連載

5分で分かる!? 有名論文ナナメ読み

Matthew J. Holland（大阪大学）

Duchi, J. et al. : Adaptive Subgradient Methods for Online Learning and Stochastic Optimization

Journal of Machine Learning Research, Vol.12, pp.2121-2159（2011）

空港の検査ゲート，病院の手術室，囲碁の対局，絵画の制作現場まで，「AI」（人工知能）と呼ばれる種々の機械学習手法の応用例を目にすることはもはや日常茶飯事である．「学習」というだけに，何となく数値化されたデータを「経験」として，過去の経験と照らし合わせながら，与えられた事例を分類したり，よく似た事例を新しく生成したり，予測したりするというイメージが湧く．

端的にいえば，まったくその通りである．しかし，直感的にイメージができても，それを効率的に行う手順（＝学習則）を設計することは決して自明ではない．本稿では，カリフォルニア大学バークレイ校（現在スタンフォード大学）のJohn Duchiらによる2011年の論文を取り上げて，彼らが提案した学習則「AdaGrad」の基本的なアイディアがつかめるように，その背景にある原理について解説する．

この論文自体はテクニカルな内容が多く，分野外の人にはやや難易度が高いが，それでもこの論文を選んだのは，AdaGradの影響力があまりにも大きいからである．現代のAIの代名詞ともいえる「深層学習」で広く使われている学習則といえば，AdaGrad本家か，もしくはAdaGradを踏襲した改造版である．深層学習のモデル構造が注目を浴びることが多いが，種々のモデルを輝かしいAI技術へと成長させてくれるのは，AdaGradを中心とした学習則にほかならない．

フィードバックと傾き

まず，機械学習の基本的な手順を一言でいえば，フィードバックと応答の繰り返しである．Norbert Wiener教授の名著『Cybernetics』を読まれた読者には馴染みのある発想であろう．学習器（＝学習則を実行している計算機）の現時点での「出来栄え」を評価して，数値的な罰則を与える．これがフィードバックである．この罰則をなるべく小さくしようと，自らの状態を更新する．これが応答である．

どのような応答方式がよいか．与えられた罰則を自らの状態の関数として捉えれば，計算機なので，その最小値を求めることが比較的に得意である．十分滑らかな関数の最小値を探すときには，その関数の傾きを把握していると便利である．現時点の状態から見て，ある方向の傾きが急峻であれば，ほんの少し動いただけでも，罰則の値を大幅に減らすことができる．急勾配の方向を追って，効率良く最良の状態を求めていく「最急降下」に基づく学習則は，近代の機械学習では絶大な人気を誇る．AdaGradも，罰則の勾配を計算し，それを頼りに更新していく手法の1つである（AdaGradの「Grad」は勾配を意味するgradientに由来する）．

適応能力とスケーリング

先ほどのフィードバックと応答の話では，ある種の最適化問題を解こうとしているように見えたかもしれない．実際，最適化と学習を特に区別しない人もいるが，この二者間では，重要な違いがある．前者では，関数値なり，関数の勾配なり，目的関数そのものについて何らかの確かな情報があって，それを頼りに最適解を探索するのが大前提である．一方の後者では，今持っているデータよりも，これから入ってくるデータでの性能

のほうが重要であるから，フィードバックに使うべき目的関数そのものが変わってしまうのである．したがって，単なる最小化問題として扱い，最適解を手に入れたとしても，学習問題が解けたという保証はない．

そこで，学習過程の途中でもフィードバックに用いる目的関数が変容するのであれば，その都度の「クセ」を考慮した応答方式が合理的であろう．たとえば，不良スケーリングが典型的なクセである．目的関数の不良スケーリングとは，方向によって，関数の増減率が著しく異なる性質を指す．過敏な方向だと飛びすぎに注意が必要である．鈍感な方向だと加速させてあげないと効率が悪い．AdaGrad では，飛びすぎを抑えつつ，ほどよく鈍感な方向への更新を促すという機能を備えている．データの如何によって刻一刻変わりゆくフィードバックに適応（＝adapt）できるという重要な特徴である．

スパース性と計算効率

現代の機械学習のモデルでは，学習器の「状態」は無数の数値パラメータによって決定される．学習中はフィードバックに応じて，この状態を何度も更新させていく必要がある．いくら原理的に優れている更新方法でも，その作業に時間がかかりすぎると，限られた時間内に更新回数を十分に重ねることができないため，学習則としては無用である．AdaGrad の最大の特徴は，データの「スパース性」を巧みに利用して，効率化を図っているというところにある．たとえば，学習器の状態を決定づけるパラメータが 100 万個あったとしても，ある時点の罰則最小化に「効く」パラメータがその 100 万個のうちのたった数十〜数百個であることも珍しくない．このようなスパース性の下，有効な方向を最大限に追い求める一方，無効な方向は無視して，計算コストを省くという戦略が AdaGrad の根幹にある．この効率化の便益が顕著に見られるのは，深層学習である．膨大なデータセットが必要になるが，更新するたびにデータセットの全要素を使うのではなく，無作為に小さなサブセットを取り出した上でフィードバックを算出するという「確率的最適化」のアプローチを取ることが多い．したがって，目的関数が時間とともに変わっていくことが必然である．さらに，小さなサブセットであるがゆえに，勾配のスパース性も生ずる．この不確実性がある中，先に述べた特徴を AdaGrad が兼ね備えていることから，良い解をより安定的かつ高速に探しだす実力を示していることは納得できるだろう．

AdaGrad の影響力

結びとして，学問内外の AdaGrad の影響力について簡単に触れる．学習理論の名門国際会議である COLT で最初に発表されてから 9 年が経とうとしているが，2019 年 3 月現在，Google Scholar によると，Duchi et al.（2011）が引用された件数は 4,000 件を大きく超えている．さらに，この論文を引用した論文で，1,000 件以上の引用件数を有する論文が数十本あり，驚異的な波及効果が窺える．実務に焦点を当てても，まったく同様である．たとえば，TensorFlow, PyTorch, Chainer といった深層学習の開発ライブラリはどれも AdaGrad が標準装備されている．さらに，実用性の高さで人気の ADADELTA（Zeiler, 2012）や Adam（Kingma and Ba, 2014）はいずれも AdaGrad を筆頭に引用しており，その改造版であることは自他ともに認められる事実である．

AdaGrad とその子孫アルゴリズムが近代的な深層学習を開花させた大きな要因であることは疑う余地がない．時代が進み，新しい課題が見いだされ，既存手法が頭打ちしてしまう領域も見えてきている．さて，これらの課題を打開して，機械学習を次のステージへと進化させる学習アルゴリズムを生み出すのは誰だろうか．

（2019 年 3 月 26 日受付）

Matthew J. Holland matthew-h@ar.sanken.osaka-u.ac.jp

2017 年奈良先端科学技術大学院大学 博士課程修了．博士（工学）．2019 年より大阪大学産業科学研究所 助教．統計的学習理論と最適化技法に関する基礎研究とアルゴリズム開発に従事．