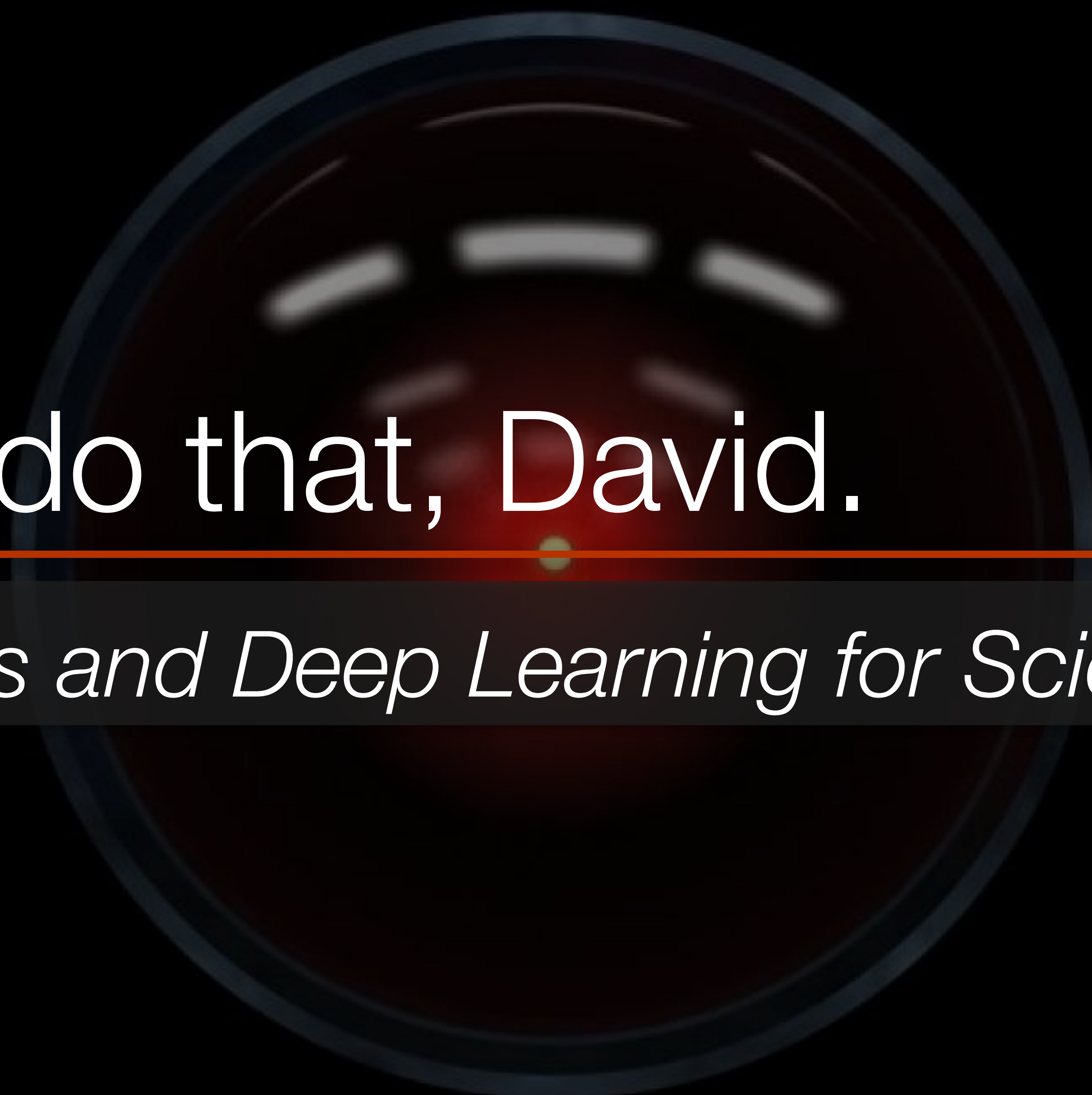


Yes, I can do that, David.

Neural Networks and Deep Learning for Science

Brian Nord (*@iamstarnord*)
9 November 2018

LSST DSFP @ Northwestern

What my parents think I do

What society thinks I do

What scientists think I do

What AI experts think I do

What my parents think I do

What society thinks I do

What scientists think I do

What AI experts think I do

What it feels like I do

Neural Networks and what they can do for you

- Discussion 1:
  Overview of Neural Networks and Deep Learning

- Activity 1:
  Coding a neural network

- Discussion 2:
  Deep learning on physical data sets

- Activity 2:
  Deep Learning Applications to Astronomy Data

- Discussion 3:
  Implications for AI and Society (Saturday)

Goals:

- You are prepared to design applications of deep learning to physics problems.
- You are prepared to consider the wide-reaching applicability and power of deep learning, including ethics.

Goals:

- You are prepared to design applications of deep learning to physics problems.

- You are prepared to consider the wide-reaching applicability and power of deep learning, including ethics.

Let’s make science’s next top model.

Let’s look around.

Discussion 1

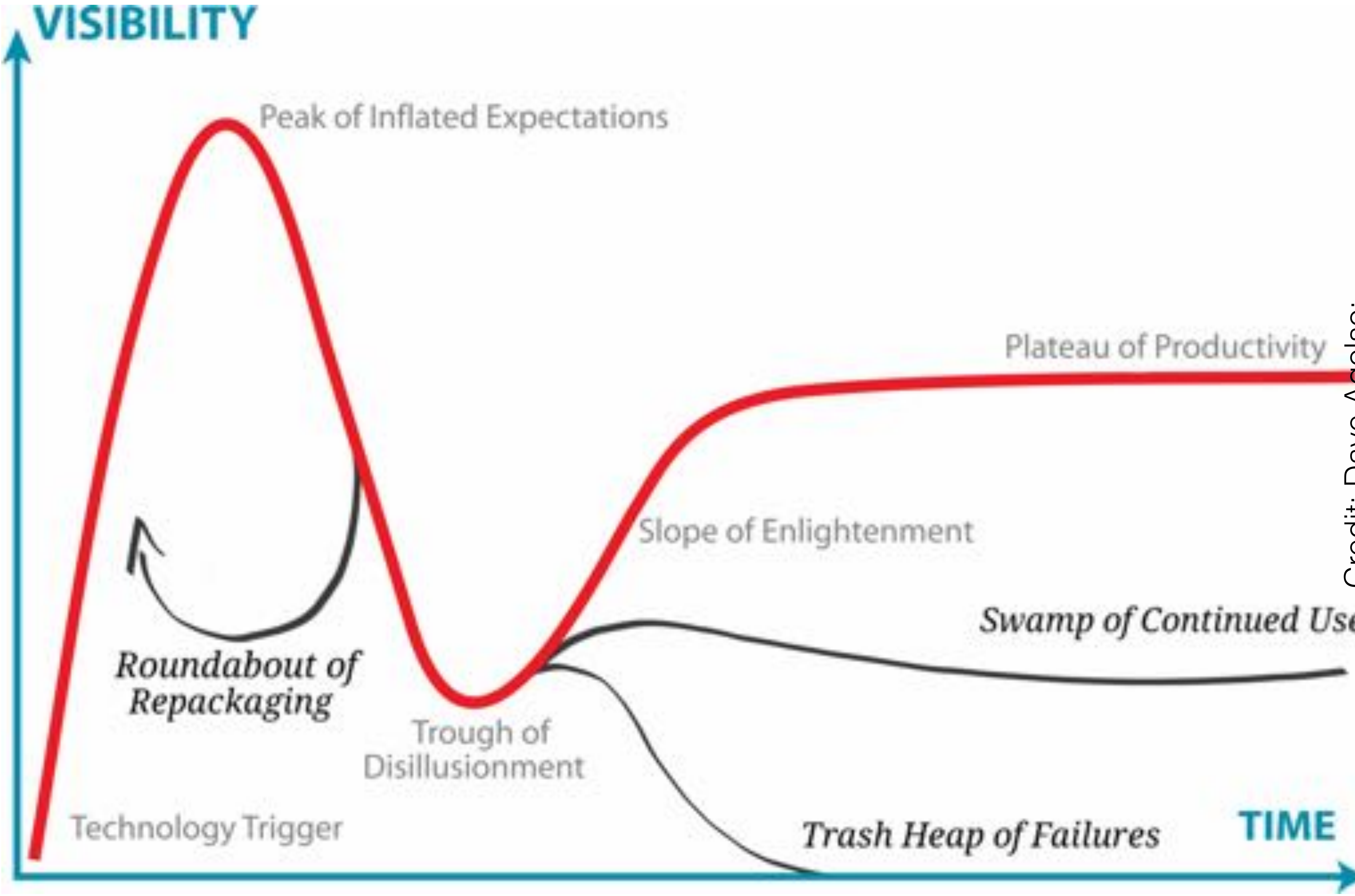
Overview of Neural Networks

Preview

- AI is Everywhere-ish!
- Scope and History of Machine Intelligence
- How does Machine Intelligence work?
  - Anatomy of a neural network

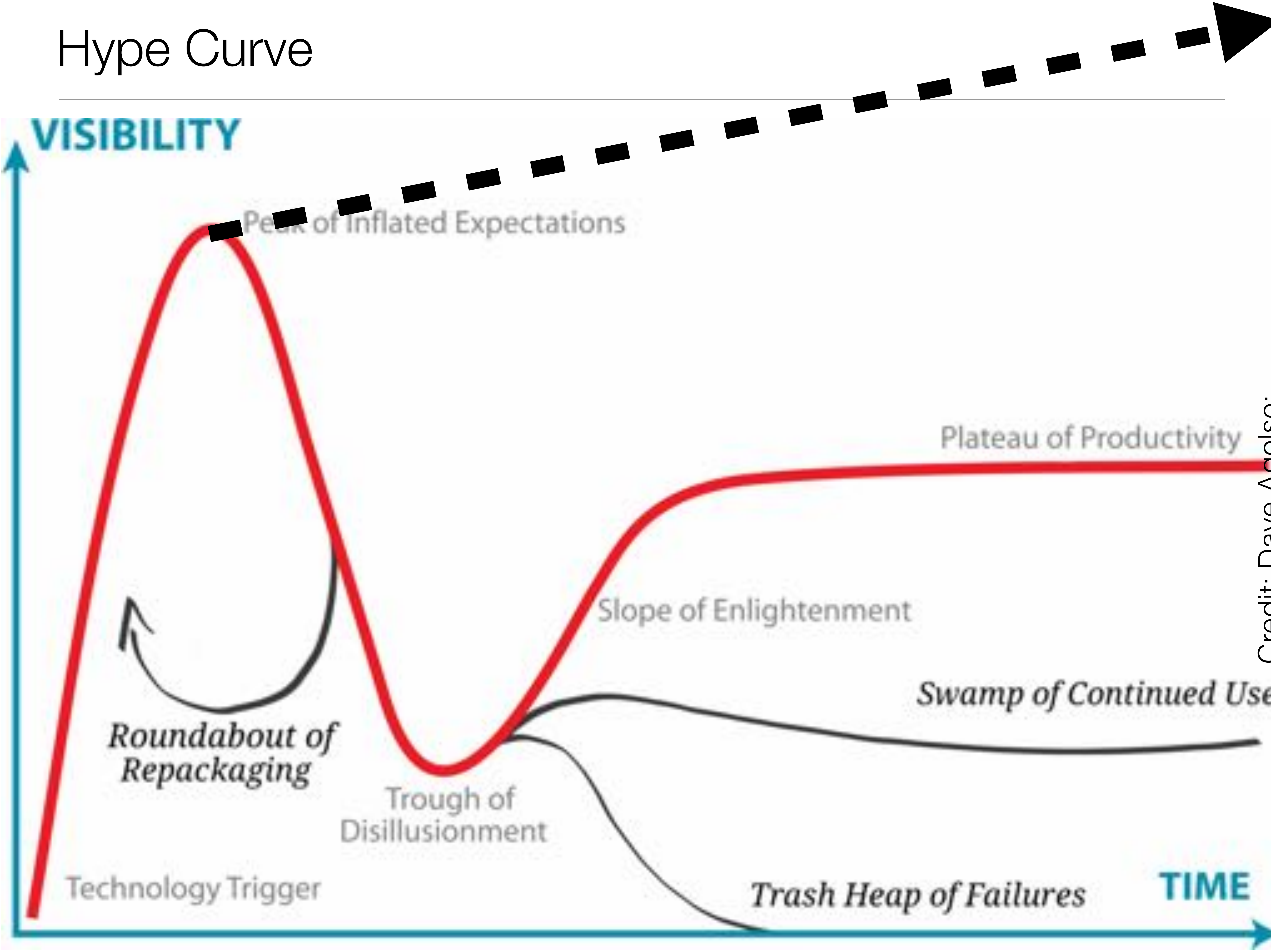
Rise of the Machines:
AI is Everywhere

Hype Curve



Credit: Dave Agolso:

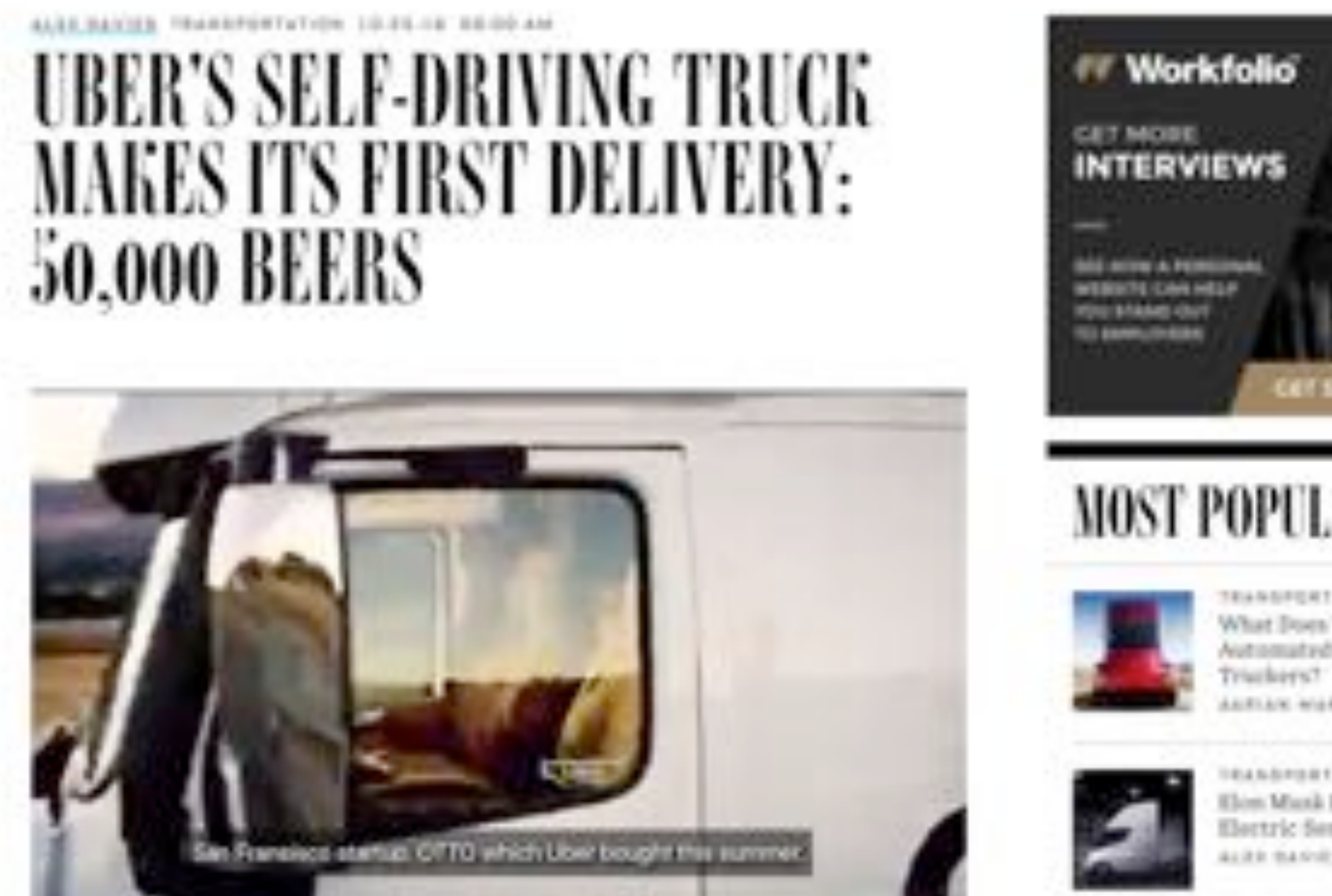
Hype Curve



Credit: Dave Agolso:

What do you think are the biggest applications for deep learning — now or in future?

Rise of the machines: Transportation and Commerce



Rise of the machines: Language Translation

The Download

What's up in emerging technology

Artificial Intelligence Can Translate Languages Without a Dictionary

Parlez-vous artificial intelligence? Two new research papers detail unsupervised machine-learning methods that can do language translation without dictionaries, as reported in *Science*. The methods also work

Rise of the machines: Entertainment

Rise of the machines: Food

TechXplore

Topics

Week's top Latest news Unread news

Deep-learning algorithm recommends ingredients and recipes based on a photo of food

July 21, 2017 by Adam Conner-Simons

Pic2Recipe, an artificial intelligence system developed at MIT, can take a photo of an entree and suggest a similar recipe to it.

Rise of the machines: Medicine

JANUARY 25, 2017

Deep learning algorithm does as well as dermatologists in identifying skin cancer

In hopes of creating better access to medical care, Stanford researchers have trained an algorithm to diagnose skin cancer.

Rise of the machines: Biochemistry

HOME NEWS MULTIMEDIA MEETINGS PORTALS ABOUT

PUBLIC RELEASE: 27-JUL-2017

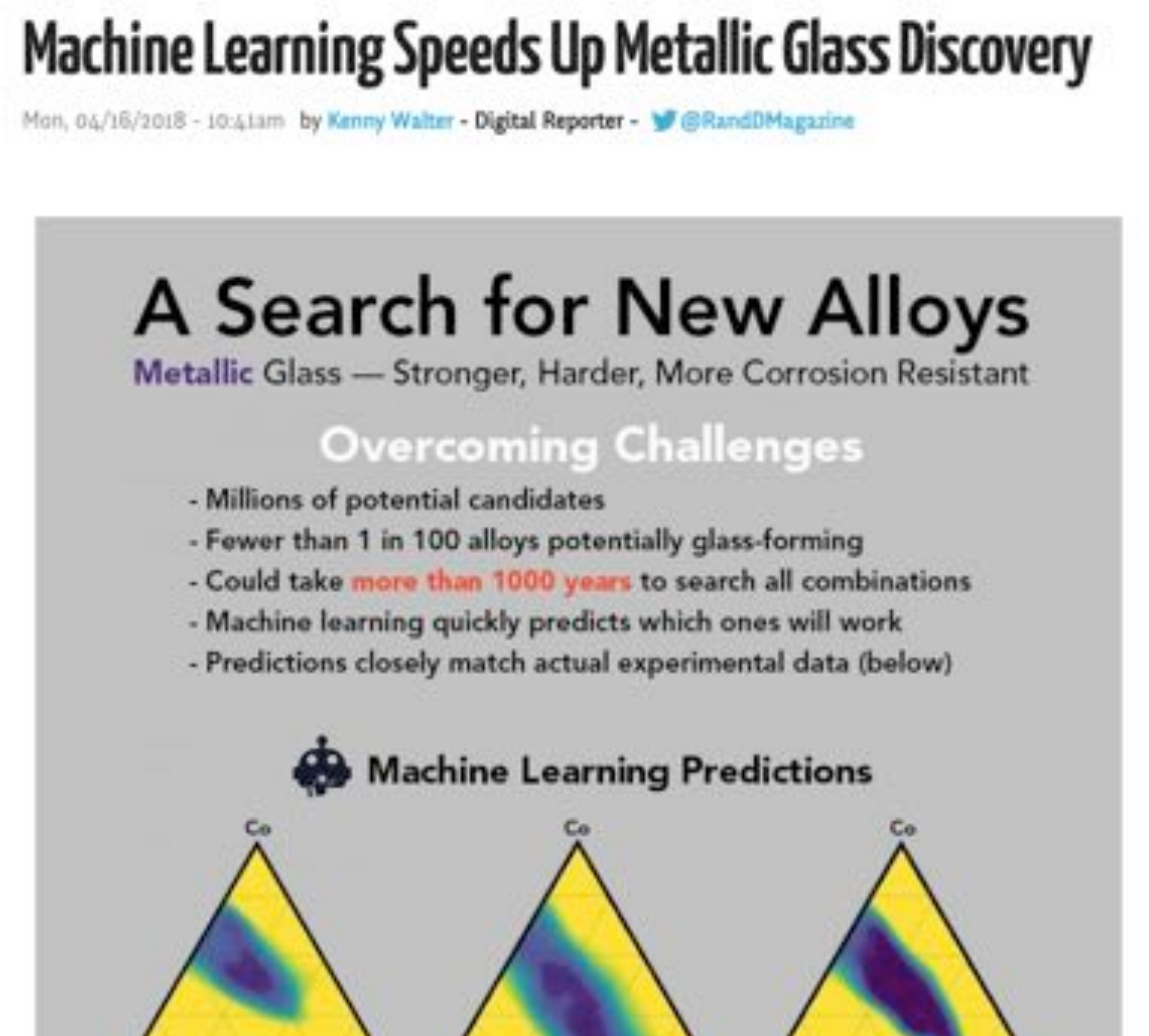
First molecules discovered by next-generation artificial intelligence to be developed into drugs

INSILICO MEDICINE, INC.

SHARE PRINT E-MAIL

Thursday, July 27, 2017, Baltimore, Md., Insilico Medicine ("Insilico"), a Baltimore-based leader in artificial intelligence ("AI") for drug discovery and biomarker development, is pleased to announce a multi-year drug development agreement

Rise of the machines: Materials



Machine Learning Speeds Up Metallic Glass Discovery

Mon, 04/16/2018 - 10:41am by Kenny Walter - Digital Reporter - @RandDMagazine

Rise of the machines: Games and Puzzles

THE VERGE TECH SCIENCE CULTURE CARS REVIEWS LONGFORM VIDEO MORE

GOOGLE \ SCIENCE \ TECH

DeepMind's Go-playing AI doesn't need human help to beat us anymore

9

The company's latest AlphaGo AI learned superhuman skills by playing itself over and over

By James Vincent | @jjvincent | Oct 18, 2017, 1:00pm EDT

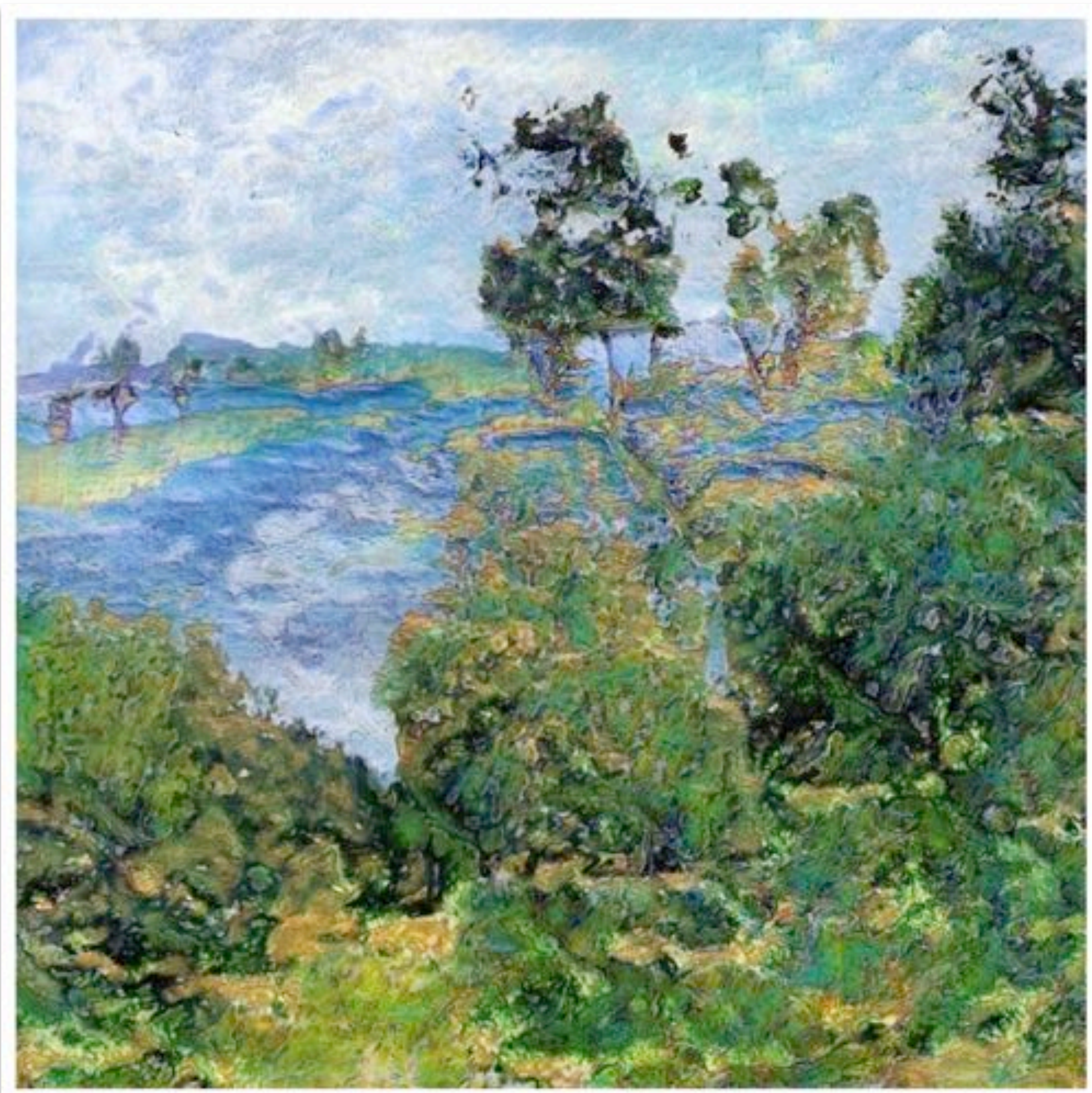
SHARE TWEET LINKEDIN

NOW TRENDING

I tried a sleep mask that bathes your eyes in light to wake you up

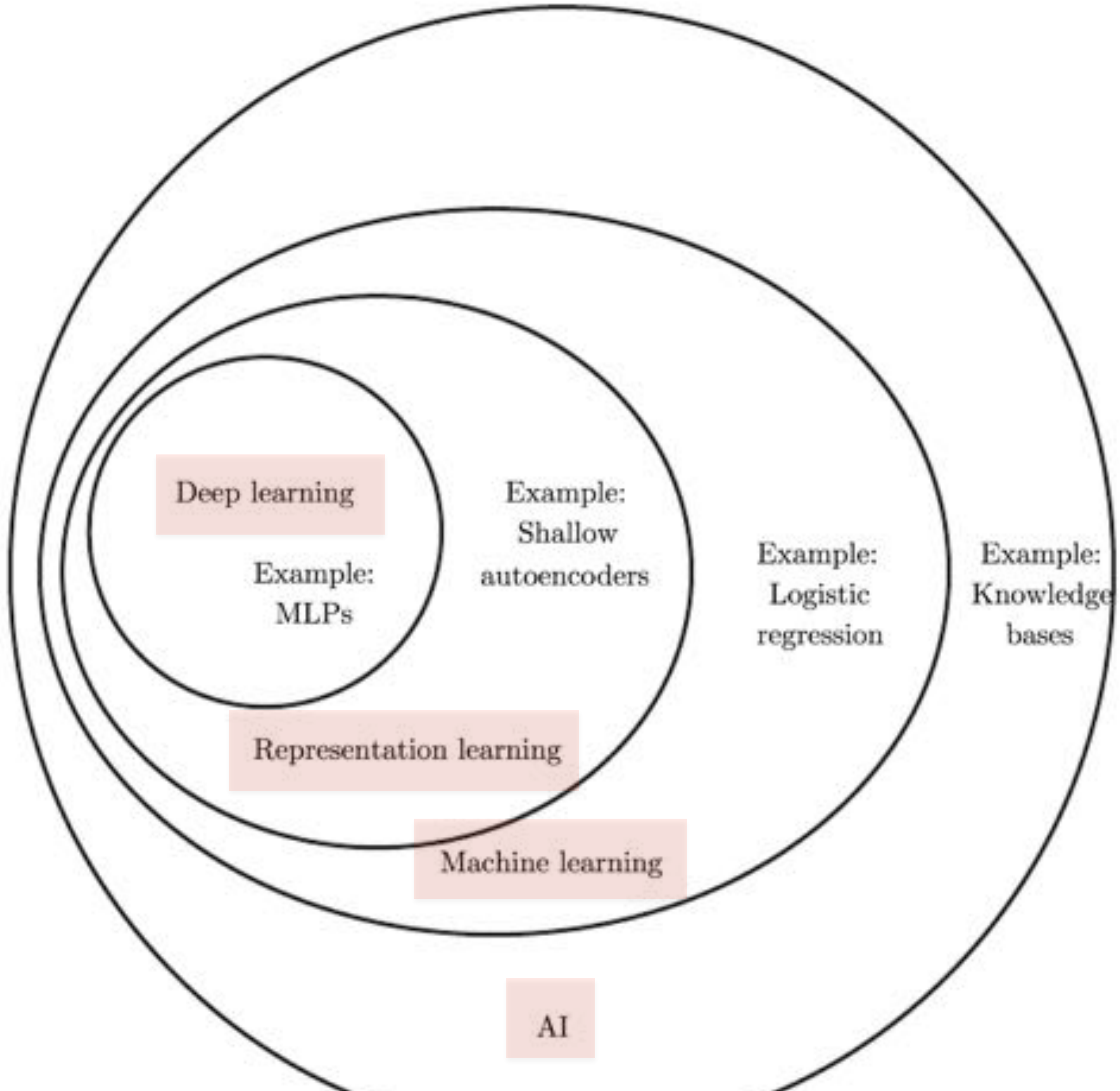
Rise of the machines: The Arts

Which of these was made by an *AI*?



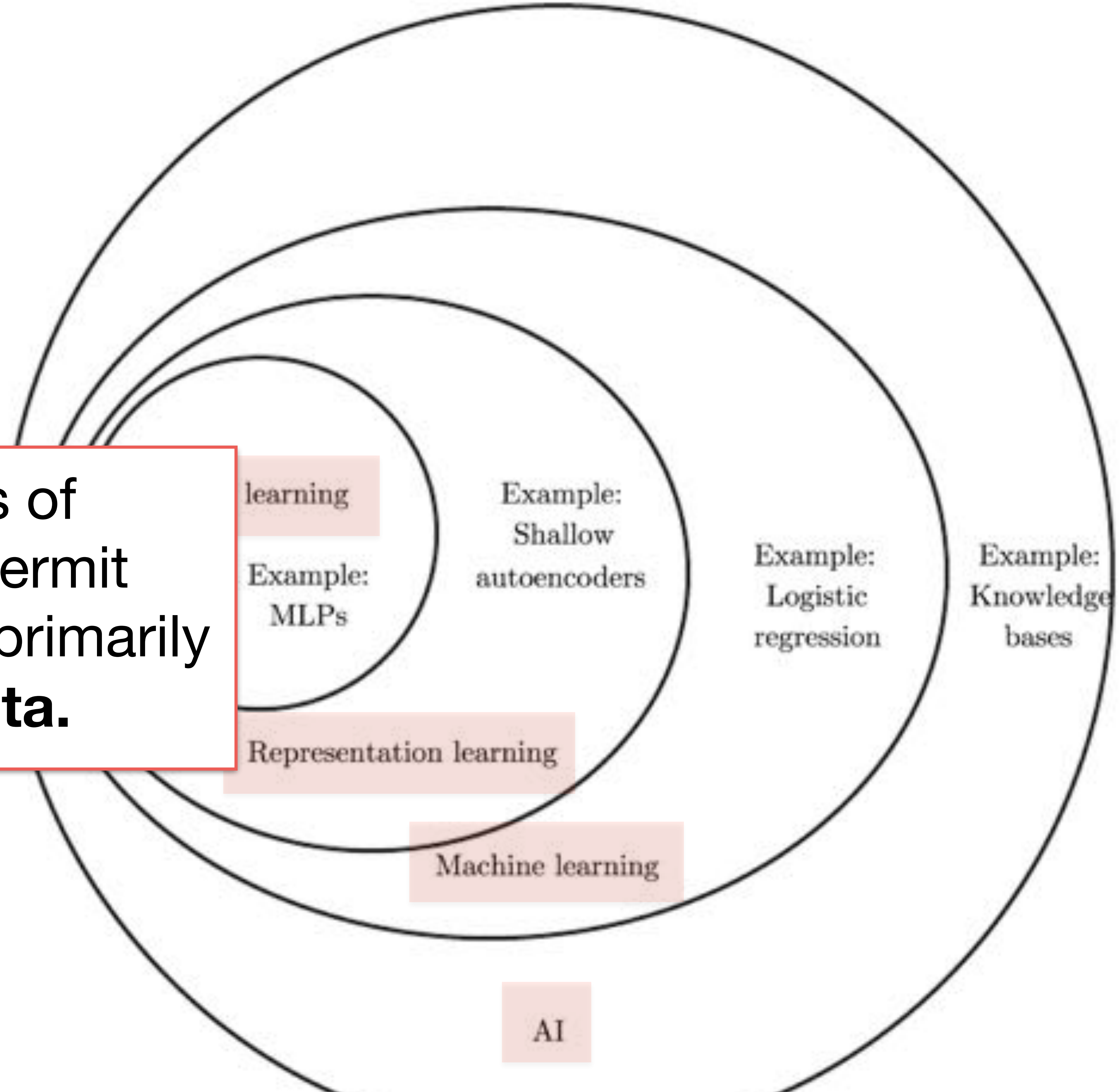
Machine Intelligence:
History and Context

Machine Intelligence Ecosystem



deeplearningbook.org: Goodfellow, Bengio, Courville

Machine Intelligence Ecosystem



These are a class of algorithms that permit modeling that is primarily **driven by the data.**

deeplearningbook.org: Goodfellow, Bengio, Courville

Historical Timeline

Ada Lovelace
The First Computer Programmer
(1815 - 1852)

Historical Timeline

Historical Timeline

1936: Turing comp sci **theory**

1943: McCulloch-Pitts **neuron**

Historical Timeline

1936: Turing comp sci **theory**

1943: McCulloch-Pitts **neuron**

1956: Dartmouth **conference**

1973: Lighthill **Report**

Historical Timeline

1936: Turing comp sci **theory**

1943: McCulloch-Pitts **neuron**

1956: Dartmouth **conference**

1973: Lighthill **Report**

1980: Fukushima **neocognitron**

1988: LeCun's Network*

Historical Timeline

1936: Turing comp sci **theory**

1943: McCulloch-Pitts **neuron**

1956: Dartmouth **conference**

1973: Lighthill **Report**

1980: Fukushima **neocognitron**

1988: LeCun's Network*

2012: **ImageNet** and CNNs

2017: **AlphaGo wins**

Historical Timeline

1936: Turing comp sci **theory**

1943: McCulloch-Pitts **neuron**

1956: Dartmouth **conference**

1973: Lighthill **Report**

1980: Fukushima **neocognitron**

1988: LeCun's Network*

2012: **ImageNet** and CNNs

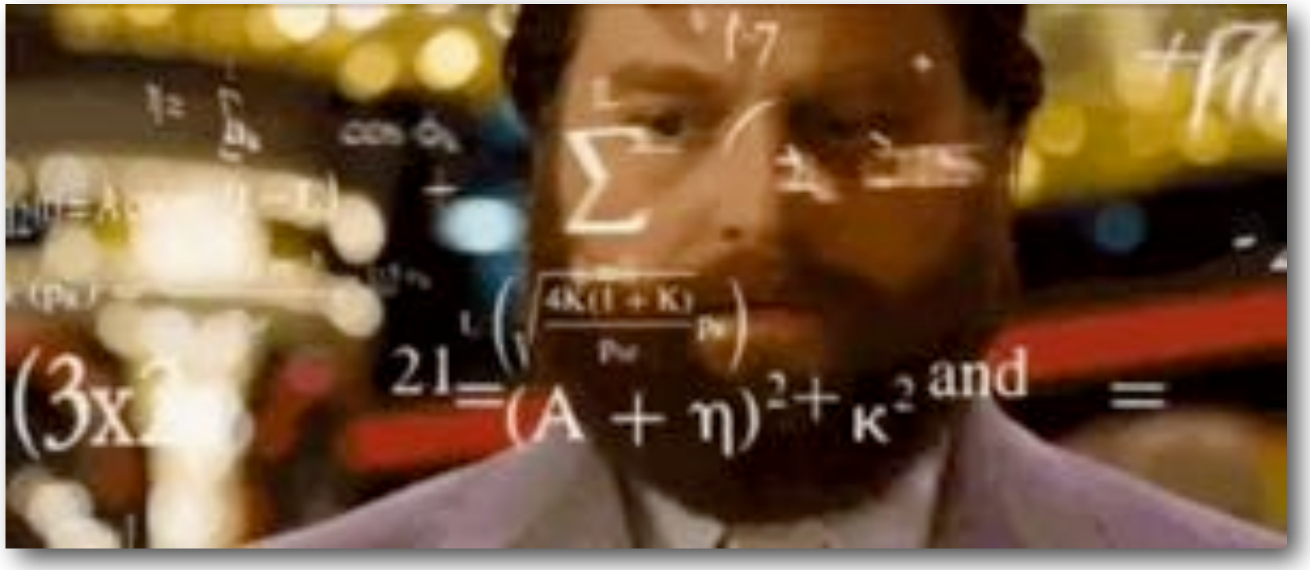
2017: **AlphaGo wins**

AI Winter 1

- Architectures not general.
- **Over-hype**

AI Winter 2

- Algorithms inefficient
- **Over-hype**

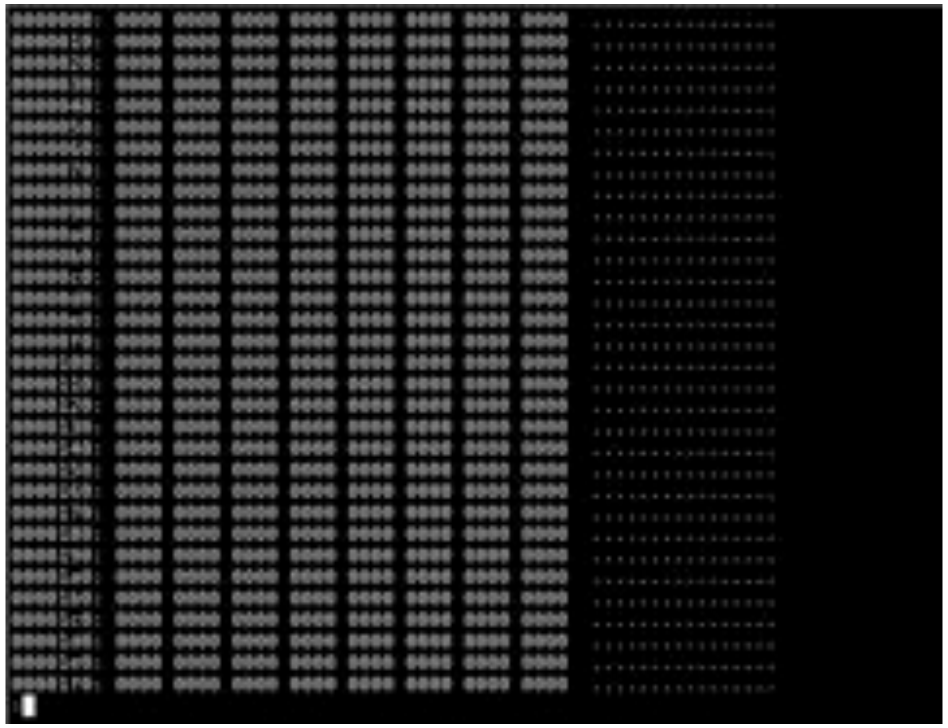


Algorithms

3 Drivers of the 3rd Age

Big Data

Computing



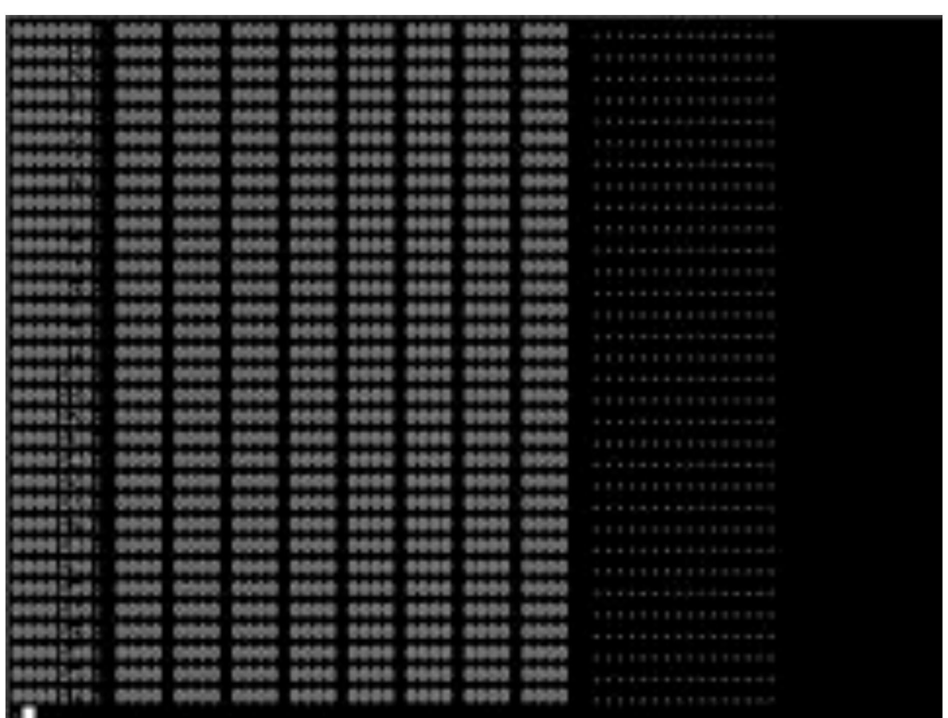
Efficient function optimization

Algorithms

3 Drivers of the 3rd Age

Big Data

Computing



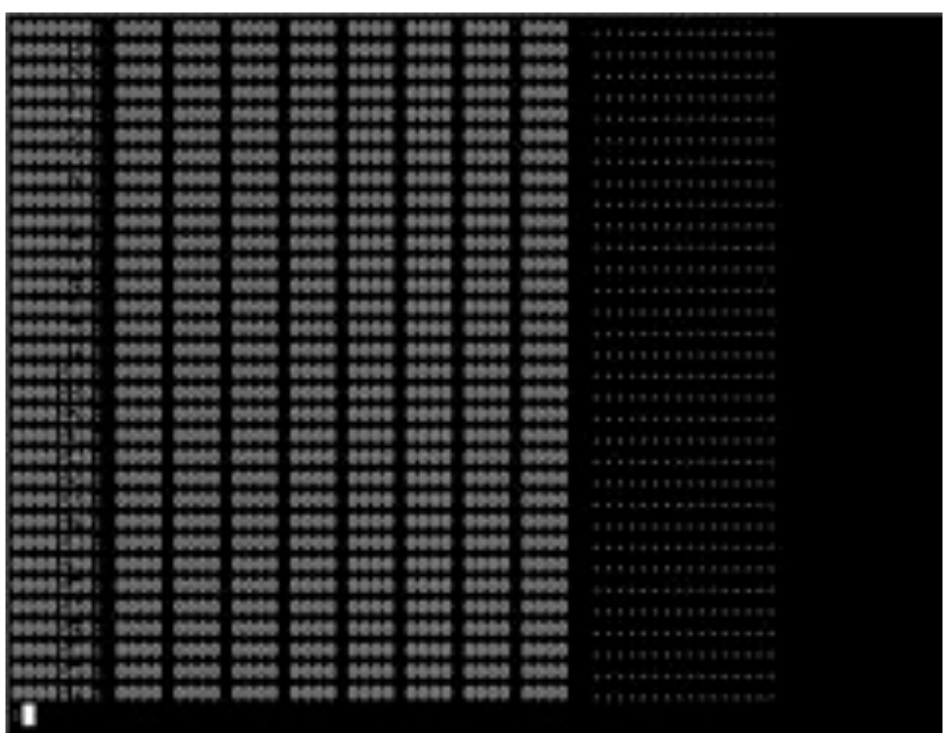
Efficient function optimization

Algorithms

3 Drivers of the 3rd Age

Big Data

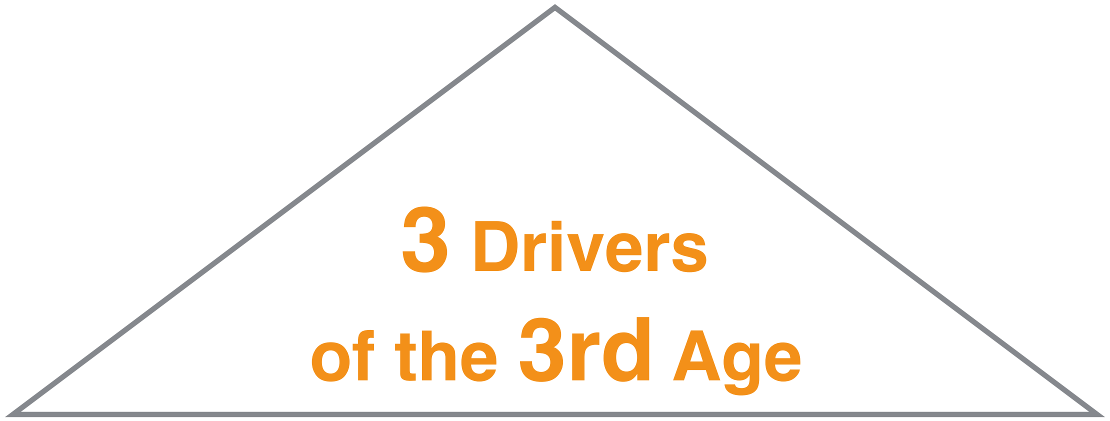
Computing



Graphical Processing Units (GPUs) make computations feasible

Efficient function optimization

Algorithms



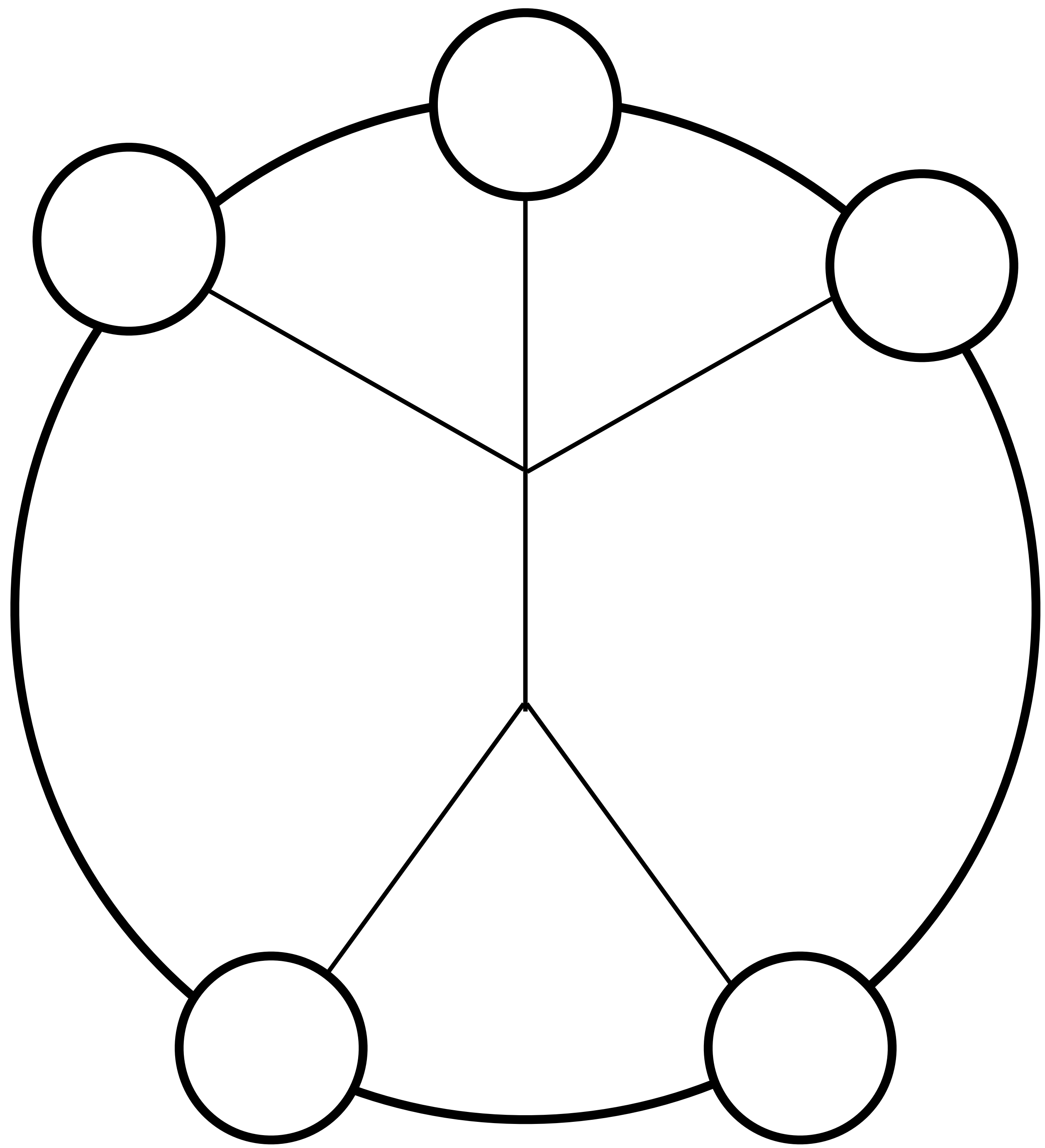
Big Data

Computing

thousands of examples per object

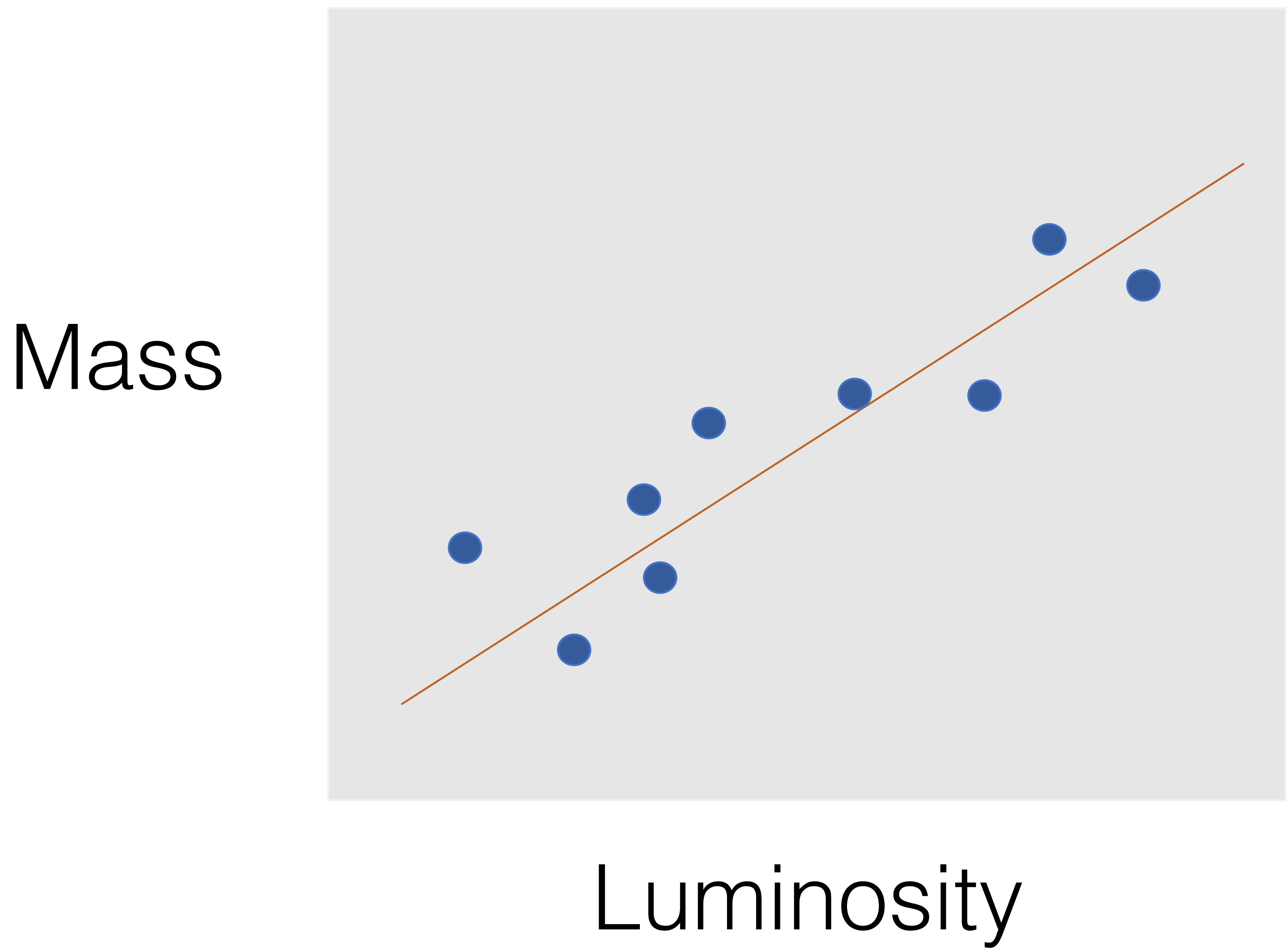
Graphical Processing Units (GPUs) make computations feasible

Anatomy of a Neural Network



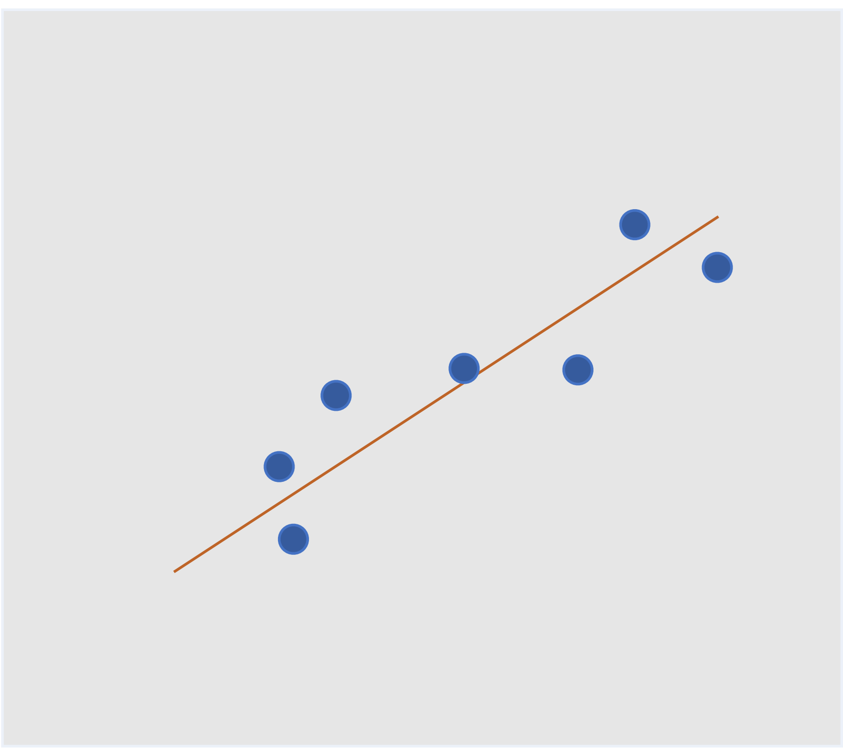
Example: Predict Galaxy Mass from its Luminosity

Mass = f(Luminosity, constants)



Perspectives: *Statistics vs. Machine Learning*

Statistics → inference:
"How can I infer the parameters/process generated my data?"

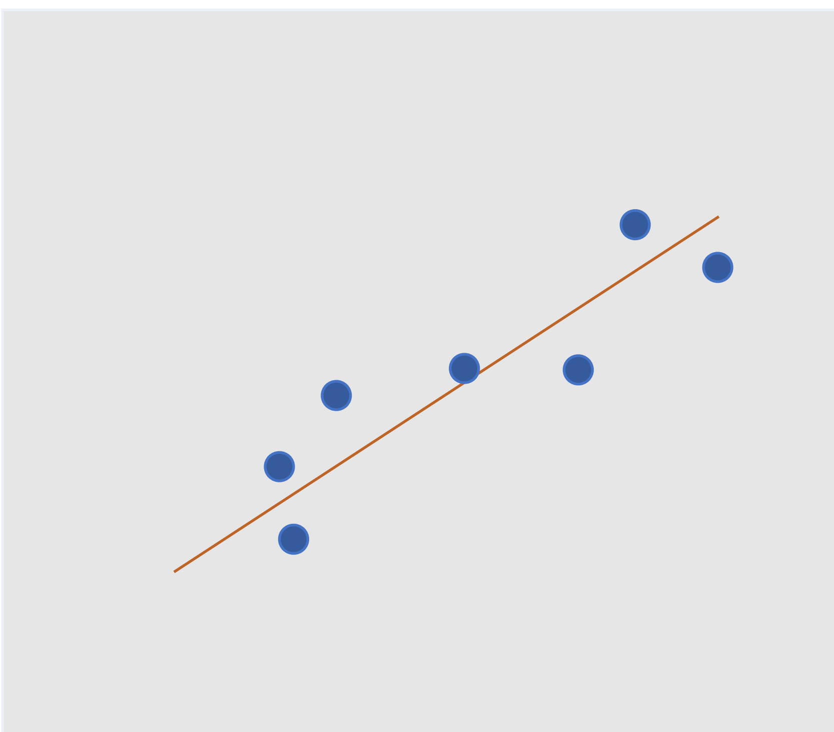


$$y = \mathbf{f}(\mathbf{x}, \mathbf{p})$$

p : Constants with *physical* motivation

Perspectives: *Statistics vs. Machine Learning*

Statistics → inference:
"How can I infer the parameters/process generated my data?"



$$y = f(\mathbf{x}, \mathbf{p})$$

p : Constants with *physical* motivation

Machine Learning → prediction:
"What will future data look like?"

$$z = g(\mathbf{x}, \mathbf{w})$$

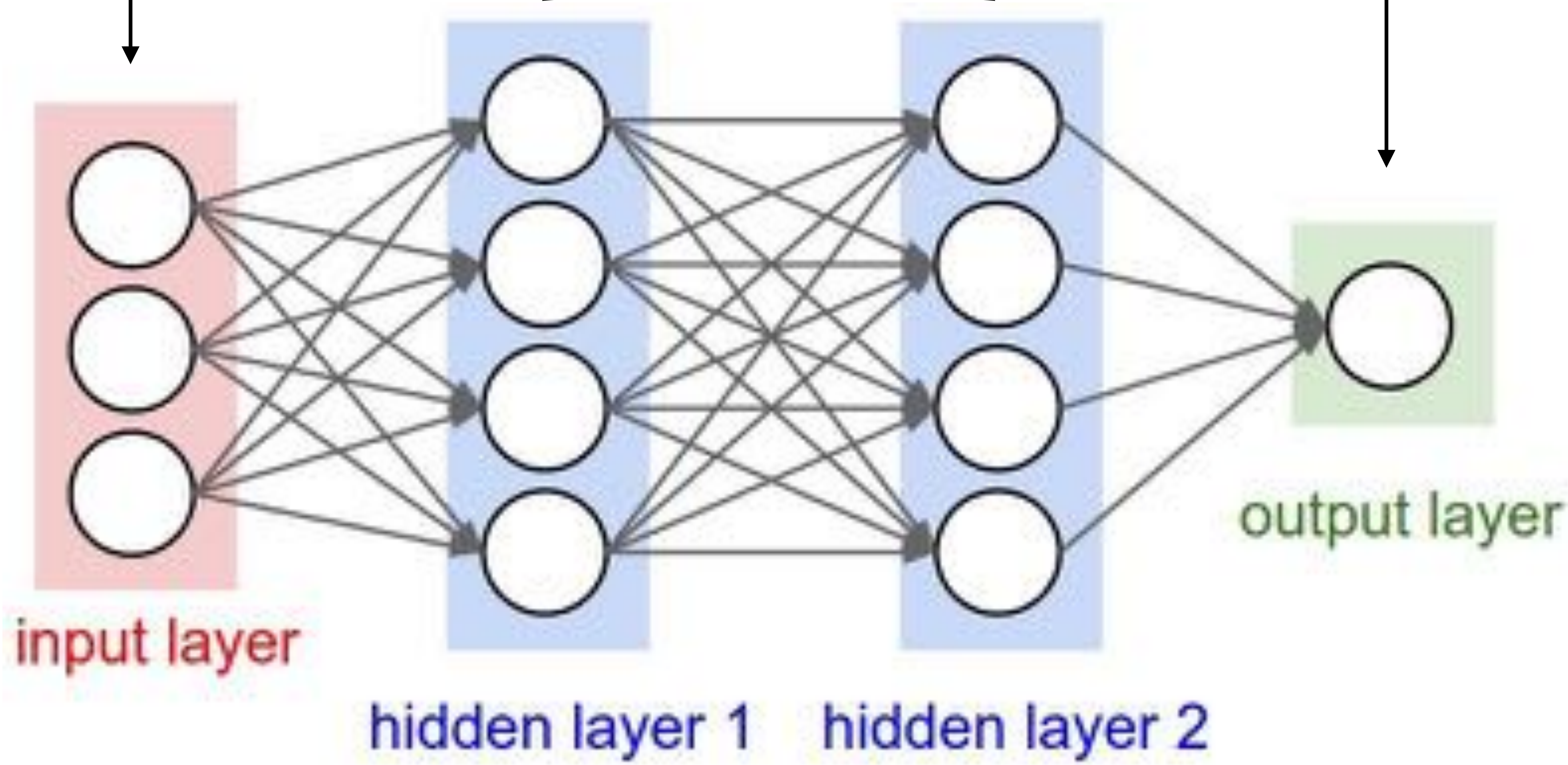
w : Constants that have no physical meaning (yet!)

Overview: Example of Dense Feed Forward Network

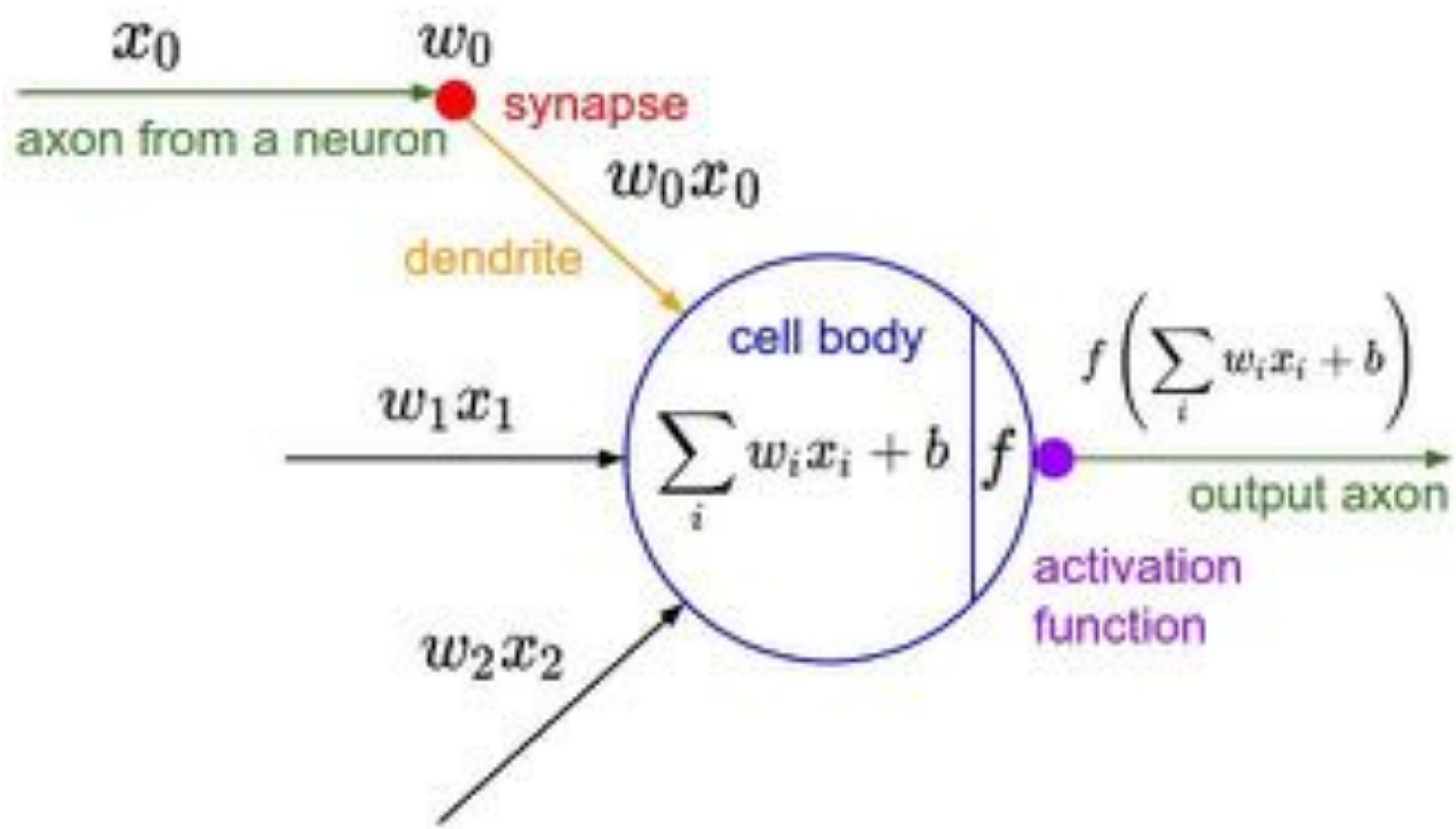
raw data
(e.g., images)

Model
(i.e., weights)

prediction
(e.g., class)

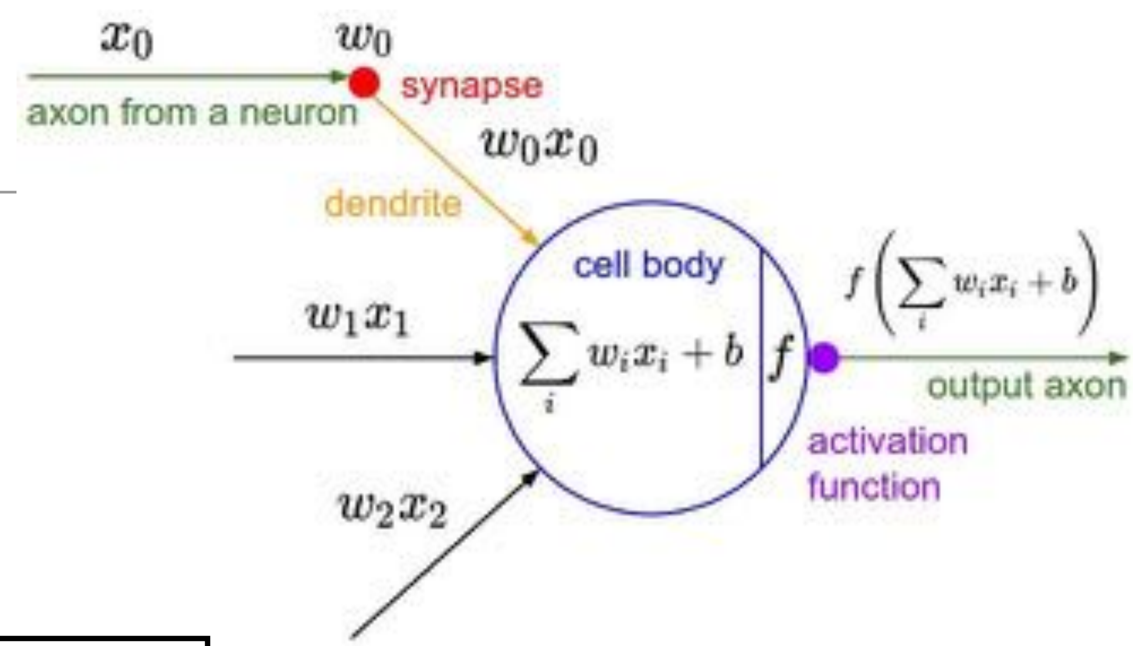


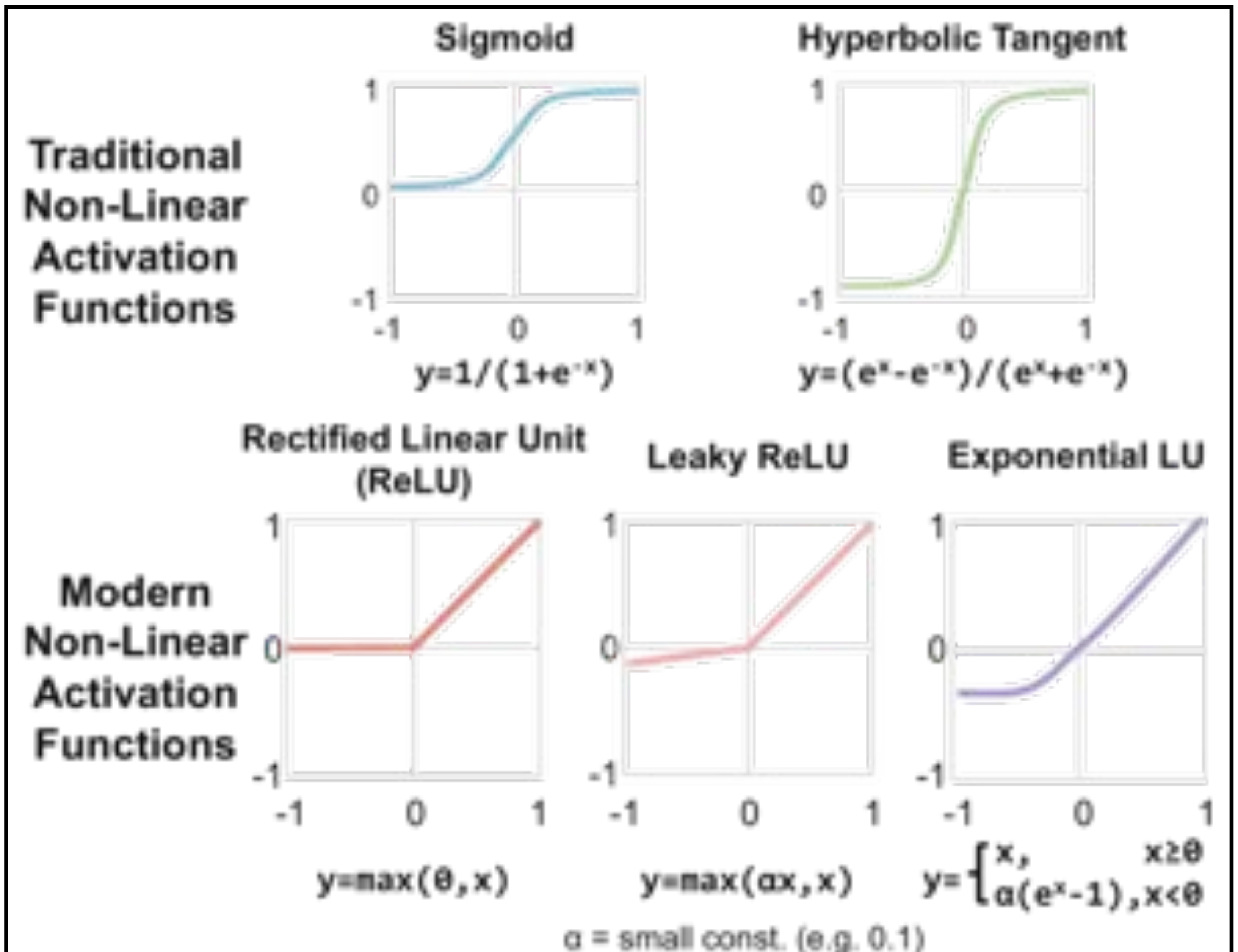
The Neuron



- Weights between neurons are applied (multiplied) to the data value and then combined inside the neuron with the activation function.

Activation Functions





- What is common to all of these activations?
- When might you want to use a sigmoid?

Loss Function

- Mean-square error for regression: prediction (**f**) and true label (**y**)

$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i) \right)^2$$

- Cross-entropy for classification:

$$H(p, q) \;=\; -\sum_i p_i \log q_i \;=\; -\, y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

Neural Network (CNN): Overview

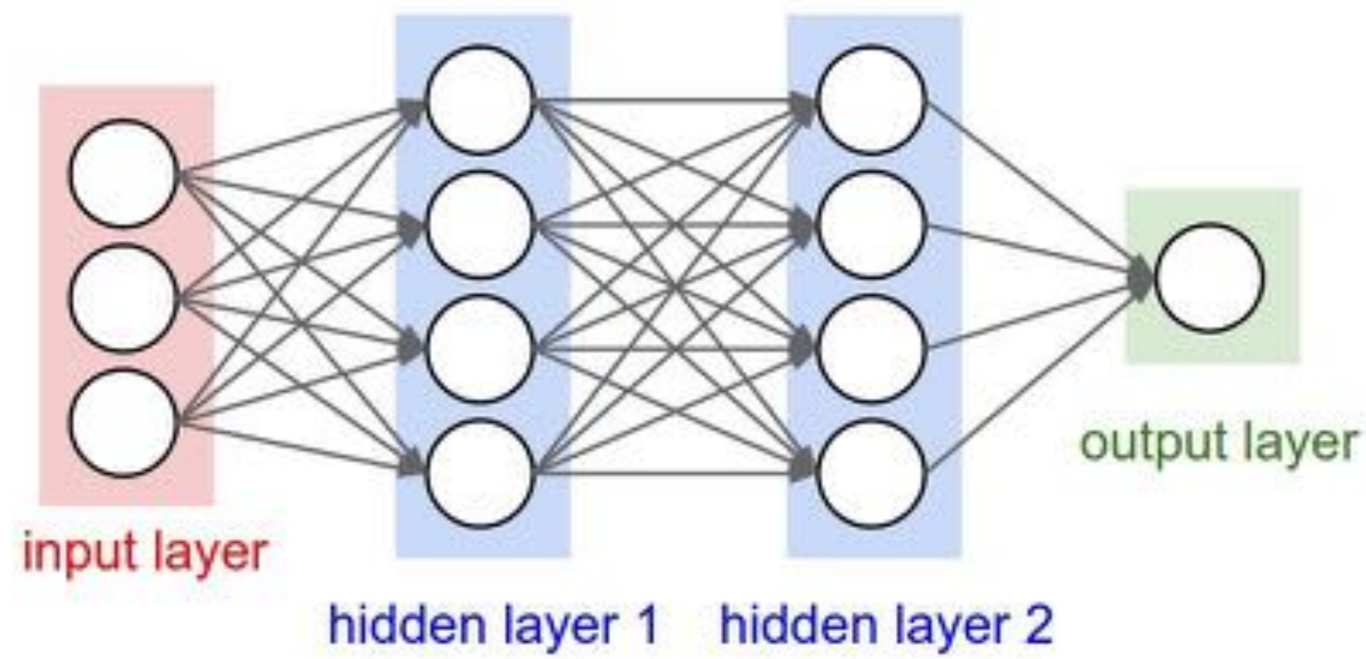
input

model

prediction

cats

dogs



cat

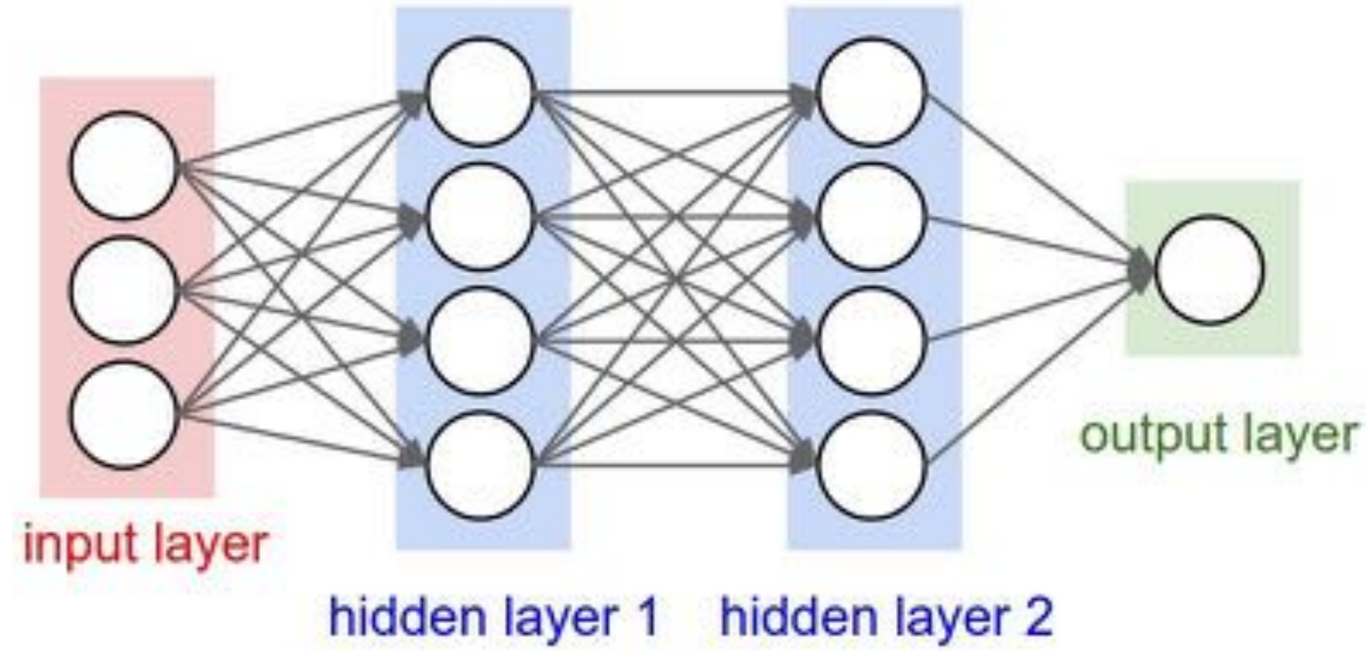
?

dog

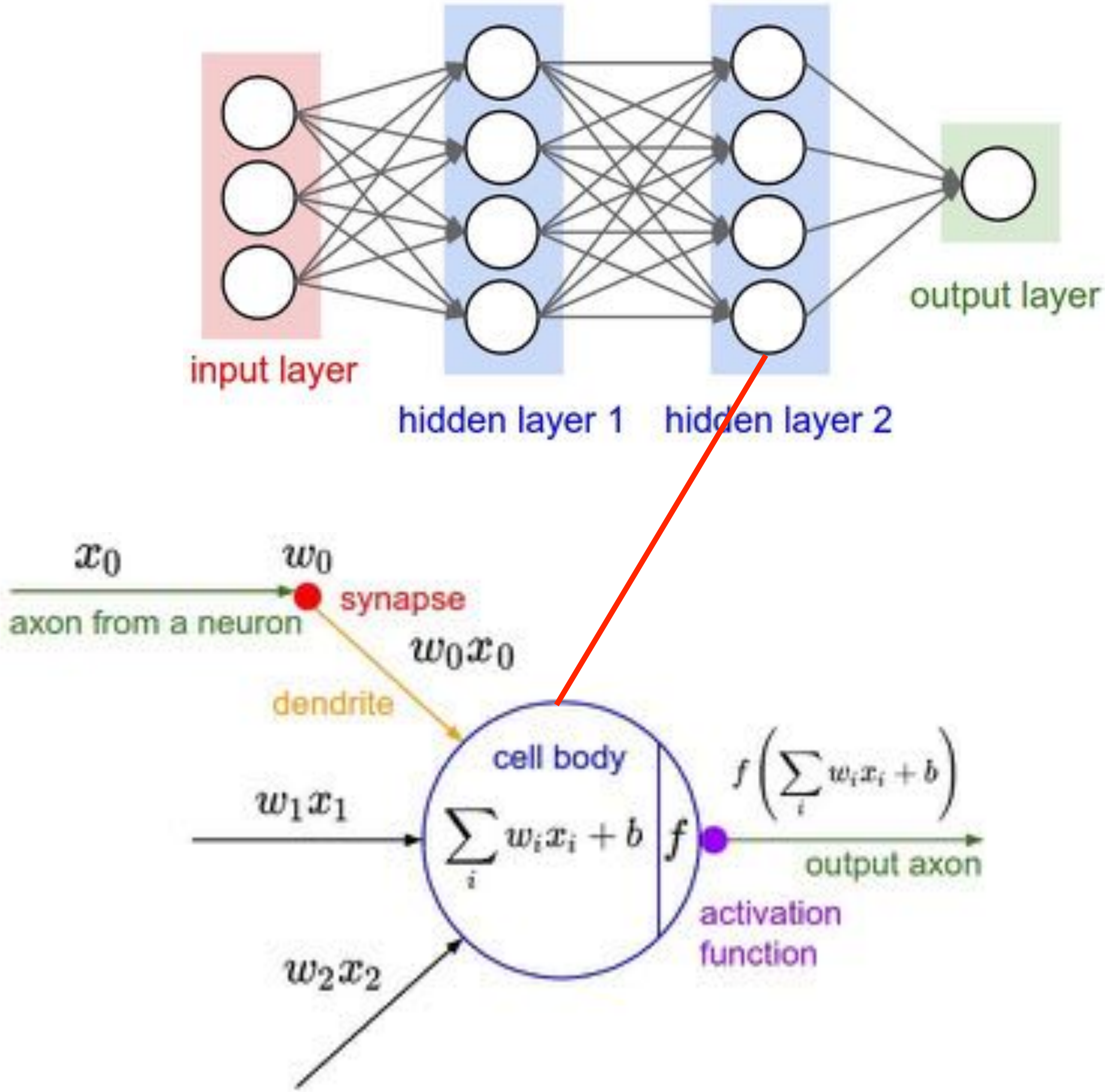
How to train your ~~dragon~~ neural network

Supervised Learning Types

Neural Network: A sum of parts

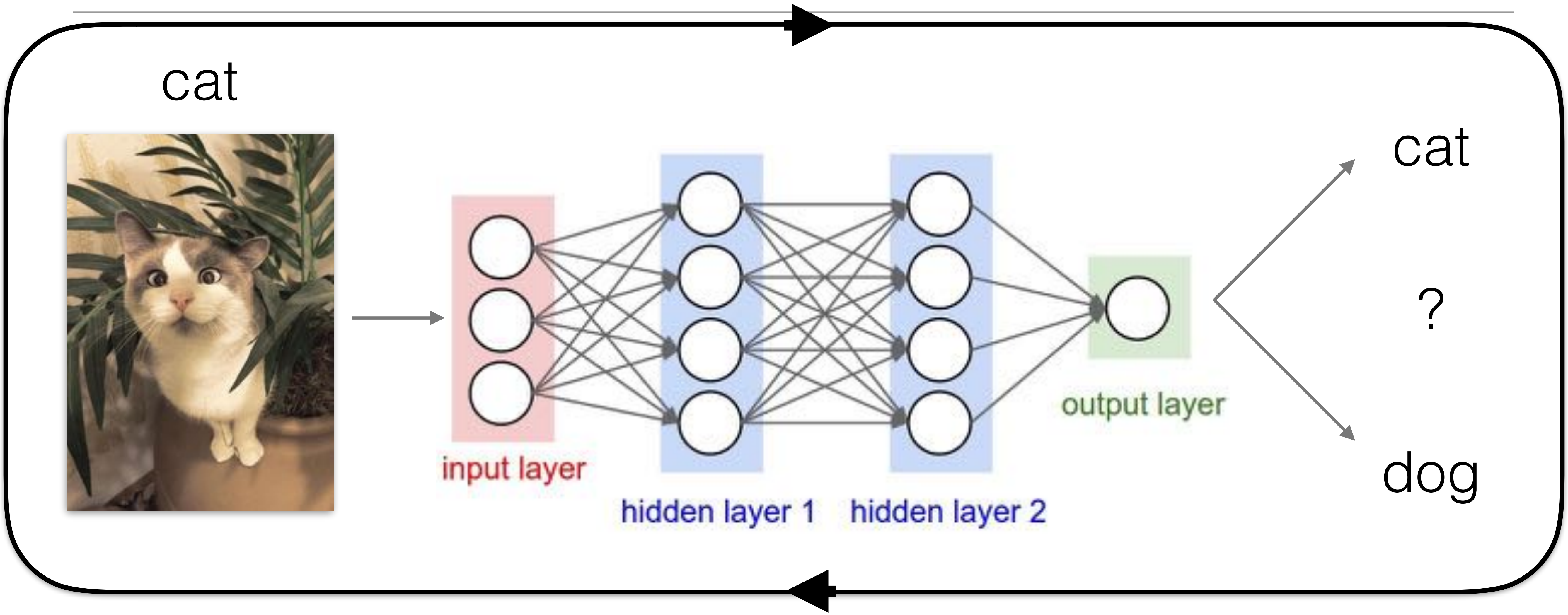


Neural Network: A sum of parts



http://cs231n.github.io/assets/nn1/neural_net2.jpeg

Convolutional Neural Network: Training (Fitting)

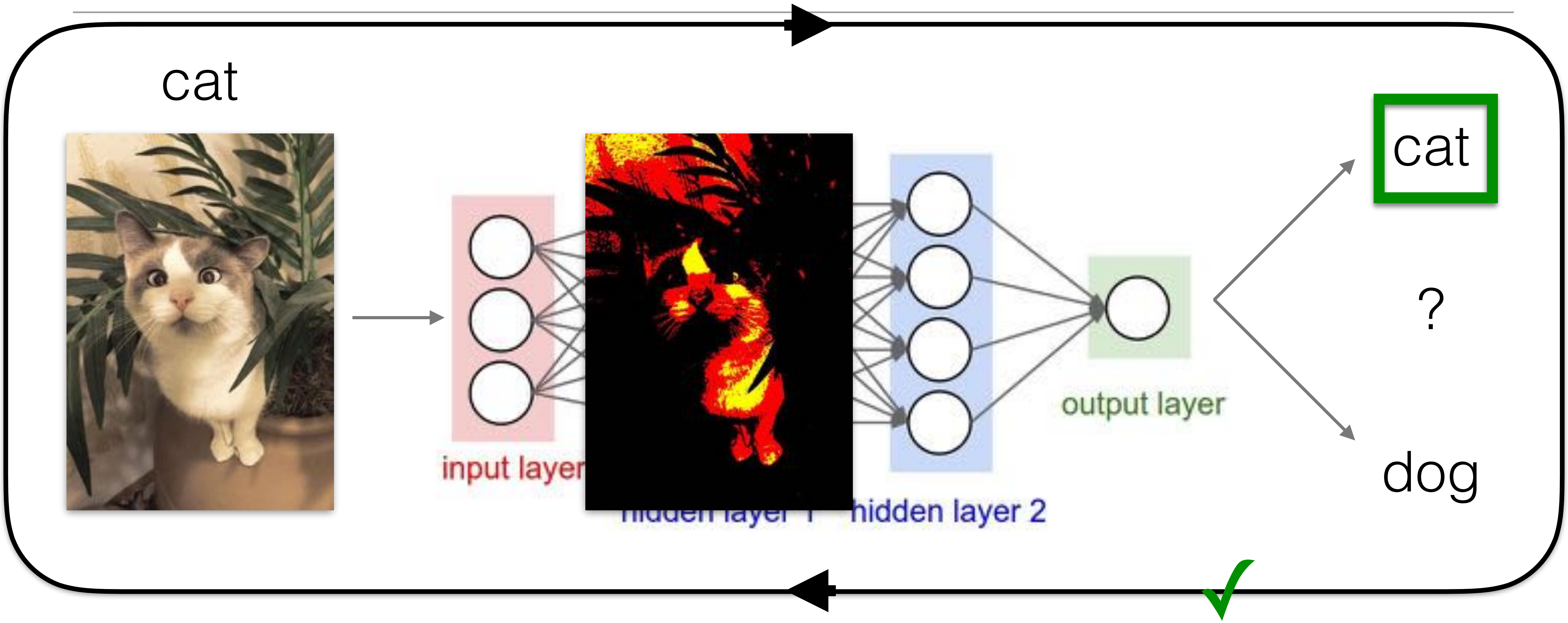


- <u>Minimize error (**E**)</u>
  minimize error between prediction (**f**) and true label (**y**)

$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i)\right)^2$$

- Keeps weights (**w**) that are good for the prediction

Convolutional Neural Network: Training (Fitting)

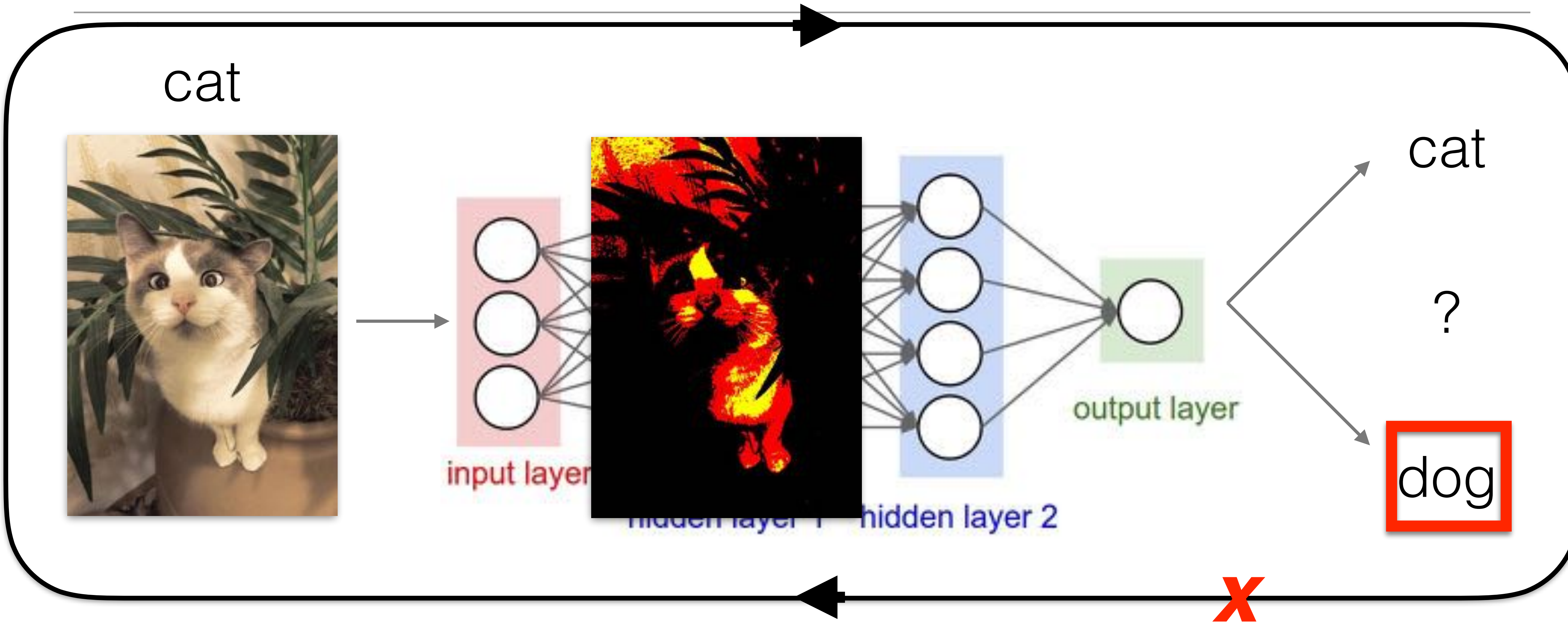


- <u>Minimize error (**E**)</u>
  minimize error between prediction (**f**) and true label (**y**)

$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i)\right)^2$$

- Keeps weights (**w**) that are good for the prediction

Convolutional Neural Network: Training (Fitting)



- <u>Minimize error (**E**)</u>
  minimize error between prediction (**f**) and true label (**y**)

$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i)\right)^2$$

- Keeps weights (**w**) that are good for the prediction

Backpropagation = Chain Rule

Loss function: $$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Update rule: $$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta)$$

Backpropagation = Chain Rule

Loss function: $$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Update rule: $$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta)$$

Derivation:
$$\begin{aligned}
\frac{\partial}{\partial\theta_j}J(\theta) &= \frac{\partial}{\partial\theta_j}\frac{1}{2}\left(h_\theta(x) - y\right)^2 \\
&= 2\cdot\frac{1}{2}\left(h_\theta(x) - y\right)\cdot\frac{\partial}{\partial\theta_j}\left(h_\theta(x) - y\right) \\
&= \left(h_\theta(x) - y\right)\cdot\frac{\partial}{\partial\theta_j}\left(\sum_{i=0}^{n}\theta_i x_i - y\right) \\
&= \left(h_\theta(x) - y\right)x_j
\end{aligned}$$

Backpropagation = Chain Rule

Loss function: $$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

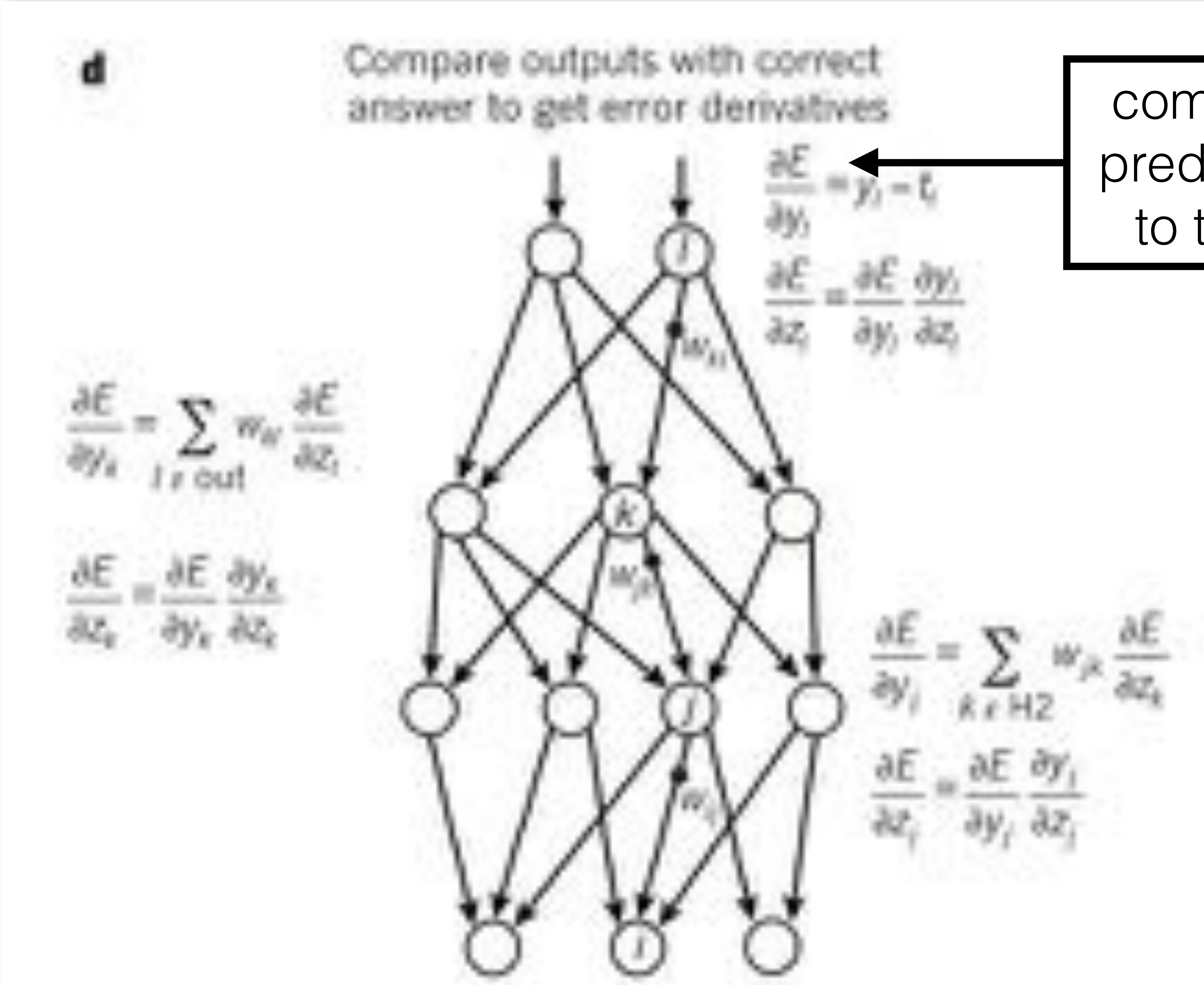
Update rule: $$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$\theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}$$

Derivation:

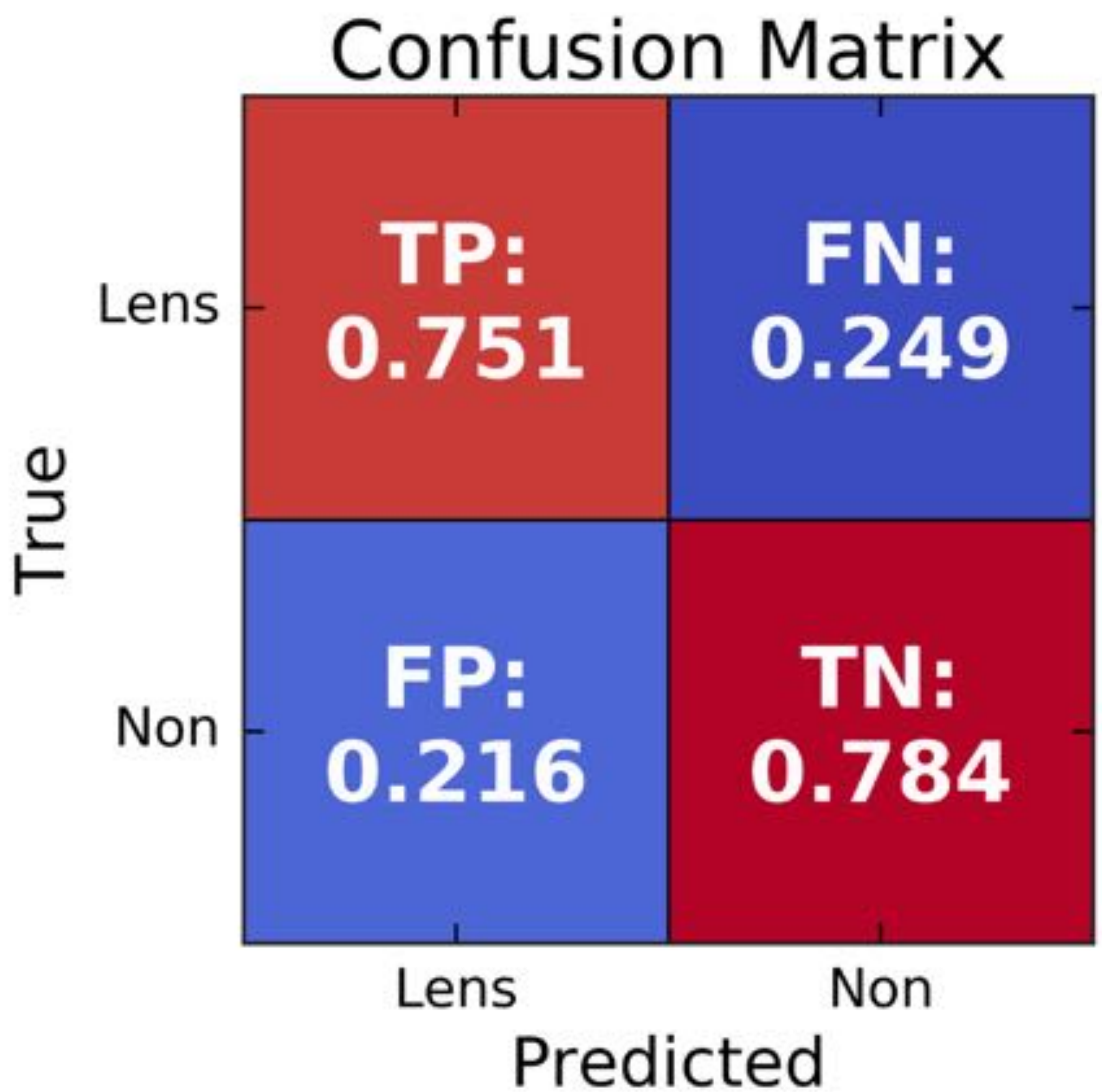
$$\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_\theta(x) - y)^2 \\
&= 2 \cdot \frac{1}{2} (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x) - y) \\
&= (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{n} \theta_i x_i - y \right) \\
&= (h_\theta(x) - y)\, x_j
\end{aligned}$$

Backpropagation = Chain Rule



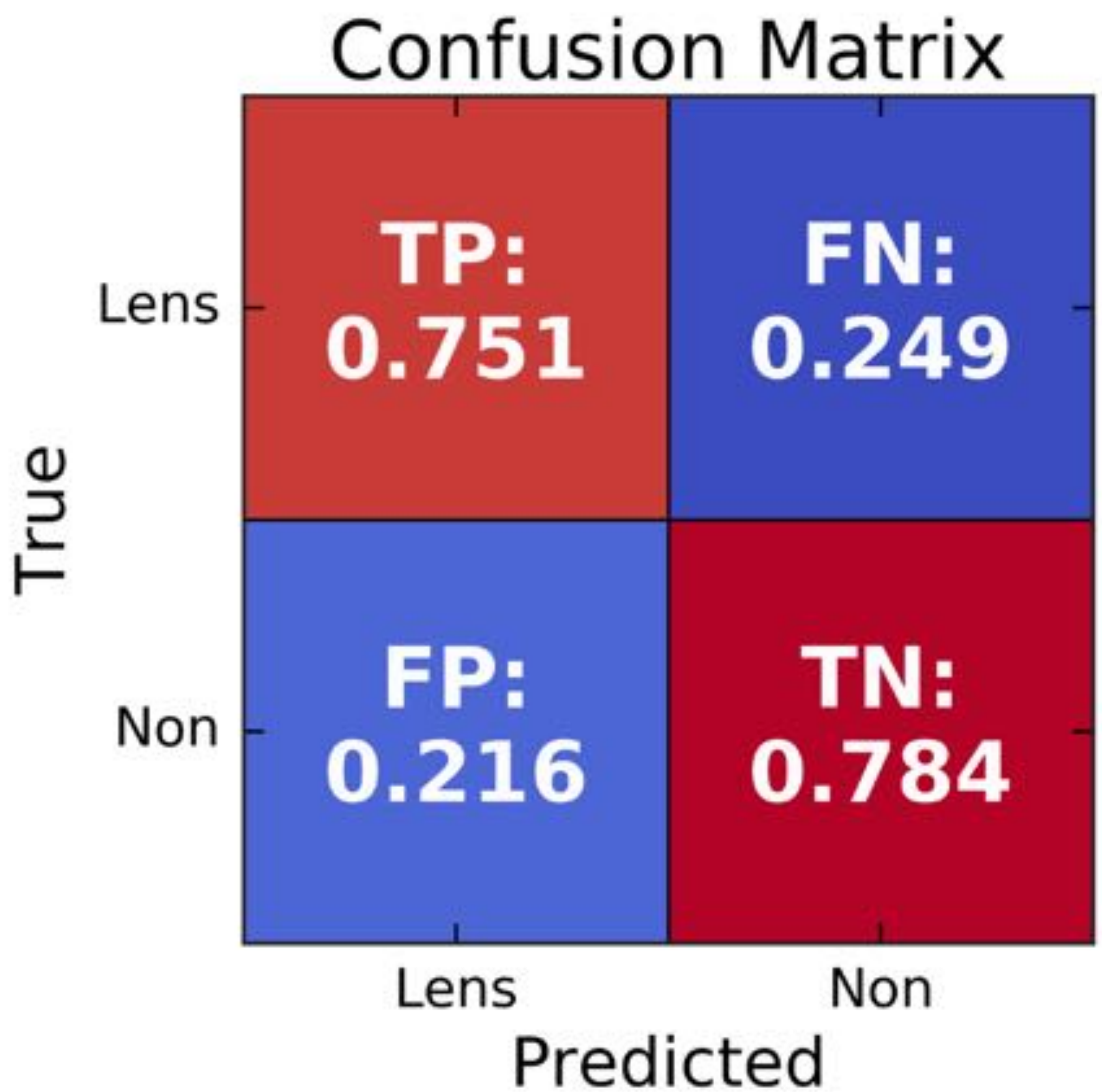
compare prediction to truth

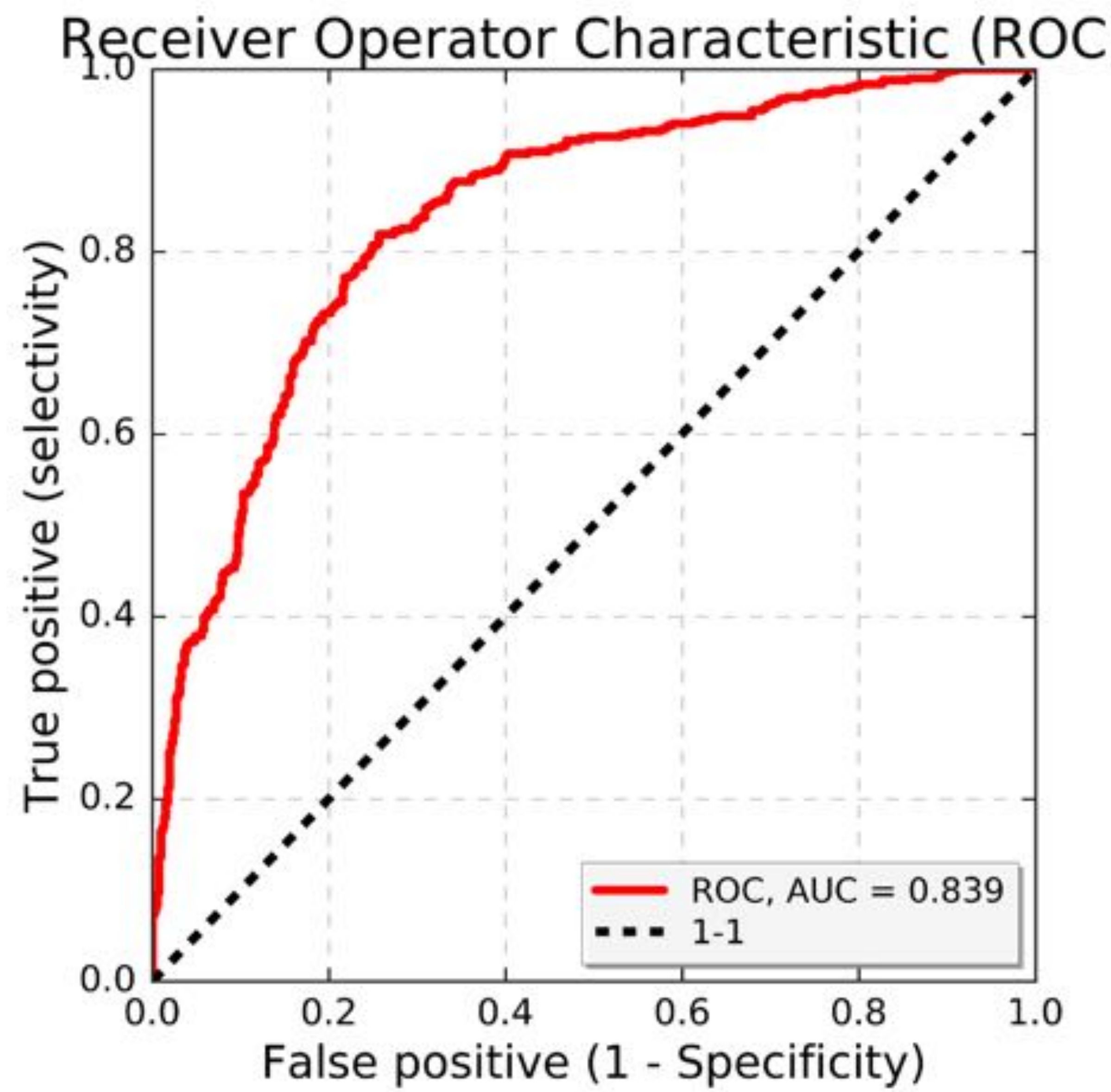
Diagnostics: Examples from a strong lensing classification



Confusion matrix shows **high precision and recall** when testing on images NOT used for training.

Diagnostics: Examples from a strong lensing classification

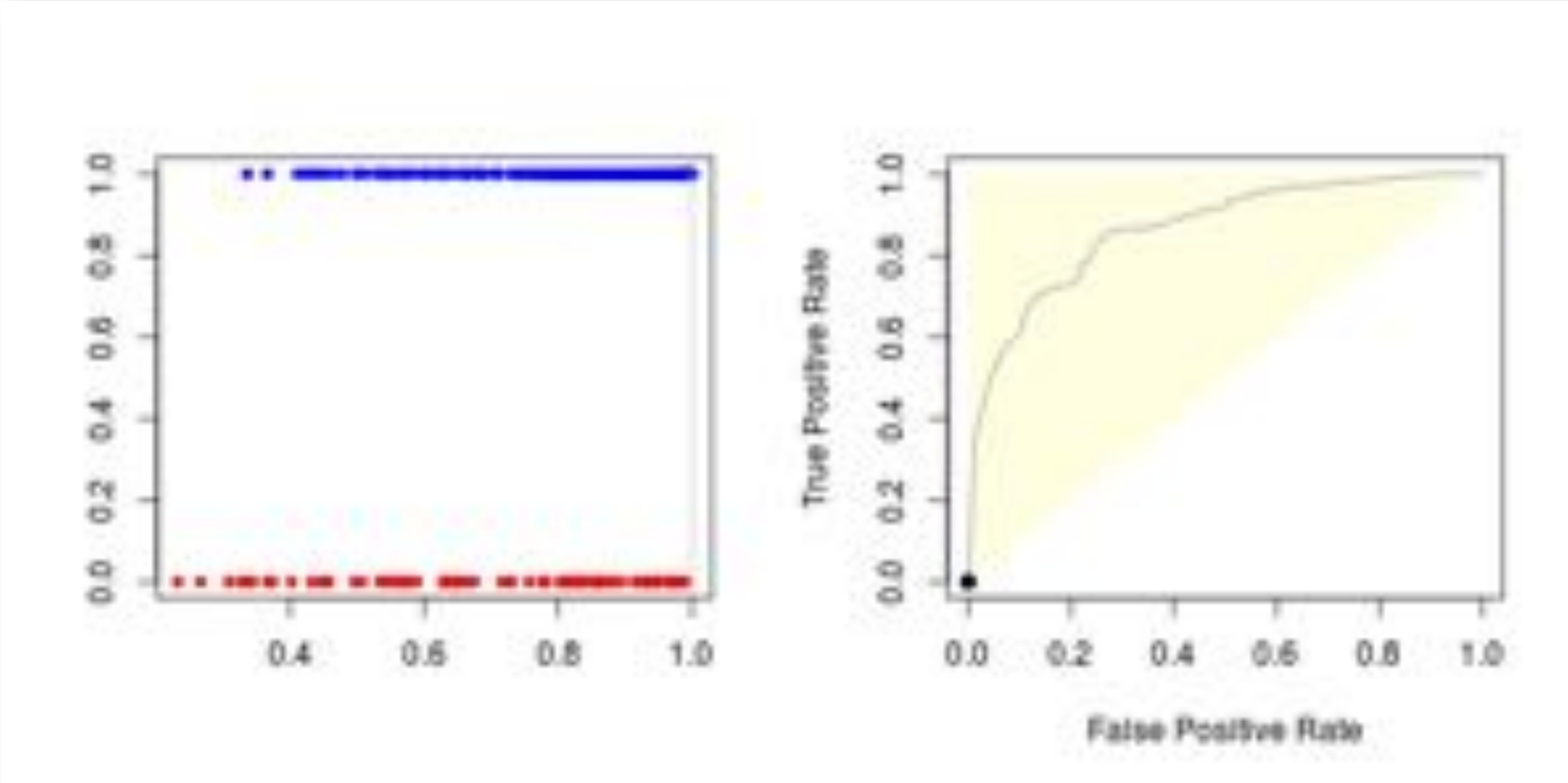




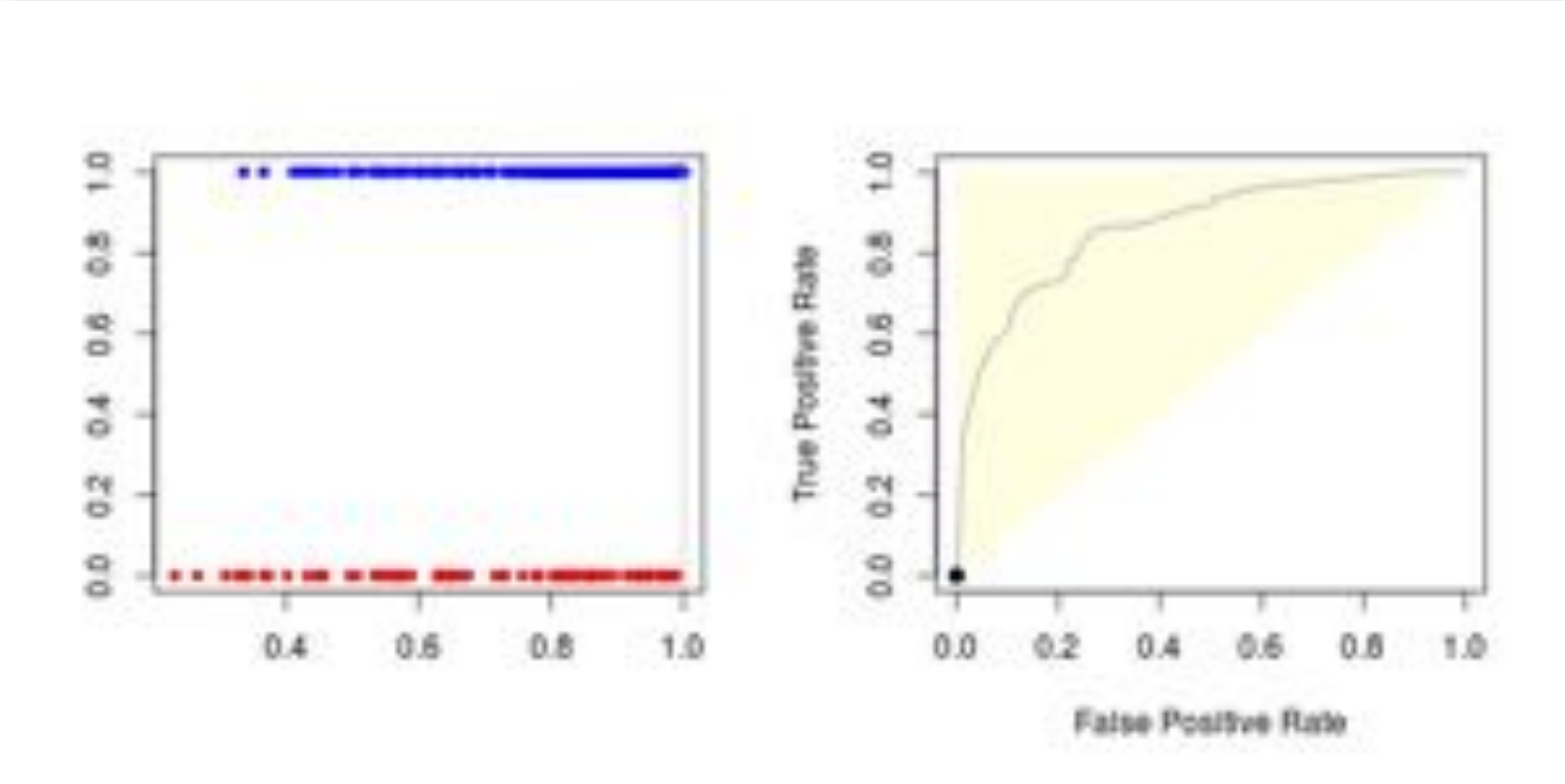
<u>Confusion matrix</u> shows **high precision and recall** when testing on images NOT used for training.

<u>ROC Curve</u> shows the accuracy as the threshold of probability for detection is incremented

ROC Curve in Action



ROC Curve in Action



ROC Curve shows the accuracy as the threshold of probability for detection is incremented

Mapping your problem to the tool: the questions

- <u>Speed</u>:
  Does your task take a long time by conventional means?
- <u>Complexity of data:</u>
  Is your raw data very complex? Too complex for humans to identify the principal features?
- <u>Simplicity of problem:</u>
  Is there a low-hanging fruit? E.g., classification

- *Advice*:
  - Think of the goal for your problem.
  - Then, stopping thinking about the science, and start thinking about the data - its structure and volume.

Challenges in Applications of Neural Nets to Science

- SO MANY PARAMETERS … that are apparently non-physical. How do we interpret this?
  - Easily 1M parameters for a large data set.
- Uncertainties are not formalized for neural networks. We don’t know how to interpret error bars on physical quantities.
- Where’s the physics? Where are the physically interpretable parameters?
- There’s no formal mathematical theory for a neural network. (Hack problem!)