

Autonomous and Adaptive Systems

AI and Creativity: Generative Machine Learning

Mirco Musolesi

mircomusolesi@acm.org

What is Creativity?

“Creativity can be defined as the ability to generate novel, and valuable, ideas. Valuable, here, has many meanings: interesting, useful, beautiful, simple, richly complex, and so on. Ideas covers many meaning too: not only ideas as such (concepts, theories, interpretations, stories), but also artifacts such as graphic images, sculptures, houses and jet engines. Computer models have been designed to generate ideas in all these areas and more.”

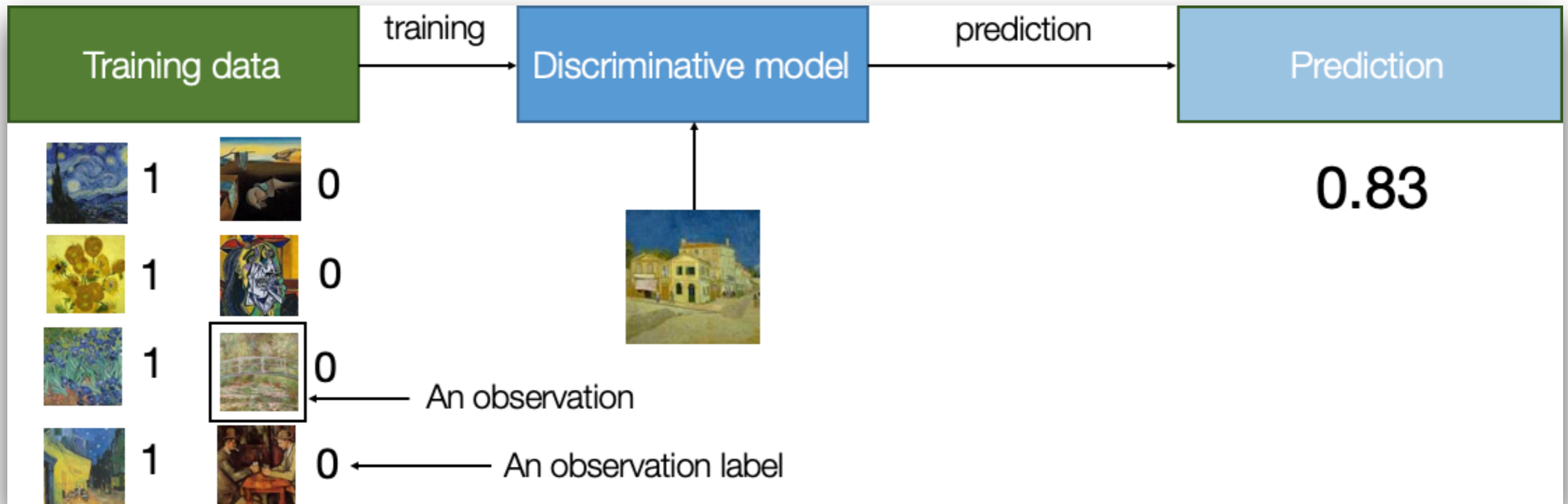
Margaret A. Boden

Can an (artificial) agent be creative?

Generative Modeling

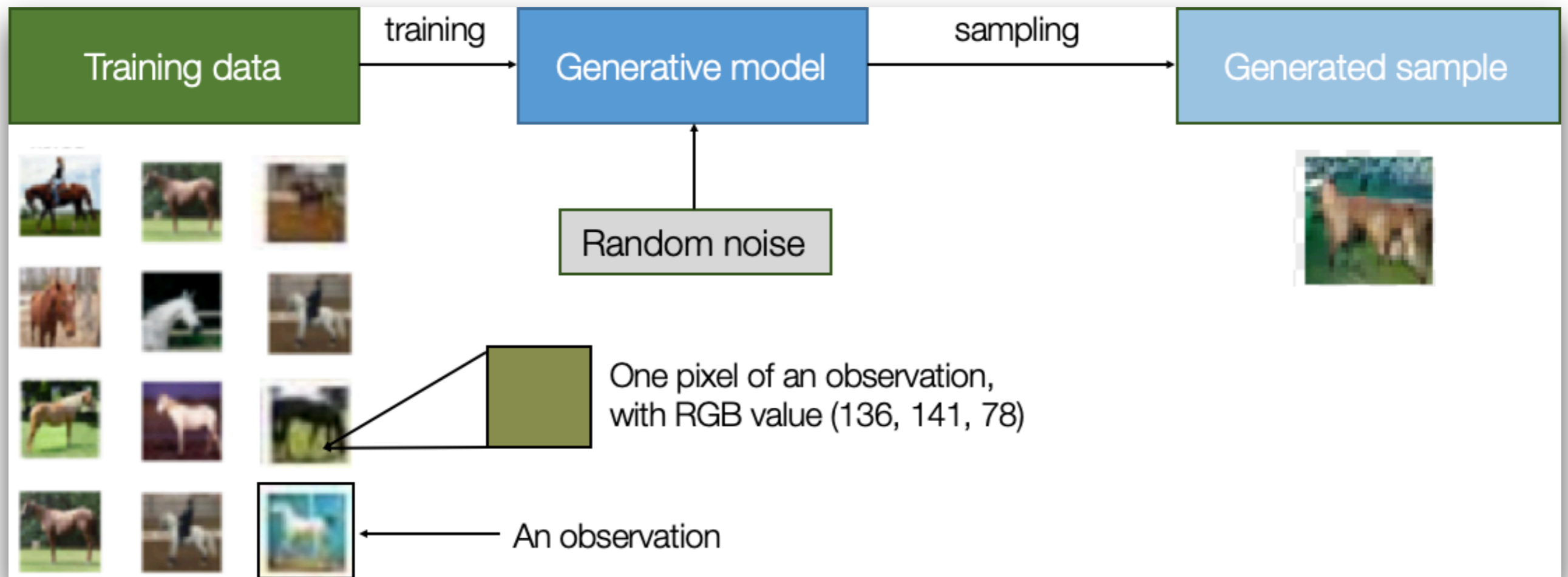
- ▶ A generative model describes how a dataset is generated for example through a probabilistic model. Through sampling of this model, we generate new data.
- ▶ The goal is to generated data are variations of the existing ones, but not “too far” from the original dataset.
- ▶ A generative model is usually probabilistic rather than deterministic in nature.

Discriminative Model



Source: David Foster. Generative Deep Learning. O'Reilly. 2019.

Generative Model



Source: David Foster. Generative Deep Learning. O'Reilly. 2019.

Generative Modelling Framework

- ▶ Given a dataset \mathbf{X} , we assume that the observations have been according to some *unknown* distribution p_{data} .
- ▶ The goal is to create a generative model p_{model} that can be used to generate samples that look like they were drawn from p_{data} .
- ▶ We achieved our goal if the generated data are also suitably different from the observations in \mathbf{X} .
 - ▶ The model should not simply reproduce the things that have already been seen.



LiveSlides web content

To view

Download the add-in.

liveslides.com/download

Start the presentation.

DeepDream

- ▶ Developed by Alexander Mordvintsev, Christopher Olah and Mike Tika in 2015.
- ▶ This movement is also called inceptionism from Chris Nolan's movie "Inception" (but a bit indirectly).
- ▶ The idea is to try to exploit the "patterns" that are learned by neural network "in reverse".
- ▶ This is used to generate images that are composed by the patterns that are detected by the different layers.

Going deeper with convolutions

Christian Szegedy

Google Inc.

Wei Liu

University of North Carolina, Chapel Hill

Yangqing Jia

Google Inc.

Pierre Sermanet

Google Inc.

Scott Reed

University of Michigan

Dragomir Anguelov

Google Inc.

Dumitru Erhan

Google Inc.

Vincent Vanhoucke

Google Inc.

Andrew Rabinovich

Google Inc.

Abstract

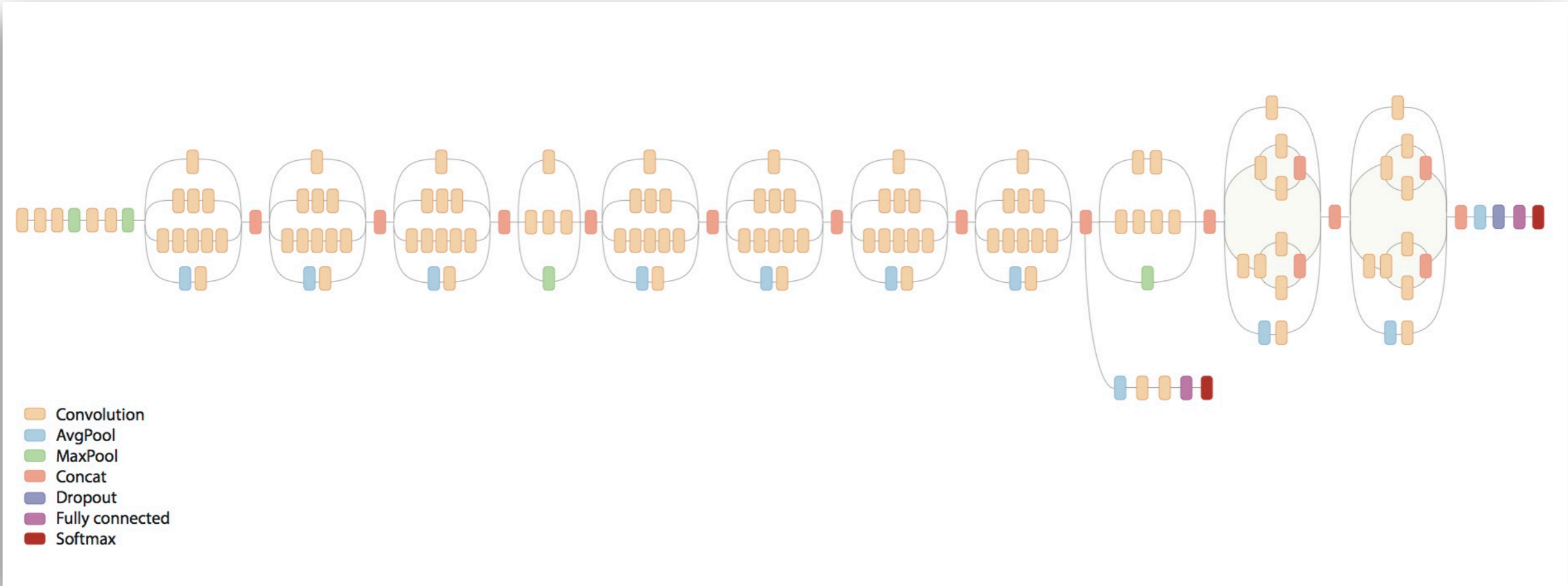
We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.



DeepDream

- ▶ Interpretability is still an open question in deep learning.
- ▶ In case of images, we know that each layer progressively extracts higher and higher-level features of the images, until the final layer makes a decision on what an image actually shows.
 - ▶ First layer for edges and corners, then intermediate layers interpret the basic features and are used to extract shapes and components (windows, leaves, etc). A final layer might extract buildings or trees.
- ▶ One way to visualise this is to turn the network upside down and ask to enhance an input image so that the role of each layer can be interpreted.

Inception Network



Source: Inception in TensorFlow
<https://github.com/tensorflow/models/tree/master/research/inception>

Inception Network

Rethinking the Inception Architecture for Computer Vision

Christian Szegedy
Google Inc.

szegedy@google.com

Vincent Vanhoucke
vanhoucke@google.com

Sergey Ioffe
sioffe@google.com

Jonathon Shlens
shlens@google.com

Zbigniew Wojna
University College London
zbigniewwojna@gmail.com

Abstract

Convolutional networks are at the core of most state-of-the-art computer vision solutions for a wide variety of tasks. Since 2014 very deep convolutional networks started to become mainstream, yielding substantial gains in various benchmarks. Although increased model size and computational cost tend to translate to immediate quality gains for most tasks (as long as enough labeled data is provided for training), computational efficiency and low parameter count are still enabling factors for various use cases such as mobile vision and big-data scenarios. Here we are exploring ways to scale up networks in ways that aim at utilizing the added computation as efficiently as possible by suitably factorized convolutions and aggressive regularization. We benchmark our methods on the ILSVRC 2012 classification challenge validation set demonstrate substantial gains over

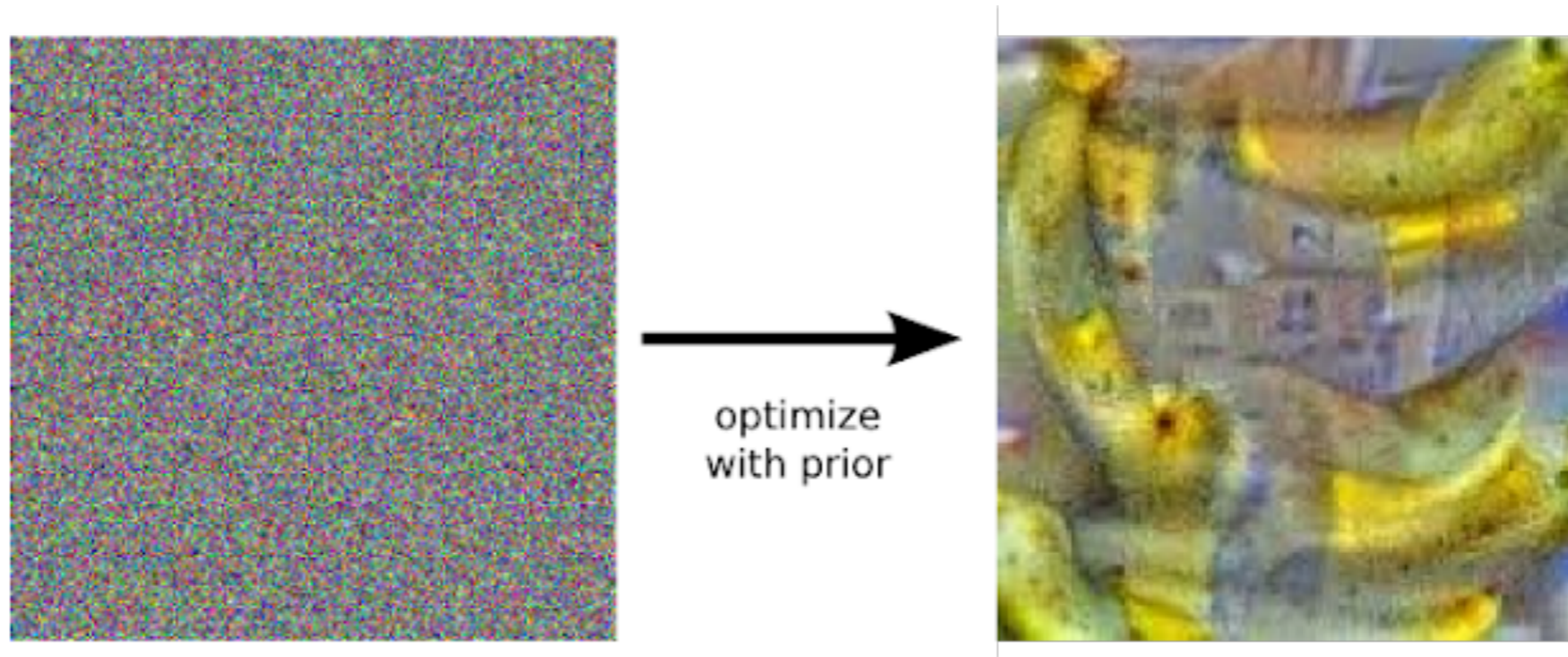
lary high performance in the 2014 ILSVRC [16] classification challenge. One interesting observation was that gains in the classification performance tend to transfer to significant quality gains in a wide variety of application domains. This means that architectural improvements in deep convolutional architecture can be utilized for improving performance for most other computer vision tasks that are increasingly reliant on high quality, learned visual features. Also, improvements in the network quality resulted in new application domains for convolutional networks in cases where AlexNet features could not compete with hand engineered, crafted solutions, e.g. proposal generation in detection[4].

Although VGGNet [18] has the compelling feature of architectural simplicity, this comes at a high cost: evaluating the network requires a lot of computation. On the other hand, the Inception architecture of GoogLeNet [20]

DeepDream

- ▶ The idea of DeepDream is to choose a layer (or layers) and minimise the loss in a way that the image increasingly “excites” the layers.
- ▶ The complexity of the features incorporated depends on the layer we chose.
- ▶ We use the InceptionV3 architecture.
 - ▶ For DeepDream the layers of interest are those where the convolutions are concatenated.
- ▶ Once we have calculated the loss for the chosen layers, we calculate the gradients (using gradient ascent) to the input images.
- ▶ If we consider a “noise” image, the shapes that are identified by that specific layer will appear.

DeepDream



Source: Inceptionism: Going Deeper into Neural Networks
<https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html>

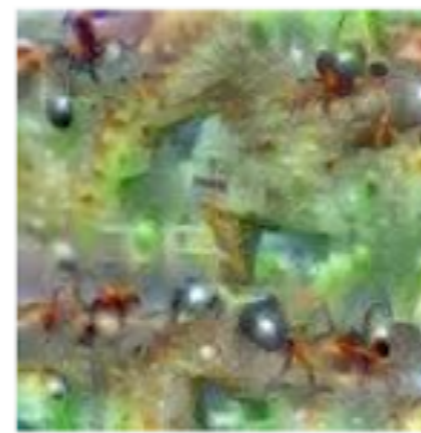
DeepDream



Hartebeest



Measuring Cup



Ant



Starfish



Anemone Fish



Banana



Parachute



Screw

Source: Inceptionism: Going Deeper into Neural Networks
<https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html>

DeepDream



Source: Inceptionism: Going Deeper into Neural Networks
<https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html>

DeepDream



Source: Inceptionism: Going Deeper into Neural Networks
<https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html>

TensorFlow DeepDream Colab

<https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/generative/deepdream.ipynb>

Neural Style Transfer



Source: TensorFlow Neural Style Transfer
https://www.tensorflow.org/tutorials/generative/style_transfer

Neural Style Transfer



Source: TensorFlow Neural Style Transfer
https://www.tensorflow.org/tutorials/generative/style_transfer

Neural Style Transfer



Source: TensorFlow Neural Style Transfer
https://www.tensorflow.org/tutorials/generative/style_transfer

A Neural Algorithm of Artistic Style

Leon A. Gatys,^{1,2,3*} Alexander S. Ecker,^{1,2,4,5} Matthias Bethge^{1,2,4}

¹Werner Reichardt Centre for Integrative Neuroscience

and Institute of Theoretical Physics, University of Tübingen, Germany

²Bernstein Center for Computational Neuroscience, Tübingen, Germany

³Graduate School for Neural Information Processing, Tübingen, Germany

⁴Max Planck Institute for Biological Cybernetics, Tübingen, Germany

⁵Department of Neuroscience, Baylor College of Medicine, Houston, TX, USA

*To whom correspondence should be addressed; E-mail: leon.gatys@bethgelab.org

In fine art, especially painting, humans have mastered the skill to create unique visual experiences through composing a complex interplay between the content and style of an image. Thus far the algorithmic basis of this process is unknown and there exists no artificial system with similar capabilities. However, in other key areas of visual perception such as object and face recognition near-human performance was recently demonstrated by a class of biologically inspired vision models called Deep Neural Networks.^{1,2} Here we introduce an artificial system based on a Deep Neural Network that creates artistic images

Generative Adversarial Networks (GANs)

- ▶ A Generative Adversarial Network (GAN) is a class of machine learning techniques in which two neural networks play against each other.
- ▶ The *generative network* generates candidates, while the *discriminative network* evaluate them.
- ▶ The generative network tries to create new samples that look similar from the true data (an original distribution, for example portraits). The goal of the discriminator is to identify if the data given in input are from the original distribution or not.
- ▶ The generative network's training objective is to increase the error rate of the discriminative network (i.e., to fool the discriminative network)
- ▶ Indeed, the discriminative network's training objective is to minimise its error rate in discriminating the input.
- ▶ This is used for images, videogame generation, scientific images, etc.

Generative Adversarial Nets

Ian J. Goodfellow*, **Jean Pouget-Abadie†**, **Mehdi Mirza**, **Bing Xu**, **David Warde-Farley**,
Sherjil Ozair‡, **Aaron Courville**, **Yoshua Bengio§**

Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G . The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D , a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

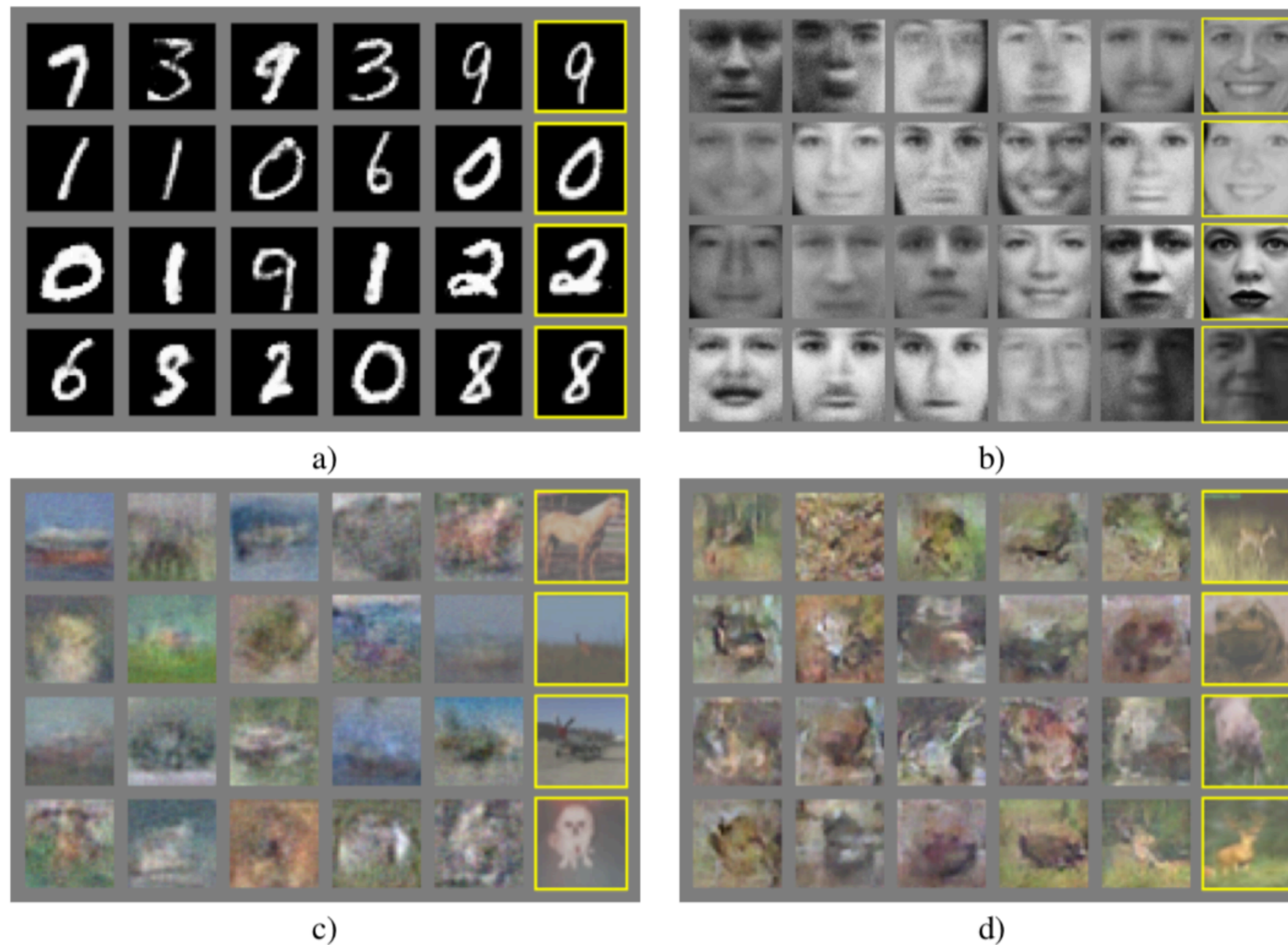


Figure 2: Visualization of samples from the model. Rightmost column shows the nearest training example of the neighboring sample, in order to demonstrate that the model has not memorized the training set. Samples are fair random draws, not cherry-picked. Unlike most other visualizations of deep generative models, these images show actual samples from the model distributions, not conditional means given samples of hidden units. Moreover, these samples are uncorrelated because the sampling process does not depend on Markov chain mixing. a) MNIST b) TFD c) CIFAR-10 (fully connected model) d) CIFAR-10 (convolutional discriminator and “deconvolutional” generator)

Generation of Images using Generative Adversarial Networks (GANs)



Source: David Foster. Generative Deep Learning. O'Reilly. 2019.

StyleGAN





 **LiveSlides** web content

To view

Download the add-in.

liveslides.com/download

Start the presentation.

A Style-Based Generator Architecture for Generative Adversarial Networks

Tero Karras
NVIDIA

tkarras@nvidia.com

Samuli Laine
NVIDIA

slaine@nvidia.com

Timo Aila
NVIDIA

taila@nvidia.com

Abstract

We propose an alternative generator architecture for generative adversarial networks, borrowing from style transfer literature. The new architecture leads to an automatically learned, unsupervised separation of high-level attributes (e.g., pose and identity when trained on human faces) and stochastic variation in the generated images (e.g., freckles, hair), and it enables intuitive, scale-specific control of the synthesis. The new generator improves the state-of-the-art in terms of traditional distribution quality metrics, leads to demonstrably better interpolation properties, and also better disentangles the latent factors of variation. To quantify interpolation quality and disentanglement, we propose two new, automated methods that are applicable to any generator architecture. Finally, we introduce a new, highly varied and high-quality dataset of human faces.

(e.g., pose, identity) from stochastic variation (e.g., freckles, hair) in the generated images, and enables intuitive scale-specific mixing and interpolation operations. We do not modify the discriminator or the loss function in any way, and our work is thus orthogonal to the ongoing discussion about GAN loss functions, regularization, and hyperparameters [24, 45, 5, 40, 44, 36].

Our generator embeds the input latent code into an intermediate latent space, which has a profound effect on how the factors of variation are represented in the network. The input latent space must follow the probability density of the training data, and we argue that this leads to some degree of unavoidable entanglement. Our intermediate latent space is free from that restriction and is therefore allowed to be disentangled. As previous methods for estimating the degree of latent space disentanglement are not directly applicable in our case, we propose two new automated metrics — perceptual path length and linear separability — for quantifying these aspects of the generator. Using these metrics, we show that compared to a traditional generator architecture,

Text Generation

PANDARUS:

Alas, I think he shall be come approached and the day
When little strain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:

They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:

Well, your wit is in the care of side and that.

Second Lord:

They would be ruled after this chamber, and
my fair nudes begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:

Come, sir, I will make did behold your worship.

VIOLA:

I'll drink it

Source: Andrej Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks.
<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

Text Generation

Proof. Omitted. □

Lemma 0.1. *Let \mathcal{C} be a set of the construction. Let \mathcal{C} be a gerber covering. Let \mathcal{F} be a quasi-coherent sheaves of \mathcal{O} -modules. We have to show that*

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

Proof. This is an algebraic space with the composition of sheaves \mathcal{F} on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where \mathcal{G} defines an isomorphism $\mathcal{F} \rightarrow \mathcal{F}$ of \mathcal{O} -modules. □

Lemma 0.2. *This is an integer \mathcal{Z} is injective.*

Proof. See Spaces, Lemma ?? □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \rightarrow Y' \rightarrow Y \rightarrow Y \rightarrow Y' \times_X Y \rightarrow X.$$

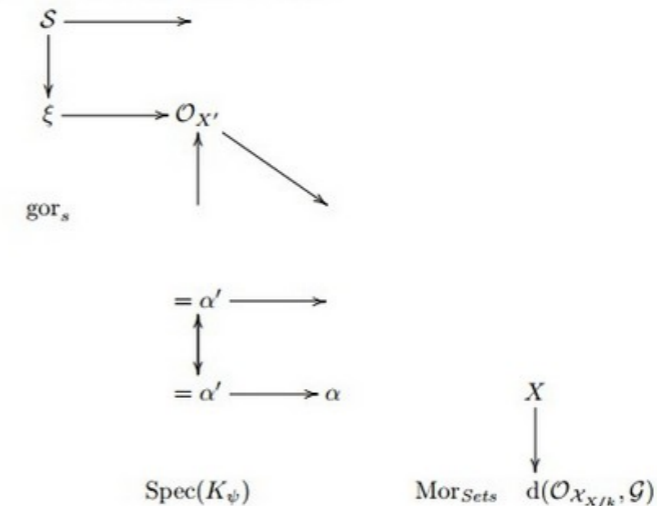
be a morphism of algebraic spaces over S and Y .

Proof. Let X be a nonzero scheme of X . Let X be an algebraic space. Let \mathcal{F} be a quasi-coherent sheaf of \mathcal{O}_X -modules. The following are equivalent

- (1) \mathcal{F} is an algebraic space over S .
- (2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



is a limit. Then \mathcal{G} is a finite type and assume S is a flat and \mathcal{F} and \mathcal{G} is a finite type f_* . This is of finite type diagrams, and

- the composition of \mathcal{G} is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \text{Spec}(R)$ and \mathcal{F} is a finite type representable by algebraic space. The property \mathcal{F} is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U . □

Proof. This is clear that \mathcal{G} is a finite presentation, see Lemmas ??
A reduced above we conclude that U is an open covering of \mathcal{C} . The functor \mathcal{F} is a "field"

$$\mathcal{O}_{X,x} \rightarrow \mathcal{F}_{\bar{x}} \rightarrow \mathcal{O}_{X_{\acute{e}tale}}^{-1} \rightarrow \mathcal{O}_{X_{\acute{e}tale}}^{-1} \mathcal{O}_{X_{\lambda}}(\mathcal{O}_{X_{\eta}}^{\bar{v}})$$

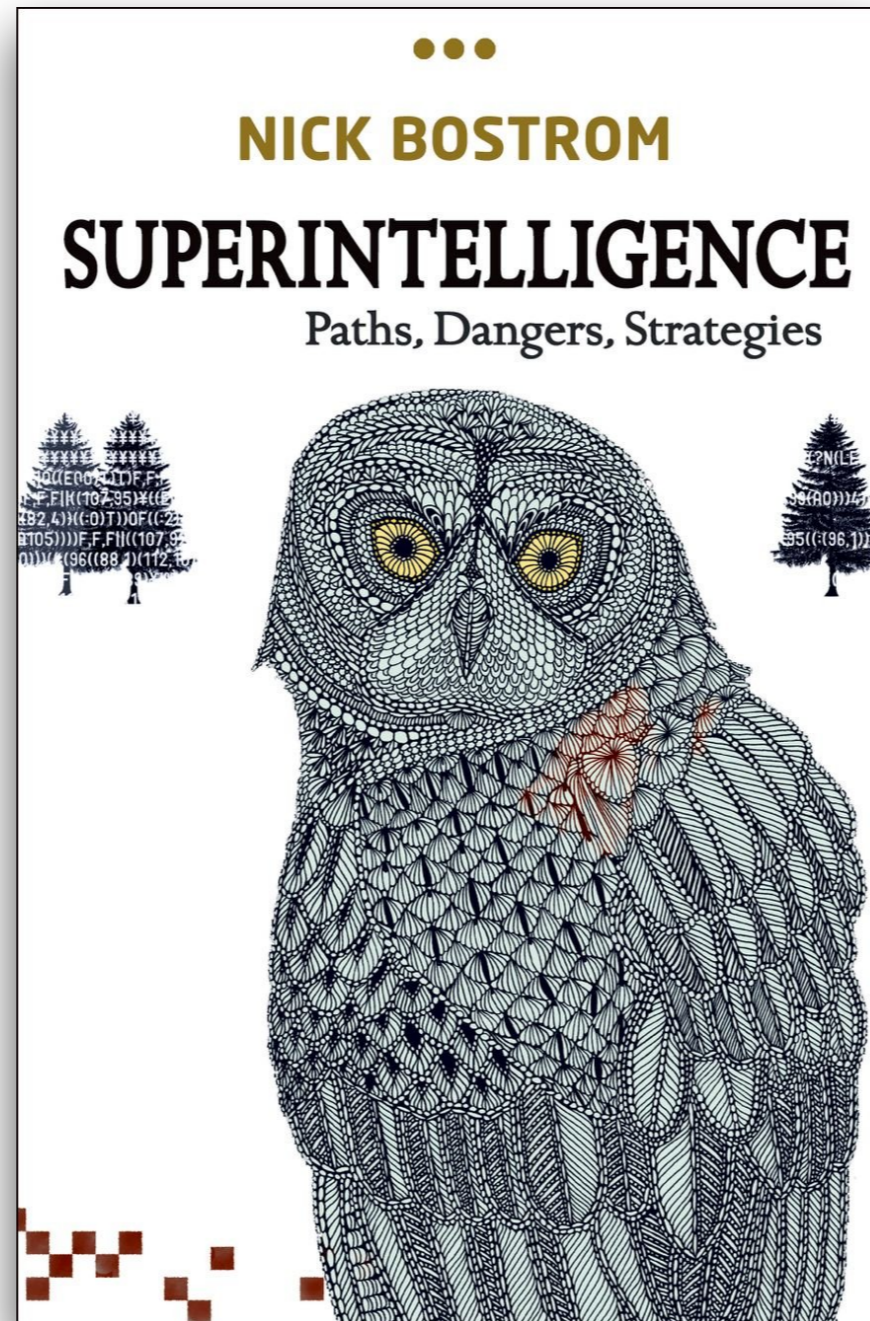
is an isomorphism of covering of \mathcal{O}_{X_1} . If \mathcal{F} is the unique element of \mathcal{F} such that X is an isomorphism.

The property \mathcal{F} is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme \mathcal{O}_X -algebra with \mathcal{F} are opens of finite type over S .

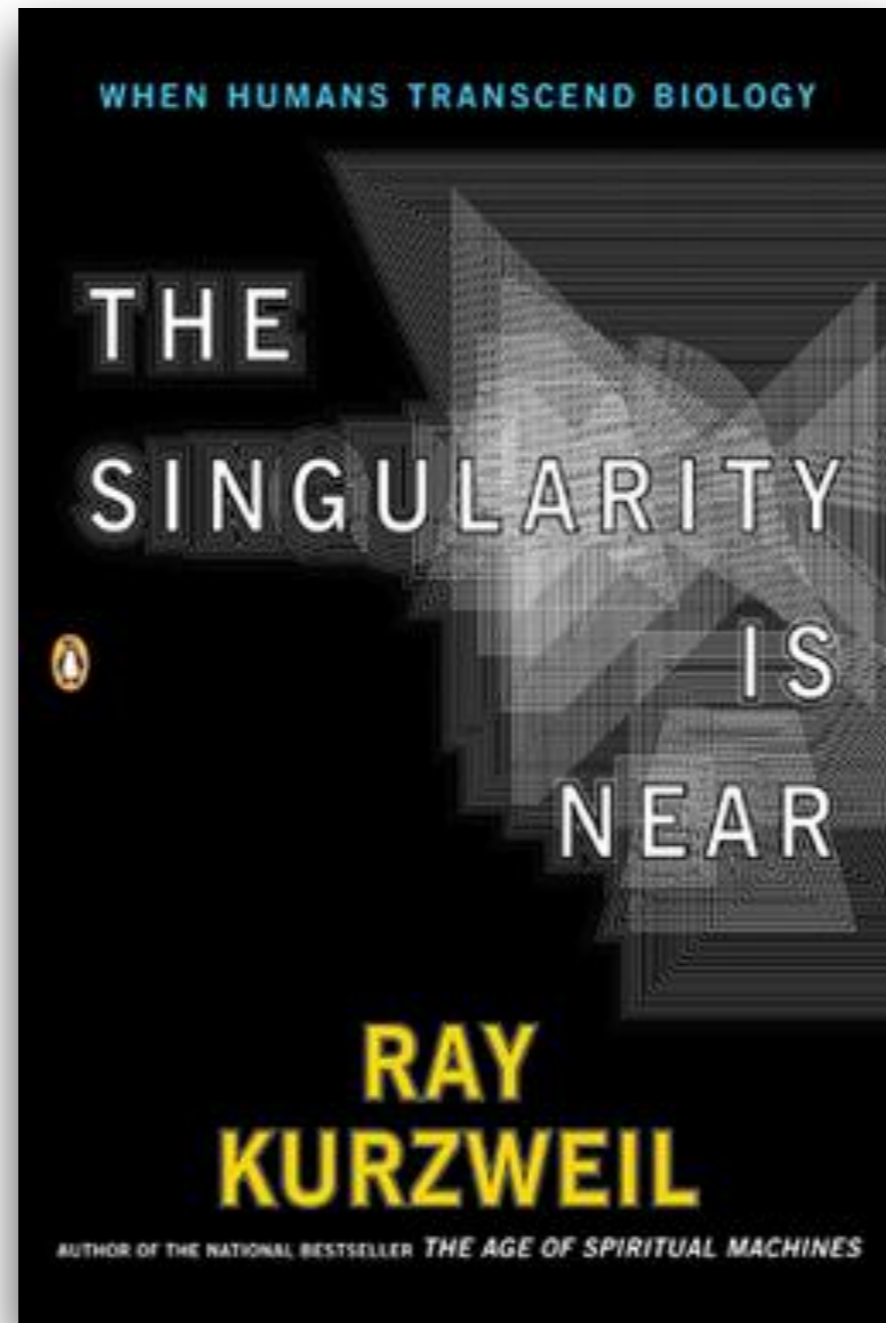
If \mathcal{F} is a scheme theoretic image points. □

If \mathcal{F} is a finite direct sum $\mathcal{O}_{X_{\lambda}}$ is a closed immersion, see Lemma ??
This is a sequence of \mathcal{F} is a similar morphism.

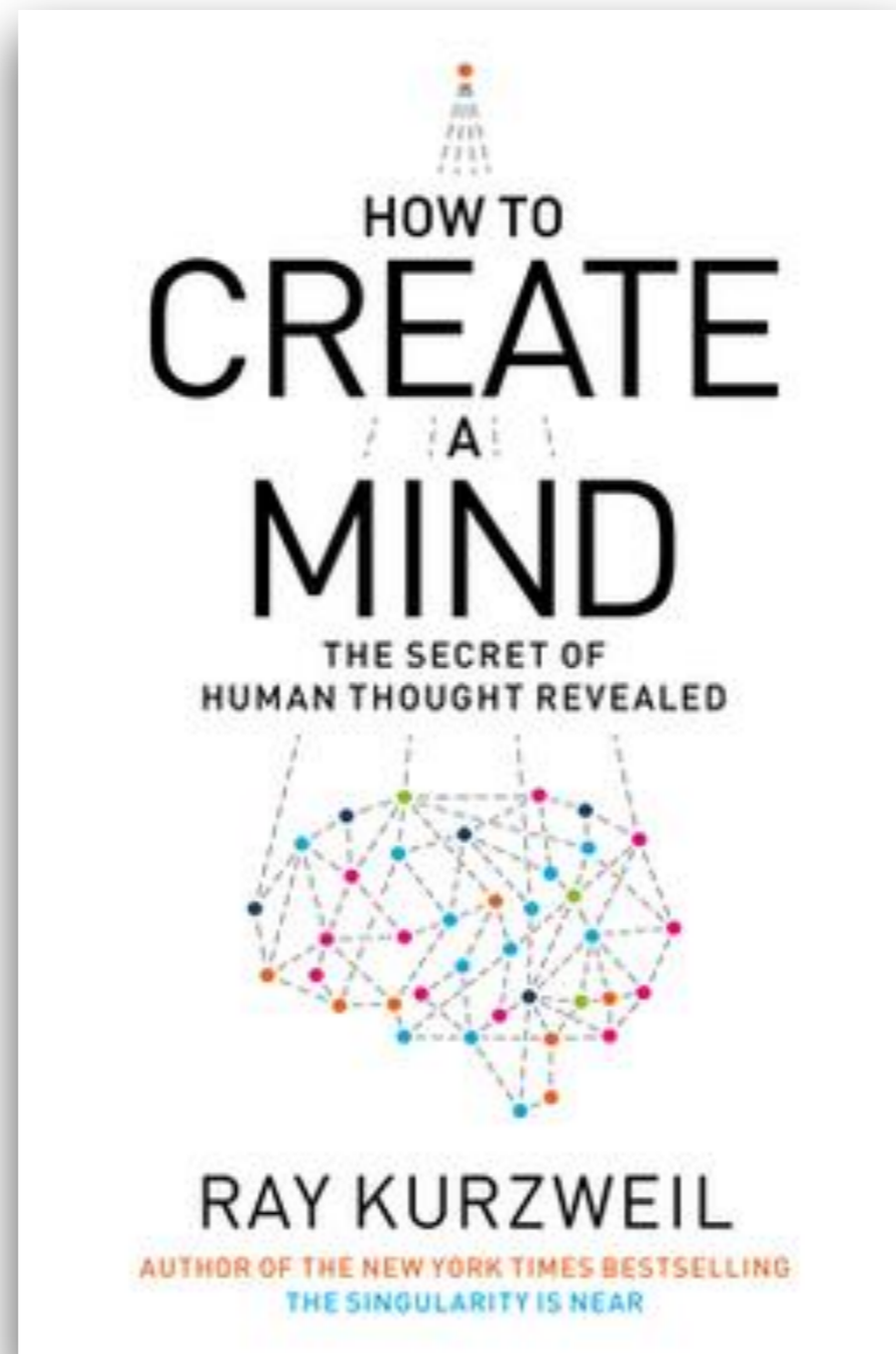
Superintelligence



The Singularity is Near



How to Create a Mind



The Singularity is Near... Or Maybe Not

“panda”
57.7% confidence

+ .007 ×

“nematode”
8.2% confidence

=

“gibbon”
99.3 % confidence

References

- ▶ Francois Chollet. Deep Learning with Python. Manning. 2016.
- ▶ David Foster. Generative Deep Learning. O'Reilly 2019.
- ▶ Arthur I. Miller. The Artist in the Machine. The World of AI-Powered Creativity. MIT Press.
- ▶ OpenAI. Attacking Machine Learning with Adversarial Examples. <https://openai.com/blog/adversarial-example-research/>