

Deep Multiagent Autoencoder による分散協調学習

Distributed Collaborative Learning with Deep Multiagent Autoencoder

黒滝 紘生 松尾 豊
Hiroki Kurotaki Yutaka Matsuo

東京大学工学系研究科技術経営戦略学専攻

The Department of Technology Management for Innovation, The University of Tokyo

We reconstructed Stacked Denoising Autoencoder, a novel Deep Learning model that achieved good performance on many tasks, with the methodology of multi-agent system. We focused on distributed and collaborative aspects of agents, and redesigned behavior of each neurons as autonomous agent. Each agent represents one feature. They have locally limited views of other agent's information. The layers are removed because they group several neurons and make them share some informations, resulting in less autonomous agents. Computations are done without explicit global manipulations. Agents automatically form feature network through the process of selections based on rewards as some neuroscientist claimed. Our contribution is to propose this new basal model that is more similar to human brain, can connect knowledge of Deep Learning and multi-agent system, and will be useful to make more flexible and multi-modal systems.

1. はじめに

我々は、ディープニューラルネットの一種である、Stacked Denoising Autoencoder(SDA)[Vincent 08]を、マルチエージェントシステムの原理に基づいて再構成した。このマルチエージェント SDA(MASDA)は、高い学習性能をもつ SDA に、マルチエージェントシステムの柔軟性を取り入れたモデルである。MASDA は、複数の知的システムをエージェントとして結合するための、基礎モデルとして期待できる。

MASDA は、特徴量を学習・提供するエージェントが多数集まった、分散協調システムである。MASDA では、良い特徴量を学習したエージェントに対し、報酬を与える。ここで特徴量の良さは、SDA にならい、データの分類ラベル・生データ・他の良い特徴量の予測に役だつかどうかで決められる。さらに、報酬を多く得たエージェントを選択的に残し、他を淘汰することで、良い特徴量をもつエージェント同士による予測ネットワークを得ることができる。この選択と淘汰というアイデアは、脳科学者ジェラルド・エーデルマンが提唱した、ニューラルダーウィニズムの考えに着想を得ている [Edelman 04]。この論文では、予測ネットワークを構成する第一歩として、Deep Learning の有力なアルゴリズムである SDA を、マルチエージェントシステムの視点で再構成する方法を示す。分散協調動作するユニットが、ローカルな情報に基づき、自律的に計算と情報交換を行うと、その振る舞いの集合として、SDA の計算が行えることを示す。

2. 全体のアーキテクチャ

我々のモデルでは、系全体は、環境およびユニットから構成される。それぞれのユニットが、マルチエージェントシステムという、自律したエージェントに相当する。環境においては、データセットにあわせた報酬が設定され、特定の条件を満たすユニットには報酬が与えられる。報酬を得られなかったユニットは確率的に死滅する。

各ユニットは、ひとつのニューロンを仮想的に表しており、

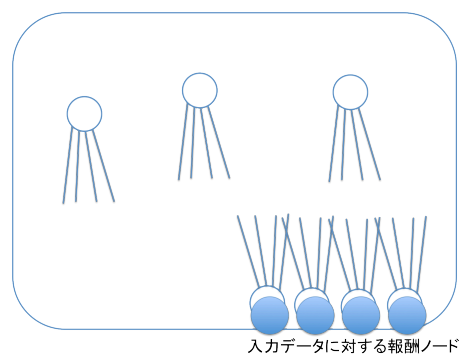


図 1: 初期状態

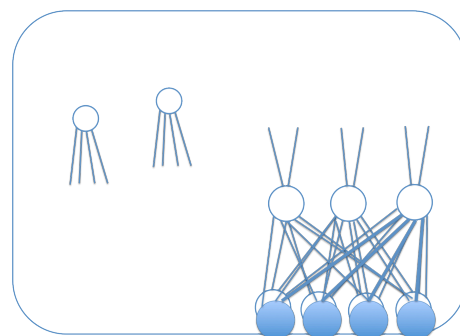


図 2: autoencoder を形成した状態

連絡先: 黒滝 紘生, 東京大学工学部, 〒113-0033 東京都文京区本郷 7-3-1 工学部 2 号館, kurotaki@weblab.t.u-tokyo.ac.jp

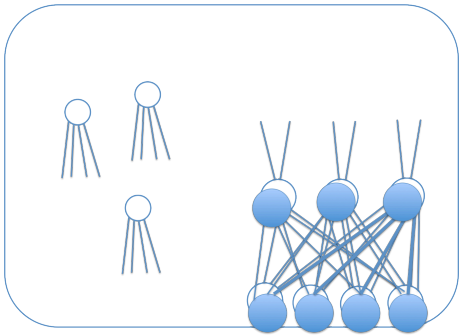


図 3: autoencoder を形成するユニットが状態 B に変化した状態

軸索および軸索末端に相当するエッジと、細胞体に相当するノードから構成される。??に示すように、予測時には、ノードからエッジを通じて情報が流れ、次のノードに情報が伝わる。逆に学習時には、エッジからノードに情報が流れ、次のエッジに伝わる。

2.1 環境の設定の概要

環境中には、入力データが送り込まれる。入力データは、ある特定のノードが入力データのある要素の値を取るという形で実装される。(なお、必ずしも入力データとノードが1対1対応する必要はないが、ここでは簡単に1対1対応すると考える。)

2.2 ユニットの設定の概要

ユニットには、状態 A(学習中) と状態 B(安定) の2種類があると考える(??)。状態 A は、学習を行っている段階であり、状態 B は学習を終えて、挙動が安定した段階である。したがって、状態 A のときには、エッジの重みを変更される。状態 B のときには、エッジの状態は変更されない。また、そのノードの値自体を予測させるようなノードがさらに付与される。このノードを報酬ノードと呼ぶことにする。報酬ノードが付与されるということは、このノード自身が予測対象になり、予測に成功すると報酬が与えられることを示している。

2.3 全体の挙動のイメージ

本研究の目的は、マルチエージェントの枠組みにより深層学習の挙動を実現することである。したがって、まず、環境とユニットを含む全体の挙動がいかにして深層学習の動作と同じになるのかを説明する。この説明はやや恣意的になるが、これをできるだけ自然な形で実装することを以下の章では試みる。

初期状態を図 1 に示す。入力データのそれぞれの要素に対して状態 B のユニットが対応しており、その値をエッジで伝えている。また、それぞれのユニットに報酬ノードが付与されている。その他に、環境中にいくつかのユニットが浮遊している状態である。

浮遊するユニットが入力ノードに一部結合して、この結合が強固になり、autoencoder を形成した状態が図 2 である。入力ノードからユニットを介して、報酬ノードに情報が流れ、報酬がその逆の経路で返るといった形になっている。

この状態で、ほぼ挙動が安定化するので、ユニットは状態 A から状態 B に変化する(図 3)。そうすると、このユニット自身を予測することに対しても報酬が与えられるようになる。また、ユニットに伸びるエッジの重みは基本的に安定し、変化しない状態になる。

その後、さらに高次の autoencoder が形成される可能性もあり得る。

*1。

以上が、マルチエージェントの枠組みでユニットという単位の挙動により、Deep Learning を実現する全体像である。

2.4 本モデルにおける仮定

本モデルは、Deep Learning の挙動をできるだけ自然な形でマルチエージェントの枠組みで実現しようとするものである。いくつかの仮定を置かなければならない。その仮定が自然で妥当であるほど良いモデルである。

本モデルで置いている仮定は以下である。

- 仮定 1: 入力データや教師データは環境に入力される外部データという意味では同じであり、いずれも予測が成功するとそれに寄与したユニットに報酬が与えられる。
- 仮定 2: ユニットの挙動が安定して、エッジの重みの変化がほぼなくなると、固定化される。と同時に、そのユニット自身を予測することで報酬が与えられるようになる。(状態 A から状態 B に変化する。)
- 仮定 3: 重み(およびユニット固有のパラメータ)の変化は、報酬ノードとの予測誤差の微分が伝搬する形で行われる。

仮定 1 に関しては、Deep Learning では最初に入力データだけで pretraining と呼ばれる処理を行い、次に教師データによって fine tuning と呼ばれる処理を行う。ただ、脳の中の神経回路にとって、こうした複数のデータに区別はないはずであり、いずれに対しても予測が成功すると報酬が与えられるというのは妥当であると考えられる。仮定 2 に関しては、離散的にある時点で固定化し状態が変わるのか、それとも徐々に変化が少なくなり、それとともに実質的に報酬が安定的に与えられるようになるという連続的な変化かについては議論の余地がある。ただし、現状の Deep Learning の実装でも、pretraining を層ごとに行っており離散的であること、またアルゴリズム上も離散的变化と考えた方がシンプルであるため、以下ではこの仮定を採用する。仮定 3 に関しては、誤差逆伝搬を仮定することであり、これまでの研究でも多く用いられてきた仮定である。神経学的な証拠もあるため、この仮定を採用することは妥当であると考えられる。

基本的には上記の 3 つの仮定で Deep Learning の挙動を実現しており、十分に自然で妥当な仮定であると考えている。

3. アルゴリズム

この章では、Stacked Denoising Autoencoder を、マルチエージェントシステムの考え方を基に再構成する。

以下では、ユニットの内部構造、接続するエッジから値を受け取ったときの計算方法(予測)、報酬ノードに接続したときのアクション(学習)、エッジの伸ばし方、状態 A から状態 B になるタイミング(termination criteria)について述べる。最後に、全体のアルゴリズムについて述べる。

*1 なお、必ずしも 2 段階の結合による autoencoder が形成される必要はない。3 段階以上のものでもよいはずである。また、1 段階の結合(すなわちユニットから出されるエッジが直接、報酬ノードにつながる)は、エッジの距離により回避されていると思われ、本モデルでもそれは禁止する。

3.1 ユニットの種類

ユニットは、一つのノードと多数のエッジで出来ている。ユニットには、変数ユニット、定数ユニット、報酬ユニットの3種類がある。変数ユニットは、学習される中間特徴量に対応する。3種のうち変数ユニットのみが、状態変化を起こし、また選択淘汰の対象になる。定数ユニットは、データセットやバイアス値など、システム外部から与えられる固定値に対応する。システムで言えばセンサの役割を果たす。始めから、変数ユニットでいう状態 B (安定)と同様に報酬ユニットを持つ。報酬ユニットは、予測値に対応する。変数ユニットから情報を受け取って値の予測を行い、予測が近いほど、多くの報酬を返す。報酬ユニットは、予測先ユニットを一つだけ持てる。

3.2 ユニットがもつ情報

ユニットは、自分が持っている情報に基づいて、自律的に行動を行う。この情報には、自分自身のことや、自分の接続先ノードのことが含まれる。情報のリストは、表1に示す通りである。

この他、データの処理中は、通常のニューラルネットと同じく、ノードの値を表す変数 v などがユニット毎に割り当てられる(後述)。自分に対応する `reward_node`, `target_node` の v は参照できるものとする。

3.3 メッセージパッシングと予測・最適化計算

ユニットは、メッセージを通じて、他のユニットと情報を受け渡しできる。これは、よりよい予測や学習のために使われる。順方向と逆方向の2種類のメッセージがあり、それぞれ `next_nodes`, `prev_nodes` に登録されたユニットにのみ送ることができる。各ユニットは、自分に送られてきたメッセージを全て記憶できる。また、メッセージで得た情報を使い、SDAと等価な計算を行うことができる。なお、メッセージの内容は、そのとき入力中のデータについてのみに有効である。データが変わった場合は、無効になり破棄される。ここではまず、必要なメッセージ経路の情報が全て揃っているとき、どのようにして計算が出来るかを示す。具体的な、各ユニットのメッセージ応対動作や、ネットワーク形成の方法は、後述する。

3.3.1 ユニットの予測とメッセージ

各ノードによる予測値や特徴量は、順方向メッセージの情報を使って計算される。ここで、順方向メッセージは、

- 送信元ノードの識別子 (ID, 名前)
- 予測用情報: 送信元ノードの v 、またはゼロ

で構成される。SDAと同じく、予測用情報は、送信元ノードの `corruption_rate` の確率で隠蔽され、ゼロになることにする。これらを使い、変数・報酬ノードの予測値 v は、

$$a = \sum_{n \in \text{prev_nodes}} w_n * v_n$$

$$v = \sigma(a)$$

によって更新される。ただし、

- σ はシグモイド関数
- w_n は、ノード n に対応するエッジの重み
- v_n は、ノード n より順メッセージで送られてきた v (またはゼロ)

とする。なお、定数ノードの v には、対応するデータやバイアス値が、環境より直接与えられる。

3.3.2 ユニットの学習とメッセージ

報酬ノードの学習は、順方向メッセージの情報のみで行える。 η を各ノードの学習率とすると、ノード n への辺の重みの修正値 Δw_n は、

$$\delta = v - v_{\text{target_node}}$$

$$\Delta w_n = -\eta \delta v_n$$

と計算される。また、変数ノードの学習は、逆方向メッセージの情報を使って計算される。逆方向メッセージは、

- 送信元ノードの識別子 (ID, 名前)
- 修正情報: $w * \delta$
- 報酬値

で構成される。これらを使って、変数ノードでは、

$$\delta = \sigma'(a) \sum_{n \in \text{next_nodes}} w_n \delta_n$$

$$\Delta w_n = -\eta \delta v_n$$

と学習される。なお、実際の重み更新は、全ての訓練データを処理した、1エポックの終わりにまとめて行われる。

3.4 各ユニットの振る舞い

ユニットがメッセージやデータを受信したとき、自分に見えている情報のみを元にして、どのように動作するかを記す。「受信したメッセージを保存」は共通動作のため、省略する。また、今回のモデルでは、「定数ノードが順メッセージを受信」「報酬ノードが逆メッセージを受信」の2パターンは起こらないため、省略する。

3.4.1 定数ノードの振る舞い

- 今回のモデルでは、メッセージを受信することはない。
- システム外部からデータを入力された「自分の v に代入」「全ての `next_nodes` に順メッセージ送信」を行う。

3.4.2 変数ノードの振る舞い

- 順メッセージを受信 全ての `prev_nodes` からの順メッセージが揃っていない場合、何もしない。揃ったなら、「自分の v を計算」「全ての `next_nodes` に順メッセージ送信」を行う。
- 逆メッセージを受信 全ての `next_nodes` からの逆メッセージが揃っていない場合、何もしない。揃ったら、「自分の Δw_n を計算」を行う。また、自分の `score` を更新する。

3.4.3 報酬ノードの振る舞い

- 順メッセージを受信 全ての `prev_nodes` からの順メッセージが揃っていない場合、何もしない。揃ったなら、「自分の v , Δw_n を計算」「全ての `prev_nodes` に順メッセージ送信」を行う。
- 今回のモデルで、報酬ノードが逆メッセージを受信することはない。

3.5 全体のアルゴリズム

ここまでの記述を踏まえて、MASDAによる学習アルゴリズムの全体を記す。

表 1: 各ユニットが持つ情報

定数	変数	報酬	名前	説明
○	○	×	next_nodes	ノードの集合。自分が送った順方向メッセージを、受け取る。 また、自分に向けて、逆方向メッセージを送ってくる。
×	○	○	prev_nodes	ノードの集合。自分に向けて、順方向メッセージを送ってくる。 また、自分が送った逆方向メッセージを、受け取る。
×	○	○	prev_weights	prev_nodes 内のノード 1 つに対し 1 つ割り当てられる数値。エッジの重み。 受けた順方向メッセージを、どう扱うか。
×	○	○	stable	prev_weights を変化させるか。状態 A(学習中) か、状態 B(安定) か。
○	○	○	depth	自分がどの抽象度の変数に相当するかを表す数値。大きいほど抽象度が高い(後述)。
×	○	×	score	自分が得た報酬。学習を繰り返す中で蓄積される数値。 これが一定値を超えたとき、状態 A から B に移行する。
○	○	×	reward_node	自分を予測している報酬ノード。
×	×	○	target_node	予測先の定数・状態 B(安定) 変数ノード。
×	○	○	learning_rate	学習率。エッジの重み変化を調整する係数。
○	○	×	corruption_rate	破損率。出力値を強制ゼロにする確率。

Algorithm 1 pretraining

入力データ、教師データの各次元に対応するデータユニットを、depth=0 で生成する。

対応する報酬ユニットを、depth=0 で生成する。

あらかじめ決めた個数の初期変数ユニットを生成する。

while pretraining 終了条件を満たさない **do**

while まだ接続されていない変数ユニットがある **do**

 未接続の変数ユニットを、既存の報酬ユニットのどれか一つに接続する。

 接続先ユニットと同じ depth の、全ての安定ユニットと報酬ユニットにも接続する。

 depth を、接続先ユニットの depth(共通)+1 に設定する。

end while

while 未入力の変数ユニットが残っている **do**

 ランダムに、未入力の変数ユニットを選び、対応するデータを入力

 (残りの計算は、ユニットにより、外部からの介入無しで行われる)

end while

状態 A 変数ユニットの辺の重みを更新する。

状態 A 変数ユニットの、状態 B への変化判定を行う。

状態 A 変数ユニットの score が低い場合、そのユニットはネットワークから外される(淘汰)。

淘汰されたユニットの個数と同じだけ、新しいユニットを生成する。

end while

3.5.1 pretraining

pretraining 段階は、以下のように進められる。

3.5.2 fine-tuning

fine-tuning 段階では、通常の SDA と同様に、pretraining 段階で得られたノードとエッジを抽出し、通常のニューラルネットワークと見なして学習を行う。再構成の報酬ユニットは、この段階では無視される。通常のニューラルネットワークが形成されるので、一般的な誤差逆伝播によってパラメータ最適化を行うことができる。

4. おわりに

この論文では、マルチエージェントシステムの知見を Deep Learning に取り入れる第一歩として、SDA をマルチエージェントシステムの観点から再構成した。これにより、分散協調動作するユニットの自律的な振る舞いを集めることで、SDA の計算が行えることを示した。今後の課題としては、このシステムを実際のデータセットに適用して、既存の SDA や機械学習の手法と、学習性能の比較を行うことが重要である。また、システムを拡張して、分散自律性を生かした学習モデルを構築することも挙げられる。

参考文献

[Vincent 08] Vincent, P. Larochelle, H. Bengio, Y. Manzagol, P: Extracting and composing robust features with denoising autoencoders (2008). In Proceedings of the 25th International Conference on Machine Learning. ICML '08. New York, NY, USA: ACM, pp. 1096-1103.

[Edelman 04] Edelman, G. M.: Wider than the sky: The phenomenal gift of consciousness (2004). (邦訳: 脳は空より広いか 「私」という現象を考える, 豊嶋 良一 監修, 冬樹 純子 訳, 草思社 (2006))