P9~

教師なし学習(*Unsupervised learning)* :教師なし学習では、解決したいタスクの目標値を持たず、観測された確率変数の結合分布の特徴を捉えようとする。密度推定、次元削減、クラスタリングなど様々なタスクがある。教師なし学習は、COとの関連では今のところほとんど注目されておらず、その直接的な利用は難しそうなので、これ以上議論しない。機械学習の教科書については、Bishop (2006); Murphy(2012); Goodfellow et al. (2016)を参照されたい。

強化学習 強化学習（RL）(*Reinforcement learning)*では、図2に示すように、エージェントはマルコフ決定過程（MDP）を通じて環境と相互作用する。各時間ステップにおいて、エージェントは与えられた環境の状態にあり、（おそらく確率的な）方針に従って行動を選択する。その結果、エージェントは環境から報酬を受け取り、新しい状態に入る。RLの目標は、リターンと呼ばれる将来の報酬の期待値を最大化するようにエージェントを訓練することである。与えられたポリシーに対して、現在のステート（ステートとアクションのペア）が与えられたときの期待リターンは、値関数（ステートアクション値関数）として知られています。価値関数はBellman方程式に従うので，この問題は動的プログラミングとして定式化でき，近似的に解くことができる．環境のダイナミクスはエージェントが知っている必要はなく、直接または間接的に学習され、探索か利用かのジレンマが生じる：可能な長期的な改良のために環境の知識を改良するために新しい状態を探索するか、またはこれまでに学習された最もよく知られたシナリオ（すでに訪問された状態または予測可能な状態にある傾向がある）を利用するかを選択する。

P10

つまり、将来の状態は現在の状態を介した過去の状態にのみ依存する（マルコフ特性）。マルコフ特性はもはや成立せず、MDPは部分的に観測可能であると言われる。例えば、エージェントが問題を解いたときは1、そうでないときは0というように、非常に疎な報酬を定義したい場合もあります。しかし、前述の設定は、エージェントが（ランダムに、または高度なアプローチによって）問題を解決するまで学習機会を提供しないため、困難である。さらに、政策が近似されている場合（例えば、線形関数）、学習は収束することが保証されておらず、ローカルミニマムに陥る可能性がある。例えば、自律走行車は、歩行者にぶつかり、劇的な負の報酬を受け取ることを恐れて、どこにも走行しないことを決定するかもしれない。これらの課題は、前述の探索のジレンマと強く関連している。読者は、強化学習に関する広範な教科書としてSutton and Barto(2018)を参照されたい。

ディープラーニング ディープラーニング(*Deep learning)*は、高次元空間においてパラメトリックに構成可能な関数を構築するための手法である。最も単純なニューラルネットワーク・アーキテクチャであるフィードフォワード・ニューラル・ネットワーク（多層パーセプトロン（MLP）とも呼ばれる）の場合、入力データはいくつかの層を連続して通過する。

P11

各層では、入力ベクトルに対してアフィン変換が適用され、続いて非線形スカラー関数（活性化関数と呼ばれる）が要素ごとに適用される。中間活性化と呼ばれる層の出力は次の層に渡される。すべてのアフィ変換は独立であり、実際には係数の異なる行列として表現される。これらは、選択された損失関数を最小化するために使用される最適化アルゴリズムである確率的勾配降下（SGD）を通じて学習、すなわち最適化される。確率性は、勾配更新を適用する前に損失を計算するために使用されるデータ点の数が限られていることに由来する。実際には、勾配は、逆伝播としても知られる連鎖法則に基づく実用的なアルゴリズムである逆モード自動微分を使用して計算されます。ディープニューラルネットワークは最適化が難しく、最適化をより良くするために、多くの場合、ネットワークのアーキテクチャ設計を変更することによって、多種多様な技術が開発されてきた。ニューラルネットワークは劇的な能力を持つため、つまり、基本的にどのようなデータセットにも対応できるため、オーバーフィッティングを起こしやすい。また、SGDによる訓練は、勾配に含まれるノイズのために正則化されるため、ニューラルネットワークが非常に大きく、より積極的な最適化で訓練した場合にオーバーフィットしてしまうような場合でも、一般的にオーバーフィッティングの問題に対してロバストになる。さらに、多くのハイパーパラメータが存在し、さまざまな組み合わせが評価される（ハイパーパラメータの最適化として知られる）。ディープラーニングはまた、従来のML手法とは異なり、データの利用可能なすべての生の特徴、例えば画像のすべてのピクセルを入力とすることで、従来のMLでは通常、限られた数のドメイン固有の特徴を設計する必要がある。

ディープラーニングの研究者たちは、可変長シーケンスなどの可変サイズのデータ構造を扱うことができる方法で、この多様な構造化データに取り組むためのさまざまな技術を開発してきた。この段落と次の段落では、そのような最先端の技術を紹介する。これらは完全なトピックであるが、理解できなくても本稿を読むのに支障はない。高いレベルでは、これらが異なる構造のデータを扱うために設計されたアーキテクチャであることを理解すれば十分である。その使い方、特に学習方法は、上で紹介した普通のフィードフォワード・ニューラル・ネットワークと非常によく似ている。最初に紹介するアーキテクチャは、カレントニューラルネットワーク（RNN）である。これらのモデルは、異なるシーケンスステップ間でパラメータを共有することにより、シーケンスデータ上で動作することができる。より正確には、同じニューラルネットワークブロックがシーケンスの各ステップで連続して適用される、すなわち、各タイムステップで同じアーキテクチャとパラメータ値を持つ。このようなネットワークの特徴は、リカレント層が存在することである。リカレント層は、図3に示すように、直前の層の活性化ベクトルと、直前のシーケンスステップにおける自身の活性化ベクトル（隠れ状態ベクトルと呼ばれる）の両方を入力とする。

P12

この文脈では、パラメータ共有が、異なる集合が同じサイズである必要はないという事実に対処するために使用される。アテンションは、図4に描かれているように、集合内のすべての要素に関する情報を照会し、ニューラルネットワーク内の下流処理のためにマージするために使用される。アフィニティ関数は、クエリ（注意をどこに集中させるべきかを示すあらゆる種類の文脈情報を表す）と集合の要素の表現（両方とも活性化ベクトル）を入力として受け取り、スカラーを出力する。これらのスカラーは（例えばソフトマックス関数で）正規化され、セット内の要素の表現の加重和を定義するために使用される。このコンテンツベースのソフトアテンションは、Bahdanauら(2015)によって導入された。アテンションメカニズムの一般的な説明は、Vaswaniら(2017)によって与えられている。アテンションはグラフ・ニューラル・ネットワーク（GNN）、すなわちVeliˇckovi´c et al. このアーキテクチャでは、各ノードはその近傍の集合に注目する。このプロセスは複数回繰り返され、より遠くにあるノードの情報を収集する。GNNはメッセージパッシングの一形態として理解することもできる（Gilmer et al.）

P13

ディープラーニングとバックプロパゲーションは、教師あり学習、教師なし学習、強化学習に用いることができる。ディープラーニングに特化した機械学習の教科書については、Goodfellow et al.

3 最近のアプローチ

組合せ最適化問題の解決に役立つMLの様々な利用法を調査し、直交する2つの軸に沿って整理する。まず、セクション3.1では、学習を用いる2つの主な動機、すなわち、近似と新しいポリシーの発見について説明する。次に、セクション3.2において、学習されたアルゴリズムと伝統的なアルゴリズムの要素を組み合わせる様々な方法の例を示す。

3.1 学習方法

この節は、セクション1.1で報告されたCOでMLを使用する2つの動機に関連する。ある研究では、研究者はCOアルゴリズムで行うべき決定に関する理論的・経験的知識を前提としているが、それらの決定の一部を機械学習で近似することで計算負荷を軽減したいと考えている。それとは逆に、専門的な知識では満足できないこともあり、より良い判断方法を見つけたいと考えることもある。したがって、試行錯誤の強化学習を通じてモデルを訓練するために、MLが登場する可能性がある。

我々は、2.2節で紹介した状態/アクションMDPのフレームワークで、これら2つの動機付けをフレームワーク化する。我々はCOアルゴリズムが利用するアルゴリズム的な決定を学習することに関心があり、その決定を行う関数をポリシーと呼ぶ。ポリシーは、MLを用いて学習したい関数であり、2つの動機付けがどのように2つの学習設定を自然にもたらすかを以下に示す。MDPの軌跡の長さが値1である場合（バンディット設定と呼ばれる）は、一般的なエッジケースであり、この設定は過剰に見えるかもしれないが、それにもかかわらず、手法の比較に役立つことに注意する。この設定において、学習者は性能尺度を最適化するために訓練されるのではなく、専門家を盲目的に模倣するために訓練される。

新しい方針を発見すること、すなわちアルゴリズムの決定関数を一から最適化することに関心がある場合、方針は、図6に示すように、経験を通じて強化学習によって学習されるかもしれない。RLの基本的なMDPのもとで学習問題を提示しても、RLの主要なアルゴリズム（近似動的計画法や政策勾配法）を用いて期待される報酬の総和を最大化することを制約するものではない。また、帯域アルゴリズム、遺伝的アルゴリズム、直接探索/局所探索などの別の最適化手法もRL問題を解くのに用いることができる4。