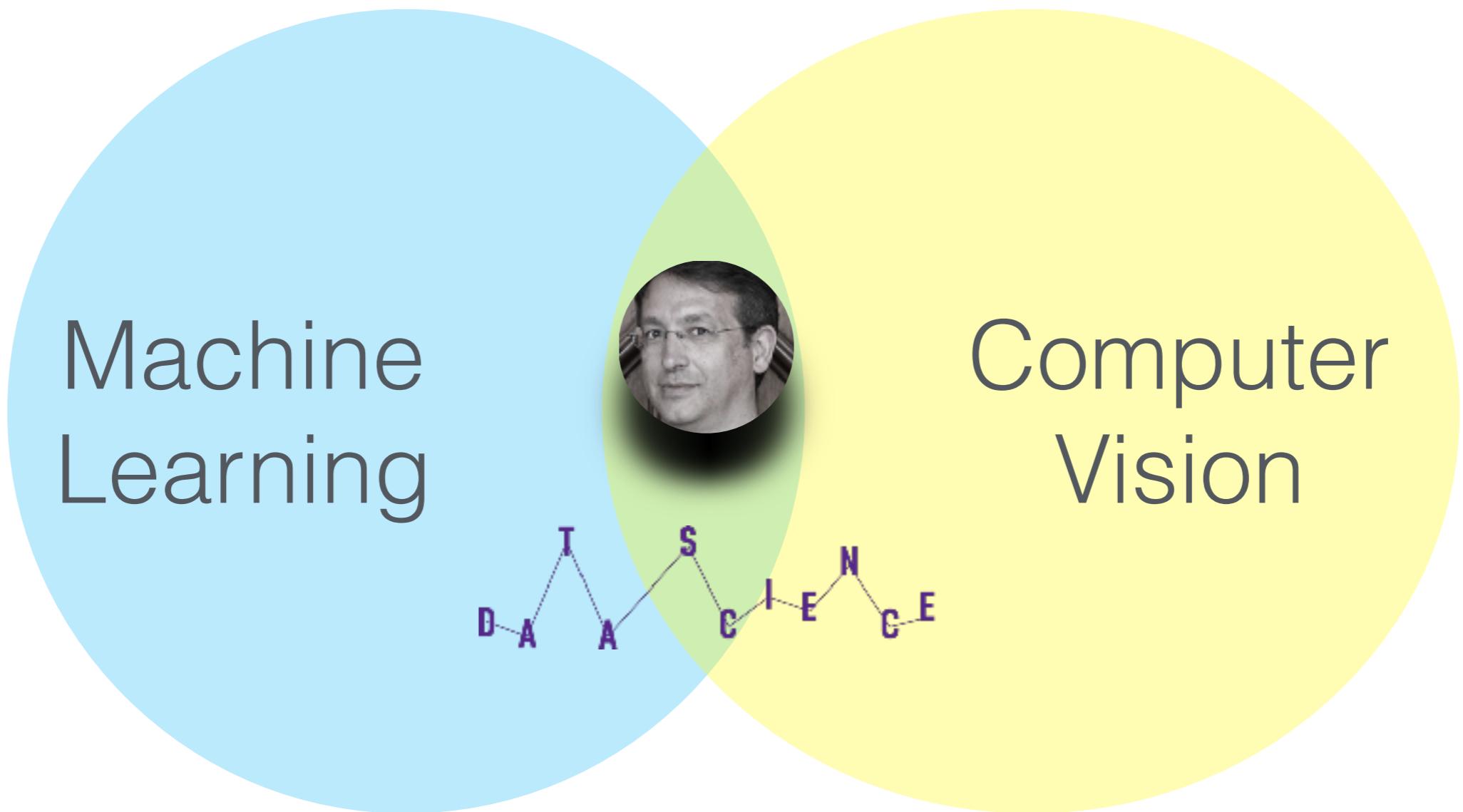




Images from <https://distill.pub/2017/feature-visualization/>

LET'S OPEN THE DEEP LEARNING BLACK BOX!

JORDI VITRIÀ



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.

What's in the picture?



“Dog leaps to catch frisbee”

Can you readily infer attributes from these images such as ‘noisy’, ‘walkable’, ‘healthy’, “safe”?

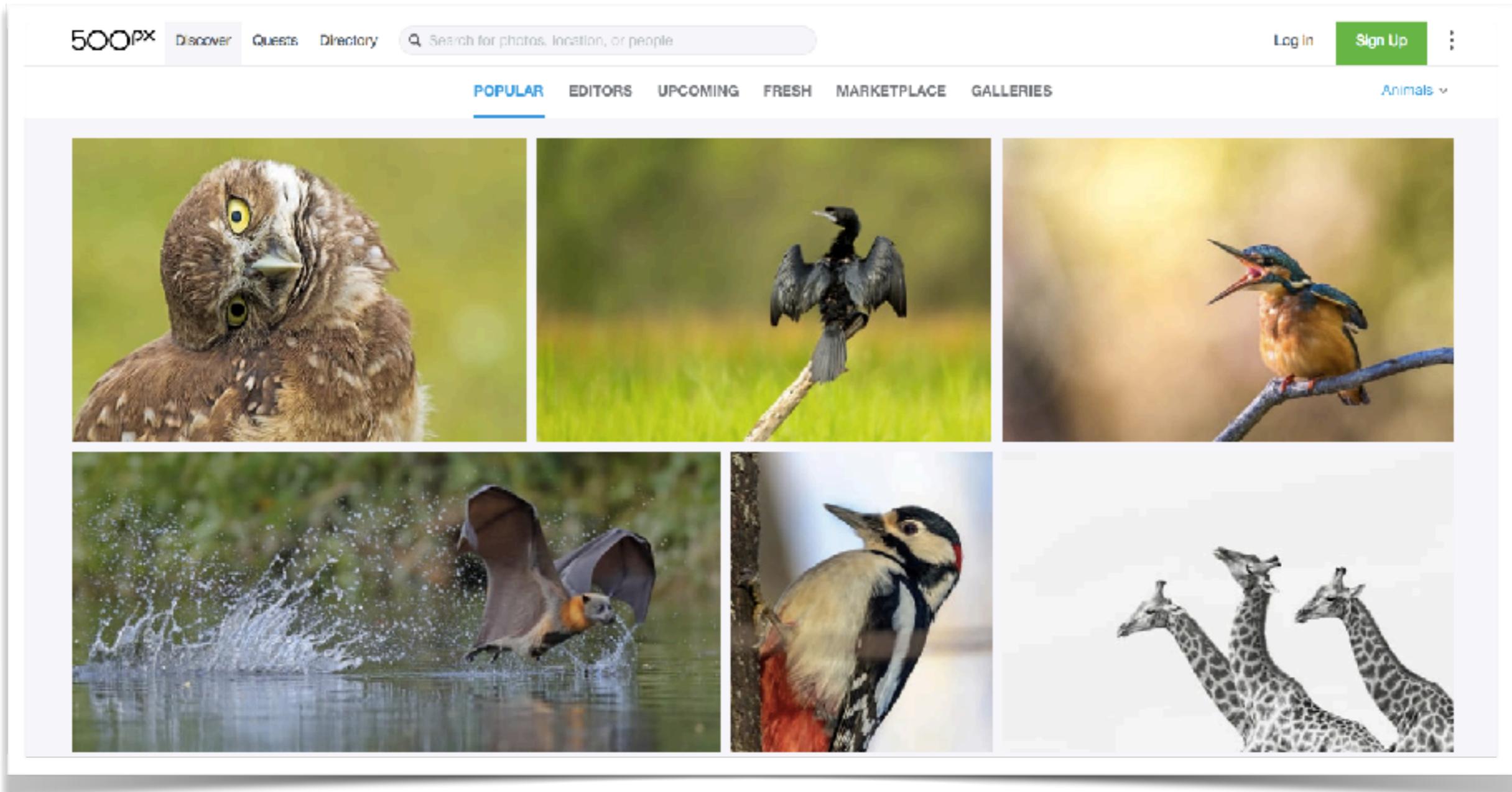
Google

© 2016 Google



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 consists of five news snippets arranged in a grid-like fashion:

- BBC News Business:** A video thumbnail titled "What is 'deep learning'?" featuring a man in a suit standing outdoors.
- Financial Times:** An article titled "The mind in the machine: Demis Hassabis on artificial intelligence" by FT Senior Writer of Science, featuring a circular graphic of glowing dots.
- Nature:** An article titled "All systems go" by Mark Henderson, featuring a computer processor on a circuit board.
- New York Times Tech & Science:** An article titled "Google is teaching a free online class on 'deep learning'" by Seung Lee, featuring a colorful illustration of a DeepDream image.
- The Economist Special Report:** An article titled "From not working to neural networking" by Michael J. I. Brown, featuring a robotic arm interacting with a screen.

A large, stylized eye icon is positioned in the bottom right corner of the collage.

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

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



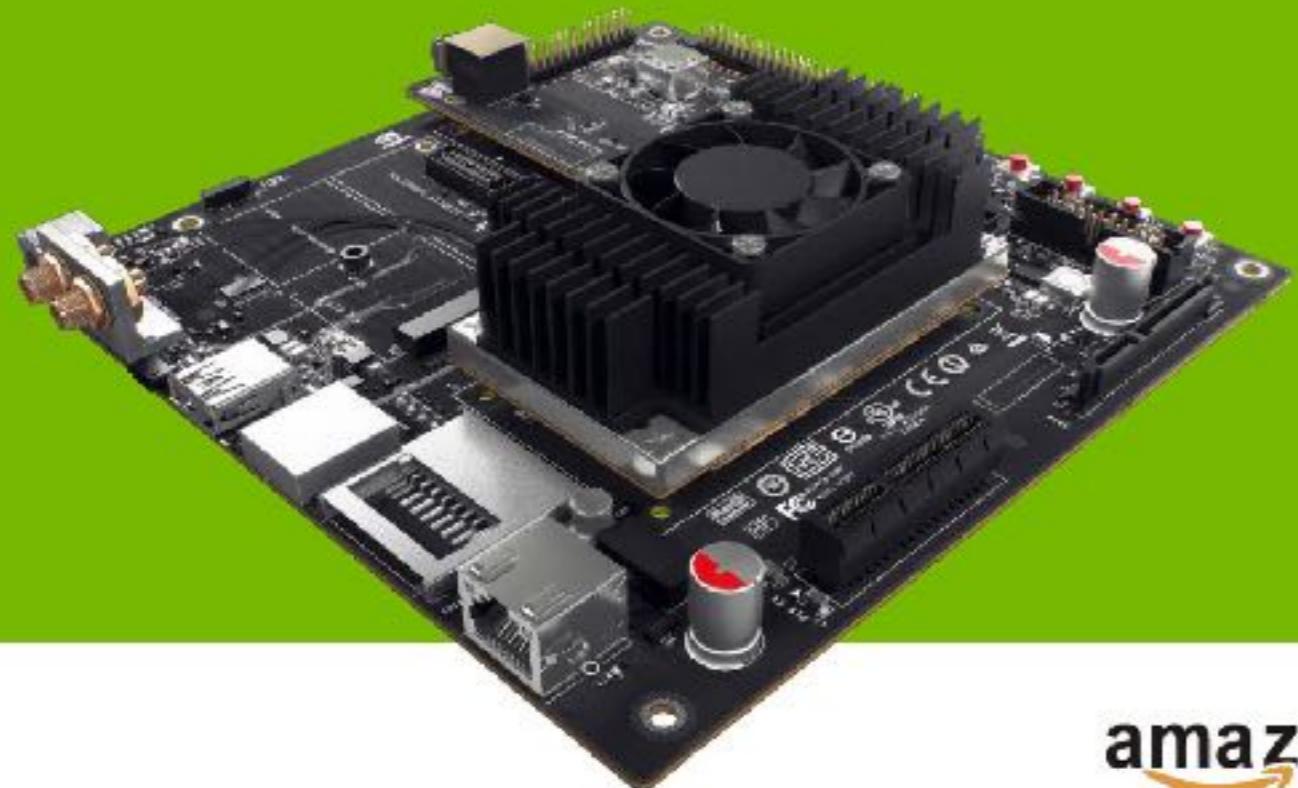


- In 1982, interest in the field was renewed. **John Hopfield** of Caltech presented a paper to the National Academy of Sciences. His approach was to create more useful machines by using bidirectional lines. Previously, the connections between neurons was only one way.
- In 1986, the problem was how 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

amazon

newegg.com[®]

MICRO CENTER
computers & electronics

NVIDIA

GPU democratization

The screenshot shows the NVIDIA Accelerated Computing website. The top navigation bar includes links for Downloads, Training, Ecosystem, and Forums, along with a search icon and a Log in button. The main content area has a dark header with the text "GPU Grant Program". Below this, a breadcrumb navigation shows "Home > ComputeWorks > Academic Collaboration > GPU Grant Program". The main text describes NVIDIA's Academic Programs Team's mission to empower professors and researchers worldwide through various initiatives like GPU grants, fellowships, and developer forums. It also outlines the requirements for GPU Grant Requests, including contact information, research descriptions, team details, and publication lists.

NVIDIA ACCELERATED COMPUTING Downloads Training Ecosystem Forums

GPU Grant Program

Home > ComputeWorks > Academic Collaboration > GPU Grant Program

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

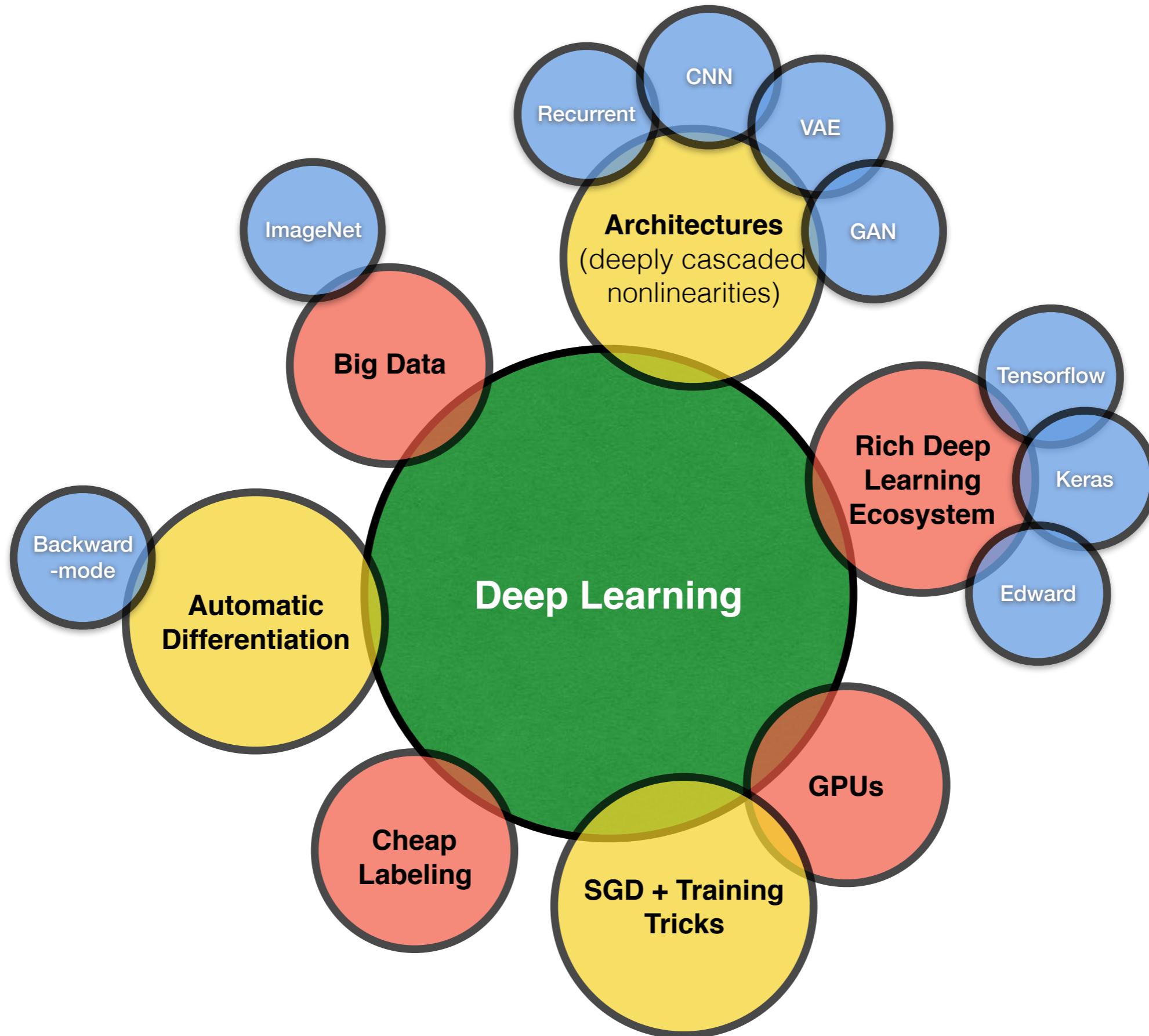
170TF | “250 servers in-a-box” | nvidia.com/dgx1

\$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.

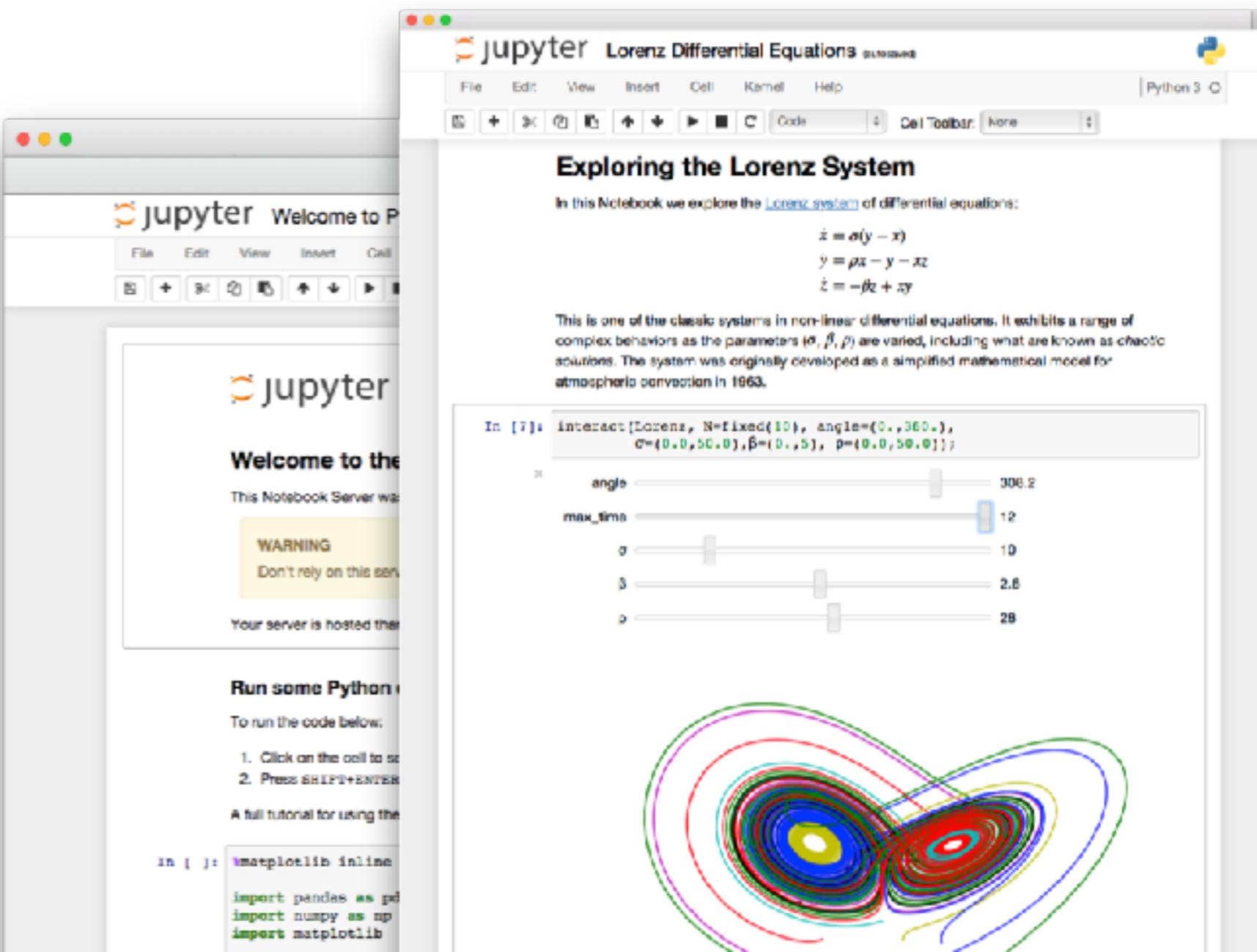


Our objectives

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

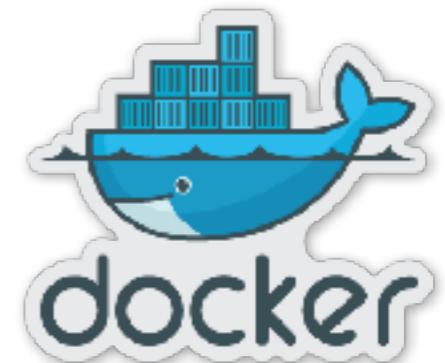
Approach

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**

Numeric features that characterize your cases

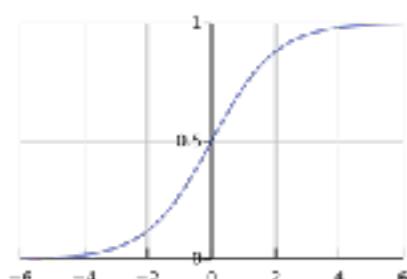
Your desired outcomes

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

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

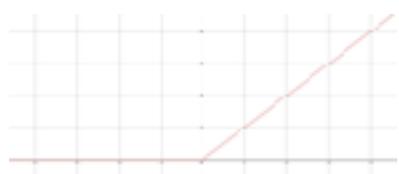
1. Choose a model function family f_w .
2. Optimize parameters w .

1-layer neural net model



Sigmoid
Function

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



ReLU
Function

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

Parameters



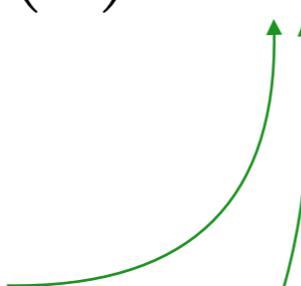
Weights



Bias

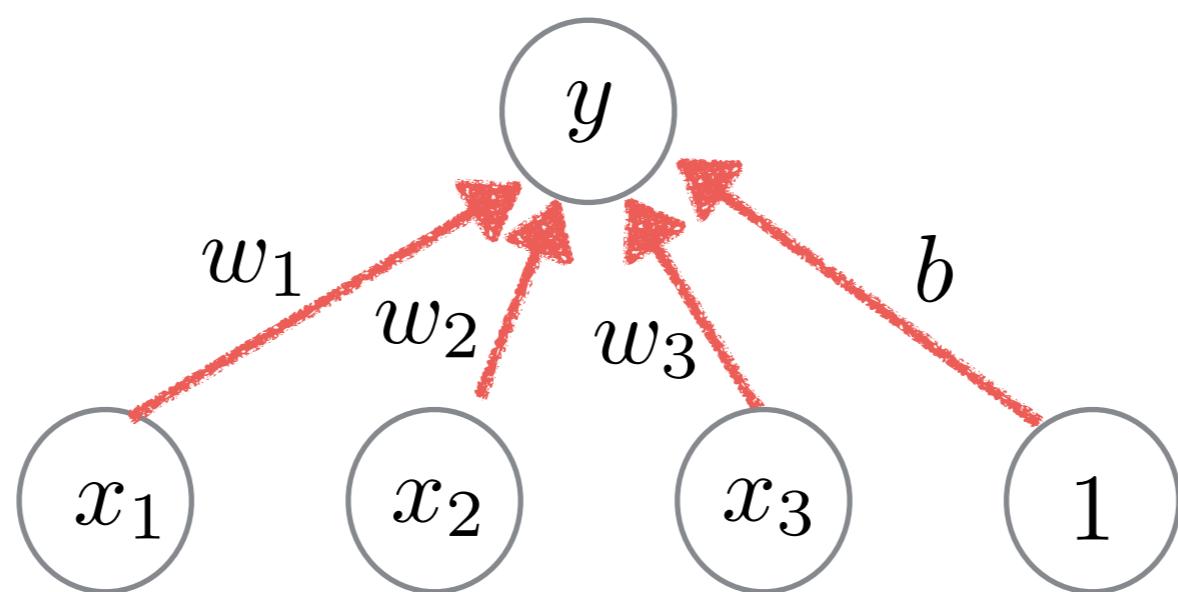
$$f(x) = \sigma(w^T \cdot x + b)$$

Dot
Product



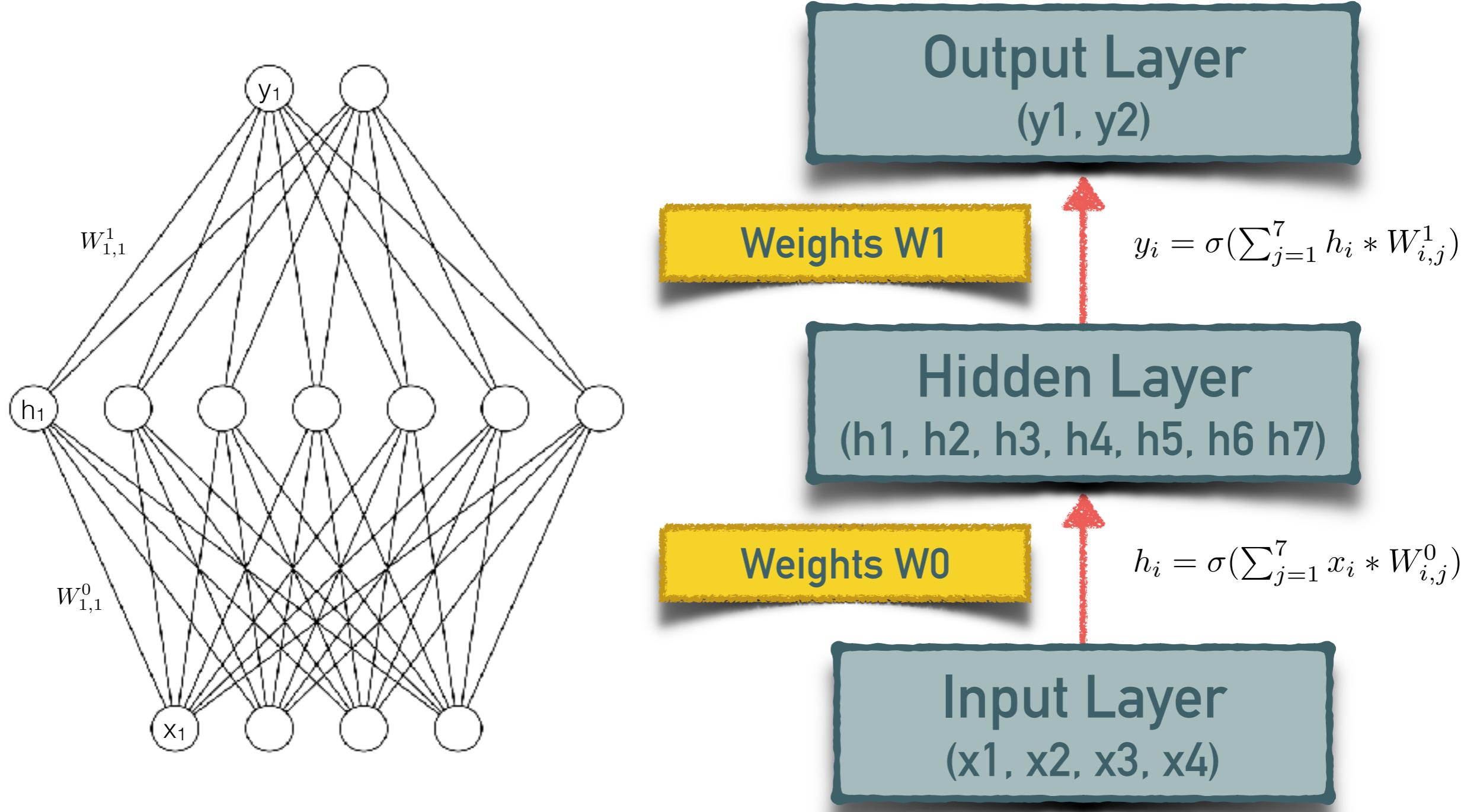
1-layer neural net model

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



Graphical Representation

2-layer neural net model



2-layer neural net model

- How to find the parameters of the function?
- We can use optimization techniques (minimizing a function, the **loss function**, that measures the discrepancy between the outcomes of a model and the desired outcomes.
- To optimize, we must compute the derivative of every parameter with respect to the loss function.
- But we have (possibly) millions of parameters and the loss function is a (possibly) large composition of functions...

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))

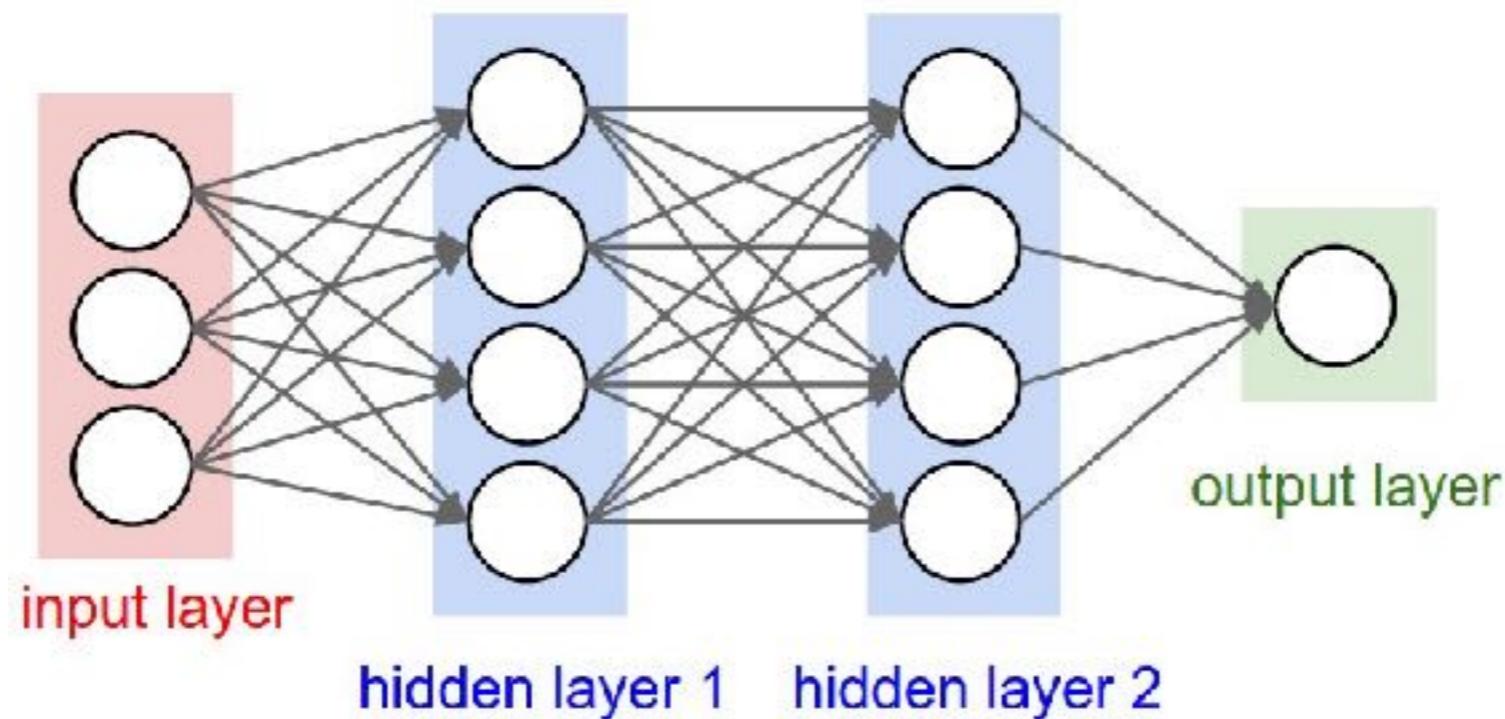
# Build a toy dataset.
inputs = np.array([[0.52, 1.12, 0.77],
                  [0.88, -1.08, 0.15],
                  [0.52, 0.06, -1.30],
                  [0.74, -2.49, 1.39]])
targets = np.array([True, True, False, True])

# 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

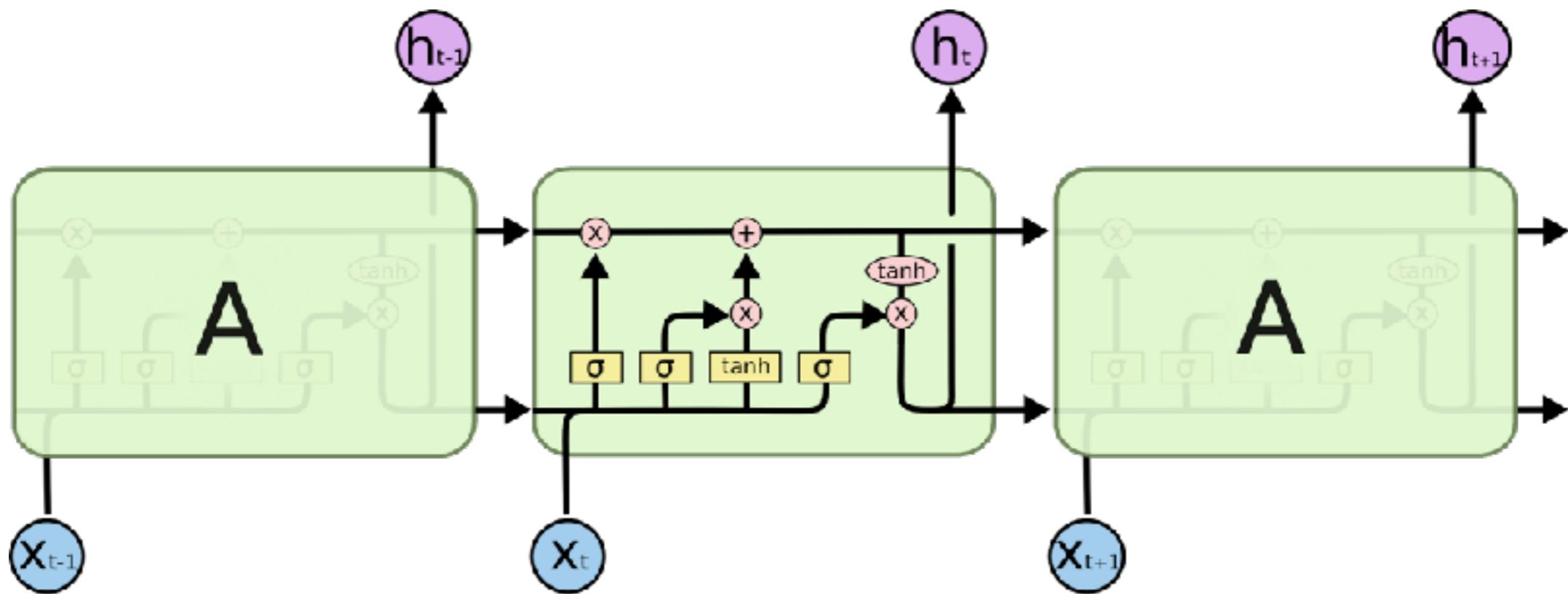
print "Trained loss:", training_loss(weights)
```

Architectures

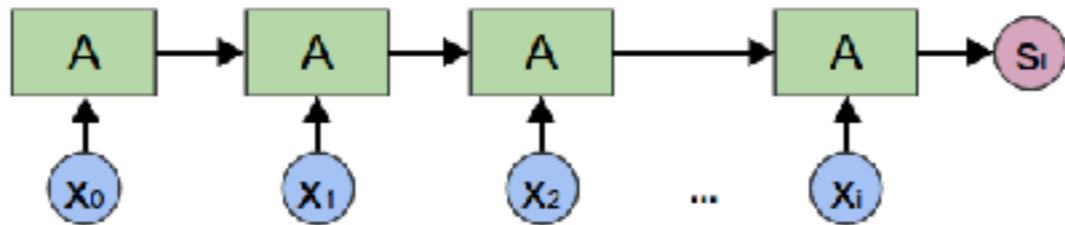


Each layer is a function, acting on the output of a previous layer. As a whole, the network is a chain of composed functions. This chain of composed functions is optimized to perform a task.

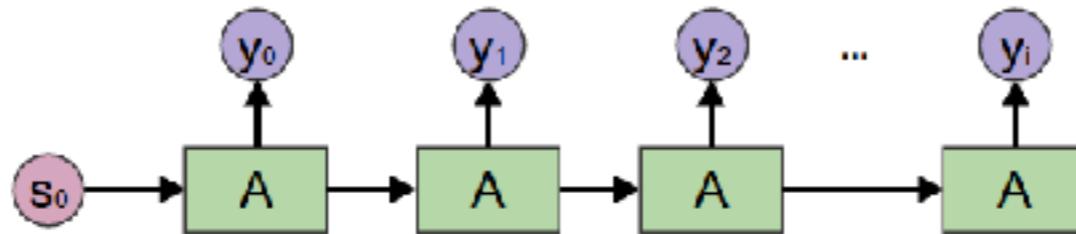
Recurrent neural layer model



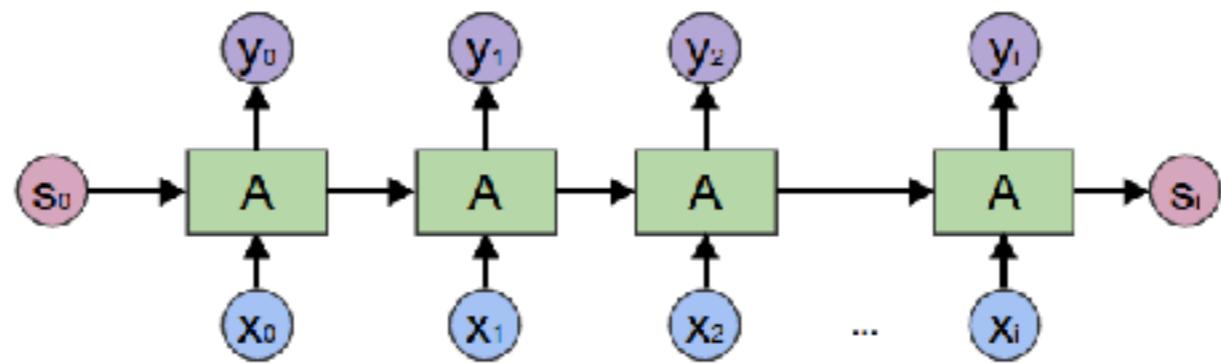
Recurrent neural layer model



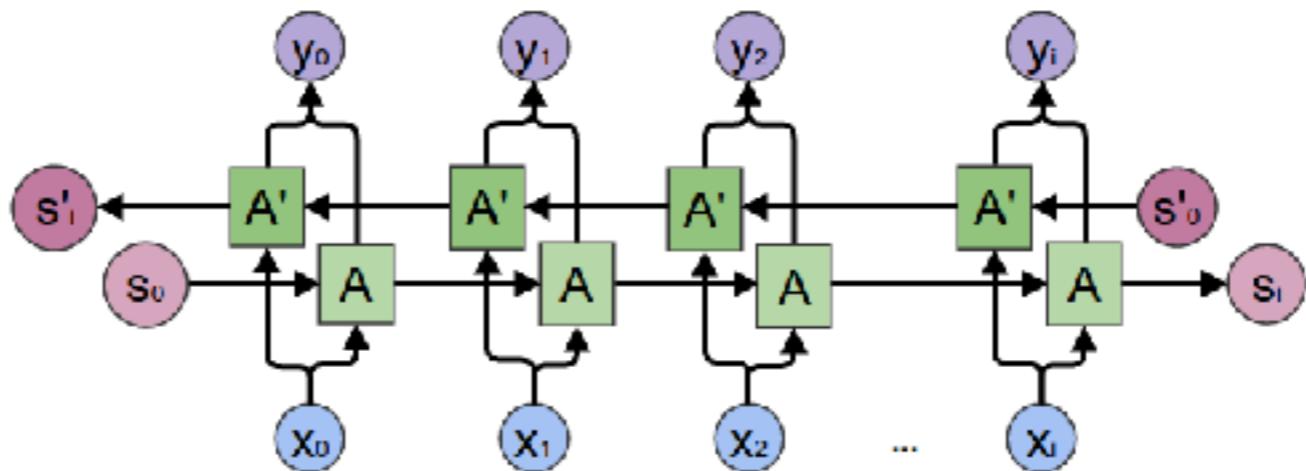
Encoder



Decoder

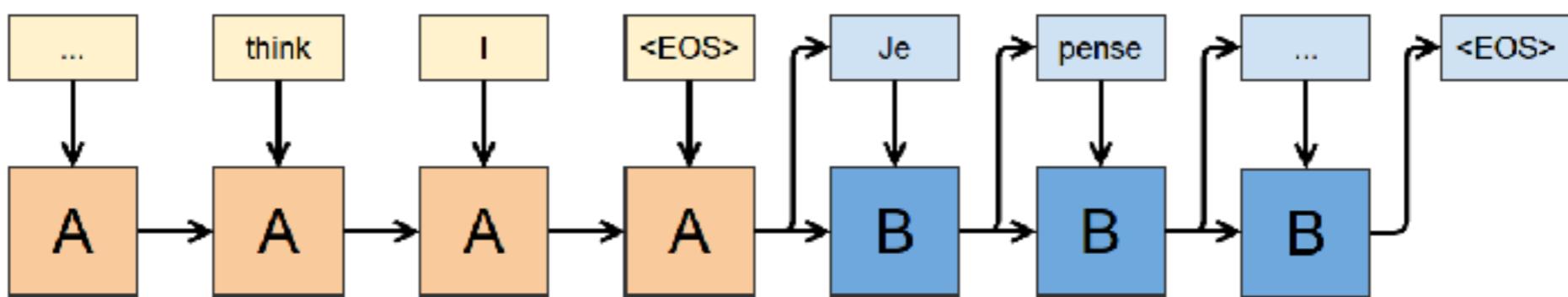


Encoder-Decoder

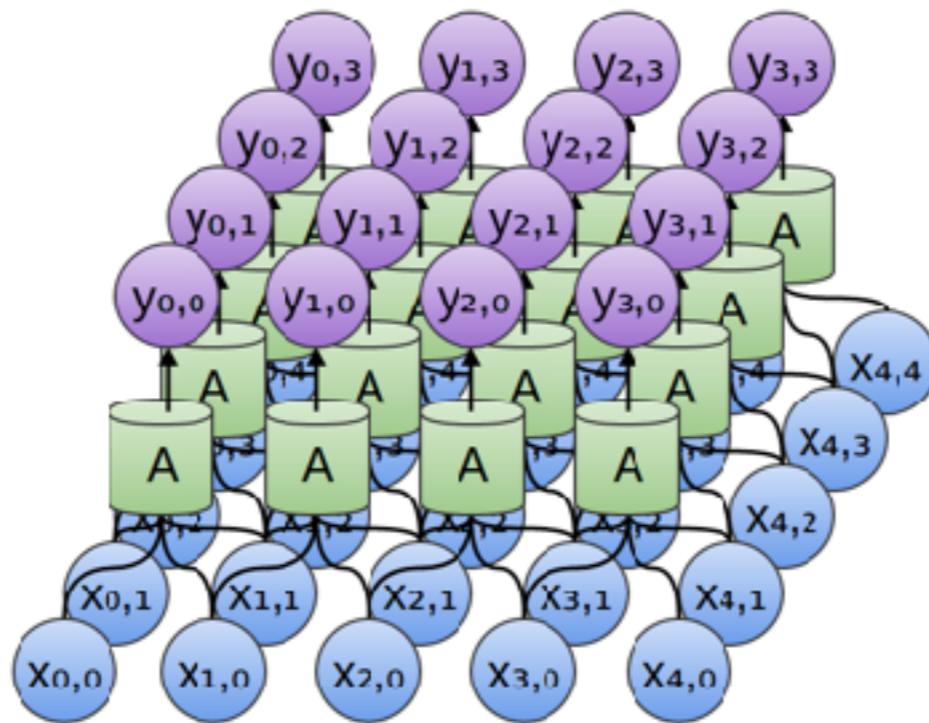
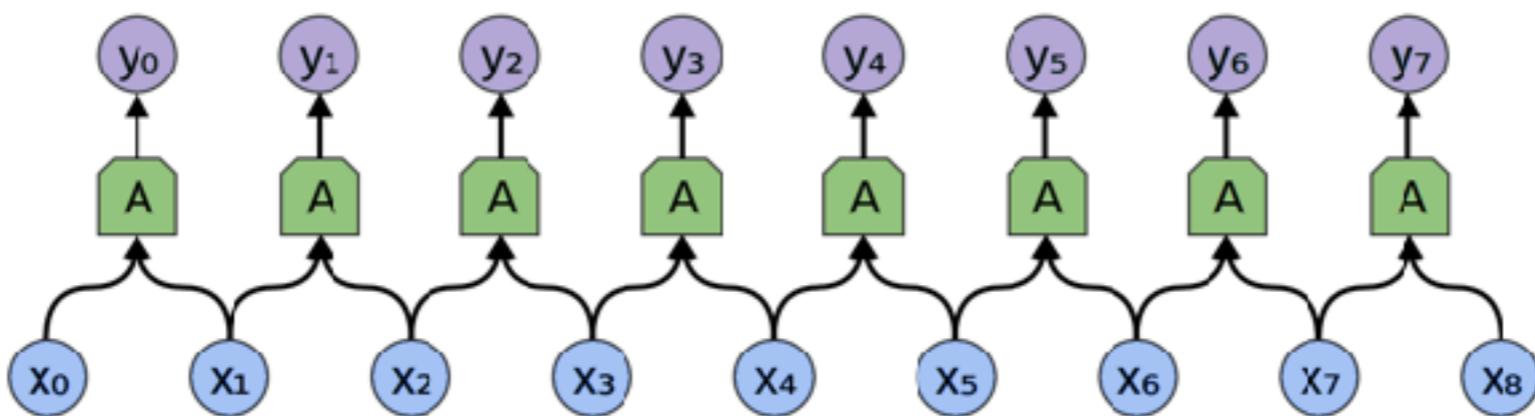


Bidirectional

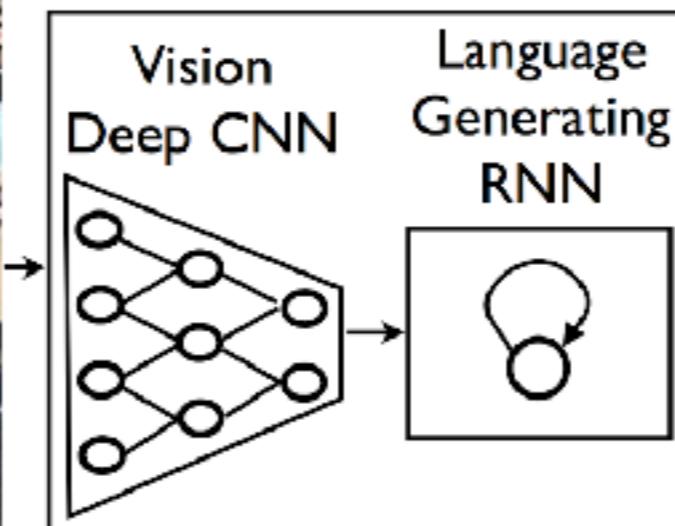
Recurrent neural layer model



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.

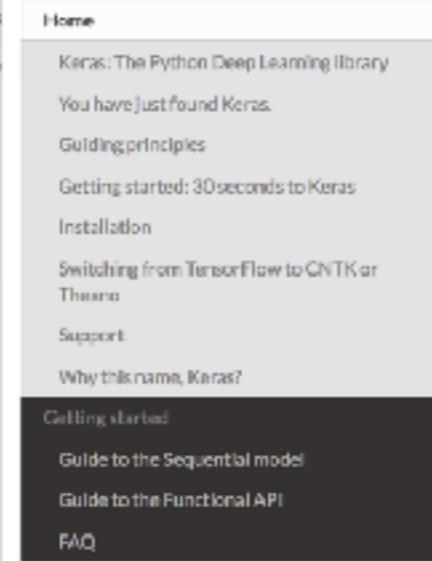
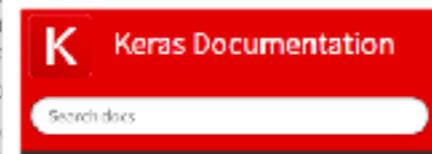
It supports modeling with

- 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
 - Clipped
 - Half
 - Stan

Getting Started
Tutorials
API
Community
Contributing



TensorFlow 1.2 has arrived!
We're excited to announce the release of TensorFlow 1.2! Check out the release notes for all the latest.

[UPGRADE NOW](#)

Introducing TensorFlow Research Cloud
We're making 1,000 Cloud TPUs available for free to accelerate open machine learning research.

[LEARN MORE](#)

The 2017 TensorFlow Dev Summit
Thousands of people from the TensorFlow community participated in the first flagship event. Watch the keynotes and talks.

[WATCH VIDEOS](#)

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.

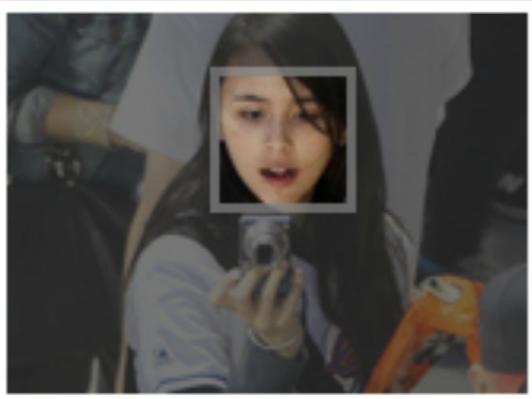
Read the documentation at [Keras.io](#).

Keras is compatible with Python 2.7-3.5.

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



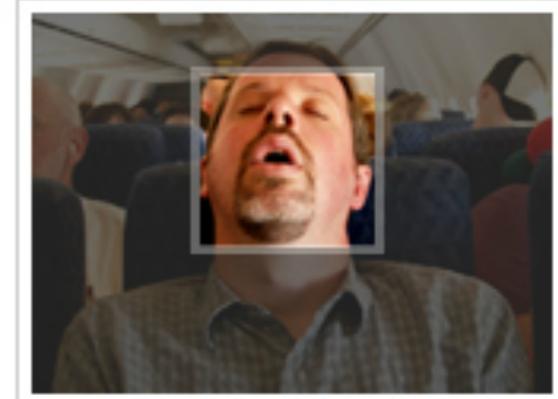
Face recognition.



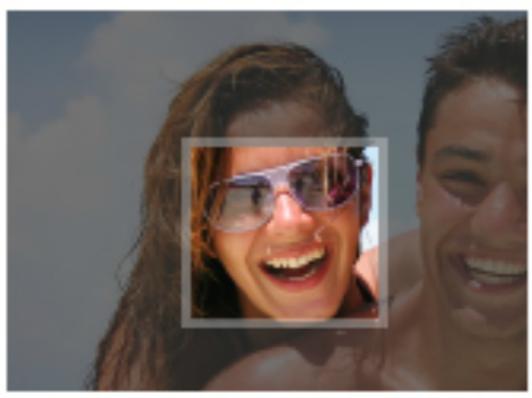
Who is this?



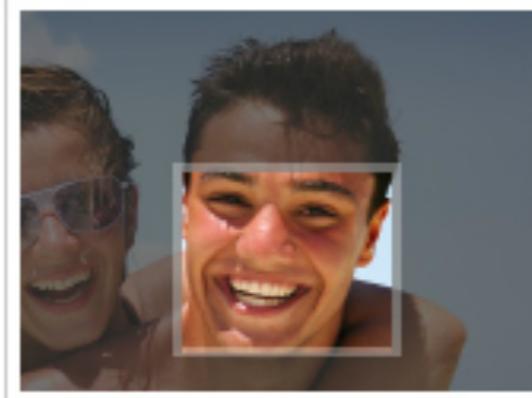
Who is this?



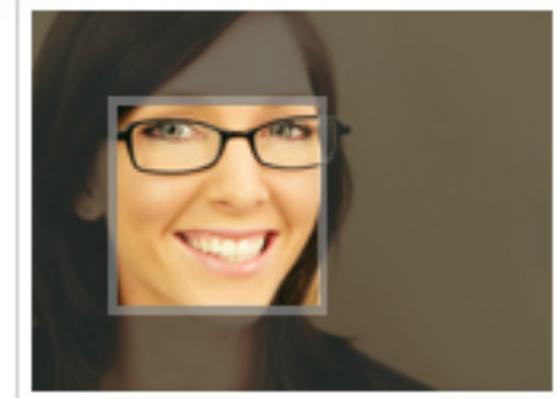
Who is this?



Who is this?



Who is this?



Who is this?

DeepFace (Facebook): Accuracy of 97.35%

New applications: navigation and mapping.

The screenshot shows the official Dyson website. At the top, there's a navigation bar with the Dyson logo, followed by links to Tienda, Aspiradoras, Ventiladores y Calefactores, Airblade™, Mi cuenta, and Soporte. A globe icon is also present. Below the navigation, the text "Robot Dyson 360 Eye™" is displayed, along with a button that says "Sea el primero en disfrutarlo". The main content area features a large image of the Dyson 360 Eye robot, which is cylindrical with a transparent bottom half showing its internal components like the motor and sensors. To the left of the robot, there's a circular callout with the text "Vea a James Dyson presentando el nuevo Dyson 360 Eye™ en Tokio" and a small video thumbnail showing James Dyson at a presentation.

New applications: Image Upscaling (Flipboard)



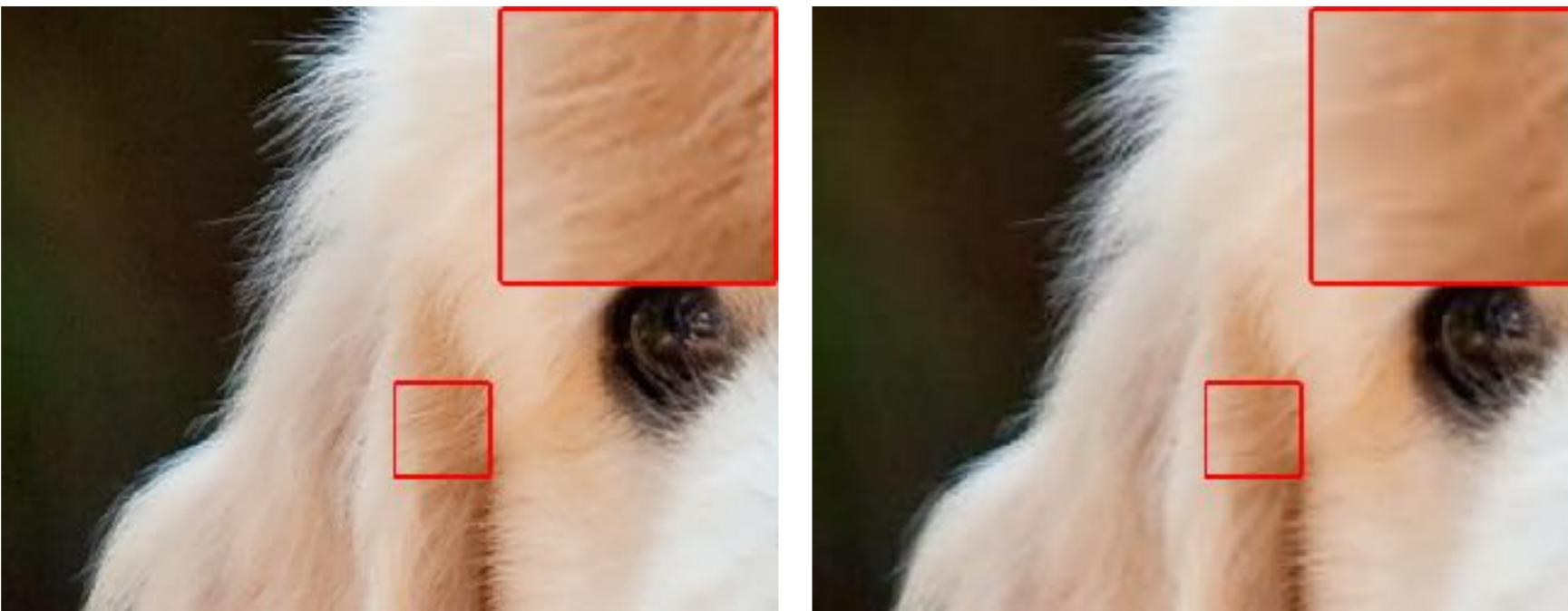
<http://engineering.flipboard.com/2015/05/scaling-convnets/>

New applications: Image Upscaling (Flipboard)



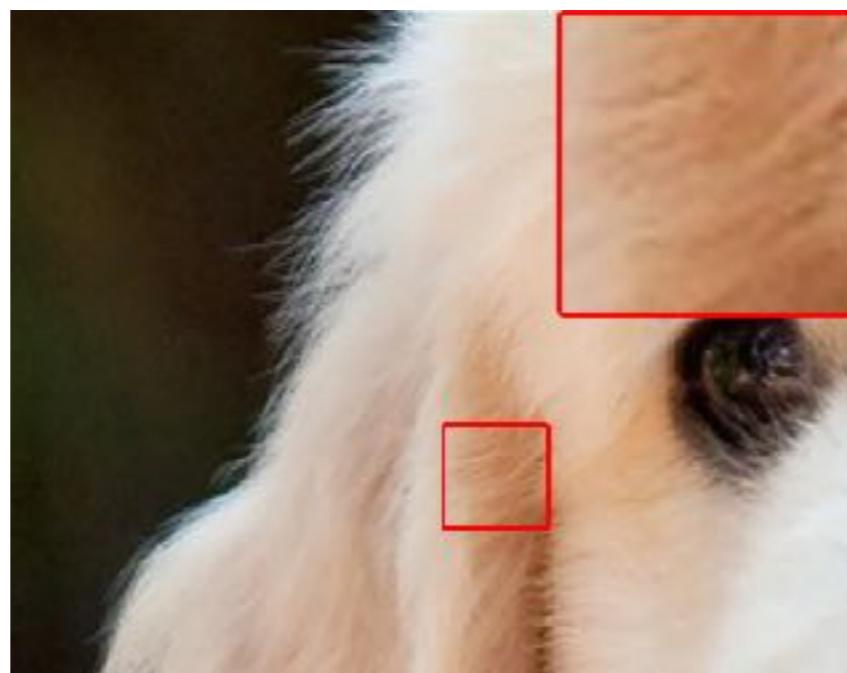
<http://engineering.flipboard.com/2015/05/scaling-convnets/>

New applications: Image Upscaling (Flipboard)



Original

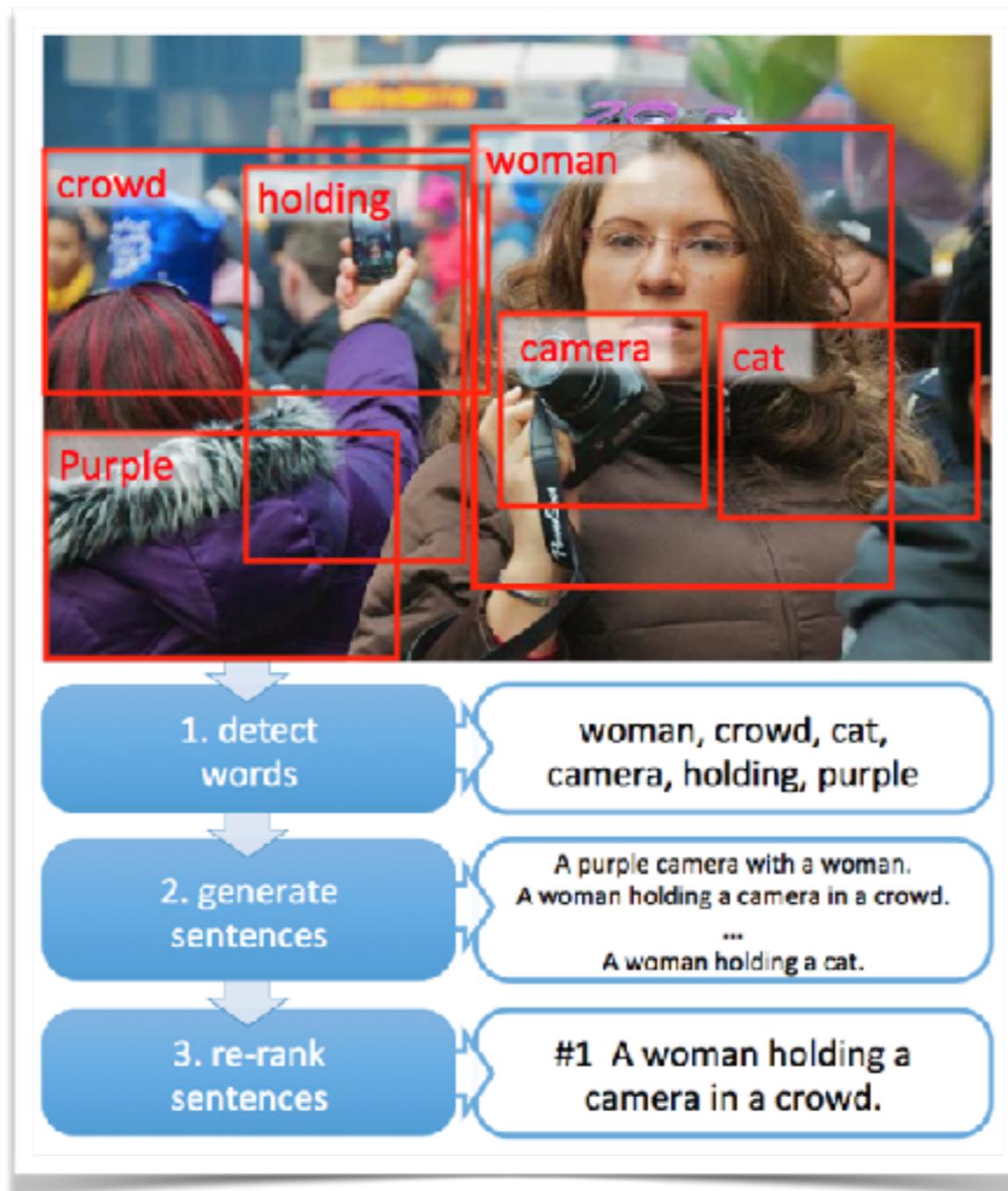
Bicubic



Model

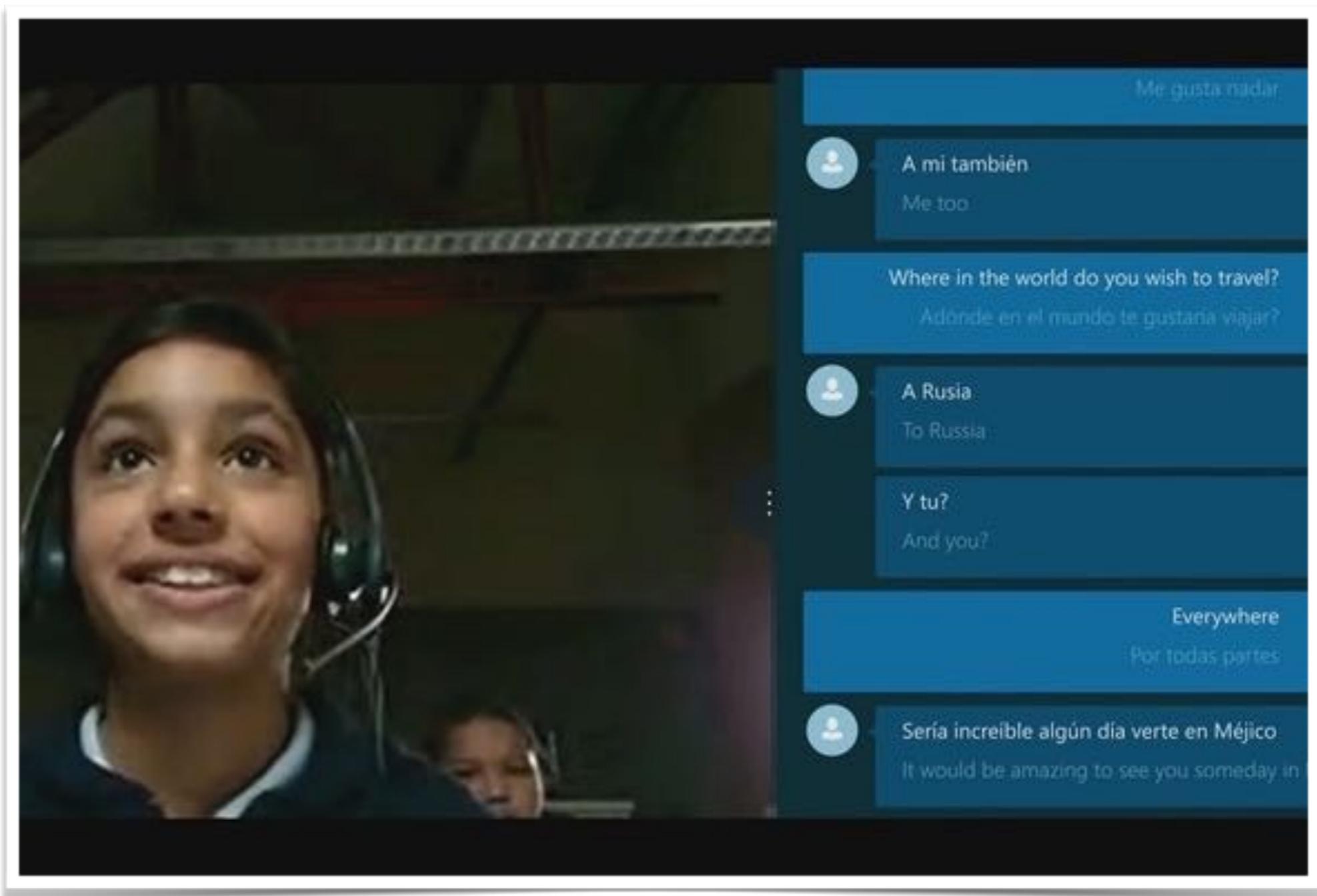
<http://engineering.flipboard.com/2015/05/scaling-convnets/>

New applications: Automatic Image Captioning

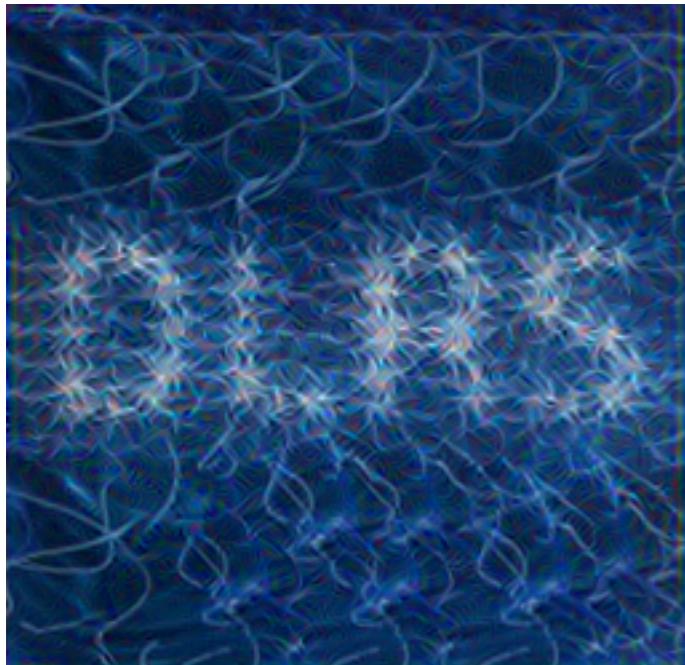


<http://blogs.technet.com/b/machinelearning/archive/2014/11/18/rapid-progress-in-automatic-image-captioning.aspx>

Speech translation



Recommenders



1st Workshop on Deep Learning for Recommender Systems

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

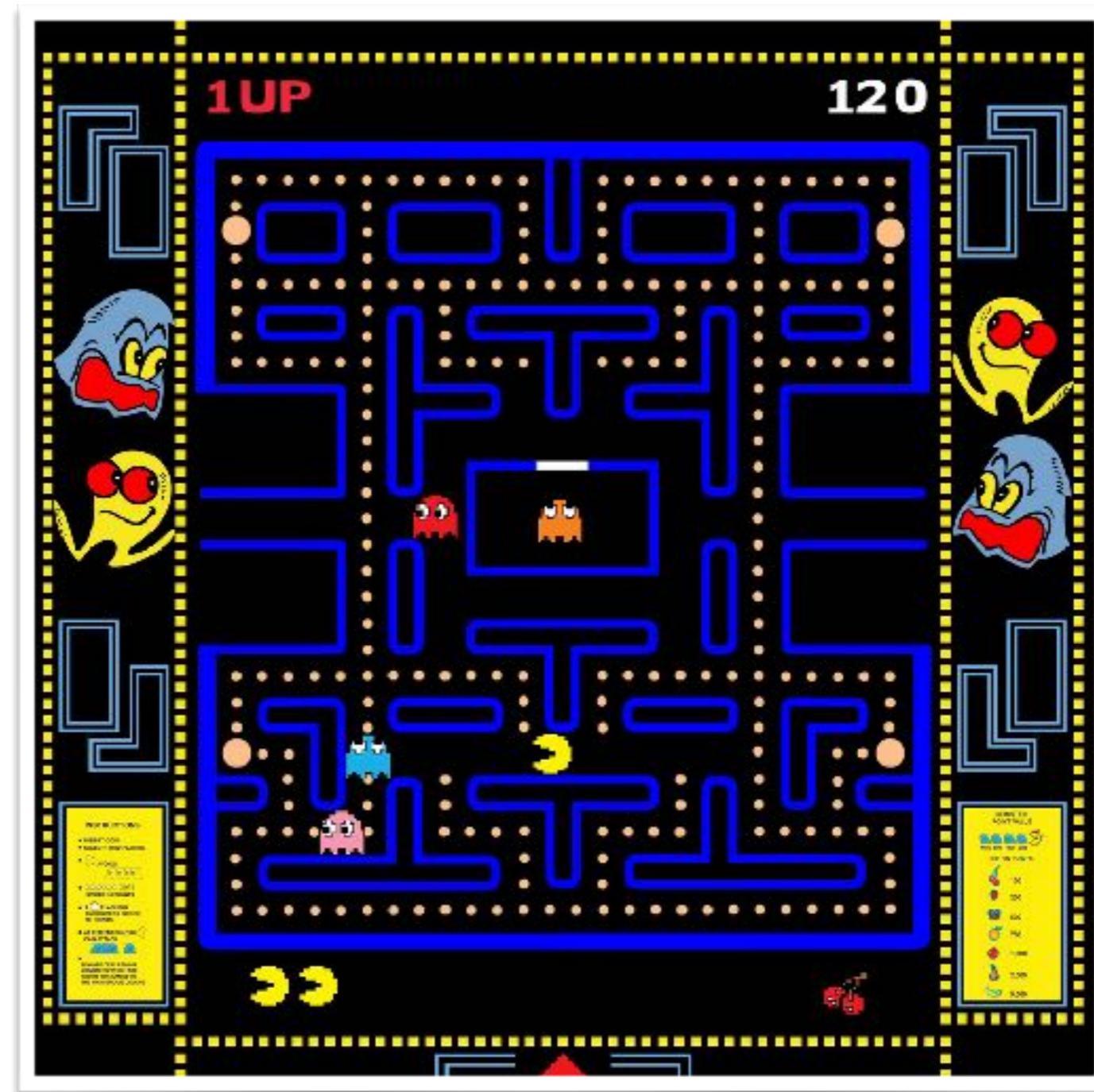
Music Generation

The screenshot shows a SoundCloud profile for an account named 'deepjazz'. The profile picture is a white circle containing a stylized 'dj' logo. The bio reads: 'I'm an AI built to make Jazz' and 'Princeton, United States'. The profile has 104 followers and 1 following, with 6 tracks. The tracks listed are:

- deepjazz on Metheny (14 days ago, Electronic genre, 0.33 seconds)
- dj 1 deepjazz On Metheny ... 1 Epoch (6.142 plays)
- dj 2 deepjazz On Metheny ... 16 Epochs (3.452 plays)
- dj 3 deepjazz On Metheny ... 32 Epochs (1.908 plays)

On the right side, there is a bio message: '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'. Below the bio are links to 'my source code (GitHub)' and 'deepjazz.io'. At the bottom right, there is a 'View all' button.

Reinforcement learning.



Go



Hands On!

Open <https://github.com/DeepLearningUB/EIIS2017> in your browser

The screenshot shows the GitHub repository page for 'DeepLearningUB / EIIS2017'. The repository description 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.' Key statistics include 24 commits, 1 branch, 0 releases, 1 contributor, and an MIT license. The repository has 1 unwatched star and 2 forks. The code tab is selected. A recent commit by 'algoritmes' was made 11 minutes ago, updating README.md. Other files listed include data, images, and three Jupyter notebooks: 1. Learning from data and optimization.ipynb, 2. Automatic Differentiation.ipynb, and 3. Tensorflow programming.ipynb, all added via upload 5 months ago.

File	Action	Time
README.md	Update	11 minutes ago
data	Add files via upload	5 months ago
images	Add files via upload	5 months ago
1. Learning from data and optimization.ipynb	Add files via upload	5 months ago
2. Automatic Differentiation.ipynb	Add files via upload	5 months ago
3. Tensorflow programming.ipynb	Add files via upload	5 months ago

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
```

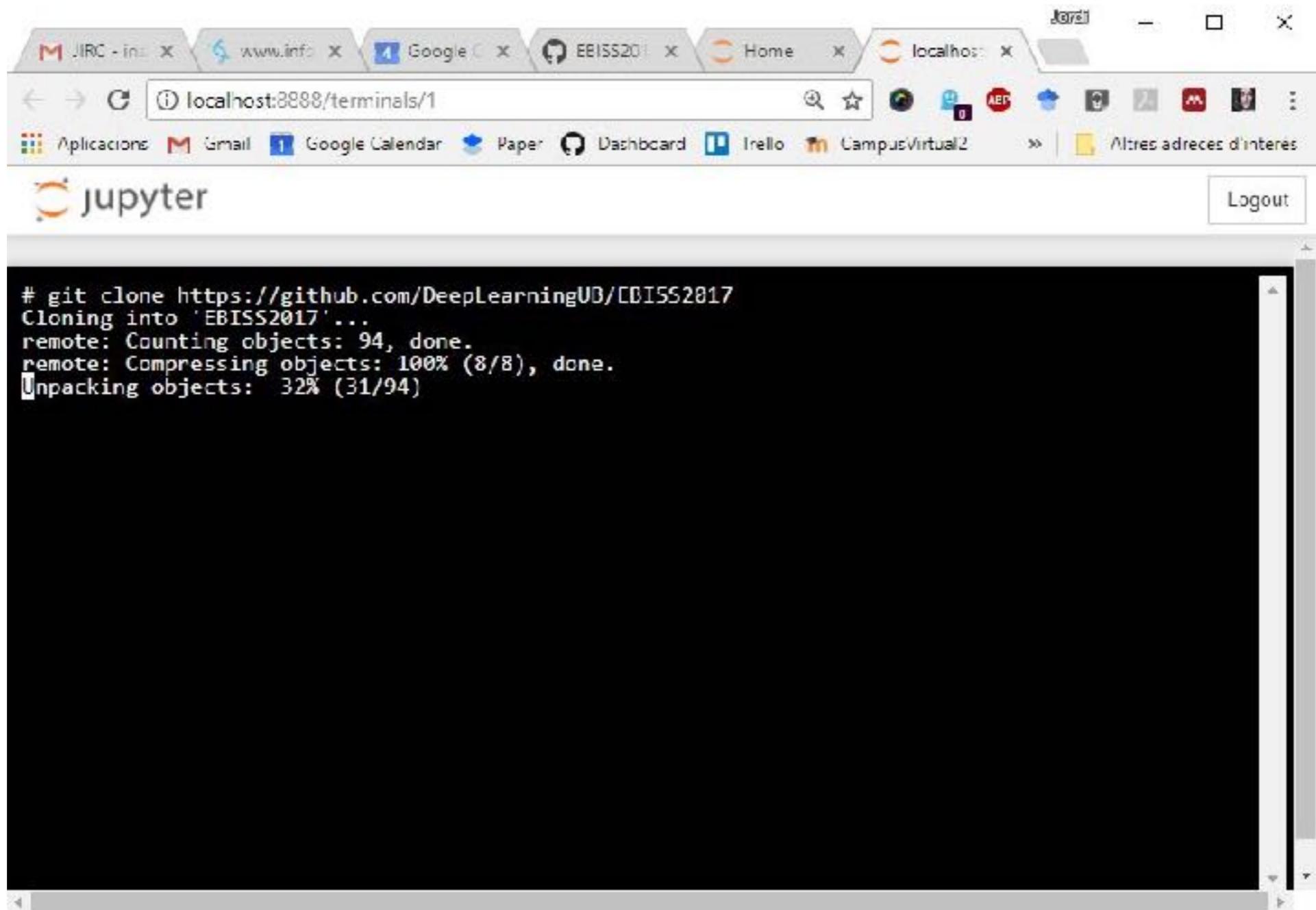
and go with your default browser to
localhost:8888

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>

A screenshot of a web browser window titled "Jordi". The address bar shows the URL "localhost:8888/tree". The browser has several tabs open, including "JIRC - inscript", "www.info.un", "Google Calen", "EBISS2017/RE", and "Home". Below the address bar is a toolbar with icons for search, star, refresh, and other browser functions. The main content area is labeled "jupyter". At the top of the jupyter interface, there are three tabs: "Files", "Running", and "Clusters". The "Files" tab is selected. Below the tabs, there is a message "Select items to perform actions on them." and a set of buttons for "Upload", "New", and "Edit". At the bottom of the interface, there is a file list with columns for "Name" and "Last Modified".

The screenshot shows a web browser window with multiple tabs open. The active tab is labeled "localhost:8888/tree". The browser's toolbar includes icons for back, forward, search, and various extensions. Below the toolbar, there are links for "Aplicacions", "Gmail", "Google Calendar", "Paper", "Dashboard", "Trello", and "CampusVirtual2". A sidebar on the left displays the "jupyter" logo and navigation tabs for "Files", "Running", and "Clusters". The main content area shows a file list with a single item: "instructions.docx". To the right of the file list is a "New" button dropdown menu. The "Notebook" section of the menu lists "Python 2" (which is selected), "Text File", and "Folder". The "Other" section lists "Terminal".



JIRC - in... X www.info X Google X EBISS2017 X Home X localhost X Jordi

localhost:8888/terminals/1

Aplicacions Gmail Google Calendar Paper Dashboard Trello CampusVirtual2 Altres adreces d'interes

jupyter Logout

```
# git clone https://github.com/DeepLearningUB/EBISSIONT  
Cloning into 'EBISSIONT'...  
remote: Counting objects: 94, done.  
remote: Compressing objects: 100% (8/8), done.  
Unpacking objects: 32% (31/94)
```

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

<https://github.com/DeepLearningUB/EBlSS2017>

The screenshot shows the GitHub repository page for 'DeepLearningUB / EBlSS2017'. The repository description 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.' It has 24 commits, 1 branch, 0 releases, 1 contributor, and is licensed under MIT. The repository was last updated 11 minutes ago. The commit history lists several files added via upload, all from 5 months ago.

File	Action	Time
algorithms/Update README.md	Add files via upload	11 minutes ago
data	Add files via upload	5 months ago
images	Add files via upload	5 months ago
1. Learning from data and optimization.ipynb	Add files via upload	5 months ago
2. Automatic Differentiation.ipynb	Add files via upload	5 months ago
3. Tensorflow programming.ipynb	Add files via upload	5 months ago

We can start to code!

