Deep Learning

Artificial neural networks with more than one hidden layer now go by the name ‘deep neural networks’, ‘deep learning’: learning multiple levels of representation and abstraction to make sense of data such as images, sound, and text.

Due to unprecedented success in a variety of machine learning tasks, deep neural networks are becoming increasingly popular and are now often implemented with more than five layers (and millions of parameters).

The theory of deep networks is an active area of research in academia. Fortunately for those of us in industry, there are now many good resources for learning about neural networks and applying them in practice, often without the need for extensive amounts of theory or mathematics up front.

Overview

Very Brief Introduction to Machine Learning for AI:

* <http://www.iro.umontreal.ca/~pift6266/H10/notes/mlintro.html>

Introduction to Deep Learning Algorithms:

* <http://www.iro.umontreal.ca/~pift6266/H10/notes/deepintro.html>

A brief overview of Deep Learning, by Ilya Sutskever (Open AI):

* <http://yyue.blogspot.co.uk/2015/01/a-brief-overview-of-deep-learning.html>

For an alternative to our guide below, see <https://metacademy.org/roadmaps/rgrosse/deep_learning>

Papers

LeCun, Y., Bengio, Y. and Hinton, G. E. (2015), *Deep Learning*, Nature:

* <http://www.cs.toronto.edu/~hinton/absps/NatureDeepReview.pdf>

Bengio, Y. (2009), *Learning Deep Architectures for AI*:

* <http://www.iro.umontreal.ca/~bengioy/papers/ftml.pdf>

1. Recommended Resources

Textbook

The Deep Learning textbook (Goodfellow, Bengio, Courville, 2016):

* Introduction: <http://www.deeplearningbook.org/contents/intro.html>

“It is easiest to understand deep learning with some historical context. Rather than providing a detailed history of deep learning, we identify a few key trends:

* Deep learning has had a long and rich history, but has gone by many names reﬂecting diﬀerent philosophical viewpoints, and has waxed and waned in popularity.
* Deep learning has become more useful as the amount of available training data has increased.
* Deep learning models have grown in size over time as computer hardware and software infrastructure for deep learning has improved.
* Deep learning has solved increasingly complicated applications with increasing accuracy over time” (p.11)
* Part II: <http://www.deeplearningbook.org/>

“This part of the book summarizes the state of modern deep learning as it is used to solve practical applications... It focuses only on those approaches that are essentially working technologies that are already used heavily in industry… These chapters are the most important for a practitioner—someone who wants to begin implementing and using deep learning algorithms to solve real-world problems today.” (p.166)

Recommended Tutorials

Stanford Course on Convolutional Neural Networks - has excellent course notes

* <http://cs231n.github.io/>

Associated videos and slides:

* <http://cs231n.stanford.edu/syllabus.html>

Alternative Tutorials

Stanford Tutorials on Deep Learning (see contents on right-hand side)

* <http://deeplearning.stanford.edu/tutorial/>

Stanford Course on Deep Learning for Natural Language Processing (NLP), though it does not have such awesome notes with code like the cs231n course does (see above).

* <http://cs224d.stanford.edu/syllabus.html>

Recommended Lectures

Weeks 4 & 5, Coursera *Machine Learning,* Andrew Ng:

* <https://www.coursera.org/learn/machine-learning/home/welcome>

Alternative Lectures

Coursera *Neural Networks for Machine Learning*, Geoffrey Hinton:

* <https://www.coursera.org/course/neuralnets>

Hugo Larochelle's online neural networks lectures:

* <https://www.youtube.com/watch?v=SGZ6BttHMPw&list=PL6Xpj9I5qXYEcOhn7TqghAJ6NAPrNmUBH>

And associated reading:

* <http://info.usherbrooke.ca/hlarochelle/neural_networks/content.html>

Victor Lavrenko (and Nigel Goddard)’s online neural networks lectures:

* <https://www.youtube.com/watch?v=jZYz0EUPYBI&list=PLBv09BD7ez_4Bs9j3o8l_ZTjQZoN_3Oqs>

“Neural Networks Demystified”

* <https://www.youtube.com/watch?v=bxe2T-V8XRs&feature=youtu.be&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU>

2. Software Packages

The two dominant software packages for neural networks are Theano and Tensorflow.

There is a overlying package called Keras which sits on top of both.

--

Theano

A Python library that lets you to define, optimize, and evaluate mathematical expressions, especially ones with multi-dimensional arrays (numpy.ndarray)

Preliminary tutorial on NumPy:

* <http://cs231n.github.io/python-numpy-tutorial/>

Very brief theano tutorial:

* <http://nbviewer.jupyter.org/github/craffel/theano-tutorial/blob/master/Theano%20Tutorial.ipynb>

Recommended in-depth theano tutorial:

* <http://deeplearning.net/software/theano/tutorial/>

Alternative / further exercises:

* <https://github.com/goodfeli/theano_exercises>

Guide to using theano for machine learning:

* <http://www.deeplearning.net/tutorial/>

Guide to implementing a neural network:

* <http://peterroelants.github.io/>

PyData survey talk with a brief overview of deep learning and how you can get started:

* <https://www.youtube.com/watch?v=MVyauNNinC0>

--

TensorFlow  
An open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them.

* Home: <https://www.tensorflow.org/>
* Tutorials: <https://www.tensorflow.org/versions/r0.7/tutorials/index.html>
* Wiki: <https://github.com/tobigithub/tensorflow-deep-learning/wiki>
* Google/Udacity online course: <https://www.udacity.com/course/deep-learning--ud730>
* Playground: <http://playground.tensorflow.org/>

--

Keras

A minimalist, highly modular neural networks library, written in Python and capable of running on top of either [TensorFlow](https://github.com/tensorflow/tensorflow) or [Theano](https://github.com/Theano/Theano). It was developed with a focus on enabling fast experimentation.

* Documentation: <http://keras.io/>
* Quick start: <http://keras.io/#getting-started-30-seconds-to-keras>

3. Further Resources

Videos

Hinton (2010), *Recent Developments in Deep Learning*

* <https://www.youtube.com/watch?v=VdIURAu1-aU>

Blog Posts

Deep Learning in a Nutshell, by Nikhil Buduma:

* <http://nikhilbuduma.com/2014/12/29/deep-learning-in-a-nutshell/>

The Landscape of Deep Learning, by Xiang Zhang:

* <http://xzh.me/posts/deeplearninglandscape/>

Cheat Sheet of Neural Network Architectures, by Fjodor van Veen:

* <http://www.asimovinstitute.org/neural-network-zoo/>

How Spotify uses Convolutional Neural Networks to use music, by Sander Dieleman:

* <http://benanne.github.io/2014/08/05/spotify-cnns.html>

How to make use of Deep Reinforcement Learning, by Andrej Karpathy:

* <http://karpathy.github.io/2016/05/31/rl/>

The Unreasonable Effectiveness of Recurrent Neural Networks, by Andrej Karpathy:

* <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

Neural Networks, Manifolds, and Topology, by Chris Olah:

* <http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/>

Evaluation of Deep Learning Frameworks:

* <https://github.com/zer0n/deepframeworks/blob/master/README.md>

Demos

* <http://deeplearning.net/demos/>

Interviews

Talking Machines, S1 E5 & 6, [*The Future of Machine Learning from the Inside Out:*](http://www.thetalkingmachines.com/blog/2015/3/13/how-machine-learning-got-where-it-is-and-the-future-of-the-field)

Interviews with Hinton, Bengio and LeCun on the history (and future) of research on neural networks. (Second half of the episodes).

* E5: <http://www.thetalkingmachines.com/blog/2015/2/26/the-history-of-machine-learning-from-the-inside-out>
* E6: [*http://www.thetalkingmachines.com/blog/2015/3/13/how-machine-learning-got-where-it-is-and-the-future-of-the-field*](http://www.thetalkingmachines.com/blog/2015/3/13/how-machine-learning-got-where-it-is-and-the-future-of-the-field)

Talking Machines S1 E19, *Strong AI and Autoencoders*

Intro to Autoencoders and interview with Hugo Larochelle (see lectures above)

* <http://www.thetalkingmachines.com/blog/2015/9/10/strong-ai-and-autoencoders>

Talking Machines S2 E16, Eric Lander and Restricted Boltzmann Machines

Introduction to (Restricted) Boltzmann Machines in the first 8 minutes

* <http://www.thetalkingmachines.com/blog/2016/8/18/eric-lander-and-restricted-boltzmann-machines>

4. Tips and Tricks for training and debugging

A Practical Guide to Training Restricted Boltzmann Machines, by Hinton

* <https://www.cs.toronto.edu/~hinton/absps/guideTR.pdf>

[http://lamda.nju.edu.cn/.../project/CNNTricks/CNNTricks.html](http://l.facebook.com/l.php?u=http%3A%2F%2Flamda.nju.edu.cn%2Fweixs%2Fproject%2FCNNTricks%2FCNNTricks.html&h=rAQHbLUYI)

[http://deeplearning.cs.cmu.edu/.../jmlr10\_larochelle.pdf](http://l.facebook.com/l.php?u=http%3A%2F%2Fdeeplearning.cs.cmu.edu%2Fpdfs%2F1111%2Fjmlr10_larochelle.pdf&h=aAQGDLWtH)

[http://deeplearning4j.org/trainingtricks](http://l.facebook.com/l.php?u=http%3A%2F%2Fdeeplearning4j.org%2Ftrainingtricks&h=tAQFafn8x)

[http://hagan.okstate.edu/22\_PracticalTraining.pdf](http://l.facebook.com/l.php?u=http%3A%2F%2Fhagan.okstate.edu%2F22_PracticalTraining.pdf&h=lAQHIWnrS)

[http://tedlab.mit.edu/~dr/Lens/thumb.html](http://l.facebook.com/l.php?u=http%3A%2F%2Ftedlab.mit.edu%2F~dr%2FLens%2Fthumb.html&h=RAQGyVV3g)

[http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf](http://l.facebook.com/l.php?u=http%3A%2F%2Fyann.lecun.com%2Fexdb%2Fpublis%2Fpdf%2Flecun-98b.pdf&h=oAQF8YE-E)

[https://www.quora.com/.../What-are-some-tips-and-tricks...](https://l.facebook.com/l.php?u=https%3A%2F%2Fwww.quora.com%2FMachine-Learning%2FWhat-are-some-tips-and-tricks-for-training-deep-neural-networks&h=wAQFYvm6A)

[https://cs224d.stanford.edu/lectures/CS224d-Lecture6.pdf](https://l.facebook.com/l.php?u=https%3A%2F%2Fcs224d.stanford.edu%2Flectures%2FCS224d-Lecture6.pdf&h=NAQFgDS8W)

[http://arxiv.org/abs/1206.5533](http://l.facebook.com/l.php?u=http%3A%2F%2Farxiv.org%2Fabs%2F1206.5533&h=XAQGA_nbT)

[http://karpathy.github.io/neuralnets/](http://l.facebook.com/l.php?u=http%3A%2F%2Fkarpathy.github.io%2Fneuralnets%2F&h=bAQGad6M0)

<http://www.deeplearningbook.org/> ch11

5. How can one get started with machine learning? [(Quora question)](https://www.quora.com/Yoshua-Bengio-How-can-one-get-started-with-machine-learning?no_redirect=1)

Yoshua Bengio [(Quora profile)](https://www.quora.com/profile/Yoshua-Bengio)

Head of Montreal Institute for Learning Algorithms, Professor @ U. Montreal

“First you need to be trained with the appropriate basis in mathematics and computer science. In the case of deep learning, you can see part 1 of the MIT Press [Deep Learning book](http://deeplearningbook.org) to either brush up on these or see which areas of math and CS are most relevant.

Then you need to read on machine learning (there are several good books, such as [Chris Bishop's](http://research.microsoft.com/en-us/um/people/cmbishop/prml/) and [Kevin Murphy's](https://www.cs.ubc.ca/~murphyk/MLbook/), online videos such as [Andrew Ng's Coursera class](https://www.coursera.org/learn/machine-learning/home/welcome) and [Hugo Larochelle's videos on neural networks](https://www.youtube.com/watch?v=SGZ6BttHMPw&list=PL6Xpj9I5qXYEcOhn7TqghAJ6NAPrNmUBH), and you can get a summary of many of the basic issues in [chapter 5](http://www.deeplearningbook.org/contents/ml.html) of the Deep Learning book).

Then you need to start practicing, i.e., programming some learning algorithms yourself and playing with them on data, try to compete in some [Kaggle competitions](https://www.kaggle.com/competitions), for example. Try to become an expert at optimizing hyper-parameters and choosing models appropriately.

In parallel, continue reading. If you are interested in deep learning, part 2 of my book will give you the basis for the most common algorithms. At that point you should have enough background to start a steady regimen of reading [papers](http://deeplearning.net/reading-list/) that tickle your fancy.”