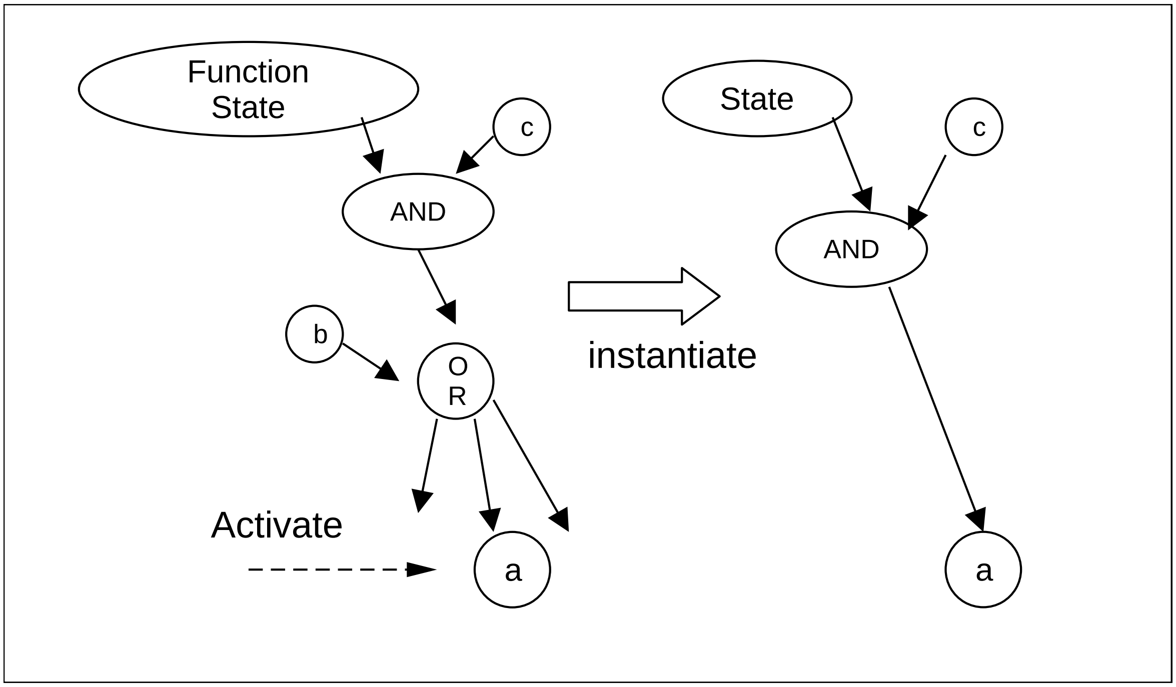
Assign between maps

1.6.4 「特化」ネットワークのインスタンス部分複製

２つの経路を通過した結果、ノードの集合の拡大が可能になることがある。この場合、リンクの経路に複数の出力リンクや論理演算入力リンクなどの分岐が存在する場合、経路のネットワークを部分的に複製して一種のインスタンスとする。インスタンスは不要なリンクの分岐が省略される。

下の図で例を示す。FunctionStateは右側のStateを部分集合として事前に代入している。代入されたStateから関数を通過して、値aを示す活性値に関数実行結果として伝搬が行われる。この伝搬集合はStateとの重複はないがStateの伝搬集合を矛盾なく拡大することが可能になる。このため、Stateは関数の結果として結果へのネットワークの複製として追加する。

この「特化」には、ネットワークの構造の中からさらに確率が確定した部分を抽出することができると同時に、伝搬の分岐を減らすことで伝搬の探索分岐を減らす作用がある。この「特化」は、関数の結果生成にも用いられる。



Network instantiate

1.6.5 「制御」リンク選択ノードによるリンク伝搬選択制御

あるノードに対して、結合される出力リンクは大量に存在しうる。リンクの探索を効率化するため、条件に対して、大量のリンクからのリンク選択を行う必要がある。この選択作用をSOL自身で決定、効率化することが必須となる。

リンク選択を制御するということは、選択したリンクに対応するリンク選択ノードを生成して、連想、活性値伝搬対象にする。その結果、選択先の活性値から得られた効用等と、リンク選択ノードとの連想を行うことができる。逆に、効用からリンク選択ノードが活性化され、効用に対応したリンクが選択されることになる。

リンク選択ノードの選択制御は、リンクに条件論理演算を追加して制御する方法とは異なり、リンクの伝搬確率を維持することができる。選択されていない状態でもリンクの伝搬自体はとくに阻害されないので伝搬自体は可能ではあるが、伝搬対象として選択される可能性は低い。

1.6.6 確率的自律論理生成

以上が確率的自律論理生成アルゴリズムの基本構成要素となる。AND、OR、XOR、NOT、写像リンク、リンク制御は、以上の手段を用いて自動生成できる。確率変動や集合に対して、伝搬確率が100%か0%に近い確定確率リンクを可能な限り選択する。

観測数が多く、かつ不確定な伝搬確率のリンクは、別の条件を順次追加して可能な限り確定確率リンクに置き換えていく。

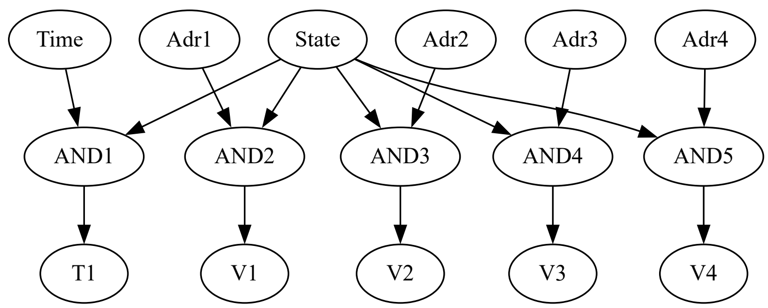
つまり、この確率的自律論理生成アルゴリズムは、アナログ値の重みを用いるニューラルネットワーク等と異なり、観測対象を可能な限りデジタル化して再現を試みるアルゴリズムである。確定的な

デジタル論理が生成されるので、論理式の形で学習結果の出力が可能になる。

1.7 順序回路の自律生成の例

1.7.1 変動連想からステートへ

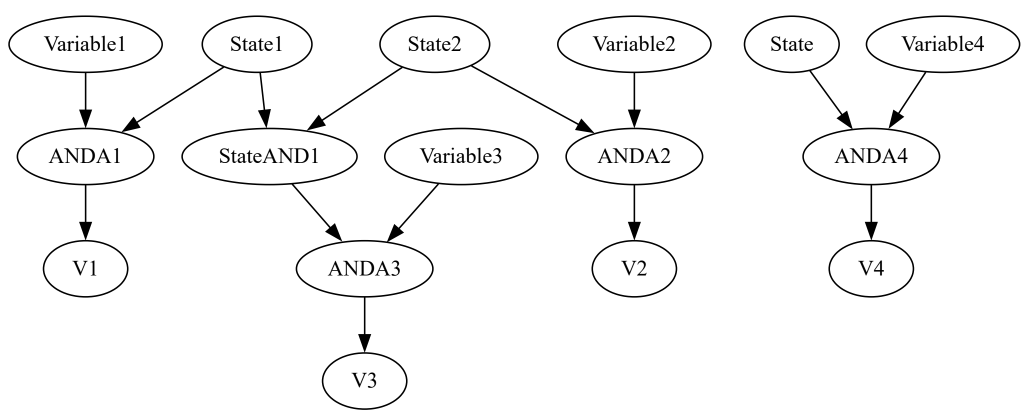
一般化された観測とは、ある時間ノードと、その時間に観測された複数の観測ノードとを連想で連結する作用と考える。つまり、複数ビットの値と時間ノードとのグループが生成される。



Form associations between time and state

1.7.2 ステート間の論理演算の自律生成

X1,X2,X3,X4はある時間での観測結果でありBool値である。時間によってそれぞれの観測結果は異なる。



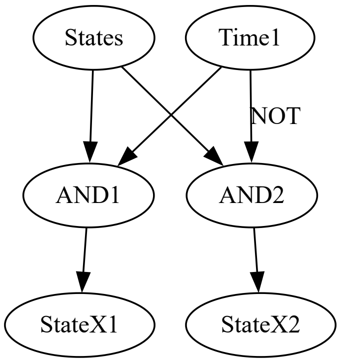
Associate with time passage and state

事前に形成された順序回路のネットワークに対して、V_1からV_4までの観測から再度伝搬が行われる。

StateとV1との確率の衝突からフィードバックが発生してStateAND1が生成され、StateがState1とState2とに分割されてStateAND1の入力条件となる。この結果、以下の論理式が自律的に形成される。

$$V_3 = V_1 and V_2$$

1.7.3 時系列で連想形成



Associations between time variability and state

Time1ノードの前後で、変動連想によってStateX1とStateX2が排他的かつ連続の時系列で結合される。この結果、隣接時間のStateX1とStateX2との間の関連が導き出される。さらにStatesからAND2へのリンクが弱く条件が必要になることにより、State間の論理式が形成されて順序回路が形成されていく。

SOLのノードは、この例ではBool値であるが、さらなる抽象的な数値、座標、文字トークンなどのあらゆる対象の連結が可能である。抽象的なノード同士の演算などの相互作用は組み込み関数などで実現され、SOLは組み込み関数同士の連結を確率で選択することになる。

2 双方向論理演算

2.1 現実の時空間と写像に対する仮定

1. 時空間を分割する集合が複数存在して、集合間の包含関係は未知である。
   集合はそれぞれの時空間の物質等に該当し、時空間方向にある程度の広がりを持つ。複数の集合は排他的な関係もありうるし、包含関係になることもあり、部分重複していることもある。これらすべての関係に対して、写像によって関連を導き出す。
2. 複数の集合の間で、確率の一致が観測されるもの同士は「写像」として連結できる。
   実集合のある部分空間に属していれば、別の部分空間がかならず確率的に一致する場合がある。この場合、実集合そのものの重複関係とは無関係に「写像」として双方の部分集合を含む包含集合を形成できる。この包含集合を「写像集合」とみなす。この「写像集合」が、一般的に因果関係と呼ばれるものを表現すると予想できる。写像元は写像集合に完全に含まれるが、写像先は写像集合に対して部分集合だけを含むことがある。これは「含意」の定義による。
3. 写像間の論理演算は、写像の成立条件となる。
   複数の写像先の集合的論理演算を行うことにより、論理演算の結果の集合が最終的な写像先の部分集合となる。これが写像で一般化された論理演算である。

現実の時空間は、このような構造になっていると仮定する。その上で、現実の集合と写像の構造がいかなるものであっても、観測からSOLで再現することを目指す。

2.2 ノードとリンク

SOLは、ノードとリンクとで構成される双方向ネットワークである。以下の全ての種類のノードはリンクで結合される。

1. 観測値の実体であるRealノード
2. リンクを等価結合するJointノード
3. リンク同士の論理演算を示すLogicノード
4. リンク間の排他性を示すExclusiveノード
5. リンクと結合して外部と入出力するためのFuncitionノード

Realノードは空間に存在する抽象的な部分集合であるが、スカラ、ベクトル値、文字などの実際の値をノードと一対一に対応させることも可能である。

論理演算ノードは、入力リンクに対するBool代数演算を行う。AND、OR、XORなどの種類がある。AND、OR、XORなどの論理演算は、複数の入力に対して論理演算と、集合演算の双方を適用する。出

力リンクは論理演算の結果を伝搬するだけではなく、出力リンク同士の値と集合とが等価であることを示す。

Jointノードは、出力のみを持つノードであり、複数のリンクの値と集合とが等価であることを示す。複数のリンクが等価であるということは、複数のリンクからの到達でフィードバックを発生させるということでもある。

Exclusiveノードは、入力リンク間の排他性を示すノードである。入力ノードすべての間のNOTリンクとほぼ等価であるが効率のために使用する。

Functionノードは、Realノードで示された実際の値を用いて、実際の演算や外部入出力を行う。結果に相当する数値などのRealノードをその都度生成する。

リンクは、リンクを通過する際に適用する二項組の確率 P^{f11},P^{f00} と経験数 N^{f11},N^{f00} とで構成される。

リンクの属性は以下の通りとなる。

内容	記号
リンクの順方向伝搬確率	P^{f11},P^{f00}
リンクの順方向経験数	N^{f11},N^{f00}
リンクの逆方向伝搬確率	P^{r11},P^{r00}
リンクの逆方向経験数	N^{r11},N^{r00}

Activation（活性値）は、複数のリンクと論理演算を通過して、起点からの伝搬確率を計算する。

2.3 リンク伝搬と論理演算

伝搬

リンク伝搬は、伝搬確率 P_{11}, P_{00} を使用して演算する。

$$P' = P_{11}P + (1 - P_{00})(1 - P)$$

NOT伝搬

リンクの伝搬確率を以下のように設定すると、リンク自体が論理値のNOTの挙動を示す。

$P_{11} = 0$
$P_{00} = 0$

AND演算

リンク伝搬の結果をAND論理演算で合成する。P_1,P_2...を入力して、P'を出力する。

$$P' = P_1 P_2 P_3 ... P_n$$

OR演算

リンク伝搬の結果をOR論理演算で合成する。P_1,P_2...を入力して、P'を出力する。

$$P' = 1 - (1 - P_1)(1 - P_2)(1 - P_3)...(1 - P_n)$$

XOR演算

リンク伝搬の結果をXOR論理演算で合成する。P_1,P_2...を入力して、P'を出力する。

$$P' = P_1(1 - P_2) + (1 - P_1)P_2$$

３変数以上のXORは再帰的に演算を適用する。

Exclusive演算

リンク伝搬の結果をExclusive演算に入力する。この場合は演算を適用せずに伝搬を行う。

$$P' = P_n$$

逆方向伝搬

リンクにはそれぞれ逆方向伝搬確率P_r^{11}, P_r^{00}が定義される。

通常のリンクの場合は、順方向伝搬確率から即座に求められる。順方向の伝搬確率を P^{f11} 、 P^{f00} 、逆方向の伝搬確率を P^{r11} 、P^{r00} とする。この場合、逆方向の伝搬確率P^{r11} 、P^{r00}は以下の式を用いて算出される。

$$P^{f11}P^{r11} + (1 - P^{f11})(1 - P^{r11}) = 1$$

$$P^{r11} = \frac{P^{f00}}{P^{f11} + P^{f00} - 1}$$

$$P^{r00} = \frac{P^{f11}}{P^{f11} + P^{f00} - 1}$$

OR,ANDなどの論理演算からの逆伝搬の場合は、ベイズの定理に従って算出される。通常のリンクの情報に加え、リンクの伝搬元確率を P 、伝搬先確率を $P^{'}$ とする。伝搬元確率と伝搬先確率の双方が得られなければ、逆方向伝搬確率は算出できない。

$$P^{r11} = \frac{PP^{f11}}{P'}$$

$$P^{r00} = \frac{(1-P)P^{f00}}{1-P'}$$

エントロピーとリンクの確定化

確率に対するエントロピーは、二値のエントロピーの公式から算出できる。

$$S = \sum_{n} P_n log P_n$$

これを二値化すると

$$S = PlogP + (1-P)log(1-P)$$

この結果、確率Pが1か0に近いものはエントロピーSが最小で、確率Pが0.5に近いものはエントロピーSが最大になる。SOLの目的は、リンクの確率を可能な限り1か0に近づけることである。つまり、エントロピーを最小化することが、SOLの自己組織化の目的の１つと言える。そのために、エントロピーの高いリンクは、適切な論理演算の挿入で分割してそれぞれのエントロピーを低下させる。

リンクへのフィードバック

リンクへのフィードバックによって、リンクを通過する集合ごとに複数のリンクに縦方向に分離する。全体確率が1か0に近づくポジティブフィードバックは、分離の必要がない。全体確率が0.5に近づくネガティブフィードバックは、リンクを通過する集合の中に伝搬確率1と0の部分集合が混在していると考える。

例えば、これまでは確率1で通過していたが、今回は確率0で通過した場合は、通過した確率0の伝搬集合は、リンクを縦方向に分離する部分集合と考える。この縦に分割する手段が以下の論理演算の挿入になる。

2.4 リンクの縦分割と論理演算挿入

ノードからノードへと伝搬するリンクは、起点ノードと終点ノードをそれぞれ部分集合に分割して、リンクを縦分割することができる。
リンクを分割するのは、リンクを通過する集合の中で、伝搬確率が異なる集合があると観測された場合である。伝搬された観測確率が0.5に近づく場合に、観測確率１の要素と観測確率０の要素とに分離する。
分離された要素は、要素の集合を示す起点ノードの情報を利用して、そのノードと論理演算を形成する。

- P^{11}側が不確定で、P^{00}側が1確定であればANDを挿入する
- P^{00}側が不確定で、P^{11}側が1確定であればORを挿入する
- P^{11}側が不確定で、P^{00}側が0確定であればNOT(AND)を挿入する
- P^{00}側が不確定で、P^{11}側が0確定であればNOT(OR)を挿入する
- P^{11}側とP^{00}側の双方が不確定であれば、XORの挿入の可能性がある

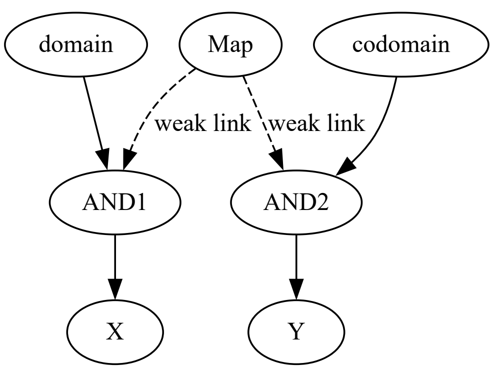
3 写像

3.1 写像の定義

論理演算は、起点を集合ととらえて、集合代数として記述できる。だが、より汎用的な論理を記述するためには、フリップフロップやメモリというような、様々な時間の状態を切り替えて使用するための記憶素子が不可欠である。そうでなければ、時間や空間の変動に対して汎用的な論理演算を作りだすことができない。

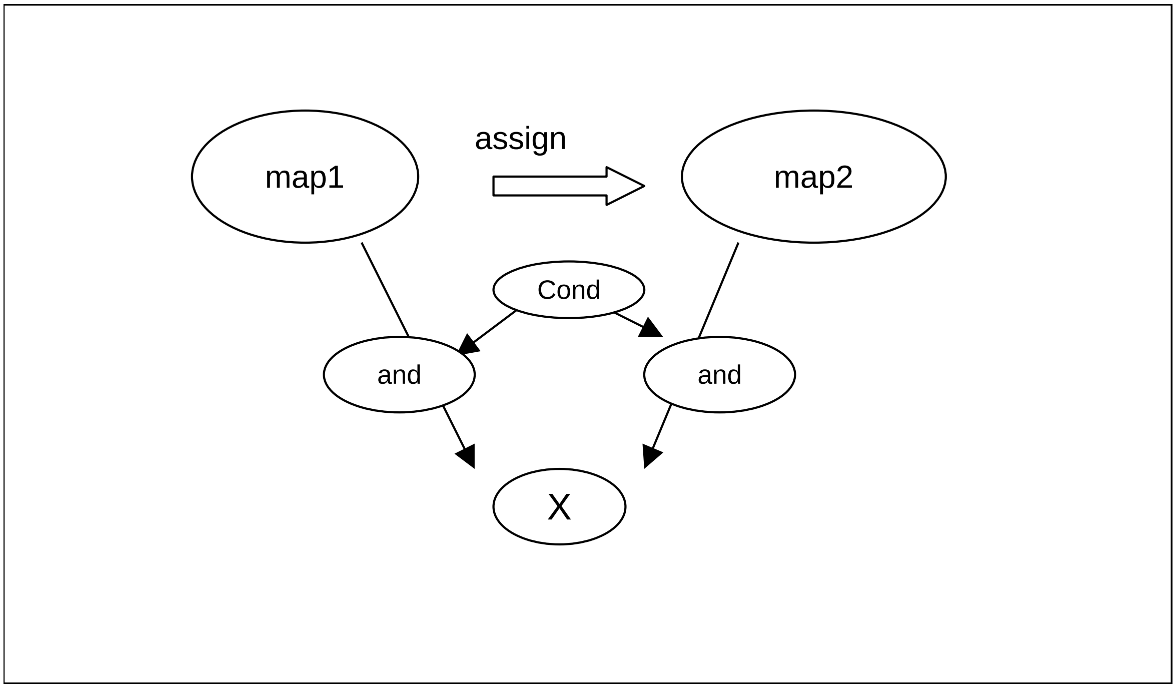
SOLは、論理演算に、写像概念を加えた。写像は、時間や空間の異なる状態を結びつけ、新たな起点として利用する。この写像の作用こそが記憶素子の一般化でもある。さらに、複数の状態の一致を状態として管理することを可能にする。これにより状態間の順序などの関係そのものを状態として利用することも可能にする。

写像を論理演算だけで表現する方法を示す。以下の例では写像元の部分集合Xから、写像先集合の部分集合Yに写像する。MapノードはJointノードの一種となる。



weak linkはN_{11}かN_{00}の観測数が少ないリンクであり、次の観測によるネガティブフィードバックによって不確定になる。このリンクの確定確率の部分を分離するために論理演算が追加されていくことになる。２つのweak linkのうちのどちらにも論理演算が追加できることになるが、整合性のある方の論理演算だけが確定することになる。

3.2 写像を通過する逆伝搬と代入



Assign between maps

この図は複数の写像の間で写像元と写像先を共有する方法である。Map1ノードから見るとXは値域であり、Map2ノードから見るとXは定義域である。重要なのは、Map1ノードの値域の条件となるCondノードは、Map2ノードの定義域の条件となるCondノードが同一である点である。

ここで、map1ノードからmap2ノードへと活性値を伝搬する。map1ノードでの活性値の値は１である。map1ノードからXへの伝搬は通常の伝搬であるが、XからMap2ノードへの伝搬は逆伝搬となる。ただし、Map2ノードで拡大された確率は均一に値が１であるという保証はない。だが、活性値の代入の考え方を用いれば、Map2ノードに到達した伝搬集合の確率が均一に１になりうるという理由を示す。

Map1ノード上に存在する確率１の活性値は、Map1ノードからXに向けて伝搬を行う。その際にノードCondでANDをとり、伝搬集合が分断される。だが、さらにXからMap2ノードへと伝搬される際に、ANDの逆伝搬を通過する。その際に、Map1ノード側と同じCondノードで補完される。集合としてはMap1ノードの活性値と同じ大きさになる。Map2ノードの活性値は０と１の値のどちらかを完全に特定することはできない。だが、値１からのANDの逆伝搬であるので、Map2ノードの活性値は伝搬集合すべての値が１と仮定することは可能である。値が０と仮定することはできない。

ここで、Map1ノードにある活性値が、Map2ノードの活性値と同一であると仮定できる。これがSOLにおける代入であると考える。そのための条件は以下の通りである。

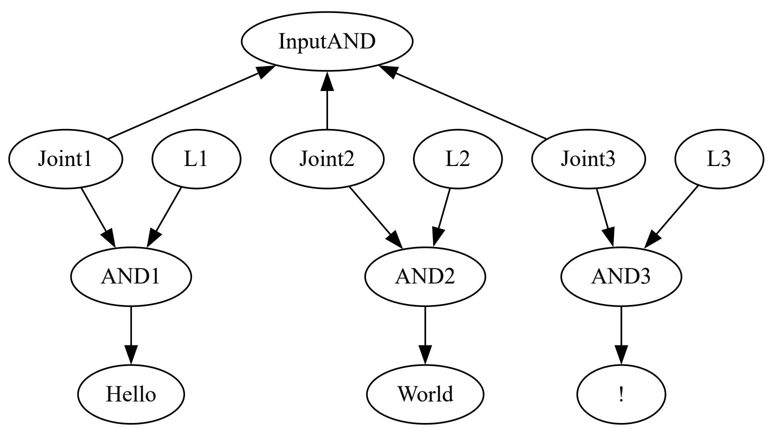
1. 伝搬元の値と、伝搬先の確定部分の値が等しいので矛盾がない。
2. ２つの伝搬集合の仮定ベクトルが同一である。

この考え方は厳密にいえばオッカムの剃刀である。だが、この逆伝搬に見えない条件がある場合は、今後の観測でネガティブフィードバックが発生して条件の追加が行われることになる。

3.3 写像と双方向伝搬による階層的なマッチング

一例として、Input入力からのテキスト解析を挙げる。以下の例では、互いに排他的な単語が３つが、互いに排他的なノードL1,L2,L3を用いて活性化される。一致判定は、単語から逆伝搬してInputANDが逆に活性化される。L1,L2,L3は互いに排他的なことが確認されればよい。そしてAND1からJoint1への逆伝搬は集合的に拡大されて排他的でなくなる。Joint1,2,3からのANDが結合されてInputANDが活性化される。逆に言えばすべての要素がなければInputANDは活性化されない。

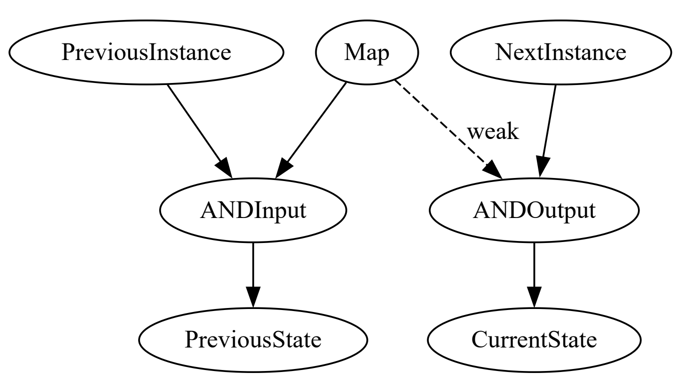
このように、双方向伝搬を用いれば、集合的に重複のない単語間での集合論理演算が可能になる。



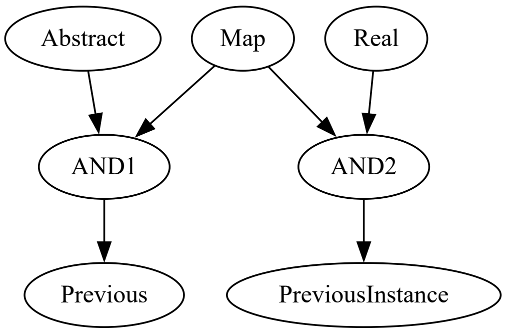
3.4 序列と時空間

順序リンクの写像ノードによる表現方法

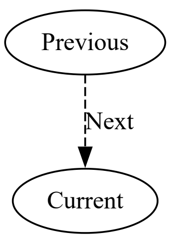
集合と写像を用いて順序の表現するには以下の手法をとる。PreviousStateとCurrentStateは連続した時間のステートであり、これらは写像で連結される。写像の条件であるNextInstanceノード,PreviousInstanceノードは共に写像ごとに生成されるノードである。



PreviousInstanceノードは抽象的なPreviousノードの部分集合であるが、Previousとは写像で連結される。ある写像でのNextは、別の写像にとってはPreviousでありうるため、PreviusInstanceノードは、集合の実体としては別の写像のNextInstanceノードとも重複があり得る。そのため、抽象的な空間にのみ存在するノードであるPreviousノードと写像で連結するという形にする。NextInstanceノードも同様である。

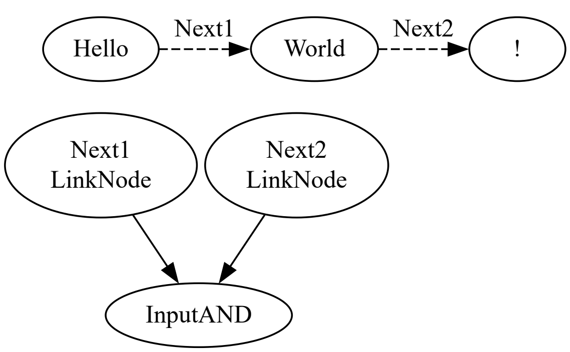


必要に応じてこれらの記述を省略して、以下のように表記する。



順序を含む単語のリンクノードによる一致判定

以下の例では、互いに排他的な単語が３つが、相互に順序を持つノードL1,L2,L3を用いて活性化される。Next1はL1,L2の順序が成立したときに活性化される。一致判定は、Joint1,Joint2,Joint3ではなくNext1,Next2のそれぞれのリンクノードからInputANDへと入力される。前の例と異なり、単語の順序が入れ替わるとNextのリンクノードが活性化されない。



このような序列の表現方法は効率が悪いように見えるかもしれないが、時空間の構造に左右されない厳密な方法である。実装次第ではベクトルや行列に代入することで効率化を行うという方法も考えられるが、ベクトルへの代入の時にこの手法で序列に従って代入する。

4 活性値の双方向伝搬

SOLは、ニューラルネットワークと類似する活性値の伝搬を行うが、起点からの伝搬確率を厳密に管理する。それに加えて、仮定ベクトルによって伝搬された集合の情報を管理する。これに加え、SOLでは論理演算の双方向への伝搬とそれによる写像の表現を可能にした。これによって、集合の空間上のあらゆるノードの間で、ネットワークで接続されている限りは伝搬確率を求めることが可能になる。

SOLのネットワークのリンクに対して、活性値を示す以下のActivationのオブジェクトをリンクごとに分配して伝搬し、階層的にActivationを生成する。Activationの持つ情報は基本的には以下の通りである。

1. PropagatedProbability

伝搬された確率値であり、値は0から1までのスカラ値である。

2. Assumption Vectors

このActivationの伝搬集合を示す仮定ベクトルである。

3. Propagating link

このActivationが通過しているリンクを示す。フィードバックはこのリンクに対して適用される。

4. Parent activations

伝搬元のActivationの情報を示す。論理演算ノードでは複数のActivationが合成される。Jointノードなどでは伝搬元のActivationを１つだけ継承する。

4.1 活性値の伝搬の流れ

1. 仮定からの確率伝搬
   活性値は仮定を起点とする。仮定は起点となるノードの起点確率を１か０に仮定する。この起点確率に対してリンク伝搬や論理演算を適用して活性値を伝搬させていく。
2. 仮定ベクトル同士の論理演算と複数の仮定ベクトルの統合
   異なる仮定ベクトルを持つ複数の入力を用いて論理演算を行う際に、仮定ベクトルが示す集合の重複部分だけを抽出して伝搬する。仮定ベクトルが重複しない部分は論理演算は適用されない。
3. 論理演算の逆伝搬と仮定ベクトルの相補的合成
   論理演算の出力側の集合を、論理演算の入力側に逆に伝搬する。
   値１をANDから逆伝搬させるときはそのまま全ての入力に対して確率１と仮定ベクトルを伝搬す

る。
値０をANDから逆伝搬させるときは、１つの入力を除いてすべての１の入力がある場合は、入力が０として確定する。
値０をANDから逆伝搬させるとき、さらに入力の中にすでに０がある場合は、逆伝搬する入力は確率が不確定となる。だがここで、不確定な確率を確定させる手段が別に存在する。これが逆伝搬を写像に用いることができる理由である。

4. 活性値衝突と、仮定ベクトルの要素比較
   複数の活性値が別の経路を通過して同一のノードに到達した場合、双方の仮定ベクトルの間に共通部分があれば双方の確率を比較する。確率が異なっていれば確率を等しくするようにフィードバックを行う。
5. 活性値が組み込み関数に到達することで、外部からの観測や外部動作などを行う。観測とは入力集合に対応した出力の活性化である。

4.2 仮定ベクトル

仮定ベクトル要素

仮定ベクトルの要素はそれぞれ、特定のノードのBool値が１か０であるという仮定を意味する。ノードの値により、仮定ベクトルの持つ伝搬集合が２つに分断される。

観測からの仮定ベクトル生成

観測は典型的には時空間の値と観測する物理的な値との連想である。この連想のMapノードに対して値を仮定することにより、仮定ベクトルの起点となる。仮定ベクトルはMapノードから実体のある値へとそれぞれ伝搬されて、その値を用いるネットワークへと伝搬されていく。

論理演算の相互作用

伝搬集合同士でAND、ORなどの論理演算を行うと仮定ベクトルが合成される。この際の合成は以下のルールに従う。

- 仮定ベクトル間で異なる仮定ベクトル要素はそのまま合成される。
- 仮定ベクトル間で同一の仮定ベクトル要素は１つに集約される。
- 集合的に包含関係の仮定ベクトル要素との合成は伝搬集合的に小さい要素に取って代わる。
- 集合的に排他的な仮定ベクトル要素との合成は伝搬集合自体が空集合になる。

活性値の衝突と仮定ベクトル

複数の活性値が同一のJointノードや論理演算ノードに到達した場合は、それらの活性値は等価なものとして衝突対象となる。論理演算に関して言えば、入力の合成結果と出力とが衝突対象となる。

仮定ベクトルが示す伝搬集合に重複があれば、伝搬された結果の確率値が一致していなくてはならない。一致していない場合はフィードバックが実行される。

4.3 活性値の確率伝播

活性値(Activation)をリンクに沿って伝搬して、リンクの伝搬確率やノードの確率合成に従って確率を伝搬させ、リンクの先の組み込み関数を実行するのがSOLの基本となる。

P^{f11} はPの確率１からP'の確率１への伝搬確率であり、 P^{f00} は、Pの確率０からP'の確率０への伝搬確率になる。確率論の自明な適用である。

$$P' = PP^{f11} + (1-P)(1-P^{f00})$$

AND演算では複数のActivationの確率を統合して、出力リンクに伝搬する。確率計算は、入力された伝搬確率 P_1 , P_2 , P_3... を複数利用して、以下のように計算する。

$$P' = P_1 P_2 P_3 ...$$

OR演算の確率計算は以下のようになる。

$$P' = 1 - \{(1-P_1)(1-P_2)(1-P_3)...\}$$

出力リンクの数が膨大となる場合は、いくつかのリンクに限定してActivationを伝搬することになる。リンクの限定の方法は別に提供する。

4.4 確率の逆伝搬

Activationは、リンクの方向と逆にたどる逆伝搬(backward activation)も可能である。ニューラルネットワークの技法のback propagationとは意味が異なる。

逆伝搬は、リンクごとの逆伝搬確率 P^{r11},P^{r00} を利用して、順方向伝搬の確率計算と同じ方法で行う。ノードの論理演算は適用されない。

$$P' = PP^{r11} + (1-P)(1-P^{r00})$$

結果、全体の伝搬確率は、以下のような複数のリンクに対する多項式として一般化される。

$$P = f(P_1^{f11}, P_1^{f00}, P_2^{r11}, P_2^{r00}, P_3^{f11}, ...)$$

4.5 活性値の衝突

活性値の衝突とは、同一ノードを起点として同一ノードへと到達した複数の伝搬経路で、伝搬された仮定ベクトルが重複しているにもかかわらず伝搬確率が異なることを言う。

原則として、$P = 0.5$、つまり確率不確定な伝搬集合との部分重複は、伝搬集合としては無関係とみなせるので衝突ではない。逆に、確率が双方ともに確定であれば、伝搬集合に重複があれば確率の衝突と見なすことができる。確率0.5を確率1（あるいは0）側が伝搬集合的に完全に包含する場合は、確率1側の確定性に矛盾が発生するので、衝突とみなされる。

以上の手段で伝搬集合の大きさの比較が行われ、伝搬集合が完全に重複している部分に対して確率の衝突、フィードバックが適用される。

5 確率変動分配階層フィードバック

このフィードバックの手法は、ベイジアンネットワークの無制限の階層に対して、どのリンクが観測確率の誤差の原因になっているかを的確に発見できる、新たに開発した手法である。

従来のニューラルネットワークでは、バックプロパゲーションを用いて深いリンク層に対して影響を及ぼすことができるが、より深いリンク階層に対する誤差原因の特定はやはり難しくなる。ディープラーニングと呼ばれるものも、相対的に階層が深いと言っているに過ぎない。

このフィードバックの前提は以下の通りである。

1. SOLのリンクには伝搬確率があり、論理演算もまた確率計算となる。複数のリンクと論理演算とを通過した結果の全体伝搬確率が求められる。観測された新たな伝搬確率P'に対して、ネットワーク伝搬確率を近づけるようにリンクごとのΔP_nに対してフィードバックされる。

$$P' = P_1P_2 ...（P_n + \Delta P_n）...P_{x-1}P_x = Q_n(P_n + \Delta P_n)$$

2. それぞれの確率PのリンクがP'になる変動発生確率は、過去の観測確率Pと観測数Nとに依存し、純粋な確率論で算出できる。
3. リンク全体の変動発生確率は、すべてのリンクの変動発生確率の積となる。この全体変動発生確率を最大にする。
4. 個々のリンクにはフィードバックごとに観測数Nが加算される。フィードバックされた伝搬確率を用いて、観測数に応じて既存の伝搬確率を補正する。観測数が多いリンクはフィードバックによる補正もわずかとなる。

以上の条件を満足するように、計算方法が決定された。この計算方法を確率変動分配階層フィードバックと呼んでいる。

5.1 リンク上の二項組確率伝播

伝搬する活性値は、１つの伝搬確率Pを持つ。この確率Pは、現在運搬している仮定ベクトルの集合の中に存在する確率となる。補集合に存在する確率は(1-P)になる。

活性値が確率的に確定したリンクを通過するときは、以下の伝搬確率に従って確率が伝達される。

$$P^{11} = 1$$

$$P^{00} = 1$$

確率的に確定したNOTリンクは、正集合と補集合の２つの集合の確率を入れ替えることになる。具体的には、リンクの伝搬確率が以下の通りになる。通過する仮定ベクトルの集合を反転させるもので

はないことに注意する。

$$P^{11} = 0$$

$$P^{00} = 0$$

リンクは、P^{11}, P^{00} とは別に、リンクの強度を示すN^{11} , N^{00} というパラメータを持つ。この強度は、フィードバックごとに加算され、リンクの確率を確定させる。

5.2 双方向伝搬と論理演算の確率計算

SOLでの活性値伝搬における、確率計算の手法を示す。一般論として、伝播確率の計算方法は以下のようになる。

$$P' = PP^{11} + (1 - P)(1 - P^{00})$$

AND演算

$$P'_1 = P^1_1 P^2_1 P^3_1 ...$$

$$P'_0 = 1 - P'_1$$

OR演算

$$P'_0 = P^1_0 P^2_0 P^3_0 ...$$

$$P'_1 = 1 - P'_0$$

リンクの逆伝搬確率は、リンクの順方向確率からベイズの定理を適用して算出する。 P_1,P_0 はノードAの確率。 P^{f11},P^{f00} はノードAからBへの順方向伝搬確率。 P^{r11},P_{r00} が求めるノードBからノードAの逆方向伝搬確率とする。

$$\{P^{f11} + (1 - P)(1 - P^{f00})\}P^{r11} = P_1 P^{f11}$$

$$P^{r11} = \frac{P^{f11}}{P^{f11} + P_0(1 - P^{f00})}$$

$$\{P^{f11}(1 - P^{f11}) + (1 - P)P^{f00}\}P^{r00} = (1 - P)P^{f00}$$

$$P^{r00} = \frac{1 - P^{f00}}{P^{f11}(1 - Pf^{11}) + (1 - P)P^{f00}}$$

AND演算、OR演算などの論理演算から入力への逆方向伝搬もまたベイズの定理を用いて算出する。ただし、リンク入力の観測確率P(A)と論理演算出力の観測確率P(B)を用いる必要がある。

$$P^{r11} = \frac{P(A)P^{f11}}{P(B)}$$

$$P^{r00} = \frac{(1 - P(A))P^{f00}}{1 - P(B)}$$

以上のように、SOLのノードごとの伝搬確率合成は、確率論に忠実で自明なものである。シグモイド関数などの非線形要素は利用しない。

5.3 フィードバックの適用対象

現在利用しているネットワーク全体の伝播確率で算出された P_{total} を、同じ状況で確率が異なる衝突相手の伝搬確率 P'_{total} に合わせるのがフィードバックとなる。
SOLのフィードバックは、衝突した２つのパスが通過するリンクの、 P^{11} , P^{00} それぞれに対して適用される。その強度は、リンクの確率、リンクの経験数に応じて変わってくる。

フィードバックされる結果となる $P^{11'}$, $P^{10'}$ を、補正値 ΔP_n^{11} , ΔP_n^{00} を使って補正する。
２つのパスの全体確率を、それぞれ補正値を追加した多項式として伝搬させて計算する。大規模な方程式になる。

$$P_n^{11'} = P_n^{11} + \Delta P_n^{11}$$

$$P_n^{00'} = P_n^{00} + \Delta P_n^{00}$$

$$P_{total} = (P_1^{11'})(P_2^{11'})(1 - P_3^{11'})...$$

２つのパスA,Bそれぞれの全体確率は、大量の ΔP_n^{11} や ΔP_n^{00} に対する伝搬多項式になる。さらに、この２つのパスの式が等しくなることより、

$$P_{totalA} - P_{totalB} = \Delta P_{total} = 0$$

この式を満たし、かつ変動確率が最小となるΔP_nをN個求めればよい。だが、まともな方法では、ΔP_n をそれぞれ変動させて解を求めるという、組み合わせ爆発問題となる。

そのため、それぞれの確率観測確率の変動量 ΔP_n を近似的に求める手段を提供する。

5.4 伝搬確率Q_nの算出

元のリンクの伝搬結果をPとし、観測結果をP'とする。リンクを伝搬することで、それぞれのリンクの ΔP_n に対する多項式が生成される。ここからは、 ΔP_n は、 ΔP_n^{11} と ΔP_n^{00} とを一括で扱うものとする。以下は一例である。

$$P_{total} = P_1(1-P_2)...(1-P_{n-1})P_nP_{n+1}...P_{x-1}P_x$$

$$P'_{total} = （P_1+\Delta P_1）（1-P_2-\Delta P_2）(P_3+\Delta P_3)...（1-P_{n-1}-\Delta P_{n-1}）（P_n+\Delta P_n$$

$$...（P_{x-1}+\Delta P_{x-1}）（P_x+\Delta P_x）$$

$$= P + \sum Q_n \Delta P_n + ...$$

$$Q_n = P_1(1-P_2)...(1-P_{n-1})P_{n+1}...P_{x-1}Px$$

全体確率P'_{total}は多変数Taylor展開で近似される。ここで、リンクに対して二次以降、２つ以上の組み合わせの項は省略して近似式とした。確率は1以下なので確実に収束する。その結果、全体確率P'_{total}は ΔP_nのベクトルの内積の形になる。そして、P_nを除く伝搬確率Q_n こそが、それぞれのリンク ΔP_n に対する作用係数となる。ORやNOTなどが入るとより複雑になるが近似としては同等とみなせる。

$$\sum_n Q_n \Delta P_n \approx P'_{total} - P_{total}$$

5.5 重み値w_nの式の導出

伝搬で通過したリンクにそれぞれフィードバックされる確率補正値 ΔP_n は、伝搬リンクの確率をすべて以下の形で書き換えて計算することで、それぞれ求められる。

$$P'_n = P_n + \Delta P_n$$

リンクnの伝搬確率 P_n に対して、観測結果としての伝搬確率 P'_n が観測される変動発生確率 t_n は、一種の二項分布に従い、以下の式で得られるとする。

$$t_n = \lim_{m\to\infty}\{\binom{m}{mP'_n}P^{mP'_n}(1-P_n)^{m(1-P'_n)}\}^{-m}$$

この式は具体的に言えば、確率P_nのコイントスをm回利用して、結果の合計がmP'_nとなる確率を求め、mに対する極限をとっている。以下のコンビネーションの式を利用している。

$$\binom{m}{mP'_n} = \frac{m!}{mP'_n!m(1-P'_n)!}$$

さらに、リンクの過去の観測回数N_nを乗数することで、結果が$mN_nP'_n$となる確率を求める。これが、確率 P_n が複数回観測されたリンクでの変動発生確率になる。

$$T_n = \lim_{m\to\infty}\{\binom{mN_n}{mN_nP'_n}P^{mN_nP'_n}(1-P_n)^{mN_n(1-P'_n)}\}^{-m}$$

まずは、コンビネーションの値の計算には、以下の「スターリングの近似式」を使う。

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\{1-\frac{1}{12n}+\frac{1}{288n^2}+...\}$$

最後の1以外の定数は近似として省略して、以下の式を用いる。

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

これを用いてコンビネーションの式に代入すると、

$$\binom{mN_n}{mN_nP'_n} = \frac{mN_n!}{(mN_nP'_n)!\{mN_n(1-P'_n)\}!}$$

$$\sim \frac{\sqrt{2\pi mN_n}}{\sqrt{2\pi mN_n 2\pi mN_nP'_n(1-P'_n)}}\left(\frac{N_n}{e}\right)^{mN_n}\left(\frac{N_nP'_n}{e}\right)^{-mN_nP'_n}\left(\frac{N_n(1-P'_n)}{e}\right)^{-mN_n(1-P'_n)}$$

$$= \frac{1}{\sqrt{2\pi mN_nP'_n(1-P'_n)}}\frac{1}{P_n'^{mN_nP'_n}(1-P'_n)^{mN_n(1-P'_n)}}$$

このコンビネーションの値を使って、T_nに代入する

$$T_n = \lim_{m\to\infty}\{P_n^{mN_nP'_n}(1-P_n)^{mN_n(1-P'_n)}P_n'^{-mN_nP'_n}(1-P'_n)^{-mN_n(1-P'_n)}$$

$$\{2\pi mN_nP'_n(1-P'_n)\}^{-1/2}\}^{-m}$$

$$= P_n^{N_nP'_n}(1-P_n)^{N_n(1-P'_n)}P_n'^{-N_nP'_n}(1-P'_n)^{-N_n(1-P'_n)}\lim_{m\to\infty}\{2\pi mN_nP'_n(1-P'_n)\}^{-m/2}$$

このT_nこそがリンクでP'_nが観測される確率であり、この確率変動をすべてのリンクで平均化するようにP_nの変動を求めれば良い。両辺logを取って、左辺をτ_nで置き換える。最後の項はmの極限で収束するので省略する。

$$\tau_n = \log T_n = N_nP'_n\log P_n + N_n(1-P'_n)\log(1-P_n)$$

$$-N_nP'_n\log P'_n - N_(1-P'_n)\log(1-P'_n)$$

$$= N_n\{P'_n\log P_n + (1-P'_n)\log(1-P_n) - P'_n\log P'_n - (1-P'_n)\log(1-P'_n)\}$$

このτ_nの式に対してP_nで微分を行う。

$$\frac{d\tau}{dP_n} = N_n\{\frac{P'_n}{P_n} - \frac{1-P'_n}{1-P_n}\}$$

$$= N_n\{\frac{P'_n(1-P_n)-(1-P'_n)P_n}{P_n(1-P_n)}\}$$

$$= N_n \frac{P'_n - P_n}{P_n(1-P_n)}$$

$$P'_n = P_n + \frac{P_n(1-P_n)}{N_n}\frac{d\tau_n}{dP_n}$$

これにより、τの係数としてのP_nの変動の重みw_nが計算できる。

$$P_n + \Delta P_n$$

$$\Delta P_n = w_n \Delta\tau_n$$

$$w_n = \frac{P_n(1-P_n)}{N_n}$$

こうして、それぞれのリンクのウェイトw_nはリンクの既知の確率P_nと観測回数N_nとで決定された。

5.6 確率補正値 ΔP_n の算出

ここからは複数のリンクへのフィードバック分配方法について述べる。観測された全体確率を、P'_{total} とする。この確率に合致するように、それぞれのリンクの変動 ΔP_n を求める。Q_n は、前述したとおりリンクnに対する伝搬確率で決定される伝搬係数である。

$$P'_{total} = P_{total} + \Delta P_{total} = P_{total} + \sum_n Q_n \Delta P_n$$

さらにもう一つの制約条件は、全体変動発生確率 T を最大値にすることである。

$$t_{all} = \prod_n t_n$$

この式はτ_nを用いると以下のようになる。

$$T = \log t_{all} = \sum_n \log t_n = \sum_n \tau_n$$

この全体変動発生確率のlogを取ったT（ラージタウ）を最大にして、かつリンクごとのτ_nの値を均一に分配するように、リンクごとのP_nの変動値を定めるのがフィードバックの目的である。

$$\Delta P_{total} = \sum_n Q_n w_n \frac{d\tau_n}{dP_n}$$

そのために、それぞれのリンクnの変動発生確率τ_nを共通パラメータτで置き換える。つまり、重みに応じて均等分配を行うことになる。

$$\Delta\tau = \Delta\tau_n$$

それぞれの ΔP_nに対して、変動全体 ΔP_{total} を$Q_n w_n$の比率を用いて分配する。

$$\Delta P_{total} = \sum_n Q_n w_n \Delta\tau$$

$$\Delta\tau = \frac{\Delta P_{total}}{\sum_n Q_n w_n}$$

こうして求められた$\Delta\tau$を用いて、全体のフィードバック ΔP_{total} を、それぞれのリンクの ΔP_n に対して分配することができる。

$$P'_n = P_n + \Delta P_n = P_n + w_n \Delta\tau$$

P'_nが1以上や0以下になることもありうるので、そういう場合はP'_nを1か0に飽和させてP_{total}をその都度再計算する。そののちに飽和したリンクを除去して再び$\Delta\tau$を計算する。

5.7 フィードバック観測数Nを用いたリンクの伝搬確率補正

このフィードバック補正された確率は、そのままではリンク自体が更新する確率にはならない。リンクのそれまでの観測数 N_n が大きければ、それだけリンク確率補正量が小さくなる。フィードバックの加算値は１ではなく、実際の伝搬に使用された伝搬確率Q_nとフィードバックされる相手の観測数N_fによって決定される。

$$N'_n = N_n + N_f Q_n$$

結果、最終的に更新される確率P''_nは以下の通りになる。

$$P''^{N'_n}_n = P_n{}^{N_n} P'_n{}^{N_f Q_n}$$

乗算平均と加算平均はそれほど差がないため以下の式で近似を行うこともできる。

$$P''_n = \frac{N_n P_n + N_f Q_n P'_n}{N'_n}$$

P_nはリンクのP^{11}, P^{00}が混在しており、フィードバックはリンクのP^{11}, P^{00}それぞれに対して適用される。

算出された確率 P_n^{11} へのフィードバック結果に基づき、 P_n^{11} がエントロピー減少方向、つまり確率が1か0のいずれかに近づいたリンクは、ポジティブフィードバックされたとみなす。逆に、 P_n^{11} のエントロピーが上昇する方向、つまり確率0.5に近づいたリンクをネガティブフィードバックされたとみなす。ネガティブフィードバックは、そのリンクになんらかの見えない条件があるために発生したと考え、その条件を検索する。P_n^{00} についても同様にフィードバックを算出する。

リンクの利用頻度が0の場合は、確率 $P_n = 0.5$ とみなされる。ポジティブフィードバックだけでは、確率1か0に近づくことはできるが、確率1か0そのものにはなれない。

この条件リンク形成については、後述する自律生成写像論理回路アルゴリズムの一部として実装される。

6 連想

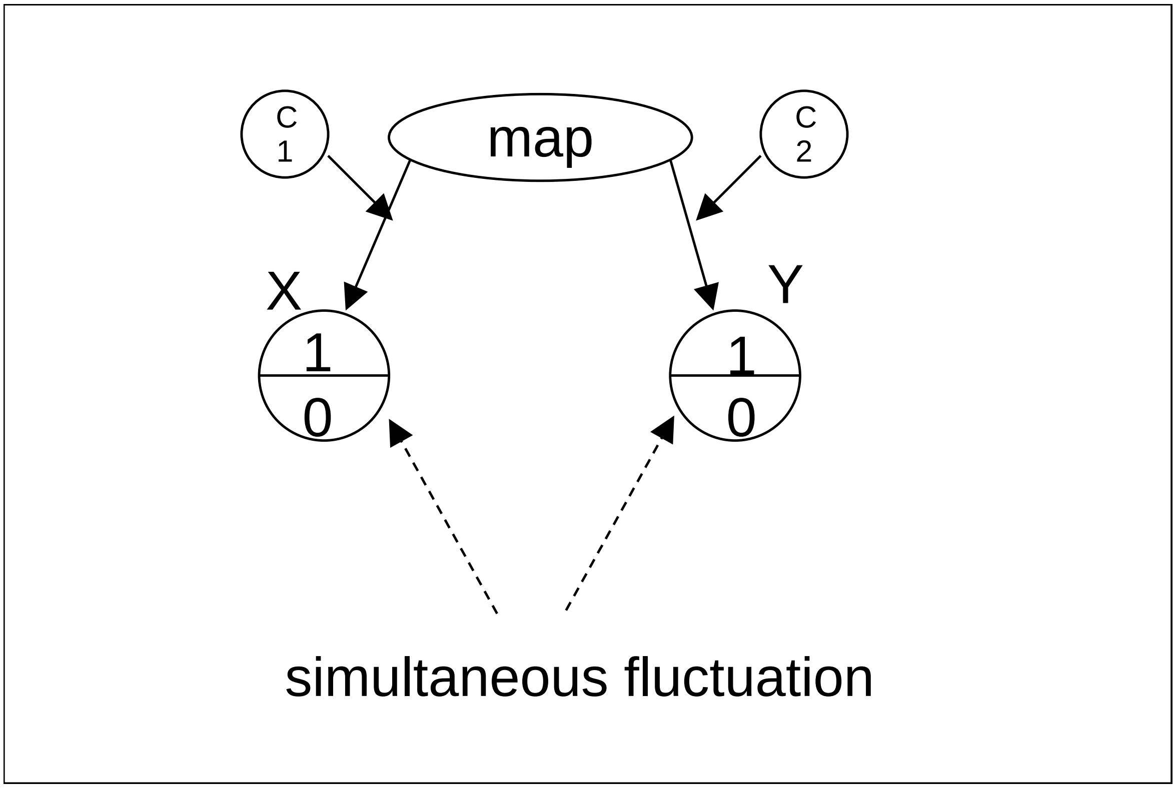
6.1 連想対象選択

連想とは、集合的に独立した２つのノード間を写像で連結して因果関係を予測する作用である。あらゆる事象の間で、因果関係がありうる関連を導き出すには、同時に変動したものだけを選択するのが最も効率が良いためである。ただしこの場合の変動は、時間軸、座標などあらゆる変動を対象とする。
２つのノードで、値の変動が同時に観測されることにより、連想の候補とする。すべての変動が同時に観測されていれば確実に連想が可能になるが、たとえすべての値の変動を観測していなくても、部分的な観測だけでも連想の可能性がある。具体的には、それぞれのノードの変動発生の確率によって同時変動の確率、つまりは連想の価値を定量的に定めることができる。

そのための条件は以下の通りである。

1. 同時に観測された確率エントロピーがほぼ等しい（多くの場合は、確率１か０のどちらかの確定確率）。
2. 同一起点からの仮定ベクトルが同一か、あるいは包含関係がある。同時変動はそれ自体が仮定ベクトルが完全に一致しているか包含関係がある場合とみなされる。

ここで、関連がない事象同士が偶然同時活性化されて連想対象になることがありうるが、偶然の連想は複数回の観測でフィードバックされて不確定なリンクとなる。



Create association

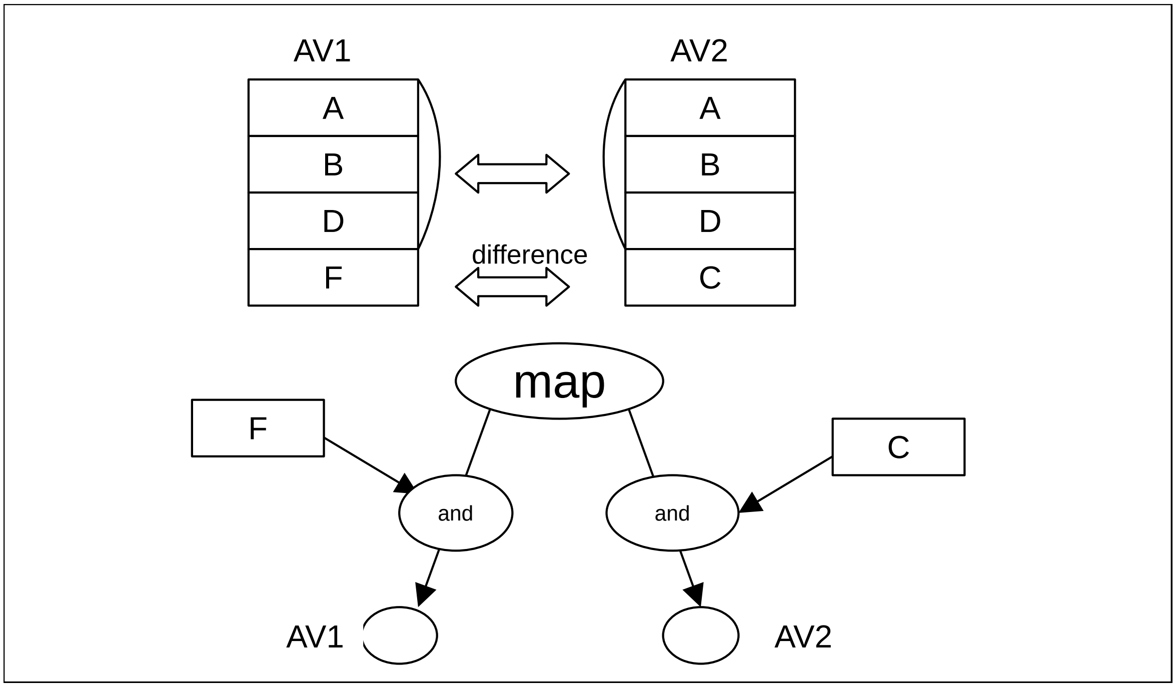
フィードバックからの連想候補

　連想の候補となるノードは、ネガティブフィードバックが適用されたリンクの終点となる。ネガティブフィードバックが適用されたことにより、そのリンクには何らかの条件が追加されていると見なすことができる。
このネガティブフィードバックが同時に適用されたリンク同士が、連想の可能性があると見なされる。ただし、単純にネガティブフィードバックされた候補となるノードは大量に存在しうる。それらの関連をすべて連想を行うのは非常に効率が悪いと考えられる。では意味のある連想だけを的確に選択するにはどうするのか？そのための方法として、ノードの変化の確率が低く、かつ同時に変動が観測されたものを最優先で選択する。

典型的な例としての観測結果による連想候補

　観測とは、観測された状態を示すノードの部分集合が活性化されることである。この活性化は、典型的には現在の時空間をノードと観測値を示すノードとの写像連結を行う作用であり、基本的に観測ごとに生成される。

6.2 連想形成と条件の付加



Difference between assumption vectors

連想はmapからのandを行うための条件である２つのノードを選択、あるいは生成する必要がある。これらの条件ノードは、仮定ベクトルの差分から生成される。上の図ではAV1とAV2との差分であるノードF,ノードCがそれぞれ条件となる。

6.3 伝搬確率、経験数の算出

連想リンクは確定なリンクではなく、観測された同時活性化の確率からその連想の経験数Nを決定することができる。

$$N^{11} = log_2(P_Y)$$

この算出方法は以下の通りである。未計測で不確定なリンク、つまり想定伝搬確率$P = 0.5$の連想リンクにおいて、現実に$P = 1$がN^{11}回観測された結果、連想対象のYへのリンクの過去の伝搬確率P_Yが計測されたと考える。

$$0.5^{N^{11}} = P_Y$$

つまりは、$P = 0.5$がN回観測されたリンクとみなすことができる。

X=1,Y=1が観測された時の連想確率は$P' = 1$となる。その結果、初期値となるP^{11}, N^{11}は以下の式になる。X=1,Y=0の時は$P = 0$となる。

$$N^{11} = log_2(P_Y)$$

$$P^{11} = \frac{0.5 + P'N^{11}}{1 + N^{11}}$$

X=0,Y=0が観測された時は、P^{00}側にのみ作用する。この場合も連想確率は$P' = 1$となる。その結果、初期状態となるP^{00}, N^{001}は以下の式で表される。X=0,Y=1の時は$P' = 0$となる。

$$N^{00} = log_2(1 - P_Y)$$

$$P^{00} = \frac{0.5 + P'N^{00}}{1 + N^{00}}$$

この経験数の算出は、めったに発生しない事象間の変動連想は関連の確実性も高いということを意味する。

7 確率的自律論理生成アルゴリズム

　このアルゴリズムは、ノードとリンクを自律的に生成、修正するための一般的な手法である。

7.1 「条件」リンク分割と論理演算ノード挿入

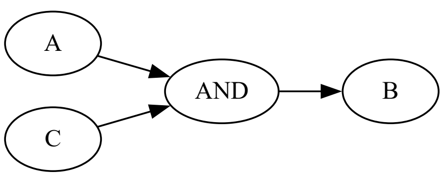
ネガティブフィードバック

　リンクへのネガティブフィードバックを起点にして、リンクを補正するための条件付けを行う。ネガティブフィードバックとは、現在のリンクの確定値と逆方向の値へのフィードバックが発生したということであり、定量的にはリンクのエントロピーを上昇させるフィードバックである。

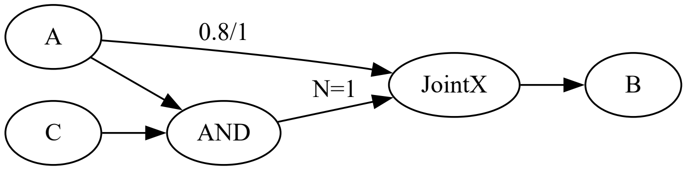
　フィードバックが行われたリンクはノードAからノードBへのリンクであるとする。以下の例では、P^{11} の伝搬確率がフィードバックによって0.8に減少したとみなす。

$$A \rightarrow B \quad \begin{cases} P^{11} = 0.8 \\ P^{00} = 1 \end{cases}$$

　このフィードバックに対応する条件Cを使用して、条件を付加する。条件はANDとなる。

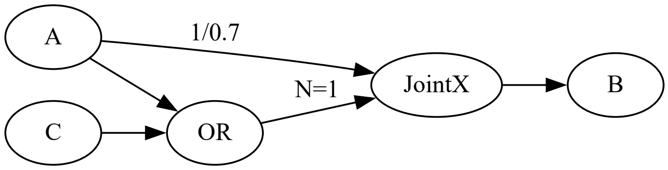


　正確には、条件Cは一観測事実であって、既存のリンクの全てを代替できないので、既存のリンクを維持してJointYで結合することになる。
新規のリンクは、経験数はN=1とする。



　P_{00} に対してネガティブフィードバックが行われたら、その場合は条件ノードCをORで結合する。

$$A \rightarrow B \quad \begin{cases} P^{11} = 1 \\ P^{00} = 0.7 \end{cases}$$



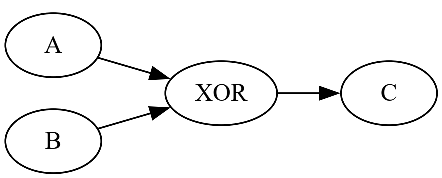
フィードバックによって完全に双方の伝搬確率が反転したリンクもありうる。すなわち、双方とも伝搬確率0付近となる、NOTリンクとなる。

$$A\neg \to B \quad \begin{cases} P^{11} = 0.1 \\ P^{00} = 0 \end{cases}$$

P_{11},P_{00} 双方に対してネガティブフィードバックが行われた場合は、リンクはランダムとなり、基本的には無効となる。ただし、２つのリンクで P_{11},P_{00} のネガティブフィードバックが同時に行われた場合は、XORノードでの結合の候補となる。

$$A \to C \quad \begin{cases} P^{11} = 0.5 \\ P^{00} = 0.5 \end{cases}$$

$$B \to C \quad \begin{cases} P^{11} = 0.5 \\ P^{00} = 0.5 \end{cases}$$

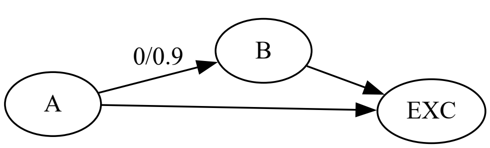


条件の検索

フィードバックに付加される条件となるノードとは、つまりフィードバックのときに通過した活性値の部分集合に相当する。この部分集合を表現する手段が仮定ベクトルとなる。

exclusiveノードの形成

Exclusiveノードは２変数においてはXORと同じ挙動を示すが、多変数ではすべての要素が完全に排他的であるという点が異なる。このExclusiveノードの目的は自然界に頻発する排他的な事象を効率的に表現するためである。P11=0が条件だが、P00は0以上（典型的には1近く）である必要がある。



7.2 「因果」ポジティブフィードバックからのリンクノード活性化と、連想対象化

以下のリンクへのフィードバックが発生した場合、リンクの前後のノードA,Bから、リンクノードが生成される。

$$A \to B$$

リンクノードと等価な論理演算

AからBへのリンクにフィードバックされた結果、活性化されるのはリンクノードLとなる。AからBへのリンクをLを用いた論理演算に書き換える。すなわち、リンクノードLの正体は、リンクに挿入されるXNOR(= NOT XOR)演算に対する、新規の入力ノードである。

```
B=!(A xor L)
```

活性値の衝突からのフィードバックが発生した結果、リンクノードへと逆に伝播される。

リンクの伝搬確率の確定から、リンクノードの生成へ

一般的なリンク伝搬の確率は以下の式になる。 P^{11}, P^{00} の２つの伝搬確率を用いる。

$$P_B = P^{11}P_A + (1 - P^{00})(1 - P_A)$$

２リンクにおけるXNORの式。Pxをリンクノードの確率とする。

$$P_B = P_x P_A + (1 - P_x)(1 - P_A)$$

$$B =!(A xor X)$$

つまり、$P^{11} = P^{00} = P_x$ ならば、リンクノードXを用いたXNORで表現できる。リンクの伝搬確率が確定して、P11とP00が等しくなった場合は、XNORを挿入する。すなわち、リンクノードが生成される。

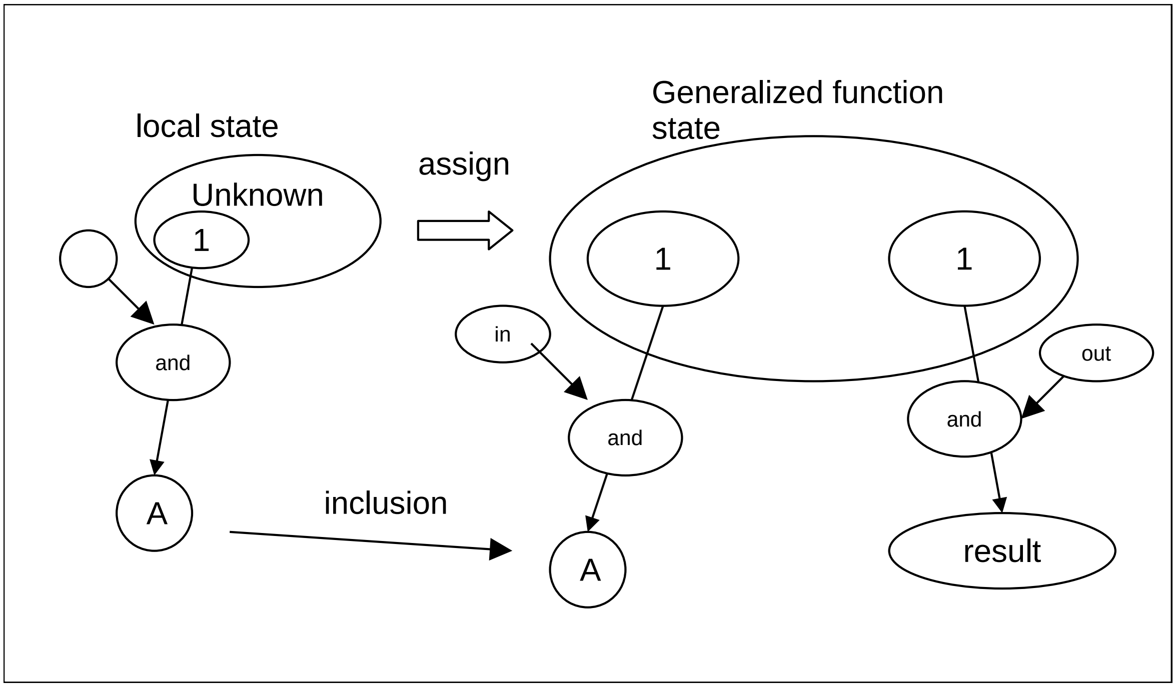
リンクノードへの伝搬は、XNORを通過した逆伝搬。やはり、 $P_x = P^{11} = P^{00}$ ならばリンクノードへ向けて伝搬できる。XORの逆方向伝搬から以下の伝搬確率の式が得られる。

$$P_x = \frac{P_A + P_B - 1}{2P_A - 1}$$

この式は、P_A,P_Bの確率が同じならば１に、相補的ならば０になる($P_A \neq 0.5$であることが条件）。つまりは条件式の結果に応じてリンクノードが活性化される。さらに、このリンクノードに対する仮定ベクトルの変動から、リンクノードとその結果との変動連想を形成することも可能である。

7.3 「代入」　伝搬集合的に包含関係にあるノード間の結合

代入とは、複数のノードに到達したそれぞれの活性値に対して、それぞれの仮定ベクトルが包含関係であることが確認できたら、ノードの間で活性値全体の伝搬を行うことである。



Assign state into function

図では複数のStateの間で代入が行われている。この代入の成立条件を説明する。

localstateノードは値に対する部分状態を包含しているが、それ以外の部分が不確定である。ここで、localstateノードの値Aはfunctionstateノードにも同一条件で存在するため、functionstateノードはlocalstateノードを包含すると見なしても矛盾がない。この結果、functionstateノードはResultノードを部分状態として包含する。

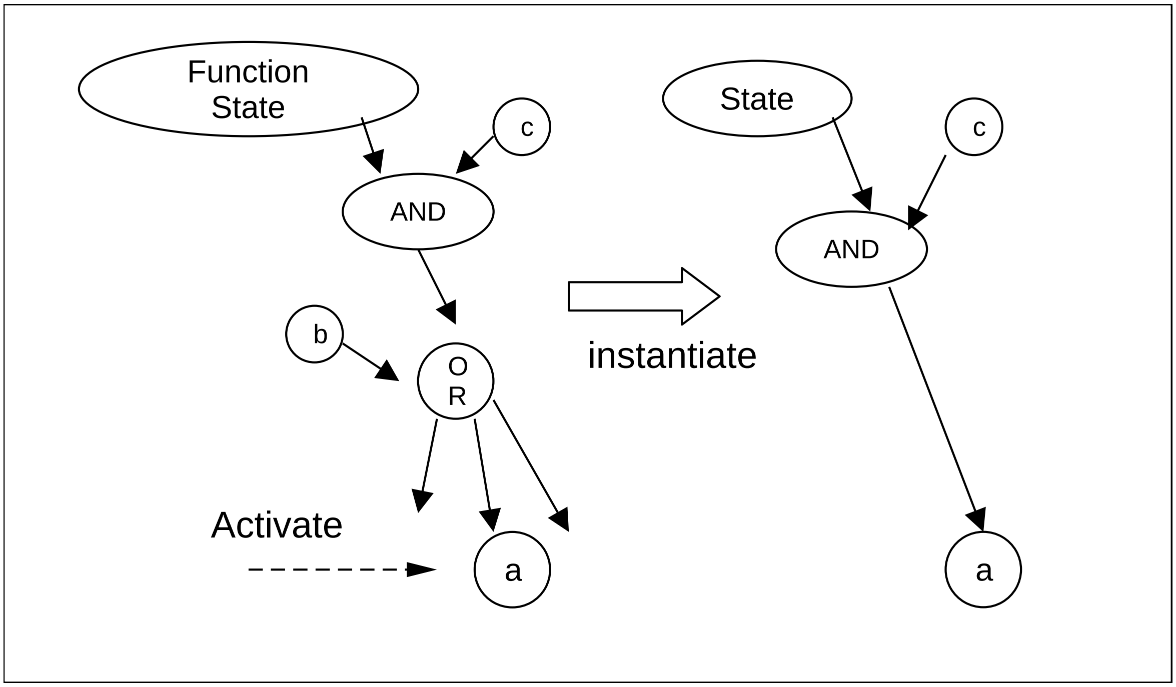
function stateノードにもう１つ部分状態として包含するResultノードは、localstateノードには無いものであるが、localstateノードの未定義部分にも存在すると推定しても矛盾がないと考えられる。このResultは、次の「特化」でlocalstateノードに対して追加が可能になる。

7.4「特化」伝搬部分集合によるネットワークの一部のインスタンス複製

ネットワークの一部を縦に分割して部分集合をとり、簡略化されたネットワークをインスタンスとして生成する。特化で生成されたリンクは、既存のリンクと構造的には変わらないが、多数のリンクの出力の選択が省略できる。

この特化を行うのは、起点の部分集合と終点の部分集合の間で、同じ伝搬集合に対する伝搬確率が定まったときである。起点と終点の間の経路全体を伝搬集合で分離する。

入力されたStateに対して新たな部分集合を関数の結果としてStateに加えるような目的に用いることができる。

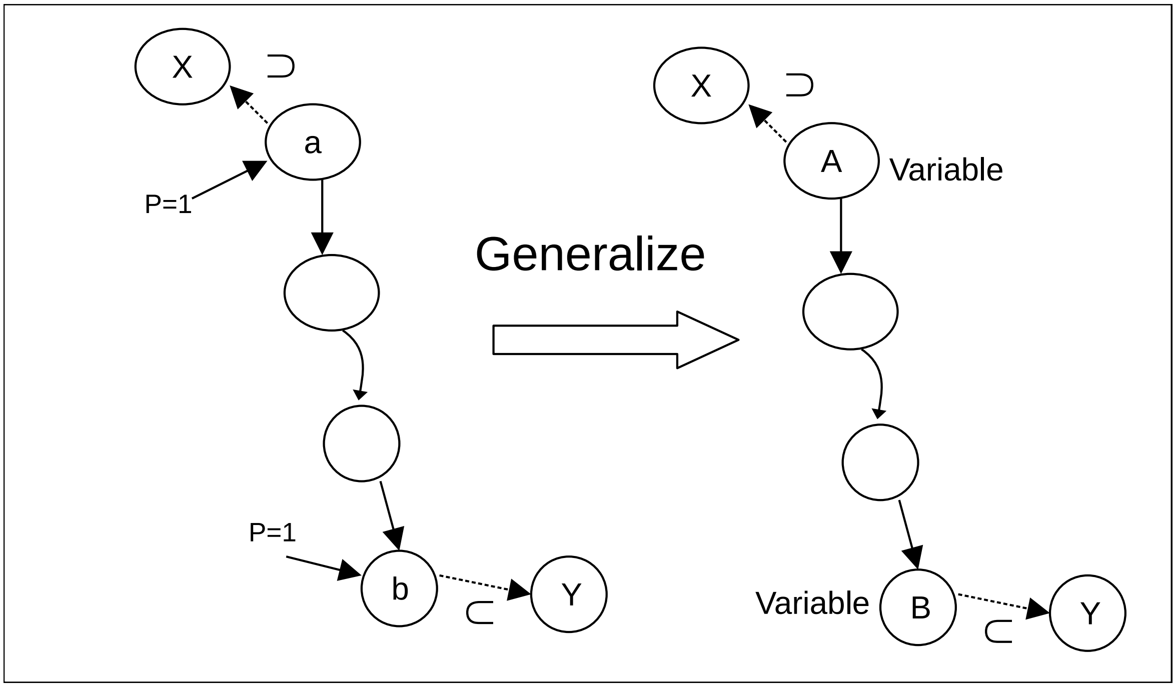


Network instantiate

7.5「汎化」 伝搬部分集合によるネットワークの一部の一般化複製

成立したネットワークの一部に対して、起点の包含集合と終点の包含集合同士で、より集合的に一般化されたネットワークとして複製する。これが汎化である。特化と類似しているが、ネットワークの伝搬集合が完全に観測されていない時点でも実行される。つまり、これは暫定的なネットワークであり、後にフィードバックによって修正される。

この汎化を行うのは、起点の部分集合と終点の部分集合の間で、伝搬が確定したときである。このとき、起点と終点の間の経路全体を、集合としてある程度拡大して分離する。特にXの部分集合であるaは、aよりも集合が大きい変数Aに置き換える。bも同様に変数Bに置き換える。



Network instantiate

7.6「選択」 多数のリンク伝搬の選択制御

あるノードに対して、リンクは大量に接続されうる。とくに、関数間の接続などでは、その数は万単位以上になりうる。そのため、必要なリンク選択を促すノードとして、リンク選択ノードを生成して、そのノードへの入出力ノードを規定する。

リンク選択ノード

リンクの選択は、「リンク選択ノード」を部分活性値にする。リンクノードは、活性値として連想対象になる。リンクノードへの連想リンクは、連想リンクへのフィードバックによって条件が更に追加されていく。

7.7「収斂」リンクの最適化

２つの経路と確率とが一致するポジティブフィードバックが発生すると、伝搬集合的に小さくかつ観測数が大きくない経路のリンクは削除される。SOLのリンクの最適化に用いられる。「特化」の逆の作用であるが、利用されないネットワークに対して行われる。

```
(A&(B|C)&D)|(A&(B|C)) = (A&(B|C))
```

7.8 「統合」同一論理の論理式ノードの統合

直列に結び付けられたリンク前後のノードがAND同士などの同一種類のノードであった場合、かつノード間のリンクの確率が１である場合、ノードを結合する。異なるAND要素があれば、異なる要素だけを抽出して分離する。論理ネットワークの最適化に用いられる。

```
A&(B&C)=A&B&C
A|(B|C)=A|B|C
```

単一ノードからリンクで直接接続された複数の同種のノード同士は、他の入力がほぼ同一ならば統合を試みる。

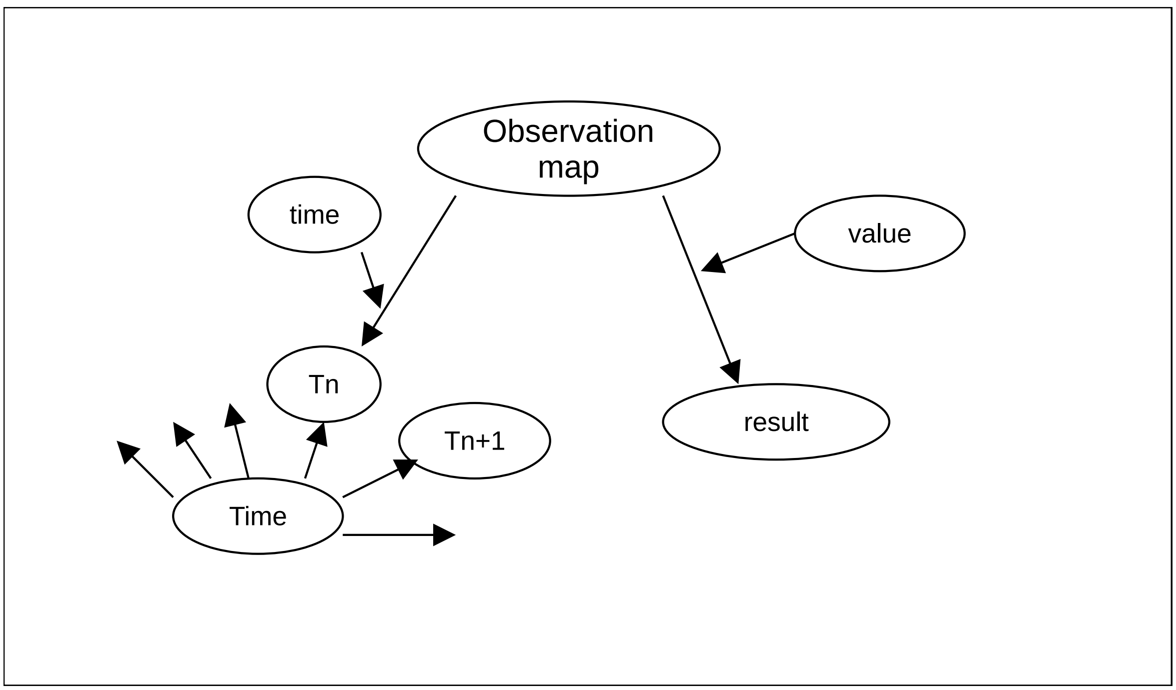
```
(A&B&C),(A&B&C&D)=(A&B&C)&D
```

8 順序回路生成

写像によるステートの概念を階層的に用いることにより、一般化されたステート間の回路を自律生成する。さらにステートを変数などで一般化することにより、汎用的に利用できる関数を生成する。

8.1 観測関数

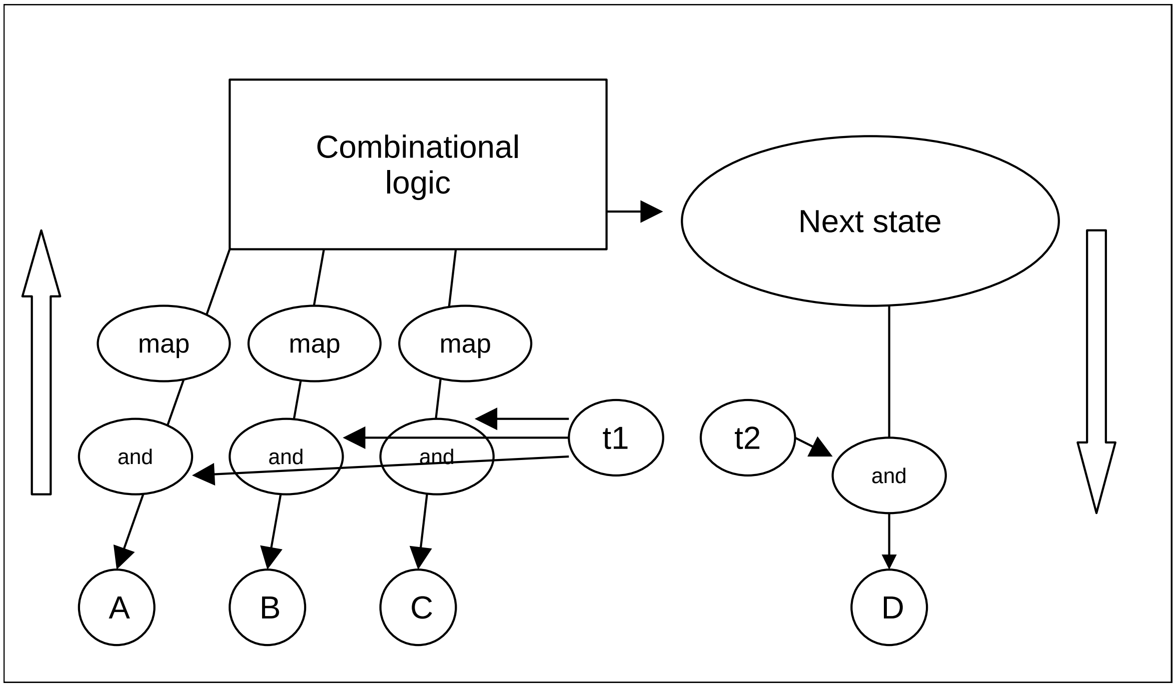
状態観測関数は、観測した現在時間T_nと、同時に観測された結果resultとの連想を形成する。現在の時間に同時に観測された結果同士がさらに因果関係として写像で連結され、それらの間の論理的な関連が観測される。



Observation

8.2 状態遷移生成

時系列の状態遷移は、時間を差分としたときの写像から形成される。A,B,CとDとの写像がフィードバックされて、互いに成立条件となり組み合わせ回路が形成される。

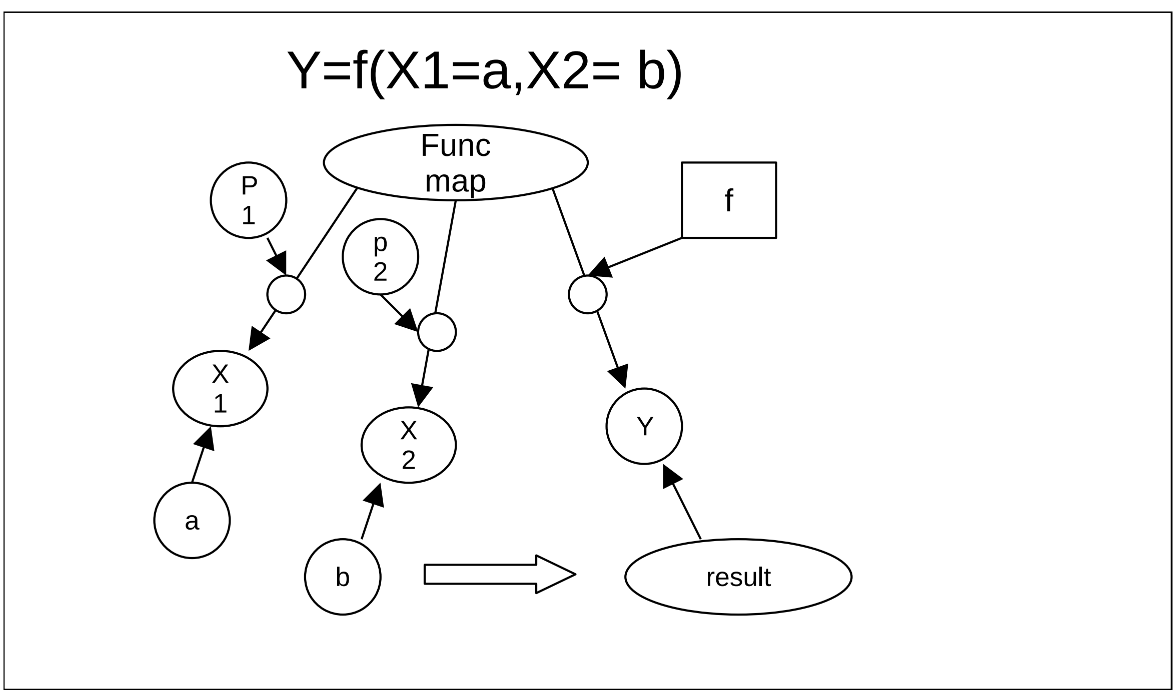


Sequencial circuit by map

入力された連続の状態は、逆方向に活性化されれば連続の出力となる。

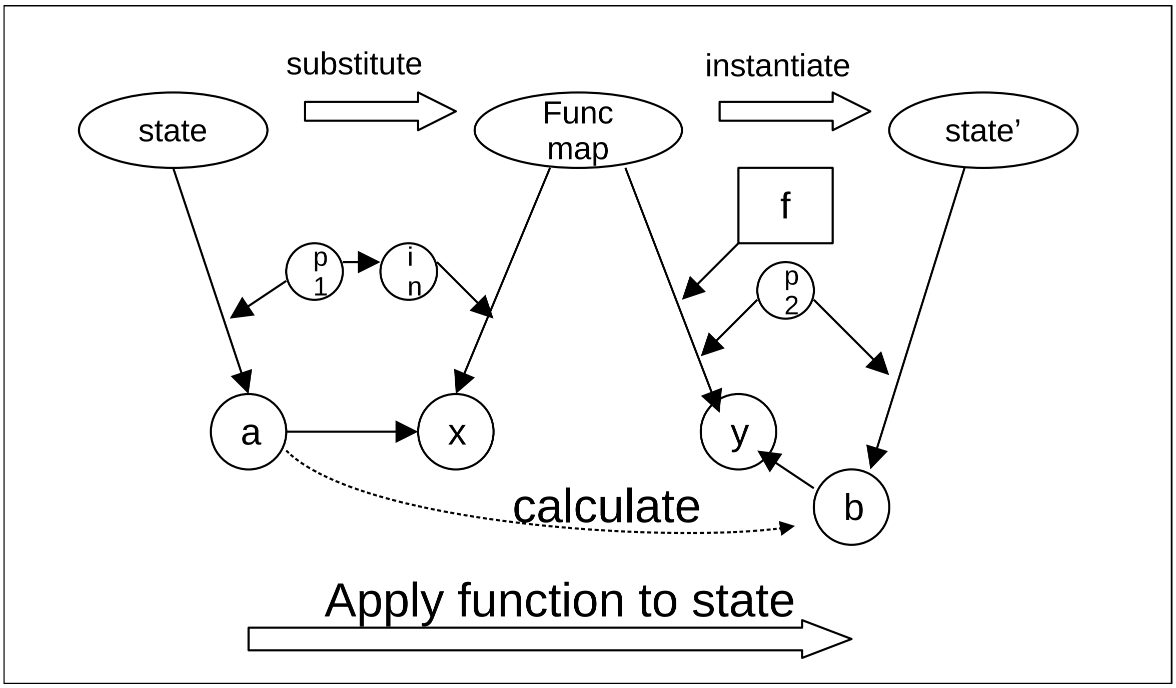
8.3 一般化ノードから関数生成

a,b,resultなどの観測されたノードの間で形成された写像は、リンクへのP00へのフィードバックにより拡大して抽象的なノードに変化する。たとえは値aが「汎化」されて変数int Xとの連想が形成される。こうして一般化されたノード同士の連想こそが関数と予想される。途中の条件のp1,p2は元のネットワークよりも拡大されるがその大きさは未確定であることに注意する。関数は個別のa,b,result同士の関連そのものを写像で示すのではなく、別に作成する値ごとの個別の写像の動作条件となる。



8.4 関数適用

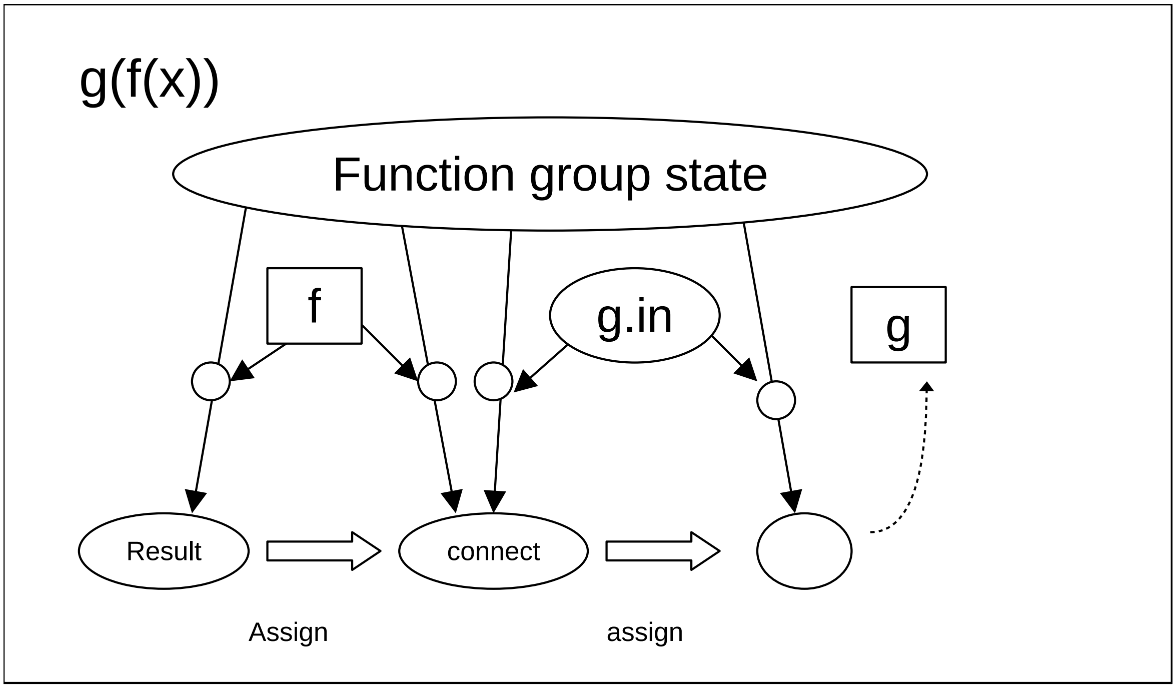
一般化で形成された関数は、入力値を持つステートを代入する。関数は代入されたステートよりも伝搬集合的に大きいとみなされるため、関数の結果の値も入力ステートに含まれるとみなす。これがステートへの関数の適用である。そして適用された結果は、元のステートにインスタンスとして追加される。



Apply function

8.5 関数のグループ化

生成された関数を連続適用する方法を示す。関数fと関数gとを連続適用するために、ｆの出力とｇの入力を接続するネットワークを追加する。

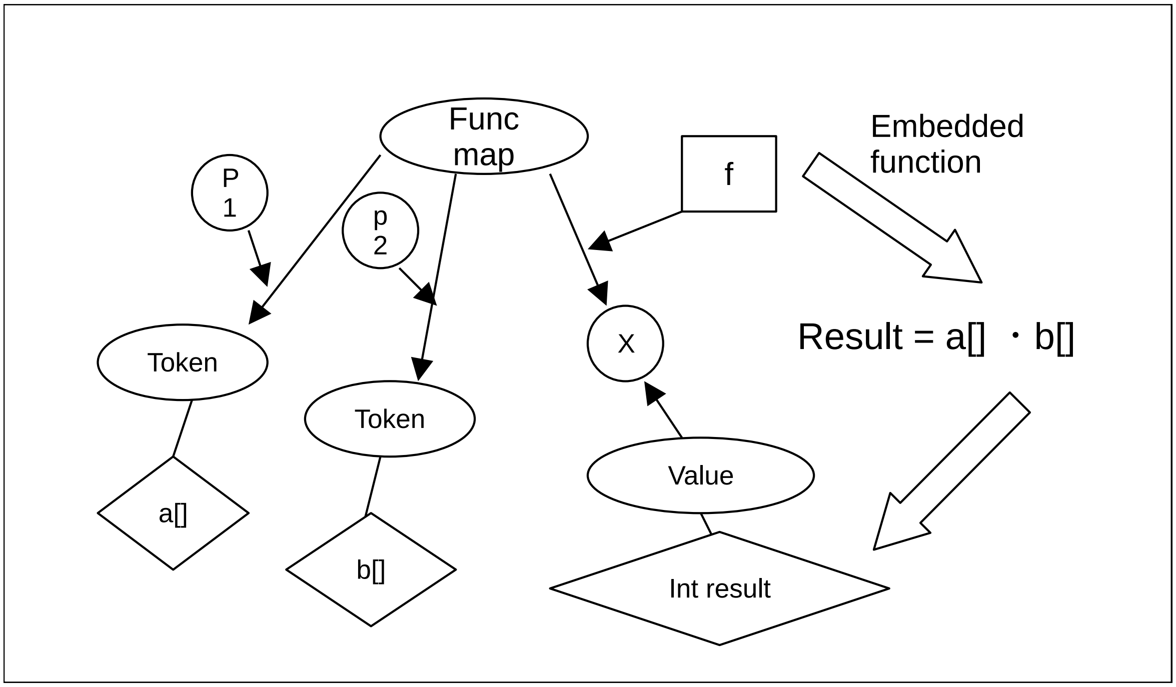


Function group

関数の結果は一致を示すリンクノードも用いることができる。すなわち一致判断をBooleanに変換する条件判断も実装が可能になる。さらに、関数を条件次第で起動するように、関数を別の関数の出力とする。これらの関数をステートグループで連想することにより、一連のソフトウェアを形成することができると考えられる。

8.6 組み込み関数とその利用

組み込み関数とは、入力ノードに対して内部動作を記述せずに出力ノードだけが活性化される外部関数と定義する。SOLの外部との入出力にも利用できる。組み込み関数で用いられるノードは、Bool値であるが、組み込み関数はノードに対応する実際のオブジェクトを取得することができる。具体的にはIntegerなどのスカラ値、ベクトル値、オブジェクトなどを対応させて、組み込み関数が演算に利用する。結果は結果の値に相当するノードをその都度生成して、組み込み関数が対応する実体を結合させる。以下の例では文字Tokenの持つ実数ベクトル値を用いて、内積を組み込み関数で計算してResultの値を生成し、相当するノードを生成している。[2]



Embedded function

9 結論

以上の内容を統合して要約すると、この自己組織化論理は以下の特徴を持つ。

1. 双方向論理は、集合間の論理演算に逆伝搬を適用することで写像の概念を実現する。写像を用いれば、時空間の重複がない任意の集合同士の関連を表現できる。これにより、順序回路が論理回路だけで記述できる。
2. 論理ノード間の接続リンクにP^{11}、P^{00}の二項組の伝搬確率を設定し、リンクの双方向伝搬を可能にすることで、双方向二項確率ベイジアンネットワークとなる。逆伝搬はベイズの定理そのものとなる。写像を通過した双方向伝搬は、任意の２つの集合間の厳密な伝搬確率計算を可能にする。
3. 仮定ベクトルによる集合確率伝搬は、リンクや写像を通過する伝搬部分集合を仮定ベクトルで厳密に管理し、仮定ベクトルの部分一致や差分を用いることで、関連のある任意の２つのノード間の集合包含関係を算出できる。複数の確率伝搬が同一ノードで衝突した際に、双方の仮定ベクトルが一致した場合、確率の一致、不一致に応じて双方のパスにフィードバックを行う。特に不確定リンクを確定値へとフィードバックする場合は一致を示すリンクノードを活性化させる。
4. 双方向二項確率ベイジアンネットワークは、リンクの過去の伝搬確率と観測回数とに基づき、確率変動分配階層フィードバックアルゴリズムを使用することで、自動的にすべてのリンク伝搬確率を正確にフィードバック補正する。
5. 連想は、同時に変動が観測された集合的に排他的なノード同士を確率的に結合して写像にする。連想の対象は観測された値だけではなく、フィードバックで発生した論理演算の条件ノードやリンクノードも同様に連想対象となる。この作用により、どのような部分集合の間でも連想を形成することができる。
6. 確率的自律論理生成アルゴリズムは、連想で結合されたリンク確率へのフィードバック結果に応じて、論理演算と写像とリンクノードとを追加することによって、確率100%あるいは0%の確定的デジタル論理を自動的に追加生成する。P^{11}、P^{00}のペアの伝搬確率を用いることで、論理演算の正確な挿入が可能になる。P^{00}の変動に対するORノードの挿入は、定数を変数へと一般化する作用でもある。
7. 時間に対する値の変動に対して連想を形成し、連想へのフィードバックに対してさらに論理演算を追加していくことにより、メモリなどの入力と出力の間で一般化された論理回路を生成することが可能になり、一般的な順序回路が自律形成されていく。一般化された順序回路は何度も利用されて関数となる。
8. 写像を階層的に用いれば、一般的なソフトウェアの階層的なデータ構造と関数とを表現できる。これらの写像の間で伝搬集合の包含関係を管理することにより、データ構造に対するパラメータの一致を判定して、関数の結果をデータ構造に反映することができる。そのためにSOLで定義した写像間の「代入」と「特化」を用いる。

このSOLは確率、集合、写像などの数学的な基本原理のみに基づいて構築されている。生物のニューロンの模倣は行っていない。さらに、ニューラルネットワークの非線形活性化関数のような厳密ではないヒューリスティックスはほとんど使用していない。そのため、一般的な機械学習よりも厳密な数学理論に近いものであり、自律生成ソフトウェアがこのSOLの延長線上にあることはほぼ確信できる。ニューラルネットワークなどの既存の機械学習も、このSOLの一部の近似的な実装であるとすら考えられる。そのため、正確で安全な機械学習を目指すならば何らかの形でこのSOLの原理を導入するしかない。

実装による動作の実証、実装の効率化、簡略化、並列化などが今後の課題となる。また、SOLが論理回路やソフトウェアを網羅的に自律生成ができるかどうかも、今後の研究課題になるだろう。

数理論理学や圏論等との関連も明白なので理論の一般化の余地がある。確率の伝搬の考え方は量子力学の経路積分などとの関連も強く、物理学との何らかな形での統合を行うことで相互の正当性の証明になると考えている。[3]

参考文献

[1] Artificial Intelligence A Modern Approach
Stuart Russel, Peter Norvig

[2] Attention Is All You Need
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin

[3] Pictureing Quantum Processes A First Couse in Quantum Theory and Diagrammatic Reasoning
Bob Coecke, Aleks Kissinger