

DEEP LEARNING FROM SCRATCH

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

Our Background



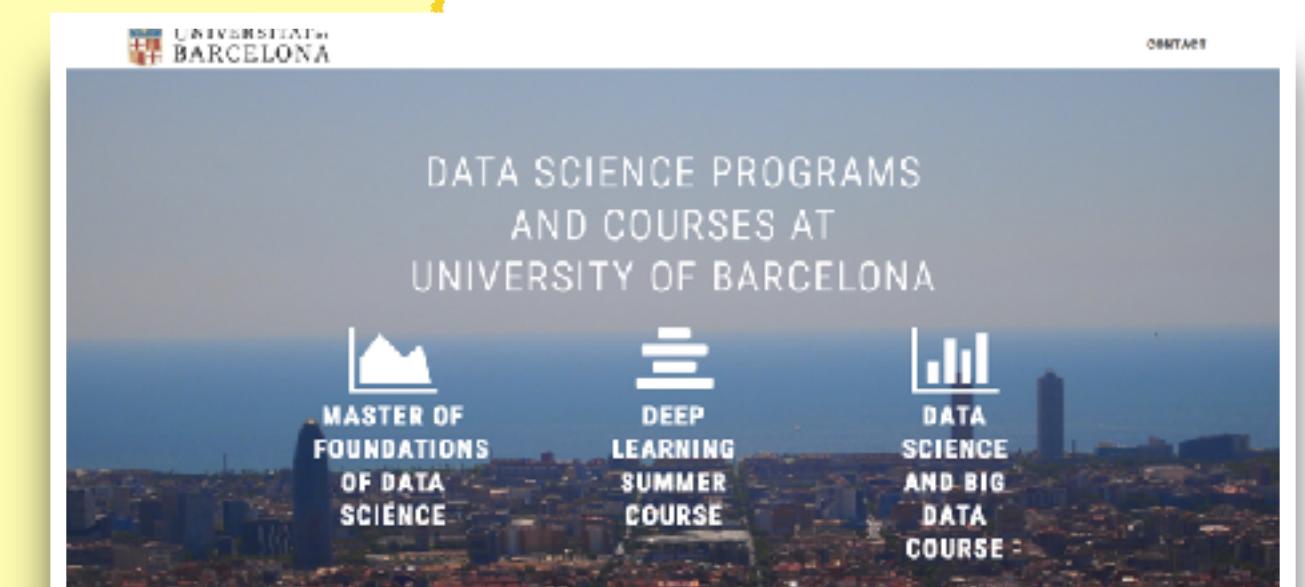
MACHINE
LEARNING
RESEARCH

Oriol Pujol,
Associate Professor at UB.

COMPUTER
VISION
RESEARCH

Santi Seguí,
Lecturer at UB.

Jordi Vitrià.
Full Professor at UB.



Day 1

Course Agenda

Thursday and Friday, 8:30h to 17:15h

1. Using the Jupyter notebook & Docker. Software stack. 8:30h-9:00h
2. Introduction to Deep Learning and its applications. 9:00h-10:00h
3. Coffee Break 10:00h-10:30h
4. Learning from Data. 10:30h-12:30h
5. Lunch. 12:30h-13:30h
6. Backpropagation. Training a MLP Neural Network from Scratch. 13:30h-15:30h
7. Coffee Break 15:30h-15:45h
8. Tensorflow programming model. 15:45h-17:15h

Day 2

Course Agenda

Thursday and Friday, 8:30h to 17:15h

1. Hands-on Tensorflow. 8:30h-9:30h
2. Convolutional Neural Networks. 9:30h:10:00h
3. Coffee Break. 10:00-10:30h
4. Convolutional Neural Networks . 10:30h:11:30h
5. Recurrent Neural Networks. 11:30h:12:30h
6. Lunch. 12:30h-13:30h
7. Recurrent Neural Networks. 13:30h:14:30h
8. Unsupervised Learning. 14:30h:15:30h
9. Coffee Break. 15:30h-15:45h
10. Applications: Embeddings, Sentiment, Time Series. 15:45h:17:15h

Course Repository

<https://github.com/DataScienceUB/Vodafone2017>

The screenshot shows the GitHub repository page for 'DataScienceUB / Vodafone2017'. The repository is private, has 13 commits, 1 branch, 0 releases, 1 contributor, and is licensed under MIT. It contains files for 'algoritmes', 'dataset', 'files', 'images', 'slides', and '0. Deep Learning from Scratch.ipynb'. All files were added via upload 3 days ago.

No description, website, or topics provided.

13 commits | 1 branch | 0 releases | 1 contributor | MIT

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

File	Action	Last Commit
algoritmes	Add files via upload	Latest commit 9cb9933 3 days ago
dataset	Add files via upload	3 days ago
files	Add files via upload	3 days ago
images	Add files via upload	3 days ago
slides	Add files via upload	3 days ago
0. Deep Learning from Scratch.ipynb	Add files via upload	3 days ago

What

1. Describe how a (deep) neural network works and combine different types of layers and activation functions. Deep Learning is not magic.
2. Describe how these models can be applied in computer vision, text analytics, time series analysis, etc. Deep Learning is not the final machine learning method.
3. Develop your own models in **Tensorflow** and derivates. You can train (mid-size) deep models in your laptop.

Why Now?



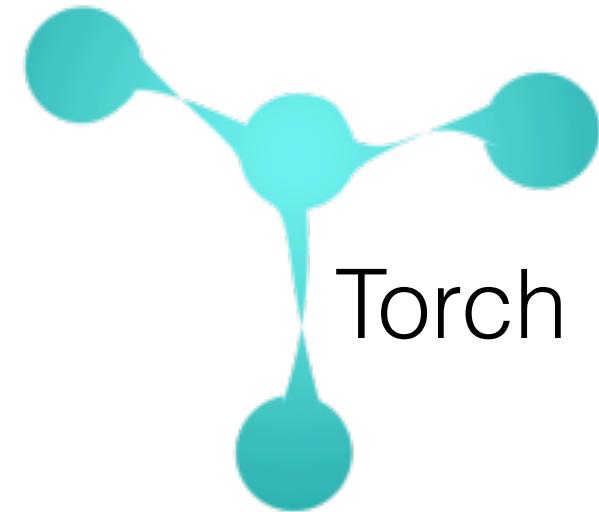
How Earth Globes were made back in the 50's

Deep Learning BackEnds



TensorFlow

Python, C++
MultiGPU
Distributed



Lua
MultiGPU

Google



theano

Python
Large amount
of sample
code

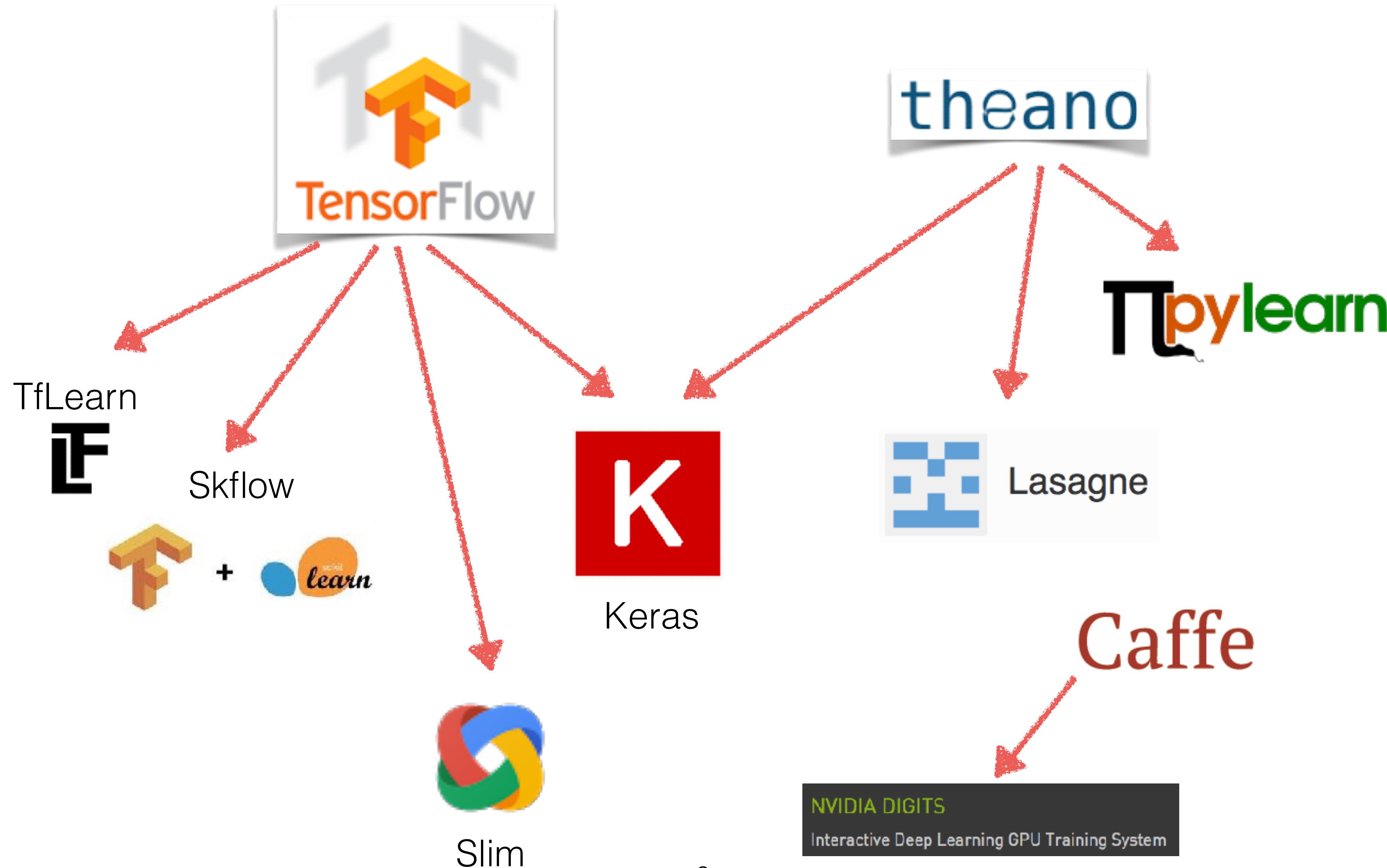
Caffe

Université de Montréal

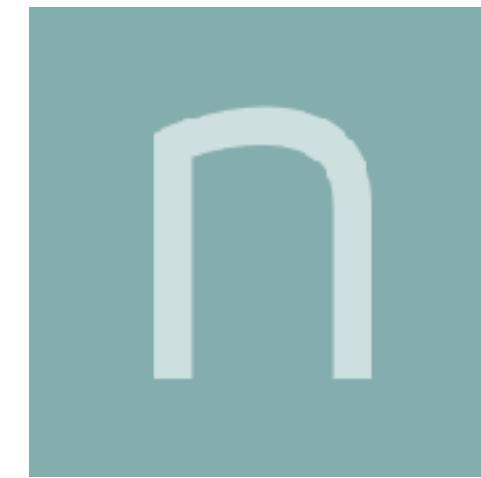
Python, C++
MultiGPU

Berkeley
UNIVERSITY OF CALIFORNIA

Tools



Other BackEnds



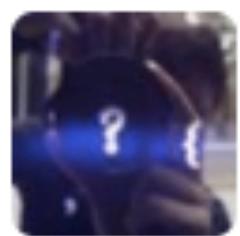
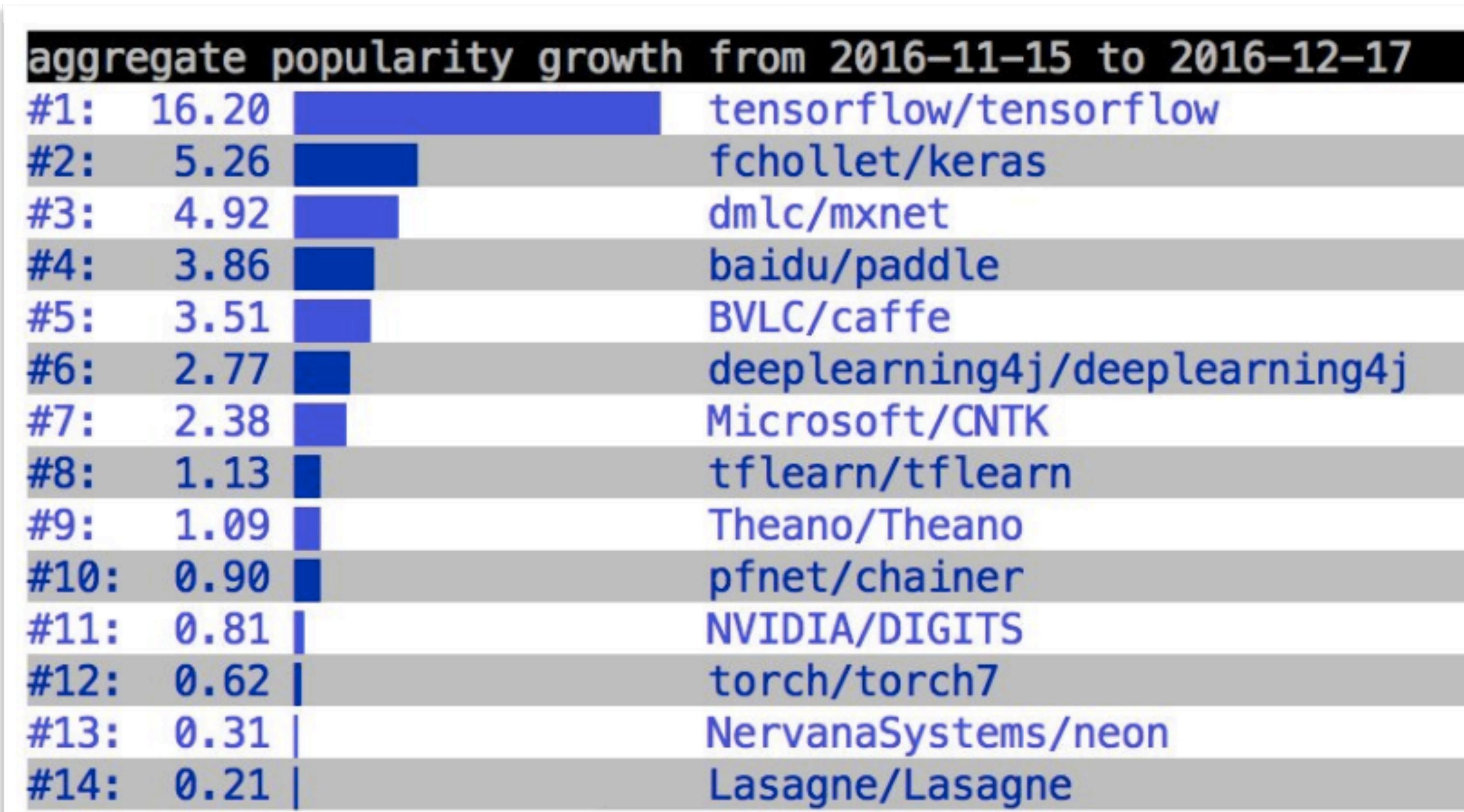
Neon

DL4J Deep Learning for Java



ConvNetJS
Deep Learning in your browser

The ConvNetJS logo features a small icon of a neural network layer with three input nodes and three output nodes connected by lines. To the right of the icon, the word "ConvNetJS" is written in a bold, black, sans-serif font. Below it, the text "Deep Learning in your browser" is written in a smaller, regular black font.



François Chollet @fchollet · 17 des. 2016

Deep learning frameworks growth over the past month.

◀ 14

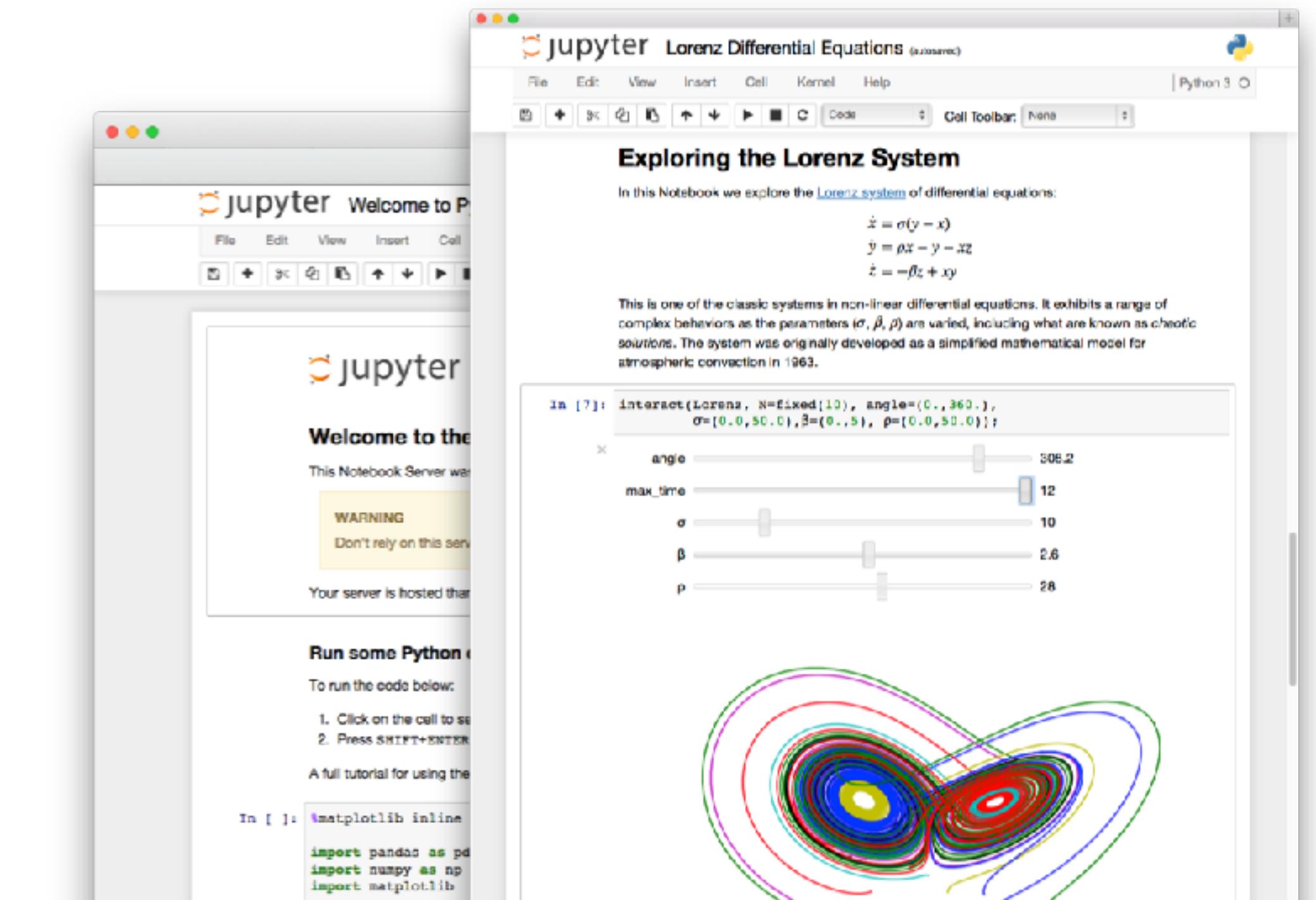
↑ 324



509

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



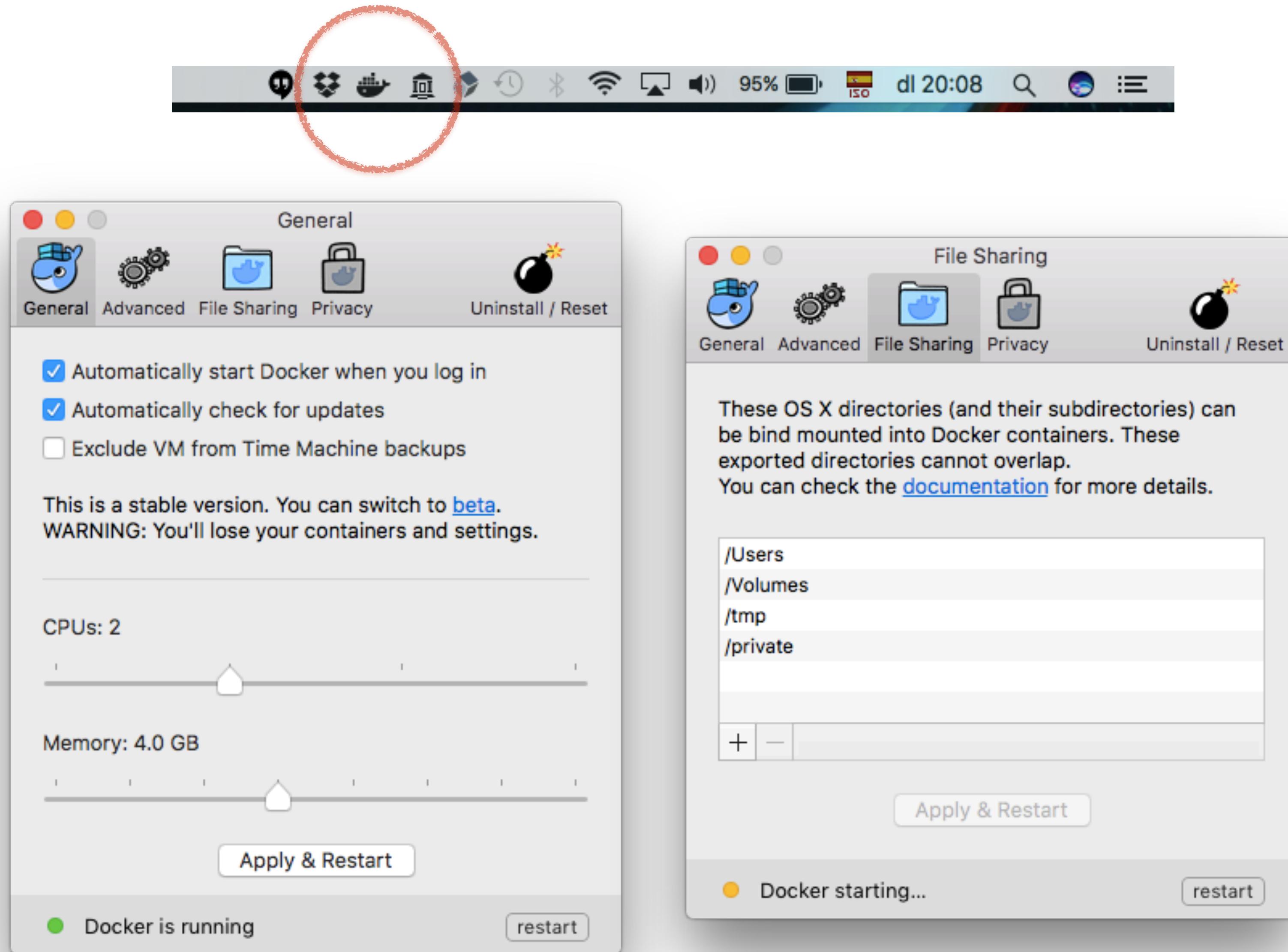
We will use a **Docker Container**.

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.

Course Software Installation

The best way to run the course software is to use a **Docker container**. There's full documentation on installing Docker at [docker.com](https://docs.docker.com), but in a few words, the steps are:

- Go to docs.docker.com in your browser.
- Step one of the instructions sends you to download Docker.
- Run that downloaded file to install Docker.
- At the end of the install process a whale in the top status bar indicates that Docker is running, and accessible from a terminal.
- Click the whale to get [Preferences](#), and other options.
- Open a command-line terminal, and run some Docker commands to verify that Docker is working as expected. Some good commands to try are `docker version` to check that you have the latest release installed, and `docker ps` and `docker run hello-world` to verify that Docker is running.
- By default, Docker is set to use 2 processors. You can increase processing power for the app by setting this to a higher number in [Preferences](#), or lower it to have Docker for Mac use fewer computing resources.
- Memory - By default, Docker is set to use 2 GB runtime memory, allocated from the total available memory on your computer. You can increase the RAM on the app to get faster performance by setting this number higher (for example to 3) or lower (to 1) if you want Docker to use less memory.

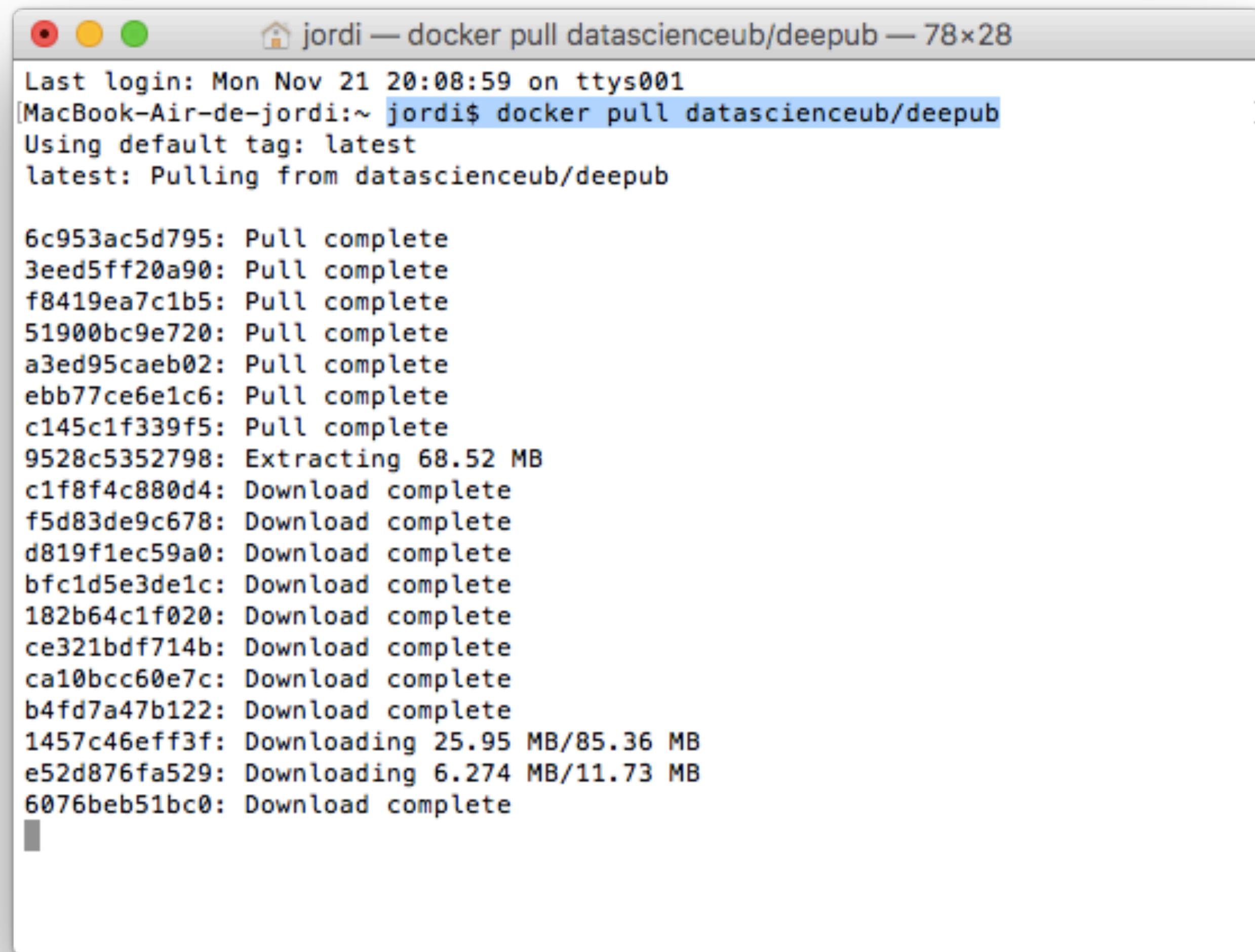


Once Docker is installed, you can download the image of this course:

- In a terminal, go to your course folder and run (This operation requires a good internet connection; it will take some minutes): `docker pull datascienceub/deepub`
- Run the `deepub` image on your system: `docker run -it -p 8888:8888 -p 6006:6006 -v $(pwd):/notebooks datascienceub/deepub`
- Once these steps have been done, you can check the installation by starting your web browser and introducing this URL: `http://localhost:8888`.
- Open a new Jupyter notebook and execute this instruction in a code cell: `!git clone https://github.com/DataScienceUB/Vodafone2017`

Note:

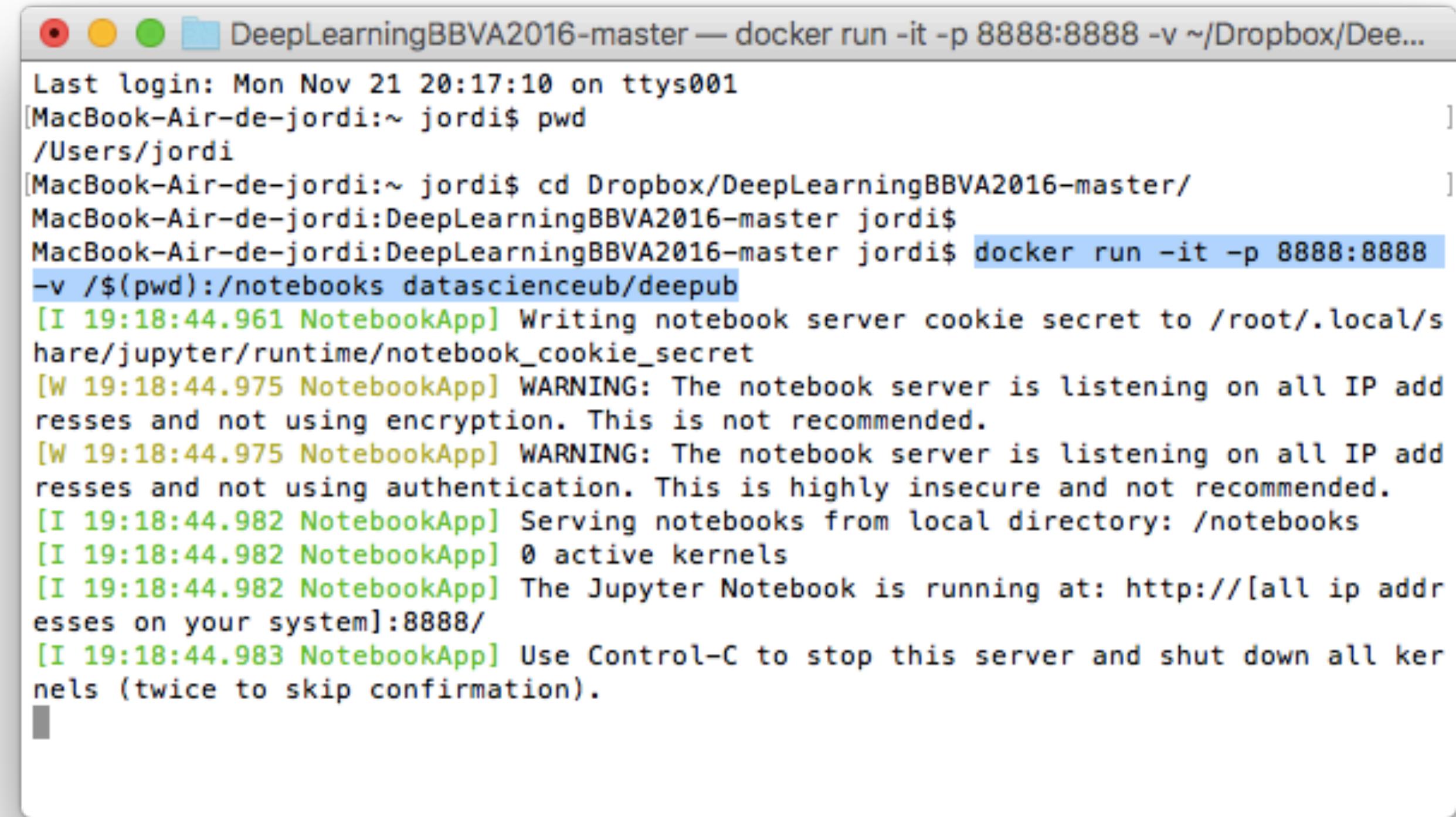
Docker for Windows requires 64bit Windows 10 Pro, Enterprise and Education. The Hyper-V package must be enabled for Docker for Windows to work. The Docker for Windows installer will enable it for you, if needed. (This requires a reboot). If your system does not satisfy these requirements, you can install Docker Toolbox, which uses Oracle Virtual Box instead of Hyper-V.



A screenshot of a macOS terminal window titled "jordi — docker pull datascienceub/deepub — 78x28". The window shows the command "docker pull datascienceub/deepub" being run, followed by a series of log messages indicating the progress of pulling Docker images. The log includes messages like "Pull complete" for various image IDs and "Extracting" for the main image, along with download progress for specific layers.

```
Last login: Mon Nov 21 20:08:59 on ttys001
[MacBook-Air-de-jordi:~ jordi$ docker pull datascienceub/deepub
Using default tag: latest
latest: Pulling from datascienceub/deepub

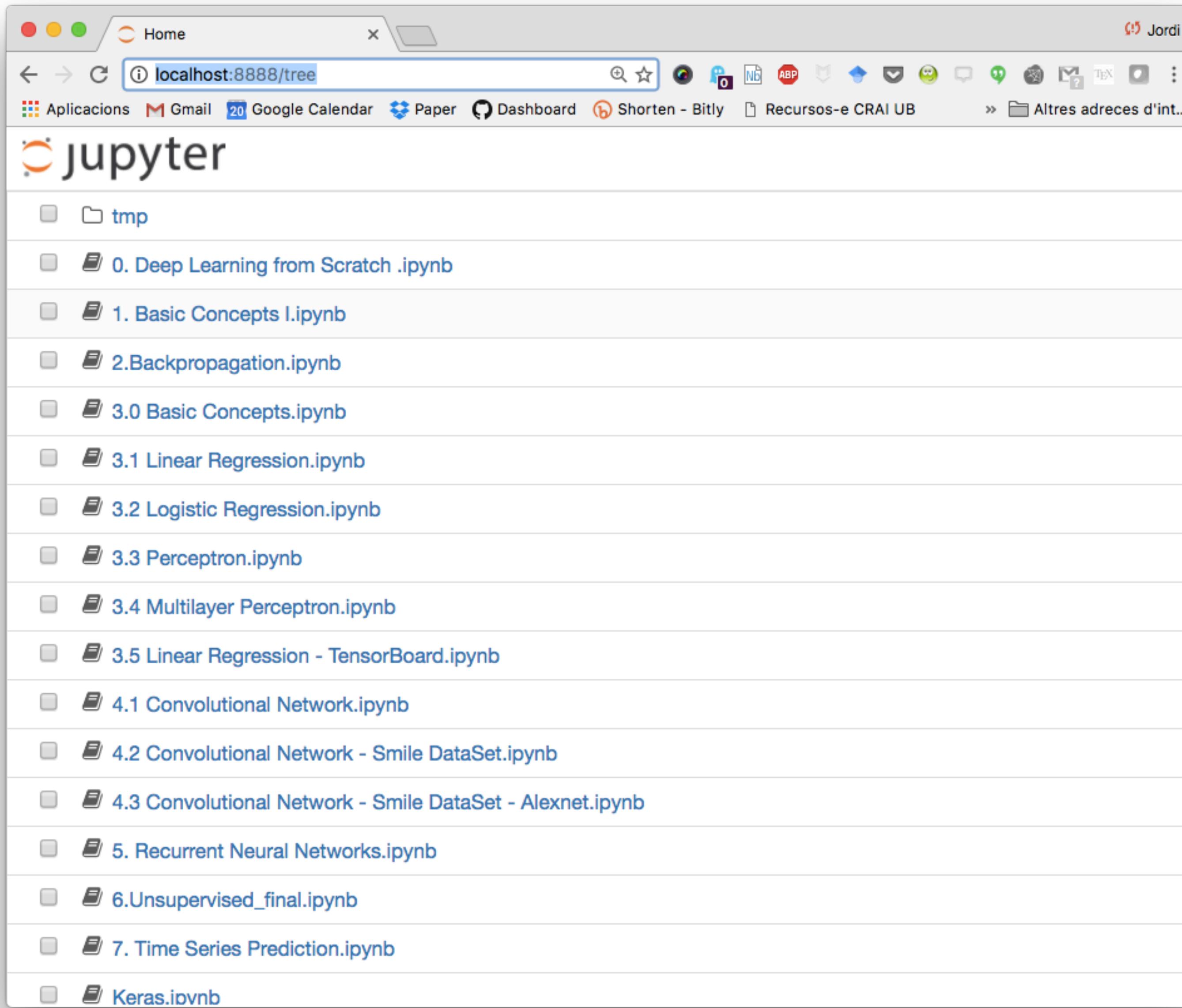
6c953ac5d795: Pull complete
3eed5ff20a90: Pull complete
f8419ea7c1b5: Pull complete
51900bc9e720: Pull complete
a3ed95caeb02: Pull complete
ebb77ce6e1c6: Pull complete
c145c1f339f5: Pull complete
9528c5352798: Extracting 68.52 MB
c1f8f4c880d4: Download complete
f5d83de9c678: Download complete
d819f1ec59a0: Download complete
bfc1d5e3de1c: Download complete
182b64c1f020: Download complete
ce321bdf714b: Download complete
ca10bcc60e7c: Download complete
b4fd7a47b122: Download complete
1457c46eff3f: Downloading 25.95 MB/85.36 MB
e52d876fa529: Downloading 6.274 MB/11.73 MB
6076beb51bc0: Download complete
```



```
Last login: Mon Nov 21 20:17:10 on ttys001
[MacBook-Air-de-jordi:~ jordi$ pwd
/Users/jordi
[MacBook-Air-de-jordi:~ jordi$ cd Dropbox/DeepLearningBBVA2016-master/
MacBook-Air-de-jordi:DeepLearningBBVA2016-master jordi$ docker run -it -p 8888:8888
-v /$(pwd):/notebooks datascienceub/deepub
[I 19:18:44.961 NotebookApp] Writing notebook server cookie secret to /root/.local/share/jupyter/runtime/notebook_cookie_secret
[W 19:18:44.975 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[W 19:18:44.975 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using authentication. This is highly insecure and not recommended.
[I 19:18:44.982 NotebookApp] Serving notebooks from local directory: /notebooks
[I 19:18:44.982 NotebookApp] 0 active kernels
[I 19:18:44.982 NotebookApp] The Jupyter Notebook is running at: http://[all ip addresses on your system]:8888/
[I 19:18:44.983 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

```
docker run -it -p 8888:8888 -p 6006:6006
-v /$(pwd):/notebooks datascienceub/deepub
```

```
docker run -it -p 8888:8888 -p 6006:6006 -v
C:\jordi\DeepLearning:/notebooks datascienceub/
deepub
```



THE REVENANT

INSPIRED BY TRUE EVENTS
JANUARY 8



Why Deep Learning?





Historical metaphors of the brain:

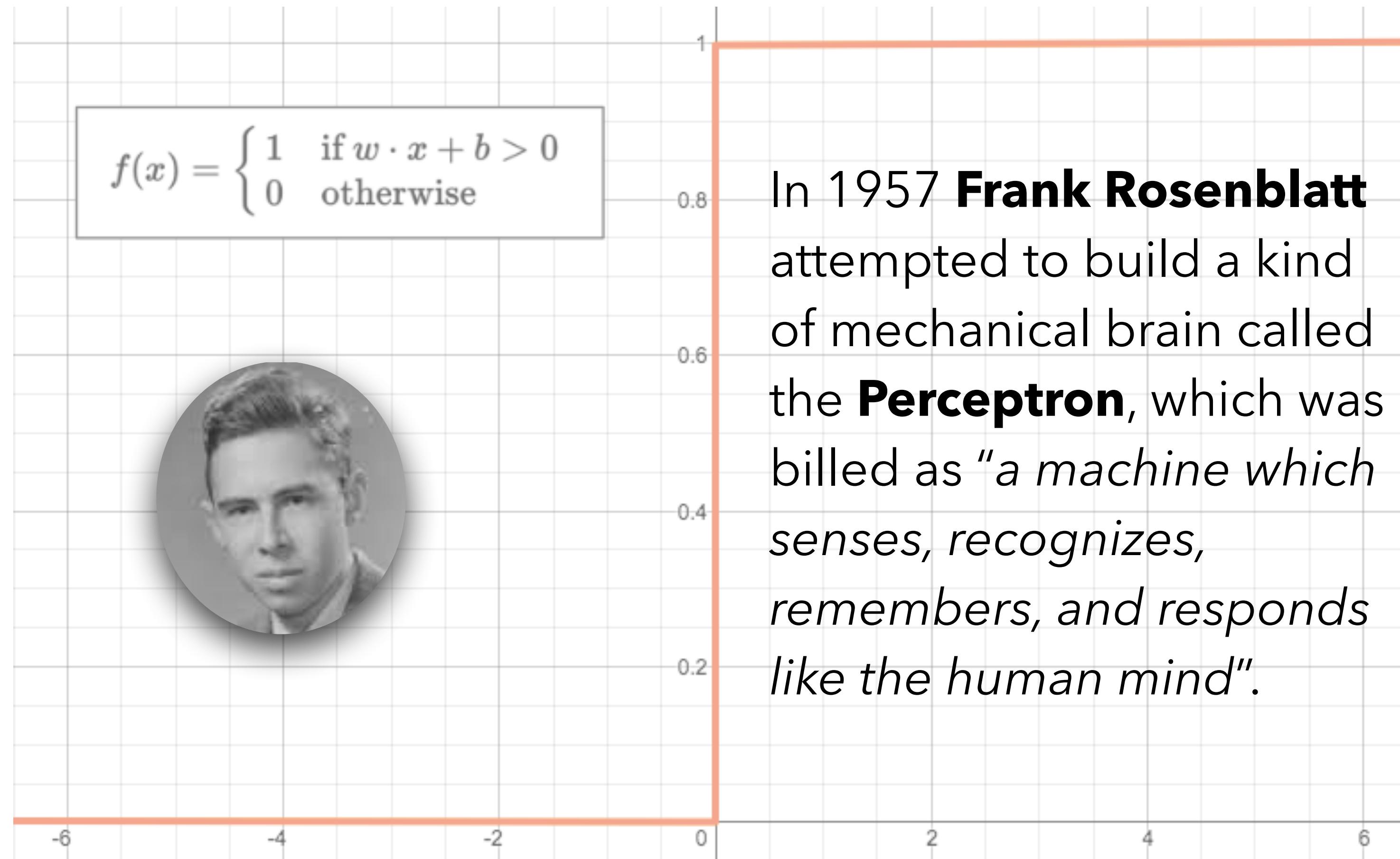
Hydraulic (blood cooler, spirits),
Mechanical (clock, steam machine), ...



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**.





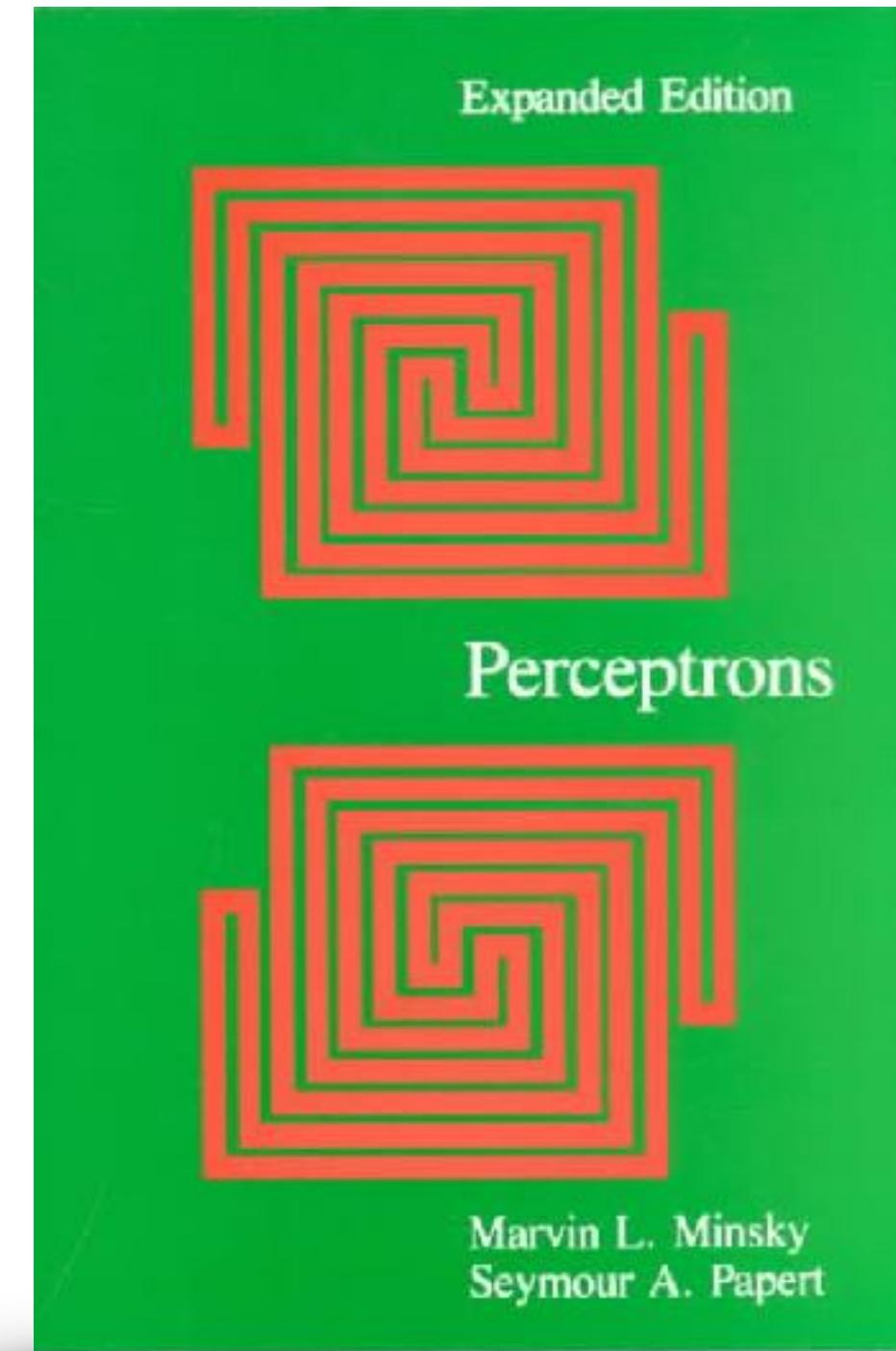
In 1962, **Widrow & Hoff** developed a **learning rule** that examines the value before the weight adjusts it (i.e. 0 or 1) according to the rule:

$$\text{Weight Change} = (\text{Pre-Weight line value}) * (\text{Error} / (\text{Number of Inputs})).$$

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".



It is claimed that pessimistic predictions made by the authors were responsible for an erroneous change in the direction of research in AI, concentrating efforts on so-called "symbolic" systems, and contributing to the so-called AI winter. This decision, supposedly, proved to be unfortunate in the 1980s, when new discoveries showed that the prognostics in the book were wrong.

Source: Wikipedia

70's: First neural network winter





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**), recurrent networks (**J.Schmidhuber**), 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.

90's-00's: Second neural network winter



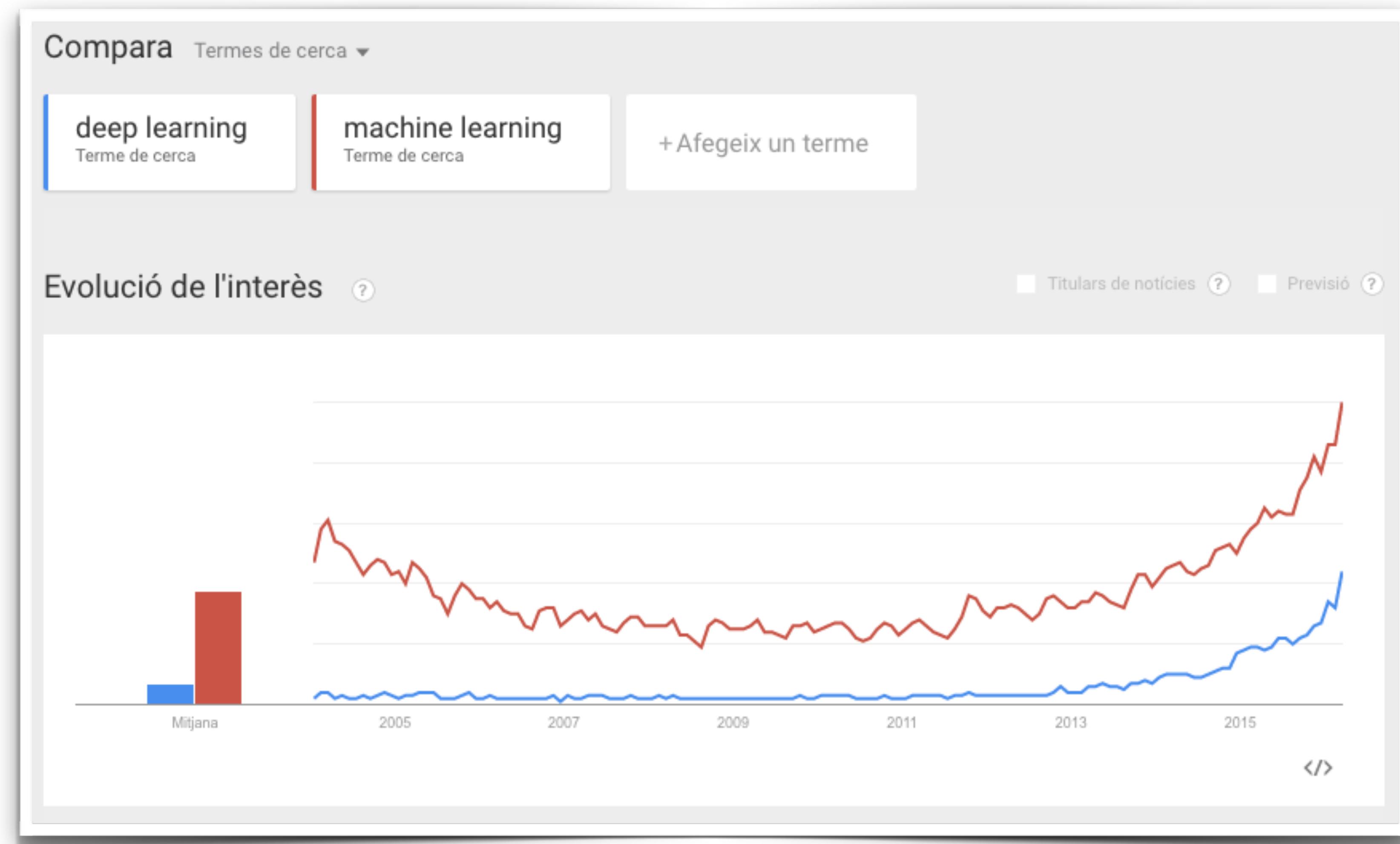


With the ascent of Support Vector Machines and the **failure of backpropagation**, the early 2000s were a dark time for neural net research.

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.

Neural Networks Reborn



Google Trends



amazon



MICRO CENTER
computers & electronics



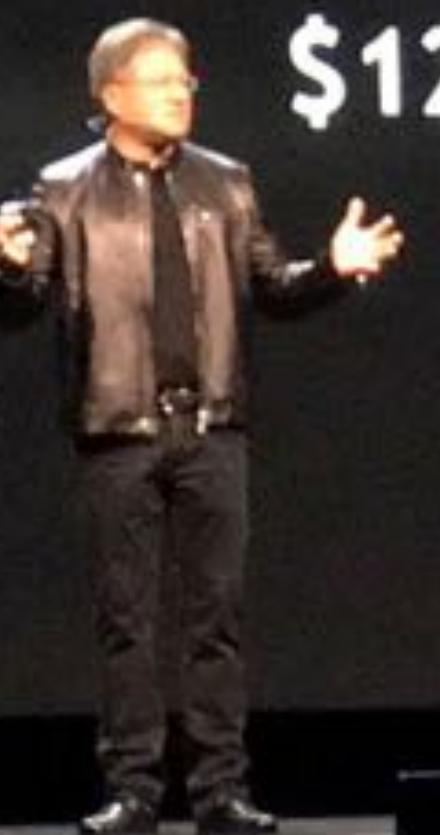


NVIDIA DGX-1

WORLD'S FIRST
DEEP LEARNING SUPERCOMPUTER

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

\$129,000



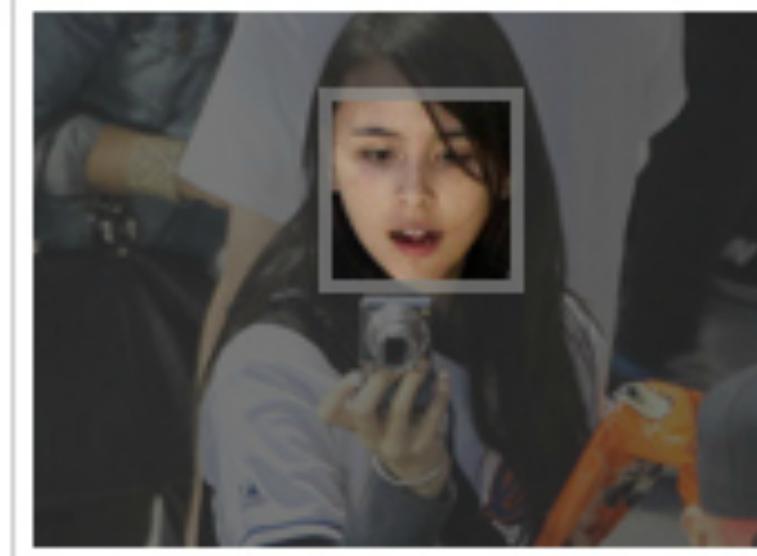
Definitions

- **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 for learning in neural networks.
- NN and DL currently provide the best solutions to many problems in image recognition, speech recognition, and natural language processing.

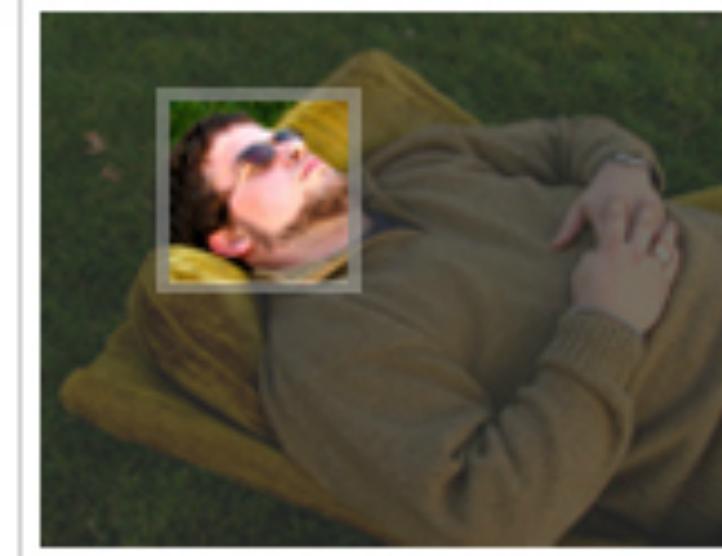
“Classical” applications: object classification, detection and segmentation.



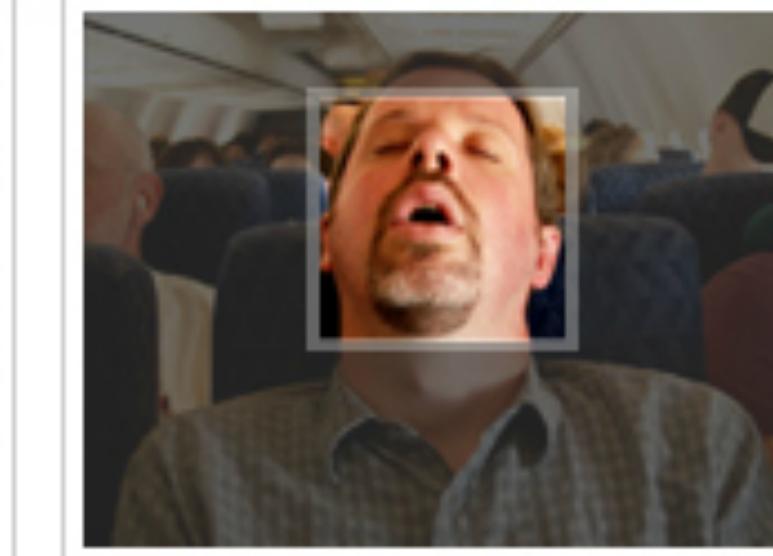
Face recognition.



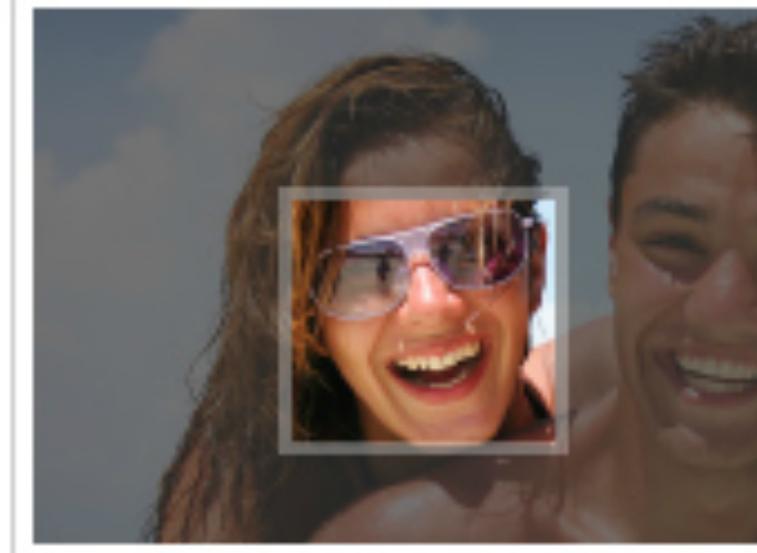
Who is this?



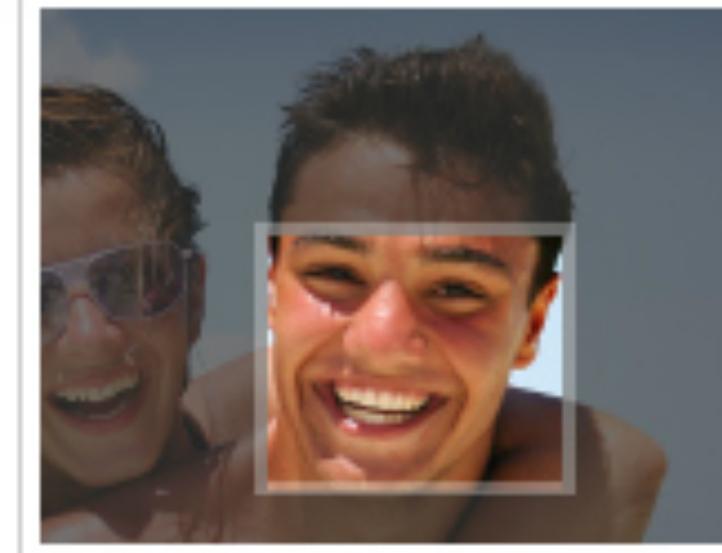
Who is this?



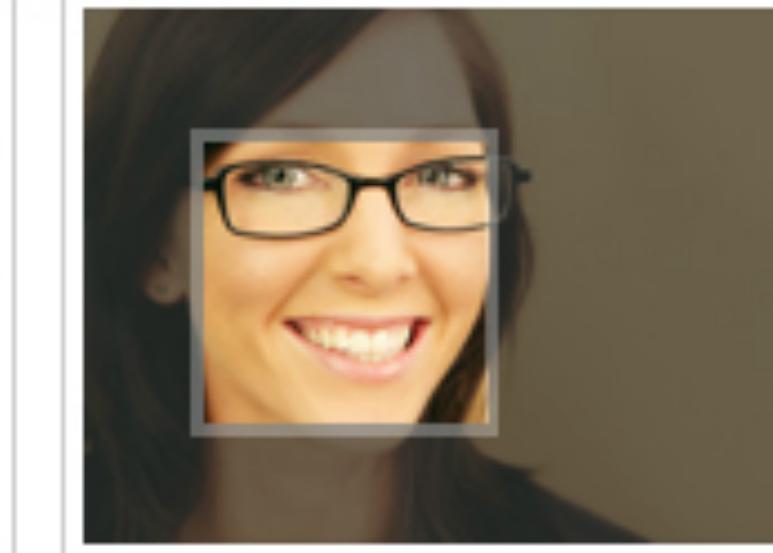
Who is this?



Who is this?



Who is this?



Who is this?

DeepFace (Facebook): Accuracy of 97.35%

New applications: navigation and mapping.

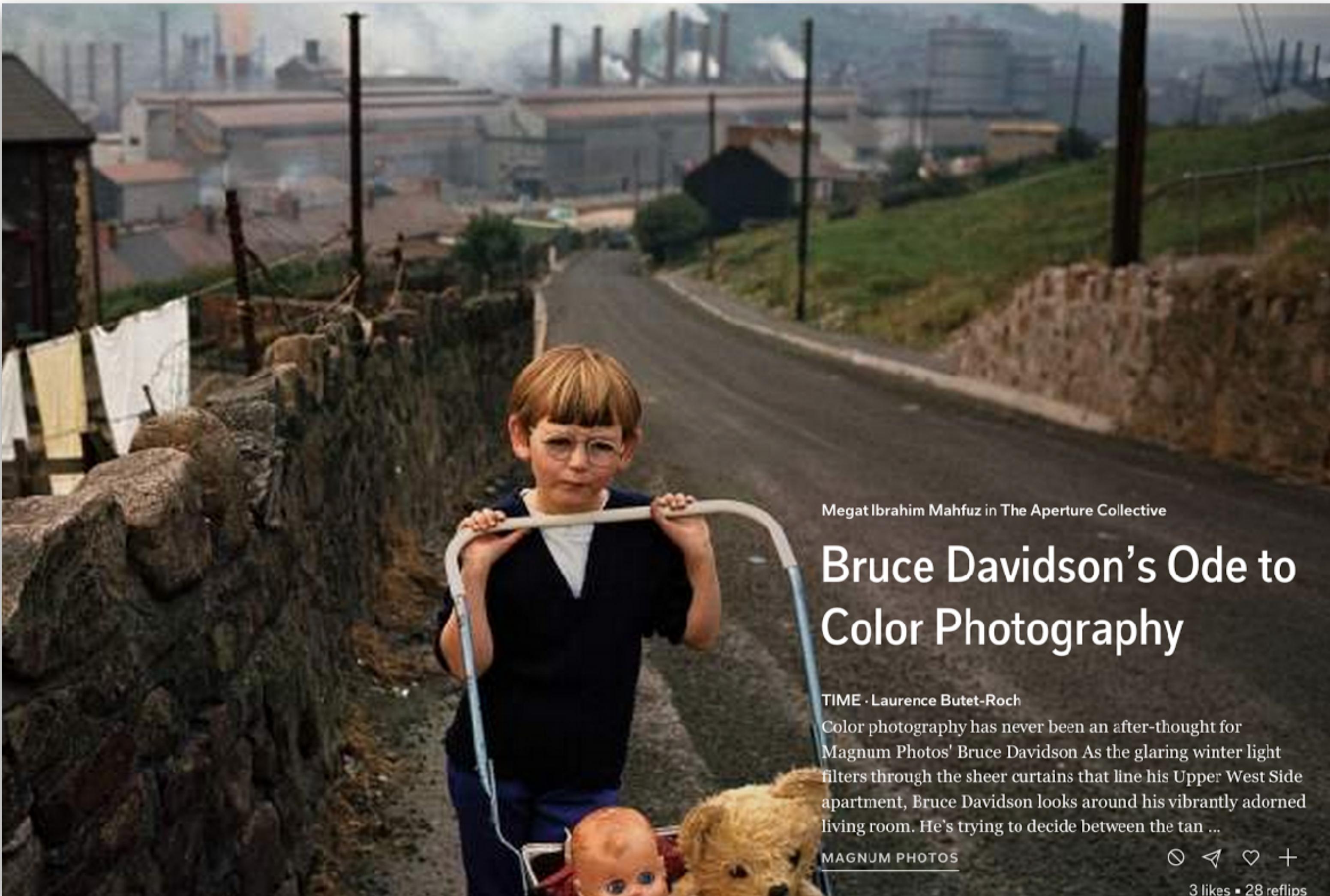


New applications: Image Upscaling (Flipboard)



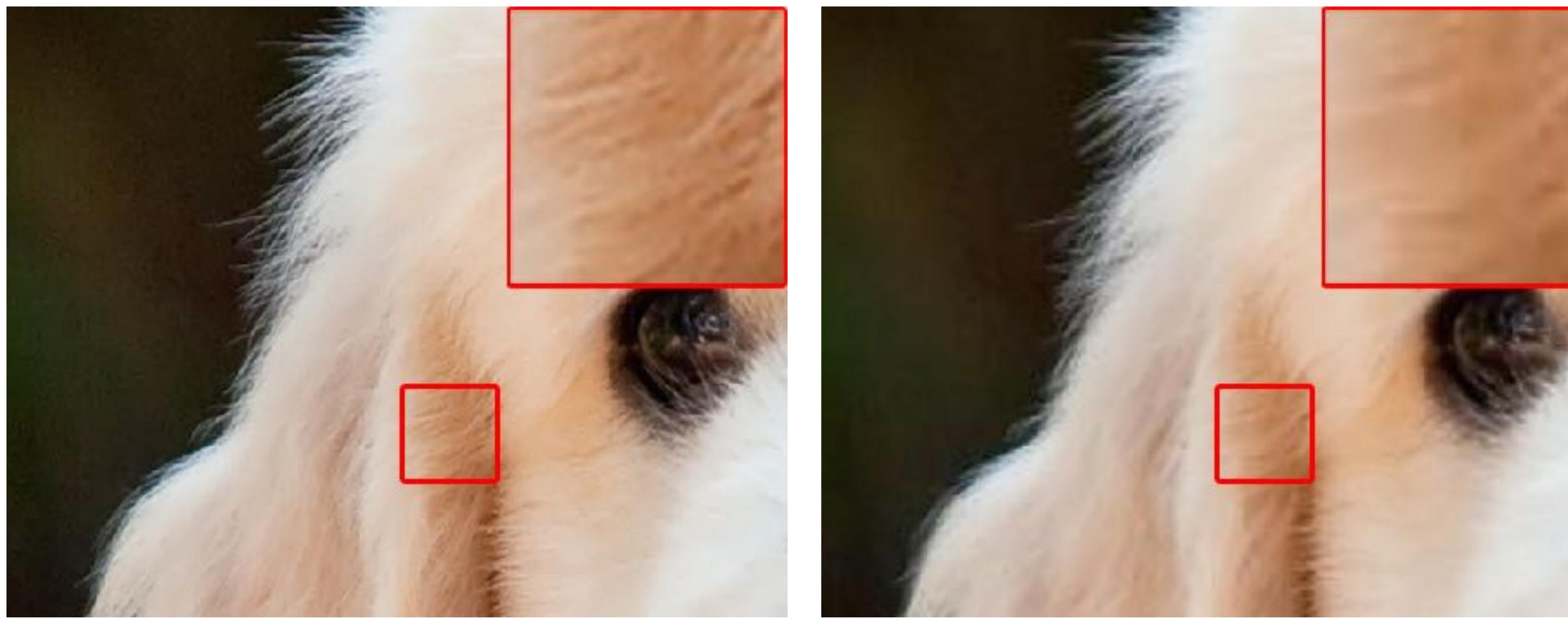
<http://engineering.flipboard.com/2015/05/scaling-convnets/>

New applications: Image Upscaling (Flipboard)



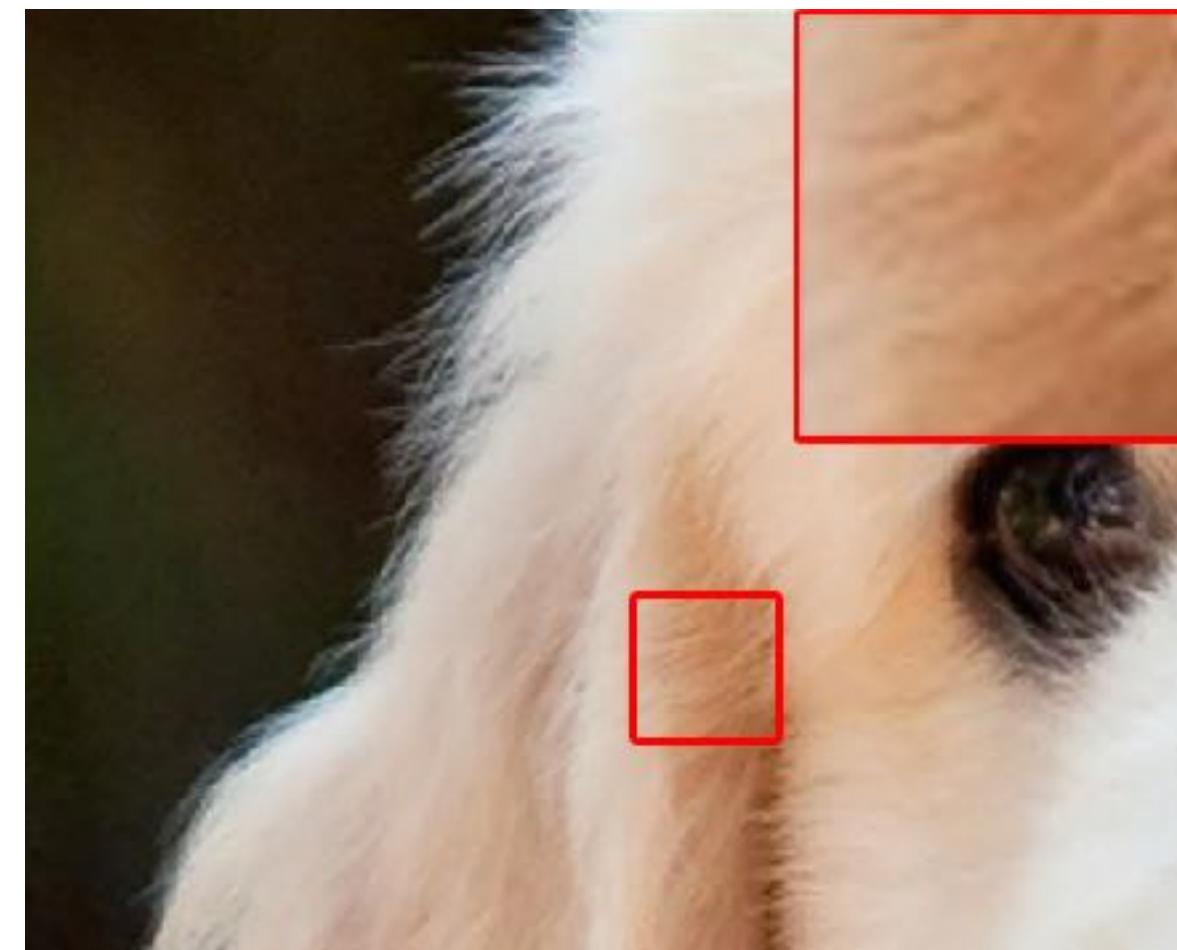
<http://engineering.flipboard.com/2015/05/scaling-convnets/>

New applications: Image Upscaling (Flipboard)



Original

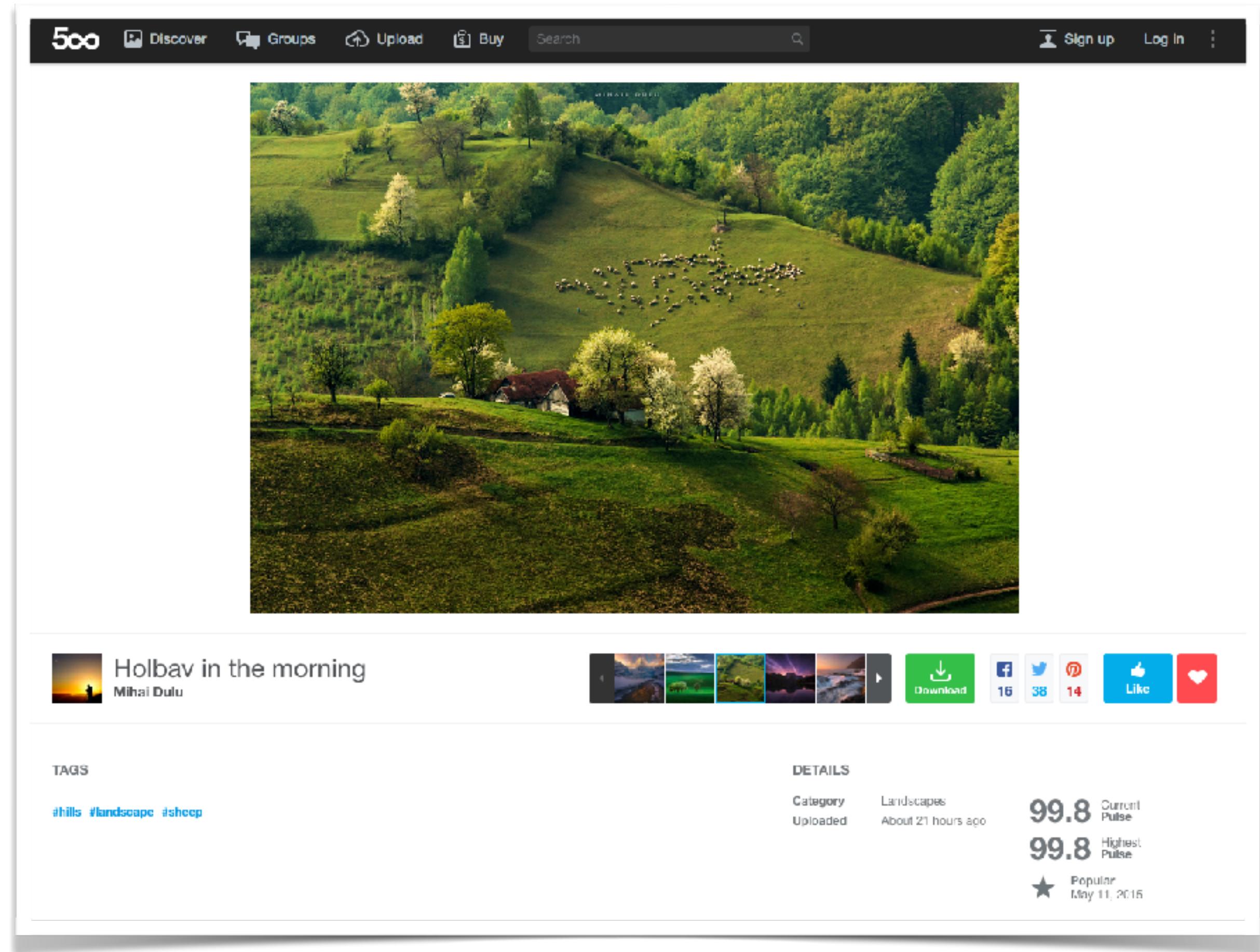
Bicubic



Model

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

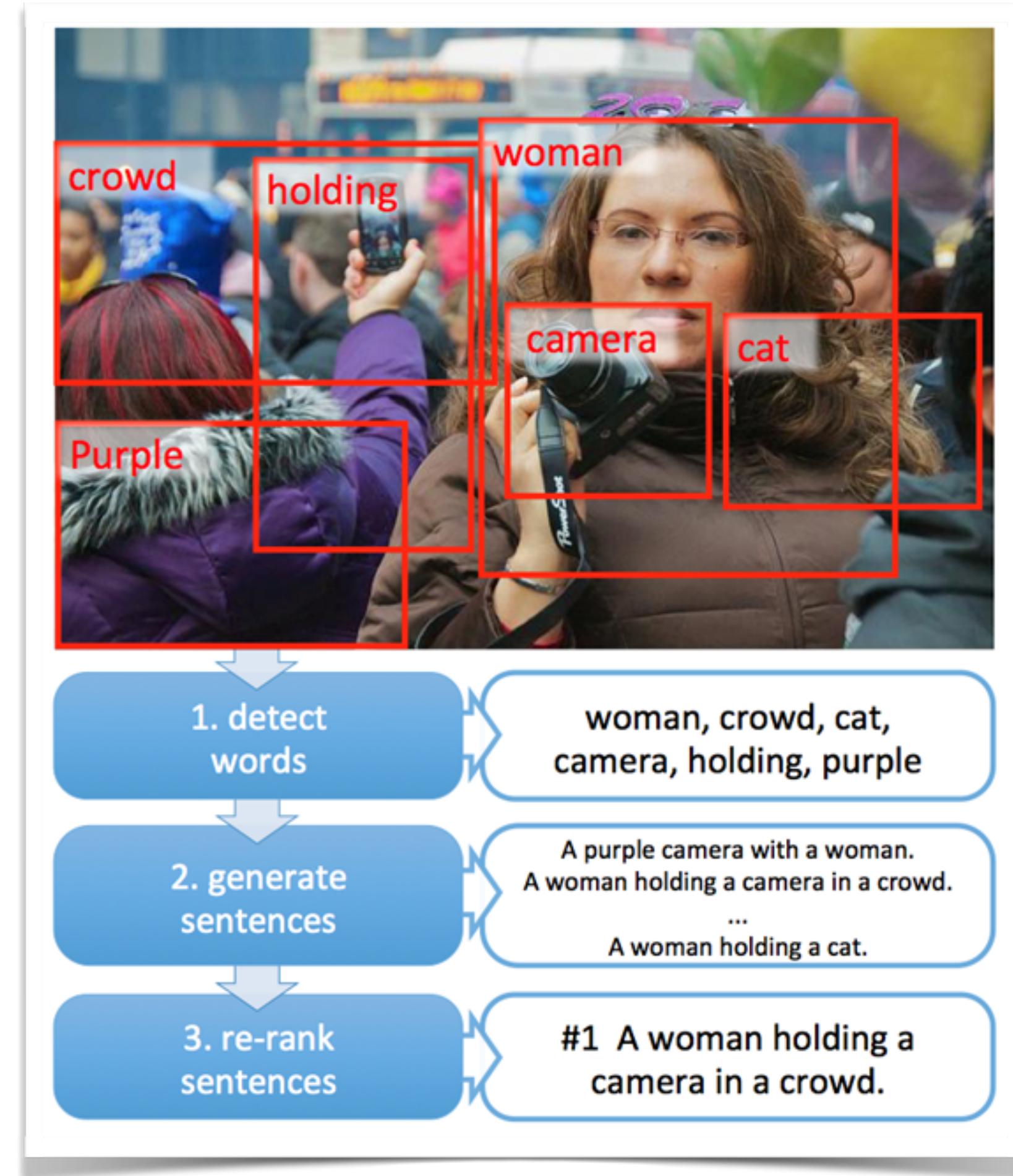
New applications: Non visual data prediction



What is Pulse?

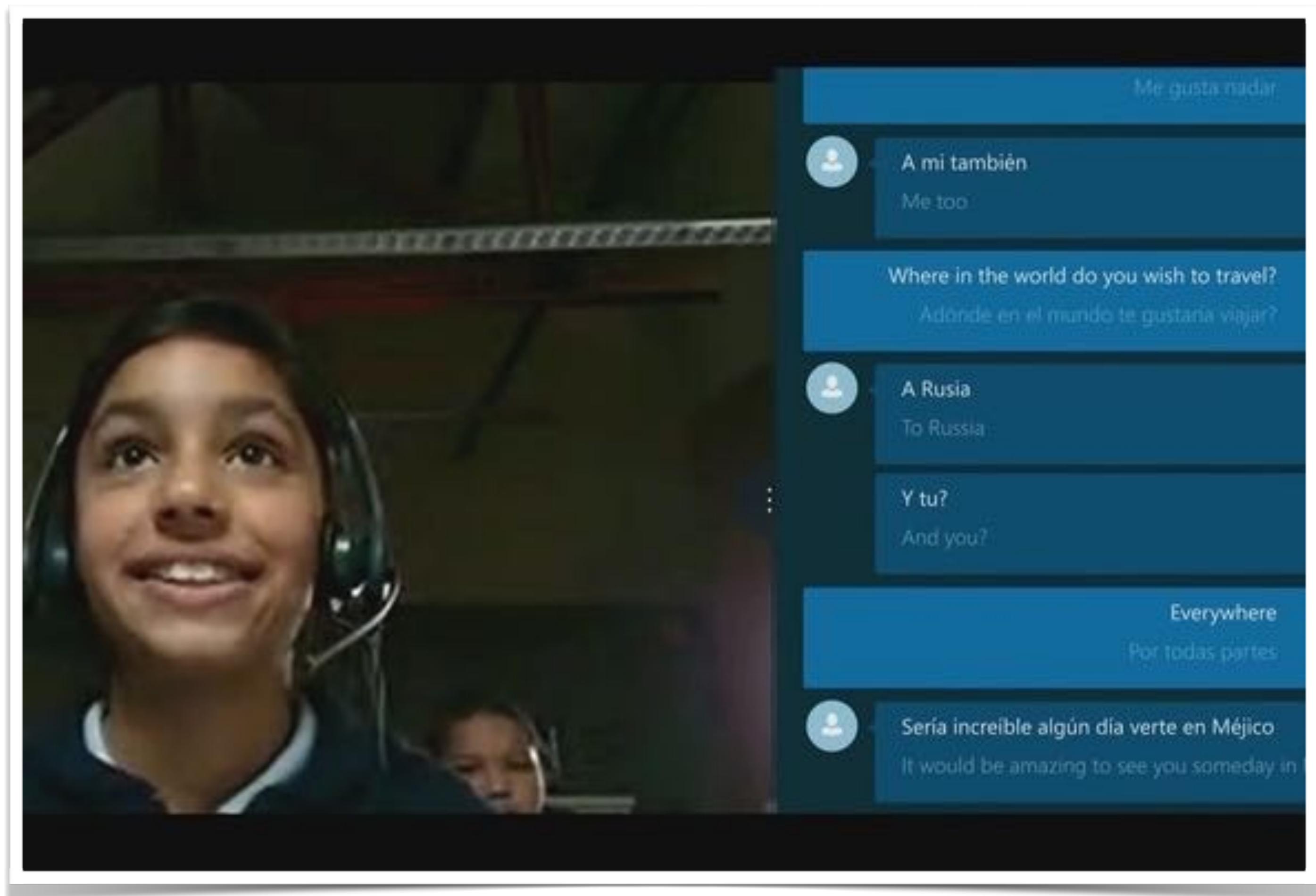
Pulse is a score out of 100 points that measures how **popular** a photo is. Pulse is calculated by an algorithm, which is unique to 500px and is based on votes (Likes & Favorites) on your photo from the community. The Pulse algorithm was designed to promote daily exposure of new photographs and photographers. It is not necessarily a measure of photograph's quality.

New applications: Automatic Image Captioning

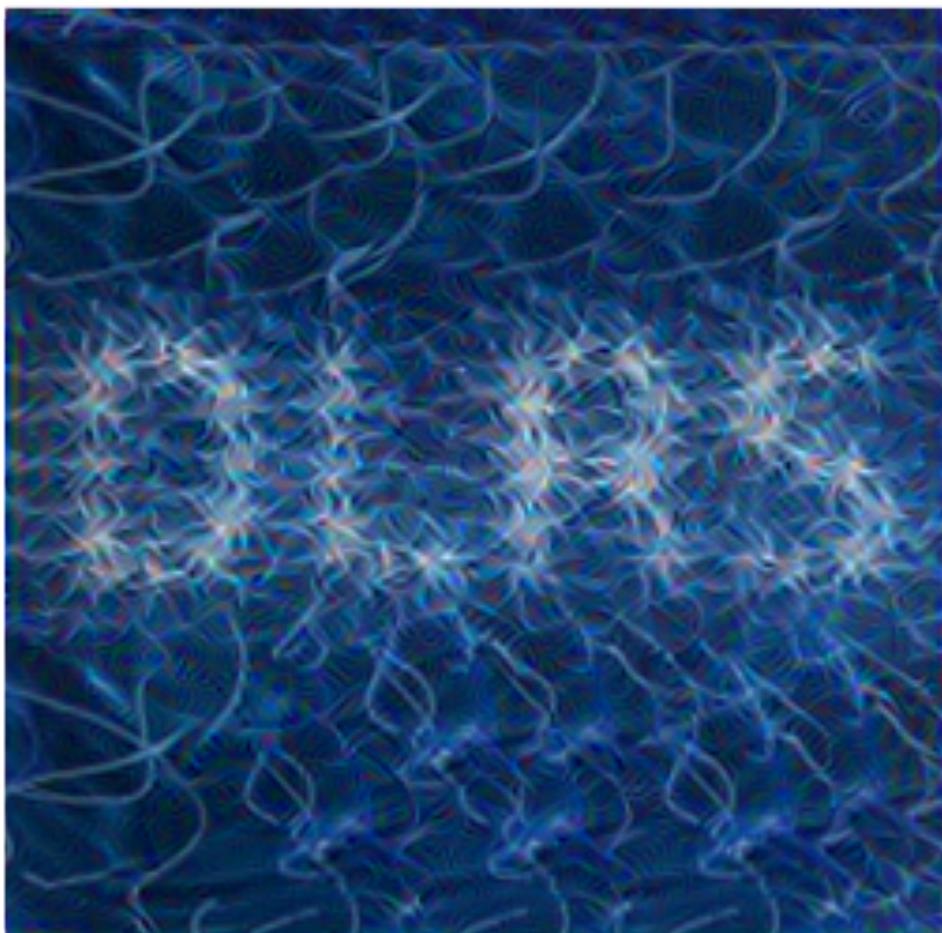


<http://blogs.technet.com/b/machinelearning/archive/2014/11/18/rapid-progress-in-automatic-image-captioning.aspx>

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'. Below the bio, there are tabs for 'All', 'Tracks', 'Playlists', and 'Reposts', with 'All' being the active tab. Under the tracks section, there is a preview for a track titled 'deepjazz on Metheny' posted 14 days ago, categorized under '#Electronic'. Three specific tracks are listed below: '1 deepjazz On Metheny ... 1 Epoch', '2 deepjazz On Metheny ... 16 Epochs', and '3 deepjazz On Metheny ... 32 Epochs'. To the right of the tracks, there are statistics: 104 Followers, 1 Following, and 6 Tracks. A message from the AI states: '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'. There are also links to 'my source code (GitHub)!' and 'deepjazz.io'. At the bottom right, there is a 'View all' link.

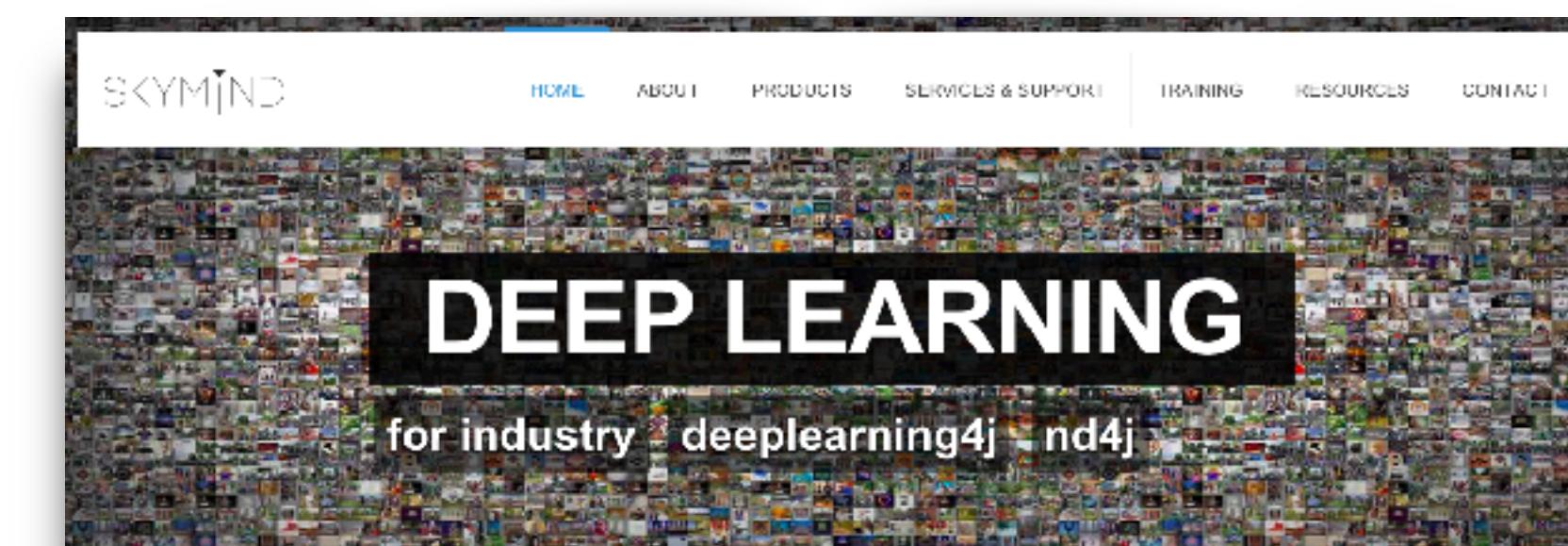
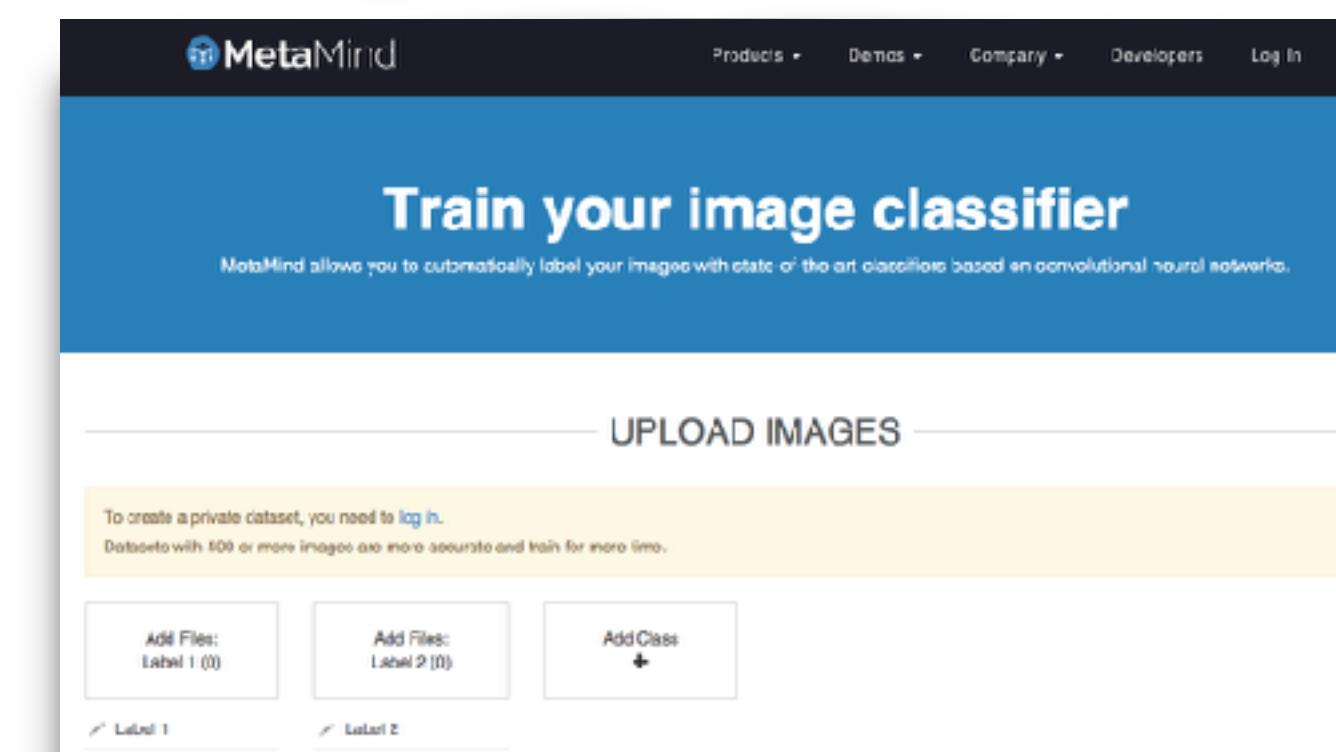
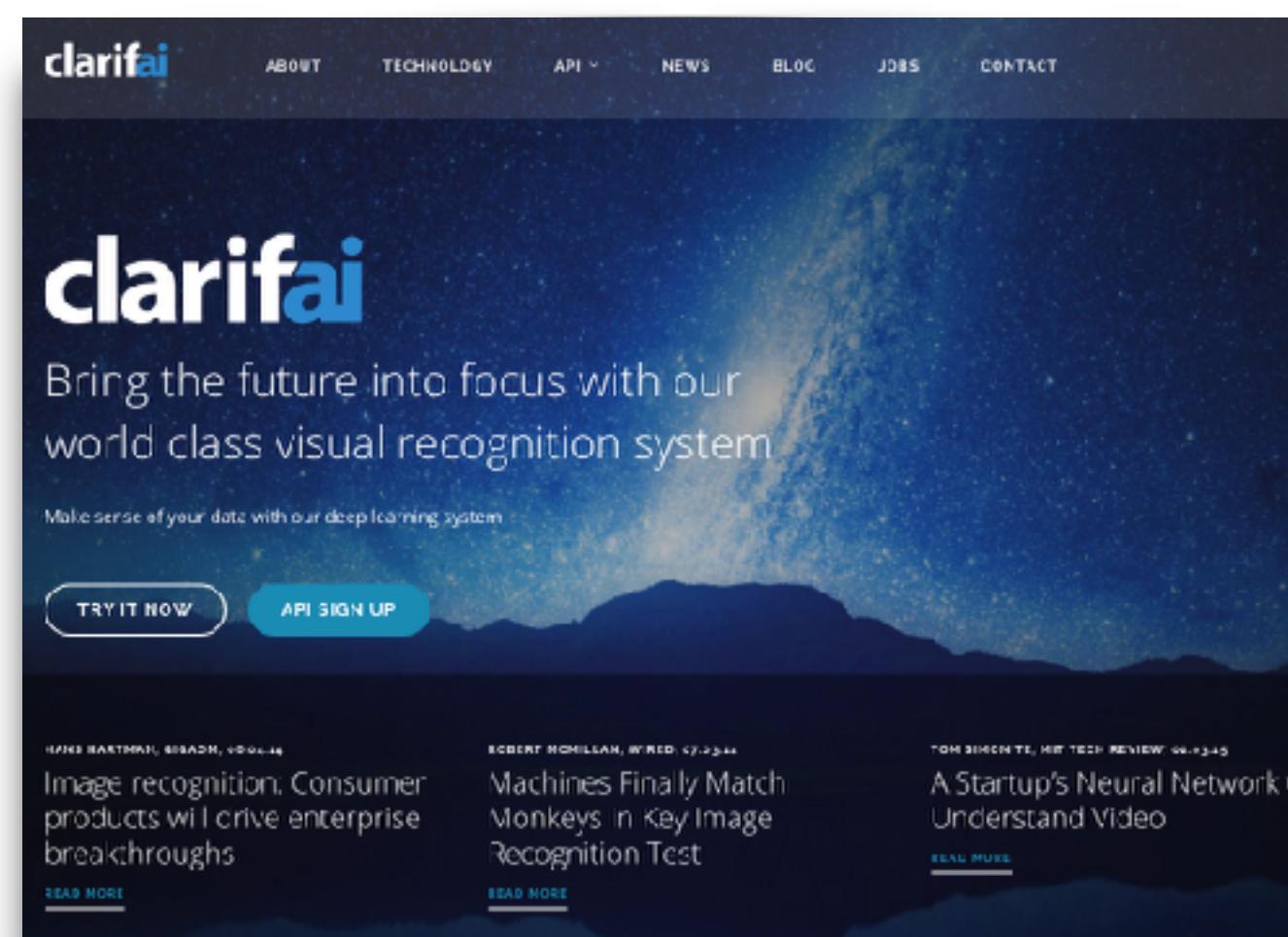
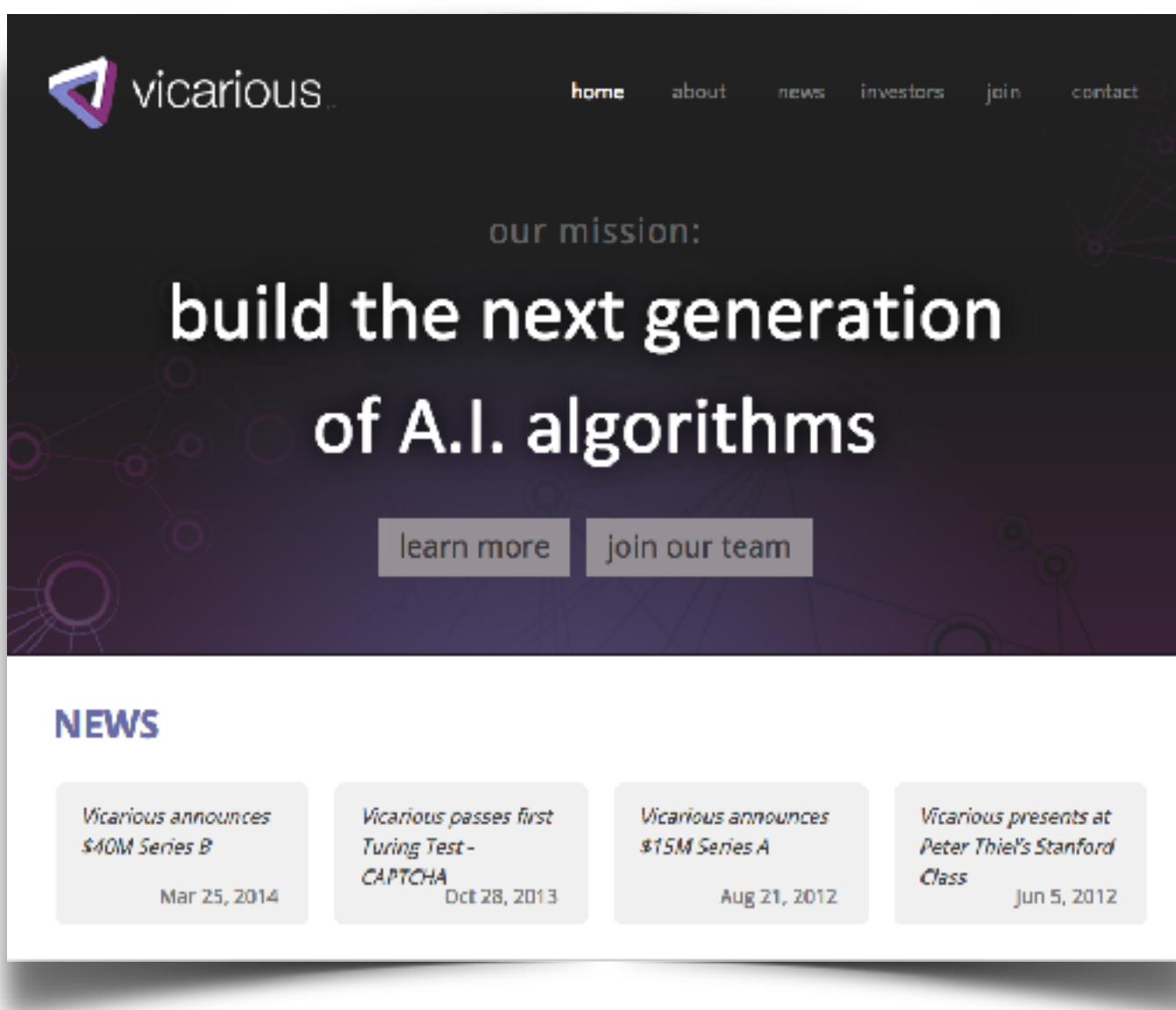
Reinforcement learning.

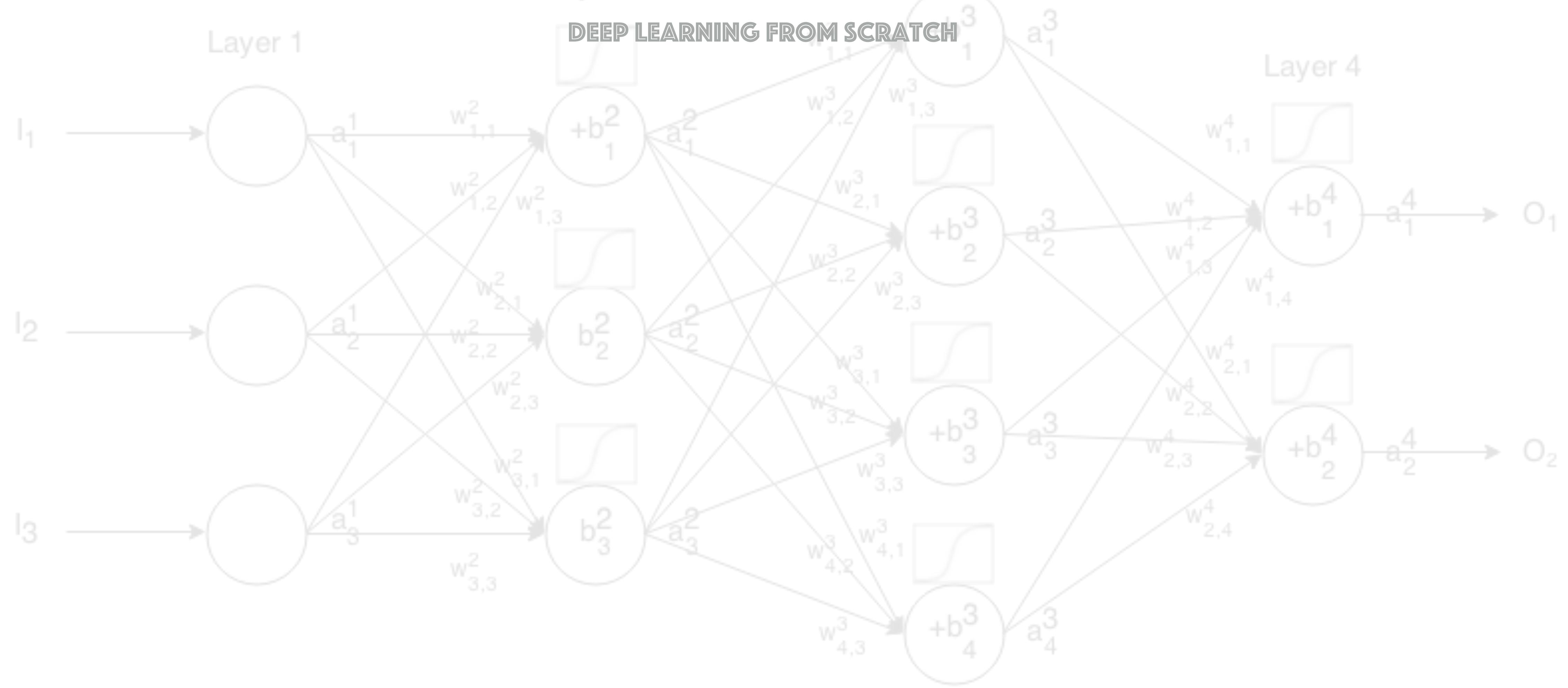


Go



Start Ups





What is Deep Learning?

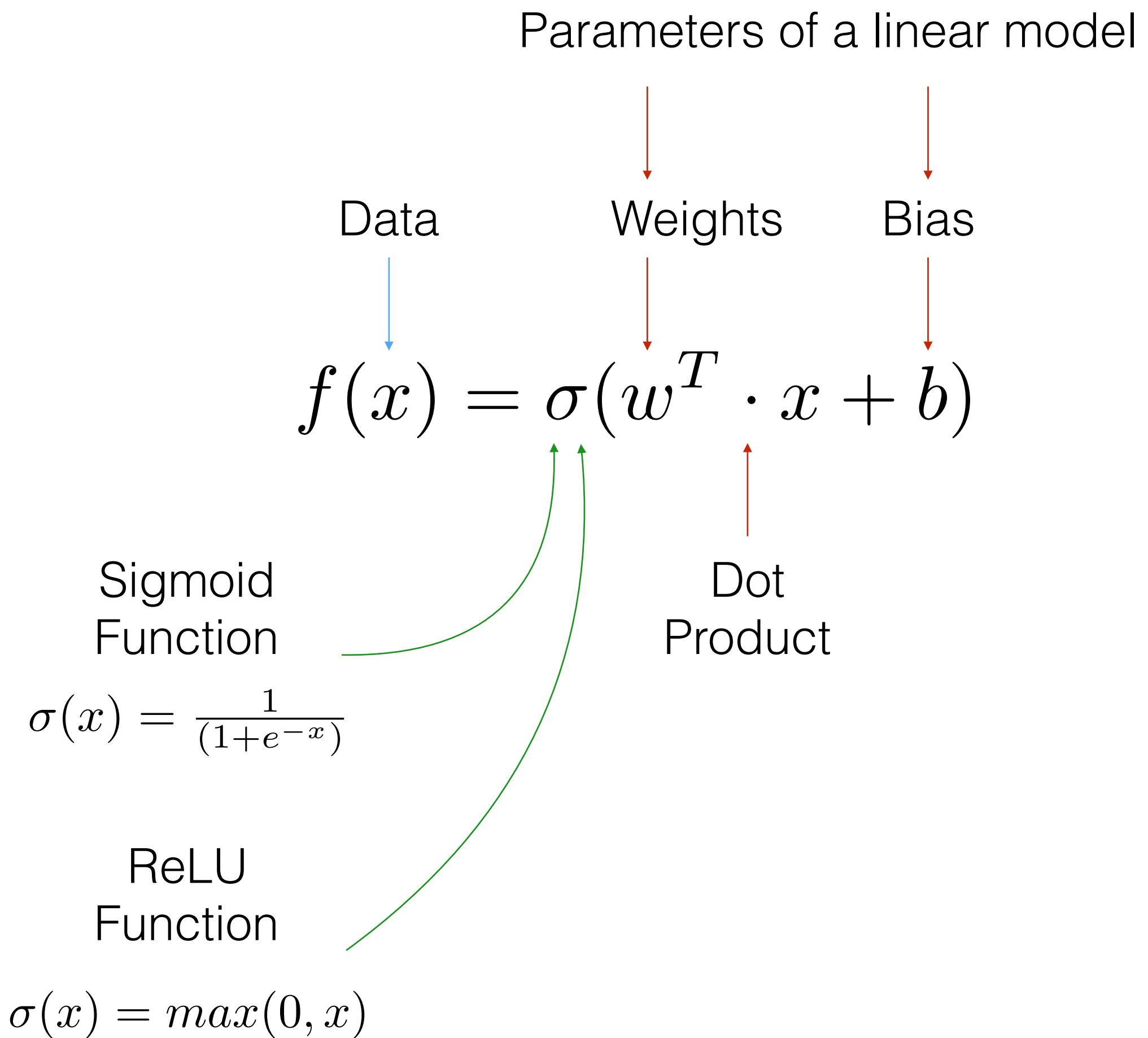
Learning from Data

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

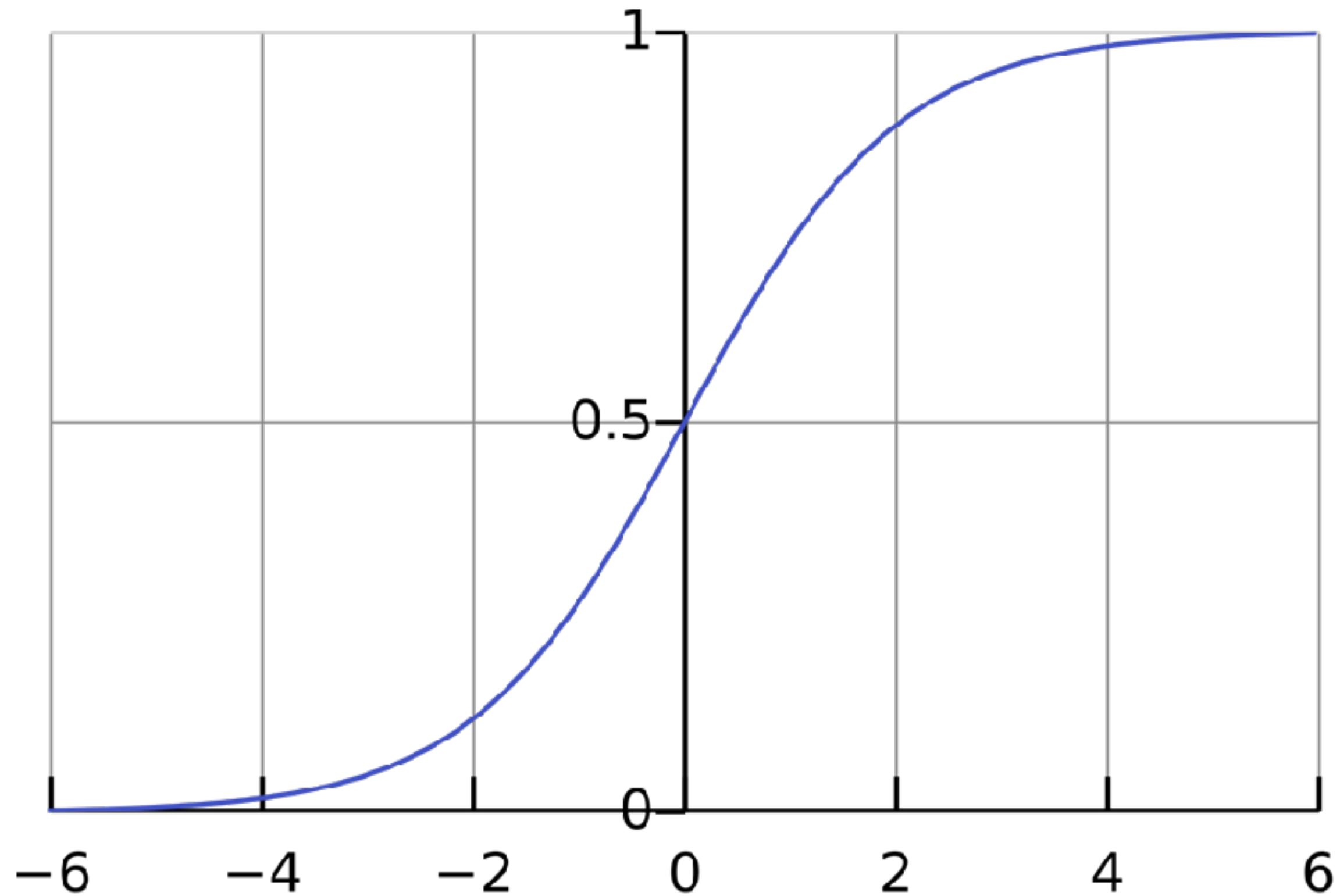
Learn a function $f_w : x \rightarrow y$ to predict on new inputs x .

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

1-layer neural net model

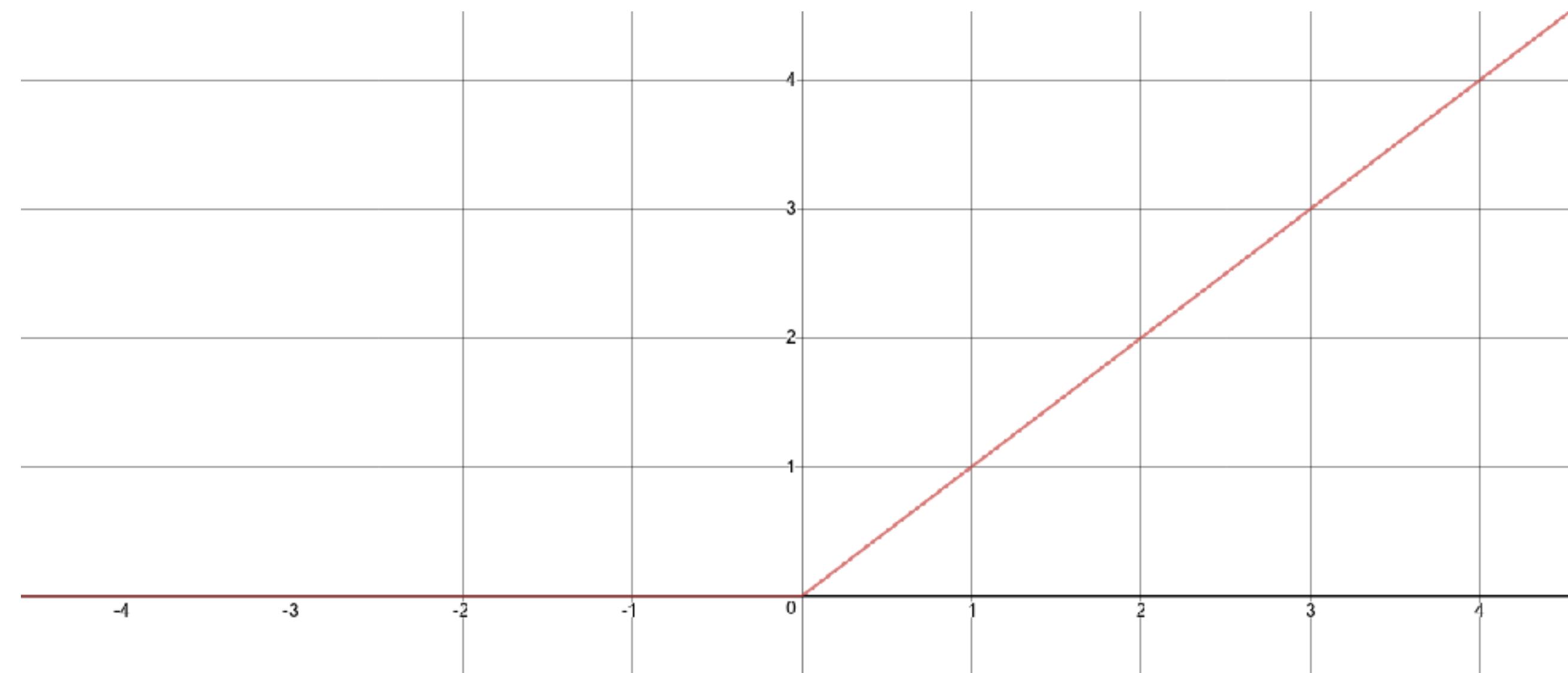


Non-linearity



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

Non-linearity



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

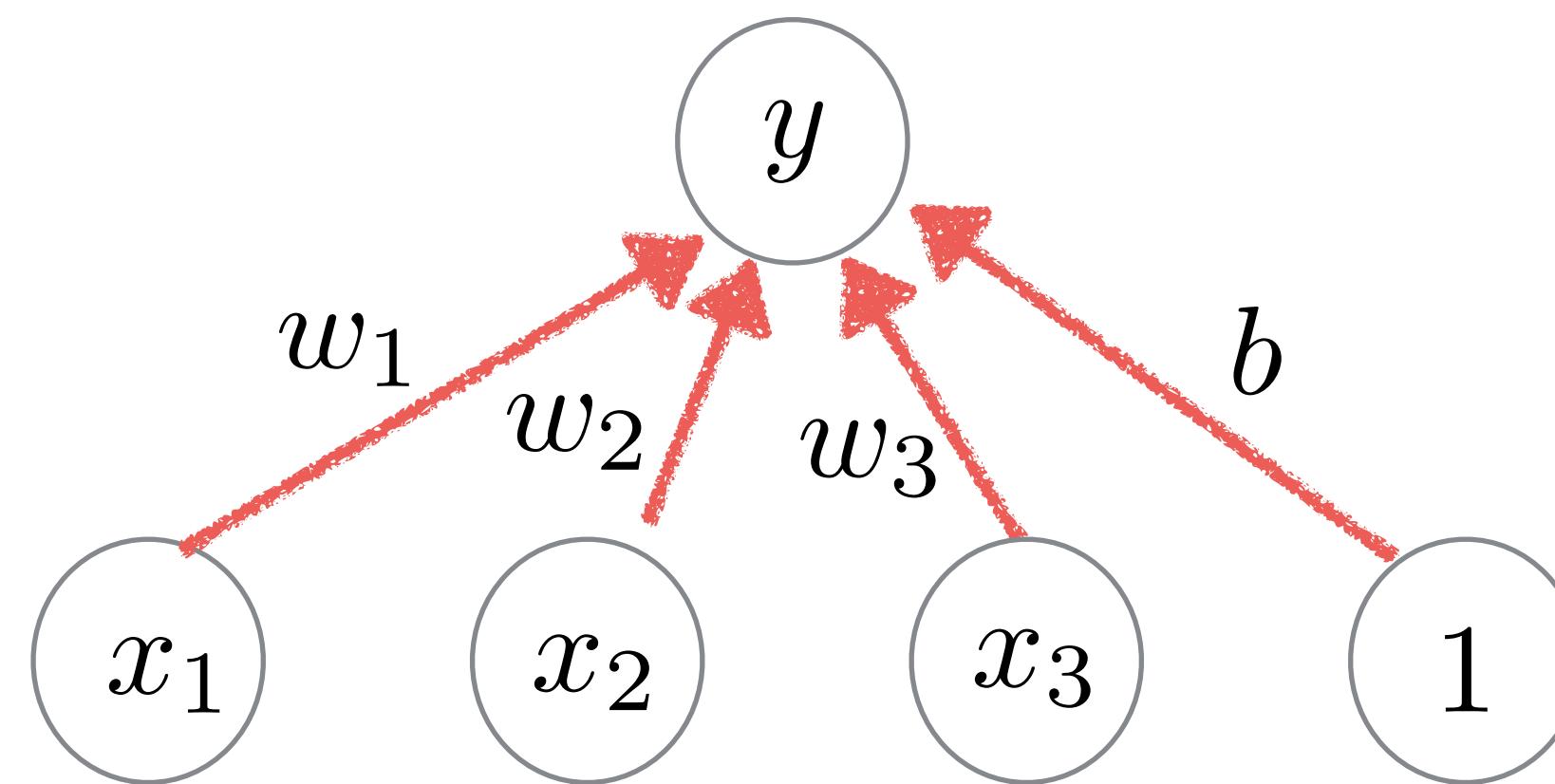
Non-linearity

Table 3: Non-linearities tested.

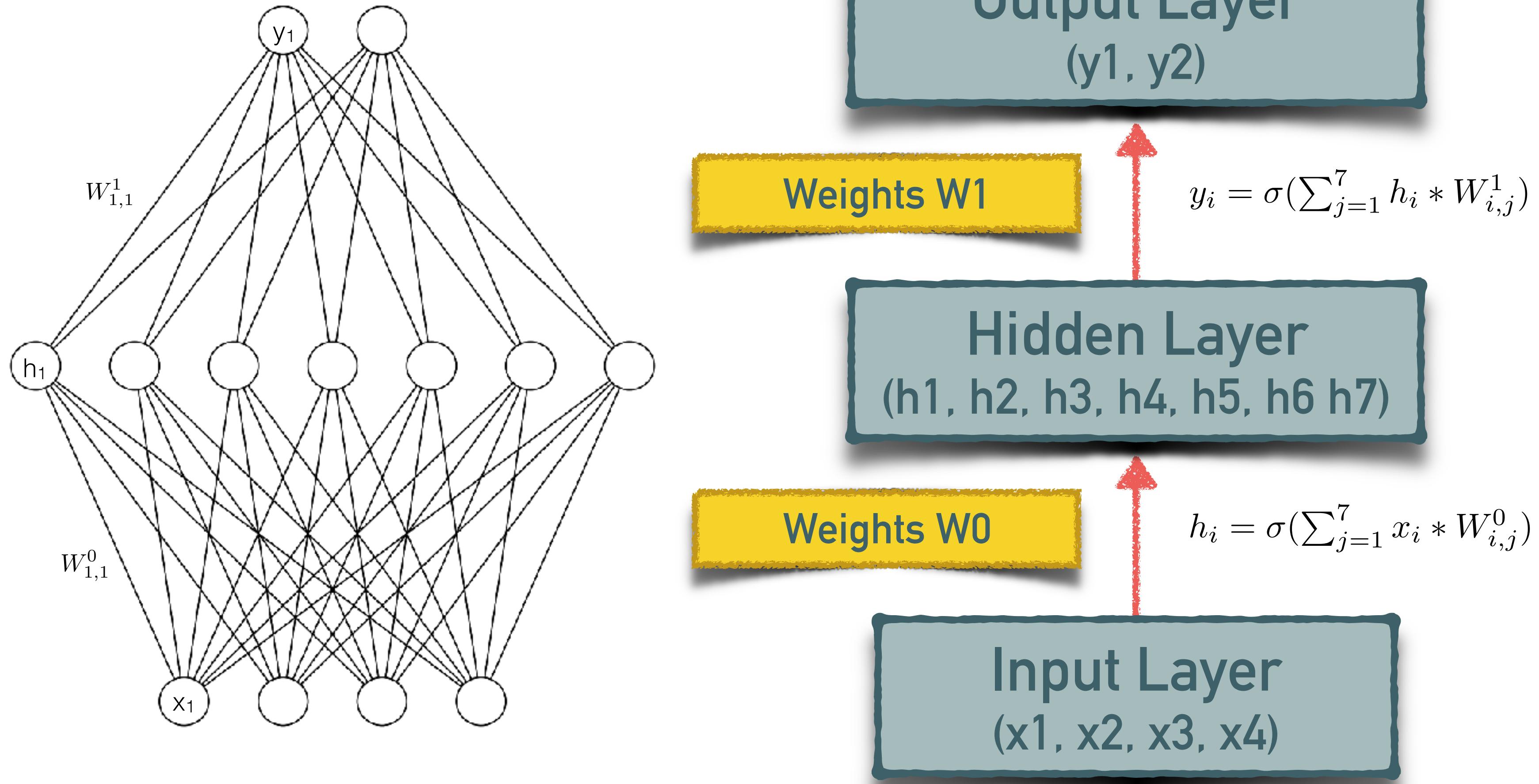
Name	Formula	Year
none	$y = x$	-
sigmoid	$y = \frac{1}{1+e^{-x}}$	1986
tanh	$y = \frac{e^{2x}-1}{e^{2x}+1}$	1986
ReLU	$y = \max(x, 0)$	2010
(centered) SoftPlus	$y = \ln(e^x + 1) - \ln 2$	2011
LReLU	$y = \max(x, \alpha x), \alpha \approx 0.01$	2011
maxout	$y = \max(W_1x + b_1, W_2x + b_2)$	2013
APL	$y = \max(x, 0) + \sum_{s=1}^S a_i^s \max(0, -x + b_i^s)$	2014
VLReLU	$y = \max(x, \alpha x), \alpha \in [0.1, 0.5]$	2014
RReLU	$y = \max(x, \alpha x), \alpha = \text{random}(0.1, 0.5)$	2015
PReLU	$y = \max(x, \alpha x), \alpha$ is learnable	2015
ELU	$y = x, \text{ if } x \geq 0, \text{ else } \alpha(e^x - 1)$	2015

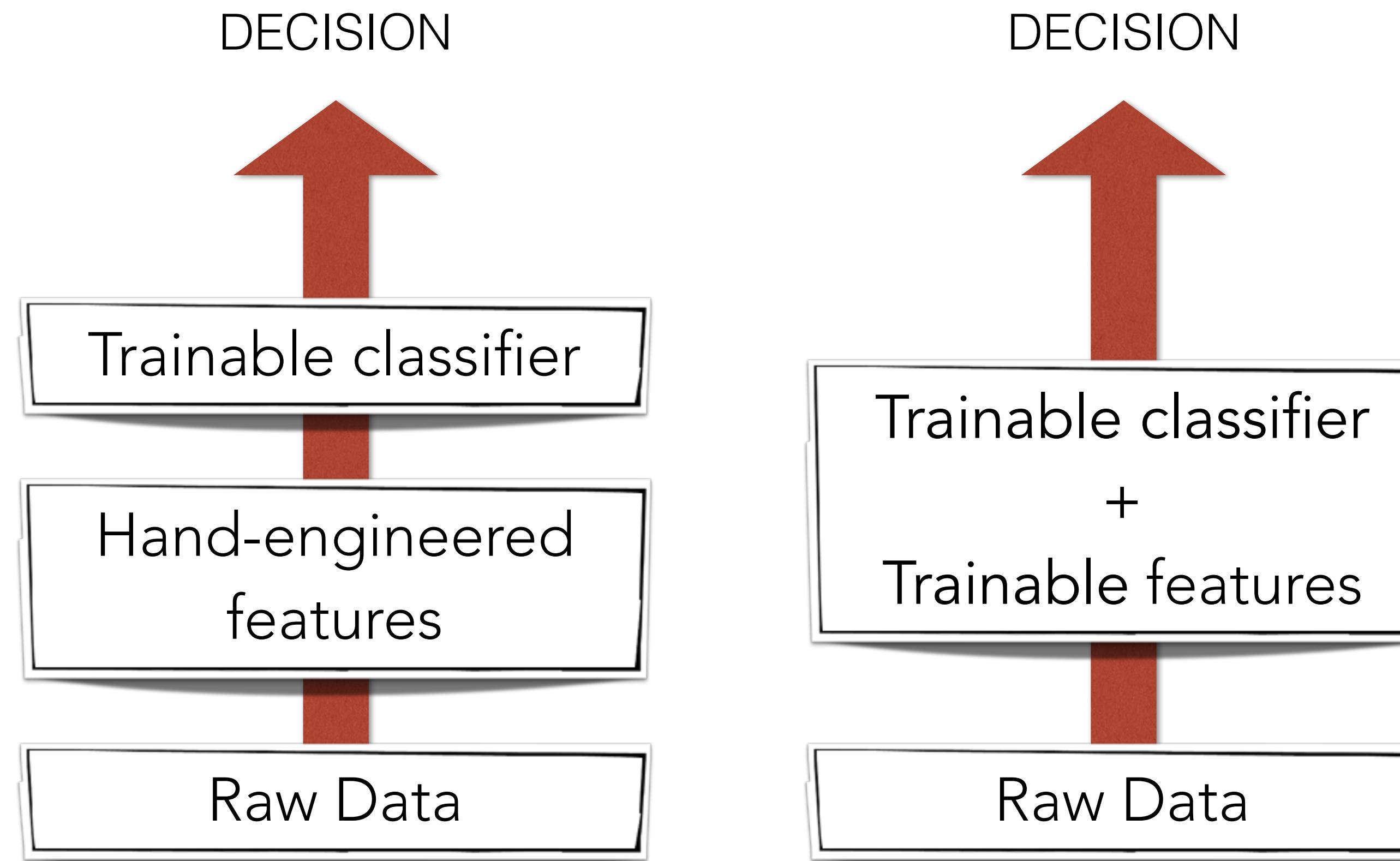
Graphical representation of 1-layer neural net model

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



2-layer neural net model

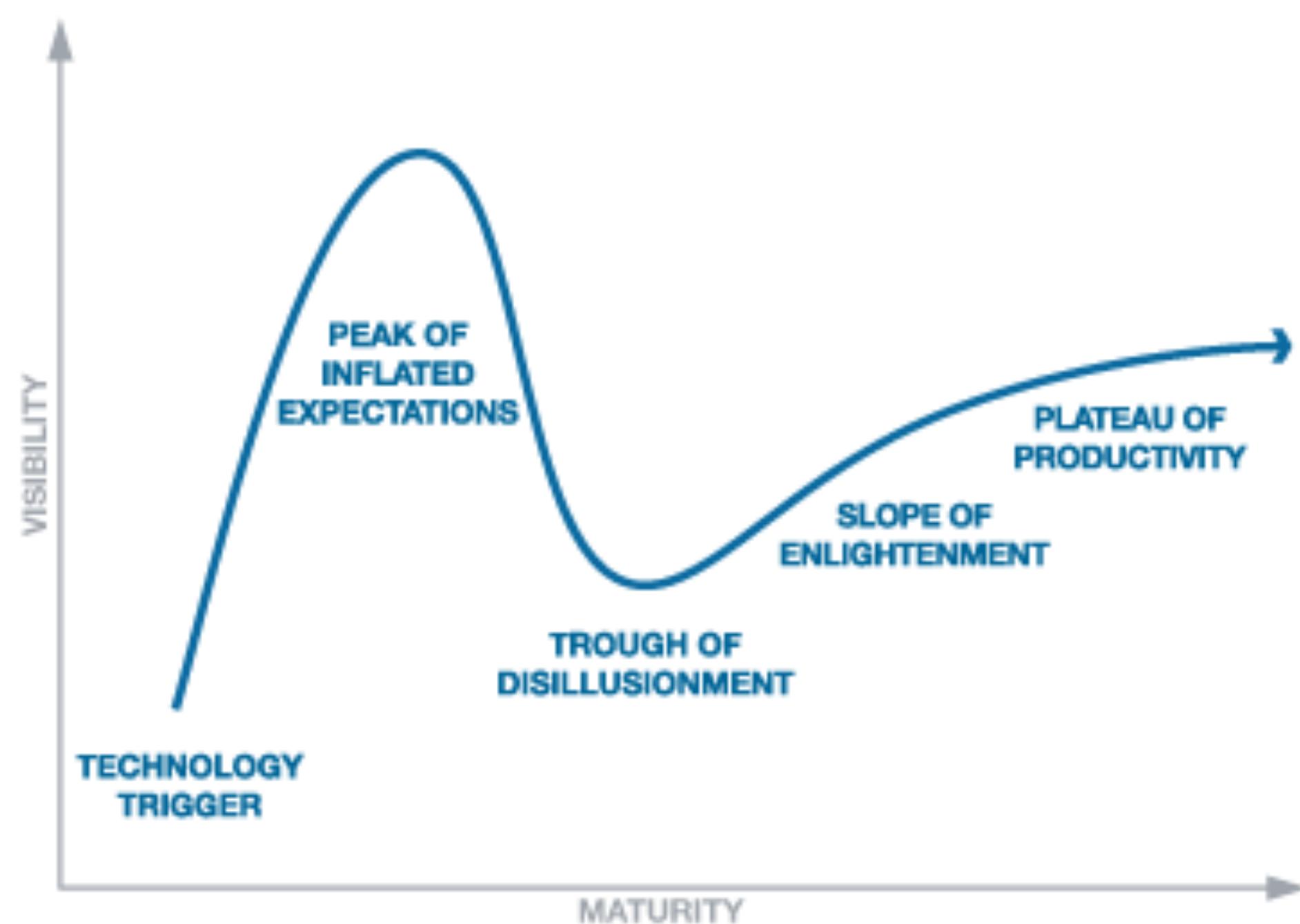




STANDARD MACHINE
LEARNING

DEEP LEARNING

Is Deep Learning Overhyped?



Hype is not new

1958 New York Times...



NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's \$2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of \$100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

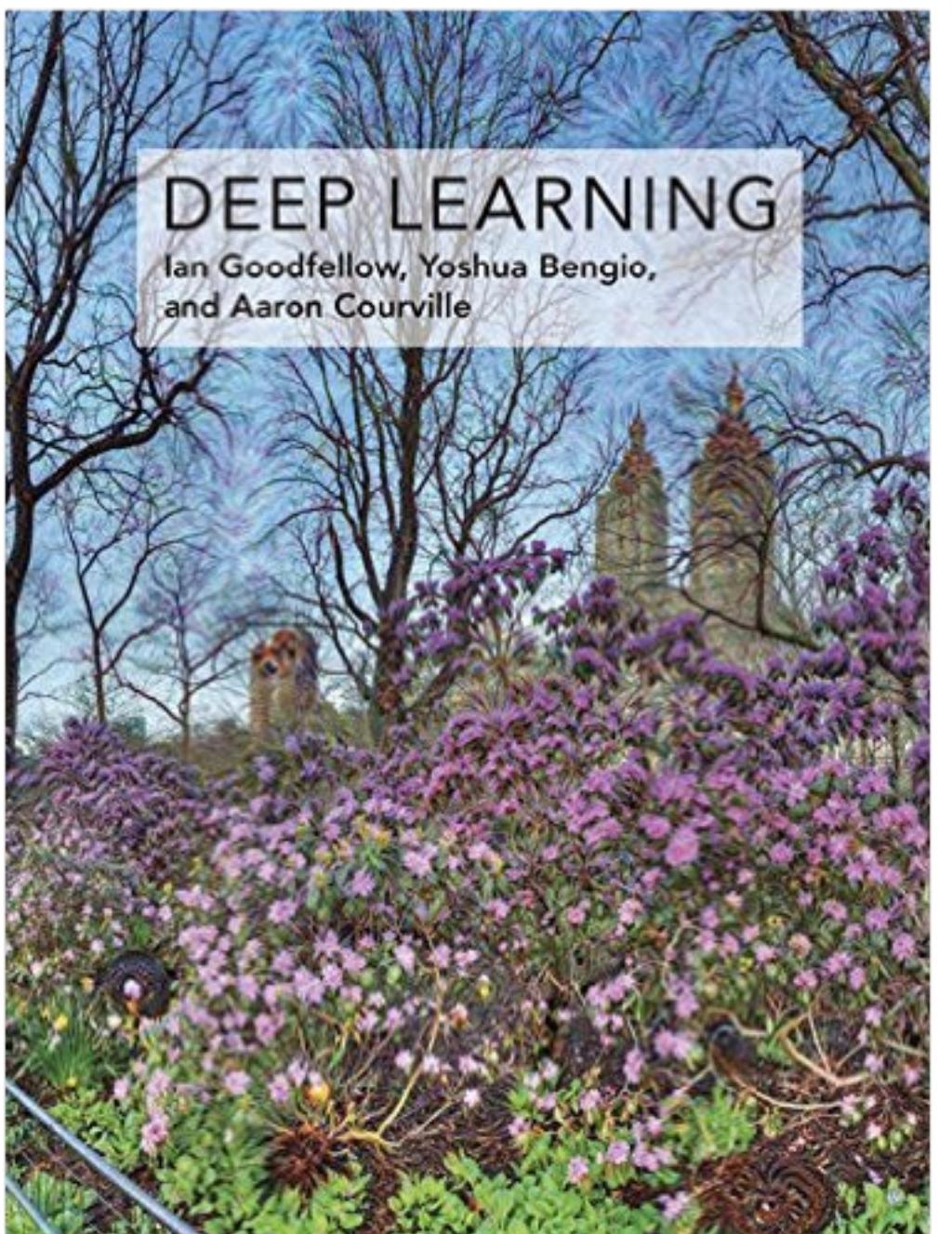
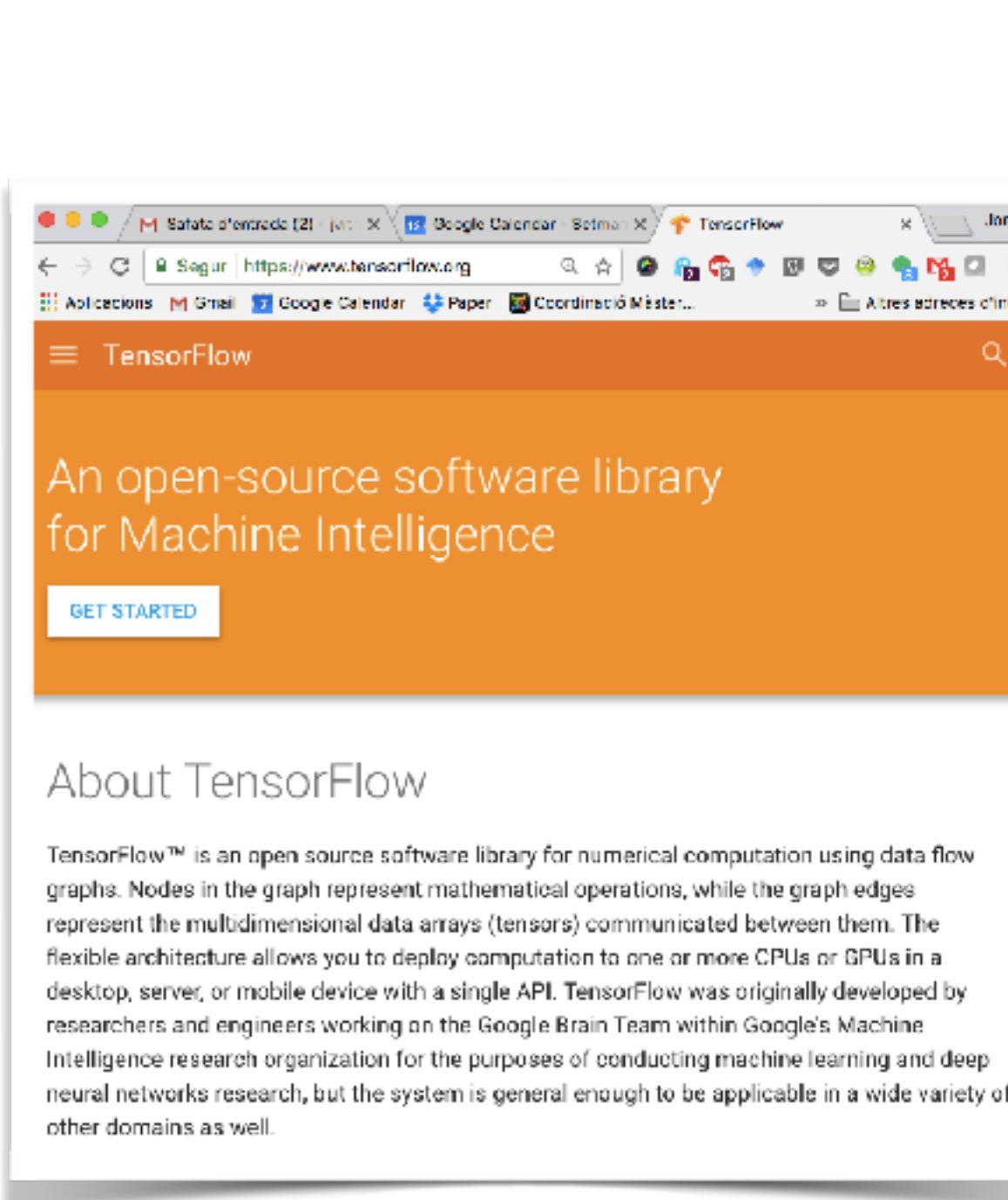
Learns by Doing

In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

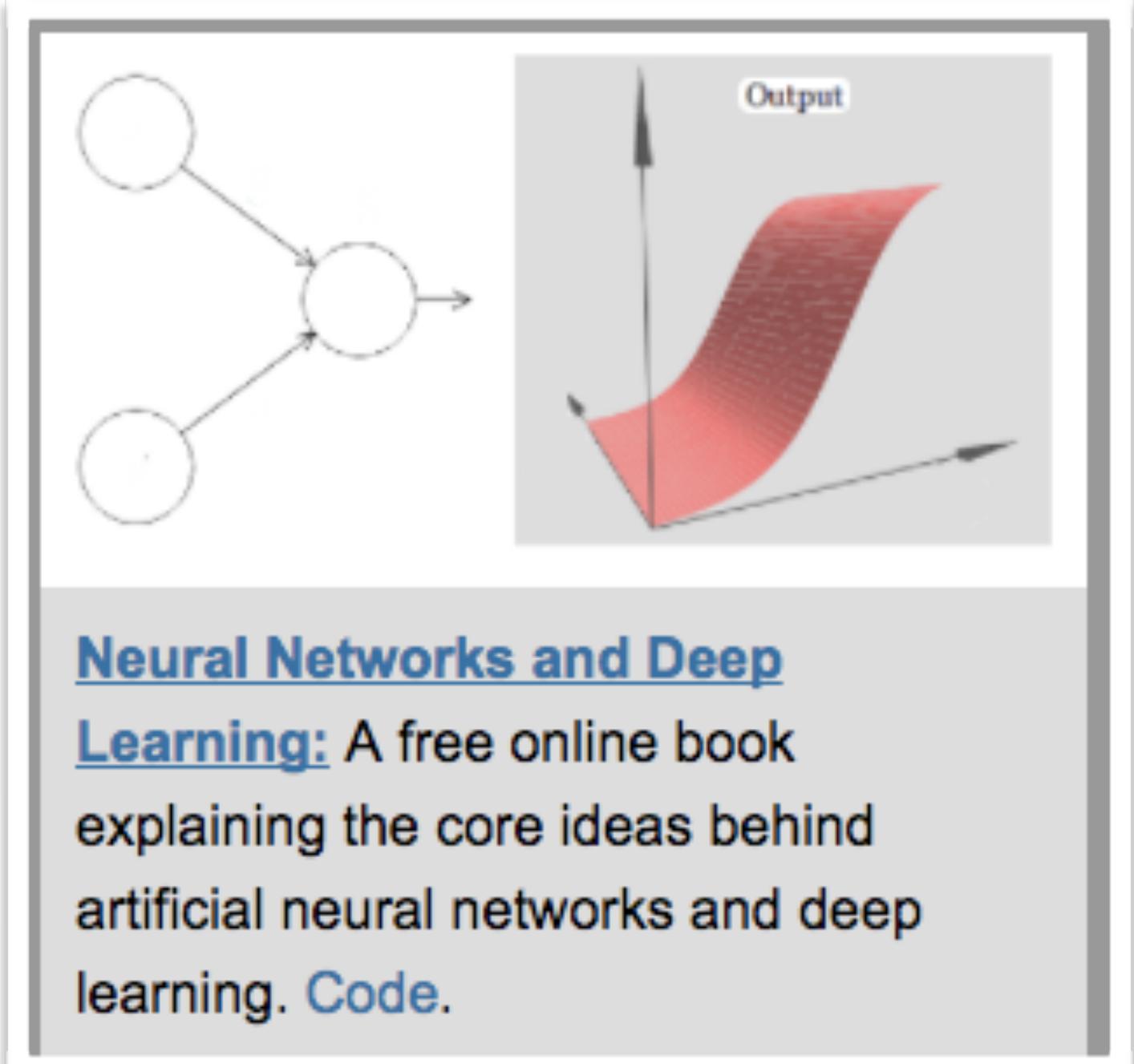
Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

Bibliography



Deep Learning, Yoshua Bengio, Ian Goodfellow, Aaron Courville, MIT Press, 2017
<http://www.deeplearningbook.org/>



[Neural Networks and Deep Learning](#): A free online book explaining the core ideas behind artificial neural networks and deep learning. [Code](#).

By Michael Nielsen / Jan 2017
<http://neuralnetworksanddeeplearning.com/>

The screenshot shows the Keras Documentation website. The header reads "Keras Documentation". The main content area is titled "Keras: Deep Learning library for Theano and TensorFlow". It includes sections like "Getting started", "FAQ", "Models", and "Layers". The "Getting started" section contains a bulleted list of features, and the "Models" section lists "Sequential", "Model (functional API)", and "Layers". At the bottom, it says "Read the documentation at [Keras.io](#)".