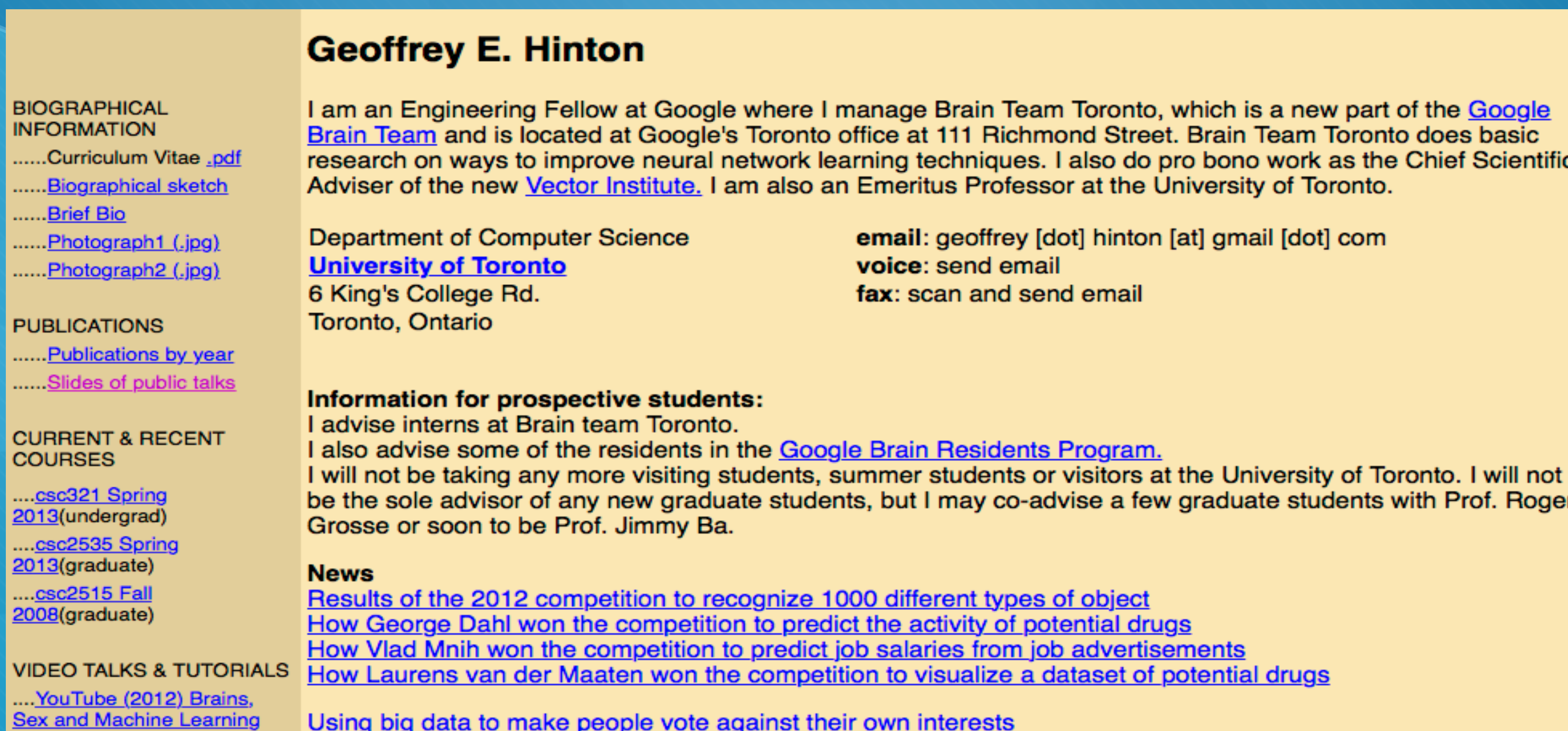
Advanced Neural Networks

2017統計專論 東華大學應用數學研究所

吳建銘

http://www.cs.toronto.edu/~hinton/



BIOGRAPHICAL INFORMATION

......Curriculum Vitae .pdf

......Biographical sketch

......Brief Bio

......Photograph1 (.jpg)

......Photograph2 (.jpg)

PUBLICATIONS

......Publications by year

......Slides of public talks

CURRENT & RECENT COURSES

....csc321 Spring 2013(undergrad)

....csc2535 Spring 2013(graduate)

....csc2515 Fall 2008(graduate)

VIDEO TALKS & TUTORIALS

....YouTube (2012) Brains, Sex and Machine Learning

Geoffrey E. Hinton

I am an Engineering Fellow at Google where I manage Brain Team Toronto, which is a new part of the Google Brain Team and is located at Google's Toronto office at 111 Richmond Street. Brain Team Toronto does basic research on ways to improve neural network learning techniques. I also do pro bono work as the Chief Scientific Adviser of the new Vector Institute. I am also an Emeritus Professor at the University of Toronto.

Department of Computer Science
University of Toronto
6 King's College Rd.
Toronto, Ontario

email: geoffrey [dot] hinton [at] gmail [dot] com
voice: send email
fax: scan and send email

Information for prospective students:
I advise interns at Brain team Toronto.
I also advise some of the residents in the Google Brain Residents Program.
I will not be taking any more visiting students, summer students or visitors at the University of Toronto. I will not be the sole advisor of any new graduate students, but I may co-advise a few graduate students with Prof. Roger Grosse or soon to be Prof. Jimmy Ba.

News
Results of the 2012 competition to recognize 1000 different types of object
How George Dahl won the competition to predict the activity of potential drugs
How Vlad Mnih won the competition to predict job salaries from job advertisements
How Laurens van der Maaten won the competition to visualize a dataset of potential drugs

Using big data to make people vote against their own interests

Researchers

- Hinton
- Widrow
- Hopfield
- Lecun

http://www.cs.toronto.edu/~hinton/science.pdf

esonance in Fig.
ation), we find
above the noise
is signal closely
incident power
e SHG emission
mall angle with
deviations from
the SRRs (see
Small detuning
ller wavelength
duces the SHG
20%. For ex-
e with vertical
we find a small
. For excitation
izontal incident
but significant
h is again po-

Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.

Dimensionality reduction facilitates the classification, visualization, communication, and storage of high-dimensional data. A simple and widely used method is principal components analysis (PCA), which finds the directions of greatest variance in the data set and represents each data point by its coordinates along each of these directions. We describe a nonlinear generalization of PCA that uses an adaptive, multilayer "encoder" network http://www.cs.toronto.edu/~hinton/absps/NatureDeepReview.pdf

REVIEW

doi:10.1038/nature14539

Deep learning

Yann LeCun[1,2], Yoshua Bengio[3] & Geoffrey Hinton[4,5]

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

http://www.genomics.princeton.edu/hopfield/Biography.html

John J.Hopfield

PrincetonUniversity LEWIS-SIGLER INSTITUTE FOR INTEGRATIVE GENOMICS

Home Biography Publications Pre & RePrints Simulation Programs Contact

John J. Hopfield, Professor
Department of Molecular Biology
Princeton University
Princeton NJ 08544
Tel: (609) 258-1239
Fax: (609) 258-6730
Email: hopfield@princeton.edu

Education Degree Year(s) Field Of Study
Swarthmore College A.B 1954 Physics
Cornell University Ph.D. 1958 Physics

Research Experience
1958-1960 Member of Technical Staff, Bell Laboratories

"Searching for memories, Sudoku, implicit check-bits, and the iterative use of not-always-correct rapid neural computation"

PDF

Hopfield network (An elementary introduction to the topic), Scholarpedia, 2007

Scholarpedia article

"Neural" Computations of Decisions in Optimization Problems, Biological Cybernetics 1985

PDF

Neural networks and physical systems with emergent collective computational abilities, Pro. Natl. Acad. Sci. 1982 http://www.genomics.princeton.edu/hopfield/PDFs/2008 neural computation.pdf

J. J. Hopfield
hopfield@princeton.edu
Carl Icahn Laboratory, Princeton University, Princeton, NJ 08544, U.S.A.

The algorithms that simple feedback neural circuits representing a brain area can rapidly carry out are often adequate to solve easy problems but for more difficult problems can return incorrect answers. A new excitatory-inhibitory circuit model of associative memory displays the common human problem of failing to rapidly find a memory when only a small clue is present. The memory model and a related computational network for solving Sudoku puzzles produce answers that contain implicit check bits in the representation of information across neurons, allowing a rapid evaluation of whether the putative answer is correct or incorrect through a computation related to visual pop-out. This fact may account for our strong psychological feeling of right or wrong when we retrieve a nominal memory from a minimal clue. This information allows more difficult computations or memory retrievals to be done in a serial fashion by using the fast but limited capabilities of a computational module multiple times. The mathematics of the excitatory-inhibitory circuits for associative memory and for Sudoku, both of which are understood in terms of energy or Lyapunov functions, is described in detail.

http://www.scholarpedia.org/article/Hopfield_network

SCHOLARPEDIA
the peer-reviewed
open-access encyclopedia

- Main page
- About
- Propose a new article
- Instructions for Authors
- Random article
- FAQs
- Help
- Blog

Focal areas
- Astrophysics
- Celestial mechanics
- Computational neuroscience
- Computational intelligence
- Dynamical systems
- Physics

Page | Discussion | Read | View source | View history | Search

Hopfield network

John J. Hopfield (2007), Scholarpedia, 2(5):1977. doi:10.4249/scholarpedia.1977 revision #91362 [link to/cite this article]

- **Dr. John J. Hopfield**, Princeton University, NJ, USA

Post-publication activity

Curator: John J. Hopfield

A **Hopfield net** is a recurrent neural network having synaptic connection pattern such that there is an underlying Lyapunov function for the activity dynamics. Started in any initial state, the state of the system evolves to a final state that is a (local) minimum of the Lyapunov function.

There are two popular forms of the model:

- Binary neurons with discrete time, updated one at a time

$$V_j(t+1) = \begin{cases} 1, & \text{if } \Sigma_k T_{jk} V_k(t) + I_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

- Graded neurons with continuous time

$$dx_j/dt = -x_j/\tau + \Sigma_k T_{jk} g(x_k) + I_j \ .$$

http://www-isl.stanford.edu/~widrow/

Bernard Widrow

Professor Emeritus
Electrical Engineering Department
Stanford University

Research
Prof. Widrow's research focuses on adaptive signal processing, adaptive control systems, adaptive neural networks, human memory, and human-like memory for computers. Applications include signal processing, prediction, noise cancelling, adaptive arrays, control systems, and pattern recognition.

Courses Taught
EE373A **Adaptive Signal Processing**
EE373B **Adaptive Neural Networks**

Hearing Aid Device
A directional acoustic receiving system is constructed in the form of a

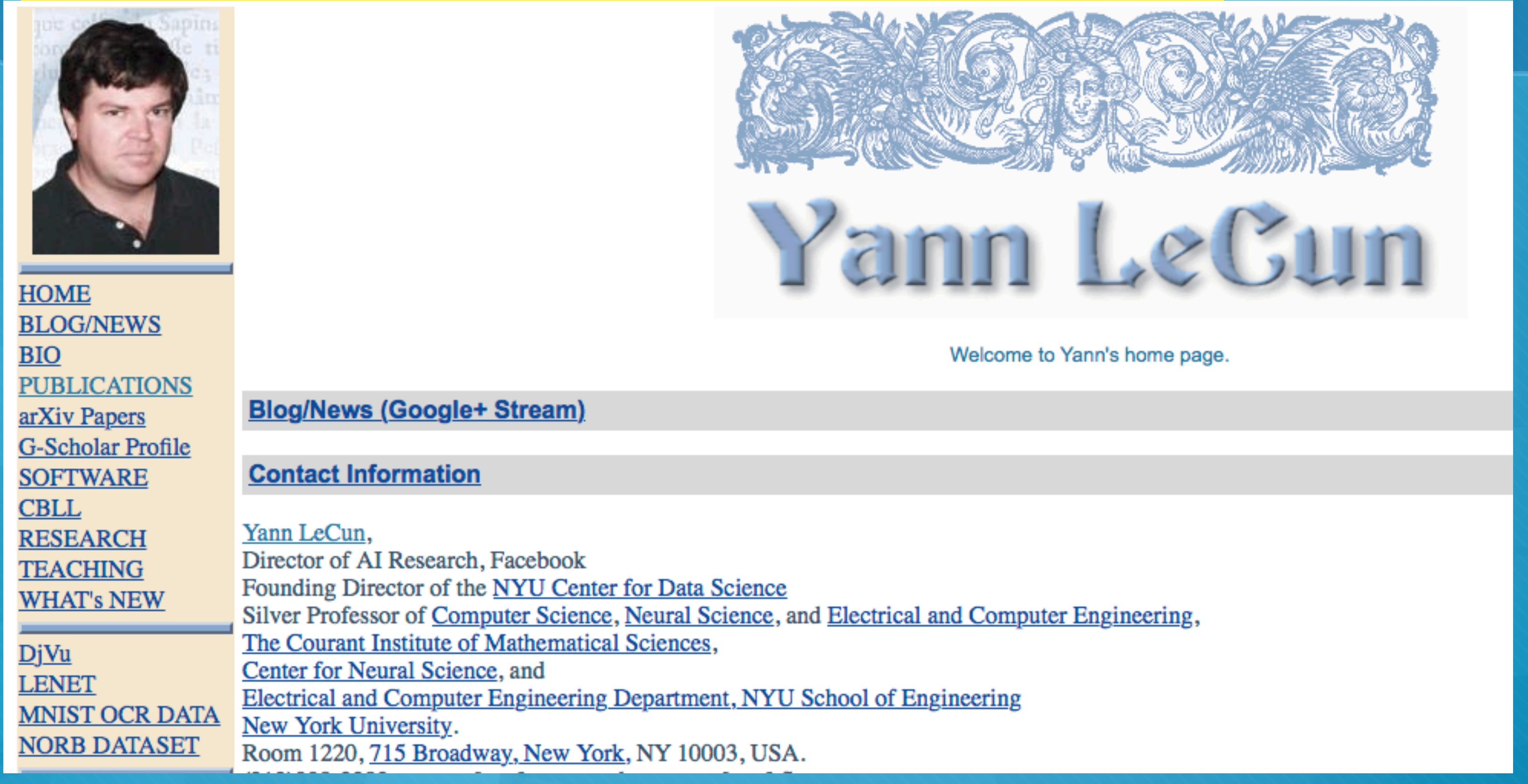
necklace, including an array of two or more microphones mounted on a housing supported on the chest of the user by a conducting loop encircling the user's neck. This method enables the design of highly-drective-hearing instruments which are comfortable, inconspicuous, and convenient to use. The array provides the user with a dramatic improvement in speech perception over existing hearing aid designs, particularly in the presence of background noise, reverberation, and feedback.

http://www-isl.stanford.edu/~widrow/papers/130.Hebbian_LMS.pdf

The Hebbian-LMS Learning Algorithm

http://www-isl.stanford.edu/~widrow/papers/c1960adaptiveswitching.pdf

http://yann.lecun.com



Yann LeCun

Welcome to Yann's home page.

- HOME
- BLOG/NEWS
- BIO
- PUBLICATIONS
- arXiv Papers
- G-Scholar Profile
- SOFTWARE
- CBLL
- RESEARCH
- TEACHING
- WHAT's NEW

- DjVu
- LENET
- MNIST OCR DATA
- NORB DATASET

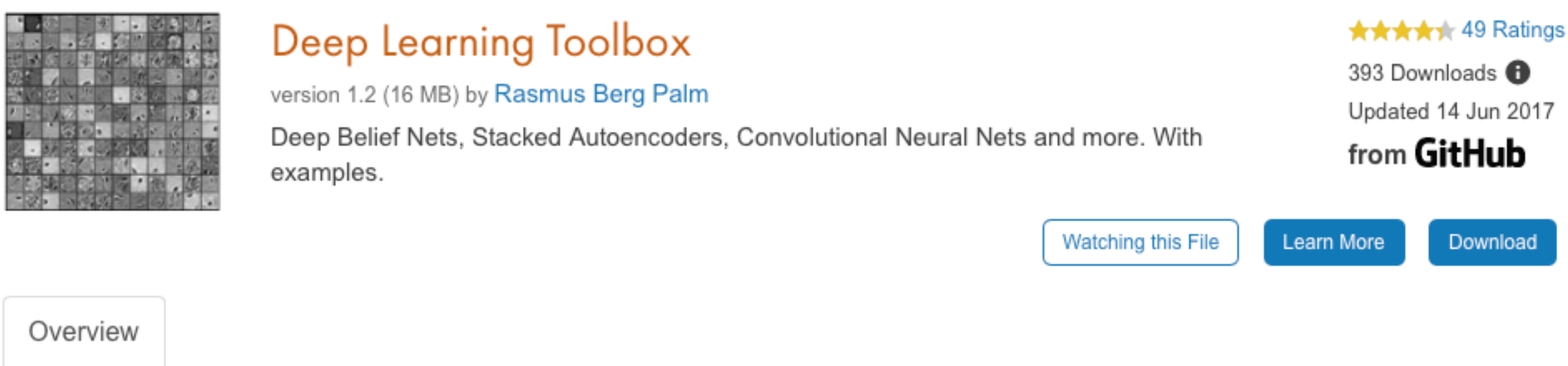
Blog/News (Google+ Stream)

Contact Information

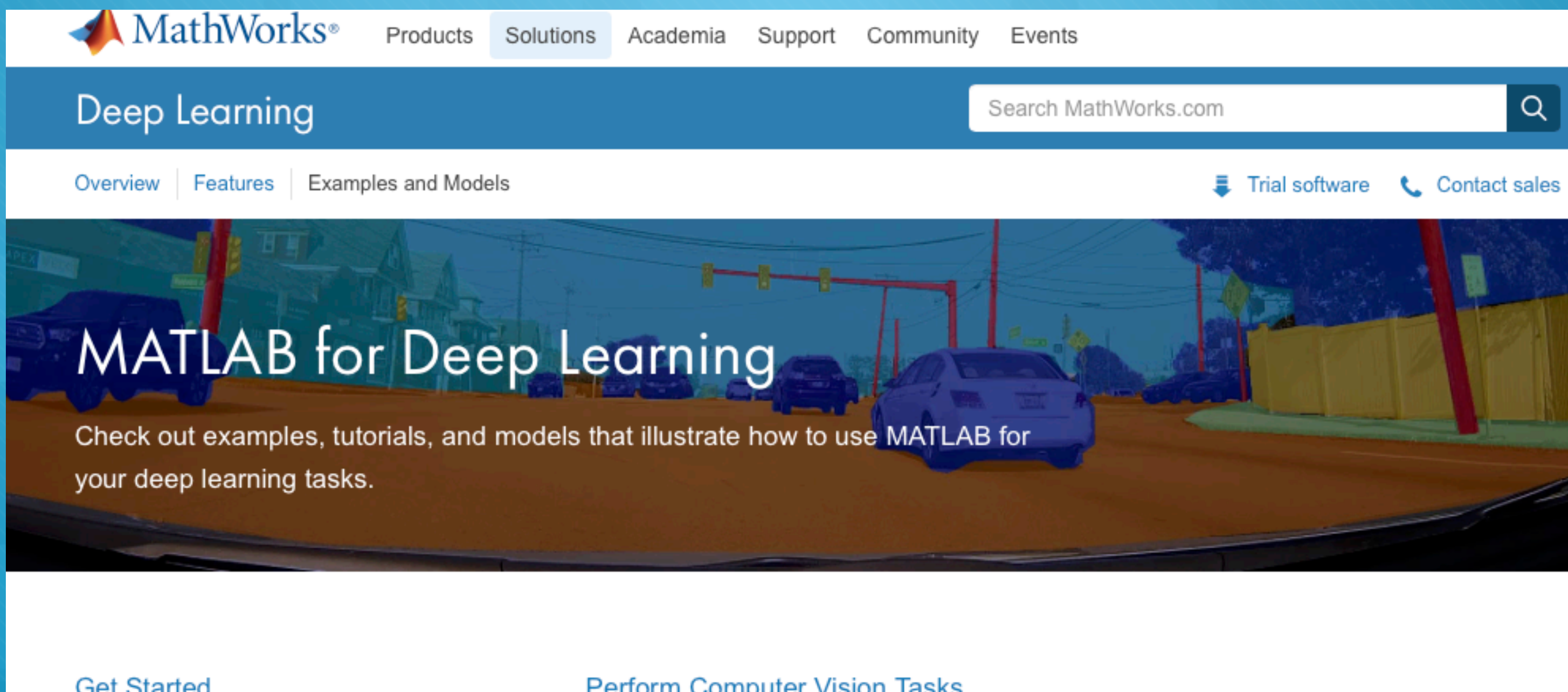
Yann LeCun,
Director of AI Research, Facebook
Founding Director of the NYU Center for Data Science
Silver Professor of Computer Science, Neural Science, and Electrical and Computer Engineering,
The Courant Institute of Mathematical Sciences,
Center for Neural Science, and
Electrical and Computer Engineering Department, NYU School of Engineering
New York University.
Room 1220, 715 Broadway, New York, NY 10003, USA.

SOFTWARE-MATLAB

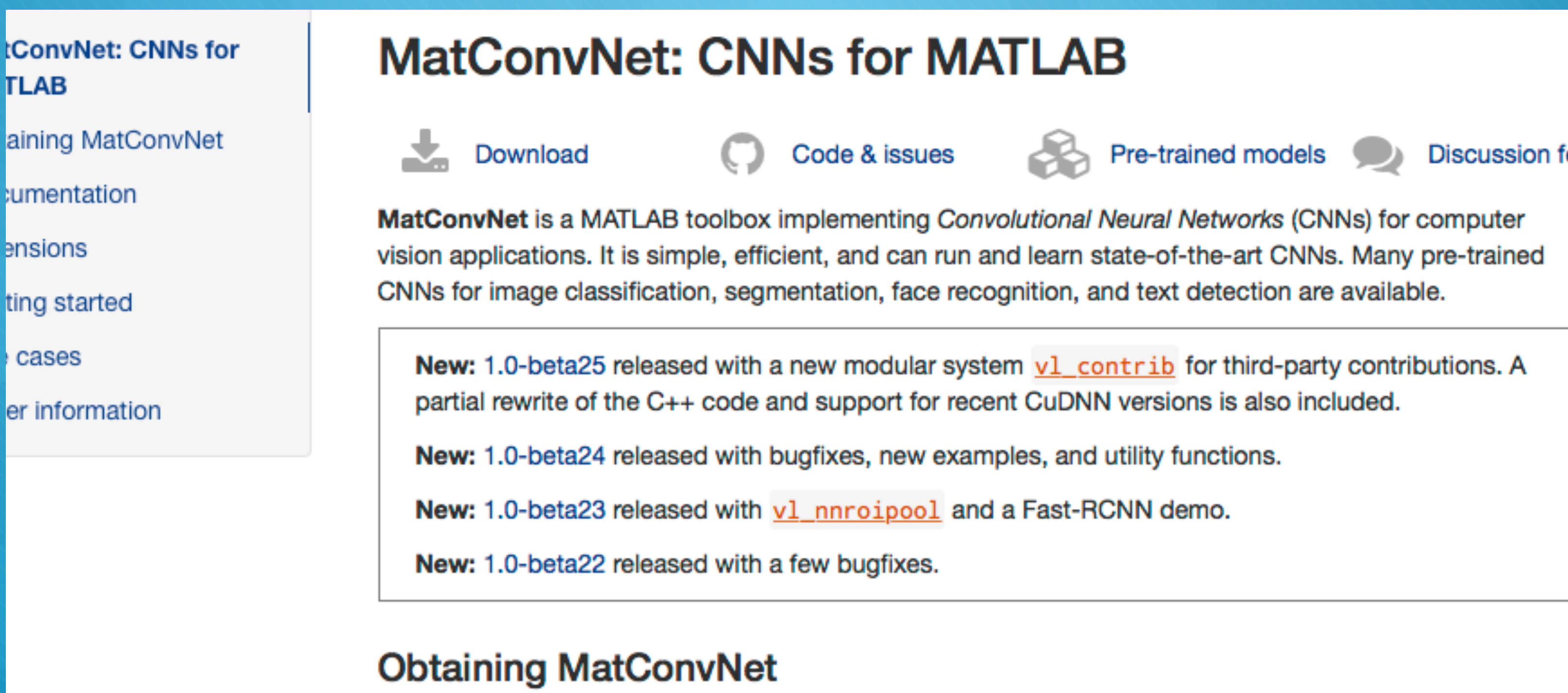
- MATLAB https://www.mathworks.com/matlabcentral/fileexchange/38310-deep-learning-toolbox



https://www.mathworks.com/solutions/deep-learning/examples.html



http://www.vlfeat.org/matconvnet/


tConvNet: CNNs for
TLAB

aining MatConvNet

cumentation

ensions

ting started

cases

er information

MatConvNet: CNNs for MATLAB

Download Code & issues Pre-trained models Discussion fo

MatConvNet is a MATLAB toolbox implementing *Convolutional Neural Networks* (CNNs) for computer vision applications. It is simple, efficient, and can run and learn state-of-the-art CNNs. Many pre-trained CNNs for image classification, segmentation, face recognition, and text detection are available.

New: 1.0-beta25 released with a new modular system `vl_contrib` for third-party contributions. A partial rewrite of the C++ code and support for recent CuDNN versions is also included.

New: 1.0-beta24 released with bugfixes, new examples, and utility functions.

New: 1.0-beta23 released with `vl_nnroipool` and a Fast-RCNN demo.

New: 1.0-beta22 released with a few bugfixes.

Obtaining MatConvNet

SOFTWARE-SWIFT

The Swift machine learning library.

swift machine-learning deep-learning artificial-intelligence ios macos ocr

341 commits | 1 branch | 2 releases | 19 contributors | MIT

Branch: **master** | New pull request | Find file | Clone or download

collinhundley committed on **GitHub** Update README.md — Latest commit 220f53a on 4 May

SiteAssets	New banner	5 months ago
.gitignore	Swift 3.1 rewrite	6 months ago
LICENSE.txt	Updated OS X example project structure	2 years ago
Package.swift	Swift 3.1 rewrite	6 months ago
README.md	Update README.md	5 months ago

README.md

https://github.com/alexsosn/iOS_ML

Curated list of resources for iOS developers in following topics:

- Core ML
- Machine Learning Libraries
- Deep Learning Libraries
  - Deep Learning: Model Compression
- Computer Vision
- Natural Language Processing
- Speech Recognition (TTS) and Generation (STT)
- Text Recognition (OCR)
- Other AI

https://github.com/KevinCoble/AIToolbox

AIToolbox

A toolbox of AI modules written in Swift: Graphs/Trees, Linear Regression, Support Vector Machines, Neural Networks, PCA, KMeans, Genetic Algorithms, MDP, Mixture of Gaussians, Logistic Regression

This framework uses the Accelerate library to speed up computations, except the Linux package versions. Written for Swift 3.0. Earlier versions are Swift 2.2 compatible

SVM ported from the public domain LIBSVM repository See https://www.csie.ntu.edu.tw/~cjlin/libsvm/ for more information

The Metal Neural Network uses the Metal framework for a Neural Network using the GPU. While it works in preliminary testing, more work could be done with this class

Use the XCTest files for examples on how to use the classes

Playgrounds for Linear Regression, SVM, and Neural Networks are available. Now available in both macOS and iOS versions.

###New - Convolution Program For the Deep Network classes, please look at the Convolution project that uses the

A Mini Review of Neuromorphic Architectures and Implementations

3
Author(s)

Robert A. Nawrocki ; Richard M. Voyles ; Sean E. Shaheen

View All Authors

Abstract:
Neuromorphic architectures are hardware systems that aim to use the principles of neural function for their basis of operation. Their goal is to harness biologically inspired concepts such as weighted connections, activation thresholds, short-and long-term potentiation, and inhibition to solve problems in distributed computation. Compared with today's methods of emulating neural function in software on conventional von Neumann hardware, neuromorphic systems provide the promise of inherently low power and fault-tolerant operation directly implemented into hardware, for application in distributed and embedded computing tasks, where the vast scaling of today's architectures does not provide a long-term solution. This mini review is intended for a general engineering audience not currently familiar with this exciting research area. It provides descriptions of some of the recent advances, including supercomputer and single-device implementations, approaches based on spiking and nonspiking neurons, machine learning hardware accelerators, and those utilizing memristive devices. Hardware implementations utilizing both conventional electronic

Browse Journals & Magazines > IEEE Transactions on Neural N... > Volume: 14 Issue: 5

VLSI implementations of threshold logic-a comprehensive survey

View Document

120 Paper Citations

7 Patent Citations

981 Full Text Views

Related Articles

Face recognition with radial basis function (RBF) neural networks

Extraction of rules from artificial neural networks for nonlinear regression

View All

3 Author(s)

V. Beiu ; J.M. Quintana ; M.J. Avedillo

View All Authors

Abstract | Authors | Figures | References | Citations | Keywords | Metrics | Media

Abstract:
This paper is an in-depth review on silicon implementations of threshold logic gates that covers several decades. In this paper, we will mention early MOS threshold logic solutions and detail numerous very-large-scale integration (VLSI) implementations including capacitive (switched capacitor and floating gate with their variations), conductance/current (pseudo-nMOS and output-wired-inverters, including a plethora of solutions evolved from them), as well as many differential solutions. At the end, we will briefly mention other implementations, e.g., based on negative resistance devices and on single electron technologies.

Published in: IEEE Transactions on Neural Networks (Volume: 14, Issue: 5, Sept. 2003)

Memory and Information Processing in Neuromorphic Systems

View Document

Related Articles

Competitive learning with floating-gate circuits

A local neural classifier for the recognition of EEG patterns associated to ment...

View All

2 Author(s)

Giacomo Indiveri ; Shih-Chii Liu

View All Authors

Abstract | Authors | Figures | References | Citations | Keywords | Metrics | Media

Abstract:
A striking difference between brain-inspired neuromorphic processors and current von Neumann processor architectures is the way in which memory and processing is organized. As information and communication technologies continue to address the need for increased computational power through the increase of cores within a digital processor, neuromorphic engineers and scientists can complement this need by building processor architectures where memory is distributed with the processing. In this paper, we present a survey of brain-inspired processor architectures that support models of cortical networks and deep neural networks. These architectures range from serial clocked implementations of multineuron systems to massively parallel asynchronous ones and from purely digital systems to mixed analog/digital systems which implement more biological-like models of neurons and synapses together with a suite of adaptation and learning mechanisms analogous to the ones found in biological nervous systems. We describe the advantages of the different approaches being pursued and present the challenges that need to be addressed for building artificial neural processing systems that can display the richness of behaviors seen in biological systems.

Published in: Proceedings of the IEEE (Volume: 103, Issue: 8, Aug. 2015)

http://ieeexplore.ieee.org/document/5607329/?denied

Development and Implementation of Parameterized FPGA Based General Purpose Neural Networks for Online Applications

Alexander Gomperts, Abhisek Ukil, *Senior Member, IEEE*, Franz Zurfluh

***Abstract*—This paper presents the development and implementation of a generalized backpropagation multilayer perceptron (MLP) architecture described in VLSI hardware description language (VHDL). The development of hardware platforms has been complicated by the high hardware cost and quantity of the arithmetic operations required in an online artificial neural networks (ANNs), i.e., general purpose ANNs with learning capability. Besides, there remains a dearth of hardware platforms for design space exploration, fast prototyping, and testing of these networks. Our general purpose architecture seeks to fill that gap and at the same time serve as a tool to gain a better understanding of issues unique to ANNs implemented in hardware, particularly using field programmable gate array (FPGA). The challenge is thus to find an architecture that minimizes hardware costs while maximizing performance, accuracy, and parameterization. This work describes a platform that offers a high degree of parameterization while maintaining generalized network design with performance comparable to other hardware based MLP implementations. Application of the hardware implementation of ANN with backpropagation learning algorithm for a realistic application is also presented.**

those neurons, etc. There remains a lack of a reliable means for determining the optimal set of network characteristics for a given application.

Numerous implementations of ANNs already exist [5], [6], [7], [8], but most of them being in software on sequential processors [2]. Software implementations can be quickly constructed, adapted and tested for a wide range of applications. However, in some cases the use of hardware architectures matching the parallel structure of ANNs is desirable to optimize performance or reduce the cost of the implementation, particularly for applications demanding high performance [9], [10]. Unfortunately, hardware platforms suffer from several unique disadvantages such as difficulties in achieving high data precision with relation to hardware cost, the high hardware cost of the necessary calculations, and the inflexibility of the platform as compared to software.

In our work we aimed to address some of these disadvan-