2010 年 7 月 2 日
情報システム工学科 塩貝亮宇

4.THE ENCODER PROBLEM

'encoder problem'(我々と Sanjaya Addanki により提案) は並列ネットワークの様々なコンポーネント中における (情報伝達,communicationinformation) の再帰的処理のシンプルな抽象化だ.

私たちは, この問題を学習アルゴリズムを (試す, テストする,test out) ために用いる.

なぜならば, 最適解と (好み, 似たもの, 同類,like) なものは (明らか, 確実,clear) であり,

そして, それを発見することは (簡単でない, ささいでない, 重要な,nontrivial) だ.

可視素子の 2 つのグループ, 指定された V_1 と V_2,

それらの状態を (伝達,communicate) することを望む 2 つのシステムを示す.

各グループは v 個の素子を持つ.

私たちがここで考える (単純, 平易, シンプル,simple) な定式化は,

各グループが 1 つの素子だけ (同時に, 一度に, 一回に,at a time) を持つ,

だから, 各グループの状態は v だけ異なっている.

V_1 と V_2 は直接的に結合していないが,

どちらも一群の隠れ素子群 H の h 素子のグループと結合しており,

(意-)$h < v$ であるため, H は (能力, 容量,capacity) を制限されたボトルネックとして振る舞うだろう. このボトルネックは V_1 と V_2 の状態についての情報が (絞る, 圧力を掛ける, 押し込む,squeezed) まれるものである.(-意)

全てのシミュレーションは全ての重みにゼロが代入されたところから始まり,

そのような問題の解を見つけることは,

H を通したコミュニケーションについてどんな (先進的な, 先験的な, 演繹的な, 推測的な,prior) (監修, 慣例, 協定, 在来技法,convention) なしでコードセットの意味に (ついて一致する, agree upon) (関係する, に及ぶ, に興味を持つ, come to) 2 つの可視素子群のグループを必要とする.

可視グループ間の完全なコミュニケーションを (許可する, 許す,permit) ために, それは $h \geq log_2 v$ の場合でなければならない。

私たちは、$h = log_2 v$ の最小の場合と h が $log_2 v$ より幾らか大きい場合を (調査し, 研究し,investigated) た.

全ての場合において,

V_1 内の素子の 1 つと V_2 内で一致する素子を (指定された, 指定した?,specified) 長さ $2v$ の (蓋然性が等しい,equiprobale) が等しいベクトル v(から成る, consist of) ネットワークについての環境は他の全ての素子を'off' にして, ともに'on' にすべきだ。(should be on together with all other unit off.)

各可視グループは (内部,internally) で完全に結合されており, そして, 各 v は H へ向けて完全に結合しているが, H 内の素子群は互いに結合していない.

なぜならば,

逐次的機械上でのシミュレーションの (極端な, 厳密な, 厄介な,severe) 速度の制限であり, そして, その学習は多くの焼きなましを必要とするため,

私たちは最初に (符号化問題,encoder problem) の小さな (版,version) について実験した.

例えば, 図 2 は $v = 4$ と $h = 2$ の条件のもとで'4-2-4' 符号化問題への良い解を示す。

可視グループと H 間の相互接続は 2 値符号化が開発された.

-各可視素子は H 内の素子群で'on' と'off' 状態の (異なるパターン,different pattern) を (引き起こす, 原因となる,cause), そして,V_1 と V_2 内での一致は (全く同じ, 同一の, そっくりの,identical) パターンを (支持する, 養う, 貫く,support).

留意すべきは, どのような方法で

2 番目の素子のバイアスが全ての H 素子を (オフ,turn off) にした素子を表現するコードという事実について (正に, 確かな,positive)(釣り合いをとる,compensate) (意-) 点である (-意).

コードは表現する素子は全ての H の素子をオフにした

how the bias of the second unit of V_1 and V_2 is positive to compensate for the fact that the code which represents that unit has all the H unit turned off.

how the bias of the second unit of V_1 and V_2 V_1 と V_2 の 2 番目の素子のバイアス

how the bias of the second unit of V_1 and V_2 is positive V_1 と V_2 の 2 番目の素子のバイアスは正

how the bias of the second unit of V_1 and V_2 is positive to compensate V_1 と V_2 の 2 番目の素子のバイアスは (相殺する—釣り合いをとる) ために正

how the bias of the second unit of V_1 and V_2 is positive to compensate for the fact that V_1 と V_2 の 2 番目の素子のバイアスは を (償う—相殺する) ために正

the code which represents that the unit has all the H unit turned off. =¿ the code represents that the unit has all the H unit turned off.

the code represents the unit コードはその素子を表現する

the code represents the unit has all the H unit コードは「その素子は全ての H 素子を持つ」ことを表現する

the code represents the unit has all the H unit turned off コードは「全ての H 素子がオフにされたことを持つ素子」を表現する

¡= the code which represents the unit その素子を表現するコードは

the code which represents the unit has all the H unit 「その素子は全ての H 素子を持つ」ことを表現するコードは

the code which represents the unit has all the H unit turned off 「全ての H 素子がオフにされたことを持つ素子」を表現するコードは

V_1 と V_2 の 2 番目の素子のバイアスを「1 」という事実 (を償う—を相殺する) ために正にする方法

(how) turned off. (how the bias of) is (positive to turn off). (how the bias of) is (positive to) turn off.

コードはそのユニットを表現するコードは「そのユニットは全ての H 素子を持つ」ことを表現するコードは「全ての H 素子がオフにされたものを持つそのユニットは」を表現する

は「素子 (?) が全ての H 素子を持つことを表現するコードは消された。」という事実について、(釣り合いをとる—補う) ための正

全ての H 素子を (持つ—有する) 素子を表わすコードをオフにするという事実について (釣り合いをとる—補う) ための (正—ポジティブ) は V_1 と V_2 の 2 番目の素子のバイアスをどのようにしてオフにするか V_1 と V_2 の素子の 2 番目のバイアスは、どのようにして正となる

1 The 4-2-4 Encoder

$v = 4$ と $h = 2$ としたときのネットワーク上での実験は以下の学習サイクルによって形成された:

1.p_{ij} の推定: 各環境ベクトルは順番に可視素子群の (上,over) へ固定される. 各環境ベクトルについて, そのネットワークは 2(回, 倍,twice) の釣り合い状態にいたることが (許可される,allowed).

素子群の対が (どのくらいの頻度,how often) で一緒になるかの (統計, 確率,statistics) は釣り合い (に, へ?,) 集められる。

大きく成長しすぎた重みを防ぐために, 私たちは節 3.2 に記述した (雑音,noisy) を固定する技術を用いた.

各固定されたベクトルの'on' ビットは 0.15 の確率で'off' になると定められており, そして, 各'off' ビットは 0.05 の確率で'on' になると定められている.

2.p'_{ij} の推定: そのネットワークは完全に固定されず, そして, 10 の温度の釣り合いへいたるために許される. p_{ij} を推定するために用いられた (だけの数, と同数の,) 焼きなましの数について, 共起性についての統計 (Statistics about co-occurrences) は, そのときに集められる.

3. 重みの更新: ネットワーク中の全ての重みは 2 の重みステップでの固定によって増減させられるが, $p_{ij} - p'_{ij}$ の符号によって決定する増加の符号となっているものと一緒に.

釣り合うための (沈殿, 沈下, 沈降,settling) が必要とされるとき, 固定されていない全ての素子は'on' か'off' の等しい確率と一緒に無作為抽出される.(無限大へ向かう温度に上昇)

そして, そのときネットワークは以下の時間と温度について [2@20, 2@15, 2@12, 4@10] を実行することが許される.

and then the network was allowed to run for the following times at the following temperatures:[2@20,2@15,2@12,4@10].

アニーリングスケジュールの後に, ネットワークが釣り合いに到達すると仮定すれば, 時間の 10 素子群について 10 の温度で (統計, 統計データ, 統計値,statistics) が集められる.

私たちは G の大域最小値について探索する 3 つの主要なフェーズを観測し, そして, それらのフェーズでの (事件, 発生, 存在, 出来事,occurence) は (比較, 相対,relatively) 的に (無神経に,insensitive) 使用されている (高価な, 貴重な, 重要な, 大事な,precise) パラメータを発見した.

最初のフェーズでは, 全ての重みにゼロが代入され, そして, ネットワークの大部分が (至るところに, の間中,) に

負の重みの (発展, 発達, 開発, 作成,development) によって特性が示され, 2 つの'(winner-take-all, 勝者が全てを得る)' 環境の構築物の側面で最もシンプルにモデル化されるネットワークが実行される.

-各可視グループ中の 1 つの素子だけが (普通は, いつもは, 正常に,normally)(1 度に, 同時に, 一回に,at a time) 振る舞う.

$4-2-4$ エンコードの中で, 例えば,

その可視素子群のもとで可能なパターンの数は 2^8 である.

各 4 つのグループのなかで,'winner-take-all' が実行されるネットワークにより, 4×4 の低エネルギーのパターンに減少することができる.

最終的に 2^4 から 2^2 へ (減少, 縮小,reduction) する低エネルギーパターンだけが 2 つの可視素子グループの間での (伝達,communicating) に用いられる隠れ素子を必要とする.

図 3a は,4 つの学習サイクル後の $4-2-4$ 符号化ネットワークを示している.

隠れ素子は最初のフェーズで (抑止, 抑制, 阻止, 阻害,inhibition) として使われるけれども,

(抑止, 抑制, 阻止, 阻害,inhibition) 処理の側面は可視グループの中での単独な結合によって (に触れる, を待つ, を動かす,) れることができる.

2 番目の側面は, 隠れ素子は可視グループ内の素子群のいくつかの正の重みを開発し始め, そして, それらは

その符号と V_1 の素子と V_2 に対応する素子の規模の近似

の対称性を維持する傾向にあり,

そして

その異なったコードの多くは使われ始め,

しかし

1 度かいくつか以上使われたいくつかのコードがある。

図 3b は 60 学習サイクル後の同じネットワークを示している。

時々、問題が解ける場合の 2 番目のフェーズの終わりで全てのコードが使われることになる。

(普段は, 通常, 大抵,) は, しかしながら, 3 番目と最も長いフェーズ の間中その学習アルゴリズムは残っている (対立—矛盾) (と—そして) フィールドの (大域最適解—最小値) を整理する。

2 つの基本的なメカニズムを整理する過程の間に含む。

図 3 で (矛盾—対立) は最初と 4 番目の素子の間に考えられるが、それにはコード $< -, + >$ が採用されている。

環境からの入力無しでシステムが実行されているとき、2 つの素子は結構頻繁に。

その結果として、$p'_{1,4}$ は $p_{1,4}$ よりも高くなるだろう。なぜならば、環境からの入力は (being on together) からの 2 つの素子を阻害する傾向にあるからだ。

だから、学習アルゴリズムは各グループでの最初と 4 番目の素子の間の結合の重みを減少を保つ。

(この影響は図 2 の抑制性の重みの (変化—変動) を説明する。よく似たコードを持つ可視素子は、それをお互いに強く抑制する。)

それ故に、

可視素子は可能なコードの空間内で'territory' について、完全に競合し、そして、この反発力の影響を 似ている隣のものから移動するための (原因) となる。

反発する影響に加えて、未使用のコードが隣接した (ハミング距離の項) (矛盾—衝突) を (引きおこされる—含まれる) コードへと最終的に運ばれる傾向にある別のプロセスを観測することができる。

このプロセスのメカニズムはいくらか微妙で、そして、私たちはここでそれらの話を膨らませる時間をとらない。

3 番目のフェーズは全てのコードが使われたときに終了し、そして、重みはそのときに解が固定されて、共起統計によるランダムな値によって引き起こる (変化—変動) に対して安定したのままでいられるように増加する傾向にある。(図 2 は図 3 に示したものと同じネットワークで、120 学習サイクル)

その $4-2-4$ エンコーダの 250 の異なるテストのなかで、常に最小値のひとつを発見し、そして、かつてのそこに、それがそこにとどまる。

4 つの異なるコードを発見するために必要なその平均期間は 110 学習サイクルだった。最長は 1810 学習時間だった。

4.2. The 4-3-4 Encoder 2 進符号化問題の派生は

H を与えるための

V_1 と V_2 のパターンの符号化について絶対に必要となる素子群である。

単純な例では $4-2-4$ エンコーダとみなして同じパラメータを与えて実行する $4-3-4$ エンコーダである。

この場合、学習アルゴリズムは高速に 4 つの異なるコードを発見する。

そのとき、それらが最適な間隔であり、ビットが 1 つだけ異なってペアがないようにコードを変化し続ける、図 4 のように。

4 つの (十分に—ゆったりとした) コードを発見するための平均時間は 270 学習サイクルで、200 試行のその最大時間は 1090。

The 8-3-8 Encoder

$v = 8$ と $h = 3$ にともなう、全ての 3 ビットの 8 つのコードを見つけるための学習サイクルの多くをとる。

私たちは 20 シミュレーションを行い、

$4-2-4$ の場合に関しては同じパラメータを用いる 4000 学習サイクルのそれぞれについて実行する。(しかし、ノイズを固定している間中は各 off 素子を 0.02 確率で反転させる)

そのアルゴリズムは 20 のシミュレーション中で 16 回、8 つ全てのコードを発見し、そして残りは 7 つのコードを発見した。

7 つのコードを発見するための平均時間は 210 学習サイクルで、そして 8 つの全コードを発見するための平均時間は 1570 サイクルだった。

8 つの全コードを発見することの難しさは

'4-2-4' の場合よりもずっと小さい解とみなして数えられる重み空間のほんの一部 (から—して以来—に続いて—のときから)

驚くべきことではない。

8 つ異なるコードのうち 7 つを使った重みのセットは正確に素早く発見され、そして、それらはグローバルな最小値よりも遥かに多い局所最適解と G の良い値 (として—みなす) ほとんどを持つ局所最適解によって構成されている。

このタイプの G 空間は

学習アルゴリズムは大域最小値に到達するまで注意深く微調整する必要があり、そして、それはとても遅い。

私たちは

適切なアルゴリズムについての G 空間はよりよい多くの可能な解決策とそのとても良い解をえるために本質的でない。

と信じている。

大きなネットワークについて学習するための (妥当—合理的) な時間は、十分な素子と重みと

だから、(no single unit) 単一素子や本質的でない重み。

その次の例は、いくつかの余分な能力を所有するものの発展を説明する。

4.4 The 40-10-40 Encoder

いくらか大きな例の'40-10-40' 符号化。

H 内の 10 素子は理論的な最小値のほぼ 2 倍であるが、H はまだボトルネックのバンド幅の制限としてまだ振る舞う。

この学習アルゴリズムは、この問題上でよく働く。

図 5 は V_1 内のパターンを与えたときの振舞いを示し、そして、V_2 内で一致するパターンと落ち着く。

各学習サイクルは各 40 の環境ベクトルを固定しながら、一回のアニーリングを (含む—取り込む—を伴う—引き起こす)、そして、固定無しでその同じ回数。

その最終的なパフォーマンスの漸近線は 98.6% を修正する。

V_1 と V_2 内のパターンを表現するために選択したネットワークのコードは少なくとも 2 のハミング距離によって全て離れており、そしてチャンスによって起こりそうにもない。

テストとして、可視素子と隠れ素子の間の結合の重みを比較した。

可視素子群の 2 つ差についての、その 10 次元の重みベクトルは同等とすべきでない。

2 つのベクトル間の角の余弦は類似度として使われ、そして、0.73 よりよい類似度を持つ 2 つのコードは存在しない、それに対して、多くのペアは 0.8 以上の類似度を持ち、

(するとき— なので— ならば— する場合は)

その同じ重みはランダムに再配置され、(対象—類似—比較) についての制御グループを提供する。

(完了—終了) テスト上の良いパフォーマンスに至るために、テストの間中、とても優しいアニーリングスケジュールが必要とされ使われる。

そのスケジュールは各温度 (に—で) 長さの 2 倍を (費やし—過ごし)、そして、スケジュールの最終温度の半分を下回る。学習の間中、使う

アニーリングはより速くされ、そのエラー率は増加し、それ故に、とても自然な速度が与えられ、正確さとのトレードオフ。

私たちはこの結果についてこれ以上議論しない。

しかし、実を結ぶだろう。

その速度/精度トレードオフの現在のモデルより良いいくつかの

なぜならば、いくつかの

そして、アニーリング探索は

数学を基本に置いた似ている