

Connectionist models and their properties

Feldman, J. A. & Ballard, D. H. (1982)
Cognitive Science 6, 205-254.

Abstract

コネクショニストモデル(=ニューラルネット)の黎明期に書かれた論文. コネクショニストモデルの一般的・形式的な特徴づけを目指す. ニューロンの情報処理モデルに基づき, コネクショニストネットワークの基本ブロックを構築する. 構築されるネットワークは, 複雑な行動の発現を見据えつつも, 計算論的な堅実さを押えたものである. 現行のコネクショニストモデル研究の基盤となる論文といえる.

Key words: 脳科学, 計算機科学, コネクショニストモデル(ニューラルネット)

1. はじめに

- 認知科学の発展 「情報処理モデルの利用に基づく」
 - ◇ 認知システムを記述する形式言語としての利用
 - ◆ 認知に関する精緻な仮説・理論の構成, 厳密な心理実験を導く
 - ◇ 従来は系列処理を仮定(通常系列計算機をお手本に)
⇒並列処理(コネクショニストモデル)へのシフトが必要
- なぜ並列処理なのか, 並列処理の動機
 - ◇ 解剖学的に「脳と計算機は構造が異なる」
 - ◆ 計算機: 単一のCPUによる系列的な処理
 - ◆ 脳: 多数の処理ユニット(ニューロン)による同時並列的な処理
 - ◇ 計算論的に「系列処理では, 脳内情報処理を計算不可能」
 - ◆ ニューロンの計算処理は数ミリ秒で完結
 - ◆ 人間は数百ミリ秒単位で複雑な行動を生成
 - ◆ 複雑な行動が数百のタイムステップで遂行
 - ◆ 現在のAIシステムは数百万のステップにより, 複雑な行動を実現
 - ◇ 技術的に「並列計算機の構築が現実的に」
 - ◆ 1,000,000 ゲートのチップを作ることが可能
 - ◆ 将来の並列計算機のために, 設計やプログラミングの方法を洗練させる必要
 - ◇ 心理学的に「系列処理とはみなすことができない現象が存在」
 - ◆ 連想記憶・プライミング・知覚競合などに同時並行的な処理が関与
- 従来のコネクショニストモデル (McClelland & Rumelhart, 1981)
 - ◇ 単語認知のモデル
 - ◆ 網膜に投影された画像特徴からのパターン認識
 - ◇ ネットワークの構成要素: ユニットとリンク
 - ◆ 各ユニットは同時並列的に情報を処理
 - ◆ リンクを介したユニット間の相互影響
 - ◇ ユニットの特徵
 - ◆ 活性化値を格納するメモリ
 - 活性化値は時間に応じて減衰
 - 活性化値に上限と下限

- ◆ 入出力の値
 - ユニットへの入力値は、結合する各ユニットの出力値の合計
 - 各時点の入力値に応じて活性値が変化
 - 各時点の活性値に応じて出力値が変化
- ◇ リンクの種類
 - ◆ ポジティブリンク(終点が矢印): 起点から終点へ正の影響を与える
 - 起点ユニットの活性値が高いほど、終点ユニットの活性値は上昇
 - ◆ ネガティブリンク(終点が丸印): 起点から終点へ負の影響を与える
 - 起点ユニットの活性値が高いほど、終点ユニットの活性値は減少
- ◇ ユニットの配置
 - ◆ 第1層: 特徴レベル(網膜に投影される単純な直線)
 - ◆ 第2層: 文字レベル(直線の組み合わせから構成される文字)
 - ◆ 第3層: 単語レベル(文字の組み合わせから構成される単語)
- ◇ ネットワークが説明すること
 - ◆ 網膜像からの単語認知(ボトムアップ的知覚)
 - 特徴→文字→単語への流れ
 - ◇ 特徴(1, 5)の認識
 - ◇ 文字(A)の認識(他の文字を抑制)
 - ◇ 単語(ABLE)の認識(他の単語を抑制)
 - ◆ 文字認知の文脈効果(トップダウン的知覚)
 - 文字の認識は、単一の文字を示される場合よりも、文字列の形で単語を提示された場合のほうが素早い(単語の活性化による影響: 第3層から第2層への矢印)

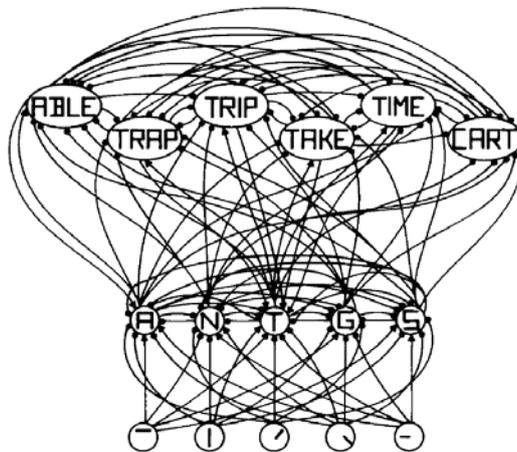


Figure 1. A few of the neighbors of the node for the letter "t" in the first position in a word, and their interconnections (McClelland & Rumelhart, 1981).

- コネクショニストモデルの前提
 - ◇ 個々のニューロンは、多量の記号情報を伝達しない
 - ◇ 伝達する情報は数値のみ
 - ◇ ユニット間の適切な結合によって処理が進行
- コネクショニストモデルと系列処理モデルの違い
 - ◇ 虫が付いているリンゴを見て、「虫食リンゴがある」と発声するまでのプロセス
 - ◆ 系列処理の場合(二本線の矢印)

- 網膜像にオブジェクトが投影
- 視覚システムが, オブジェクトを, 虫付きのリンゴ[虫食リンゴ]とエンコード
- 視覚システムから音声システムへのシグナルの伝達
- 音声システムがシグナルをデコード
- 「虫食リンゴ」と発声
- ◆ 並列処理の場合(一本線の矢印)
 - 網膜像にオブジェクトが投影
 - 投影されたオブジェクトに対して, 複数のニューロンが同時発火
 - それぞれのニューロンが並列的に聴覚システムへ活性を伝達
 - 音声システムにおいて[虫]と[リンゴ]が結合
 - 「虫食リンゴ」と発声
- ◆ 2種類のモデルの差異
 - サブシステム間での情報の転送
 - ◇ 系列処理(複合概念の伝達) ⇔ 並列処理(活性値の伝達)
 - 概念の定義
 - ◇ 系列処理(各概念に対応した信号) ⇔ 並列処理(ユニットの発火パターン)

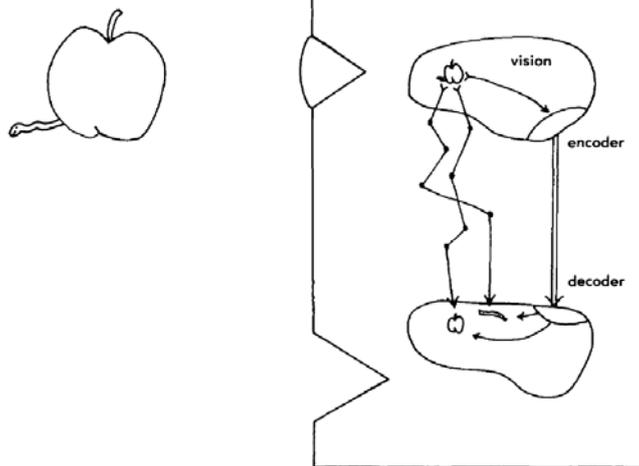


Figure 2. Connectionism vs. symbolic encoding.
⇒ Assumes some general encoding
→ Assumes individual connections

- コネクショニストモデルの限界, なぜコネクショニストが流行らないか
 - ◇ 実装方法の未整備
 - ◆ 個々のユニットをどう結合すればよいか
 - ◇ 記号処理能力の限界
 - ◆ 記号表現の未整備
 - ◆ 高次認知は説明できないという共通認識
 - ニューロン活動の上に, 記号処理のレイヤがある. それは, プロダクションシステム・述語論理・Lispのような高次言語で記述できる. 知能の研究はそちらを重視すべきではないか.
- 本論文
 - ◇ ニューロンベースの認知モデルへ向けた試み
 - ◆ 人間の脳内に記号システムがあるのならば

- ◆
 - ニューロンの結合によってどのようにして記号システムを表現できるか
- ◆ もしニューロンが直接記号システムを構成するのならば
 - モデリングの方法を知りたい.
- ◇ 複雑な問題を見据えつつ, コネクショニストモデルの基礎・体系化を目指す
 - ◆ コネクショニストによるモデリングのための基礎的な要素 (building block) を構築
 - × 本論文で示すネットワークは実装例
 - × 特定の認知機能の正確なモデリングを目指したわけではない
 - × 例として知覚・運動を扱い, 学習は扱わない
- ◇ 従来研究との差異
 - ◆ より一般的・抽象的な形式化
- 本論文の構成
 - ◇ セクション2: ユニットの定義
 - ◇ セクション3: ユニットから構成されるネットワークの基本ブロックを定義
 - ◇ セクション4: より妥当なネットワーク構成の方法を提案
 - ◇ セクション5: 複雑な現象への応用

2. Neuron-like Computing Unit (ユニットの定義)

- モデリングの最小単位: ユニット
 - ◇ 現時点までのニューロンの理解に基づく
 - ◇ 複雑さと機能の異なる複数種類のユニットを定義

2.1. P ユニット (最も単純なユニット)

- ニューロンの電位 (Potential) を取り入れたユニット
 - ◇ $p \leftarrow p + \beta \sum w_k i_k$ $[0 \leq w_k \leq 1]$
 - ◇ $v \leftarrow \text{if } p > \theta \text{ then round}(p - \theta) \text{ else } 0$ $[v = 0 \dots 9]$
 - ◆ p : $-10 \sim 10$ の範囲をとる連続値
 - ◆ v : アウトプット $0 \sim 9$ の範囲をとる整数
 - ◆ i : 入力ベクトル i_1, \dots, i_n
 - ◆ w : 各入力値に対する重み (学習によって調整)
 - ◆ β, θ : 定数
- 生理学的・計算論的裏づけ
 - ◇ ニューロンには膜電位が存在
 - ◆ 各時間の電位は直前の電位に影響 (メモリの役割)
 - ◇ v が整数 (値の種類が10しかない) である理由
 - ◆ ニューロンの発火の振幅は1秒間に数百程度
 - ◆ 1/10秒で心的活動は発現
 - ◆ 発火の振幅 (出力) は少数の値になるはず
- P ユニットの拡張
 - ◇ 電位の減衰: ネガティブリンクを自身に結合

2.2. Disjunctive Firing Conditions and Conjunctive Connections (選言的な発火と連接結合)

- ニューロンによる選択発火をモデリング
 - ◇ 多数の選択肢から一つの反応を返すユニット
 - ◇ ニューロンにおける樹状突起をモデルに組み込むことで実現

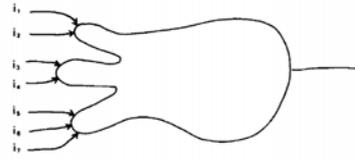


Figure 3. Conjunctive connections and disjunctive input sites.

- Max-of-Sum ユニット
 - ◇ $p \leftarrow p + \beta \text{Max}(i_1 + i_2 - \phi, i_3 + i_4 - \phi, i_5 + i_6 - i_7 - \phi)$
 - ◆ ϕ : 定数 (入力値を制御するため)
 - ◆ 入力値の符号(+: ポジティブリンク, -: ネガティブリンク)
- OR-of-AND ユニット
 - ◇ $p \leftarrow p + \alpha \text{OR}(i_1 \& i_2, i_3 \& i_4, i_5 \& i_6 \& (\text{not } i_7))$
 - ◆ OR 選言的結合, & 接続結合
- Max-of-Sum ユニットと OR-of-AND ユニット
 - ◇ Max-of-Sumユニット: ニューロンに近い電位を実現
 - ◇ OR-of-ANDユニット: シンプルな表現

2.3. Q-Units and Compound Units (状態空間を取り入れたユニット)

- コネクショニストによる状態空間の表現
 - ◇ 電位 p に代わり有限の状態集合 $\{q\}$ を仮定
離散的にユニットの状態を表現
 - ◇ 入力 i に区別された名前をつける
→有限オートマトンに
- Q ユニット
 - ◇ 状態集合 $\{\text{Firing/Null}\}$
 - ◇ $v \leftarrow \text{if } q = \text{Firing then } 6 \text{ else } 0$
 - ◆ ユニットの状態が条件に合致したときのみ出力
 - ◇ 状態遷移の定義

		入力	
		i1 (start)	i2 (stop)
遷移前の状態	Firing	Firing	Null
	Null	Firing	Null

遷移後の状態

- 複合ユニット
 - ◇ 複数のQユニットをカプセル化
 - ◆ 単純なユニットの結合により複雑な動作の発現
 - ◇ 例: ターンユニット
 - ◆ Qユニット: left turner, right turner
 - ◆ Qユニットへの入力: start信号, stop信号
 - ◆ ターンユニットの入力: left信号, right信号
 - ◆ ターンユニットの出力: 左への動作, 右への動作
 - ◆ 状態集合 {left turnerの活動, right turnerの活動}

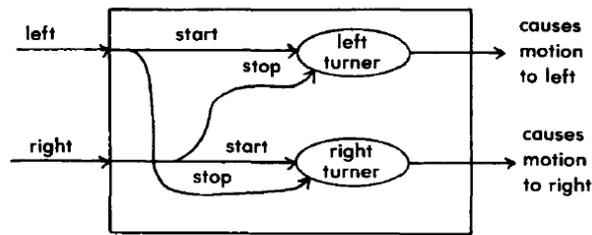


Figure 4. A Turn Unit.

◇ ターンユニットの状態空間

	I_1 (Left)	i_2 (Right)	Output value
Left-Turner	Left-Turner	Right-Turner	$v_1 = 7, v_2 = 0$
Right-Turner	Left-Turner	Right-Turner	$v_1 = 0, v_2 = 8$

2.4. Units Employing p and q (電位と状態を併用)

- ターンユニットに p と q を併用した場合
 - ◇ 曲がる度合いを表現することができる
- ニューロンの飽和を実現 (電位が上限に到達した後に、一定の回復期間へ入ること)
 - ◇ 状態集合 $q \in \{\text{normal}, \text{recover}\}$
 - ◇ 通常状態ではPユニットのように振舞う
 - ◇ リカバー状態では入力を無視し、アウトプットを0に
 - ◇ pの値に応じて状態が変化するものとして定式化

	$-1 < p < 9$	$p > 9$	Output value
Normal	$p \leftarrow p + \sum i$	$p \leftarrow -1 / \text{recover}$	$v \leftarrow \alpha p - \theta$
recover	normal		$v \leftarrow 0$

◇ リカバーの期間=減衰(0へ向かう)によって実現

- ◆ $p \leftarrow p_0 e^{-kt}$
- ◆ パラメータkによってリカバー期間を定めることができる

2.5. ユニットの一般化

- 3つの関数の集合
 - ◇ $p \leftarrow f(i, p, q)$
 - ◇ $q \leftarrow g(i, p, q)$
 - ◇ $v \leftarrow h(i, p, q)$

2.6. 入力値に対する関数

• 修飾子の定義

- ◇ $\forall j, i = r_j \cdot w_j \cdot m_j \quad [j=1, \dots, n]$
 - ◆ r: 先行ユニットから得る値
 - ◆ w: 重み
 - ◆ m: 0か1の値を取る
- ◇ 修飾子mにより、特定の入力をシャットダウンすることが可能

◇ シンプルなネットワークの構築が可能

2.7. ネットワークの例

- P ユニットの対称結合

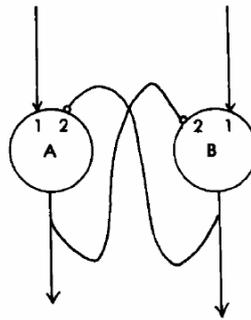
◇ 仮定

- ◆ $w_1 = 1, w_2 = -.5$
- ◆ $p(t+1) = p(t) + r_1 - (.5)r_2$
- ◆ $v = \text{round}(p)$ [0...9]
- ◆ $t = 1$ において, Aは6, Bは5の電位
- ◆ $t > 1$ において, A, Bは2の入力

◇ 特徴

- ◆ リンク間で相互抑制
- ◆ ユニットAは各ステップで1の増加
- ◆ ユニットBは各ステップで0.5の減少
- ◆ ステップ8でユニットAが飽和

Figure 5a.



Suppose A, received an input of 6 units, then 2 per time step
Suppose B, received an input of 5 units, then 2 per time step

t	P(A)	P(B)
1	6	5
2	5.5	4
3	5.5	3.5
4	6	3
5	6.5	2
6	7.5	1
7	9.5	0
8	Sat	0

- 2つのPユニット連合の対称結合

◇ 仮定

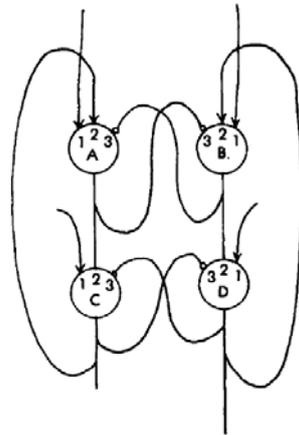
- ◆ $w_1 = 1, w_2 = .5, w_3 = -.5$
- ◆ $p(t+1) = p(t) + r_1 + .5(r_2 - r_3)$
- ◆ $v = \text{round}(p)$
- ◆ $t = 1$ において, AとCは6, BとDは5の電位
- ◆ $t > 1$ において, インプットなし

◇ 特徴

- ◆ 素早い収束

- ◆ AC, BDで連合(ユニット間にポジティブリンク)
- ◆ 一般的に連合ユニットがある場合は素早い収束

Figure 5b.



t	P(A)	P(B)	P(C)	P(D)
1	6	5	6	5
2	6.5	4.5	6.5	4.5
3	7.5	3.5	7.5	3.5
4	9.5	1.5	9.5	1.5
	Sat	0	Sat	0

- ネットワークキューブネットワーク

◇ 特徴

- ◆ 3層のユニット配置
- ◆ 各ユニットは、それぞれ単一の主張を表現「AはBよりも近い」など
- ◆ 整合した主張にポジティブリンク, 矛盾した主張にはネガティブリンク
- ◆ 整合した単一の見えへ収束

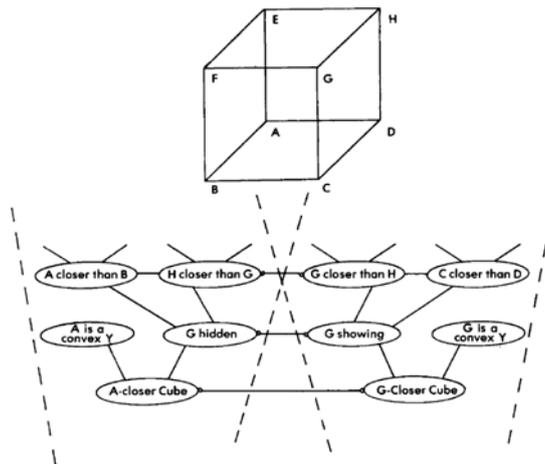


Figure 6. The Necker Cube.

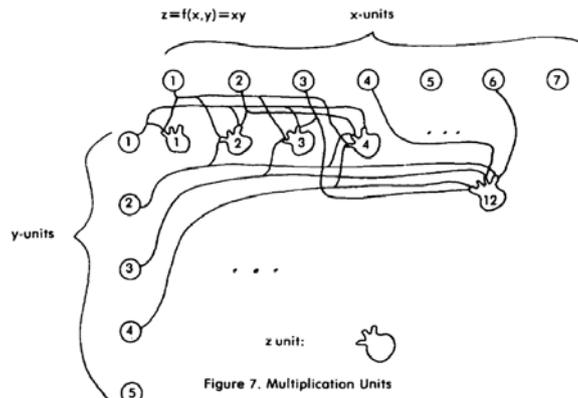
3. Networks of Units (ビルディングブロックの構成)

- コネクショニストの制約

- ◇ ユニット間で記号情報を伝達しない
- 問題
 - ◇ ネットワークをどのように結合すれば問題を表現できるか
 - ◇ ネットワークはどのようにして動的な行動をしめすか

3.1. Using a Unit to Represent a Value

- Unit/value の原則
 - ◇ 単一のユニットは単一の意味ある値を表す
- Unit/value の原則に従ったユニットの例
 - ◇ ネットワークにおけるG-hiddenノード, Gが隠れていることに対する確信度
- Unit/value の原則に対する生理学的根拠
 - ◇ 特定の角度のエッジに敏感なニューロン (Hubel & Wiesel, 1979)
 - ◇ 特定の色に敏感なニューロン (Zeki, 1980)
- 掛け算 ($z = xy$) ネットワーク
 - ◇ Look-upテーブルの形で掛け算を実現 (表頭: x , 表側: y)
 - ◇ x, y, z の各次元について, それぞれ値と対応したユニットを用意
 - ◇ 組み合わせ爆発の可能性
 - ◆ z ユニットの数は $N_x N_y$ に



3.2. Modifiers and Mappings

- Unit/value の原則にしたがうことで, 変数の写像が可能
- 変数「草」の色に値を写像するネットワーク
 - ◇ 保持する知識「カリフォルニア以外で草は緑色. カリフォルニアで茶色」
 - ◇ 修飾子を利用 (丸い頭の線で表現)
 - ◆ 通常の状態では, [Grass]に対し, [Green], [Brown]が写像
 - ◆ ユニット[Calif.]が活性→ユニット[Green]からユニット[Grass]への入力をキャンセル→ユニット[Grass]にユニット[Brown]が写像
 - ◆ ユニット[not Calif.]が活性→ユニット[Brown]からユニット[Grass]への入力をキャンセル→ユニット[Grass]にユニット[Green]が写像

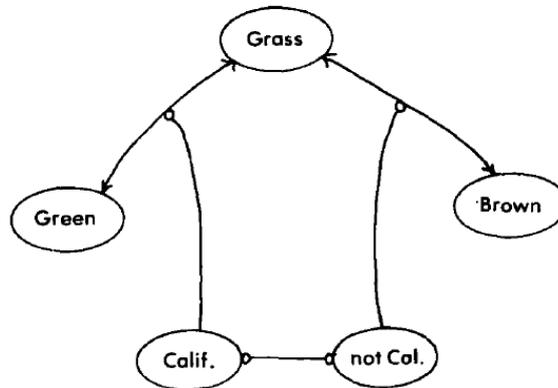


Figure 8. Grass is Green connection modified by California.

- 奥行きを考慮して、網膜上のオブジェクトと物理的なサイズを対応付けるネットワーク
 - ◇ 修飾子を利用したネットワーク
 - ◆ 奥行きユニットが物理サイズと網膜サイズの対応を決定付ける制約に
 - 左側の奥行きユニットが活性化している場合、網膜サイズ2と物理サイズ2が対応
 - 右側の奥行きユニットが活性化している場合、網膜サイズ2と物理サイズ1が対応
 - ◇ 接続結合を利用したネットワーク(3者の結合)
 - ◆ 奥行きユニット・物理ユニットの出力によって網膜ユニットが決定
 - ◆ 物理ユニット・網膜ユニットの出力によって奥行きユニットが決定
 - ◆ 網膜ユニット・奥行きユニットの出力によって物理ユニットが決定

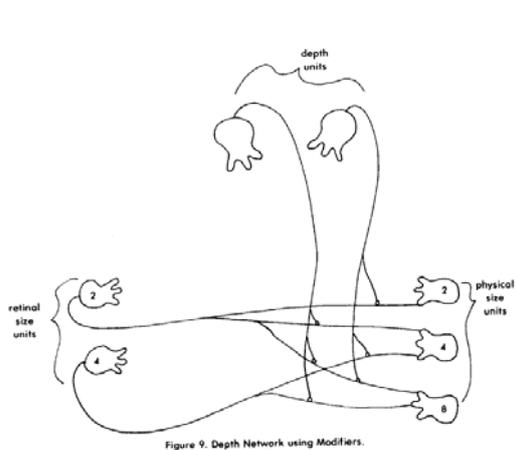


Figure 9. Depth Network using Modifiers.

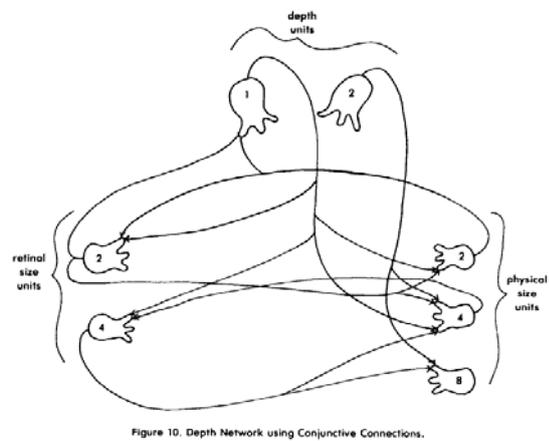


Figure 10. Depth Network using Conjunctive Connections.

3.3. Winner-Take-All Networks and Regulated Networks

- ネットワークによって意思決定はどのように表現できるのか
 - ◇ Winner-Take-All 勝者全奪の利用
- 定義
 - ◇ $p \leftarrow \text{if } p > \max(i_j, .1) \text{ then } p \text{ else } 0$
 - ◆ 各ユニットは、自身の電位とネットワーク中の他の全ユニットからの入力値を比較
 - ◆ 自分の電位よりも高い入力値があれば、活動を停止

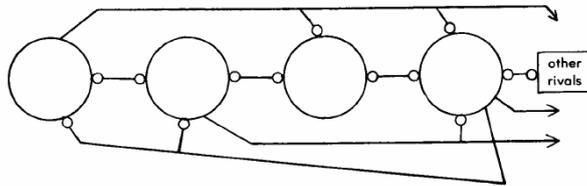


Figure 12. Winner-Take-All
Each unit stops if it sees a higher value

- WTA の弱点: 誤った解への急速な収束
 - ◇ ネットワーク全体の活動が停止
- 安定化ネットワーク
 - ◇ ネットワーク全体における活性値に上限と下限を設ける
 - ◆ UBユニット: ネットワーク全体の活性の合計と上限とを比較, 上限を上回った場合は全ユニットを抑制
 - ◆ LBユニット: ネットワーク全体の活性の合計と下限とを比較, 下限を下回った場合は全ユニットを抑制
 - ◇ ネットワークの活動が持続

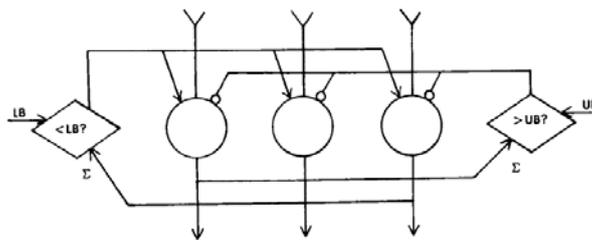


Figure 13. Regulated Network
if sum exceeds UB all units get uniform inhibition

3.4. Stable Coalitions (ネットワーク安定化の条件)

- ネットワークによる意思決定
 - ◇ 連合関係にある特定のユニット群が安定すること
 - ◇ 安定の条件を明らかにすれば, モデリングに便利
- 連合が安定する条件
 - ◇ $p(u) \leftarrow p(u) + Exc - Inh$
 - ◇ 安定条件 $\forall u \in \pi; Exc|_{\pi} > INH$
 - ◆ 集合 π : 潜在的な連合ユニットが含まれる集合
 - ◆ $Exc|_{\pi}$: 集合単体からの活性値
 - ◆ INH はユニットが受け取る最大の抑制値

4. Conserving Connections (ネットワークの改良)

- 脳内のニューロンの数 (10^{11}), リンクの数 (10^{15}), 単一ニューロンが受け取る入力の数 (10^3)
 - ◇ 膨大だが膨大すぎるわけではない
- 使用するユニット, リンクをどう減らしたらよいか

4.1. Functional Decomposition (機能分解)

- ベクトル $t = f(u, v, w, x, y, z)$
 - ◇ t, u, v, w, x, y, z のそれぞれが100の値を持ち, 全ての値が結合される場合
 - ◇ 各ユニットへの入力数 (10^{10})
- 結合するユニットを削減する方法
 - ◇ $f(u, v, w, x, y, z) = g(u, v) \circ h(w, x, y, z)$
 - ◇ $f(g, h), g(u, v), h(w, x, y, z)$
 - ◇ \circ はベクトル間の結合を表す
- 具体例⇒MacClelland & Rumelhart
 - ◇ 中間層を介在することでリンクの数を削減

4.2. Limited Precision Computation (限定された正確さの計算)

- 計算の目的に合わせて計算の粒度を調整
- 例: 粒度の粗い掛け算ネットワーク
 - ◇ z ユニットとして2つの数字範囲を用意

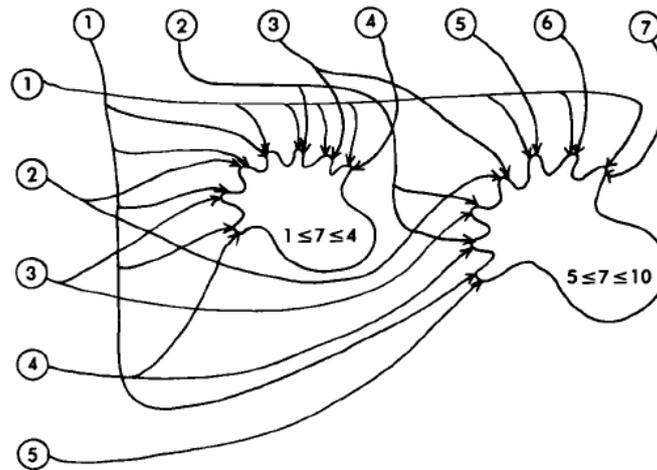


Figure 14. Modified Multiplication Table using Less Units.

4.3. Coarse and Coarse-Fine Coding (コースコーディングとコースファインコーディング)

- コーディングの粒度を調整するテクニック
- コースコーディング
 - ◇ 同時に発火するユニット群を単一のユニットへ置き換え
 - ◆ Figure 15aにおけるb: 重なりあう空間上で発火するユニット群
 - ◆ Figure 15bにおけるa: bがカバーする範囲を単一のユニットに置き換えたもの
 - ◇ 必要なユニット数が F^k から F^k/D^{k-1} へ削減
 - ◆ F: 全体の範囲, D: 同時に発火するユニット群, k: 特徴の次元数

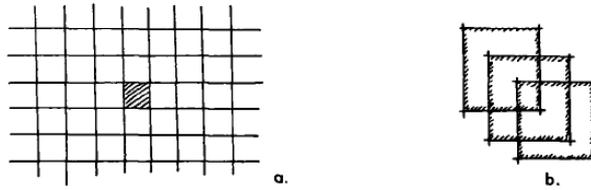


Figure 15a. Coarse coding example. In a two-dimensional measurement space, the presence of a measurement can be encoded by making a single unit in the fine resolution space have a high confidence value. The same measurement can be encoded by making overlapping coarse units in three distinct coarse arrays have high confidence values.

- コースファインコーディング

- 例: 三面頂点(Y, ↑)の認識

- ◆ 認識に必要な特徴次元
 - 形状を決める2つの角度, 方向を決める1つの角度, x, yの位置
 - ◆ 必要なユニット数
 - 5度角の精度で 10^5 のサンプリングポイントをとった場合
 - $N_r = 3.6 \times 10^5$, $N_s = 10^5$ $N_r N_s = 3.6 \times 10^{10}$

- ◇ 一方のユニット数を選択的に減少させた2つのネットワークを用意

- ◆ $N_r = 0.01N_r / N_s = 0.01N_s$
 - ◆ 空間に特化したネットワーク: $N_r N_s = 3.6 \times 10^8$
 - ◆ 角度に特化したネットワーク: $N_r N_s = 3.6 \times 10^8$
 - ◆ 2つのネットワークのアウトプットを接続結合することで空間と角度の両方で出力

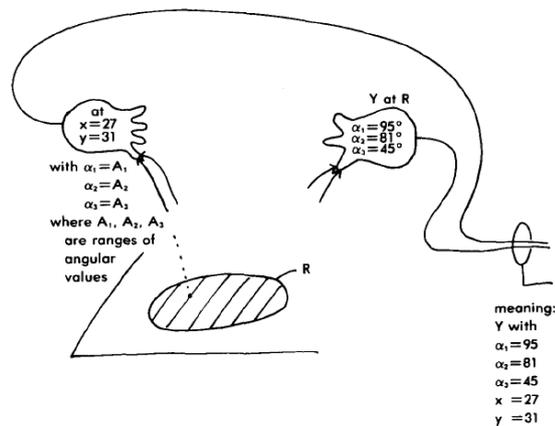


Figure 15b. Coarse angle—fine position and coarse position—fine angle units combine to yield precise values of all five parameters.

- ◇ コースファインコーディングの欠点

- ◆ ゴーストの存在 (何も存在しないところでニューロンが発火)
 - ◆ A, Bが正しい入力だがC, Dも発火

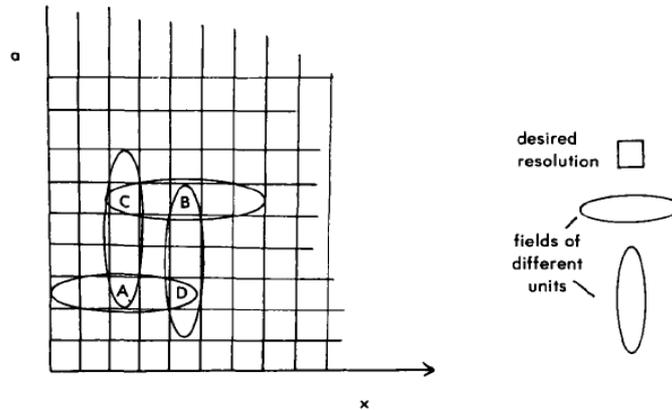


Figure 16. Inputs at A & B cause ghosts at C & D.

4.4. Tuning (調整)

- コーディングの粒度におけるトレードオフ
 - ◇ 粒度をあげれば生理学的な制限を超過
 - ◇ 粒度を下げれば刺激間の区別が困難. 弱い強度のユニット発火を誤って解釈
- WTA は粒度の低い特徴空間で有効な手法

4.5. Spatial Coherence (空間的一貫性)

- 複雑な概念を表現する方法
 - ◇ 複合概念を全座標位置に配置するのは困難
 - ◇ 座標位置には特徴のみに反応するユニットを配置し, 座標位置と独立した概念ユニットを用意
- カラー図形ユニット Red Circle, Blue Square, Blue Circle を仮定
 - ◇ 赤い円が $x=7$, 青い四角が $x=11$ に同時に現れた場合
 - ◆ 入力を単純に加算すれば, Blue Circleが誤って選択(クロストーク)

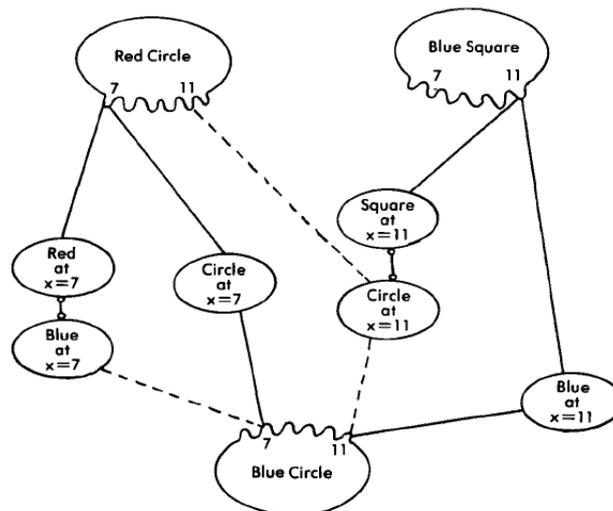


Figure 17. Spatial coherence on inputs can represent complex concepts without cross-talk. Solid lines show active inputs and dashed lines (some of the) inactive inputs.

- クロストークの排除
 - ◇ 相互結合的な入力 (AND) に切り替える
 - ◇ 各座標位置が突起に対応
 - ◇ 空間的に一貫して現れる刺激に収束

5. Applications (応用)

- コネクショニストの力を見せるための例を挙げる
 - ◇ 複雑なオブジェクトの知覚
 - ◇ 系列を伴う運動
- 知覚と運動を共通の形式でモデル化することで、知覚-行動サイクルが実現
 - ◇ 動物は、外界の情報を取り入れ、取り入れられた情報を元に運動し、入力に変化

5.1. Object Recognition

- オブジェクト
 - ◇ 単一の座標位置に複数次元の特徴が存在すること
 - ◇ 例: Fredのフリスビー (ベビーブルー・すばやい・丸い)
- 複合概念の表現
 - ◇ 多数のユニットを単一の概念にどのように結合したらよいかという問題
 - ◇ データベース管理におけるインデックス問題と同構造
- 空間上の単一の点へ注意を向けることで負荷を軽減
 - ◇ Figure 17に注視ユニットを追加したネットワーク
 - ◇ 注視ユニットと他の特徴との結合
 - ◆ 直接的なAND結合
 - ◆ ブースターとしての間接的な結合
- ブースターが双方向性であれば、色から注意位置への影響も表現可能
 - ◇ 何かの理由でベビーブルーに注意
 - ◇ $x=7$ にベビーブルーが存在
 - ◇ $x=7$ へ注意を移動

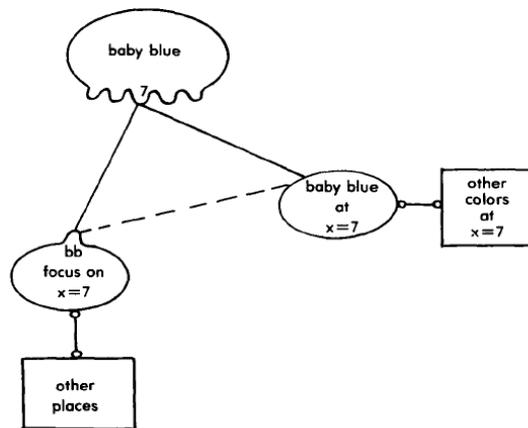


Figure 18. Spatial focus unit can gate only input from attended positions.

- Fred のfrisビーを認識するネットワーク

- ◇ 構成

- ◆ 多次元(青・速度・形状)の特徴を結合
- ◆ $x=7$ への注視ユニットは各特徴ユニットの7に対応する突起と結合
- ◆ 位置に特化した特徴ユニットは注視ユニットから影響を受けない

- ◇ 動作

- ◆ たまたま $x=7$ へ注意を向けた場合(ボトムアップ)
 - 位置に独立した特徴ユニットが発火しFredのfrisビーを認識
- ◆ Fredのfrisビーを探している場合(トップダウン)
 - 位置に独立した特徴ユニットの発火を経由し、 $x=7$ の発火へいたる
- ◆ 物体が混在している場合
 - ストラテジー1(ボトムアップ): 注視ユニットをしらみつぶしに移動し、 $x=7$ へいたる
 - ストラテジー2(トップダウン): Fredのfrisビーの特徴(ベビーブルー)を発火させ、位置に特化した特徴ユニットを得る。 $x=7$ へいたる

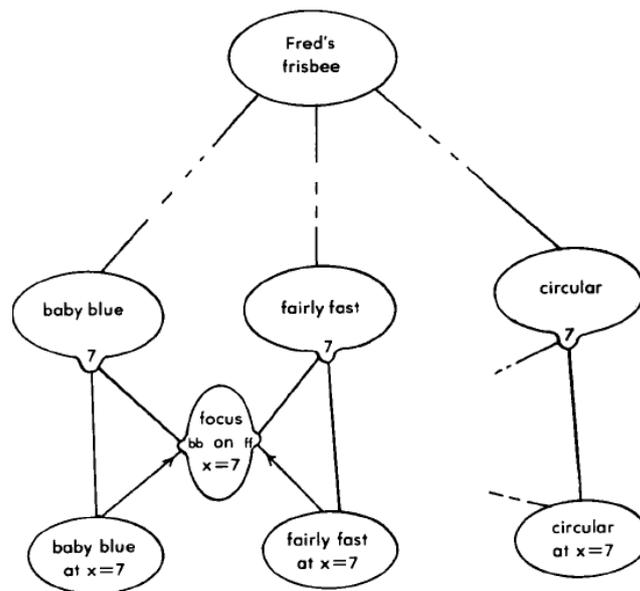


Figure 19. Spatial focus and indexing.

5.2. Time and Sequence

- コネクションリストは時間的な変化を表現することが苦手とされてきた
- 著者らが取り組んできた時間的変化
 - ◇ 長期的変化: 学習(ウェイトの変化)
 - ◇ 短期的変化: 各ユニットの発火系列
本論文では短期的変化を検討
- 長期的変化と短期的変化の差異
 - ◇ シナプスのウェイトの変化は生理学的に確認
 - ◇ だが、読むこと・聞くこと、発話することを説明するのに十分な速度はない
 - ◇ ネットワークへの入力、ユニット間での入出力によって、どのように複雑な行動を表現できるか

- 眼球運動のモデル

- ◇ 網膜座標と対応したユニット(網膜ユニット)が存在
- ◇ 網膜ユニットは筋肉を操作するユニット(筋肉ユニット)と結合
- ◇ 特定の網膜ユニットが飽和→飽和したユニット中心に眼球を移動(座標の更新)
- ◇ 複数の網膜ユニットが飽和した場合→各飽和ユニットは独立に筋肉ユニットへ伝達
→筋肉ユニットの平均に応じて眼球移動
2,4,5,6の網膜ユニットが飽和した場合, 4.25に眼球を移動

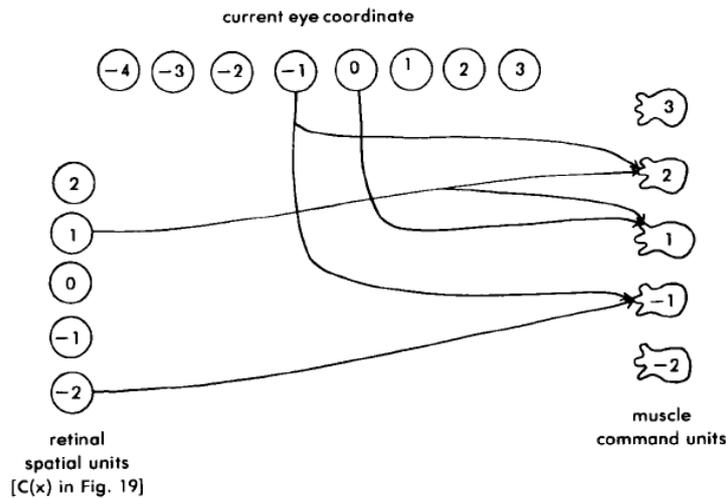
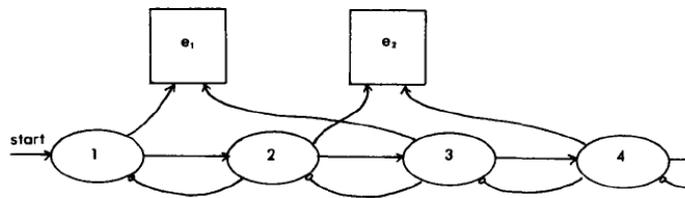


Figure 20. Distributed Control of Eye Fixations

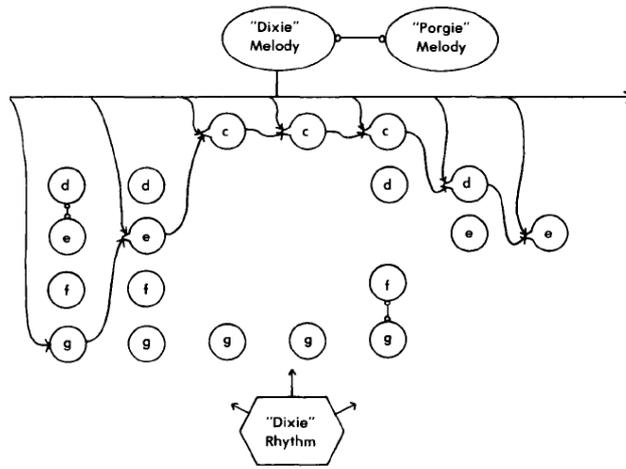
- 系列的な行動の表現

- ◇ ボールを投げること
 - ◆ 筋肉運動の一連の系列
 - ◆ 筋肉を制御する何らかの操作器がある
- ◇ ネットワークのイメージ
 - ◆ 先行ユニットを抑制し, 後続ユニットを抑制
 - ◆ 時間に応じて異なるエフェクタが優位に



a. Sequence and Suppression

- ◇ ディクシーメロディの発声(系列 [g→e→c→c→c→d→e] の発声)
 - ◆ 全体をコントロールするユニットの仮定
 - ◆ 各時点で発声する音をWTAで選択



b. Whistling Dixie

Figure 21. Mapping Space to Time.

- 系列的な入力の理解
 - ◇ 前節の例(あらかじめ系列が定められている)
 - ◇ 音声の理解(あらかじめ何が聞こえてくるか予測できない)
 - ◆ 独立した2つのバッファ(修飾子をもち, 入力を選択)
 - ◆ ネットワーク: 各時点(バッファの長さに対応)にそれぞれの音韻特徴と対応したユニット
 - ◆ 各時点の音声入力に対応し, いずれかの音韻特徴が活性化(複数の活性がおきるときもある)
 - ◆ Sound Analysisにおいて, バッファ中の音韻系列をもとに解釈
 - ◆ トップダウン的に解釈が起きる

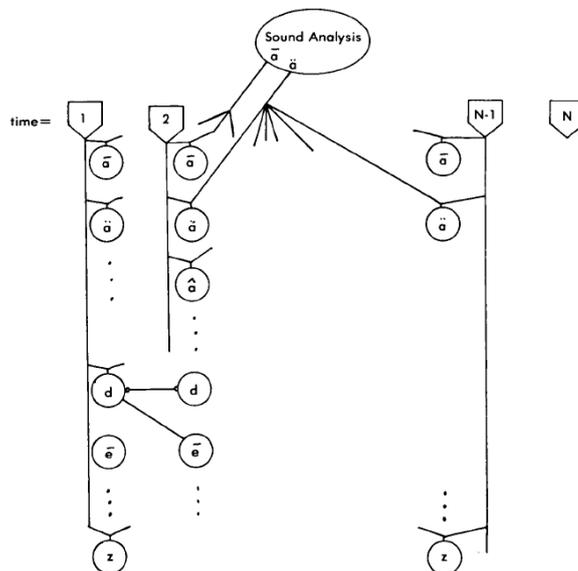


Figure 22. Mapping Time to Space.

6. まとめ

- 本論文からは, 記号処理的アプローチが必要か必要でないかはわからない
- 複雑な高次処理へアプローチするための下地を整えた
- 本論文によって示唆されるアプローチは将来的な脳と行動の研究に有用と信じる