—論文題目—

IL-RBMの強化学習タスクへの応用

英語のタイトル

指導教授

萩原 将文 教授

学習指導副主任

遠山 元道 准教授

慶應義塾大学 理工学部 情報工学科

平成　27年度

学籍番号 61212346

筒井佑一郎

目 次

あらまし

fffffffffffffffffffffffffffffffffffffffffffff こ

第 1章

RBM, IL-RBM

本章では、提案システムを構成する主要なネットワークである Restricted Boltzmann Bachine(RBM) と IL-RBM について説明する。

1.1 Restricted Boltzmann Machine

1.1.1 Restricted Boltzmann Machine の概要

Restricted Boltzmann Machine(RBM) とは、1982 年、J.J. Hopfield が発表した [] ホップフィールドネットワークの一種である Boltzmann Machine の可視層間、隠れ層間の結合を制限したものである。

Boltzmann Machine は、連想記憶のモデルとして有力なニューラルネットワークであったが、学習時間が発散してしまうため、問題のサイズがある程度以上になると現実的に学習が不可能であるという欠点が存在した。

RBM は Boltzmann Machine の可視層間、隠れ層間の各ノードの結合を制限することで、学習時間を現実的な値まで減らしたモデルであり、後述する Contrastive Divergence 法と呼ばれる洗練された学習方法の登場もあり、現在様々な分野で広く利用されているニューラルネットワークの一つである。

図 1.1 に示すように、RBM は可視層と隠れ層の二層で構成される。基本的な RBM では、各ノードは 0 か 1 の値を取る。

可視層と隠れ層のノードの値を示すベクトルを以下のように定義する。

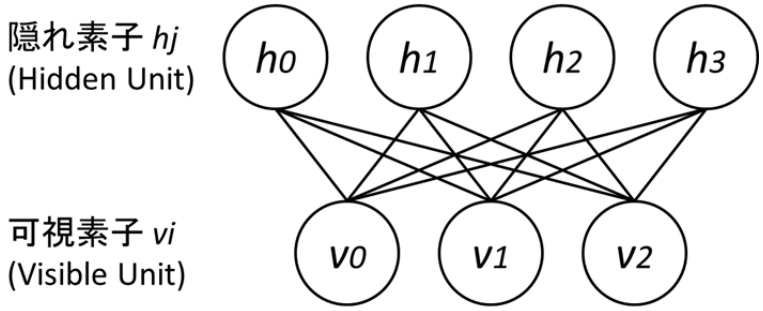


図 1.1 Restricted Boltzmann Machine

$$\boldsymbol{v} = (v_0, v_1, \ldots v_{V-1}), \forall v_i \in 0, 1 \tag{1.1}$$

$$\boldsymbol{h} = (h_0, h_1, \ldots h_{H-1}), \forall h_i \in 0, 1 \tag{1.2}$$

この時、RBM では可視層と隠れ層の結合確率が以下のように定義される。

$$(\boldsymbol{v}, \boldsymbol{h}; \theta) = \frac{1}{Z(\theta)} \exp(-E(\boldsymbol{v}, \boldsymbol{h}; \theta)) \tag{1.3}$$

$$E(\boldsymbol{v}, \boldsymbol{h}; \theta) = -\sum_i b_i v_i - \sum_j c_j h_j - \sum_i \sum_j v_i W_{ij} h_j \tag{1.4}$$

ここで b_i, c_j, W_{ij} はそれぞれ可視素子 v_i のバイアス，隠れ素子 h_j のバイアス，可視素子 v_i と隠れ素子 h_j の間のウェイトパラメータである．これらのパラメータをまとめて θ とする．また Z は正規化定数であり，以下のように定義される．

$$Z(\theta) = \sum_{\boldsymbol{v}} \sum_{\boldsymbol{h}} \exp(-E(\boldsymbol{v}, \boldsymbol{h}; \theta)) \tag{1.5}$$

式 1.3、式 1.4 より、片方の層の状態が入力された時、他方の層の各ノードの取る値の条件付き確率が計算できる。そのため、ある隠れ層の状態から入力を確率的に生成することが可能なため、RBM は生成モデルであるという側面を持つ。

RBM は、可視層に入力データを入れ学習させることで、隠れ層にその特徴をよく表すようなパラメータを出力することができる。この特徴を用いて、後述する Deep

Belief Net のように、ディープニューラルネットワークのプレトレーニングに用いられることも多い。

1.1.2 RBM の学習

1.1.2.1 計算の概略

RBM の学習では、可視層の観測データ $\boldsymbol{v}$ に対する $p(\boldsymbol{v})$ について最尤推定を行う。$p(\boldsymbol{v})$ を計算するため結合確率 $p(\boldsymbol{v}, \boldsymbol{h})$ を $\boldsymbol{h}$ について周辺化する。

$$
\begin{aligned}
p(\boldsymbol{v};\theta) &= \sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h};\theta) && (1.6)\\
&= \sum_{\boldsymbol{h}} \frac{1}{Z(\theta)} \exp(-E(\boldsymbol{v},\boldsymbol{h};\theta)) && (1.7)
\end{aligned}
$$

計算の簡単のため、以下では尤度の対数をとった対数尤度を取り扱う。

$$
\begin{aligned}
J &= < \ln p(\boldsymbol{v};\theta) >_q && (1.8)\\
&= < \ln \sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h};\theta) >_q && (1.9)\\
&= < \ln \sum_{\boldsymbol{h}} \exp(-E(\boldsymbol{v},\boldsymbol{h};\theta) >_q - \ln Z(\theta) && (1.10)\\
\theta^* &= \operatorname*{argmax}_{\theta} J && (1.11)
\end{aligned}
$$

ここで $<\cdot>_q$ は確率分布 $q(\boldsymbol{v})$ の期待値を表す.

$$
< f(\boldsymbol{v}) >_q = \sum_{\boldsymbol{v}} f(\boldsymbol{v}) q(\boldsymbol{v}) \tag{1.12}
$$

また，$q(\boldsymbol{v})$ は観測データに関しての確率分布である.

$$
q(\boldsymbol{v}) = \frac{1}{N_{data}} \sum_{n} \delta(\boldsymbol{v} - \boldsymbol{v}^n) \tag{1.13}
$$

ここで $\boldsymbol{v}^k$ は k 番目の観測データを，$\delta(x)$ は以下のように定義される関数である.

$$\delta(x) = \begin{cases} 1 & (n = 0) \\ 0 & (otherwise) \end{cases} \tag{1.14}$$

ここで，対数尤度をパラメータ θ について微分する．

$$\begin{aligned} \frac{\partial J(\theta)}{\partial \theta} &= - < \frac{1}{\sum_{\boldsymbol{h}} \exp(-E(\boldsymbol{v}, \boldsymbol{h}; \theta))} \sum_{\boldsymbol{h}} \frac{\partial E(\boldsymbol{v}, \boldsymbol{h}; \theta)}{\partial \theta} \exp(-E(\boldsymbol{v}, \boldsymbol{h}; \theta)) >_q \\ &\quad + \frac{1}{Z(\theta)} \sum_{\boldsymbol{v}} \sum_{\boldsymbol{h}} \frac{\partial E(\boldsymbol{v}, \boldsymbol{h}; \theta)}{\partial \theta} \exp(-E(\boldsymbol{v}, \boldsymbol{h}; \theta)) \qquad (1.15) \\ &= - \sum_{\boldsymbol{v}} \sum_{\boldsymbol{h}} \frac{\partial E(\boldsymbol{v}, \boldsymbol{h}; \theta)}{\partial \theta} \frac{\exp(-E(\boldsymbol{v}, \boldsymbol{h}; \theta)}{\sum_{\boldsymbol{h}} \exp(-E(\boldsymbol{v}, \boldsymbol{h}; \theta))} q(v) \\ &\quad + \frac{1}{Z(\theta)} \sum_{\boldsymbol{v}} \sum_{\boldsymbol{h}} \frac{\partial E(\boldsymbol{v}, \boldsymbol{h}; \theta)}{\partial \theta} \exp(-E(\boldsymbol{v}, \boldsymbol{h}; \theta)) \qquad (1.16) \\ &= < \frac{\partial E(\boldsymbol{v}, \boldsymbol{h}; \theta)}{\partial \theta} >_{data} - < \frac{\partial E(\boldsymbol{v}, \boldsymbol{h}; \theta)}{\partial \theta} >_{model} \qquad (1.17) \end{aligned}$$

ここで、$<>_{data}$、$<>_{model}$ はそれぞれ $p_{data}(\boldsymbol{v}, \boldsymbol{h}) = p(\boldsymbol{h}|\boldsymbol{v})q(\boldsymbol{v})$ と $p_{model}(\boldsymbol{v}, \boldsymbol{h}) = p(\boldsymbol{v}, \boldsymbol{h})$ に対する期待値を表す。

ここで、第一項に関しては、可視層の状態から隠れ層の条件付き確率の厳密解が用意に計算できることから、計算はかなり容易である。一方、第二項に関しては、全ての $\boldsymbol{v}$,$\boldsymbol{h}$ の組み合わせを計算しなければいけないため計算量が指数的に爆発してしまう。

したがって第二項を計算するため、 サンプリング的手法や、近似的に解を求める手法などが用いられてきた。現在では次節で説明する Contrastive Divergence 法と呼ばれるサンプリング手法が最も有力であり、広く用いられている。

1.1.2.2 Contrastive Divergence 法

Contrastive Divergence 法 (CD 法) は、2002 年に Hinton によって発表された RBM の学習におけるサンプリング手法である。

厳密な $p(\boldsymbol{v}, \boldsymbol{h})$ を計算する代わりに、$\boldsymbol{v}$ と $\boldsymbol{h}$ をサンプリングして、$p(\boldsymbol{v}, \boldsymbol{h})$ を近似的に求める手法であるが、従来のギブスサンプリングとは遷移回数と可視層の初期値の選び方の点で異なる。

CD 法では、$\boldsymbol{v}$ と $\boldsymbol{h}$ をサンプリングする際の可視層の初期値に、実際の観測データを用いる。RBM では、各層のノードの取る値の条件付き確率は、他方の層のノードの状態にのみ依存しており、その確率はロジスティック関数 $\sigma(x) = \frac{1}{1+\exp(-x)}$ を用いて

$$p(v_i = 1 \mid \boldsymbol{h}) = \sigma(b_i + \sum_j W_{ij} h_j) \tag{1.18}$$

$$p(h_i = 1 \mid \boldsymbol{v}) = \sigma(c_i + \sum_i v_i W_{ij}) \tag{1.19}$$

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{1.20}$$

で表されるので、入力された可視層から隠れ層の状態を求めることができる。その後、同じように求めた隠れ層の状態から可視層の状態を計算可能である。

ギブスサンプリングでは、この可視層と隠れ層の間の遷移を通常多数回行う必要があったが、CD 方では少ない回数でも上手く学習が進むことが経験的に知られており、多くの場合では遷移回数が 1 回でも十分である。

このようにして得られた可視層の状態と条件付き確率より結合確率

$$p_{model}(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{N} \sum_k \delta(\boldsymbol{v} - \boldsymbol{v}_k) p(\boldsymbol{h} | \boldsymbol{v}_k) \tag{1.21}$$

を求め、近似的に式 1.17 の第二項を求める。

1.2 Deep Belief Net

ディープニューラルネットワークを教師あり学習させようとした場合、過学習や局値解に陥ってしまうと言った問題が起きることがある。このような問題を解決するためにプレトレーニングが行われることがあるが、プレトレーニングに前述した RBM の学習を用いたものを Deep Belief Net(DBN)[?] という.

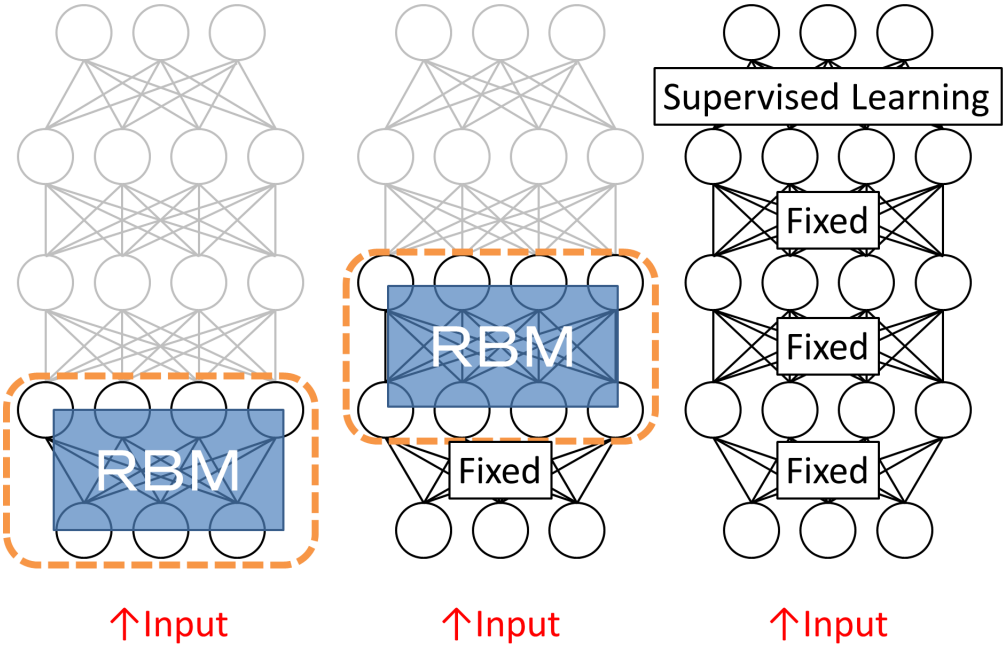


図 1.2 DBN 学習過程

図 1.2 にプレトレーニング [?] の手順を示す.

まず、ディープニューラルネットワークの第一層と第二層を RBM とみなし RBM の学習を行う。可視層に入力した入力データの特徴の表現が隠れ層に表れるという RBM の利点を、ディープニューラルネットワークのプレトレーニングに活用する。

第一層と第二層にて学習を終えた後、この RBM の重みを固定し、入力データを伝播させて表れた隠れ層の状態を新たな入力データとして、第二層と第三層を新たな RBM とみなし学習を行う。

このように連鎖的に下層から上層に向かい二層ずつ RBM とみなして教師なし学習を行い、出力層では教師あり学習を行う。

このようにプレトレーニングを終えた後、ネットワーク全体で誤差逆伝播法にて教師あり学習を行う手法もしばしば用いられ、これをファインチューニングと呼ぶ。

1.3 Incremental Learning-RBM(IL-RBM)

提案システムの IL-RBM は追加学習を行うと既学習情報を失うという RBM の欠点を改良したものであり、追加学習を行う際に隠れ層の素子数を追加するという特徴がある。追加する素子数を学習データセットを考慮し自動で決定するアルゴリズムと、ネットワークのエネルギーより既学習データと未学習データを判別するアルゴリズムも同時に説明する。

1.3.1 隠れ層ノード数の自動決定法

RBM の隠れ層のノード数の自動決定を行うアルゴリズム [] について説明する。文献 [] において、図 1.3[1] に示すように、学習済み RBM のクロスエントロピーと隠れ層ノード数には以下の関係が有ることが述べられている。

- 隠れ層のノード数が非常に少ない場合、ノード数にかかわらずクロスエントロピーが一定である領域が存在する場合がある。
- 隠れ層のノード数が不十分な場合、隠れ層のノード数を増やした際に学習済み RBM のクロスエントロピーは一般に減少し、その減少は線形に近似することができる。
- 隠れ層ノード数が十分な値に達した場合、隠れ層のノード数を増やしても学習済み RBM のクロスエントロピーは減少しない。

この関係を用いて、文献 [] にて RBM の素子数の自動決定法が提案されている。

RBM の素子数の自動決定法は傾斜検出フェーズと傾斜予測フェーズの 2 つのフェーズに分けることができる。

1.3.1.1 傾斜検出フェーズ

傾斜検出フェーズは、ニューロン数にかかわらずクロスエントロピーの変化しない領域をスキップし、傾斜の始まりを正しく検出するためのフェーズである．図 1.4 に

[1] テキスト

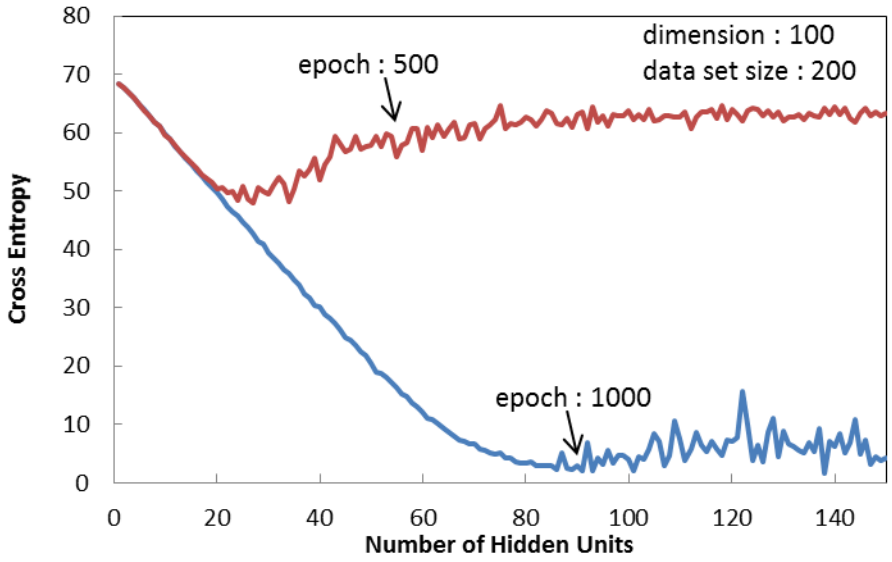


図 1.3 隠れ層ノード数とクロスエントロピーの関係

傾斜検出フェーズの実行過程を示す.

ニューロン数が 1 の時のクロスエントロピーを E_{init} と置く。その後、ノード数を増加させつつサンプリングしたクロスエントロピーと E_{init} との差がある閾値を超えた場合にクロスエントロピーの減少が始まったと判断し、その点を傾斜の開始とみなす。

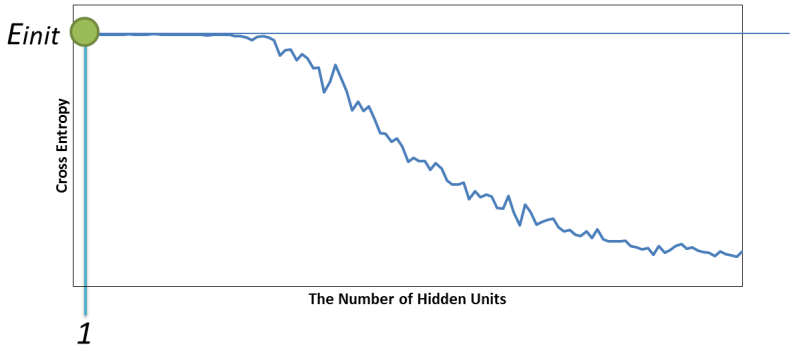
傾斜の開始を検知した後、ノード数をいくつかサンプリングしてクロスエントロピーを求める。サンプリングした点から傾斜の傾きを計算し、クロスエントロピーの減少を線形で近似する。

最終的な隠れ層のノード数は、クロスエントロピーの近似式の値が 0 となる時のノード数が選ばれる。

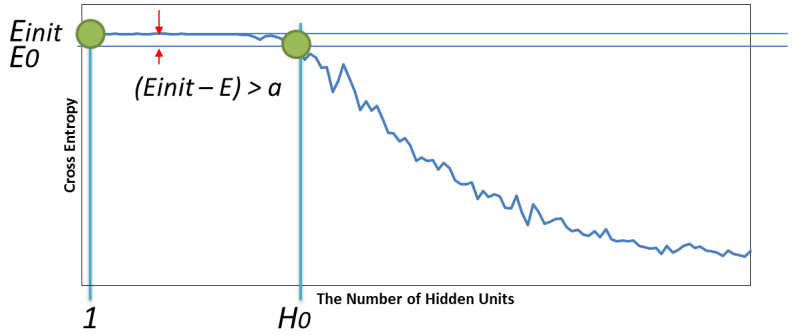
1.3.2 追加学習

II-RBM では追加学習を行う。追加学習とは、既にあるデータセットに対し学習済みのニューラルネットワークが、新たなデータセットに対し、既学習情報を失わない形で学習を行うことである。

通常の RBM の隠れ層のノード数は一定である。しかし、IL-RBM は新たなデータ



(a) E_{init} の決定



(b) 閾値 a との比較

図 1.4 傾斜検出フェーズの実行過程

セットを学習する際に隠れ層のノード数を追加する操作を行う。IL-RBM の学習は次のような流れをとる。

1. 初期データセットに対し IL-RBM を学習させる。
2. 追加データセットに対し、適切な追加ノード数を決定し、Il-RBM の隠れ層にノードを追加する。
3. 上記手順で追加したノードのみを用いて、追加データセットを学習させる。
4. 上記 2、3 の手順を追加データセットの分だけ繰り返し行う。

追加データセットに対する適切な追加ノード数の決定には 1.3.1 節で説明したノード数の自動決定法を用いる。

1.3.3 システム全体の流れ

本システムは、強化学習の基本的な考え方に則り、”環境”と”エージェント”が”行動”と”報酬”により相互に影響を及ぼし合うように構成されている。

システムの流れは以下のとおりである。

1. 二つのエージェントを定義する。
2. 先行のエージェントに盤面の状態を与える。
3. 先行のエージェントは盤面の状態から行動を選択する。
4. 環境は、エージェントの行動を受け取り、自身の状態を更新する。
5. 環境の状態に応じて、エージェントに報酬を与える。
6. 環境の状態が終了条件を満たせばゲームを終了する。
7. 上記手順を先攻、後攻を交代し繰り返す。

環境部分は、常にある状態をを持ち、その状態はエージェントの行動によって変化させられる。また、環境部はその状態とエージェントの行動によってエージェントに与える報酬を決定する。

1.4 Incremental Learning-RBM(IL-RBM)

この節では、本システムの根幹となるIL-RBMについて説明する。IL-RBMは追加学習を行うと既学習情報を失うというRBMの欠点を改良したものである。

1.4.1 追加学習

IL-RBMでは追加学習を行う。追加学習とは、既にあるデータセットに対し学習済みのニューラルネットワークが、新たなデータセットに対し、既学習情報を失わない形で学習を行うことである。

通常のRBMの隠れ層のノード数は一定である。しかし、IL-RBMは新たなデータセットを学習する際に隠れ層のノード数を追加する操作を行う。IL-RBMの学習は次のような流れをとる。

1. 初期データセットに対しIL-RBMを学習させる。
2. 追加データセットに対し、適切な追加ノード数を決定し、Il-RBMの隠れ層にノードを追加する。
3. 上記手順で追加したノードのみを用いて、追加データセットを学習させる。
4. 上記2、3の手順を追加データセットの分だけ繰り返し行う。

1.4.2 未学習データセットの判別

エージェントに与えられたあるデータを既学習であるか、未学習であるかを判別する手法について述べる。エージェントに与えられたデータの既学習判定にRBMのエネルギーを用いる。RBMは学習済みのデータに対して、エネルギーが低くなる、という特徴を持つため、データを入力した際のRBMのエネルギーを見ることで入力されたデータが既学習であるか、未学習であるかを判別することができる。

第 2章

IL-RBMの強化学習への応用

この章は??にて説明したIL-RBMの, 本システムにおける強化学習への応用について説明する。

2.0.1 強化学習

強化学習とは、あるエージェントがある環境内にて、得られる報酬を最大化するような行動を学習するような機械学習のことである。

強化学習には重要な4つの概念がある。

- 環境 … エージェントの行動に応じて、報酬をエージェントに与える。また、エージェントの行動に応じて、エージェントの観測する環境も更新される。
- 報酬 … エージェントの行動に応じて環境からエージェントに与えられる。この得られる報酬を最大化するようエージェントは行動を学習する。
- 行動 … エージェントは観測した環境に応じて、行動を選択する。
- エージェント … 環境を観測し、報酬を受け取り、行動を選択する。

明確な教師データが与えられる教師あり学習や、全く教師データが与えられない教師なし学習と異なり、報酬という限定されたフィードバックのみが与えられる点に特徴がある。不確実な環境を取り扱えるという点で、応用上非常に有望な機械学習手法の一つである。

2.0.2 エージェントの概要

エージェントの概要について具体的に説明する。

本システムで用いたエージェントは、IL-RBM と出力層からなる DBN(以下、IL-DBN) を用いている。この IL-DBN は入力データを環境、出力を行動として学習する。エージェントは、常に今自分が置かれている環境を観測することが可能であり、環境から報酬を与えられた場合には、それを感知することが可能である。

エージェントは環境と行動の組からなるデータセットを自動的に構築し、そのデータセットを逐次的に学習することで、最適行動を学習する。また、IL-RBM のエネルギーに注目することで、未学習データセットを検出できるという特徴を用い、後述するサブゴールを獲得し、長期的な戦略を獲得することが可能である。

2.0.3 未学習データ判定法

1.1.2 で説明したとおり、RBM は以下に J で表される対数尤度を最大化するように学習が行われる。すなわち、エネルギー E を最小化するように学習が行われることと同値である。

$$
\begin{aligned}
p(\boldsymbol{v},\boldsymbol{h};\theta) &= \frac{1}{Z(\theta)}\exp(-E(\boldsymbol{v},\boldsymbol{h};\theta)) \\
E(\boldsymbol{v},\boldsymbol{h};\theta) &= -\sum_i b_i v_i - \sum_j c_j h_j - \sum_i \sum_j v_i W_{ij} h_j \\
p(\boldsymbol{v};\theta) &= \sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h};\theta) \\
&= \sum_{\boldsymbol{h}} \frac{1}{Z(\theta)}\exp(-E(\boldsymbol{v},\boldsymbol{h};\theta)) \\
J &= < \ln \sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h};\theta) >_q \\
&= < \ln \sum_{\boldsymbol{h}} \exp(-E(\boldsymbol{v},\boldsymbol{h};\theta) >_q - \ln Z(\theta)
\end{aligned}
$$

したがって、学習済み RBM は既学習データが入力された場合はエネルギーが低くなり、未学習データが入力された場合にはエネルギーが高くなる。この原理を応用し、

あるデータが入力された際の RBM のエネルギーを判定することで未学習データか既学習データかの判定を行う。

未学習データを入力した際のエネルギーと既学習データを入力した際のエネルギーをそれぞれ記録しておき、この値に基づいてある閾値を決定し、未学習か既学習か不明であるデータが入力された際の RBM のエネルギーが、この閾値を上回れば未学習、下回れば既学習と判定する。

2.0.4 データセットの獲得

本システムがデータセットを環境中から自動で学習するメカニズムについて説明する。以下、三目並べタスクの最適行動を学習するエージェントを例に説明する。

学習エージェントが対戦相手と三目並べを行う環境下で、三目並べに勝利した場合と敗北した場合にそれぞれ正の報酬と負の報酬を与えられるとする。この時、エージェントが観測する環境は盤面の状態であり、出力する行動は次に選択する手の盤面上の位置となる。

2.0.4.1 勝利データセットの獲得

まず、エージェントは勝利データセットを収集する。ある行動を出力した際、盤面が勝利条件を満たし、正の報酬が与えられたとする。その勝利条件を満たした際の行動と、その行動をとった際の環境を組みにしてデータセットとしてエージェントは保存する。

データセットがある一定数に達した場合、または試行回数が一定数に達した場合、エージェントは保存したデータセットを用いて学習を行う。

2.0.4.2 サブゴールの獲得

前述した、既学習、未学習判定を用いてサブゴールを獲得し、長期的な戦略を獲得するメカニズムについて説明する。

勝利データセットの獲得において、盤面が勝利条件を満たした場合、その時にとった行動と、行動を選択した際の環境を組みにしてデータセットとして保存した。この場合、勝利条件を満たした盤面の状態をゴールとしてデータセットを採集している。

サブゴールとは、ゴールにつながり得る環境の状態を指す。サブゴールを適切に設定し、サブゴールに至る行動と、その行動を選択した際の環境を更にデータセットに加える事で、長期的な戦略を獲得することができる。

サブゴールの設定方法について説明する。ある行動を選択し、環境が更新されたとする。その際の環境をエージェントの RBM が既学習であるか未学習であるかを前述したエネルギーによる判定法で判定する。そして、得られた環境が既学習であった場合、その環境はゴールへと至る可能性が高いため、その環境をサブゴールとして設定する。そして、サブゴールに至る直前の環境と、その状態にて選択した行動をデータセットとして新たに保存し、追加学習を行う。

このようにして、サブゴールについても学習した IL-RBM は、また新たに”サブゴールのサブゴール”を扱うことが可能になる。観測された環境が、既に学習済みであるサブゴールである環境と一致、近似すると判定された場合、さらにその環境に至る行動とその直前の環境をデータセットとして加えることで、サブゴールのサブゴールを学習でき、長期的な戦略を獲得することが可能である。

2.0.5 負のネットワーク、負のサブゴール

前述したサブゴールの設定法には負の報酬と行動の抑制を扱えない、という欠点があった。ある RBM がある環境を表す入力データを未学習か既学習か判定できたとしても、その環境が正の報酬に紐付いているか、負の報酬に紐付いているかを判定できないからである。

そこで、負の報酬と負の報酬を得る環境へ至る行動を抑制するために負の報酬をあつかうネットワークをエージェントに追加することにした。

エージェントは IL-RBM を二つ使用する。それぞれ正の報酬を扱うネットワークと負の報酬を扱うネットワークである。また、保持するデータセットも二種類用意する。正の報酬を得ることのできるゴールに至る行動と、その直前の環境の組、サブゴール

に至る行動とその直前の環境の組をデータセットとして扱う正のデータセットと、負の報酬について同様に扱う負のデータセットである。

負のデータセットを扱う負のネットワークでは、負の報酬を得ることになるゴール、サブゴールに至る行動が出力され、その直前の環境が入力データとなる。したがって、観測された環境をこのネットワークに入力した際の出力された行動は抑制されるべき行動である。

本エージェントでは負のネットワークからの出力を正のネットワークからの出力から引くことで、負の報酬を得る環境へ至る行動を抑制している。しかし、正のネットワークと負のネットワークを同列に扱うべきか、荷重をかけどちらかを優先すべきかなどは研究途中である。

第 3章

結論

本論文では、IL-RBM を改良し、強化学習タスクへの応用方法を示した。

文献で提案された IL-RBM は与えられたデータが既学習か未学習かを判定することが可能であるが、強化学習への応用の際、与えられたデータが正の報酬に関連づくものか、負の報酬に関連づくものかを区別することが出来なかった。

そこで、IL-RBM に正と負の二種類のネットワークを持たせることで、IL-RBM が正の報酬と関連の強いデータと負の報酬と関連の強いデータを区別することが可能であることを示した。

IL-RBM が正のネットワークと負のネットワークを持つことで、既存手法では扱えなかった負のサブゴールを設定可能となり、負の報酬を避けるような長期的な戦略を獲得できることを示した。

また、提案した IL-RBM をもちいたエージェントに三目並べタスクを実際に解かせた。エージェントは正の報酬へ繋がる行動のデータセットと負の報酬へ繋がる行動のデータセットを採集し、それぞれを提案した正負のネットワークで学習することで、負のネットワークを持たない IL-RBM より高い勝率で勝つことができることを示した。また、負の報酬を避けるような長期的な戦略により、より顕著に敗北率が減少することを示した。

参考文献

[1] 小林一郎. 人工知能の基礎. サイエンス社, 2008.