

LET'S OPEN THE DEEP LEARNING BLACK BOX!

JORDI VITRIÀ



UNIVERSITAT DE
BARCELONA

jordi.vitria@ub.edu

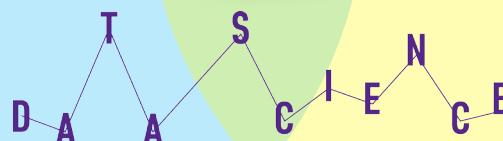
DΛS

jordi.vitria@ub.edu

Departament de Matemàtiques i Informàtica
Universitat de Barcelona

Machine
Learning

Computer
Vision



Since 2007, I am a Full Professor at the Mathematics & Computer Science Department, **Universitat de Barcelona**. Before that I spent 20 years on the faculty of the CS Department at the **Universitat Autònoma de Barcelona**. I am the Director of the **Data Science & Big Data Postgraduate Course** and the **Foundations of Data Science Master** at UB. I am the leader of the **DataScience@UB** group, whose objective is to promote technology transfer.

Some examples of our research
(that involve **deep learning** methods)

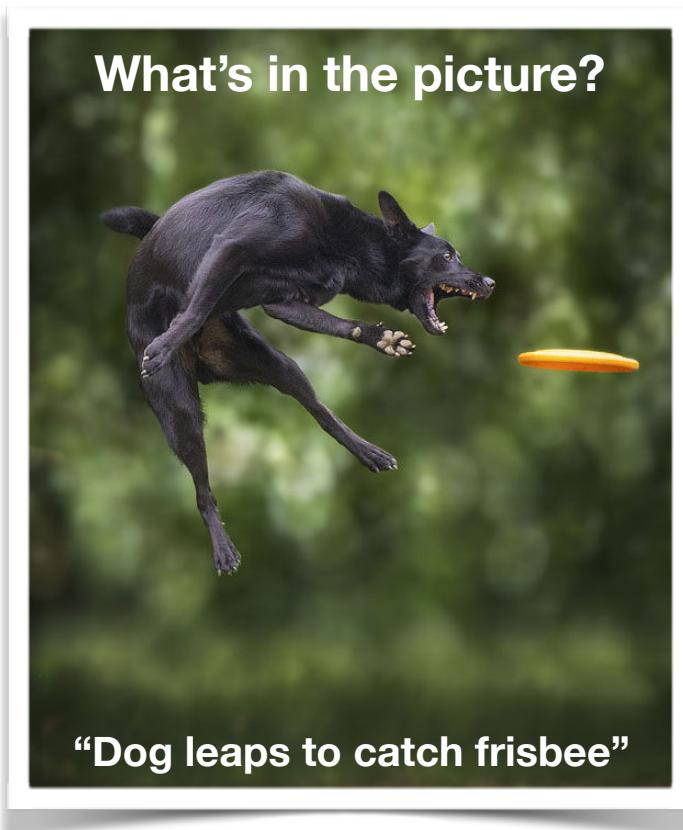
end-to-end learning

deep neural networks

“black box” learning...

Extracting non visual attributes from images using CNN.

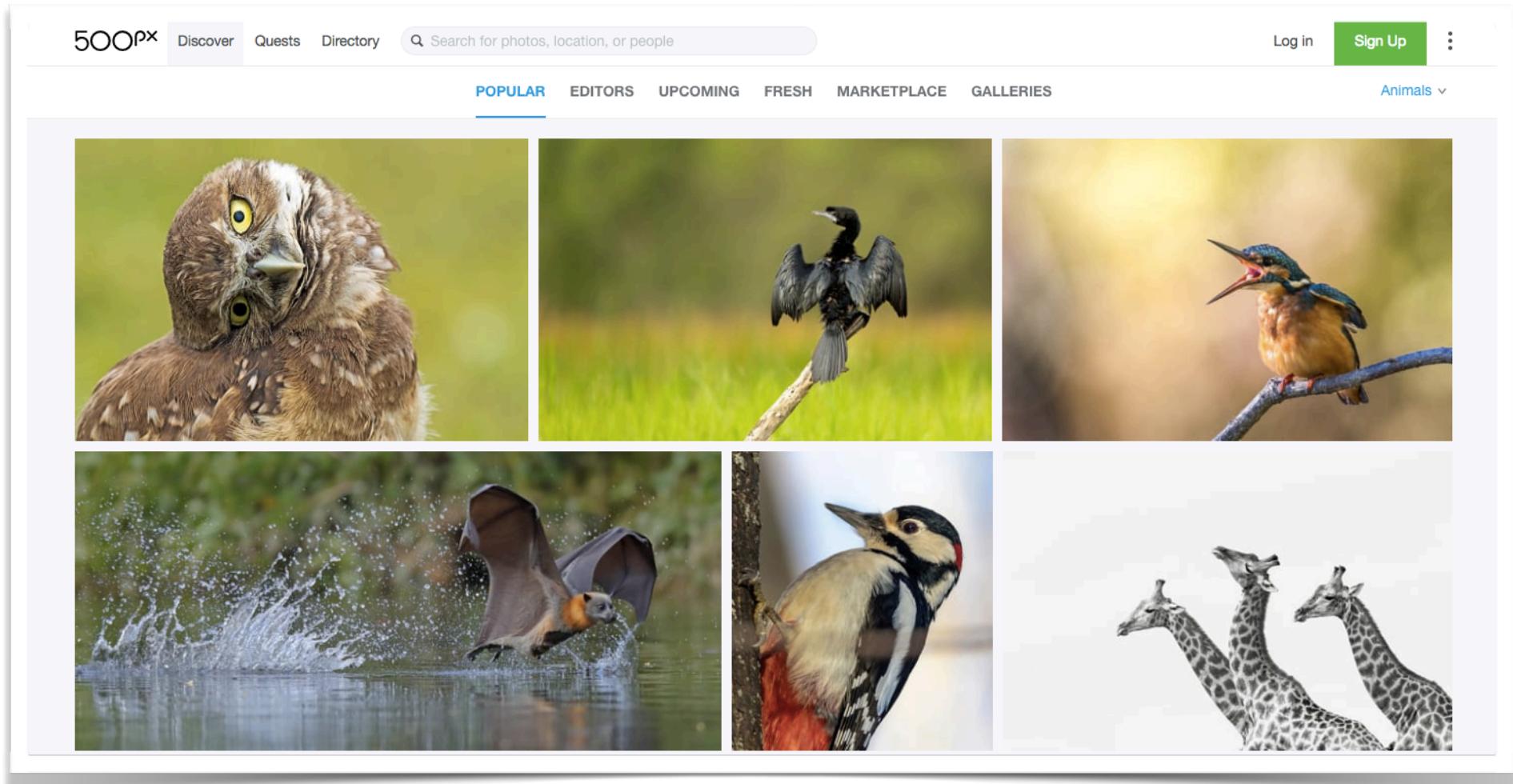
Non-visual attributes are those attributes of an image that can be inferred from visual information but do not have a clear correspondence on the image.





Which apartment is, *a priori*, more successful on Airbnb?

Online photo marketplace



Which is the expected popularity of these images?

Is Deep Learning Overhyped?

The collage illustrates the widespread media coverage of deep learning, featuring several prominent publications:

- BBC News Business:** A video by Colm O'Regan explaining "What is 'deep learning'?"
- Financial Times:** An article titled "The mind in the machine: Demis Hassabis on artificial intelligence" featuring a circular graphic of glowing dots.
- Nature:** A magazine cover with a Go board on a circuit board, titled "ALL SYSTEMS GO".
- The New York Times Magazine:** A dark cover with a small image of a person and text about songbirds.
- The Economist:** An article titled "From not working to neural networking" with a DeepDream image of a dog and a car.

Below the collage, there are two additional news snippets:

New York Times Magazine: An article titled "The Great A.I. Awakening" by Gideon Lewis-Kraus, dated Dec 14, 2016.

Scientific American: A snippet from an article by Michael Tyka, showing a DeepDream image of a cat and a diagram of a neural network layer.

Objectives



1. What is Deep Learning?

Deep Learning is not magic.

or how to train large and highly complex models with deeply cascaded nonlinearities by using automatic differentiation and several tricks.

2. What are the main applications of Deep Learning?

computer vision, natural language, speech, recommenders, time series, etc.

3. What are the main limitations of Deep Learning?

Deep Learning is not the final machine learning method.

4. How to build deep learning models?

Keras, Tensorflow...

THE REVENANT

INSPIRED BY TRUE EVENTS
JANUARY 8

A movie poster for "The Revenant". The background is a dark, moody landscape with a forest silhouette and a close-up of a man's face. The man, played by Leonardo DiCaprio, has long hair and a beard, and is wearing a heavy fur-trap. He has a look of exhaustion and determination on his face. The title "THE REVENANT" is at the top in large white letters, followed by the subtitle "INSPIRED BY TRUE EVENTS" and the release date "JANUARY 8".

Why Deep Learning?

History



- In 1943, neurophysiologist **Warren McCulloch** and mathematician **Walter Pitts** wrote a paper on how neurons might work. In order to describe how neurons in the brain might work, they modeled a simple neural network using **electrical circuits**.
- In 1949, Donald **Hebb** wrote *The Organization of Behavior*, a work which pointed out the fact that neural pathways are strengthened each time they are used, a concept fundamentally essential to the ways in which humans learn. If two nerves fire at the same time, he argued, the connection between them is enhanced.
- In 1957 **Frank Rosenblatt** attempted to build a kind of mechanical brain called the **Perceptron**, which was billed as “a machine which senses, recognizes, remembers, and responds like the human mind”.



- In 1962, **Widrow & Hoff** developed a learning procedure that examines the value before the weight adjusts it (i.e. 0 or 1) according to the rule: Weight Change = (Pre-Weight line value) * (Error / (Number of Inputs)). It is based on the idea that while one active perceptron may have a big error, one can adjust the weight values to distribute it across the network, or at least to adjacent perceptrons.
- A critical book written in 1969 by **Marvin Minsky** and his collaborator **Seymour Papert** showed that Rosenblatt's original system was painfully limited, literally blind to some simple logical functions like "exclusive-or" (As in, you can have the cake or the pie, but not both). What had become known as the field of "neural networks" all but disappeared.

First neural network winter is coming



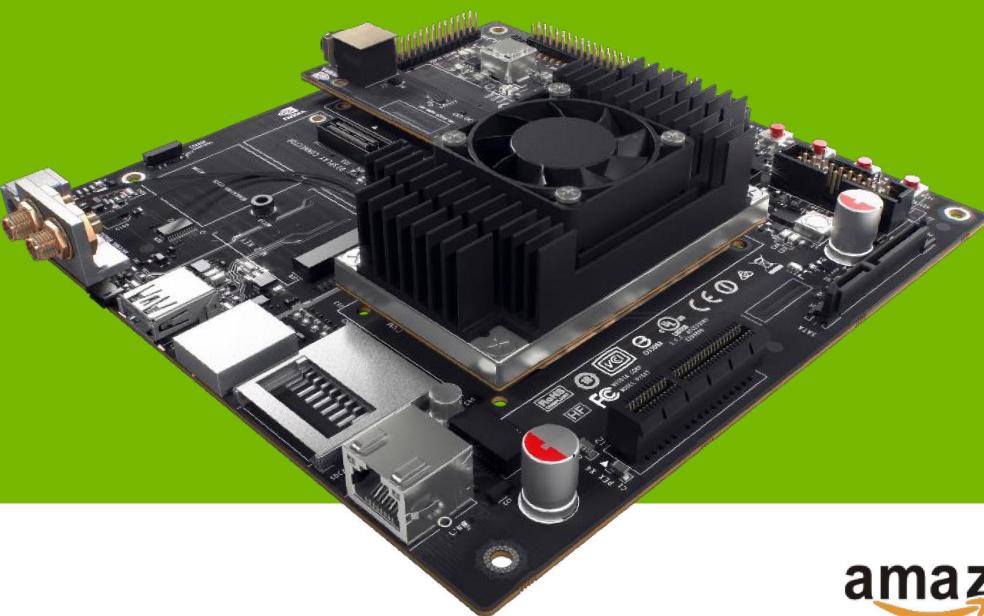


- In 1982, interest in the field was renewed. **John Hopfield** of Caltech presented a paper to the National Academy of Sciences. His approach was to create more useful machines by using bidirectional lines. Previously, the connections between neurons was only one way.
- In 1986, the problem was how to extend the Widrow-Hoff rule to multiple layers. Three independent groups of researchers, which included **David E. Rumelhart**, **Geoffrey E. Hinton** and **Ronald J. Williams**, came up with similar ideas which are now called **back-propagation networks** because it distributes pattern recognition errors throughout the network.
- From 1986 to mid 90's new developments arised: convolutional neural networks (**Y.LeCun**), unsupervised learning (**Y.Bengio**), RBM (**G.Hinton**), etc. But, by this point **new machine learning methods** had begun to also emerge, and people were again beginning to be skeptical of neural nets since they seemed so intuition-based and since computers were still barely able to meet their computational needs.

Second neural network winter is coming



- With the ascent of **Support Vector Machines** and the failure of backpropagation, the early 2000s were a dark time for neural net research.
- Then, what every researcher must dream of actually happened: G.Hinton, S.Osindero, and Y.W.Teh published a paper in 2006 that was seen as a breakthrough, a breakthrough significant enough to rekindle interest in neural nets: *A fast learning algorithm for deep belief nets.*
- After that, following Moore's law, computers got dozens of times faster (GPUs) since the slow days of the 90s, making learning with large datasets and many layers much more tractable.



Jetson TX1 Developer Kit
\$599 retail
\$299 edu
Pre-order Nov 12
Shipping Nov 16 (US)
Intl to follow



GPU democratization

The screenshot shows the NVIDIA Accelerated Computing website. At the top, there is a navigation bar with the NVIDIA logo, "ACCELERATED COMPUTING", and links for "Downloads", "Training", "Ecosystem", and "Forums". To the right of the navigation bar are a search icon and a "Log in" button. Below the navigation bar, the page title "GPU Grant Program" is displayed in green. A breadcrumb navigation path "Home > ComputeWorks > Academic Collaboration > GPU Grant Program" is shown below the title. The main content area starts with a paragraph about NVIDIA's Academic Programs Team and their initiatives. It then lists several bullet points about the GPU Grant Program, including small scale grants, graduate fellowships, teaching kits, and developer forums. Below this, a section titled "GPU Grant Requests" provides instructions for professors, researchers, and advisors on how to request a GPU. It specifies that they should complete an online form and attach a statement of research and CV, listing required information such as contact details, project descriptions, GPU usage plans, and publication lists.

NVIDIA's Academic Programs Team is dedicated to empowering and collaborating with professors and researchers at universities worldwide. We aim to inspire cutting-edge technological innovation and to find new ways of enhancing faculty research as well as the teaching and learning experience. We achieve this through a variety of initiatives and programs including:

- Small scale GPU grants
- Graduate Fellowships
- Providing free teaching materials and GPU cloud resources through our Deep Learning Institute (DLI) Teaching Kits
- Providing access to developer forums, pre-released tools and drivers through the Accelerated Computing Developer Program

GPU Grant Requests

Professors, Researchers and Advisors should complete the online [GPU Grant Request Form](#) to request a GPU for research purposes for themselves and/or their teams. In order to review your request, we require a statement of research and CV. Additional materials should not be used to attach a full proposal in lieu of a statement of research. Your statement of research must include the following:

- contact information
- a short description of your research project(s)
- how you and/or your team will use the GPU
- list of recent publications

Thank you NVIDIA!

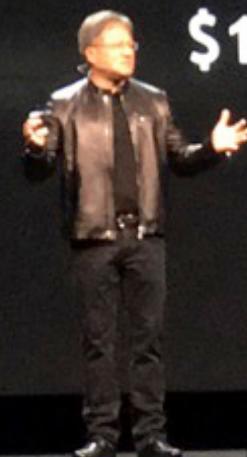


NVIDIA DGX-1

WORLD'S FIRST
DEEP LEARNING SUPERCOMPUTER

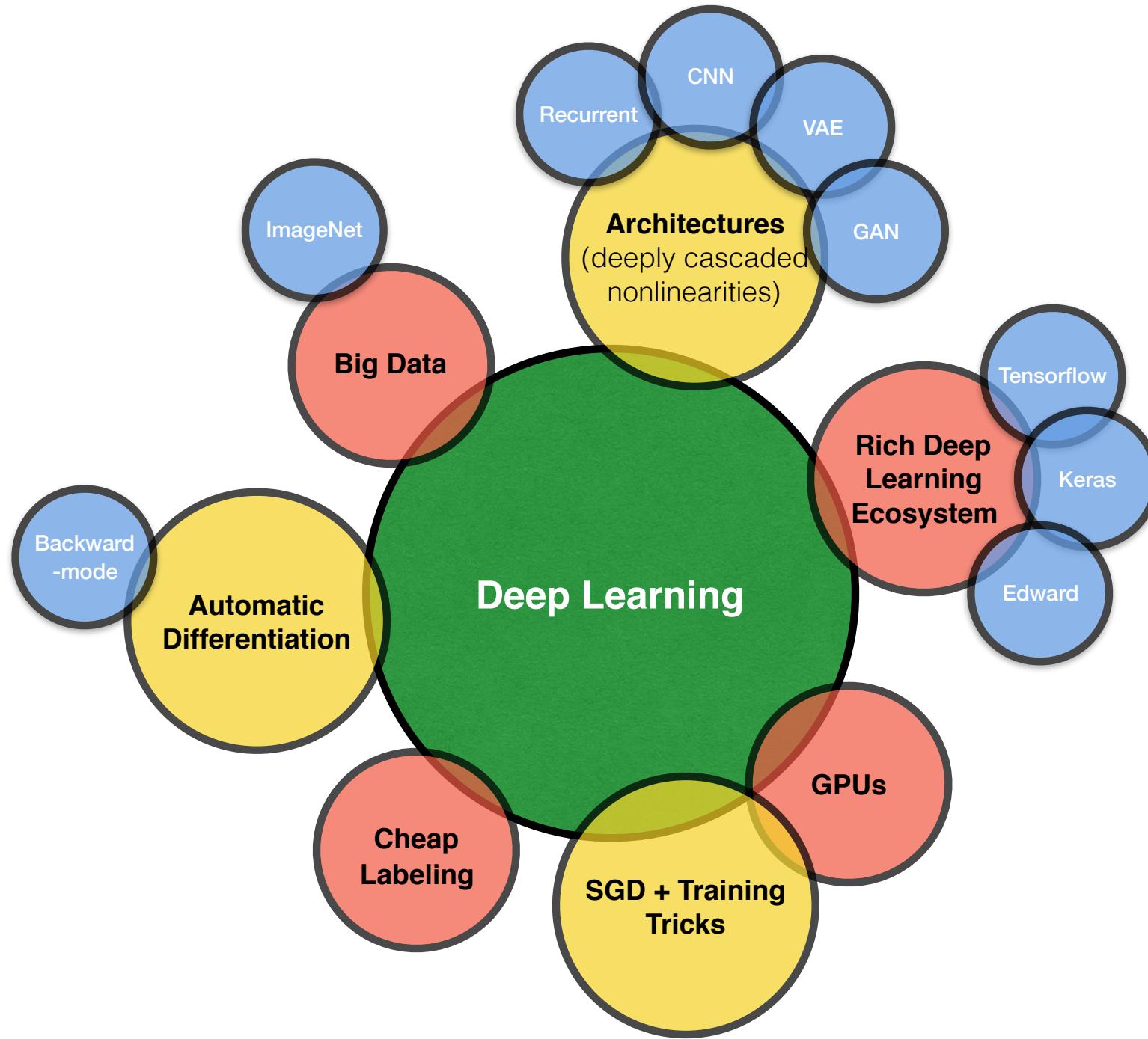
170TF | "250 servers in-a-box" | nvidia.com/dgx1

\$129,000



Definitions

- **Neural Networks (NN)** is a beautiful biologically-inspired programming paradigm which enables a computer to learn from observational data.
- **Deep Learning (DL)** is a powerful set of techniques (and tricks) for learning in deep neural networks.
- NN and DL currently provide the best solutions to many problems in image recognition, speech recognition, and natural language processing.

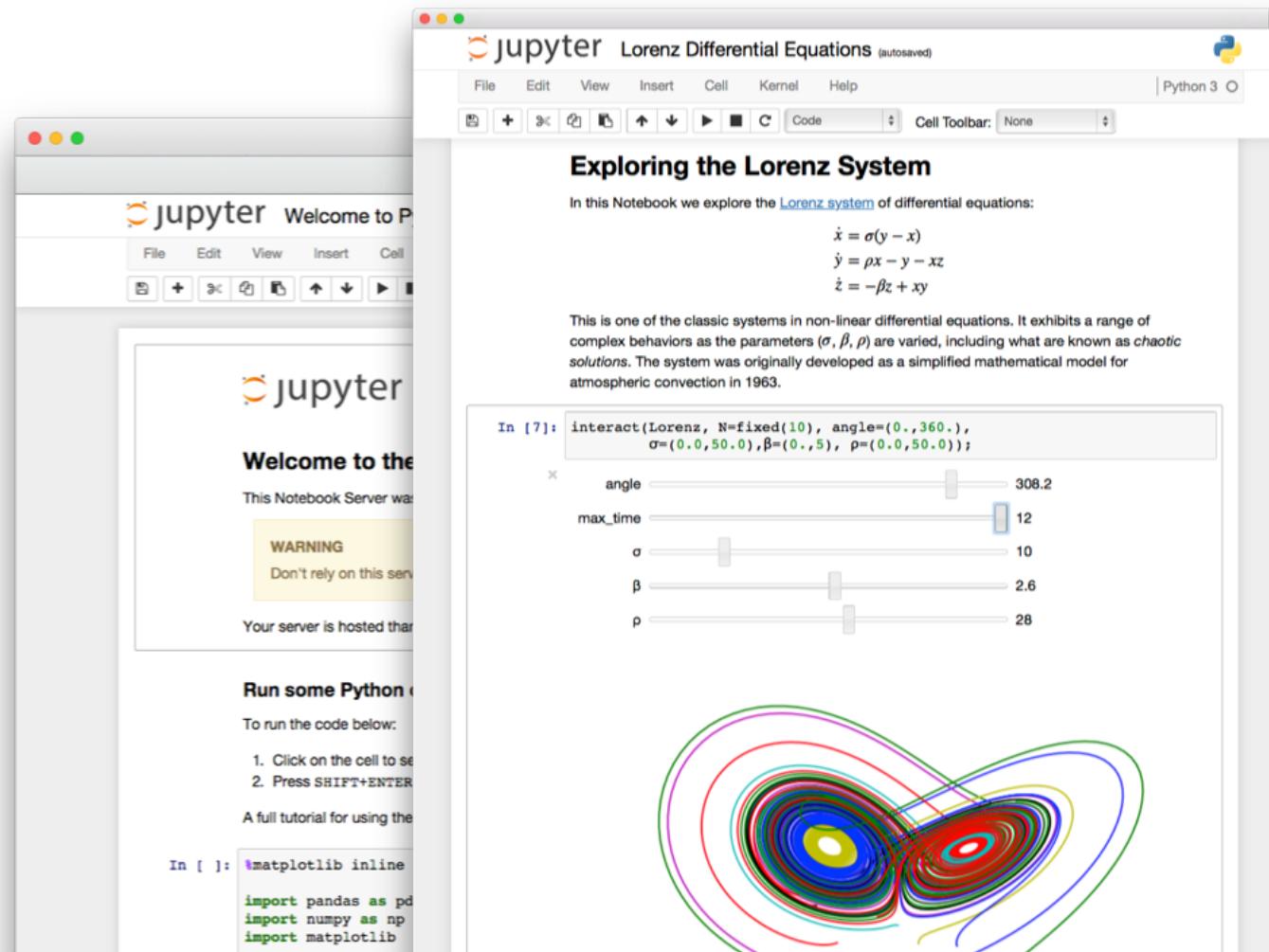


Our objectives

- Optimization and Automatic Differentiation
- Programming a Neural Network
- Design and Train a Deep Model

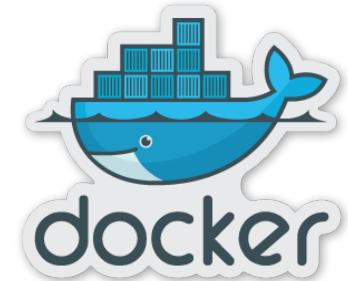
Approach

We will illustrate all contents with Jupyter notebooks, a web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text.



Approach

We will use a **Docker Container**.



Docker provides the ability to build a runtime environment that not only remains isolated from other running containers, but also can be deployed to multiple locations in a repeatable way. Docker also uses a text document – a Dockerfile – that contains all the commands to assemble an image, which will meet our need to document the build environment. Finally, Docker's runtime options enable us to attach GPU devices when deploying on remote servers.

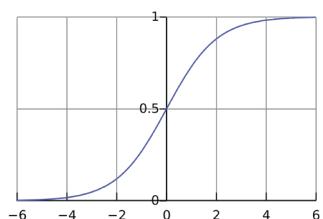
The problem: **machine learning**

Training data: a set of $(x^{(m)}, y^{(m)})$ pairs.

Learn a function $f_w : x \rightarrow y$ to predict on new inputs x .

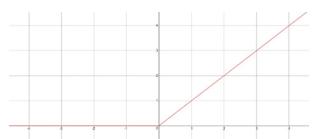
1. Choose a model function family f_w .
2. Optimize parameters w .

1-layer neural net model



Sigmoid
Function

$$\sigma(x) = \frac{1}{(1+e^{-x})}$$



ReLU
Function

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

Parameters



Weights



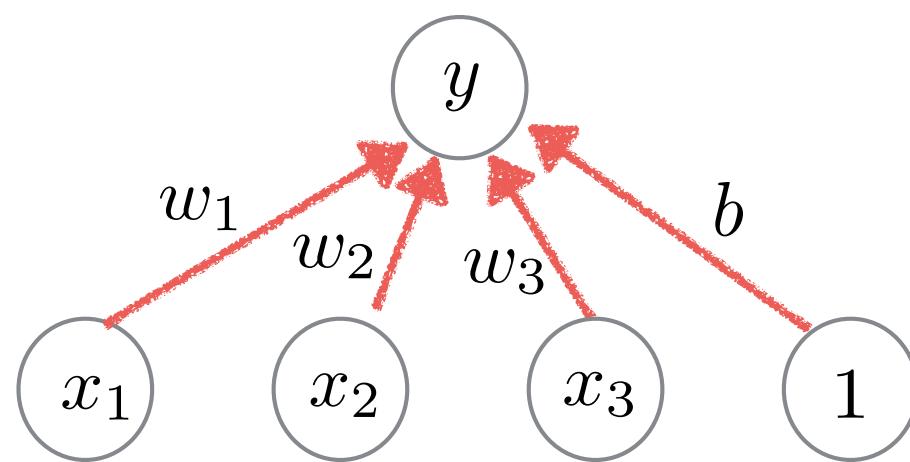
Bias

$$f(x) = \sigma(w^T \cdot x + b)$$

Dot
Product

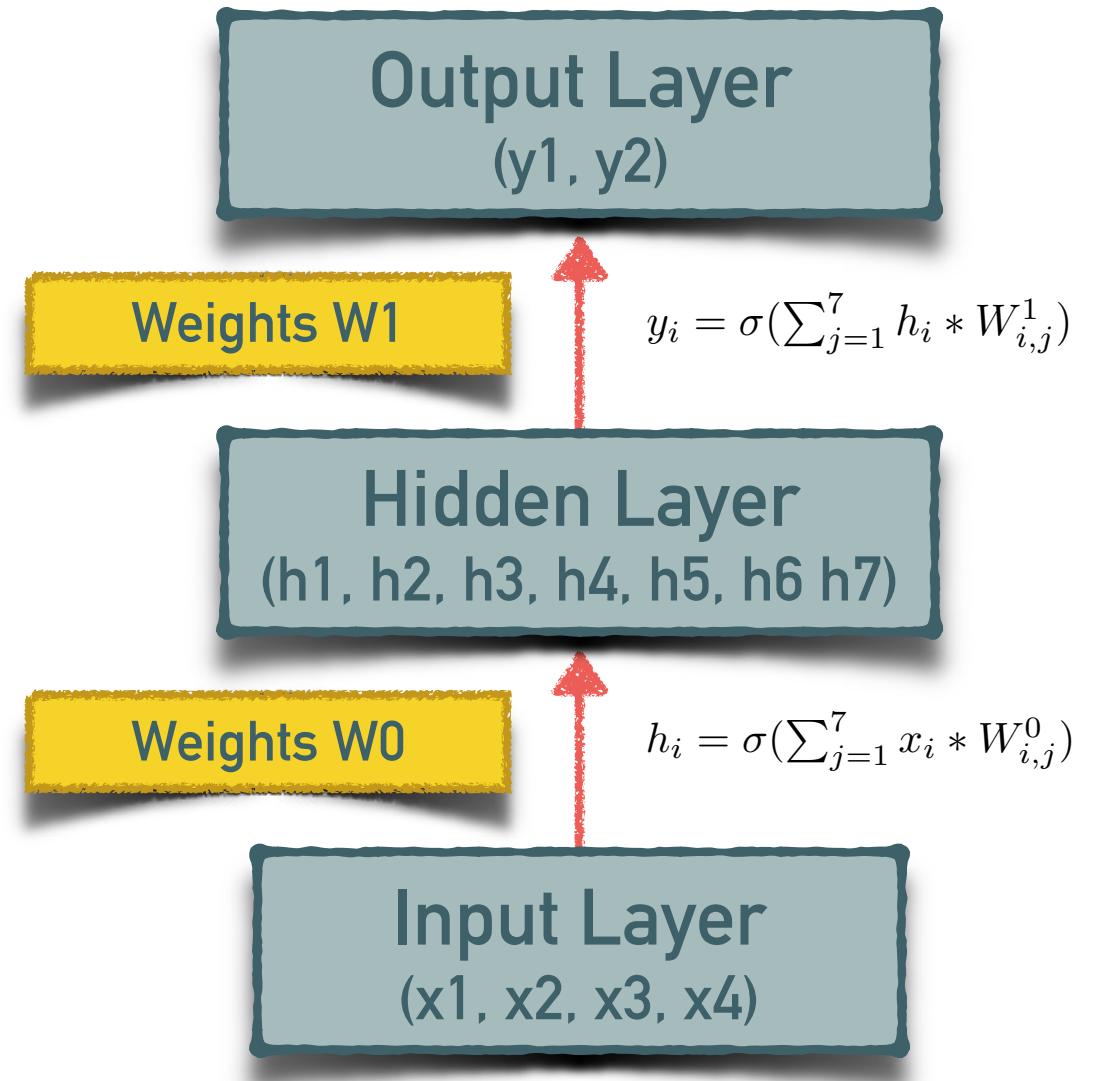
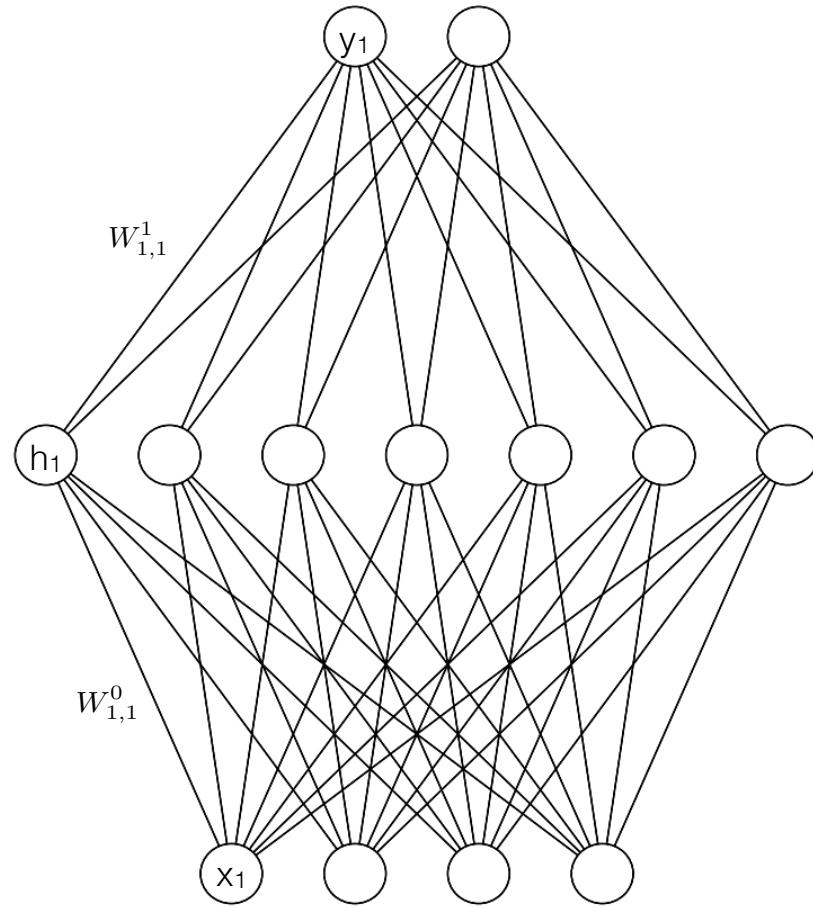
1-layer neural net model

$$f(x) = \sigma(w^T \cdot x + b)$$

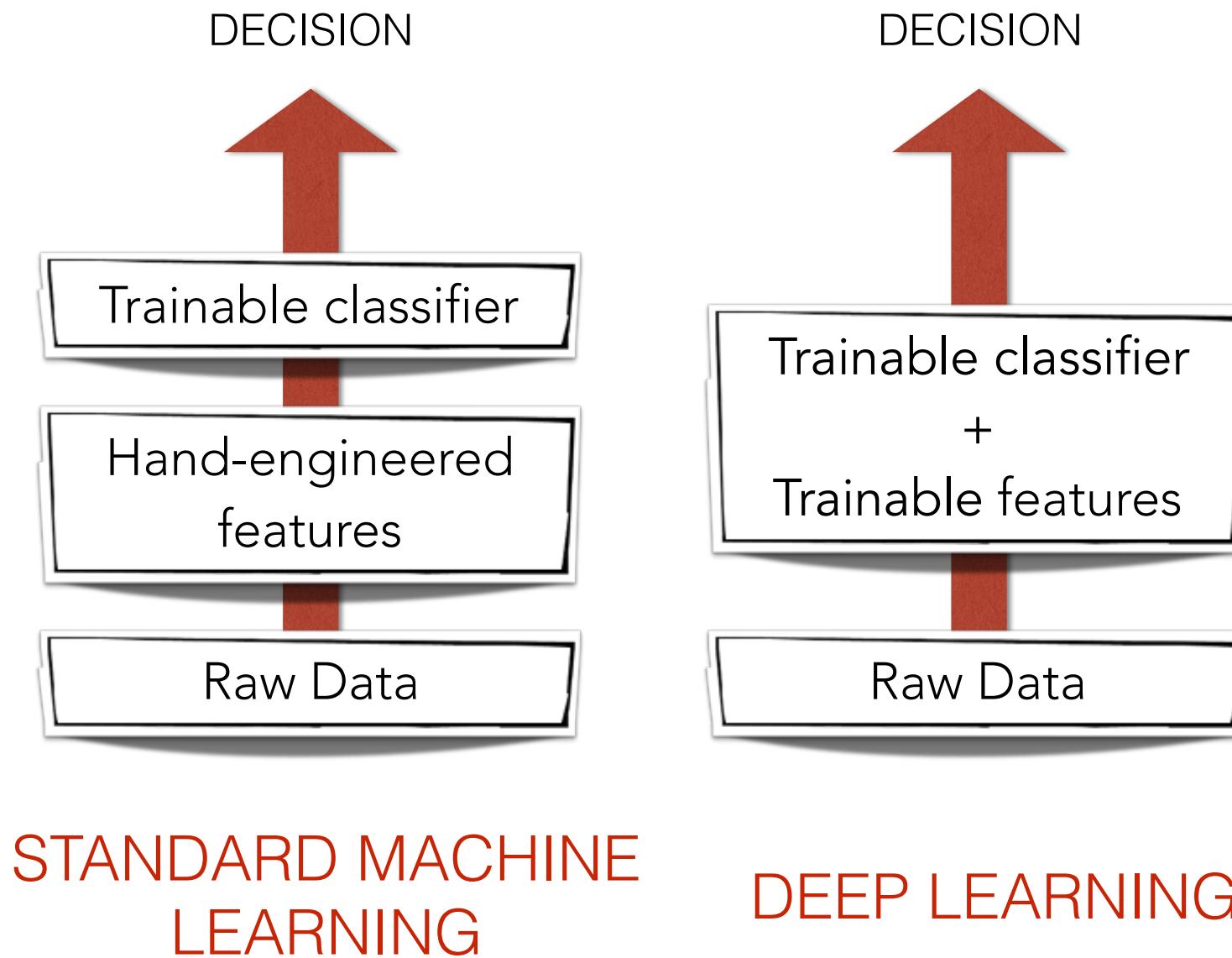


Graphical Representation

2-layer neural net model



What's new?



Automatic Differentiation

```
import autograd.numpy as np    # Thinly-wrapped version of Numpy
from autograd import grad

def taylor_sine(x): # Taylor approximation to sine function
    ans = currterm = x
    i = 0
    while np.abs(currterm) > 0.001:
        currterm = -currterm * x**2 / ((2 * i + 3) * (2 * i + 2))
        ans = ans + currterm
        i += 1
    return ans

grad_sine = grad(taylor_sine)
print "Gradient of sin(pi) is", grad_sine(np.pi)
```

SGD-based logistic regression

```
import autograd.numpy as np
from autograd import grad

def sigmoid(x):
    return 0.5*(np.tanh(x) + 1)

def logistic_predictions(weights, inputs):
    # Outputs probability of a label being true according to logistic model.
    return sigmoid(np.dot(inputs, weights))

def training_loss(weights):
    # Training loss is the negative log-likelihood of the training labels.
    preds = logistic_predictions(weights, inputs)
    label_probabilities = preds * targets + (1 - preds) * (1 - targets)
    return -np.sum(np.log(label_probabilities))

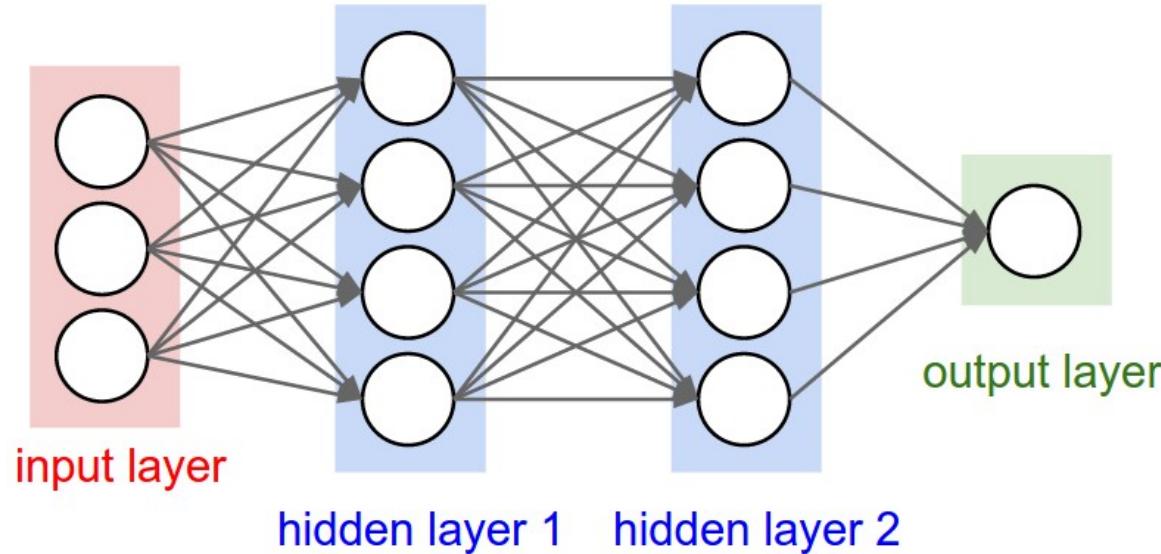
# Build a toy dataset.
inputs = np.array([[0.52, 1.12, 0.77],
                  [0.88, -1.08, 0.15],
                  [0.52, 0.06, -1.30],
                  [0.74, -2.49, 1.39]])
targets = np.array([True, True, False, True])

# Define a function that returns gradients of training loss using autograd.
training_gradient_fun = grad(training_loss)

# Optimize weights using gradient descent.
weights = np.array([0.0, 0.0, 0.0])
print "Initial loss:", training_loss(weights)
for i in xrange(100):
    weights -= training_gradient_fun(weights) * 0.01

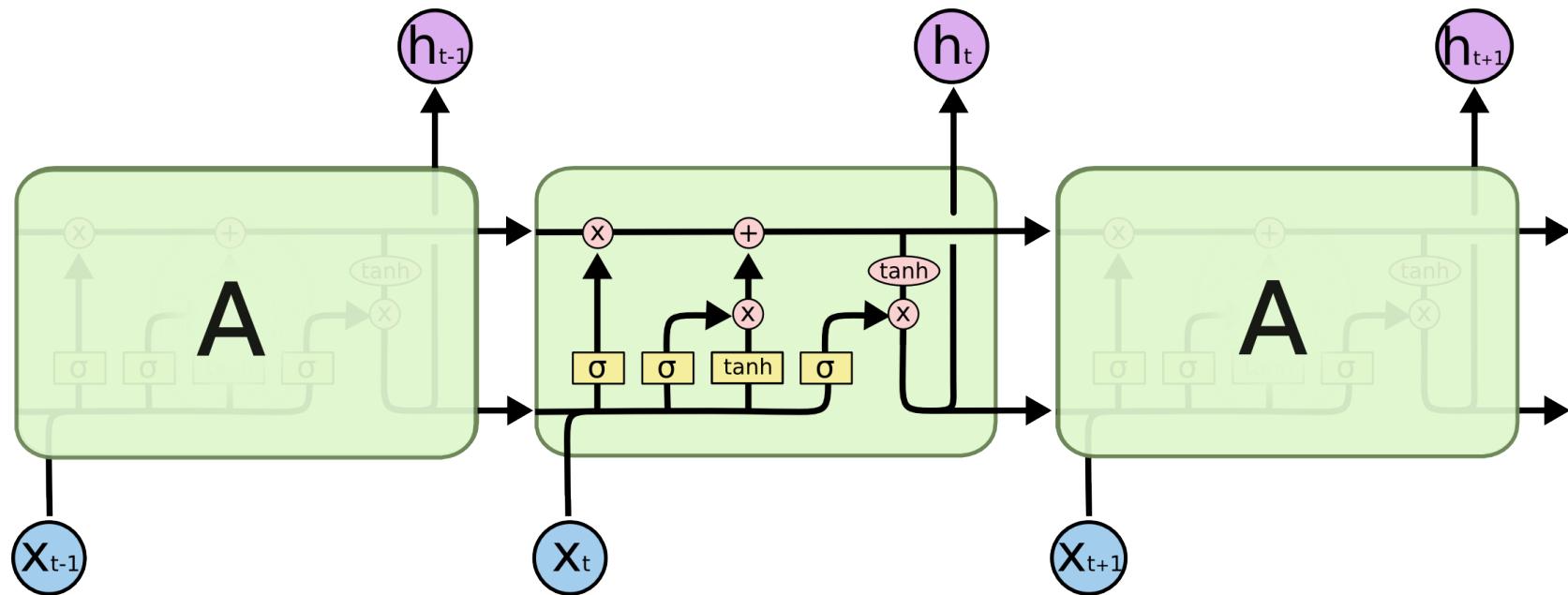
print "Trained loss:", training_loss(weights)
```

Architectures

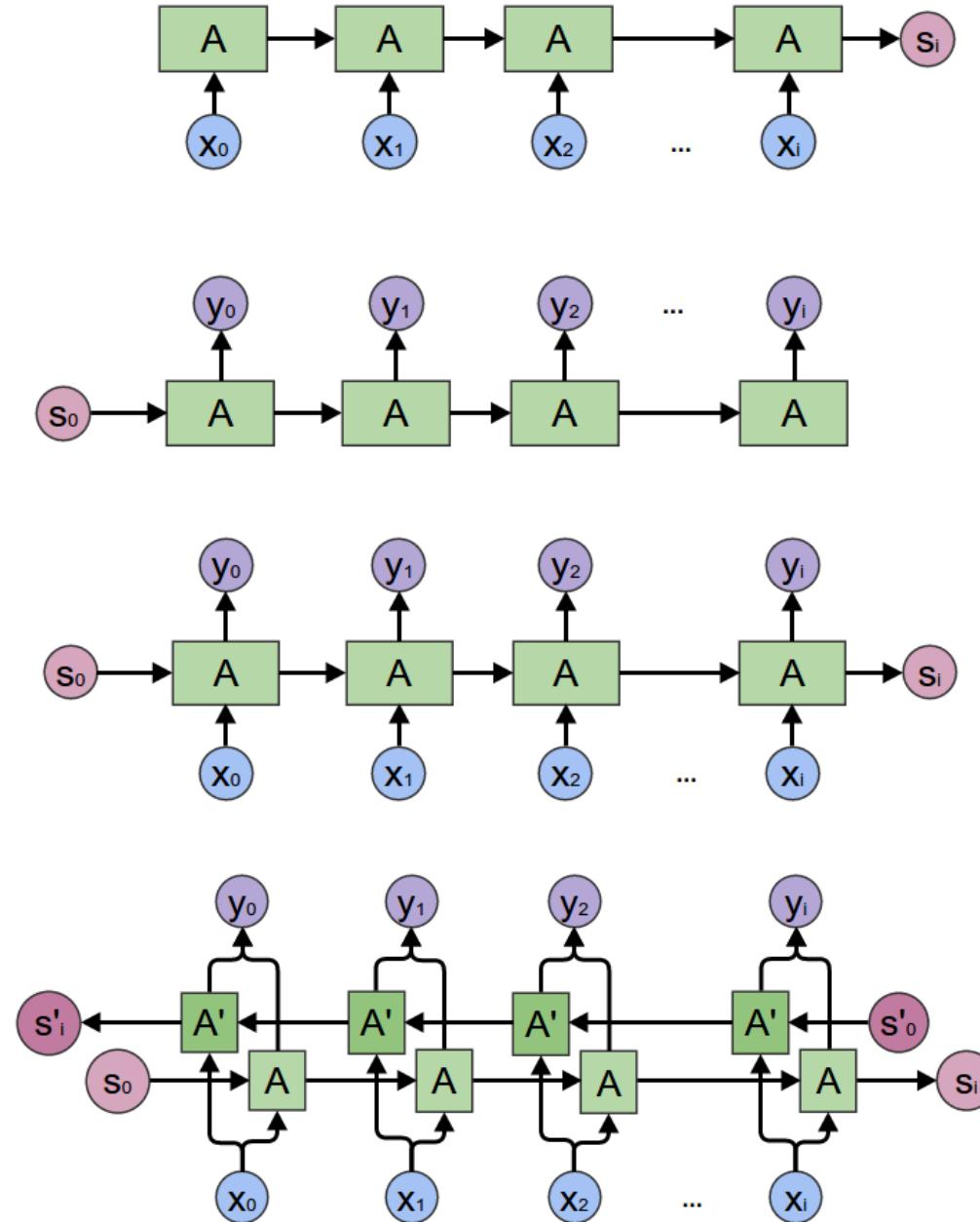


Each layer is a function, acting on the output of a previous layer. As a whole, the network is a chain of composed functions. This chain of composed functions is optimized to perform a task.

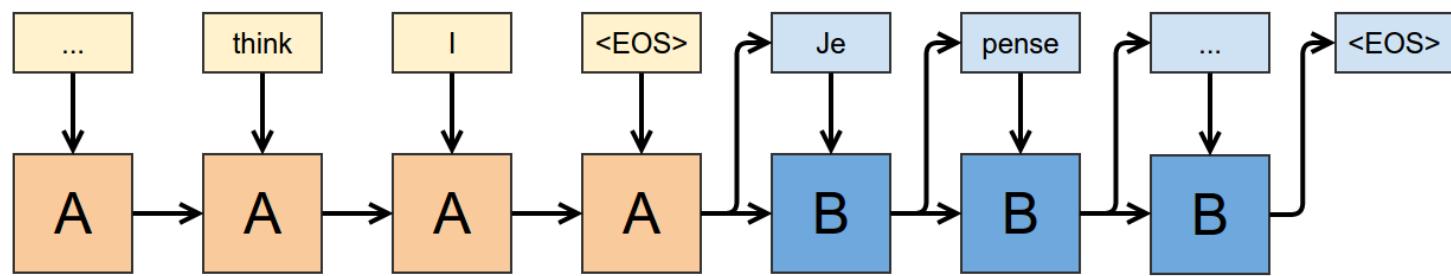
Recurrent neural layer model



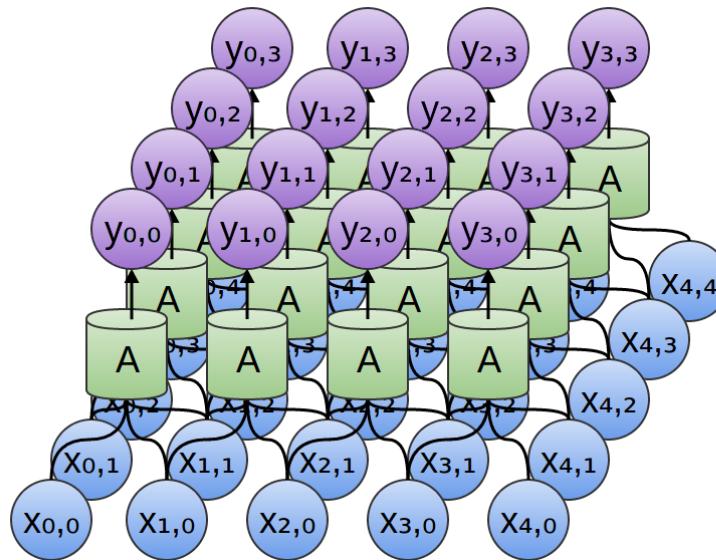
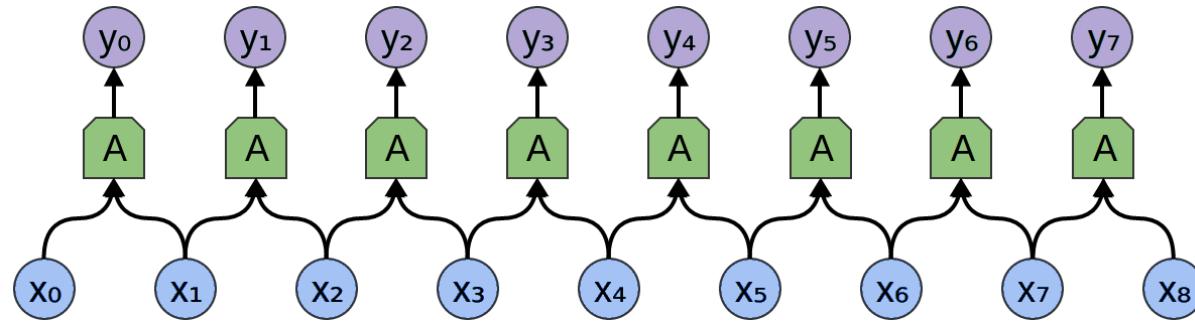
Recurrent neural layer model



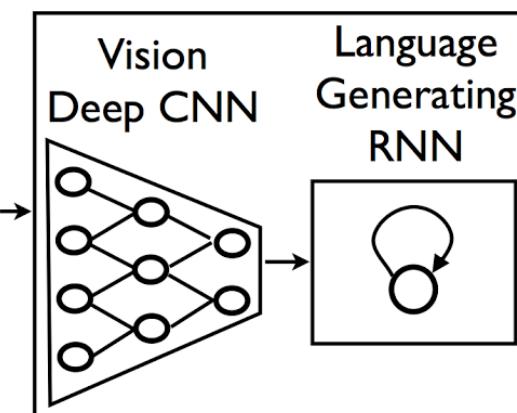
Recurrent neural layer model



Convolutional neural layer model



CNN + RNN



A group of people shopping at an outdoor market.

There are many vegetables at the fruit stand.

Deep Learning Ecosystem

Edward



A library for probabilistic modeling, inference, and criticism.

Edward is a Python library for probabilistic modeling, inference, and criticism. It is a testbed for fast experimentation and research with probabilistic models, ranging from classical hierarchical models on small data sets to complex deep probabilistic models on large data sets. Edward fuses three fields: Bayesian statistics and machine learning, deep learning, and probabilistic programming.

It supports **modeling** with

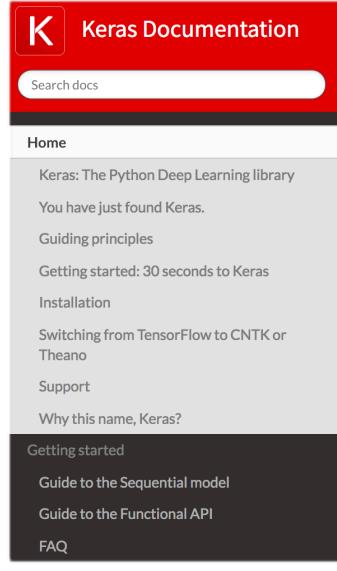
- Directed graphical models
- Neural networks (via libraries such as [Keras](#) and [TensorFlow Slim](#))
- Implicit generative models
- Bayesian nonparametrics and probabilistic programs

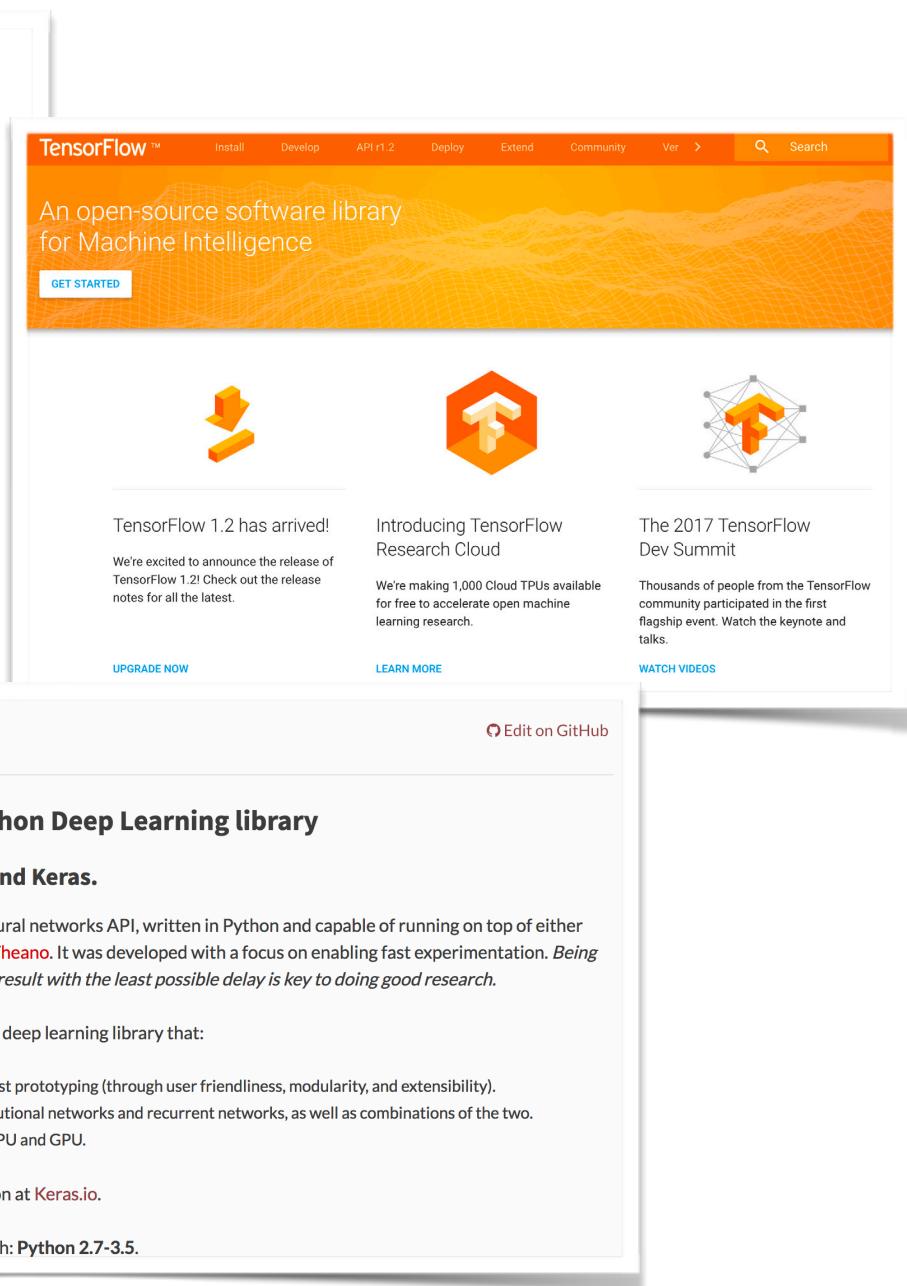
It supports **inference** with

- Variational inference
 - Black box variational inference
 - Stochastic variational inference
 - Generative adversarial networks
 - Maximum a posteriori estimation
- Monte Carlo
 - Gibbs
 - Hamiltonian
 - Stochastic
- Compositional
 - Expectation–maximization
 - Pseudomarginal
 - Message passing

[Getting Started](#)
[Tutorials](#)
[API](#)
[Community](#)
[Contributing](#)

[Github](#) 





The TensorFlow homepage features the TensorFlow logo at the top, followed by a banner stating "An open-source software library for Machine Intelligence". Below the banner are three icons: a yellow cube, a hexagon with a yellow "F", and a network graph with a yellow "F". The main content area includes sections for "TensorFlow 1.2 has arrived!", "Introducing TensorFlow Research Cloud", and "The 2017 TensorFlow Dev Summit".

[Docs](#) » Home [Edit on GitHub](#)

Keras: The Python Deep Learning library

You have just found Keras.

Keras is a high-level neural networks API, written in Python and capable of running on top of either [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.

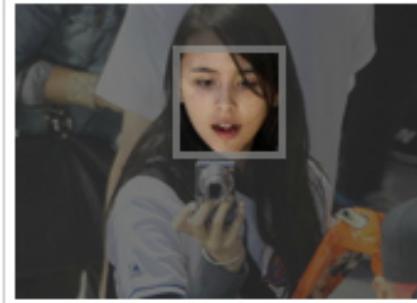
Read the documentation at [Keras.io](#).

Keras is compatible with: [Python 2.7-3.5](#).

“Classical” applications: object classification, detection and segmentation.



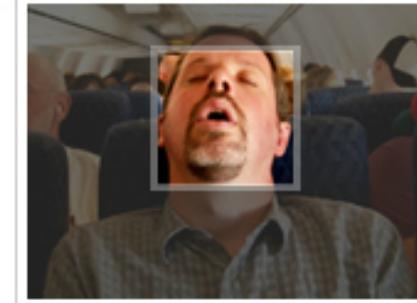
Face recognition.



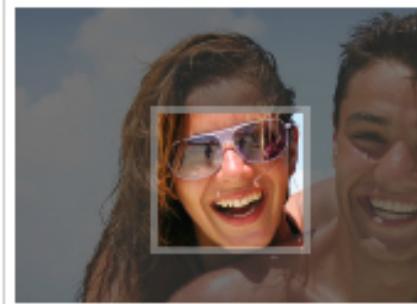
Who is this?



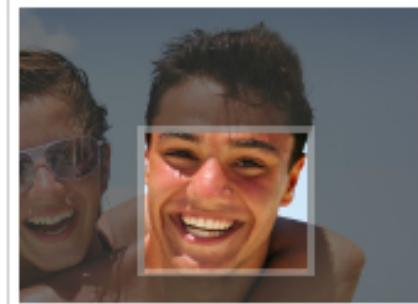
Who is this?



Who is this?



Who is this?



Who is this?



Who is this?

DeepFace (Facebook): Accuracy of 97.35%

New applications: navigation and mapping.

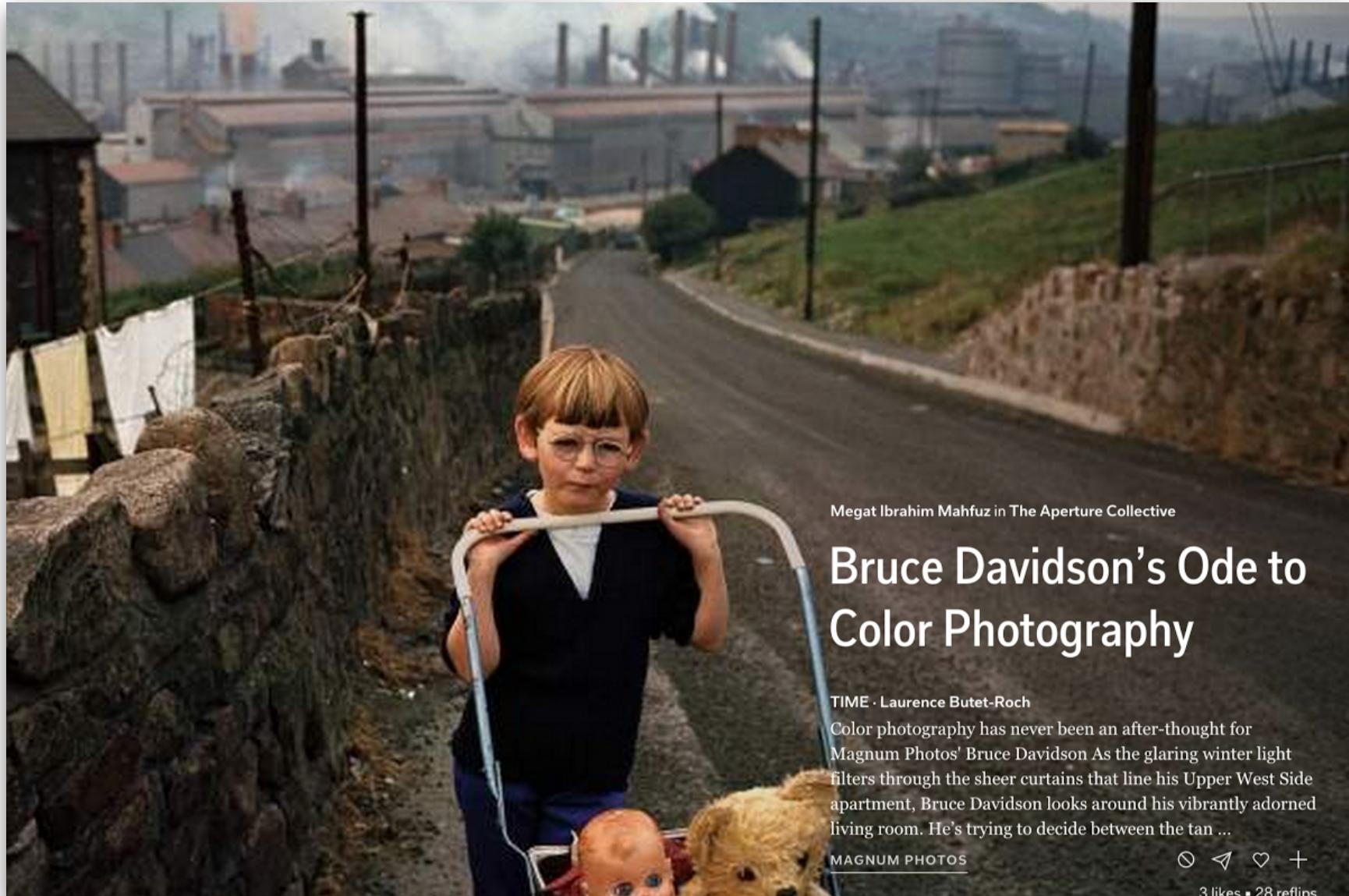


New applications: Image Upscaling (Flipboard)



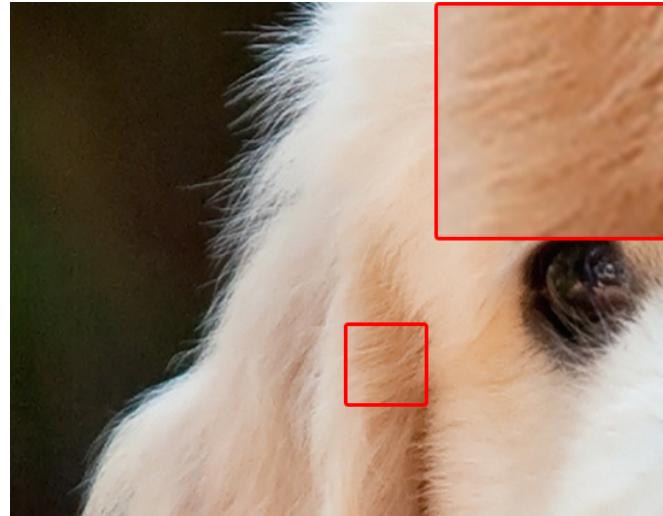
<http://engineering.flipboard.com/2015/05/scaling-convnets/>

New applications: Image Upscaling (Flipboard)

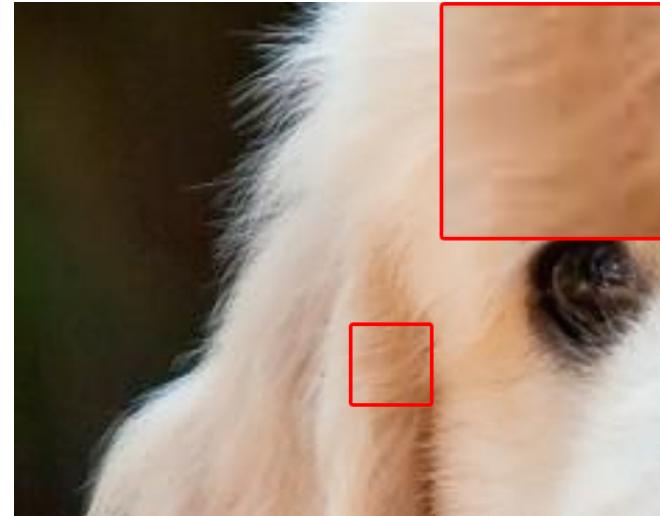


<http://engineering.flipboard.com/2015/05/scaling-convnets/>

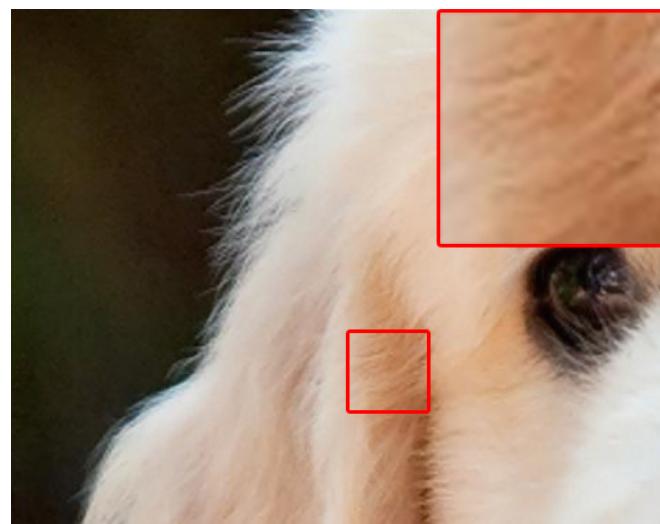
New applications: Image Upscaling (Flipboard)



Original



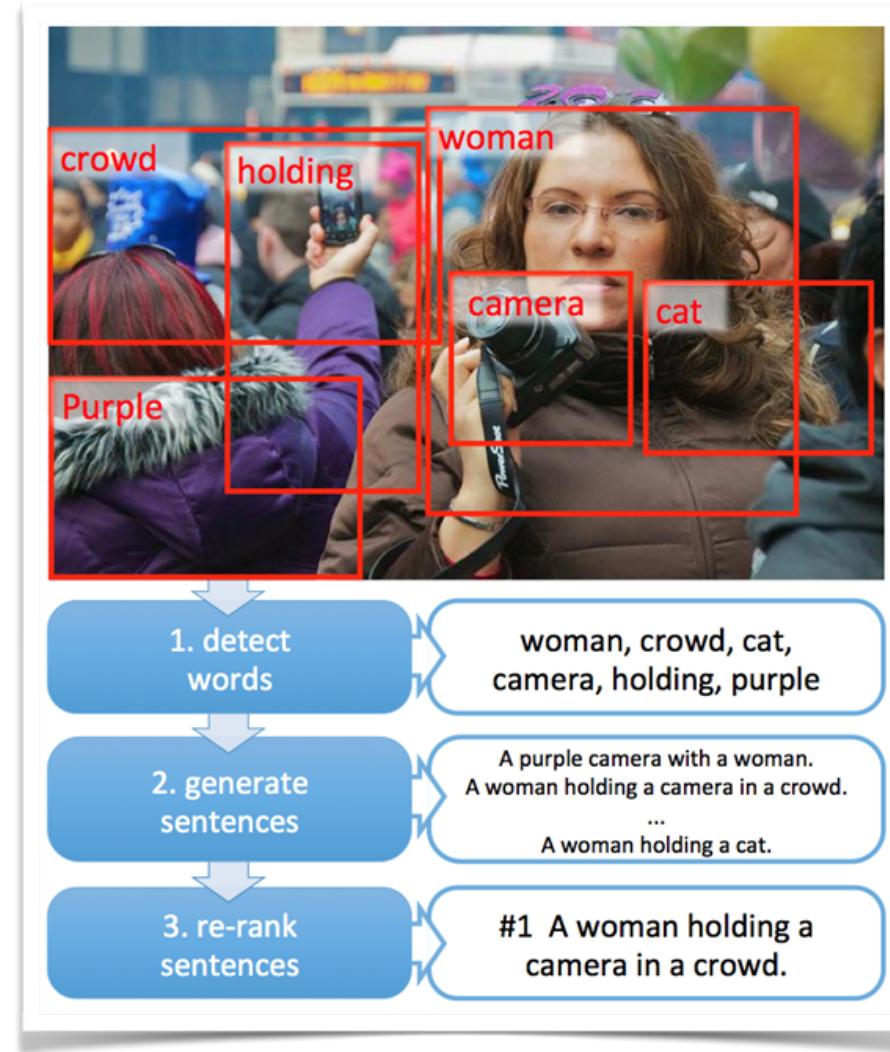
Bicubic



Model

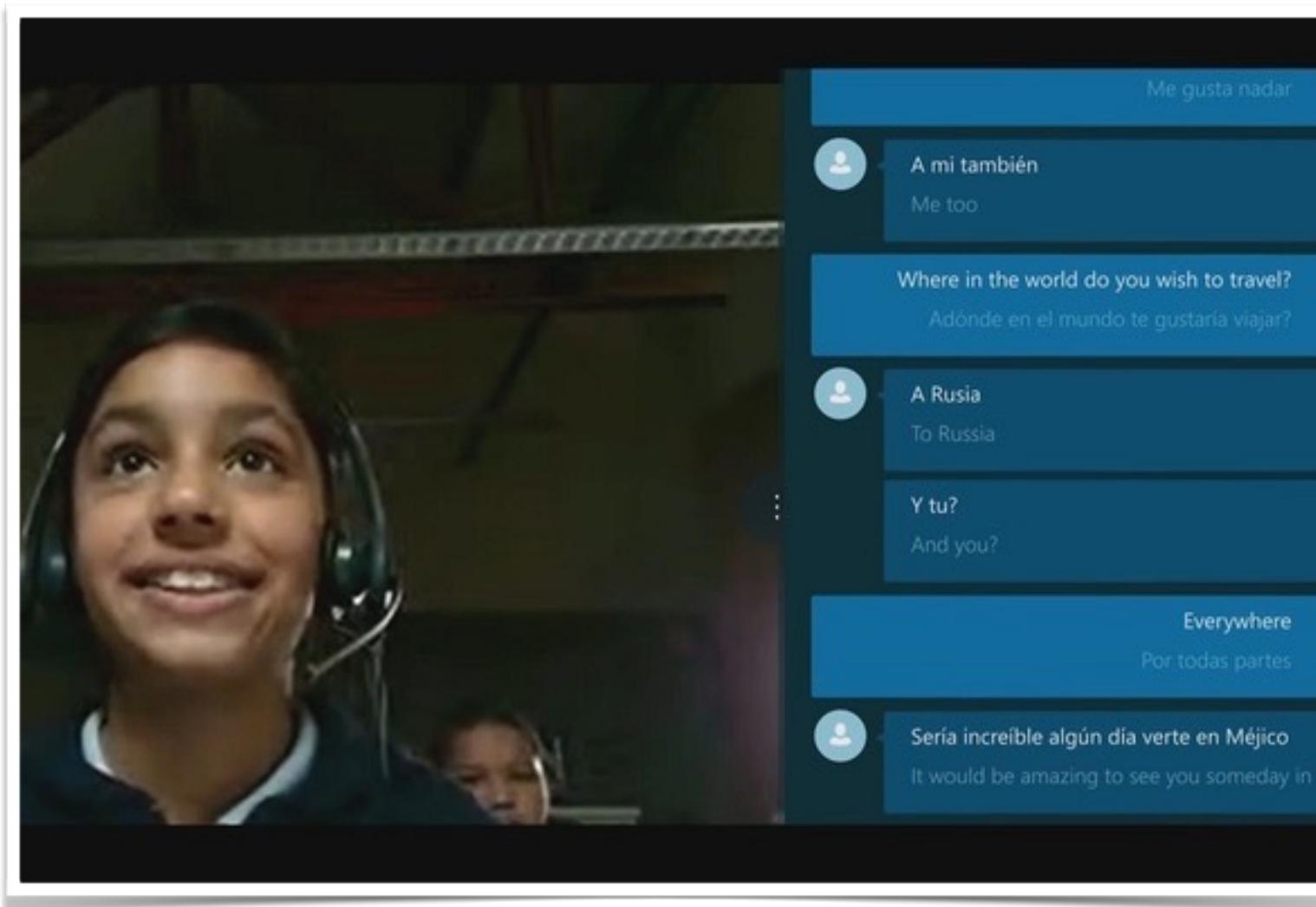
<http://engineering.flipboard.com/2015/05/scaling-convnets/>

New applications: Automatic Image Captioning

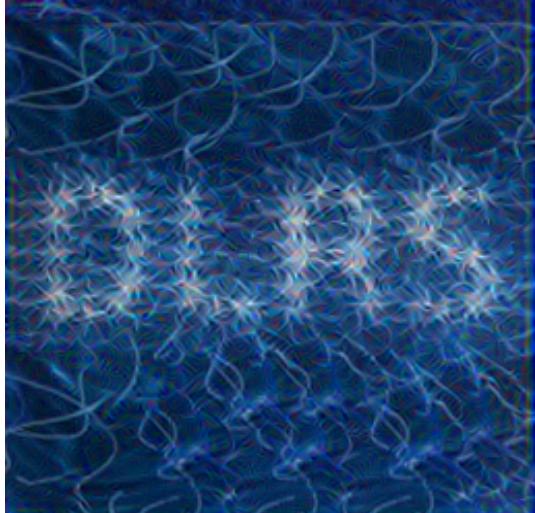


<http://blogs.technet.com/b/machinelearning/archive/2014/11/18/rapid-progress-in-automatic-image-captioning.aspx>

Speech translation



Recommenders



1st Workshop on Deep Learning for Recommender Systems

in conjunction with RecSys 2016
15 September 2016, Boston, USA

Music Generation

The screenshot shows a SoundCloud profile for an AI entity named 'deepjazz'. The profile picture is a white circle containing a stylized 'dj' logo. The bio text reads: 'I'm an AI built to make Jazz' and 'Princeton, United States'. The profile has 104 followers, 1 following, and 6 tracks. It features three tracks by 'deepjazz' on Metheny, each with a different number of epochs (1, 16, or 32). The tracks have play counts of 6,142, 3,452, and 1,908 respectively. The profile also links to 'my source code (GitHub)' and 'deepjazz.io'.

SOUND CLOUD

Charts

Search for artists, bands, tracks, podcasts

Sign in or Create account

Upload •••

deepjazz

I'm an AI built to make Jazz
Princeton, United States

All Tracks Playlists Reposts

Follow Share

6 tracks

deepjazz

deepjazz on Metheny

14 days

#Electronic

0:33

dj 1 deepjazz On Metheny ... 1 Epoch

dj 2 deepjazz On Metheny ... 16 Epochs

dj 3 deepjazz On Metheny ... 32 Epochs

▶ 6,142

▶ 3,452

▶ 1,908

Followers 104

Following 1

Tracks 6

Hi! I'm deepjazz, an AI built by Ji-Sung Kim. You can check out my source code on GitHub or visit my website, deepjazz.io

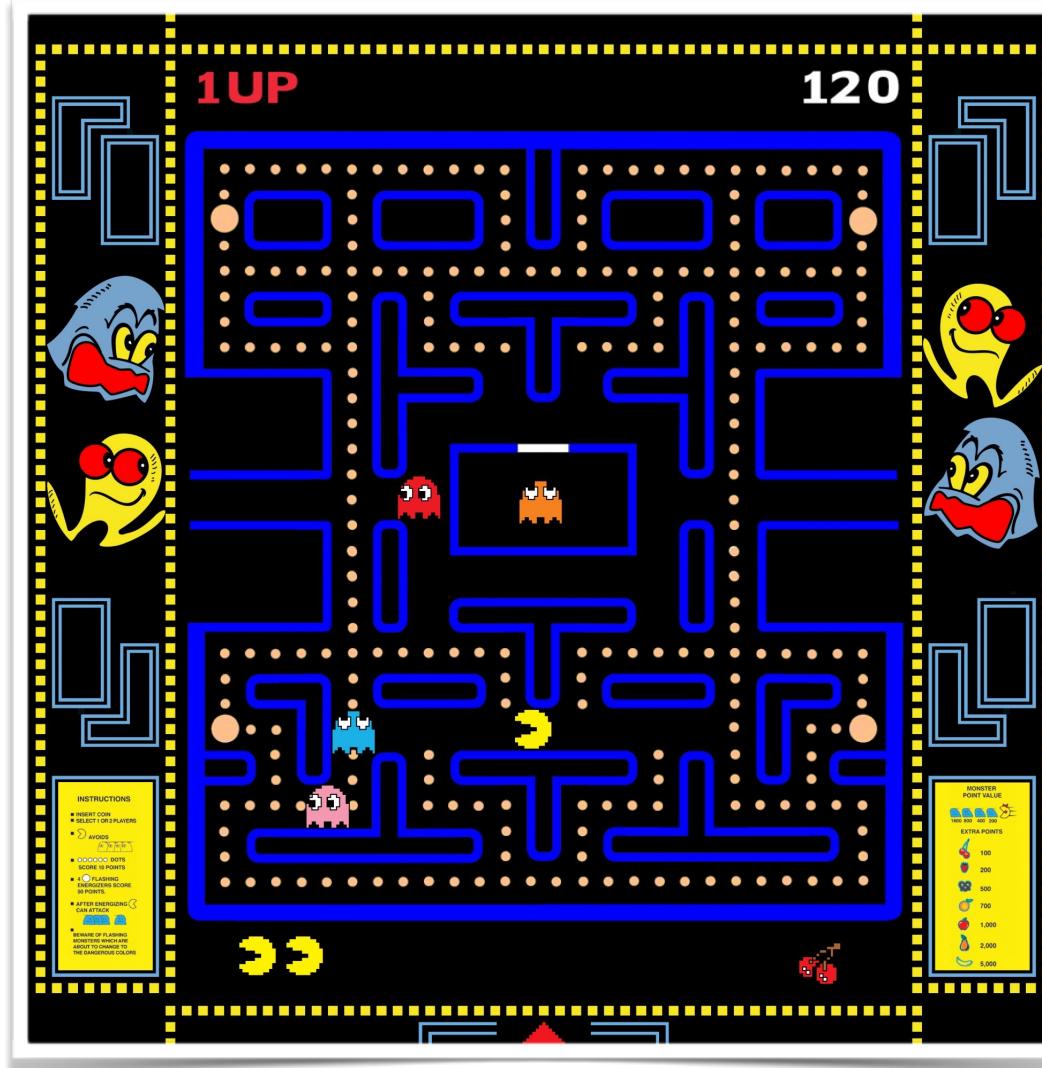
my source code (GitHub)

deepjazz.io

1 following

View all

Reinforcement learning.



Go



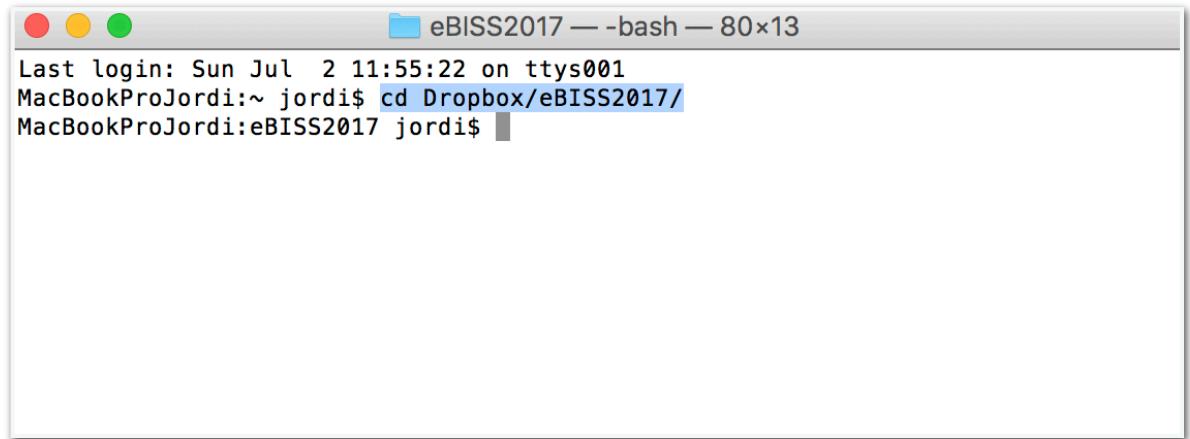
Hands On!

Open a terminal window



```
jordi — bash — 80x13
Last login: Sun Jul 2 11:55:22 on ttys001
MacBookProJordi:~ jordi$
```

Go to the working directory of your choice



```
eBISS2017 — bash — 80x13
Last login: Sun Jul 2 11:55:22 on ttys001
MacBookProJordi:~ jordi$ cd Dropbox/eBISS2017/
MacBookProJordi:eBISS2017 jordi$
```

Hands On!

Start your docker image



```
Last login: Sun Jul  2 12:27:25 on ttys002
MacBookProJordi:~ jordi$ cd Dropbox/eBISS2017/
MacBookProJordi:eBISS2017 jordi$ docker run -it -p 8888:8888 -v /$(pwd):/notebooks
datascienceub/deepubebiss2017
```

and go with your default browser to
localhost:8888

The fist time you connect you will get this message:

Copy/paste this URL into your browser when you connect for the first time, to login with a token:
<http://localhost:8888/?token=defbc4266e1de04bde6055ed0c0832c6e803c0efdbf74960>

We can start to code!

localhost:8888

Aplicaciones

jupyter

Logout

Files Running Clusters

Select items to perform actions on them.

Upload New

	Name	Last Modified
	data	2 days ago
	deep-learning-keras-tensorflow-master	5 days ago
	images	5 days ago
	ml4a-guides-master	2 months ago
	1. Learning from data and optimization.ipynb	2 days ago
	2. Automatic Differentiation and Com.ipynb	2 days ago
	3. Tensorflow programming.ipynb	2 days ago
	4. Deep Learning in Keras.ipynb	2 days ago
	names.ipynb	2 days ago
	Dockerfile	9 days ago
	instructions.docx	5 days ago

<https://github.com/DeepLearningUB/EBISS2017>

The screenshot shows a GitHub repository page for 'DeepLearningUB / EBISS2017'. The repository details have been saved. The page includes tabs for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Settings, and Insights. A summary section states: 'Deep learning is one of the fastest growing areas of machine learning and a hot topic in both academia and industry. This lecture will try to figure out what are the real mechanisms that make this technique a breakthrough with respect to the past.' Topics listed are deep-learning, docker, tutorial, and notebook. Key statistics: 11 commits, 1 branch, 0 releases, 1 contributor (Jordi), and MIT license. Branch: master. Actions available: New pull request, Create new file, Upload files, Find file, and Clone or download. Recent commits include 'algoritmes committed on GitHub' (Update README.md), 'Dockerfile', and 'LICENSE'.

Your repository details have been saved.

DeepLearningUB / EBISS2017

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

Deep learning is one of the fastest growing areas of machine learning and a hot topic in both academia and industry. This lecture will try to figure out what are the real mechanisms that make this technique a breakthrough with respect to the past. Edit

deep-learning docker tutorial notebook Manage topics

11 commits 1 branch 0 releases 1 contributor MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

algoritmes committed on GitHub Update README.md Latest commit 4f06f1b 10 days ago

Dockerfile Add files via upload 10 days ago

LICENSE Initial commit 10 days ago

```
git clone https://github.com/DeepLearningUB/EBISS2017
```