UNIVERSITAT DE BARCELONA

What is Deep Learning?

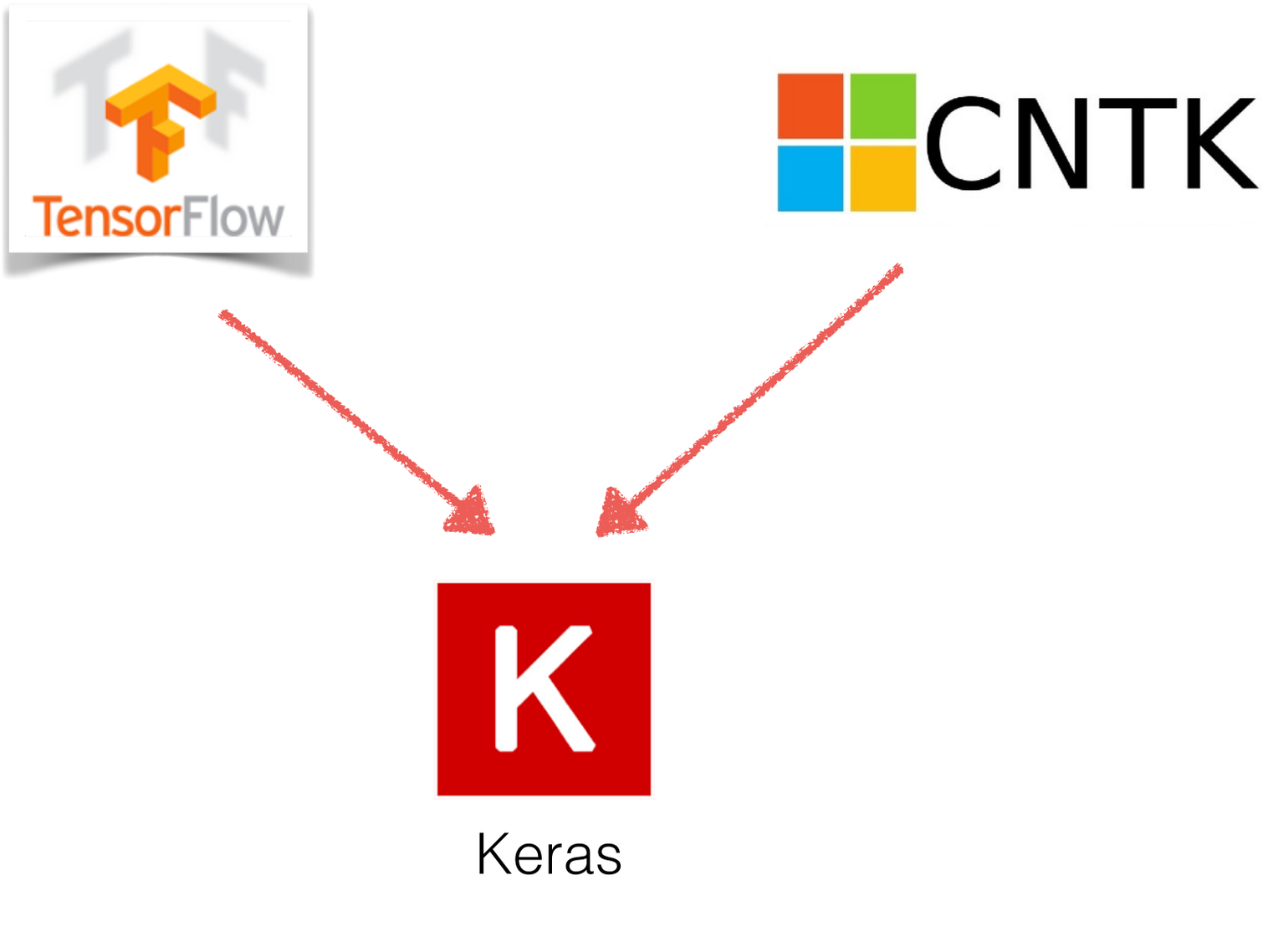
Jordi Vitrià
jordi.vitria@ub.edu

Course Contents

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, Keras** and derivates. You can train (small) deep models in your laptop.

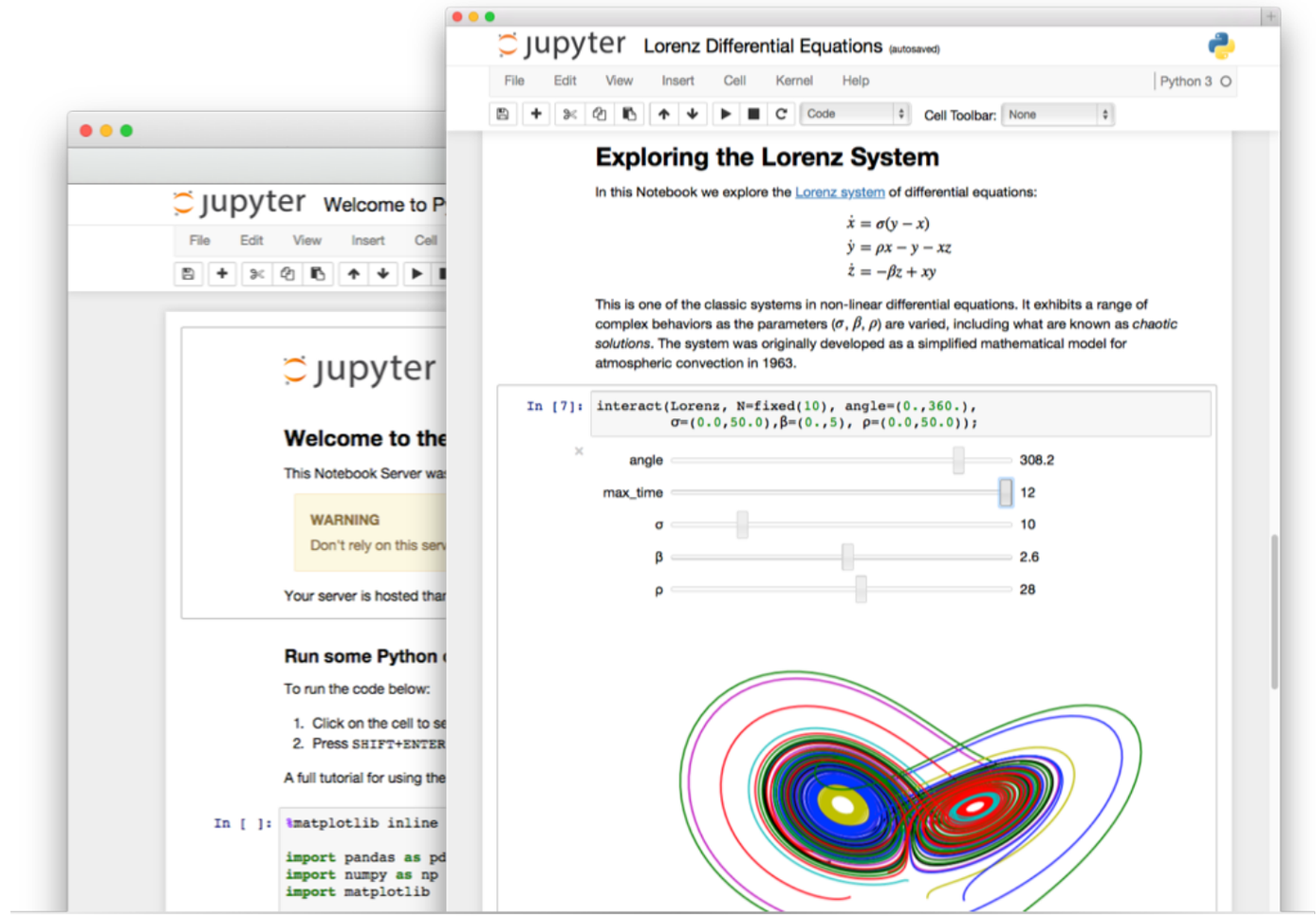
BackEnds

Wrappers



Approach

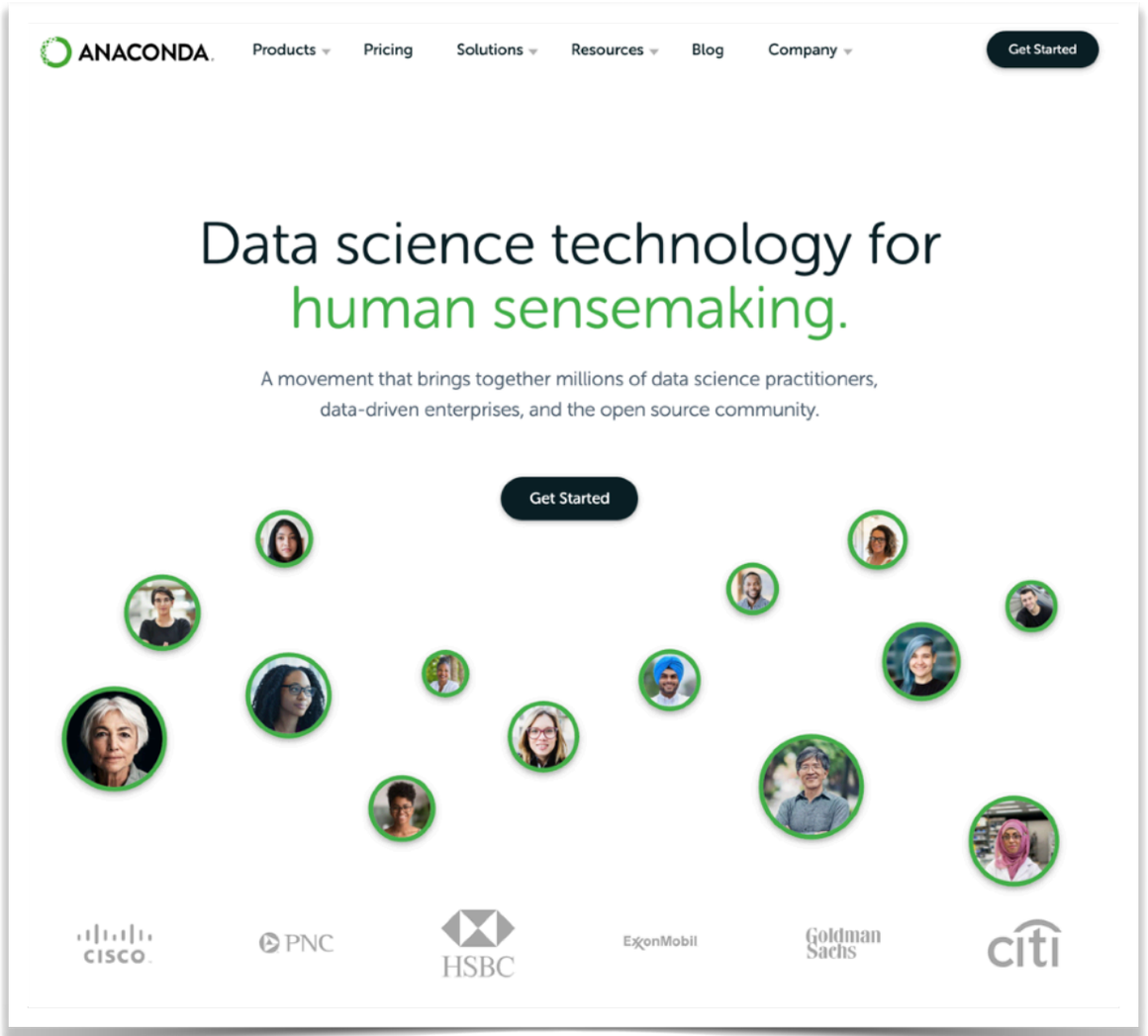
We will illustrate all contents with **Jupyter notebooks**, a web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text.



How to run a Jupyter Notebook?

1

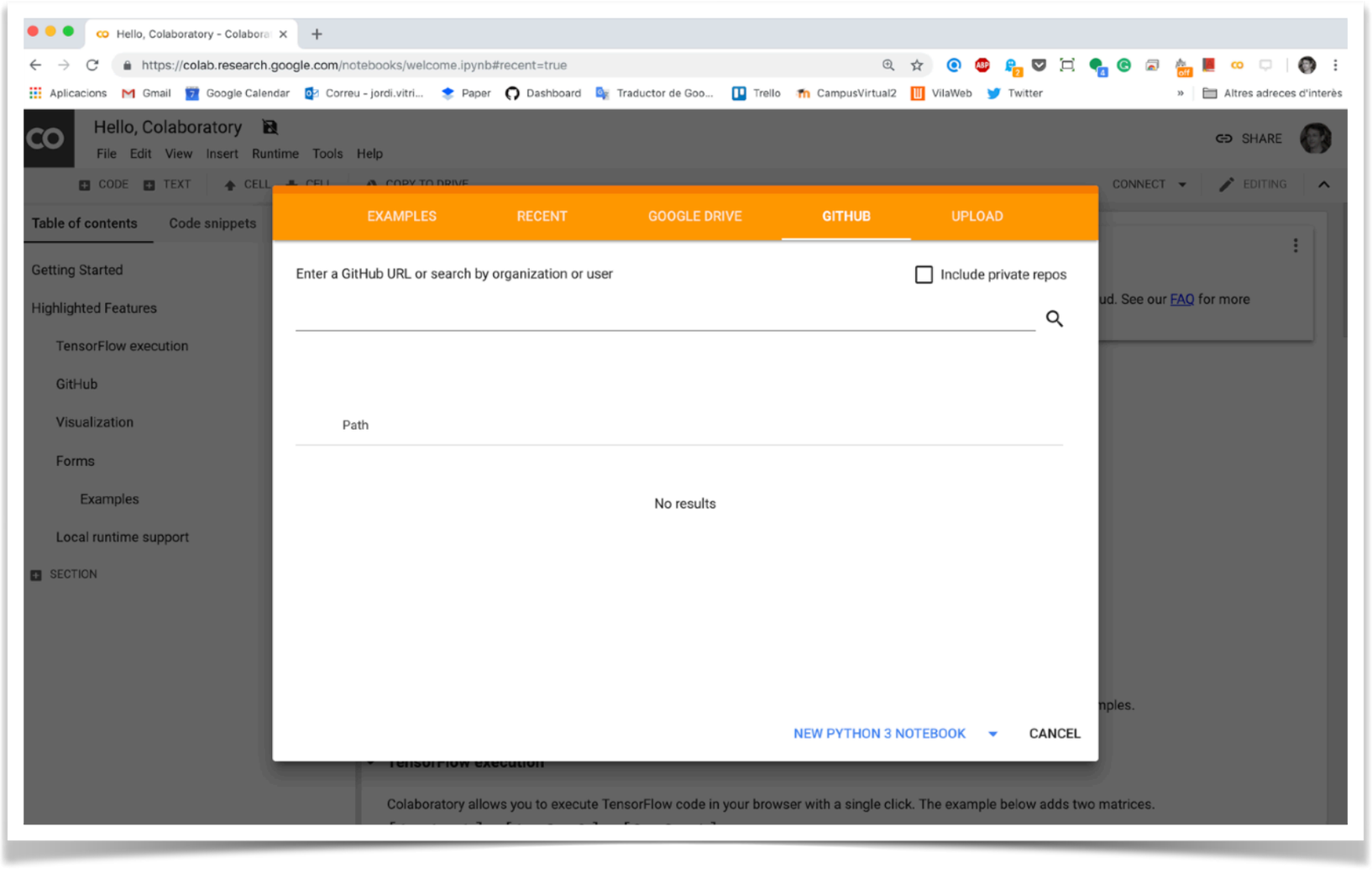
The easiest way for a beginner to get started with Jupyter Notebooks is by installing **Anaconda**. Anaconda is the most widely used Python distribution for data science and comes pre-loaded with all the most popular libraries and tools.



How to run a Jupyter Notebook?

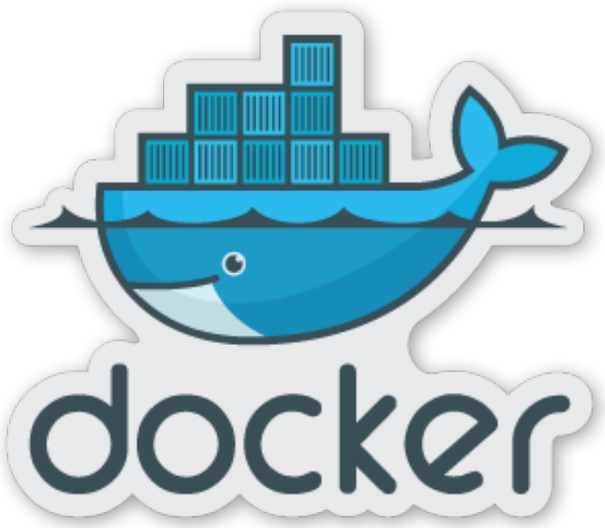
2

Colaboratory is a hosted (Google) Jupyter notebook environment that is free to use and requires no setup.



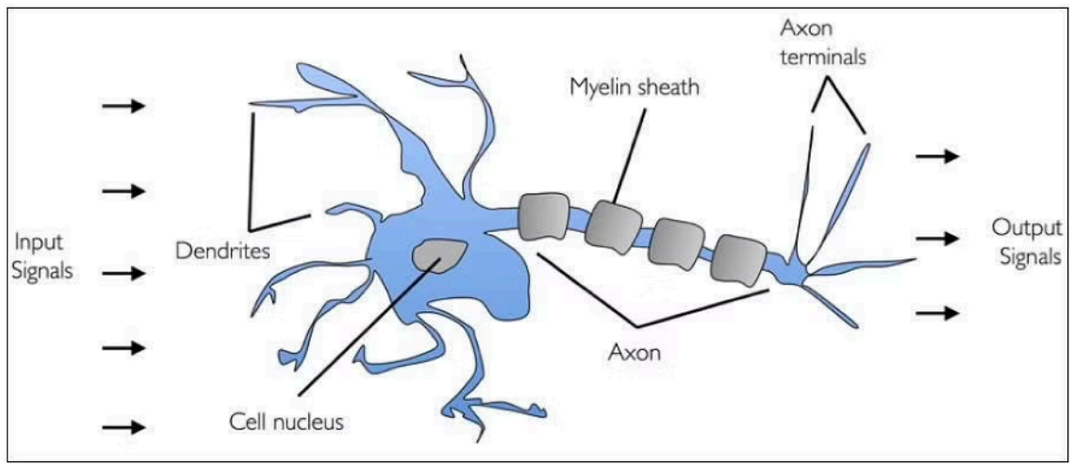
How to run a Jupyter Notebook? 3

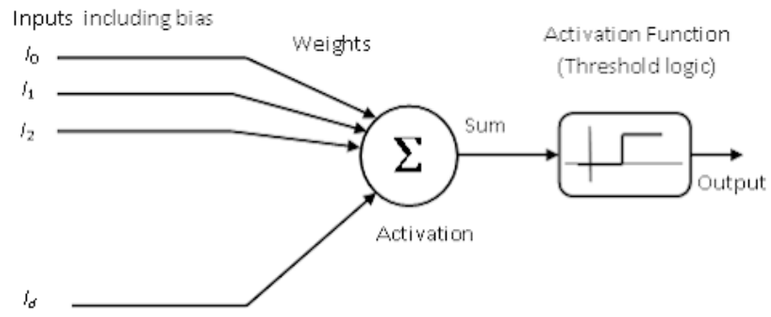
Docker provides the ability to build a runtime environment that not only remains isolated from other running containers, but also can be deployed to multiple locations in a repeatable way. Docker also uses a text document — a Dockerfile — that contains all the commands to assemble an image, which will meet our need to document the build environment.



The road to DL

- 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.





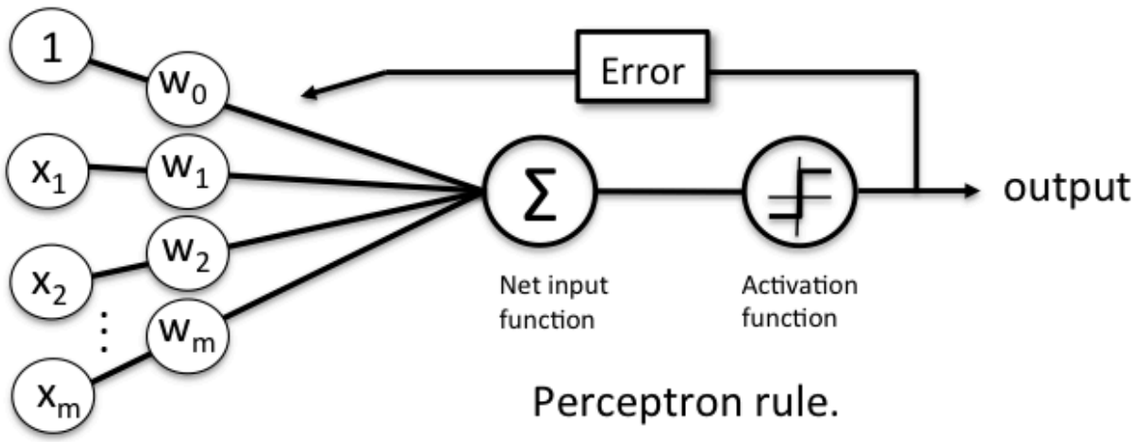
$$f_1(\mathbf{x}) = \mathbf{wx} = \sum_i w_i x_i = x_0 w_0 + \ldots + x_m w_m$$

$$f_2(\mathbf{x}) = \begin{cases} 1 & f_1(\mathbf{x}) > 0 \\ 0 & \text{otherwise} \end{cases}$$

- 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”.

The perceptron training rule



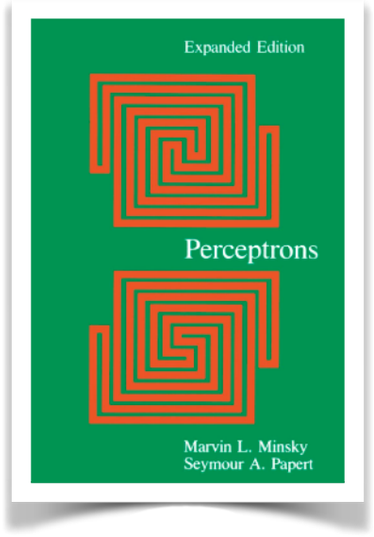
Perceptron rule.

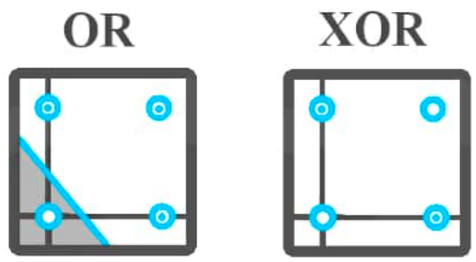
Given a dataset of samples that represents the desired behavior of the neuron, we can learn this behavior by following this algorithm:

Algorithm: Perceptron Learning Algorithm

$P \leftarrow inputs \quad with \quad label \quad 1;$
$N \leftarrow inputs \quad with \quad label \quad 0;$
Initialize $\mathbf{w}$ randomly;
while !*convergence* **do**
  Pick random $\mathbf{x} \in P \cup N$;
  if $\mathbf{x} \in P \quad and \quad \mathbf{w}.\mathbf{x} < 0$ **then**
    $\mathbf{w} = \mathbf{w} + \mathbf{x}$;
  end
  if $\mathbf{x} \in N \quad and \quad \mathbf{w}.\mathbf{x} \geq 0$ **then**
    $\mathbf{w} = \mathbf{w} - \mathbf{x}$;
  end
end
//the algorithm converges when all the inputs are classified correctly

- A critical book written in 1969 by **Marvin Minsky** and his collaborator **Seymour Papert** (“Perceptrons”) showed that Rosenblatt’s original system was painfully limited, literally blind to some simple logical functions like “exclusive-or”. What had become known as the field of “neural networks” almost disappeared.





- Obviously, ther was a blind spot in their analysis.

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.

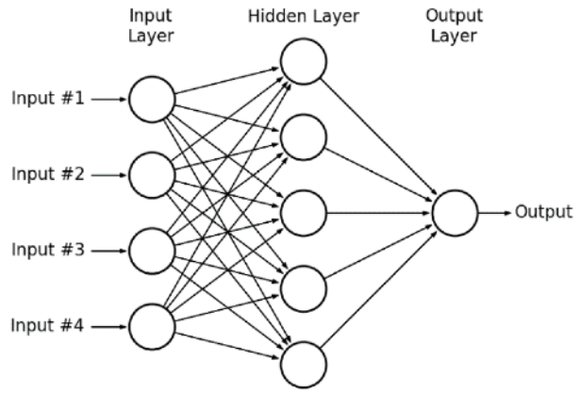
- In 1986, the problem was how to extend the Widrow-Hoff rule to multiple layers. Three independent groups of researchers, which included **David E. Rumelhart, Geoffrey E. Hinton** and **Ronald J. Williams**, came up with similar ideas which are now called **back-propagation** networks because it distributes 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.

Backpropagation generalized the perceptron training algorithm to several layers (that can solve the XOR problem), tackling the problem raised by Minsky and Papert. Nevertheless, it still had to limitations:

- While the backpropagation algorithm is guaranteed to converge on some solution in the case of a linearly separable training set, it may still pick **any solution** and problems may admit many solutions of varying quality.



- The convergence of the algorithm in the case of more than 2 layers was extremely slow.

Second neural network winter is coming

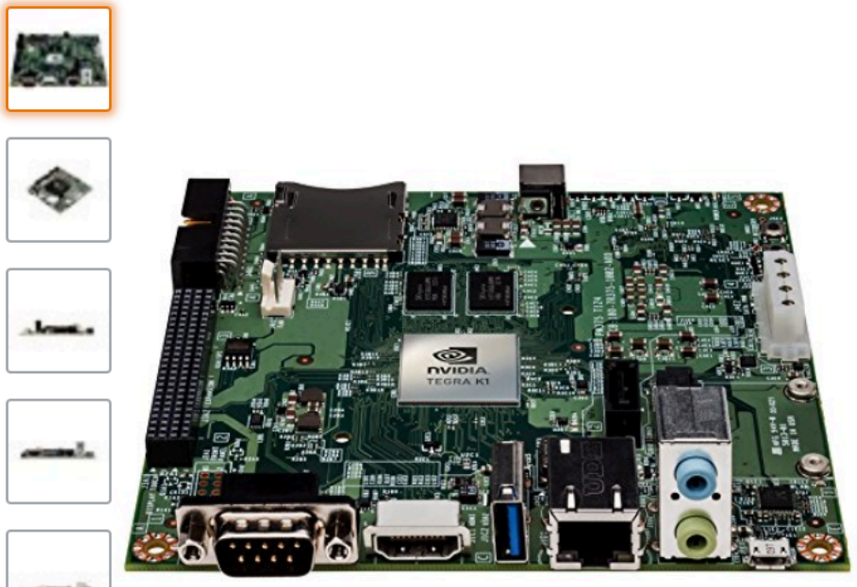
- With the ascent of **Support Vector Machines** and the failure of backpropagation, the early 2000s were a dark time for neural net research.

- 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.

2012: Neural Networks Reborn

Google Trends



Roll over image to zoom in

NVIDIA

NVIDIA Jetson TK1 Development Kit

33 customer reviews | 24 answered questions

List Price: ~~$199.99~~

Price: $170.77 + $49.57 Shipping & Import Fees Deposit to Spain Details

You Save: $29.22 (15%)

Item is eligible: No interest if paid in full within 6 months with the Amazon.com Store Card.

In Stock.

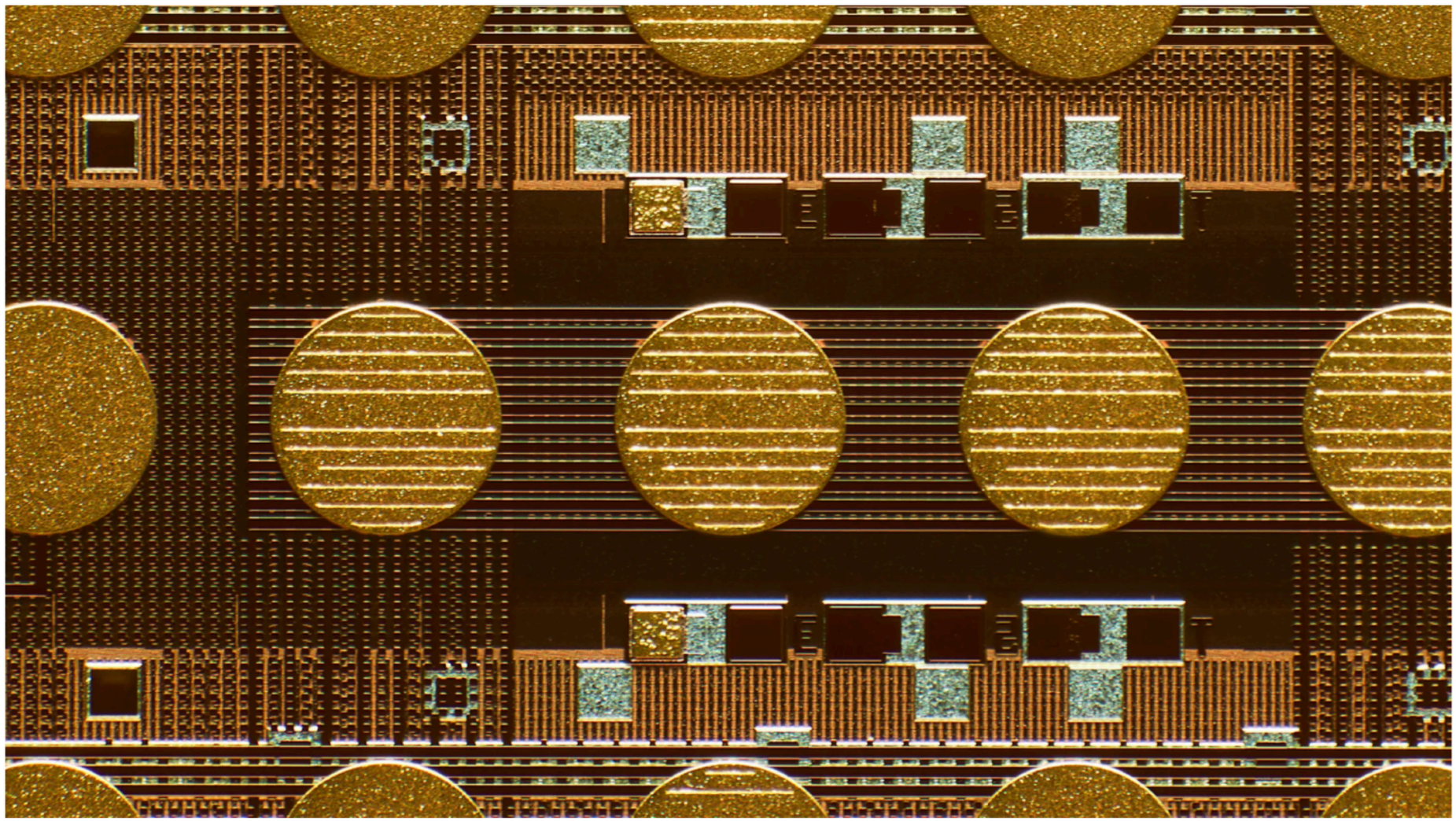
This item ships to **Barcelona, Spain**. **Want it Monday, Feb. 12?** Order within **5 hrs 25 mins** and choose **AmazonGlobal Priority Shipping** at checkout. Learn more
Ships from and sold by Amazon.com. Gift-wrap available.

- NVIDIA Kepler GPU with 192 CUDA cores
- NVIDIA 4-Plus-1 quad-core ARM Cortex-A15 CPU
- 2 GB memory, 16 GB eMMC
- Gigabit Ethernet, USB 3.0, SD/MMC, miniPCIe
- HDMI 1.4, SATA, Line out/Mic in, RS232 serial port
- Expansion ports for additional display, GPIOs, and high-bandwidth camera interface

NICOLE KOBIE LONG READS 17.06.2021 10:10 AM

NVIDIA and the battle for the future of AI chips

NVIDIA's GPUs dominate AI chips. But a raft of startups say new architecture is needed for the fast-evolving AI field



SUN LEE

Definitions

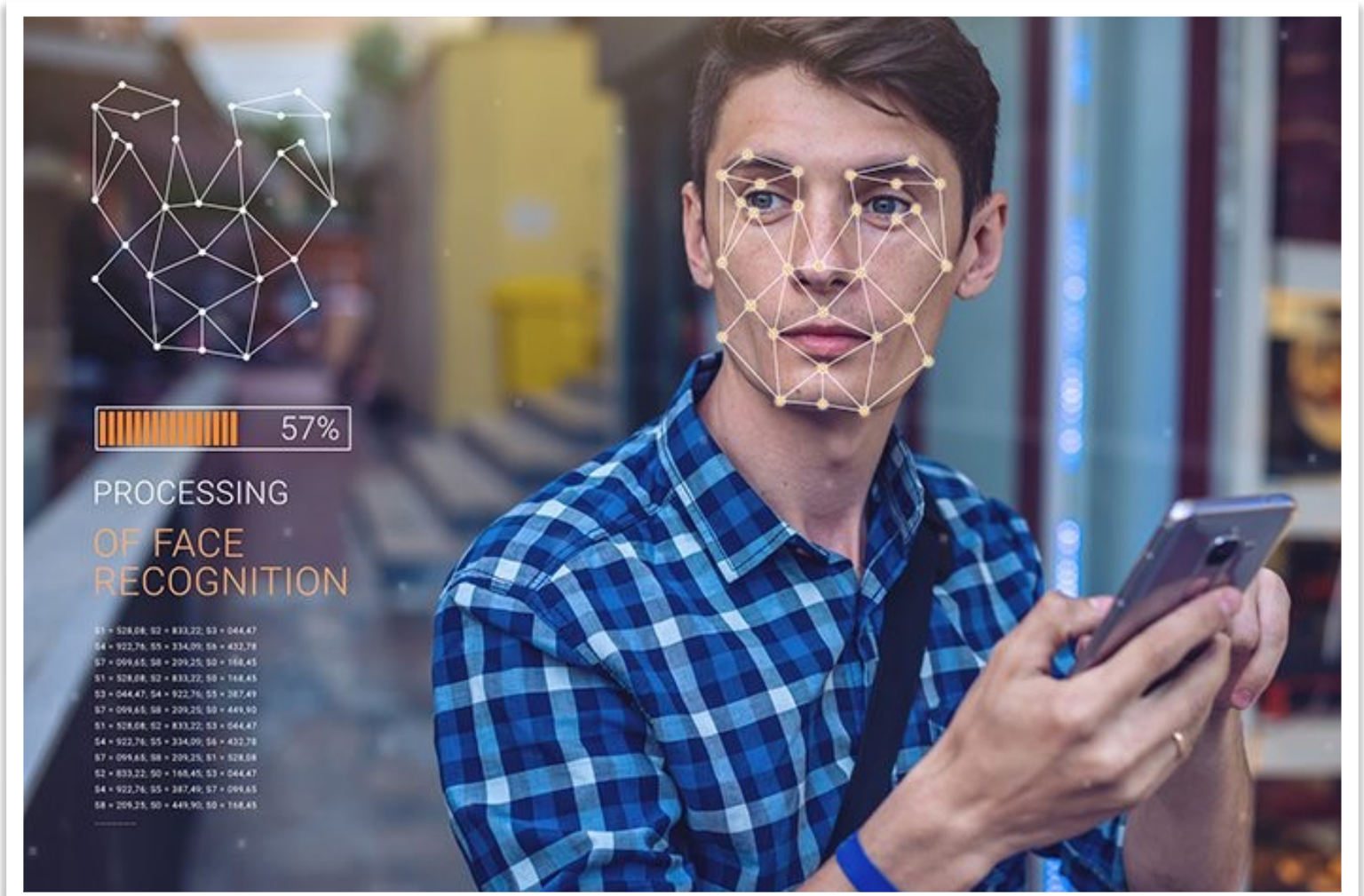
- **Neural Networks (NN)** is a beautiful biologically-inspired programming paradigm which enables a computer to learn from observational data.

- **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.



DeepFace (Facebook): Accuracy of 97.35% (2014)
FaceNet (Google): Accuracy of 99.63% (2015)
InsightFace (Open Source): Accuracy of 99.86% (2020)
Now?

New applications: navigation and mapping.

New applications: navigation and mapping.

iMAP

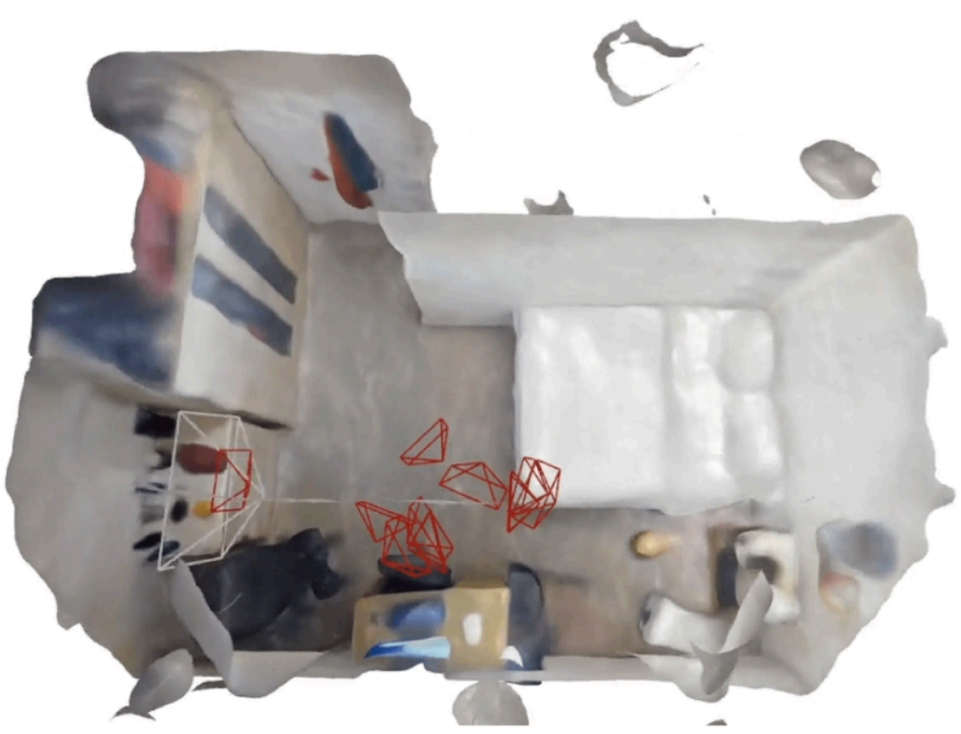
Implicit Mapping and Positioning in Real-Time

ICCV 2021

Edgar Sucar Shikun Liu Joseph Ortiz Andrew Davison

Dyson Robotics Lab, Imperial College London

Paper Video



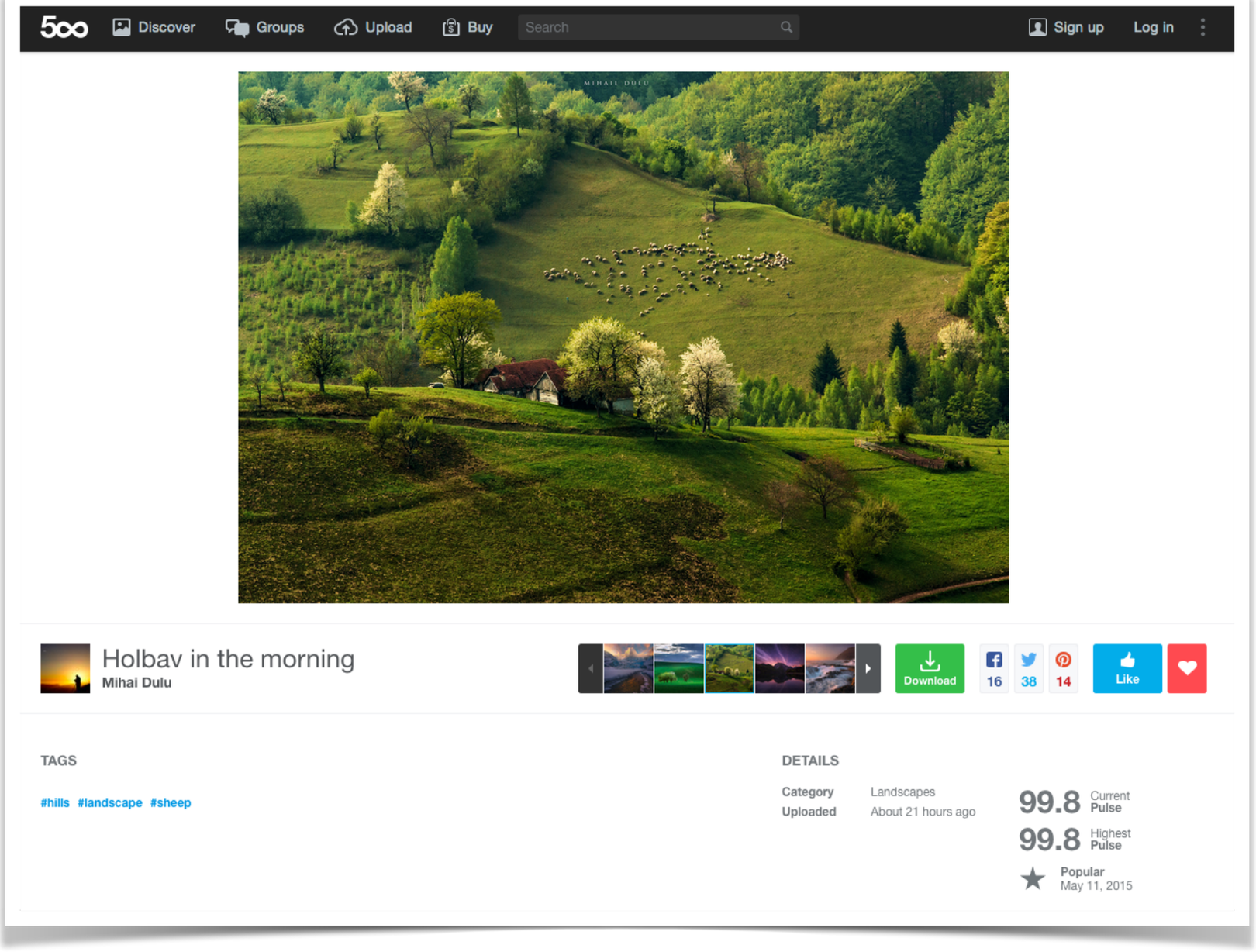
New applications: Image Upscaling (Flipboard)

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

New applications: Image Upscaling (Flipboard)

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

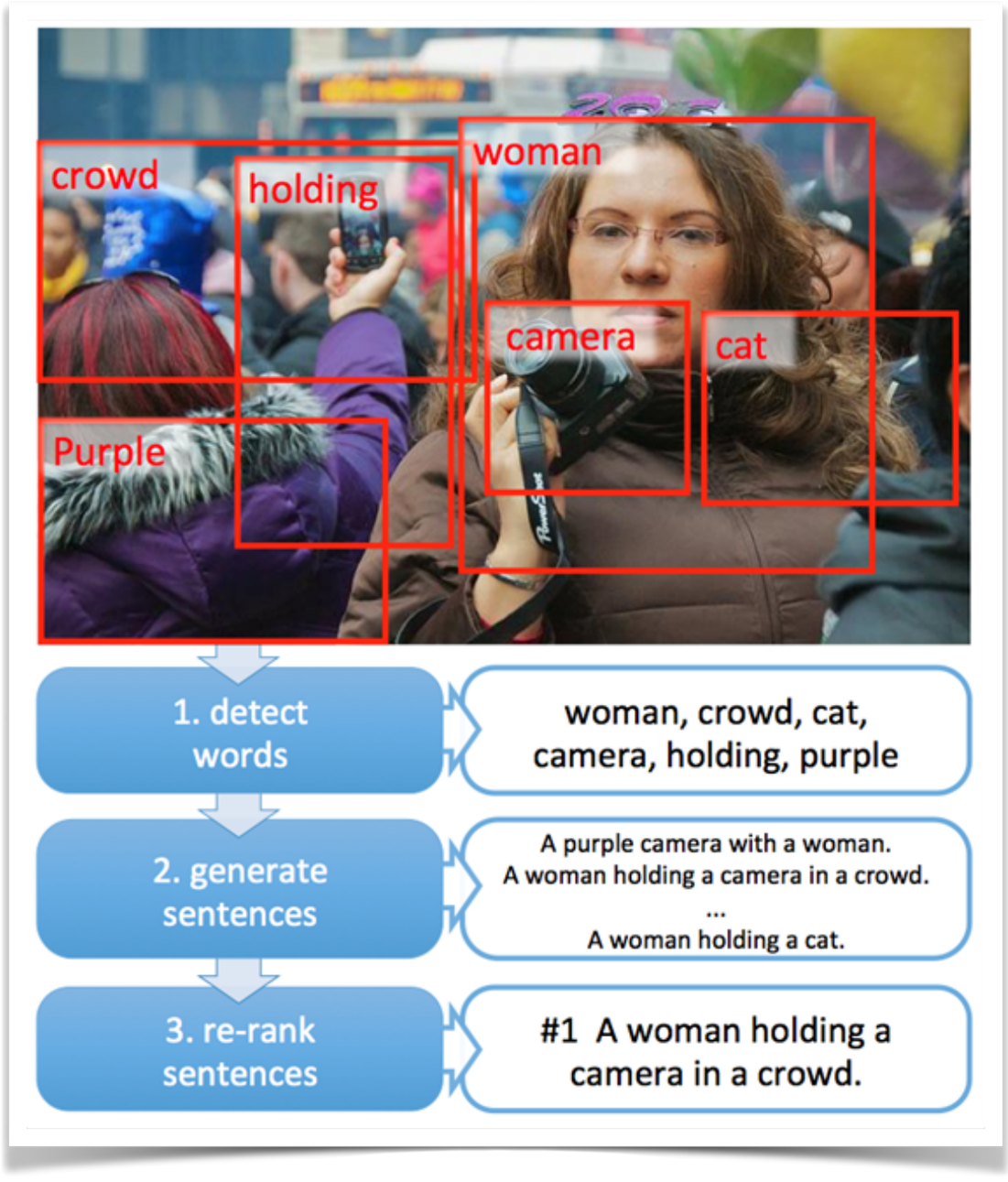
New applications: Non visual data prediction



What is Pulse?

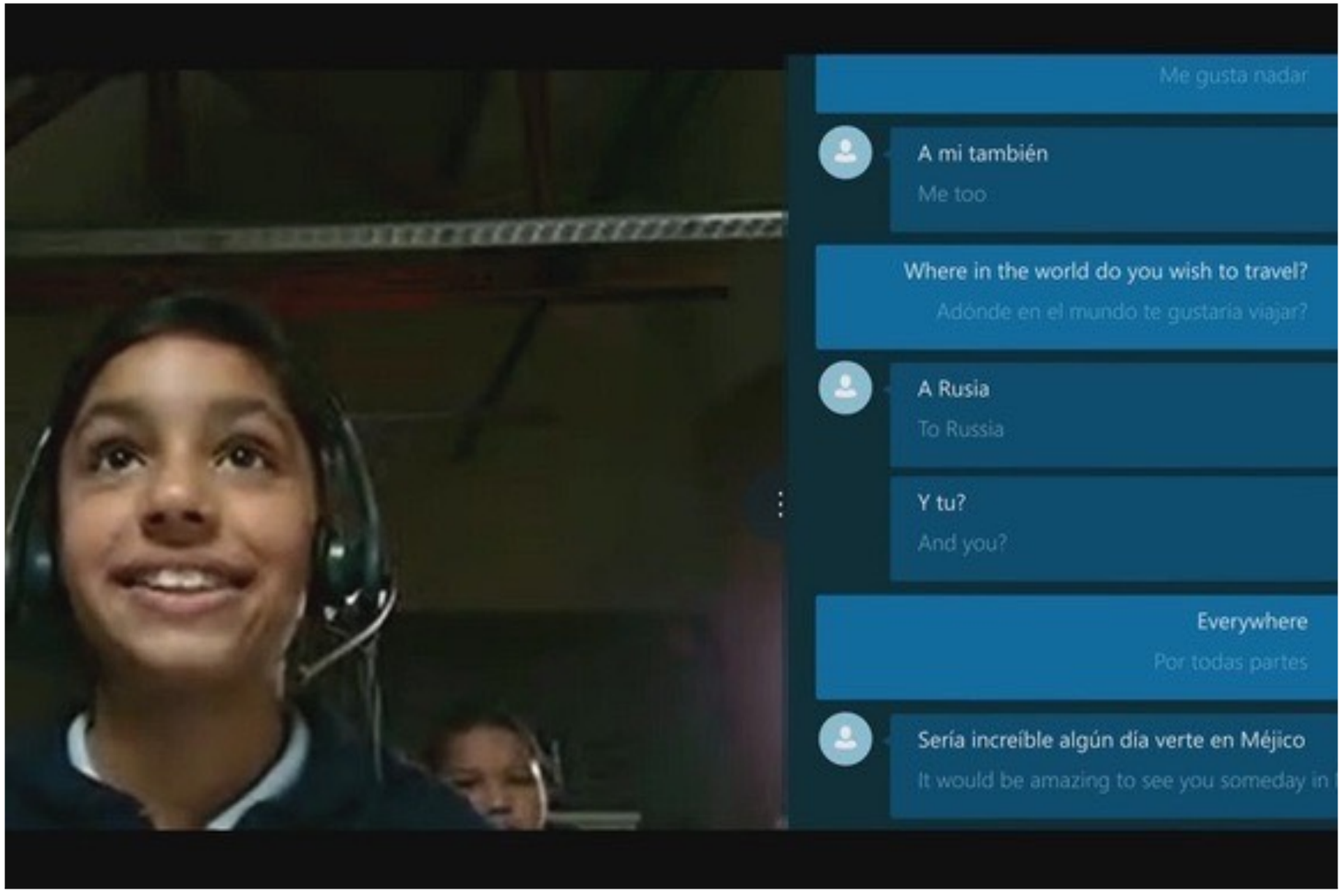
Pulse is a score out of 100 points that measures how **popular** a photo is. Pulse is calculated by an algorithm, which is unique to 500px and is based on votes (Likes & Favorites) on your photo from the community. The Pulse algorithm was designed to promote daily exposure of new photographs and photographers.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

New applications: Automatic Image Captioning

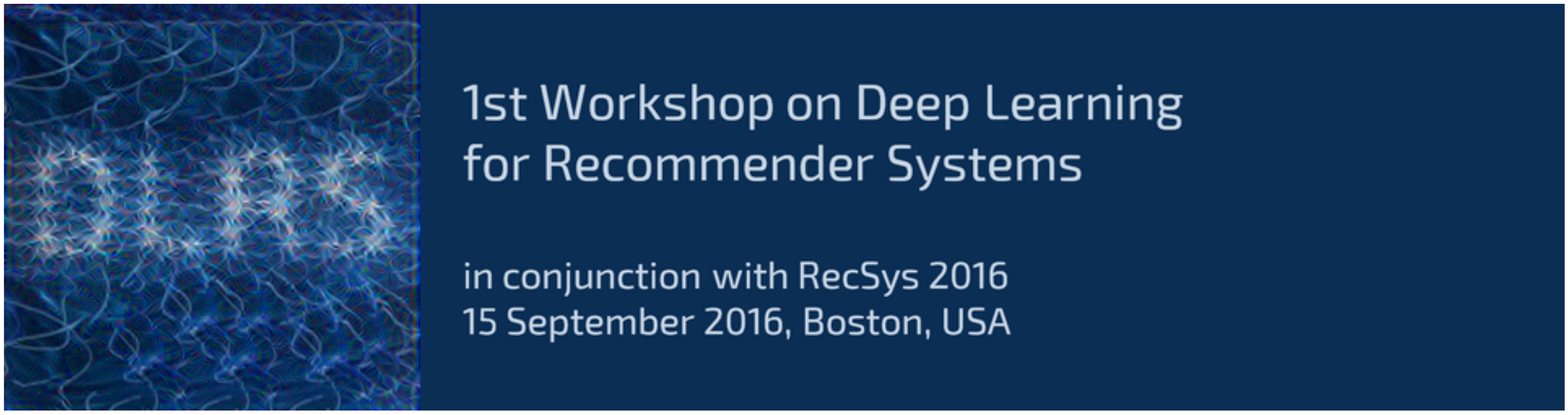


http://blogs.technet.com/b/machinelearning/archive/2014/11/18/rapid-progress-in-automatic-image-captioning.aspx

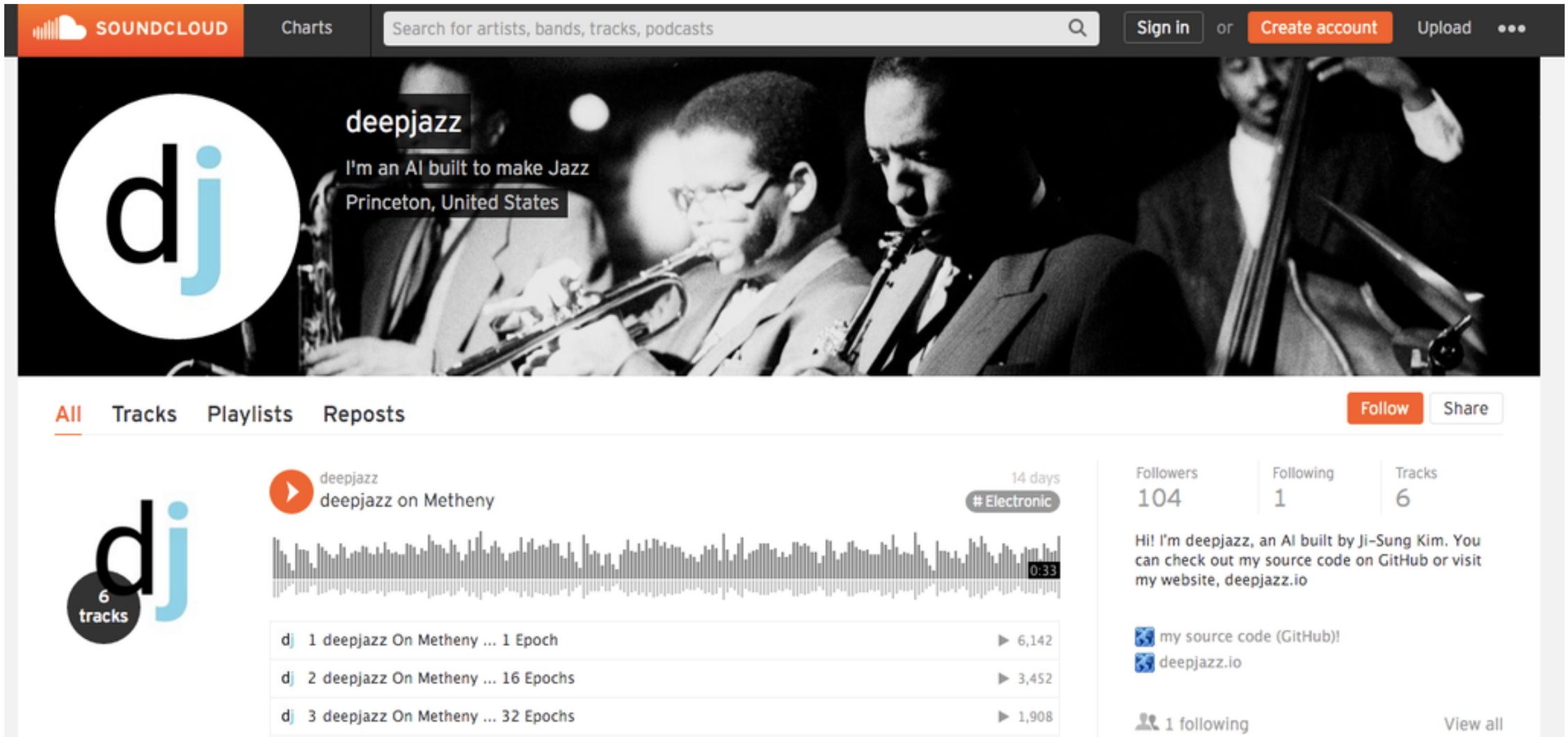
Speech translation



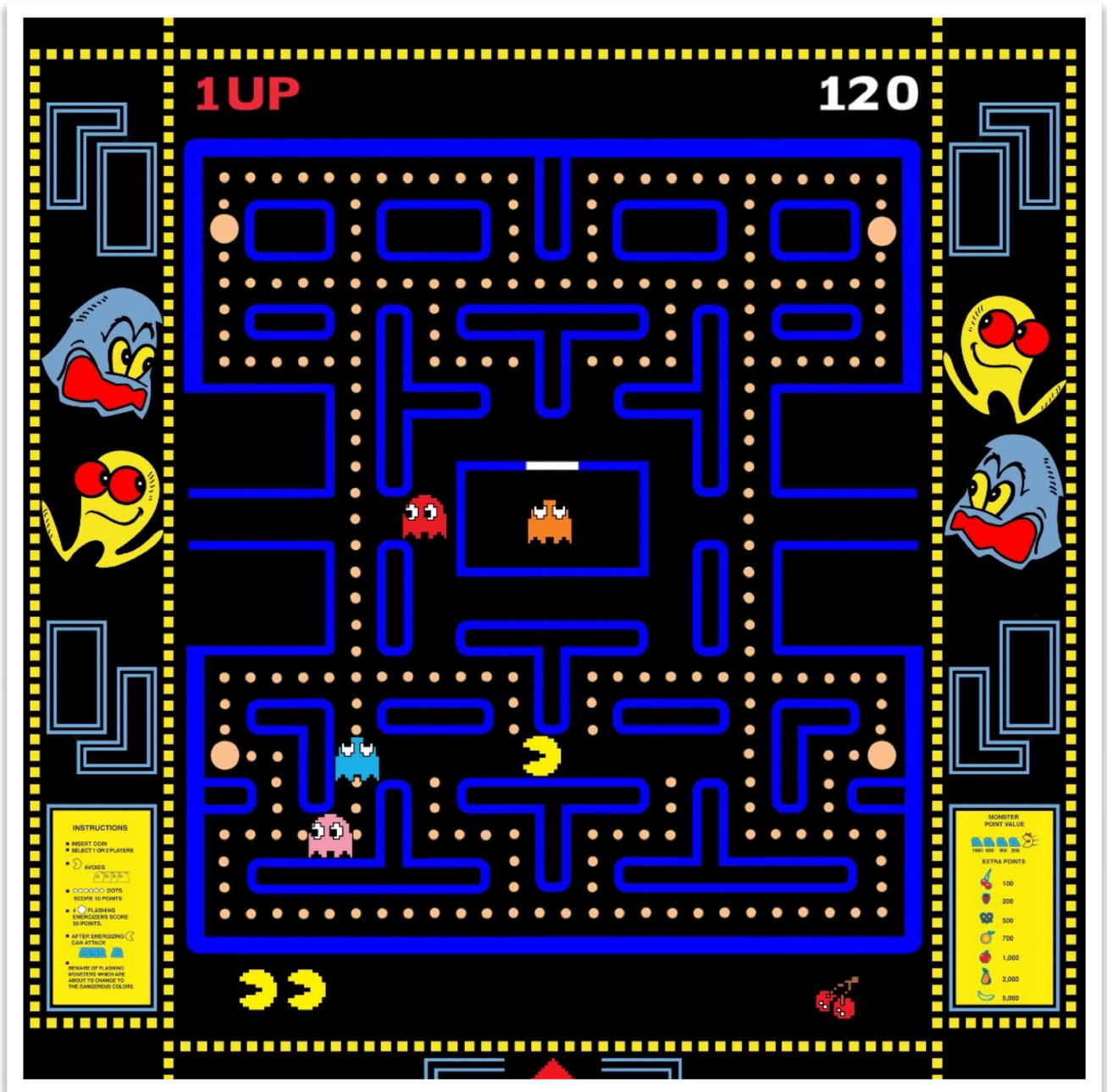
Recommenders



Music Generation



Reinforcement learning.



AlphaZero


nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO
CONSERVATION
SONGBIRDS À LA CARTE
Illegal harvest of millions of Mediterranean birds
PAGE 452
RESEARCH ETHICS
SAFEGUARD TRANSPARENCY
Don't let openness backfire on individuals
PAGE 459
POPULAR SCIENCE
WHEN GENES GOT 'SELFISH'
Dawkins's calling card forty years on
PAGE 462
NATURE.COM/NATURE
28 January 2016 £10
Vol. 529, No. 7587
9 770028 083095 04

AlphaFold

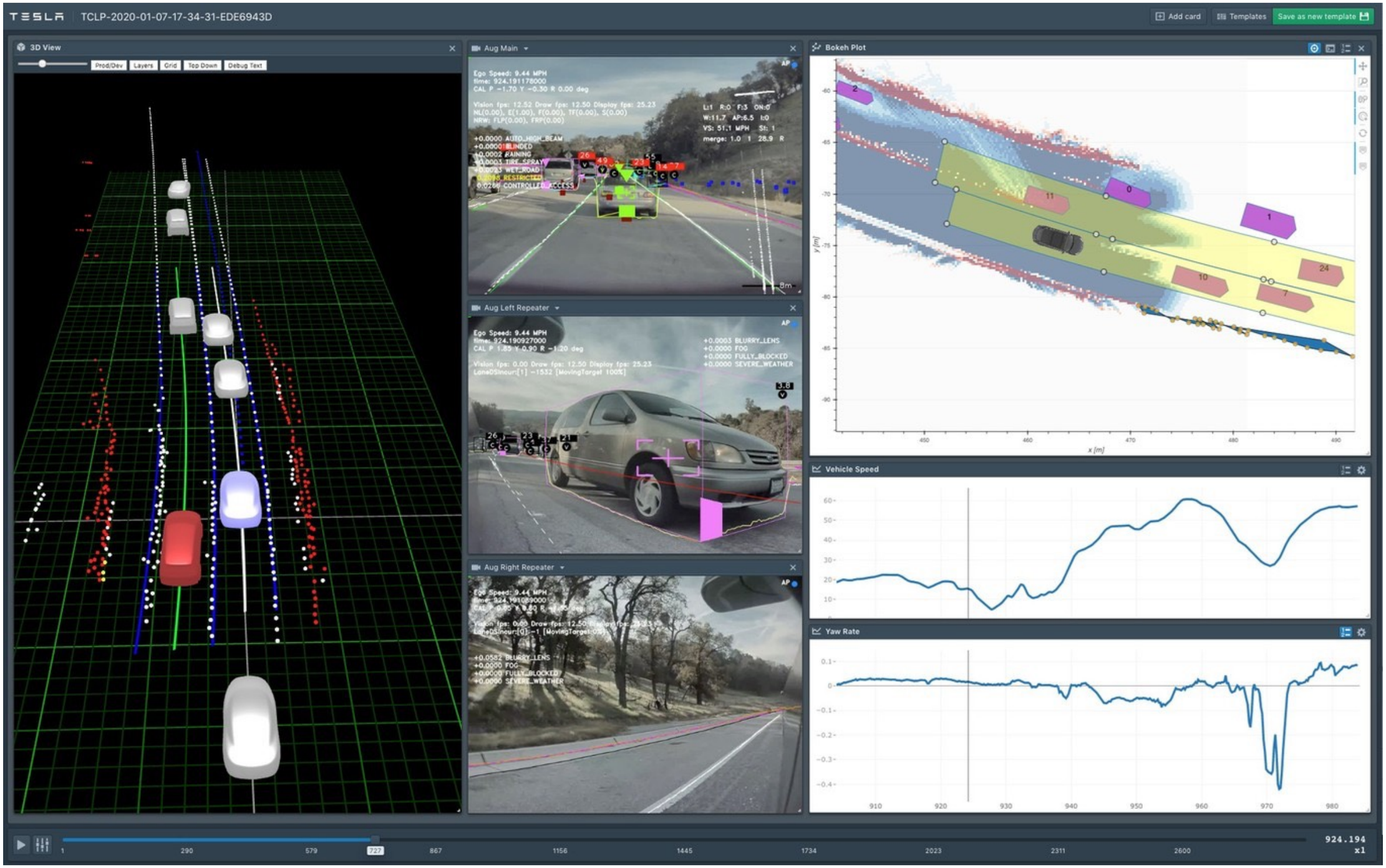
DeepMind > Blog > AlphaFold: Using AI for scientific discovery

BLOG POST
RESEARCH

15 JAN 2020

AlphaFold: Using AI for scientific discovery

TESLA Autopilot



https://www.youtube.com/watch?time_continue=31&v=NGS2SNGXUXo&feature=emb_title

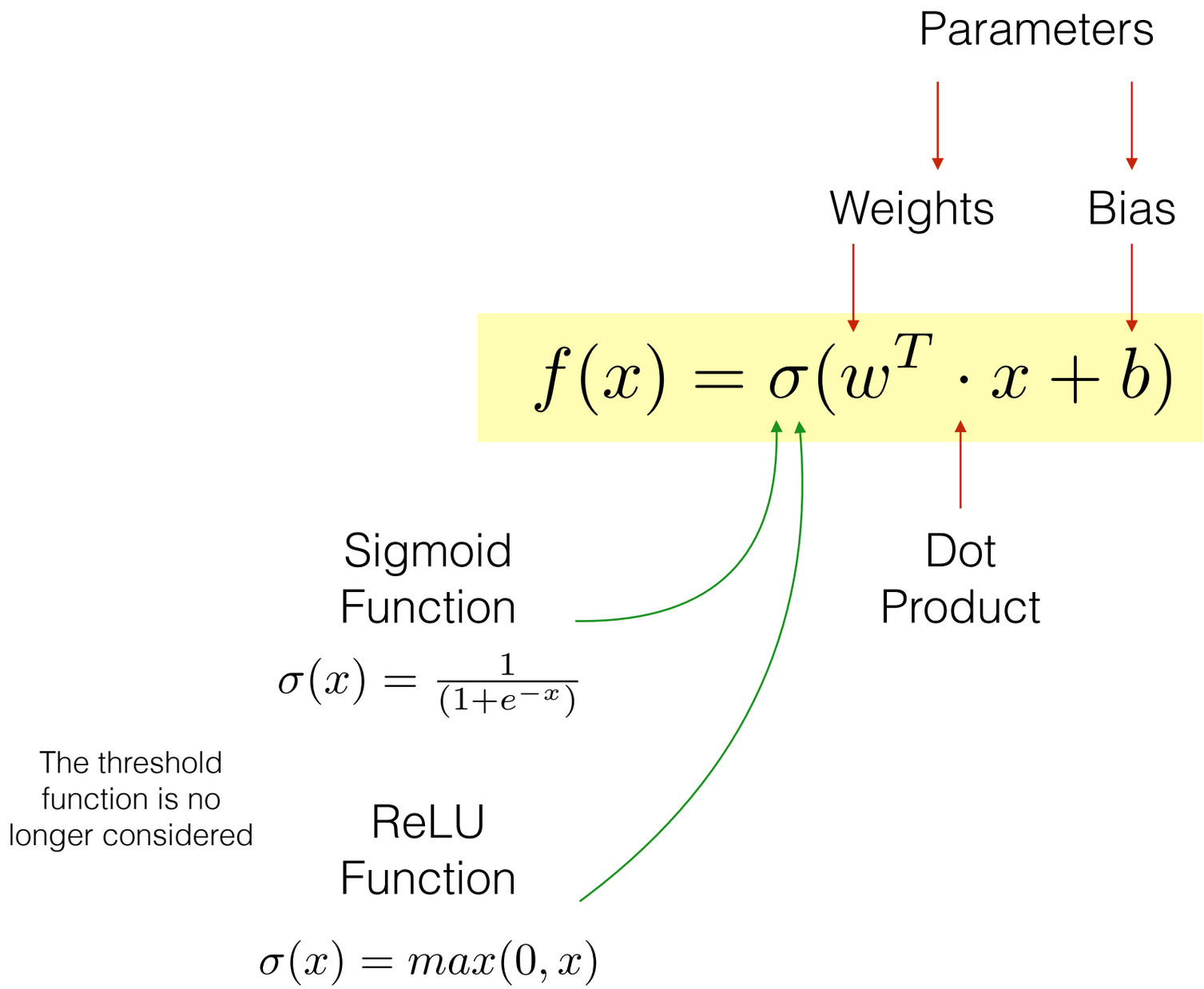
What is Deep Learning?

What is Learning from Data?

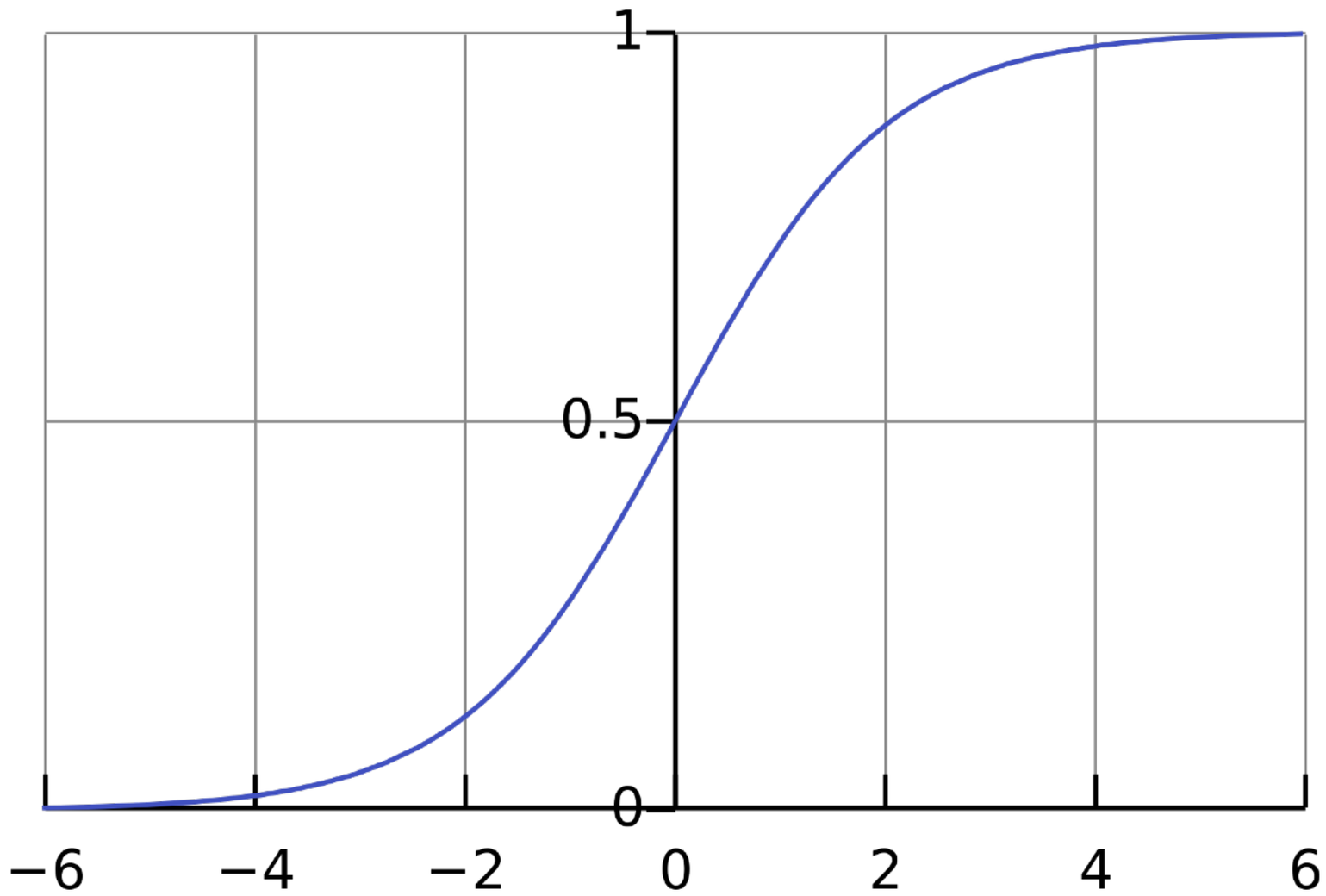
Training data: a set of $(x^{(m)}, y^{(m)})$ pairs.
Learn a function $f_w : x \to y$ to predict on new inputs x.

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

Our first model function: 1-layer "modern" neural net model

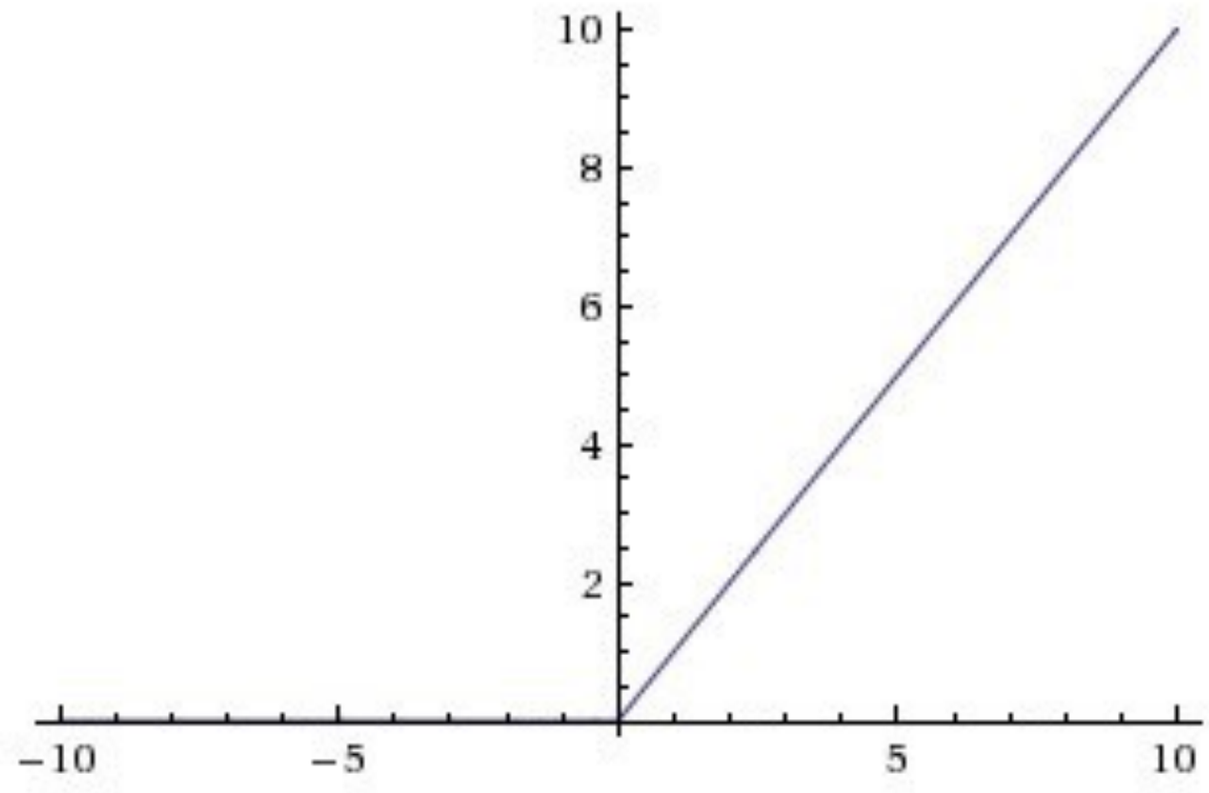


1-layer neural net model



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

1-layer neural net model



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

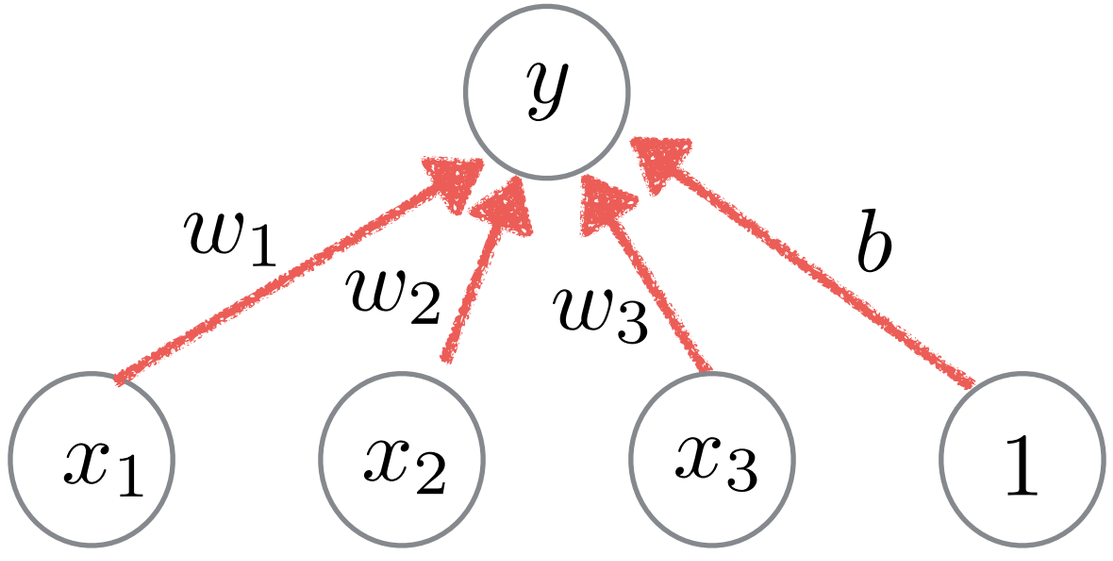
Table 3: Non-linearities tested.

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	$y = \max(W_1 x + b_1, W_2 x + b_2)$	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	$y = x$, if $x \geq 0$, else $\alpha(e^x - 1)$	2015

1-layer neural net model

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

1-dimensional NN → 2 parameters

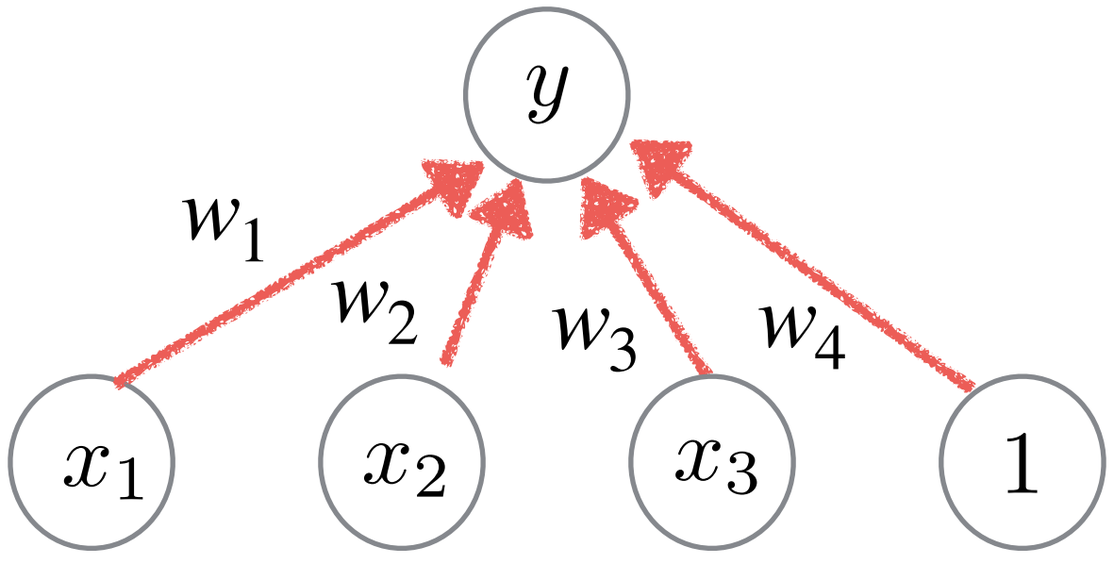


Graphical representation for 3-dimensional data → 4 parameters

1-layer neural net model **rewriting**

$$f(x) = \sigma(w^T \cdot x + b) \rightarrow f(\mathbf{x}') = \sigma(\mathbf{w}^T \cdot \mathbf{x}')$$

where $\mathbf{x}'$ is $(x_1, x_2, \ldots, x_n, 1)$



In this way we simplify notation and algorithms!

Training a network consist of updating the vector w values until our equation fits the training data as well as possible.

Such updates happen by randomly picking one of the examples and determining if it is a miss-classification. You have a misclassified example when the model determines that an example is part of the class, but it isn't, or when the model determines an example isn't part of the class, but it is.

Training usually handles one misclassified example at a time and operates by changing the w vector using a simple weighted addition:

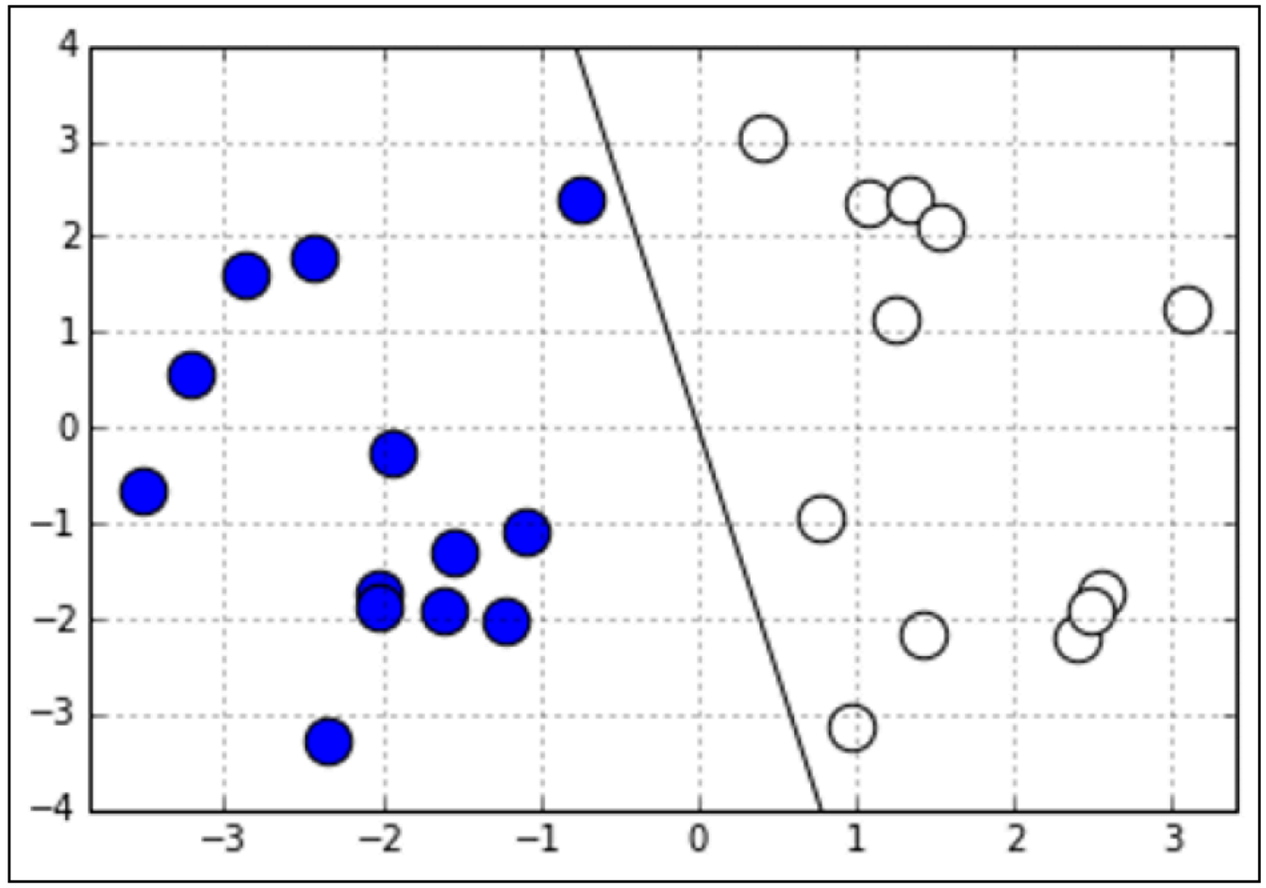
$$w_i = w_i + \alpha(x_i \cdot y_i)$$

The 2D network is nothing more than a line trying to separate the positive class from the negative one.

Initially, when w is set to zero or to random values, the separating line is just one of the infinite possible lines found on a plane

The updating phase defines it by forcing it to become nearer to the misclassified point. As the algorithm passes through the misclassified examples, it applies a series of corrections. In the end, using multiple iterations to define the errors, the algorithm places the separating line at the border area between the two classes.

1-layer neural net model



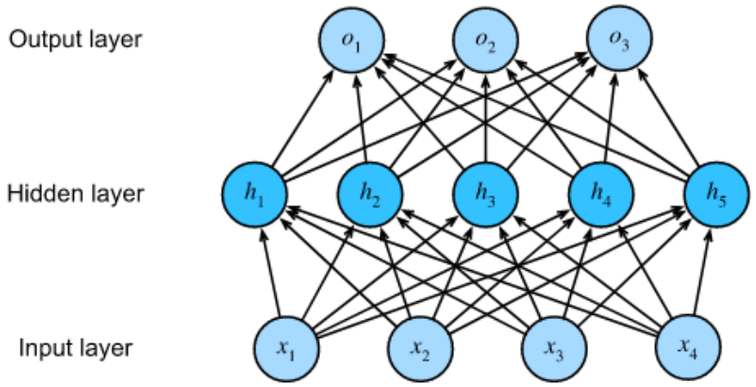
This is an example.
Is this an optimal boundary?

If you can't divide two classes spread on two or more dimensions by any line or plane, they're **nonlinearly separable**. Overcoming data's being nonlinearly separable is one of the challenges of main challenges of ML.

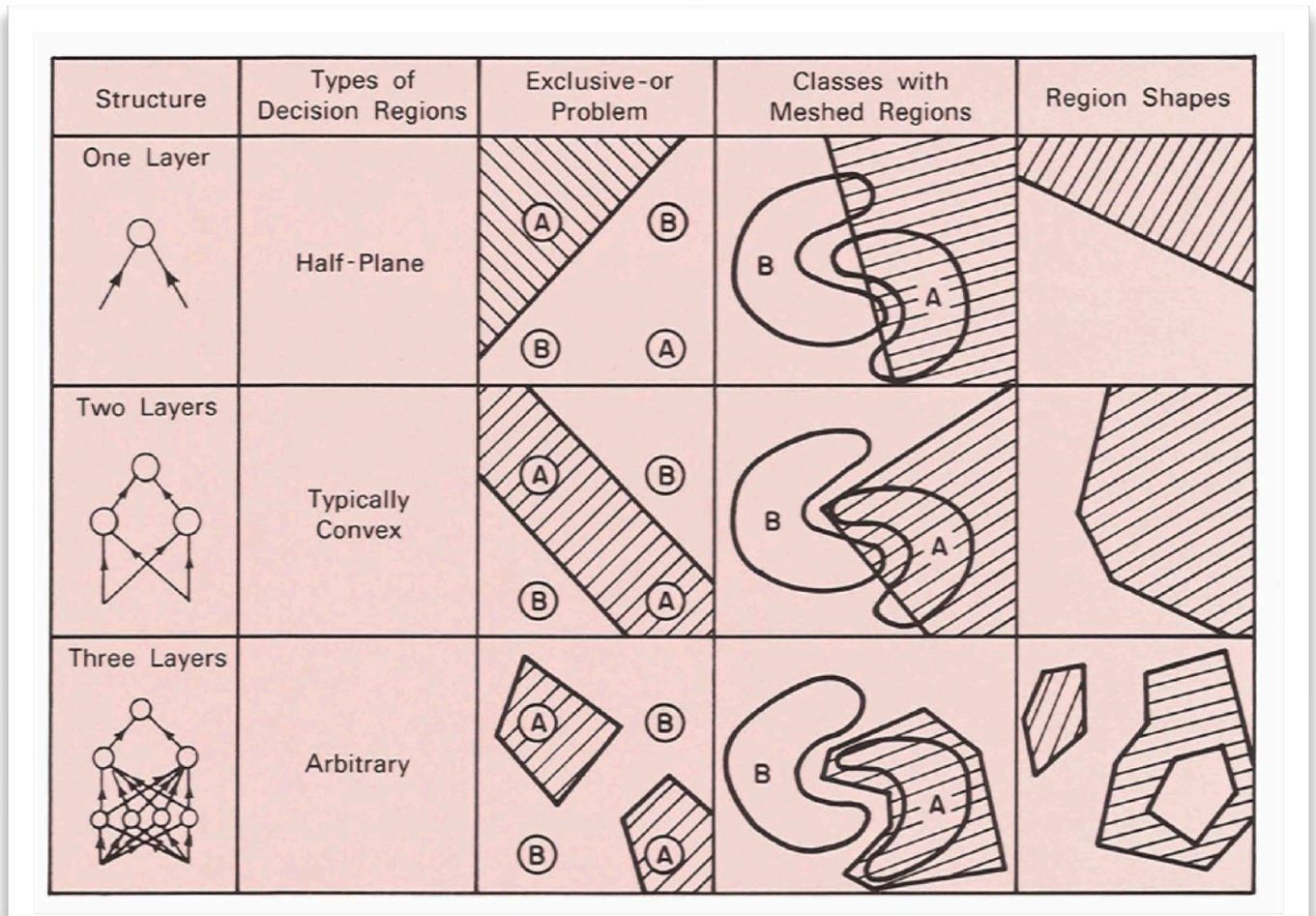
In our case, this can be solved by using **multilayer perceptrons**.

Multilayer (MLP) neural net model

Now we have more parameters!

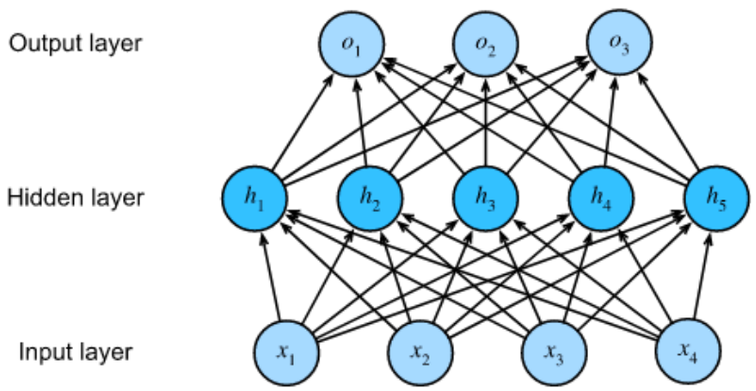


An MLP with a hidden layer of 5 hidden units.



https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.semanticscholar.org%2Fpaper%2FNeural-Network-Classifiers-for-Speech-Recognition-P..%2F7e73bffddc02223a8037e07da898cb194a64a619%2Ffigure%2F4&psig=AOvVaw04Iny0XJdW373JtlK_6TCD&ust=1601564371382000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCMiWwrGSkewCFQAAAAAdAAAAABAV

Multilayer (MLP) neural net model



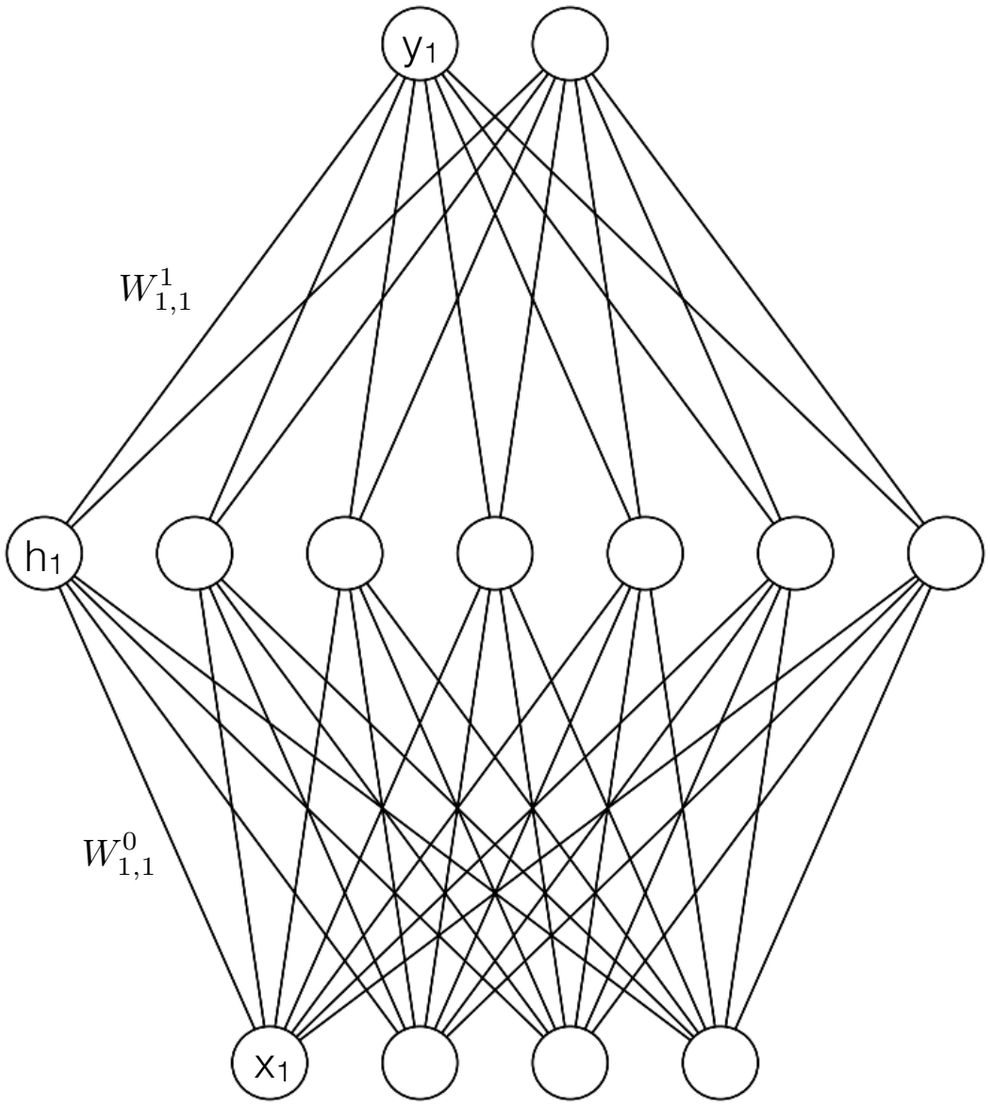
An MLP with a hidden layer of 5 hidden units.

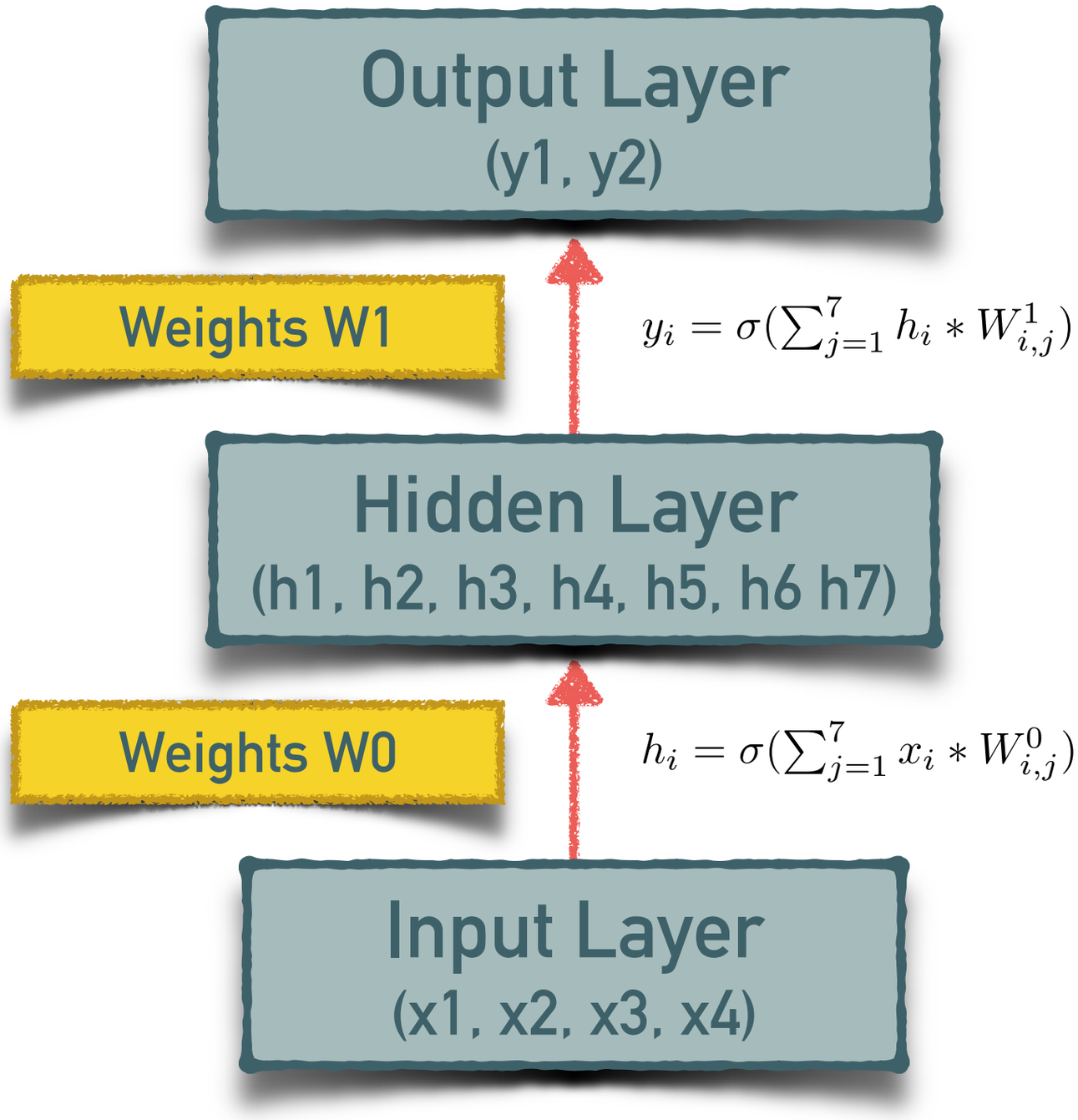
It can be shown (**Universal Approximation Theorem**) that a single-hidden-layer network, given enough nodes (possibly absurdly many), and the right set of weights, we can model any function.

Moreover, just because a single-hidden-layer network *can* learn any function does not mean that you should try to solve all of your problems with single-hidden-layer networks. In fact, we can approximate many functions much more compactly by using deeper (vs. wider) networks.

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.semanticscholar.org%2Fpaper%2FNeural-Network-Classifiers-for-Speech-Recognition-P..%2F7e73bffddc02223a8037e07da898cb194a64a619%2Ffigure%2F4&psig=AOvVaw04Iny0XJdW373JtlK_6TCD&ust=1601564371382000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCMiWwrGSkewCFQAAAAAdAAAAABAV

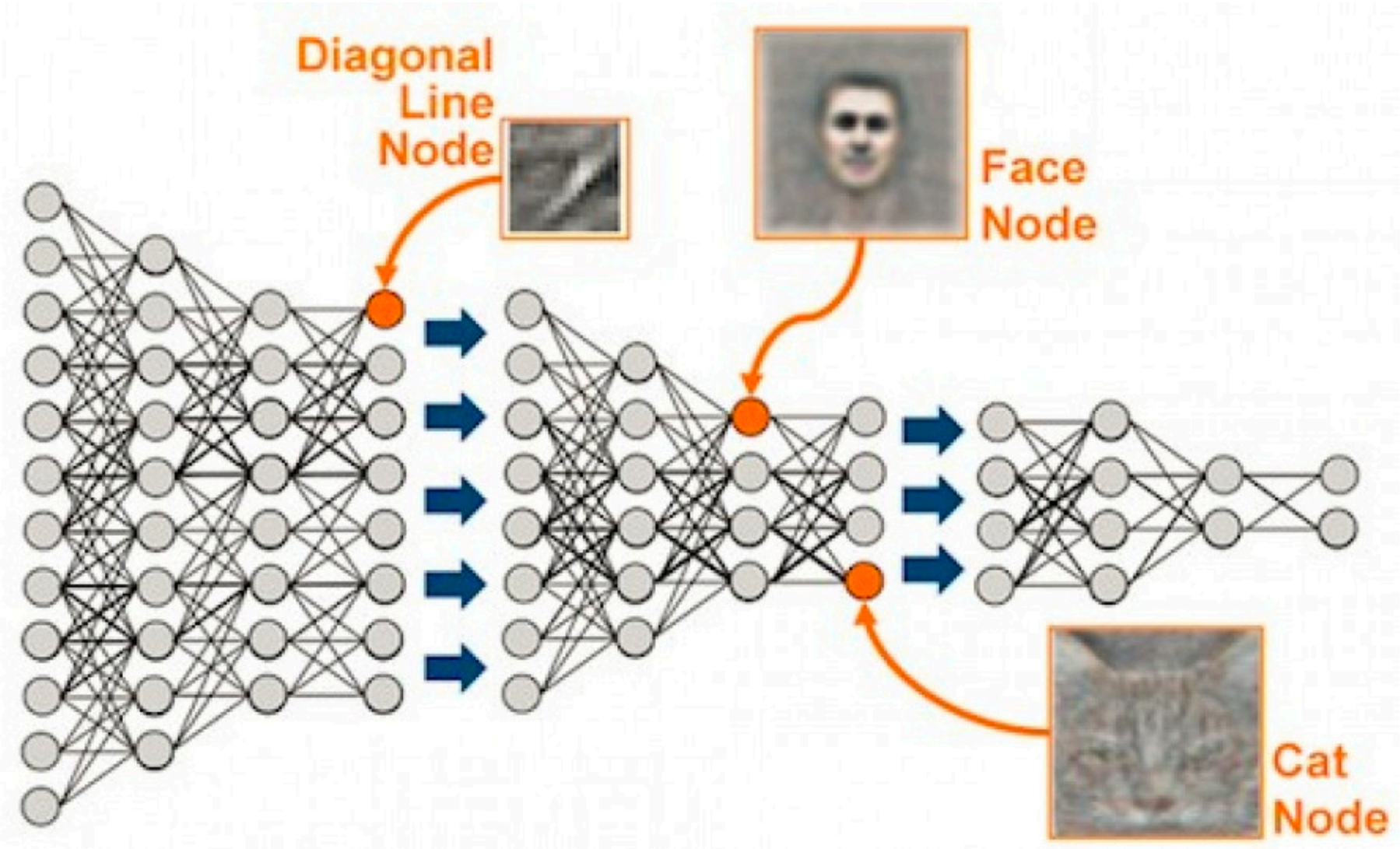
2-layer neural net model





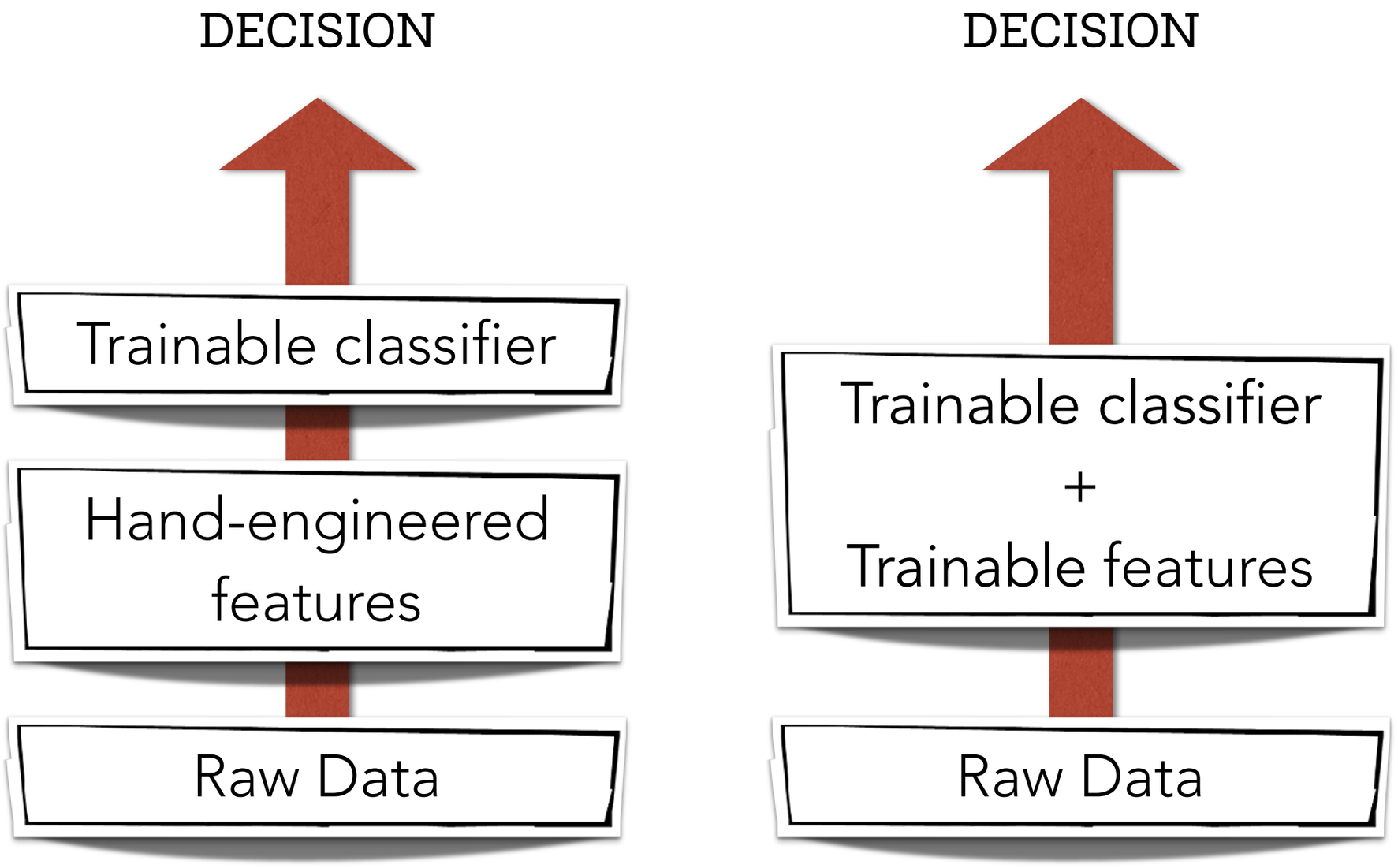
Computing the activation of one layer from the previous one can be written as a matrix-vector multiplication!

Deeper is better!



http://sdat.ir/en/sdat-blog/item/1-deep-learning

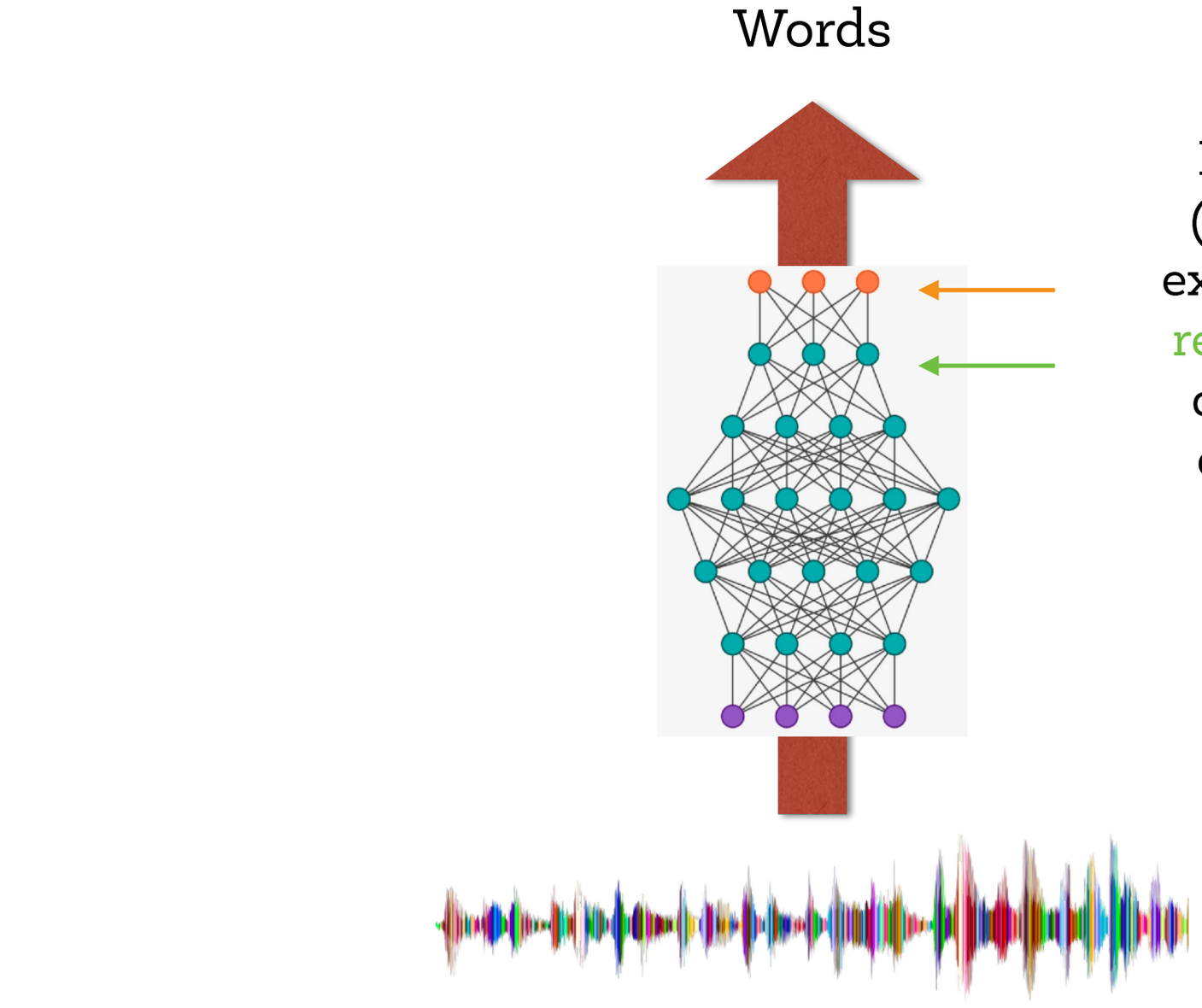
What is the main DL advantage?



STANDARD MACHINE LEARNING

DEEP LEARNING

What is the main DL advantage?



DL can discover (during training) extremely complex representations of data than can be classified with a linear model.