

Deep Learning 利用法と知見の体系化

Knowledges and Usages of Deep Learning

黒滝 紘生^{*1}
Hiroki Kurotaki

松尾 豊^{*2}
Yutaka Matsuo

^{*1} 東京大学工学部
Faculty of Engineering, The University of Tokyo

^{*2} 東京大学大学院 工学系研究科
School of Engineering, The University of Tokyo

Deep Learning has achieved significant results in recognition of images and sounds and in prediction of molecular activities. It is desirable to accumulate knowledge of Deep Learning to apply in various kinds of machine learning problems easily and effectively. We show that the combination of Pylearn2, a Deep Learning framework, Maxout Network, a learning model with deep architecture, and utilization of machine with Graphics Processing Unit(GPU) is effective to deal with problems with implementing Deep Learning.

1. 概要

深層学習(Deep Learning)は、画像認識や音声認識、化合物の活性予測といった分野で優れた性能を示しており、機械学習の様々な分野において、深層学習技術を容易に活用出来ることが望まれる。この研究では、深層学習の実装における課題を解決するための対策として、深層学習ライブラリ Pylearn2 と学習モデル Maxout Network を併用し、Graphics Processing Unit (GPU)を搭載したマシンを用いることが有効であると示した。

2. 背景

深層学習とは機械学習の手法の 1 つで、人間の脳の構造に類似した、多層ニューラルネットワーク構造をモデルに用いて、学習を行わせる方法である。深層学習には、今まで専門家によって行われてきた特徴量の変換方法それ自体も、習得する能力があると考えられている。生のデータを入力して学習させると、データから自動的に素性を作り、抽象表現を習得することが出来るようになるのである。この深層学習は、画像認識や音声認識、化合物の活性予測といった分野で優れた性能を示し、注目を浴びるようになった。図 1 は、深層学習が従来の機械学習と異なり、入力データから素性への変換方法までもを学習していることを模式的に表している。

深層学習が高い識別性能を持つことがわかり、深層学習を身近な問題に適用して、良い成果を得たいという機運が高まって

いる。出来るだけ簡易に、深層学習を様々な問題に応用するための方法論が求められている。しかし、深層学習は歴史の浅い発展途上の技術であり、実装の上では様々な問題点があり、国内での技術開発は特に遅れている。

3. 深層学習の実装における課題とその対策の提案

3.1 深層学習の実装における問題点

深層学習のアルゴリズムを、ウェブ工学などに応用するための具体的な実装を行う際の問題点として、「分類精度の再現の問題」「実装難易度の問題」「学習時間の問題」という 3 つが考えられる。

「分類精度の再現の問題」とは、深層学習が目された大きな理由である、高い識別精度を本当に実現できるアルゴリズムかどうか、また、論文において掲げられている精度を再現できる実装を手元に用意したり、あるいは制作できるかどうかの問題である。アルゴリズムを紹介する論文では、対応する実装が公開されていなかったり、例えメインのアルゴリズムについては書いてあっても、実験の際に設定したハイパーパラメータの詳細について記していない場合がある。よって、本当に論文に記述してある性能が発揮できるのかどうか、実際に利用してみないとわからない状況になっている。

「実装難易度の問題」とは、実際の問題に応用するにあたって、必要なプログラミング技術や数学的知識は、出来るだけ低くて済む方が望ましいということである。しかし、実際には深層学習の実装は、他の機械学習に比べて困難である。この問題は、主に標準といえる公開ライブラリが確立していないことが原因で起こる。深層学習は歴史の浅い発展途上の技術であり、どのアルゴリズムを定番とすれば良いのか、試行錯誤の段階にある。公開されているライブラリも、現状では、開発用途や実験的なものが多くなってしまっている。ライブラリが使えないと、利用者が自らコードを書くことになるが、深層学習の原理は多くの数学的背景の下成り立っており、理解には多くの時間がかかる。結果としては、プログラム開発に長い時間がかかってしまい、ビジネスシーンにおいて不利であると考えられる。

「学習時間の問題」は、深層学習のモデルを訓練するには、無視できない長さの時間が必要となり、刻一刻と変化するウェブサービスに対して応用するのは難しくなってしまう、という問題である。例えば、先述した Google による Supervision の学習では、1000 台のマシンによるクラスタを用いても、訓練に 3 日間かかっており、個人レベルでは実験自体がほぼ不可能である。一回

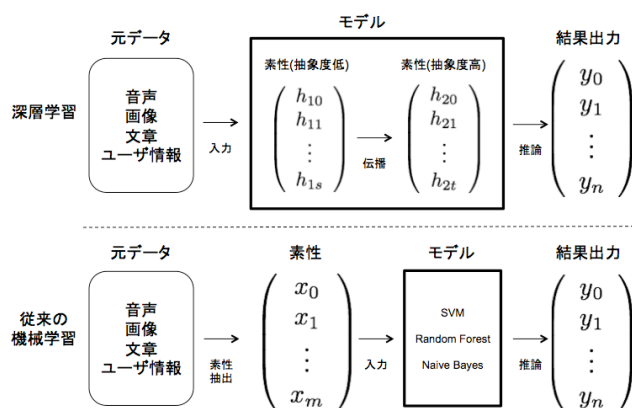


図 1. 深層学習と従来の機械学習の差異

連絡先: 黒滝 紘生, 東京大学工学部, 〒135-8656 東京都文京区弥生 2-11-16 工学部 9 号館, kurotaki@weblab.t.u-tokyo.ac.jp

の学習の時間が長くなれば、それだけ技術開発にかかる時間も長くなり、特に、スタートアップ企業が深層学習の技術を用いて市場での優位を得ようとする場合を考えると、開発期間の短縮は大きなポイントである。必要実行時間の長さをカバーするため、GPU を用いて演算をスピードアップさせる手法も確立されつつある。しかし、GPU は比較的高価なパーツである上、GPU の利用には特殊なプログラミングが要求され、開発における障壁の1つとなっている。

3.2 実装上の問題への対策

3.1 にて挙げた問題のうち、「実行時間の問題」は、GPU を搭載したマシンを学習に用いることで、大幅な改善が出来る。このとき、GPU 計算用ライブラリの利用が有効である。また、「実装難易度の問題」は、適切なライブラリを用いることで対応できる。「分類精度の再現の問題」については、良い分類精度を達成できると論文などで保証されたモデル及び、論文で用いたハイパーパラメータのセットを用意し、これを実際に手で走らせて、分類精度の再現性をチェックする必要がある。このようなセットは、やはり既存のライブラリから探すことになる。最終的に、3つの問題は、「GPU を搭載・利用する」「適切なライブラリを選択する」「そのライブラリの実装により、再現実験を行ってみる」という行程で解決できる。

現在深層学習を実装しているライブラリ 17 種を比較検討した結果、「実装難易度」が低く、GPU による「学習時間」の短縮が可能なライブラリとして、Pylearn2 と Deep Learning Tutorial の2つを選定した。この絞り込まれた中から、最後に残った「分類精度の再現性の問題」を解決できる分類モデルを選ぶことにした。

3.3 実装上の問題への対策

3.1 にて挙げた問題のうち、「実行時間の問題」は、GPU を搭載したマシンを学習に用いることで、大幅な改善が出来る。このとき、GPU 計算用ライブラリの利用が有効である。また、「実装難易度の問題」は、適切なライブラリを用いることで対応できる。「分類精度の再現の問題」については、良い分類精度を達成できると論文などで保証されたモデル及び、論文で用いたハイパーパラメータのセットを用意し、これを実際に手で走らせて、分類精度の再現性をチェックする必要がある。このようなセットは、やはり既存のライブラリから探すことになる。最終的に、3つの問題は、「GPU を搭載・利用する」「適切なライブラリを選択する」「そのライブラリの実装により、再現実験を行ってみる」という行程で解決できる。

現在深層学習を実装しているライブラリ 17 種を比較検討した結果、「実装難易度」が低く、GPU による「学習時間」の短縮が可能なライブラリとして、Pylearn2 と Deep Learning Tutorial の2つを選定した。この絞り込まれた中から、最後に残った「分類精度の再現性の問題」を解決できる分類モデルを選ぶことにした。

4. 深層学習の実装例とその検証

実際に Pylearn2 と Deep Learning Tutorial に実装されている深層学習モデルを用いて、機械学習のタスクを実行した。実際の分類精度を測定することにより、「分類精度の再現の問題」の検証を行った。また、GPU の利用により、どこまで「学習時間の問題」が解消されたのかもチェックした。実験には、手書き文字識別の MNIST と、カラー画像識別の CIFAR10 の2種のデータセットを用いた。

表 1. Maxout Network による MNIST 分類誤差の再現

手法	論文誤差	実験誤差	増加分
Maxout +畳み込み	0.45%	0.51%	+0.06%
Maxout (順序不変)	0.94%	1.16%	+0.12%

表 2. GPU の有無による CIFAR10 分類の実行時間

	GPU	CPU	DBN	SDA	CNN
1	有	3.1GHz×8	115 分	92 分	28 分
2	無	2.5GHz×4	236 分	1335 分	766 分

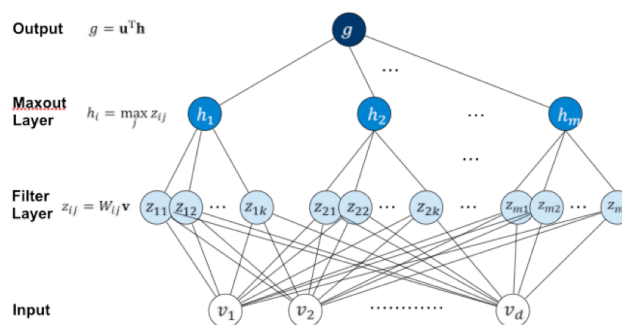


図 2. Maxout Network の模式図(レイヤー1層分)

結果として、表 1 のように、Pylearn2 に実装されている Maxout Network(図 2 参考)を用いると、「分類精度の再現の問題」は起こらず、論文で公表されていた分類誤差とほぼ同じ値を再現できることがわかった。また、表 2 のように、GPU を搭載したマシンと搭載していないマシンでは、同じ分類モデルを使っても最大で 27.35 倍まで実行時間の差が出た。GPU を利用することで、「学習時間の問題」を大きく改善できることがわかった。

5. 結論

深層学習モデルを、ウェブ工学などの応用問題に適用する際には、深層学習ライブラリの Pylearn2 と、そこに実装されている Maxout Network をモデルに選び、GPU を搭載したマシン上で学習を実行させることで、想定される問題点を全て解決できることがわかった。

参考文献

- [Goodfellow 13a] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. : Maxout networks, Proceedings of The 30th International Conference on Machine Learning, Vol. 28 (2013)
- [Goodfellow 13b] Goodfellow, I. J., Warde-Farley, D., Lamblin, P., Dumoulin, V., Mirza, M., Pascanu, R., Bergstra, J., Bastien, F., and Bengio, Y. : Pylearn2 : a machine learning research library, arXiv preprint arXiv:1308.4214 (2013)
- [Hinton 12] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. : Improving neural networks by preventing co-adaptation of feature detectors, Computing Research Repository, Vol. abs/1207.0580 (2012)