OUKA
The University of Osaka
Institutional Knowledge Archive

Title	ゴム材料中のナノ粒子構造に対するディープラーニング画像認識モデルの分散学習による高速最適化技術手法の検討
Author(s)	萩田, 克美
Citation	サイバーメディアHPCジャーナル. 2018, 8, p. 71-74
Version Type	VoR
URL	https://doi.org/10.18910/70838
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

ゴム材料中のナノ粒子構造に対するディープラーニング画像認識モデルの分散学習による高速最適化技術手法の検討

萩田　克美
防衛大学校　応用物理学科

1. はじめに

ディープラーニング技術は、ビジネスや産業の応用を中心として、急速に発達している[1-8]。特に、自動運転技術[5]のコアである画像認識の性能が、従来よりも圧倒的に高いことが牽引力である。また、Alpha GO [6]は、ディープラーニング技術の応用として有名である。ディープラーニングは、ニューラルネットワークを用いた機械学習の１つであり、多層のニューラルネットワークを用いることが特徴である。機械学習の主たるタスクは、教師付データの認識、教師なしデータの分類、回帰予測の３つである。様々な分野でこれらのタスクに対して、ディープラーニングが高い性能を示すことが報告されている。また、ソフト環境や入門書も充実し、比較的簡単に利用することができる[9,10]。ゴム材料に関する計算科学的研究でも活用が進んでいる[11,12]。我々は、ゴム中のフィラーの断面像について、ディープラーニングによる画像分類が、従来の HOG 特徴量の SVM による分類よりも、遙かに高性能であることを報告した[11]。また、ディープラーニング技術の１つである SRGAN を用いた超解像化処理について、集束イオンビーム走査型電子顕微鏡（FIB-SEM）の非対称な解像度での観察における有効性について報告した[12]。ディープラーニングの性能を高めるためには、ネットワーク構成や、様々な閾値パラメータや関数などを最適化する必要がある。これは、ハイパーパラメータの最適化と呼ばれる。

本課題では、ハイパーパラメータの最適化において、遺伝的アルゴリズム等を適用することで、ゴム材料に関するデータに対するディープラーニングの性能を、半自動的に向上させる手法・仕組みの構築に向けて、準備的な検討を実施した。

2. Deep Learning による画像分類と最適化

Deep Learning による画像分類について、情報理論的な側面ではなく、計算機利用の観点で簡単に説明する。畳み込みニューラルネットワーク(CNN)による事例としては、TensorFlow のチュートリアルの MNIST の手書き数字の分類がよく知られている。この事例では、畳み込み層が３層のモデルが用いられている。MNIST と同じようにデータを準備すれば、顔写真などの任意の画像データに対するディープラーニングは、Tensorflow などのツールを用いて簡単に実施することができる[9]。

2.1 TensorFlow と Keras について

Tensorflow は、Google 社が開発したオープンソースのディープラーニングのライブラリである。Keras は、Tensorflow などを簡便に使うための python ベースのラッパーであり、バランスの良さから広く普及している。図１のように、python 中でモデルを定義するスタイルで利用する。

```
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Convolution2D(64, 3, 3, border_mode='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Convolution2D(128, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Convolution2D(256, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(1024))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(256))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(self.nb_classes))
model.add(Activation('softmax'))
```

図１　Keras でのモデルの記述例

2.2 CNN のハイパーパラメータの最適化

Keras の python スクリプトでは、ネットワークの層の数や、種々のパラメータを系統的に変えた性能評価を簡単に行うことができる。多数のパラメータの探索的な最適化手法としては、単純なグリッドサーチやランダムサーチに加えて、python ベースのライブラリを利用することでより効率的な探索法を、手軽に実施することができる。最近、python の進化計算ライブラリ(Deap)を用いて、遺伝的アルゴリズムで２層 CNN のパラメータ探索をする事例も出てきた[13]。本課題では、ハイパーパラメータの最適化において、進化計算の手法と HPC を活用して、効率化する検討を目的とした。

3. ゴム中フィラーの画像分類の計算機実験

我々は、ゴム中のフィラーの断面像に関して、ディープラーニングの画像分類がどのような特性を持ち、必要な条件や最適な条件がどう決まるかに関心がある。特に、実験データを多数用意することが困難であるため、有意な判断を得るために最低限必要な実験データ数や撮影条件に関する情報を得たい。

3.1 撮影領域やデータ数に応じた性能に関する検討

ゴム中のフィラーの画像分類に関する基礎的な検討を行うために、実験データの代わりに、SPring-8 で計測した結果を「京」を用いて逆問題推定したフィラーの３次元構造のデータを用いた（図 2)。末端変性 SBR(M-SBR)と、未変性 SBR(n-SBR)の 2 種類の 3 次元構造を得ている。

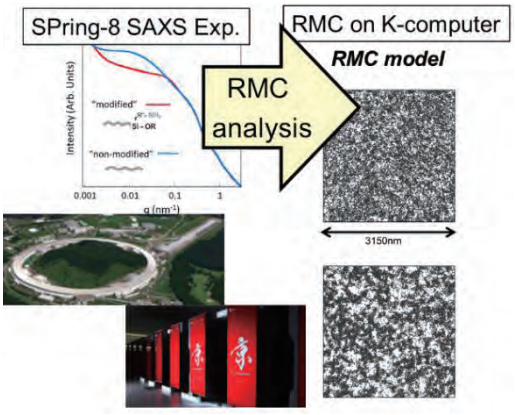

図 2　散乱スペクトルからのフィラー構造推定の概要

本検討では、画像データとして、この 2 種の構造とランダムにナノ粒子を配置した 3 次元構造(Rnd)から、ランダムな位置でランダムな方向のスライス像を多数作成した。ディープラーニングによる画像認識の性能評価では、学習用、テスト用、評価用の３つのデータセットを利用する（図 3)。最初に、撮影領域の大きさと、画像数に対する性能の変化を調べた。評価用の画像数は 1000 に固定し、学習用とテスト用の画像数を、8000 枚毎、2000 枚毎、500 枚毎とした。撮影領域の大きさは、$(250nm)^2$、$(500nm)^2$、$(1000nm)^2$ の 3 種とした。表 1 に示すように、画像数が多いほど、正答数が大きくなった。また、$(1000nm)^2$ のでは、500 枚毎でも 9 割の正答率となった。一方で、$(500nm)^2$ では、画像数が減ると、n-SBR の正答率が低下した。

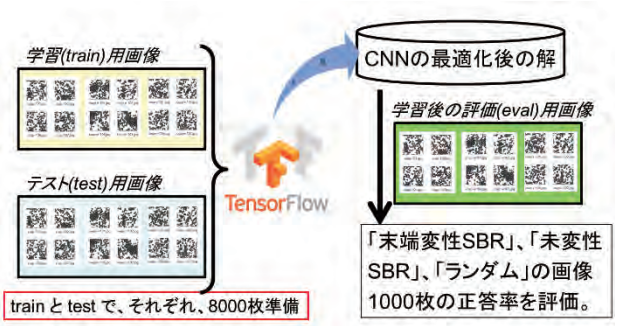

図 3 性能評価におけるデータと処理の流れの概要

表 1　全結合層の数に対する正答数の挙動

撮影領域	画像数	M-SBR	n-SBR	Rnd
$(250nm)^2$	8000+8000	434	925	516
$(500nm)^2$	500+500	981	663	925
	2000+2000	965	791	950
	8000+8000	908	889	993
$(1000nm)^2$	500+500	914	915	1000
	2000+2000	978	974	1000
	8000+8000	940	995	1000

3.2 FIB-SEM 観察データに対する予備検討

2nm 解像度で FIB-SEM で観察したデータ[14]から、$(1000nm)^2$ で 880x4 枚の画像を取得した。(図 4)。M-SBR, n-SBR, Rnd の 3 種類の画像分類を考えた。ハイパーパラメータ最適化の予備検討として、全結合層の数を変えて、正答数の挙動を確かめた（表 2)。ディープラーニングのモデルの違いにより、性能が変わることを確かめた。

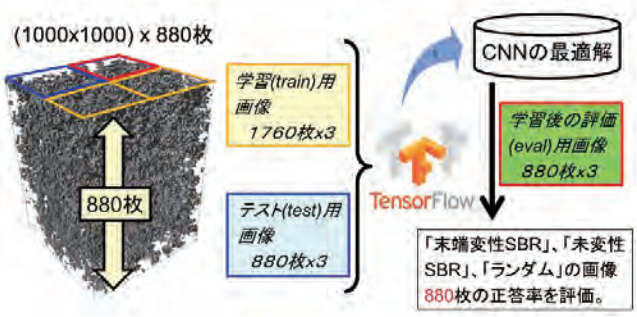

図 4　FIB-SEM で観察した 3 次元像とデータの流れ

表 2　全結合層の数に対する正答数の挙動

	M-SBR	n-SBR	Rnd
C4F2	753	767	880
C4F3	744	717	880
C4F4	863	812	880
C4F5	880	766	880
C4F6	878	823	880
C4F7	878	741	880
C4F8	878	833	880
C4F9	880	542	880

3.3 ハイパーパラメータの最適化の検討

基礎的な検討を行うために、3.1 で利用したデータを用いて、ハイパーパラメータの最適化の手法を検討した。最近、ハイパーパラメータの最適化に関しては、多くの研究がなされ、それらの報告がインターネット上でも紹介されている。最終的には、Keras-2 で、python の最適化ツールを併用し、Scikit-learn の GridSearchCV を用いたグリッドサーチやランダムサーチや、Gopt を用いたベイス最適化、hyperopt を用いた SMBO TPE 探索[15]などのコードが、ゴム中のナノ粒子の断面像のデータに対して動作することを確認した。現在、これらの計算特性や HPC をベースとした処理の最適化について検討し始めたところである。

上記のツールでは、基本的には、シリアルで探索を行い、1 つの探索過程で 1 つのハイパーパラメータセットの学習の計算処理を行う。ディープラーニングの場合、一般に、1 つのハイパーパラメータのセットに対する学習には、多くの計算時間を要する。この時間短縮には、1 ノードに多数の GPU を搭載した計算装置を用いることが 1 つの対応である。分散ノードでの高速化としては、データ並列やモデル並列が考えられるが、これらの検討は今後の課題である。加えて、ハイパーパラメータをシリアルで探索する代わりに、投機的な並列計算で経過時間を高速化する方法も検討する価値は高いと考えられる。さらに、学習済みの結果を再利用することで効率が高められる場合は、その際利用を積極的に行う高速化も重要と考えられる。

4. おわりに

本課題では、ディープラーニングによる画像分類におけるハイパーパラメータの最適化について検討した。Keras-Tensorflow をベースとしたハイパーパラメータの最適化について、実際に動作検証を行った。本課題の計算資源である VCC では、CentOS6 であったため、CUDA7.5、Python3、Tensorflow1.3 などの動作環境整備に多くの時間を費やす結果となった。分散処理については、ノード内のマルチ GPU 利用までは検討したが、ノード間での分散処理の検討は十分にできなかった。現状は、ディープラーニング処理自体の高速化よりも、ハイパーパラメータの探索の高速化の方が、全体性能に与える影響が大きいと思われる。この方向の検討として、効率的な探索法を用いたハイパーパラメータの探索法の検討を引き続き行うとともに、進化的アルゴリズムでハイパーパラメータを探索する方法などの検討を行っていく。

さらに、FIB-SEM などで観察した現実実験のデータに対して、ハイパーパラメータの最適化を検討していく予定である。また、単純な画像分類外にも、高分子材料系の機能-構造相関の予測として、構造に関するデータから、機械的特性などの機能がディープラーニングで予測可能であるかについて検討していく予定である。

謝辞

本研究は、大規模計算機システム公募型利用を通じて、ディープラーニングに関する計算機利用技術を検討した成果の一部である。この公募型利用は、大阪大学　李天鎬博士、名古屋大学　荻野正雄博士、

JSR（株）冨永哲雄博士との共同研究である。また、画像分類の検討に用いたデータは、理化学研究所のスーパーコンピュータ「京」を利用して得られた成果（課題番号：hp130050, hp140082, hp140239, hp150050, hp150064）を利用している。さらに、JHPCN 公募課題（課題番号：10-MD01, 11-MD02, 12-MD03, jh130028-NA19, jh140026-MD02, 14-NA28, jh150002-NA01, jh160036-NAH）や、HPCI 公募課題（課題番号：hp130062, hp130122, hp140191）などの実施を通じて得られた知見の一部を活用している。

参考文献

(1) Le Cun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature Vol. 521 (2015) pp. 436-444.

(2) Schmidhuber, J.: Deep Learning in Neural Networks: An Overview. Neural Netw. Vol. 61 (2015) pp. 85-117.

(3) Hinton, G. E., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science Vol. 313 (2006) pp. 504–507.

(4) Russakovsky, O. et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. Vol. 115 (2015) pp.211-252.

(5) Bojarski, M. et al.: End to End Learning for Self-Driving Cars. arXiv:1604.07316 (2016).

(6) Silver, D. et al.: Mastering the game of Go with deep neural networks and tree search. Nature Vol. 529 (2016) pp. 484–489.

(7) Goodfellow, I., Bengio, Y., Courville, A. Deep Learning. MIT Press; http://www.deeplearningbook.org (2016).

(8) Géron, A.: Hands-On Machine Learning with Scikit-Learn and TensorFlow. (O'Reilly Media 2017).

(9) 有山圭二: TensorFlow はじめました 実践! 最新 Google マシンラーニング, インプレス R&D (2016).

(10) 有山圭二: TensorFlow はじめました 2 機械学習で超解像, インプレス R&D (2017).

(11) 萩田克美, 冨永哲雄, 曽根卓男, 高橋一郎, Lee Chonho, 荻野正雄: ディープラーニングを用いたゴム中フィラー凝集構造の画像判別の特性評価, 日本ゴム協会誌 91 (2018) 3-8

(12) Hagita, K., Higuchi, T., Jinnai, H.: Super-resolution for asymmetric resolution of FIB-SEM 3D imaging using AI with deep learning. Sci. Rep. 8 (2018) 5877.

(13) https://github.com/shibuiwilliam/keras_opt/blob/master/ga_nn.ipynb

(14) K. Hagita, T. Tominaga, and T. Sone, T.: Large-scale reverse Monte Carlo analysis for the morphologies of silica nanoparticles in end-modified rubbers based on ultra-small-angle X-ray scattering data. Polymer Vol. 135 (2018) pp. 219-229.

(15) Bergstra, J., Bardenet, R., Bengio, Y. and Kegl. B. Algorithms for Hyper-Parameter Optimization, NIPS Vol. 24 (2011) 2546-2554.