

<資料>

コネクショニストモデルシミュレータ *tlearn*
を使った心理学実験実習課題
—対称性の学習における中間ユニットの数と学習率の効果—

守 一雄 信州大学教育学部教育科学講座

An Exercise for Experimental Studies in Psychology
Using a Connectionist Model Simulator *tlearn*:
—Effects of Hidden Units and Learning Rate in Learning Symmetry—

MORI Kazuo: Faculty of Education, Shinshu University

The present paper proposes to use the connectionist model simulator software *tlearn* for exercises in laboratory class of introductory psychology courses. This paper includes a student manual for an exercise of making neural networks to learn symmetrical patterns in the inputs using *tlearn* and analyzing the data treating each set of randomly assigned initial connection weights as subject in a usual psychology experiment. The software *tlearn* can be run either on Windows OS or Mac OS personal computers, and is available from the following web site, <http://crl.ucsd.edu/innate/tlearn.html>.

【キーワード】 コネクショニストモデル *tlearn* 心理学実験実習 対称性の検出

行動主義心理学の刺激-反応モデル, 認知心理学のプロダクションシステムモデルに代わる新しいモデルとして, 脳神経系を模したコネクショニストモデルが提案されてきた。コネクショニストモデルは心理学の分野でも適用範囲は広いと期待されているが, 難しい数式が使われること, コンピュータシミュレーションのための技術が必要であることなどから, 日本の心理学研究者への浸透はまだ質量共に貧弱である。守らは, コネクショニストモデルについての入門書(守 1996)や研究例を紹介した概説書(守・都築・楠見 2001)を刊行してきたが, 多くの心理学研究者や心理学を学ぶ学生たちにとってコネクショニストモデルはまだ心理学から遠い存在であるようである。

従来, 科学的心理学はその方法論として実験を中心に据え, これに前段階としての観察と調査を加えた3つの研究方法(観察・調査・実験)を重要視してきた。大学での心理学教育においても, 伝統的に「一般実験」と呼ばれる研究方法実習の授業が必修科目として置かれるのが常であった。コンピュータシミュレーションはこうした3つの伝統的な研究方法に付け加えられるべき第4の研究方法である。心理学以外の他の多くの実証科学においてコン

ピュータシミュレーションが新しい研究方法としてその地位を不動のものにしている。

心理学におけるコンピュータシミュレーションの活用は、認知心理学の隆盛と同時に発展した人工知能学におけるものが始まりであると思われる。しかし、人工知能学におけるコンピュータシミュレーションは理論や仮説を言語や数式で表現する代わりにコンピュータプログラムで記述し、そのプログラムが適切にふるまうことをもって理論や仮説の正しさを証明するという使われ方をしていた。代表的なものとしては J.R.アンダーソンの ACT*モデルや A.ニューウェルの SOAR などがある。しかし、アンダーソンらが主張したように (Anderson & Bower 1973)、認知心理学におけるコンピュータシミュレーションの活用や壮大なモデル作りは、伝統的な実験を中心とした心理学研究法へのアンチテーゼとしてなされたものであった。その結果、こうしたコンピュータプログラミングのためのリスト処理言語 (Lisp や Prolog など) を学ぶことが、従来の心理学の研究方法に付け加えられて広く普及するということにはならなかった。

これに対し、コネクショニストモデルにおけるコンピュータシミュレーションの活用は、伝統的な心理学の研究法との親和性が高い。にもかかわらず、前世代のコンピュータシミュレーションの影響から、新しいコンピュータシミュレーションも心理学の研究法として受け入れられてこなかった。これには、心理学研究者や心理学の学習者が簡便に使うことのできるソフトウェアがなかったことも大きく影響している。

そうしたなかで、Plunkett らが開発した tlearn (Plunkett & Elman 1997) は Windows や Mac のパソコンで実行でき、また、GNU の考え方に基づいた無料配布ソフトウェアであるため、誰でも使うことができる。ここでは、この tlearn を用いて、従来の心理学研究法における課題の一つとして組み込むことができるような実習課題を提案する。

1. 課題の目的

提示刺激の対称性を検出できるようなニューラルネットワークを構築し、学習率と中間ユニットの数が学習に与える影響を調べる。

2. 方法

<装置> Windows パソコンまたは Mac OS パソコン・tlearn ソフトウェア (<http://crl.ucsd.edu/innate/tlearn.html> からダウンロードできる。)

<学習課題> 6つの入力ユニットからなるニューラルネットワークに、入力される2値ベクトルの対称性を検知して1を出力するよう学習させる。

<実験計画> 中間ユニットが2つと3つの2種類のネットワークを作る。それぞれのネットワークに異なる学習率 (0.1, 0.4, 0.7 の3水準) で学習させ、その学習成績を比較する。

2種類のネットワークそれぞれについて、各ユニット間の結合強度の初期値をランダムに20種類用意する。これは各セルに20人ずつ被験者を配置した2×3の2要因実験計画に相当する。第1の要因は被験者間要因で2水準 (中間ユニット2/中間ユニット3)、第2

の要因は被験者内要因で3水準（学習率 0.1/0.4/0.7）である。
<実験手続き>種々の学習遂行尺度の中から、最適なものを選び、条件間の比較を行う。

3. 具体的手順

<シミュレーションの準備>

(1) tlearn を起動する。tlearn のアイコンをダブルクリックする。

(2) 画面のメニューバー（一番上の部分）が

「File Edit Search Network Displays Special Window」 に変わる。

3.1 メニューバーの Network をクリックし、New Project を選ぶ。

(1) 新しいプロジェクトに symmetry と名前を付け、OK ボタンをクリックする。

(2) 自動的に、symmetry.cf, symmetry.data, symmetry.teach が用意される。

(3) コンフィグファイル(symmetry.cf)に以下のように書き入れ、3層のネットワークを構築する。（カッコ内は説明である。ファイルには英字のみを書き込むこと。「=」の前後の空白など空欄もそのとおりに入力すること。）

```
NODES:                                (ユニットについての指定部分。すべて大文字で記入すること.)
nodes = 3                               (入力ユニットを除くユニットの数が3であることを指定する.)
inputs = 6                               (入力ユニットが6つであることを指定する.)
outputs = 1                             (出力ユニットが1つであることを指定する.)
output node is 3                         (出力ユニットに付けられる番号を3と指定する.)
CONNECTIONS:                            (ユニット間の結合についての指定部分。すべて大文字で記入.)
groups = 0                               (同じ結合強度に設定するグループの数を指定する。
                                         ここではそうしたグループを作らないため0とする。)
1-2 from i1-i6                          (2つの中間ユニット1と2に6つの入力ユニットから結合がなされる。
                                         3つのユニットのうち、番号3を出力ユニットとしたので、
                                         残りの1と2は自動的に中間ユニットを表す番号となる。6つの
                                         入力ユニットの番号にはそれぞれ i がつく。「i1-i6」と書くこと
                                         で、「i1 から i6 まですべて」を表す。)
3 from 1-2                               (2つの中間ユニットから出力ユニットへ結合がなされる。)
1-2 from 0                               (中間ユニットと出力ユニットのすべてにバイアスユニットから結合がなされる。
                                         バイアスユニットは常に番号0で表される。)
SPECIAL:                                (ユニットとそれらの間の結合以外の指定事項を書き込む部分。)
selected = 1                             (中間ユニットの活性レベルを後で分析できるよう選び出すための指定。)
weight_limit = 1                         (結合強度の初期値をランダムに設定する幅を指定する。この場合0を中心
                                         に1の幅、つまり初期値は±0.5となる。)
```

(4) コンフィグファイル(symmetry.cf)が正しく記入できたら保存(save)しておく。

(5) メニューバーの Display から Network Architecture を選ぶと、図 1 のようなネットワークの構造図を描いてくれる。(cf ファイルに間違いがある場合は、間違っている部分を示して cf ファイルが開く。どこが間違っているか確認し修正すること。)

(6) 上の(3)では、中間ユニットが 2 つのネットワークが作られた。中間ユニットが 3 つのネットワークにするためには、コンフィグファイルのどこを変更すればいいだろうか？

(① 2 行目のノード数, ② 5 行目のノード番号, ③ 8 行目の入力ユニットからの結合, ④ 9 行目の中間ユニットから出力ユニットへの結合, ⑤ 10 行目のバイアスユニットからの結合, ⑥ 12 行目の selected の指定. **網掛**の数字がそれぞれ一つ大きいものになる。)

3.2 データファイル (data file : symmetry.data) と正解ファイル (teach file:symmetry.teach)を作成する。

(1) 最初の 2 行はどちらも同じである。ここでは data file を例示する。

```
distributed          (1 行目には「distributed」か「localist」のどちらかを書く。)  
64                  (2 行目にはデータの数を書く。)  
0 0 0 0 0 0        (3 行目以降は、データを 1 行ずつ記入する。  
0 0 0 0 0 1        データ間のスペースも意味を持つので必ずスペースを入れること。)  
0 0 0 0 1 0  
0 0 0 0 1 1  
(中略)  
1 1 1 1 1 0  
1 1 1 1 1 1
```

(2) 正解ファイルには、3 行目以降にデータファイルの入力データに対応する正しい出力を 0 か 1 かで指定する。

(3) それぞれのファイルを保存する。

(4) 中間ユニットが 3 つの場合にも、データファイルと正解ファイルはまったく同じものが使われる。

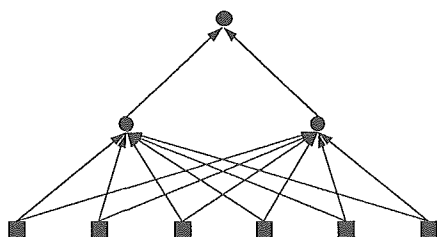


図 1 tlearn が描く Network Architecture (■は入力ユニットを●はその他のユニットをあらわす.バイアスユニットは表示していない。)

<シミュレーションの実行>

3.3 学習条件の設定

- (1)メニューバーの Network から Training Options を選ぶ.
- (2)Training Sweeps のボックスに 192000 と打ち込む. (64 とおりのデータを一揃い提示して1エポックであるから, 3000 エポックは 64×3000 で 192000sweeps となる.)
- (3)その下の Seed は, 下の Seed randomly の方を選ぶ. (Seed とは「種を植えること」を意味する英語であるが, ここでは「仮想的な乱数表のスタート地点を決めること」と考えるとわかりやすいだろう. 上のボックスに数値 (-32766 から 32767 の範囲の整数) を入力すれば, それがスタート地点となる. 下の Seed randomly を選ぶと, tlearn が自動的にランダムに乱数表のスタート地点を選んでくれる.)
- (4)その下の, Train sequentially/randomly は, データファイル内のデータを上から順に提示する(sequentially)かランダムに提示する(randomly)かの選択肢である. ここではランダムに提示を選ぶことにする.
- (5)右側の Learning Rate:には, 実験条件に応じて, 0.1, 0.4, 0.7 のどれかを入力する.
- (6)その下の Momentum:には, 0.5 を入力することにする.

3.4 学習の実行

- (1)メニューバーの Display から Error Display を選ぶ. (これは学習の実行に必須ではないが, 学習の進行状況が RMS エラーの減少の様子からわかる. 同様に, Connection Weights を表示させても結合強度の変化の様子が変わっておもしろい.)
- (2)メニューバーの Network から Train the network を選ぶ. (これだけで学習が自動的になされる.)

3.5 学習指標の決定

「解説」の「5.0 学習指標のいろいろ」を参考にして, 今回のシミュレーション実験でどんな学習指標を用いるのがよいかを定める. そのためには, 学習がどのようになされるかを何度か仮に行ってみる必要がある.

4. 結果の分析

- 4.1 各条件ごとの学習指標を記録する.
- 4.2 各条件ごとの学習指標を適切な統計的手法で分析する.
- 4.3 中間ユニットの数の効果と学習率の効果について考察する.

5. 解説

5.0 学習指標のいろいろ

(1)代表的指標

心理学における学習実験では種々の指標が用いられる. 以下に心理学の研究でよく用いられる代表的な指標を示す.

- ① 正答数（またはエラー数）：初めはほとんど正答が得られないが、学習が進むに連れて徐々に正答が増えていく。学校での試験は通常この考えに従って行われる。tlearn では、出力ユニットの出力（学習者の解答に相当する）と正解との差（＝エラー、誤差）を、すべての問題について平均した値である RMS エラーを自動的に計算してくれる。RMS エラーは Root Mean Square Error のことで、誤差(Error)を自乗(Square)してから平均し(Mean)、それを平方根(Root)に直したものである。
- ② 学習成立までの試行数（時間）：どれだけ早く学習できるようになるかを学習の指標とすることも多い。「一定の時間内でどれだけできるか」を調べる代わりに、「一定の課題がどれだけ時間でできるか」を調べるわけである。
- ③ 学習が成立した回数（被験者数）：何時間かかっても学習が成立しない場合、①や②の指標を使うことはできない。そういう場合には、各条件ごとの学習成立数（通常の心理学実験ならば被験者数）を学習の容易さの指標とすることができる。統計的な検定には χ^2 検定などのノンパラメトリックな検定法が用いられる。

(2)学習基準

「学習成立までの試行数」や「学習が成立した回数」を学習の指標として用いるためには、何をもって「学習成立」とするか「学習基準」を決める必要がある。では、ネットワークが対称性を正しく検出できるようになったと判断できるための基準はどう決められるだろうか。以下に3種類の基準を紹介する。

①正答率（誤答率）に基づく基準

学習基準の例として RMS エラーが一定水準以下になることを考えることができる。全部で 64 の入力データに対し、出力ユニットは 0 から +1 までの連続量を出力する。正答は 0（＝非対称）か 1（＝対称）であるから、すべての入力に 0.5 という中間値が出力されたとすれば、RMS エラーは 0.5 となる。そこで、学習開始直後の RMS エラーはほぼこの値となる。出力値を四捨五入して、 $x \geq 0.5$ のとき 1、 $x < 0.5$ のとき 0 とすれば、RMS エラーが 0.5 付近でも学習が成立している可能性もある。一方、不正解であるにもかかわらず、RMS エラーが小さくなる場合も考えておく必要がある。たとえば、この課題では 64 の入力パターンのうち「対称パターン」は 8 通りしかない。そこで、ネットワークがすべての入力パターンに対して 0 を出力するよう学習したとすると、この 8 パターンにはエラーが最大値の 1 となるが、残りの 56 パターンではエラーは 0 となり、エラーの平均である RMS エラーは 0.125 とかなり小さくなる。そこで、学習基準はこの 0.125 よりも小さいものにすべきである。少し余裕をもたせて、対称な入力に対しては 0.9 以上（エラー 0.1 未満）、非対称な入力に対しては 0.1 未満（エラーも 0.1 未満）の出力をするようになれば学習成立と見なして間違いないだろう。そこで、RMS エラーが 0.1 を下回った時点で学習成立とするような学習基準が考えられる。

学習基準を定め、基準に達するまでの試行数を指標とする実験を行う場合には、Training Options のウィンドウの左下の more ボタンをクリックして、学習基準に到達したところ

で学習を終了させるオプションを選ぶとよい。具体的には、右の段中央やや下に、Halt if RMS error falls below という指定があるので、このボックスに学習基準とする RMS エラーの数値、たとえば 0.1 を入力する。メニューバーの Display から tlearn status を選んで学習状況をモニターしておけば、RMS エラーが設定値を下回り学習が停止したときの sweep 数が表示されるので、それを記録する。

②個々の解答の正誤を調べる

ネットワークが 64 個の入力パターンについて、それぞれどんな出力をしているかを調べ、すべての入力パターンに「適切な出力」をしていれば、学習成立とみなすことができる。ネットワークの出力は連続量であるために、「0」か「1」かではなく、0.1 や 0.8 といった出力になってしまう。この出力を四捨五入したとき「正答」になるものを「適切な出力」と考える。

メニューバーの Display から Node Activations を選ぶと、各入力パターンに対する出力ユニットの活性化が白い四角形の大きさで表される。64 の入力パターンのうち、8 つの対称な入力にのみ出力ユニットが活性化するようになれば、学習が成立していることが確認できる。

③学習過程からの判断

正答率（誤答率）を基準とする場合の問題点として、偶然に基準に達してしまう可能性（まぐれ当たり）があることである。これに対し、学習過程を追っていくと、学習が着実に進行しているのか、まぐれ当たりを繰り返しているのかの判断ができる。tlearn では、RMS エラーの増減で学習の進行がわかるようになっているので、こうした判断をすることが可能である。学習が成立する場合には、RMS エラー曲線が振動しながらも着実に減っていく。一方、学習が成立しない場合には、RMS エラー曲線はいつまでも振動を繰り返し収束しない。

5.1 出力関数

tlearn では特別な設定をしないかぎり、各ユニット i における出力 a_i は、そのユニットへの入力の総和(net_i)を式 1 のようなロジスティック関数で変換したものとなる。この変換により、すべてのユニットの出力は 0 から 1 の間に収まることになる。この変換に用いる関数を「出力関数」という。また、式 2 に示すように、各ユニットへの入力の総和(net_i)は、そのユニットに結合するユニット j からの出力 a_j とそれらユニット間の結合強度 w_{ij} の積をすべて足したものである。

$$a_i = 1 / (1 + e^{-net_i}) \quad (式 1)$$

$$net_i = w_{ij} a_j \quad (式 2)$$

上記のロジスティック関数はコネクショニストモデルに最もよく使われる S 字型の（＝シグモイド）関数であり、これにより出力値が入力値に比例しない非線形の変換が施されることになる。

5.2 閾値とバイアスユニット

各ユニットはそのユニットへの入力の総和が一定の値を越えだすと興奮しだし、出力するようになる。こうした「一定の値」を「閾値(threshold)」と呼ぶ。各ユニットごとに閾値は異なるため、興奮しやすいユニットやなかなか興奮しないユニットができることになる。tlearn では、各ユニットの閾値を「常に 1 を出力しているバイアスユニットからの結合強度」とみなす方法がとられている。tlearn ではバイアスユニットはユニット番号 0 で表されている。たとえば、「入力の総和が 0.8 を越えるあたりから興奮し始めるユニット(=閾値が 0.8 であるユニット)」は、tlearn では、「入力の総和が 0 を越えるあたりから興奮し始めるユニット(=閾値が 0 であるユニット)に常に-0.8 の抑制的な入力があること」として表されるわけである。こうすることで、tlearn ではすべてのユニットが基本的には同じ閾値 0 のユニットとなる。

また、こうすることで、通常のオン・オフを逆転させたようなユニットを作ることでもできる。バイアスユニットから興奮性の(正の)結合があるとすれば、バイアスユニットは常に 1 を出力しているため、このユニットには常に一定の正の入力があることになる。通常のユニットは基本的には興奮していない状態にあり、一定以上の入力があったときに興奮を始めることになるが、この逆転ユニットは基本的に興奮している状態にあり、一定以上の負の入力があった場合だけ興奮を止めることになる。

5.3 学習率 (Learning rate) とモーメント (Momentum)

tlearn では一般化デルタルールと呼ばれる方法で各ユニット間の結合強度が修正される。デルタルールとは、出力ユニットの出力(o)と教師信号(t)との差(=デルタ)に基づいて結合を修正する学習規則である。修正すべき量 Δw_{ij} は tlearn では式 3 のように計算される。

$$\Delta w_{ij} = \eta \delta_{ip} o_j \quad (\text{式 3})$$

これは入力パターン p に対するあるユニット i の出力 o_{ip} と正しい出力 t_{ip} の差 δ_{ip} とそのユニットに結合しているユニット j の出力 o_j との積によって結合強度が修正されることを意味している。(厳密には、 δ の計算式はもう少し複雑である。) 各ユニットを擬人的に表現するなら、「自分の間違いの程度 (δ) と情報源 j の興奮度 (o_j) とを考慮して、次回その情報源からの情報をどれだけ重視するか(w_{ij})を修正する」ということである。このとき、修正を目一杯行うかどうかを決めているのが、係数 η (イータ) であり、これを「学習率 (learning rate)」と呼ぶ。学習率は通常 0 から 1 までの値をとる。学習率は大きいに越したことはないような気がするが、実際には必ずしも大きい方がいいわけではない。検出すべきパターンは複数あるため、個々のパターンごとに最適な修正を施してしまうと、全体としては「修正しすぎ」となってしまふからである。

結合強度の修正にはもう一つのパラメータが使われる。それは「モーメント μ (momentum)」と呼ばれるものである。モーメントは、結合強度の修正に前回の修正をどれだけ加味するかを決める係数である。簡易式で示せば式 4 のようになる。

$$\text{今回の修正} = \text{結合強度の修正} + \mu (\text{前回の修正}) \quad (\text{式 4})$$

momentum という英語は「勢い・はずみ」などと訳されるが、前回の修正の勢いが次ぎ

にある程度持ち越されると考えると直感的にわかりやすいだろう。デルタルールによる学習は、谷底(=到達目標)に向かって斜面を転がり落ちていくボールにたとえられる。斜面に部分的なコブがあっても、ボールの動きに勢いがあれば、このコブを乗り越えていくことができる。学習率の説明に用いた擬人化をここでも用いるとすると、「特定の情報源からの情報をどれだけ重視するか(w_{ij})を修正する際に、前回の修正をも加味する」ことにたとえられる。日常の人間関係においても、最近の一回だけでなく、過去の経験が影響するのと同じことである。

5.4 ネットワークは対称性をどう検出しているのだろうか？

対称性の検出は人間の目には一目瞭然だが、ネットワークに検出させようとする一筋縄ではいかない。課題と同じネットワークを紙に描いて、それぞれのユニット間の結合強度をどのように設定すれば、正しく対称性が検出できるようになるか考えてみよう。(人間の頭で考えても、この問題の解決はかなり難しい。)

次に、正しく検出できるようになったネットワークの結合強度がどのようになっているか調べてみよう。対称性を正しく検出するための結合パターンは一通りではない。学習が成立したネットワークのいくつかについて、それぞれ結合パターンを調べ、比べてみよう。

結合強度がどのようになっているかを見るために2つの方法が用意されている。それは、①視覚的に見るか②結合強度の詳細な数値を調べるかである。まず初めに①結合強度がどのようになっているかを視覚的に見てみよう。このためには、メニューバーの Display から Connection Weights を選んで、Connection Weights Diagram というウィンドウを開く。このウィンドウには横軸に結合の開始ユニット、縦軸に結合の終着ユニットが配置され、特定のユニットから特定のユニットへの結合強度がマトリクスとして表わされる。結合強度は長方形の面積として示され、正負は白黒で示される。そこで、一番左上に大きな黒い長方形があるとすれば、バイアスユニット(b)からユニット1(1)へ大きなマイナスの結合がなされていることを意味している。結合強度の最大値は網掛けで示されている。網掛けのない白いままの部分とはもともとそうした結合がないことを表している。このウィンドウを表示させたまま学習を進行させることもできる。そうすることで、学習の進行に伴って結合強度がどう変化していくのかを観察することもできる。

結合強度の大きさを正確に数値で知るためには、結合強度ファイルを開く必要がある。結合強度ファイルは symmetry.192000.wts(Windows 版では symmetry-192000.wts)という名前でも自動的に作られているはずである。このファイルをメニューバーの File から Open を選んで開くと

```
NETWORK CONFIGURED BY TLEARN
```

```
# weights after 192000 sweeps
```

と書かれたファイルが開く。このファイルには、①で視覚的に示された結合強度が結合の終着ユニット順に左から順に数値で示されている。①では空欄になっていた部分(結合が存在しない部分)は0.0000となっている。〔結合がない〕=〔結合強度が0〕

対称性を検出するための結合強度パターンを見つけだすことができたでしょうか？対称性の検出ができるようになったネットワークの結合強度パターンをいくつか調べてみればわかるように、「正解」は一つだけというわけではない。しかし、対称性を見つけだす「原理」は同じになっているはずである。

それは、左右の対応するユニット対が「同時に興奮（入力が1）しているかどうか」の検出を、3つの対について行なうことである。具体的には、左右の対応する対で結合強度の絶対値を揃え、正負だけ逆転させることで、同時に興奮している場合には興奮がちょうど打ち消しあうようにする。こうすれば、どちらかのユニットだけが興奮している場合には、正または負の興奮が伝わるが、どちらも興奮していない場合や両方が興奮している場合にはゼロとなるからである。

ただし、ある対における非対称を示す興奮と別の対における非対称を示す興奮とが打ち消しあうことになってはいけない。さらには、ある対の非対称を示す興奮と他の2つの対の非対称を示す興奮とが打ち消しあうのでもまずい。そこで、非対称を示す興奮の絶対値およびその差の絶対値がそれぞれ違うものとなるようにする必要がある。具体例で言えば、0.1, 0.2, 0.4 ならば、どのような組み合わせになってもゼロにはならない。このような3つの数の組み合わせは無数に存在するため、「正解」は一つにならないのである。

出力ユニットが興奮すべきなのは、入力ユニット群から中間ユニットへの入力が全体としてゼロになったときだけである。中間ユニットへの入力がゼロならば、中間ユニットは興奮しないから、逆に考えて、「中間ユニットが興奮しない場合だけ出力ユニットが興奮する」ようにすればよい。これは、バイアスユニットから常にプラスの興奮を伝えることで出力ユニットが常に興奮するようにしておき、中間ユニットからの抑制（負の）結合によって中間ユニットが興奮した場合だけ出力ユニットの興奮が抑制されるようにすればよいことになる。ただし、中間ユニットは入力がゼロの場合だけでなく、入力が負の場合にも興奮しないため、非対称の検出が正の値でも負の値でも可能になるよう、結合強度の正負を入れ替えたものを2つ用意しておく必要がある。そこで、中間ユニットが最低でも2つ必要となるわけである。

文献

Anderson, J. R. & Bower, G. H. 1973 *Human Associative Memory*. Winston & Sons: Washington, D.C.

守 一雄 1996 『やさしい PDP モデルの話：文系読者のためのニューラルネットワーク入門』新曜社

守 一雄・都築誉史・楠見 孝 2001 『コネクショニストモデルと心理学：脳のシミュレーションによる心の理解』北大路書房

Plunkett, K. & Elman, J. L. 1997 *Exercises in Rethinking Innateness: A Handbook for Connectionist Simulations*. MIT Press: Cambridge, MA.