

音韻論， 読解力の獲得， そして失読症：コネクショニストモデルからの洞察

• Harm and Seidenberg (1999)

要約

コネクショニストモデルを用いて、読解力の発達と発達性ディスレクシアの基盤を探った。読書に先立つ音韻知識の獲得、この知識がどのように読書学習を促進するか、失読症の音韻的・非音韻的基盤、音韻表現に対するリテラシーの影響、という4つの問題を検討した。単純なフィードフォワードネットワークと比較して、音韻知識をアトラクタネットワークで表現すると、学習と一般化が向上することがわかった。発達性失読症の音韻型と表層型は、通常、語彙処理と非語彙処理の異なる「経路」の障害に起因するとされているが、ネットワークへの異なるタイプの損傷から得られたものである。これらの結果は、コネクショニストの原理を用いて、読解習得の多くの側面を計算によって明示的に説明するものである。

1. 音韻知識の獲得

我々の最初の目標は、子供が読むことを学ぶ前に音韻的な知識を獲得することを近似的に再現することであった。我々は、子供と同じように、単語の音韻形態に曝され、その音韻形態を記憶に表現することを学習するモデルを構築した。我々のモデルは、英語の音韻構造に関する一般的な情報を引き出すことができる。とりわけ、音素セグメントの構造（音素のクラスターで構成されている）と、音素配列に対する制約（音韻戦略）について学習した。モデルが音韻構造について何を符号化したかを評価試験を行ったところ、この音韻表現は、初級読者の音韻知識を近似していると考えられ、正書法から音韻へのマッピングを学習する後続のモデルに使用された。

1.1 音韻表現

我々が採用した音韻表現方式には2つの主要な設計上の特徴がある。まず、音素を分散して表現し、その単位を音素に対応させた。また、この表現は、図1に示した大きなネットワークの一部を構成しており、すべての音素ユニットは、互いに接続され Hinton and Shallice (1991) や Plaut and Shallice (1993) が採用した意味クリーンアップユニットに類似した音韻クリーンアップユニットのセットに接続されている。この表現は、入力ベクトルが一連の音素に対応するという意味でスロットベースであり、スロットベースのアプローチの既知の制限のいくつかを継承している (Plaut et al. 1996)。例えば、最初の子音スロットの /b/ 音素は、最後の子音スロットの /b/ とは別個に表現されるため、ある位置の音素について学習したことが、別の位置の同じ音素に自動的に引き継がれるわけではない。Plaut ら (1996) はこれを「分散問題」と呼んでいる。この問題は、モデルが1つの /b/ の表現ではなく、複数の /b/ の表現を学習しなければならないという点で、音韻表現の学習課題をより複雑にしている。しかし、この表現には、我々の目的にとってより重要な他の特性がある。まず、入力ユニットとクリーンアップ装置の間の相互接続により、ネットワークはスロット間の依存関係を符号化することができる。単純なフィードフォワードネットワークでは、このような依存性がまったく表現できないため、スロット問題はより深刻になる。第2に、この表現により、異なる位置にある音素が音韻的に異なることがあるという事実をモデルで捉えることができる。よく知られている例では、英語の単語の最初の有声破裂音（例 /p/, /t/）は一般的にアスピレートされている。(脚注1) 最後に、音韻コードは母音を中心にした。母音は、音韻や正書法と音韻の対応における主な変動要因である。したがって、母音を中心にすることで、母音が最も顕著な表現の側面における分散問題を最小限に抑えることができた。要約すると、本表現は音素に対応するスロットに音素を使用しているが、入力ユニット、クリーンアップユニットのセット、および母音中心化の間に直接的な接続を組み込むことで、分散問題を最小化している。音素は音素記号で表現されるが、標準的な特徴的行列 (Chomsky & Halle, 1968) とは異なり、ネットワークは特徴やセグメント間の依存関係を符号化することができる。

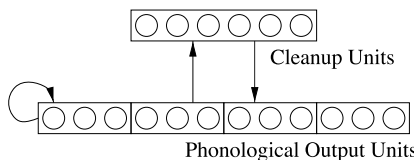


Fig. 1

脚注1：このような同音異義語のバリエーションは、今回の研究では利用されていないが、今後の研究で検討すべきものである。

音素は 11 次元の実数ベクトル で表現され、各次元が、各音素特徴に対応している。音素特徴表現は、最近の音声学や音韻論の理論から導き出されたものである (Gorecka, 1992; Steriade, 1993)。ユニットは -1 から 1 の間の値を表現できる。口唇音や咽頭音のようないくつかの特徴は -1 と 1 の値をとる 2 値性とした。また、声のように -1, 0, 1 の 3 段階の値をとるものもある。音韻特徴は、連続的な勾配に沿って値を取り、音韻のヒエラルキーの符号化を表した。単音節は、これらの音素スロットのうちの 6 つ ccvvc 形式で表現された。単音節は母音中心で、中央の 2 つの母音スロットには 2 重母音が入っている。2 つの母音スロットのうち 2 つ目の母音スロットは、2 重母音でない場合は空の音素としてコード化された（特徴は -1 の値）。例えば BAT という単語の音韻は /bae_t_/ であり、BLADE という単語の音

韻は /bleid_/ である。 使用した音素と特徴を表 1 と表 2 にまとめた。 単音節の音韻表現は 66 ユニットで 11 ユニットの 6 スロットで音素を定義した。 無声音素はすべての特徴を -1 にしてコード化した。

Table 1 Phonological Feature Representation: Consonants													
Symbol	Sonorant	Consonantal	Voice	Nasal	Degree	Labial	Palatal	Pharyngeal	Round	Tongue	Radical	Example	
/p/	-1	1	-1	-1	1	1	0	-1	1	0	0	pat	
/b/	-1	1	0	-1	1	1	0	-1	1	0	0	bat	
/t/	-1	1	-1	-1	1	-1	1	-1	-1	1	0	top	
/d/	-1	1	0	-1	1	-1	1	-1	-1	1	0	dog	
/k/	-1	1	-1	-1	1	-1	-1	-1	-1	-1	0	kite	
/g/	-1	1	0	-1	1	-1	-1	-1	-1	-1	0	give	
/f/	-0.5	1	-1	-1	0	-1	-1	-1	1	0	0	fit	
/v/	-0.5	1	0	-1	0	-1	1	-1	1	0	0	vine	
/θ/	-0.5	1	-1	-1	0	-1	1	-1	-1	0	0	with	
/ð/	-0.5	1	0	-1	0	-1	1	-1	-1	0	0	the	
/s/	-0.5	1	-1	-1	0	-1	1	-1	-1	1	0	sit	
/z/	-0.5	1	0	-1	0	-1	1	-1	-1	1	0	jazz	
/h/	-0.5	1	0	-1	0	-1	-1	-1	-1	-1	-1	hat	
/ʃ/	-0.5	1	-1	-1	0	-1	0	-1	-1	0	0	shot	
/ʒ/	-0.5	1	0	-1	0	-1	0	-1	-1	0	0	beige	
/tʃ/	-0.8	1	-1	-1	1	-1	0	-1	0	0	0	catch	
/dʒ/	-0.8	1	0	-1	1	-1	0	-1	-1	0	0	gin	
/m/	0	0	1	1	1	1	0	-1	1	0	0	mop	
/n/	0	0	1	1	1	-1	1	-1	-1	1	0	not	
/ŋ/	0	0	1	1	1	-1	-1	-1	-1	-1	0	sing	
/r/	0.5	0	1	0	-1	-1	1	1	1	-1	-1	rat	
/l/	0.5	0	1	0	-1	-1	1	-1	-1	1	0	loop	
/w/	0.8	0	1	0	0	1	-1	-1	1	-1	0	win	
/j/	0.8	0	1	0	0	-1	0	-1	-1	0	1	yes	

Table 2 Phonological Feature Representation: Vowels													
Symbol	Sonorant	Consonantal	Voice	Nasal	Degree	Labial	Palatal	Pharyngeal	Round	Tongue	Radical	Example	
/i/	1	-1	1	0	0	-1	0	-1	-1	0	1	heed	
/ɪ/	1	-1	1	0	0	-1	0	-1	-1	0	-1	hid	
/e/	1	-1	1	0	-1	-1	0	-1	-1	-1	1	paid	
/ɛ/	1	-1	1	0	-1	-1	0	-1	-1	-1	-1	head	
/æ/	1	-1	1	0	-1	-1	-1	1	-1	-1	-1	hat	
/ʌ/	1	-1	1	0	-1	-1	-1	1	-1	-1	-1	hot	
/ɑ/	1	-1	1	0	-1	-1	-1	-1	1	-1	-1	paw	
/ɔ/	1	-1	1	0	-1	-1	-1	-1	1	-1	1	toad	
/o/	1	-1	1	0	0	-1	-1	-1	1	0	-1	took	
/ʊ/	1	-1	1	0	0	-1	-1	-1	1	0	1	boot	
/u/	1	-1	1	0	-1	-1	-1	-1	-1	-1	-1	hut	

表 1, 表 2

このテンプレートでは表現できない非屈折の単音節英単語が 95 個あった。 の中には、特に、母音の前後に 3 つの子音を必要とする単語 (例 STREET, WHILST) であった。 これらの単語は純粋に実用的な理由から除外した。 先頭および末尾の子音のスロットを追加すると、音韻単位の数 が 66 から 88 になる。 これにより、音韻要素のサイズは 6,996 個のウェイトから 11,264 個のウェイトに増加し、ネットワークの訓練時間が大幅に増加する。 後述する多数のシミュレーションを考慮すると、訓練時間のコストは 95 語の追加を表現することで得られるわずかな利益を正当化するものではないと考えられた。 以下では、より多くの単語を使ったシミュレーションを行い、これらの単語を追加しても我々のアプローチに他の問題が生じないことを示している。

音韻アトラクタネットワークは、すべての特徴量ユニットを相互に接続するとともに、一連のクリーンアップユニット (事実上、音素表現から自分自身への計算を媒介する隠れユニットのセット) に接続することで作成された。 このような接続を行うことで、ユニットの動作を時間とともに変化させることができ、音韻構成要素は、その状態が自ら変化することができる動的システムとなる。 このようなシステムは、適切に訓練されると、アトラクタの状態、つまり引力の基底を形成することができる (Hinton & Shallice, 1991; Plaut & Shallice, 1993)。このような流域は、固定アトラクタに近い状態がそのアトラクタ状態に引き込まれるような、状態空間の表面と考えることができる。理想的には 66 次元の状態空間全体が、重なり合ったアトラクター流域のランドスケープによって特徴づけられ、ネットワークが置かれ得る無限の状態のいずれもが、時間の経過とともに音素的、音声的に合法的な最終状態に解決されるようになる。

音韻単位が直接つながっていることで、音韻特徴間いくつかの単純なタイプの依存関係を符号化することができた。例えば、ある音素は子音と子音の両方を持つことはできない (表1, 2)。子音が正であれば、子音は負でなければならず、その逆も同様である。この制約は、音素内の子音特徴から子音特徴への負の重みによって符号化することができ、両方とも 0 でない場合には、反対の符号を持つように強制することができる。しかし、英語の音韻には、単純な直接接続では表現できない、より複雑な依存関係も存在する。例えば、音節の最初の 2 つの音素の度数特徴の関係を考えてみよう。表 3 に示すように、両方の次数特徴が 1 に設定されることはないという制約があるが、両方とも -1 になることもあるし、符号が反対になることもある。この制約は 2 つのユニット間の直接的な接続では符号化できず、いわゆる隠れユニットの層を持つネットワークの使用の動機となるような現象である (Rumelhart, Hinton, & Williams, 1986)。隠れユニットが自己アトラクタで利用される場合、つまり、時間の経過とともに自分から自分へと活性化をマッピングするユニットのセットを意味する場合、隠れユニットは、ユニットが出力活性化値を「クリーンアップ」するのを助ける (Plaut & Shallice, 1993)、つまり、パターンを合法的な構成に強制することから、クリーンアップユニットと呼ばれる。

Table 3
Distributions of Degree Features in Consonant Slots Preceding the Vowel

Degree C1	Degree C2	Example	Legal
-1	-1	_RAT	Yes
-1	1	_PAT	Yes
1	-1	BRAT	Yes
1	1	BPAT	No

図 1 に示した音韻アトラクタネットワークでは、各ユニットのダイナミクスは、入力の非線形スキャッシュド和として記述することができる。伝統的なロジスティック関数ではなく、双曲正接活性化関数 (以下 tanh) を選択したのは、この実装にとって魅力的な特性をいくつか持っているからである。双曲線タンジェント (図 2) は、おなじみの s 字型をしているが、その出力曲線は点 (0, 0) を通過する。より伝統的なロジスティック活性化関数は (0, 0.5) を通過し、ユニットへの 0 の入力は 0.5 の出力を意味する。tanh 関数では、入力がない場合、ユニットはゼロの出力を出し、アンビエントな非アクティブ状態を可能にする。さらに、tanh 活性化関数は入力の符号を保持する。負の入力は出力が負になり、正の入力は出力が正になる。これにより、重みをユニット間の相関関係として読み取ることが容易になった。各音韻ユニットには 0.75 に設定された重みとその値に凍結された重みという自動接続がある。音韻単位の活性化 (正または負) は、外部からの駆動入力がない場合、時間の経過とともにゼロに向かって落ちていく傾向がある。

1.2 訓練コーパス

過去の研究で使用されたリストやオンライン辞書など、さまざまな情報源から 3,123 個の単音節の単語を選んだ。固有名詞や形態変化 (複数形、過去形など) は、訓練コーパスのサイズを小さくするために除外した。前述の複雑な発音の単語と同様に、屈折した単語を除外するという決定は、純粋に計算時間に関する実用的なものであった。p.14 では 8,000 語近い単音節の単語でも問題なくモデルを学習できることを示す追加シミュレーションの結果を紹介している。

各単語には Wall Street Journal Corpus (Marcus, Santorini, & Marcinkiewicz, 1993) での出現頻度から得られた頻度を割り当てた。このコーパスは、一般的に使用されている Brown コーパスよりもはるかに大きく、より確実な頻度推定が可能である。この頻度を対数変換して提示確率に変換した (対数頻度圧縮については Plaut et al.1996 参照)。

$$p_i = \frac{\log\left(\frac{f_i}{100} + 1\right)}{\log\frac{m}{100}} \quad (1)$$

ここで f_i は単語 i の頻度, m はコーパス中で最も頻度の高い単語の頻度 (THE 頻度 270 万回) である。確率 p_i が 0.05 よりも小さい単語は 0.05 とした。対数頻度は学習を容易にするために使用されたもので、すべての単語を合理的にカバーするためには、もっと大量の学習が必要となる。例えば Wall Street Journal コーパスでは THE という単語は ISLE という単語の約 5 万倍、CZAR という単語の約 1 万 6 千倍の頻度で出現する。単語の頻度の線形関数である提示確率を使用することは、オンライン学習では計算上困難である。学習コーパスに使われている単語の WSJ 頻度の総和は約 2100 万である。FILCH のようにカウント 1 のアイテムが 90 %の確率で 1 回以上出現するようにするには、約 5,000 万語をサンプリングする必要がある。対数圧縮された確率を使うと、必要なサンプル数は約 2 万 2000 語になる。

1.3 学習方法

続いて、コーパスに含まれる単語の音韻コードを用いてネットワークを学習した。訓練の目的は、ネットワークが英語の単音節の音韻構造を表現できるようになることであった。実際には、子どもたちは音声言語を理解し、生成することを学ぶ過程で、このような表現を身につけていくのだが、私たちはそれをシミュレートすることができなかった。そこで我々は、モデルが時間をかけて音韻コードを表現し、保持することを学習するだけのシンプルな手順を採用した。各試行において、単語の音韻表現は音韻ユニットに固定された。これらのユニットには、時間の経過とともに活性化値が減衰する傾向が組み込まれていた。ネットワークの課題は、ユニットの活性化が減衰する傾向があるにもかかわらず、入力パターンを保持することであった。この課題により、ネットワークは訓練セットの統計的規則性を具現化する動的アトラクタを形成することになる。ウェイト調整は、ネットワークの出力とターゲットとなる単語の音韻との差に基づいて行われた。

最初に、音韻アトラクタネットワークのすべての重みは -0.1 から 0.1 の間の小さなランダムな値が割り当てられた。例外は、各音韻ユニットから自分自身への 66 個の接続で、これは 0.75 で固定されていた。ユニットは、以前の活性化レベルの一部を維持し、出力レベルはすぐには低下せず、徐々に低下する傾向があった。図 3 は、初期値を 1, 0.6, -0.6, -1 とし、ユニットに他の入力がないと仮定した場合の、ユニットの出力の経時変化を示している。ユニットの活性化は、最終的にゼロまで低下する。

ネットワークの学習には Backpropagation through time training algorithm (Williams & Peng, 1990) を用いた。ある時点での各ユニットの出力は、以下の式にしたがって、そのユニットの集約的な活性化の合計の関数となる。

$$o_i^t = f(x_i^t) \quad (2)$$

$$x_i^t = \sum_{j \in U} w_{ij} o_j^{t-1} \quad (3)$$

ここで o_i^t は時間 t におけるユニット i の出力、 x_i^t は時間 t におけるそのユニットへの入力、 f はユニットの入力をその出力にマッピングする活性化 (squashing) 関数、 U はすべてのユニットの集合、 w_{ij} はユニット j から i への重みを表している。各ユニットは、前の時間 tick $t-1$ における他のユニットの出力の加重和 (w で加重されたもの) を取り、これが tick t のそのユニットへの入力 x となる。そのユニットの出力である o は、入力値 x に活性化関数 f を適用した結果である。

これは、出力ベクトル o と目標ベクトル d の間の二乗差の総和であり、式 $E = \sum_i^I \sum_t^T (o_i^t - d_i^t)^2$ に従って、すべての時間刻み T とすべてのユニット I で合計したものである。 E は o と d の間のユークリッド距離の二乗であるため、誤差 E を最小化することは、本質的にベクトル o と d の間の距離を最小化することを意味する。

このネットワークは、事前に設定された時間刻みで実行され、各ユニットは、式 (2) および式 (3) に従って、前のタイムスライスにおける他のすべてのユニットの重みと活性化に従って、各タイムスライスで活性化を更新した。次に、標準的なバックプロパゲーション方程式 (Rumelhart et al., 1986) に従って、ネットワークの各重み w に関する誤差 E の微分を計算した。そして、誤差の微分値に学習率 (μ) と呼ばれる小さな定数を乗じた負の値に応じて、各重みの値を調整した。初期の段階では、学習率 $\mu = 0.01$ が適切であることが判明し、音韻ネットワークの学習にはこの値を使用した。

ネットワークの学習にはオンライン学習を使用した。学習の際には、0.1 未満の誤差はゼロとみなす「ゼロ誤差半径」を 0.1 に設定した。コネクションリストネットワークで一般的に使用される活性化関数は、無限に大きな入力がある場合を除き、極値に達することはできない。0.1 未満の誤差をゼロとすることで、極値を正確に得ることができないネットワークの過学習を避けることができる。

単語は、式 (1) で算出される確率値 p に応じて、学習セットから確率的にサンプリングされた。平均して、 $p = 0.5$ の単語は $p = 0.25$ の単語の約 2 倍の頻度でネットワークによって選択される。ネットワークの学習には、次のようなアルゴリズムを用いた。

1. 訓練コーパスから、頻度分布 (式1) にしたがって、ランダムに単語を抽出する。
2. 時間刻み 0 で 66 個の音韻単位をその単語に適した値でクランプした。
3. ネットワークを 4 回実行し、ユニットのクランプを解除した。
4. tick 2-4 では、各音韻ユニットの出力がその単語の実際の値と比較され、その差がネットワークを通じて逆方向に伝搬され、各重みに誤差の勾配が生成された。
5. 重みは、その誤差勾配に応じて更新された。
6. ステップ 1 に戻る。

各ユニットは正の自動接続重みを持っていたので、その初期値の符号を維持する傾向があった。自動接続重みは、ユニットの活性化が低下するほど低い値で固定されていたため、各ユニットが目標出力に到達するためには、他のユニットや後始末ユニットからの入力活性化を増やす必要があった。100 万回の試行の後、二乗誤差の合計が減少しないことが確認されたため、訓練を中止した。

1.4. 結果

モデルが開発した音韻表現の性質を評価するために、いくつかのテストが考案された。一般的には、劣化した、あるいは不完全な音韻表現を保持・修復するモデルの能力を定量化し、モデルが形成したアトラクタダイナミクスを特徴づけることを目的とした。後の節では、これらのアトラクタの質を、音声知覚の研究で観察された特定の現象（音素復元効果や子音のカテゴリー別知覚など）と関連付けることにする。ここでは読解に焦点を当てているため、このような音声知覚効果をシミュレートするモデルの能力を徹底的に検証したり、幅広いデータをシミュレートしたりすることはしていない。しかし、ネットワークが音韻情報を十分に符号化しているため、単音節に限定するなどの実装上の制限はあるものの、人間で観察された効果のいくつかを生み出すことができることを示している。これらの結果は、この種のアーキテクチャと音声知覚現象との関連性をさらに調査することが有益であることを示唆している。

1.4.1 パターン保持

音韻アトラクターを評価する最初の方法は、モデルが訓練された課題でどれだけうまく動作するかを観察することであった。音韻の出力の正しさを評価するために、最近傍測定が使用され、6つの出力音素スロットのそれぞれについて11の特徴量を既存の音素セットと照合した。実際の出力に最も近いユークリッド距離を持つ音素が、その位置の出力音素とされた。すべての出力音素が正しいものであれば、その単語は正しいと評価された。

また、不正な音素を識別するために、より厳密な2つ目の方法も用いた。モデルの特徴的な出力は、最近傍測定では正しくても、的確な特徴の組み合わせに対応していない場合がある（例えば、特徴的には /k/ に最も近いが、子音ではない出力）。そこで11個の出力特徴のすべてがその音素の表現から0.5以内に収まっている音素が存在する場合に限り、その出力を的確と判断した。

図4は、訓練データセットの評価をまとめたものである。元の3123項目のうち誤った項目はわずか11項目となり、暫時、不正な音素数は10にまで減少した。

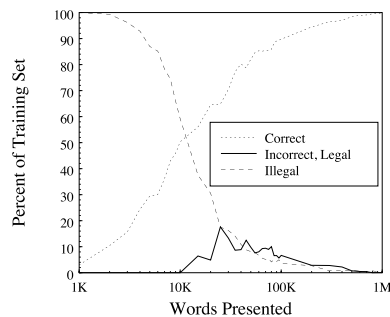
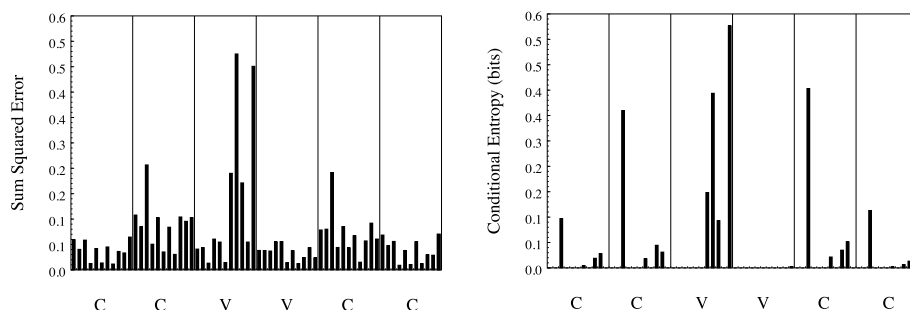


図. 4

1.4.2 パターン補完

次に、パターン補完課題を用いてネットワークを評価した。入力パターンから情報を削除し、音韻構造に関するコード化された情報をもとに、モデルがこのギャップをどの程度埋められるかを観察した。このテストは、訓練セットの全項目に対して行った。66個の特徴量はそれぞれ連続して取り出され、その値は指定されないままであった。与えられた単語の他のすべての特徴は、適切な値に固定された。ネットワークは、学習時と同じように4時刻刻みで動作させた。そして、クランプされていない特徴量の値とその目標値との差を測定し、記録した。この作業は、3123語のうち66個の特徴量すべてについて行われた。このテストでは、ネットワークが不完全な音韻パターンをターゲット言語の音韻に適したパターンに変換する能力を評価した。

各特徴 j に対する誤差 e_j の平均的な大きさは、学習データセットの全3,123語について計算した $e_j = \sum_{i=1}^{3123}$ 。結果を図5に示す。一般に、ほとんどの特徴で誤差は非常に小さく、ほとんどの項目で平均1となっている。学習時には誤差半径0.1を使用していたので、多くの値が0.1付近で推移するのは驚くことではない。各ユニットの活性度（およびターゲット）は-1から1であるため、可能な2乗誤差の範囲は0から4となり、0.1の誤差は非常に小さい。このように、大部分の特徴では、ネットワークは隣接特徴に基づいて、高い精度で正しい値を埋めている。



しかし、いくつかの特徴は高い誤差値を持つ。2番目と5番目のスロットの有声音の特徴が高く、調音場所や調音方法に関連する母音の特徴も高くなっている。このような効果は、すべての音素の特徴的な表現を見ることで説明できる。濁音の特徴は /p/ と /b/ を区別する唯一の特徴

であり、/t/ と /d/, /k/ と /g/ とをはじめとするいくつかの他にもいくつかの最小対がある。他にもいくつかの最小対がある。濁音のビットが指定されていない単語形式があると、ネットワークは正しい値を知る方法がない。つまり、有声音の特徴は、隣接する特徴の影響を比較的受けにくい。一方、子音特徴は、セグメント内の他の特徴によって完全に制約される。そのため、制約の少ない項目で誤差が多くなるのは当然である。

このことをより厳密に示すために、各特徴の条件付きエントロピーを計算した。分布 \mathcal{X} のエントロピー $H(\mathcal{X})$ は次のように定義される。

$$H(\mathcal{X}) = \sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \quad (4)$$

導出と議論については Cover and Thomas (1991) を参照。環境 \mathcal{Y} が与えられたときの分布 \mathcal{X} の条件付きエントロピー（条件付き不確実性）は次のように与えられる：

$$H(\mathcal{X}|\mathcal{Y}) = - \sum p(y) \sum p(x|y) \log_2 p(x|y) \quad (5)$$

音素出力配列の各特徴量について、その音素スロットに含まれる他の 10 個の特徴量に対する条件付きエントロピーを計算した。エントロピーは、学習データセットに含まれるすべての単語を対象に、各単語の頻度で重み付けして計算した。

図 5 下（訳注：右）は、音韻出力の各ユニットの条件付きエントロピー（不確実性）をビット単位でプロットしたものである。予想通り、子音スロットの有声音の特徴は高い不確かさを示している。スロット 1 と 6 の子音は、内側の子音に比べて空の場合が多く、空のスロットの特徴はその環境から容易に予測できるため、不確実性が低いことがわかる。

視覚的にも、平均平方誤差と条件付エントロピーの一致は非常に良好である。2 つの数字のセットに対するピアソン相関は $r = 0.88$, $t(64) = 15$, $p < 0.0001$ とよく一致した。このように、ネットワークの誤差は、訓練セットの条件付き不確実性に近似している。この結果は、二乗誤差の和を用いて学習したネットワークの残余誤差が、学習セットの条件付き分散に近似していることを考えると、驚くべきことではない (Bishop, 1995)。

やや飛んで

2. 読みの学習

本節では、音読学習における事前音韻知識の役割について考察する。音韻表現、読解力獲得、失読症に関連する広範な行動学的証拠から、このような知識があれば、綴りと発音の間のマッピングの学習が促進されると予想した。また、本シミュレーションは、これまでの単語読みモデルよりも子どもの経験に近いものであり、モデルと子どもの成績の比較をより妥当なものにすることができる。

以下のシミュレーションでは 3 つの条件を比較した。訓練アトラクター条件では、事前訓練によって得られた重みが、読解モデルの初期状態となった。非訓練アトラクター条件では、同じネットワーク・アーキテクチャと課題を使用した、ネットワークの音韻アトラクター部分を小さなランダムな重みで初期化した。このモデルは、特徴間の高次の依存関係を符号化する能力を持っていたが、「訓練アトラクター」モデルとは異なり、読解学習の開始時にはこのような知識が備わっていなかった。3 番目のフィードフォワード条件では、単純なフィードフォワードネットワークを使用した。音声特徴ユニットとクリーンアップ装置の間の接続は排除され、正書層から隠れユニットへ、隠れユニットから音声ユニットへの接続のみが残された。このモデルでは、特徴間の依存関係を表現する能力がより限定されていた。これらの条件により、音韻的知識が読解学習の前に備わっていることと、単に読解学習の過程でそのような知識を学習し表現する能力があることとの相対的な重要性を調べることができた。これまでの知見から、フィードフォワードネットワークは、音韻構造を表現する能力が制限されているため、特に非単語の一般化において成績が低下することが予想された。

2.1 アーキテクチャ

訓練アトラクタと非訓練アトラクタで使用されたアーキテクチャを図 9 に示す。入力層は、単語のスペルを表す 208 個のユニットで構成されている。これらは中間レベルの 100 個の隠れユニットに完全に接続されており、さらにその隠れユニットは前述の音韻アトラクタネットである出力表現に全結合された。フィードフォワード条件では、音韻ユニットとクリーンアップユニットの間の接続を排除した。いずれの場合も、単語の綴りを発音に対応させるようにモデルを訓練した。

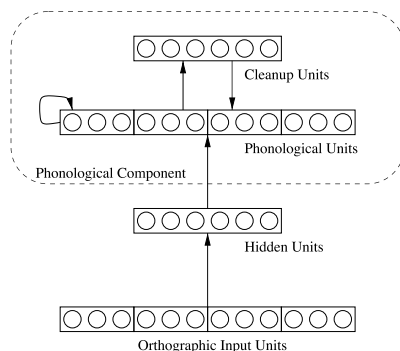


Figure 9. The architecture of the reading model.

8 つのスロット（26 個）で 8 文字までの単語を表現し、各スロットは文字の位置に対応し、各ユニットは 1 文字を表す。単語は母音中心で、最初の母音はスロット 4 に、子音は母音から外側に向かって伸びていくように表現されている。ある文字がスロット内に存在することは、そのユニットを 1 に、それ以外を 0 にすることで示された。

この正書法表現には、かなりの非効率性がある。例えば、オンセットとコーダの位置にも母音ユニットがあるが、これらの位置には母音が発生しないため、常に 0 に設定される。逆に、母音の位置にある子音のユニットは常にオフになる。この表現はできるだけシンプルなものを選び、英語の正書法の構造を、基本的なアーキテクチャによって課せられたものの以外はほとんど組み込まないようにした。

2.2 訓練方法

訓練には backpropagation through time アルゴリズムを使用し、オンライン学習を行い、1 節と同じ手順で訓練用の単語を選択した。音韻学習で使用したのと同じ訓練コーパスを読解訓練に使用した。**単語が選択された後、正書法ユニットはその単語の綴りに対応するパターンで 6 回の時間刻みで固定された。ユニットの活性化は 6 回の時間刻みで更新された。**最後の時点で、音韻ユニットの出力を単語の音韻ターゲットと比較した。BPTT アルゴリズムでは、各出力ユニットの出力とターゲットの間の不一致がネットワークの出力ユニットに入力される。各出力ユニットの誤差は、その出力ユニットに接続されているすべてのユニットの誤差を計算するために使用される。例えば、隠れユニットの誤差は、出力ユニットの誤差への貢献度の関数である。各隠れユニットが蓄積した誤差は、同じ方法ですべての入力ユニットの誤差を決定するために使用される。同様に、音韻アトラクタネットワークのクリーンアップユニットは、それらが接続されている出力ユニットの誤差に基づいて誤差を蓄積する。この意味で、全体的なエラーの「責任」は、出力ユニットからネットワークを介して後方に伝搬される。各ユニットに「責任」が割り当てられると、誤差を減らす方向に重みを少しずつ変えて更新することができる。その後、誤差は破棄され、新しい単語が選択されるというサイクルが繰り返される。このようにして学習を進めていくと、学習セット内の各単語の誤差を最小化するような値が重みとして設定されていく。規則語は、ターゲットと入力に類似している限り、重みに同様の影響を与える、例外語は重みを異なる方向に導く。例えば GAVE, BRAVE, SAVE といった単語の正書法的ライムから得られる重みは、いずれも音韻ターゲットが似ているため、重みの値への影響も類似する。これらの単語が作成する隠れユニットに対する活性値パターンは、綴りが重複していることによる類似性と、オンセットが異なることによる相違性がある。隠れユニットの活性値の類似性と出力ターゲットの類似性が相まって、規則に従った動作が生み出される。一方、例外語である HAVE の場合、ネットワークは、規則的な近傍で作られたデフォルトの動作を上書きするために AVE の環境における H の存在に対する感度を獲得しなければならない。

試行錯誤の結果、モデルの読解成分（正字層から隠れユニットへの結合、および隠れユニットから音韻表現への結合）には、学習率 μ を 0.005 とするのが適切であることがわかった。値が低いと学習時間が非常に長くなり、値が高いとネットワークが不安定になる。この値は、音韻アトラクタで用いられた値（0.001）よりも高い。初期の研究では、モデル全体の学習率を 0.001 とし、学習時間は長くなったが（1 回の実行に数日かかる）、学習率を高くした場合と定性的には同様の性能を示した。

もう 1 つの重要な特徴は、2 種類の訓練試行を織り交ぜて行ったことである。前述の音韻獲得の段階では、モデルは音韻保持課題で訓練された。ここでは、正書層から音韻論へのマッピングという 2 つ目の課題をモデルに学習させる。しかし、モデルには音韻構造の知識も保持させたいと考えている。読解課題のみでモデルをトレーニングすると、このタスクには最適な重みが得られるが、音韻保持課題には必ずしも最適ではない。異なる課題での訓練がブロックしていると「破局的干渉」(McCloskey & Cohen, 1989) と呼ばれる現象が発生する。このような条件では 2 番目の課題の訓練では 1 番目のタスクに関連して学習したこと的一切を保持できないという結果になる。この問題を解決するには、単純に 2 つの課題の訓練をインターリーブすることである(Hetherington & Seidenberg, 1989 参)。第 2 課題の訓練中に、この種の試行を数回追加すれば、第 1 課題の忘却は避けられる。また、このような課題のインターリーブは、課題によって厳密にブロックされていない子供の経験に関して、より現実的である。子供は話し言葉に触れることで広範囲の音韻知識を獲得するが、読み聞かせが始まると話し言葉の経験が終わるわけではない。

このように、読解課題の訓練には、音韻保持課題の追加試行を織り交ぜた。後者の課題は、音韻パターンを符号化して保持する課題であるため、ここでは「聞き取り」課題と呼ぶことにする。各訓練試行では、乱数が計算される。この乱数に基づいて、その訓練サイクルは、読解または聴解のいずれかのサイクルとなった。80 %の試行では読解課題、20 %の試行ではリスニング課題でモデルを学習させた（脚注2）。読解試行では、ネットワークは上述のように訓練された。リスニング試行では、1 節と同様にモデルを訓練した。したがって、音韻結合係数は、両課題での成績を促進するような値を仮定する必要がある。

脚注 2 この比率は、子供が経験するであろう読書と聴取の比率よりも高くなっている。様々な比率を検討した結果、2 つの発見があり、この比率に至った。1 つ目は、リスニング試行回数を 20 %にすることで、音韻の著しい脱落を防ぐことができたこと。2 つ目は、リスニング試行回数の割合を増やしても、リスニング課題と読解タスクのどちらの習得にもほとんど影響しないことであった。そこで、訓練時間を比較的短くするため、80 対 20 の比率を採用した。リスニング試行回数の割合を増やしても、基本的には同じ結果となるが、ネットワークの学習には時間がかかる。

2.2.1 訓練の長さ

3 つの条件のそれぞれについて、異なる被験者の 8 回のシミュレーションを行った。各試行では、正字ユニットから隠れユニットへの重み、および隠れユニットから音韻ユニットへの重みを -0.1 から 0.1 の間の値にランダム化した。訓練アトラクタ条件では、8 回の実行ごとに 2 節と同様に音韻ネットワークを訓練した。未訓練アトラクタ条件では、アトラクタネットワークにランダムな重みが割り当てられ、フィードフォワードネットワークではアトラクタネットワークは存在しなかった。全ネットワークにおいて、初期重みと単語の正確な提示順序は、初期乱数シードによって決定された。各学習条件において、8 回の学習は、初期シードを除いてすべて同じであった。以下に示す結果は、特に断りのない限り、各条件における 8 つのネットワークの平均性能を示している。

各試行では、1 節と同様、音韻訓練段階で訓練されたアトラクタネットワークに 100 万語の単語を提示した。続いて、読解フェーズで 1000 万語の単語を使って訓練を行った。読解段階では 80 %が読解、20 %がリスニングであったため、各試行では、合計 300 万回のリスニング例と 800 万回の読解例を経験させた。未訓練アトラクタネットワークには音韻の事前学習は行わず、読解段階で 1000 万語の単語を学習し、読解と聴解の試行回数は上記の分布とした。フィードフォワードネットワークは 1000 万回の読解試行で学習した。

各段階（リスニングと読解）では、多数の学習試行を行い、成績の漸近的なレベルを検討することができた。後述するように、結果として得られた学習曲線は非常に非線形であり、最初の 100 万語程度の単語提示では急速に学習が進み、その後はゆっくりとした学習となった。

2.3 Results

まず、訓練アトラクタネットワークの性能を説明し、次に未訓練アトラクタおよびフィードフォワード条件との比較を行う。

2.3.1 単語の成績

各モデルが訓練データセットの単語に対して出力したものを、2つの基準で評価した。まず、単語中の各音素は、前述の最近隣テストを用いて評価した。訓練データセットに含まれる音素のうち、ネットワークの出力にユークリッド距離で最も近い音素を各位置で決定した。そして、これらの音素をターゲット音素と比較した。音素がターゲットと同一であるか、あるいはあらかじめ定義された音素の等価クラスのメンバーである場合に、正しいと判断した。同等クラスの根拠は、参加者が作り出すことができても、検出できないような特定の音素の生成にはいくつかのバリエーションがあるということである。これらのクラスは、/a/ と /l/ (例 COT と CAUGHT), /ow/ と /o_/ (例: 英語のいくつかの方言で、末尾に /m/ の音を持つ DOE の母音と、持たない DOME の母音の違い), /ej/ と /e_/ (後者は /ej/ をより短く、より切り取ったもの) であった。単語 (または非単語) に含まれるすべての音素がターゲットの音素に最も近いものでなければ、その項目は正解とみなされない。

さらに、各出力音素の特徴量は、訓練セット内の正規の音素に対応していなければならない。つまり、訓練セット内には、各出力特徴量がその音素の値から 0.5 以内に収まる音素がなければならない。最近傍基準は、特定の特徴の組み合わせがどの音素にも対応していなくても、計算された出力が他のどの音素よりも正しい音素に近いという可能性を与えるものである。2つ目の採点基準を導入することで、このような試行は不正解と判定されるようになった。

図 10 左は、訓練セットの項目に対する3つのネットワークの平均成績を時間経過とともに示している。訓練アトラクタネットワークと未訓練アトラクタネットワークの間にはほとんど差がない。フィードフォワードネットワークの性能はやや劣り、アトラクタネットワークと同じ漸近レベルに達するが、よりゆっくりとしている。漸近では、訓練アトラクタネットワークの平均正解率は 98 %であった。エラーの大半は、低頻度例外語 (例: CHOIR) や、正書法的に珍しい低頻度語 (例: MYRRH) であった。これらの結果は Seidenberg and McClelland (1989) が提示した結果や, Waters and Seidenberg (1985) などが報告した行動データと一致している。

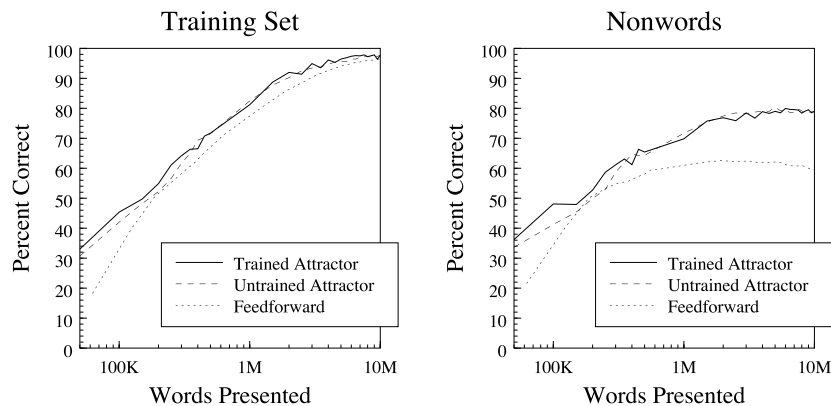


Fig. 10

2.3.2 非単語生成

モデルの新規項目への汎化能力は、364 個の非単語を使って評価した。Glushko (1979) の実験 1 の 86 個の疑単語と McCann and Besner (1987) (脚注3) の160 個の項目のうち 156 個を合わせて 239 個の項目とした (Glushko と McCann & Besner の研究では 3 個の項目が重複していたので、2つのセットの合計よりも 3 個少ない)。さらに 125 の項目は、既存の単語の本体からオンセットを置き換えて非単語を形成することによって生成された (Seidenberg, Plaut, Petersen, McClelland, & McRae, 1994) (脚注4)。ほとんどの項目では、ネットワークの出力を採点する際、正しい発音は 1 つしか認められていない。いくつかの項目では、ネットワークが 2 つの可能な出力のうちの 1 つを生成した場合、正しいと評価された (例えば非単語 DOMB は BOMB のとき /dam/ と COMB のとき /dom/ のどちらにも発音できる)。このスコアリングは Seidenberg ら (1994) が報告した行動データと一致している。Seidenberg らの研究では 500 以上の非単語の中で最も一般的な 2 つの発音が、参加者の回答の 90 %以上を占めていた。非単語の一般化能力の発達は、図 10 右にまとめられている。漸近線において、訓練アトラクタネットワークは、厳しい基準で測定された非単語セットの平均的な正解率が 79 %であり、より緩い最近傍測定では 88 %の正解率であった。訓練済アトラクタネットワークと未訓練アトラクタネットワークの性能は、学習初期に未訓練アトラクタネットワークの性能がわずかに劣っただけで、非常に似通っていた。しかし、フィードフォワードネットワークは、訓練中、非単語セットの成績が非常に悪かった。

脚注 3. 4 項目 (BINJE, FAIJE, JINJE, WAIJE) は、母音の 2 つ後の位置に J という文字が含まれていたため、除外された。このスロットにあるこの文字は、訓練セットでは見られない。このように、モデルはこれらの項目を正解することができなかった。この問題は、スロットベースの表現方法の本質的な限界を表している (議論は PMSP を参照)。この問題は、このギャップをカバーする多義語 (BANJO, CONJURE など) を使ってモデルを学習した場合には発生しなかった。この問題が発生したのは McCann と Besner と Glushko の研究では、これらの単語だけであった。

脚注 4 非単語の全セットは <http://siva.usc.edu/~mharm/papers/dyslex.psyrev/nonword.stim.pdf>