

Big Data Mining

巨量資料探勘

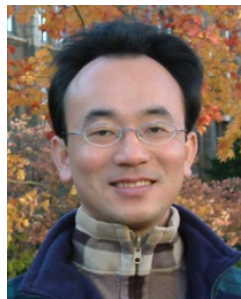
Deep Learning with Google TensorFlow

(Google TensorFlow 深度學習)

1042DM10

MI4 (M2244) (3094)

Tue, 3, 4 (10:10-12:00) (B216)



Min-Yuh Day

戴敏育

Assistant Professor

專任助理教授

Dept. of Information Management, Tamkang University

淡江大學 資訊管理學系

<http://mail.tku.edu.tw/myday/>

2016-05-10



課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
1	2016/02/16	巨量資料探勘課程介紹 (Course Orientation for Big Data Mining)
2	2016/02/23	巨量資料基礎：MapReduce典範、Hadoop與Spark生態系統 (Fundamental Big Data: MapReduce Paradigm, Hadoop and Spark Ecosystem)
3	2016/03/01	關連分析 (Association Analysis)
4	2016/03/08	分類與預測 (Classification and Prediction)
5	2016/03/15	分群分析 (Cluster Analysis)
6	2016/03/22	個案分析與實作一 (SAS EM 分群分析)： Case Study 1 (Cluster Analysis – K-Means using SAS EM)
7	2016/03/29	個案分析與實作二 (SAS EM 關連分析)： Case Study 2 (Association Analysis using SAS EM)

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
8	2016/04/05	教學行政觀摩日 (Off-campus study)
9	2016/04/12	期中報告 (Midterm Project Presentation)
10	2016/04/19	期中考試週 (Midterm Exam)
11	2016/04/26	個案分析與實作三 (SAS EM 決策樹、模型評估) : Case Study 3 (Decision Tree, Model Evaluation using SAS EM)
12	2016/05/03	個案分析與實作四 (SAS EM 迴歸分析、類神經網路) : Case Study 4 (Regression Analysis, Artificial Neural Network using SAS EM)
13	2016/05/10	Google TensorFlow 深度學習 (Deep Learning with Google TensorFlow)
14	2016/05/17	期末報告 (Final Project Presentation)
15	2016/05/24	畢業班考試 (Final Exam)

**LeCun, Yann,
Yoshua Bengio,
and Geoffrey Hinton.**

"Deep learning."

**Nature 521, no. 7553 (2015):
436-444.**

Deep learning

Yann LeCun^{1,2}, Yoshua Bengio³ & Geoffrey Hinton^{4,5}

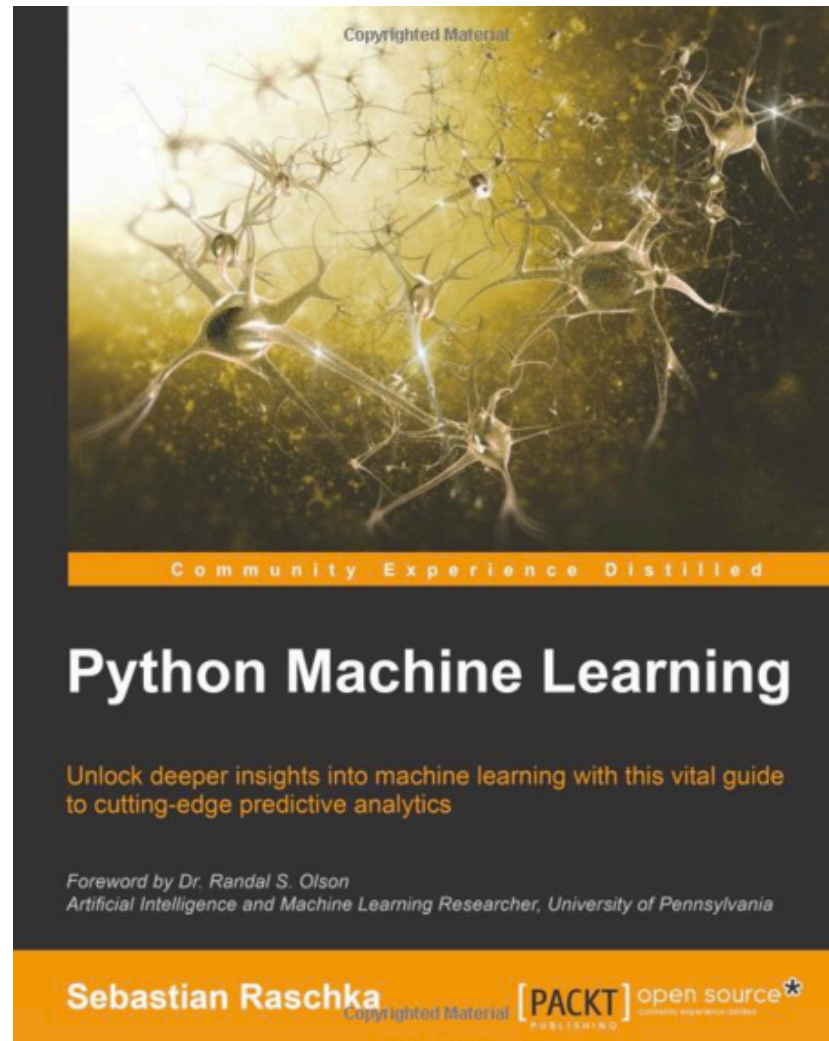
Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other 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.

Machine-learning technology powers many aspects of modern society: from web searches to content filtering on social networks to recommendations on e-commerce websites, and it is increasingly present in consumer products such as cameras and smartphones. Machine-learning systems are used to identify objects in images, transcribe speech into text, match news items, posts or products with users' interests, and select relevant results of search. Increasingly, these applications make use of a class of techniques called deep learning.

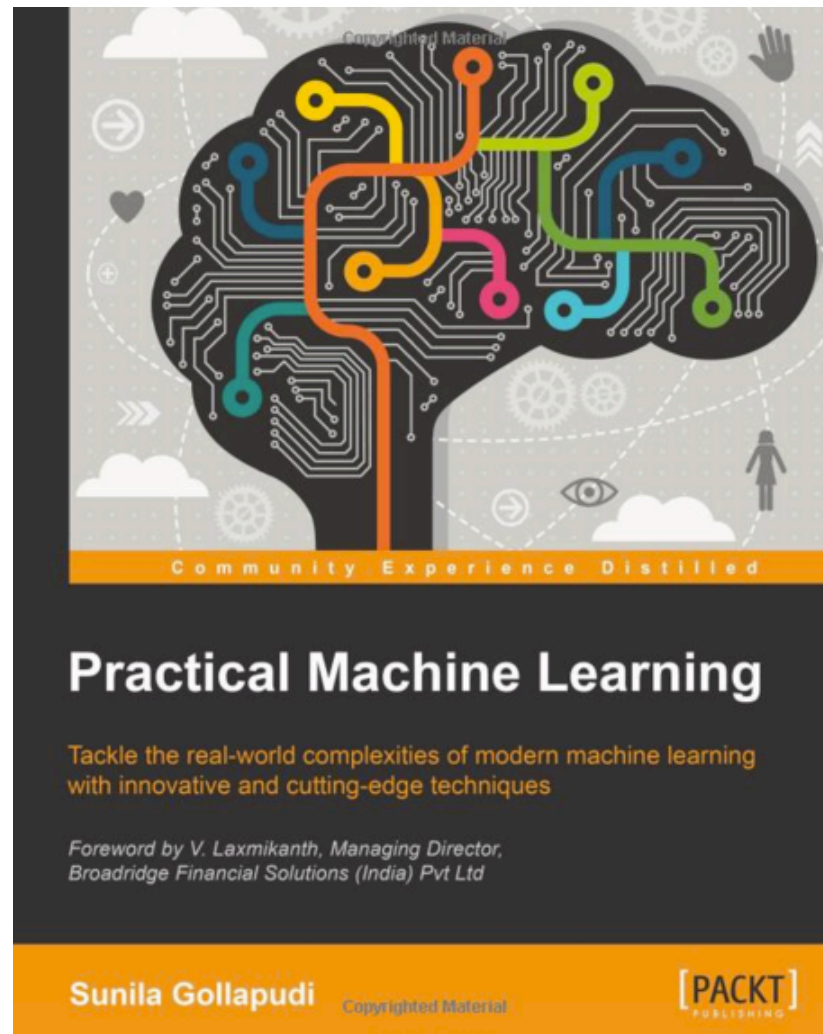
Conventional machine-learning techniques were limited in their ability to process natural data in their raw form. For decades, con-

intricate structures in high-dimensional data and is therefore applicable to many domains of science, business and government. In addition to beating records in image recognition¹⁻⁴ and speech recognition⁵⁻⁷, it has beaten other machine-learning techniques at predicting the activity of potential drug molecules⁸, analysing particle accelerator data^{9,10}, reconstructing brain circuits¹¹, and predicting the effects of mutations in non-coding DNA on gene expression and disease^{12,13}. Perhaps more surprisingly, deep learning has produced extremely promising results for various tasks in natural language understanding¹⁴, particularly topic classification, sentiment analysis, question answering¹⁵ and language translation^{16,17}.

Sebastian Raschka (2015),
Python Machine Learning,
Packt Publishing



Sunila Gollapudi (2016),
Practical Machine Learning,
Packt Publishing



Machine Learning Models

Deep Learning

Association rules

Decision tree

Clustering

Bayesian

Kernel

Ensemble

Dimensionality reduction

Regression Analysis

Instance based

Neural networks (NN) 1960

Multilayer Perceptrons (MLP) 1985

Restricted Boltzmann Machine (RBM) 1986

Support Vector Machine (SVM) 1995



Hinton presents the

Deep Belief Network (DBN)

**New interests in deep learning
and RBM**

State of the art MNIST

2005

Deep Recurrent Neural Network (RNN) 2009

Convolutional DBN 2010

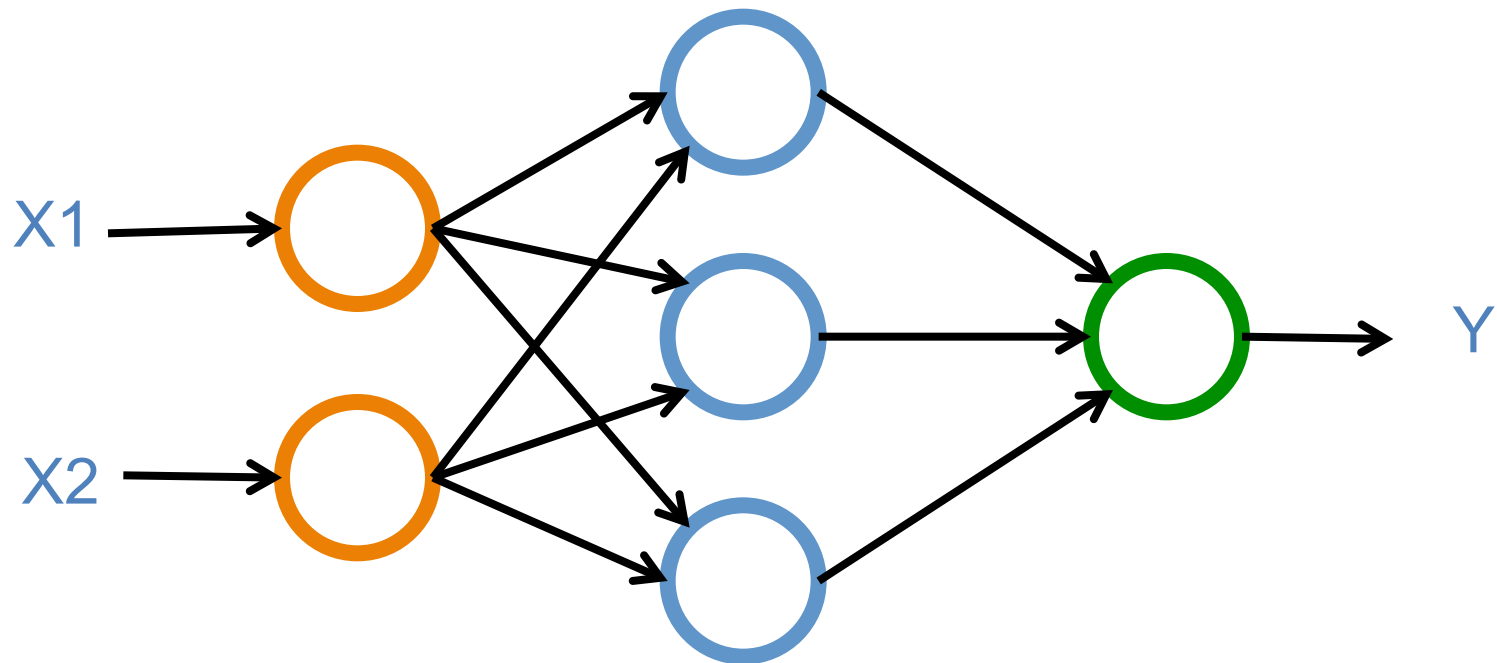
Max-Pooling CDBN 2011

Neural Networks

Input Layer
(X)

Hidden Layer
(H)

Output Layer
(Y)



Deep Learning

Geoffrey Hinton

Yann LeCun

Yoshua Bengio

Andrew Y. Ng



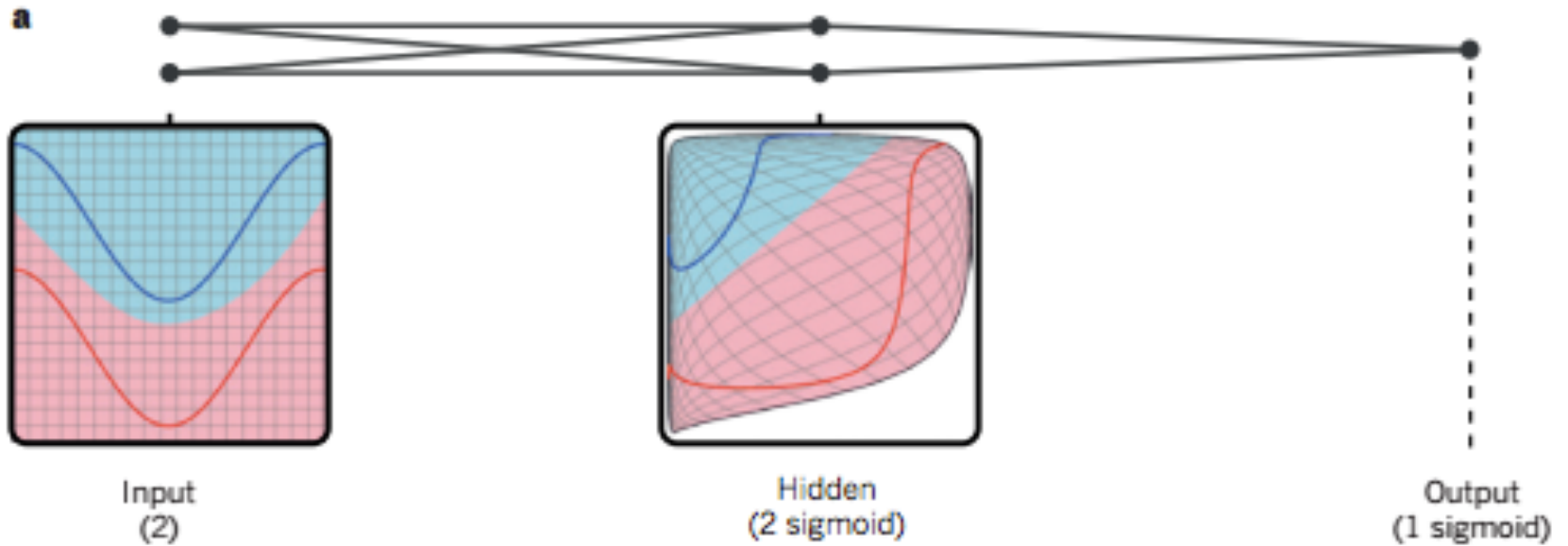
Geoffrey Hinton
Google
University of Toronto

**LeCun, Yann,
Yoshua Bengio,
and Geoffrey Hinton.**

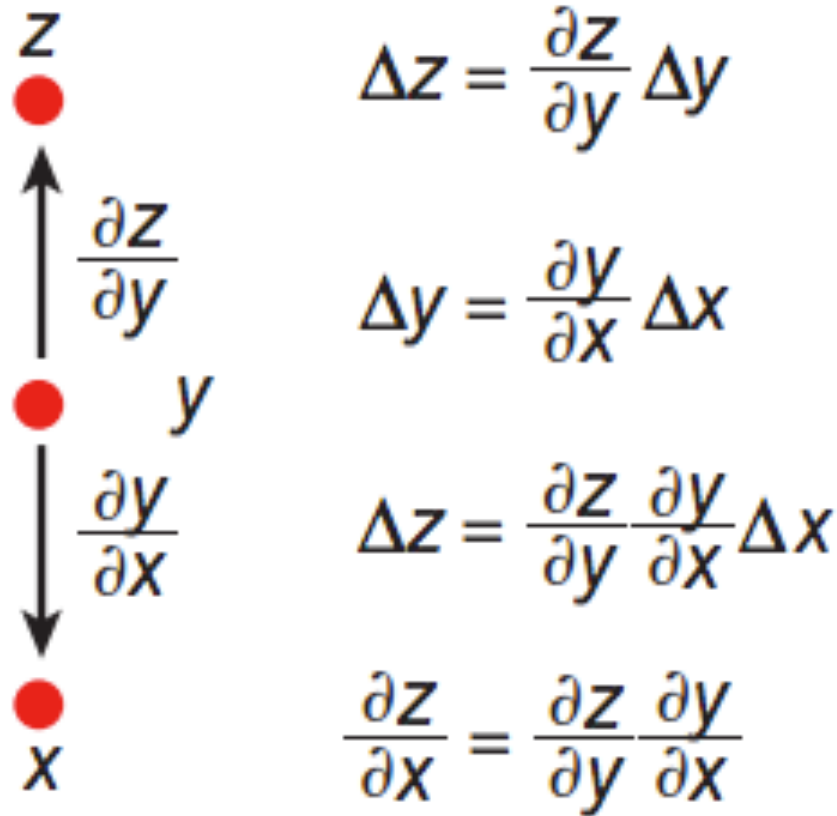
"Deep learning."

**Nature 521, no. 7553 (2015):
436-444.**

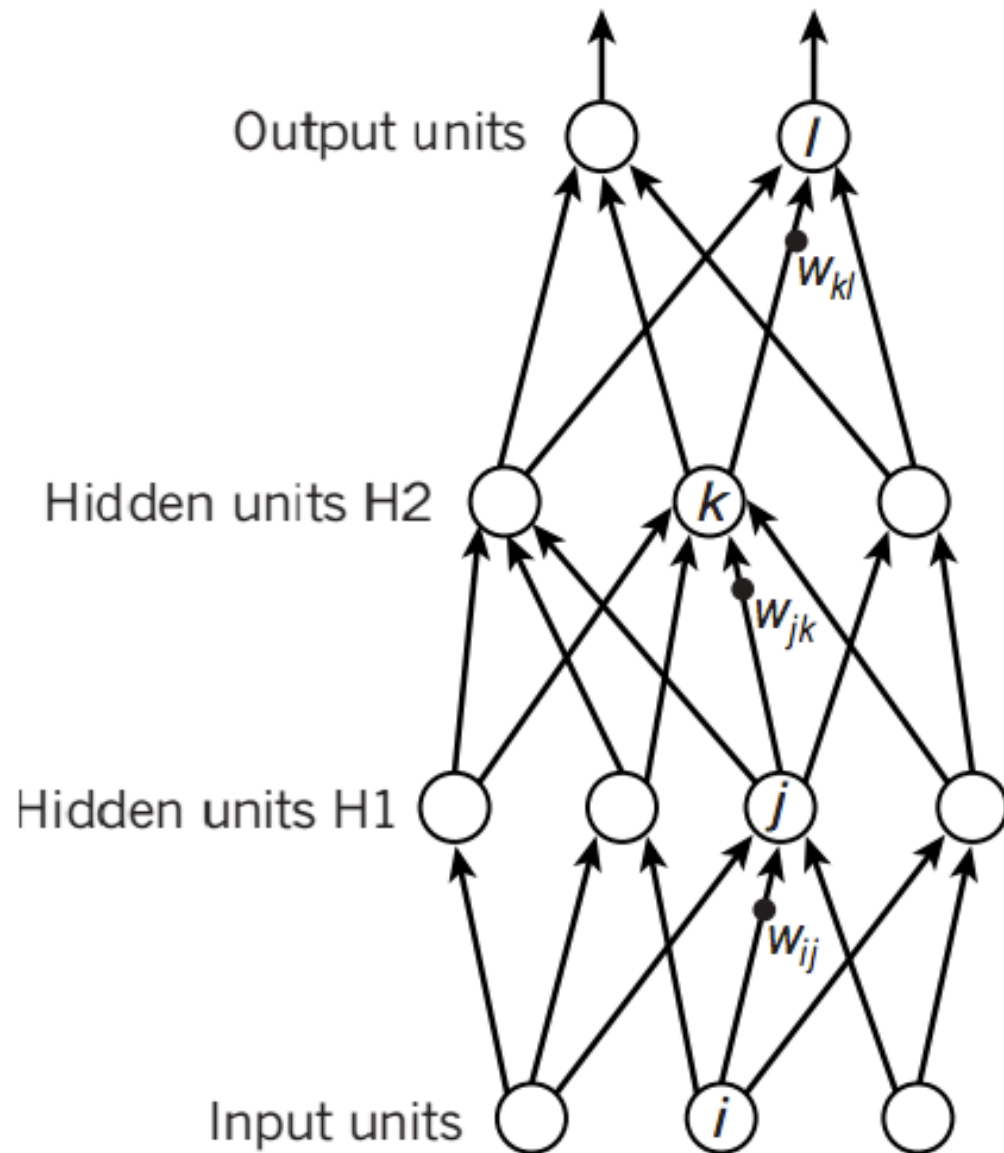
Deep Learning



Deep Learning



Deep Learning



$$y_l = f(z_l)$$

$$z_l = \sum_{k \in H2} w_{kl} y_k$$

$$y_k = f(z_k)$$

$$z_k = \sum_{j \in H1} w_{jk} y_j$$

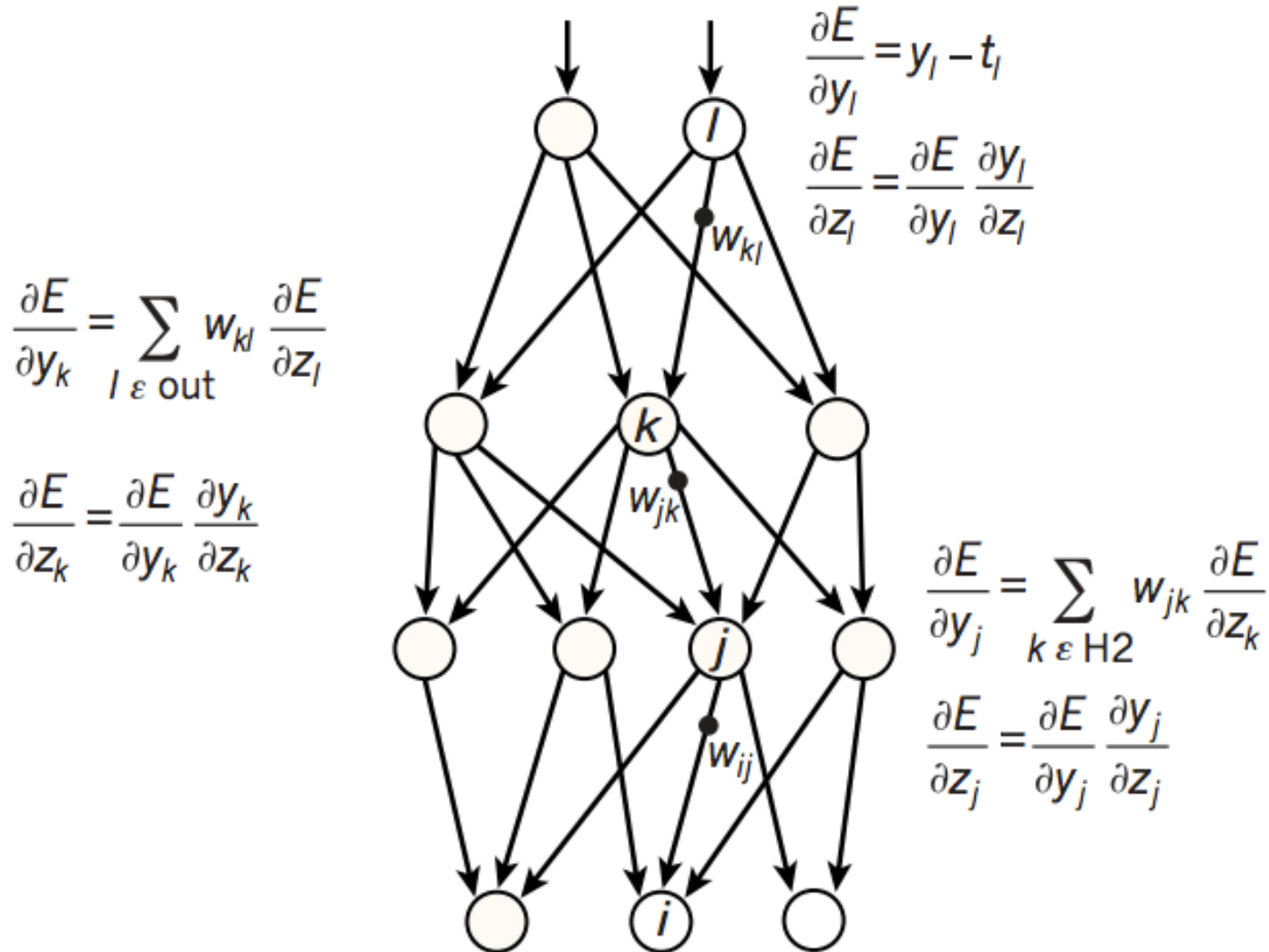
$$y_j = f(z_j)$$

$$z_j = \sum_{i \in \text{Input}} w_{ij} x_i$$

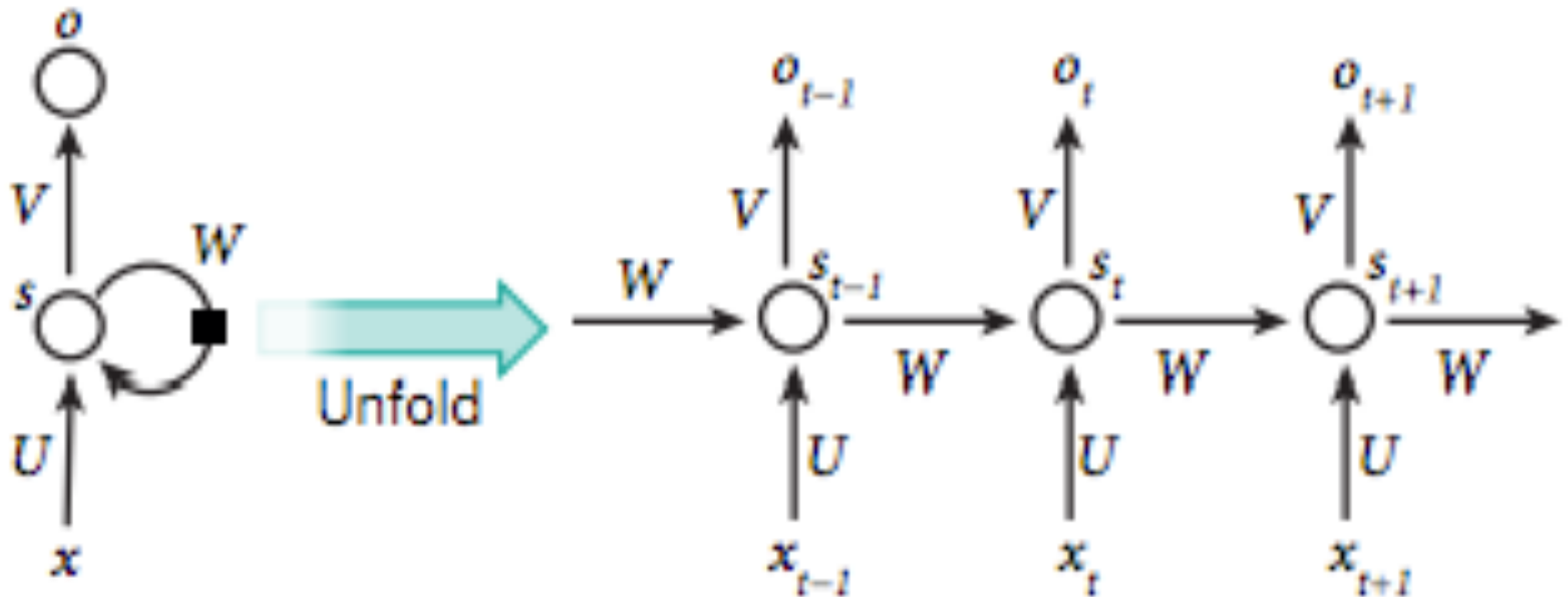
Deep Learning

d

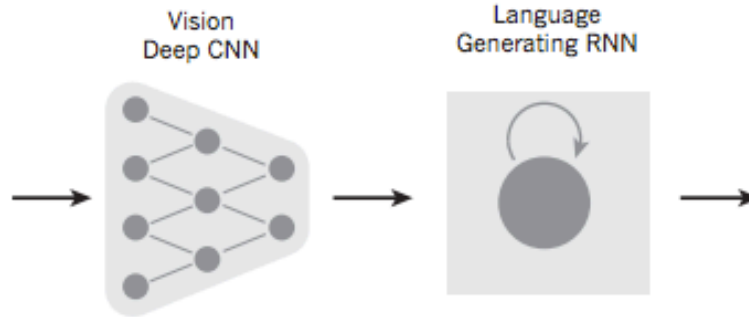
Compare outputs with correct answer to get error derivatives



Recurrent Neural Network (RNN)



From image to text



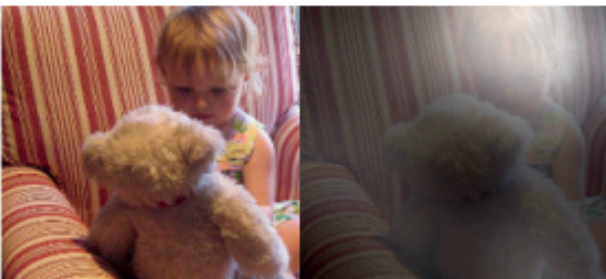
A woman is throwing a **frisbee** in a park.



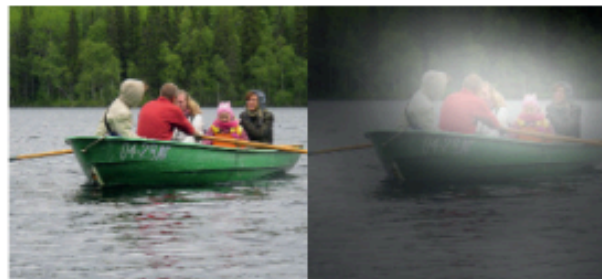
A **dog** is standing on a hardwood floor.



A **stop** sign is on a road with a mountain in the background



A little **girl** sitting on a bed with a teddy bear.



A group of **people** sitting on a boat in the water.



A giraffe standing in a forest with **trees** in the background.

From image to text

Image: deep convolution neural network (CNN)

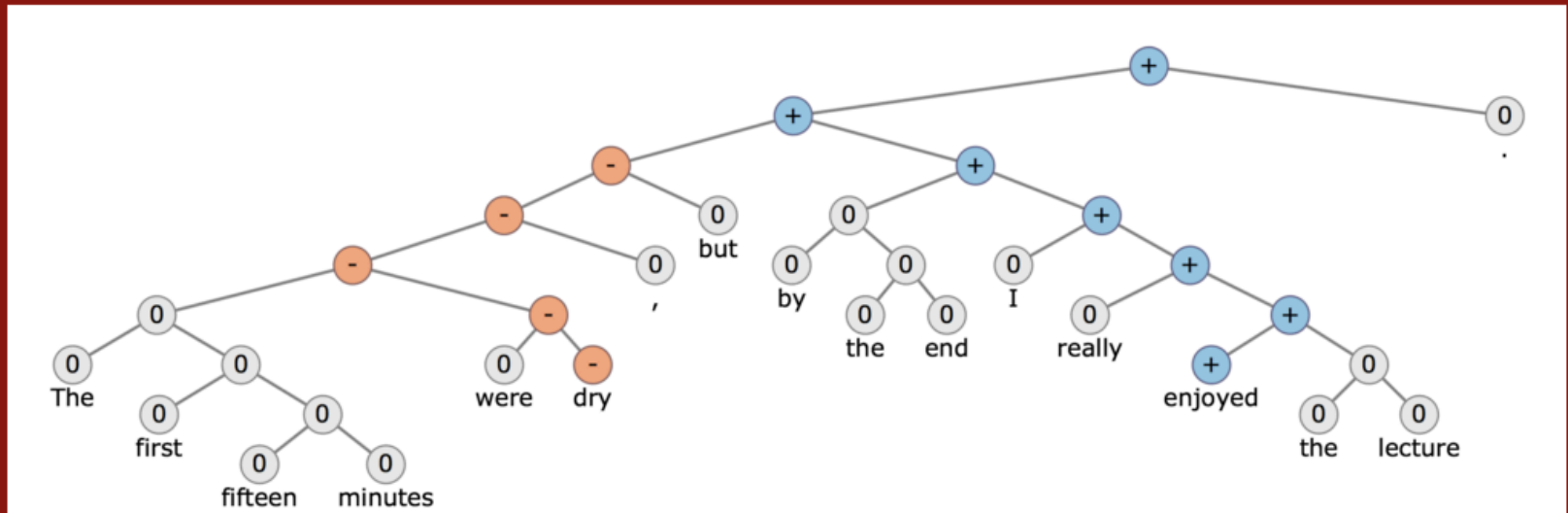
Text: recurrent neural network (RNN)



A group of **people** sitting on a boat in the water.

CS224d: Deep Learning for Natural Language Processing

CS224d: Deep Learning for Natural Language Processing

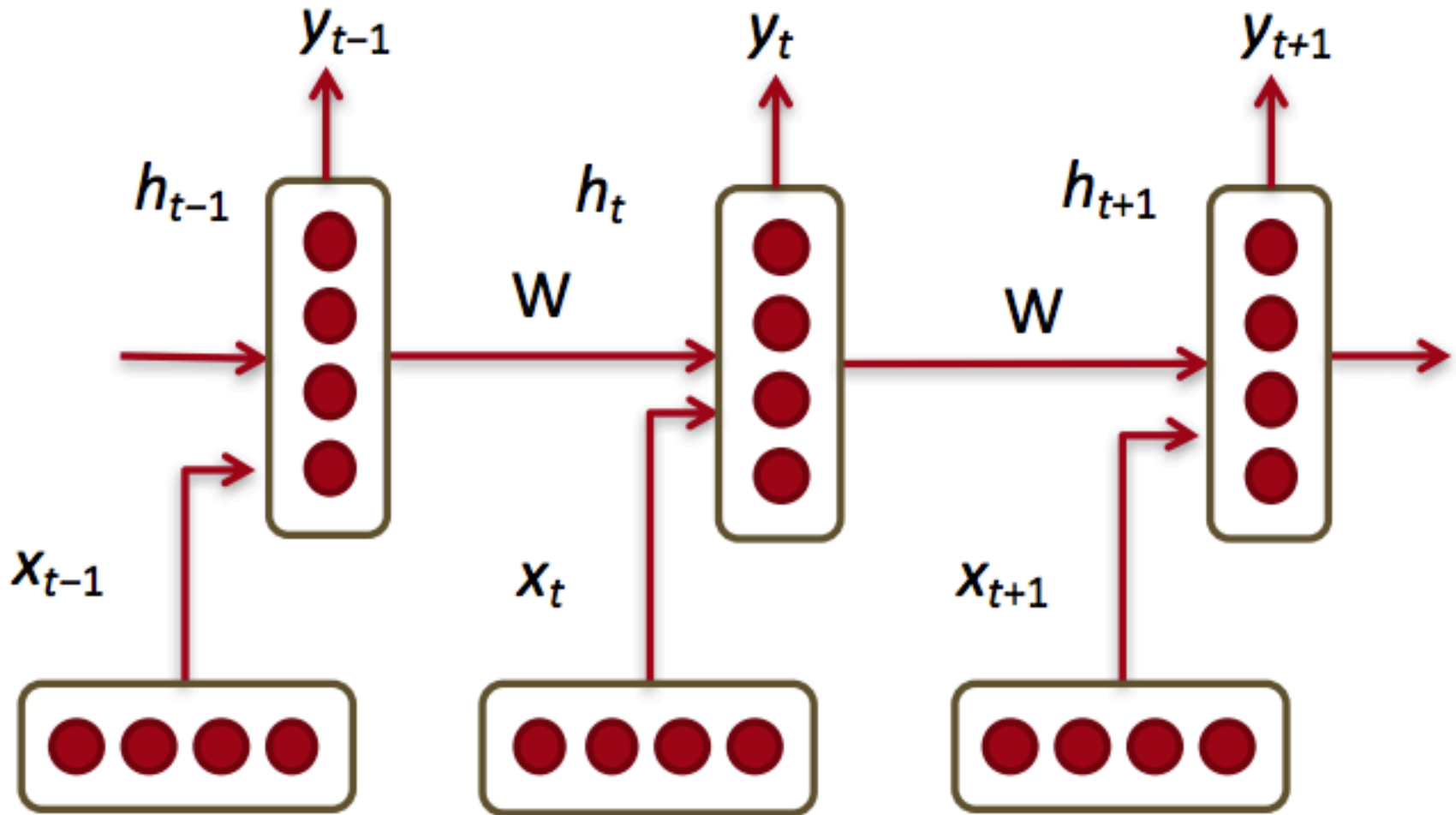


Course Description

Natural language processing (NLP) is one of the most important technologies of the information age. Understanding complex language utterances is also a crucial part of artificial intelligence. Applications of NLP are everywhere because people communicate most everything in language: web search, advertisement, emails, customer service, language translation, radiology reports, etc. There are a large variety of underlying tasks and machine learning models powering NLP applications. Recently, deep learning approaches have obtained very high performance across many different NLP tasks. These models can often be trained with a single end-to-end model and do not require traditional, task-specific feature engineering. In this spring quarter course students will learn to implement, train, debug, visualize and invent their own neural network models. The course provides a deep excursion into cutting-edge research in deep learning applied to NLP. The final project will involve training a complex recurrent neural network and applying it to a large scale NLP problem. On the model side we will cover word vector representations,

<http://cs224d.stanford.edu/>

Recurrent Neural Networks (RNNs)

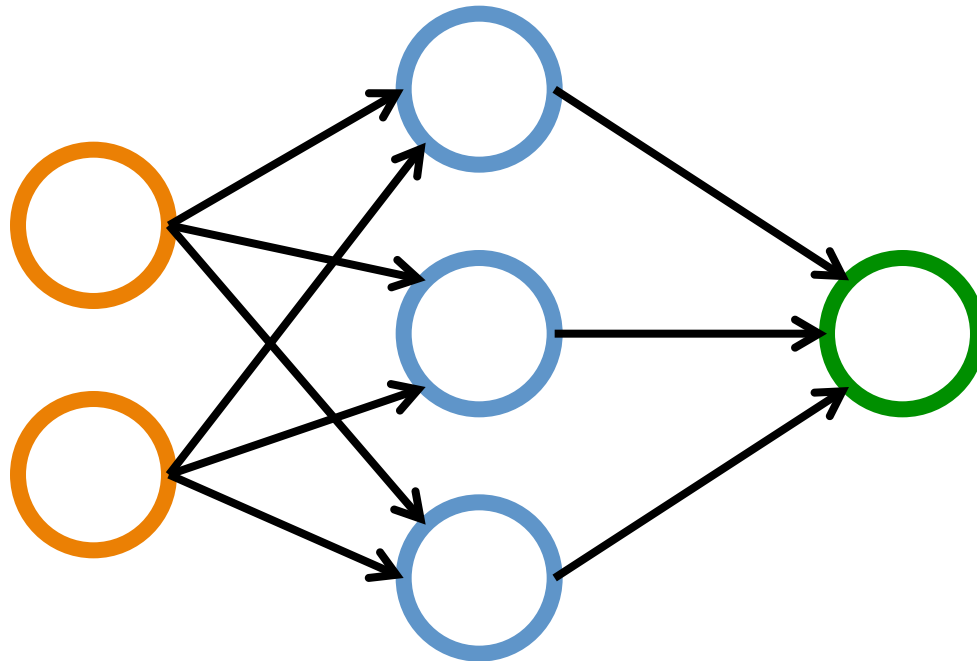


X		Y
Hours Sleep	Hours Study	Score
3	5	75
5	1	82
10	2	93
8	3	?

	X		Y
	Hours Sleep	Hours Study	Score
Training	3	5	75
	5	1	82
	10	2	93
Testing	8	3	?

Training a Network
=
Minimize the Cost Function

Neural Networks

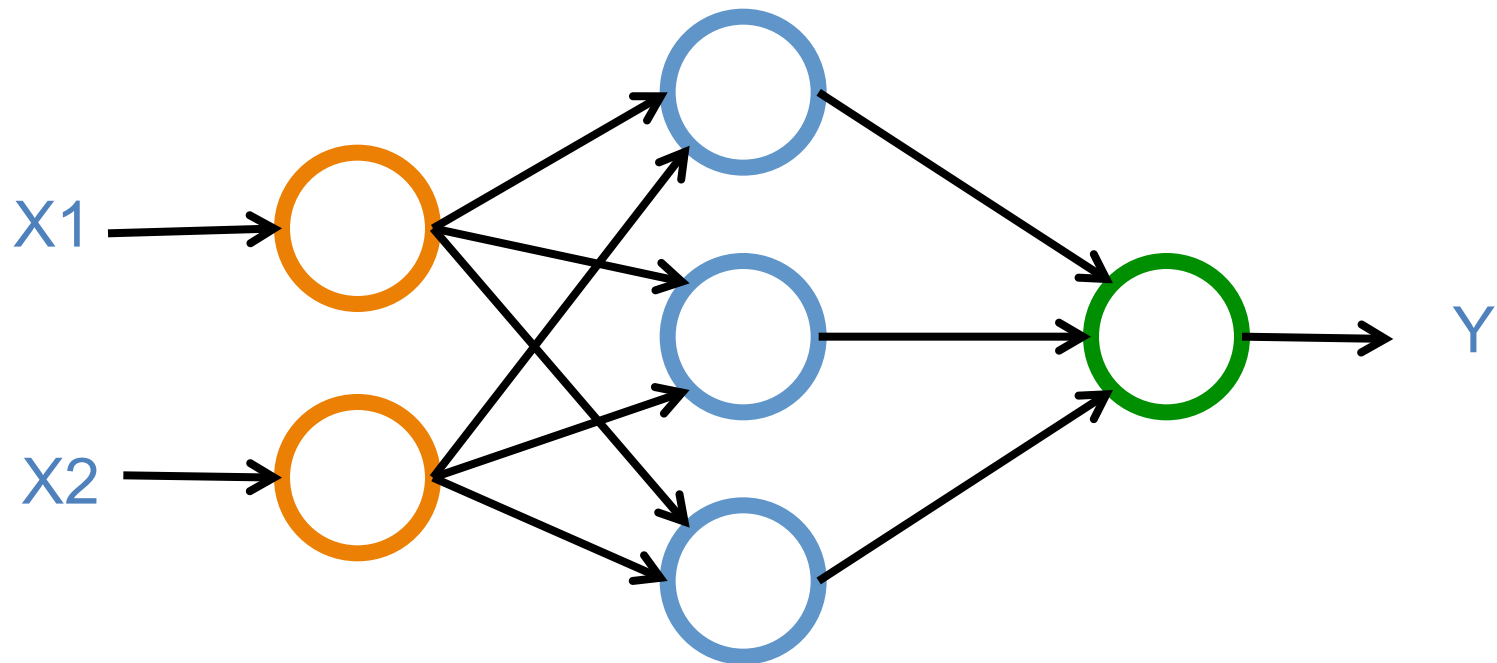


Neural Networks

Input Layer
(X)

Hidden Layer
(H)

Output Layer
(Y)



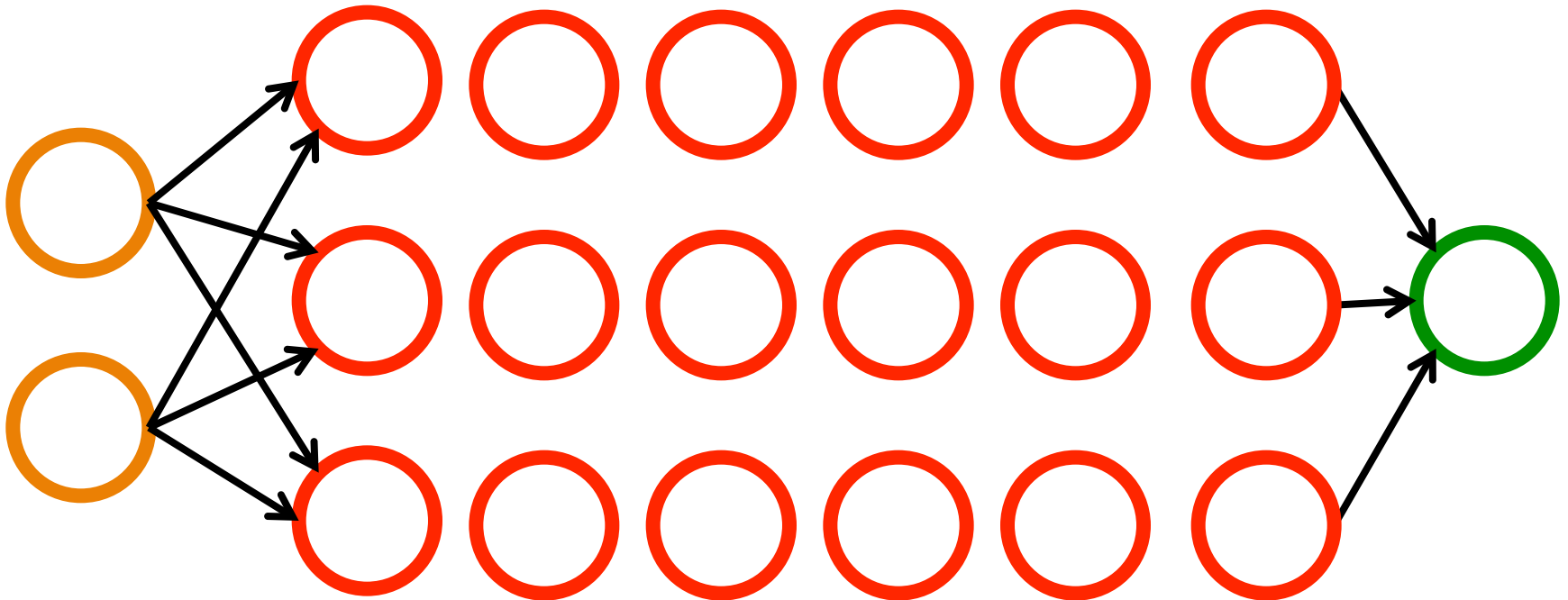
Neural Networks

Input Layer
(X)

Hidden Layers
(H)

Output Layer
(Y)

Deep Neural Networks
Deep Learning

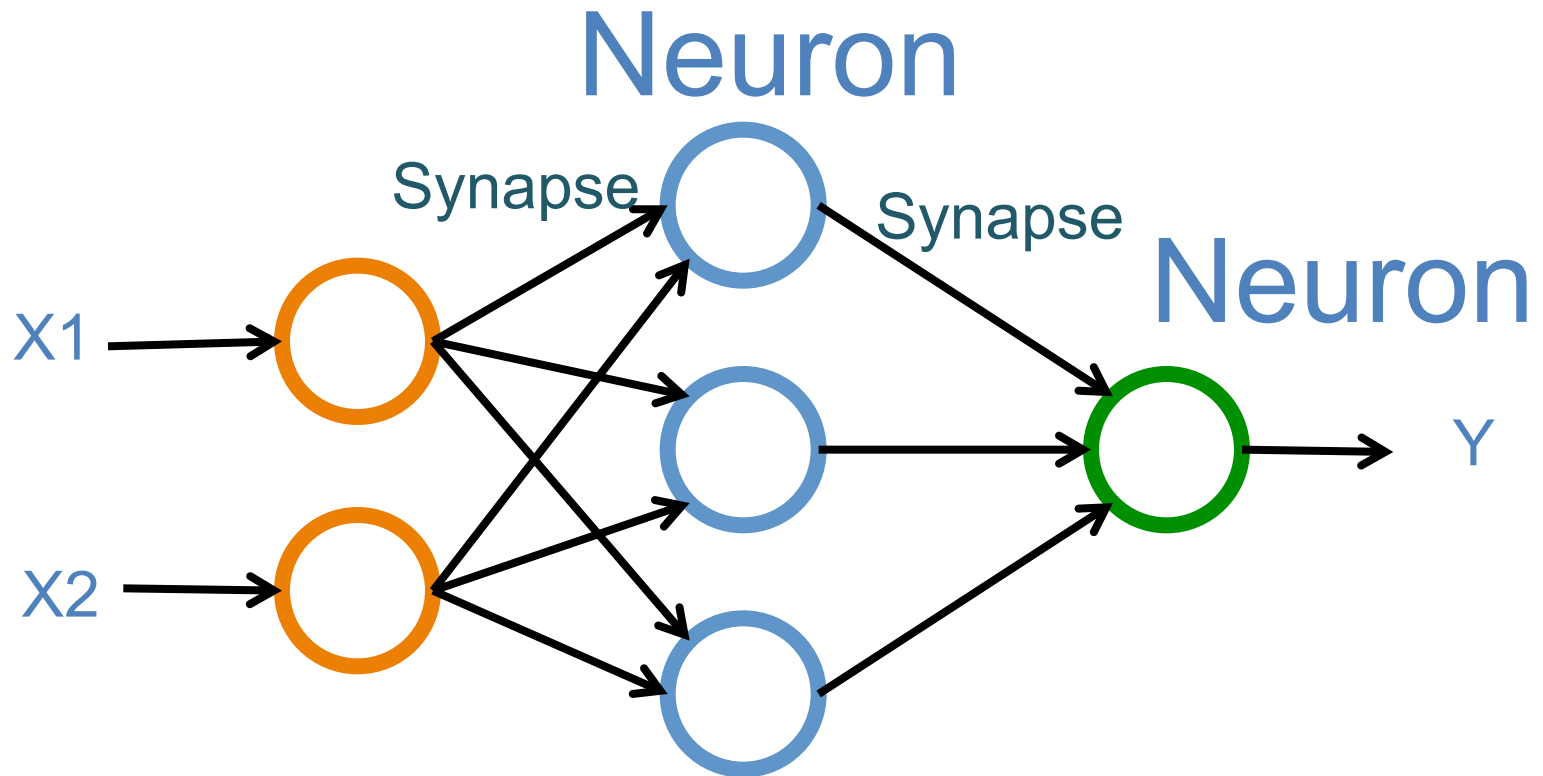


Neural Networks

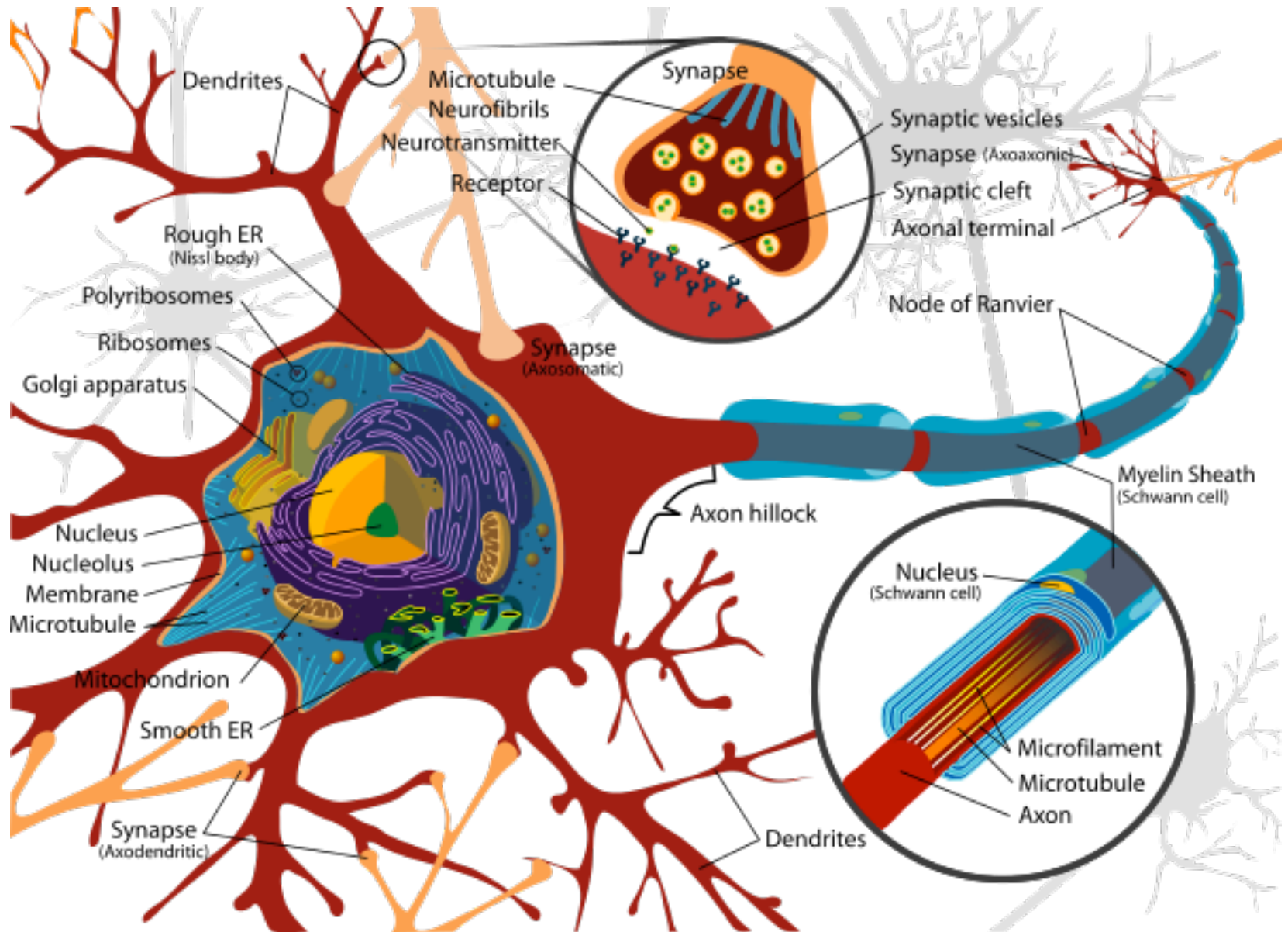
Input Layer
(X)

Hidden Layer
(H)

Output Layer
(Y)



Neuron and Synapse

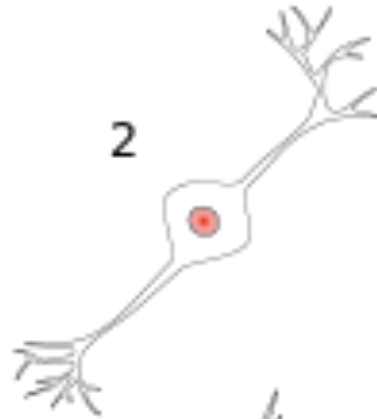


Neurons

1 Unipolar neuron



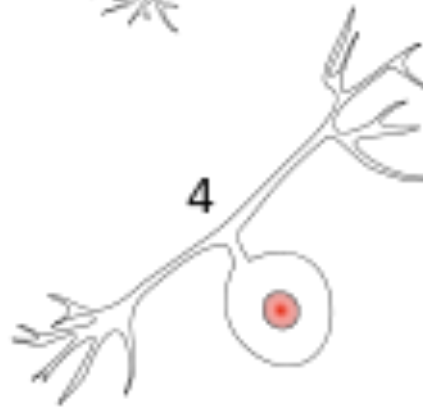
2 Bipolar neuron



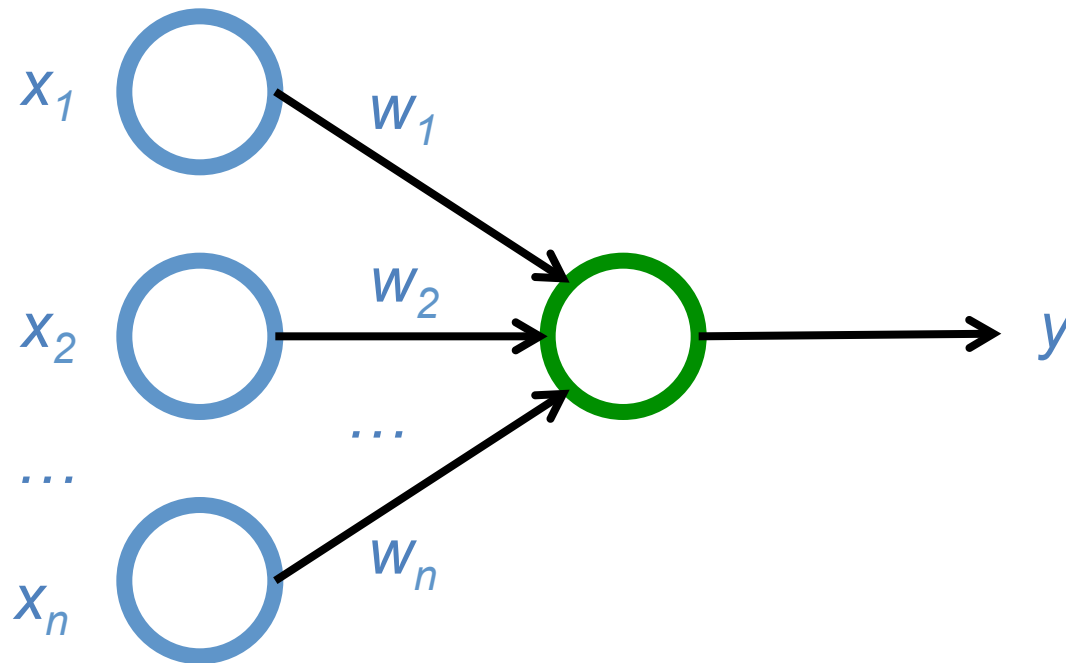
3 Multipolar neuron



4 Pseudounipolar neuron

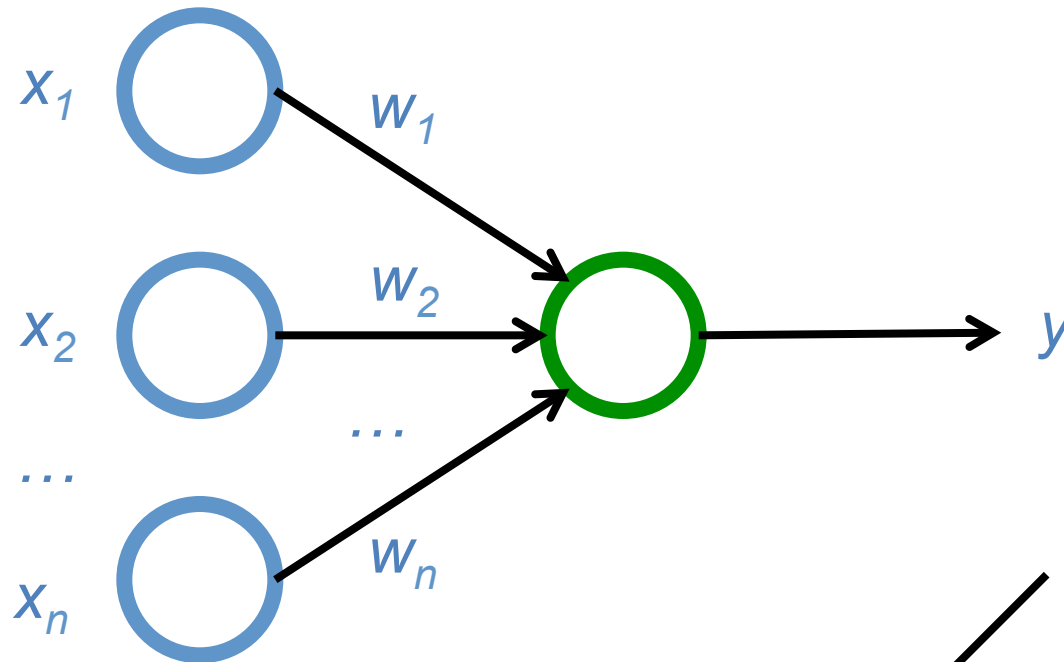


The Neuron



The Neuron

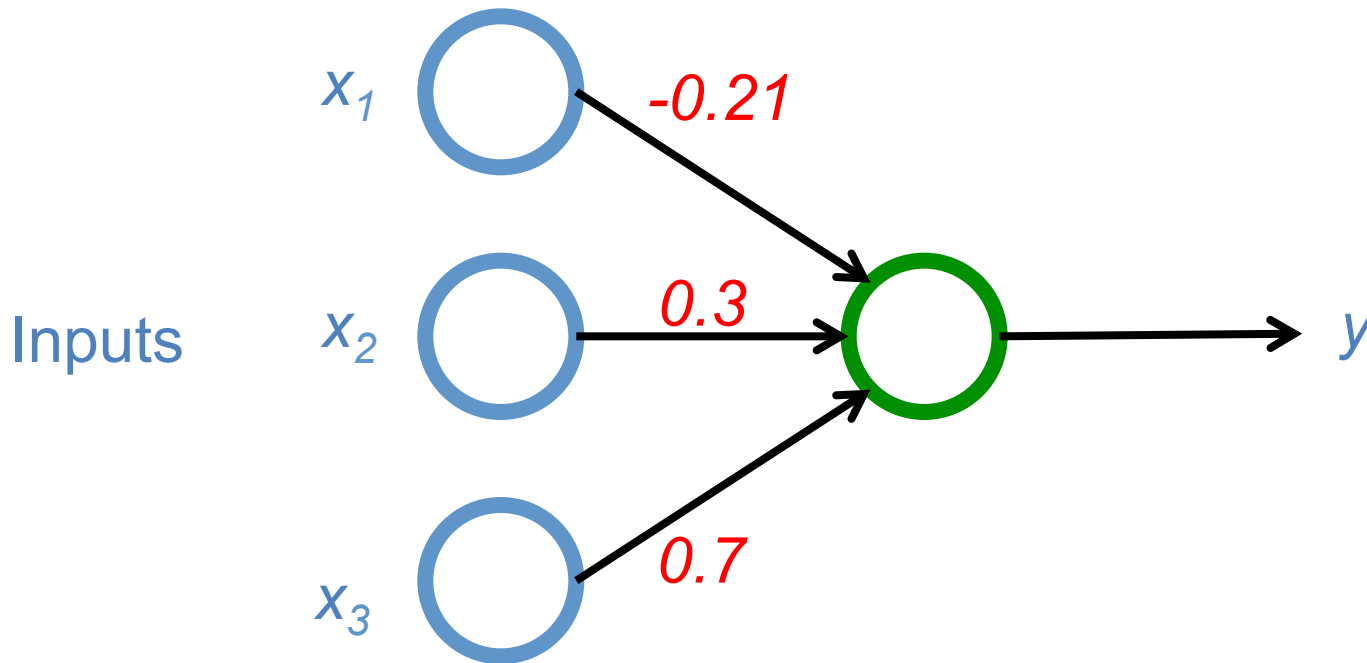
$$y = F\left(\sum_i w_i x_i\right)$$



$$F(x) = \max(0, x)$$

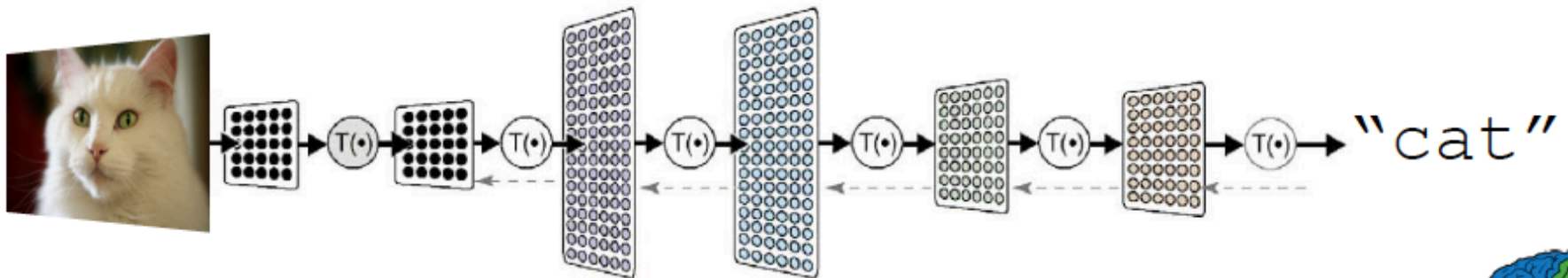
$$y = \max (0, -0.21 * x_1 + 0.3 * x_2 + 0.7 * x_3)$$

Weights



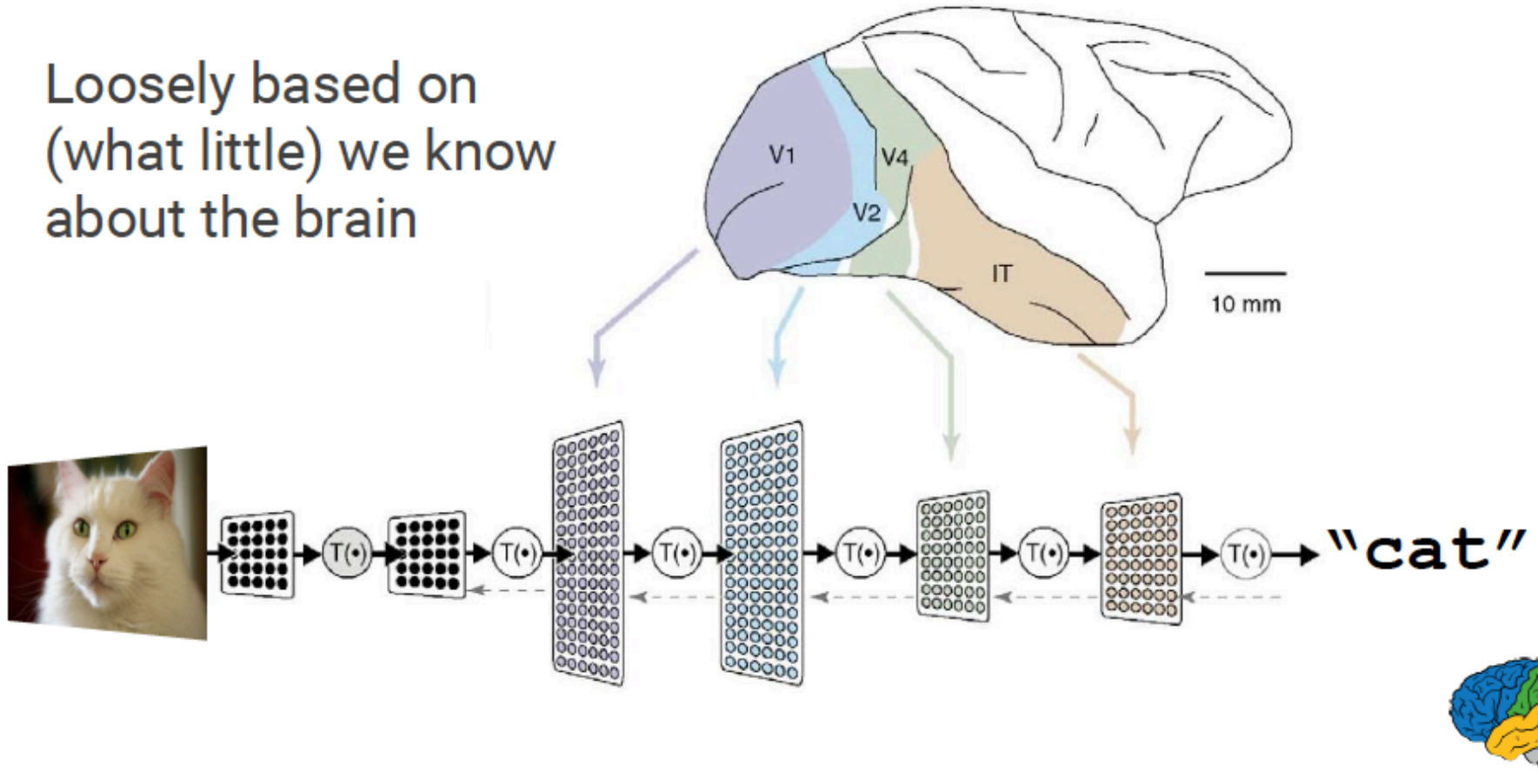
Deep Learning

- A powerful class of **machine learning** model
- **Modern reincarnation** of **artificial neural networks**
- Collection of simple, trainable mathematical functions
- Compatible with many variants of machine learning



What is Deep Learning?

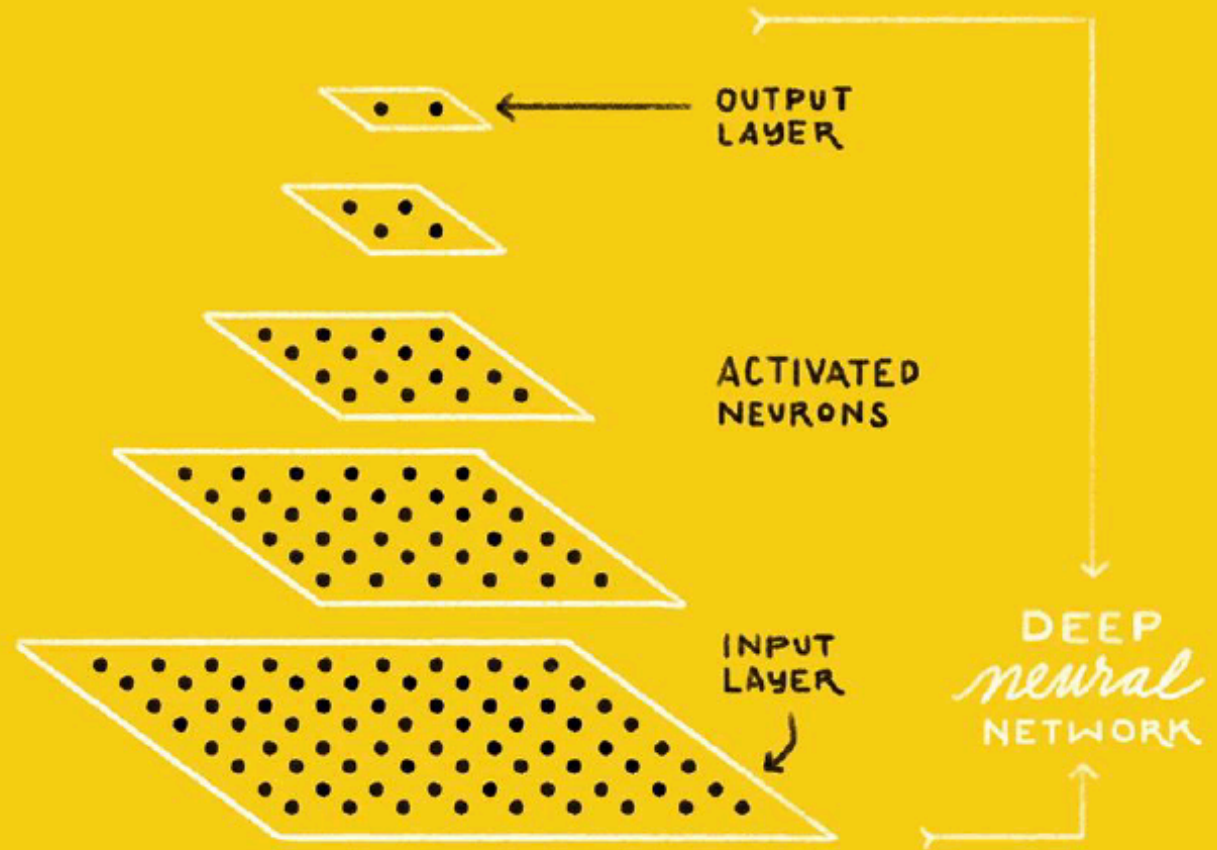
- Loosely based on (what little) we know about the brain



IS THIS A
CAT or **DOG**?



CAT **DOG**



Learning Algorithm

While not done:

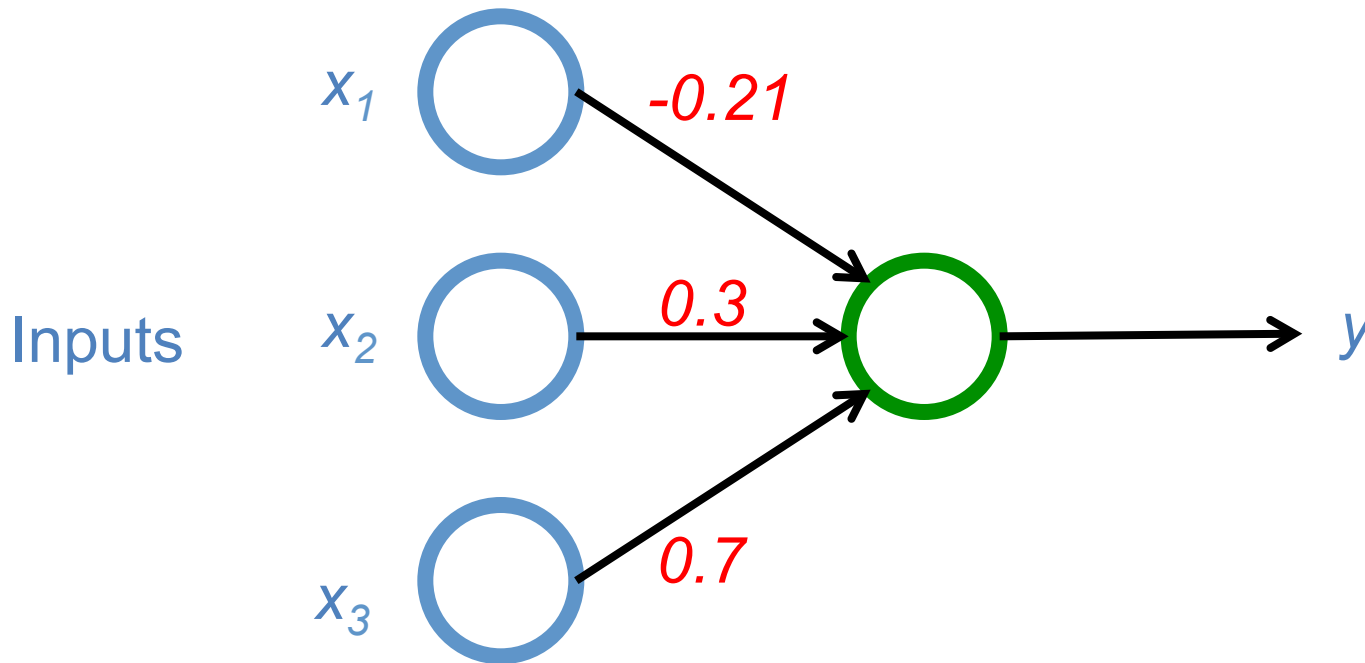
Pick a random training example “(input, label)”

Run neural network on “input”

Adjust weights on edges to make output closer to “label”

$$y = \max (0, -0.21 * x_1 + 0.3 * x_2 + 0.7 * x_3)$$

Weights

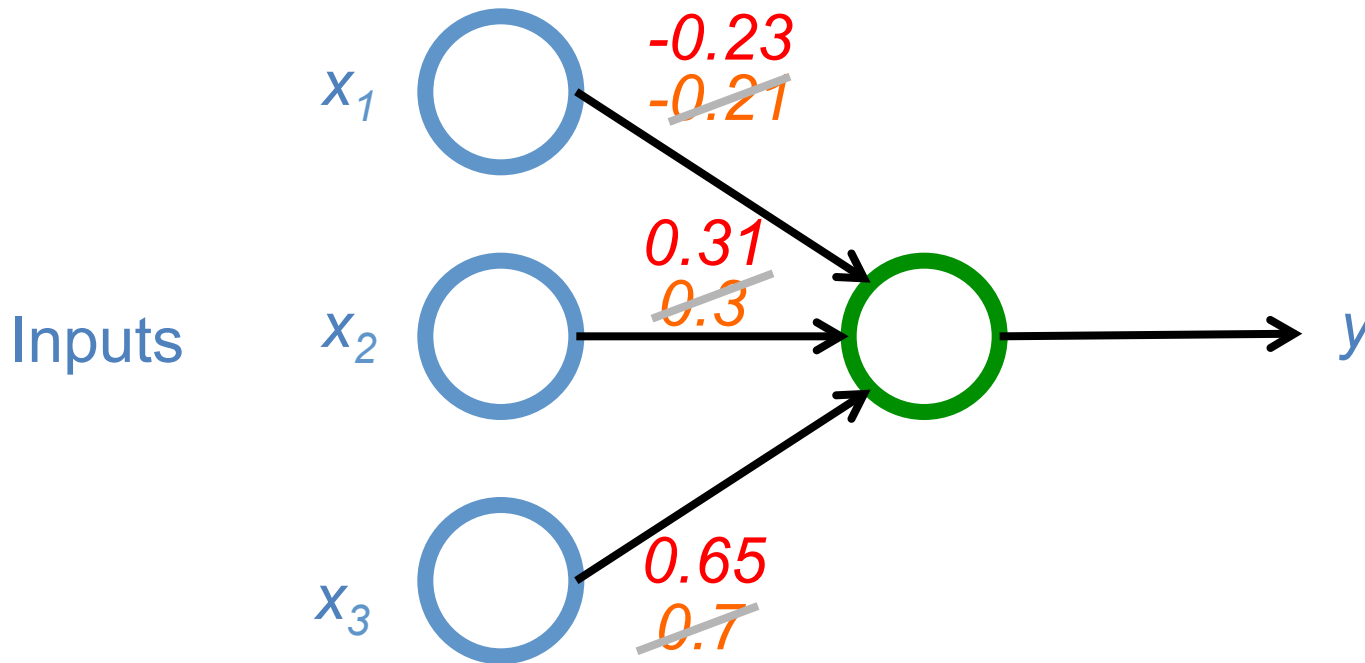


Next time:

$$y = \max(0, -0.23 * x_1 + 0.31 * x_2 + 0.65 * x_3)$$

~~$$y = \max(0, -0.21 * x_1 + 0.3 * x_2 + 0.7 * x_3)$$~~

Weights

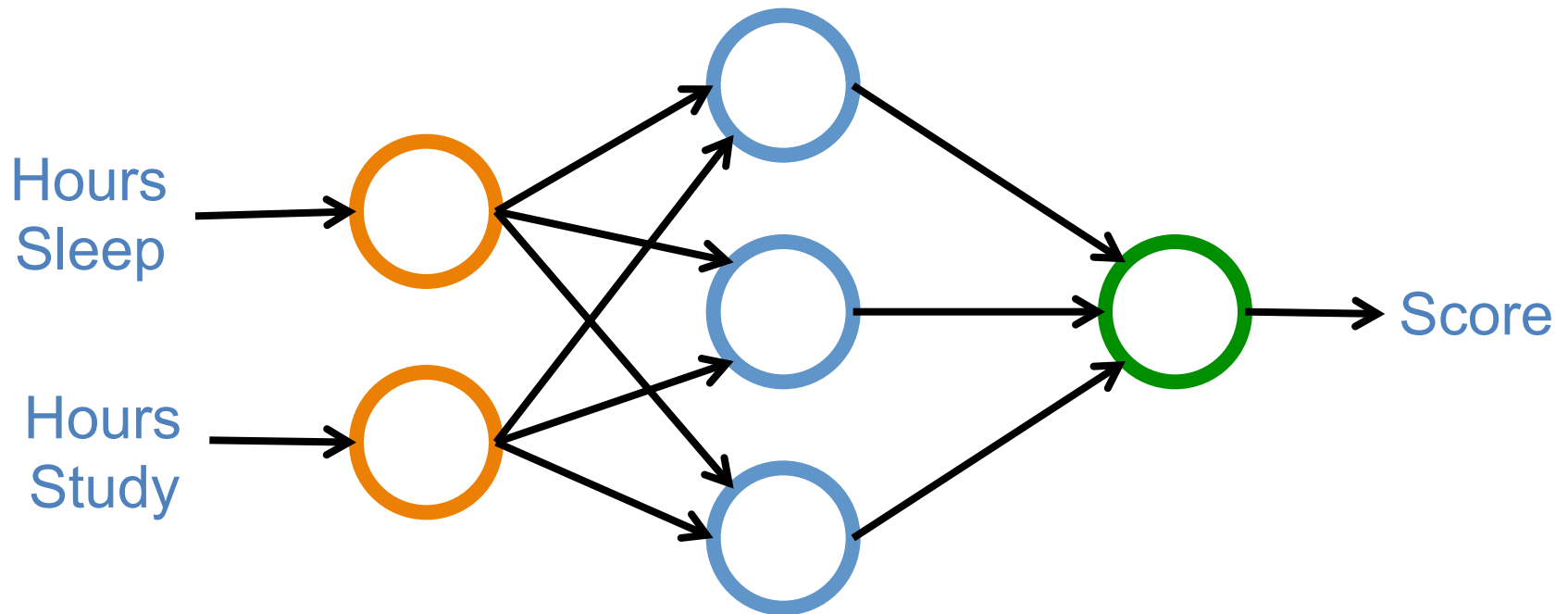


Neural Networks

Input Layer
(X)

Hidden Layer
(H)

Output Layer
(Y)

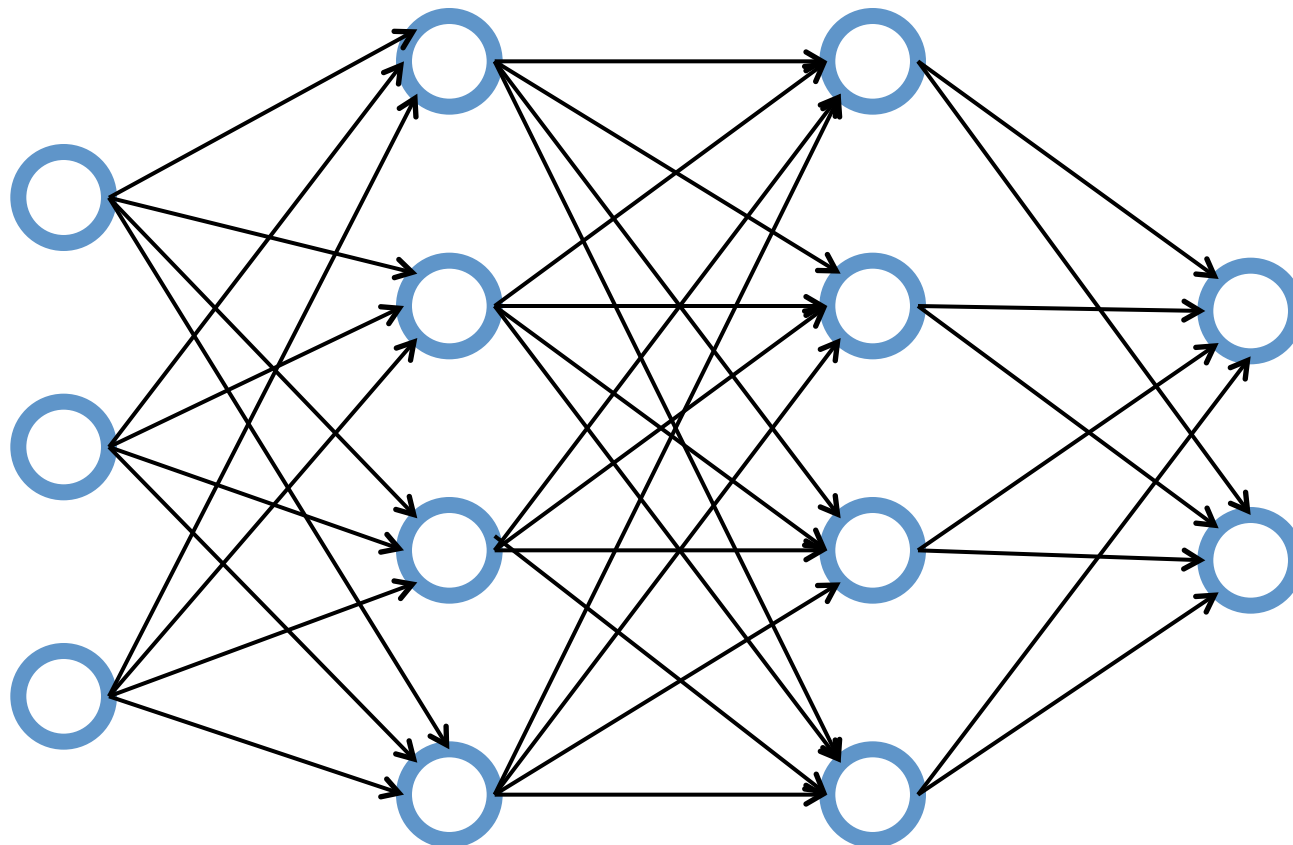


Neural Networks

Input Layer
(X)

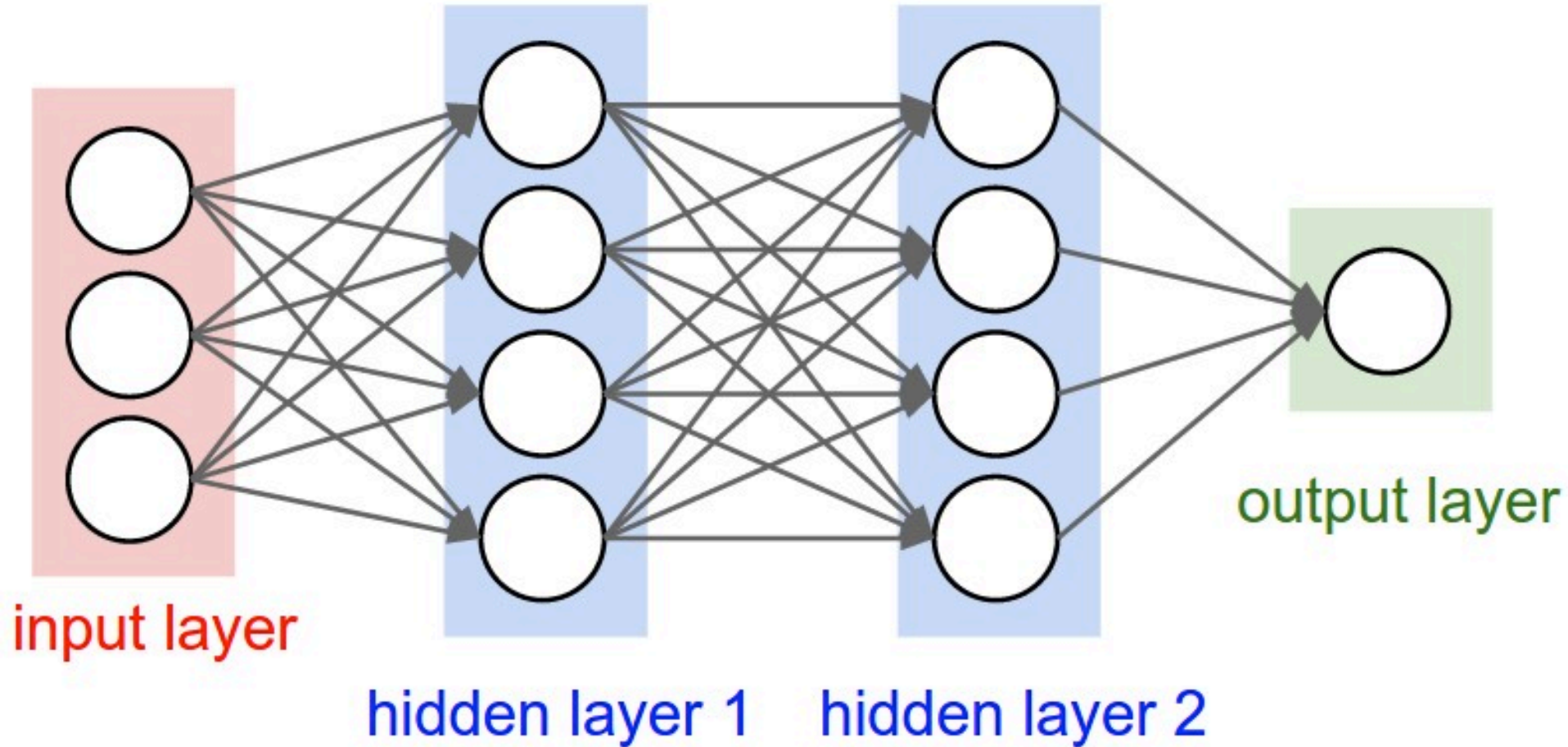
Hidden Layer
(H)

Output Layer
(Y)

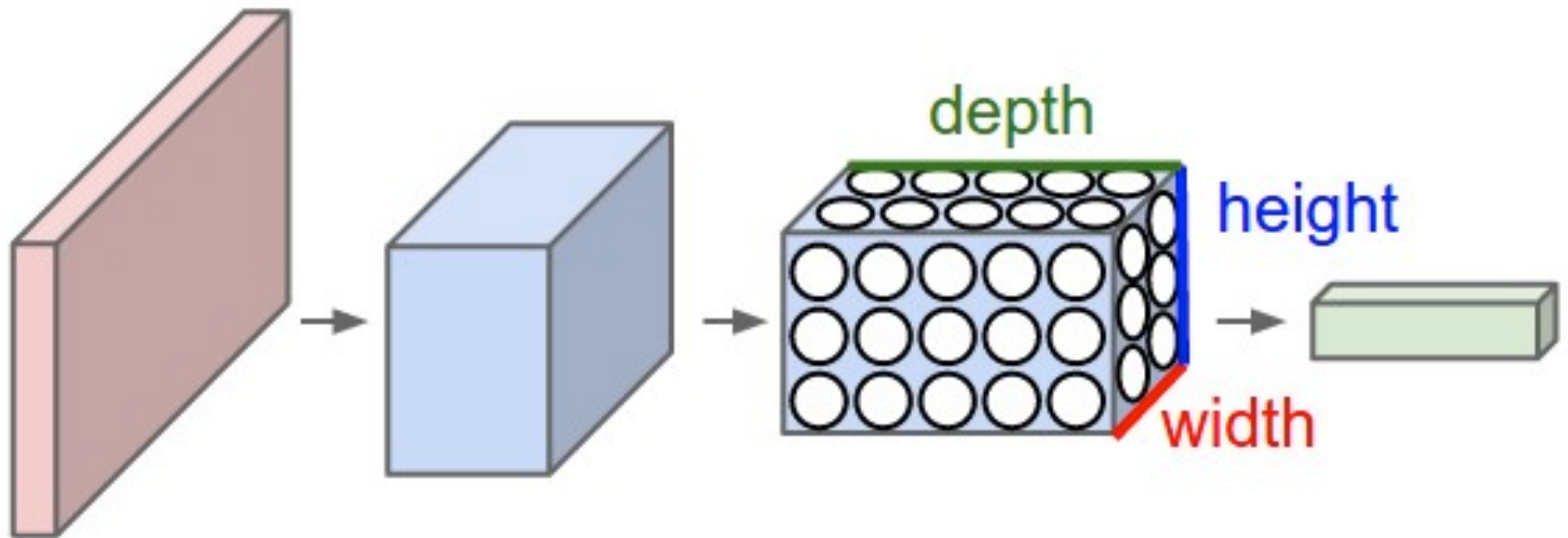


Convolutional Neural Networks (CNNs / ConvNets)

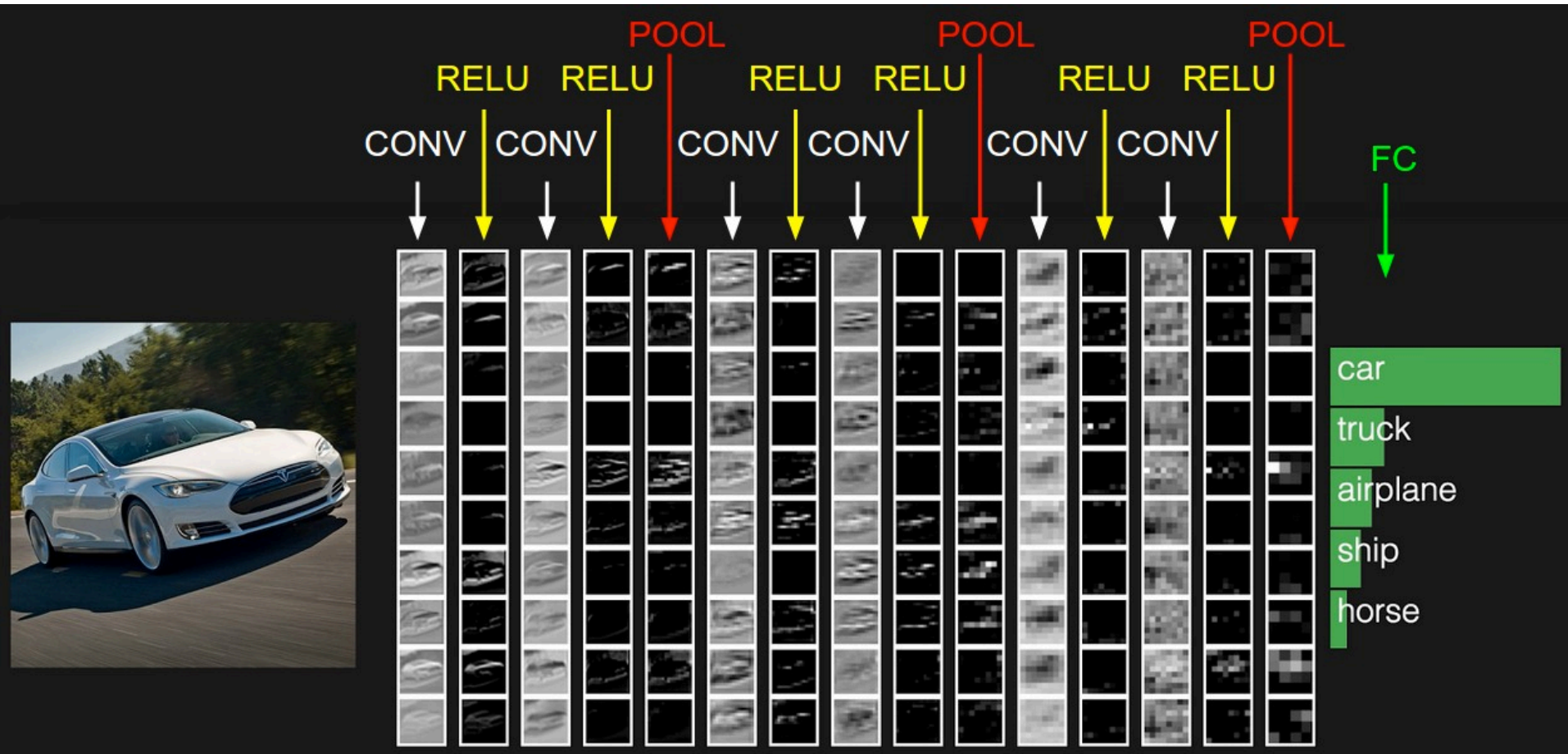
A regular 3-layer Neural Network



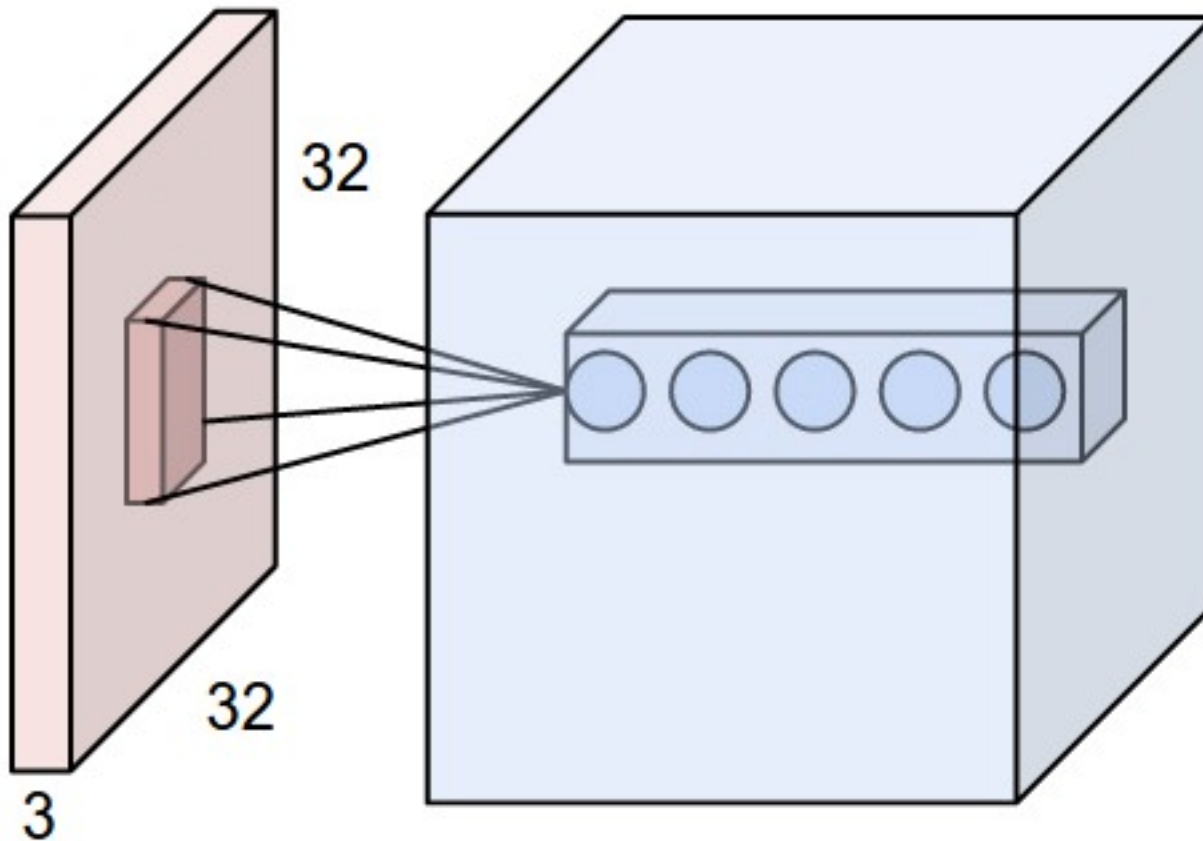
A ConvNet arranges its neurons in three dimensions (width, height, depth)



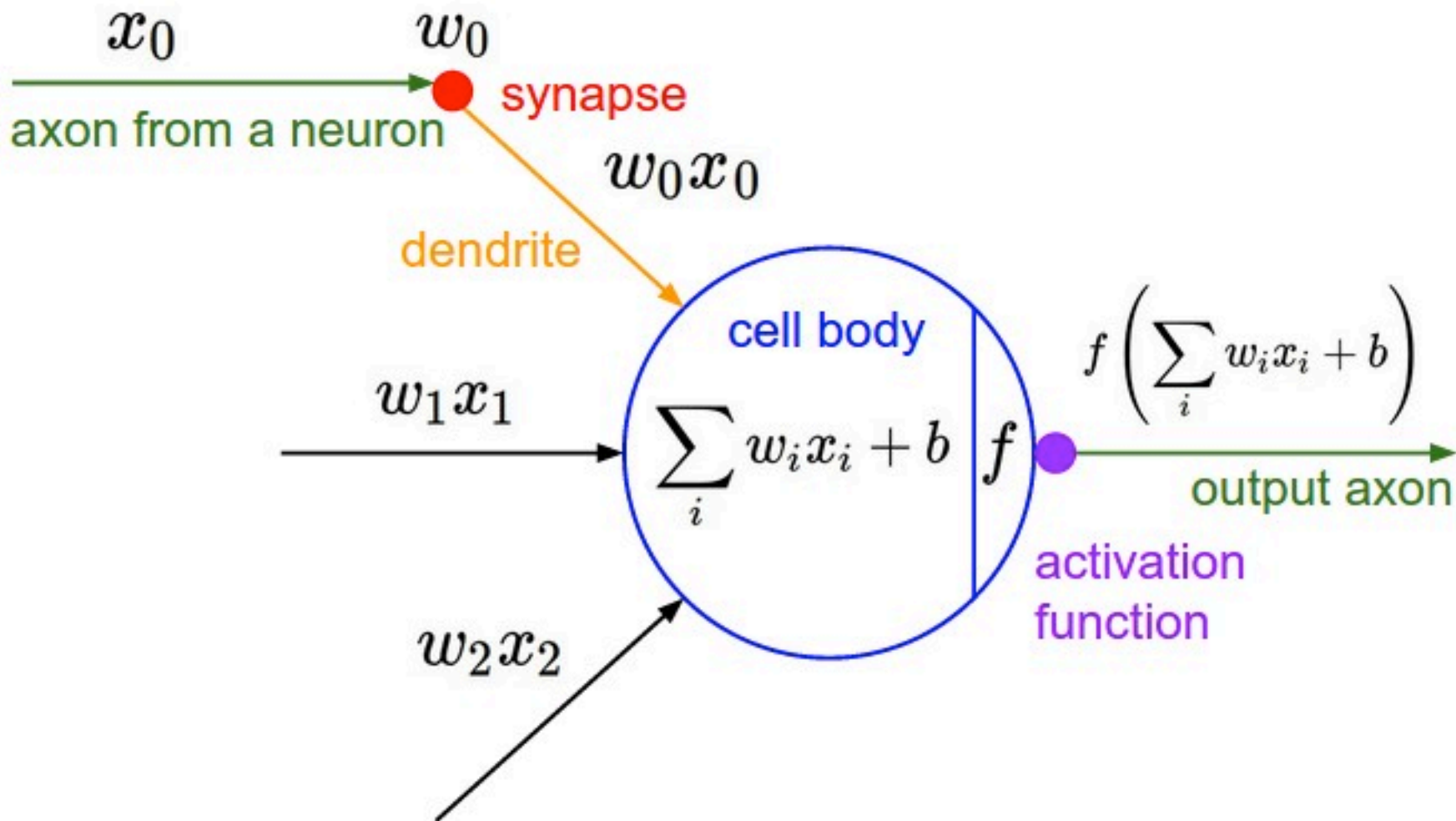
The activations of an example ConvNet architecture.



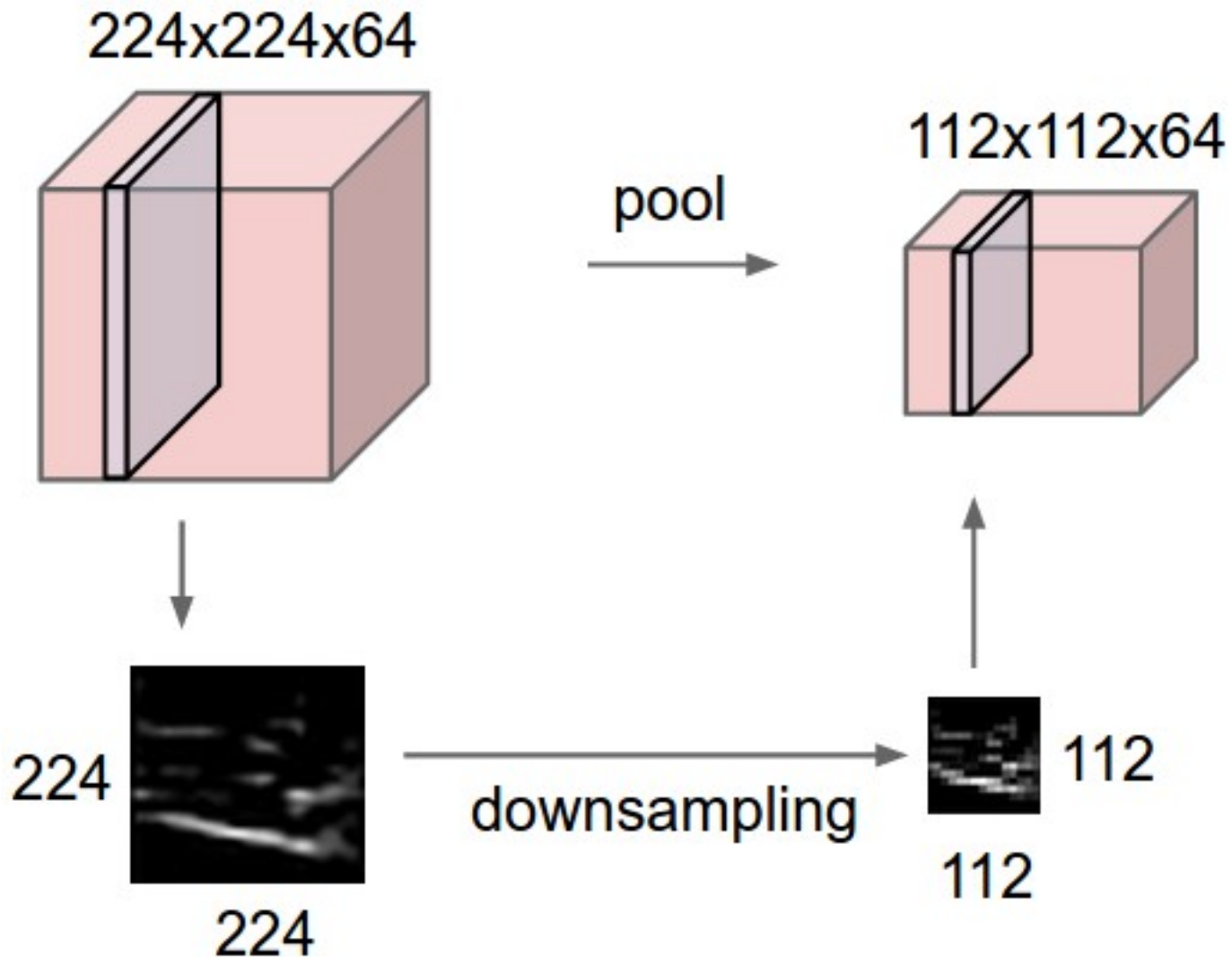
ConvNets



ConvNets



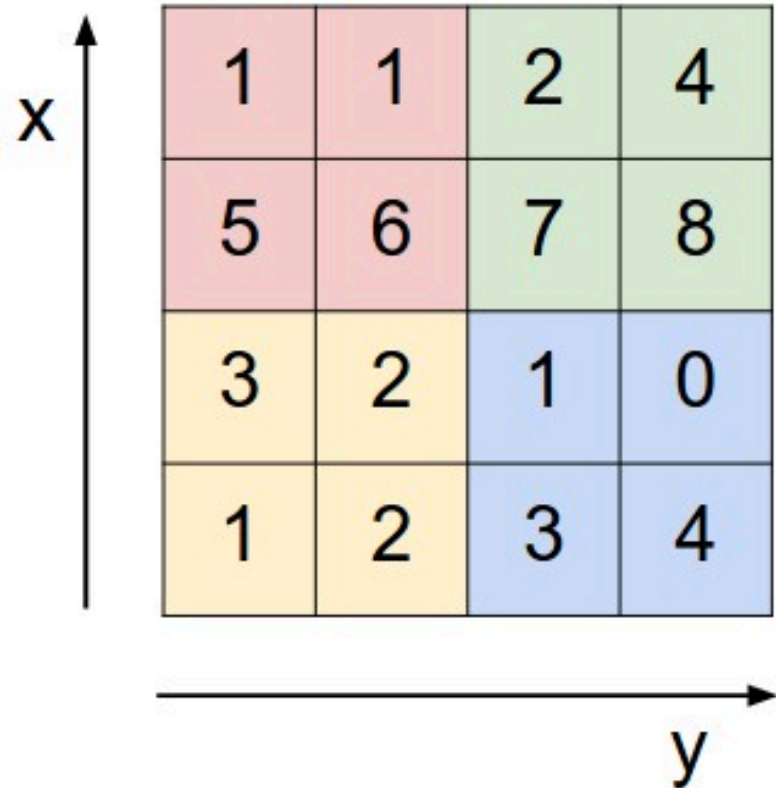
ConvNets



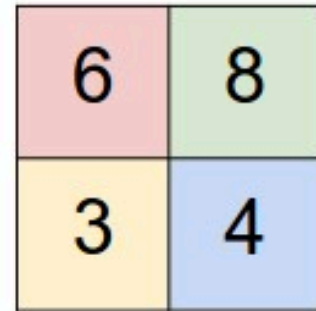
ConvNets

max pooling

Single depth slice



max pool with 2x2 filters
and stride 2

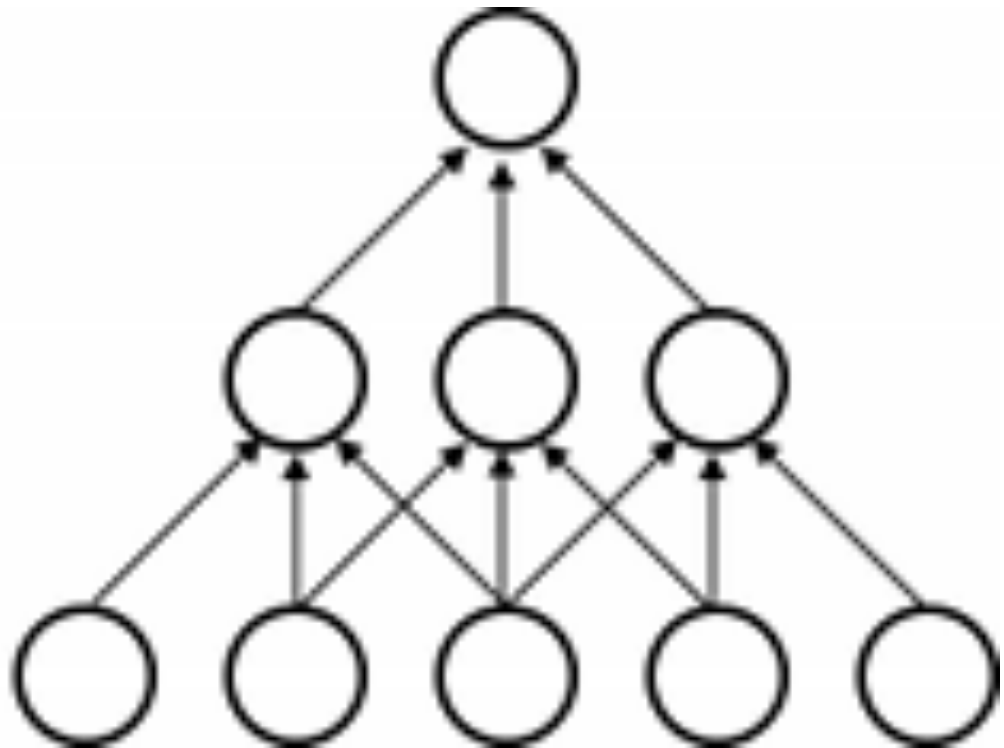


Convolutional Neural Networks (CNN) (LeNet) Sparse Connectivity

layer $m+1$

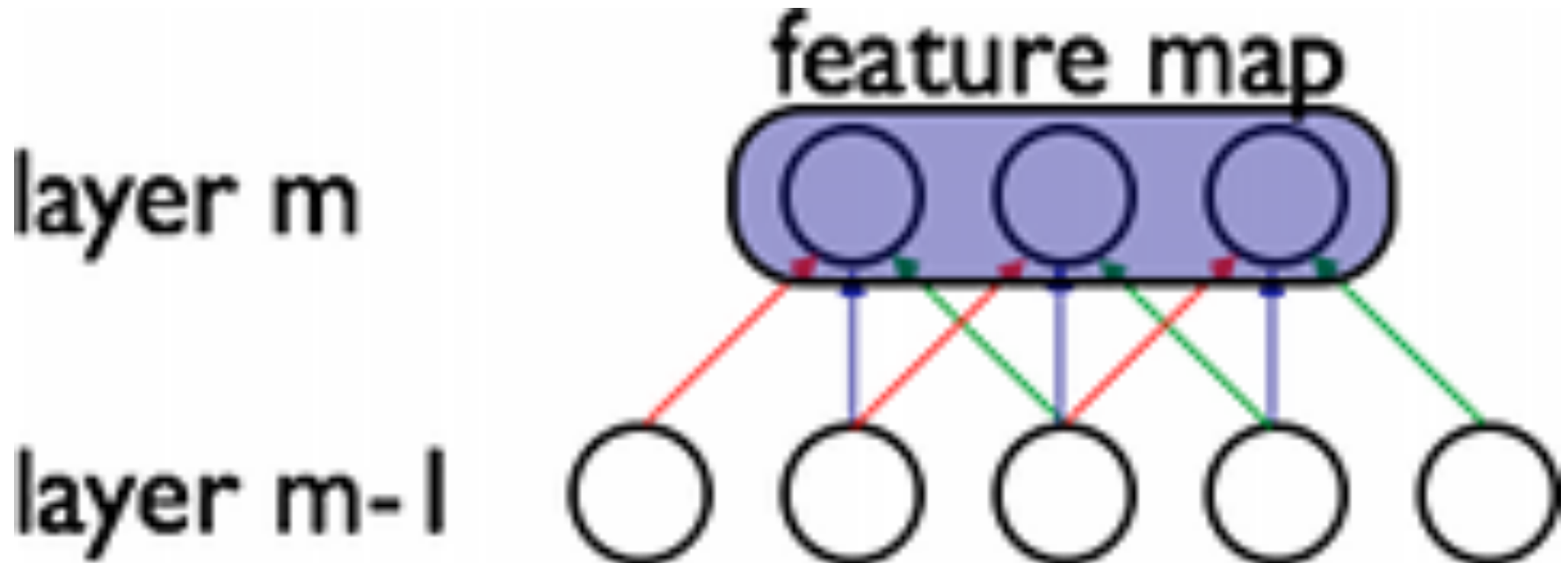
layer m

layer $m-1$



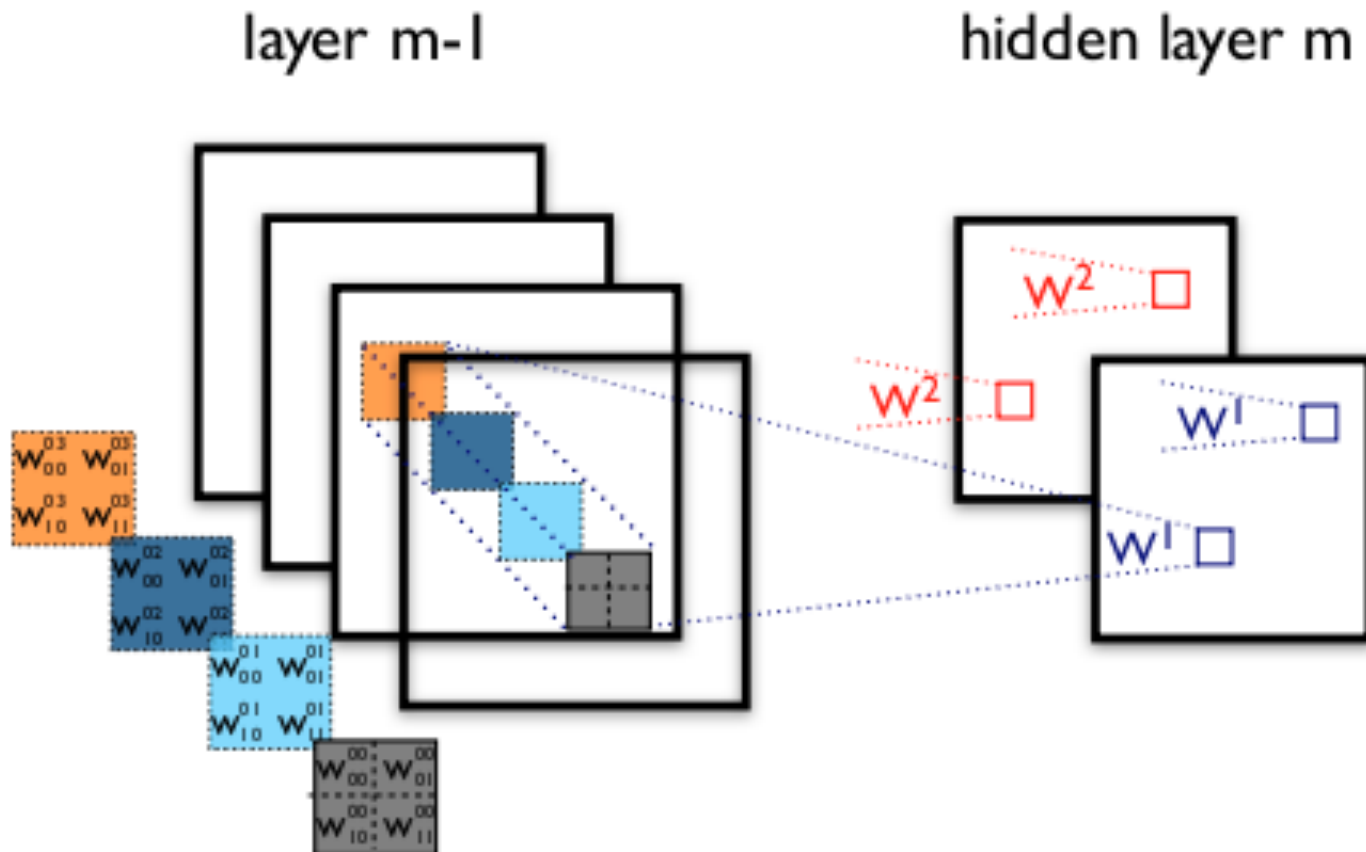
Convolutional Neural Networks (CNN) (LeNet)

Shared Weights



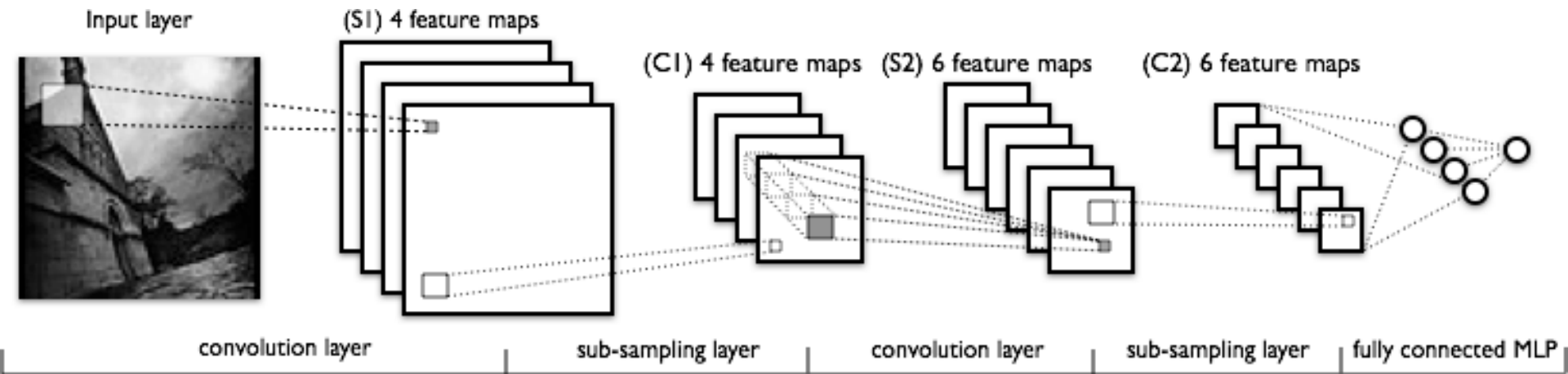
Convolutional Neural Networks (CNN) (LeNet)

example of a convolutional layer



Source: <http://deeplearning.net/tutorial/lenet.html>

Convolutional Neural Networks (CNN) (LeNet)



Source: <http://deeplearning.net/tutorial/lenet.html>

show flights from Boston to New York today

Recurrent Neural Networks with Word Embeddings

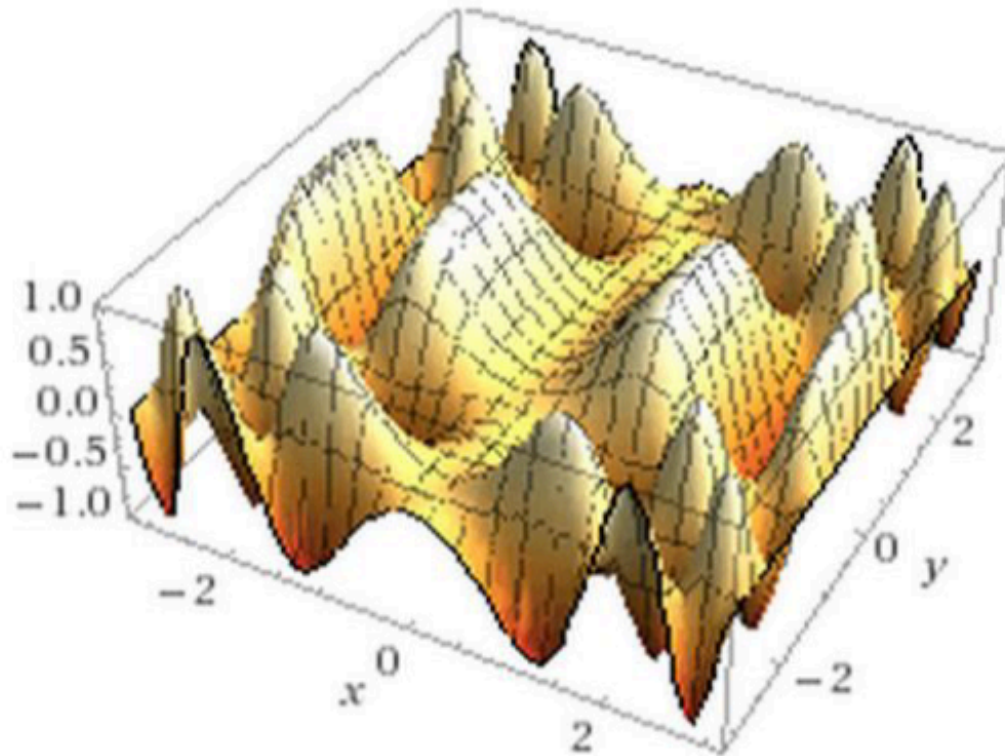
Semantic Parsing / Slot-Filling (Spoken Language Understanding)

Input (words)	show	flights	from	Boston	to	New	York	today
Output (labels)	O	O	O	B-dept	O	B-arr	I-arr	B-date

show flights from Boston to New York today

show flights from Boston to New York today

Input (words)	show	flights	from	Boston	to	New	York	today
Output (labels)	O	O	O	B-dept	O	B-arr	I-arr	B-date



This shows a function of 2 variables: real neural nets are functions of hundreds of millions of variables!

Important Property of Neural Networks

Results get better with

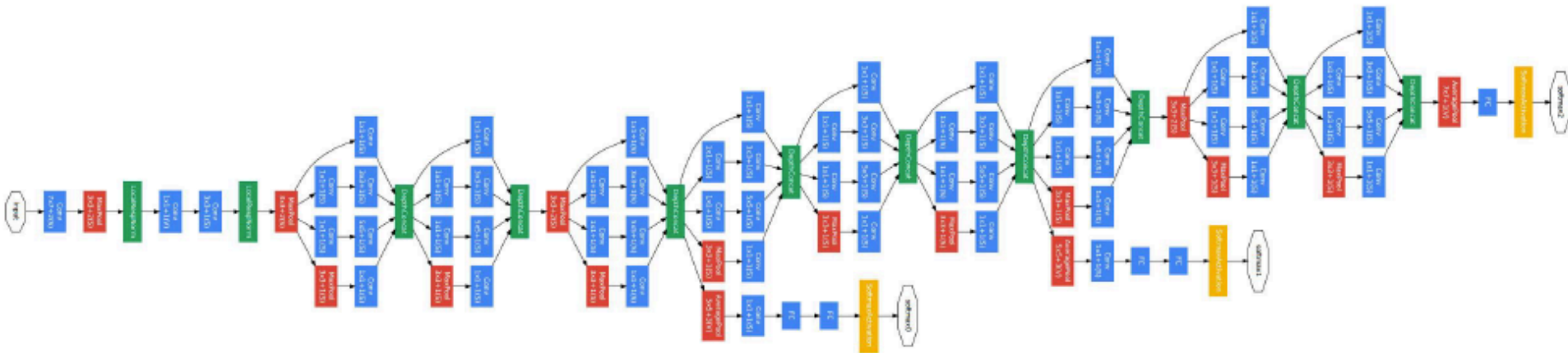
More data +

Bigger models +

More computation

(Better algorithms, new insights
and improved techniques always help, too!)

The Inception Architecture (GoogLeNet, 2014)



Going Deeper with Convolutions

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich

ArXiv 2014, CVPR 2015



DeepDream

GitHub, Inc. [US] <https://github.com/tensorflow/tensorflow/blob/master/tensorflow/examples/tutorials/deepdream/deepdream.ipynb>

```
In [15]: render_deepdream(tf.square(T('mixed4c')), img0)
```



Note that results can differ from the [Caffe's](#) implementation, as we are using an independently trained network. Still, the network seems to like dogs and animal-like features due to the nature of the ImageNet dataset.

Source: <https://github.com/tensorflow/tensorflow/blob/master/tensorflow/examples/tutorials/deepdream/deepdream.ipynb>

Deep Learning Software

- **Theano**
 - CPU/GPU symbolic expression compiler in python (from MILA lab at University of Montreal)
- **Keras**
 - A theano based deep learning library.
- **Tensorflow**
 - TensorFlow™ is an open source software library for numerical computation using data flow graphs.



TensorFlow

Google TensorFlow

TensorFlow™

GET STARTED TUTORIALS HOW TO API RESOURCES ABOUT

Fork me on GitHub

TensorFlow is an Open Source Software Library for Machine Intelligence

GET STARTED

About TensorFlow

TensorFlow™ is 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. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.



<https://www.tensorflow.org/>

TensorFlow
is an
Open Source
Software Library
for
Machine Intelligence

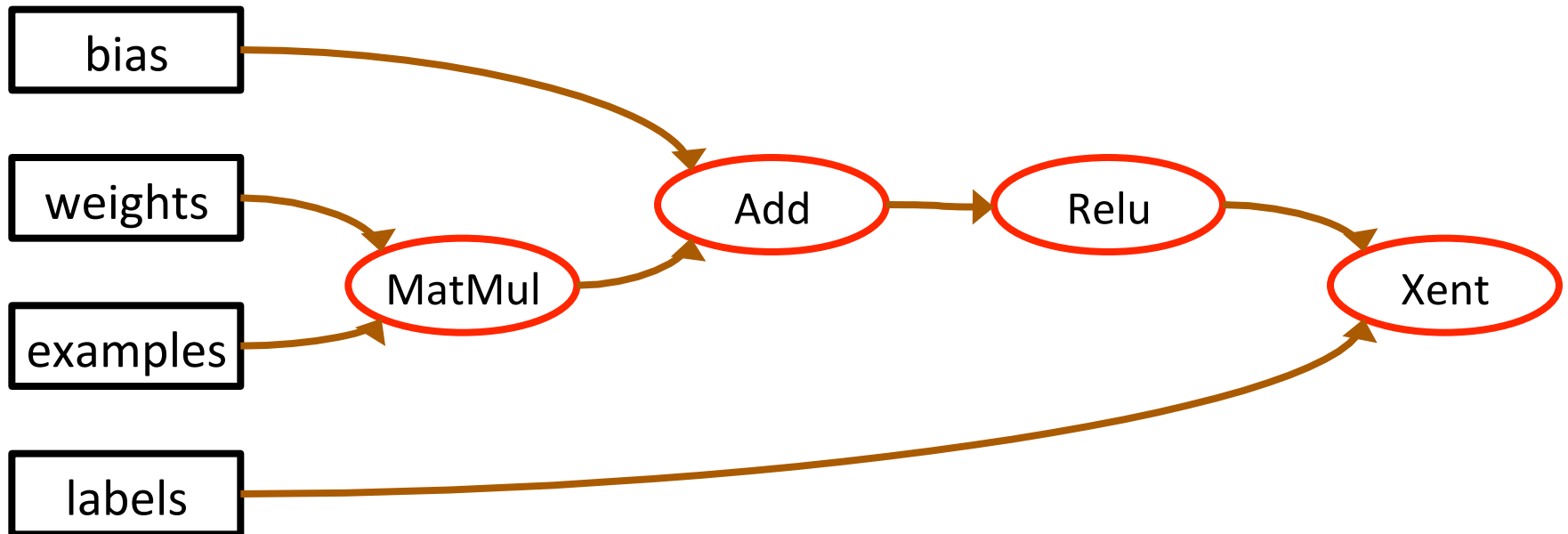
numerical computation using data flow graphs

Nodes:
mathematical operations

edges:
multidimensional data arrays
(tensors)
communicated between nodes

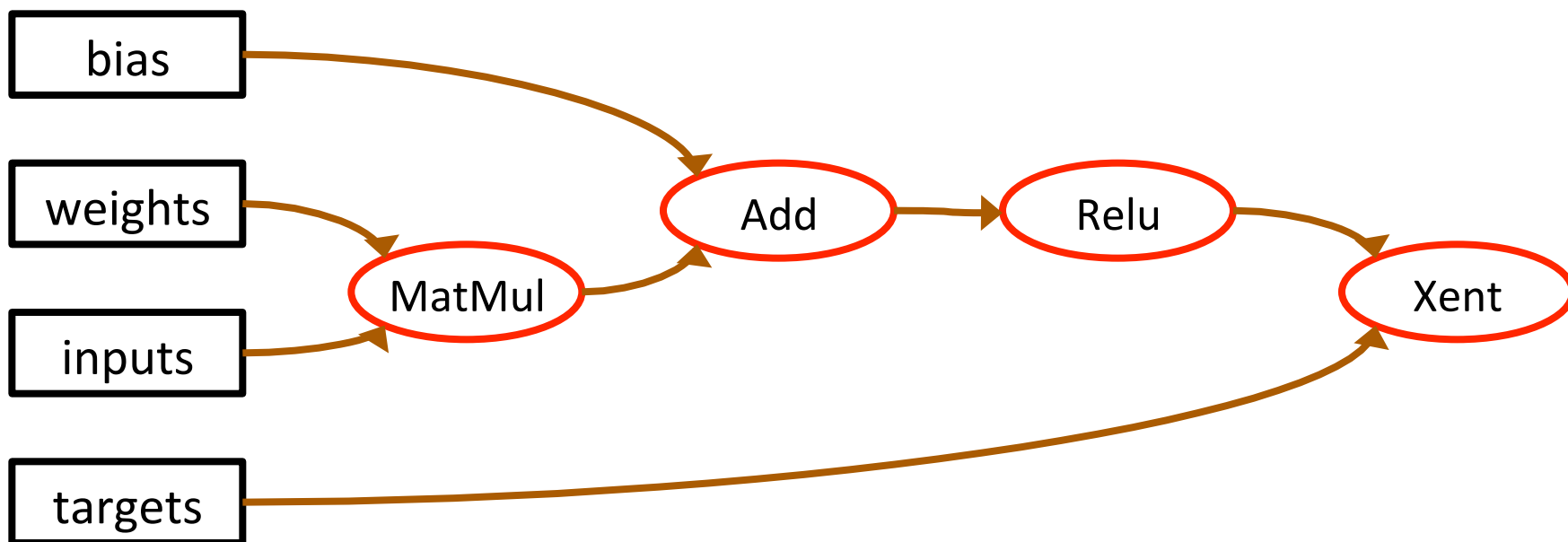
Computation is a Dataflow Graph

Graph of **Nodes**,
also called **Operations** or **ops**.

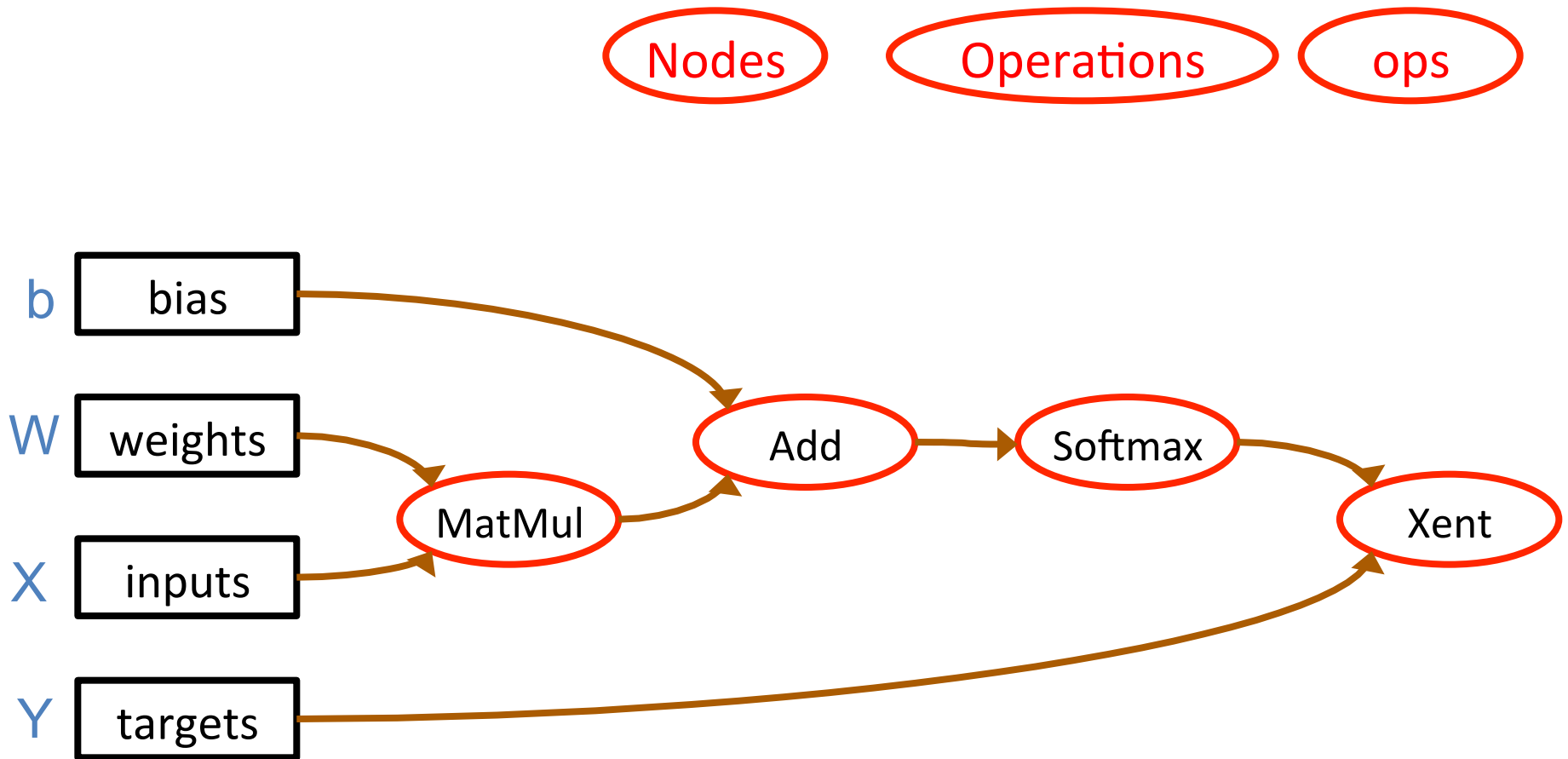


Computation is a Dataflow Graph

Edges are N-dimensional arrays: **Tensors**



Logistic Regression as Dataflow Graph

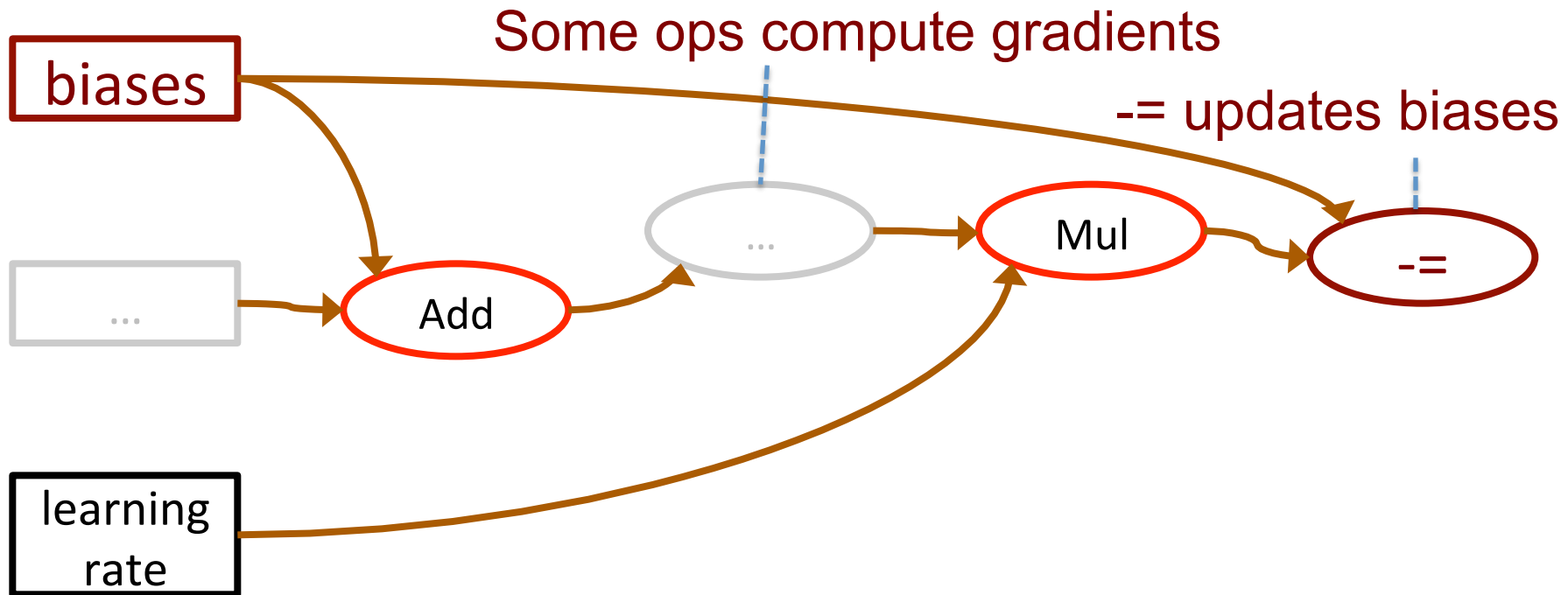


Edges are N-dimensional arrays: **Tensors**

Computation is a Dataflow Graph

with state

'Biases' is a variable

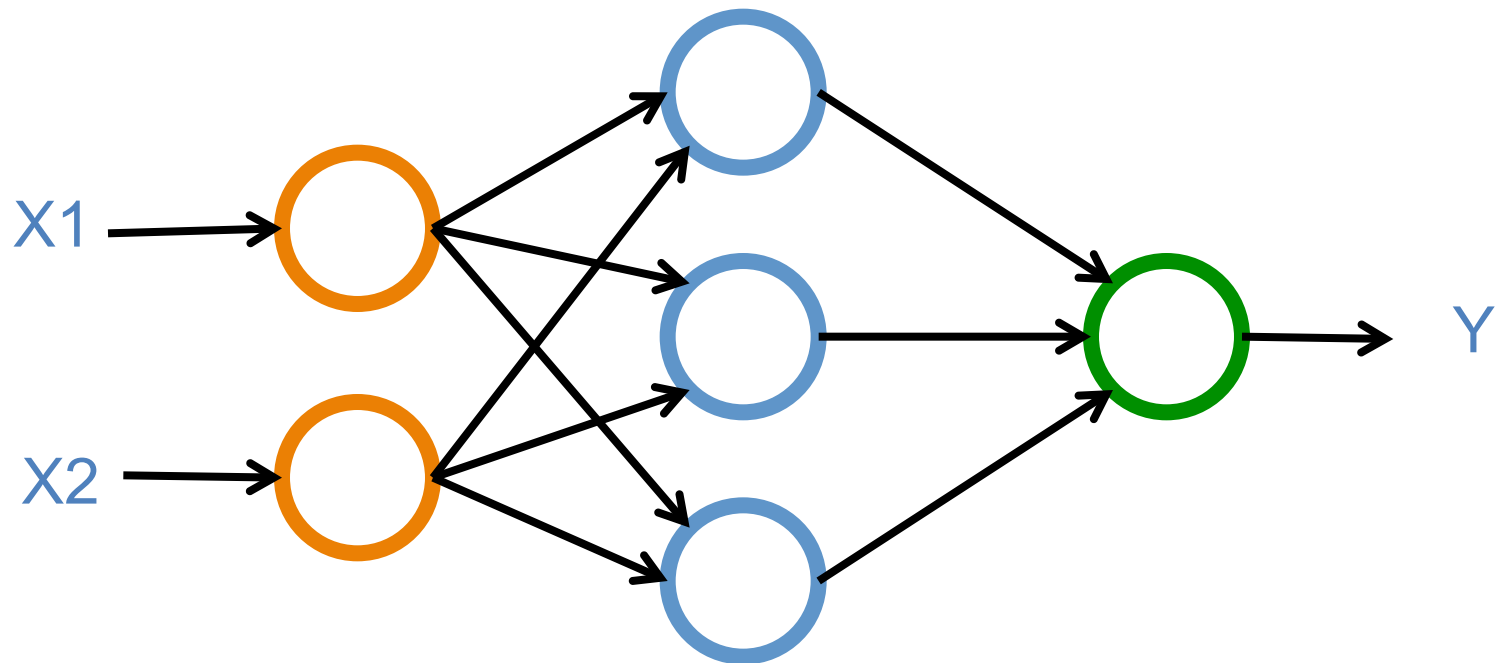


Neural Networks

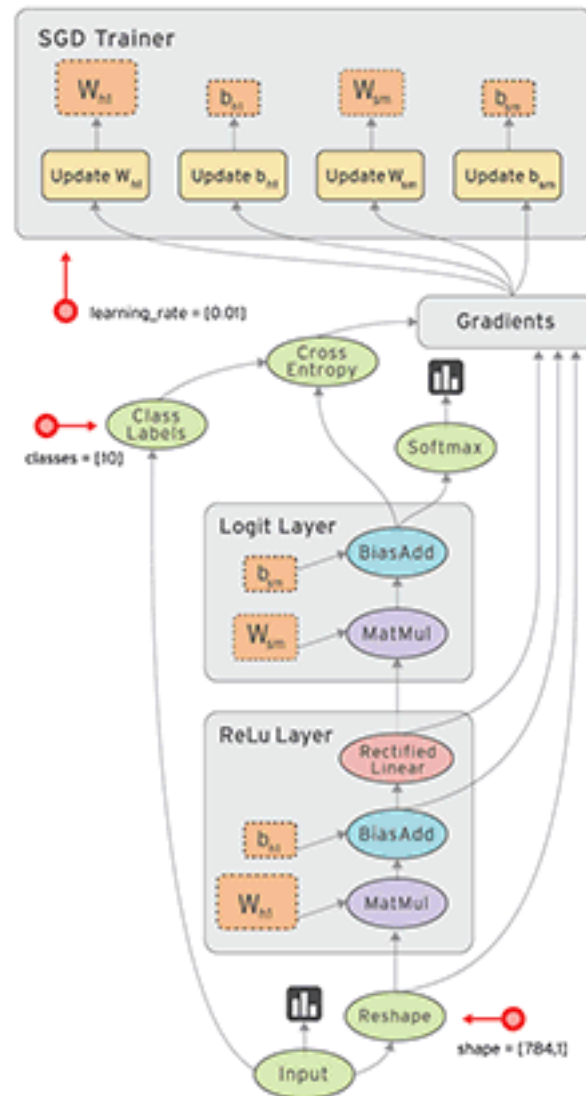
Input Layer
(X)

Hidden Layer
(H)

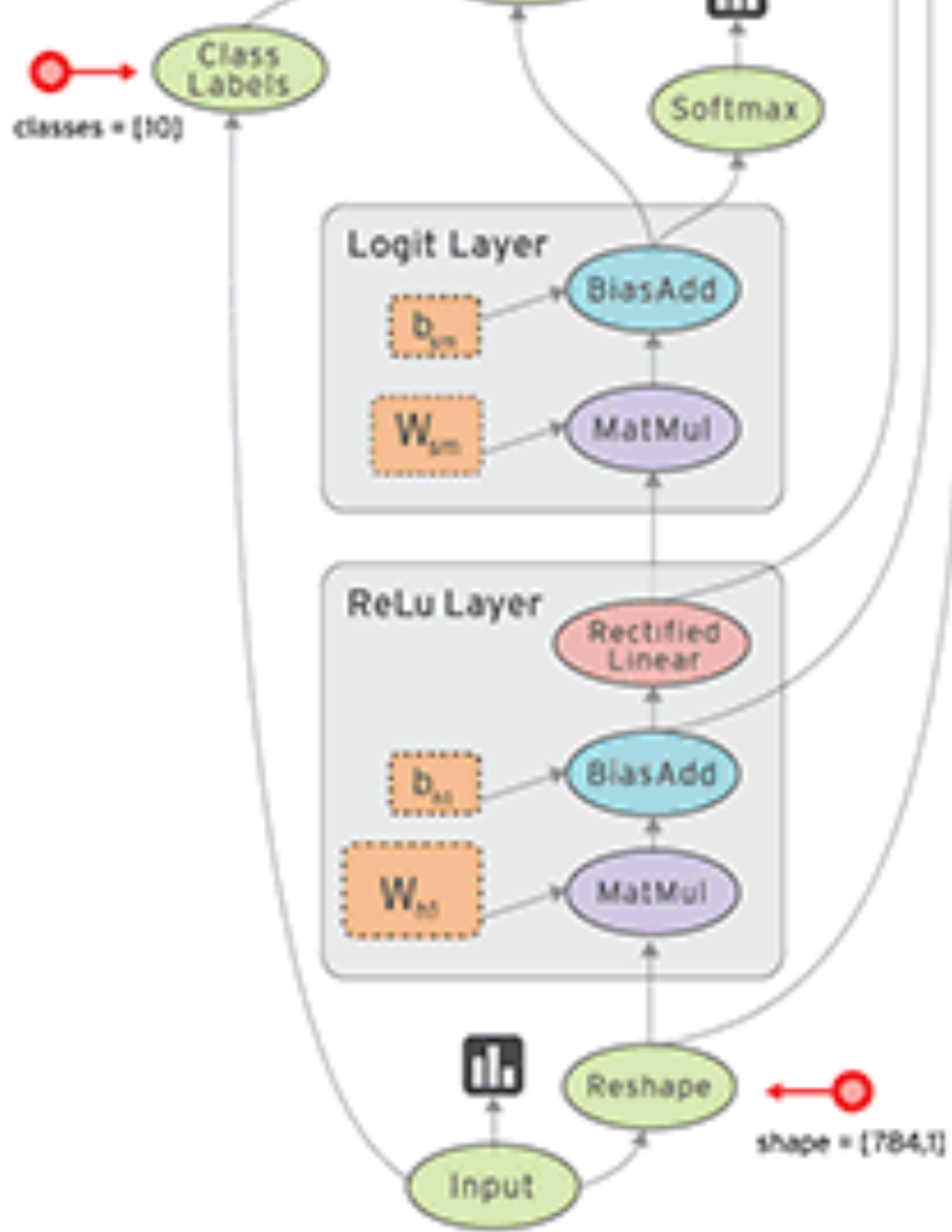
Output Layer
(Y)



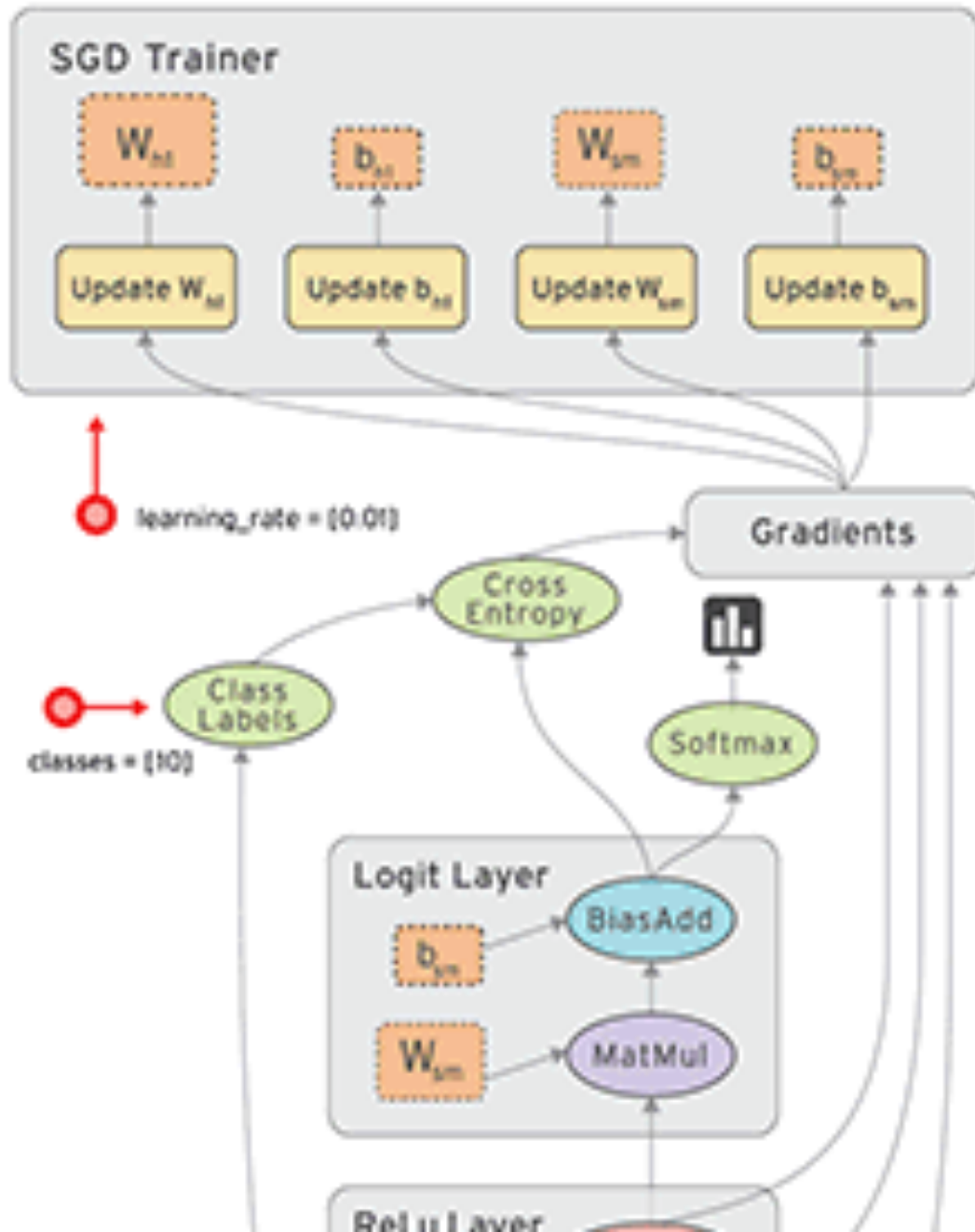
Data Flow Graph



Data Flow Graph



Data Flow Graph



TensorFlow Playground

Tinker With a **Neural Network** Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.



Iterations
000,582

Learning rate
0.03

Activation
Tanh

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%



Noise: 0



Batch size: 10

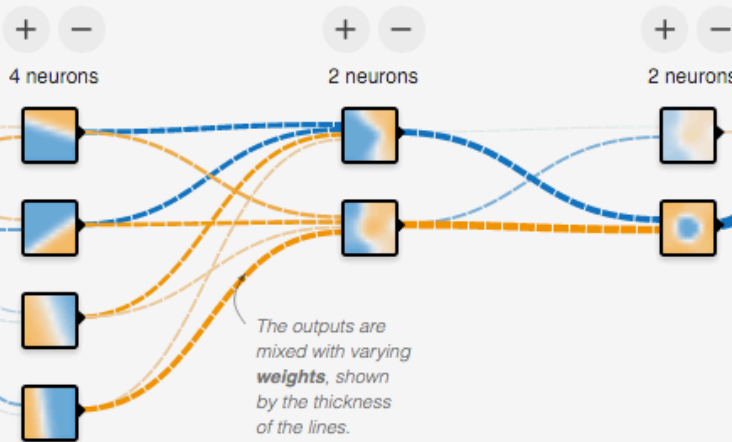


INPUT

Which properties do you want to feed in?



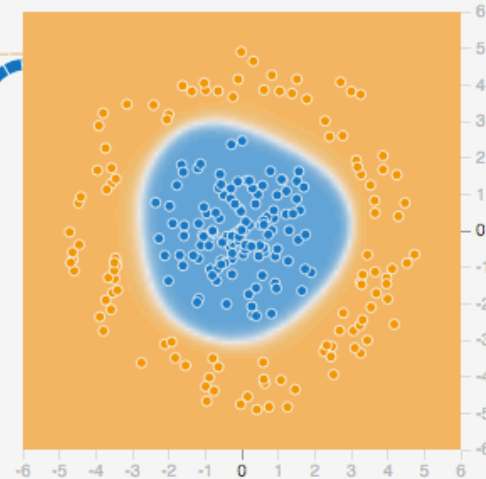
3 HIDDEN LAYERS



This is the output from one **neuron**. Hover to see it larger.

OUTPUT

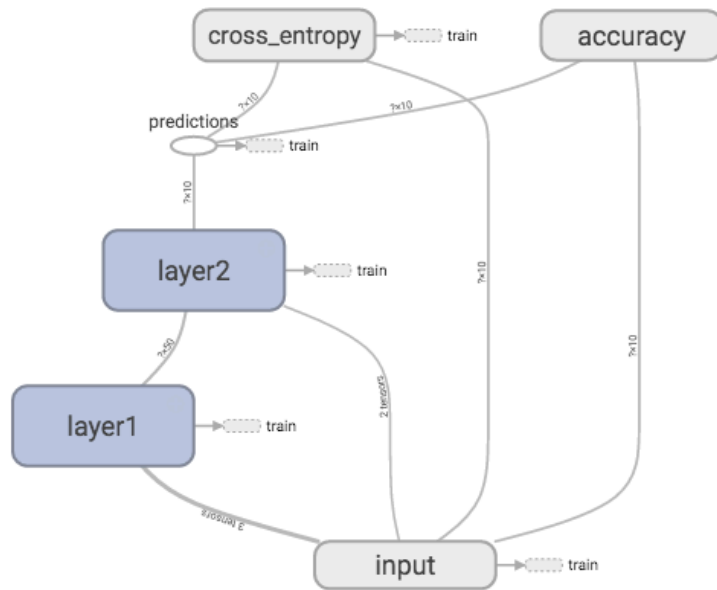
Test loss 0.000
Training loss 0.000



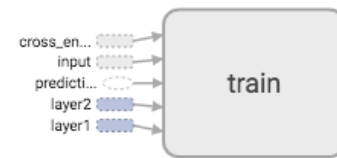
TensorBoard

Fit to screen
 Download PNG
 Run train (1)
 Session runs (0)
 Upload
 Color Structure
 Device
 color: same substructure
 gray: unique substructure
 Graph (* = expandable)
 Namespace*
 OpNode
 Unconnected series*
 Connected series*
 Constant
 Summary
 Dataflow edge
 Control dependency edge
 Reference edge

Main Graph



Auxiliary nodes

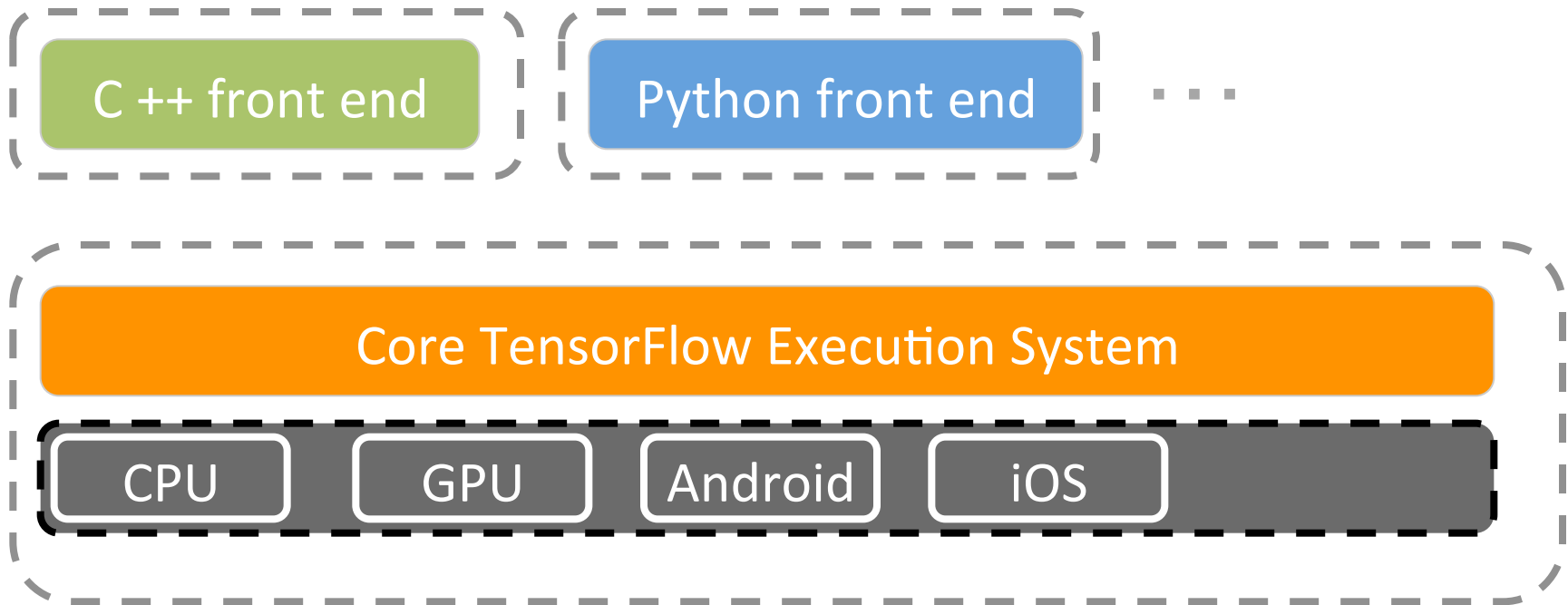


Try your first TensorFlow

```
$ python
```

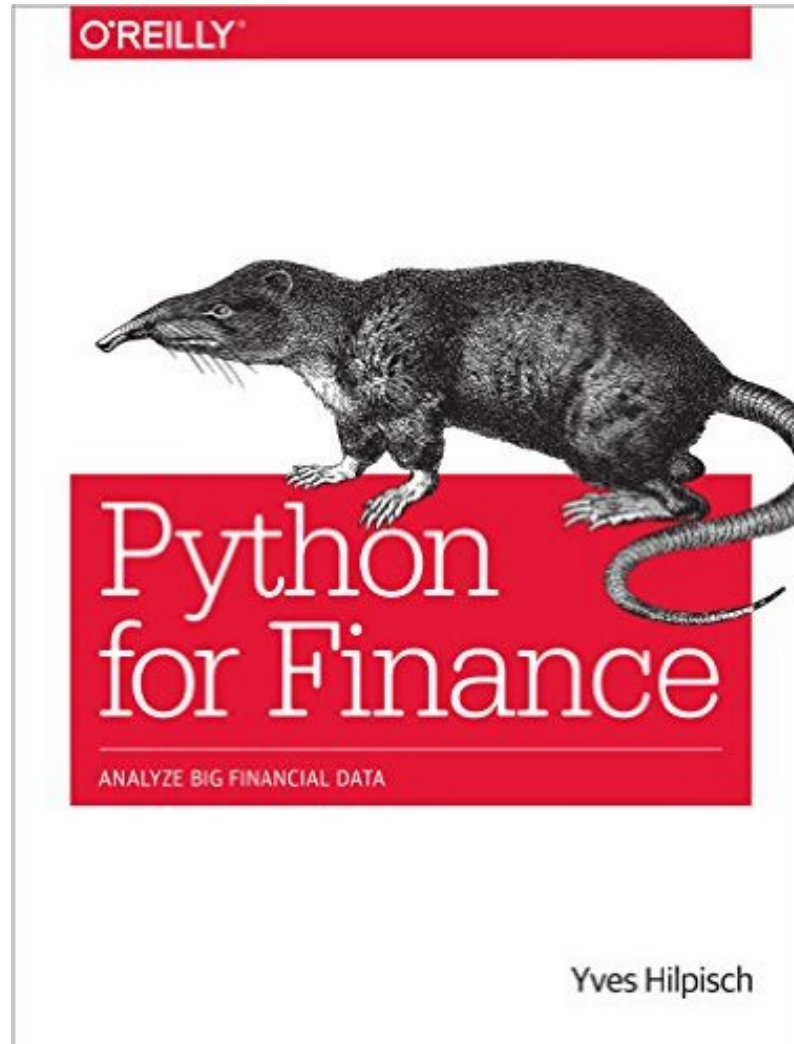
```
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> sess.run(hello)
Hello, TensorFlow!
>>> a = tf.constant(10)
>>> b = tf.constant(32)
>>> sess.run(a+b)
42
>>>
```

Architecture of TensorFlow

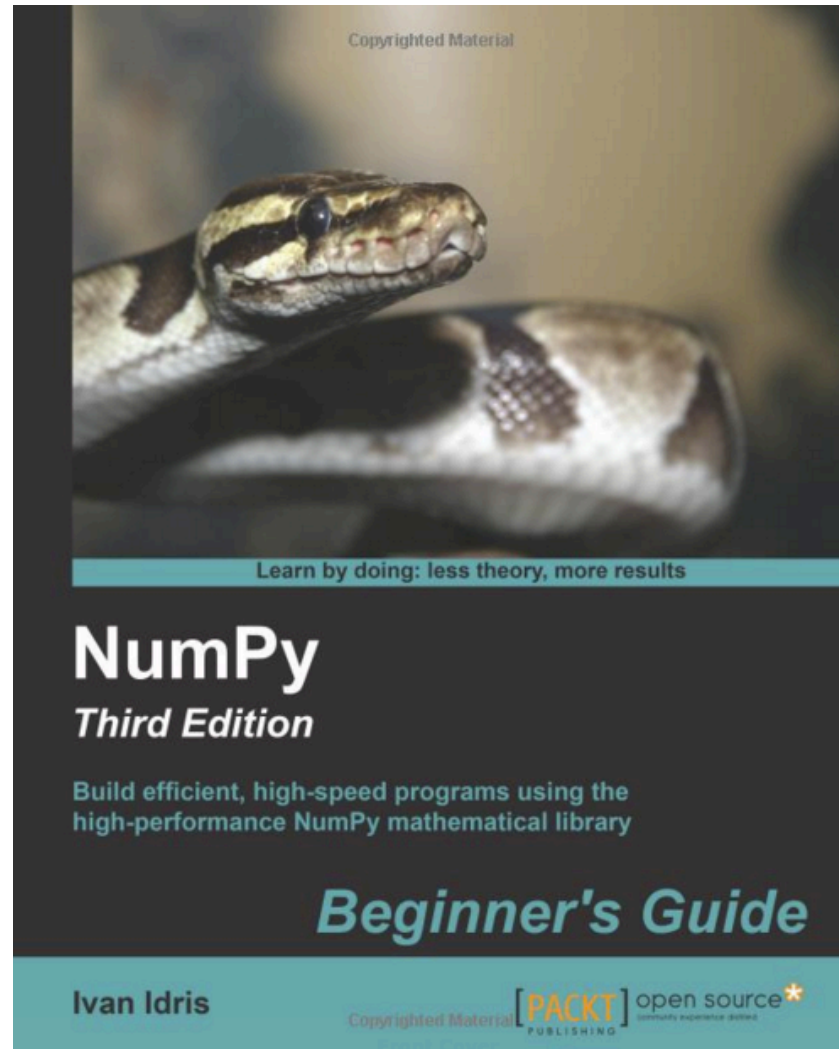




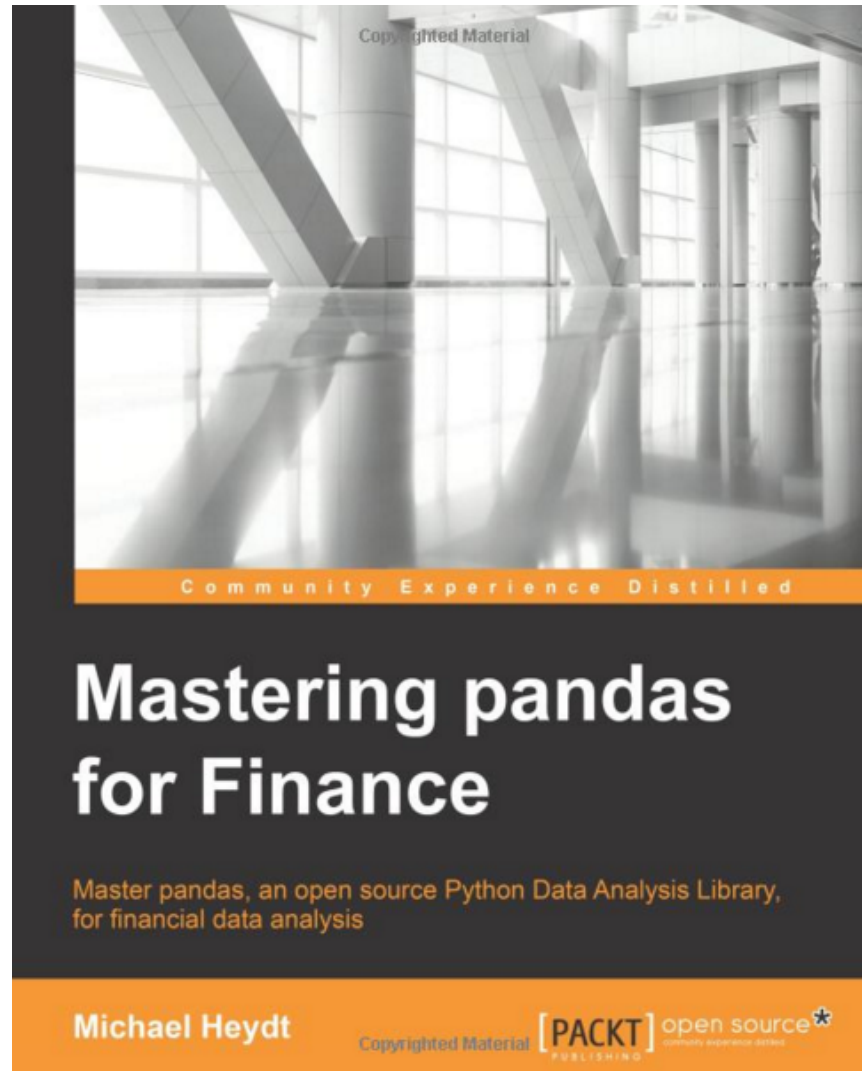
Yves Hilpisch, Python for Finance: Analyze Big Financial Data, O'Reilly, 2014



Ivan Idris, Numpy Beginner's Guide, Third Edition Packt Publishing, 2015

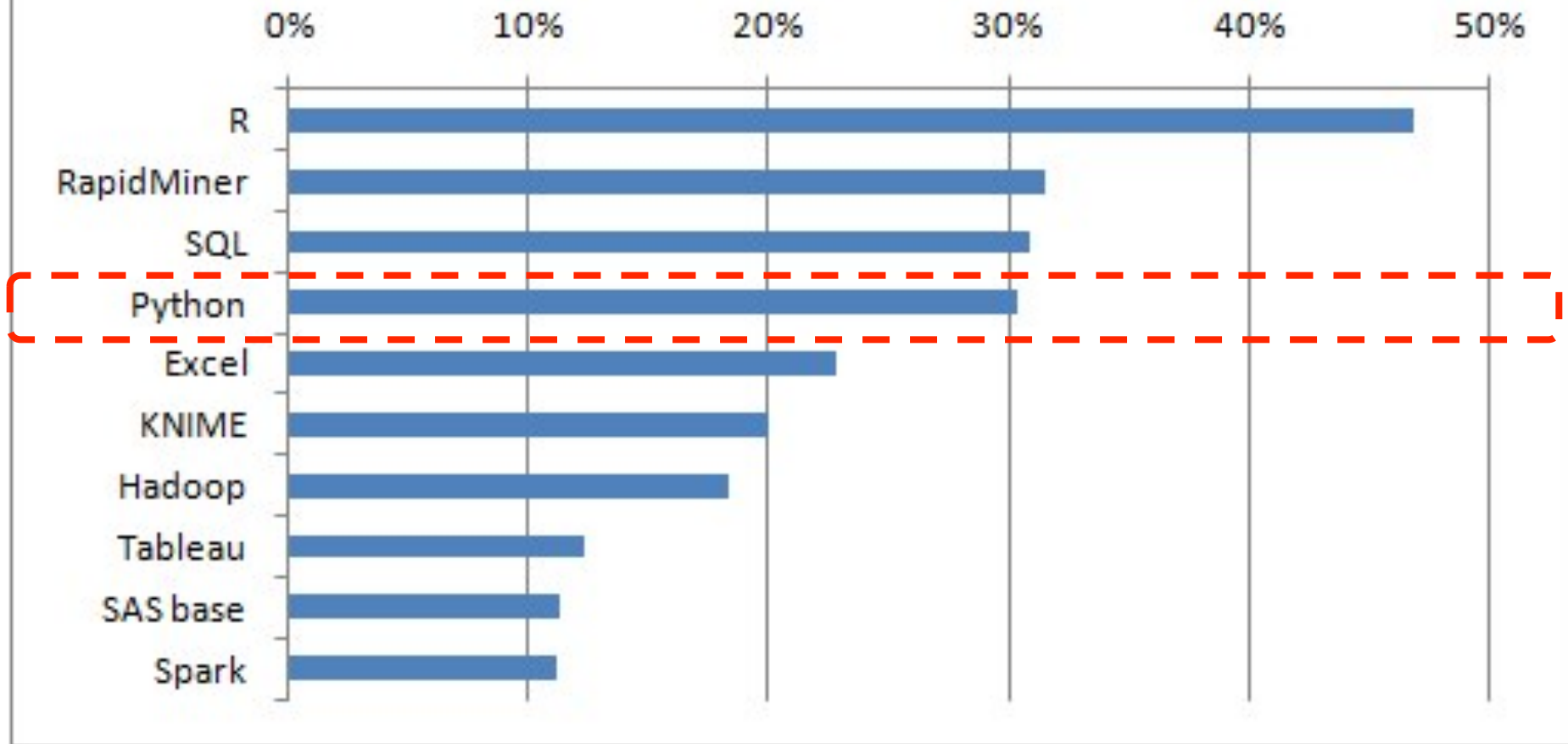


Michael Heydt , Mastering Pandas for Finance, Packt Publishing, 2015



Python: Analytics and Data Science Software

Top Analytics, Data Mining, Data Science software used, 2015



Python

Python

PSF

Docs

PyPI

Jobs

Community



GO

Socialize

Sign In

About

Downloads

Documentation

Community

Success Stories

News

Events

```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in
fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```



Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. [More about lists in Python 3](#)

1

2

3

4

5

Python is a programming language that lets you work quickly and integrate systems more effectively. [>>> Learn More](#)

Get Started

Download

Docs

Jobs

<https://www.python.org/>

Python is an
interpreted,
object-oriented,
high-level
programming language
with
dynamic semantics.

NumPy



Scipy.org

NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

NumPy is licensed under the [BSD license](#), enabling reuse with few restrictions.

Getting Started

- [Getting Numpy](#)
- [Installing the SciPy Stack](#)
- [NumPy and SciPy documentation page](#)
- [NumPy Tutorial](#)
- [NumPy for MATLAB® Users](#)
- [NumPy functions by category](#)
- [NumPy Mailing List](#)

About Numpy

License

Old array packages

NumPy
is the
fundamental package
for
scientific computing
with **Python.**

Python versions (py2 and py3)

- Python 0.9.0 released in 1991 (first release)
- Python 1.0 released in 1994
- Python 2.0 released in 2000
- Python 2.6 released in 2008
- **Python 2.7 released in 2010**
- Python 3.0 released in 2008
- Python 3.3 released in 2010
- Python 3.4 released in 2014
- Python 3.5 released in 2015

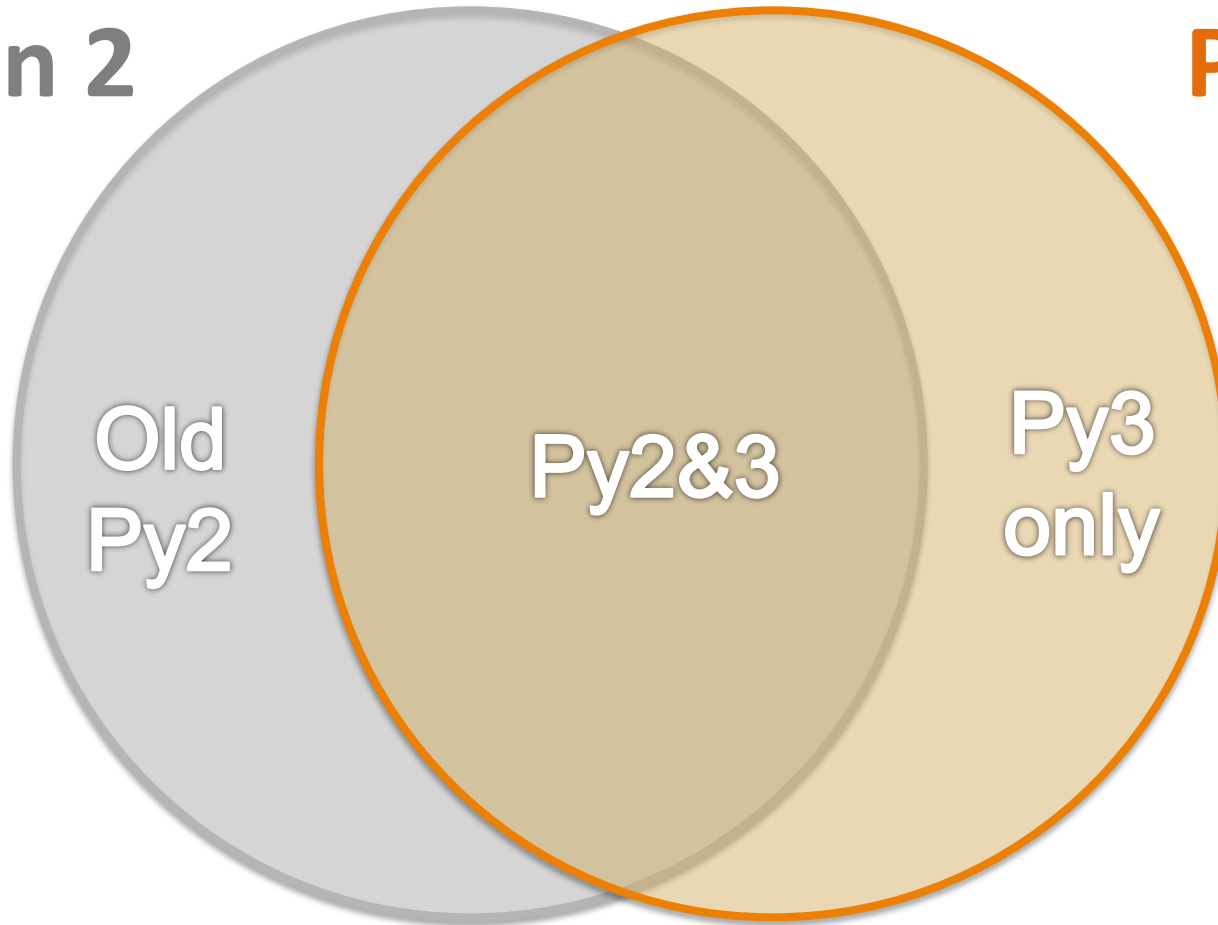
Python (Python 2.7 & Python 3.5)



Standard Syntax

Python 2

Python 3



Source: PyCon Australia (2014), Writing Python 2/3 compatible code by Edward Schofield

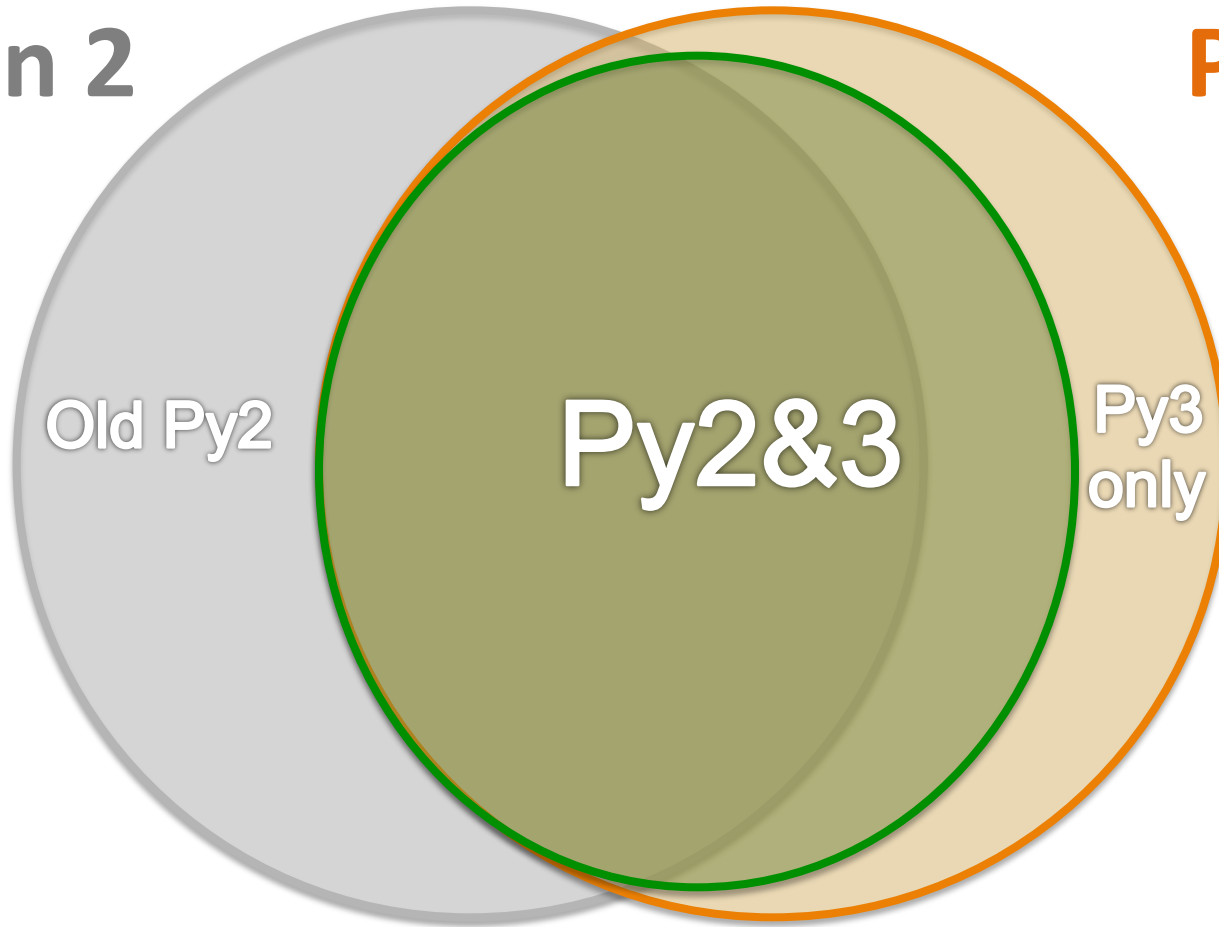
<https://www.youtube.com/watch?v=KOqk8j11aAI>

```
from __future__ import ...
```



Python 2

Python 3



Source: PyCon Australia (2014), Writing Python 2/3 compatible code by Edward Schofield

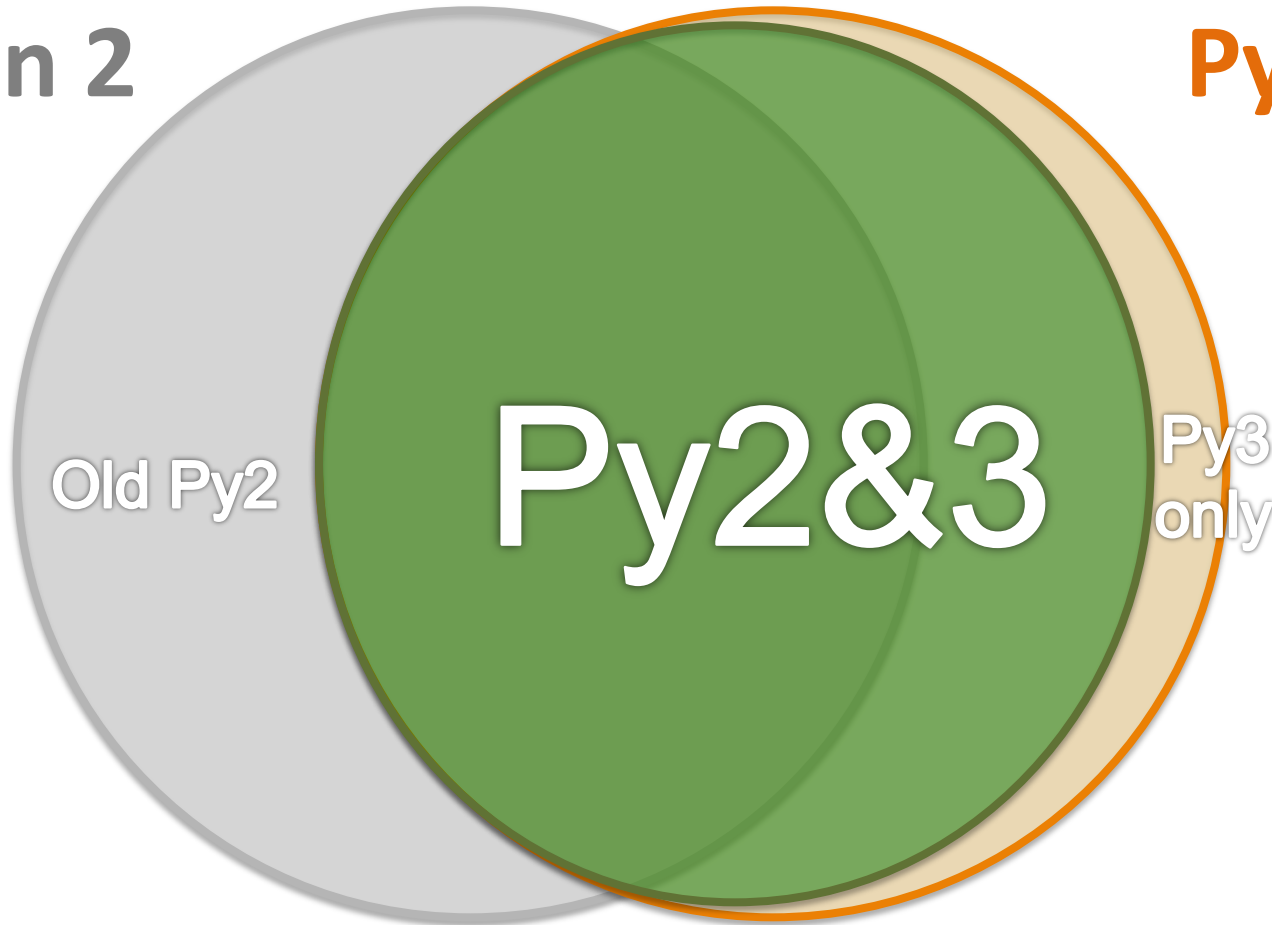
<https://www.youtube.com/watch?v=KOqk8j11aAI>

```
from future.builtins import *
```



Python 2

Python 3



Source: PyCon Australia (2014), Writing Python 2/3 compatible code by Edward Schofield

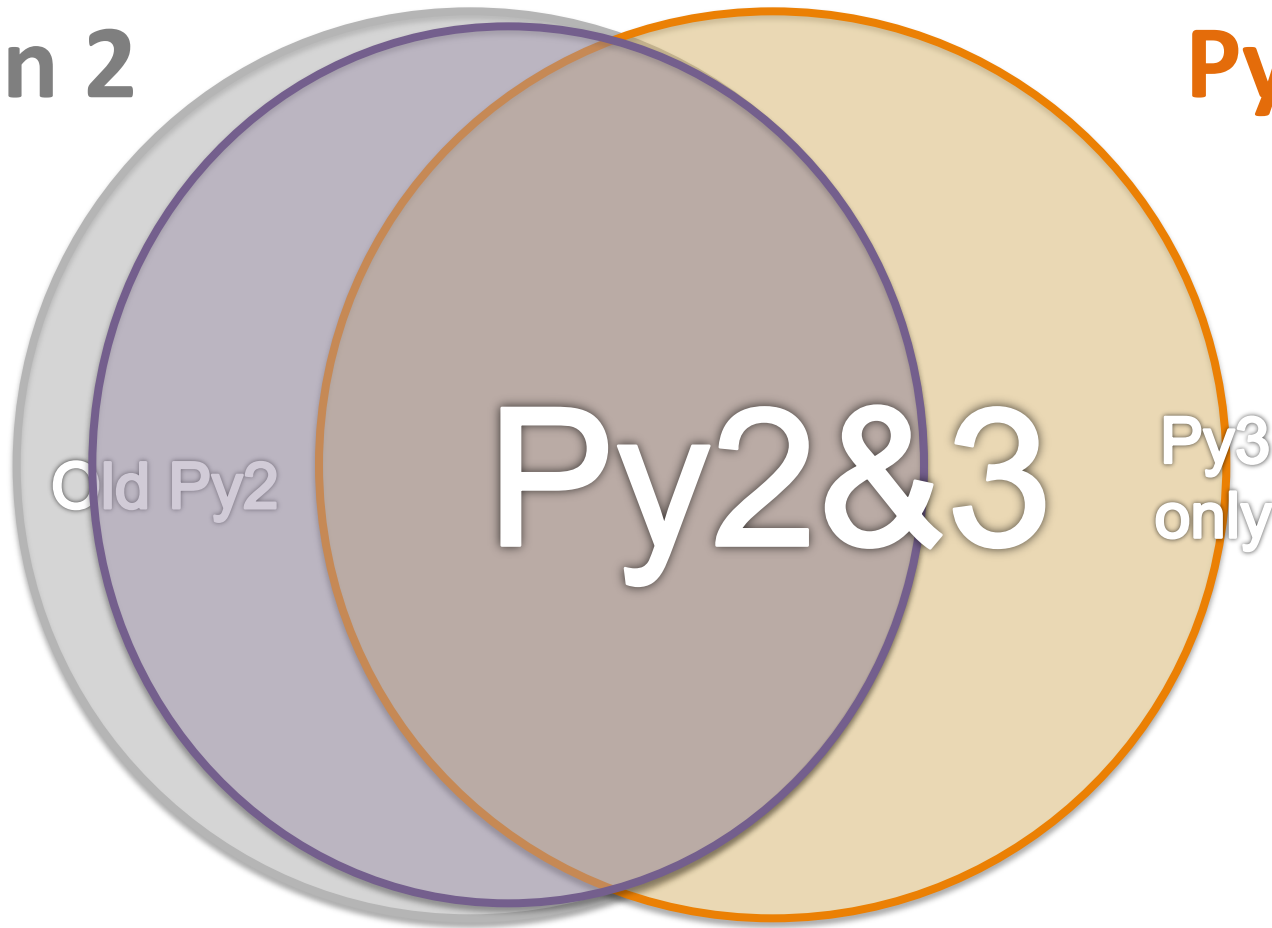
<https://www.youtube.com/watch?v=KOqk8j11aAI>

```
from past.builtins import *
```



Python 2

Python 3



Source: PyCon Australia (2014), Writing Python 2/3 compatible code by Edward Schofield

<https://www.youtube.com/watch?v=KOqk8j11aAI>

Anaconda

CONTINUUM[®]
ANALYTICS

ANACONDA

COMMUNITY

SERVICES

SOLUTIONS

ABOUT

RESOURCES

LOG IN SUPPORT CONTACT

ANACONDA GIVES
SUPERPOWERS TO
PEOPLE WHO CHANGE
THE WORLD



ANACONDA[®]

Modern open source analytics platform powered
by Python

DOWNLOAD FOR FREE

ANACONDA NOW AVAILABLE FOR CLOUDERA CDH

WHY YOU'LL LOVE ANACONDA

Making it easy to install, intuitive to discover, quick to analyze, simple to collaborate, and accessible to all.

Committed to Open Source. Now and forever.

Tested and certified packages to cover your back.

Explore and visualize complex data easily.

All the analytics you ever wanted and more.

<https://www.continuum.io/>

Download Anaconda

🏠 ANACONDA ▶ DOWNLOAD

DOWNLOAD ANACONDA NOW!

Jump to: [Windows](#) | [OS X](#) | [Linux](#)

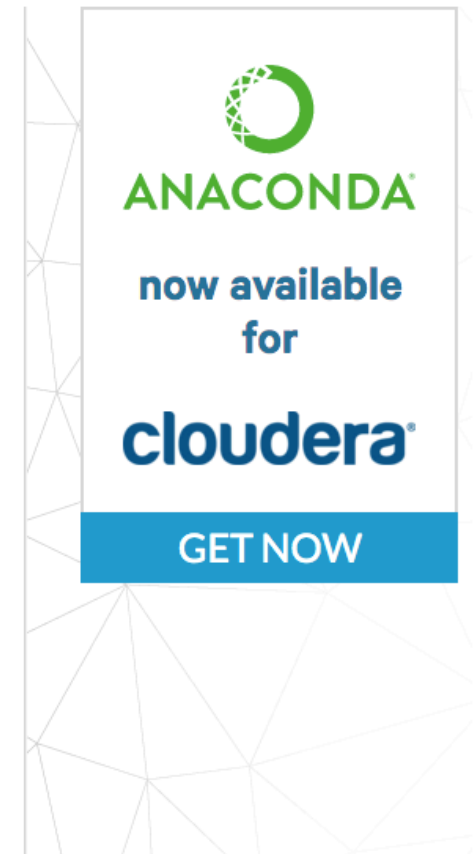
Get Superpowers with Anaconda

Anaconda is a completely free Python distribution (including for commercial use and redistribution). It includes more than 400 of the most popular Python packages for science, math, engineering, and data analysis. See the packages included with Anaconda and [the Anaconda changelog](#).

Which version should I download and install?

Because Anaconda includes installers for Python 2.7 and 3.5, either is fine. Using either version, you can use Python 3.4 with the conda command. You can create a 3.5 environment with the conda command if you've downloaded 2.7 — and vice versa.

If you don't have time or disk space for the entire distribution, try [Miniconda](#), which contains only conda and Python. Then install just the individual packages you want through the conda command.



Download Anaconda Python 2.7

Anaconda for OS X

PYTHON 2.7	PYTHON 3.5
Mac OS X 64-bit Graphical Installer 274M (OS X 10.7 or higher)	Mac OS X 64-bit Graphical Installer 267M (OS X 10.7 or higher)
Mac OS X 64-bit Command-Line installer 239M (OS X 10.7 or higher)	Mac OS X 64-bit Command-Line installer 233M (OS X 10.7 or higher)

OS X Anaconda Installation

Choose either the graphical installer or the command line installer for OS X.

Graphical Installer:

1. Download the graphical installer.
2. Double-click the downloaded .pkg file and follow the instructions.

<https://www.continuum.io/downloads>

OS X Anaconda Installation

OS X Anaconda Installation

Choose either the graphical installer or the command line installer for OS X.

Graphical Installer:

1. Download the graphical installer.
2. Double-click the downloaded .pkg file and follow the instructions.

Command Line Installer:

1. Download the command line installer.
2. In your terminal window, type one of the below and follow the instructions:

Python 2.7:

```
bash Anaconda2-2.5.0-MacOSX-x86_64.sh
```

Python 3.5:

```
bash Anaconda3-2.5.0-MacOSX-x86_64.sh
```

NOTE: Include the "bash" command even if you are not using the bash shell.

3. Optional: [Verify data integrity with MD5.](#)

<https://www.continuum.io/downloads>

OS X Anaconda Installation

Anaconda2-2.5.0-MacOSX-x86_64.pkg



Anaconda2-2.5.0-
MacOSX-x86_64.pkg

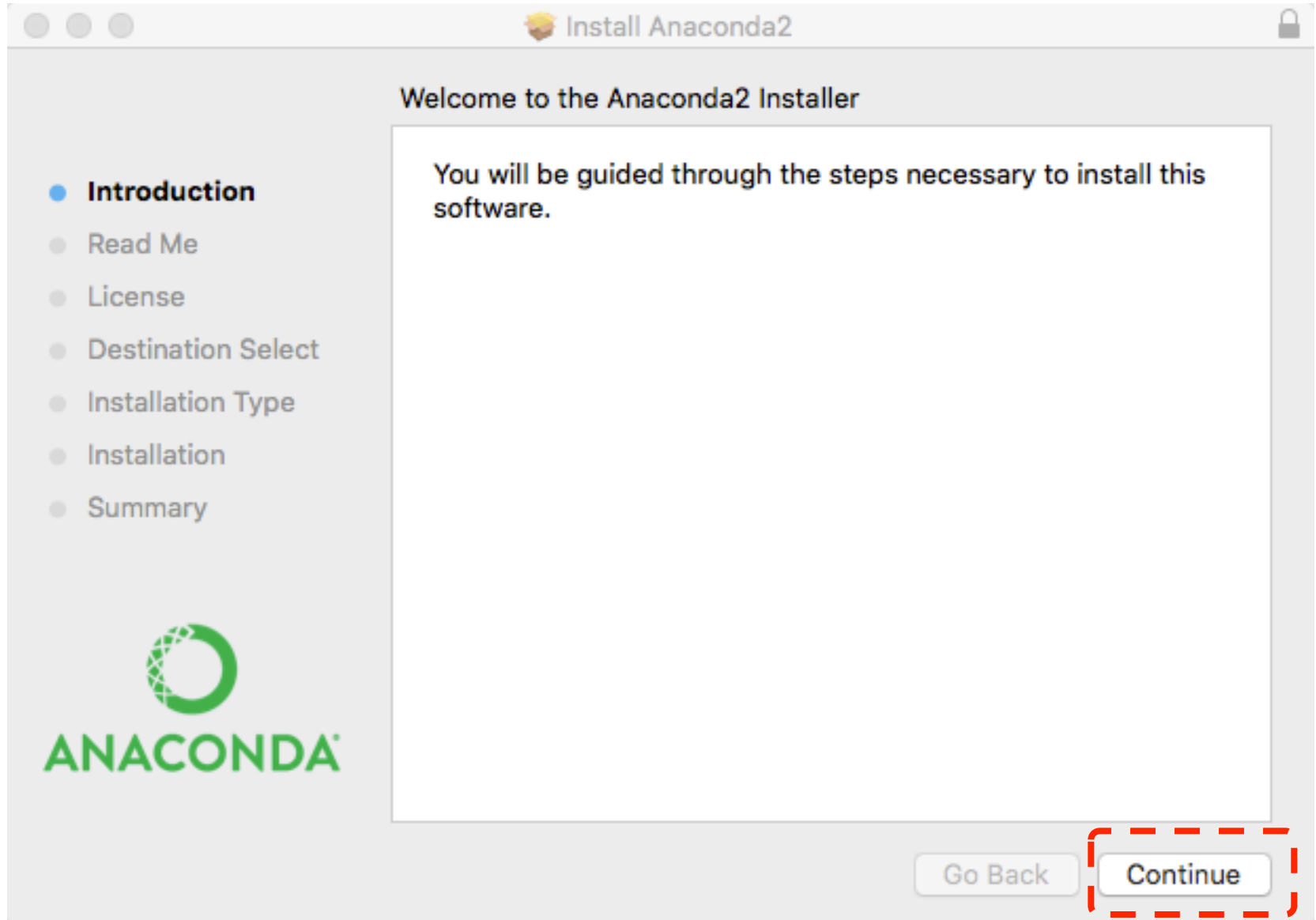
385.8 MB



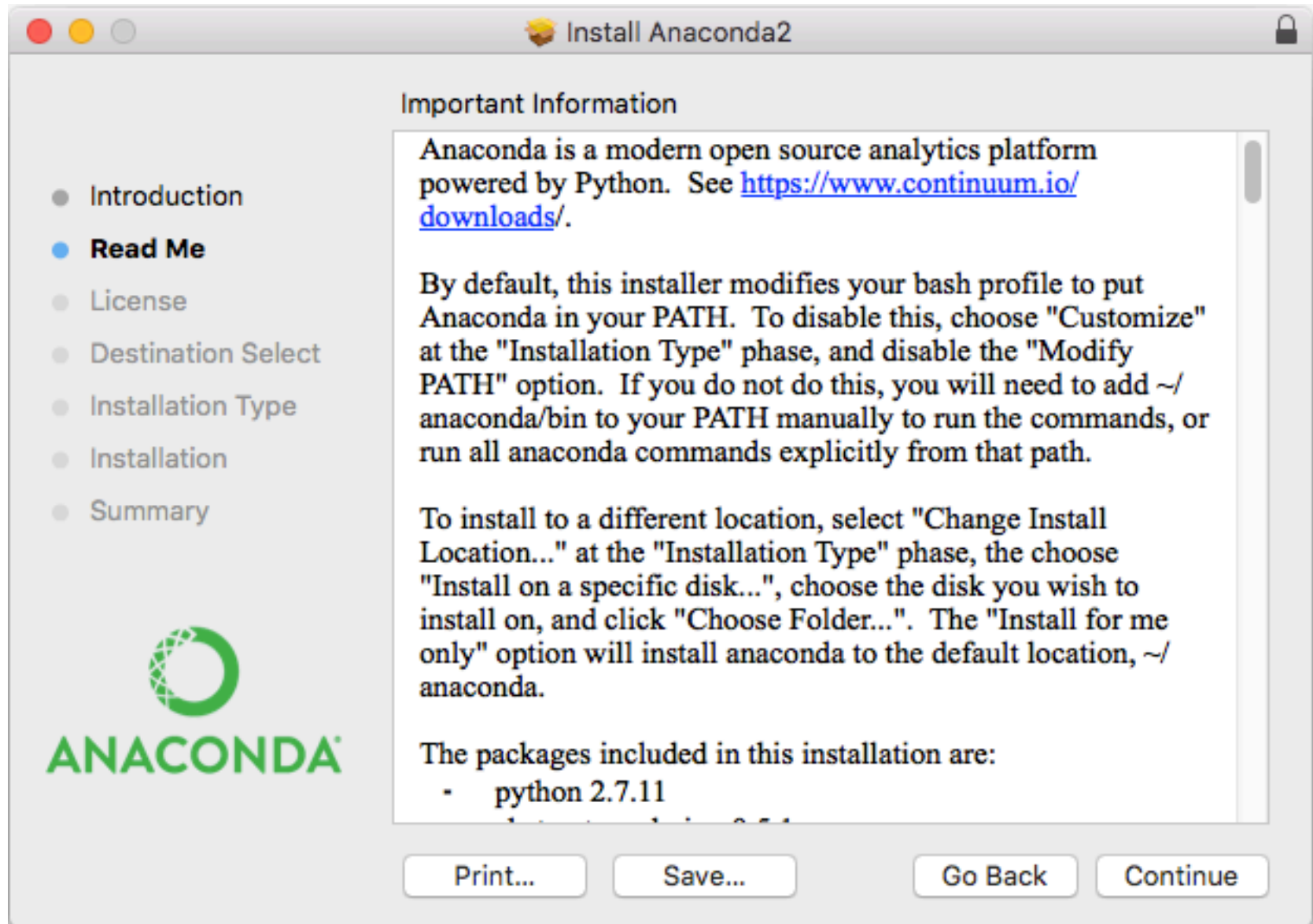
Anaconda2-2.5.0-MacOSX-x86_64.pkg

Installer package - 385.8 MB

OS X Anaconda Installation



OS X Anaconda Installation



OS X Anaconda Installation

Install Anaconda2

Important Information

The packages included in this installation are:

- python 2.7.11
- abstract-rendering 0.5.1
- alabaster 0.7.7
- anaconda-client 1.2.2
- appnope 0.1.0
- appscript 1.0.1
- argcomplete 1.0.0
- astropy 1.1.1
- babel 2.2.0
- backports_abc 0.4
- beautifulsoup4 4.4.1
- bitarray 0.8.1
- blaze-core 0.9.0
- bokeh 0.11.0
- boto 2.39.0
- bottleneck 1.0.0
- cdecimal 2.3

Print... Save... Go Back Continue

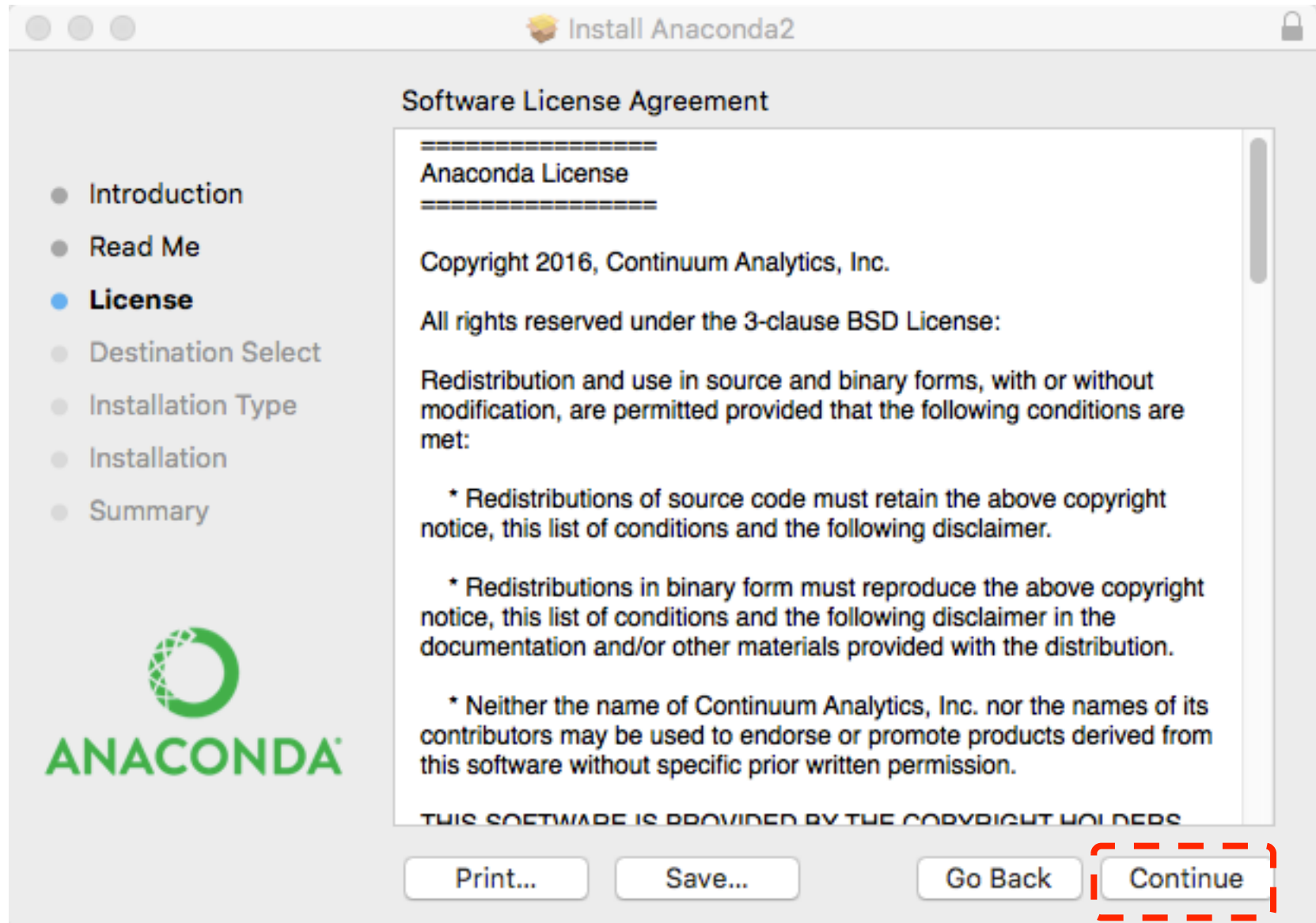
ANACONDA

Install Anaconda 2:

165 packages included

1	-	python 2.7.11	59	-	jupyter 1.0.0	128	-	scikit-learn 0.17
2	-	abstract-rendering 0.5.1	60	-	jupyter_client 4.1.1	129	-	scipy 0.17.0
3	-	alabaster 0.7.7	61	-	jupyter_console 4.1.0	130	-	setuptools 19.6.2
4	-	anaconda-client 1.2.2	62	-	jupyter_core 4.0.6	131	-	simplegeneric 0.8.1
5	-	appnope 0.1.0	63	-	launcher 1.0.0	132	-	singledispatch 3.4.0.3
6	-	appsript 1.0.1	64	-	libdynd 0.7.1	133	-	sip 4.16.9
7	-	argcomplete 1.0.0	65	-	libpng 1.6.17	134	-	six 1.10.0
8	-	astropy 1.1.1	66	-	libtiff 4.0.6	135	-	snowballstemmer 1.2.1
9	-	babel 2.2.0	67	-	libxml2 2.9.2	136	-	sockjs-tornado 1.0.1
10	-	backports_abc 0.4	68	-	libxslt 1.1.28	137	-	sphinx 1.3.5
11	-	beautifulsoup4 4.4.1	69	-	llvmlite 0.8.0	138	-	sphinx_rtd_theme 0.1.9
12	-	bitarray 0.8.1	70	-	lxml 3.5.0	139	-	spyder 2.3.8
13	-	blaze-core 0.9.0	71	-	markupsafe 0.23	140	-	spyder-app 2.3.8
14	-	bokeh 0.11.0	72	-	matplotlib 1.5.1	141	-	sqlalchemy 1.0.11
15	-	boto 2.39.0	73	-	mistune 0.7.1	142	-	sqlite 3.9.2
16	-	bottleneck 1.0.0	74	-	mkl 11.3.1	143	-	ssl_match_hostname 3.4.0.2
17	-	cdecimal 2.3	75	-	mkl-service 1.1.2	144	-	statsmodels 0.6.1
18	-	cffi 1.2.1	76	-	multipledispatch 0.4.8	145	-	sympy 0.7.6.1
19	-	clyent 1.2.0	77	-	nbconvert 4.1.0	146	-	terminado 0.5
20	-	colorama 0.3.6	78	-	nbformat 4.0.1	147	-	tk 8.5.18
21	-	configobj 5.0.6	79	-	networkx 1.11	148	-	toolz 0.7.4
22	-	cryptography 1.0.2	80	-	nlTK 3.1	149	-	tornado 4.3
23	-	curl 7.45.0	81	-	node-webkit 0.10.1	150	-	traitlets 4.1.0
24	-	cycler 0.9.0	82	-	nose 1.3.7	151	-	unicodcsv 0.14.1
25	-	cython 0.23.4	83	-	notebook 4.1.0	152	-	werkzeug 0.11.3
26	-	cytoolz 0.7.5	84	-	numba 0.23.1	153	-	wheel 0.26.0
27	-	datashape 0.5.0	85	-	numexpr 2.4.6	154	-	xlrd 0.9.4
28	-	decorator 4.0.6	86	-	numpy 1.10.4	155	-	xlsxwriter 0.8.4
29	-	docutils 0.12	87	-	odo 0.4.0	156	-	xlwings 0.6.4
30	-	dynd-python 0.7.1	88	-	openpyxl 2.3.2	157	-	xlwt 1.0.0
			89	-	openssl 1.0.2f	158	-	xz 5.0.5
			90	-	pandas 0.17.1	159	-	yaml 0.1.6
			91	-	path.py 8.1.2	160	-	zeromq 4.1.3
			92	-	patsy 0.4.0	161	-	zlib 1.2.8
			93	-	pep8 1.7.0	162	-	anaconda 2.5.0
			94	-	pexpect 3.3	163	-	conda 3.19.1
			95	-	pickleshare 0.5	164	-	conda-build 1.19.0
						165	-	conda-env 2.4.5

OS X Anaconda Installation



The screenshot shows the 'Install Anaconda2' window on OS X. The window title is 'Install Anaconda2' and it has a lock icon in the top right corner. On the left side, there is a navigation menu with the following items: Introduction, Read Me, License (highlighted with a blue dot), Destination Select, Installation Type, Installation, and Summary. Below the menu is the Anaconda logo, which consists of a green circle with a white grid pattern and the word 'ANACONDA' in green capital letters below it.

The main content area is titled 'Software License Agreement'. It contains the following text:

```
=====
Anaconda License
=====
```

Copyright 2016, Continuum Analytics, Inc.

All rights reserved under the 3-clause BSD License:

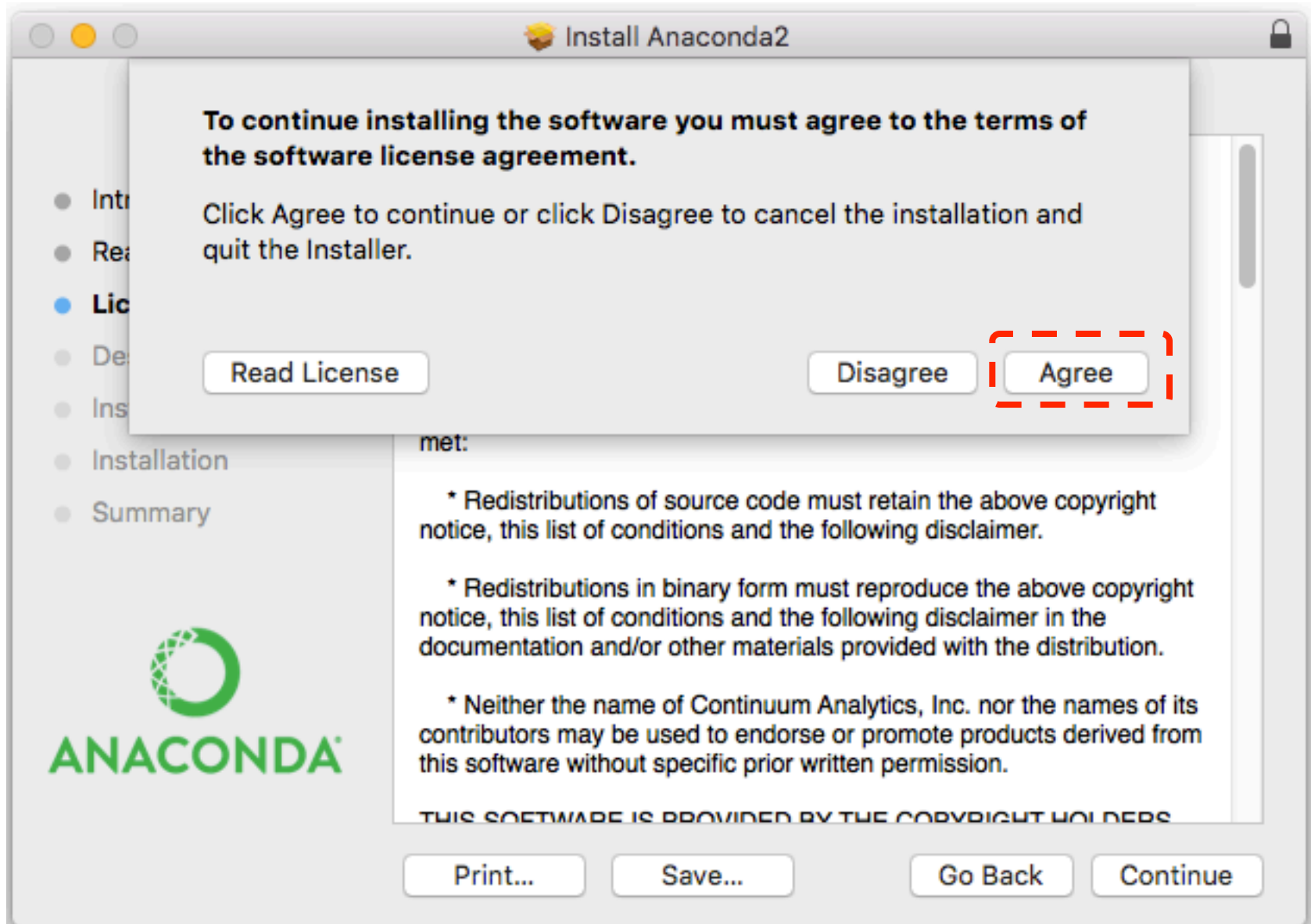
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Continuum Analytics, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

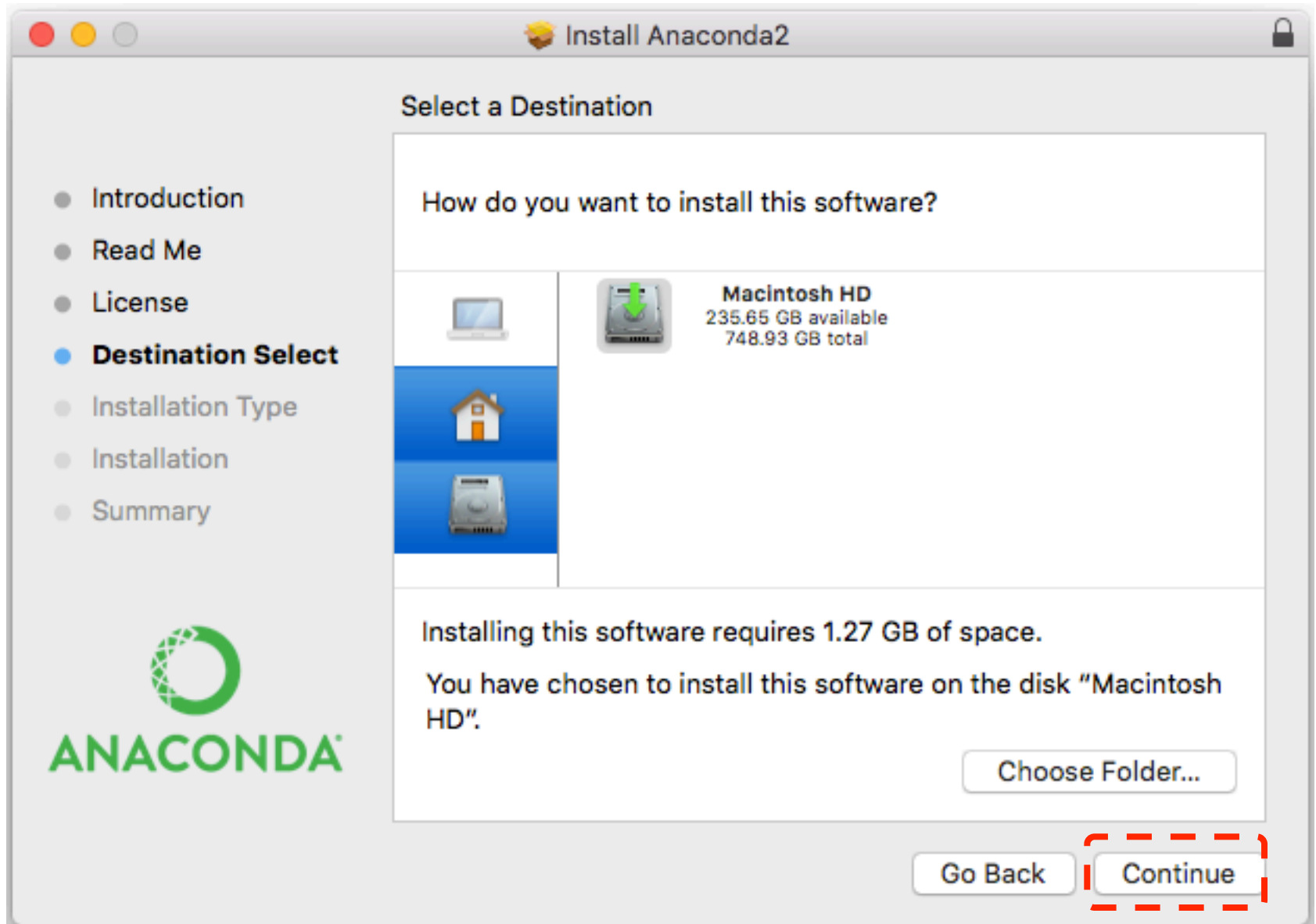
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS

At the bottom of the window, there are four buttons: 'Print...', 'Save...', 'Go Back', and 'Continue'. The 'Continue' button is highlighted with a red dashed border.

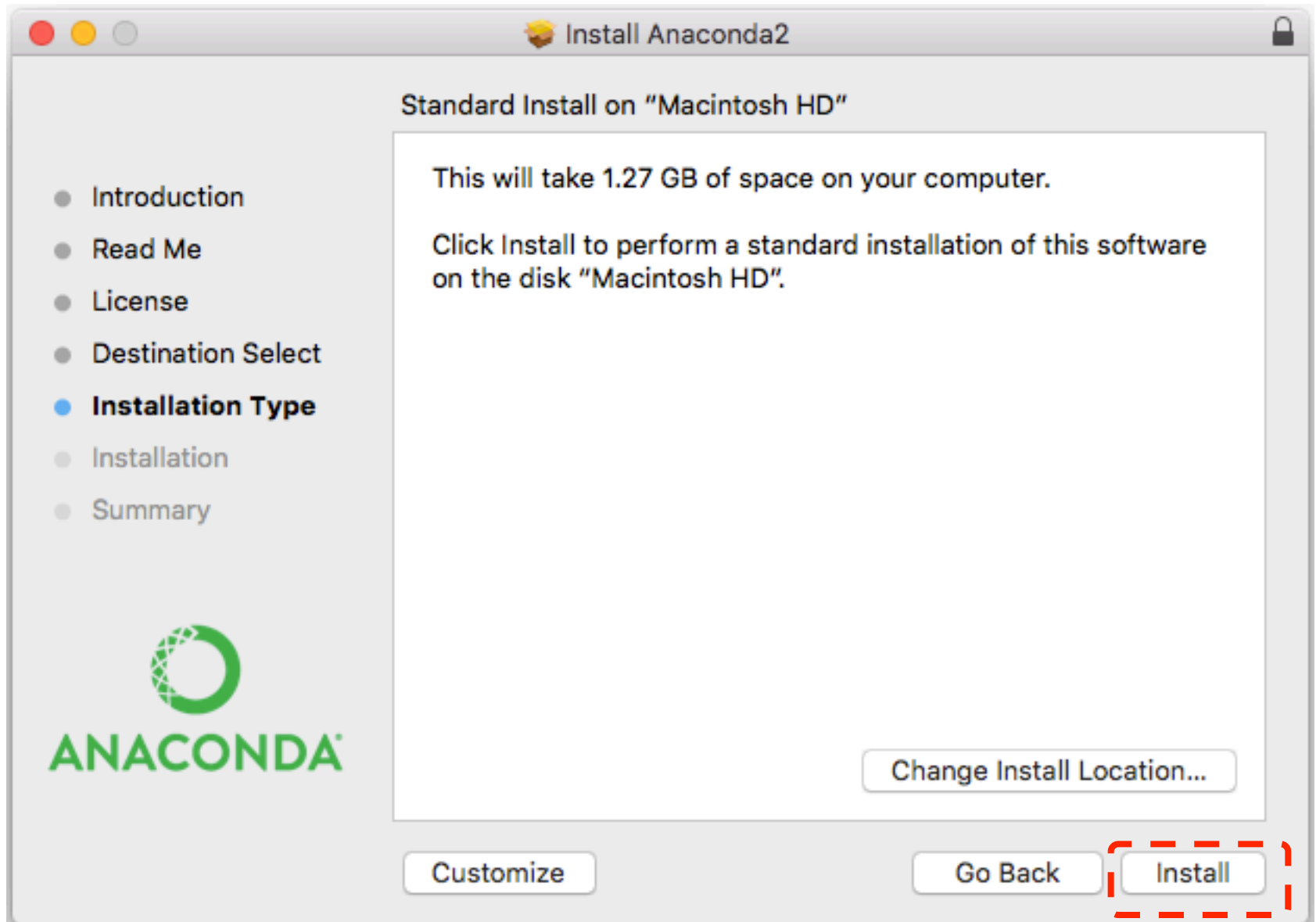
OS X Anaconda Installation



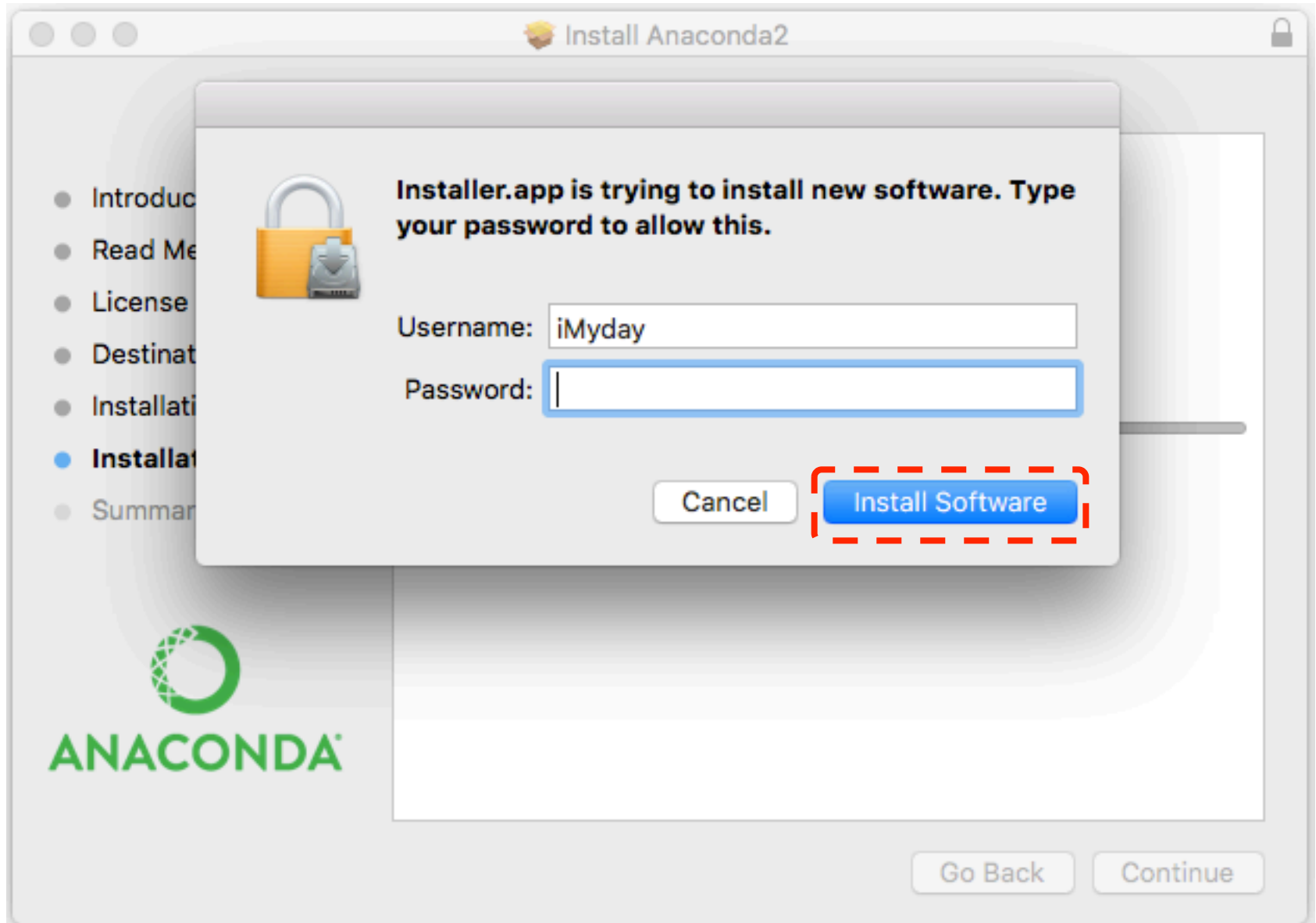
OS X Anaconda Installation



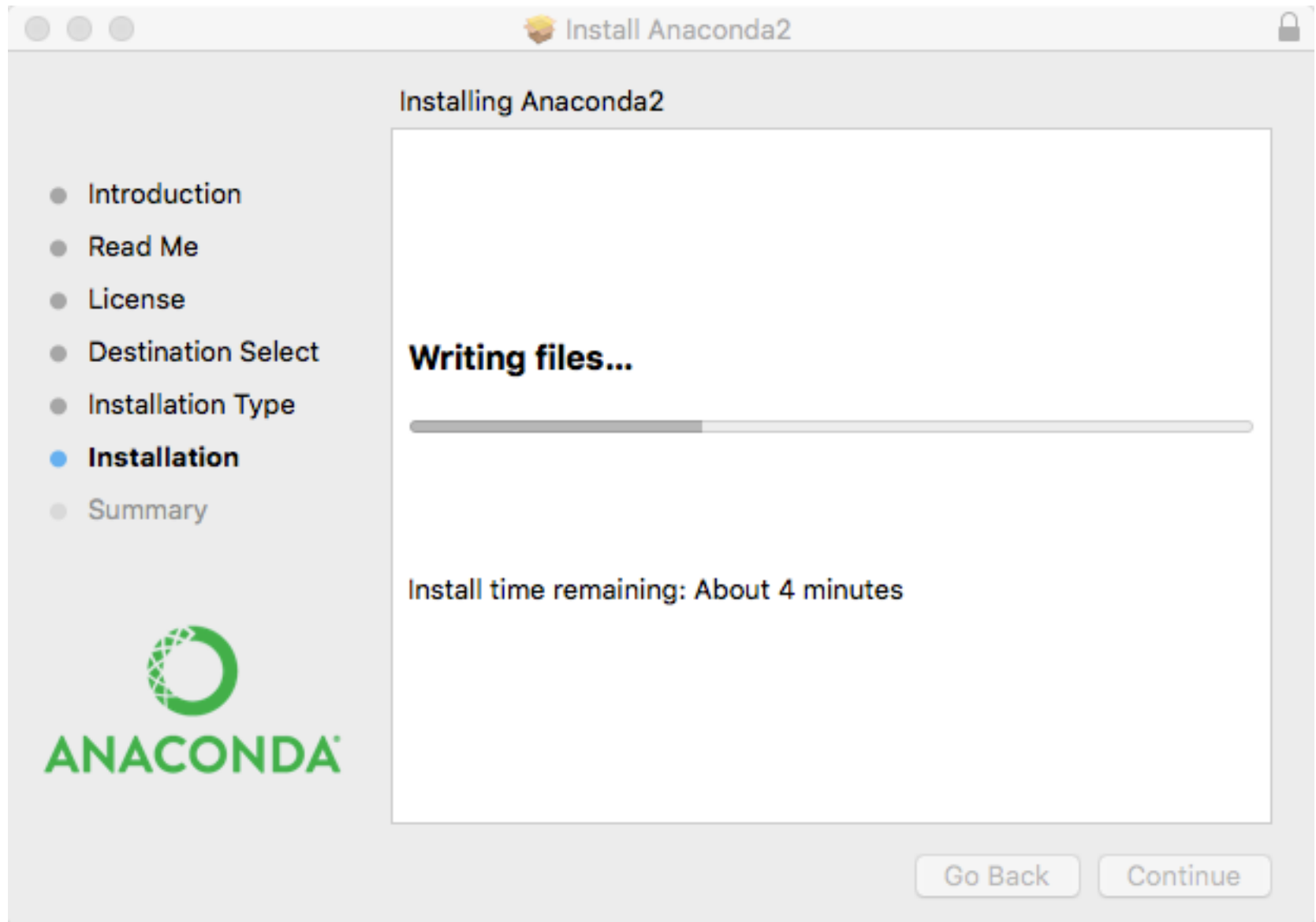
OS X Anaconda Installation



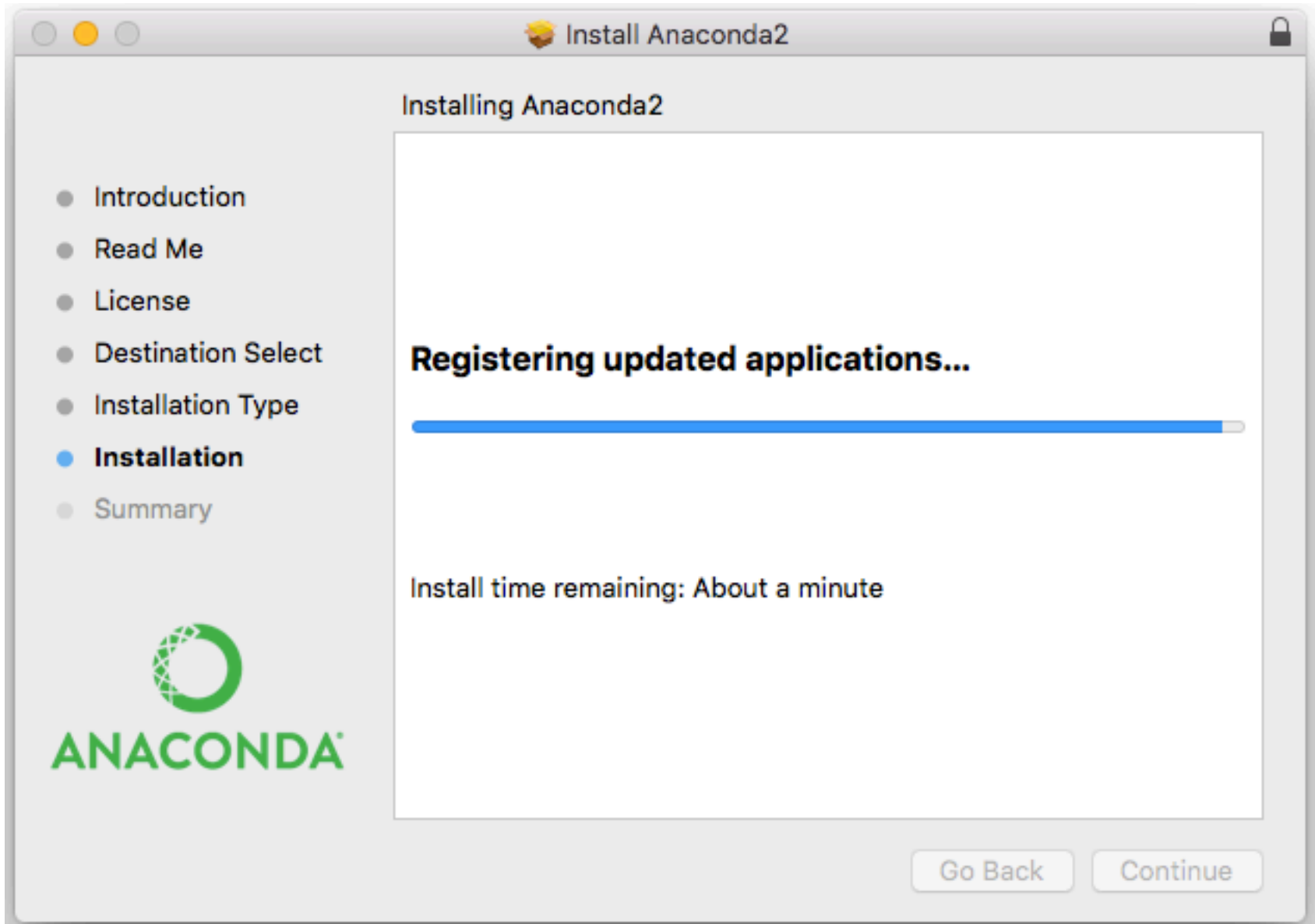
OS X Anaconda Installation



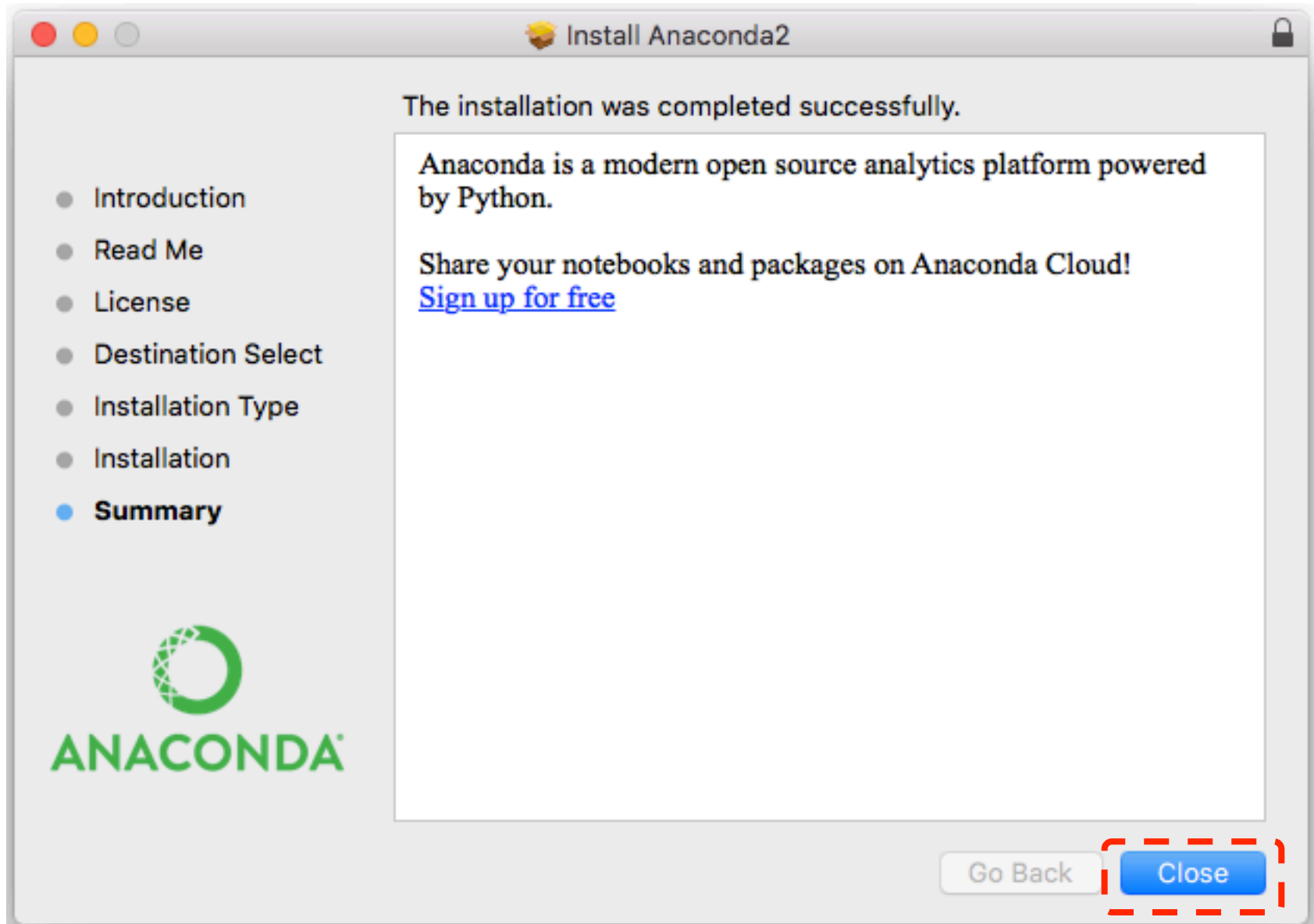
OS X Anaconda Installation



OS X Anaconda Installation



OS X Anaconda Installation



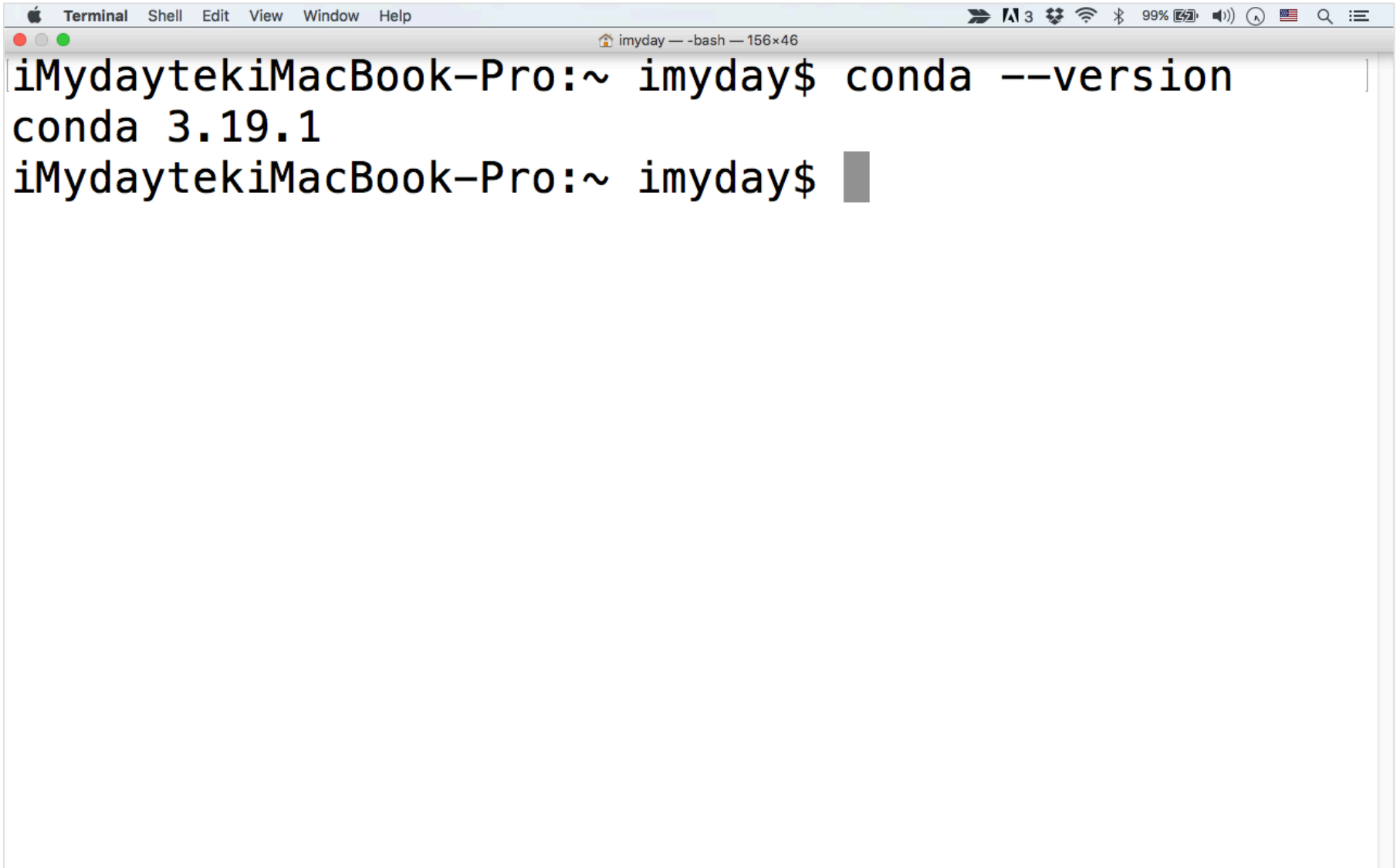
conda list

```
imyday — -bash — 80x24
[iMydaytekiMacBook-Pro:~ imyday$ conda list
# packages in environment at //anaconda:
#
abstract-rendering      0.5.1      np110py27_0
alabaster               0.7.7      py27_0
anaconda               2.5.0      np110py27_0
anaconda-client        1.2.2      py27_0
appnope                0.1.0      py27_0
appscript              1.0.1      py27_0
argcomplete            1.0.0      py27_1
astropy                1.1.1      np110py27_0
babel                  2.2.0      py27_0
backports-abc          0.4        <pip>
backports.ssl-match-hostname 3.4.0.2    <pip>
backports_abc          0.4        py27_0
beautifulsoup4         4.4.1      py27_0
bitarray               0.8.1      py27_0
blaze                  0.9.0      <pip>
blaze-core             0.9.0      py27_0
bokeh                  0.11.0     py27_0
boto                   2.39.0     py27_0
bottleneck             1.0.0      np110py27_0
cdecimal               2.3        py27_0
cffi                   1.2.1      py27_0
```

conda --version

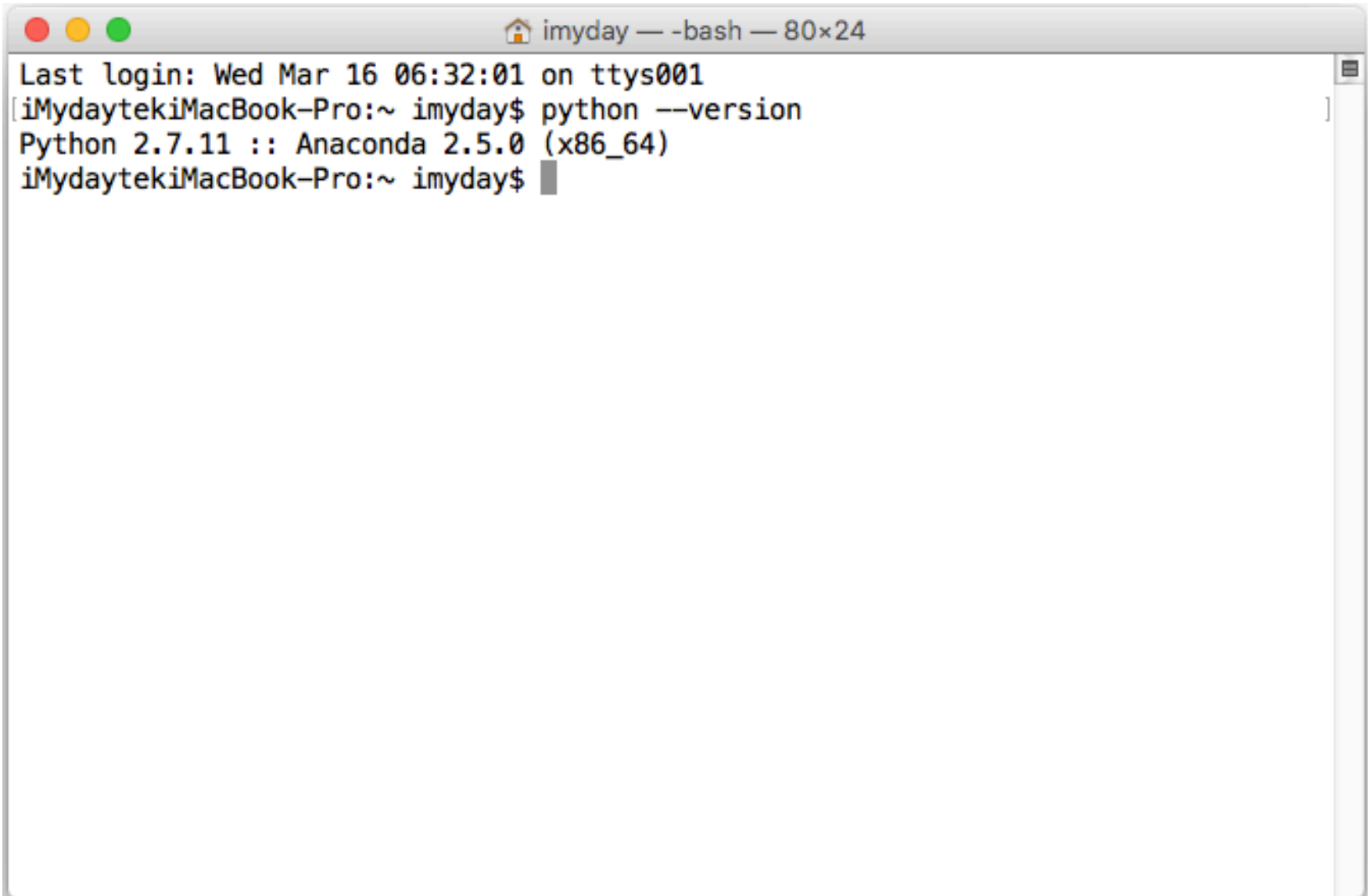
```
sqlite 3.9.2 0
ssl_match_hostname 3.4.0.2 py27_0
statsmodels 0.6.1 np110py27_0
sympy 0.7.6.1 py27_0
tables 3.2.2 <pip>
terminado 0.5 py27_1
tk 8.5.18 0
toolz 0.7.4 py27_0
tornado 4.3 py27_0
traitlets 4.1.0 py27_0
unicodcsv 0.14.1 py27_0
werkzeug 0.11.3 py27_0
wheel 0.26.0 py27_1
xlrd 0.9.4 py27_0
xlsxwriter 0.8.4 py27_0
xlwings 0.6.4 py27_0
xlwt 1.0.0 py27_0
xz 5.0.5 0
yaml 0.1.6 0
zeromq 4.1.3 0
zlib 1.2.8 0
[iMydaytekiMacBook-Pro:~ imyday$ conda --version
conda 3.19.1
iMydaytekiMacBook-Pro:~ imyday$
```

conda --version



```
Terminal Shell Edit View Window Help
imyday -- -bash -- 156x46
iMydaytekiMacBook-Pro:~ imyday$ conda --version
conda 3.19.1
iMydaytekiMacBook-Pro:~ imyday$
```


python --version

A terminal window titled 'imyday — -bash — 80x24' with a home icon. The window shows the following text: 'Last login: Wed Mar 16 06:32:01 on ttys001', '[iMydaytekiMacBook-Pro:~ imyday\$ python --version]', 'Python 2.7.11 :: Anaconda 2.5.0 (x86_64)', and '[iMydaytekiMacBook-Pro:~ imyday\$]'. The prompt is followed by a cursor.

```
imyday — -bash — 80x24
Last login: Wed Mar 16 06:32:01 on ttys001
[iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Anaconda 2.5.0 (x86_64)
iMydaytekiMacBook-Pro:~ imyday$ ]
```

ipython notebook

```
imyday — python • python.app //anaconda/bin/ipython notebook — 80x24
[iMydaytekiMacBook-Pro:~ imyday$ ipython notebook
[I 14:26:49.944 NotebookApp] Serving notebooks from local directory: /Users/imyday
ay
[I 14:26:49.944 NotebookApp] 0 active kernels
[I 14:26:49.944 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/
[I 14:26:49.944 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
[W 14:26:56.639 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c960677/channels?session_id=265FB16817FB4AB79202F6D3C3BDB0E6 (::1): Kernel does not exist: a87ab95b-6d6e-44d3-aaa7-c1901c960677
[W 14:26:56.663 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c960677/channels?session_id=265FB16817FB4AB79202F6D3C3BDB0E6 (::1) 95.43ms referer=None
[W 14:26:56.681 NotebookApp] 404 GET /api/kernels/b7fae9a6-d77b-4ead-832c-c070b18d642b/channels?session_id=EF4C761633E541C88568CDBCDE1091B7 (::1): Kernel does not exist: b7fae9a6-d77b-4ead-832c-c070b18d642b
[W 14:26:56.683 NotebookApp] 404 GET /api/kernels/b7fae9a6-d77b-4ead-832c-c070b18d642b/channels?session_id=EF4C761633E541C88568CDBCDE1091B7 (::1) 6.62ms referer=None
[W 14:27:29.595 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c960677/channels?session_id=265FB16817FB4AB79202F6D3C3BDB0E6 (::1): Kernel does not exist: a87ab95b-6d6e-44d3-aaa7-c1901c960677
[W 14:27:29.631 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c
```

conda search python

```
imyday — -bash — 80x24
iMydaytekiMacBook-Pro:~ imyday$ conda search python
Using Anaconda Cloud api site https://api.anaconda.org
Fetching package metadata: ....
biopython                1.60                np17py27_0 defaults
                        1.60                np17py26_0 defaults
                        1.1                  py27_0 defaults
                        1.1                  py26_0 defaults
                        1.1                  py27_1 defaults
                        1.1                  py26_1 defaults
                        1.2                  py27_0 defaults
                        1.2                  py26_0 defaults
                        1.2                  py33_1 defaults
                        1.2                  py27_1 defaults
                        1.2                  py26_1 defaults
                        1.2                  py34_2 defaults
                        1.2                  py33_2 defaults
                        1.2                  py27_2 defaults
                        1.2                  py26_2 defaults
                        1.2                  py34_3 defaults
                        1.2                  py33_3 defaults
                        1.2                  py27_3 defaults
                        1.2                  py26_3 defaults
                        1.2                  py35_4 defaults
                        1.2                  py34_4 defaults
                        1.2                  py33_4 defaults
                        * 1.2                  py27_4 defaults
                        1.2                  py26_4 defaults
wxpython                  3.0                  py27_0 defaults
iMydaytekiMacBook-Pro:~ imyday$
```

conda create -n py35 python=3.5 anaconda

```
imyday — conda create -n py35 python=3.5 anaconda — 80x24
[iMydaytekiMacBook-Pro:~ imyday$ conda create -n py35 python=3.5 anaconda
Using Anaconda Cloud api site https://api.anaconda.org
Fetching package metadata: ....
Solving package specifications: .....
Package plan for installation in environment //anaconda/envs/py35:

The following packages will be downloaded:
```

package	build	
openssl-1.0.2g	0	3.0 MB
xz-5.0.5	1	173 KB
python-3.5.1	0	12.7 MB
anaconda-custom	py35_0	3 KB
setuptools-20.2.2	py35_0	458 KB
wheel-0.29.0	py35_0	82 KB
pip-8.1.0	py35_0	1.6 MB
Total:		18.0 MB

```
The following NEW packages will be INSTALLED:

anaconda:    custom-py35_0
openssl:    1.0.2g-0
```

Create a Python 3.5 environment

Create a Python 3.5 environment

```
imyday — conda create -n py35 python=3.5 anaconda — 80x24
```

xz-5.0.5		1	173 KB
python-3.5.1		0	12.7 MB
anaconda-custom		py35_0	3 KB
setuptools-20.2.2		py35_0	458 KB
wheel-0.29.0		py35_0	82 KB
pip-8.1.0		py35_0	1.6 MB

Total:			18.0 MB

The following NEW packages will be INSTALLED:

```
anaconda:  custom-py35_0
openssl:   1.0.2g-0
pip:       8.1.0-py35_0
python:    3.5.1-0
readline:  6.2-2
setuptools: 20.2.2-py35_0
sqlite:    3.9.2-0
tk:        8.5.18-0
wheel:     0.29.0-py35_0
xz:        5.0.5-1
zlib:      1.2.8-0
```

Proceed ([y]/n)? **y**

Create a Python 3.5 environment

```
imyday — -bash — 80x24
zlib:      1.2.8-0

Proceed ([y]/n)? y

Fetching packages ...
openssl-1.0.2g 100% |#####| Time: 0:00:04 743.84 kB/s
xz-5.0.5-1.tar 100% |#####| Time: 0:00:00 228.62 kB/s
python-3.5.1-0 100% |#####| Time: 0:00:16 815.46 kB/s
anaconda-custo 100% |#####| Time: 0:00:00 2.70 MB/s
setuptools-20. 100% |#####| Time: 0:00:01 257.08 kB/s
wheel-0.29.0-p 100% |#####| Time: 0:00:00 95.97 kB/s
pip-8.1.0-py35 100% |#####| Time: 0:00:05 305.56 kB/s
Extracting packages ...
[ COMPLETE ] |#####| 100%
Linking packages ...
[ COMPLETE ] |#####| 100%
#
# To activate this environment, use:
# $ source activate py35
#
# To deactivate this environment, use:
# $ source deactivate
#
iMydaytekiMacBook-Pro:~ imyday$
```

source activate py35

```
imyday — -bash — 80x24

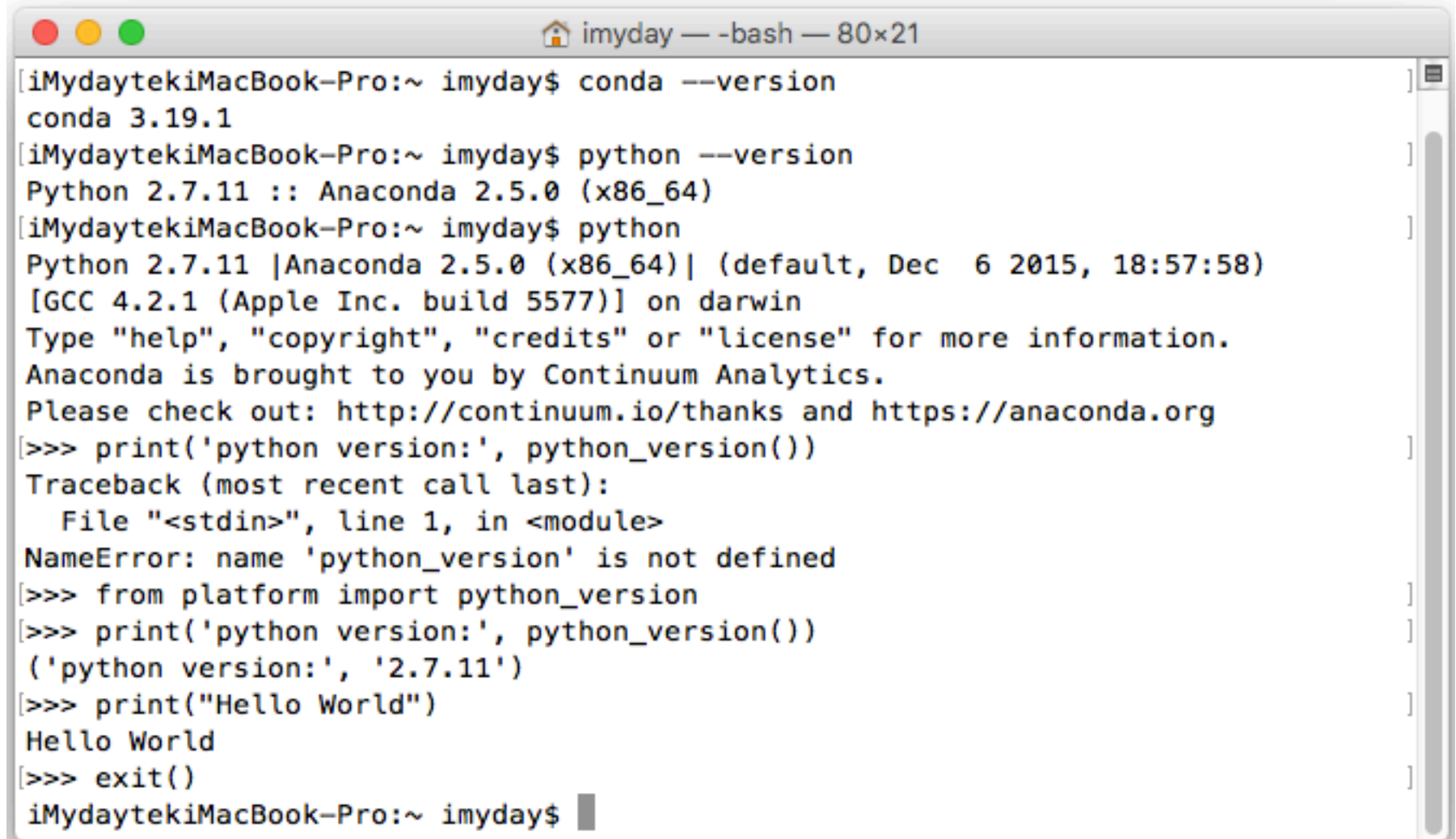
Fetching packages ...
openssl-1.0.2g 100% |#####| Time: 0:00:04 743.84 kB/s
xz-5.0.5-1.tar 100% |#####| Time: 0:00:00 228.62 kB/s
python-3.5.1-0 100% |#####| Time: 0:00:16 815.46 kB/s
anaconda-custo 100% |#####| Time: 0:00:00 2.70 MB/s
setuptools-20. 100% |#####| Time: 0:00:01 257.08 kB/s
wheel-0.29.0-p 100% |#####| Time: 0:00:00 95.97 kB/s
pip-8.1.0-py35 100% |#####| Time: 0:00:05 305.56 kB/s
Extracting packages ...
[ COMPLETE ]|#####| 100%
Linking packages ...
[ COMPLETE ]|#####| 100%
#
# To activate this environment, use:
# $ source activate py35
#
# To deactivate this environment, use:
# $ source deactivate
#
[iMydaytekiMacBook-Pro:~ imyday$ source activate py35
discarding //anaconda/bin from PATH
prepending //anaconda/envs/py35/bin to PATH
(py35)iMydaytekiMacBook-Pro:~ imyday$
```

python --version

```
imyday — -bash — 80x24
setuptools-20. 100% |#####| Time: 0:00:01 257.08 kB/s
wheel-0.29.0-p 100% |#####| Time: 0:00:00 95.97 kB/s
pip-8.1.0-py35 100% |#####| Time: 0:00:05 305.56 kB/s
Extracting packages ...
[ COMPLETE ]|#####| 100%
Linking packages ...
[ COMPLETE ]|#####| 100%
#
# To activate this environment, use:
# $ source activate py35
#
# To deactivate this environment, use:
# $ source deactivate
#
[iMydaytekiMacBook-Pro:~ imyday$ source activate py35
discarding //anaconda/bin from PATH
prepending //anaconda/envs/py35/bin to PATH
(py35)iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 3.5.1 :: Anaconda custom (x86_64)
(py35)iMydaytekiMacBook-Pro:~ imyday$ source deactivate
discarding //anaconda/envs/py35/bin from PATH
[iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Anaconda 2.5.0 (x86_64)
iMydaytekiMacBook-Pro:~ imyday$
```



```
from platform import python_version
print('Python Version:', python_version())
```



```
imyday — -bash — 80x21
[iMydaytekiMacBook-Pro:~ imyday$ conda --version
conda 3.19.1
[iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Anaconda 2.5.0 (x86_64)
[iMydaytekiMacBook-Pro:~ imyday$ python
Python 2.7.11 |Anaconda 2.5.0 (x86_64)| (default, Dec 6 2015, 18:57:58)
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
[>>> print('python version:', python_version())
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'python_version' is not defined
[>>> from platform import python_version
[>>> print('python version:', python_version())
('python version:', '2.7.11')
[>>> print("Hello World")
Hello World
[>>> exit()
imydaytekiMacBook-Pro:~ imyday$
```

conda info --envs

```
imyday — -bash — 80x12
[iMydaytekiMacBook-Pro:~ imyday$ conda --version
conda 3.19.1
[iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Anaconda 2.5.0 (x86_64)
[iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py35                //anaconda/envs/py35
root                * //anaconda

iMydaytekiMacBook-Pro:~ imyday$
```

conda --version

python --version

```
imyday — -bash — 80x24
[iMydaytekiMacBook-Pro:~ imyday$ conda --version
conda 3.19.1
[iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Anaconda 2.5.0 (x86_64)
[iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py35                //anaconda/envs/py35
root                * //anaconda

[iMydaytekiMacBook-Pro:~ imyday$ conda create -n py27 python=2.7 anaconda
Using Anaconda Cloud api site https://api.anaconda.org
Fetching package metadata: ....
Solving package specifications: .....
Package plan for installation in environment //anaconda/envs/py27:

The following packages will be downloaded:
```

package	build	
anaconda-custom	py27_0	3 KB
setuptools-20.2.2	py27_0	453 KB
wheel-0.29.0	py27_0	81 KB

conda info --envs

```

iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
anaconda-custo 100% |#####| Time: 0:00:00 2.23 MB/s
setuptools-20. 100% |#####| Time: 0:00:01 297.52 kB/s
wheel-0.29.0-p 100% |#####| Time: 0:00:00 98.03 kB/s
pip-8.1.0-py27 100% |#####| Time: 0:00:05 303.28 kB/s
Extracting packages ...
[ COMPLETE ]|#####| 100%
Linking packages ...
[ COMPLETE ]|#####| 100%
#
# To activate this environment, use:
# $ source activate py27
#
# To deactivate this environment, use:
# $ source deactivate
#
[iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py27 //anaconda/envs/py27
py35 //anaconda/envs/py35
root * //anaconda
iMydaytekiMacBook-Pro:~ imyday$
```

py27
py35

Source activate py35 conda install notebook ipykernel

```
iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py27          //anaconda/envs/py27
py35          //anaconda/envs/py35
root         * //anaconda

[iMydaytekiMacBook-Pro:~ imyday$ source activate py35
discarding //anaconda/bin from PATH
prepending //anaconda/envs/py35/bin to PATH
(py35)iMydaytekiMacBook-Pro:~ imyday$ conda install notebook ipykernel
Using Anaconda Cloud api site https://api.anaconda.org
Fetching package metadata: ....
Solving package specifications: .....
Package plan for installation in environment //anaconda/envs/py35:

The following packages will be downloaded:
```

package	build	
appnope-0.1.0	py35_0	4 KB
decorator-4.0.9	py35_0	7 KB
ipython_genutils-0.1.0	py35_0	33 KB

conda install notebook ipykernel

```
imyday — -bash — 80x24

ipython:          4.1.2-py35_0
ipython_genutils: 0.1.0-py35_0
jinja2:          2.8-py35_0
jsonschema:      2.4.0-py35_0
jupyter_client:  4.2.1-py35_0
jupyter_core:    4.1.0-py35_0
markupsafe:      0.23-py35_0
mistune:         0.7.2-py35_0
nbconvert:       4.1.0-py35_0
nbformat:        4.0.1-py35_0
notebook:        4.1.0-py35_1
path.py:         8.1.2-py35_1
pexpect:         3.3-py35_0
pickleshare:     0.5-py35_0
ptyprocess:      0.5-py35_0
pygments:       2.1.1-py35_0
python.app:      1.2-py35_4
pyzmq:          15.2.0-py35_0
simplegeneric:    0.8.1-py35_0
terminado:       0.5-py35_1
tornado:         4.3-py35_0
traitlets:       4.1.0-py35_0

Proceed ([y]/n)? y
```

conda install notebook ipykernel

```
imyday — -bash — 80x24
mistune-0.7.2- 100% |#####| Time: 0:00:01 121.50 kB/s
path.py-8.1.2- 100% |#####| Time: 0:00:00 92.90 kB/s
pexpect-3.3-py 100% |#####| Time: 0:00:00 108.41 kB/s
ptyprocess-0.5 100% |#####| Time: 0:00:00 63.16 kB/s
pygments-2.1.1 100% |#####| Time: 0:00:05 233.98 kB/s
python.app-1.2 100% |#####| Time: 0:00:00 3.27 MB/s
pyzmq-15.2.0-p 100% |#####| Time: 0:00:03 198.64 kB/s
simplegeneric- 100% |#####| Time: 0:00:00 2.32 MB/s
tornado-4.3-py 100% |#####| Time: 0:00:01 333.25 kB/s
jinja2-2.8-py3 100% |#####| Time: 0:00:01 176.77 kB/s
pickleshare-0. 100% |#####| Time: 0:00:00 2.84 MB/s
terminado-0.5- 100% |#####| Time: 0:00:00 59.95 kB/s
traitlets-4.1. 100% |#####| Time: 0:00:01 72.11 kB/s
ipython-4.1.2- 100% |#####| Time: 0:00:02 381.63 kB/s
jupyter_core-4 100% |#####| Time: 0:00:00 102.62 kB/s
jupyter_client 100% |#####| Time: 0:00:00 121.92 kB/s
nbformat-4.0.1 100% |#####| Time: 0:00:00 127.42 kB/s
ipykernel-4.3. 100% |#####| Time: 0:00:01 108.68 kB/s
nbconvert-4.1. 100% |#####| Time: 0:00:01 142.17 kB/s
notebook-4.1.0 100% |#####| Time: 0:00:09 480.58 kB/s
Extracting packages ...
[ COMPLETE ] |#####| 100%
Linking packages ...
[ COMPLETE ] |#####| 100%
```

Source activate py27 conda install notebook ipykernel

```
iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py27          //anaconda/envs/py27
py35          //anaconda/envs/py35
root         * //anaconda

iMydaytekiMacBook-Pro:~ imyday$ source activate py27
discarding //anaconda/bin from PATH
prepending //anaconda/envs/py27/bin to PATH
(py27)iMydaytekiMacBook-Pro:~ imyday$ conda install notebook ipykernel
Using Anaconda Cloud api site https://api.anaconda.org
Fetching package metadata: ....
Solving package specifications: .....
Package plan for installation in environment //anaconda/envs/py27:

The following packages will be downloaded:
```

package	build	
decorator-4.0.9	py27_0	12 KB
mistune-0.7.2	py27_0	178 KB
pygments-2.1.1	py27_0	1.2 MB

conda install notebook ipykernel

```
imyday — -bash — 80x24

jsonschema:      2.4.0-py27_0
jupyter_client:  4.2.1-py27_0
jupyter_core:    4.1.0-py27_0
markupsafe:      0.23-py27_0
mistune:         0.7.2-py27_0
nbconvert:       4.1.0-py27_0
nbformat:        4.0.1-py27_0
notebook:        4.1.0-py27_1
path.py:         8.1.2-py27_1
pexpect:         3.3-py27_0
pickleshare:    0.5-py27_0
ptyprocess:     0.5-py27_0
pygments:       2.1.1-py27_0
python.app:     1.2-py27_4
pyzmq:          15.2.0-py27_0
simplegeneric:   0.8.1-py27_0
singledispatch: 3.4.0.3-py27_0
six:            1.10.0-py27_0
ssl_match_hostname: 3.4.0.2-py27_0
terminado:      0.5-py27_1
tornado:        4.3-py27_0
traitlets:     4.1.0-py27_0

Proceed ([y]/n)? y
```

conda install notebook ipykernel

```
simplegeneric:      0.8.1-py27_0
singledispatch:    3.4.0.3-py27_0
six:                1.10.0-py27_0
ssl_match_hostname: 3.4.0.2-py27_0
terminado:         0.5-py27_1
tornado:           4.3-py27_0
traitlets:         4.1.0-py27_0
```

Proceed ([y]/n)? y

Fetching packages ...

```
decorator-4.0. 100% |#####| Time: 0:00:00 48.24 kB/s
mistune-0.7.2- 100% |#####| Time: 0:00:01 177.69 kB/s
pygments-2.1.1 100% |#####| Time: 0:00:03 354.58 kB/s
ipython-4.1.2- 100% |#####| Time: 0:00:04 210.93 kB/s
jupyter_core-4 100% |#####| Time: 0:00:00 71.68 kB/s
jupyter_client 100% |#####| Time: 0:00:00 111.21 kB/s
ipykernel-4.3. 100% |#####| Time: 0:00:00 145.24 kB/s
notebook-4.1.0 100% |#####| Time: 0:00:05 816.06 kB/s
```

Extracting packages ...

```
[ COMPLETE ] |#####| 100%
```

Linking packages ...

```
[ COMPLETE ] |#####| 100%
```

```
(py27)iMydaytekiMacBook-Pro:~ imyday$
```

python --version

```
imyday — -bash — 80x24
Fetching packages ...
decorator-4.0. 100% |#####| Time: 0:00:00 48.24 kB/s
mistune-0.7.2- 100% |#####| Time: 0:00:01 177.69 kB/s
pygments-2.1.1 100% |#####| Time: 0:00:03 354.58 kB/s
ipython-4.1.2- 100% |#####| Time: 0:00:04 210.93 kB/s
jupyter_core-4 100% |#####| Time: 0:00:00 71.68 kB/s
jupyter_client 100% |#####| Time: 0:00:00 111.21 kB/s
ipykernel-4.3. 100% |#####| Time: 0:00:00 145.24 kB/s
notebook-4.1.0 100% |#####| Time: 0:00:05 816.06 kB/s
Extracting packages ...
[ COMPLETE ] |#####| 100%
Linking packages ...
[ COMPLETE ] |#####| 100%
[ (py27) iMydaytekiMacBook-Pro:~ imyday$ conda info --envs ]
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py27          * //anaconda/envs/py27
py35          //anaconda/envs/py35
root         //anaconda

[ (py27) iMydaytekiMacBook-Pro:~ imyday$ python --version ]
Python 2.7.11 :: Anaconda custom (x86_64)
(py27) iMydaytekiMacBook-Pro:~ imyday$
```

source deactivate

```
imyday — -bash — 80x24
Extracting packages ...
[ COMPLETE ]|#####| 100%
Linking packages ...
[ COMPLETE ]|#####| 100%
[ (py27) iMydaytekiMacBook-Pro:~ imyday$ conda info --envs ]
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py27          * //anaconda/envs/py27
py35          //anaconda/envs/py35
root          //anaconda

[ (py27) iMydaytekiMacBook-Pro:~ imyday$ python --version ]
Python 2.7.11 :: Anaconda custom (x86_64)
[ (py27) iMydaytekiMacBook-Pro:~ imyday$ source activate py35 ]
discarding //anaconda/envs/py27/bin from PATH
prepending //anaconda/envs/py35/bin to PATH
[ (py35) iMydaytekiMacBook-Pro:~ imyday$ python --version ]
Python 3.5.1 :: Anaconda custom (x86_64)
[ (py35) iMydaytekiMacBook-Pro:~ imyday$ source deactivate ]
discarding //anaconda/envs/py35/bin from PATH
[ iMydaytekiMacBook-Pro:~ imyday$ python --version ]
Python 2.7.11 :: Anaconda 2.5.0 (x86_64)
iMydaytekiMacBook-Pro:~ imyday$
```

ipython notebook

ipython notebook

```
imyday — python • python.app //anaconda/bin/ipython notebook — 80x24
[iMydaytekiMacBook-Pro:~ imyday$ ipython notebook
[I 14:26:49.944 NotebookApp] Serving notebooks from local directory: /Users/imyday
ay
[I 14:26:49.944 NotebookApp] 0 active kernels
[I 14:26:49.944 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/
[I 14:26:49.944 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
[W 14:26:56.639 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c960677/channels?session_id=265FB16817FB4AB79202F6D3C3BDB0E6 (::1): Kernel does not exist: a87ab95b-6d6e-44d3-aaa7-c1901c960677
[W 14:26:56.663 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c960677/channels?session_id=265FB16817FB4AB79202F6D3C3BDB0E6 (::1) 95.43ms referer=None
[W 14:26:56.681 NotebookApp] 404 GET /api/kernels/b7fae9a6-d77b-4ead-832c-c070b18d642b/channels?session_id=EF4C761633E541C88568CDBCDE1091B7 (::1): Kernel does not exist: b7fae9a6-d77b-4ead-832c-c070b18d642b
[W 14:26:56.683 NotebookApp] 404 GET /api/kernels/b7fae9a6-d77b-4ead-832c-c070b18d642b/channels?session_id=EF4C761633E541C88568CDBCDE1091B7 (::1) 6.62ms referer=None
[W 14:27:29.595 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c960677/channels?session_id=265FB16817FB4AB79202F6D3C3BDB0E6 (::1): Kernel does not exist: a87ab95b-6d6e-44d3-aaa7-c1901c960677
[W 14:27:29.631 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c
```

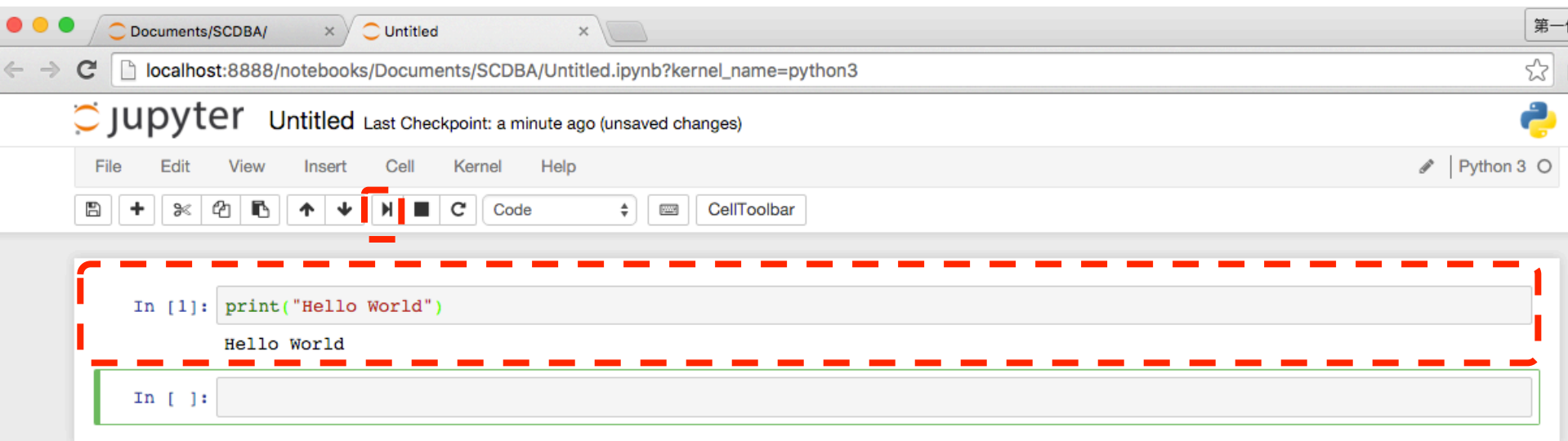
jupyter notebook

Python 3

The screenshot shows the Jupyter Notebook web interface. The browser address bar displays `localhost:8888/tree/Documents/SCDBA`. The Jupyter logo is visible in the top left. Below the logo, there are tabs for 'Files', 'Running', and 'Clusters'. A message reads 'Select items to perform actions on them.' The file browser shows the path `/ Documents / SCDBA` with a '..' folder icon. Below the file browser, it says 'Notebook list empty.' On the right side, there is an 'Upload' button and a 'New' dropdown menu. The 'New' menu is open, showing options: 'Text File', 'Folder', 'Terminal', 'Notebooks', and 'Python 3'. The 'New' button and the 'Python 3' option are highlighted with a red dashed border.

jupyter notebook

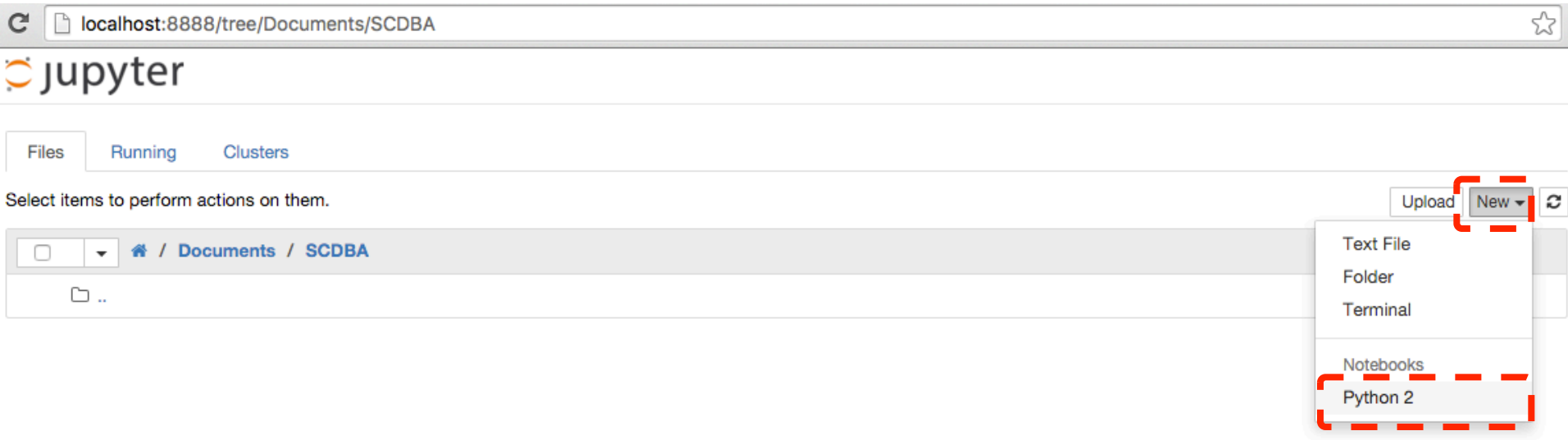
Python 3



The image shows a web browser window displaying a Jupyter Notebook. The browser's address bar shows the URL `localhost:8888/notebooks/Documents/SCDBA/Untitled.ipynb?kernel_name=python3`. The notebook interface includes a menu bar with options: File, Edit, View, Insert, Cell, Kernel, and Help. Below the menu is a toolbar with icons for saving, adding cells, undo, redo, and running code. The main area contains two code cells. The first cell, highlighted with a red dashed border, contains the code `In [1]: print("Hello World")` and has executed, showing the output `Hello World`. The second cell is empty and contains `In []:`.

jupyter notebook

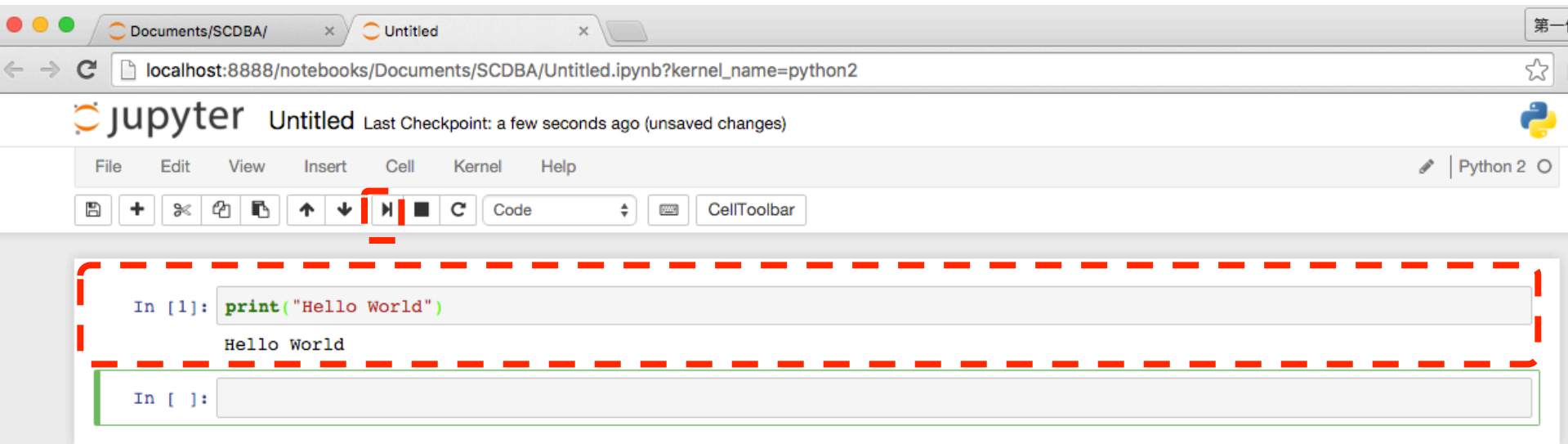
Python 2



The screenshot shows the Jupyter Notebook interface in a web browser. The address bar displays the URL `localhost:8888/tree/Documents/SCDBA`. The Jupyter logo is visible in the top left. Below the logo, there are tabs for `Files`, `Running`, and `Clusters`. A message states "Select items to perform actions on them." Below this, the breadcrumb path is `Documents / SCDBA`. A file browser shows a folder icon and `..`. On the right side, there are buttons for `Upload`, `New`, and a refresh icon. The `New` dropdown menu is open, showing options: `Text File`, `Folder`, `Terminal`, `Notebooks`, and `Python 2`. The `Python 2` option is highlighted with a red dashed border.

jupyter notebook

Python 2



The screenshot displays a web browser window with a Jupyter Notebook interface. The browser's address bar shows the URL `localhost:8888/notebooks/Documents/SCDBA/Untitled.ipynb?kernel_name=python2`. The notebook's title bar reads "jupyter Untitled" and includes a status message: "Last Checkpoint: a few seconds ago (unsaved changes)". A menu bar is visible with options: File, Edit, View, Insert, Cell, Kernel, and Help. Below the menu is a toolbar with icons for saving, adding cells, undo, redo, and running code. The main area contains two code cells. The first cell, highlighted with a red dashed border, contains the code `In [1]: print("Hello World")` and has executed, displaying the output `Hello World`. The second cell, highlighted with a green border, is currently empty and shows `In []:`.

ipython notebook

jupyter notebook

The screenshot shows the Jupyter Notebook interface in a web browser. The address bar displays "localhost:8888/tree". The Jupyter logo is visible at the top left. Below the logo, there are three tabs: "Files", "Running", and "Clusters". The "Files" tab is active. Below the tabs, there is a prompt "Select items to perform actions on them." and two buttons: "Upload" and "New". A list of files and folders is displayed, each with a checkbox and a folder icon. The list includes: AndroidStudioProjects, app, Applications, AppsPro, bin, Desktop, Development, Documents, Downloads, Dropbox, imtkuapp5, jEdit, man, Movies, Music, OneDrive, and Pictures.

localhost:8888/tree

jupyter

Files Running Clusters

Select items to perform actions on them. Upload New

- AndroidStudioProjects
- app
- Applications
- AppsPro
- bin
- Desktop
- Development
- Documents
- Downloads
- Dropbox
- imtkuapp5
- jEdit
- man
- Movies
- Music
- OneDrive
- Pictures

jupyter notebook

Home x

localhost:8888/tree

第一位使用者

jupyter

Files Running Clusters

Select items to perform actions on them.

Upload New ↕

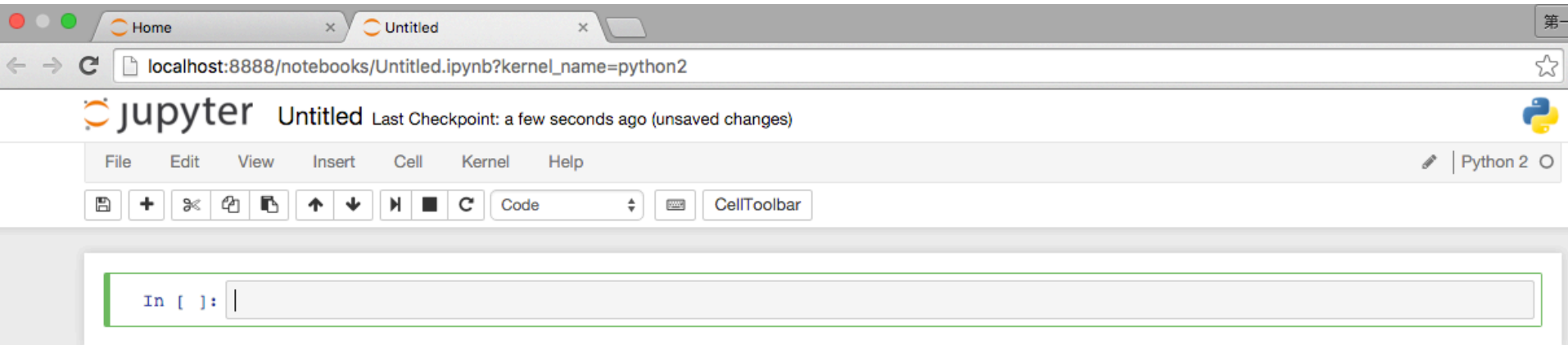
- Home
- AndroidStudioProjects
- app
- Applications
- AppsPro
- bin
- Desktop
- Development
- Documents
- Downloads
- Dropbox
- imtkuapp5
- jEdit
- man
- Movies
- Music
- OneDrive
- Pictures

Text File
Folder
Terminal
Notebooks
Python 2

Create a new notebook with Python 2

localhost:8888/tree#

jupyter notebook



```
print('Hello World, Python')
```

The screenshot shows a web browser window with the address bar containing `localhost:8888/notebooks/Untitled.ipynb?kernel_name=python2`. The Jupyter interface includes a header with the logo, the name "Untitled", and a status message "Last Checkpoint: a minute ago (unsaved changes)". A menu bar contains "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help". On the right, there is a "Python 2" indicator. Below the menu is a toolbar with icons for file operations and cell actions, along with a "Code" dropdown and a "CellToolbar" button. The main area contains a single code cell with the text `In []: print('Hello World, Python')`.

```
print('Hello World, Python')
```

The screenshot shows a Jupyter Notebook interface. At the top, the browser address bar displays `localhost:8888/notebooks/Untitled.ipynb?kernel_name=python2`. Below the browser, the Jupyter logo and the text "Untitled" are visible, along with a status message "Last Checkpoint: 3 minutes ago (unsaved changes)". The main menu includes "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help". On the right side of the menu, there is a pencil icon and the text "Python 2". Below the menu is a toolbar with various icons for file operations, navigation, and execution. The main content area contains two code cells. The first cell, labeled "In [1]:", contains the code `print('Hello World, Python')` and has the output "Hello World, Python" displayed below it. The second cell, labeled "In []:", is currently empty.

Conda Get-Started

← → ↻ conda.pydata.org/docs/get-started.html ☆

🏠 Conda

Search docs

Get started

Intro to conda

Download conda

Installation

Test drive

Conda cheat sheet

Using conda

Building packages

Help & reference

Get involved

Docs » Get started

[Edit on GitHub](#)

Get started

- [Intro to conda](#)
- [Download conda](#)
 - [Should I download Anaconda or Miniconda?](#)
 - [Which version of Anaconda or Miniconda should I choose?](#)
 - [Should I choose GUI installer or command line installer?](#)
 - [What version of Python should I choose?](#)
 - [What about cryptographic hash verification?](#)
- [Installation](#)
 - [Quick install](#)
 - [Miniconda quick install requirements](#)
 - [Windows Miniconda install](#)
 - [OS X Miniconda install](#)
 - [Linux Miniconda install](#)
 - [Full installation](#)
 - [Anaconda requirements](#)
 - [Install instructions](#)
 - [Windows Anaconda install](#)
 - [OS X Anaconda install](#)
 - [Linux Anaconda install](#)
 - [Configuration](#)
 - [The conda configuration file \(.condarc\)](#)
 - [General configuration](#)
 - [Channel locations \(channels\)](#)
 - [Always yes \(always yes\)](#)

<http://conda.pydata.org/docs/get-started.html>

Update or Upgrade Python

If you are in an environment with Python version 3.4.2, this command will update Python to 3.4.3, which is the latest version in the 3.4 branch:

```
$ conda update python
```

Upgrade Python to another branch such as 3.5 by installing that version of Python:

```
$ conda install python=3.5
```

Python-Future

python-future.org/index.html

Python-Future



Overview

Cheat Sheet

FAQ

Contents ▾

Page ▾

Search



Fork me on GitHub

Easy, clean, reliable Python 2/3 compatibility

Table of Contents

What's New

Overview: Easy, clean, reliable Python 2/3 compatibility

Quick-start guide

Cheat Sheet: Writing Python 2-3 compatible code

Imports

What else you need to know

Automatic conversion to Py2/3

Frequently Asked Questions (FAQ)

Standard library incompatibilities

Older interfaces

Changes in previous versions

Licensing and credits

API Reference (in progress)

Easy, clean, reliable Python 2/3 compatibility

`python-future` is the missing compatibility layer between Python 2 and Python 3. It allows you to use a single, clean Python 3.x-compatible codebase to support both Python 2 and Python 3 with minimal overhead.

Contents:

- [What's New](#)
 - [What's new in version 0.15.2 \(2015-09-11\)](#)
 - [What's new in version 0.15.1 \(2015-09-09\)](#)
 - [What's new in version 0.15.0 \(2015-07-25\)](#)
 - [Previous versions](#)
- [Overview: Easy, clean, reliable Python 2/3 compatibility](#)
 - [Features](#)
 - [Code examples](#)
 - [Automatic conversion to Py2/3-compatible code](#)
 - [Automatic translation](#)
 - [Licensing](#)
 - [Next steps](#)
- [Quick-start guide](#)
 - [Installation](#)
 - [If you are writing code from scratch](#)
 - [To convert existing Python 3 code](#)
 - [To convert existing Python 2 code](#)
 - [Standard library reorganization](#)
 - [Python 2-only dependencies](#)
 - [Next steps](#)
- [Cheat Sheet: Writing Python 2-3 compatible code](#)
 - [Setup](#)
 - [Essential syntax differences](#)
 - [Strings and bytes](#)

<http://python-future.org/index.html>

pip install future

pip install six

The imports below refer to these pip-installable packages on PyPI:

```
import future           # pip install future
import builtins         # pip install future
import past             # pip install future
import six              # pip install six
```

```
futurize                # pip install future
pasteurize              # pip install future
```

print

```
# Python 2 only:  
print 'Hello'
```

```
# Python 2 and 3:  
print('Hello')
```

```
# Python 2 only:  
print 'Hello', 'Guido'
```

```
# Python 2 and 3:  
from __future__ import print_function #(at top of module)  
  
print('Hello', 'Guido')
```

Writing Python 2-3 compatible code

Essential syntax differences

print

```
# Python 2 only:  
print 'Hello'
```

```
# Python 2 and 3:  
print('Hello')
```

To print multiple strings, import `print_function` to prevent Py2 from interpreting it as a tuple:

```
# Python 2 only:  
print 'Hello', 'Guido'
```

```
# Python 2 and 3:  
from __future__ import print_function    # (at top of module)  
  
print('Hello', 'Guido')
```

Unicode (text) string literals

```
# Python 2 only  
s1 = 'The Zen of Python'  
s2 = u'きたないのよりきれいな方がいい\n'
```

```
# Python 2 and 3  
s1 = u'The Zen of Python'  
s2 = u'きたないのよりきれいな方がいい\n'
```

Unicode (text) string literals

```
# Python 2 and 3
from __future__ import unicode_literals # at top of module

s1 = 'The Zen of Python'
s2 = 'きたないのよりきれいな方がいい\n'
```



Text input and output

```
print("Hello World")
```

```
print("Hello World\nThis is a message")
```

```
x = 3  
print(x)
```

```
x = 2  
y = 3  
print(x, ' ', y)
```

```
name = input("Enter a name: ")
```

```
x = int(input("What is x? "))
```

```
x = float(input("Write a number"))
```

Variables

```
x = 2
```

```
price = 2.5
```

```
word = 'Hello'
```

```
word = 'Hello'
```

```
word = "Hello"
```

```
word = '''Hello'''
```

```
x = 2
```

```
x = x + 1
```

```
x = 5
```

Python Basic Operators

```
print('7 + 2 =', 7 + 2)
print('7 - 2 =', 7 - 2)
print('7 * 2 =', 7 * 2)
print('7 / 2 =', 7 / 2)
print('7 // 2 =', 7 // 2)
print('7 % 2 =', 7 % 2)
print('7 ** 2 =', 7 ** 2)
```

```
print('7 + 2 =', 7 + 2)
print('7 - 2 =', 7 - 2)
print('7 * 2 =', 7 * 2)
print('7 / 2 =', 7 / 2)
print('7 // 2 =', 7 // 2)
print('7 % 2 =', 7 % 2)
print('7 ** 2 =', 7 ** 2)
```

```
7 + 2 = 9
7 - 2 = 5
7 * 2 = 14
7 / 2 = 3.5
7 // 2 = 3
7 % 2 = 1
7 ** 2 = 49
```

BMI Calculator in Python

```
height_cm = float(input("Enter your height in cm: "))
weight_kg = float(input("Enter your weight in kg: "))

height_m = height_cm/100
BMI = (weight_kg/(height_m**2))

print("Your BMI is: " + str(round(BMI,1)))
```

If statements

> greater than
< smaller than
== equals
!= is not

```
score = 80
if score >=60 :
    print( "Pass" )
else:
    print( "Fail" )
```

For loops

```
for i in range(1,11):  
    print(i)
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

For loops

```
for i in range(1,10):  
    for j in range(1,10):  
        print(i, ' * ', j, ' = ', i*j)
```

```
9 * 1 = 9  
9 * 2 = 18  
9 * 3 = 27  
9 * 4 = 36  
9 * 5 = 45  
9 * 6 = 54  
9 * 7 = 63  
9 * 8 = 72  
9 * 9 = 81
```

Functions

```
def convertCMtoM(xcm) :  
    m = xcm/100  
    return m
```

```
cm = 180  
m = convertCMtoM(cm)  
print(str(m))
```

1.8

Lists

```
x = [60, 70, 80, 90]
print(len(x))
print(x[0])
print(x[1])
print(x[-1])
```

```
60
70
90
```

Tuples

A **tuple** in Python is a collection that **cannot be modified**.

A tuple is defined using **parenthesis**.

```
x = (10, 20, 30, 40, 50)
```

```
print(x[0])
```

```
print(x[1])
```

```
print(x[2])
```

```
print(x[-1])
```

```
10
```

```
20
```

```
30
```

```
50
```

Python Ecosystem

Python Ecosystem

import math

```
x = log(1)
print(x)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-64-55d85b4998db> in <module>()
----> 1 x = log(1)
      2 print(x)
```

NameError: name 'log' is not defined

math.log?

```
import math
x = math.log(1)
print(x)
```

0.0

```
math.log(8, 2)|
```

3.0

Docstring:

```
log(x[, base])
```

Return the logarithm of x to the given base.

If the base not specified, returns the natural logarithm (base e) of x.

Type: builtin_function_or_method

NumPy

- NumPy provides a **multidimensional array** object to store homogenous or heterogeneous data; it also provides **optimized functions/methods** to operate on this array object.

NumPy

```
v = range(1, 6)  
print(v)  
2 * v  
import numpy as np  
v = np.arange(1, 6)  
v  
2 * v
```

```
v = range (1, 6)
print(v)
```

```
[1, 2, 3, 4, 5]
```

```
2 * v
```

```
[1, 2, 3, 4, 5, 1, 2, 3, 4, 5]
```

```
import numpy as np
v = np.arange(1, 6)
v
```

```
array([1, 2, 3, 4, 5])
```

```
2 * v
```

```
array([ 2,  4,  6,  8, 10])
```

Compatible Python 2 and Python 3 Code

- `print()`
- Exceptions
- Division
- Unicode strings
- Bad imports

Compatible Python 2 and Python 3 Code

```
print()  
print("This works in py2 and py3")  
  
from __future__ import print_function  
print("Hello", "World")
```

What version of Python should I choose?

- The latest version of Python 2 is 2.7, and that is included with Anaconda and Miniconda.
- The newest stable version of Python is 3.5, and that is included with Anaconda3 and Miniconda3.
- You can easily set up additional versions of Python such as 3.4 by downloading any version and creating a new environment with just a few clicks.

Create Python 2 or 3 environments

conda.pydata.org/docs/py2or3.html#create-python-2-or-3-environments

Conda

Search docs

Get started

Using conda

Overview

Managing environments

Managing Python

Managing packages

Using R with conda

Using Microsoft R Open

Building packages

Help & reference

Get involved

Create Python 2 or 3 environments

Anaconda supports Python 2.7, 3.4, and 3.5. The default is Python 2.7 or 3.5, depending on which installer you used. If the installer you used is Anaconda or Miniconda, the default is 2.7. If the installer you used is Anaconda3 or Miniconda3, the default is 3.5.

Create a Python 3.5 environment

To create a new environment with a different version of Python, use the `conda create` command. In this example, we'll make the new environment for Python 3.5:

```
$ conda create -n py35 python=3.5 anaconda
```

Here, the 'py35' is the name of the environment you want to create, and 'anaconda' is the meta-package that includes all of the actual Python packages comprising the Anaconda distribution. When creating a new environment and installing Anaconda, you can specify the exact package and Python versions, for example, `numpy=1.7` or `python=3.5`.

Create a Python 2.7 environment

In this example, we'll make a new environment for Python 2.7:

```
$ conda create -n py27 python=2.7 anaconda
```

File IO with open()

```
# Python 2 only
f = open('myfile.txt')
data = f.read() # as a byte string
text = data.decode('utf-8')

# Python 2 and 3: alternative 1
from io import open
f = open('myfile.txt', 'rb')
data = f.read() # as bytes
text = data.decode('utf-8') # unicode, not bytes

# Python 2 and 3: alternative 2
from io import open
f = open('myfile.txt', encoding='utf-8')
text = f.read() # unicode, not bytes
```

Six: Python 2 and 3 Compatibility Library

← → ↻ <https://pythonhosted.org/six/> ☆ 🌐 ☰

six 1.10.0 documentation » modules | index

Table Of Contents

- Six: Python 2 and 3 Compatibility Library
 - Indices and tables
 - Package contents
 - Constants
 - Object model compatibility
 - Syntax compatibility
 - Binary and text data
 - unittest assertions
 - Renamed modules and attributes compatibility
 - urllib parse
 - urllib error
 - urllib request
 - urllib response
 - Advanced - Customizing renames

This Page

[Show Source](#)

Quick search

Enter search terms or a module, class or function name.

Six: Python 2 and 3 Compatibility Library

Six provides simple utilities for wrapping over differences between Python 2 and Python 3. It is intended to support codebases that work on both Python 2 and 3 without modification. six consists of only one Python file, so it is painless to copy into a project.

Six can be downloaded on PyPi. Its bug tracker and code hosting is on [BitBucket](#).

The name, “six”, comes from the fact that 2*3 equals 6. Why not addition? Multiplication is more powerful, and, anyway, “five” has already been snatched away by the (admittedly now moribund) Zope Five project.

Indices and tables

- Index
- Search Page

Package contents

six.PY2
A boolean indicating if the code is running on Python 2.

six.PY3
A boolean indicating if the code is running on Python 3.

Constants

Six provides constants that may differ between Python versions. Ones ending `_types` are mostly useful as the second argument to `isinstance` or `issubclass`.

six.class_types
Possible class types. In Python 2, this encompasses old-style and new-style classes. In Python 3, this is just new-styles.

Conda Test Drive

conda.pydata.org/docs/test-drive.html



Conda

Search docs

Get started

Intro to conda
Download conda
Installation

Test drive

Conda cheat sheet

Using conda

Building packages

Help & reference

Get involved

Docs » [Get started](#) » Test drive

[Edit on GitHub](#)

Test drive

To start the conda 30-minute test drive, you should have already followed our 2-minute [Quick install](#) guide to download, install and update Miniconda, OR have downloaded, installed and updated Anaconda or Miniconda on your own.

NOTE: After installing, be sure you have closed and then re-opened the terminal window so the changes can take effect.

Conda test drive milestones:

1. **USING CONDA.** First we will verify that you have installed Anaconda or Miniconda, and check that it is updated to the current version. 3 min.
2. **MANAGING ENVIRONMENTS.** Next we will play with environments by creating a few environments, so you can learn to move easily between the environments. We will also verify which environment you are in, and make an exact copy of an environment as a backup. 10 min.
3. **MANAGING PYTHON.** Then we will check to see which versions of Python are available to install, install another version of Python, and switch between versions. 4 min.
4. **MANAGING PACKAGES.** We play with packages. We will a) list packages installed on your computer, b) see a list of available packages, and c) install and remove some packages using conda install. For packages not available using conda install, we will d) search on Anaconda.org. For packages that are in neither location, we'll e) install a package with the pip package manager. We will also install a free 30 day trial of Continuum's commercial package IOPro. 10 min.
5. **REMOVING PACKAGES, ENVIRONMENTS, OR CONDA.** We'll end the test drive by removing

<http://conda.pydata.org/docs/test-drive.html>

Managing Conda and Anaconda

Managing conda and anaconda

conda info

Verify conda is installed, check version #

conda update conda

Update conda package and environment manager to current version

conda update anaconda

Update the anaconda meta package (the library of packages ready to install with **conda** command)

Managing environments

Managing environments

conda info --envs or **conda info -e** Get a list of all my environments, active environment shown with *

conda create --name snowflakes biopython Create an environment and install program(s)

or

conda create -n snowflakes biopython

TIP: To avoid dependency conflicts, install all programs in the environment (snowflakes) at the same time.

*TIP: Environments install by default into the envs directory in your conda directory. You can specify a different path; see **conda create --help** for details.*

source activate snowflakes (Linux, Mac)

activate snowflakes (Windows)

Activate the new environment to use it

TIP: Activate prepends the path to the snowflakes environment.

conda create -n bunnies python=3.4 astroid Create a new environment, specify Python version

conda create -n flowers --clone snowflakes Make exact copy of an environment

conda remove -n flowers --all Delete an environment

conda env export > puppies.yml

Save current environment to a file

conda env create -f puppies.yml

Load environment from a file

Managing Python

Managing Python

```
conda search --full-name python  
or  
conda search -f python
```

Check versions of Python available to install

```
conda create -n snakes python=3.4
```

Install different version of Python in new environment

```
source activate snakes (Linux, Mac)  
activate snakes (Windows)
```

Switch to the new environment that has a different version of Python

TIP: *Activate prepends the path to the snakes environment.*

Managing Packages in Python

Managing packages, including Python

<code>conda list</code>	View list of packages and versions installed in active environment
<code>conda search beautiful-soup</code>	Search for a package to see if it is available to conda install
<code>conda install -n bunnies beautiful-soup</code>	Install a new package <i>NOTE: If you do not include the name of the new environment (-n bunnies) it will install in the current active environment.</i> <i>TIP: To view list of all packages available through conda install, visit http://docs.continuum.io/anaconda/pkg-docs.html.</i>
<code>conda update beautiful-soup</code>	Update a package in the current environment
<code>conda search --override-channels -c pandas bottleneck</code>	Search for a package in a specific location (i.e. the pandas channel on Anaconda.org) <i>NOTE: Or go to Anaconda.org in the browser and search by package name. This will show the specific channel (owner) through which it is available.</i>
<code>conda install -c pandas bottleneck</code>	Install a package from a specific channel
<code>conda search --override-channels -c defaults beautiful-soup</code>	Search for a package to see if it is available from the Anaconda repository
<code>source activate bunnies</code> (Linux, Mac) <code>activate bunnies</code> (Windows) <code>pip install see</code>	Activate the environment where you want to install a package and install it with pip (included with Anaconda and Miniconda)
<code>conda install iopro accelerate</code>	Install commercial Continuum packages
<code>conda skeleton pypi pyinstrument</code> <code>conda build pyinstrument</code>	Build a Conda package from a Python Package Index (PyPI) Package



TensorFlow

Version:

Introduction

Recommended Next Steps

Download and Setup

Requirements

Overview

Pip Installation

Virtualenv installation

Anaconda environment installation

Docker installation

Test the TensorFlow installation

(Optional, Linux) Enable GPU Support

Run TensorFlow from the Command Line

Run a TensorFlow demo model

Installing from sources

Clone the TensorFlow repository

Installation for Linux

Download and Setup

You can install TensorFlow either from our provided binary packages or from the github source.

Requirements


The TensorFlow Python API supports Python 2.7 and Python 3.3+.

The GPU version (Linux only) works best with Cuda Toolkit 7.5 and cuDNN v4. other versions are supported (Cuda toolkit >= 7.0 and cuDNN 6.5(v2), 7.0(v3), v5) only when installing from sources. Please see [Cuda installation](#) for details.

Overview

We support different ways to install TensorFlow:

- **Pip install:** Install TensorFlow on your machine, possibly upgrading previously installed Python packages. May impact existing Python programs on your machine.
- **Virtualenv install:** Install TensorFlow in its own directory, not impacting any existing Python programs on your machine.
- **Anaconda install:** Install TensorFlow in its own environment for those running the Anaconda Python distribution. Does not impact existing Python programs on your machine.

Version: 

Introduction

Recommended Next Steps

Download and Setup

Requirements

Overview

Pip Installation

Virtualenv installation

Anaconda environment installation

Docker installation

Test the TensorFlow installation

(Optional, Linux) Enable GPU Support

Run TensorFlow from the Command Line

Run a TensorFlow demo model

Installing from sources

Clone the TensorFlow repository

Installation for Linux

Follow the instructions on the [Anaconda download site](#)

Create a conda environment called `tensorflow`:

```
# Python 2.7
$ conda create -n tensorflow python=2.7

# Python 3.5
$ conda create -n tensorflow python=3.5
```

Activate the environment and use pip to install TensorFlow inside it. Use the `--ignore-installed` flag to prevent errors about `easy_install`.

```
$ source activate tensorflow
(tensorflow)$ # Your prompt should change

# Ubuntu/Linux 64-bit, CPU only:
(tensorflow)$ pip install --ignore-installed --upgrade https://storage.googleapis.com/tensorflow/tensorflow-1.0.1-cp27-none-linux_x86_64.whl

# Ubuntu/Linux 64-bit, GPU enabled. Requires CUDA toolkit 7.5 and CuDNN v4. For
# other versions, see "Install from sources" below.
(tensorflow)$ pip install --ignore-installed --upgrade https://storage.googleapis.com/tensorflow/tensorflow-1.0.1-cp27-cuda75-linux_x86_64.whl

# Mac OS X, CPU only:
(tensorflow)$ pip install --ignore-installed --upgrade https://storage.googleapis.com/tensorflow/tensorflow-1.0.1-cp27-none-macosx_x86_64.whl
```

conda create -n tensorflow python=2.7

```
imyday — -bash — 80x30
[iMydaytekiMacBook-Pro:~ imyday$ conda create -n tensorflow python=2.7
Using Anaconda Cloud api site https://api.anaconda.org
Fetching package metadata: ....
Solving package specifications: .....

Package plan for installation in environment //anaconda/envs/tensorflow:

The following packages will be downloaded:
```

package	build	
openssl-1.0.2h	0	3.0 MB
setuptools-20.7.0	py27_0	453 KB
pip-8.1.1	py27_1	1.5 MB
Total:		5.0 MB

```
The following NEW packages will be INSTALLED:

openssl:    1.0.2h-0
pip:        8.1.1-py27_1
python:     2.7.11-0
readline:   6.2-2
setuptools: 20.7.0-py27_0
sqlite:     3.9.2-0
tk:         8.5.18-0
wheel:      0.29.0-py27_0
zlib:       1.2.8-0

Proceed ([y]/n)? y
```

source activate tensorflow

```
imyday — -bash — 80x20
Proceed ([y]/n)? y

Fetching packages ...
openssl-1.0.2h 100% |#####| Time: 0:00:04 778.75 kB/s
setuptools-20. 100% |#####| Time: 0:00:05 79.42 kB/s
pip-8.1.1-py27 100% |#####| Time: 0:00:07 216.54 kB/s
Extracting packages ...
[ COMPLETE ]|#####| 100%
Linking packages ...
[ COMPLETE ]|#####| 100%
#
# To activate this environment, use:
# $ source activate tensorflow
#
# To deactivate this environment, use:
# $ source deactivate
#
[iMydaytekiMacBook-Pro:~ imyday$ source activate tensorflow]
discarding //anaconda/bin from PATH
prepending //anaconda/envs/tensorflow/bin to PATH
```

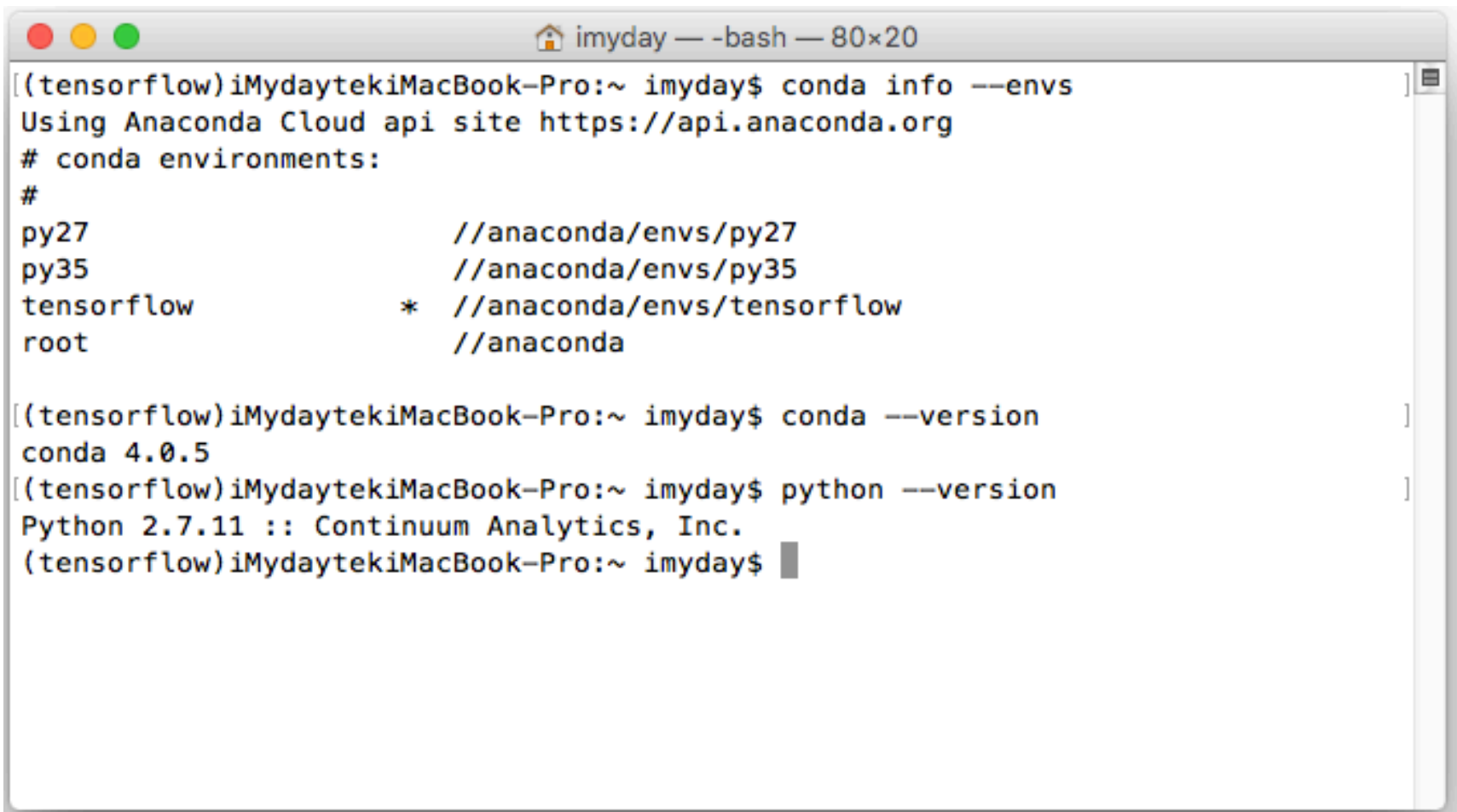
```
pip install --ignore-installed --upgrade  
https://storage.googleapis.com/tensorflow/  
mac/tensorflow-0.8.0-py2-none-any.whl
```



```
(tensorflow) iMydaytekiMacBook-Pro:~ imyday$ pip install --ignore-installed --upgrade https://storage.googleapis.com/tensorflow/mac/tensorflow-0.8.0-py2-none-any.whl  
Collecting tensorflow==0.8.0 from https://storage.googleapis.com/tensorflow/mac/tensorflow-0.8.0-py2-none-any.whl  
  Downloading https://storage.googleapis.com/tensorflow/mac/tensorflow-0.8.0-py2-none-any.whl (19.3MB)  
    100% |████████████████████████████████████████| 19.3MB 32kB/s  
Collecting six>=1.10.0 (from tensorflow==0.8.0)  
  Downloading six-1.10.0-py2.py3-none-any.whl  
Collecting protobuf==3.0.0b2 (from tensorflow==0.8.0)  
  Downloading protobuf-3.0.0b2-py2.py3-none-any.whl (326kB)  
    100% |████████████████████████████████████████| 327kB 482kB/s  
Collecting wheel (from tensorflow==0.8.0)  
  Downloading wheel-0.29.0-py2.py3-none-any.whl (66kB)  
    100% |████████████████████████████████████████| 71kB 1.3MB/s  
Collecting numpy>=1.10.1 (from tensorflow==0.8.0)  
  Downloading numpy-1.11.0-cp27-cp27m-macosx_10_6_intel.macosx_10_9_intel.macosx_10_9_x86_64.macosx_10_10_intel.macosx_10_10_x86_64.whl (3.9MB)  
    100% |████████████████████████████████████████| 3.9MB 159kB/s  
Collecting setuptools (from protobuf==3.0.0b2->tensorflow==0.8.0)  
  Downloading setuptools-21.0.0-py2.py3-none-any.whl (509kB)  
    100% |████████████████████████████████████████| 512kB 682kB/s  
Installing collected packages: six, setuptools, protobuf, wheel, numpy, tensorflow  
Successfully installed numpy-1.11.0 protobuf-3.0.0b2 setuptools-20.7.0 six-1.10.0 tensorflow-0.8.0 wheel-0.29.0  
(tensorflow) iMydaytekiMacBook-Pro:~ imyday$
```



```
conda info --envs
conda --version
python --version
```



```
imyday — -bash — 80x20
[(tensorflow)iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py27                //anaconda/envs/py27
py35                //anaconda/envs/py35
tensorflow          * //anaconda/envs/tensorflow
root                //anaconda

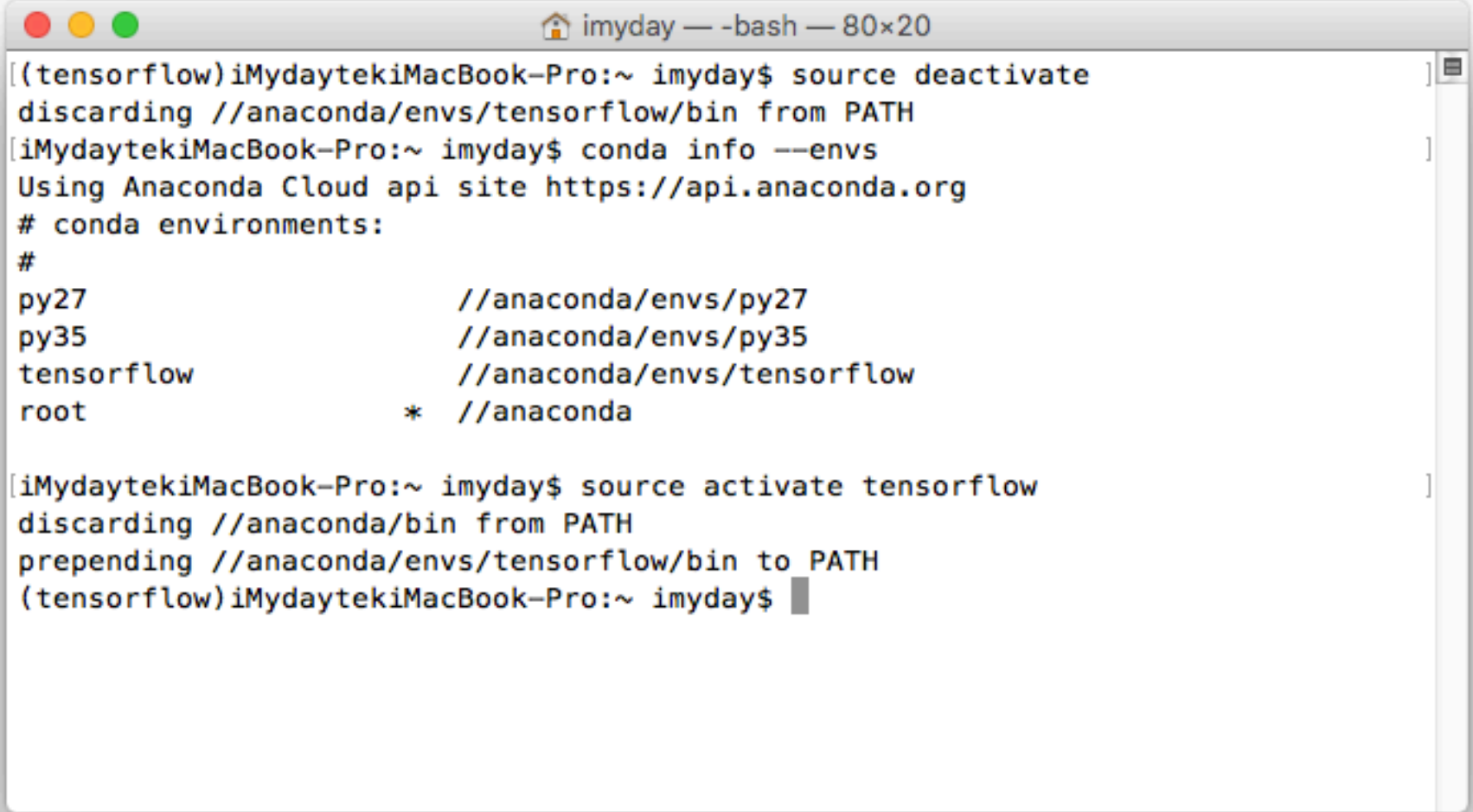
[(tensorflow)iMydaytekiMacBook-Pro:~ imyday$ conda --version
conda 4.0.5

[(tensorflow)iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Continuum Analytics, Inc.
(tensorflow)iMydaytekiMacBook-Pro:~ imyday$
```

source deactivate

```
conda info --envs
```

source activate tensorflow



```
imyday — -bash — 80x20
[(tensorflow) iMydaytekiMacBook-Pro:~ imyday$ source deactivate
discarding //anaconda/envs/tensorflow/bin from PATH
[iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py27          //anaconda/envs/py27
py35          //anaconda/envs/py35
tensorflow    //anaconda/envs/tensorflow
root          * //anaconda

[iMydaytekiMacBook-Pro:~ imyday$ source activate tensorflow
discarding //anaconda/bin from PATH
prepending //anaconda/envs/tensorflow/bin to PATH
(tensorflow) iMydaytekiMacBook-Pro:~ imyday$
```

```
$ python
```

```
...
```

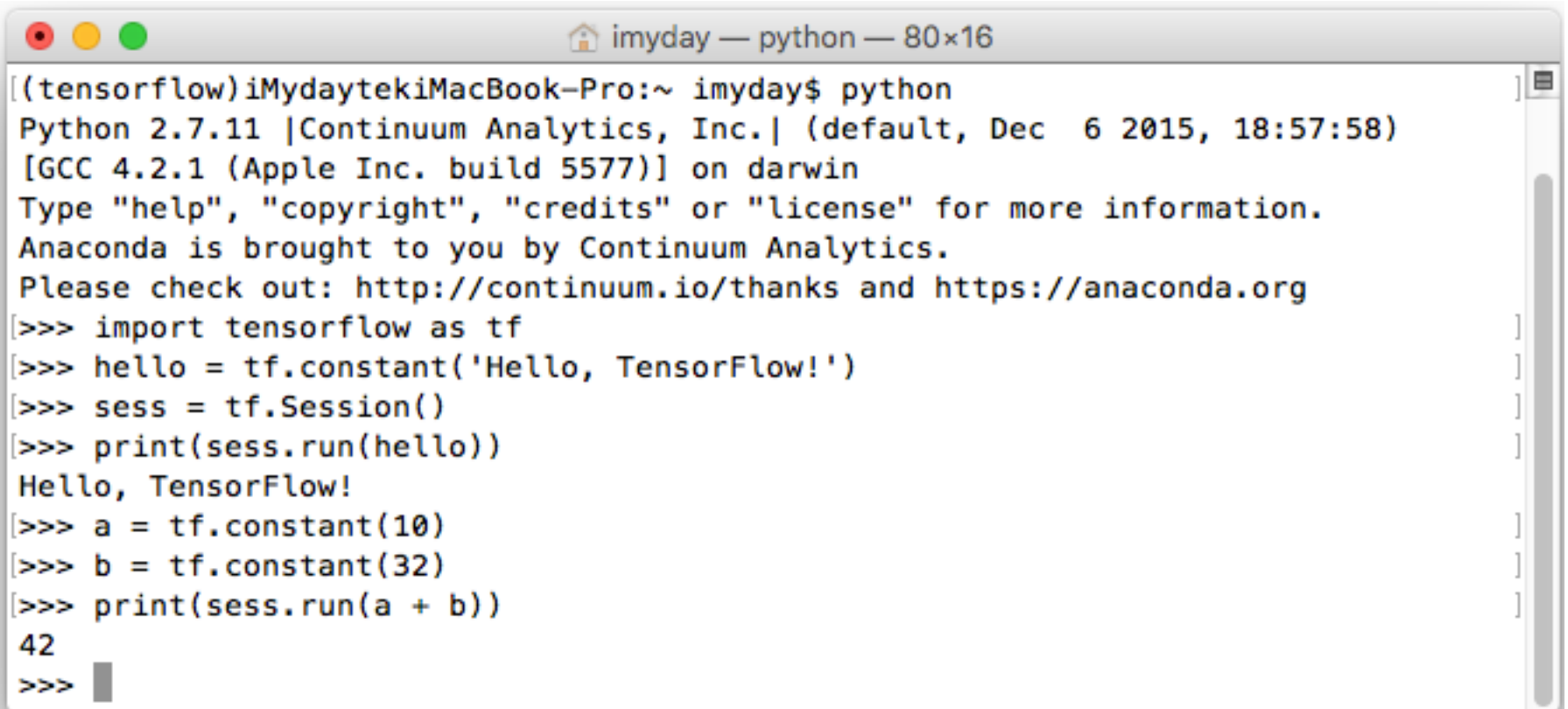
```
>>> import tensorflow as tf
```

```
>>> hello = tf.constant('Hello, TensorFlow!')
```

```
>>> sess = tf.Session()
```

```
>>> print(sess.run(hello))
```

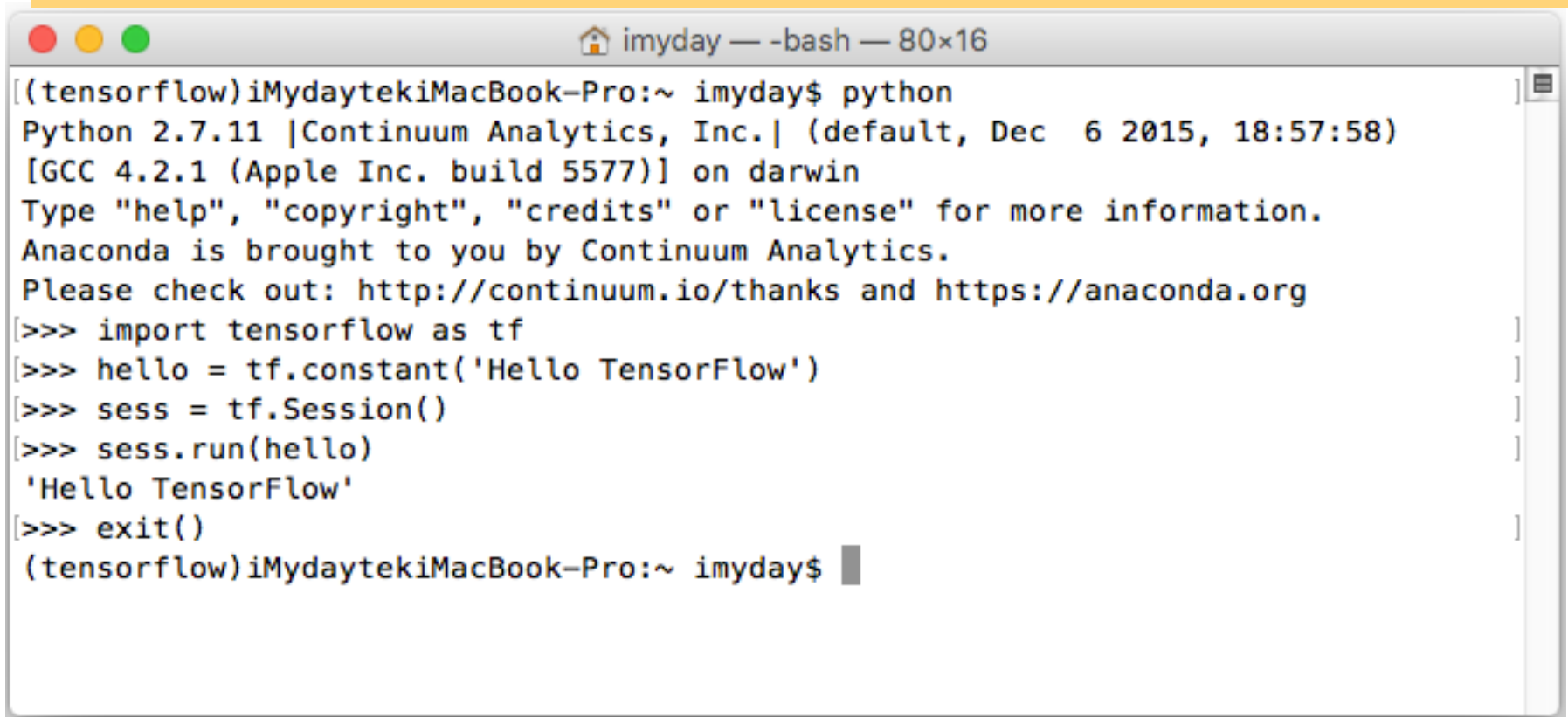
```
Hello, TensorFlow!
```



The screenshot shows a terminal window titled "imyday — python — 80x16". The terminal output is as follows:

```
(tensorflow)iMydaytekiMacBook-Pro:~ imyday$ python
Python 2.7.11 |Continuum Analytics, Inc.| (default, Dec 6 2015, 18:57:58)
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
[>>> import tensorflow as tf
[>>> hello = tf.constant('Hello, TensorFlow!')
[>>> sess = tf.Session()
[>>> print(sess.run(hello))
Hello, TensorFlow!
[>>> a = tf.constant(10)
[>>> b = tf.constant(32)
[>>> print(sess.run(a + b))
42
>>>
```

```
$ python
>>> import tensorflow as tf
>>> hello = tf.constant('Hello TensorFlow')
>>> sess = tf.Session()
>>> sess.run(hello)
'Hello TensorFlow'
>>> exit()
$
```



The screenshot shows a terminal window titled "imyday — -bash — 80x16". The prompt is "(tensorflow)iMydaytekiMacBook-Pro:~ imyday\$". The user enters "python", which starts the Python interpreter. The interpreter displays version information: "Python 2.7.11 |Continuum Analytics, Inc.| (default, Dec 6 2015, 18:57:58) [GCC 4.2.1 (Apple Inc. build 5577)] on darwin". It then shows the user's code being executed: "import tensorflow as tf", "hello = tf.constant('Hello TensorFlow')", "sess = tf.Session()", "sess.run(hello)", and "exit()". The output of the script is "Hello TensorFlow". The terminal prompt returns to "(tensorflow)iMydaytekiMacBook-Pro:~ imyday\$".

```
(tensorflow)iMydaytekiMacBook-Pro:~ imyday$ python
Python 2.7.11 |Continuum Analytics, Inc.| (default, Dec 6 2015, 18:57:58)
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
[>>> import tensorflow as tf
[>>> hello = tf.constant('Hello TensorFlow')
[>>> sess = tf.Session()
[>>> sess.run(hello)
'Hello TensorFlow'
[>>> exit()
(tensorflow)iMydaytekiMacBook-Pro:~ imyday$
```

conda list

```
imyday — -bash — 80x17
[(tensorflow) iMydaytekiMacBook-Pro:~ imyday$ conda list
# packages in environment at //anaconda/envs/tensorflow:
#
numpy                1.11.0                <pip>
openssl              1.0.2h                0
pip                  8.1.1                 py27_1
protobuf             3.0.0b2               <pip>
python               2.7.11                0
readline             6.2                   2
setuptools           20.7.0                py27_0
six                  1.10.0                <pip>
sqlite               3.9.2                 0
tensorflow            0.8.0                 <pip>
tk                   8.5.18                0
wheel                0.29.0                py27_0
zlib                  1.2.8                 0
(tensorflow) iMydaytekiMacBook-Pro:~ imyday$
```

pip install ipython

```
imyday — -bash — 80x20
[(tensorflow)iMydaytekiMacBook-Pro:~ imyday$ pip install ipython
Collecting ipython
  Downloading ipython-4.2.0-py2-none-any.whl (736kB)
    100% |████████████████████████████████████████| 737kB 191kB/s
Collecting traitlets (from ipython)
  Downloading traitlets-4.2.1-py2.py3-none-any.whl (67kB)
    100% |████████████████████████████████████████| 71kB 315kB/s
Collecting pickleshare (from ipython)
  Downloading pickleshare-0.7.2-py2.py3-none-any.whl
Collecting simplegeneric>0.8 (from ipython)
  Downloading simplegeneric-0.8.1.zip
Collecting backports.shutil-get-terminal-size (from ipython)
  Downloading backports.shutil_get_terminal_size-1.0.0-py2.py3-none-any.whl
Collecting decorator (from ipython)
  Downloading decorator-4.0.9-py2.py3-none-any.whl
Requirement already satisfied (use --upgrade to upgrade): setuptools>=18.5 in /anaconda/envs/tensorflow/lib/python2.7/site-packages (from ipython)
Collecting gnureadline (from ipython)
  Downloading gnureadline-6.3.3-cp27-none-macosx_10_9_x86_64.whl (132kB)
    100% |████████████████████████████████████████| 133kB 155kB/s
```

conda list

```
imyday — -bash — 80x29
[(tensorflow)iMydaytekiMacBook-Pro:~ imyday$ conda list
# packages in environment at //anaconda/envs/tensorflow:
#
appnope                0.1.0                <pip>
backports.shutil-get-terminal-size 1.0.0                <pip>
decorator              4.0.9                <pip>
gnureadline           6.3.3                <pip>
ipython               4.2.0                <pip>
ipython-genutils     0.1.0                <pip>
numpy                 1.11.0               <pip>
openssl               1.0.2h                0
pathlib2              2.1.0                <pip>
pexpect               4.0.1                <pip>
pickleshare          0.7.2                <pip>
pip                   8.1.1                py27_1
protobuf              3.0.0b2              <pip>
ptyprocess            0.5.1                <pip>
python                2.7.11                0
readline              6.2                  2
setuptools            20.7.0               py27_0
simplegeneric          0.8.1                <pip>
six                   1.10.0               <pip>
sqlite                3.9.2                0
tensorflow            0.8.0                <pip>
tk                    8.5.18               0
traitlets             4.2.1                <pip>
wheel                 0.29.0               py27_0
zlib                  1.2.8                0
(tensorflow)iMydaytekiMacBook-Pro:~ imyday$
```


conda list

```
imyday — ipython notebook ▶ python — 80×25
[(tensorflow)iMydaytekiMacBook-Pro:~ imyday$ conda list
# packages in environment at //anaconda/envs/tensorflow:
#
alabaster                0.7.7                <pip>
appnope                  0.1.0                <pip>
babel                    2.3.4                <pip>
backports-abc            0.4                  <pip>
backports.shutil-get-terminal-size 1.0.0                <pip>
backports.ssl-match-hostname 3.5.0.1              <pip>
certifi                  2016.2.28            <pip>
configparser             3.3.0.post2          <pip>
decorator                4.0.9                <pip>
docutils                 0.12                 <pip>
entrypoints              0.2.1                <pip>
functools32              3.2.3.post2          <pip>
futures                  3.0.5                <pip>
gnureadline              6.3.3                <pip>
imagesize                0.7.1                <pip>
ipykernel                 4.3.1                <pip>
ipyparallel              5.0.1                <pip>
ipython                  4.2.0                <pip>
ipython-genutils         0.1.0                <pip>
ipywidgets               5.1.3                <pip>
jinja2                   2.8                  <pip>
jsonschema               2.5.1                <pip>
```

conda list

```
imyday — ipython notebook ▶ python — 80×25
ipykernel                4.3.1                <pip>
ipyparallel              5.0.1                <pip>
ipython                  4.2.0                <pip>
ipython-genutils        0.1.0                <pip>
ipywidgets               5.1.3                <pip>
jinja2                   2.8                  <pip>
jsonschema               2.5.1                <pip>
jupyter-client           4.2.2                <pip>
jupyter-core             4.1.0                <pip>
markupsafe               0.23                 <pip>
mistune                  0.7.2                <pip>
nbconvert                 4.2.0                <pip>
nbformat                 4.0.1                <pip>
nose                      1.3.7                <pip>
notebook                  4.2.0                <pip>
numpy                    1.11.0               <pip>
openssl                  1.0.2h                0
path.py                  8.2.1                <pip>
pathlib2                 2.1.0                <pip>
pexpect                  4.0.1                <pip>
pickleshare              0.7.2                <pip>
pip                       8.1.1                py27_1
protobuf                 3.0.0b2              <pip>
ptyprocess               0.5.1                <pip>
pygments                 2.1.3                <pip>
```

ipython notebook

```
ipython notebook
```

simplegeneric	0.8.1	<pip>
singledispatch	3.4.0.3	<pip>
six	1.10.0	<pip>
snowballstemmer	1.2.1	<pip>
sphinx	1.4.1	<pip>
sqlite	3.9.2	0
tensorflow	0.8.0	<pip>
terminado	0.6	<pip>
testpath	0.3	<pip>
tk	8.5.18	0
tornado	4.3	<pip>
traitlets	4.2.1	<pip>
wheel	0.29.0	py27_0
widgetsnbextension	1.2.2	<pip>
zlib	1.2.8	0

```
[(tensorflow)iMydaytekiMacBook-Pro:~ imyday$ ipython notebook ]
[TerminalIPythonApp] WARNING | Subcommand `ipython notebook` is deprecated and will be removed in future versions.
[TerminalIPythonApp] WARNING | You likely want to use `jupyter notebook` in the future
//anaconda/envs/tensorflow/lib/python2.7/site-packages/widgetsnbextension/__init__.py:30: UserWarning: To use the jupyter-js-widgets nbextension, you'll need to update
  the Jupyter notebook to version 4.2 or later.
  the Jupyter notebook to version 4.2 or later."")
```

```
import tensorflow as tf
hello = tf.constant('Hello TensorFlow')
sess = tf.Session()
print(sess.run(hello))
```

localhost:8889/notebooks/SCDBA/DeepLearningTensorFlow1.ipynb

Jupyter DeepLearningTensorFlow1 (autosaved)



File Edit View Insert Cell Kernel Help Python 2

File operation icons: Save, Add, Copy, Paste, Undo, Redo, Run, Stop, Refresh, Code, Keyboard, CellToolbar

```
In [1]: import tensorflow as tf
hello = tf.constant('Hello TensorFlow')
sess = tf.Session()
print(sess.run(hello))
```

Hello TensorFlow

```
In [2]: a = tf.constant(10)
b = tf.constant(32)
c = sess.run(a+b)
print(c)
```

42

```
a = tf.constant(10)
b = tf.constant(32)
c = sess.run(a+b)
print(c)
```

TensorFlow Example

```
import tensorflow as tf
import numpy as np

# Create 100 phony x, y data points in NumPy,  $y = x * 0.1 + 0.3$ 
x_data = np.random.rand(100).astype(np.float32)
y_data = x_data * 0.1 + 0.3

# Try to find values for W and b that compute  $y\_data = W * x\_data + b$ 
# (We know that W should be 0.1 and b 0.3, but Tensorflow will
# figure that out for us.)
W = tf.Variable(tf.random_uniform([1], -1.0, 1.0))
b = tf.Variable(tf.zeros([1]))
y = W * x_data + b

# Minimize the mean squared errors.
loss = tf.reduce_mean(tf.square(y - y_data))
optimizer = tf.train.GradientDescentOptimizer(0.5)
train = optimizer.minimize(loss)

# Before starting, initialize the variables. We will 'run' this first.
init = tf.initialize_all_variables()

# Launch the graph.
sess = tf.Session()
sess.run(init)

# Fit the line.
for step in xrange(201):
    sess.run(train)
    if step % 20 == 0:
        print(step, sess.run(W), sess.run(b))

# Learns best fit is W: [0.1], b: [0.3]
```

TensorFlow Example



```
import tensorflow as tf
import numpy as np

# Create 100 phony x, y data points in NumPy,  $y = x * 0.1 + 0.3$ 
x_data = np.random.rand(100).astype(np.float32)
y_data = x_data * 0.1 + 0.3

# Try to find values for W and b that compute  $y\_data = W * x\_data + b$ 
# (We know that W should be 0.1 and b 0.3, but Tensorflow will
# figure that out for us.)
W = tf.Variable(tf.random_uniform([1], -1.0, 1.0))
b = tf.Variable(tf.zeros([1]))
y = W * x_data + b

# Minimize the mean squared errors.
loss = tf.reduce_mean(tf.square(y - y_data))
optimizer = tf.train.GradientDescentOptimizer(0.5)
train = optimizer.minimize(loss)

# Before starting, initialize the variables. We will 'run' this first.
init = tf.initialize_all_variables()

# Launch the graph.
sess = tf.Session()
sess.run(init)

# Fit the line.
for step in xrange(201):
    sess.run(train)
    if step % 20 == 0:
        print(step, sess.run(W), sess.run(b))

# Learns best fit is W: [0.1], b: [0.3]
```

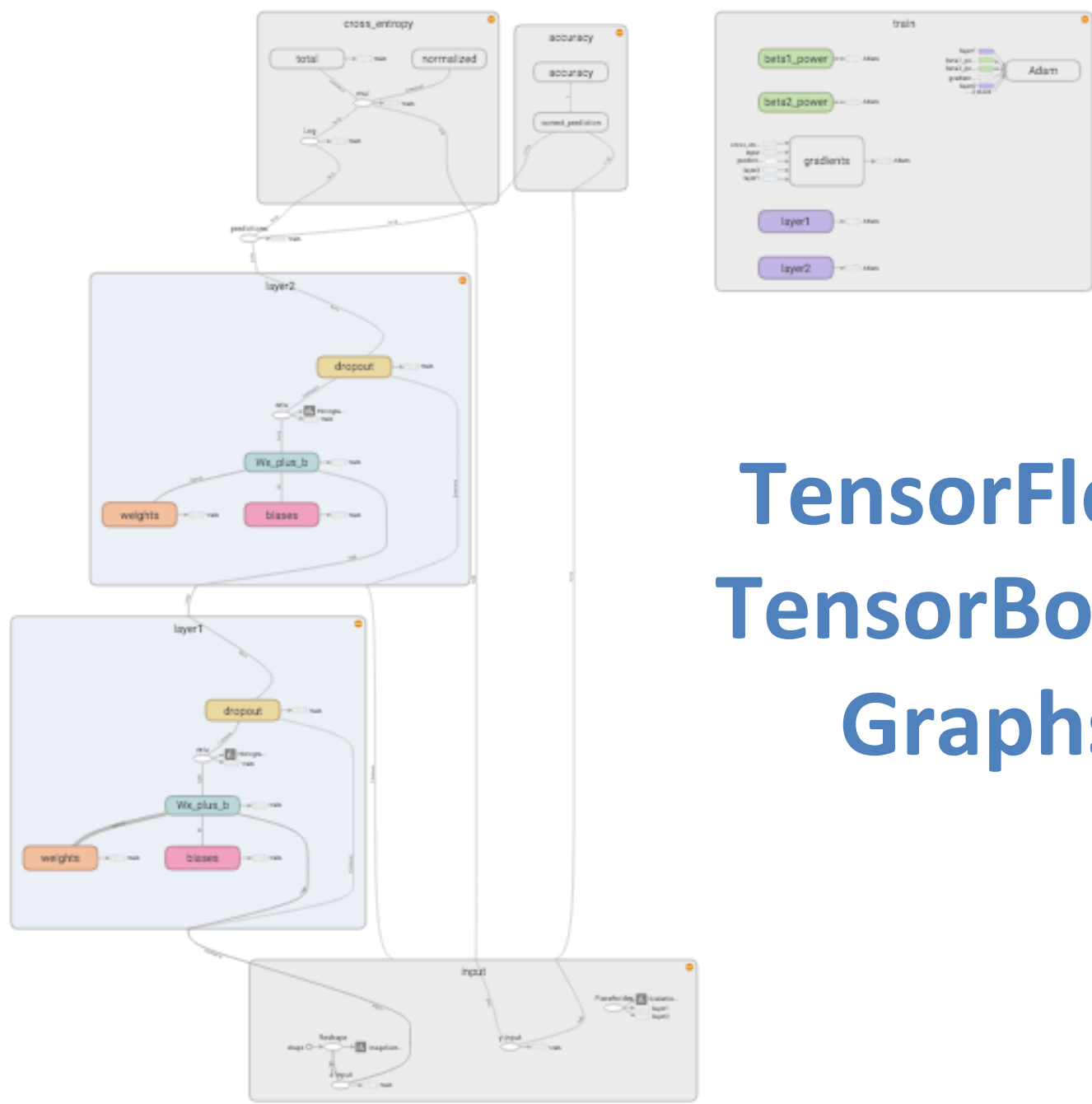
TensorFlow Example

```
# Launch the graph.
sess = tf.Session()
sess.run(init)

# Fit the line.
for step in xrange(201):
    sess.run(train)
    if step % 20 == 0:
        print(step, sess.run(W), sess.run(b))

# Learns best fit is W: [0.1], b: [0.3]
```

```
(0, array([-0.07500112], dtype=float32), array([ 0.54339123], dtype=float32))
(20, array([ 0.05152683], dtype=float32), array([ 0.32598534], dtype=float32))
(40, array([ 0.09009784], dtype=float32), array([ 0.30530834], dtype=float32))
(60, array([ 0.09797716], dtype=float32), array([ 0.3010844], dtype=float32))
(80, array([ 0.09958676], dtype=float32), array([ 0.30022153], dtype=float32))
(100, array([ 0.09991558], dtype=float32), array([ 0.30004525], dtype=float32))
(120, array([ 0.09998275], dtype=float32), array([ 0.30000925], dtype=float32))
(140, array([ 0.09999647], dtype=float32), array([ 0.30000189], dtype=float32))
(160, array([ 0.09999929], dtype=float32), array([ 0.3000004], dtype=float32))
(180, array([ 0.09999986], dtype=float32), array([ 0.3000001], dtype=float32))
(200, array([ 0.09999999], dtype=float32), array([ 0.30000007], dtype=float32))
```



TensorFlow TensorBoard Graphs

TensorFlow Example

MNIST Softmax

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

# Import data
from tensorflow.examples.tutorials.mnist import input_data

import tensorflow as tf

flags = tf.app.flags
FLAGS = flags.FLAGS
flags.DEFINE_string('data_dir', '/tmp/data/', 'Directory for storing data')

mnist = input_data.read_data_sets(FLAGS.data_dir, one_hot=True)

sess = tf.InteractiveSession()

# Create the model
x = tf.placeholder(tf.float32, [None, 784])
W = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))
y = tf.nn.softmax(tf.matmul(x, W) + b)

# Define loss and optimizer
y_ = tf.placeholder(tf.float32, [None, 10])
cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y), reduction_indices=[1]))
train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)

# Train
tf.initialize_all_variables().run()
for i in range(1000):
    batch_xs, batch_ys = mnist.train.next_batch(100)
    train_step.run({x: batch_xs, y_: batch_ys})

# Test trained model
correct_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y_, 1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
print(accuracy.eval({x: mnist.test.images, y_: mnist.test.labels}))
```

TensorFlow Example MNIST Softmax



```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

# Import data
from tensorflow.examples.tutorials.mnist import input_data

import tensorflow as tf

flags = tf.app.flags
FLAGS = flags.FLAGS
flags.DEFINE_string('data_dir', '/tmp/data/', 'Directory for storing data')

mnist = input_data.read_data_sets(FLAGS.data_dir, one_hot=True)

sess = tf.InteractiveSession()

# Create the model
x = tf.placeholder(tf.float32, [None, 784])
W = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))
y = tf.nn.softmax(tf.matmul(x, W) + b)

# Define loss and optimizer
y_ = tf.placeholder(tf.float32, [None, 10])
cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y), reduction_indices=[1]))
train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)

# Train
tf.initialize_all_variables().run()
for i in range(1000):
    batch_xs, batch_ys = mnist.train.next_batch(100)
    train_step.run({x: batch_xs, y_: batch_ys})

# Test trained model
correct_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y_, 1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
print(accuracy.eval({x: mnist.test.images, y_: mnist.test.labels}))
```

TensorFlow Example MNIST Softmax

```
flags.DEFINE_string('data_dir', '/tmp/data/', 'Directory for storing data')

mnist = input_data.read_data_sets(FLAGS.data_dir, one_hot=True)

sess = tf.InteractiveSession()

# Create the model
x = tf.placeholder(tf.float32, [None, 784])
W = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))
y = tf.nn.softmax(tf.matmul(x, W) + b)

# Define loss and optimizer
y_ = tf.placeholder(tf.float32, [None, 10])
cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y), reduction_indices=[1]))
train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)

# Train
tf.initialize_all_variables().run()
for i in range(1000):
    batch_xs, batch_ys = mnist.train.next_batch(100)
    train_step.run({x: batch_xs, y_: batch_ys})

# Test trained model
correct_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y_, 1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
print(accuracy.eval({x: mnist.test.images, y_: mnist.test.labels}))
```

```
Extracting /tmp/data/train-images-idx3-ubyte.gz
Extracting /tmp/data/train-labels-idx1-ubyte.gz
Extracting /tmp/data/t10k-images-idx3-ubyte.gz
Extracting /tmp/data/t10k-labels-idx1-ubyte.gz
0.9179
```

TensorFlow

Deep MNIST for Experts

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import gzip
import os
import tempfile

import numpy
from six.moves import urllib
from six.moves import xrange # pylint: disable=redefined-builtin
import tensorflow as tf
from tensorflow.contrib.learn.python.learn.datasets.mnist import read_data_sets

def weight_variable(shape):
    initial = tf.truncated_normal(shape, stddev=0.1)
    return tf.Variable(initial)

def bias_variable(shape):
    initial = tf.constant(0.1, shape=shape)
    return tf.Variable(initial)

def conv2d(x, W):
    return tf.nn.conv2d(x, W, strides=[1, 1, 1, 1], padding='SAME')

def max_pool_2x2(x):
    return tf.nn.max_pool(x, ksize=[1, 2, 2, 1],
                          strides=[1, 2, 2, 1], padding='SAME')

W_conv1 = weight_variable([5, 5, 1, 32])
b_conv1 = bias_variable([32])

x_image = tf.reshape(x, [-1,28,28,1])

h_conv1 = tf.nn.relu(conv2d(x_image, W_conv1) + b_conv1)
h_pool1 = max_pool_2x2(h_conv1)

W_conv2 = weight_variable([5, 5, 32, 64])
b_conv2 = bias_variable([64])

h_conv2 = tf.nn.relu(conv2d(h_pool1, W_conv2) + b_conv2)
h_pool2 = max_pool_2x2(h_conv2)

W_fc1 = weight_variable([7 * 7 * 64, 1024])
b_fc1 = bias_variable([1024])

h_pool2_flat = tf.reshape(h_pool2, [-1, 7*7*64])
h_fc1 = tf.nn.relu(tf.matmul(h_pool2_flat, W_fc1) + b_fc1)

keep_prob = tf.placeholder(tf.float32)
h_fc1_drop = tf.nn.dropout(h_fc1, keep_prob)

W_fc2 = weight_variable([1024, 10])
b_fc2 = bias_variable([10])

y_conv=tf.nn.softmax(tf.matmul(h_fc1_drop, W_fc2) + b_fc2)

cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y_conv), reduction_indices=[1]))
train_step = tf.train.AdamOptimizer(1e-4).minimize(cross_entropy)
correct_prediction = tf.equal(tf.argmax(y_conv,1), tf.argmax(y_,1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
sess.run(tf.initialize_all_variables())
for i in range(20000):
    batch = mnist.train.next_batch(50)
    if i%100 == 0:
        train_accuracy = accuracy.eval(feed_dict={
            x:batch[0], y_: batch[1], keep_prob: 1.0})
        print("step %d, training accuracy %g"%(i, train_accuracy))
        train_step.run(feed_dict={x: batch[0], y_: batch[1], keep_prob: 0.5})

print("test accuracy %g"%accuracy.eval(feed_dict={
    x: mnist.test.images, y_: mnist.test.labels, keep_prob: 1.0}))
```

TensorFlow Deep MNIST for Experts



```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import gzip
import os
import tempfile

import numpy
from six.moves import urllib
from six.moves import xrange # pylint: disable=redefined-builtin
import tensorflow as tf
from tensorflow.contrib.learn.python.learn.datasets.mnist import read_data_sets

def weight_variable(shape):
    initial = tf.truncated_normal(shape, stddev=0.1)
    return tf.Variable(initial)

def bias_variable(shape):
    initial = tf.constant(0.1, shape=shape)
    return tf.Variable(initial)

def conv2d(x, W):
    return tf.nn.conv2d(x, W, strides=[1, 1, 1, 1], padding='SAME')

def max_pool_2x2(x):
    return tf.nn.max_pool(x, ksize=[1, 2, 2, 1],
                          strides=[1, 2, 2, 1], padding='SAME')
```

TensorFlow Deep MNIST for Experts



```
W_conv1 = weight_variable([5, 5, 1, 32])
b_conv1 = bias_variable([32])

x_image = tf.reshape(x, [-1,28,28,1])

h_conv1 = tf.nn.relu(conv2d(x_image, W_conv1) + b_conv1)
h_pool1 = max_pool_2x2(h_conv1)

W_conv2 = weight_variable([5, 5, 32, 64])
b_conv2 = bias_variable([64])

h_conv2 = tf.nn.relu(conv2d(h_pool1, W_conv2) + b_conv2)
h_pool2 = max_pool_2x2(h_conv2)

W_fc1 = weight_variable([7 * 7 * 64, 1024])
b_fc1 = bias_variable([1024])

h_pool2_flat = tf.reshape(h_pool2, [-1, 7*7*64])
h_fc1 = tf.nn.relu(tf.matmul(h_pool2_flat, W_fc1) + b_fc1)

keep_prob = tf.placeholder(tf.float32)
h_fc1_drop = tf.nn.dropout(h_fc1, keep_prob)

W_fc2 = weight_variable([1024, 10])
b_fc2 = bias_variable([10])

y_conv=tf.nn.softmax(tf.matmul(h_fc1_drop, W_fc2) + b_fc2)
```

TensorFlow Deep MNIST for Experts



```
cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y_conv), reduction_indices=[1]))
train_step = tf.train.AdamOptimizer(1e-4).minimize(cross_entropy)
correct_prediction = tf.equal(tf.argmax(y_conv,1), tf.argmax(y_,1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
sess.run(tf.initialize_all_variables())
for i in range(20000):
    batch = mnist.train.next_batch(50)
    if i%100 == 0:
        train_accuracy = accuracy.eval(feed_dict={
            x:batch[0], y_: batch[1], keep_prob: 1.0})
        print("step %d, training accuracy %g"%(i, train_accuracy))
        train_step.run(feed_dict={x: batch[0], y_: batch[1], keep_prob: 0.5})

print("test accuracy %g"%accuracy.eval(feed_dict={
    x: mnist.test.images, y_: mnist.test.labels, keep_prob: 1.0}))
```

TensorFlow Deep MNIST for Experts



```
step 0, training accuracy 0.06
step 100, training accuracy 0.7
step 200, training accuracy 0.94
step 300, training accuracy 0.94
step 400, training accuracy 0.94
step 500, training accuracy 0.92
step 600, training accuracy 0.88
step 700, training accuracy 0.94
step 800, training accuracy 0.96
step 900, training accuracy 0.94
step 1000, training accuracy 0.92
step 1100, training accuracy 0.98
step 1200, training accuracy 0.92
step 1300, training accuracy 0.96
step 1400, training accuracy 0.98
step 1500, training accuracy 0.98
step 1600, training accuracy 0.9
step 1700, training accuracy 0.94
step 1800, training accuracy 0.98
step 1900, training accuracy 0.92
step 2000, training accuracy 1
step 2100, training accuracy 0.96
step 2200, training accuracy 1
step 2300, training accuracy 1
step 2400, training accuracy 0.98
step 2500, training accuracy 0.94
step 2600, training accuracy 0.98
step 2700, training accuracy 1
step 2800, training accuracy 0.94
step 2900, training accuracy 0.98
step 3000, training accuracy 0.98
```


References

- Sunila Gollapudi (2016), Practical Machine Learning, Packt Publishing
- Sebastian Raschka (2015), Python Machine Learning, Packt Publishing
- TensorFlow: <https://www.tensorflow.org/>
- Rajat Monga (2016), TensorFlow: Machine Learning for Everyone, https://www.youtube.com/watch?v=wmw8Bbb_eIE
- Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, The 9th ACM International Conference on Web Search and Data Mining (WSDM 2016), San Francisco, California, USA., February 22-25, 2016. <http://www.wsdm-conference.org/2016/slides/WSDM2016-Jeff-Dean.pdf>
- Deep Learning Basics: Neural Networks Demystified, <https://www.youtube.com/playlist?list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU>
- Deep Learning SIMPLIFIED, <https://www.youtube.com/playlist?list=PLjJh1vISEYgvGod9wWiydumYl8hOXixNu>
- Theano: <http://deeplearning.net/software/theano/>
- Keras: <http://keras.io/>