Machines Are Learning

Bringing Powerful Artificial Intelligence Tools to Developers

Danilo Poccia

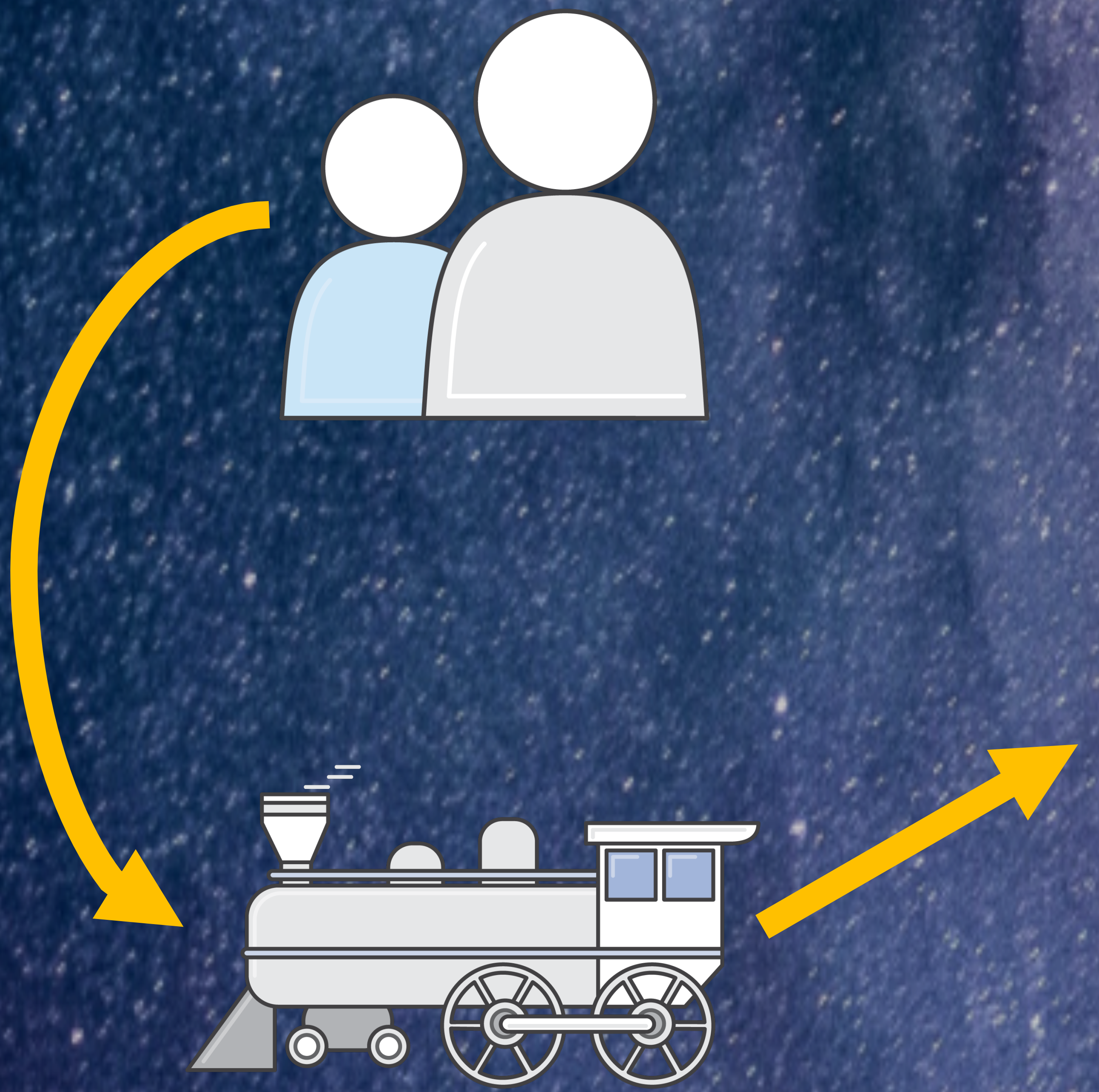
AWS Technical Evangelist

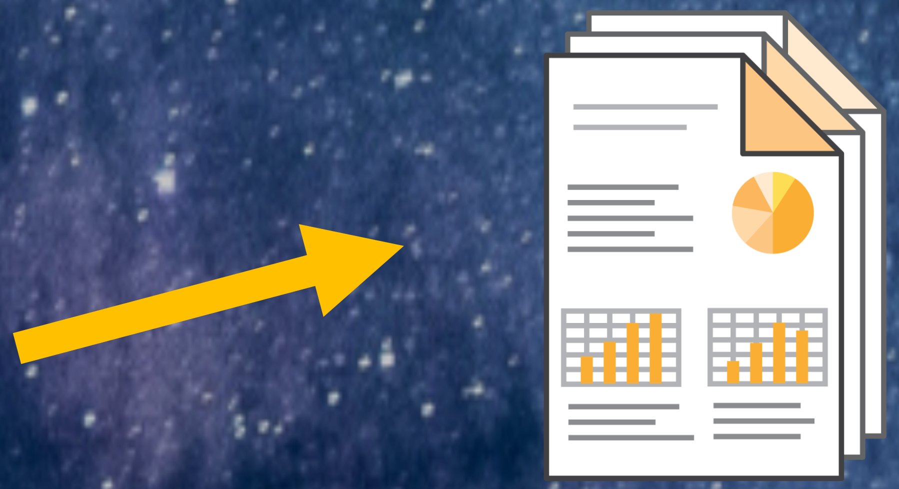
@danilop danilop

CloudConf 2017

Credit: Gerry Cranham/Fox Photos/Getty Images
http://www.telegraph.co.uk/travel/destinations/europe/united-kingdom/england/london/galleries/The-history-of-the-Tube-in-pictures-150-years-of-London-Underground/1939-ticket-examin/

1939 London Underground





Credit: Gerry Cranham/Fox Photos/Getty Images
http://www.telegraph.co.uk/travel/destinations/europe/united-kingdom/england/london/galleries/The-history-of-the-Tube-in-pictures-150-years-of-London-Underground/1939-ticket-examin/


Data
Predictions


Data
Model
Predictions

Model

1959 Arthur Samuel

http://www.thehudsonvalley.com/articles/60-years-ago-today-local-technology-demonstrated-artificial-intelligence-for-the-first-time

Machine Learning

Supervised Learning

↑

Machine Learning

Inferring a model from labeled training data

Supervised Learning

Inferring a model from labeled training data

Machine Learning

Unsupervised Learning

Inferring a model to describe hidden structure from unlabeled data

Reinforcement Learning

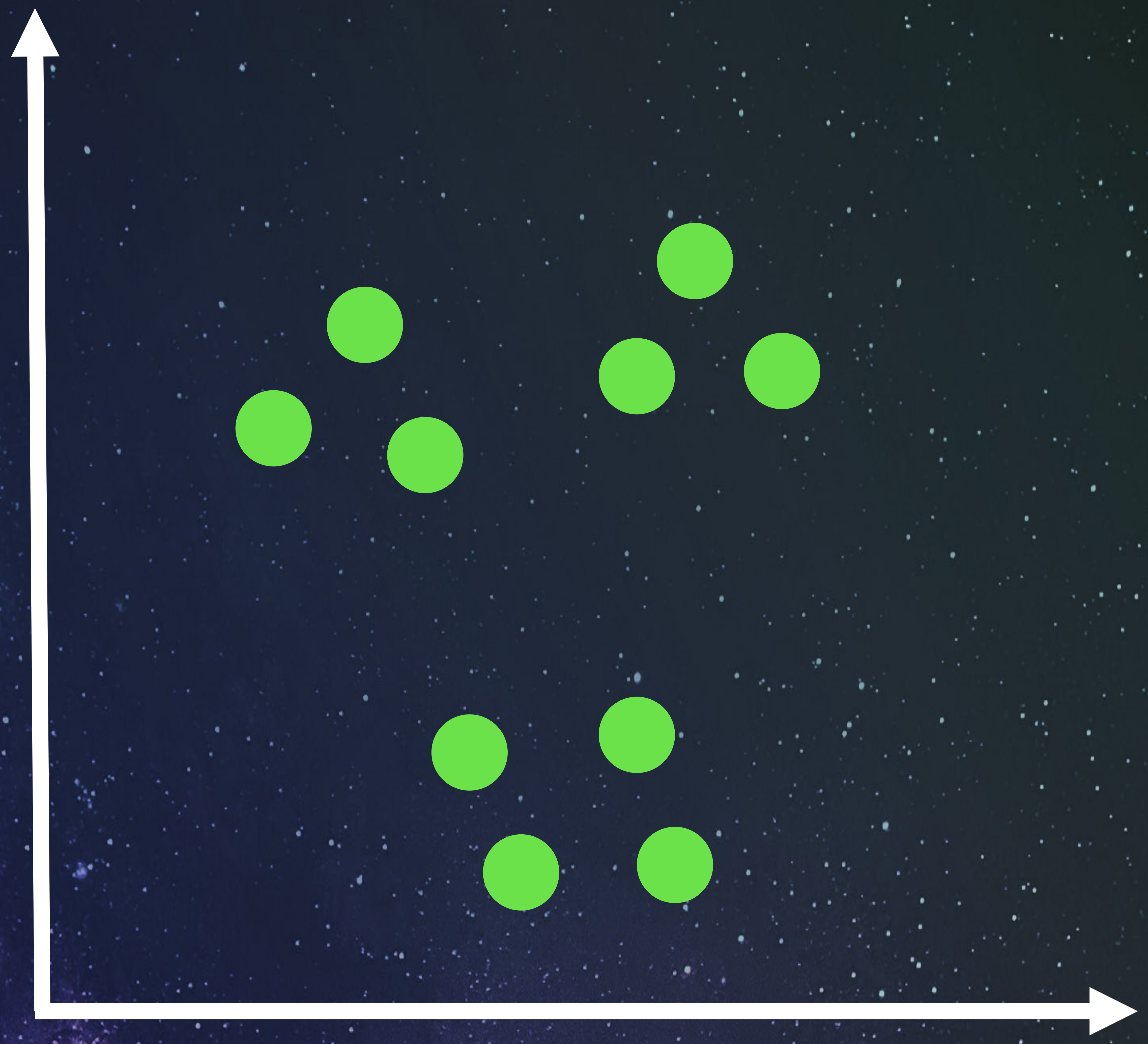
Perform a certain goal in a dynamic environment

Driving a vehicle

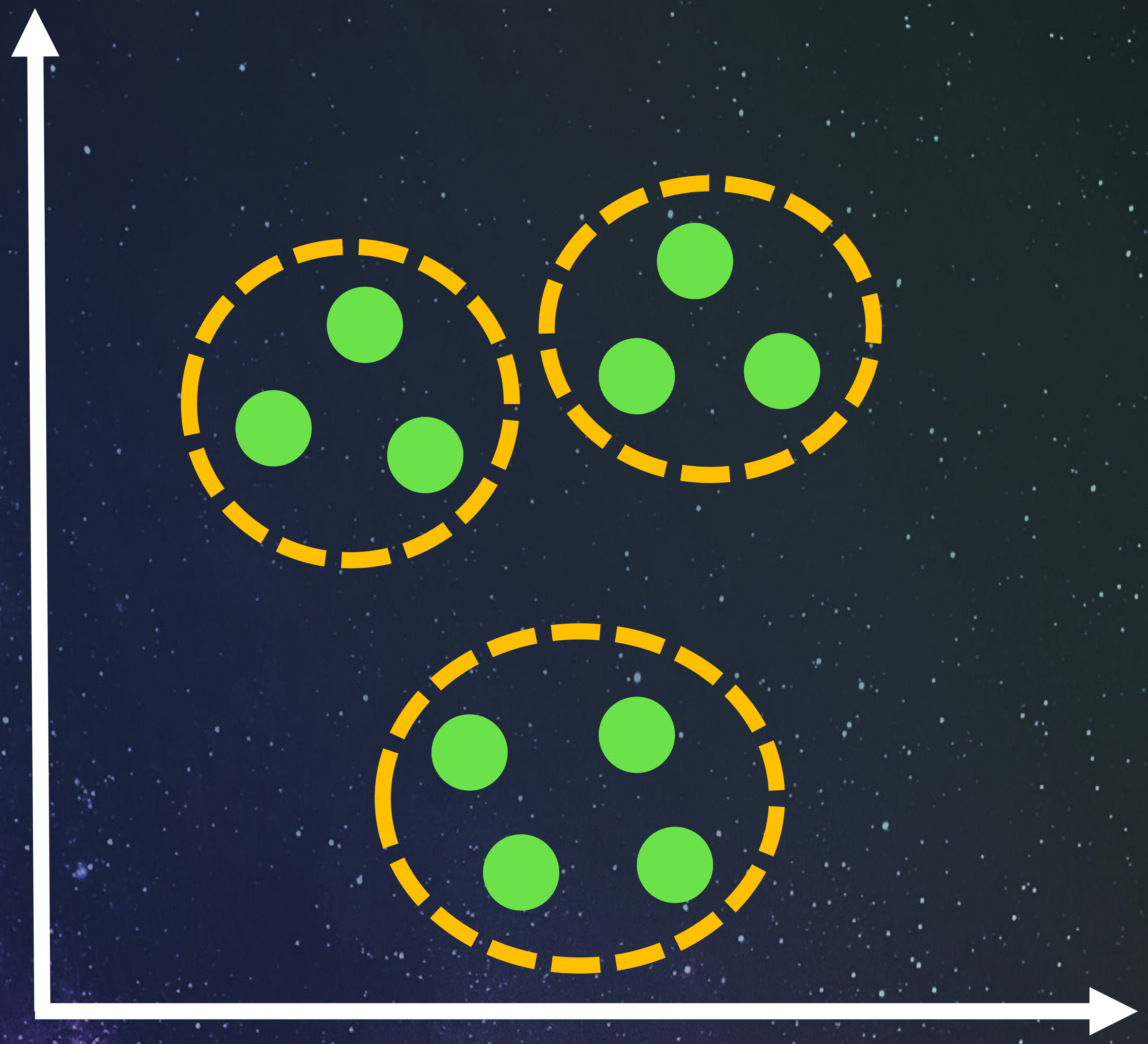
Playing a game
against an opponent

Unsupervised Learning

Clustering



Clustering



Supervised Learning

Regression	"How many bikes will be rented tomorrow?"	1, 0, 100K
Binary Classification	"Is this email spam?"	Yes / No True / False %
Multi-Class Classification	"What is the sentiment of this tweet, or of this social media comment?"	Happy, Sad, Angry, Confused, Disgusted, Surprised, Calm, Unknown

Supervised Learning

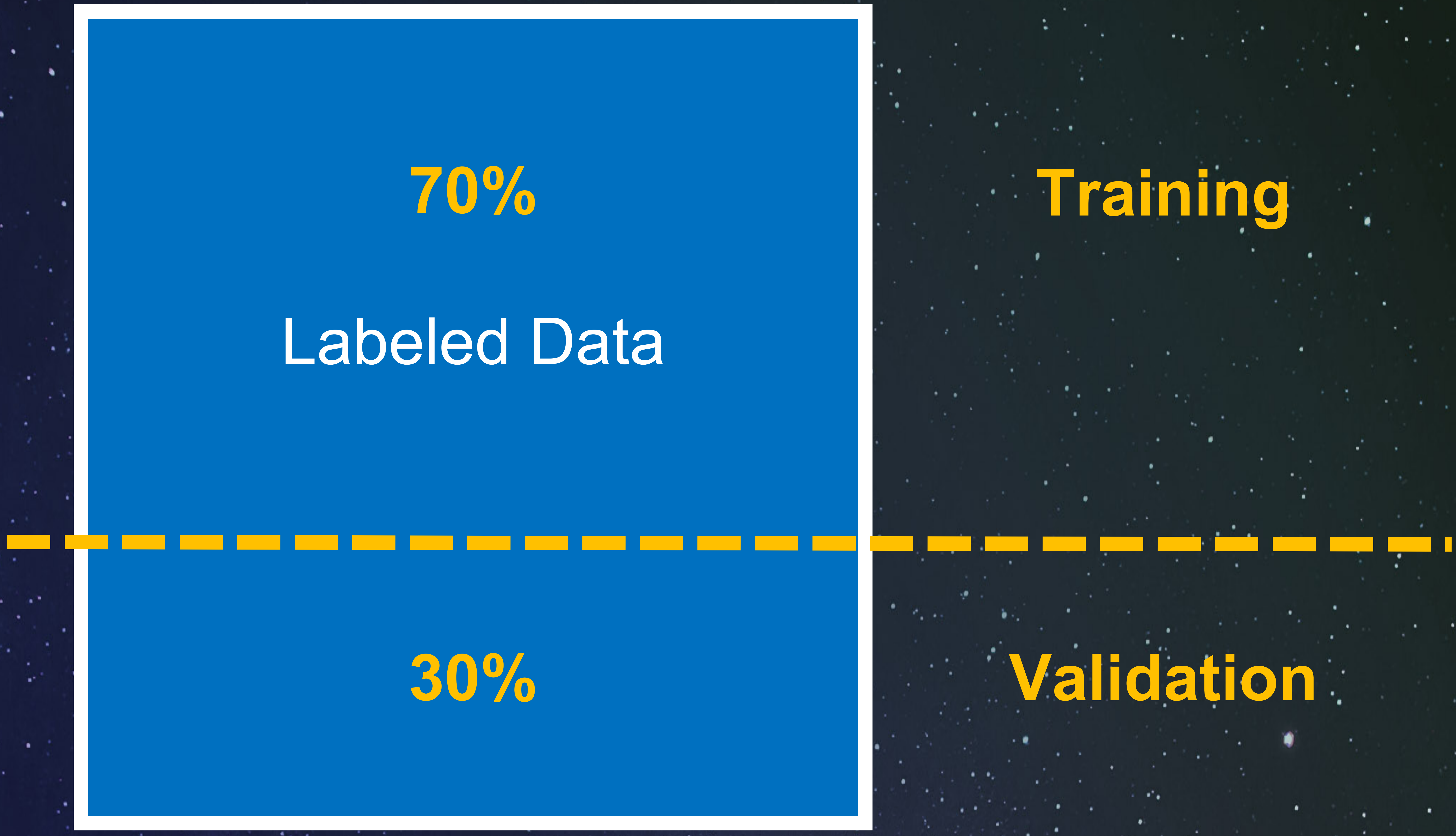
Validation

How well will this model work on new data?

Supervised Learning

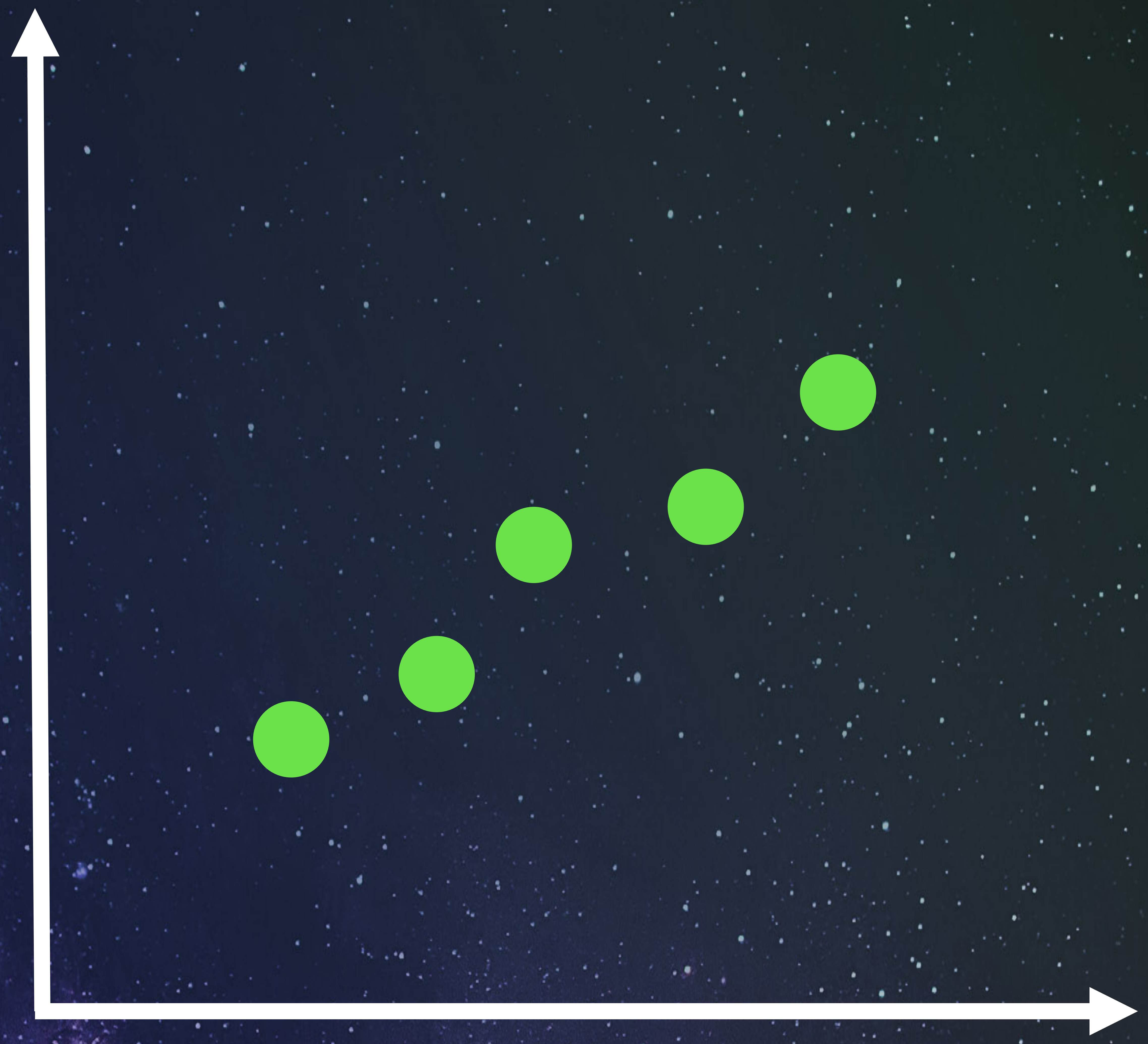
Labeled Data

Supervised Learning



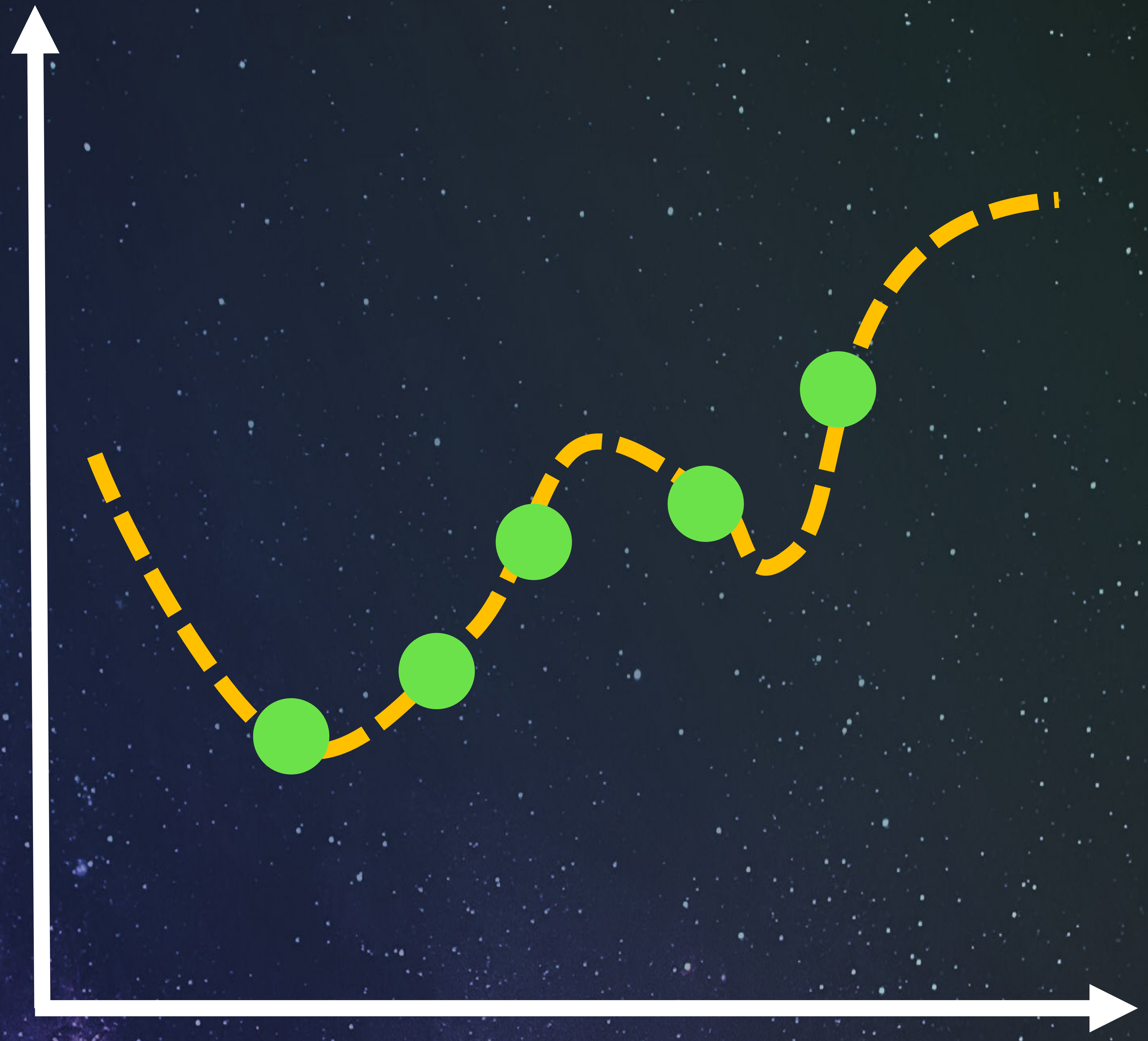
Supervised Learning

Be Careful of Overfitting



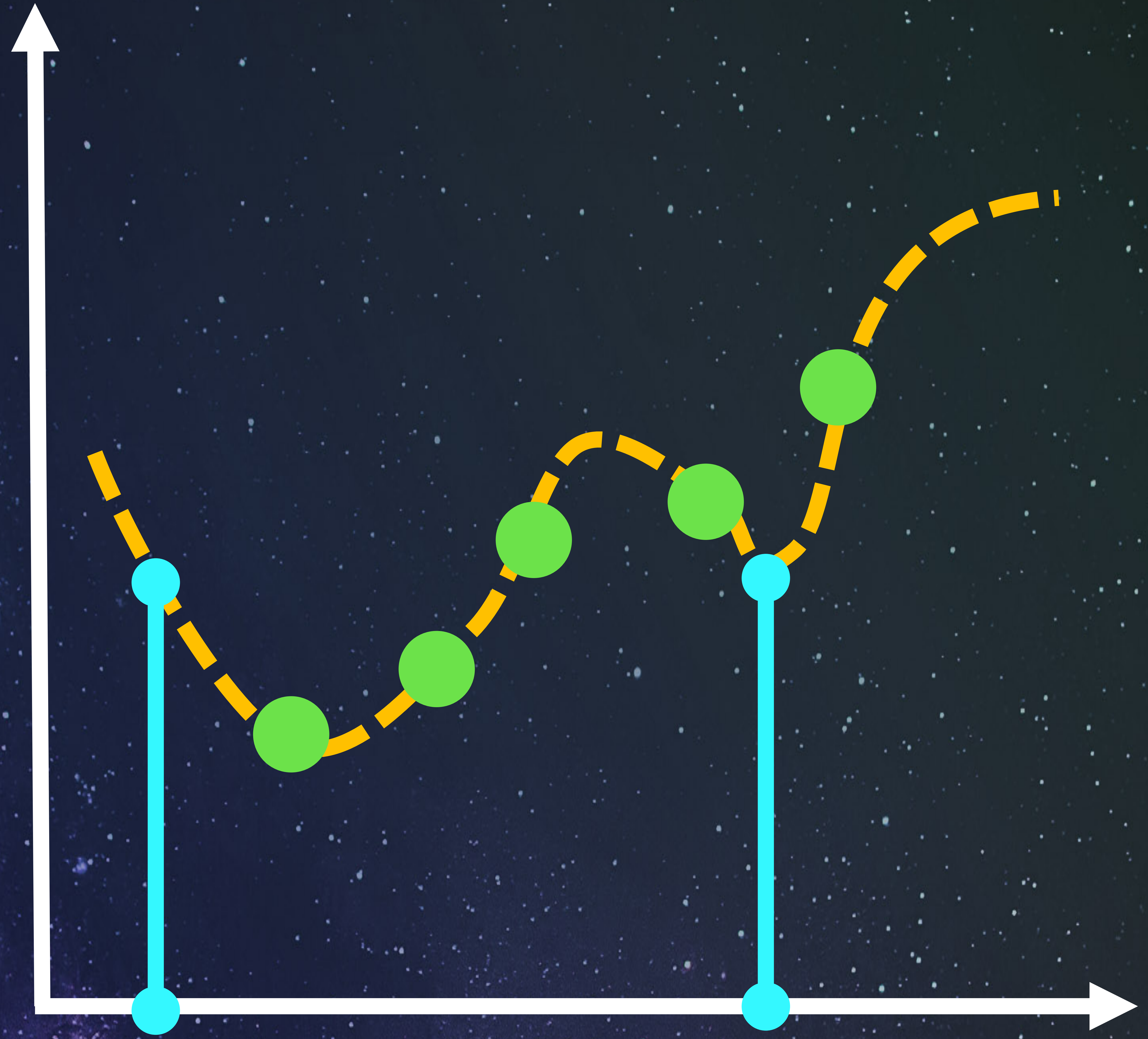
Supervised Learning

Be Careful of Overfitting



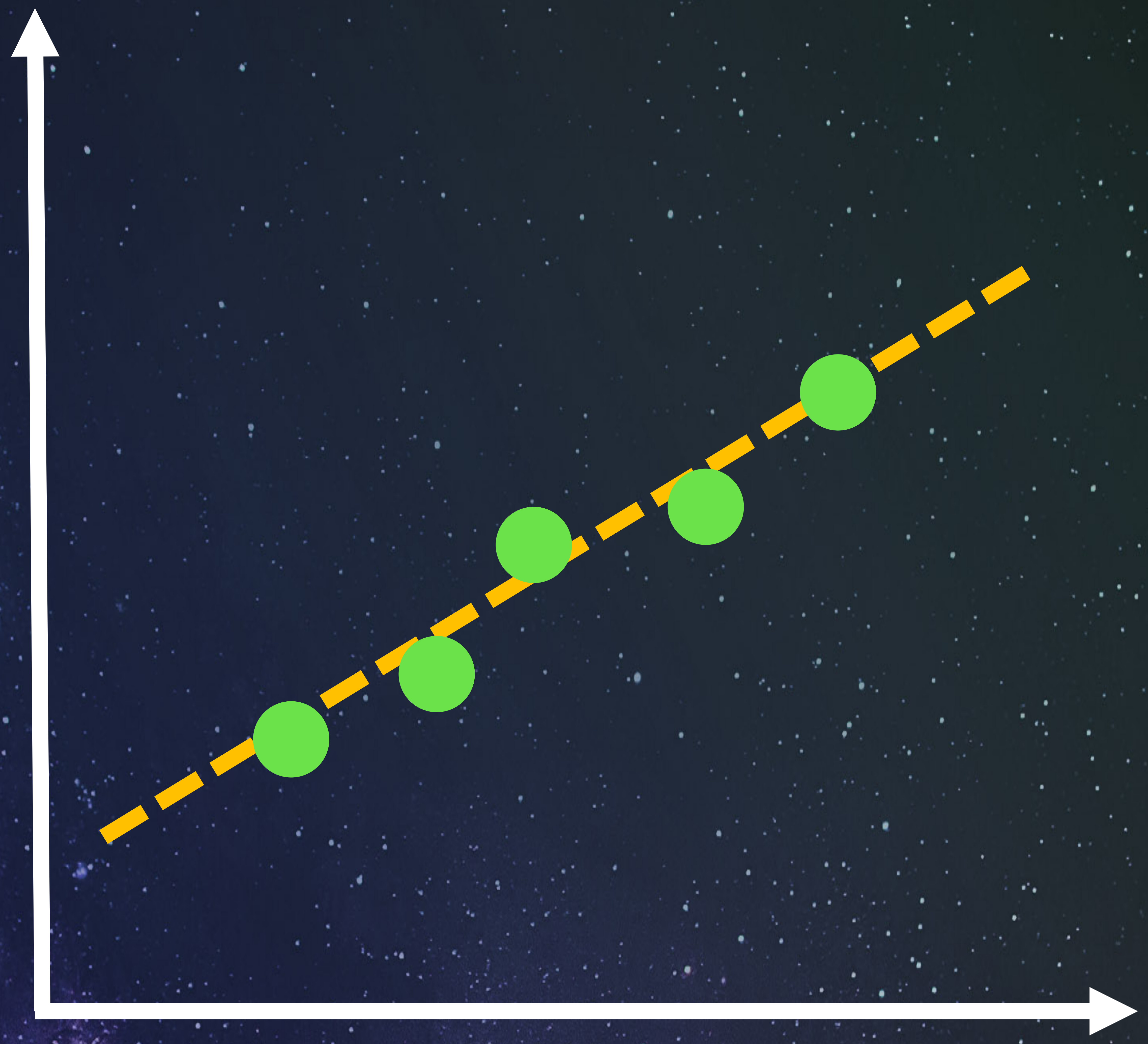
Supervised Learning

Be Careful of Overfitting

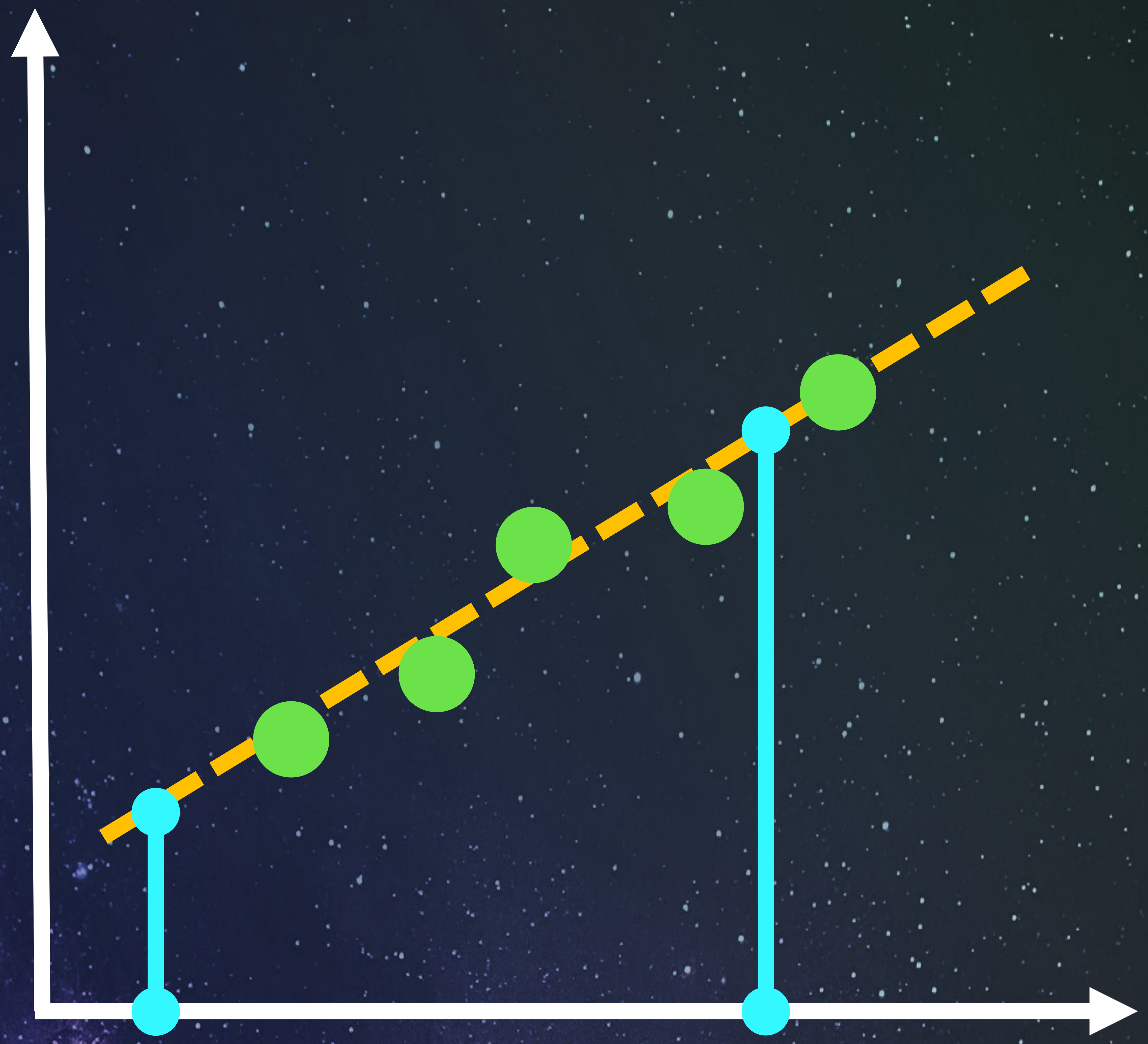


Supervised Learning

Better Fitting

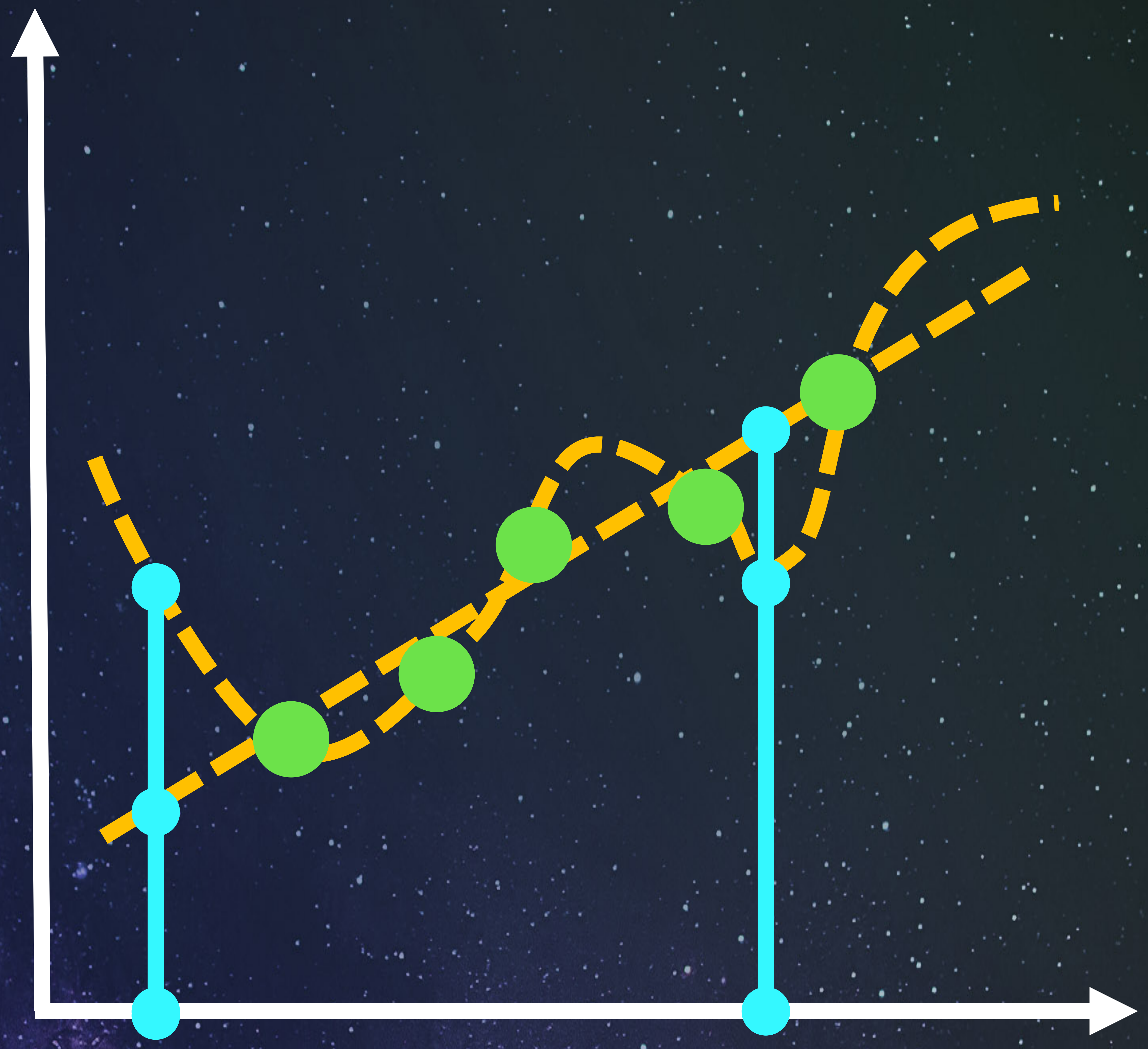


Better Fitting



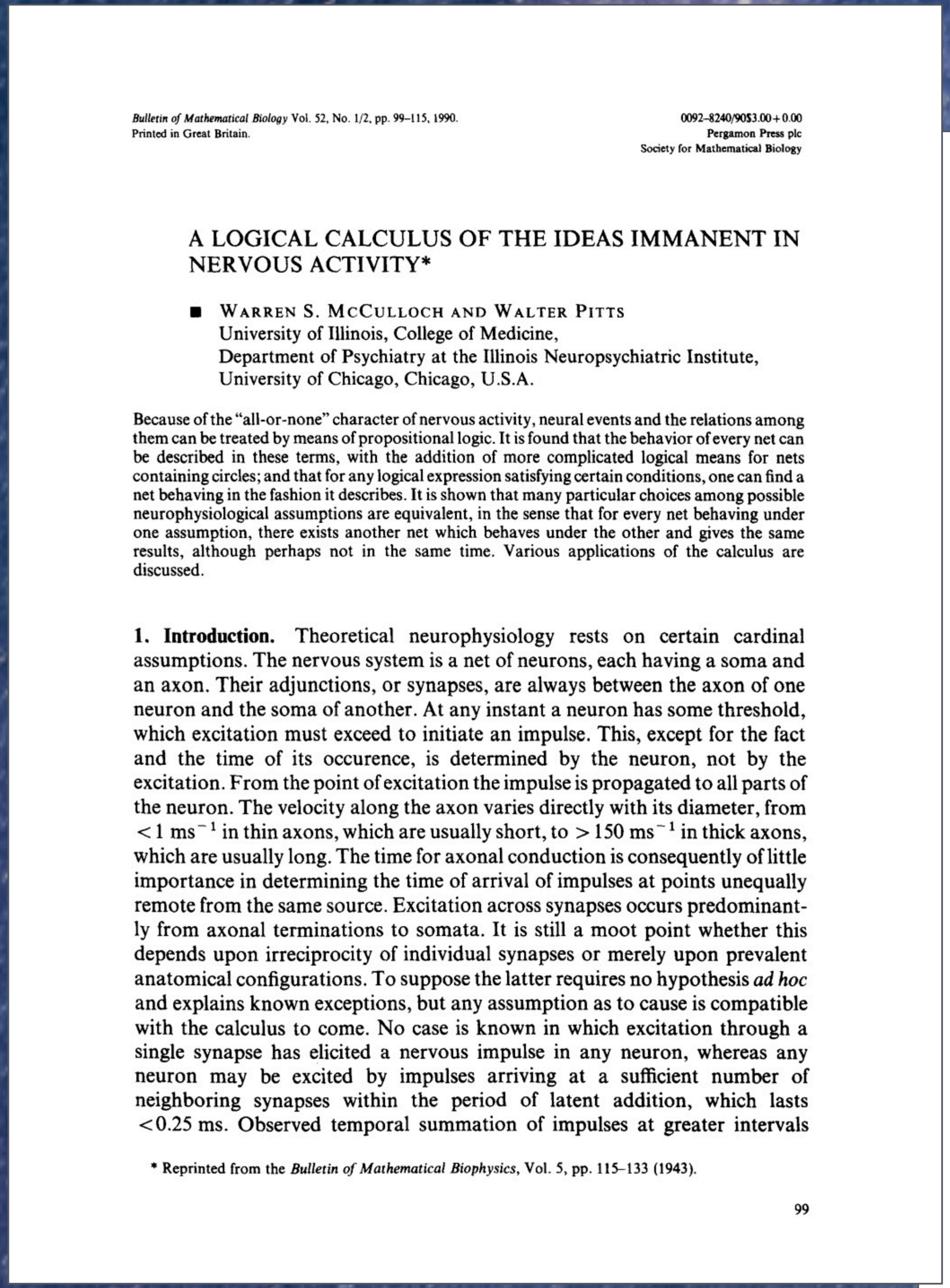
Supervised Learning

Different Models ⇒ Different Predictions



Neural Networks

1943 Warren McCulloch, Walter Pitts


Bulletin of Mathematical Biology Vol. 52, No. 1/2, pp. 99–115, 1990.
Printed in Great Britain.

0092–8240/90$3.00+0.00
Pergamon Press plc
Society for Mathematical Biology

A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY*

■ WARREN S. MCCULLOCH AND WALTER PITTS
University of Illinois, College of Medicine,
Department of Psychiatry at the Illinois Neuropsychiatric Institute,
University of Chicago, Chicago, U.S.A.

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.

1. Introduction. Theoretical neurophysiology rests on certain cardinal assumptions. The nervous system is a net of neurons, each having a soma and an axon. Their adjunctions, or synapses, are always between the axon of one neuron and the soma of another. At any instant a neuron has some threshold, which excitation must exceed to initiate an impulse. This, except for the fact and the time of its occurence, is determined by the neuron, not by the excitation. From the point of excitation the impulse is propagated to all parts of the neuron. The velocity along the axon varies directly with its diameter, from $<1\ \text{ms}^{-1}$ in thin axons, which are usually short, to $>150\ \text{ms}^{-1}$ in thick axons, which are usually long. The time for axonal conduction is consequently of little importance in determining the time of arrival of impulses at points unequally remote from the same source. Excitation across synapses occurs predominantly from axonal terminations to somata. It is still a moot point whether this depends upon irreciprocity of individual synapses or merely upon prevalent anatomical configurations. To suppose the latter requires no hypothesis *ad hoc* and explains known exceptions, but any assumption as to cause is compatible with the calculus to come. No case is known in which excitation through a single synapse has elicited a nervous impulse in any neuron, whereas any neuron may be excited by impulses arriving at a sufficient number of neighboring synapses within the period of latent addition, which lasts <0.25 ms. Observed temporal summation of impulses at greater intervals

* Reprinted from the *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115–133 (1943).

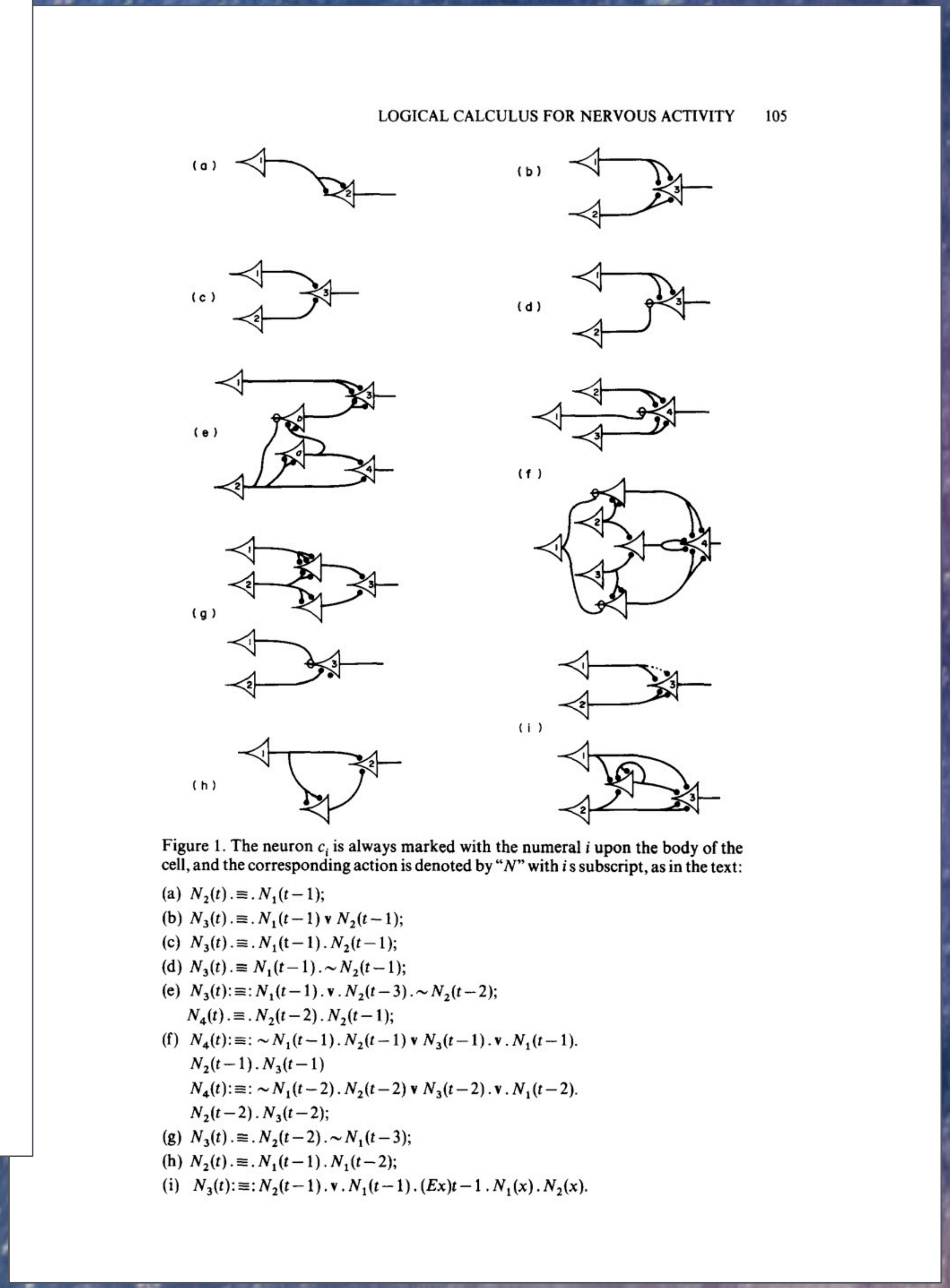
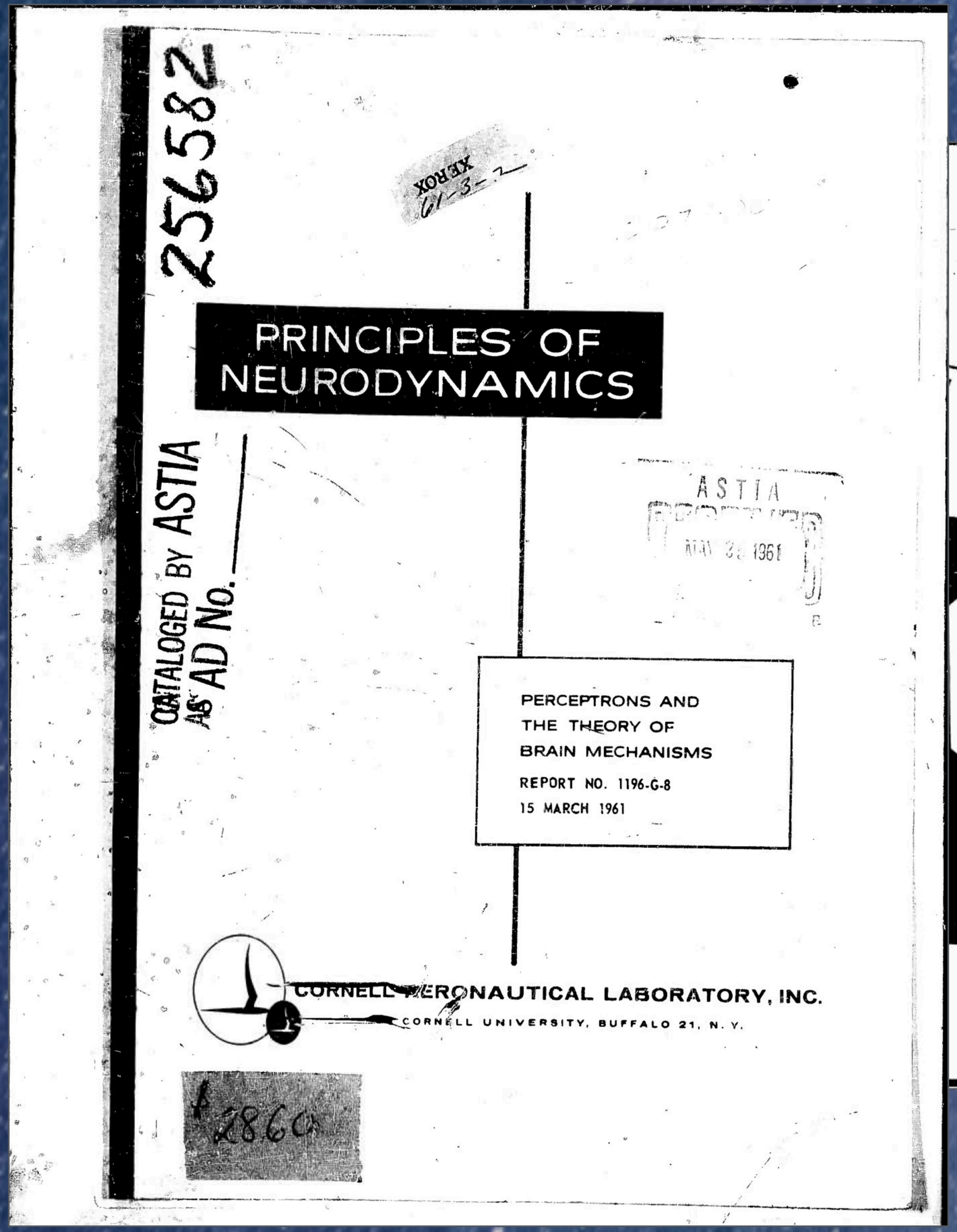


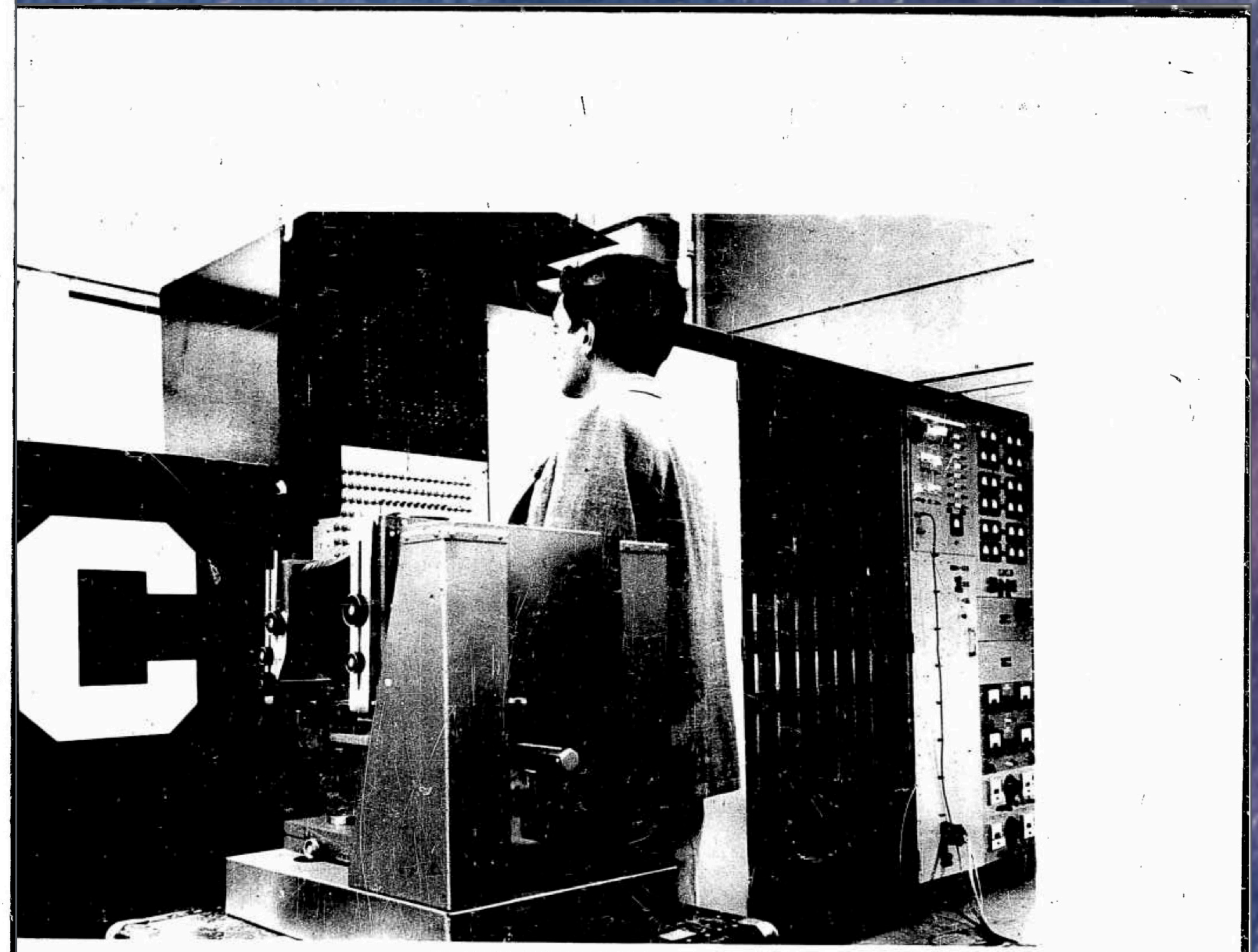
Figure 1. The neuron c_i is always marked with the numeral i upon the body of the cell, and the corresponding action is denoted by "N" with i s subscript, as in the text:

(a) $N_2(t) . \equiv . N_1(t-1)$;
(b) $N_3(t) . \equiv . N_1(t-1) \vee N_2(t-1)$;
(c) $N_3(t) . \equiv . N_1(t-1) . N_2(t-1)$;
(d) $N_3(t) . \equiv N_1(t-1) . \sim N_2(t-1)$;
(e) $N_3(t) : \equiv : N_1(t-1) . \vee . N_2(t-3) . \sim N_2(t-2)$;
$N_4(t) . \equiv . N_2(t-2) . N_2(t-1)$;
(f) $N_4(t) : \equiv : \sim N_1(t-1) . N_2(t-1) \vee N_3(t-1) . \vee . N_1(t-1).$
$N_2(t-1) . N_3(t-1)$
$N_4(t) : \equiv : \sim N_1(t-2) . N_2(t-2) \vee N_3(t-2) . \vee . N_1(t-2).$
$N_2(t-2) . N_3(t-2)$;
(g) $N_3(t) . \equiv . N_2(t-2) . \sim N_1(t-3)$;
(h) $N_2(t) . \equiv . N_1(t-1) . N_1(t-2)$;
(i) $N_3(t) : \equiv : N_2(t-1) . \vee . N_1(t-1) . (Ex)t-1 . N_1(x) . N_2(x)$.

Threshold Logic Units

1962 Frank Rosenblatt

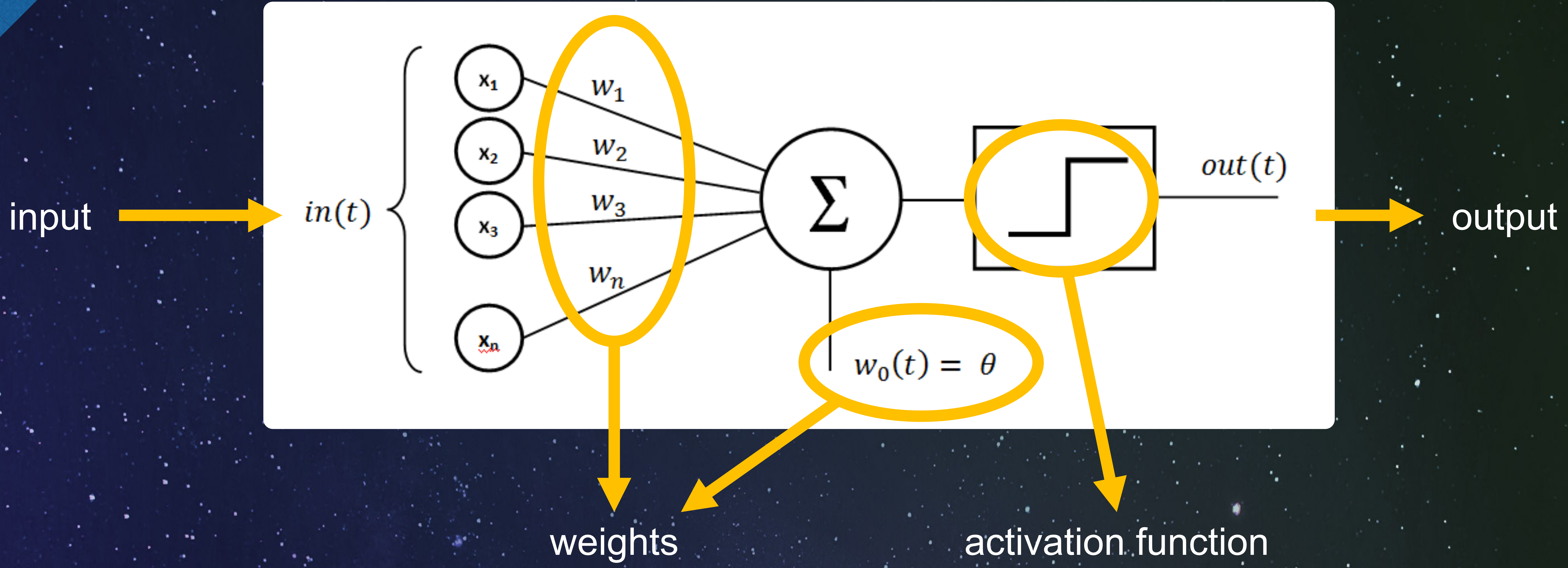




THE MARK I PERCEPTRON

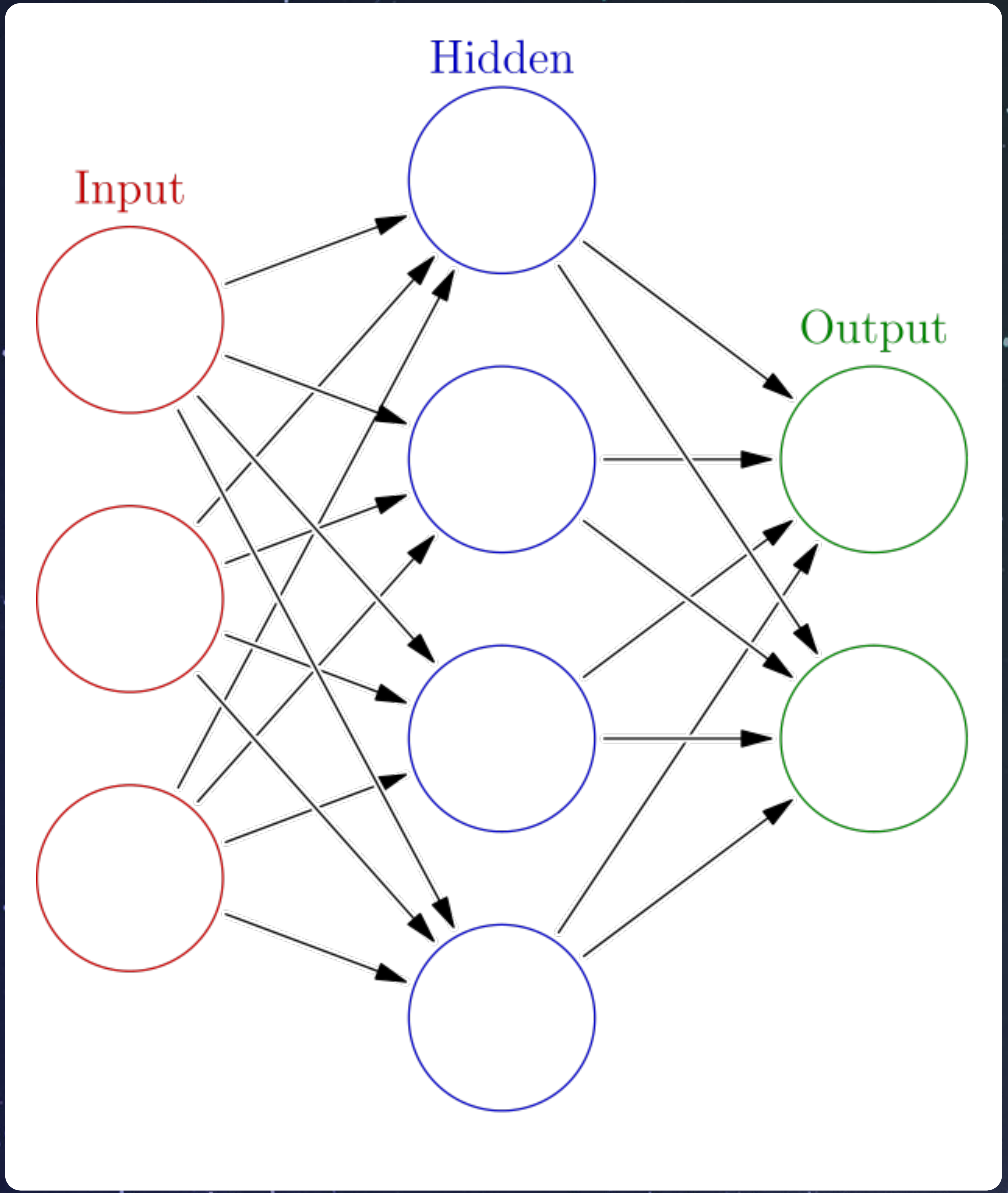
Perceptron

Perceptron



https://github.com/cdipaolo/goml/tree/master/perceptron

Neural Network



https://en.wikipedia.org/wiki/Artificial_neural_network

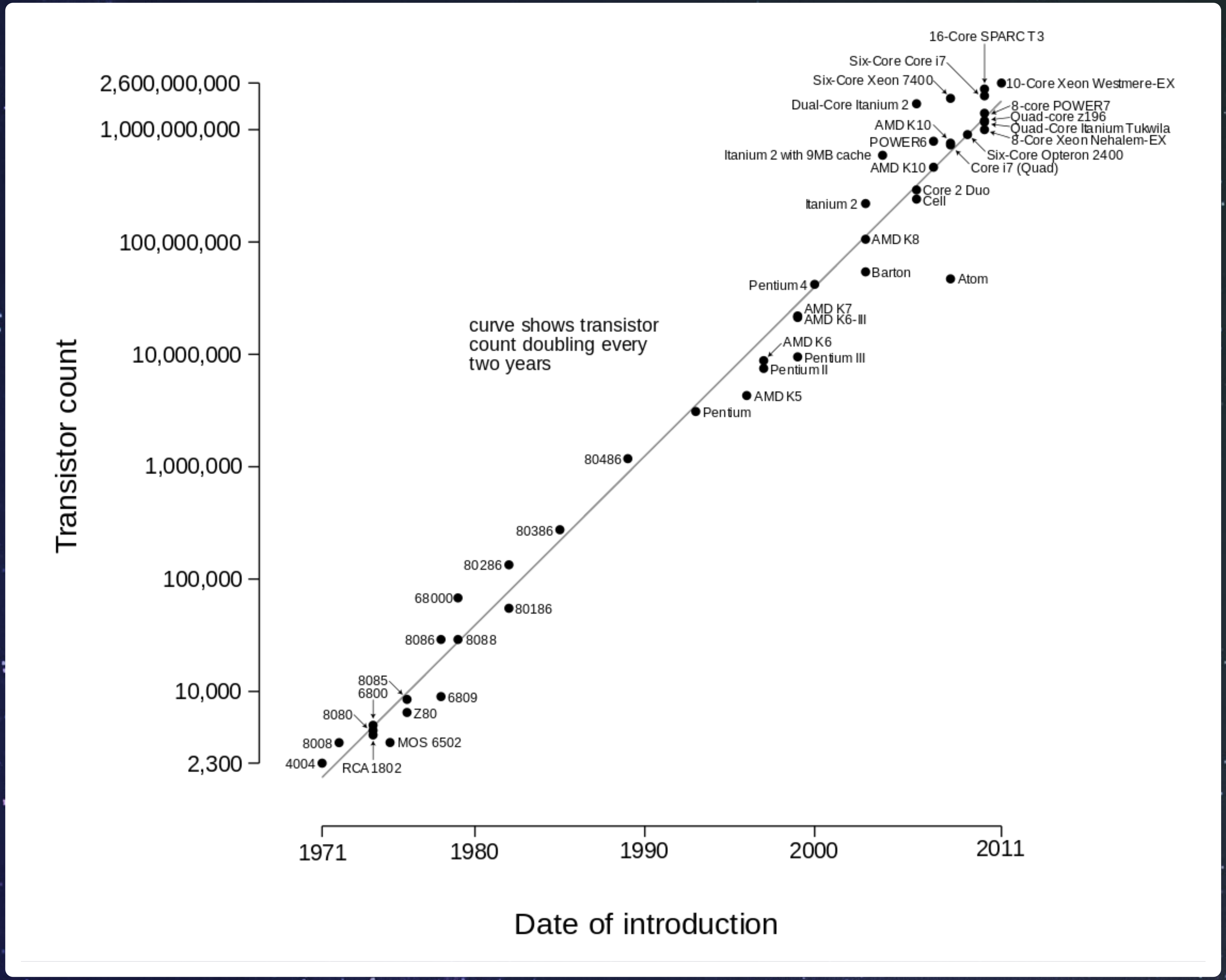
Multiple Layers

↓

Backpropagation

Moore's Law

Microprocessor Transistor Counts 1971-2011



Intel E7 CPU
4-24 cores

NVIDIA K80 GPU
2,496 cores

https://en.wikipedia.org/wiki/Moore's_law

Deep Learning

Advances in Research 1998-2009

PROC. OF THE IEEE, NOVEMBER 1998 1

Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

LeCun, Gradient-Based Learning Applied to Document Recognition,1998

LETTER Communicated by Yann Le Cun

A Fast Learning Algorithm for Deep Belief Nets

Geoffrey E. Hinton
hinton@cs.toronto.edu
Simon Osindero
osindero@cs.toronto.edu
Department of Computer Science, University of Toronto, Toronto, Canada M5S 3G4

Yee-Whye Teh
tehyw@comp.nus.edu.sg
Department of Computer Science, National University of Singapore, Singapore 117543

Neural Computation 18, 1527–1554 (2006) © 2006 Massachusetts Institute of Technology

Hinton, A Fast Learning Algorithm for Deep Belief Nets, 2006

Learning Deep Architectures for AI

Yoshua Bengio
Dept. IRO, Université de Montréal
C.P. 6128, Montreal, Qc, H3C 3J7, Canada
Yoshua.Bengio@umontreal.ca
http://www.iro.umontreal.ca/~bengioy

Technical Report 1312

Bengio, Learning Deep Architectures for AI, 2009

Deep Learning

Image Processing

Image Processing

Convolution Matrix

Image Processing

Convolution Matrix

Image Processing

Convolution Matrix

Identity

0	0	0
0	1	0
0	0	0

Image Processing

Convolution Matrix

Left Edges

1	0	-1
2	0	-2
1	0	-1

Image Processing

Convolution Matrix

Right Edges

-1	0	1
-2	0	2
-1	0	1

Image Processing

Convolution Matrix

Top Edges

1	2	1
0	0	0
-1	-2	-1

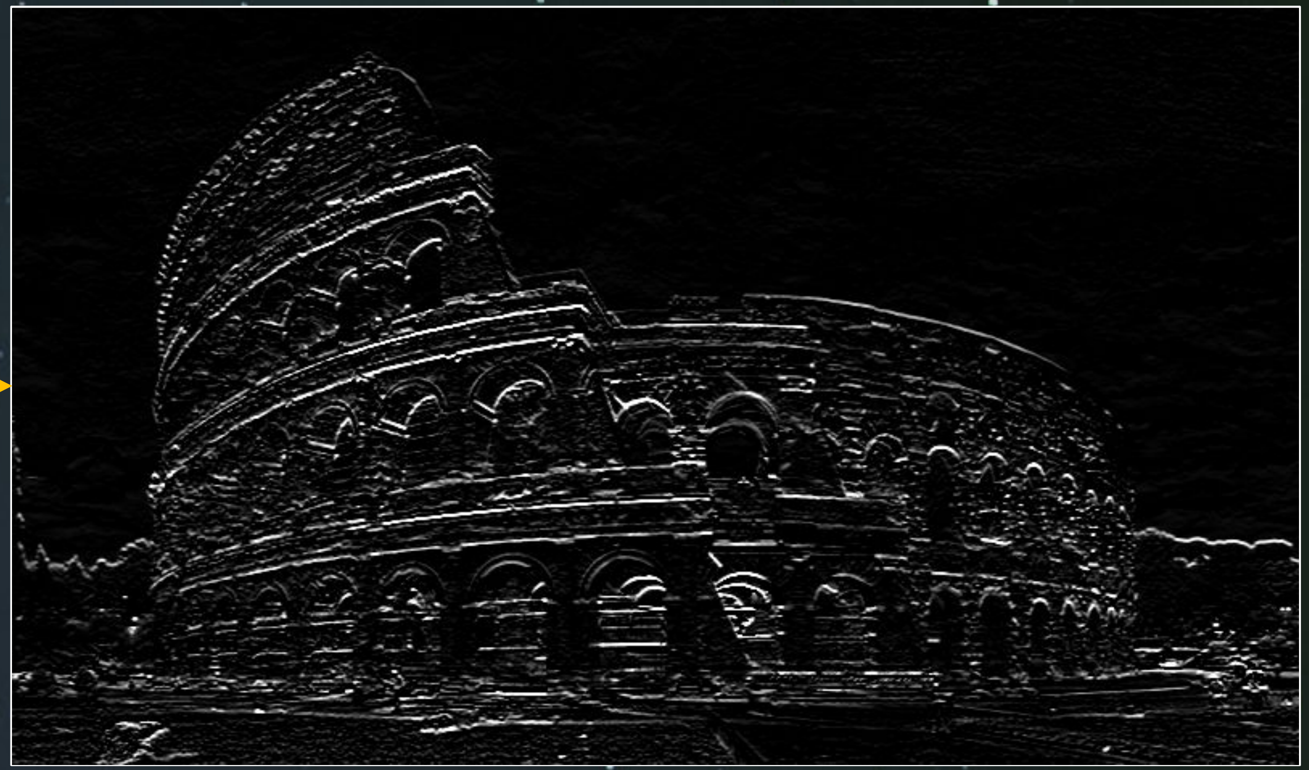


Image Processing

Convolution Matrix

Bottom Edges

-1	-2	-1
0	0	0
1	2	1

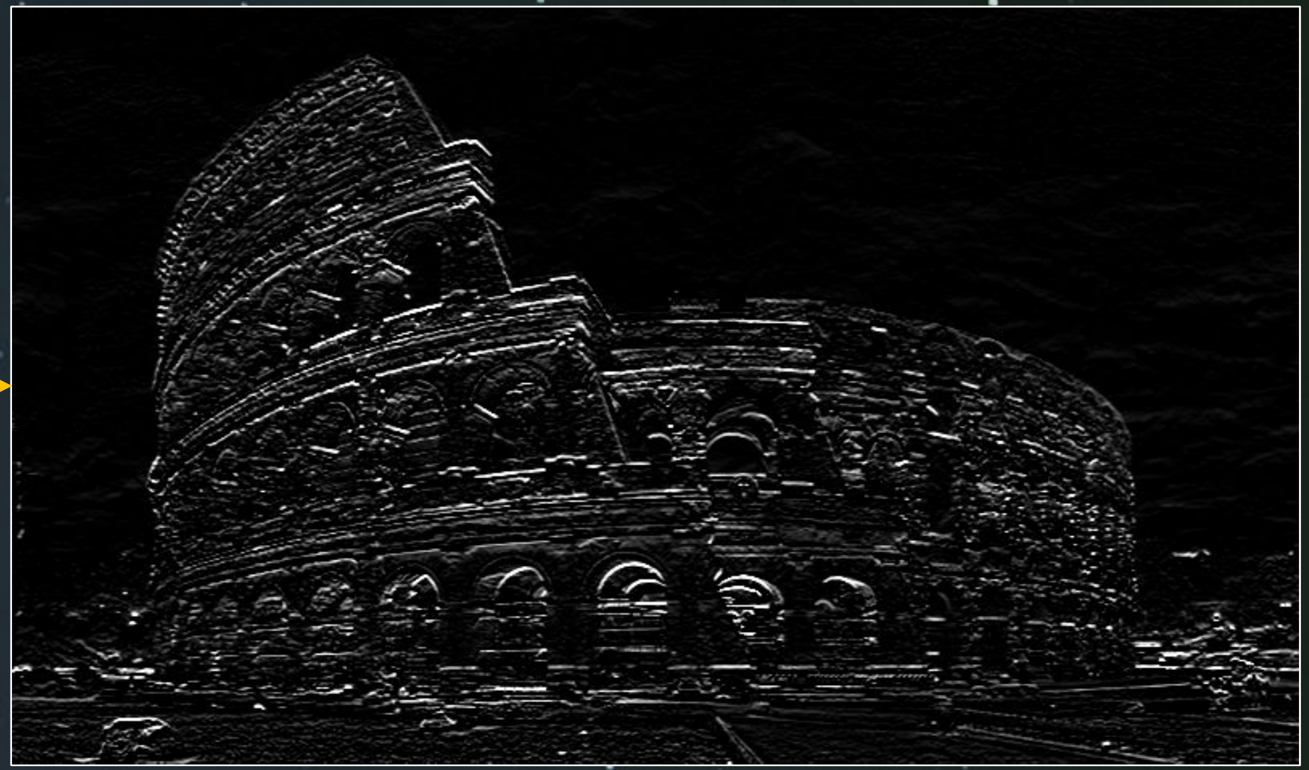


Image Processing

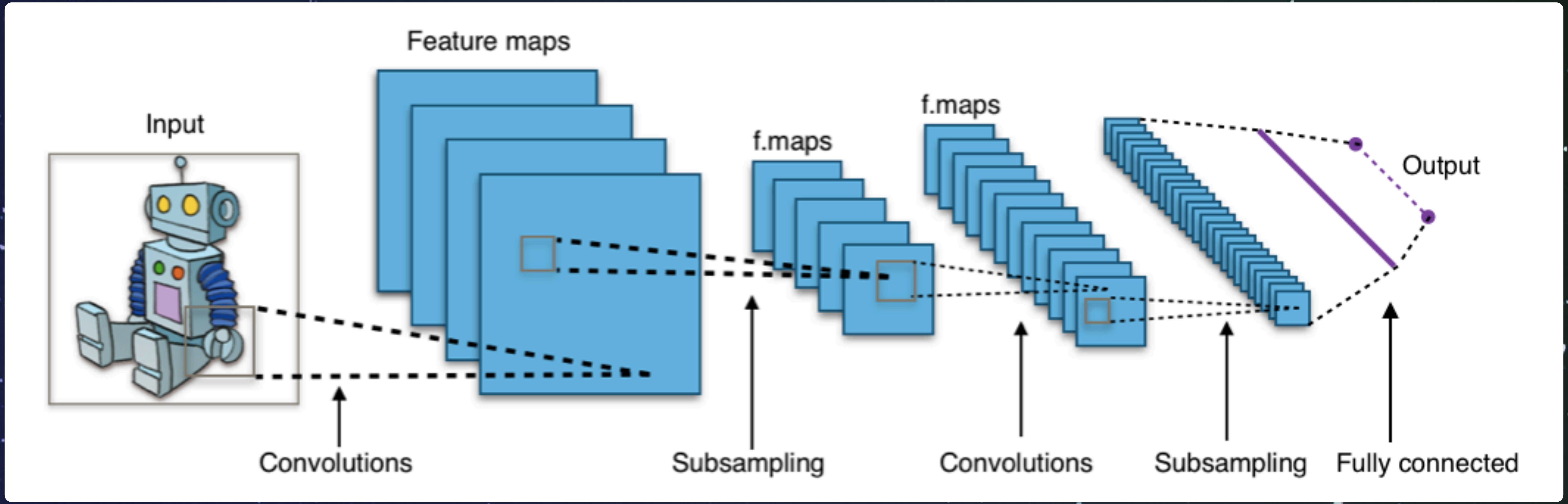
Convolution Matrix

Random Values

0.6	-0.6	1.2
-1.4	1.2	-1.6
0.8	-1.4	1.6

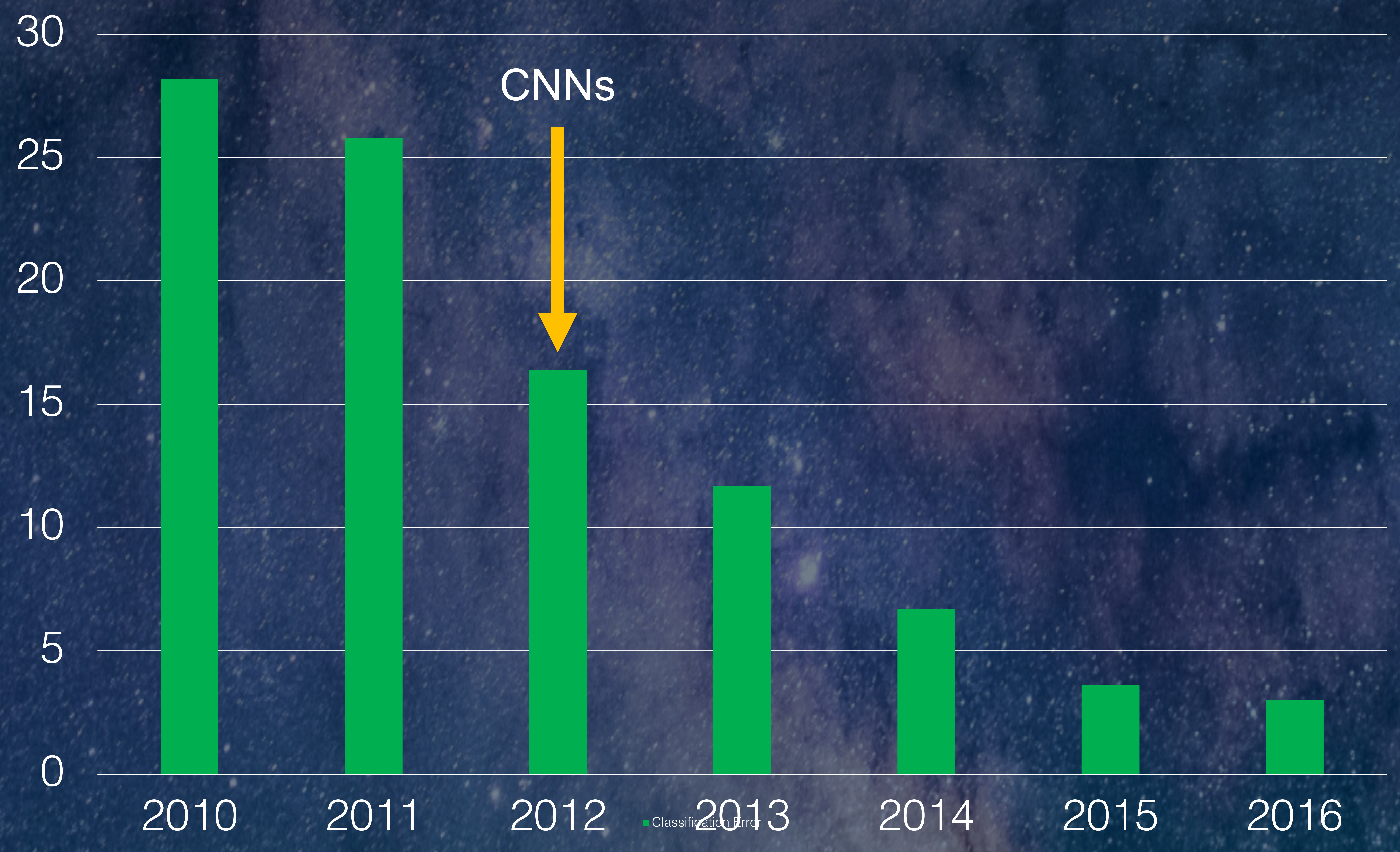
CNNs

Convolutional Neural Networks (CNNs)

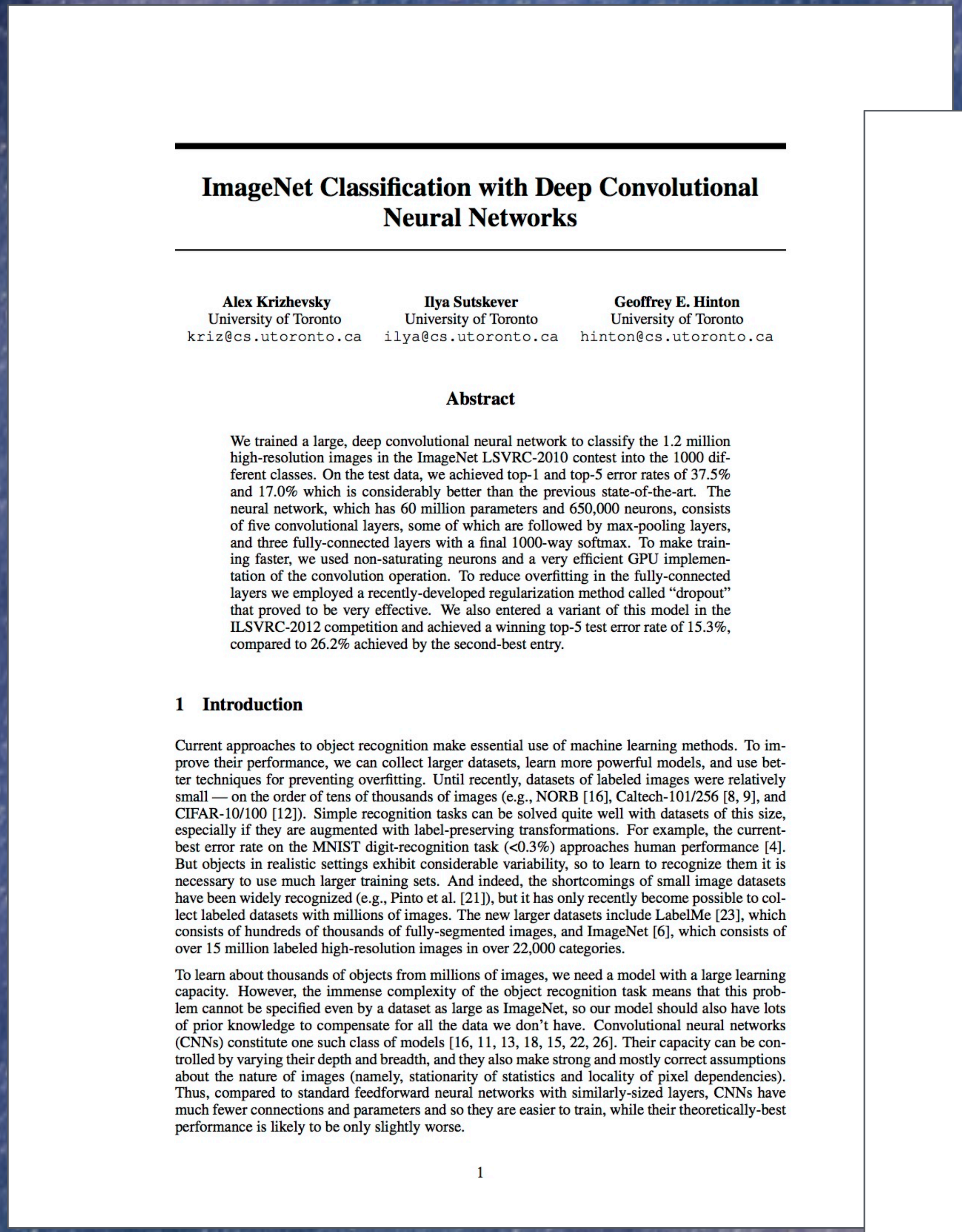


https://en.wikipedia.org/wiki/Convolutional_neural_network

ImageNet Classification Error Over Time



2012 ImageNet Classification with Deep Convolutional Neural Networks



ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

1 Introduction

Current approaches to object recognition make essential use of machine learning methods. To improve their performance, we can collect larger datasets, learn more powerful models, and use better techniques for preventing overfitting. Until recently, datasets of labeled images were relatively small — on the order of tens of thousands of images (e.g., NORB [16], Caltech-101/256 [8, 9], and CIFAR-10/100 [12]). Simple recognition tasks can be solved quite well with datasets of this size, especially if they are augmented with label-preserving transformations. For example, the current-best error rate on the MNIST digit-recognition task (<0.3%) approaches human performance [4]. But objects in realistic settings exhibit considerable variability, so to learn to recognize them it is necessary to use much larger training sets. And indeed, the shortcomings of small image datasets have been widely recognized (e.g., Pinto et al. [21]), but it has only recently become possible to collect labeled datasets with millions of images. The new larger datasets include LabelMe [23], which consists of hundreds of thousands of fully-segmented images, and ImageNet [6], which consists of over 15 million labeled high-resolution images in over 22,000 categories.

To learn about thousands of objects from millions of images, we need a model with a large learning capacity. However, the immense complexity of the object recognition task means that this problem cannot be specified even by a dataset as large as ImageNet, so our model should also have lots of prior knowledge to compensate for all the data we don't have. Convolutional neural networks (CNNs) constitute one such class of models [16, 11, 13, 18, 15, 22, 26]. Their capacity can be controlled by varying their depth and breadth, and they also make strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies). Thus, compared to standard feedforward neural networks with similarly-sized layers, CNNs have much fewer connections and parameters and so they are easier to train, while their theoretically-best performance is likely to be only slightly worse.

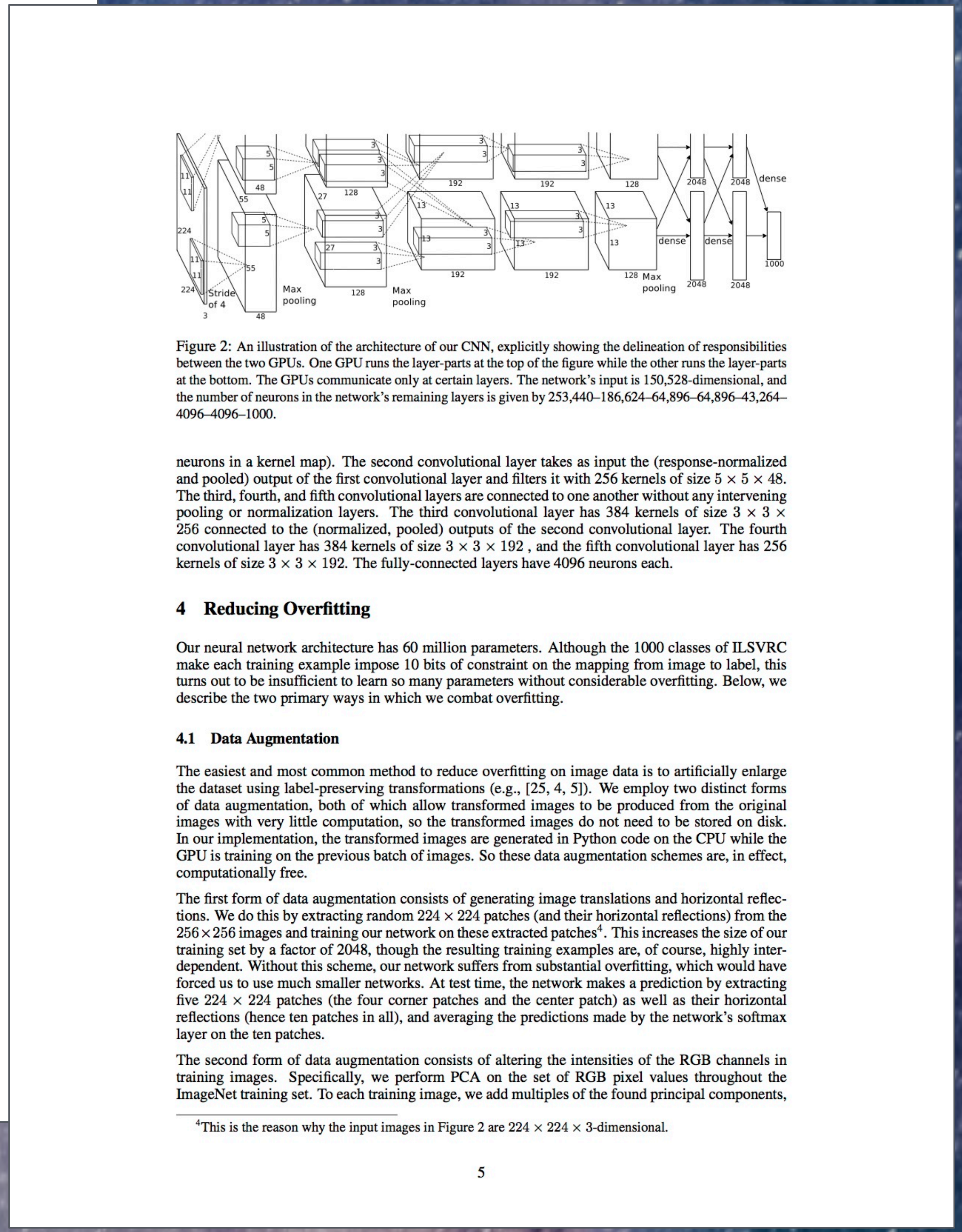


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

neurons in a kernel map). The second convolutional layer takes as input the (response-normalized and pooled) output of the first convolutional layer and filters it with 256 kernels of size $5 \times 5 \times 48$. The third, fourth, and fifth convolutional layers are connected to one another without any intervening pooling or normalization layers. The third convolutional layer has 384 kernels of size $3 \times 3 \times 256$ connected to the (normalized, pooled) outputs of the second convolutional layer. The fourth convolutional layer has 384 kernels of size $3 \times 3 \times 192$, and the fifth convolutional layer has 256 kernels of size $3 \times 3 \times 192$. The fully-connected layers have 4096 neurons each.

4 Reducing Overfitting

Our neural network architecture has 60 million parameters. Although the 1000 classes of ILSVRC make each training example impose 10 bits of constraint on the mapping from image to label, this turns out to be insufficient to learn so many parameters without considerable overfitting. Below, we describe the two primary ways in which we combat overfitting.

4.1 Data Augmentation

The easiest and most common method to reduce overfitting on image data is to artificially enlarge the dataset using label-preserving transformations (e.g., [25, 4, 5]). We employ two distinct forms of data augmentation, both of which allow transformed images to be produced from the original images with very little computation, so the transformed images do not need to be stored on disk. In our implementation, the transformed images are generated in Python code on the CPU while the GPU is training on the previous batch of images. So these data augmentation schemes are, in effect, computationally free.

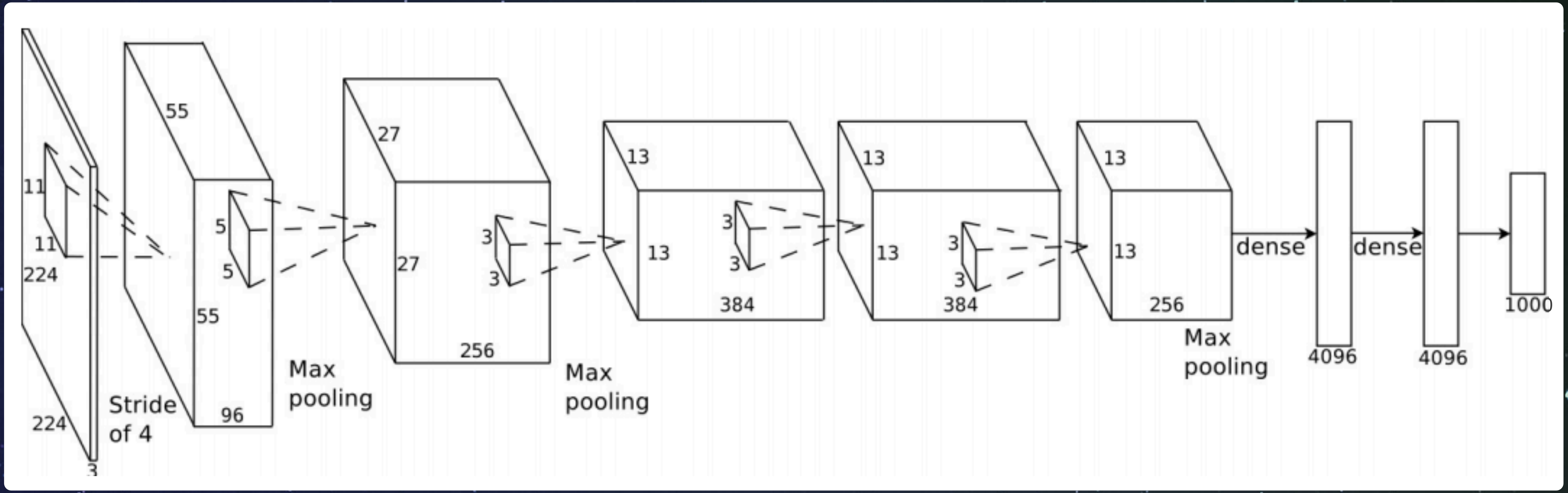
The first form of data augmentation consists of generating image translations and horizontal reflections. We do this by extracting random 224×224 patches (and their horizontal reflections) from the 256×256 images and training our network on these extracted patches[4]. This increases the size of our training set by a factor of 2048, though the resulting training examples are, of course, highly interdependent. Without this scheme, our network suffers from substantial overfitting, which would have forced us to use much smaller networks. At test time, the network makes a prediction by extracting five 224×224 patches (the four corner patches and the center patch) as well as their horizontal reflections (hence ten patches in all), and averaging the predictions made by the network's softmax layer on the ten patches.

The second form of data augmentation consists of altering the intensities of the RGB channels in training images. Specifically, we perform PCA on the set of RGB pixel values throughout the ImageNet training set. To each training image, we add multiples of the found principal components,

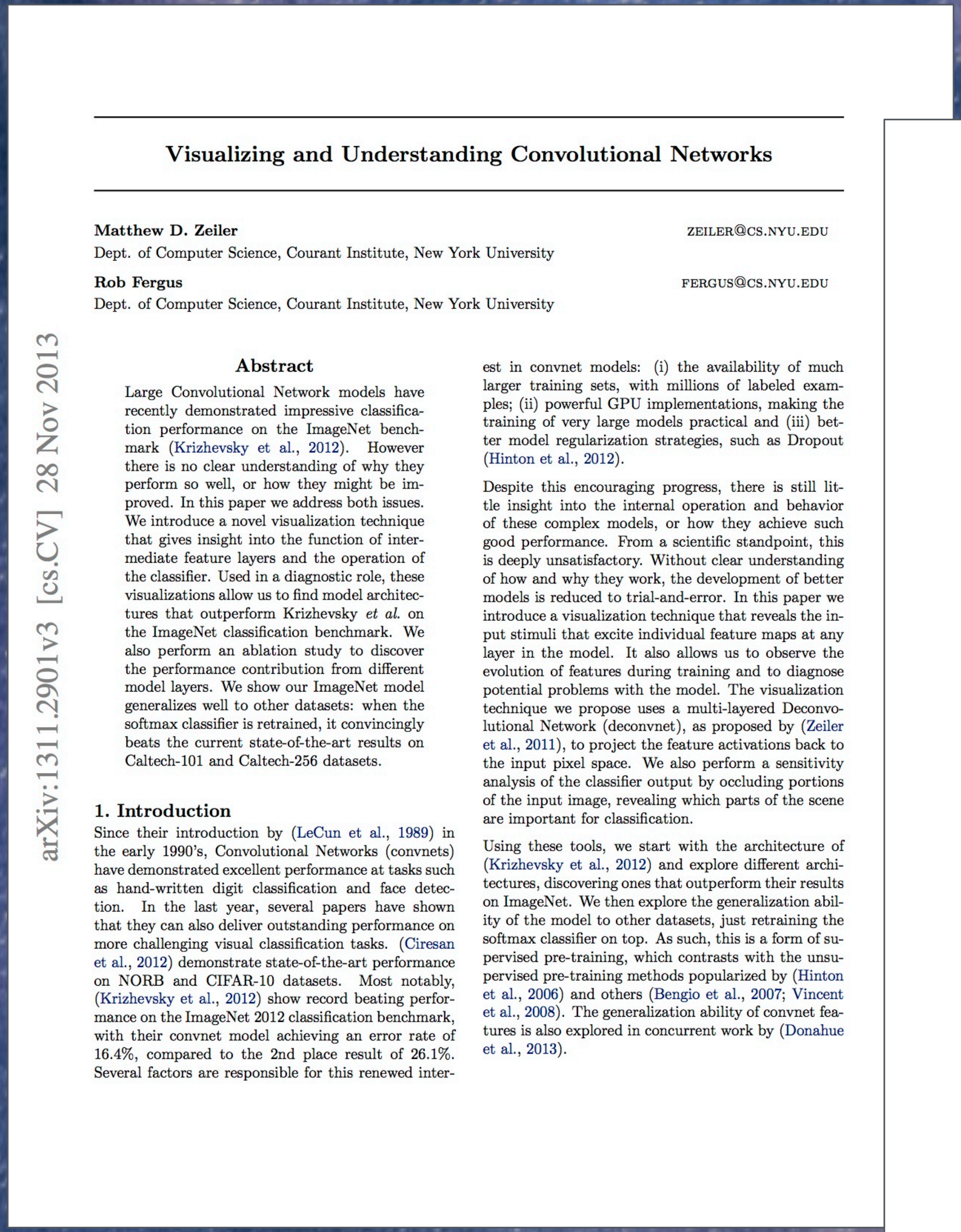
[4] This is the reason why the input images in Figure 2 are $224 \times 224 \times 3$-dimensional.

CNNs

SuperVision: 8 layers, 60M parameters



2013 Visualizing and Understanding Convolutional Networks



arXiv:1311.2901v3 [cs.CV] 28 Nov 2013

Visualizing and Understanding Convolutional Networks

Matthew D. Zeiler ZEILER@CS.NYU.EDU
Dept. of Computer Science, Courant Institute, New York University

Rob Fergus FERGUS@CS.NYU.EDU
Dept. of Computer Science, Courant Institute, New York University

Abstract

Large Convolutional Network models have recently demonstrated impressive classification performance on the ImageNet benchmark (Krizhevsky et al., 2012). However there is no clear understanding of why they perform so well, or how they might be improved. In this paper we address both issues. We introduce a novel visualization technique that gives insight into the function of intermediate feature layers and the operation of the classifier. Used in a diagnostic role, these visualizations allow us to find model architectures that outperform Krizhevsky *et al.* on the ImageNet classification benchmark. We also perform an ablation study to discover the performance contribution from different model layers. We show our ImageNet model generalizes well to other datasets: when the softmax classifier is retrained, it convincingly beats the current state-of-the-art results on Caltech-101 and Caltech-256 datasets.

1. Introduction

Since their introduction by (LeCun et al., 1989) in the early 1990's, Convolutional Networks (convnets) have demonstrated excellent performance at tasks such as hand-written digit classification and face detection. In the last year, several papers have shown that they can also deliver outstanding performance on more challenging visual classification tasks. (Ciresan et al., 2012) demonstrate state-of-the-art performance on NORB and CIFAR-10 datasets. Most notably, (Krizhevsky et al., 2012) show record beating performance on the ImageNet 2012 classification benchmark, with their convnet model achieving an error rate of 16.4%, compared to the 2nd place result of 26.1%. Several factors are responsible for this renewed interest in convnet models: (i) the availability of much larger training sets, with millions of labeled examples; (ii) powerful GPU implementations, making the training of very large models practical and (iii) better model regularization strategies, such as Dropout (Hinton et al., 2012).

Despite this encouraging progress, there is still little insight into the internal operation and behavior of these complex models, or how they achieve such good performance. From a scientific standpoint, this is deeply unsatisfactory. Without clear understanding of how and why they work, the development of better models is reduced to trial-and-error. In this paper we introduce a visualization technique that reveals the input stimuli that excite individual feature maps at any layer in the model. It also allows us to observe the evolution of features during training and to diagnose potential problems with the model. The visualization technique we propose uses a multi-layered Deconvolutional Network (deconvnet), as proposed by (Zeiler et al., 2011), to project the feature activations back to the input pixel space. We also perform a sensitivity analysis of the classifier output by occluding portions of the input image, revealing which parts of the scene are important for classification.

Using these tools, we start with the architecture of (Krizhevsky et al., 2012) and explore different architectures, discovering ones that outperform their results on ImageNet. We then explore the generalization ability of the model to other datasets, just retraining the softmax classifier on top. As such, this is a form of supervised pre-training, which contrasts with the unsupervised pre-training methods popularized by (Hinton et al., 2006) and others (Bengio et al., 2007; Vincent et al., 2008). The generalization ability of convnet features is also explored in concurrent work by (Donahue et al., 2013).

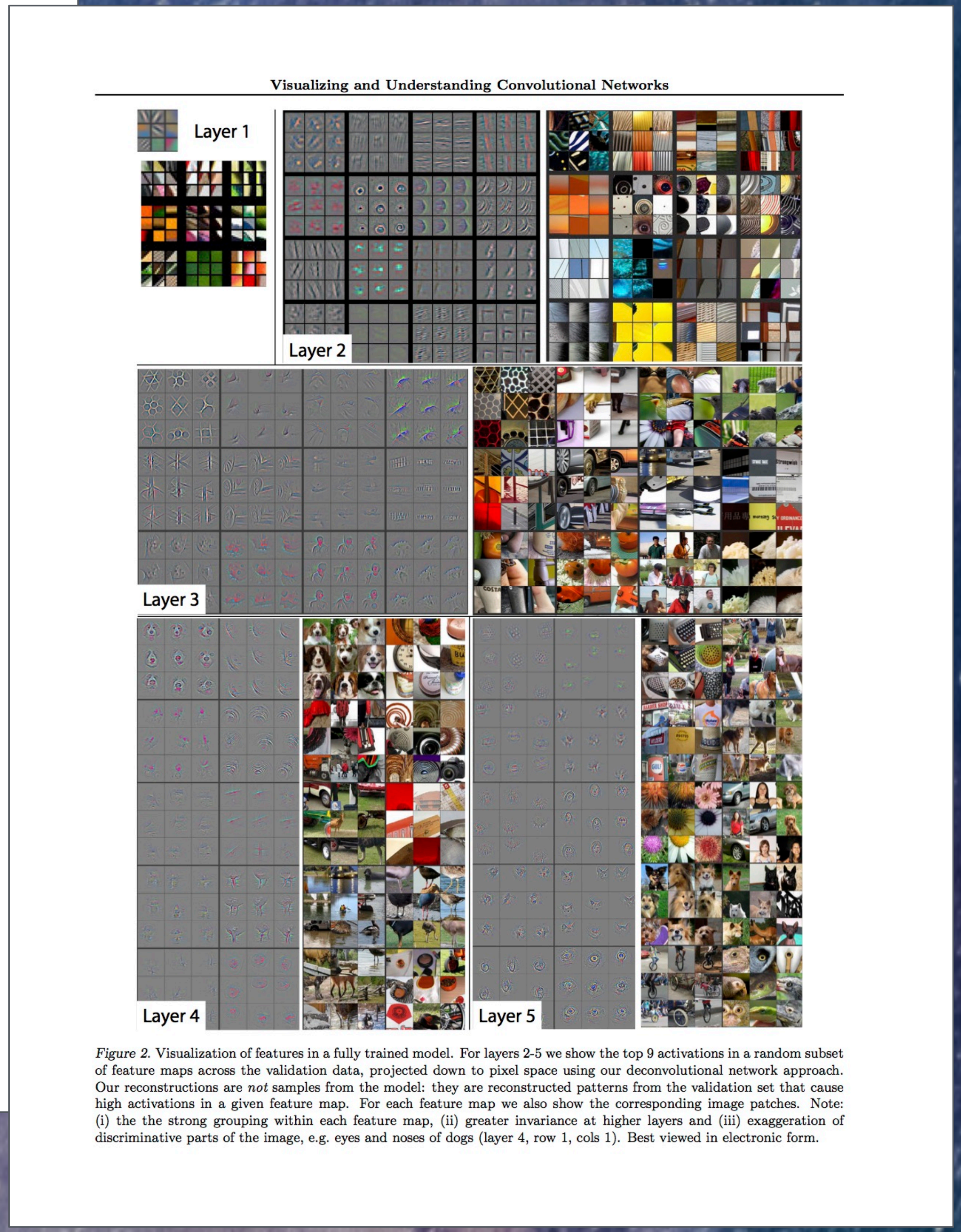
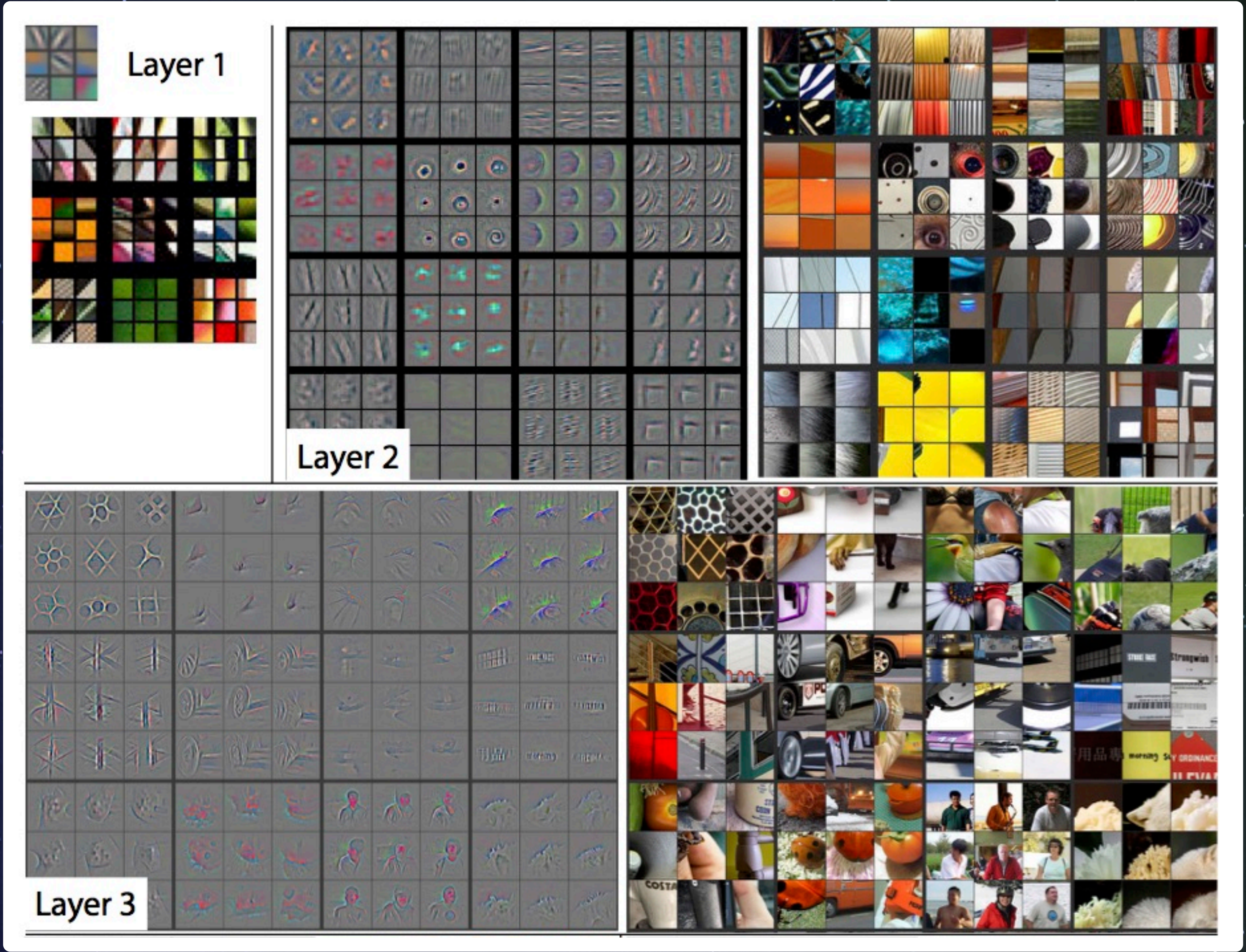
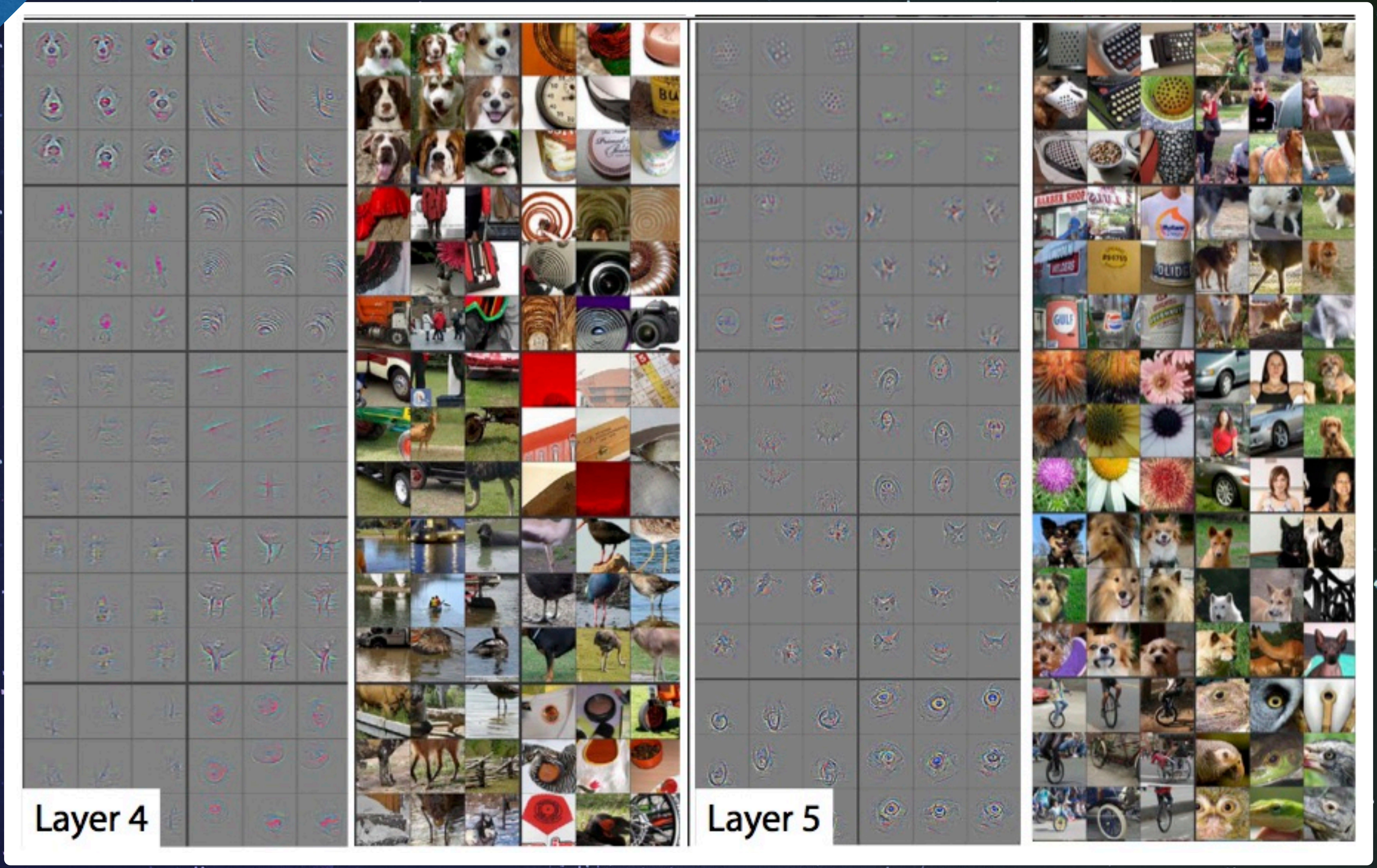


Figure 2. Visualization of features in a fully trained model. For layers 2-5 we show the top 9 activations in a random subset of feature maps across the validation data, projected down to pixel space using our deconvolutional network approach. Our reconstructions are *not* samples from the model: they are reconstructed patterns from the validation set that cause high activations in a given feature map. For each feature map we also show the corresponding image patches. Note: (i) the the strong grouping within each feature map, (ii) greater invariance at higher layers and (iii) exaggeration of discriminative parts of the image, e.g. eyes and noses of dogs (layer 4, row 1, cols 1). Best viewed in electronic form.

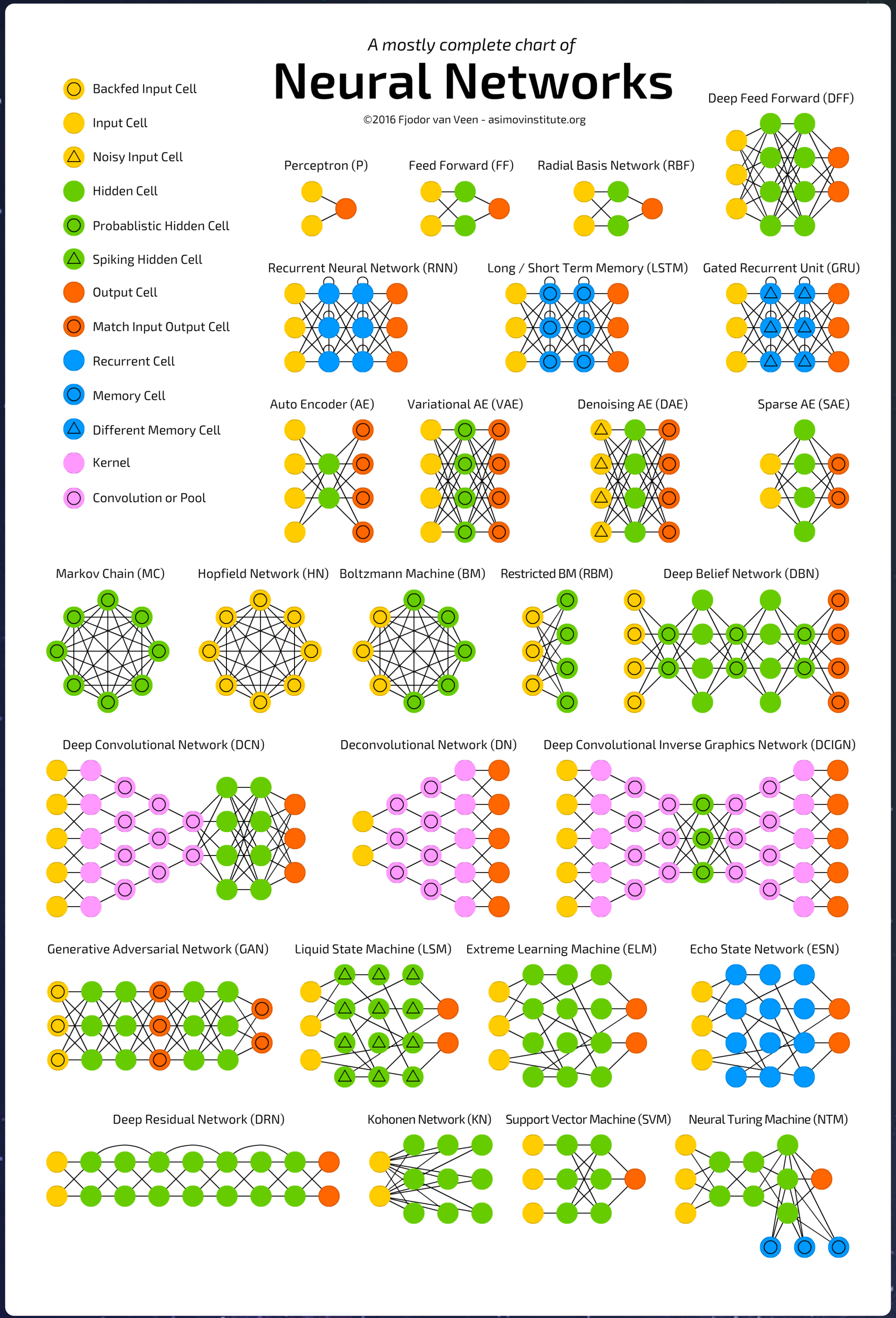
CNNs



CNNs



Neural Network Architectures



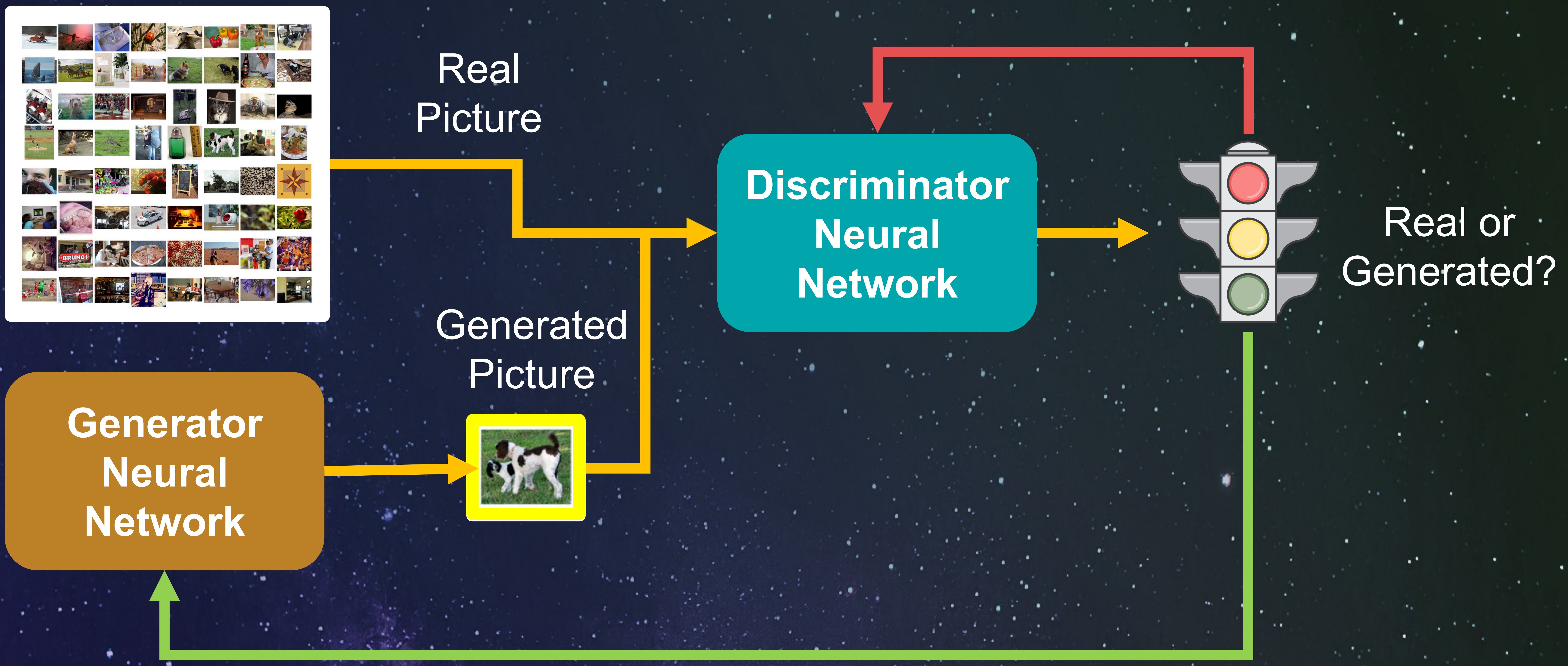
Lots of Parameters

Network Architectures defined by Hyperparameters

Dropout Layers for Regularization

http://www.asimovinstitute.org/neural-network-zoo/

Generative Adversarial Networks (GANs)



Unsupervised Learning

Generative Adversarial Networks (GANs)

arXiv:1511.06434v2 [cs.LG] 7 Jan 2016

Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

Alec Radford & Luke Metz
indico Research
Boston, MA
{alec,luke}@indico.io

Soumith Chintala
Facebook AI Research
New York, NY
soumith@fb.com

Abstract

In recent years, supervised learning with convolutional networks (CNNs) has seen huge adoption in computer vision applications. Comparatively, unsupervised learning with CNNs has received less attention. In this work we hope to help bridge the gap between the success of CNNs for supervised learning and unsupervised learning. We introduce a class of CNNs called deep convolutional generative adversarial networks (DCGANs), that have certain architectural constraints, and demonstrate that they are a strong candidate for unsupervised learning. Training on various image datasets, we show convincing evidence that our deep convolutional adversarial pair learns a hierarchy of representations from object parts to scenes in both the generator and discriminator. Additionally, we use the learned features for novel tasks - demonstrating their applicability as general image representations.

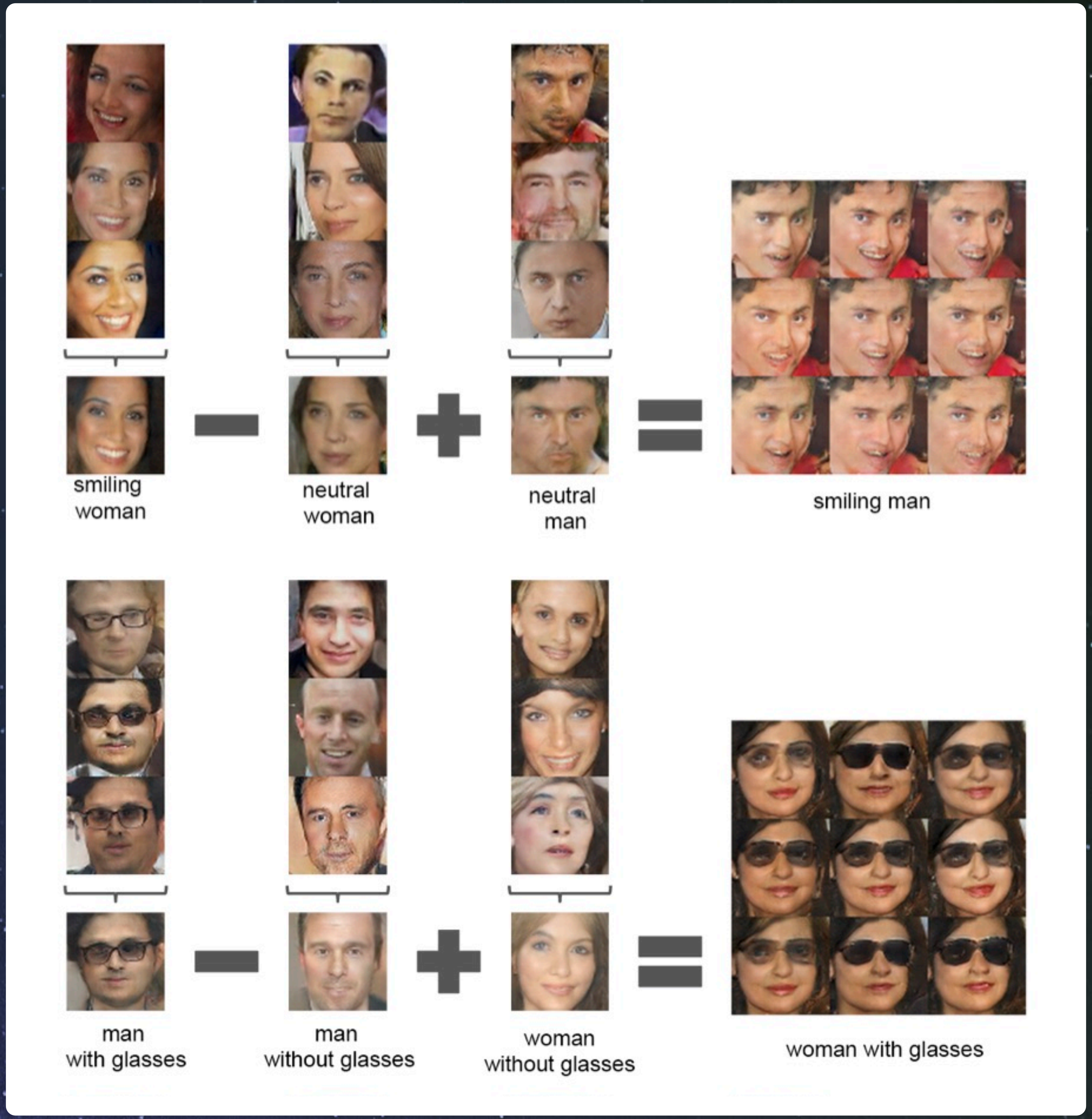
1 Introduction

Learning reusable feature representations from large unlabeled datasets has been an area of active research. In the context of computer vision, one can leverage the practically unlimited amount of unlabeled images and videos to learn good intermediate representations, which can then be used on a variety of supervised learning tasks such as image classification. We propose that one way to build good image representations is by training Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), and later reusing parts of the generator and discriminator networks as feature extractors for supervised tasks. GANs provide an attractive alternative to maximum likelihood techniques. One can additionally argue that their learning process and the lack of a heuristic cost function (such as pixel-wise independent mean-square error) are attractive to representation learning. GANs have been known to be unstable to train, often resulting in generators that produce nonsensical outputs. There has been very limited published research in trying to understand and visualize what GANs learn, and the intermediate representations of multi-layer GANs.

In this paper, we make the following contributions

- We propose and evaluate a set of constraints on the architectural topology of Convolutional GANs that make them stable to train in most settings. We name this class of architectures Deep Convolutional GANs (DCGAN)
- We use the trained discriminators for image classification tasks, showing competitive performance with other unsupervised algorithms.
- We visualize the filters learnt by GANs and empirically show that specific filters have learned to draw specific objects.

2016

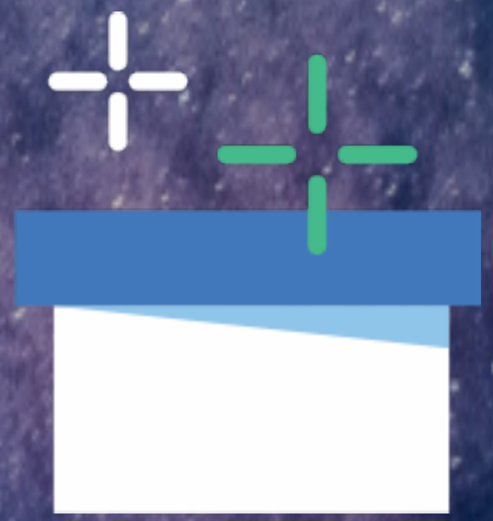


Artificial Intelligence & Deep Learning At Amazon

Thousands Of Employees Across The Company Focused on AI

Discovery &
Search

Fulfilment &
Logistics



Add ML-powered
features to existing products

Echo &
Alexa

Artificial Intelligence on AWS

P2, F1 &
Elastic GPUs

Deep Learning
AMI and template

Investment in
Apache MXNet

Open Source

Apache MXNet

Portable

Scala

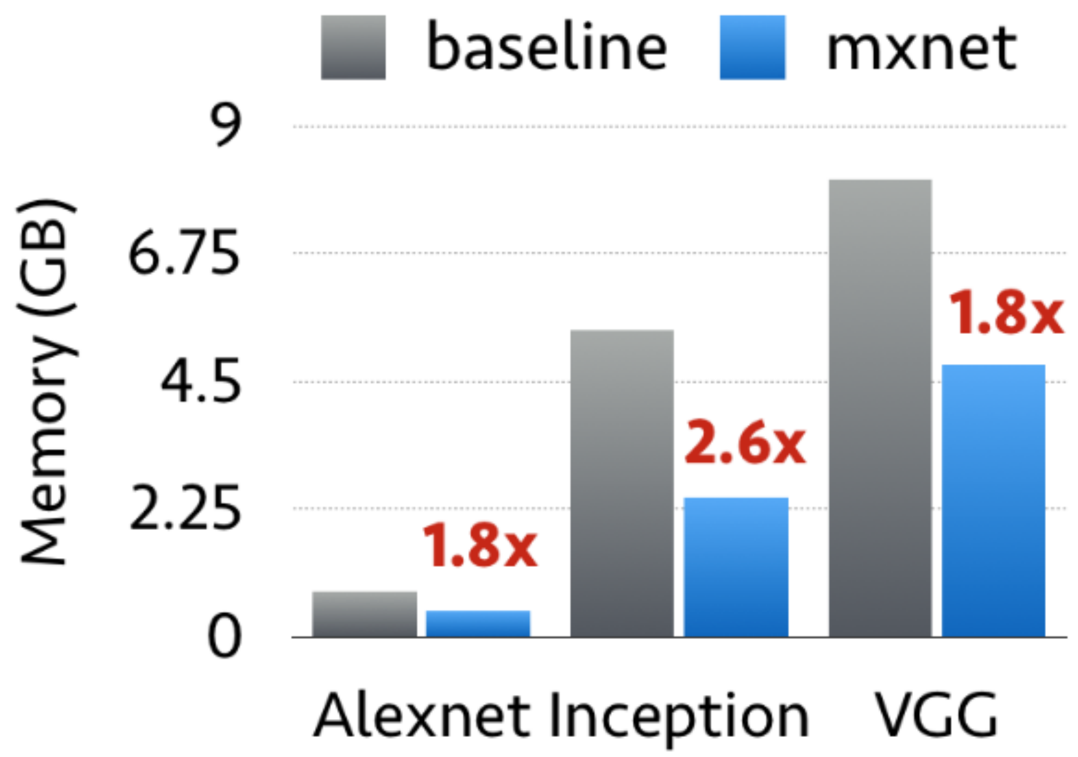
JS

python

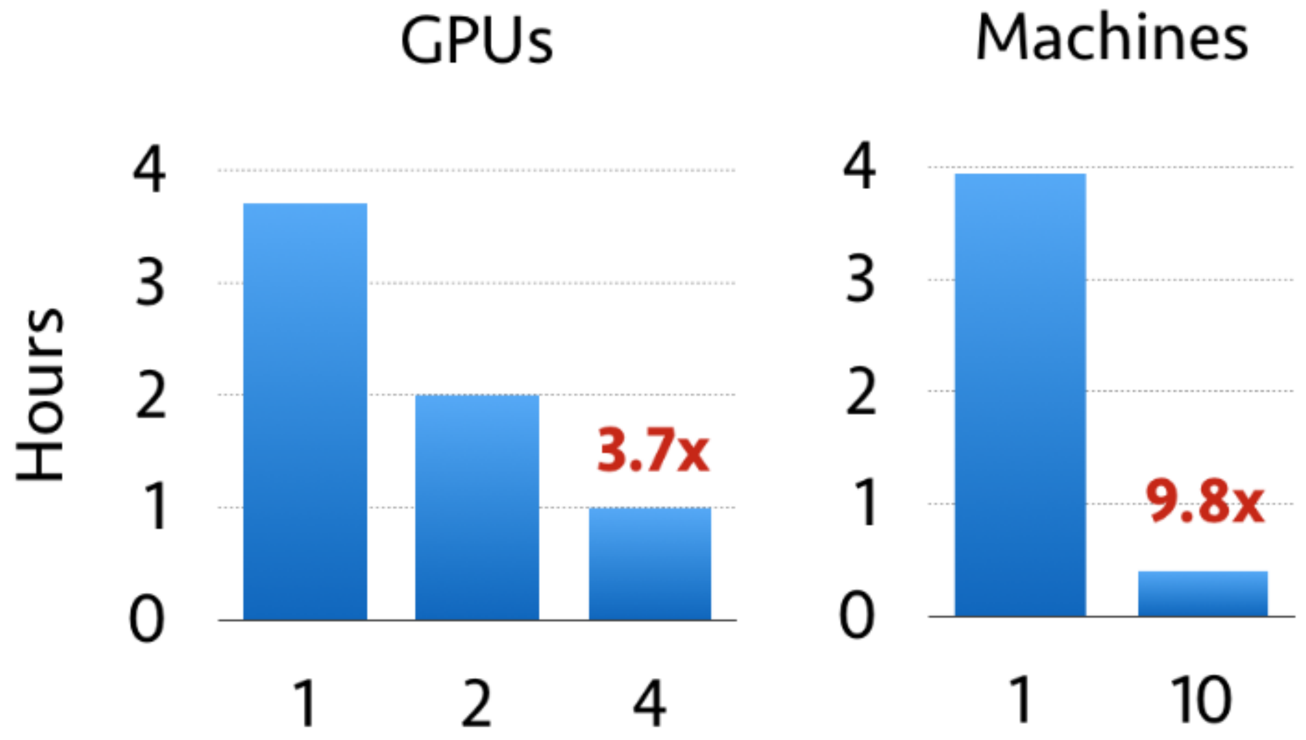
julia

Go

Efficient

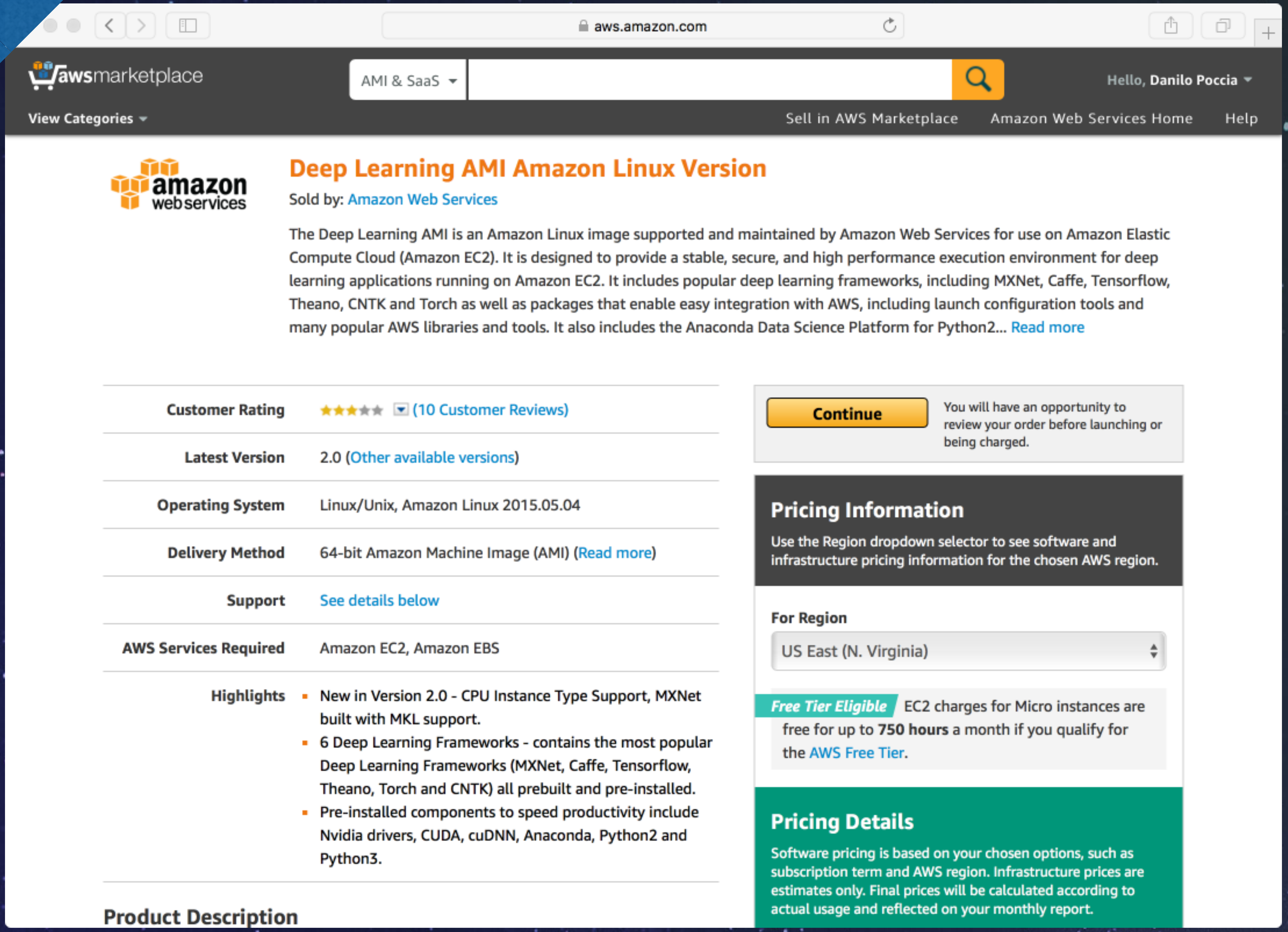


Scalable



Amazon EC2

Deep Learning AMI



Deep Learning Frameworks

MXNet, Caffe, Tensorflow, Theano, Torch, CNTK and Keras

Pre-installed components to speed productivity, such as Nvidia drivers, CUDA, cuDNN, Intel MKL-DNN with MXNet, Anaconda, Python 2 and 3

AWS Integration

Amazon AI

Bringing Powerful Artificial Intelligence To All Developers

Amazon AI

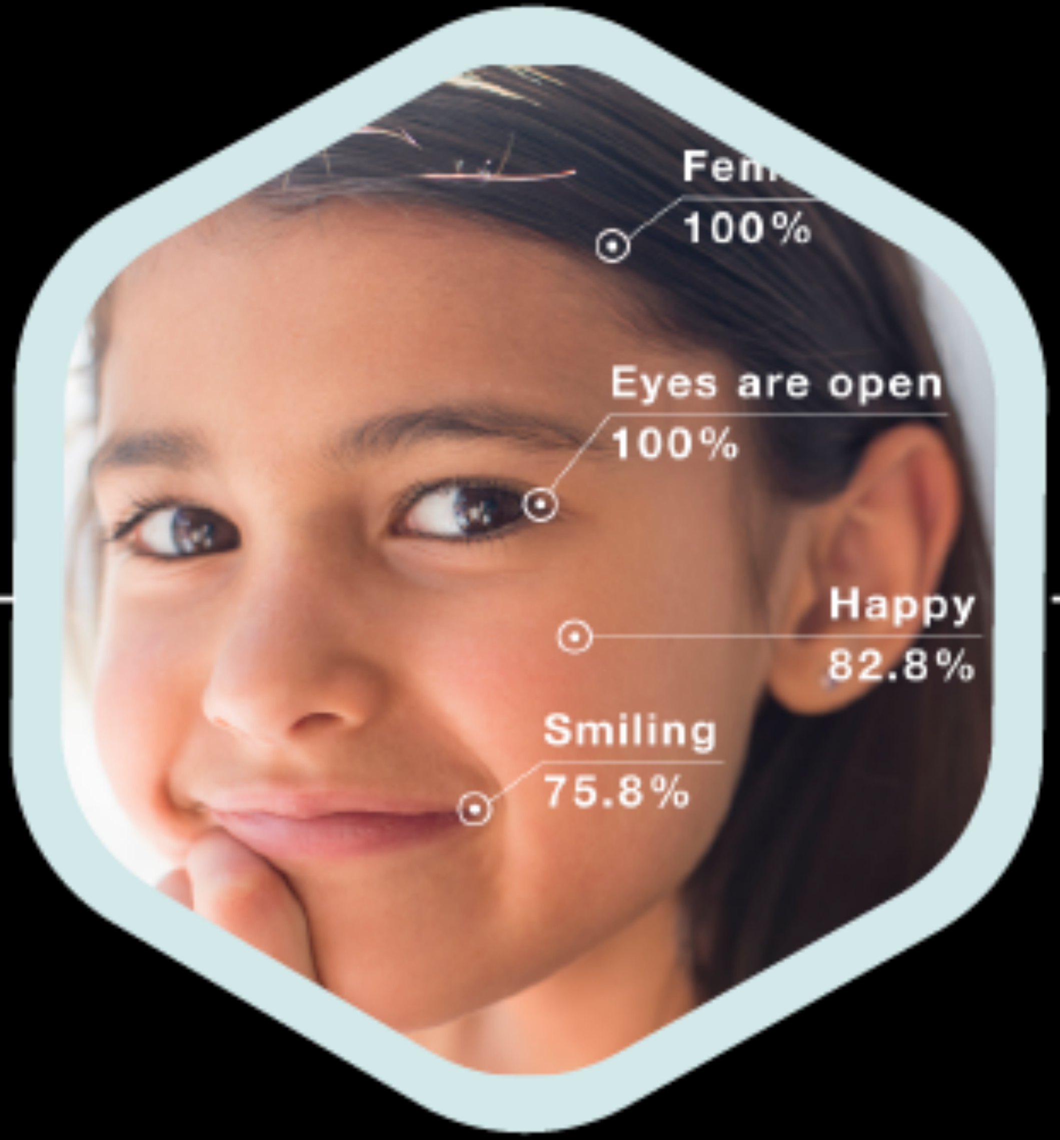
1 **Amazon Rekognition**

Image Recognition And Analysis
Powered By Deep Learning

Amazon Rekognition

Deep learning-based image recognition service
Search, verify, and organize millions of images

Object and Scene Detection



Facial Analysis

Face Comparison

Facial Recognition

Amazon Rekognition: Images In, Categories and Facial Analysis Out

Amazon Rekognition

Objects & Scenes

Car

Outside

Daytime

Driving

Faces

Female

Smiling

Sunglasses

Amazon Rekognition Facial Analysis
al analysis
a complete analysis of facial attributes, including confidence scores. (Your images aren't stored.)
Done with the demo?
Download SDKs
Results
looks like a face 99.9%
appears to be female 100%
age range 26 - 43 years old
not smiling 99.8%
appears to be happy 72%
not wearing eyeglasses 99.6%
not wearing sunglasses 99.7%
eyes are open 97.9%
mouth is closed 99.9%
does not have a mustache 99.6%
does not have a beard 99.3%
Show less
Choose a sample Image
Use your own image
Upload
or
Type or paste image URL
Go

Deep Learning Process

With Rekognition, Bynder revolutionizes marketing admin tasks with AI capabilities

“

With our new AI capabilities, Bynder’s software… now allows users to save hours of admin labor when uploading and organizing their files, adding exponentially more value.

bynder

Chris Hall
CEO, Bynder

”

Bynder allows you to easily create, find and use content for branding automation and marketing solutions.

Amazon AI

2 Amazon Polly

Text To Speech Powered By Deep Learning

Amazon Polly: Text In, Life-like Speech Out

"The temperature
in WA is 75°F"

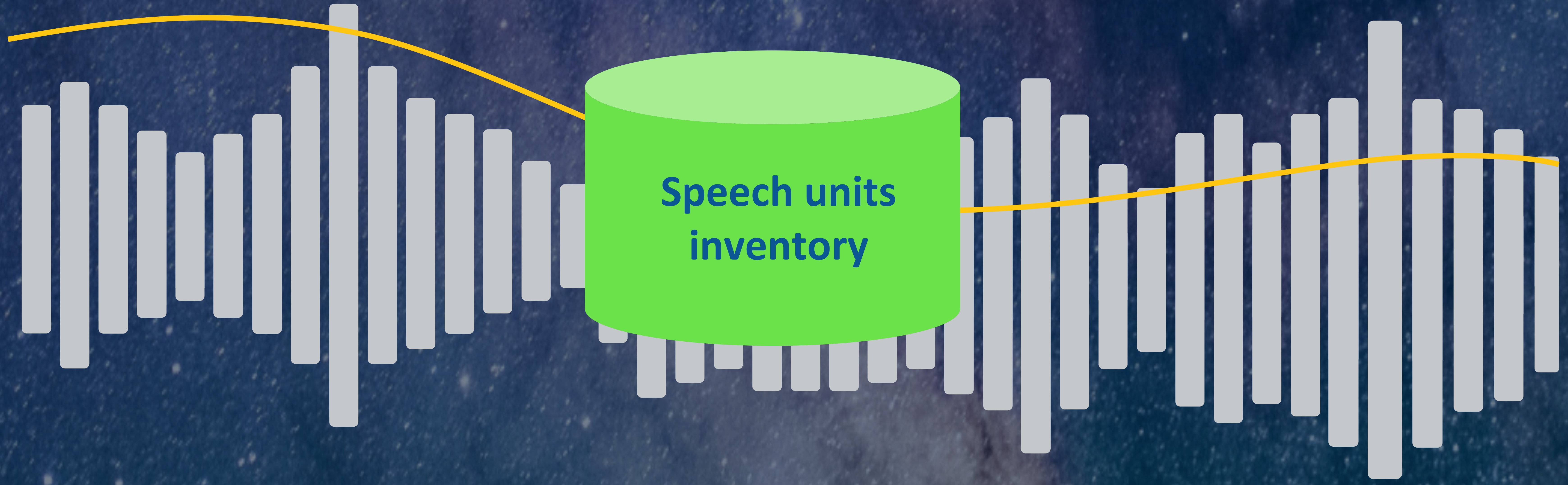
Amazon Polly

**"The temperature
in Washington is 75 degrees
Fahrenheit"**

TEXT PROCESSING

Market	grew	by	>	20%.
Market	grew	by	more than	twenty percent
ˈmɑɹ.kət	ˈgɹu	baɪ	ˈmoʊɹ ˈðæn	ˈtwɛn.ti pɚ.ˈsɛnt

UNIT SELECTION



Amazon Polly
AWS CLI

```
aws polly synthesize-speech
--text "It was nice to live such a wonderful live show."
--output-format mp3
--voice-id Joanna
--text-type text joanna.mp3)
```

Amazon Polly

*“Nel mezzo del cammin di nostra vita
mi ritrovai per una selva oscura
ché la diritta via era smarrita.”*

Duolingo voices its language learning service Using Polly

“

With Amazon Polly our users benefit from the most lifelike Text-to-Speech voices available on the market.

Severin Hacker
CTO, Duolingo

duolingo

”

Duolingo is a free language learning service where users help translate the web and rate translations.

- Spoken language crucial for language learning
- Accurate pronunciation matters
- Faster iteration thanks to TTS
- As good as natural human speech

With Polly, GoAnimate gives voice to the characters in their animations

“

Amazon Polly gives GoAnimate users the ability to immediately give voice to the characters they animate using our platform.

Alvin Hung
CEO, GoAnimate

GoAnimate

”

GoAnimate is a cloud-based, animated video creation plarform.

- Multi-language communication
  - Training or HR professionals who have to create content in many languages
- Video preproduction
  - Video makers who need to iterate and fine-tune before the text-to-speech is eventually replaced by a professional voiceover
- K–12 education
  - Students who make videos and don't have access to professional voices or time for or knowledge of voiceover

RNIB provides the largest library in the UK for people with sight loss

"Amazon Polly delivers incredibly lifelike voices which captivate and engage our readers."

John Worsfold
Solutions Implementation Manager, RNIB

RNIB Supporting people with sight loss

Royal National Institute of Blind People creates and distributes accessible information in the form of synthesized content

- RNIB delivers largest library of audiobooks in the UK for nearly 2 million people with sight loss
- Naturalness of generated speech is critical to captivate and engage readers
- No restrictions on speech redistributions enables RNIB to create and distribute accessible information in a form of synthesized content

Amazon AI

③ Amazon ALEXA

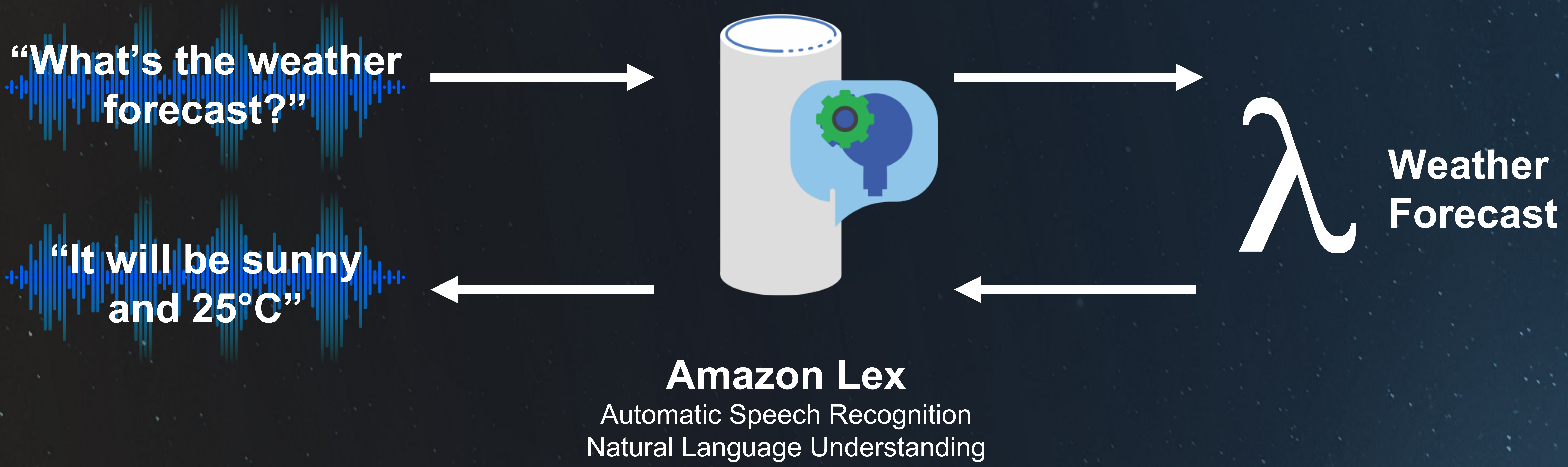
(It’s what’s inside Alexa)

Natural Language Understanding (NLU) &
Automatic Speech Recognition (ASR) Powered By Deep Learning

Amazon Lex: Speech Recognition & Natural Language Understanding

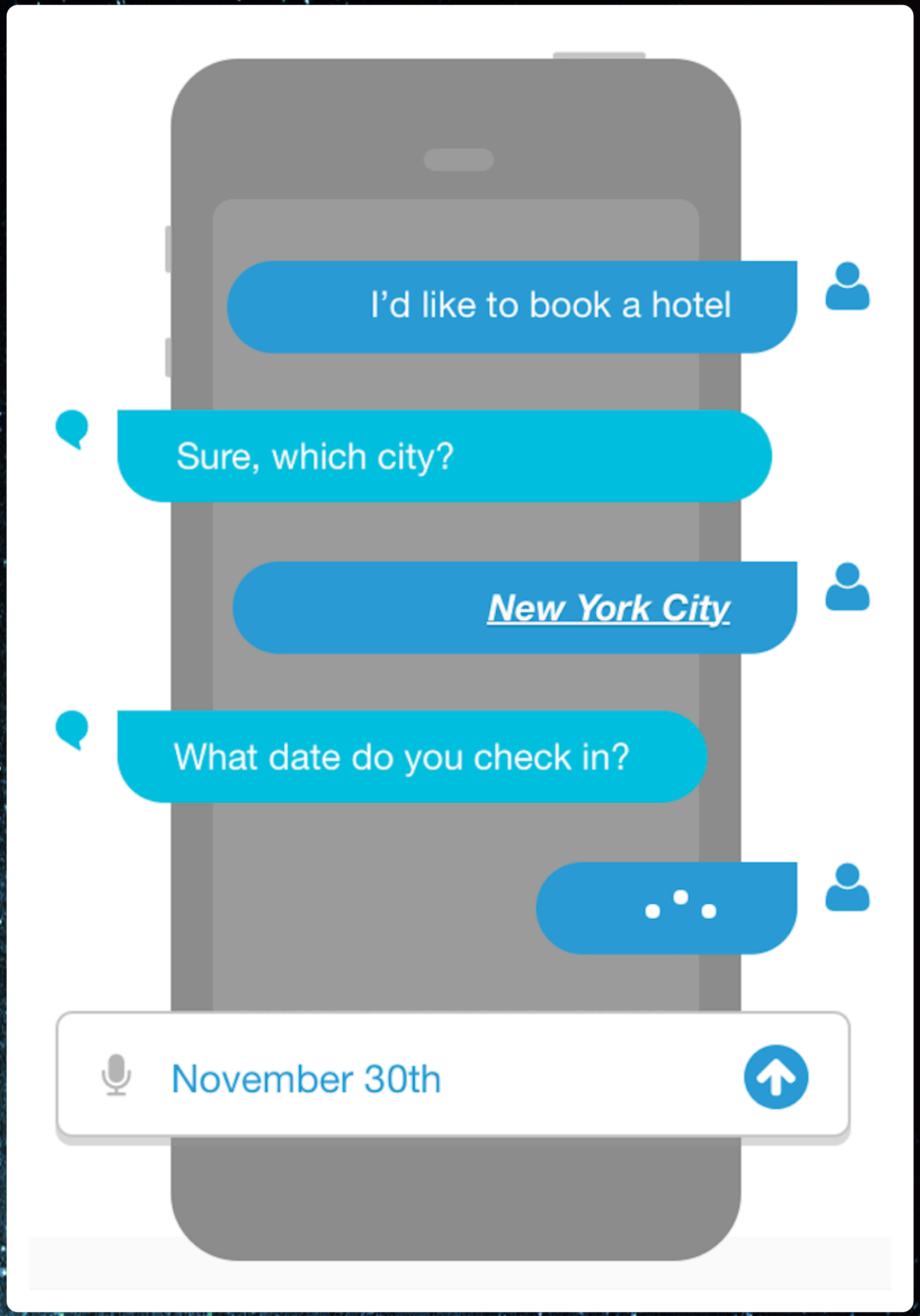
Amazon Lex
Automatic Speech Recognition
Natural Language Understanding

Amazon Lex: Speech Recognition & Natural Language Understanding



Lex Bot Structure

BookHotel

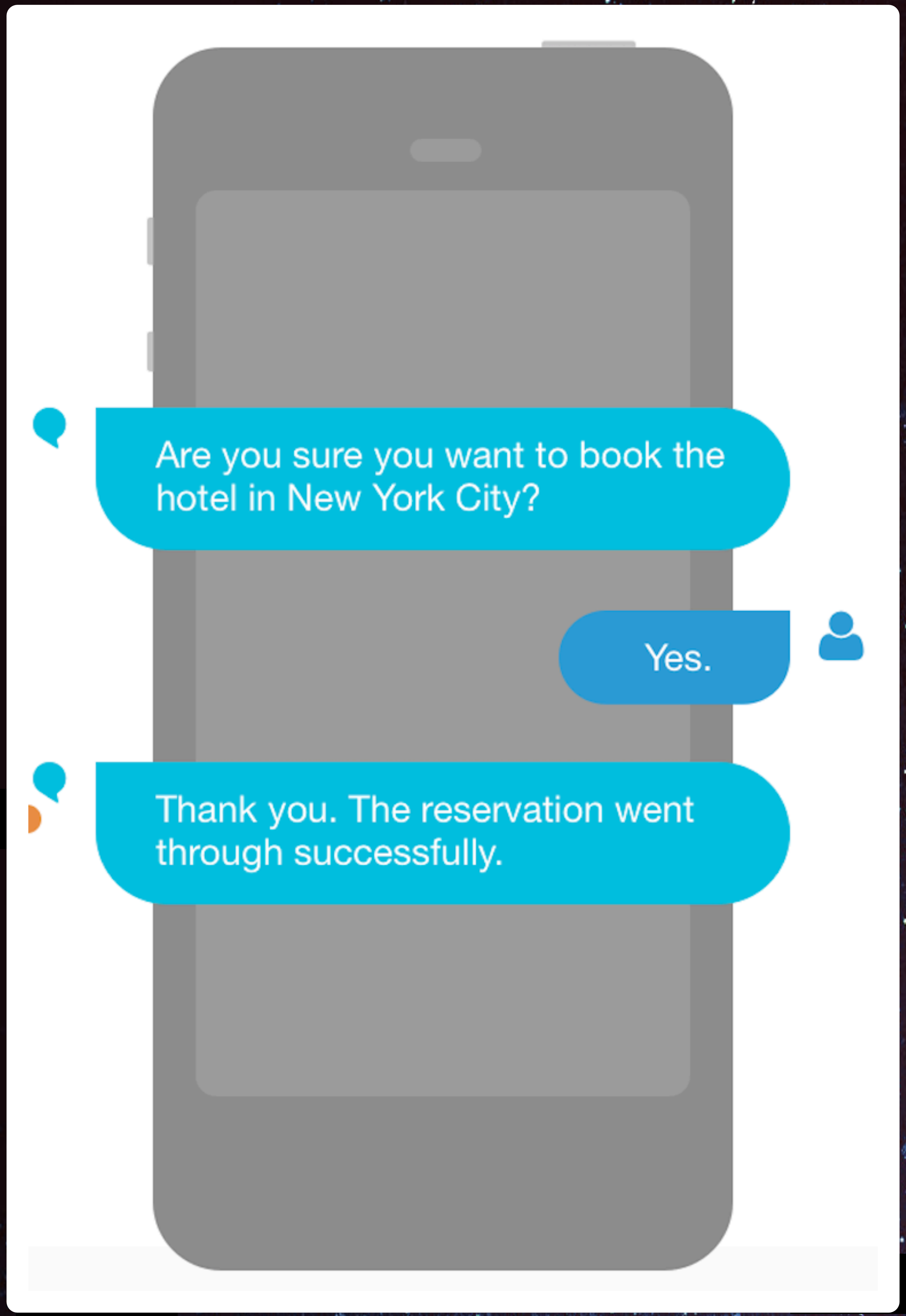


Intents
An Intent performs an action in response to natural language user input

Utterances
Spoken or typed phrases that invoke your intent

Slots
Slots are input data required to fulfill the intent

Fulfillment
Fulfillment mechanism for your intent



“Book a Hotel”

“Book a Hotel in NYC”

Book a Hotel in NYC

Automatic Speech Recognition

Hotel Booking

New York City

Natural Language Understanding

Intent/Slot Model

Hotel Booking	
City	New York City
Check In	Nov 30^{th}
Check Out	Dec 2^{nd}

Utterances

“Can I go ahead with the booking?”

λ

Confirmation: “Your hotel is booked for Nov 30^{th}”

Polly

“Your hotel is booked for Nov 30^{th}”

Motorola Solutions is using Amazon AI to help finding missing persons

"

Finding missing persons:
~100,000 active missing persons cases in the U.S. at any given time
~60% are adults,
~40% are children

MOTOROLA SOLUTIONS

"

Motorola Solutions keeps utility workers connected and visible to each other with real-time voice and data communication across the smart grid.

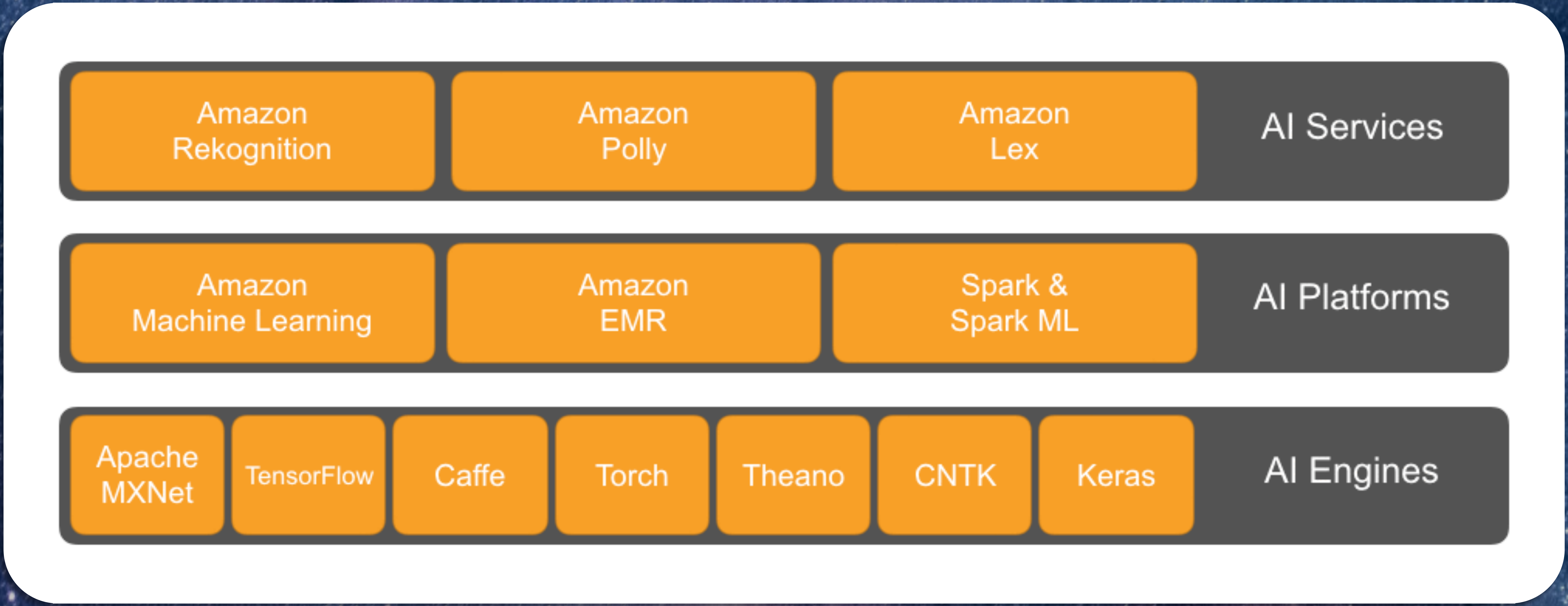
- Motorola Solutions applies Amazon Rekognition, Amazon Polly and Amazon Lex
- Image analytics and facial recognition can continually monitor for missing persons
- Tools that understand natural language can enable officers to keep eyes up and hands free

<demo>

I See

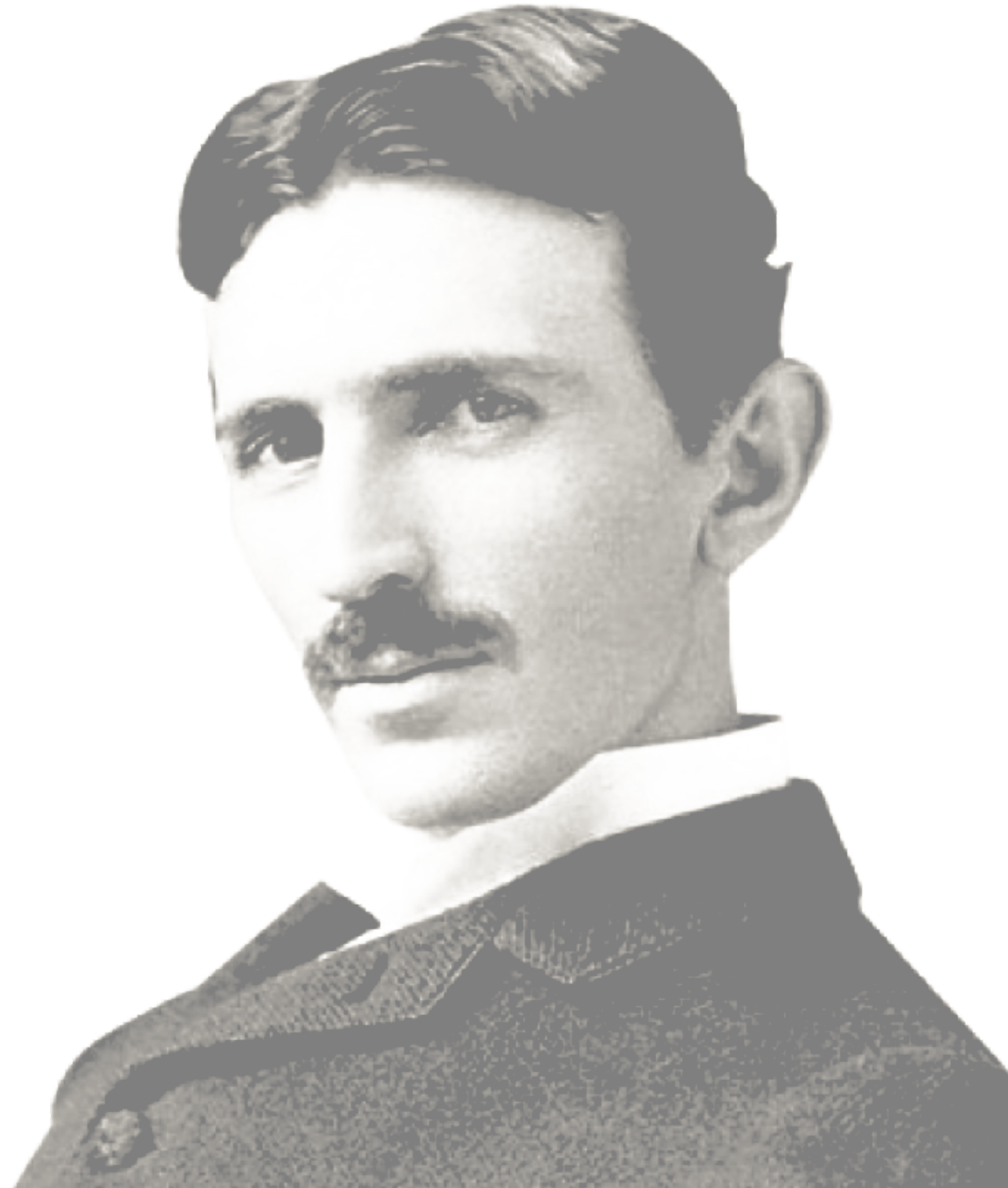
</demo>


I See
Demo
Raspberry Pi
I see…
Camera
Voice
Detect Labels
Detect Faces
Synthesize
Speech
Amazon
Rekognition
Amazon
Polly


Amazon Rekognition
Amazon Polly
Amazon Lex
AI Services
Amazon Machine Learning
Amazon EMR
Spark & Spark ML
AI Platforms
Apache MXNet
TensorFlow
Caffe
Torch
Theano
CNTK
Keras
AI Engines

"When wireless is perfectly applied, the whole earth will be converted into a huge brain…"

Nikola Tesla, 1926



Machines Are Learning

Bringing Powerful Artificial Intelligence Tools to Developers

Danilo Poccia

AWS Technical Evangelist

@danilop danilop

CloudConf 2017