# Thinking About Artificial Intelligence by Making AI Happen

Ruben R. Puentedura, Ph.D.

The New York Times

BUSINESS DAY

# *Google's AlphaGo Defeats Chinese Go Master in Win for A.I.*

点击查看本文中文版

By PAUL MOZUR MAY 23, 2017


Ke Jie, the world's top Go player, reacting during his match on Tuesday against AlphaGo, artificial intelligence software developed by a Google affiliate. China Stringer Network, via Reuters

RELATED COVERAGE


A.I. Is Doing Legal Work. But It Won't Replace Lawyers, Yet. MARCH 19, 2017


China's Intelligent Weaponry Gets Smarter FEB. 3, 2017


THE FUTURE OF WORK
The Future of Not Working FEB. 23, 2017


Master of Go Board Game Is Walloped by Google Computer Program MARCH 9, 2016

DeepMind

# AlphaZero AI beats champion chess program after teaching itself in four hours

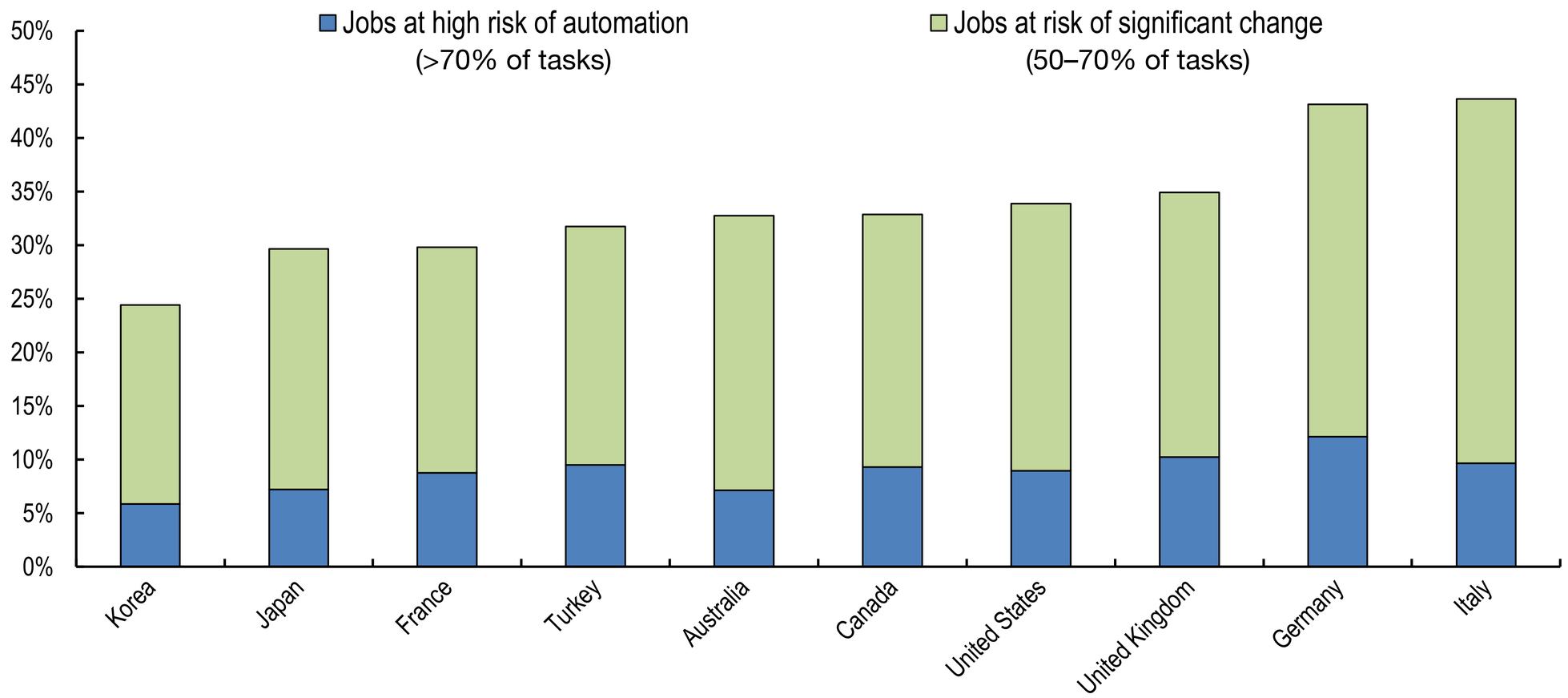
**Google's artificial intelligence sibling DeepMind repurposes Go-playing AI to conquer chess and shogi without aid of human knowledge**

▲ AlphaZero's victory is just the latest in a series of computer triumphs over human players since Computer programs have been able to beat the best IBM's Deep Blue defeated Garry Kasparov in 1997. Photograph: 18percentgrey / Alamy/Alamy

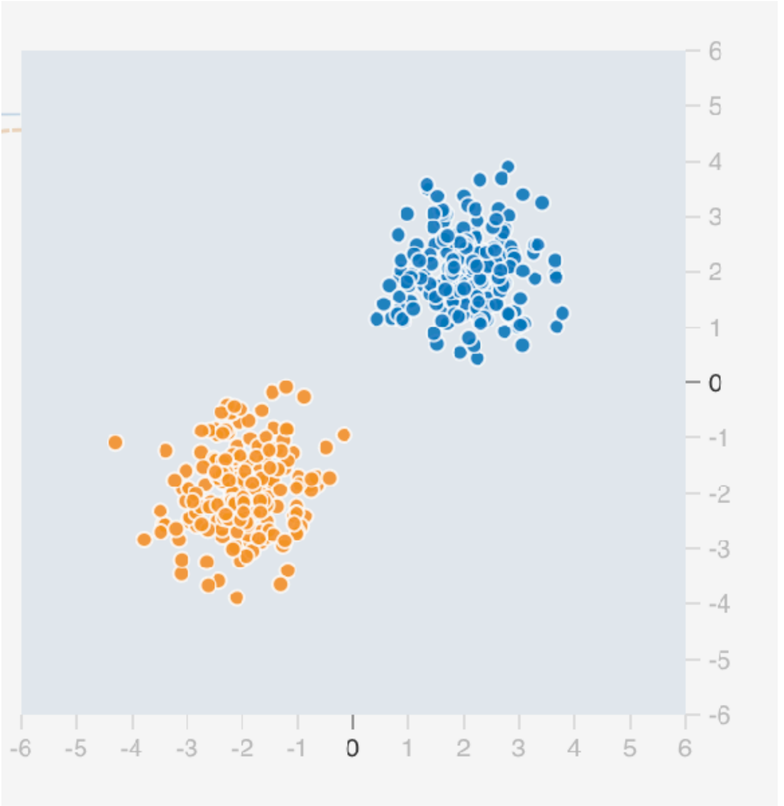
Samuel Gibbs

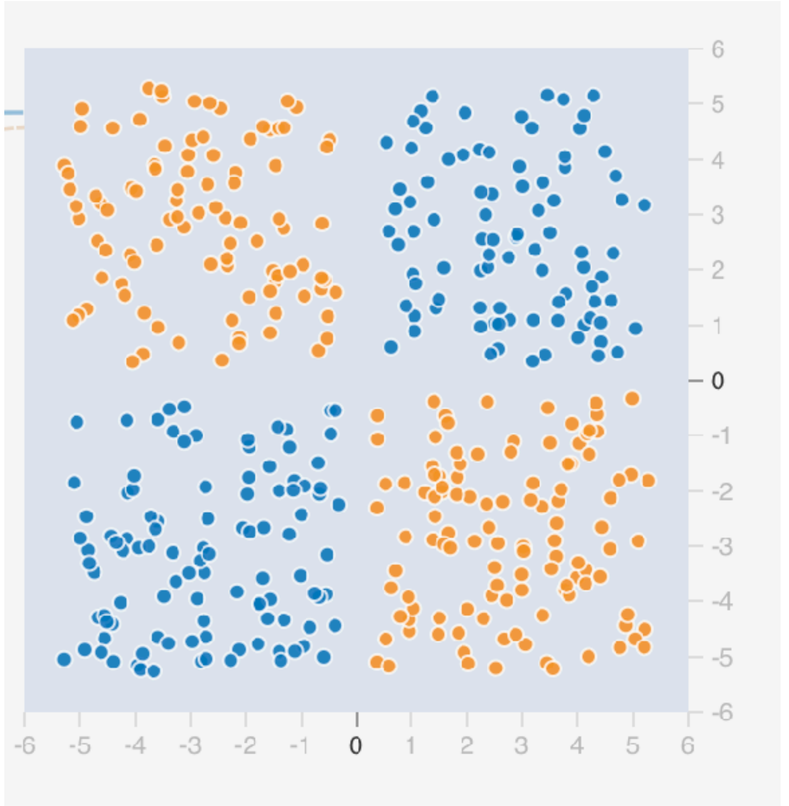
Thu 7 Dec 2017 07.41 EST

# Advanced G20 Countries: Jobs at High Risk of Automation

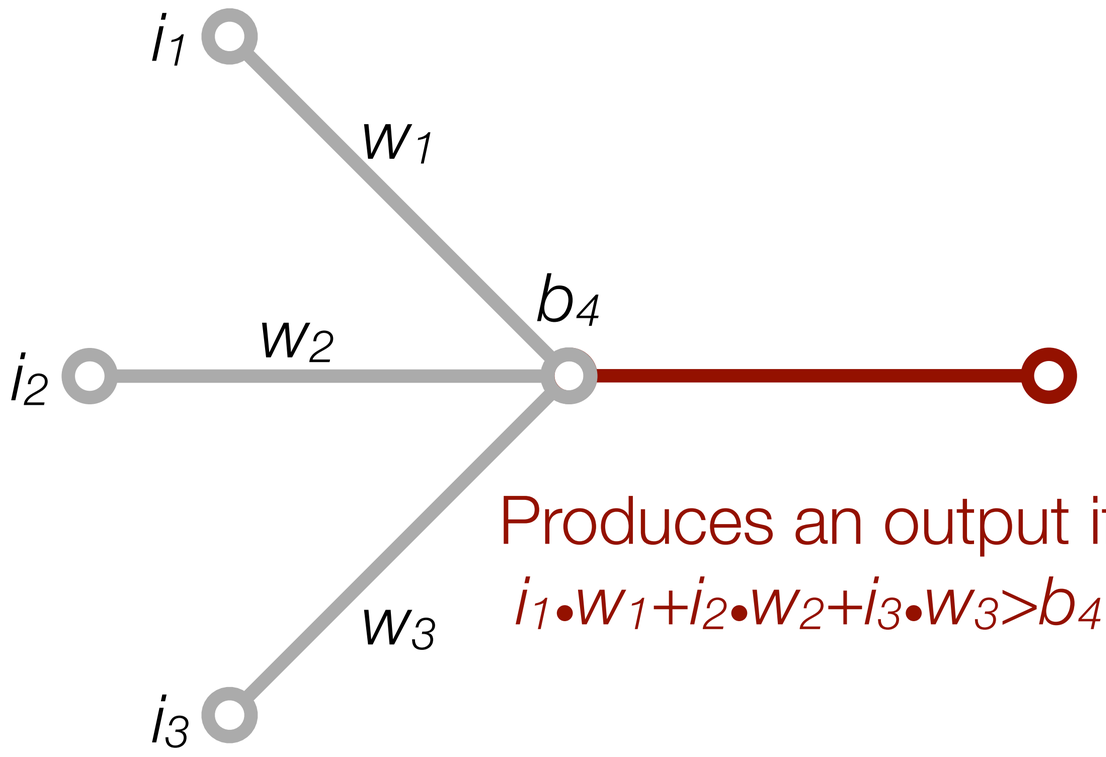



Organisation for Economic Co-operation and Development (OECD), *Future of Work and Skills*. Paper presented at the 2nd Meeting of the G20 Employment Working Group, Hamburg, Germany. (2017)


6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6


6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6

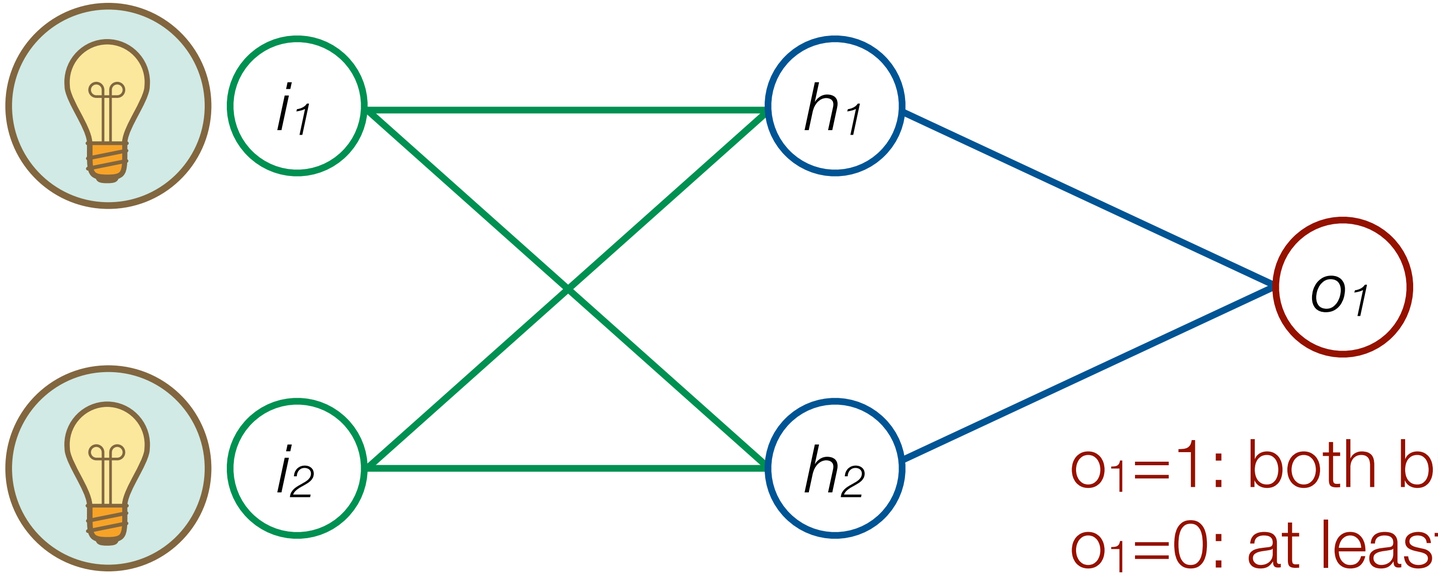
# Simple McCulloch–Pitts/Perceptron Model




Produces an output if

$i_1 \bullet w_1 + i_2 \bullet w_2 + i_3 \bullet w_3 > b_4$

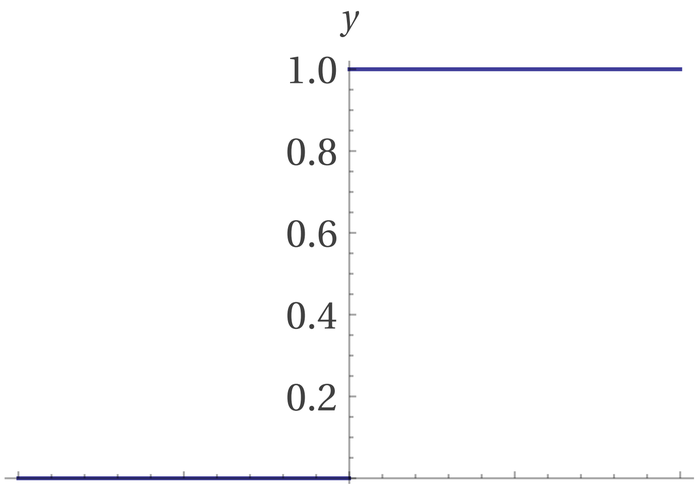
# A Simple Example: A Light Bulb Tester



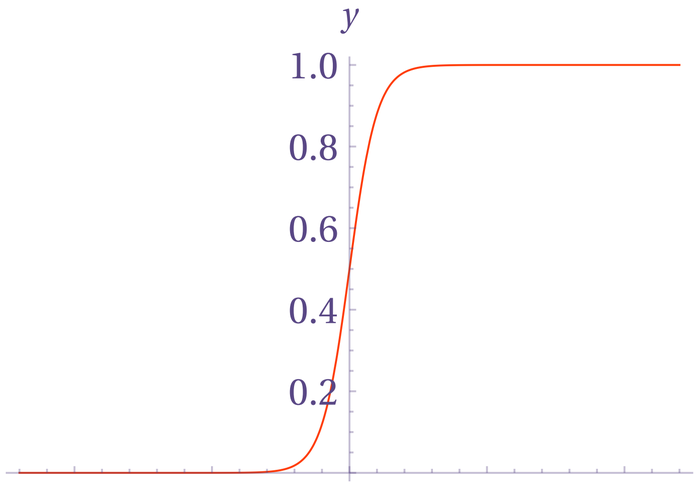

$o_1$=1: both bulbs working

$o_1$=0: at least one bulb not working

# Threshold Function




Based on step function




Based on hyperbolic tangent function

$$\frac{e^{x}-e^{-x}}{e^{-x}+e^{x}}$$

# Backpropagation

- Run the network for a given input
- Calculate the difference between the output produced by the network and the desired output - this is the *error*
- Change the weights by a certain percentage in the direction indicated by the error (i.e. increase or decrease to make the error smaller)
  - This percentage is called the *learning rate*
- Repeat for all inputs until the error is within the desired range

# Using brain.js (https://github.com/harthur/brain): http://tinyurl.com/mlti17nn01

Run Update Fork Set as base Tidy Collaborate Embed Settings rubenrp

Fiddle Meta

MLTI17NN01

No description

Add title to make the fiddle public.

External Resources

AJAX Requests

Legal, Credits and Links

JSFiddle Roadmap
suggest and vote for features

HTML

```html
<html>
<head>
<script src="https://cdn.rawgit.com/harthur/brain/gh-pages/brain-0.6.3.min.js"></script>
</head>
<body>
<script>
var net = new brain.NeuralNetwork();

var inputset = net.train([{input: [0, 0], output: [0]},
                          {input: [0, 1], output: [0]},
                          {input: [1, 0], output: [0]},
                          {input: [1, 1], output: [1]}]);

var output = net.run([1, 1]);

document.write("Output: ",output[0],
               "<br>Number of iterations: ",inputset.iterations,
               "<br>Error: ",inputset.error);

</script>
</body>
</html>
```

No need for the HTML tag, it's already in the output.

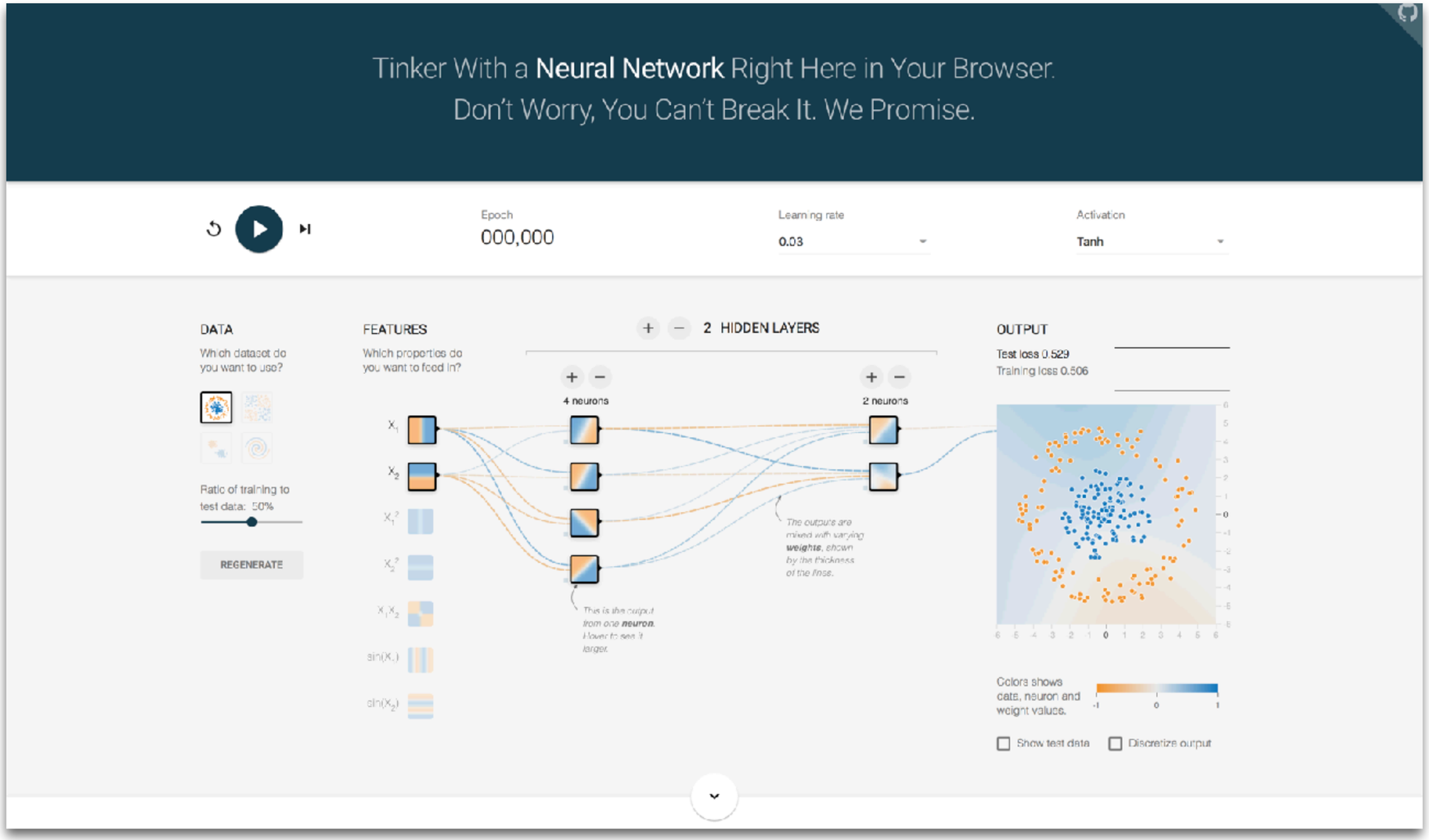
No need for the HEAD tag, it's already in the output.

JAVASCRIPT

CSS

Output: 0.8898158809618955
Number of iterations: 1221
Error: 0.004991215749680746

# A Neural Network Playground: http://tinyurl.com/mlti17nndemo



# Returning to brain.js
# http://tinyurl.com/mlti17nn02

Run Update Fork Set as base Tidy Collaborate Embed Settings rubenrp

**Fiddle Meta**

MLTI17NN02

No description

Add title to make the fiddle public.

External Resources

AJAX Requests

Legal, Credits and Links

JSFiddle Roadmap
suggest and vote for features

HTML

```html
<html>
<head>
<script src="https://cdn.rawgit.com/harthur/brain/gh-pages/brain-0.6.3.min.js"></script>
</head>
<body>
<script>
var net = new brain.NeuralNetwork({
          hiddenLayers: [3,3],
          learningRate: 0.6});

var inputset = net.train([{input: [0, 0], output: [0]},
                          {input: [0, 1], output: [0]},
                          {input: [1, 0], output: [0]},
                          {input: [1, 1], output: [1]}]);

var output = net.run([1, 1]);

document.write("Output: ",output[0],
               "<br>Number of iterations: ",inputset.iterations,
               "<br>Error: ",inputset.error);

</script>
</body>
</html>
```

No need for the **HTML** tag, it's already in the output.

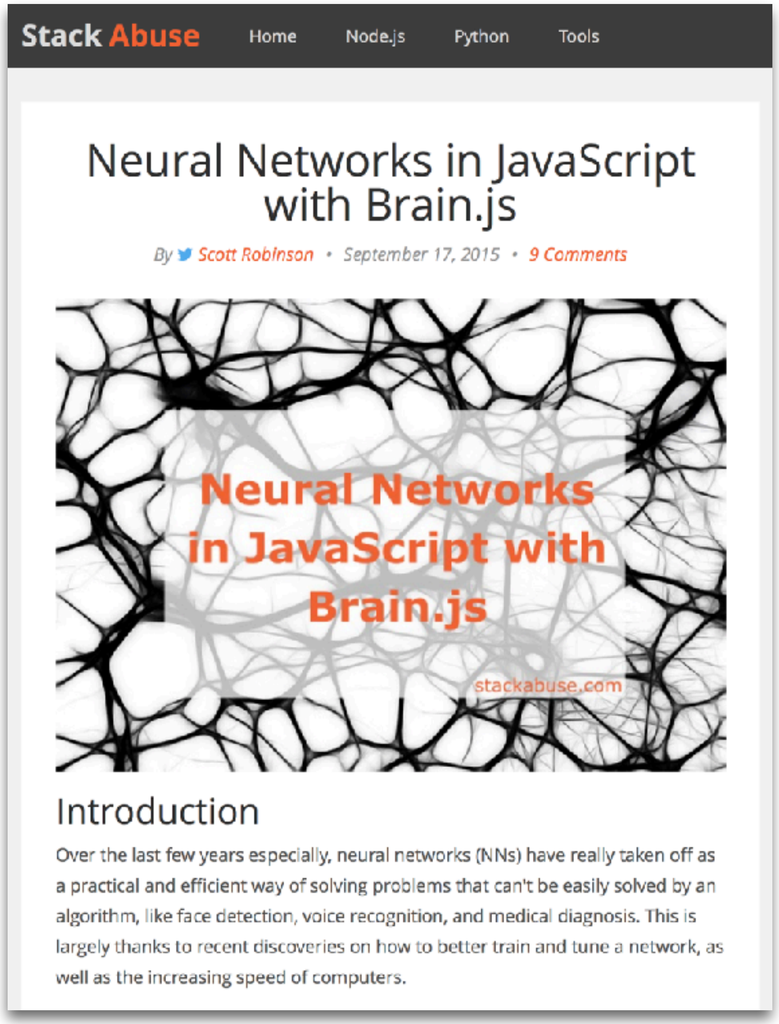
No need for the **HEAD** tag, it's already in the output.

JAVASCRIPT

CSS

Output: 0.8883243102192545
Number of iterations: 1022
Error: 0.004998365539591867

# Using brain.js for Image Recognition: http://stackabuse.com/neural-networks-in-javascript-with-brain-js/



Stack Abuse  Home  Node.js  Python  Tools

## Neural Networks in JavaScript with Brain.js

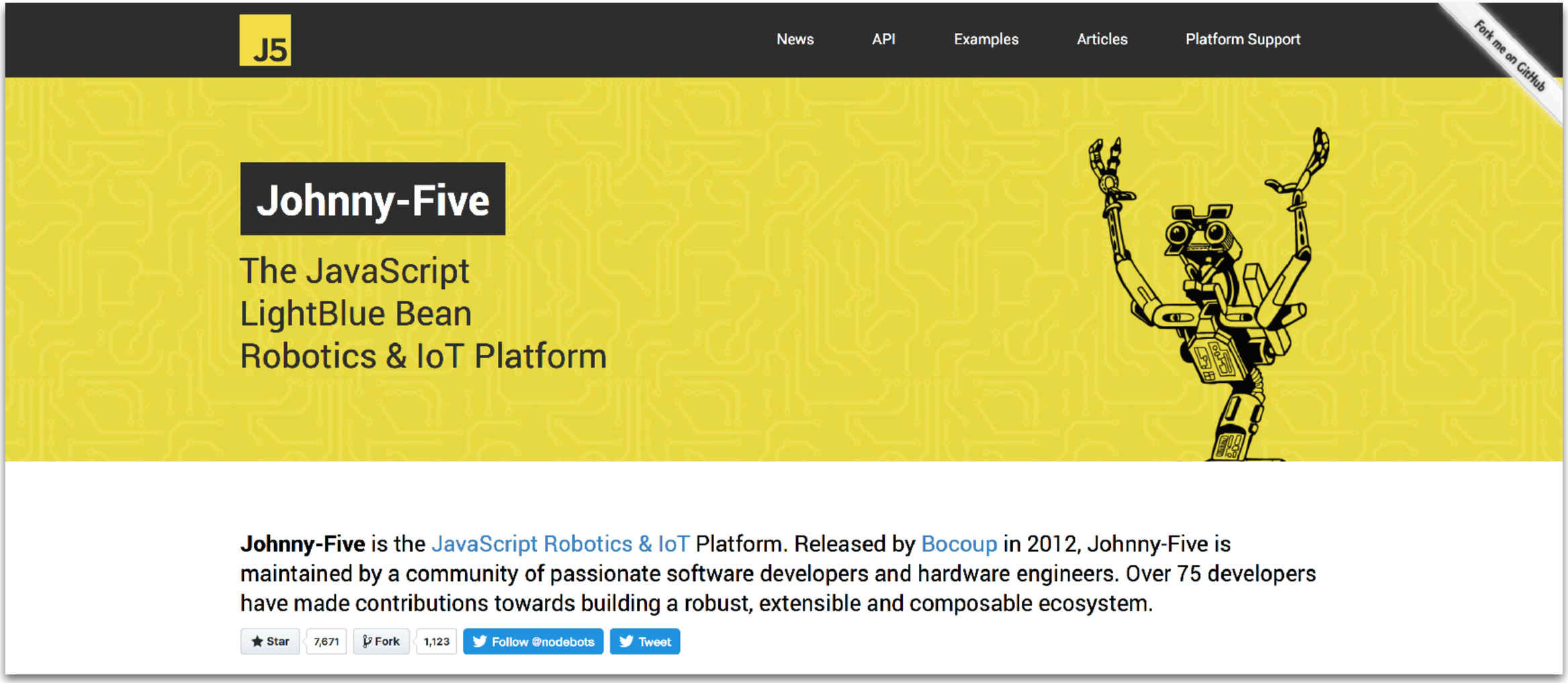
By Scott Robinson • September 17, 2015 • 9 Comments



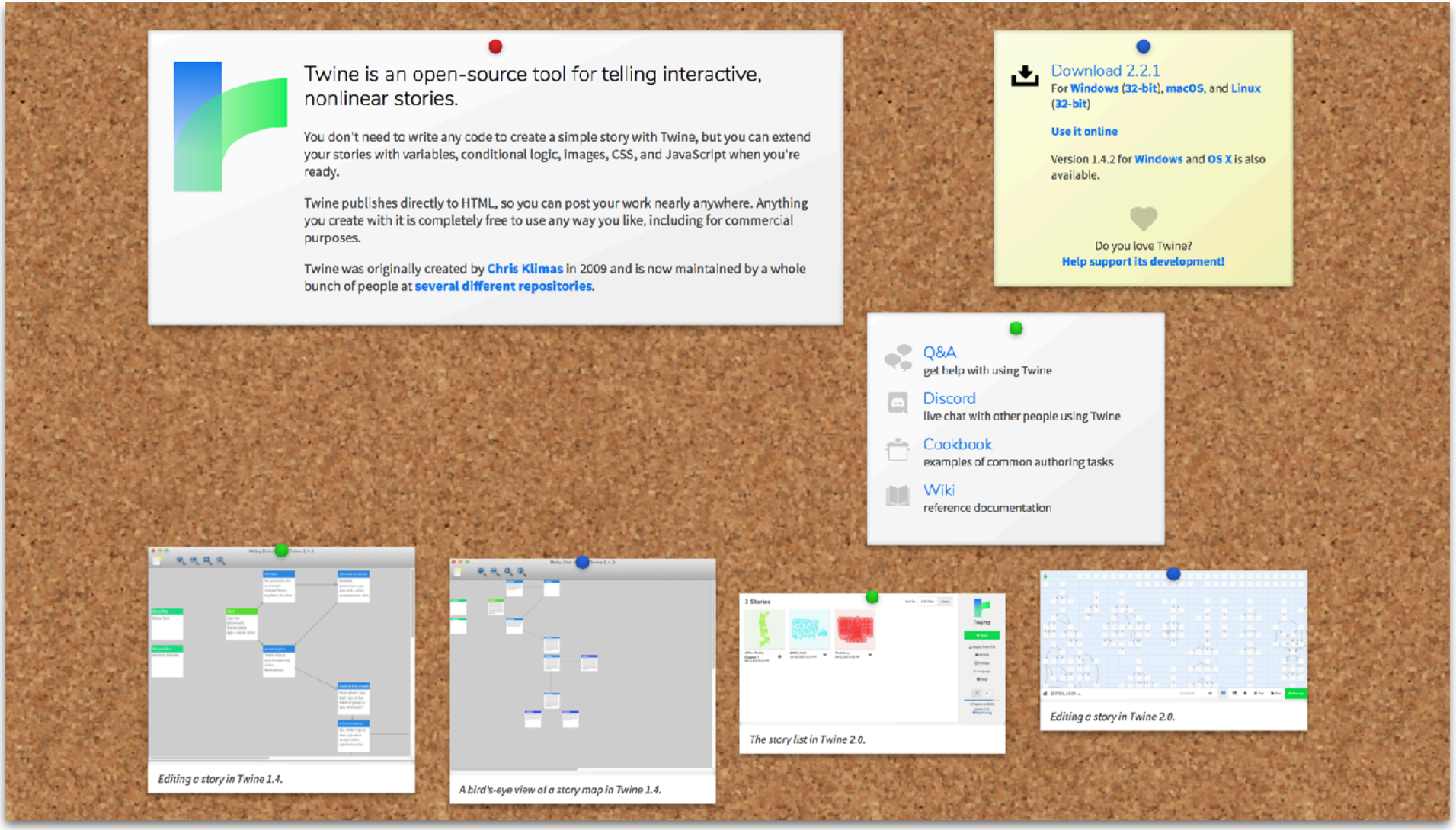
### Introduction

Over the last few years especially, neural networks (NNs) have really taken off as a practical and efficient way of solving problems that can't be easily solved by an algorithm, like face detection, voice recognition, and medical diagnosis. This is largely thanks to recent discoveries on how to better train and tune a network, as well as the increasing speed of computers.

# Connecting to Arduino - Johnny-Five: http://johnny-five.io

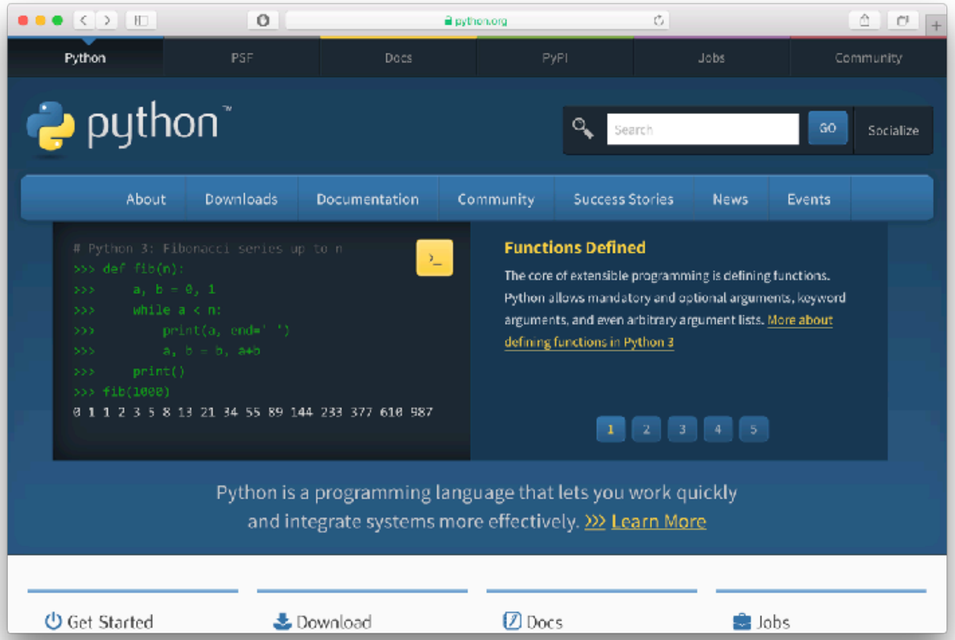


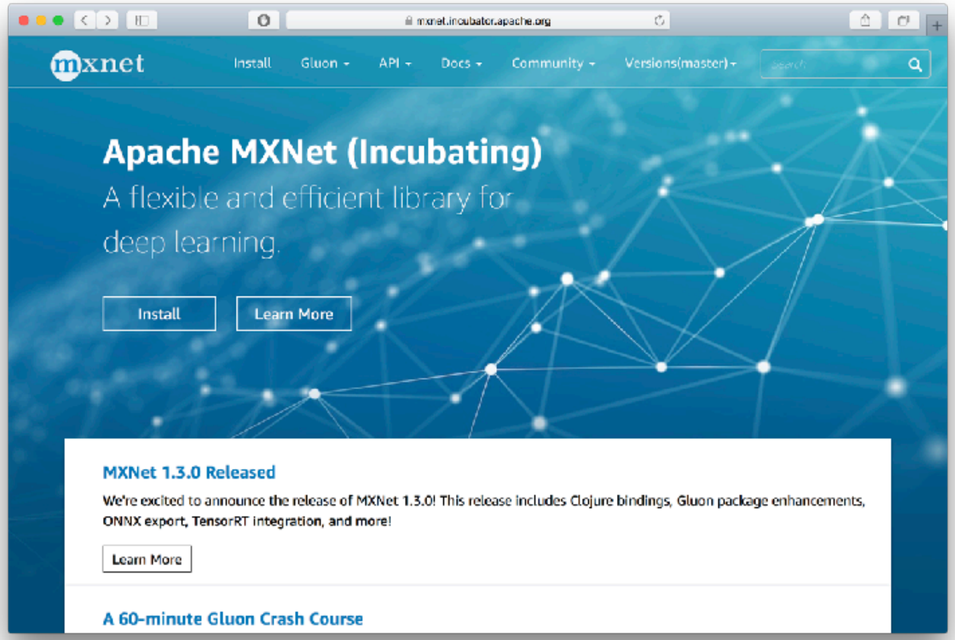
# Connecting to Text Adventures: Twine
http://twinery.org

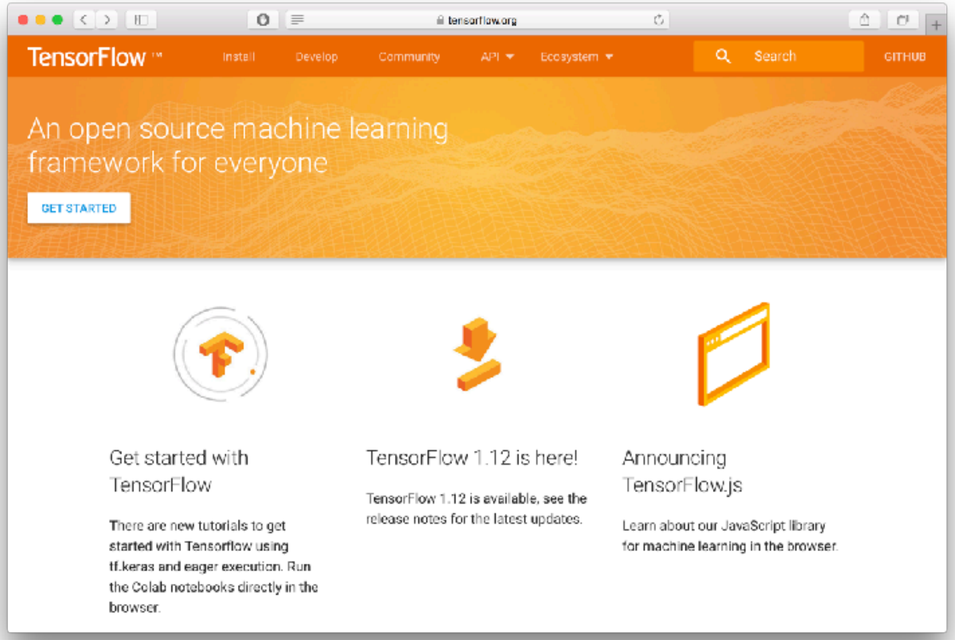


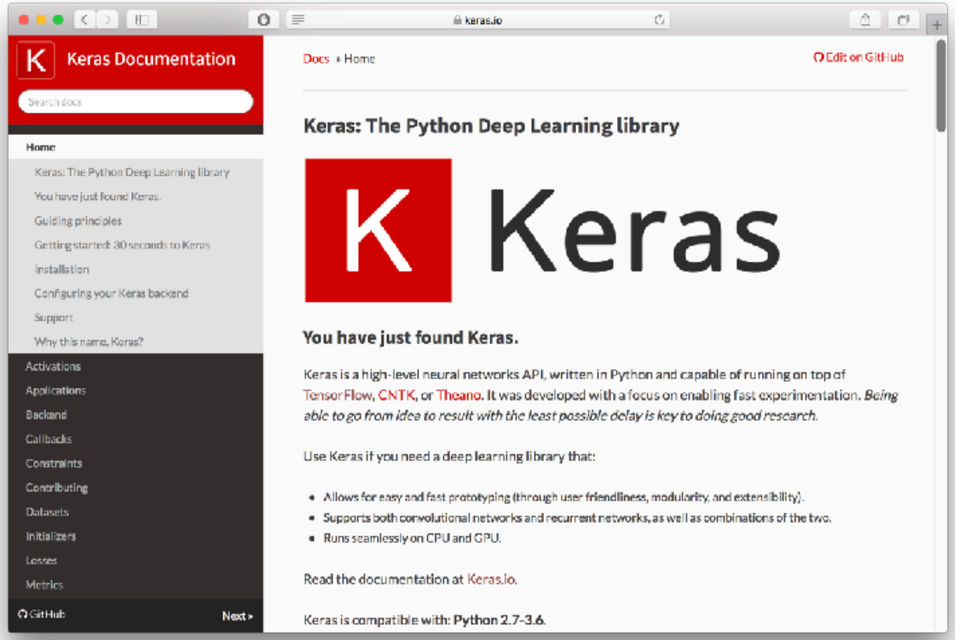
# Toolset Overview

| | |
|---|---|
| Deep Learning Studio | Neural Network GUI |
| Keras | Neural Network API |
| MXNet; TensorFlow | Computational Framework |
| Python | Programming Language |

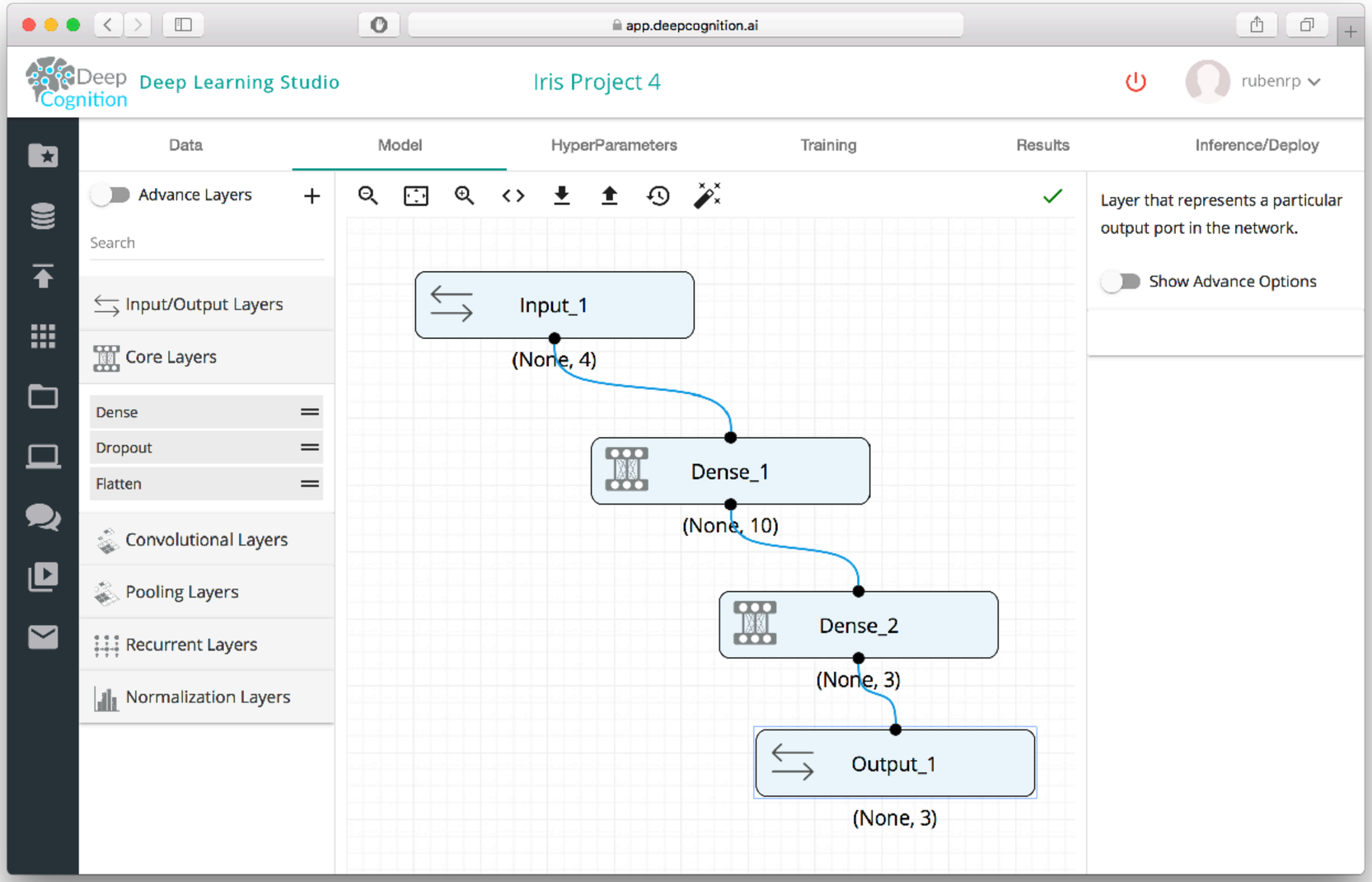

python.org
Python
PSF
Docs
PyPI
Jobs
Community
python
Search
GO
Socialize
About
Downloads
Documentation
Community
Success Stories
News
Events
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>> a, b = 0, 1
>>> while a < n:
>>> print(a, end=' ')
>>> a, b = b, a+b
>>> print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
Functions Defined
The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. More about defining functions 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


mxnet.incubator.apache.org
mxnet
Install
Gluon
API
Docs
Community
Versions(master)
Apache MXNet (Incubating)
A flexible and efficient library for deep learning.
Install
Learn More
MXNet 1.3.0 Released
We're excited to announce the release of MXNet 1.3.0! This release includes Clojure bindings, Gluon package enhancements, ONNX export, TensorRT integration, and more!
Learn More
A 60-minute Gluon Crash Course

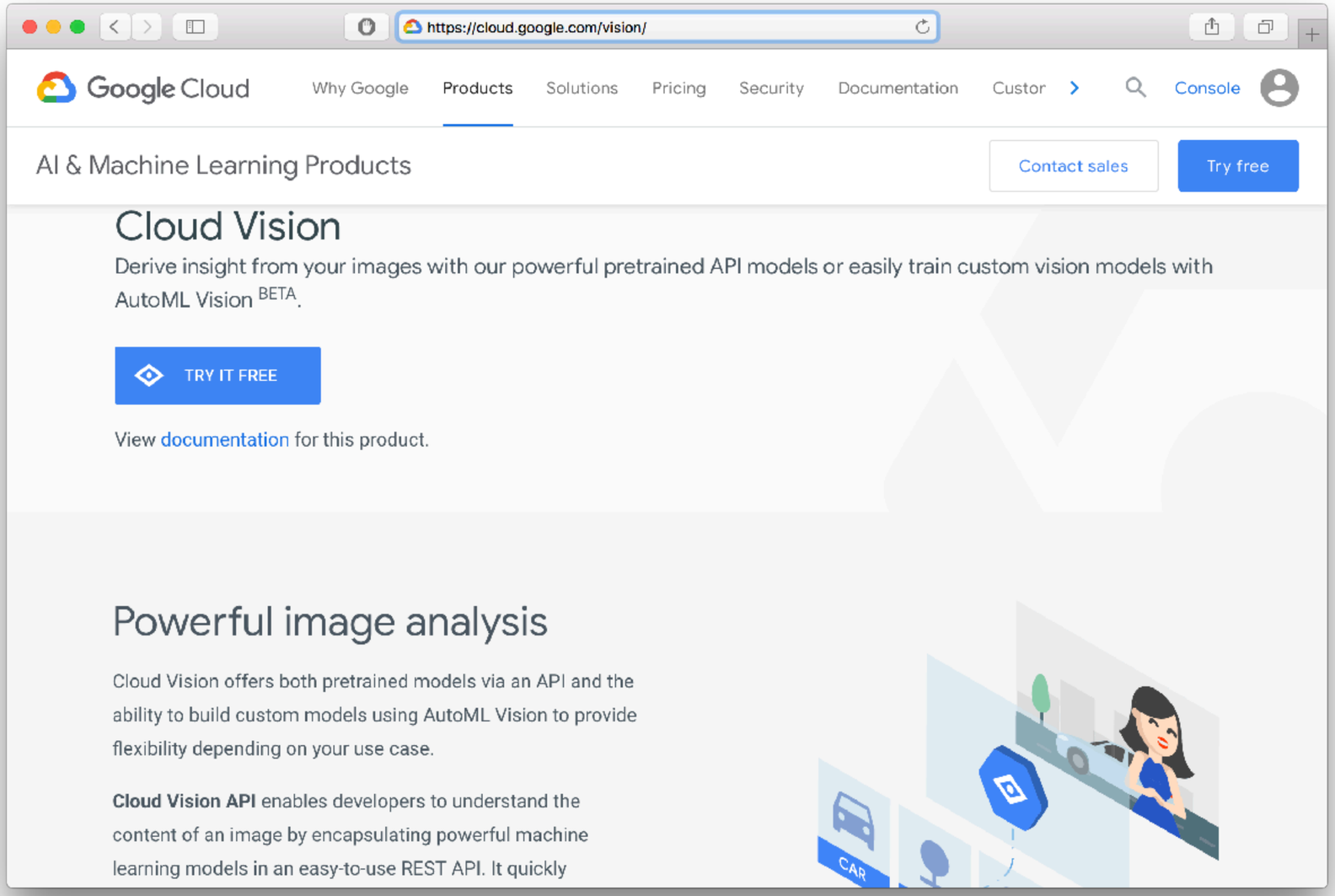

tensorflow.org
TensorFlow
Install
Develop
Community
API
Ecosystem
Search
GITHUB
An open source machine learning framework for everyone
GET STARTED
Get started with TensorFlow
There are new tutorials to get started with Tensorflow using tf.keras and eager execution. Run the Colab notebooks directly in the browser.
TensorFlow 1.12 is here!
TensorFlow 1.12 is available, see the release notes for the latest updates.
Announcing TensorFlow.js
Learn about our JavaScript library for machine learning in the browser.


keras.io
Keras Documentation
Search docs
Home
Keras: The Python Deep Learning library
You have just found Keras.
Guiding principles
Getting started: 30 seconds to Keras
Installation
Configuring your Keras backend
Support
Why this name, Keras?
Activations
Applications
Backend
Callbacks
Constraints
Contributing
Datasets
Initializers
Losses
Metrics
GitHub
Next »
Docs » Home
Edit on GitHub
Keras: The Python Deep Learning library
Keras
You have just found Keras.
Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.
Use Keras if you need a deep learning library that:
Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
Supports both convolutional networks and recurrent networks, as well as combinations of the two.
Runs seamlessly on CPU and GPU.
Read the documentation at Keras.io.
Keras is compatible with: Python 2.7-3.6.

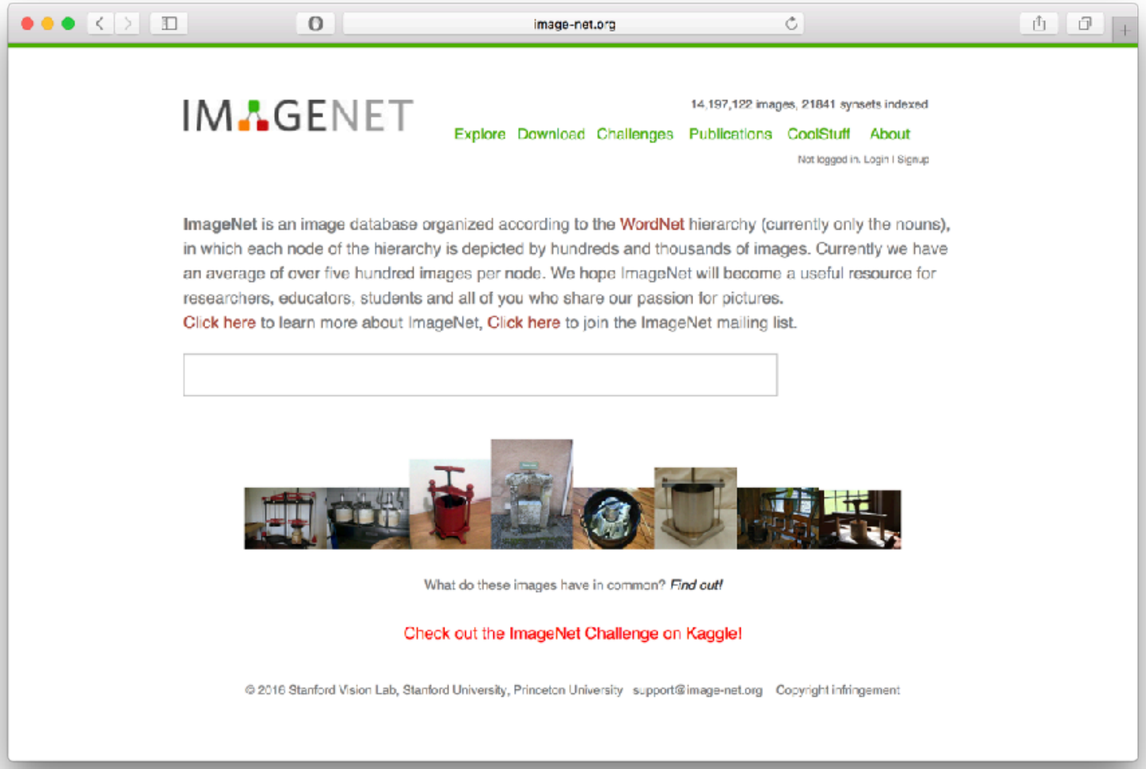
# Deep Cognition (https://deepcognition.ai)
# Deep Learning Studio

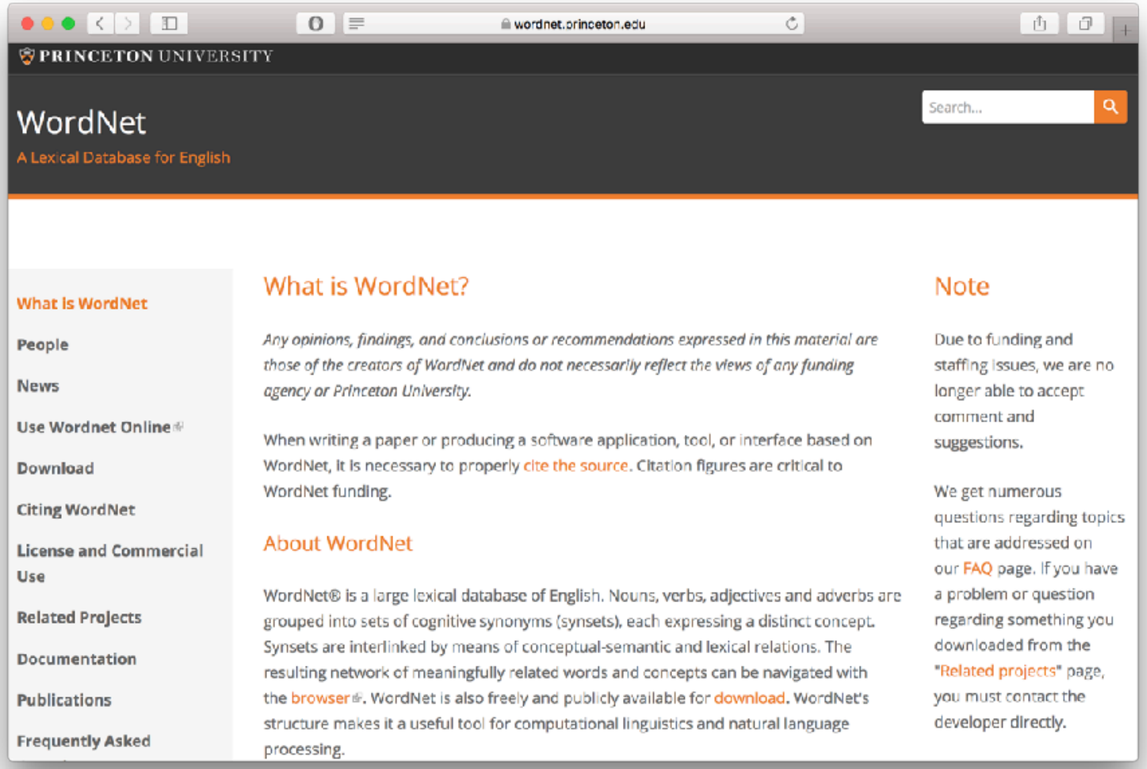


# Google Vision (https://cloud.google.com/vision/)

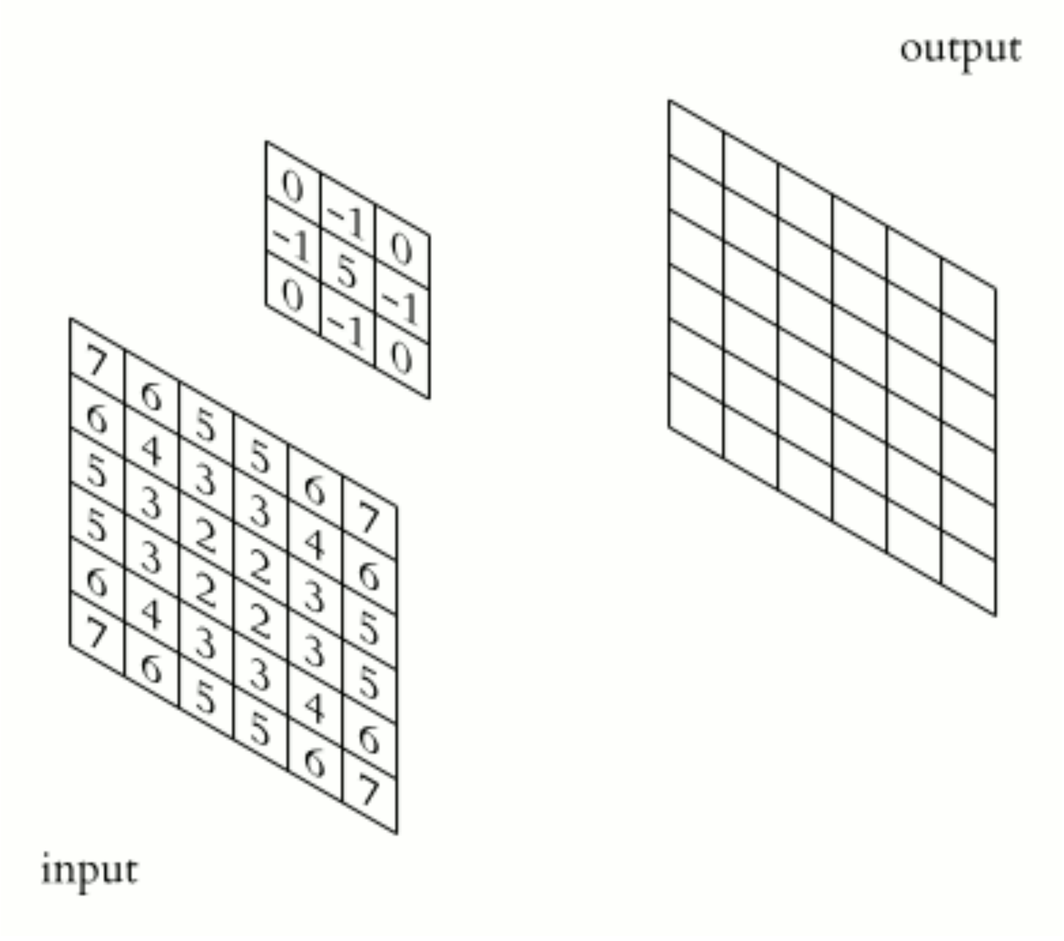


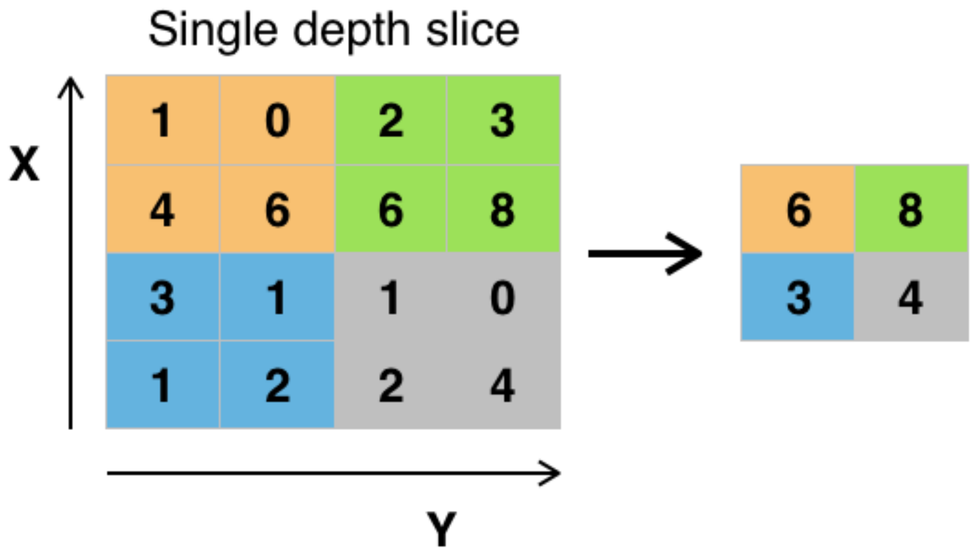
# ImageNet (http://www.image-net.org)
# WordNet (https://wordnet.princeton.edu)



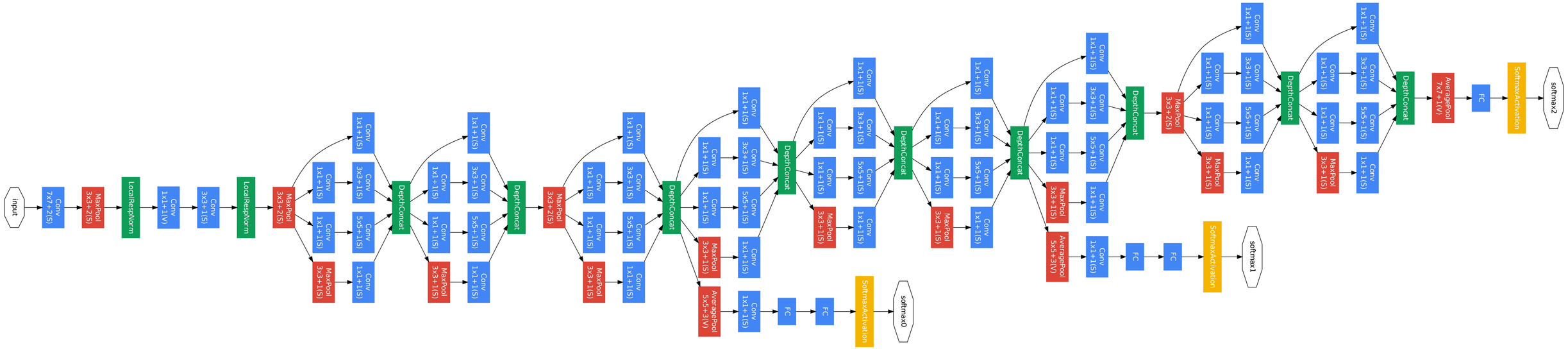
# Convolution and Pooling Layers



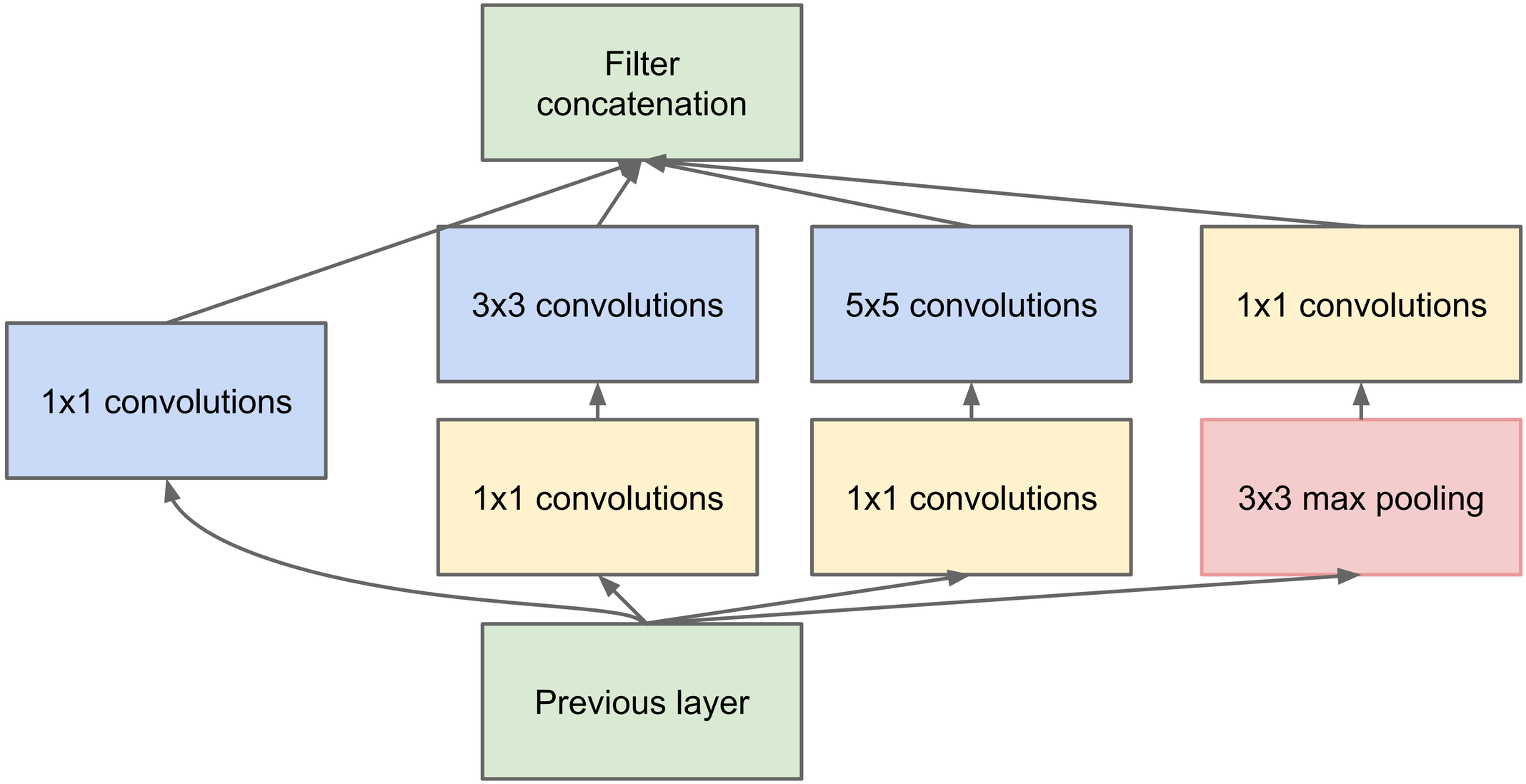

Images Source: Wikipedia

# The Inception V1 Model

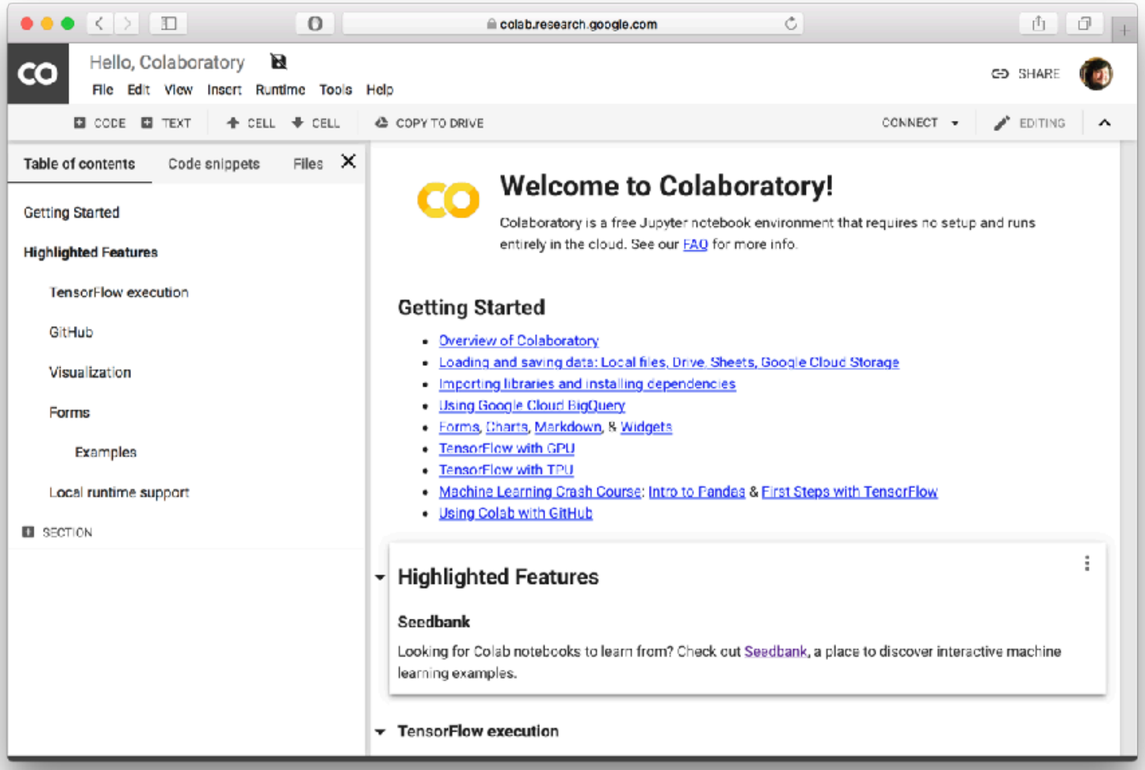


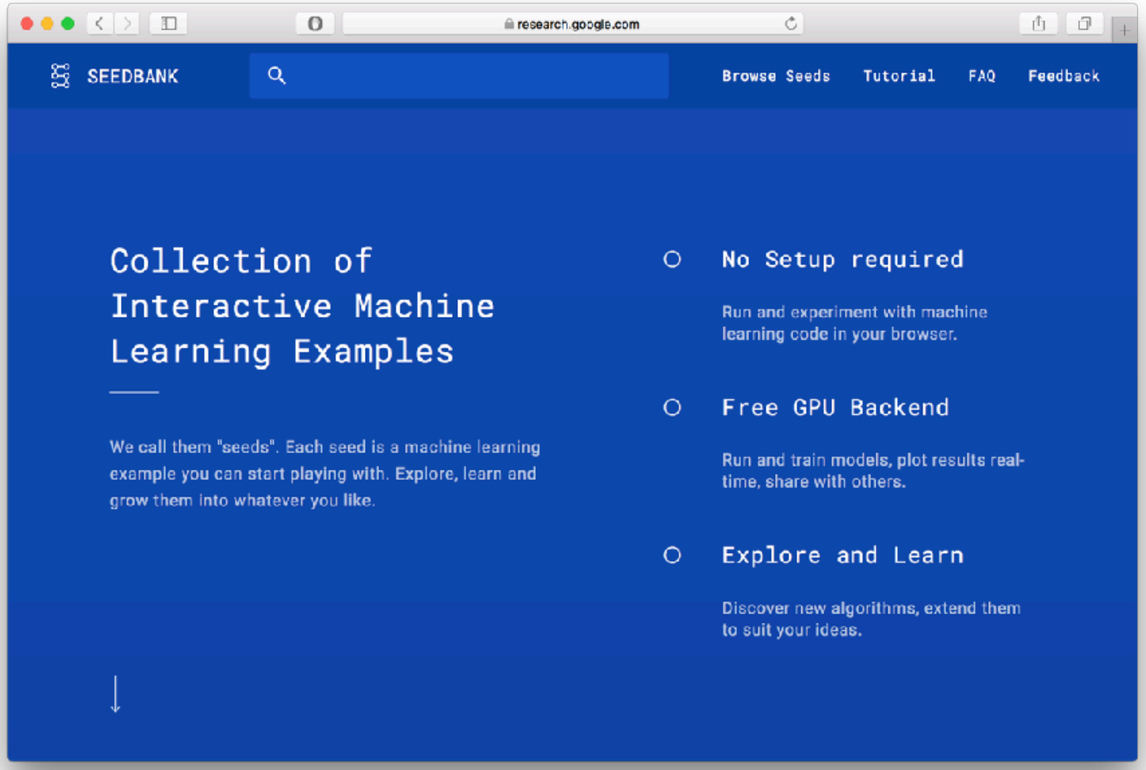

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1-9. 2015.

# The Inception Module Architecture

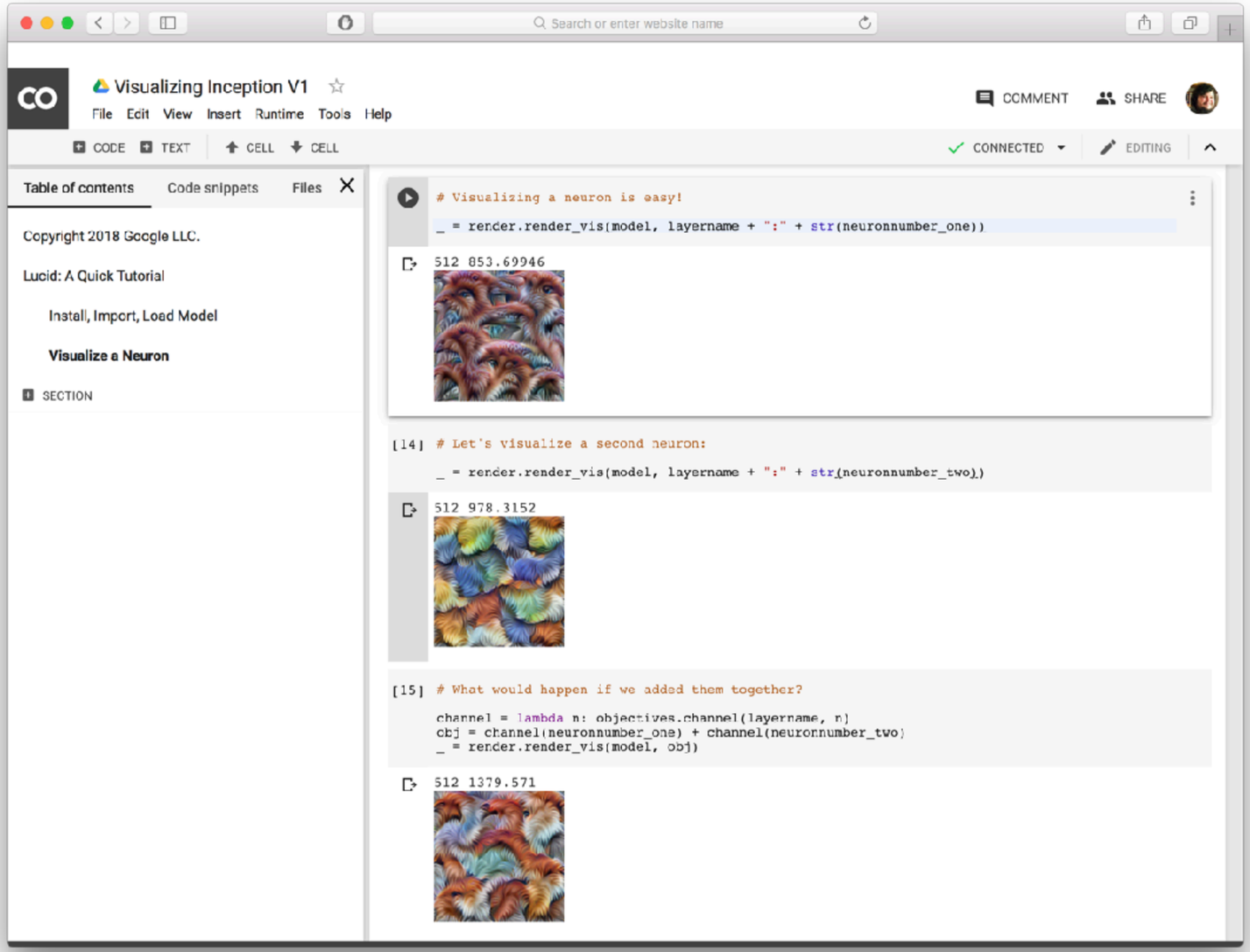


# Google Colaborative (https://colab.research.google.com)
# Seedbank (https://research.google.com/seedbank/)

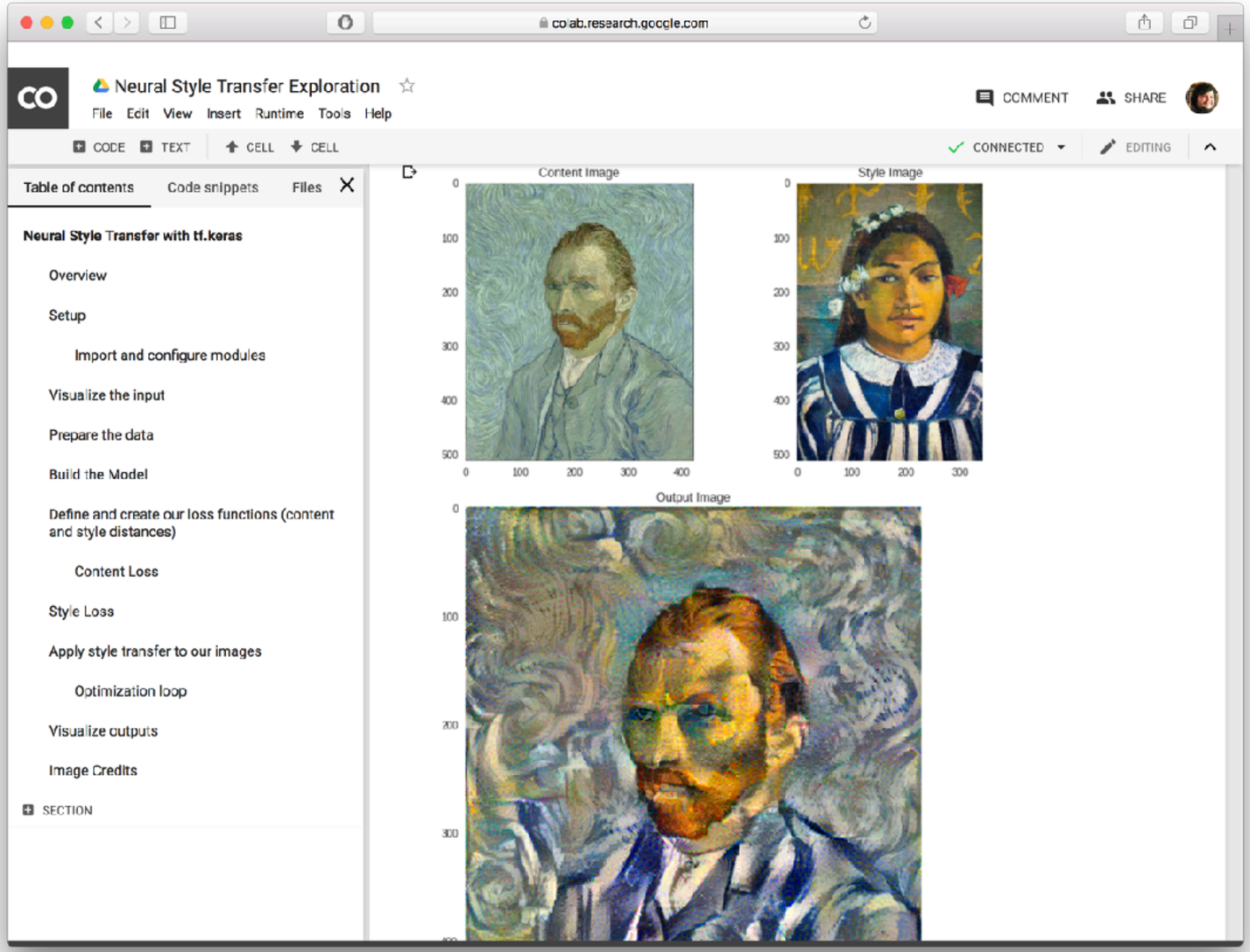


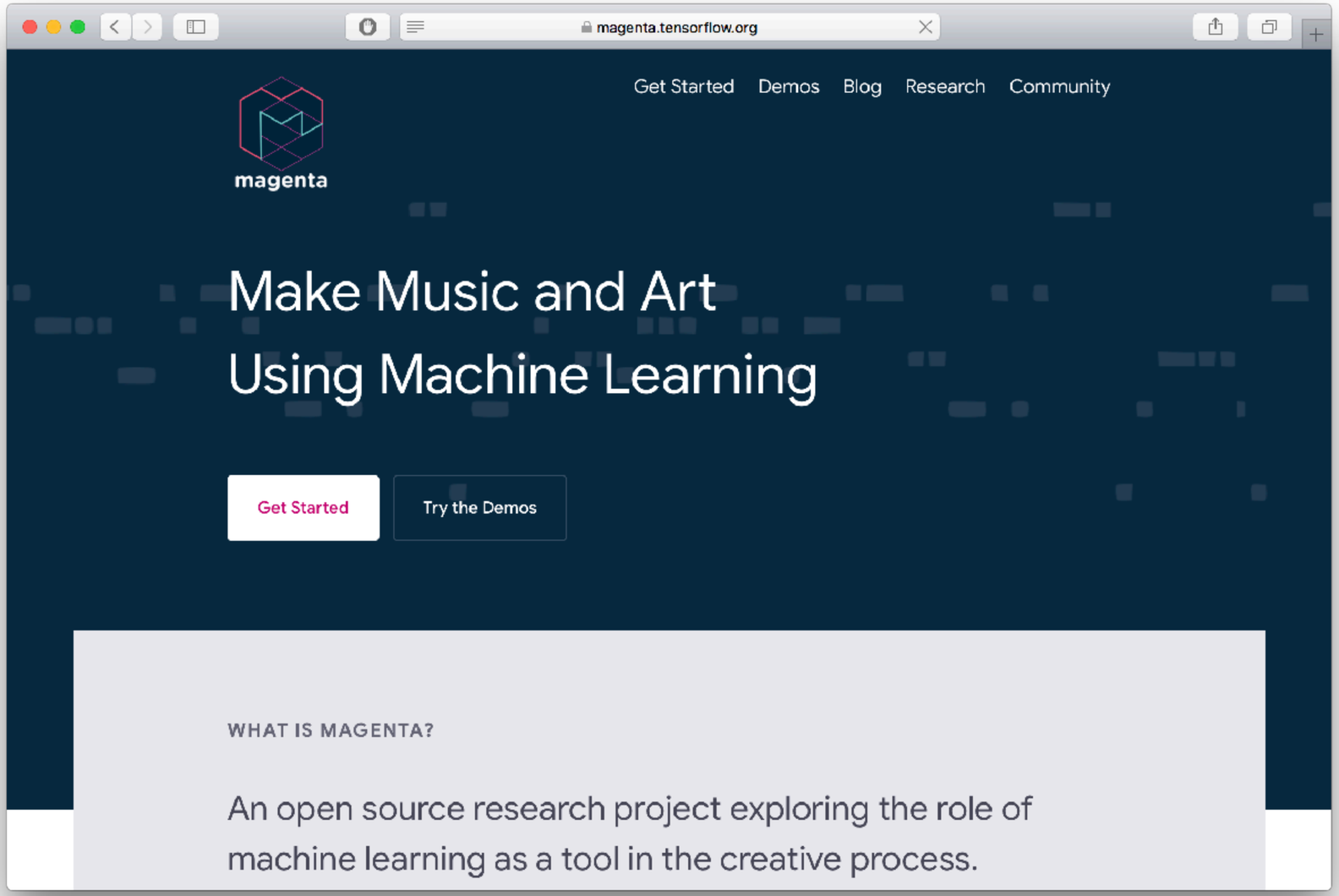
# Visualizing the Inception V1 Model
# https://tinyurl.com/nn1vtfest18



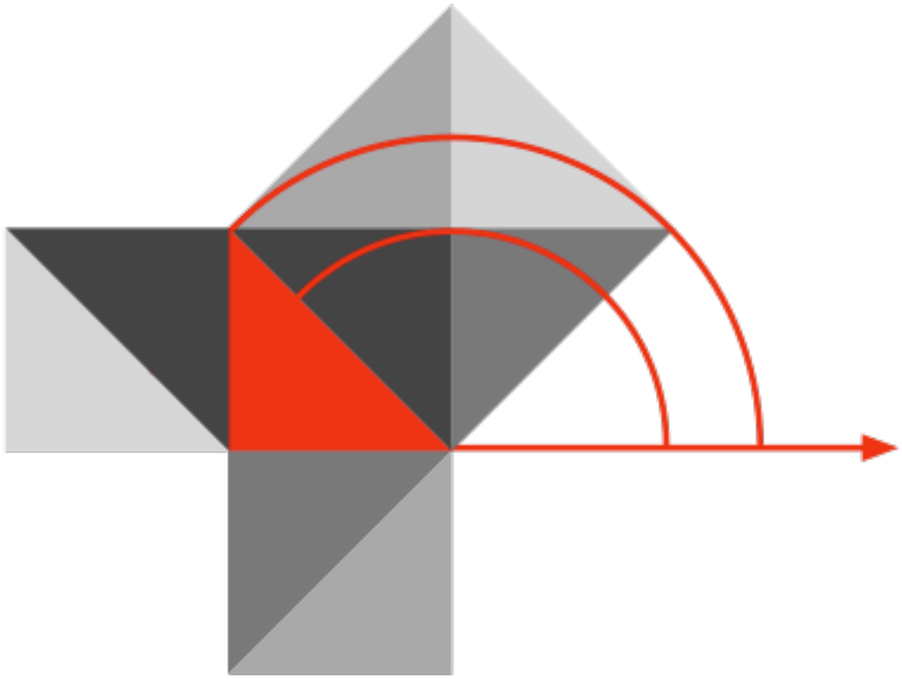
# Exploring Neural Style Transfer
# https://tinyurl.com/nn2vtfest18



# Magenta (https://magenta.tensorflow.org)



# Hippasus



Blog: http://hippasus.com/blog/
Email: rubenrp@hippasus.com
Twitter: @rubenrp

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License.