数理工学概論
データサイエンスの数理１

情報学研究科　システム科学専攻
下平英寿

データサイエンスの数理

- 深層学習によるパターン認識などAI技術が近年注目を集めている
- その基盤となる確率論，統計学，機械学習について紹介する
- 特に多変量解析，自然言語処理，画像認識，DNA解析などを取り上げる

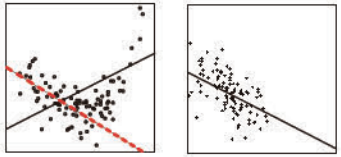
- かなりチャレンジングに詰め込んでいる．．．

もうひとつのテーマ

数理 vs 計算

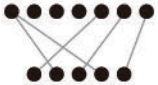
数理と計算どちらも重要

共変量シフト：予測分布と確率密度比



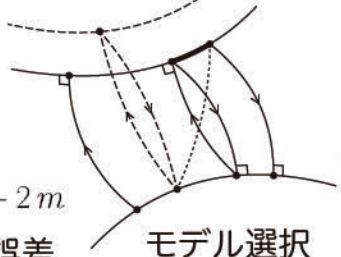
$$E_0\left\{\frac{q_1(x)}{q_0(x)}\log p(y|x;\theta)\right\} = E_1\left\{p(y|x;\theta)\right\}$$

リサンプリング



情報幾何学

$$D(q,p) = -\int q(x)\log\frac{p(x)}{q(x)}\,dx$$

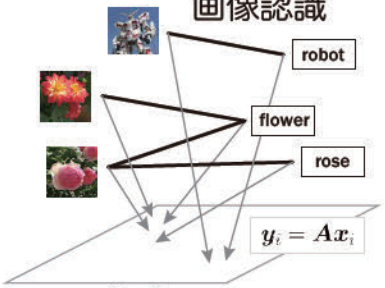


$$AIC = -2logL(\theta) + 2m$$

情報量規準・汎化誤差

モデル選択

画像認識



$$\phi(A) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\|Ax_i - Ax_j\|^2$$

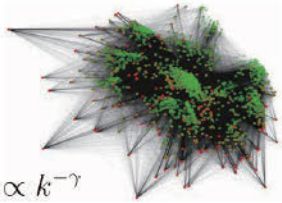
グラフの埋め込み

機械学習　ディープラーニング　情報統合の多変量解析

GPGPU
並列計算

統計学

単語のベクトル表現



$$P(k) \propto k^{-\gamma}$$

ベイズ統計学と頻度論をつなぐ

仮説検定の高次漸近理論

$$P(\sigma^2) = \Phi\left\{\sigma\,\Phi^{-1}(Q(\sigma^2))\right\}$$

複雑ネットワークの統計解析　分子進化系統樹と遺伝子発現解析　自然言語処理

システム科学専攻　下平研サイト　http://stat.sys.i.kyoto-u.ac.jp

下平英寿

学生

- 1990.3 東大　工学部　卒業
- 1995.3 東大　工学系研究科　博士（工学）

若手

- 1995.10-1996.3 University of Washington, Dept. Genetics　客員研究員
- 1996.7 統数研　予測制御　助手
- 1999.7-2001.1 Stanford University, Dept. Statistics　客員研究員

教員

- 2002.6 東工大　情報理工　講師　（数理計算　ORグループ）
- 2005.5 東工大　情報理工　助教授，准教授
- 2012.4 阪大　基礎工　教授　（数理科学　統計グループ）
- 2016.9 理研　革新知能統合研究センター　チームリーダー
- 2017.4 京大　情報学　教授

人工知能，AIブーム！

- ディープラーニング（深層学習）

- 画像認識の飛躍 (2012)
- アルファ碁 (2016)

http://sao-movie.net/story/character.html

RIKEN's New Research Center for Brain Science: 脳神経科学研究センター（理研CBS）ウェブサイト

RIKEN BRAIN SCIENCE INSTITUTE 文字サイズ A A A ENGLISH Find us on Facebook

Home | BSIについて | 研究室 | 連携研究 | プレスリリース・お知らせ | イベント | 人材育成 | 採用情報 | 研究に参加する

About
- データ集

Home » BSIについて » データ集

- BSI について
- データ集
  - 組織図
  - Leadership
- 一般の皆様へ

Organization Chart 組織図

理研BSI 組織図 (2017年7月1日現在)

Leadership

2017年7月～

センター長代行	合田裕紀子, Ph.D.
副センター長	岡本仁, M.D., Ph.D. 宮脇敦史, M.D., Ph.D. 加藤忠史, M.D., Ph.D. 大河内眞
特別顧問	伊藤正男, M.D., Ph.D. 甘利俊一, D.Eng.
研究コーディネーター（研究業務担当）	Charles Yokoyama, Ph.D.

過去のセンター長

2009年4月～2017年6月

センター長	利根川 進, Ph.D.

2008年4月～2009年3月

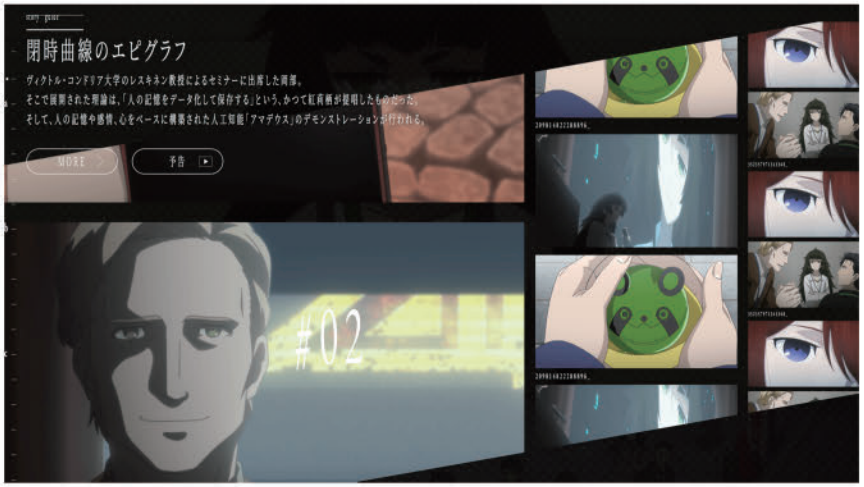
センター長代行	田中啓治, Ph.D.

2003年～2008年3月

センター長	甘利俊一, D.Eng.

1997年～2003年3月

初代センター長	伊藤正男, M.D., Ph.D.



http://steinsgate0-anime.com

http://www.brain.riken.jp/jp/about/fact.html

甘利俊一先生

ロボスタ | aiboニュースメール登録者限定の「aibo特別抽選販売」情報

「人工知能に人間らしさは必要か?」「シンギュラリティは来るか?」数理脳科学研究者 甘利俊一氏に聞く

2017年9月5日 By 神崎 洋治

その日、情報幾何学の創始者であり、計算論的神経科学研究の第一人者である甘利俊一先生を訪ねて、国立研究開発法人 理化学研究所 脳科学総合研究センターに足を運んだ。

講談社の「週刊 鉄腕アトムを作ろう!」の冊子でインタビューを行うことになったからだ。甘利先生が現在の人工知能ブームをどう捉え、鉄腕アトムのようなロボットにAIが搭載されていることについて、どう考えるのか、その答えが聞きたかった(インタビュー記事は8月29日発売の「週刊 鉄腕アトムを作ろう!」19号に掲載)。

人工知能の研究には大きく分けて2つの源流がある。
最初に「人工知能」という言葉が使われたのは1956年のダートマス会議だが「コンピュータにはこれほどの計算能力があるのだから、すぐに論理をこなす機械にもなる。人間の知能も言語と論理で構成されているのだから、きっといずれ人間の知能をコンピュータで実現で

編集部

第三次人工知能(AI)ブームと呼ばれていますが、先生はどのように評価されていますか

甘利

ニューラルネットワークを使って機械学習が成果を上げましたね。コンピュータによる画像認識の世界的な競技会(IMAGENETのILSVRC)が定期的に開催されていますが、ここ数年はニューラルネットワークを使った機械学習を活用したチームが軒並み好成績を収めたために注目を集めました。画像認識や音声認識の分野では人間を超えたと評価する声もあるようです。
わたしは囲碁が大好きなんですが(笑)、深層学習と強化学習が成果を出したおかげで、今では名人級の棋士がコンピュータに歯が立たなくなってしまいました。これも素晴らしい実績だと言えるでしょう。人工知能研究ではじめて実用的な技術が登場し、その成果を証明したと言えるでしょう。

https://robotstart.info/2017/09/05/dramari-riken.html

編集部

どのような課題があると感じていますか

甘利

あえて現状の課題を言うと、ディープラーニングで学習したシステムがどうしてその答えを導き出したのか、そのプロセスを開発者すら理解できていないことです。数理科学者の立場から言えば「結果良ければすべて良し」というだけでなく、深層学習によって何が捉えられているのか、何ができて何ができないのか、それを解明するのが重要です。もちろん、ディープラーニングが成果を生み出す謎を、数理で解明できないかと研究はしていますが、答えはまだ先になりそうですね。

編集部

ディープラーニングによる機械学習は、言語の認識や意味理解でも人間に近付くことができるのでしょうか? 言語は画像認識のように感覚的なものではなく、もっと論理的なものなので、同じように成果が出せるのでしょうか・・

甘利

たしかにニューラルネットワークは画像や音声などの「パターン認識」においてとても優れた成果を出し、研究者の多くは「研究を更に先に進めれば言語認識や時系列の把握までできるようになる」と言っています。しかし、本当にそんなところまでいけるのかは疑問です。
とはいえ、論理的な「言語」の分野でも、「Google翻訳」はニューラルネットワークを導入してから、実際に精度が良くなったと多くのユーザーが言っていますので、言語の分野でも一定の成果は上げられるのでしょう。

編集部

ディープラーニングによる機械学習は今後、どのように進展していくべきだと考えますか?

甘利

深層学習は一種のブームだと思っています。あと5年もしないうちにブームは冷めてしまうでしょう。しかし、ブームは冷めてもニューラルネットワークの良い技術は残っていって欲しいと思っています。それと同時に深層学習がなぜ成果を残せたのか、どこが良かったのか、論理や言語をどう処理していけば、更に発展させることができるのか、その原理をきちんと解明する必要があります。
パターン認識の技術は素晴らしいが、人間の知能にはまだまだ遠く及びません。ディープラーニングの層を何百にも増やせばもっと高度なことが認識できるようになると言う意見がありますが、それは本当でしょうか。人間の脳でもせいぜい10層程度なのに・・。それよりもディープラーニングの層を重ねていくときに何が起こっているのか、統計神経力学モデルを使ってそれを解明したいと考えています。

人工知能が更に人間に近付くために必要なこととは

編集部

人工知能が更に人間の知能に近付くにはどんなことが必要だと、先生はお考えですか。

甘利

人間の知能に近付くには2つの面が重要だと考えています。ひとつは、ニューラルネットワークのように素早く実行して結果を導き出すという面です。もうひとつが、じっくりと考えを導き出す「人間特有の意識」の面です。動物にも意識があると言われてはいますが、じっくりと考えたりはしません。

Imagenet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)

IMAGENET **Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)**

Held in conjunction with PASCAL Visual Object Classes Challenge 2012 (VOC2012)

Back to Main page

All results

- Task 1 (classification)
- Task 2 (localization)
- Task 3 (fine-grained classification)
- Team information and abstracts

Task 1

Team name	Filename	Error (5 guesses)	Description
SuperVision	test-preds-141-146.2009-131-137-145-146.2011-145f.	0.15315	Using extra training data from ImageNet Fall 2011 release
SuperVision	test-preds-131-137-145-135-145f.txt	0.16422	Using only supplied training data
ISI	pred_FVs_wLACs_weighted.txt	0.26172	Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.
ISI	pred_FVs_weighted.txt	0.26602	Weighted sum of scores from classifiers using each FV.
ISI	pred_FVs_summed.txt	0.26646	Naive sum of scores from classifiers using each FV.
ISI	pred_FVs_wLACs_summed.txt	0.26952	Naive sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.
OXFORD_VGG	test_adhocmix_classification.txt	0.26979	Mixed selection from High-Level SVM scores and Baseline Scores, decision is performed by looking at the validation performance
XRCE/INRIA	res_1M_svm.txt	0.27058	
OXFORD_VGG	test_finecls_classification.txt	0.27079	High-Level SVM over Fine Level Classification score, DPM score and Baseline Classification

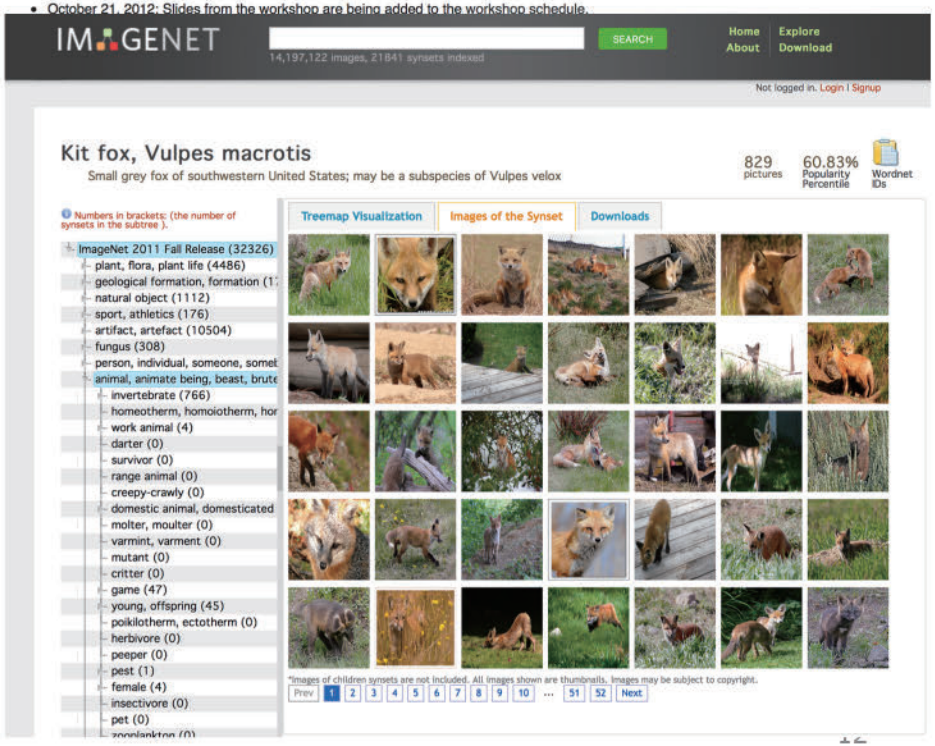
IMAGENET **Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)**

Held in conjunction with PASCAL Visual Object Classes Challenge 2012 (VOC2012)

Introduction Task Timetable Citation[new] Organizers Contact Workshop Download Evaluation Server

News

- September 2, 2014: A new paper which describes the collection of the ImageNet Large Scale Visual Recognition Challenge dataset, analyzes the results of the past five years of the challenge, and even compares current computer accuracy with human accuracy is now available. *Please cite it when reporting ILSVRC2012 results or using the dataset.*
- March 19, 2013: Check out ILSVRC 2013!
- January 26, 2012: Evaluation server is up. Now you can evaluate you own results against the competition entries.
- December 21, 2012: Additional analysis of the ILSVRC dataset and competition results is released.
- October 21, 2012: Slides from the workshop are being added to the workshop schedule.



http://image-net.org/challenges/LSVRC/2012/

優勝チームの論文
Krizhevsky et al. (NIPS 2012)

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

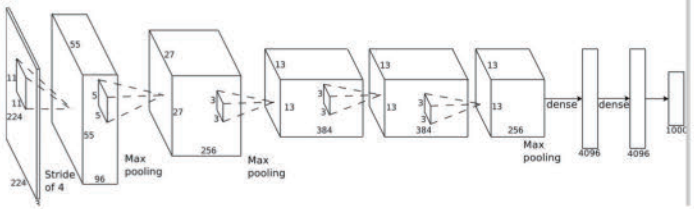
1 Introduction

Current approaches to object recognition make essential use of machine learning methods. To improve their performance, we can collect larger datasets, learn more powerful models, and use better techniques for preventing overfitting. Until recently, datasets of labeled images were relatively small — on the order of tens of thousands of images (e.g., NORB [16], Caltech-101/256 [8, 9], and CIFAR-10/100 [12]). Simple recognition tasks can be solved quite well with datasets of this size, especially if they are augmented with label-preserving transformations. For example, the current-best error rate on the MNIST digit-recognition task (<0.3%) approaches human performance [4]. But objects in realistic settings exhibit considerable variability, so to learn to recognize them it is necessary to use much larger training sets. And indeed, the shortcomings of small image datasets have been widely recognized (e.g., Pinto et al. [21]), but it has only recently become possible to collect labeled datasets with millions of images. The new larger datasets include LabelMe [23], which consists of hundreds of thousands of fully-segmented images, and ImageNet [6], which consists of over 15 million labeled high-resolution images in over 22,000 categories.

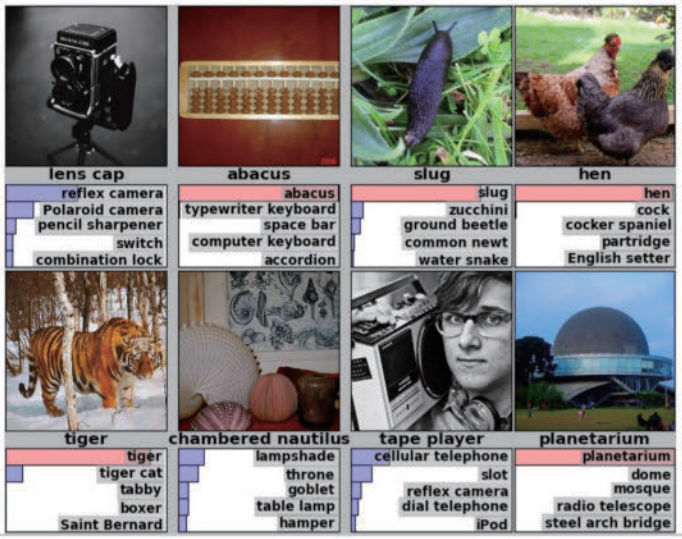
To learn about thousands of objects from millions of images, we need a model with a large learning capacity. However, the immense complexity of the object recognition task means that this problem cannot be specified even by a dataset as large as ImageNet, so our model should also have lots of prior knowledge to compensate for all the data we don't have. Convolutional neural networks (CNNs) constitute one such class of models [16, 11, 13, 18, 15, 22, 26]. Their capacity can be controlled by varying their depth and breadth, and they also make strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies). Thus, compared to standard feedforward neural networks with similarly-sized layers, CNNs have much fewer connections and parameters and so they are easier to train, while their theoretically-best performance is likely to be only slightly worse.

Our model

- Max-pooling layers follow first, second, and fifth convolutional layers
- The number of neurons in each layer is given by 253440, 186624, 64896, 64896, 43264, 4096, 4096, 1000



Validation classification



https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Imagenet Large Scale Visual Recognition Challenge 2014 (ILSVRC2014)

IMAGENET **Large Scale Visual Recognition Challenge 2014 (ILSVRC2014)**

Introduction History Data Tasks FAQ Development kit Timetable Citation[new] Organizers Sponsors Contact

News

- June 2, 2015: Additional announcement regarding submission server policy is released.
- May 19, 2015: Announcement regarding submission server policy is released.
- December 17, 2014: ILSVRC 2015 is announced.
- September 2, 2014: A new paper which describes the collection of the ImageNet Large Scale Visual Recognition Challenge dataset, analyzes the results of the past five years of the challenge, and even compares current computer accuracy with human accuracy is now available. *Please cite it when reporting ILSVRC2014 results or using the dataset.*
- August 18, 2014: Check out the New York Times article about ILSVRC2014.
- August 18, 2014: Results are released.
- August 18, 2014: Test server is open.
- July 25, 2014: Submission server is now open.
- July 15, 2014: Computational resources available, courtesy of NVIDIA.
- July 3, 2014: **Please note that the August 15th deadline is firm this year and will not be extended.**
- June 25, 2014: You can now browse all annotated detection images.
- May 3, 2014: ILSVRC2014 development kit and data are available. Please register to obtain the download links.
- April 8, 2014: Registration for ILSVRC2014 is open. Please register your team.
- January 19, 2014: Preparations for ImageNet Large Scale Visual Recognition Challenge 2014 (ILSVRC2014) are underway. Stay tuned!

Introduction

This challenge evaluates algorithms for object detection and image classification at large scale. This year there will be two competitions:

1. A PASCAL-style detection challenge on fully labeled data for 200 categories of objects, and
2. An image classification plus object localization challenge with 1000 categories.

   NEW: This year all participants are encouraged to submit object localization results; in past challenges, submissions to classification and classification with localization tasks were accepted separately.

One high level motivation is to allow researchers to compare progress in detection across a wider variety of objects -- taking advantage of the quite expensive labeling effort. Another motivation is to measure the progress of computer vision for large scale image indexing for retrieval and annotation.

History

- ILSVRC 2013
- ILSVRC 2012
- ILSVRC 2011
- ILSVRC 2010

Data

Dataset 1: Detection

As in ILSVRC2013 there will be object detection task similar in style to PASCAL VOC Challenge. There are 200 basic-level categories for this task which are fully annotated on the test data, i.e. bounding boxes for all categories in the image have been labeled. The categories were carefully chosen considering different factors such as object scale, level of image clutterness, average number of object instance, and several others. Some of the test images will contain none of the 200 categories.

NEW: The training set of the detection dataset will be significantly expanded this year compared to ILSVRC2013. 60658 new images have

Classification+localization

Task 2a: Classification+localization with provided training data

Classification+localization with provided training data: Ordered by localization error

Team name	Entry description	**Localization error**	Classification error
VGG	a combination of multiple ConvNets (by averaging)	0.253231	0.07405
VGG	a combination of multiple ConvNets (fusion weights learnt on the validation set)	0.253501	0.07407
VGG	a combination of multiple ConvNets, including a net trained on images of different size (fusion done by averaging); detected boxes were not updated	0.255431	0.07337
VGG	a combination of multiple ConvNets, including a net trained on images of different size (fusion weights learnt on the validation set); detected boxes were not updated	0.256167	0.07325
GoogLeNet	Model with localization ~26% top5 val error.	0.264414	0.14828
GoogLeNet	Model with localization ~26% top5 val error, limiting number of classes.	0.264425	0.12724
VGG	a single ConvNet (13 convolutional and 3 fully-connected layers)	0.267184	0.08434
SYSU_Vision	We compared the class-specific localization accuracy of solution 1 and solution 2 by the validation set. Then we chosen better solution on each class based on the accuracy. General speaking, solution 2 outformed solution 1 when there were multiple objects in the image or the objects are relatively small.	0.31899	0.14446
MIL	5 top instances predicted using FV-CNN	0.337414	0.20734
MIL	5 top instances predicted using FV-CNN + class specific window size rejection. Flipped training images are added.	0.33843	0.21023
SYSU_Vision	We just simply averaged the result between solution 1 and solution 2 to form our solution 4.	0.338741	0.14446
MIL	5 top instances predicted using FV-CNN + class specific window size rejection	0.340038	0.20823
MSRA Visual	Multiple CPP nets further tuned on validation set (A)	0.354700	0.09060

基礎理論は３０年まえにできている

バックプロパゲーション

ニューラルネットワーク > **バックプロパゲーション**

バックプロパゲーション（英: Backpropagation）または**誤差逆伝播法**（ごさぎゃくでんぱほう）[1]は、機械学習において、ニューラルネットワークを学習させる際に用いられるアルゴリズムである。1986年に*backwards propagation of errors*（後方への誤差伝播）の略からデビッド・ラメルハート らによって命名された[2]。

隠れ層のない2層のニューラルネットワークでの出力誤差からの確率的勾配降下法は1960年にB. Widrow と M.E. Hoff, Jr. らがWidrow-Hoff 法（デルタルール）という名称で発表した[3][4]。隠れ層のある3層以上の物は、1967年に甘利俊一が発表した[5][6]。その後、何度も適用され、1969年にアーサー・E・ブライソン（英語版）（Arthur E. Bryson）と何毓琦（英語版）が多段動的システム最適化手法として提案した[7][8]。ニューラルネットワークにおける応用を示唆した文献として、1974年のポール・ワーボス（英語版）[9]がある。1986年のデビッド・ラメルハート、ジェフリー・ヒントン、ロナルド・J・ウィリアムス（英語版）[10][2]らの適用により定着し、特に1986年の発表以降ニューラルネットワーク研究が注目を浴び再活性化することになった。

バックプロパゲーションでは、人工ニューロン(または「ノード」)で使われる活性化関数が可微分でなければならない。

https://ja.wikipedia.org/wiki/バックプロパゲーション

ネオコグニトロン

ネオコグニトロン（英: Neocognitron）は、1980年代に福島邦彦によって提唱された階層的、多層化された人工ニューラルネットワークである。手書き文字認識やその他のパターン認識の課題に用いられており、畳み込みニューラルネットワークの発想の元となった[1]。

ネオコグニトロンはヒューベルとウィーセルが1959年に提唱したモデルから発想を得ている。彼らは「単純細胞（英語版）」および「複雑細胞（英語版）」と呼ばれる一次視覚野の2種類の細胞を発見し、パターン認識タスクにおいて使用されるこれら2種類の細胞のカスケードモデルを提唱した[2][3]。

ネオコグニトロンはこれらのカスケードモデルが自然に発展したものである。ネオコグニトロンは複数の種類の細胞から構成され、その中で最も重要な細胞は「S細胞」および「C細胞」と呼ばれる[4]。局所特徴量はS細胞によって抽出され、微小変位（local shift）といったこれらの特徴の変形はC細胞に委ねられている。入力中の局所特徴量は、隠れ層によって徐々に統合され、分類される[5]。局所特徴量の統合の発想は、*LeNet*モデルや*SIFT*（英語版）モデルといったその他複数のモデルでも見られる。

ネオコグニトロンには様々な種類が存在する[6]。例えば、ある種のネオコグニトロンは、逆伝播シグナルを用いることによって同一入力中の複数のパターンを検出でき、選択的注意（selective attention）を達成する[7]。

https://ja.wikipedia.org/wiki/ネオコグニトロン

ヒントン先生の受賞スピーチ

Geoffrey Hinton receives the IEEE/RSE James Clerk Maxwell Medal - Honors Ceremony 2016

https://ieeetv.ieee.org/ieeetv-specials/geoffrey-hinton-receives-the-ieee-rse-james-clerk-maxwell-medal-honors-ceremony-2016?rf=channels%7C9&

ryugo hayano
@hayano

フォロー中

う〜ん．Amazonの研究開発費，すごい．2.5兆円！
ちなみに日本の科学研究費補助金（科研費）は2000億円程度．

twitter.com/goando/status/…

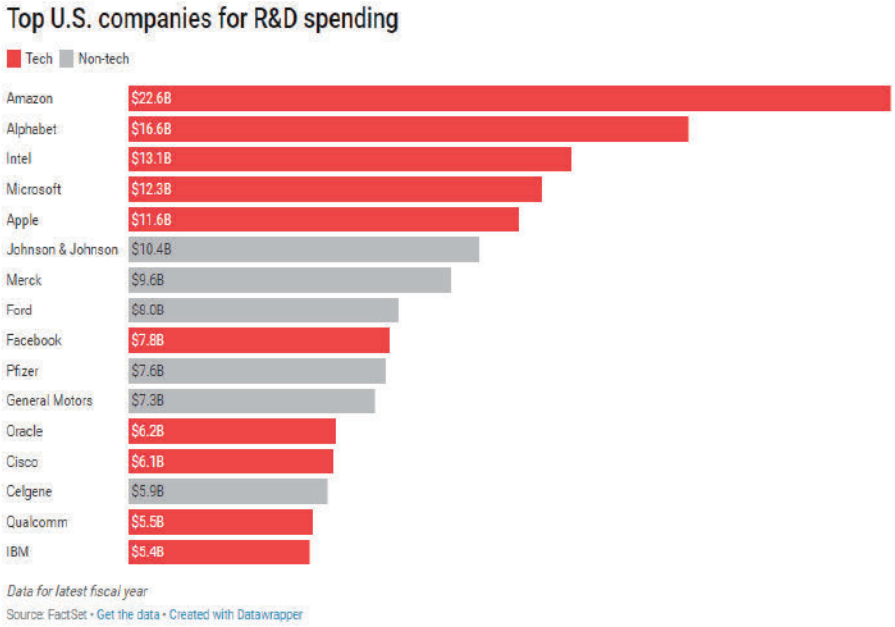
2018年04月11日 06時00分 メモ

Amazonが2017年の研究開発費に総額約2.5兆円を投資していたことが判明

By Robert Scoble

2017年にアメリカ国内で研究開発費に投資した額が高い順に企業をランク付けしたところ、Amazonがトップの約230億ドル(約2.5兆円)で、2位以下を大きく引き離す巨額の投資を行っていたことが明らかになりました。

Amazon spent nearly $23 billion on R&D last year - Recode
https://www.recode.net/2018/4/9/17204004/amazon-research-development-rd

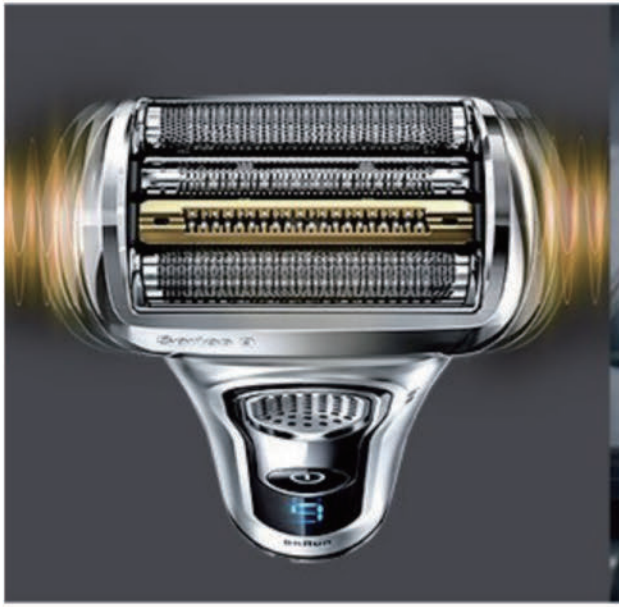


https://gigazine.net/news/20180411-amazon-research-development-rd/

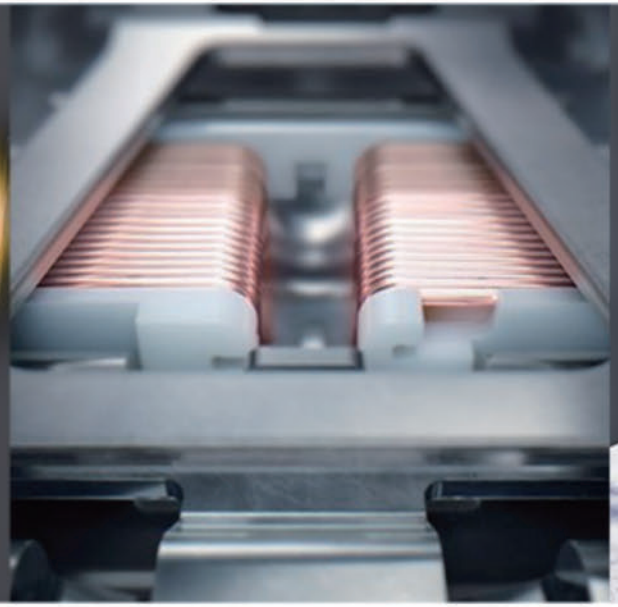
人工知能搭載シェーバー

https://www.braun.jp/ja-jp/male-grooming/shavers-for-men/autosense-technology

人工知能テクノロジー搭載シェーバー



カットシステム
と連動



ヒゲの濃さを読み取り
自動でパワーを調節

一度で*剃りきる

人工知能テクノロジー

ブラウン最新の人工知能テクノロジーはヒゲの濃さを読み取り毎秒13回自動でパワーを調節。必要な箇所ではパワーを最大化し、一度で*剃りきる。少ないストローク数で肌にやさしいシェービングを。*最少ストロークで。

ニューロ・ファジー洗濯機

1990年（平成2年）
ファジィ制御の全自動洗濯機、誕生。
布量、汚れの質と量、洗剤の種類を見分けて、約600通りの洗濯方法から適した洗い方を選びます。
ファジィ家電ブームの火つけ役となりました。

1991年（平成3年）
ニューロ・ファジィ全自動洗濯機、登場。
ファジィよりさらにきめ細やかに洗濯条件の違いに対応。布質や水質まで判断して、約3800通りの洗い方から最適の洗い方を選ぶニューロ・ファジィ制御です。

http://panasonic.jp/labo/history/product/kaji/wash/chr_table/
http://web.archive.org/web/20090322154817/http://panasonic.jp/labo/history/product/kaji/wash/chr_table/

ニューロ・ファジー洗濯機のその後

ナショナル洗濯機　ニューロファジー50　　２０年あまり前に購入して... 中古 売ります・あげます　　地元の掲

ジモティー 大阪版
都道府県選択

ジモティー > 売ります・あげます > 家電 > 生活家電 > 洗濯機 > 大阪府の洗濯機 > 大阪市の洗濯機 > ナショナル洗濯機

▼記事に違反内容を見つけたらこちら

ナショナル洗濯機　ニューロファジー50　（投稿ID：86yf4 ）

更新2018年3月22日 11:16
作成2018年1月27日 12:19

閲覧数：571



価格　：0 円
ジャンル　：生活家電 > 洗濯機
投稿者　：plant
投稿: 7
評価: 3 0 0
身分証認証 （認証とは）
取引き場所：大阪市 - 住吉区 - 大領
大阪市営地下鉄御堂筋線 - 西田辺駅

友達におしえる　ツイート　いいね！ 0　イイネ！

２０年あまり前に購入して使ったもので、一応洗濯機として動作はしています。
大きさ　６４ｃｍ×６０ｃｍ×高さ１００ｃｍ
取りにこられる方おられましたら差し上げます。

※お問い合わせの受付は終了いたしました。

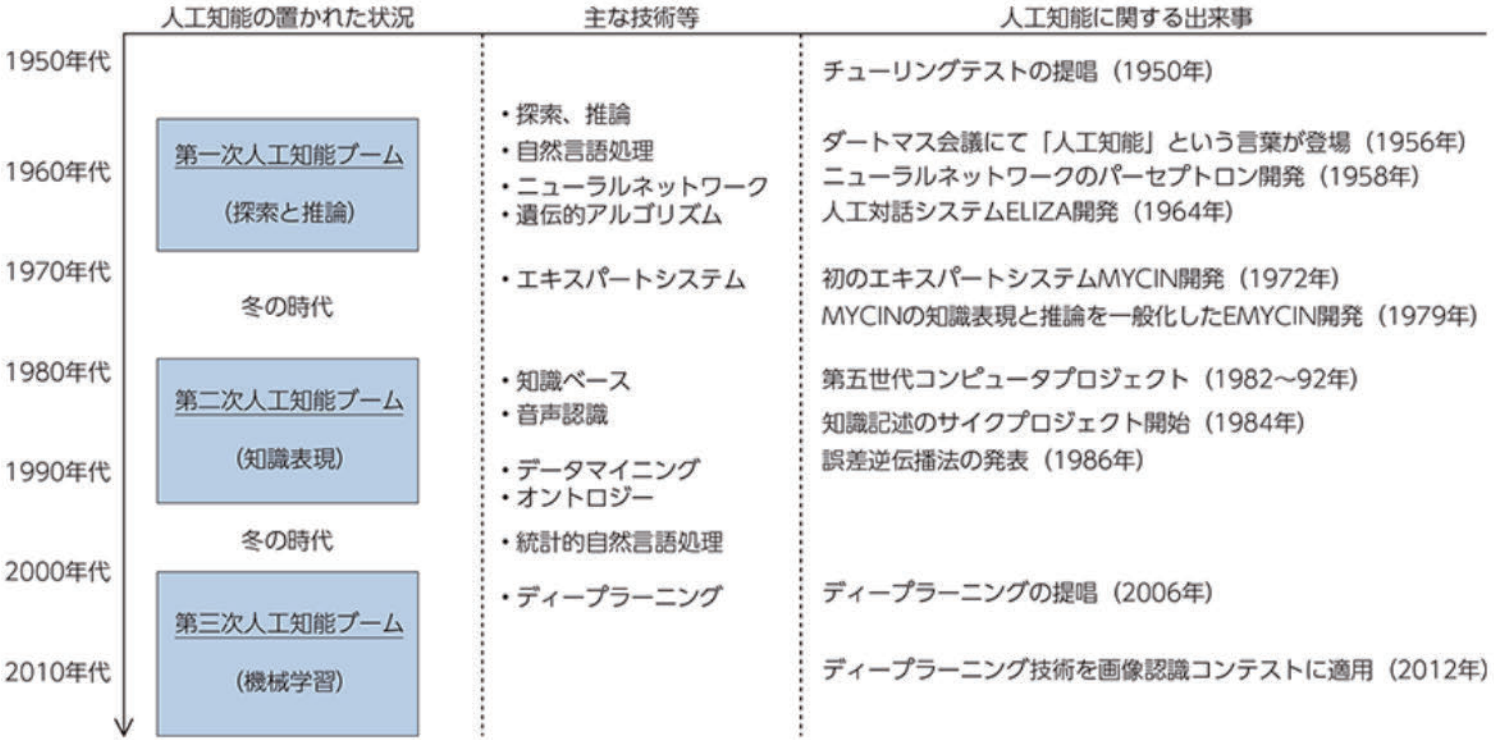
条件を登録して、似たモノの新着メールを受け取れます。

https://jmty.jp/osaka/sale-ele/article-86yf4

これまでのブーム

(2)人工知能(AI)研究の歴史[10]

人工知能(AI)の研究は1950年代から続いているが、その過程ではブームと冬の時代が交互に訪れてきたとされ、現在は第三次のブームとして脚光を浴びている(図表4-2-1-5)。



総務省のサイト http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/html/nc142120.html

エ　これまでの人工知能ブームをふりかえって

過去2回のブームにおいては、人工知能（AI）が実現できる技術的な限界よりも、社会が人工知能（AI）に対して期待する水準が上回っており、その乖離が明らかになることでブームが終わったと評価されている。このため、現在の第三次ブームに対しても、人工知能（AI）の技術開発や実用化が最も成功した場合に到達できる潜在的な可能性と、実現することが確実に可能と見込まれる領域には隔たりがあることを認識する必要がある、との指摘がある[12]。例えば、ディープラーニングによる技術革新はすでに起きているものの、実際の商品・サービスとして社会に浸透するためには実用化のための開発であったり社会環境の整備であったりという取組が必要である。実用化のための地道な取組が盛んになるほど、人工知能（AI）が社会にもたらすインパクトも大きくなり、その潜在的な可能性と実現性の隔たりも解消されると考えられる。

注目ニュース
» コンセプトをリニューアルした「CEBIT 2018」6月11日から15日に開催
Experience the new CEBIT

注目の特集
日本のハードウェアスタートアップ、SaaS、オープンイノベーション、RPAのこれからは？
プロ級のPR企画を自社で立案する方法

ライトニングトーク
Amazon Echoで睡眠をサポートする『アクセーラ』

アクセスランキング
1 日本のハードウェア

アスキーエキスパート — 第42回

AIはもはや先進テクノロジーではなく、誰もが使うコモディティに

SXSW 2018で見た「AIブーム」の終焉

2018年05月10日 09時00分更新　　文● 帆足啓一郎／アスキーエキスパート

国内の"知の最前線"から、変革の先の起こり得る未来を伝えるアスキーエキスパート。KDDI総合研究所の帆足啓一郎氏による人工知能についての最新動向をお届けします。

日本での注目度がますます高まっている巨大テクノロジーイベント「SXSW Interactive 2018」に、今年も参加することができた。筆者は、2016年から3年連続でSXSWに参加しており、この3年間は一貫して「人工知能」を調査の主テーマと設定し、関連するセッションの聴講を中心に動向を調査している。

AlphaGoの華々しい登場によりシンギュラリティ議論が巻き起こった2016年（参考：本連載2016年6月掲載「弁護士の仕事も奪われる対象に SXSWで見た人工知能最前線」）、人工知能と人類との共生が議論された2017年（参考：本連載2017年6月掲載「SXSW 2017が示したAI時代の共通課題とは？」）に続く動きは何か？　人工知能というテーマを中心に連続してSXSWに参加している筆者の目線ならではの考察を示す。

スタートアップ連載一覧
働き方改革に直接効く「RPA」最前線
何がすごい？働き方改革に直接効く「RPA」とは
ASCII STARTUP ライトニングトーク
Amazon Echoで睡眠をサポートする『アクセーラ』
ASCII STARTUP 業界ポジトーク
ブロックチェーンや自律走行ドローンの大学発ベンチャーがアツい
石川温の「動き出している5Gビジネス」
5Gで可能になる建築重機の遠隔操作

http://ascii.jp/elem/000/001/673/1673587/

Hiroki MORI (森 裕紀)
@HirokiMori

フォローする

人工知能ブームがなぜ滅びたのか、私よくわかる。PRMLの谷の詩にあるもの。
「データに根を下ろし　モデルと共に生きよう　実験と共に冬を越え　解析と共に春を歌おう」
どんなに恐ろしいライブラリを持っても、たくさんのかわいそうなGPUを操っても、データから離れては生きられないのよ。

21:36 - 2018年6月3日

475件のリツイート　**903**件のいいね

2　475　903

人工知能ブームは終わりそうだが？

- それでも着実に技術は発展します．大丈夫です．
- 今後も応用，商品化はほっておいてもどんどん進みます．
- いつか将来，ドラえもんだってできるでしょう．
- 理論の基礎研究をしっかり進めるべきです．

いまこそ．．．

数理を勉強しよう

研究室のサーバー

下平英寿 @hshimodaira · 2016年10月15日

評価用にTesla P100を貸してくれたNVIDIAに感謝です．ディープラーニングにはGPUが必要だから．さくっと動かしてくれた研究室の学生にも感謝．

下平英寿 @hshimodaira
研究室のサーバーです．日本で最初に稼働したTesla P100らしいです．みためではM40と区別つきません．最近のGPUはファンレスなんですね．

1　12　16

下平英寿 @hshimodaira · 2016年10月15日

研究室のサーバーです．日本で最初に稼働したTesla P100らしいです．みためではM40と区別つきません．最近のGPUはファンレスなんですね．

1　80　107

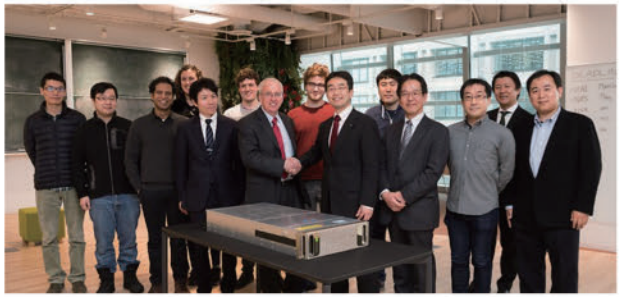
下平英寿 @hshimodaira · 2016年10月15日

ディープラーニングにはGPUが必要ということで，NVIDIAから評価用にお借りしたP100を研究室で動かした．日本で最初に稼働したTesla P100 PCIEらしい．

理研AIPのサーバー

理化学研究所が世界最大の NVIDIA DGX-1 システムを導入

BY NVIDIA JAPAN · MARCH 6, 2017



理化学研究所は、文部科学省が進める AIP プロジェクト (人工知能、ビッグデータ/IoT、サイバーセキュリティ統合プロジェクト) の研究開発拠点として昨年、「革新知能統合研究センター」を設置しました。

この度、同研究センターにおける人工知能研究を支える大規模計算リソースとして、「ディープラーニング解析システム」が導入されます。富士通株式会社様が受注されたこのシステムでは、GPU 計算ノードとして NVIDIA の「AI スーパーコンピューター」DGX-1 が採用されました。

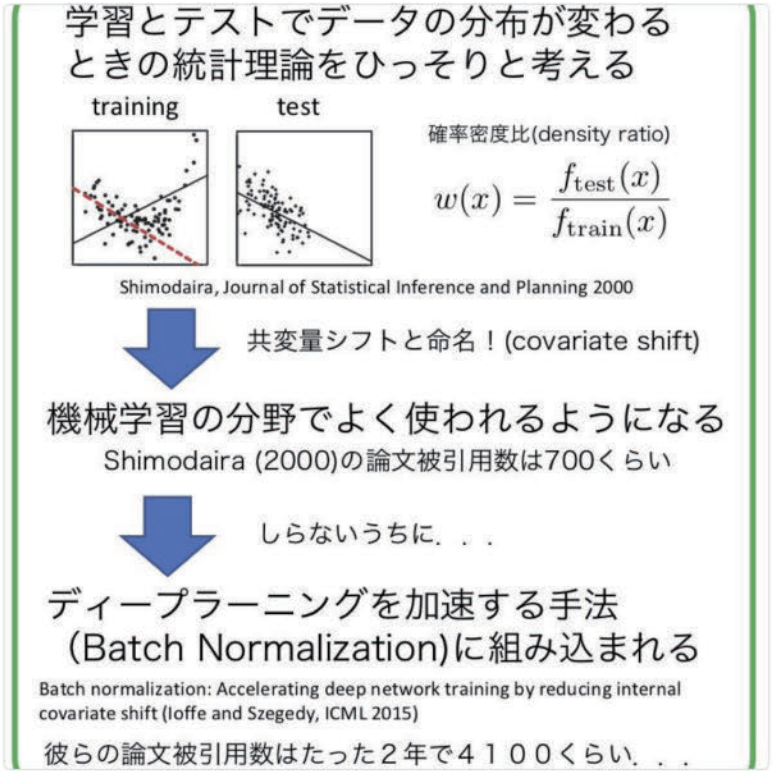
DGX-1 は、最新の Pascal アーキテクチャ GPU である Tesla P100 を 8 基搭載し、ディープラーニングの学習処理で活用される半精度浮動小数点 (FP16) 演算では 170 テラフロップスの性能を持ちます。

今回の「ディープラーニング解析システム」には 24 台の DGX-1 が導入され、FP16 演算性能は総計 4 ペタフロップスに達します。これは、現在 NVIDIA のお客様が運用する DGX-1 クラスターとして、世界最大となります。

下平英寿
@hshimodaira

アニメでも普通に登場するようになって「ディープラーニングやりたい」っていう人があまりに多いので，あえてこんな研究室説明チラシつくってみたよ．研究がどう伸びていくかなんてわからないから，すこしロングスパンでみてもいいんじゃないかな．

stat.sys.i.kyoto-u.ac.jp/post-ja/445/



13:50 - 2018年3月31日

253件のリツイート 624件のいいね

253 624

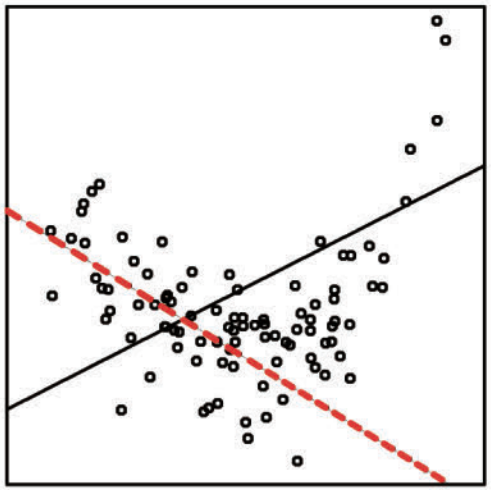
共変量シフト (covariate shift)
学習時とテスト時で xの分布f(x)が異なる

学習時

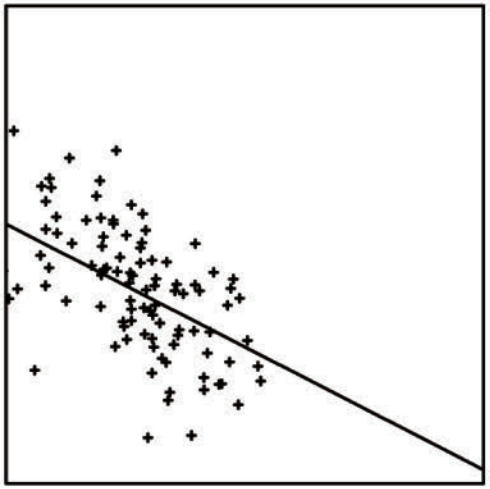
テスト時

$f(y|x)$

は共通



$f_{\mathrm{train}}(x)$



$f_{\mathrm{test}}(x)$

データの重み付けは確率密度比で行う： $w(x) = \dfrac{f_{\mathrm{test}}(x)}{f_{\mathrm{train}}(x)}$

「転移学習」として機械学習で使われている．因果推測にも関係

共変量シフトを提案した論文

ELSEVIER

Journal of Statistical Planning and Inference 90 (2000) 227–244

journal of statistical planning and inference

www.elsevier.com/locate/jspi

Improving predictive inference under covariate shift by weighting the log-likelihood function

Hidetoshi Shimodaira *

The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku, Tokyo 106-8569, Japan

Received 17 December 1998; received in revised form 21 January; accepted 25 February 2000

Abstract

A class of predictive densities is derived by weighting the observed samples in maximizing the log-likelihood function. This approach is effective in cases such as sample surveys or design of experiments, where the observed covariate follows a different distribution than that in the whole population. Under misspecification of the parametric model, the optimal choice of the weight function is asymptotically shown to be the ratio of the density function of the covariate in the population to that in the observations. This is the pseudo-maximum likelihood estimation of sample surveys. The optimality is defined by the expected Kullback–Leibler loss, and the optimal weight is obtained by considering the importance sampling identity. Under correct specification of the model, however, the ordinary maximum likelihood estimate (i.e. the uniform weight) is shown to be optimal asymptotically. For moderate sample size, the situation is in between the two extreme cases, and the weight function is selected by minimizing a variant of the information criterion derived as an estimate of the expected loss. The method is also applied to a weighted version of the Bayesian predictive density. Numerical examples as well as Monte-Carlo simulations are shown for polynomial regression. A connection with the robust parametric estimation is discussed.

MSC: 62B10; 62D05

Keywords: Akaike information criterion; Design of experiments; Importance sampling; Kullback–Leibler divergence; Misspecification; Sample surveys; Weighted least squares

1. Introduction

Let x be the explanatory variable or the covariate, and y be the response variable. In predictive inference with the regression analysis, we are interested in estimating the conditional density $q(y|x)$ of y given x, using a parametric model. Let $p(y|x,\theta)$ be the model of the conditional density which is parameterized by $\theta=(\theta^1,\ldots,\theta^m)' \in \Theta \subset \mathcal{R}^m$.

* Correspondence address: Department of Statistics, Sequoia Hall, 390 Serra Mall, Stanford University, Stanford, CA 94305-4065, USA.
E-mail address: shimo@ism.ac.jp (H. Shimodaira).

0378-3758/00/$ - see front matter
PII: S0378-3758(00)00115-4

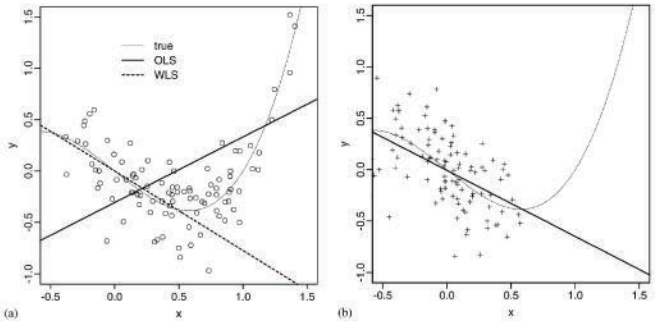


Fig. 1. Fitting of polynomial regression with degree $d=1$. (a) Samples (x_t, y_t) of size $n=100$ are generated from $q(y|x)q_0(x)$ and plotted as circles, where the underlying true curve is indicated by the thin dotted line. The solid line is obtained by OLS, and the dotted line is WLS with weight $q_1(x)/q_0(x)$. (b) Samples of $n=100$ are generated from $q(y|x)q_1(x)$, and the regression line is obtained by OLS.

On the other hand, MWLE $\hat{\theta}_w$ is obtained by weighted least squares (WLS) with weights $w(x_t)$ for the normal regression. We again consider the model with $d=1$, and the regression line fitted by WLS with $w(x)=q_1(x)/q_0(x)$ is drawn in dotted line in Fig. 1a. Here, the density $q_1(x)$ for imaginary "future" observations or that for the whole population in sample surveys is specified in advance by

$$x \sim \mathrm{N}(\mu_1, \tau_1^2), \tag{2.4}$$

where $\mu_1=0.0$, $\tau_1^2=0.3^2$. The ratio of $q_1(x)$ to $q_0(x)$ is

$$\frac{q_1(x)}{q_0(x)} = \frac{\exp(-(x-\mu_1)^2/2\tau_1^2)/\tau_1}{\exp(-(x-\mu_0)^2/2\tau_0^2)/\tau_0} \propto \exp\left(-\frac{(x-\bar{\mu})^2}{2\bar{\tau}^2}\right), \tag{2.5}$$

where $\bar{\tau}^2=(\tau_1^{-2}-\tau_0^{-2})^{-1}=0.38^2$, and $\bar{\mu}=\bar{\tau}^2(\tau_1^{-2}\mu_1-\tau_0^{-2}\mu_0)=-0.28$.

The obtained lines in Fig. 1a are very different for OLS and WLS. The question is: which is better than the other? It is known that OLS is the best linear unbiased estimate and makes small mean squared error of prediction in terms of $q(y|x)q_0(x)$ which generated the data. On the other hand, WLS with weight (2.5) makes small prediction error in terms of $q(y|x)q_1(x)$ which will generate future observations, and thus WLS is better than OLS here. To confirm this, a dataset of size $n=100$ is generated from $q(y|x)q_1(x)$ specified by (2.2) and (2.4). The regression line of $d=1$ fitted by OLS is shown in Fig. 1b, which is considered to have small prediction error for the "future" data. The regression line of WLS fitted to the past data in Fig. 1a is quite similar to the line of OLS fitted to the future data in Fig. 1b. In practice, only the past data is available. The WLS gave almost the equivalent result to the future OLS by using only the past data.

Shimodaira (2000)から削除した図

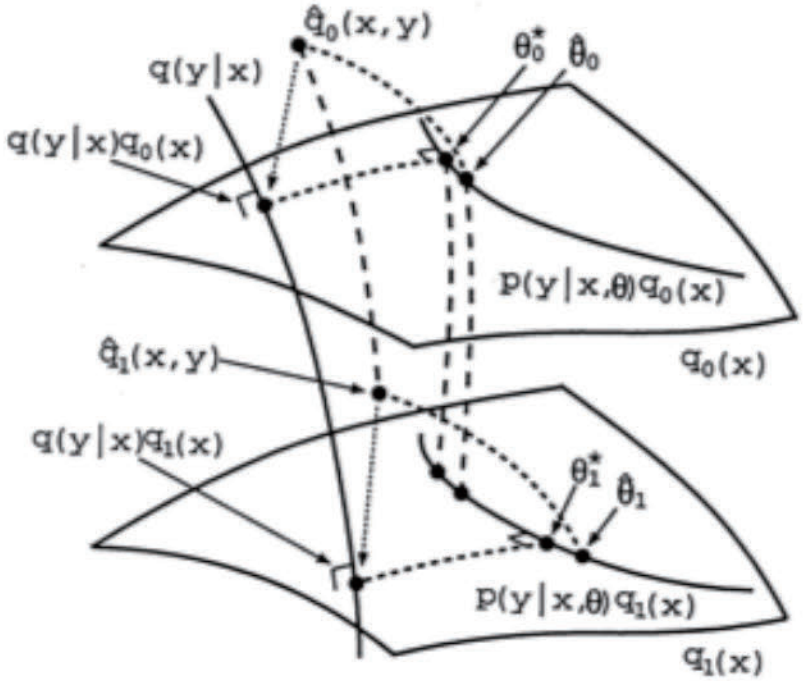
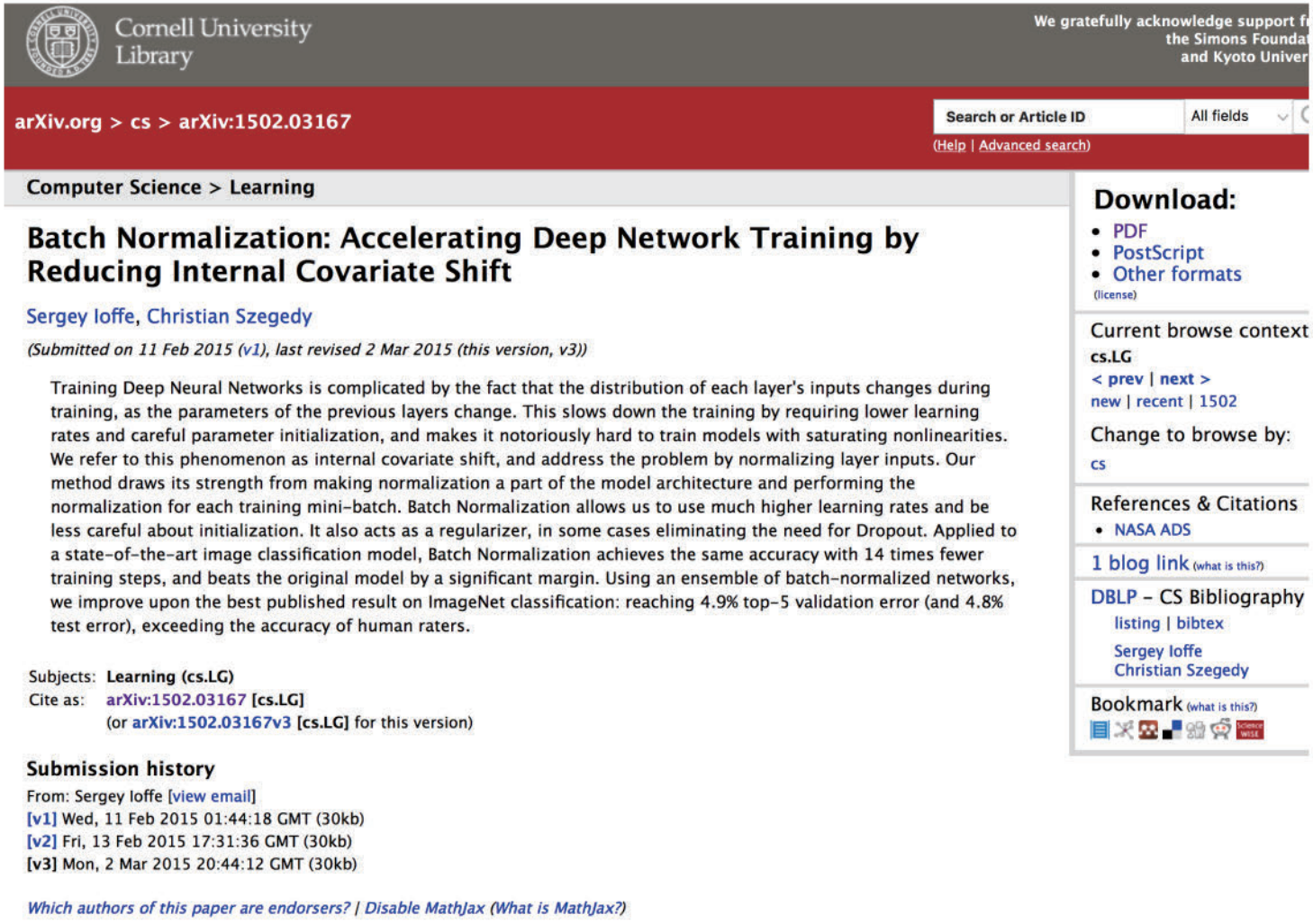


Figure 3: Schematic diagram of the space of joint densities of (x, y) in the sense of Amari (1985); each point represents a joint density of (x, y).

Improving predictive inference under covariate shift by weighting the log-likelihood function, ISM RM-712, 1998.

Batch normalizationの提案
ネットワーク内の共変量シフトを修正して
ディープラーニングを加速するアイデア



Cornell University Library

We gratefully acknowledge support f
the Simons Foundat
and Kyoto Univer

arXiv.org > cs > arXiv:1502.03167

Search or Article ID | All fields

(Help | Advanced search)

Computer Science > Learning

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

Sergey Ioffe, Christian Szegedy

(Submitted on 11 Feb 2015 (v1), last revised 2 Mar 2015 (this version, v3))

Training Deep Neural Networks is complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities. We refer to this phenomenon as internal covariate shift, and address the problem by normalizing layer inputs. Our method draws its strength from making normalization a part of the model architecture and performing the normalization for each training mini-batch. Batch Normalization allows us to use much higher learning rates and be less careful about initialization. It also acts as a regularizer, in some cases eliminating the need for Dropout. Applied to a state-of-the-art image classification model, Batch Normalization achieves the same accuracy with 14 times fewer training steps, and beats the original model by a significant margin. Using an ensemble of batch-normalized networks, we improve upon the best published result on ImageNet classification: reaching 4.9% top-5 validation error (and 4.8% test error), exceeding the accuracy of human raters.

Subjects: **Learning (cs.LG)**
Cite as: arXiv:1502.03167 **[cs.LG]**
(or arXiv:1502.03167v3 **[cs.LG]** for this version)

Submission history

From: Sergey Ioffe [view email]
[v1] Wed, 11 Feb 2015 01:44:18 GMT (30kb)
[v2] Fri, 13 Feb 2015 17:31:36 GMT (30kb)
[v3] Mon, 2 Mar 2015 20:44:12 GMT (30kb)

Which authors of this paper are endorsers? | Disable MathJax (What is MathJax?)

Download:
- PDF
- PostScript
- Other formats

(license)

Current browse context
cs.LG
< prev | next >
new | recent | 1502

Change to browse by:
cs

References & Citations
- NASA ADS

1 blog link (what is this?)

DBLP - CS Bibliography
listing | bibtex
Sergey Ioffe
Christian Szegedy

Bookmark (what is this?)

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

Sergey Ioffe
Google Inc., sioffe@google.com

Christian Szegedy
Google Inc., szegedy@google.com

Abstract

Training Deep Neural Networks is complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities. We refer to this phenomenon as *internal covariate shift*, and address the problem by normalizing layer inputs. Our method draws its strength from making normalization a part of the model architecture and performing the normalization *for each training mini-batch*. Batch Normalization allows us to use much higher learning rates and be less careful about initialization. It also acts as a regularizer, in some cases eliminating the need for Dropout. Applied to a state-of-the-art image classification model, Batch Normalization achieves the same accuracy with 14 times fewer training steps, and beats the original model by a significant margin. Using an ensemble of batch-normalized networks, we improve upon the best published result on ImageNet classification: reaching 4.9% top-5 validation error (and 4.8% test error), exceeding the accuracy of human raters.

1 Introduction

Deep learning has dramatically advanced the state of the art in vision, speech, and many other areas. Stochastic gradient descent (SGD) has proved to be an effective way of training deep networks, and SGD variants such as momentum (Sutskever et al., 2013) and Adagrad (Duchi et al., 2011) have been used to achieve state of the art performance. SGD optimizes the parameters Θ of the network, so as to minimize the loss

$$\Theta = \arg\min_{\Theta} \frac{1}{N} \sum_{i=1}^{N} \ell(\mathbf{x}_i, \Theta)$$

where $\mathbf{x}_{1\ldots N}$ is the training data set. With SGD, the training proceeds in steps, and at each step we consider a *mini-batch* $\mathbf{x}_{1\ldots m}$ of size m. The mini-batch is used to approximate the gradient of the loss function with respect to the parameters, by computing

$$\frac{1}{m} \frac{\partial \ell(\mathbf{x}_i, \Theta)}{\partial \Theta}.$$

Using mini-batches of examples, as opposed to one example at a time, is helpful in several ways. First, the gradient of the loss over a mini-batch is an estimate of the gradient over the training set, whose quality improves as the batch size increases. Second, computation over a batch can be much more efficient than m computations for individual examples, due to the parallelism afforded by the modern computing platforms.

While stochastic gradient is simple and effective, it requires careful tuning of the model hyper-parameters, specifically the learning rate used in optimization, as well as the initial values for the model parameters. The training is complicated by the fact that the inputs to each layer are affected by the parameters of all preceding layers – so that small changes to the network parameters amplify as the network becomes deeper.

The change in the distributions of layers' inputs presents a problem because the layers need to continuously adapt to the new distribution. When the input distribution to a learning system changes, it is said to experience *covariate shift* (Shimodaira, 2000). This is typically handled via domain adaptation (Jiang, 2008). However, the notion of covariate shift can be extended beyond the learning system as a whole, to apply to its parts, such as a sub-network or a layer. Consider a network computing

$$\ell = F_2(F_1(\mathbf{u}, \Theta_1), \Theta_2)$$

where F_1 and F_2 are arbitrary transformations, and the parameters Θ_1, Θ_2 are to be learned so as to minimize the loss ℓ. Learning Θ_2 can be viewed as if the inputs $\mathbf{x} = F_1(\mathbf{u}, \Theta_1)$ are fed into the sub-network

$$\ell = F_2(\mathbf{x}, \Theta_2).$$

For example, a gradient descent step

$$\Theta_2 \leftarrow \Theta_2 - \frac{\alpha}{m} \sum_{i=1}^{m} \frac{\partial F_2(\mathbf{x}_i, \Theta_2)}{\partial \Theta_2}$$

(for batch size m and learning rate α) is exactly equivalent to that for a stand-alone network F_2 with input $\mathbf{x}$. Therefore, the input distribution properties that make training more efficient – such as having the same distribution between the training and test data – apply to training the sub-network as well. As such it is advantageous for the distribution of $\mathbf{x}$ to remain fixed over time. Then, Θ_2 does

indicate that the parameters γ and β are to be learned, but it should be noted that the BN transform does not independently process the activation in each training example. Rather, $\mathrm{BN}_{\gamma,\beta}(x)$ depends both on the training example *and the other examples in the mini-batch*. The scaled and shifted values y are passed to other network layers. The normalized activations $\widehat{x}$ are internal to our transformation, but their presence is crucial. The distributions of values of any $\widehat{x}$ has the expected value of 0 and the variance of 1, as long as the elements of each mini-batch are sampled from the same distribution, and if we neglect ϵ. This can be seen by observing that $\sum_{i=1}^{m} \widehat{x}_i = 0$ and $\frac{1}{m}\sum_{i=1}^{m} \widehat{x}_i^2 = 1$, and taking expectations. Each normalized activation $\widehat{x}^{(k)}$ can be viewed as an input to a sub-network composed of the linear transform $y^{(k)} = \gamma^{(k)}\widehat{x}^{(k)} + \beta^{(k)}$, followed by the other processing done by the original network. These sub-network inputs all have fixed means and variances, and although the joint distribution of these normalized $\widehat{x}^{(k)}$ can change over the course of training, we expect that the introduction of normalized inputs accelerates the training of the sub-network and, consequently, the network as a whole.

During training we need to backpropagate the gradient of loss ℓ through this transformation, as well as compute the gradients with respect to the parameters of the BN transform. We use chain rule, as follows (before simplification):

$$\begin{aligned}
\frac{\partial \ell}{\partial \widehat{x}_i} &= \frac{\partial \ell}{\partial y_i} \cdot \gamma \\
\frac{\partial \ell}{\partial \sigma_B^2} &= \sum_{i=1}^{m} \frac{\partial \ell}{\partial \widehat{x}_i} \cdot (x_i - \mu_B) \cdot \frac{-1}{2}(\sigma_B^2 + \epsilon)^{-3/2} \\
\frac{\partial \ell}{\partial \mu_B} &= \left(\sum_{i=1}^{m} \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{-1}{\sqrt{\sigma_B^2 + \epsilon}}\right) + \frac{\partial \ell}{\partial \sigma_B^2} \cdot \frac{\sum_{i=1}^{m} -2(x_i - \mu_B)}{m} \\
\frac{\partial \ell}{\partial x_i} &= \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{1}{\sqrt{\sigma_B^2 + \epsilon}} + \frac{\partial \ell}{\partial \sigma_B^2} \cdot \frac{2(x_i - \mu_B)}{m} + \frac{\partial \ell}{\partial \mu_B} \cdot \frac{1}{m} \\
\frac{\partial \ell}{\partial \gamma} &= \sum_{i=1}^{m} \frac{\partial \ell}{\partial y_i} \cdot \widehat{x}_i \\
\frac{\partial \ell}{\partial \beta} &= \sum_{i=1}^{m} \frac{\partial \ell}{\partial y_i}
\end{aligned}$$

Thus, BN transform is a differentiable transformation that introduces normalized activations into the network. This ensures that as the model is training, layers can continue learning on input distributions that exhibit less internal covariate shift, thus accelerating the training. Furthermore, the learned affine transform applied to these normalized activations allows the BN transform to represent the identity transformation and preserves the network capacity.

3.1 Training and Inference with Batch-Normalized Networks

To *Batch-Normalize* a network, we specify a subset of activations and insert the BN transform for each of them, according to Alg. 1. Any layer that previously received x as the input, now receives $\mathrm{BN}(x)$. A model employing Batch Normalization can be trained using batch gradient descent, or Stochastic Gradient Descent with a mini-batch size $m > 1$, or with any of its variants such as Adagrad (Duchi et al., 2011). The normalization of activations that depends on the mini-batch allows efficient training, but is neither necessary nor desirable during inference; we want the output to depend only on the input, deterministically. For this, once the network has been trained, we use the normalization

$$\widehat{x} = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}}$$

using the population, rather than mini-batch, statistics. Neglecting ϵ, these normalized activations have the same mean 0 and variance 1 as during training. We use the unbiased variance estimate $\mathrm{Var}[x] = \frac{m}{m-1} \cdot \mathrm{E}_B[\sigma_B^2]$, where the expectation is over training mini-batches of size m and σ_B^2 are their sample variances. Using moving averages instead, we can track the accuracy of a model as it trains. Since the means and variances are fixed during inference, the normalization is simply a linear transform applied to each activation. It may further be composed with the scaling by γ and shift by β, to yield a single linear transform that replaces $\mathrm{BN}(x)$. Algorithm 2 summarizes the procedure for training batch-normalized networks.

Input: Network N with trainable parameters Θ; subset of activations $\{x^{(k)}\}_{k=1}^{K}$
Output: Batch-normalized network for inference, $N_{\mathrm{BN}}^{\mathrm{inf}}$

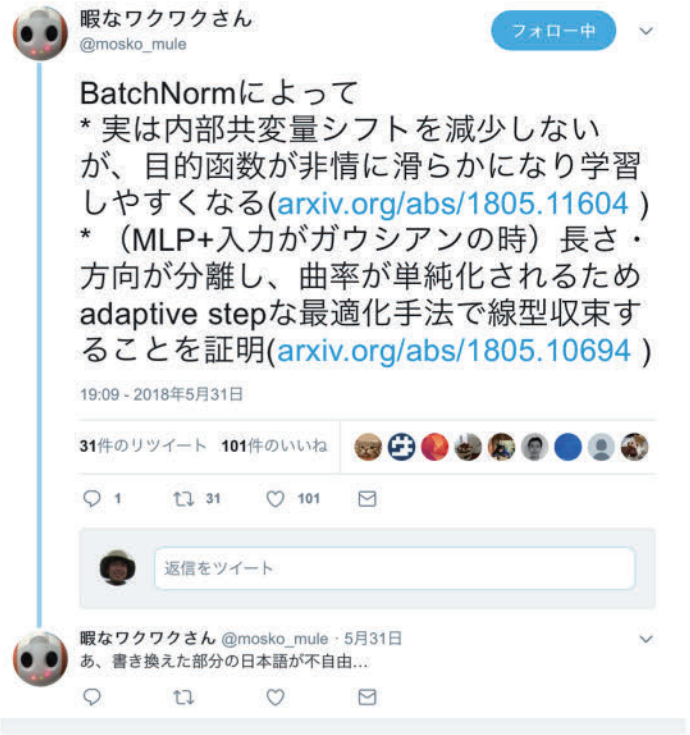
1: $N_{\mathrm{BN}}^{\mathrm{tr}} \leftarrow N$ // Training BN network
2: **for** $k = 1 \ldots K$ **do**
3: Add transformation $y^{(k)} = \mathrm{BN}_{\gamma^{(k)},\beta^{(k)}}(x^{(k)})$ to $N_{\mathrm{BN}}^{\mathrm{tr}}$ (Alg. 1)
4: Modify each layer in $N_{\mathrm{BN}}^{\mathrm{tr}}$ with input $x^{(k)}$ to take $y^{(k)}$ instead
5: **end for**
6: Train $N_{\mathrm{BN}}^{\mathrm{tr}}$ to optimize the parameters $\Theta \cup \{\gamma^{(k)}, \beta^{(k)}\}_{k=1}^{K}$
7: $N_{\mathrm{BN}}^{\mathrm{inf}} \leftarrow N_{\mathrm{BN}}^{\mathrm{tr}}$ // Inference BN network with frozen // parameters
8: **for** $k = 1 \ldots K$ **do**
9: // For clarity, $x \equiv x^{(k)}, \gamma \equiv \gamma^{(k)}, \mu_B \equiv \mu_B^{(k)}$, etc.
10: Process multiple training mini-batches $\mathcal{B}$, each of size m, and average over them:
$\mathrm{E}[x] \leftarrow \mathrm{E}_B[\mu_B]$
$\mathrm{Var}[x] \leftarrow \frac{m}{m-1}\mathrm{E}_B[\sigma_B^2]$
11: In $N_{\mathrm{BN}}^{\mathrm{inf}}$, replace the transform $y = \mathrm{BN}_{\gamma,\beta}(x)$ with $y = \frac{\gamma}{\sqrt{\mathrm{Var}[x]+\epsilon}} \cdot x + \left(\beta - \frac{\gamma \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x]+\epsilon}}\right)$
12: **end for**

Algorithm 2: Training a Batch-Normalized Network

3.2 Batch-Normalized Convolutional Networks

Batch Normalization can be applied to any set of activations in the network. Here, we focus on transforms

Covariate shift関係ないよ！？



How Does Batch Normalization Help Optimization? (No, It Is Not About Internal Covariate Shift)

Shibani Santurkar* MIT shibani@mit.edu
Dimitris Tsipras* MIT tsipras@mit.edu
Andrew Ilyas MIT ailyas@mit.edu
Aleksander Mądry MIT madry@mit.edu

Abstract

Batch Normalization (BatchNorm) is a widely adopted technique that enables faster and more stable training of deep neural networks (DNNs). Despite its pervasiveness, the exact reasons for BatchNorm's effectiveness are still poorly understood. The popular belief is that this effectiveness stems from controlling the change of the layers' input distributions during training to reduce the so-called "internal covariate shift". In this work, we demonstrate that such distributional stability of layer inputs has little to do with the success of BatchNorm. Instead, we uncover a more fundamental impact of BatchNorm on the training process: it makes the optimization landscape significantly smoother. This smoothness induces a more predictive and stable behavior of the gradients, allowing for faster training. These findings bring us closer to a true understanding of our DNN training toolkit.

1 Introduction

Over the last decade, deep learning has made impressive progress on a variety of notoriously difficult tasks in computer vision [13, 6], speech recognition [4], machine translation [24], and game-playing [14, 20]. This progress hinged on a number of major advances in terms of hardware, datasets [12, 18], and algorithmic and architectural techniques [22, 10, 15, 23]. One of the most prominent examples of such advances was batch normalization (BatchNorm) [8].

At a high level, BatchNorm is a technique that aims to improve training of neural networks by stabilizing the distributions of layer inputs. This is achieved by introducing additional network layers that control the first two moments (mean and variance) of these distributions.

The practical success of BatchNorm is indisputable. By now, it is used by default in most deep learning models, both in research (more than 4,000 citations) and real-world settings. Somewhat shockingly, however, despite its prominence, we still have a poor understanding of what the effectiveness of BatchNorm is stemming from. In fact, there are now a number of works that provide alternatives to BatchNorm[1, 2, 11, 26], but none of them seem to bring us any closer to understanding this issue. (A similar point was also raised recently in [17].)

Currently, the most widely accepted explanation of BatchNorm's success, as well as its original motivation, relates to so-called *internal covariate shift* (ICS). Informally, ICS refers to the change in the distribution of layer inputs caused by updates to the preceding layers. It is conjectured that such continual change negatively impacts training. The goal of BatchNorm was to reduce ICS and thus remedy this effect.

Even though this explanation is widely accepted, we seem to have little concrete evidence in its support. In particular, we still do not understand the link between ICS and training performance.

*Equal contribution.

Preprint. Work in progress.

standard, non-BatchNorm network, yet it *still performs better* in terms of training. (Figure 8 in Appendix B plots the variation in the mean and variance of the corresponding distributions.)

Clearly, these findings are hard to reconcile with the claim that the performance gain due to BatchNorm stems from increased stability of layer input distributions.

2.2 Is BatchNorm reducing internal covariate shift?

Our findings in Section 2.1 make it apparent that ICS is not directly connected to the training performance. At least if we tie ICS to stability of the mean and variance of input distributions. One might wonder, however. Is there a broader notion of internal covariate shift that *has* such a direct link to training performance? And if so, does BatchNorm indeed reduce this notion?

Recall that each layer can be seen as solving an empirical risk minimization problem where given a set of inputs, it is optimizing some loss function (that possibly involves later layers). An update to the parameters of any previous layer will change these inputs, thus changing this empirical risk minimization problem itself. This phenomenon is at the core of the intuition that Ioffe and Szegedy [8] provide regarding internal covariate shift. Specifically, they try to capture this phenomenon from the perspective of the resulting *distributional* changes in layer inputs. However, as demonstrated in Section 2.1, this perspective does not seem to properly encapsulate the roots of BatchNorm's success.

To address this issue, we attempt to capture internal covariate shift from a perspective that is more tied to the underlying optimization phenomenon. (After all the success of BatchNorm is largely of an optimization nature.) Since the training procedure is a first-order method, the gradient of the loss is the most natural object to study. To quantify the extent to which parameters in a layer would have to "adjust" in reaction to a parameter update in the previous layers, we measure the difference between the gradients of each layer before and after updates to all the previous layers. This leads to the following definition.

Definition. *Let $\mathcal{L}$ be the loss, $W_1^{(t)}, \ldots, W_k^{(t)}$ be the parameters and $(x^{(t)}, y^{(t)})$ be the batch of input-label pairs used to train the network at time t. We define* internal covariate shift (ICS) *of activation i at time t to be the difference $\|G_{t,i} - G'_{t,i}\|_2$, where*

$$G_{t,i} = \nabla_{W_i^{(t)}} \mathcal{L}(W_1^{(t)}, \ldots, W_k^{(t)}; x^{(t)}, y^{(t)})$$
$$G'_{t,i} = \nabla_{W_i^{(t)}} \mathcal{L}(W_1^{(t+1)}, \ldots, W_{i-1}^{(t+1)}, W_i^{(t)}, W_{i+1}^{(t)}, \ldots, W_k^{(t)}; x^{(t)}, y^{(t)}).$$

Here, $G_{t,i}$ corresponds to the gradient of the layer parameters that would be applied during a simultaneous update of all layers (as is typical). On the other hand, $G'_{t,i}$ is the same gradient *after* all the previous layers have been updated with their new values. The difference between G and G' thus reflects the change in the optimization landscape of W_i caused by the changes to its input. It thus captures precisely the effect of cross-layer dependencies that could be problematic for training.

Equipped with this definition, we measure the extent of ICS with and without BatchNorm layers. To account for the effect of non-linearities as well as gradient stochasticity, we also perform this analysis on (25-layer) deep linear networks (DLN) trained with full-batch gradient descent (see Appendix A for details). The conventional understanding of BatchNorm suggests that the addition of BatchNorm layers in the network should increase the correlation between G and G', thereby reducing ICS.

Surprisingly, as shown in Figure 3, we observe that networks with BatchNorm exhibit an *increase* in their ICS. This is particularly striking in the case of DLN at low learning rates. Here, the standard network experiences almost no ICS for the entirety of training, whereas for BatchNorm it appears that G and G' are almost uncorrelated. We emphasize that this is the case *even though BatchNorm networks continue to perform drastically better* in terms optimization of accuracy and loss. (The stabilization of the BatchNorm VGG network later in training is an artifact of faster convergence.)

This evidence suggests that, from optimization point of view, controlling the distributions layer inputs as done in BatchNorm, might not even reduce the internal covariate shift.

講義の予定．．．あくまで予定

1. (6/5)　イントロ
2. (6/12）「線形代数，ベクトル，内積，行列，固有値，固有ベクトル」　分散，共分散，多変量解析，回帰分析，PCA,
3. (6/19)　正準相関分析，グラフ埋め込み，深層学習（微分），SGD，ポアンカレ埋め込み，ミンコフスキー計量
4. (6/26)「確率，統計」確率モデル，リサンプリング，ブートストラップ，統計的仮説検定