http://www2.toyo.ac.jp/~asahi/research/simulation/docs/neuron.docx

技術資料：CAST言語による神経網のモデル（予備実験）

第1版　旭　貴朗　2017.03.31

第3版　旭　貴朗　2019.03.29

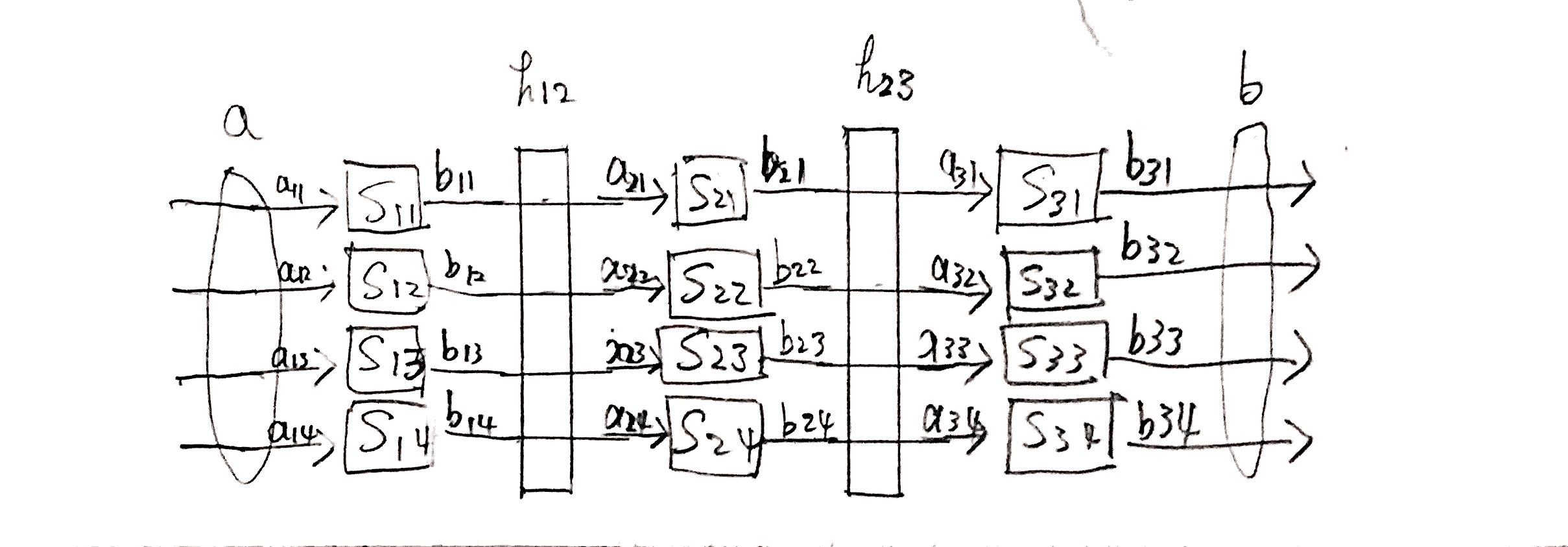


図1　人工神経網(ANN)

1. 概要

【目的】　　　モデル理論アプローチでパターン認識できるかを確認する

【今回の実験】言語CASTを使って，とりあえず神経網のモデルを構築する

【今回の結果】

　言語CASTによる図1のモデルは構築できた（付録1）

　単調なパルス伝達（固定入力の繰り返し，重み固定）はできる

【条件／制限】

　適応層（誤差逆伝播）がなく，直列ネットワーク（図1）のみ

　ただし，接続関数hを修正すれば，複雑なネットワークも構築できる

【推定】

　スプレッドシートで高速化して研究・教育用には利用できる

　適応層をつけて深層学習まで、やろうと思えばできるだろう

　ただし要素数が多くなると遅くなるから，実用化までは？

2. 実験の詳細

　機械学習のうち、特に深層学習に利用されるような神経網のモデルには、通常、次のような特徴がある。1）個別ニューロンの発火点の存在を仮定していない。つまり、パルスの電気が各ニューロンに蓄積されずに流れていく。ただし、ニューロンにおける電気の流れにくさ（重み）に応じて電圧が減少していく。2）電気パルスは入力層から出力層へ直線的に流れていく。例えば図1の左から右に直線的に電気パルスが伝達されていく神経網モデルである。つまりパルスが循環することがない。

　しかしながら、ただ単に神経網だけを考えるならば、実在の生物の脳内神経網に準じて、パルスの循環や発火点の存在があっても良いはずである。この報告書で定式化する神経網では、個別ニューロンに発火点が存在することを仮定し、神経網の電気パルスが循環することも射程に入れたモデルを以下で提示する。理論的には、このモデルは個別ニューロンを特殊化すれば上記1）と2）を満たすという意味で、深層学習で用いられるモデルを含んだものである。

2.1 モデル概要

【個別ニューロンの動作】

　図1のように，12個の神経細胞（ニューロン）が結合したシステムを考える．各ニューロンSijはMoore型オートマトンでモデル化する．それぞれのニューロンに対する入力をaij, 出力をbijと書く．それぞれの状態cijは入力パルスaijを蓄積する（次の状態はcij + aij となる）．ただし，状態が1を超えると発火し，出力1を放出する．その時状態は -0.01に遷移しており，さらに一時刻「待機」してから，状態は0に戻る．待機中はどのような入力も受け付けない．

　　状態遷移関数δij(cij, aij)

　　　cij + aij が0以上1未満なら，そのまま蓄積する（蓄積状態）

　　　cij + aij ≧1なら，（発火状態）-0.01に遷移する

　　　cij = -0.01（発火状態）なら，どのような入力でも次は必ず0に遷移する

　　出力関数λij(cij)

　　　cij = -0.01（発火状態）のときだけ1を出力する

【全体システムの構成】

　全体システムは3層からなっている．図1の左側から順に，

　　入力層S1 … S11, S12, S13, S14の並列結合

　　隠れ層S2 … S21, S22, S23, S24の並列結合

　　出力層S3 … S21, S32, S33, S34の並列結合

と呼ぶ．これら3層が並行処理直列結合したものが神経網の全体システムSである．

【層間結合】

　図1では入力層の各ニューロンS1jの出力b1jは隠れ層のニューロンS2jへの入力となるが，その強さは80%減衰するものとする（ウェイト0.2）．すなわち，a2j = 0.2・b1j である．実験モデルでは，ニューロン間の結合を関数h12(b1) = a2 <-> ︎ a2 = 0.2・b1 で定義している．隠れ層から出力層への結合も同様である．

　しかし，本来ならば，次の特徴をモデル化する必要があるが，やってない．そのような定式化は読者に任せる．固定的な結合関数hを可変的な結合に修正してモデル化せよ．

1）各ニューロンは次の層のどのニューロンとも繋がる可能性を持ち，

2）その結合の強さは時間の経過により変化する．

2.2 CASTモデル

　言語CASTマニュアル（高原他2007）およびシミュレーション作成マニュアル（旭他2011）を参考に，上記の神経網に対する言語CASTによるモデルを作成した（付録1：neuron05.set）．

　付録1のモデルでは，まずpreprocess( ) で（通常のDIALOGウインドウの他に）スプレッドシートウインドウを開いている．シミュレーション中の各ニューロンの状態をリアルタイムで表示するためのものである．

　次にinitialstate( )で，全体システムの初期状態をc0:=[[0,0,0,0],[0,0,0,0],[0,0,0,0]] とする．12個のニューロンを層別に区切っており，各層の初期状態（入力層c1:=[0,0,0,0], 隠れ層c2:=[0,0,0,0], 出力層c3:=[0,0,0,0]）をまとめて書いてある．

　個別ニューロンの状態遷移関数と出力関数はdelta(i, j, cij, aij)とlambda(i, j, cij) で定義している（層番号i = 1, 2, 3）（ニューロン番号 j = 1, 2, 3, 4）．さらに，これらの並列結合として，各層の状態遷移関数delta(i, ci, ai)と出力関数lambda(i, ci)を構成している．神経網全体の状態遷移関数delta(c, a) は3つの層の並行処理直列結合である．ただし，ウェイト付きの接続関数h12とh23は，ここでは単純なものにしている．

　全体の出力lambda(c, a) は出力層の出力と同じb:=lambda(3,c3)である．また1回の出力のたびに，スプレッドシートにその時の入力値，状態値，出力値を表示し，合わせてそれまでの状態遷移の推移（の概略）をグラフ表示するようにしている（showGraph([c, b]）．

2.3 実行結果

　次の図2は，付録1のCASTモデル（neuron05.set）を，モデル理論アプローチで提供されている開発実行環境MTA-SDKに読み込んでコンパイルし，実行している様子である．

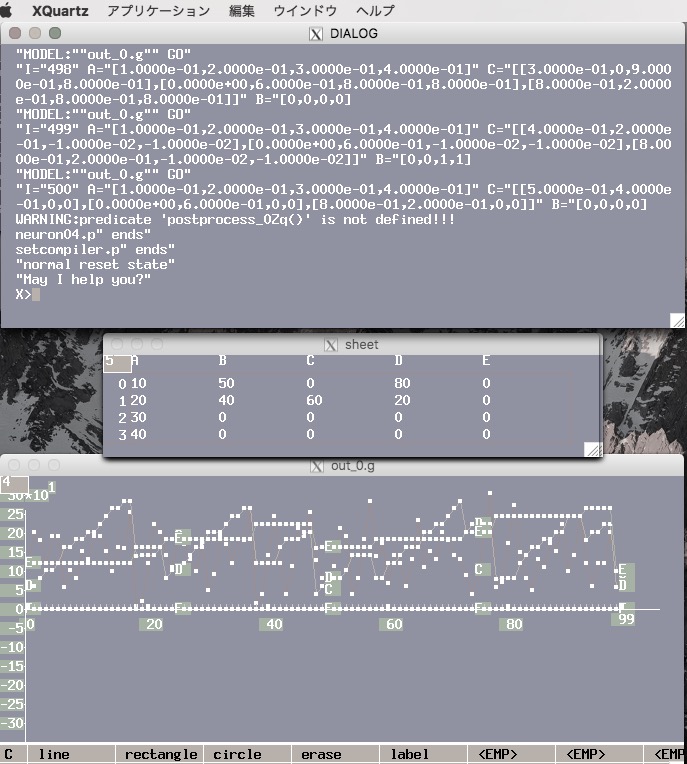


図2　ユーザモデル（付録1）を実行した様子

　図2では3つのウインドウが開かれている．上段は，シミュレーション実行サイクル毎の入力a = (a11, a12, a13, a14), 状態c = (c1, c2, c3), 出力b = (b31, b32, b33,b34) の実際の計算値を時系列で示すウインドウである．図では時刻I = 500 であるから，500回目の入力の結果までが示されている．

　中段は，その時点での入力，状態，出力の値をスプレッドシートに表示するウインドウである．図ではI=500の時の，各種の値が示されている．

　　A列は入力値a = (a11, a12, a13, a14)，

　　B列は入力層の各ニューロンの状態値c1 = (c11,c12,c13,c14)，

　　C列は隠れ層の各ニューロンの状態値c2 = (c21,c22,c23,c24)，

　　D列は出力層の各ニューロンの状態値c3 = (c31,c32,c33,c34)，

　　E列は出力b = (b31, b32, b33,b34)．

実際の計算値は小数点以下であるが，すべての数値を100倍することで視認性を高めている．

　下段は，各ニューロンの状態の変化をリアルタイムでグラフ化するものである．12個のニューロンの状態変化を表示すると複雑すぎるので，3層の概略的な数値（ci1+ci2+ci3）をグラフ化している．入力を蓄積している間は右肩上がりであるが，層内のどれかのニューロンが発火すると数値が下がる．その様子がリアルタイムで示される．

3. まとめ

【実験結果】

　CAST言語による図1のモデルは構築できた（付録1）．ただし，適応層（誤差逆伝播など）のない単純な直列ネットワーク（図1）のみをモデル化している．つまり、要素間結合は固定で，重みも変化しない．

　今回の実験では，単調なパルス伝達（固定入力に対する繰り返し処理）が，モデル理論アプローチで提供する開発実行環境MTA-SDKを使ってできることが確認されただけである。

【結論】

　パターン認識を行うには，接続関数hを修正し，可変の重み付きネットワークを構築し，さらに適応層（オートマトン）を加えることになる．適応メカニズムが分かっていれば（これまでの経験からすると）構築可能である．適応層を付加して，深層学習までやろうと思えばできるだろう．卒論や修士論文のテーマにはなるかも．そのあとは，深層学習の結果を利用して，何に応用するのかである．基礎研究か技術応用か？

　要素数が多くなると遅くなる．しかし，スプレッドシートに各要素の状態を保存する方法（旭 2015）により，ある程度の高速化はできるので，研究・教育用には利用できるだろう．実用化まではいくかどうか？

参考文献

旭　貴朗（2015.11）「モデル記述言語CASTにおける多数要素システムのモデル」『経営論集』東洋大学経営学部, Vol.86, pp.59-73.

旭，高原，齋藤，柴（2011）『モデル理論アプローチによるシミュレーション：Simcast09，開発方法論，モデル』経営情報学会「形式的アプローチ」研究部会内部資料．

高原，齋藤，旭，柴（2007）「第4章 コンピュータ可読に表現する」『形式手法 モデル理論アプローチ－情報システム開発の基礎－』日科技連出版．

付録1

//neuron05.set Neuron Network case(4\*4\*4) out(100\*c)

preprocess() <-> makesheet(4,5,wp), wp.g:=wp, hist.g:=[];

inputsequence()=[0.1,0.2,0.3,0.4];

times()=500;

initialstate()=c0 <-> c0:=[[0,0,0,0],[0,0,0,0],[0,0,0,0]];

delta(i,j,cij,aij)=ccij <-> //個別ニューロン

(cij = -0.01) -> (ccij:=0) //発火後1回待つ

otherwise(

(cij +aij >=1) -> (ccij:=-0.01), //発火状態

(cij +aij <1) -> (ccij:=cij + aij) //通常の蓄積

);

lambda(i,j,cij)=bij <->

(cij=-0.01) -> (bij:=1) otherwise (bij:=0); //発火状態なら発火する

delta(i,[ci1,ci2,ci3,ci4],[ai1,ai2,ai3,ai4])=cci <-> //個別ニューロン層

cci:=[delta(i,1,ci1,ai1),delta(i,2,ci2,ai2),delta(i,3,ci3,ai3),delta(i,4,ci4,ai4)];

lambda(i,[ci1,ci2,ci3,ci4])=bi <->

bi:=[lambda(i,1,ci1),lambda(i,2,ci2),lambda(i,3,ci3),lambda(i,4,ci4)];

func([h12,h23]); //層間接続とウエイト

h12(b1)=a2 <-> a2:=0.2\*b1; //a2:=[a21,a22,a23,a24]

h23(b2)=a3 <-> a3:=0.2\*b2; //b2:=[b21,b22,b23,b24]

delta([c1,c2,c3],a)=cc <-> //神経網全体

writess(wp.g,"c",1,1,floor(100\*a)),

[a1,a2,a3]:=[a,h12(lambda(1,c1)),h23(lambda(2,c2))],

cc:=[delta(1,c1,a1),delta(2,c2,a2),delta(3,c3,a3)];

lambda([c1,c2,c3])=b <->

b:=lambda(3,c3),

showGraph([[c1,c2,c3],b]);

showGraph([c,b]) <-> //100倍した数値をシートとグラフで表示する

cb:=append(c,[b]),

z:=floor(100\*cb),

//xwriteln(0,"z=",z),.stop,

writess(wp.g,"c",1,2,z),

zz:=floor(100\*sum(c)),

//xwriteln(0,"zz=",zz), .stop,

hist:=hist.g,

hist1:=append(hist,[zz]), len:=cardinality(hist1),

hist2:=project(hist1,["r",len-99,len]), hist.g:=hist2,

//hist3:=append([[100,100,100]],hist2),

show1(transpose(hist2),"plot");