Shallow Understanding of Deep Learning

Siva Athreya

Indian Statistical Institute, Bangalore, India

May 18, 2018

Acknowledgement

I know very little about deep learning

Acknowledgement

Why I am giving this talk ?

Acknowledgement

No one really knows what is happening in Deep Learning

Acknowledgement

It can do a lot of cool stuff

Acknowledgements

- Adrian Röllin
- Mansi Garg
- Vivek Borkar
- Rajesh Sundaresan

Preliminaries

- Review
- Artificial Neural Networks
- Calculation to establish learning
- Cool Videos

Machine Learning

Give computers the ability to learn without being explicitly programmed.

Machine Learning: Supervised by itself

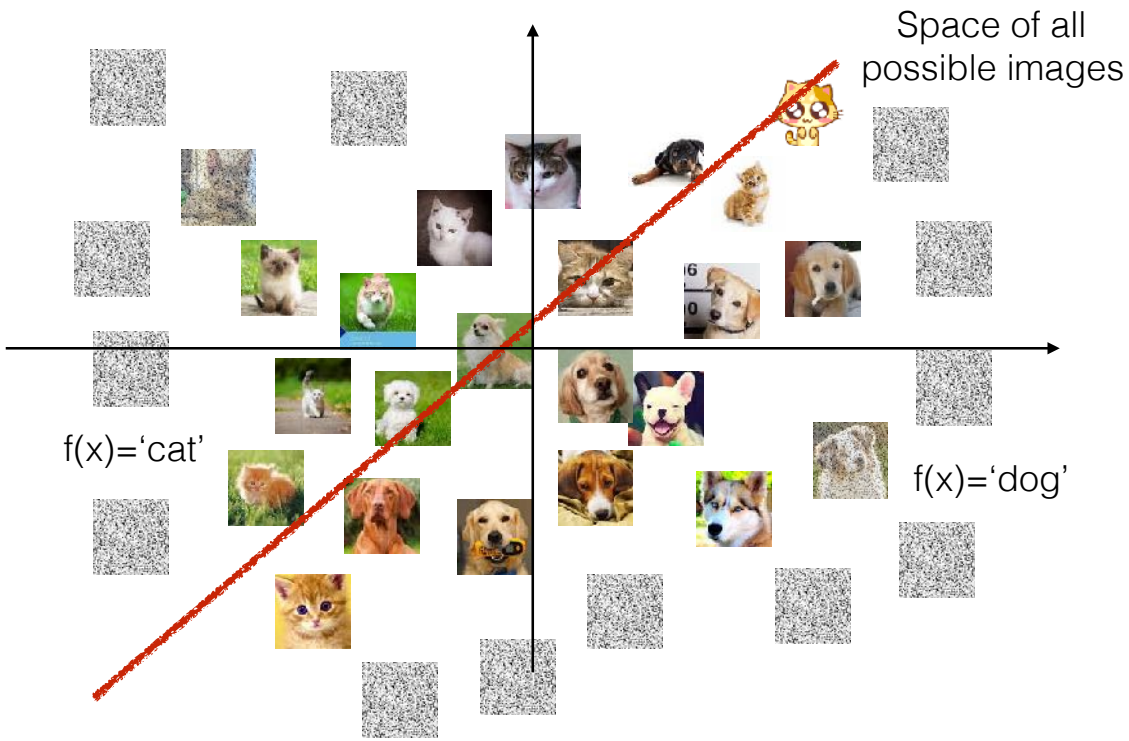
Find a function $f(x)$ so that

$$f(x_i) = y_i$$

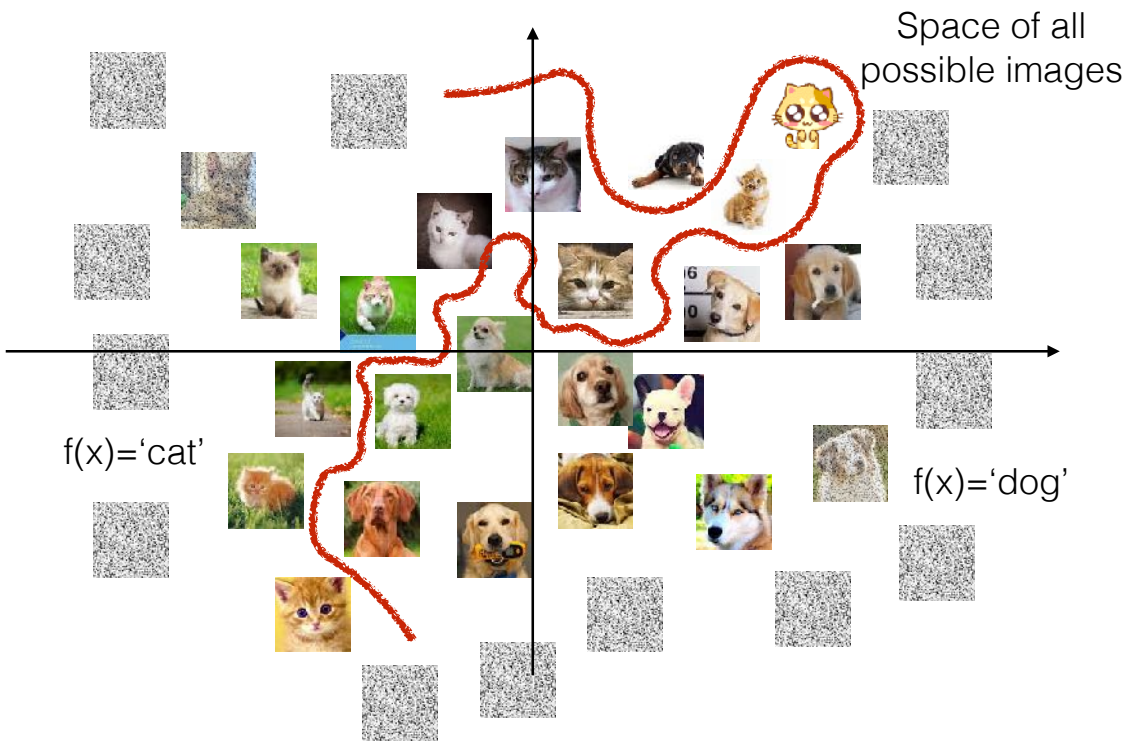
where data set is:

x_i	y_i
	Dog
	Cat

Linear Classification



Non-linear Classification



Machine Learning

a plate of food with broccoli and meat
logprob: -8.11

a motorcycle parked on the side of the street
logprob: -7.11

a table with a vase of flowers and a glass of water
logprob: -12.64

a little boy is holding a toothbrush in his mouth
logprob: -8.81

a plate of food with a fork and a glass of wine
logprob: -11.50

a surfboard is sitting on the sand near the water
logprob: -12.32

a woman standing in front of a pile of fruit
logprob: -10.31

http://cs.stanford.edu/people/karpathy/deepimagesent/generationdemo/

Image Net Competition

IMAGENET

14,197,122 images, 21841 synsets indexed

Explore Download Challenges Publications CoolStuff About

Not logged in. Login | Signup

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.

Click here to learn more about ImageNet, Click here to join the ImageNet mailing list.

What do these images have in common? *Find out!*

Check out the ImageNet Challenge on Kaggle!

AlphaGo

Nikkei ▼ 19344.15 -0.18% Hang Seng ▲ 23663.54 0.29% U.S. 10 Yr ▲ 1/32 Yield 2.497% Crude Oil ▼ 53.04 -0.30% Yen ▲ 113.90 0.00%

THE WALL STREET JOURNAL.

Subscribe Now | Sign In

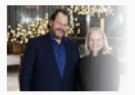
SPECIAL OFFER: JOIN NOW

Home World U.S. Politics Economy Business Tech Markets Opinion Arts Life Real Estate


Facebook Wasn't Ready for Video's Dark Side


Samsung Scandal Deepens as Prosecutors Detail Allegations


Fake Goods on Alibaba Hurt U.S. Small Businesses



TECH

AlphaGo Software Storms Back to Beat Human in Final Game

South Korean Go champion Lee Se-dol grabbed a victory from the artificial intelligence in fourth game, but couldn't repeat the feat

By JONATHAN CHENG

March 15, 2016 9:05 a.m. ET

SEOUL—Humanity didn't stand a chance.

South Korean Go grandmaster Lee Se-dol on Tuesday lost the final round of the chess-like game—and the match—to AlphaGo, an artificial-intelligence machine developed by Alphabet Inc.'s Google.

Mr. Lee, the 33-year-old representing human intelligence, had on

Recommended Videos

1. High Salaries at Charities Draw Scrutiny

2. Could North Korea's Missiles Reach the U.S.?

Machine playing poker

Machines are finally getting the best of humans at poker. dolgachov/iStockphoto

Artificial intelligence goes deep to beat humans at poker

By Tonya Riley | Mar. 3, 2017 , 2:15 PM

Two artificial intelligence (AI) programs have finally proven they "**know when to hold 'em, and when to fold 'em**," recently beating human professional card players for the

Image Net Competition

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

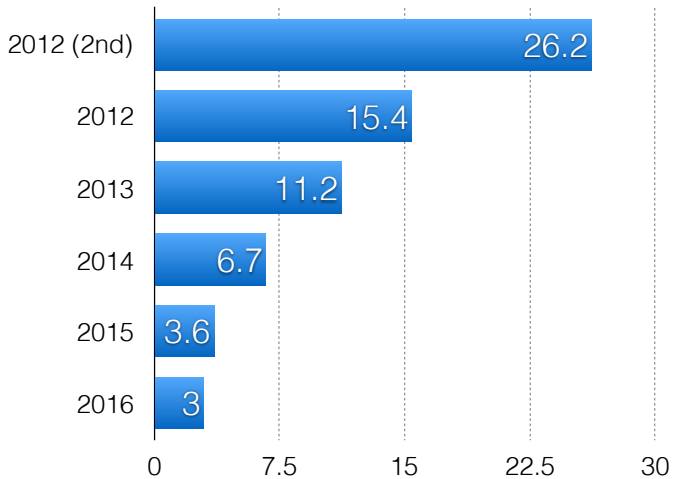
Abstract

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

Image Net Competition

ImageNet Competition



Name	Layers
AlexNet	8
ZFNet	8
GoogLeNet	22
Microsoft ResNet	152
Trimps-Soushen	

Human Top5 Error Rate: ≈ 5%

Deepmind

Jan 2014

theguardian

Google buys UK artificial intelligence startup Deepmind for £400m

Google makes its biggest EU purchase yet with the technology that aims to make computers think like humans

Google Atari

Playing Atari with Deep Reinforcement Learning

Volodymyr Mnih **Koray Kavukcuoglu** **David Silver** **Alex Graves** **Ioannis Antonoglou**

Daan Wierstra **Martin Riedmiller**

DeepMind Technologies

{vlad,koray,david,alex.graves,ioannis,daan,martin.riedmiller} @ deepmind.com

Abstract

We present the first deep learning model to successfully learn control policies directly from high-dimensional sensory input using reinforcement learning. The model is a convolutional neural network, trained with a variant of Q-learning, whose input is raw pixels and whose output is a value function estimating future rewards. We apply our method to seven Atari 2600 games from the Arcade Learning Environment, with no adjustment of the architecture or learning algorithm. We find that it outperforms all previous approaches on six of the games and surpasses a human expert on three of them.

1 Introduction

Learning to control agents directly from high-dimensional sensory inputs like vision and speech is one of the long-standing challenges of reinforcement learning (RL). Most successful RL applications that operate on these domains have relied on hand-crafted features combined with linear value functions or policy representations. Clearly, the performance of such systems heavily relies on the quality of the feature representation.

Recent advances in deep learning have made it possible to extract high-level features from raw sensory data, leading to breakthroughs in computer vision [11, 22, 16] and speech recognition [6, 7]. These methods utilise a range of neural network architectures, including convolutional networks, multilayer perceptrons, restricted Boltzmann machines and recurrent neural networks, and have exploited both supervised and unsupervised learning. It seems natural to ask whether similar techniques could also be beneficial for RL with sensory data.

However reinforcement learning presents several challenges from a deep learning perspective. Firstly, most successful deep learning applications to date have required large amounts of hand-labelled training data. RL algorithms, on the other hand, must be able to learn from a scalar reward

Deep Mind Atari

Neurons
Weights
layers
} Architecture

DEEP MIND ATARI

2014 Year

22000 Neurons

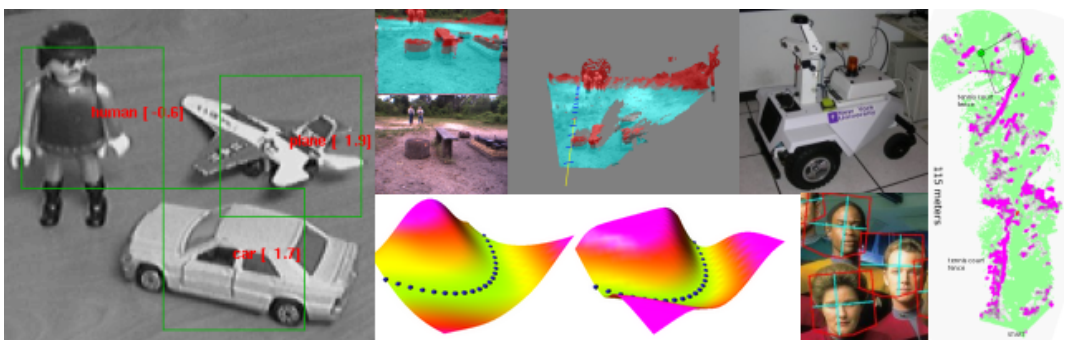
1,69,000 Weights

5 Layers

Microsoft Res-Net

MICROSOFT Res Net	
2015	Year
20,000,000	Neurons
60,000,000	Weights
152	Layers

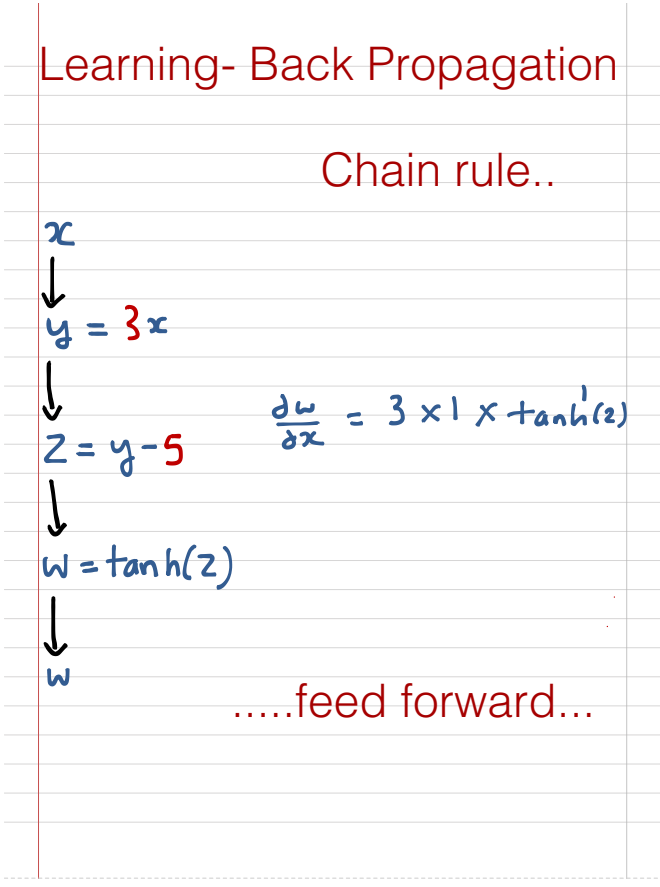
NYU Segmentation

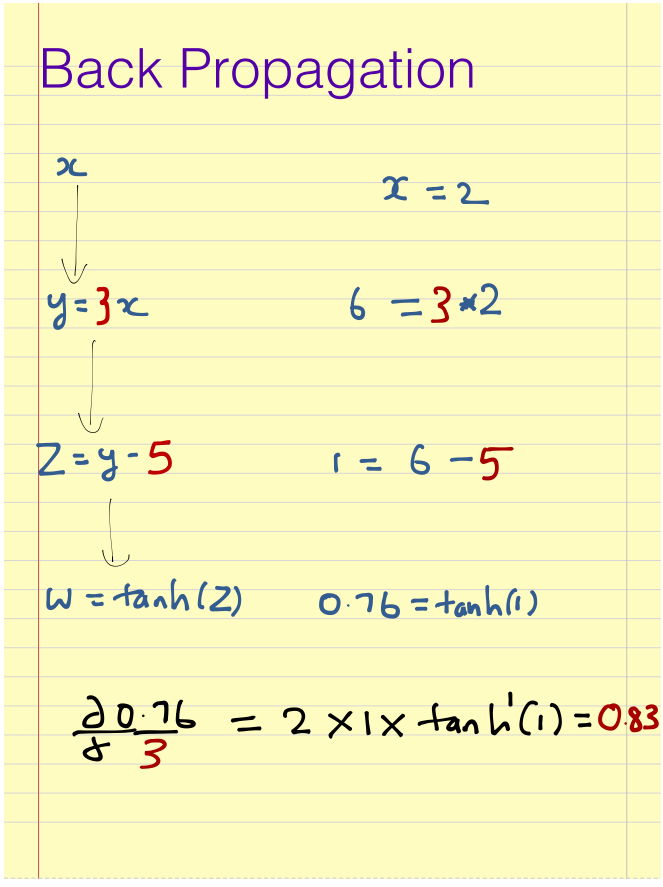


Why Now ?

- Data Sets available
- Advanced CPU
- Clever set of tricks to avoid Gradient Descent issues.

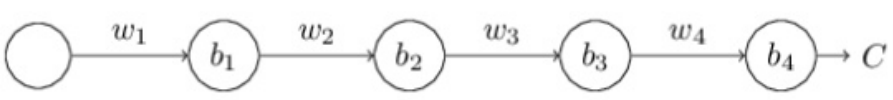
Vanishing Gradient





Deep learning Theory

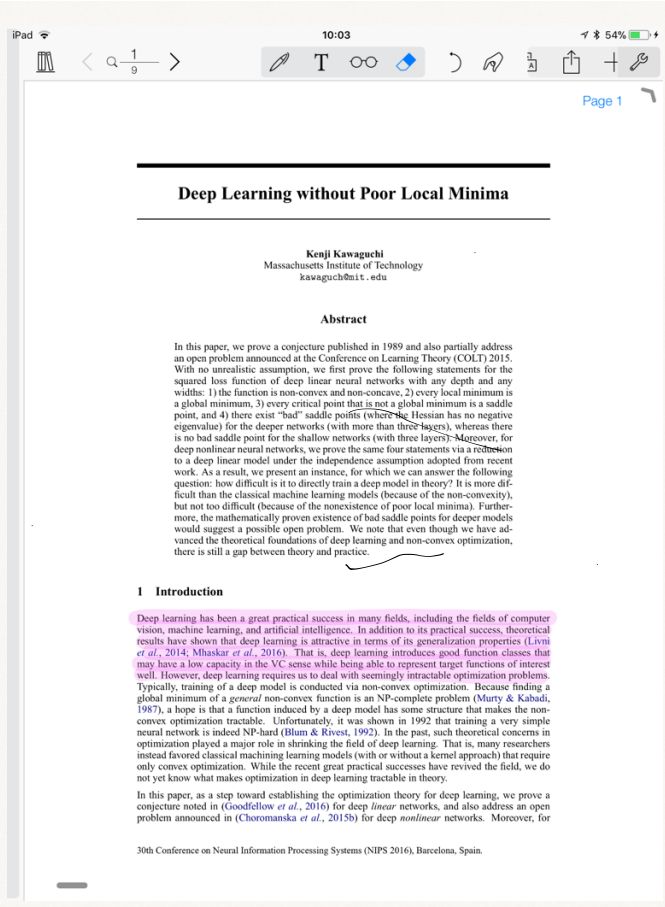
Vanishing Gradient Problem



$w_1, w_2, \ldots$ are the weights, $b_1, b_2, \ldots$ are the biases,
C is some cost function.
$a_j = \sigma(z_j)$, σ is activation function,
$z_j = w_j a_{j-1} + b_j$ is the weighted input to the neuron.

$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \times w_2 \times \sigma'(z_2) \times w_3 \times \sigma'(z_3) \times w_4 \times \sigma'(z_4) \times \frac{\partial C}{\partial a_4}$$

- All contents are collected online, listed in **Reference** page.



iPad 10:03 54%

Page 1

Deep Learning without Poor Local Minima

Kenji Kawaguchi
Massachusetts Institute of Technology
kawaguch@mit.edu

Abstract

In this paper, we prove a conjecture published in 1989 and also partially address an open problem announced at the Conference on Learning Theory (COLT) 2015. With no unrealistic assumption, we first prove the following statements for the squared loss function of deep linear neural networks with any depth and any widths: 1) the function is non-convex and non-concave, 2) every local minimum is a global minimum, 3) every critical point that is not a global minimum is a saddle point, and 4) there exist "bad" saddle points (where the Hessian has no negative eigenvalue) for the deeper networks (with more than three layers), whereas there is no bad saddle point for the shallow networks (with three layers). Moreover, for deep nonlinear neural networks, we prove the same four statements via a reduction to a deep linear model under the independence assumption adopted from recent work. As a result, we present an instance, for which we can answer the following question: how difficult is it to directly train a deep model in theory? It is more difficult than the classical machine learning models (because of the non-convexity), but not too difficult (because of the nonexistence of poor local minima). Furthermore, the mathematically proven existence of bad saddle points for deeper models would suggest a possible open problem. We note that even though we have advanced the theoretical foundations of deep learning and non-convex optimization, there is still a gap between theory and practice.

1 Introduction

Deep learning has been a great practical success in many fields, including the fields of computer vision, machine learning, and artificial intelligence. In addition to its practical success, theoretical results have shown that deep learning is attractive in terms of its generalization properties (Livni *et al.*, 2014; Mhaskar *et al.*, 2016). That is, deep learning introduces good function classes that may have a low capacity in the VC sense while being able to represent target functions of interest well. However, deep learning requires us to deal with seemingly intractable optimization problems. Typically, training of a deep model is conducted via non-convex optimization. Because finding a global minimum of a *general* non-convex function is an NP-complete problem (Murty & Kabadi, 1987), a hope is that a function induced by a deep model has some structure that makes the non-convex optimization tractable. Unfortunately, it was shown in 1992 that training a very simple neural network is indeed NP-hard (Blum & Rivest, 1992). In the past, such theoretical concerns in optimization played a major role in shrinking the field of deep learning. That is, many researchers instead favored classical machining learning models (with or without a kernel approach) that require only convex optimization. While the recent great practical successes have revived the field, we do not yet know what makes optimization in deep learning tractable in theory.

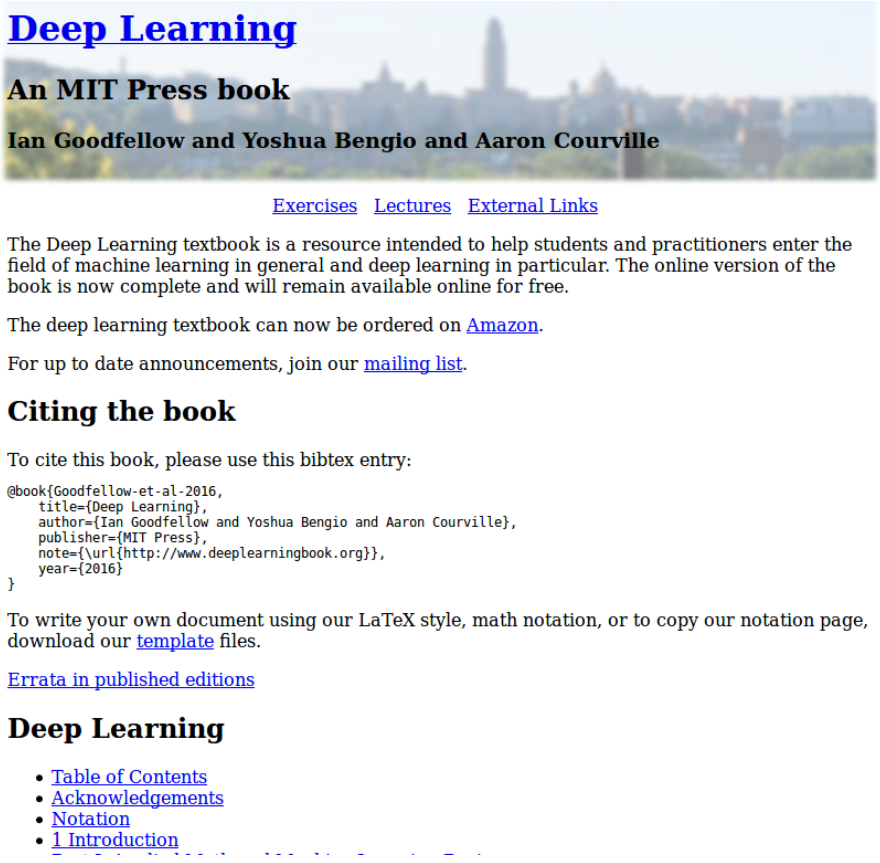
In this paper, as a step toward establishing the optimization theory for deep learning, we prove a conjecture noted in (Goodfellow *et al.*, 2016) for deep *linear* networks, and also address an open problem announced in (Choromanska *et al.*, 2015b) for deep *nonlinear* networks. Moreover, for

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Conclusions

- Powerful Tool for any minimisation problem
- Architecture, Learning and Application.
- Explosion of Powerful Artificial Neural Networks
- Mathematical Theory awaited.
- Still cannot resolve our SPAM filter!

Resources: Book



Deep Learning

An MIT Press book

Ian Goodfellow and Yoshua Bengio and Aaron Courville

Exercises Lectures External Links

The Deep Learning textbook is a resource intended to help students and practitioners enter the field of machine learning in general and deep learning in particular. The online version of the book is now complete and will remain available online for free.

The deep learning textbook can now be ordered on Amazon.

For up to date announcements, join our mailing list.

Citing the book

To cite this book, please use this bibtex entry:

```
@book{Goodfellow-et-al-2016,
    title={Deep Learning},
    author={Ian Goodfellow and Yoshua Bengio and Aaron Courville},
    publisher={MIT Press},
    note={\url{http://www.deeplearningbook.org}},
    year={2016}
}
```

To write your own document using our LaTeX style, math notation, or to copy our notation page, download our template files.

Errata in published editions

Deep Learning

- Table of Contents
- Acknowledgements
- Notation
- 1 Introduction

Plus Your favourite book on:

- Basic Programming,
- Calculus: Gradient Descent,
- Linear Algebra and
- Statistics.

Resources

Resources

Docs » Home

Edit on GitHub

Keras: The Python Deep Learning library

You have just found Keras.

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. *Being able to go from idea to result with the least possible delay is key to doing good research.*

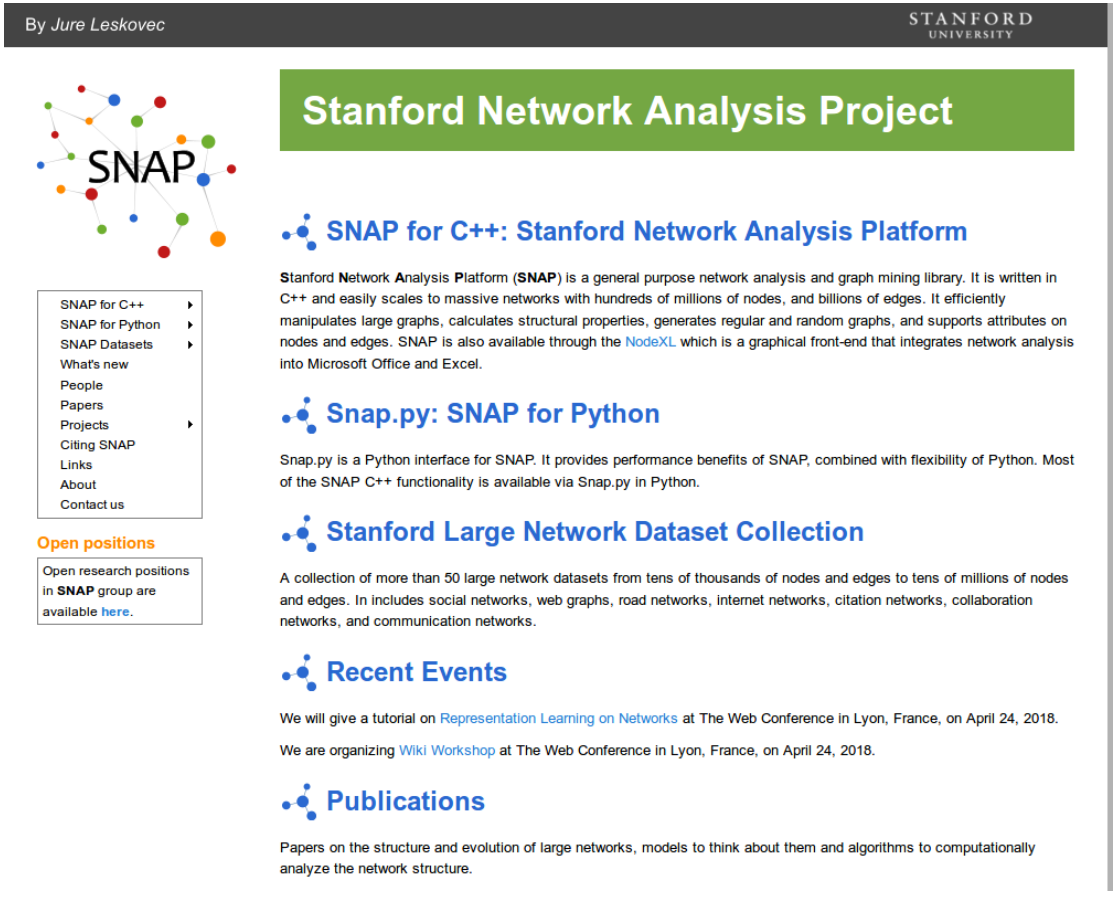
Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolutional networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU.

Read the documentation at Keras.io.

Keras is compatible with: **Python 2.7-3.6**.

Resources


By Jure Leskovec
STANFORD UNIVERSITY
SNAP
SNAP for C++
SNAP for Python
SNAP Datasets
What's new
People
Papers
Projects
Citing SNAP
Links
About
Contact us
Open positions
Open research positions in SNAP group are available here.
Stanford Network Analysis Project
SNAP for C++: Stanford Network Analysis Platform
Stanford Network Analysis Platform (SNAP) is a general purpose network analysis and graph mining library. It is written in C++ and easily scales to massive networks with hundreds of millions of nodes, and billions of edges. It efficiently manipulates large graphs, calculates structural properties, generates regular and random graphs, and supports attributes on nodes and edges. SNAP is also available through the NodeXL which is a graphical front-end that integrates network analysis into Microsoft Office and Excel.
Snap.py: SNAP for Python
Snap.py is a Python interface for SNAP. It provides performance benefits of SNAP, combined with flexibility of Python. Most of the SNAP C++ functionality is available via Snap.py in Python.
Stanford Large Network Dataset Collection
A collection of more than 50 large network datasets from tens of thousands of nodes and edges to tens of millions of nodes and edges. In includes social networks, web graphs, road networks, internet networks, citation networks, collaboration networks, and communication networks.
Recent Events
We will give a tutorial on Representation Learning on Networks at The Web Conference in Lyon, France, on April 24, 2018.
We are organizing Wiki Workshop at The Web Conference in Lyon, France, on April 24, 2018.
Publications
Papers on the structure and evolution of large networks, models to think about them and algorithms to computationally analyze the network structure.

Resources

Your favourite online resource

from: Keras,

Image net

Thank you

It has been a fun filled 10 days.