Task Decomposition Through Competition in a Modular Connectionist Architecture: The What and Where Vision Tasks

Jacobs, R. A., Jordan, M. I,, and Barto, A. G.
Cognitive Science 15 219-250 (1991)

担当: 森田純哉

平成 19 年 2 月 23 日

概 要

モジュール型のニューラルネットアーキテクチャを提案した論文です。提案したアーキテクチャは、複数のネットワークを要素として構成されるものです。それぞれのネットワークは、それぞれ別の関数を学習することができます。これによって、ネットワークは、与えられる課題に柔軟に対応した出力を計算することができます。著者は、このようなアーキテクチャは、人間の脳におけるモジュール性の発達を説明するものであると主張します。脳におけるモジュール性とは、たとえば、右脳と左脳が異なる機能をつかさどることや、脳の部位によって異なる表現が取り扱われることなどをいいます。このネットワークを使用したサンプルの実験として，物体の種類と位置を同時に認識する知覚課題が示されます。

はじめに

- 背景
  - 生理学・心理学では，脳がモジュール性を持つということについて広く同意
    - 側性化（右脳と左脳の違い）
    - 部位による機能の違い（視覚野・聴覚野・言語野など）
  - しかし，脳のモジュール性の詳細は不明
    - モジュールはいくつあるのか
    - モジュールの機能
    - モジュール間での相互作用
    - どのように発達するのか
  - コネクショニストモデル（ニューラルネットモデル）の役割
    - これらの問いの探求に，ニューラルネットによるモジュール表現は有効
    - 計算機の上で形式的にモジュールを表現することができる
    - シミュレーションによって，モジュール間の複雑な相互作用を検討できる
- 本論文の目的

  - 新たなモジュール型コネクショニストアーキテクチャを構築
    - 複数の下位ネットワークから構成されるアーキテクチャ
    - 下位ネットワークの間で相互作用が存在
    - 全体としてのシステムは，下位ネットワーク間での競合によって学習

- 本論文の構成
  - セクション 1: モジュール型アーキテクチャに関する理論的考察
  - セクション 2: モジュール型アーキテクチャの詳細
  - セクション 3: 提案したアーキテクチャと従来型アーキテクチャとの比較実験
  - セクション 4: 本論文から得られた知見
  - セクション 5: まとめ

1 モジュール型コネクショニストアーキテクチャの利点

- モジュール型ネットの利点は，学習速度，一般化，表現，ハードウェア的制約の 4 つにある

1.1 学習速度

1.1.1 機能分解による速度向上

- 機能分解とは
  - 複雑な関数を機能に分解して処理すること
- 例: 絶対値の関数

$$f(s) = \begin{cases} -x \ if \ x < 0 \\ x \ if \ x > 0 \end{cases}$$

  - 従来型アーキテクチャ
    - 絶対値のような非線形関数の計算には，少なくともひとつの中間層が必要
  - モジュール型ネット
    - 2 つの下位ネットがあれば，課題を割り振れるので中間層は必要ない
      - 一方のネットが $f(x) = x$ を計算
      - 他方のネットが $f(x) = -x$ を計算
    - 3 層の単一ネットに比べ，2 つのネットを持つモジュール型は素早い学習

1.1.2 クロストークの除外による速度向上

- クロストーク
  - 相互に矛盾した入力
  - ニューラルネットによる学習を阻害する要因として知られる
- モジュール型ネットによって，2 種類のクロストークが除外される

空間的クロストーク

- 空間的クロストークとは
  - 出力層が中間層に矛盾した誤差信号を出すこと
  - 単一の中間ユニットが 2 つ以上の出力ユニットと結合するときに生じる
    - もし，中間ユニットが 2 つの出力ユニットに正のリンクを持ち
    - かつ，一方の出力ユニットの結果は，教師信号よりも小さい，他方は大きい
    - ならば，一方はより大きい活性値を求め，他方はより小さい値を求める
- 空間的クロストークを避けるには
  - 中間ユニットを単一の出力ユニットとのみ結合する (Figure 1B)
  - モジュール型ネットに空間的クロストークはない

時間的クロストーク

- 時間的クロストークとは
  - 異なるときに異なる教師信号を受け取ることによって生じる
  - 特に，類似した入力-教師信号のペアを連続して受け取った後に，それと矛盾する入力-教師信号ペアを受け取るとき，著しい時間的クロストークを受ける
    - はじめの系列で学習した中間ユニットを，次の関数に当てはめることができない
- 時間的クロストークを避けるには
  - 2 つ以上の中間ユニットの集合があると良い
  - はじめの系列では，一つの中間ユニットの集合のみを用いる
  - 中間ユニットの集合を分離したモジュール型ネットにおいて，時間的クロストークは起きない

1.2 一般化

- モジュール型ネットは，刺激に対する反応を適度に一般化する
- 単一ネットワーク
  - 類似した刺激に対して，同じ反応を繰り返す
  - 過剰な一般化をしがち
- モジュール型ネット
  - 入力を部分的に分けることができるので，過剰な一般化をしない

1.3 表現

- モジュール型アーキテクチャによって構成される内部表現は分かりやすい
  - 内部表現＝中間層の状態
- 単一ネットの場合
  - 2 つの課題を単一のネットワークでこなさせた場合，前者の中間ユニットは 2 つの課題をごちゃ混ぜに
- モジュール型
  - 区別される中間ユニットが構成

1.4 ハードウェア的制約

- モジュール型ネットは，必要なユニット数，リンク長を削減することができる
- 刺激を区別するために必要なユニット数
  - $Nk/Dk - 1$
    - k は次元数，N は次元における値，D は一つのニューロンが扱うことのできる値
- 次元数によって，必要なユニット数は組み合わせ爆発
- モジュール型ネットは，次元を分けたネットワークを作れるので，結果的にユニット数が少なくなる

2 提案アーキテクチャ

2.1 概要

- 構成要素 (Figure 2)
  - エキスパートネット（エキスパートネット 1，エキスパートネット 2）
    - 実際に計算をするネットワーク
    - それぞれのネットワークは，ベクトル値を出力 (y_1, y_2)
  - ゲートネット（ゲートネット 1，ゲートネット 2）
    - 調整役のネットワーク
    - エキスパートネットの数に対応した出力ユニット数を持つ
    - 出力ユニットはスカラー値を出力 (g_1, g_2)
  - システムの出力
    - $y = g_1 y_1 + g_2 y_2$
      - $g_1 = 1, g_2 = 0$ のとき，エキスパートネット 1 が全出力を決定
      - $g_1 = 0, g_2 = 1$ のとき，エキスパートネット 2 が全出力を決定
- 学習

- エキスパートネット同士で競合しながら学習
- 結果的に，一つのネットが一つの関数，他方のネットが他方の関数に対応するようになる

- 出力の一般的な式

$$y = \sum_{i=1}^{n} g_i y_i$$

ただし，n はエキスパートネットの数

2.2 学習アルゴリズム

- バックプロパゲーションによる学習（誤差逆伝播法）に基づく
  - 各トライアルの後に教師信号を受け取る
  - 教師信号と出力値との誤差が最小になるように，ネットワーク内のウェイトを調整（最小二乗法）
- 提案アーキテクチャでは，エキスパートネットとゲートネットで異なる誤差関数を使用

2.2.1 エキスパートネットにおける誤差

- 各トライアルにおいて，誤差 J_y を最小にするようにウェイトを変更
- 誤差 J_y は，出力の要求値 y^* と実際の出力 y の誤差二乗和

$$J_y = \frac{1}{2}(y^* - y)^T(y^* - y)$$

2.2.2 ゲートネットにおける誤差

- 概念的説明
  - トライアルの結果，一つのエキスパートネット（勝者ネット）が他のエキスパートネット（敗者ネット）よりも望まれるアウトプットに近いとき
    - もし，システムのパフォーマンスが，過去よりも良いのなら，
      - 勝者ネットへのゲートユニットのウェイトを 1 に近づくように調整
      - 敗者ネットへのゲートユニットのウェイトを 0 に近づくように調整
    - もし，システムのパフォーマンスが過去と比べて向上しないのなら，
      - 全ネットワークのウェイトを中立的な値に近づくように調整
- 形式的説明
  1. 現在と過去のパフォーマンス
     - t を現在のステップ，$J_y(t)$ を現在のパフォーマンス（エラー），α を忘却のパラメータとし，各タイムステップごとに以下の値を算出

$$\bar{J}_y(t) = \alpha J_y(t) + (1 - \alpha)\bar{J}_y(t - 1)$$

2. パフォーマンスの向上に関する判断
   - バイナリ値 λ_{WTA}， λ_{NT}，パラメータ γ によって，システムのパフォーマンスが向上したか否かを判断.
     * If $J_y(t) < \gamma \bar{J}_y(t-1)$
     * Then $\lambda_{WTA} = 1$ and $\lambda_{NT} = 0$
     * Else $\lambda_{WTA} = 0$ and $\lambda_{NT} = 1$
3. 勝者・敗者の決定
   - 個々のエキスパートネット i について，教師信号と出力との二乗誤差を計算する
     $$J_{yi} = \frac{1}{2}(y^* - y_i)^T(y^* - y_i)$$
   - 誤差の最も小さいネットを勝者とし，他のネットを敗者とする
4. ゲートネットの要求値を決定
   - システムのパフォーマンスが向上したとき ($\lambda_{WTA} = 1$)
     * エキスパートネット i が勝者ネットであるとき
       · 対応するゲートネットの出力ユニットの要求値 (g_i^*) を 1
     * エキスパートネット i が敗者ネットであるとき
       · 対応するゲートネットの出力ユニットの要求値 (g_i^*) を 0
   - システムのパフォーマンスが向上しなかったとき ($\lambda_{NT} = 1$)
     * 全てのゲートネットの出力ユニット要求値 (g_i^*) を $1/n$ (n はエキスパートネットの数).

- 一般的定義
$$J_G = \lambda_{WTA}\frac{1}{2}\sum_{i=1}^{n}(g_i^* - g_i)^2 + \lambda_{WTA}\frac{1}{2}(1 - \sum_{i=1}^{n} g_i)^2 + \lambda_{WTA}\sum_{i=1}^{n} g_i(1 - g_i) + \lambda_{NT}\frac{1}{2}\sum_{i=1}^{n}(\frac{1}{n} - g_i)^2$$
  - システムのパフォーマンスが向上したとき，第 1 項から 3 項までを計算
    * 第 1 項: ゲートネットの出力ユニットの誤差の二乗和
    * 第 2 項: ゲートネットの合計が 1 に近いときに最小の値をとる
    * 第 3 項: 個々の出力ユニットがバイナリに近いときに小さい値となる
      · 単一のユニットが 1 に近づき，他のユニットが 0 に近づくように更新（競合学習）
  - システムのパフォーマンスが向上しなかったとき，第 4 項を計算
    * 実際の値と中立的な値との誤差の二乗和
    * 中立的な値に近づくように調整がなされる.

2.3 考察

2.3.1 システムの特性

- システムに内在する 3 つの相互作用
  - ゲートネットが，システムの出力に対する，各エキスパートネットの貢献度を決定
  - エキスパートネットが，ゲートネットの教師信号を決定
  - ゲートネットが，各エキスパートネットの学習率を決定
    - エキスパートネット 1 への誤差信号 g1(y*-y)
    - エキスパートネット 1 への誤差信号 g2(y*-y)
- 相互作用の例
  - ゲートネットの出力が g1，g2 であった場合
    - エキスパートネット 1 が受け取る誤差は y*-y
    - エキスパートネット 2 が受け取る誤差は 0
    - エキスパートネット 1 のみが，そのトレーニングパターンに学習
  - エキスパートネット 1 がエキスパートネット 2 よりも教師信号と類似し，g1 が g2 よりも若干高かった場合
    - エキスパートネット 1 はエキスパートネット 2 よりも大きな誤差を受け取り，より大きく学習する．
    - 結果的に g1 は g2 よりも大きくなる（正のフィードバック）

2.3.2 従来研究との関係

競合学習 (Rumelhart & Zipser, 1987)

- 競合学習の特徴
  - 単一ネット内のユニットが相互に競合しあう．
  - 入力ユニットベクトルと重みベクトルのマッチングに基づく競合．
  - 異なるユニットが異なる入力セットに反応
- 提案アーキテクチャとの違い
  - 本論文は入力のグルーピングが目的ではない．
  - トレーニングパターンのクラスタリングが目的．
  - ユニット間の競合ではなく，ネットワーク間での競合をする．
  - 競合に教師信号が入る

複素ネットワーク (Feldman & Ballard, 1982)

- 複素ネットワークの特徴
  - ひとつのユニットの複数の結合を持たせる
  - 異なる状況において異なる関数を計算可能
- 提案アーキテクチャとの違い
  - 複素ネットワークは，課題分割を人為的に行う
  - 提案アーキテクチャは，課題の分割をシステムが行う

確率学習オートマトン

- 確率学習オートマトンの特徴
  - 行動の集合に確率分布を与える
  - 確率分布から行動を選択
  - 報酬が与えられたら選択された行動に対する確率が上がる
  - 罰が与えられたら選択された行動に対する確率が下がる
- 提案アーキテクチャとの関連
  - ゲートネットの学習アルゴリズムとよく類似
    - システムのパフォーマンスが向上したとき，ゲートネットに報酬が入る

脳の側性化

- 脳におけるモジュール性の議論 (Kosslyn, 1987)
  - 脳には多数のネットワーク（顕著なものとして左脳，右脳）
  - ネットワーク間は競合している
  - ある入力値に反応したものは，その値に敏感に反応するようになる
  - その結果，脳は特定の刺激に対して，素早く反応できるようになる
- 提案アーキテクチャとの関連
  - 類似した議論
  - モジュール型ニューラルネットは，脳のモジュール性を検討する優れたツールであるのかもしれない

3 “ What ”課題と“ Where ”課題

3.1 物体認知に関する心理学的知見

- 人間は，角度や位置によらず物体の認識ができる
- 網膜像を変換し，脳内の物体モデルとマッチングしているから
- 物体認知のプロセスにおいて，変換前の位置情報は失われる

3.2 物体認知に関する脳科学的知見

- 物体認知と位置の認知では異なる脳部位が使われる
- 物体認知のための脳部位: 腹側経路（線状野，視覚前野，下側側頭部）
- 位置認知のための脳部位: 背側経路（線状野，視覚前野，下側頭頂部）

3.3 コネクショニストモデルによる検討 (Rueckl, 1989)

- 課題
  - 入力パターン
    - 5×5 のバイナリ値
    - オブジェクトのサイズは 3×3（9 種類）
    - 出現位置は 9 種類
  - 課題構成
    - 物体認知課題 (What 課題)
      入力の種類を同定
    - 空間決定課題 (Where 課題)
      入力の位置を同定
- システム 1
  - システム 1: 単一型
    - システムの構成
      - 入力ユニット（網膜像に対応）: 25 (5*5)
      - 中間ユニット: 18
      - 出力ユニット: 18（物体パターン 9 + 位置パターン 9）
      - 各層の間で全てのユニットが結合
  - システム 2: モジュール型
    - システムの構成
      - ユニット数は単一型と同じ
      - 中間ユニット 14 と出力ユニット 9 の結合
      - 中間ユニット 4 と出力ユニット 9 の結合
- トレーニング
  - 9 種類のいずれかのオブジェクトが 9 箇所のどこかに提示
  - What タスクではオブジェクトの種類，Where 課題では位置を出力
- 結果
  - システム 2 は課題をすばやく学習．より理解のしやすい表現を構築.
  - システム 1 は空間クロストークによって学習が遅延された
  - 矛盾する教師信号を受け取ったため

  - システム 2 はクロストークの影響を受けない
- 結論
  - 2 つの課題をやるときは 2 つのネットワークを使うべき
- 遣り残したこと（本研究で取り組む問題）
  - それ自身で課題分割のできるシステムの構築

4 シミュレーション実験

4.1 時間的クロストーク

4.1.1 単一ネットによる実験

- 入力パターン
  - 162($9 \times 9 \times 2$)
- トレーニング
  - ランダムトレーニングランダムに入力パターンを受け取る
  - ブロックトレーニングはじめ 81 は what・残り 81 は where
  - 単一ネット
    - 入力ユニット: 25 (網膜像に対応) + 1 (課題識別ユニット)
    - 中間ユニット: 18
    - 出力ユニット: 9（物体パターン 9 + 位置パターン 9）
    - 各層の間で全てのユニットが結合
  - 単一ネット強化版
    - 入力ユニット: 25 (網膜像に対応) + 1 (課題識別ユニット)
    - 中間ユニット: 36
    - 出力ユニット: 9（物体パターン 9 + 位置パターン 9）
    - 各層の間で全てのユニットが結合
- 結果
  - ネットワーク 1 の結果 Figure 5
  - ネットワーク 2 の結果 Figure 6
  - ともに，ランダムトレーニングはブロックトレーニングよりも早い学習
    - 時間的クロストークの影響
  - ネットワーク間の比較
    - ネットワーク 1 よりもネットワーク 2 のほうが早い学習（中間ユニットが多いほうが学習は効率的）
    - ともに，ランダムトレーニングが早い（中間ユニットの多さは，時間的クロストークの影響を除外できない）

4.1.2 モジュール型ネットによる実験

- 入力・トレーニング
  - 単一と同じ
- ネットワークの構成 (Figure 5)
  - エキスパートネット
    入力ユニットは 26，出力は 9
    - エキスパートネット 1 は中間が 36
    - エキスパートネット 2 は中間が 18
    - エキスパートネット 3 は中間なし
  - ゲートネット
    - 入力: 課題識別ユニットから受ける
    - 出力: 3 つのエキスパートネットに対応したユニット
- 予想される学習のパターン
  1. エキスパートネット 1，2 のどちらかが 2 つの課題をこなす
     ゲートネットは常にひとつのエキスパートネットをオンにし，他のエキスパートネットをオフに
  2. エキスパートネット 1，2 のそれぞれが 2 つの課題を分担
     ゲートネットは課題に応じて，エキスパートネット間でのオン・オフを調整
  3. エキスパートネット 1，2 のどちらかが What 課題，エキスパートネット 3 が Where 課題

     Where 課題は線形関数なので，これが最も良い解
- 実験結果
  - 第 3 の学習パターン
    - エキスパートネット 1 を What 課題
    - エキスパートネット 3 を Where 課題
  - Figure 6 学習カーブ
    - 他の学習カーブと比べてもすばやい結果
    - 時間的クロストークにラバストな結果（2 つの学習トライアル間で差なし）
- 考察
  - 機能分解ができる（適したネットに課題を割り振ることができる）
  - モジュール型ネットにより時間的クロストークを避けることができる

4.2 空間的クロストーク

4.2.1 実験 1

- 目的
  - 上記モジュール型ネットの弱点は空間的クロストークの除外
  - 各時点で使われるエキスパートネットは一つ.
  - 同時に 2 つの課題が提示されたときに，エキスパートネットを割り振ることができない.
- 課題の構成
  - What 課題と Where 課題を同時に与える
- ネットワーク
  - アーキテクチャ(Figure 7)
    - 2 つのエキスパートネットと 2 つのゲートネット
    - ゲートネット 1 はエキスパートネットのはじめの m のアウトプットユニットをゲート
    - ゲートネット 2 はエキスパートネットの残りのアウトプットユニットをゲート
  - 仕様 (Table 3)
    - 2 つのエキスパートネット
    - 入力ユニットは網膜像に対応，出力ユニットは，オブジェクトの種類×場所
    - エキスパートネット 1 は中間ユニットを 36 もち，エキスパートユニット 2 は中間ユニットなし
    - ゲートネットの入力は無し
    - ゲートネット 1 ははじめ 9 の出力ユニットをゲート，ゲートネット 2 は残りをゲート
  - 結果の予測
    - 予想 1: エキスパートネット 1 が 2 つの課題を学習
    - 予想 2: エキスパートネット 1 が What 課題，エキスパートネット 2 が Where 課題
  - 結果
    - 2 つの予測ともに裏切る
      - エキスパートネット 2 が 2 つの課題を学習
      - What 課題は線形分離できないため，正解できない
      - 理由は，学習の初期では，エキスパートネット 2 の効率がよいため
      - 局所解から抜け出せない

4.2.2 実験 2

- 目的
  - 空間的クロストークを除外するアーキテクチャの構築
- ネットワークの改良
  - トレーニングの間，ゲートネットのエラー関数の影響が徐々に減少するようにした
  - システムのパフォーマンスが向上しないときに働く関数を変更
  - 1/n（式 7 第 4 項）→
- 結果
  - 空間的クロストークが除外されたと結論付けることはできない
  - Rueckl で研究された単一ネットのパフォーマンスを上回ることはできなかった

4.3 実験のまとめ

- モジュール型アーキテクチャ単一アーキテクチャの比較
  - モジュール型アーキテクチャが異なる関数を異なるネットに割り振ることができる
  - 課題の割り振りがネットワークの形態的特性にあわせたものである
  - モジュール型アーキテクチャが時間的クロストークに対して頑健である

5 課題分割とネットワークアーキテクチャ

- 本研究から導かれるモジュール型ネットワークの設計の指針
  - 課題を良く知ることが重要
  - なんでもいいから作ればいいというものでもない.

6 まとめ

- モジュールは重要
- 領域知識が必要になるとはいっても，学習の問題は重要