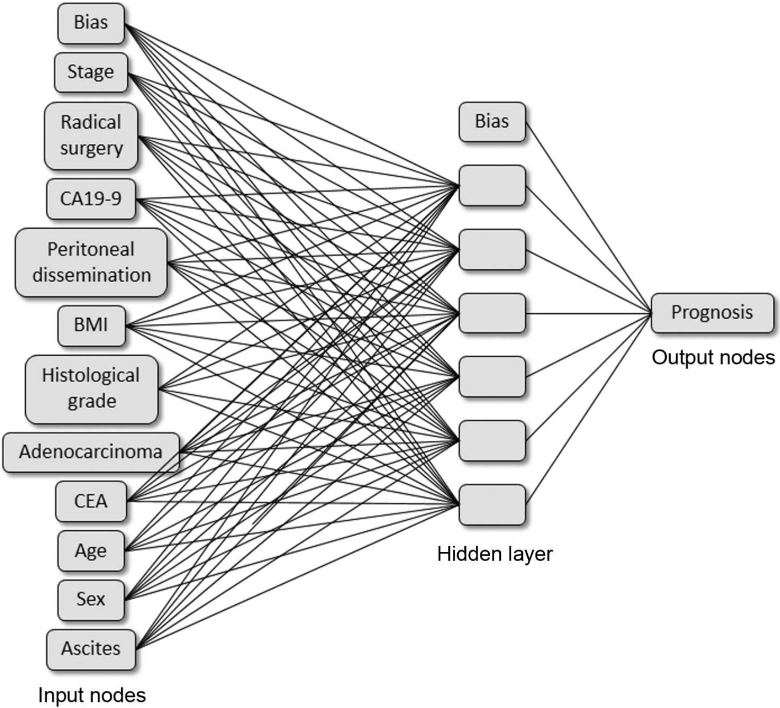
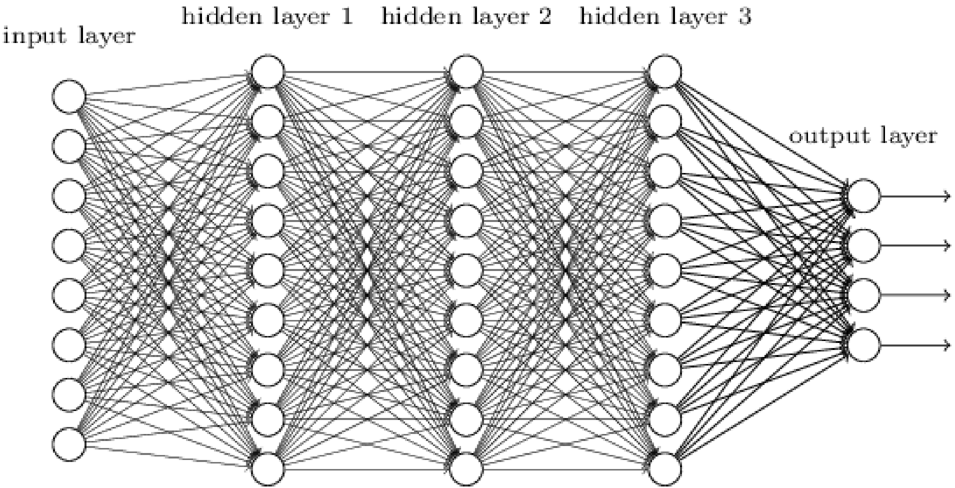
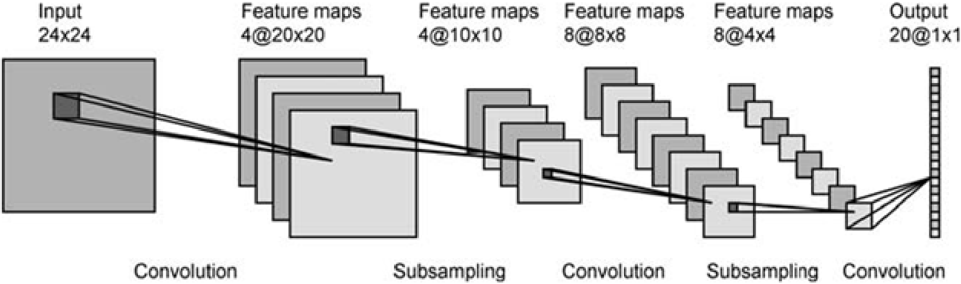
Tricks from Deep Learning

Atılım Güneş Baydin[1]
Barak A. Pearlmutter[2]
Jeffrey Mark Siskind[3]

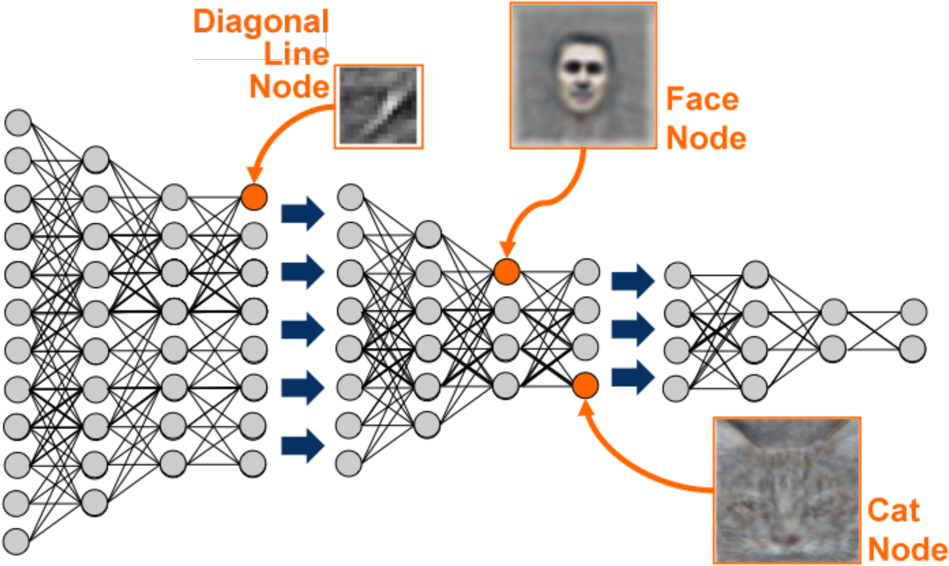
[1]Oxford University, ⟨gunes@robots.ox.ac.uk⟩
[2]Maynooth University, ⟨barak@pearlmutter.net⟩
[3]Purdue University, ⟨qobi@purdue.edu⟩


Bias
Stage
Radical surgery
CA19-9
Peritoneal dissemination
BMI
Histological grade
Adenocarcinoma
CEA
Age
Sex
Ascites
Input nodes
Bias
Hidden layer
Prognosis
Output nodes


input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer



Train: reverse AD (Speelpenning, 1980), stochastic ∇ descent (Robbins and Monro, 1951).


Diagonal Line Node
Face Node
Cat Node

Compiling Fast Partial Derivatives of Functions Given by Algorithms
(Speelpenning, 1980, PhD thesis)

vs

Learning representations by back-propagating errors
(Rumelhart et al., 1986, Nature **323**:533–6)

Compiling Fast Partial Derivatives of Functions Given by Algorithms
(Speelpenning, 1980, PhD thesis)

vs

Learning representations by back-propagating errors
(Rumelhart et al., 1986, Nature **323**:533–6)

About 166 results (**0.02** sec)

Compiling fast partial derivatives of functions given by algorithms

☐ Search within citing articles

[BOOK] Interval analysis
RE Moore - 1966 - sbras.ru
This book is intended primarily for those not yet familiar with methods for computing with intervals of real numbers and what can be done with these methods. Using a pair [a, b] of computer numbers to represent an interval of real numbers a≤ x≤ b, we define an **...**
Cited by 5782 Related articles All 6 versions Cite Saved More

[BOOK] Global optimization using interval analysis: revised and expanded
E Hansen, GW Walster - 2003 - books.google.com
Employing a closed set-theoretic foundation for interval computations, Global Optimization Using Interval Analysis simplifies algorithm construction and increases generality of interval arithmetic. This Second Edition contains an up-to-date discussion of interval methods for **...**
Cited by 2209 Related articles All 4 versions Cite Save More

[CITATION] Identification of parametric models from experimental data
E Walter, L Pronzato - 1997 - Springer Verlag
Cited by 1271 Related articles Cite Save More

Algorithm 755: ADOL-C: a package for the automatic differentiation of algorithms written in C/C++
A Griewank, D Juedes, J Utke - ACM Transactions on Mathematical …, 1996 - dl.acm.org
Abstract The C++ package ADOL-C described here facilitates the evaluation of first and higher derivatives of vector functions that are defined by computer programs written in C or C++. The resulting derivative evaluation routines may be called from C/C++, Fortran, or **...**
Cited by 846 Related articles All 9 versions Web of Science: 268 Cite Saved More

[PDF] On automatic differentiation
A Griewank - Mathematical Programming: recent developments and …, 1989 - 140.221.6.23
Abstract In comparison to symbolic di erentiation and numerical di erencing, the chain rule based technique of automatic di erentiation is shown to evaluate partial derivatives accurately and cheaply. In particular it is demonstrated that the reverse mode of automatic **...**
Cited by 728 Related articles All 10 versions Cite Save More

About 19,265 results (**0.05** sec)

Learning internal representations by error propagation

☐ Search within citing articles

[BOOK] An introduction to genetic algorithms

M Mitchell - 1998 - books.google.com

Genetic algorithms have been used in science and engineering as adaptive algorithms for solving practical problems and as computational models of natural evolutionary systems. This brief, accessible introduction describes some of the most interesting research in the **...**

Cited by 10354 Related articles All 16 versions Cite Save More

Support-vector networks

C Cortes, V Vapnik - Machine learning, 1995 - Springer

Abstract The support-vector network is a new learning machine for two-group classification problems. The machine conceptually implements the following idea: input vectors are non-linearly mapped to a very high-dimension feature space. In this feature space a linear **...**

Cited by 21971 Related articles All 44 versions Web of Science: 8327 Cite Save Mo

Unsupervised learning

T Hastie, R Tibshirani, J Friedman - The elements of statistical learning, 2009 - Springer

The previous chapters have been concerned with predicting the values of one or more outputs or response variables Y=(Y1,..., Ym) for a given set of input or predictor variables XT=(X1,..., Xp). Denote by xT i=(xi1,..., xip) the inputs for the ith training case, and let yi be **..**

Cited by 29291 Related articles All 32 versions Cite Save More

[BOOK] Pattern classification

RO Duda, PE Hart, DG Stork - 2012 - books.google.com

The first edition, published in 1973, has become a classic reference in the field. Now with the second edition, readers will find information on key new topics such as neural networks and statistical pattern recognition, the theory of machine learning, and the theory of **...**

Cited by 34317 Related articles All 24 versions Cite Saved More

Learning internal representations by error propagation

☐ Search within citing articles

Predicting the secondary structure of globular proteins using neural network models

N Qian, TJ Sejnowski - Journal of molecular biology, 1988 - Elsevier

Abstract We present a new method for predicting the secondary structure of globular proteins based on non-linear neural network models. Network models learn from existing protein structures how to predict the secondary structure of local sequences of amino **...**

Cited by 1252 Related articles All 18 versions Web of Science: 687 Cite Save More

Content-based book recommending using learning for text categorization

RJ Mooney, L Roy - Proceedings of the fifth ACM conference on Digital …, 2000 - dl.acm.org

Abstract Recommender systems improve access to relevant products and information by making personalized suggestions based on previous examples of a user's likes and dislikes. Most existing recommender systems use collaborative filtering methods that base **...**

Cited by 1272 Related articles All 30 versions Cite Save More

Detection, classification, and tracking of targets

D Li, KD Wong, YH Hu… - IEEE signal processing …, 2002 - ieeexplore.ieee.org

Networks of small, densely distributed wireless sensor nodes are being envisioned and developed for a variety of applications involving monitoring and manipulation of the physical world in a tetherless fashion [1],[16],[17],[22],[23]. Typically, each individual node can **...**

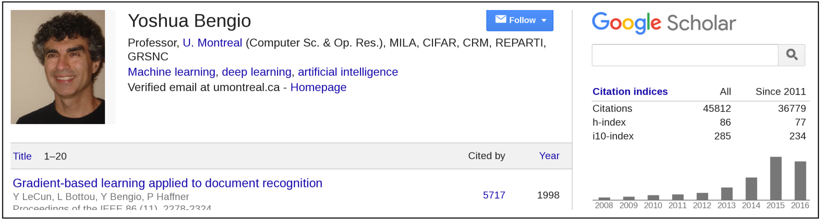
Cited by 1242 Related articles All 29 versions Web of Science: 444 Cite Save

Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights

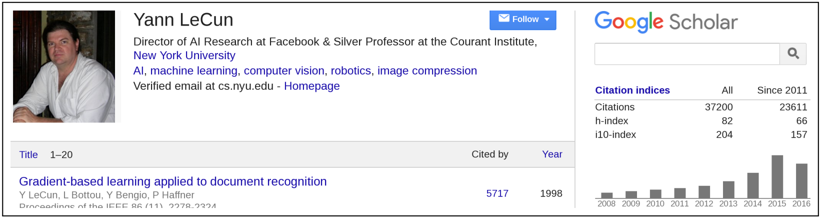
D Nguyen, B Widrow - Neural Networks, 1990., 1990 IJCNN …, 1990 - ieeexplore.ieee.org

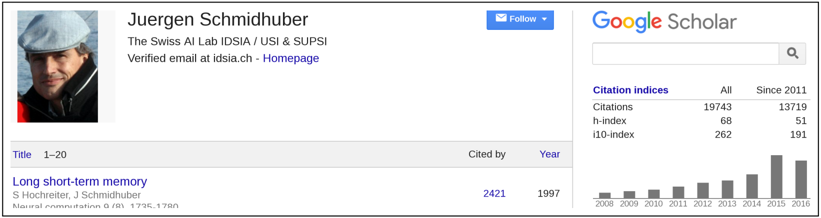
Abstract A two-layer neural network can be used to approximate any nonlinear function. The behavior of the hidden nodes that allows the network to do this is described. Networks with one input are analyzed first, and the analysis is then extended to networks with multiple **...**

Cited by 1231 Related articles All 9 versions Cite Save

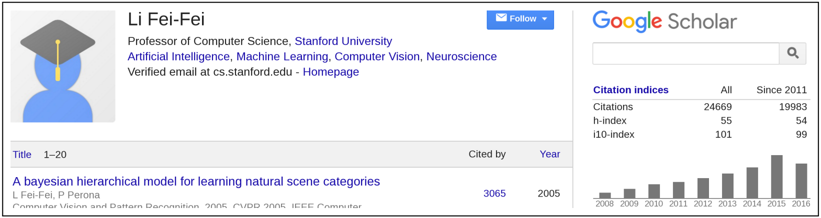

Yoshua Bengio
Follow
Professor, U. Montreal (Computer Sc. & Op. Res.), MILA, CIFAR, CRM, REPARTI, GRSNC
Machine learning, deep learning, artificial intelligence
Verified email at umontreal.ca - Homepage
Title 1–20
Cited by
Year
Gradient-based learning applied to document recognition
Y LeCun, L Bottou, Y Bengio, P Haffner
5717
1998
Google Scholar
Citation indices
All
Since 2011
Citations 45812 36779
h-index 86 77
i10-index 285 234
2008 2009 2010 2011 2012 2013 2014 2015 2016


Geoffrey Hinton
Follow
Emeritus Professor of Computer Science, University of Toronto & Distinguished Researcher, Google Inc
machine learning, neural networks, artificial intelligence, cognitive science, computer science
Verified email at cs.toronto.edu - Homepage
Title 1–20
Cited by
Year
Parallel distributed processing
DE Rumelhart, JL McClelland, PDP Research Group
IEEE 1, 354-362
20676
1988
Google Scholar
Citation indices
All
Since 2011
Citations 138909 60171
h-index 122 89
i10-index 288 205
2008 2009 2010 2011 2012 2013 2014 2015 2016

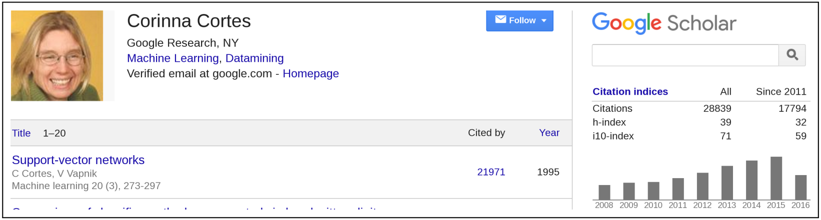

Yann LeCun
Follow
Director of AI Research at Facebook & Silver Professor at the Courant Institute, New York University
AI, machine learning, computer vision, robotics, image compression
Verified email at cs.nyu.edu - Homepage
Title 1–20
Cited by
Year
Gradient-based learning applied to document recognition
Y LeCun, L Bottou, Y Bengio, P Haffner
5717
1998
Google Scholar
Citation indices
All
Since 2011
Citations 37200 23611
h-index 82 66
i10-index 204 157
2008 2009 2010 2011 2012 2013 2014 2015 2016



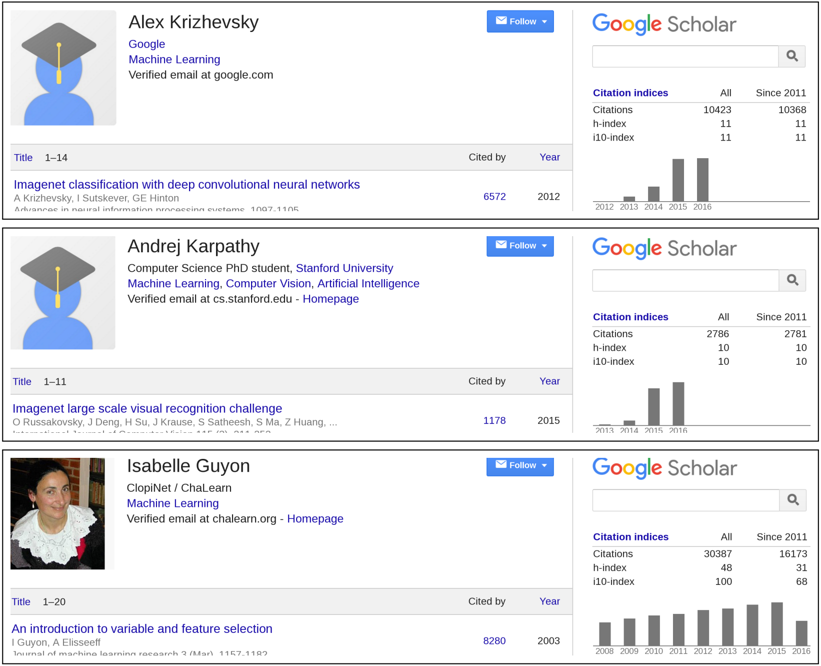
Juergen Schmidhuber
Follow
The Swiss AI Lab IDSIA / USI & SUPSI
Verified email at idsia.ch - Homepage
Google Scholar
Citation indices All Since 2011
Citations 19743 13719
h-index 68 51
i10-index 262 191
Title 1–20 Cited by Year
Long short-term memory
S Hochreiter, J Schmidhuber
Neural computation 9 (8), 1735-1780
2421 1997
2008 2009 2010 2011 2012 2013 2014 2015 2016



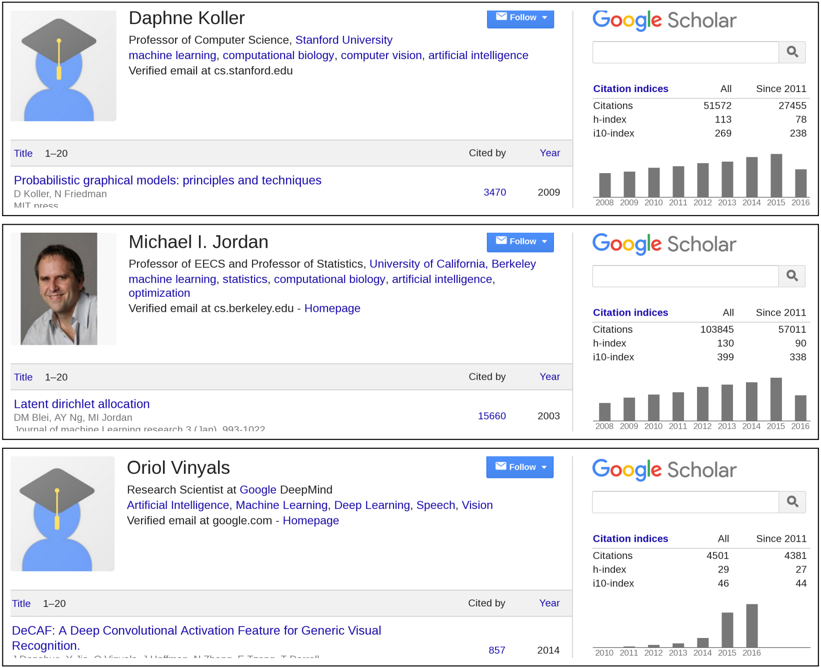
Li Fei-Fei
Follow
Professor of Computer Science, Stanford University
Artificial Intelligence, Machine Learning, Computer Vision, Neuroscience
Verified email at cs.stanford.edu - Homepage
Google Scholar
Citation indices All Since 2011
Citations 24669 19983
h-index 55 54
i10-index 101 99
Title 1–20 Cited by Year
A bayesian hierarchical model for learning natural scene categories
L Fei-Fei, P Perona
Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer
3065 2005
2008 2009 2010 2011 2012 2013 2014 2015 2016



Corinna Cortes
Follow
Google Research, NY
Machine Learning, Datamining
Verified email at google.com - Homepage
Google Scholar
Citation indices All Since 2011
Citations 28839 17794
h-index 39 32
i10-index 71 59
Title 1–20 Cited by Year
Support-vector networks
C Cortes, V Vapnik
Machine learning 20 (3), 273-297
21971 1995
2008 2009 2010 2011 2012 2013 2014 2015 2016



Alex Krizhevsky
Follow
Google
Machine Learning
Verified email at google.com
Google Scholar
Citation indices All Since 2011
Citations 10423 10368
h-index 11 11
i10-index 11 11
2012 2013 2014 2015 2016
Title 1–14 Cited by Year
Imagenet classification with deep convolutional neural networks
A Krizhevsky, I Sutskever, GE Hinton
6572 2012
Andrej Karpathy
Follow
Computer Science PhD student, Stanford University
Machine Learning, Computer Vision, Artificial Intelligence
Verified email at cs.stanford.edu - Homepage
Google Scholar
Citation indices All Since 2011
Citations 2786 2781
h-index 10 10
i10-index 10 10
Title 1–11 Cited by Year
Imagenet large scale visual recognition challenge
O Russakovsky, J Deng, H Su, J Krause, S Satheesh, S Ma, Z Huang, ...
1178 2015
Isabelle Guyon
Follow
ClopiNet / ChaLearn
Machine Learning
Verified email at chalearn.org - Homepage
Google Scholar
Citation indices All Since 2011
Citations 30387 16173
h-index 48 31
i10-index 100 68
2008 2009 2010 2011 2012 2013 2014 2015 2016
Title 1–20 Cited by Year
An introduction to variable and feature selection
I Guyon, A Elisseeff
8280 2003


Daphne Koller
Follow
Professor of Computer Science, Stanford University
machine learning, computational biology, computer vision, artificial intelligence
Verified email at cs.stanford.edu
Title 1–20
Cited by
Year
Probabilistic graphical models: principles and techniques
D Koller, N Friedman
3470
2009
Google Scholar
Citation indices
All
Since 2011
Citations 51572 27455
h-index 113 78
i10-index 269 238
2008 2009 2010 2011 2012 2013 2014 2015 2016
Michael I. Jordan
Follow
Professor of EECS and Professor of Statistics, University of California, Berkeley
machine learning, statistics, computational biology, artificial intelligence, optimization
Verified email at cs.berkeley.edu - Homepage
Title 1–20
Cited by
Year
Latent dirichlet allocation
DM Blei, AY Ng, MI Jordan
15660
2003
Google Scholar
Citation indices
All
Since 2011
Citations 103845 57011
h-index 130 90
i10-index 399 338
2008 2009 2010 2011 2012 2013 2014 2015 2016
Oriol Vinyals
Follow
Research Scientist at Google DeepMind
Artificial Intelligence, Machine Learning, Deep Learning, Speech, Vision
Verified email at google.com - Homepage
Title 1–20
Cited by
Year
DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition.
857
2014
Google Scholar
Citation indices
All
Since 2011
Citations 4501 4381
h-index 29 27
i10-index 46 44
2010 2011 2012 2013 2014 2015 2016


Tianjin Mayor Caught Up in Xi's Antigraft Campaign


NEWS ANALYSIS
Few Expect China to Punish North Korea for Latest Nuclear Test

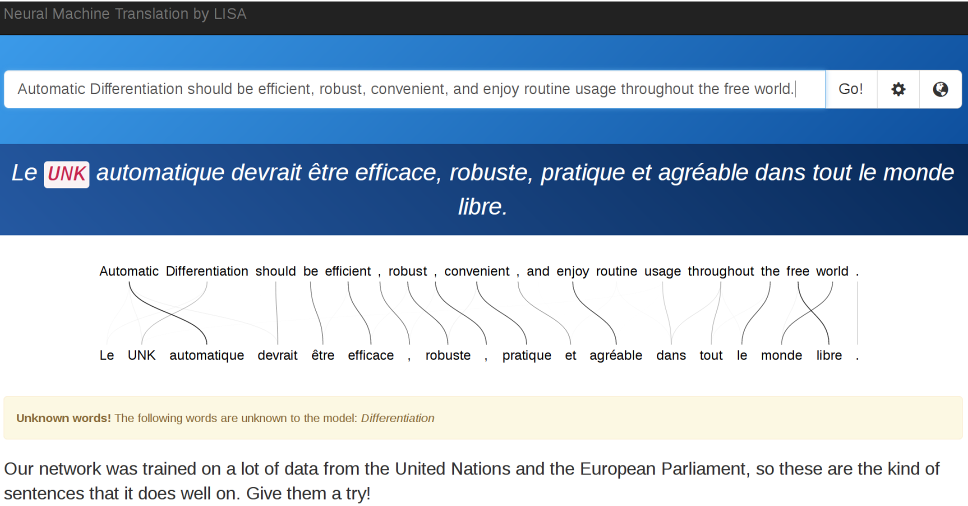

Boiler Explosion at Bangladesh Factory Kills at Least 23


THE INTERPRETER
North Korea, Far From Crazy, Is All Too Rational

ASIA PACIFIC

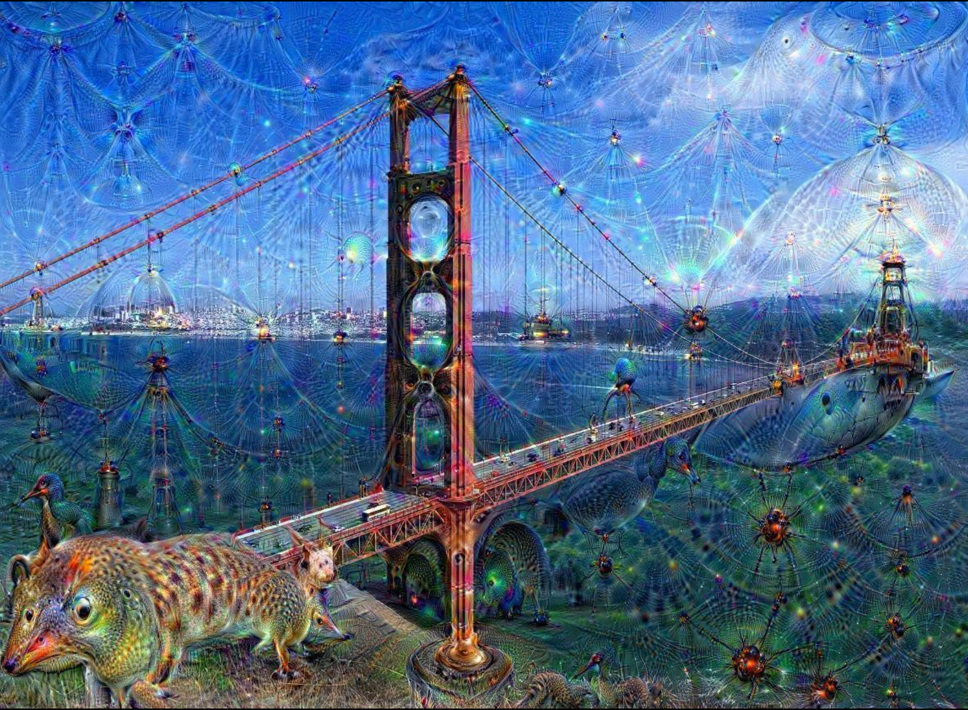
Google's Computer Program Beats Lee Se-dol in Go Tournament

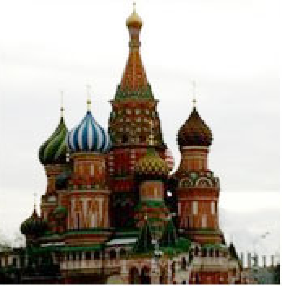
By CHOE SANG-HUN MARCH 15, 2016

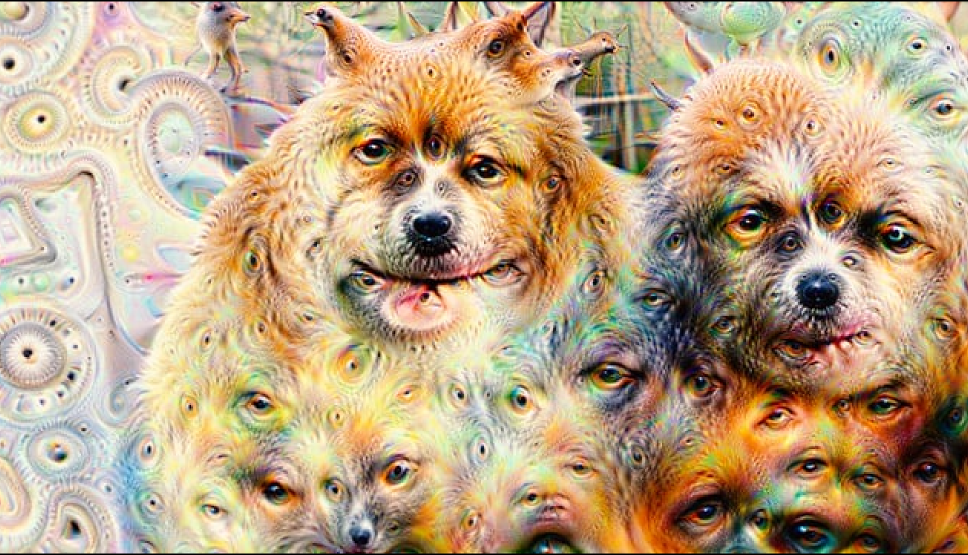


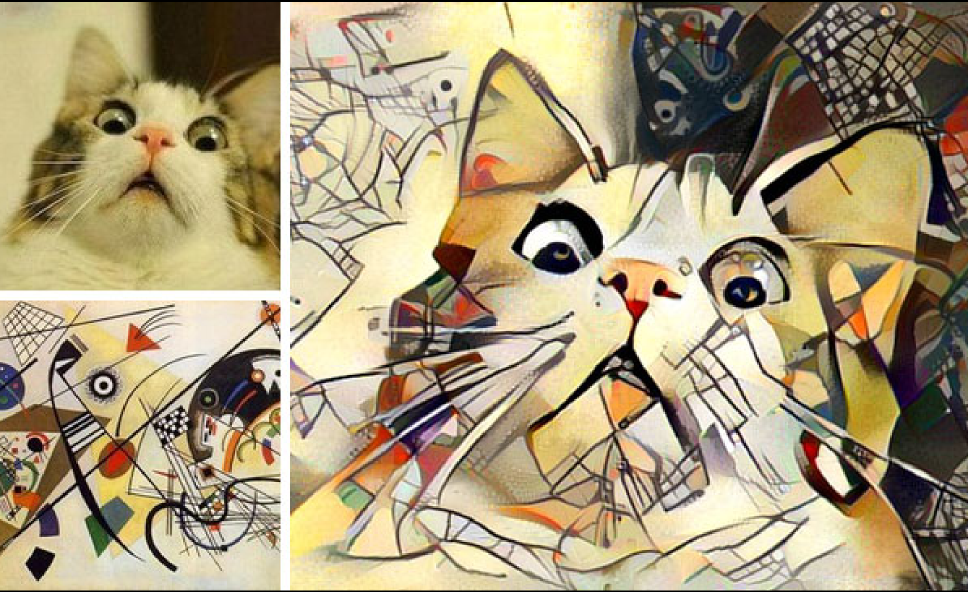
(see http://lisa.iro.umontreal.ca/mt-demo)


Image: Google Research Blog









big programs *vs* big data

some math to take home

Stochastic Gradient Descent (Robbins and Monro, 1951)

Define the stochastic hat: $\mathbf{E}[\,\hat{\cdot}\,] = \cdot$

Stochastic Gradient Descent (Robbins and Monro, 1951)

Define the stochastic hat: $\mathbf{E}[\,\hat{\cdot}\,] = \cdot$

Find the minimizer w^* of

$$E(w) = \frac{1}{n}\sum_{i=1}^{n} E_i(w)$$

by iterating

$$w(t+1) = w(t) - \eta_t \nabla_w \widehat{E}(w(t)) = w(t) - \eta_t \nabla_w E_i(w(t))$$

where i is chosen randomly and $\eta_t > 0$ is gradually decreased, $O(t^{-2}) < \eta_t \leq O(t^{-1})$.

Stochastic Gradient Descent (Robbins and Monro, 1951)

Define the stochastic hat: $\mathbf{E}[\,\hat{\cdot}\,] = \cdot$

Find the minimizer w^* of

$$E(w) = \frac{1}{n}\sum_{i=1}^{n} E_i(w)$$

by iterating

$$w(t+1) = w(t) - \eta_t \nabla_w \widehat{E}(w(t)) = w(t) - \eta_t \nabla_w E_i(w(t))$$

where i is chosen randomly and $\eta_t > 0$ is gradually decreased, $O(t^{-2}) < \eta_t \leq O(t^{-1})$.

Nowadays: choose i sequentially, mini-batches, $\eta_t = O(1)$, momentum.

Stochastic Hessian-Vector Product

Stochastic Hessian-Vector Product

(same as stochastic gradient: sample data)

Stochastic Hessian-Vector Product

(same as stochastic gradient: sample data)

$$\widehat{H}v = \nabla^2 \widehat{E} v$$

Stochastic Hessian-Inverse-Vector Product

Stochastic Hessian-Inverse-Vector Product

$$H^{-1} = \sum_{i=0}^{\infty} (I - H)^i$$

choose $i \sim p(i)$ and

$$\widehat{H^{-1}}v = p(i)^{-1} \overbrace{(I - \widehat{H})(\cdots((I - \widehat{H})}^{i \text{ times}} v))$$

(Assume in radius of convergence. See Agarwal et al., 2016)

Reversible Learning

Desire: perform reverse AD on $f : x \mapsto y(t_F)$ where $y(t_f)$ is the result of numeric integration of an ode with time-dependent x-dependent driving term,

$$y(t + \Delta t) = y(t) + \Delta t\, g(t, y(t), x)$$

Naïve reverse AD: store $y(t_0), y(t_0 + \Delta t), \ldots, t(t_F)$.

Idea: this ode is time reversible; run it backwards during reverse pass.

Problems: (a) the ode is stable, so its time-reversal is unstable;
(b) $\Delta t > 0$ and limited precision cause time reversal to diverge from primal.

Solution: store info during primal to correct time-reversal so it never diverges from primal (Maclaurin et al., 2015).

Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)

DrMAD: Distilling Reverse-Mode Automatic Differentiation for Optimizing Hyperparameters of Deep Neural Networks

Jie Fu[*] **Hongyin Luo**[ˆ] **Jiashi Feng**[*] **Kian Hsiang Low**[*] **Tat-Seng Chua**[*]
[*] National University of Singapore, Singapore
[ˆ] Tsinghua University, China

Abstract

The performance of deep neural networks is well-known to be sensitive to the setting of their hyperparameters. Recent advances in reverse-mode automatic differentiation allow for optimizing hyperparameters with gradients. The standard way of computing these gradients involves a forward and backward pass of computations. However, the backward pass usually needs to consume unaffordable memory to store all the intermediate variables to *exactly* reverse the forward training procedure. In this work we propose a simple but effective method, DrMAD, to distill the knowledge of the forward pass into a shortcut path, through which we *approximately* reverse the training trajectory. Experiments on two image benchmark datasets show that DrMAD is at least 45 times faster and consumes 100 times less memory compared to state-

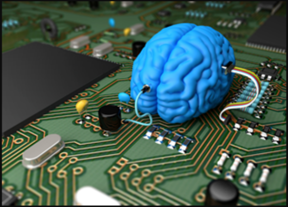
deep neural networks in a variety of benchmark datasets [Shahriari *et al.*, 2016; Snoek *et al.*, 2012].

A common choice for hyperparameter optimization is gradient-free Bayesian optimization [Wang *et al.*, 2013]. Bayesian optimization builds a probability model to describe the distribution of validation loss conditioned on specific hyperparameters, which is obtained by multiple observations over the pairs of hyperparameter and validation loss. This probability model is then used to optimize the validation loss after complete training of the model's elementary[1] parameters. Although those techniques have been shown to achieve good performance with a variety of models on benchmark datasets [Shahriari *et al.*, 2016], they can hardly scale up to handle more than 20 hyperparameters [Maclaurin *et al.*, 2015; Shahriari *et al.*, 2016]. Here we mean *effective* hyperparameters. It has been shown in [Wang *et al.*, 2013] that Bayesian optimization can handle high-dimensional inputs only if the number of effective hyperparameters is small. Due to this inability, hyperparameters are often considered nui-

deep learning tools

Deep Learning

What society thinks I do



What my friends think I do

What other computer scientists think I do

What mathematicians think I do

What I think I do

```
from theano import *
```

What I actually do

What mathematicians think I do	What I think I do	What I actually do

Theano is a Python library for efficiently handling mathematical expressions involving multi-dimensional arrays (also known as tensors). It is a common choice for implementing neural network models. Theano has been developed in University of Montreal, in a group led by Yoshua Bengio, since 2008.

Some of the features include:

- automatic differentiation – you only have to implement the forward (prediction) part of the model, and Theano will automatically figure out how to calculate the gradients at various points, allowing you to perform gradient descent for model training.
- transparent use of a GPU – you can write the same code and run it either on CPU or GPU. More specifically, Theano will figure out which parts of the computation should be moved to the GPU.
- speed and stability optimisations – Theano will internally reorganise and optimise your computations in order to make them run faster and be more numerically stable. It will also try to compile some operations into C code, in order to speed up the computation.

Technically, Theano isn't actually a machine learning library, as it doesn't provide you with pre-built models that you can train on your dataset. Instead, it is a mathematical library that provides you with

(from *What is Theano?* http://www.marekrei.com/blog/theano-tutorial/)

Gradient-based Optimization of MLP in Lua/Torch7

(runs on real FMRI data)

```lua
function classify_net(number_of_voxels, number_of_classes)
   local mlp = nn.Sequential()
   mlp:add(nn.Dropout(0.5))
   mlp:add(nn.Linear(number_of_voxels, number_of_voxels))
   mlp:add(nn.ReLU(true))
   mlp:add(nn.Dropout(0.5))
   mlp:add(nn.Linear(number_of_voxels, number_of_classes))
   mlp:add(nn.LogSoftMax())
   return mlp
end

function train_classifier(net, criterion, iterations, learning_rate,
                          training_set)
   for i = 1, iterations do
      net:training()
      local class = net:forward(training_set.fmri)
      criterion:forward(class, training_set.labels)
      net:zeroGradParameters()
      net:backward(fmri, criterion:backward(class, training_set.labels))
      net:updateParameters(learning_rate)
   end
end
```

Gradient-based Optimization of MLP in Julia/ReverseDiffSource

```
function classify_net(d1, d2, w1, w2, x1)
   x1 = d1.*x1
   x2 = w1*x1
   x2 = log1p(exp(x2))
   x2 = d2.*x2
   x3 = w2*x2
   x3 = log1p(exp(x3))
   log(1/(sum(exp(x))*exp(x)))
end

function train_classifier(criterion, iterations, learning_rate, training_set)
   d1, d2 = 0.5*rand(4096), 0.5*rand(4096)
   w1, w2 = randn(4096, 4096), randn(4096, 12)
   dnet = rdiff(classify_net, (d1, d2, w1, w2, training_set))
   for i in 1:iterations
       d1, d2 = 0.5*rand(4096), 0.5*rand(4096)
       _, _, _, dedw1, dedw2, _ = dnet(d1, d2, w1, w2, training_set)
       w1 = w1-learning_rate*dedw1
       w2 = w2-learning_rate*dedw2
   end
end
```

Gradient-based Optimization of MLP in Python/autograd

```
def neural_net_predict(parameters, inputs):
    for W, b in parameters:
        outputs = np.dot(inputs, W) + b
        inputs = np.tanh(outputs)
    return outputs - logsumexp(outputs, axis=1, keepdims=True)

def l2_norm(parameters):
    flattened, _ = flatten(parameters)
    return np.dot(flattened, flattened)

def log_posterior(parameters, inputs, targets):
    log_prior = -l2_norm(parameters)
    log_likelihood = np.sum(neural_net_predict(parameters, inputs) * targets)
    return log_prior + log_likelihood

def objective(parameters, iter):
    return -log_posterior(parameters, training_set, training_labels)

optimized_parameters = adam(grad(objective), init_parameters,
                            step_size=step_size,
                            num_iters=num_epochs * num_batches)
```

https://github.com/BVLC/caffe/issues

This repository | Search

Pull requests Issues Gist

BVLC / caffe

Watch 1,566 | Star 12,417 | Fork 7,552

Code | Issues 497 | Pull requests 259 | Wiki | Pulse | Graphs

Filters | is:issue is:open | Labels | Milestones | New issue

497 Open ✓ 2,507 Closed — Author | Labels | Milestones | Assignee | Sort

large iter_size and small batch_size hurts batch normalization
#4716 opened 5 hours ago by KeyKy

Python module doesn't appear to initialise google logging
#4715 opened 9 hours ago by ferrouswheel

Caffe make all issue on ubuntu 16.04 — 1
#4714 opened 13 hours ago by balajithoshkahna

Can't cast the same trained model to net on different machine
#4713 opened 17 hours ago by xxw345

Obtaining the amount of required top/bot blobs for a layer.
#4712 opened 18 hours ago by TimZaman

Training halted when using multiple GPUs and skipping LMDB entries
#4711 opened a day ago by shihenw

https://github.com/Theano/Theano/issues

This repository Search

Pull requests Issues Gist

Theano / Theano

Watch 430 | Star 4,500 | Fork 1,620

Code | Issues 627 | Pull requests 116 | Wiki | Pulse | Graphs

Filters | is:issue is:open | Labels | Milestones | New issue

627 Open ✓ 1,122 Closed | Author | Labels | Milestones | Assignee | Sort

theano error after upgrade of ipython. Cannot run test theano.test() 6
#4955 opened 3 days ago by kirk86

User interface change: pool_2d rename parameter ds to ws ccw Interface
#4933 opened 10 days ago by nouiz 0.9

cuDNN batch norm does not support 5D input GPU GPU - New back-end 1
#4931 opened 10 days ago by mdering

"sum" pooling crashes when being replaced by cuDNN op ccw Crash GPU GPU - New back-end Python Code Only 5
#4929 opened 11 days ago by lamblin

Naming of third dimension for 3D convolution/pooling 2
#4928 opened 11 days ago by gvtulder

error: '::hypot' has not been declared when compiling with MingGW64 3
#4926 opened 11 days ago by davikrehalt

https://github.com/tensorflow/tensorflow/issues

This repository | Search | Pull requests | Issues | Gist

tensorflow / **tensorflow**

Watch 2,997 | Star 31,703 | Fork 13,565

Code | Issues 503 | Pull requests 35 | Pulse | Graphs

Filters | is:issue is:open | Labels | Milestones | New issue

503 Open ✓ 2,293 Closed — Author | Labels | Milestones | Assignee | Sort

configure issues
#4335 opened an hour ago by fayeshine

dynamic_rnn() cannot receive an input of dynamic shape during training
#4333 opened 3 hours ago by lan2720

Tensor with inconsistent dimension size?
#4332 opened 6 hours ago by BichenWuUCB

Problem with using tf.contrib.metrics.streaming_mean_iou
#4331 opened 6 hours ago by ziky90

overparametrized convolution error in tf.nn.separable_conv2d
#4330 opened 9 hours ago by Jongchan

typo in tutorials/mnist/beginners/index.html
#4329 opened 10 hours ago by infotek

JS] | https://github.com/torch/torch7/issues

This repository | Search | Pull requests | Issues | Gist

torch / **torch7**

Watch 604 | Star 5,358 | Fork 1,541

Code | Issues 70 | Pull requests 7 | Wiki | Pulse | Graphs

Filters | is:issue is:open | Labels | Milestones | New issue

70 Open ✓ 286 Closed — Author | Labels | Milestones | Assignee | Sort

Compilation issues on Nvidia TK1
#763 opened 2 hours ago by SergeyMalashenko

[Port] Portability problem introduced by THVector.c — 2
#762 opened 2 days ago by CDLuminate

Handling Variable length sequences in training an RNN — 1
#759 opened 3 days ago by mineshmathew

illegal memory access was encountered
#752 opened 12 days ago by sunshineatnoon

test stuck at GPU() — 1
#750 opened 15 days ago by zawlin

[bug] torch.test() fails when the second time calling — 1
#749 opened 18 days ago by CDLuminate

S] | https://github.com/HIPS/autograd/issues

This repository | Search | Pull requests | Issues | Gist

HIPS / **autograd**

Watch 97 | Unstar 1,010 | Fork 128

Code | Issues 18 | Pull requests 2 | Wiki | Pulse | Graphs

Filters | is:issue is:open | Labels | Milestones | New issue

18 Open ✓ 80 Closed | Author | Labels | Milestones | Assignee | Sort

value_and_jacobian
#145 opened 14 days ago by ambushed

hessian + flatten_func: singleton dimension
#142 opened 26 days ago by wOOL

median is not implemented
#139 opened 29 days ago by yukoba

Sharing state between forward and reverse computation enhancement — 3
#129 opened on Jul 7 by timvieira

Linear systems using cholesky decomp — 1
#125 opened on Jun 28 by Bonnevie

einsum doesn't handle tracing over indices bug — 1
#115 opened on Jun 3 by mattjj

https://github.com/ekmett/ad/issues

This repository | Search

Pull requests Issues Gist

ekmett / ad

Unwatch 19 | Unstar 146 | Fork 31

Code | Issues 5 | Pull requests 1 | Pulse | Graphs

Filters | is:issue is:open | Labels | Milestones | New issue

5 Open ✓ 30 Closed — Author — Labels — Milestones — Assignee — Sort

Unexpected difference in behavior of `grad` for different modes? 8
#54 opened on Apr 27 by mstksg

Gradient descent fails to converge (to the right result). 7
#45 opened on Jul 11, 2015 by echatav

du failing on simple function bikeshedding limitation 12
#38 opened on Sep 12, 2014 by barak

Generate Specialized Code for known objective functions (recording prior dev discussion on this topic) improvement performance question
#25 opened on May 6, 2013 by cartazio

Vectored AD improvement major refactoring
#2 opened on Jun 26, 2010 by ekmett

S] | https://github.com/twitter/torch-autograd

This repository | Search | Pull requests | Issues | Gist

twitter / torch-autograd

Watch 42 | Star 423 | Fork 74

Code | Issues 17 | Pull requests 1 | Wiki | Pulse | Graphs

Autograd automatically differentiates native Torch code

585 commits | 17 branches | 0 releases | 21 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

alexbw committed on GitHub Update autograd-scm-1.rockspec — Latest commit a357155 2 days ago

benchmark	Better errors for unsupported operations.	8 months ago
doc	Doc.	10 months ago
examples	Source code printing enabled (#138)	2 months ago
src	remove globals (#147)	28 days ago
test	Fix gradients of sinh and cosh (#145)	a month ago
.gitignore	Reorganize codegen.	9 months ago
.travis.yml	Remove totem dependency (torch now has its own tester) (#124)	4 months ago
CMakeLists.txt	Modules for Autograd (#146)	a month ago
LICENSE	License touch up.	10 months ago

Autograd

slack 2/76

build passing

Autograd automatically differentiates native Torch code. Inspired by the original Python version.

Scope

Autograd has multiple goals:

- provide automatic differentiation of Torch expressions
- support arbitrary Torch types (e.g. transparent and full support for CUDA-backed computations)
- full integration with nn modules: mix and match auto-differentiation with user-provided gradients
- the ability to define any new nn compliant Module with automatic differentiation
- represent complex evaluation graphs, which is very useful to describe models with multiple loss functions and/or inputs
- graphs are dynamic, i.e. can be different at each function call: for loops, or conditional, can depend on intermediate results, or on input parameters
- enable gradients of gradients for transparent computation of Hessians

Updates

Black-Box Stochastic Variational Inference in Five Lines of Python

David Duvenaud
dduvenaud@seas.harvard.edu
Harvard University

Ryan P. Adams
rpa@seas.harvard.edu
Harvard University

Abstract

Several large software engineering projects have been undertaken to support black-box inference methods. In contrast, we emphasize how easy it is to construct scalable and easy-to-use automatic inference methods using only automatic differentiation. We present a small function which computes stochastic gradients of the evidence lower bound for any differentiable posterior. As an example, we perform stochastic variational inference in a deep Bayesian neural network.

Black-box stochastic variational inference in five lines of Python

(Duvenaud and Adams, 2015, NIPS Workshop on Black-box Learning and Inference)

```python
def lower_bound(variational_params, logprob_func, D, num_samples):
    # variational_params: the mean and covariance of approximate posterior.
    # logprob_func:       the unnormalized log-probability of the model.
    # D:                  the number of parameters in the model.
    # num_samples:        the number of Monte Carlo samples to use.

    # Unpack mean and covariance of diagonal Gaussian.
    mu, cov = variational_params[:D], np.exp(variational_params[D:])

    # Sample from multivariate normal using the reparameterization trick.
    samples = npr.randn(num_samples, D) * np.sqrt(cov) + mu

    # Lower bound is the exact entropy plus a Monte Carlo estimate of energy.
    return mvn.entropy(mu, np.diag(cov)) + np.mean(logprob(samples))

# Get gradient with respect to variational params using autograd.
gradient = grad(lower_bound)
```

Figure 1: Black-box stochastic variational inference in five lines of Python, using automatic differentiation. The variational objective gradient can be used with any stochastic-gradient-based optimizer.

Black-box stochastic variational inference in five lines of Python

(Duvenaud and Adams, 2015, NIPS Workshop on Black-box Learning and Inference)

```
def lower_bound(variational_params, logprob_func, D, num_samples):
    # variational_params: the mean and covariance of approximate posterior.
    # logprob_func:       the unnormalized log-probability of the model.
    # D:                  the number of parameters in the model.
    # num_samples:        the number of Monte Carlo samples to use.

    # Unpack mean and covariance of diagonal Gaussian.
    mu, cov = variational_params[:D], np.exp(variational_params[D:])

    # Sample from multivariate normal using the reparameterization trick.
    samples = npr.randn(num_samples, D) * np.sqrt(cov) + mu

    # Lower bound is the exact entropy plus a Monte Carlo estimate of energy.
    return mvn.entropy(mu, np.diag(cov)) + np.mean(logprob(samples))

# Get gradient with respect to variational params using autograd.
gradient = grad(lower_bound)
```

Figure 1: Black-box stochastic variational inference in five lines of Python, using automatic differentiation. The variational objective gradient can be used with any stochastic-gradient-based optimizer.

Black-box stochastic variational inference in five lines of Python

(Duvenaud and Adams, 2015, NIPS Workshop on Black-box Learning and Inference)

The Machine Learning community *cares* about the AD API.

JUN 17, 2016

Automatic differentiation for machine learning in Julia

Automatic differentiation is a term I first heard of while working on (as it turns out now, a bit cumbersome) implementation of backpropagation algorithm – after all it caused lots of headaches as I had to handle all derivatives myself with almost pen-and-paper like approach. Obviously I made many mistakes until I got my final solution working.

At that time, I was aware some libraries like Theano or Tensorflow handle derivatives in a certain “magical” way for free. I never knew exactly what happens deep in the guts of these libraries though and I somehow suspected it is rather painful than fun to grasp (apparently, I was wrong!).

I decided to take a shot and directed my first steps towards TensorFlow official documentation to quickly find out what the magic is. The term I was looking for was **automatic differentiation**.

Contents [hide]

Justin Domke's Weblog

← Hessian-Vector products The Stalin Compiler →

Automatic Differentiation: The most criminally underused tool in the potential machine learning toolbox?

Posted on February 17, 2009

> *Update: (November 2015) In the almost seven years since writing this, there has been an explosion of great tools for automatic differentiation and a corresponding upsurge in its use. Thus, happily, this post is more or less obsolete.*

I recently got back reviews of a paper in which I used automatic differentiation. Therein, a reviewer clearly thought I was using finite difference, or "numerical" differentiation. This has led me to wondering: **Why don't machine learning people use automatic differentiation more? Why don't they use it…constantly?** Before recklessly speculating on the answer, let me

Archives

- September 2015
- December 2014
- February 2014
- January 2014
- September 2013
- September 2012
- January 2012
- November 2011
- October 2011
- July 2011
- May 2011
- March 2011
- November 2009
- October 2009
- August 2009

Quora Ask or Search Quora Ask Question

Why isn't automatic differentiation more widely used in the machine learning community?

Request ▲ Follow 4 Comment Share Downvote •••

Have this question too? Request Answers:

Request From Quora
We will distribute this question to writers, and notify you about new answers.

Dsoul, Stanford AI
5 Answers in Machine Learning

Yoshua Bengio, Head of Montreal Institute for Learning Algorithms, Professor @ …
Most Viewed Writer in Machine Learning with 110 Answers

Govind Narasimman, techie
3 Answers in Machine Learning

View More or Search

Can you answer this question?

Answer

https://www.quora.com/Why-isnt-automatic-differentiation-more-widely-used-in-the-machine-learning-community

Why isn't automatic differentiation more widely used in the machine learning community?

1. the ML community doesn't use Fortran
   - it doesn't even use C much anymore, and hasn't for a while
   - it never used C++ or Java very much
   - it uses Matlab a lot, and has for a long time
   - it uses Python a lot
2. it typically doesn't write big programs
   - instead it writes small programs that express complicated mathematical ideas (nuggets) in a dense fashion
   - relies a lot on reuse through libraries widely used in the community
3. it doesn't need to support legacy code
4. it needs to be nimble
   - code written by individuals and small teams
   - but shared a lot with other groups that fork the code and build on it can't tolerate large build infrastructures that come with preprocessors
5. it processes huge amounts of data
6. recently, it relies heavily on GPUs
7. it uses idioms that are conveniently formulated with data-parallel constructs

Deep Learning Tool Uptake Requirements

In order for the Deep Learning community to embrace a tool it must provide them:

- AD
- High speed on large datasets
- Expressive power for algorithms of interest
- High speed on large datasets
- Open source
- High speed on large datasets

Baidu open-sourced
paddlepaddle
on 12-Sep-2016

https://github.com/baidu/Paddle/issues

This repository | Search | Pull requests | Issues | Gist

baidu / **Paddle**

Watch 272 | Star 2,970 | Fork 610

Code | Issues 21 | Pull requests 4 | Wiki | Pulse | Graphs

Filters | is:issue is:open | Labels | Milestones | New issue

21 Open ✓ 27 Closed | Author | Labels | Milestones | Assignee | Sort

How to use CTC loss function?
#68 opened 13 hours ago by F0REacH

gcc 6 ? (for archlinux + python2)
#67 opened 16 hours ago by fcellier

demo/sentiment$ ./train.sh error 2
#64 opened 2 days ago by hdulbj

run vgg_16_cifar wrong. 2
#63 opened 3 days ago by zhuyong0000

Cudnn Batch-normalization not working with cudnn-v4 1
#60 opened 3 days ago by zchen0211

Not defined error! 1
#57 opened 4 days ago by prakhar21

Take-Home Message

- The AD community has spent a lot of time developing
  - sophisticated methods for taking gradients (forward mode, reverse mode, checkpointing, ...)
  - that apply to *programs*
- The ML community has spent a lot of time developing
  - specialized methods for taking gradients
  - that apply to *neural nets* (feed forward data-flow graphs constructed out of a small number of node types)
  - but has made them *fast* (GPU implementations)
  - to run on *huge* datasets (ImageNet, MSCOCO, Sports1M, ...)
- The AD community is *tiny* ($\approx$50)
- The ML community is *huge* ($\approx$15000)
- The ML community is rediscovering AD
- There is a window of opportunity for the AD community to have impact in the ML community

ML and AD

opportunity for ideas
to flow
back and forth

Good News

&

Bad News

EOT

References I

N. Agarwal, B. Bullins, and E. Hazan. Second order stochastic optimization in linear time. *arXiv:1602.03943*, 2016.

D. Duvenaud and R. P. Adams. Black-box stochastic variational inference in five lines of Python. NIPS Workshop on Black-box Learning and Inference, 2015. URL http://www.cs.toronto.edu/~duvenaud/papers/blackbox.pdf.

J. Fu, H. Luo, J. Feng, K. H. Low, and T.-S. Chua. DrMAD: Distilling reverse-mode automatic differentiation for optimizing hyperparameters of deep neural networks. In *IJCAI*, 2016.

D. Maclaurin, D. Duvenaud, and R. P. Adams. Gradient-based hyperparameter optimization through reversible learning. *arXiv:1502.03492*, 2015.

MSCS Editorial Board. Editors' note: bibliometrics and the curators of orthodoxy. *Mathematical Structures in Computer Science*, 19(1):1–4, Feb. 2009. doi: 10.1017/S0960129508007391.

H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–7, 1951. URL https://projecteuclid.org/euclid.aoms/1177729586.

References II

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–6, 1986.

B. Speelpenning. *Compiling Fast Partial Derivatives of Functions Given by Algorithms*. PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, Jan. 1980.

(they also care about nesting)

Math. Struct. in Comp. Science (2009), *vol.* 19, *pp.* 1–4. © 2009 Cambridge University Press
doi:10.1017/S0960129508007391 First published online 19 January 2009 Printed in the United Kingdom

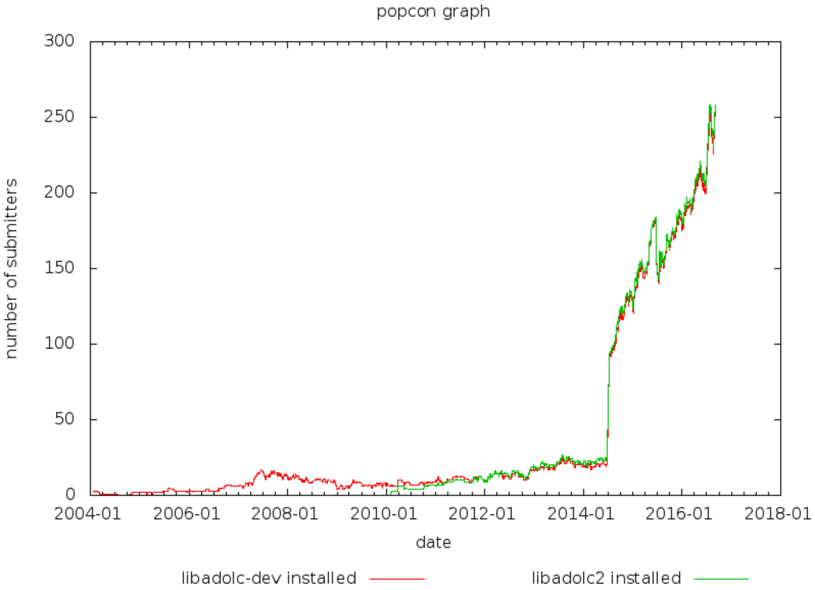
Editors' note: bibliometrics and the curators of orthodoxy

MSCS Editorial Board

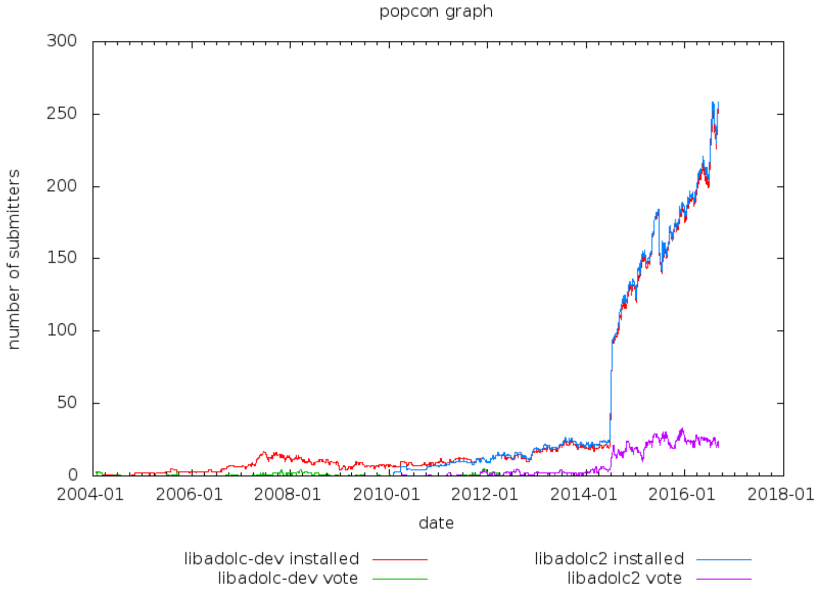
Received 1 December 2008

Have you ever seen the Citation Indexes (CIs) for the year 1600? At that time, a very active community was working on the reconstruction of planetary movements by means of epicycles. In principle, any ellipse around the Sun may be approximated by sufficiently many epicycles around the Earth. This is a non-trivial geometrical task, especially given the lack of analytical tools (sums of series). And the books and papers of many talented geometers quoted one another. Scientific knowledge, however, was already taking other directions. Science has a certain 'inertia', it is prudent (at times, it has been exceedingly so, mostly for political or metaphysical reasons), but even under the best of conditions, we all know how difficult it is to accept new ideas, to let them blossom in time, away from short-term pressures.

At best, CIs transform this slowness into a tool for judgement. If used unwisely, as is increasingly the case, they discourage people (young ones in particular) right from the



Debian package population count



Debian package population count (with usage)

Outline

1. beginning (ML rocks)
    - because they're smart
    - and generous
    - and work on important stuff
    - count their citations
    - look at their pretty pictures
    - follow the money
2. middle (technical gobbledygook)
    - caffe, theano, torch, tensorflow, grad, autograd, robust first class verbiage
    - reverse mode by reversing a reversible ode leaving bread crumbs
    - from stochastic gradient descent to stochastic hessian inverse product
    - limited by their tools but they're working hard
3. end (take home message)
    - if we don't service the ML community they will eat AD for lunch