RNP-VC

Aula 10: Visualização e Interpretabilidade

Lucas Pereira, Rafael Teixeira, Lucas Assis, Anderson Soares
Instituto de Informática
Universidade Federal de Goiás (UFG)

DEEP LEARNING BRASIL

Sumário

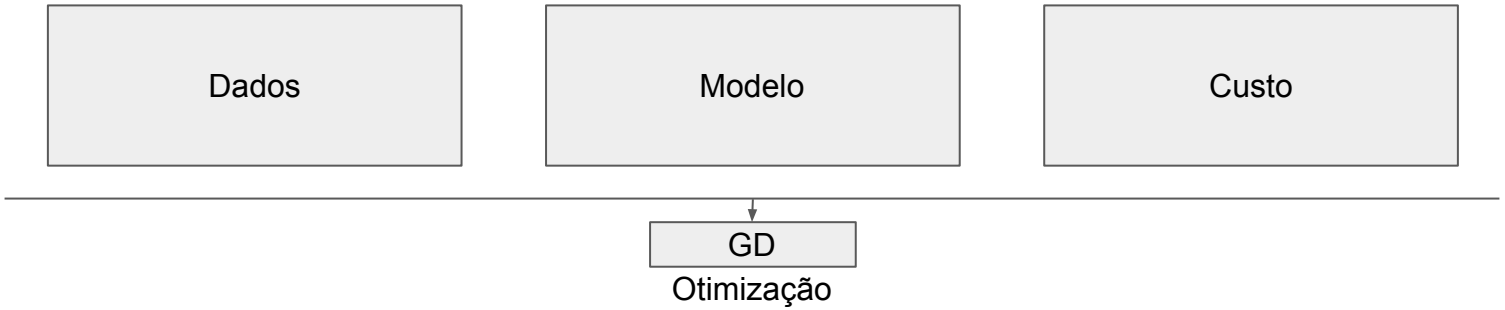
- No último episódio...
- Espaço latente
- Mapas de saliência
- T-CAV
- Pesquisa em interpretabilidade
- Deep Dream
- Style Transfer

No último episódio...

- E se…
  - não supervisionado = não label!

?

?



No último episódio...

- Autoencoders

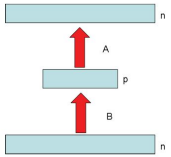
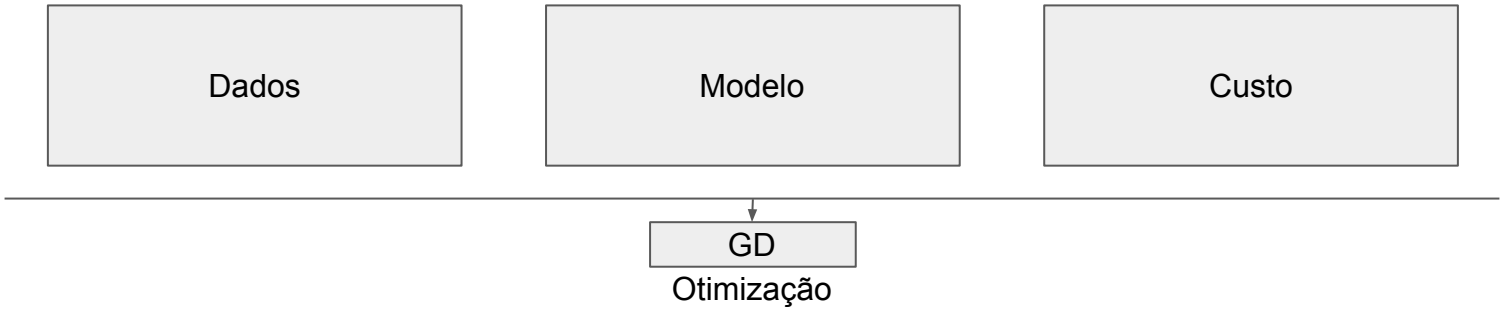


Figure 1: An n/p/n Autoencoder Architecture.

$$\|x - \hat{x}\|^2$$



No último episódio...

- Maximum Likelihood
  - Se o dataset foi coletado, ele tem probabilidade máxima de acontecer.

$$\arg\min_{\theta} \ \text{loss}(\theta, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}) = \frac{1}{n} \sum_{i=1}^{n} -\log p_\theta(\mathbf{x}^{(i)})$$

$$\text{for all } \theta, \quad \sum_{\mathbf{x}} p_\theta(\mathbf{x}) = 1 \quad \text{and} \quad p_\theta(\mathbf{x}) \geq 0 \quad \text{for all } \mathbf{x}$$

No último episódio...

- PixelRNN

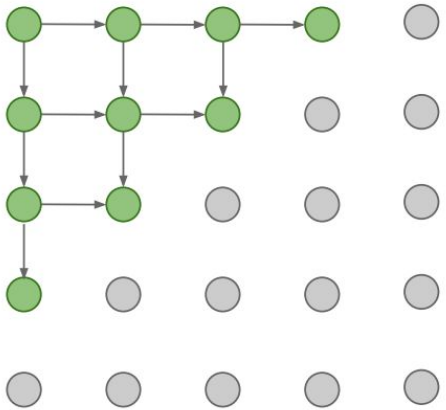
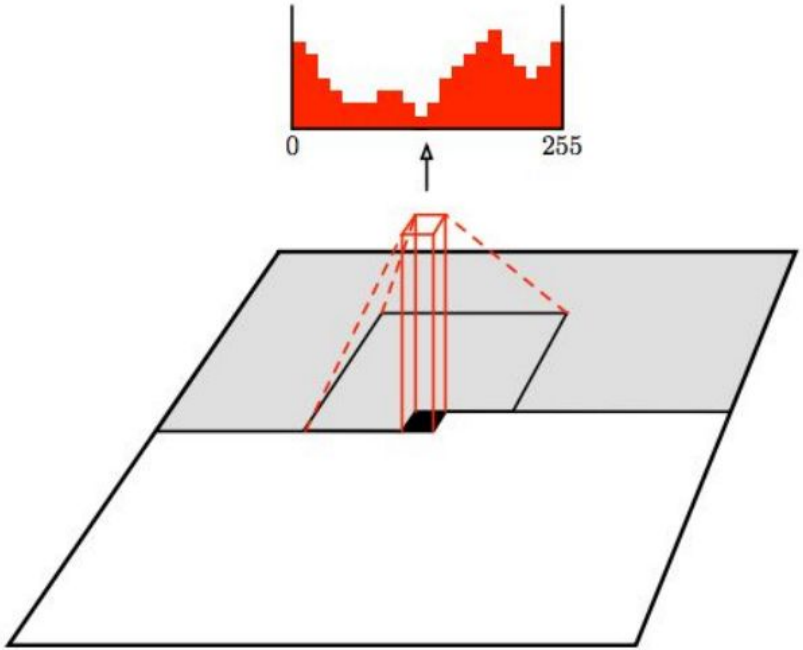


Figure 1. Image completions sampled from a PixelRNN.

No último episódio...

- PixelCNN



No último episódio...

- Variational Lower Bound (Evidence Lower Bound - ELBO)

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Bayes' Rule)}$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad \text{(Multiply by constant)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Logarithms)}$$

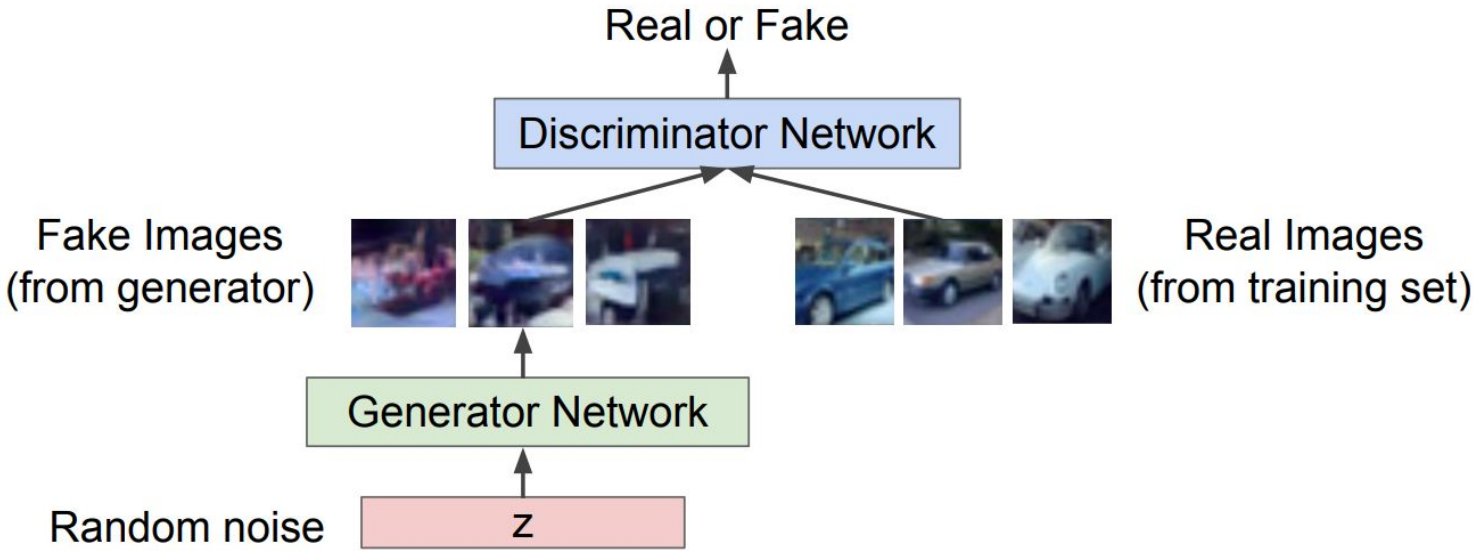
$$= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\geq 0}$$

We want to maximize the data likelihood

Tractable lower bound which we can take gradient of and optimize! ($p_\theta(x|z)$ differentiable, KL term differentiable)

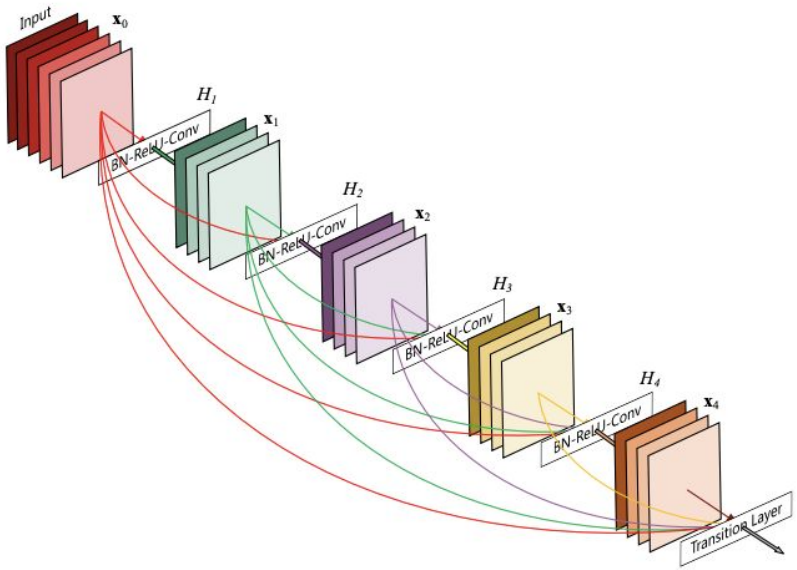
No último episódio...

- Abordagem adversária
  - Otimização segundo teoria dos jogos



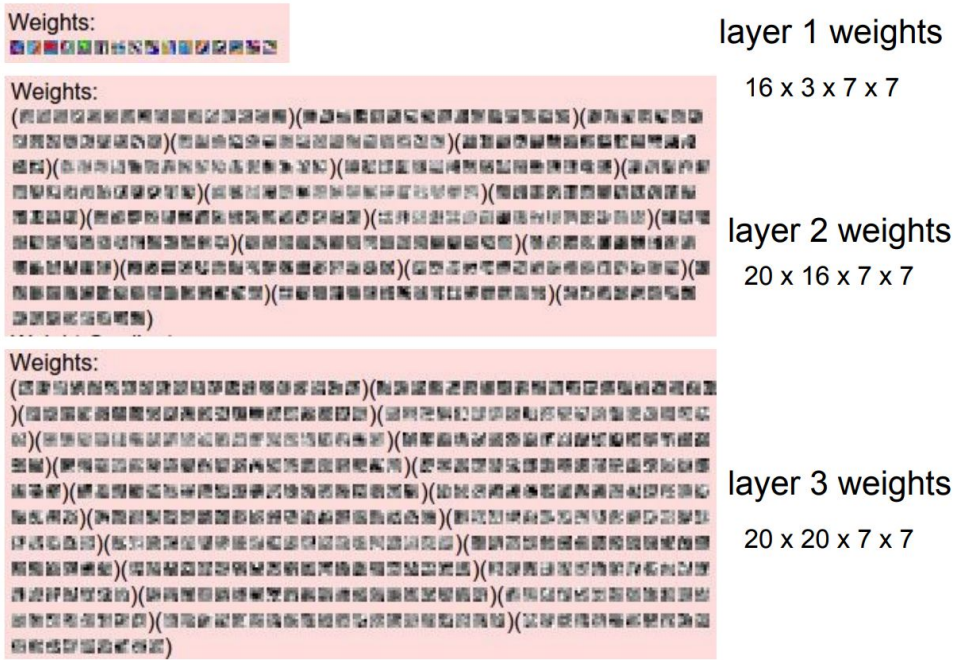
Espaço latente

- O que as camadas internas representam?



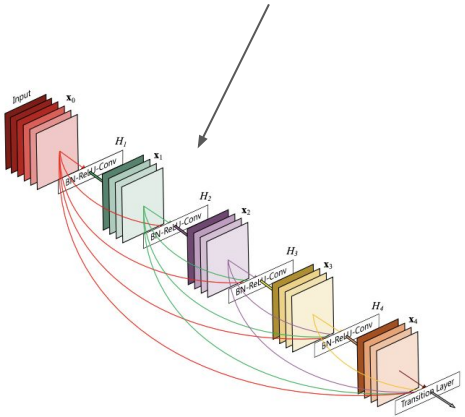
Espaço latente

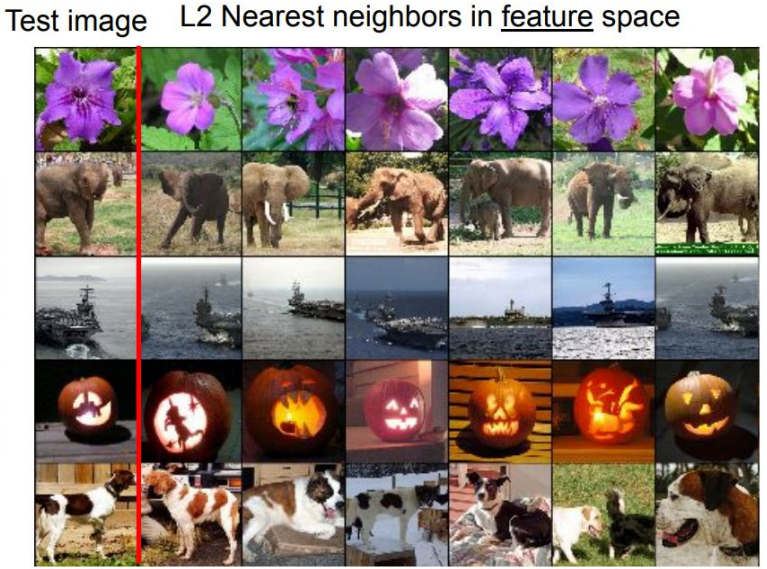
- Visualização dos pesos



Espaço latente

- Visualização de conjuntos de ativações
  - Vizinhos mais próximos no espaço latente





Espaço latente

- Redução de dimensionalidade
  - t-SNE
  - UMAP
  - …

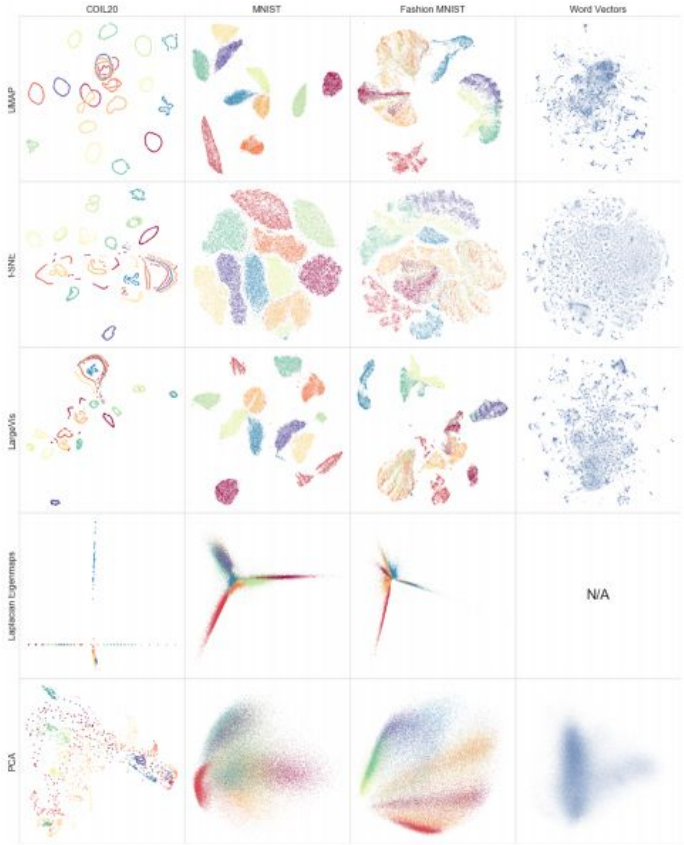


Figure 2: A comparison of several dimension reduction algorithms. We note that UMAP successfully reflects much of the large scale global structure that is well represented by Laplacian Eigenmaps and PCA (particularly for MNIST and Fashion-MNIST), while also preserving the local fine structure similar to t-SNE and LargeVis.

Espaço latente

- Ativações e gradiente ascendente

Understanding Neural Networks Through Deep Visualization

Jason Yosinski YOSINSKI@CS.CORNELL.EDU
Cornell University

Jeff Clune JEFFCLUNE@UWYO.EDU
Anh Nguyen ANGUYEN8@UWYO.EDU
University of Wyoming

Thomas Fuchs FUCHS@CALTECH.EDU
Jet Propulsion Laboratory, California Institute of Technology

Hod Lipson HOD.LIPSON@CORNELL.EDU
Cornell University

Abstract

Recent years have produced great advances in training large, deep neural networks (DNNs), including notable successes in training convolutional neural networks (convnets) to recognize natural images. However, our understanding of how these models work, especially what computations they perform at intermediate layers, has lagged behind. Progress in the field will be further accelerated by the development of better tools for visualizing and interpreting neural nets. We introduce two such tools here. The first is a tool that visualizes the activations produced on each layer of a trained convnet as it processes an image or video (e.g. a live webcam stream). We have found that looking at live activations that change in response to user input helps build valuable intuitions about how convnets work. The second tool enables visualizing features at each layer of a DNN via regularized optimization in image space. Because previous versions of this idea produced less recognizable images, here we introduce several new regularization methods that combine to produce qualitatively clearer, more interpretable visualizations. Both tools are open source and work on a pre-trained convnet with minimal setup.

1. Introduction

The last several years have produced tremendous progress in training powerful, deep neural network models that are approaching and even surpassing human abilities on a variety of challenging machine learning tasks (Taigman et al., 2014; Schroff et al., 2015; Hannun et al., 2014). A flagship example is training deep, convolutional neural networks (CNNs) with supervised learning to classify natural images (Krizhevsky et al., 2012). That area has benefitted from the combined effects of faster computing (e.g. GPUs), better training techniques (e.g. dropout (Hinton et al., 2012)), better activation units (e.g. rectified linear units (Glorot et al., 2011)), and larger labeled datasets (Deng et al., 2009; Lin et al., 2014).

While there has thus been considerable improvements in our knowledge of how to create high-performing architectures and learning algorithms, our understanding of how these large neural models operate has lagged behind. Neural networks have long been known as "black boxes" because it is difficult to understand exactly how any particular, trained neural network functions due to the large number of interacting, non-linear parts. Large modern neural networks are even harder to study because of their size; for example, understanding the widely-used AlexNet DNN involves making sense of the values taken by the 60 million trained network parameters. Understanding what is learned is interesting in its own right, but it is also one key way of further improving models: the intuitions pro-

Espaço latente

- O que maximiza as ativações

  - https://www.youtube.com/watch?v=AgkfIQ4IGaM

Figure 2. A view of the 13×13 activations of the 151st channel on the conv5 layer of a deep neural network trained on ImageNet, a dataset that does not contain a face class, but does contain many images with faces. The channel responds to human and animal faces and is robust to changes in scale, pose, lighting, and context, which can be discerned by a user by actively changing the scene in front of a webcam or by loading static images (e.g. of the lions) and seeing the corresponding response of the unit. Photo of lions via Flickr user arnolouise, licensed under CC BY-NC-SA 2.0.

Espaço latente

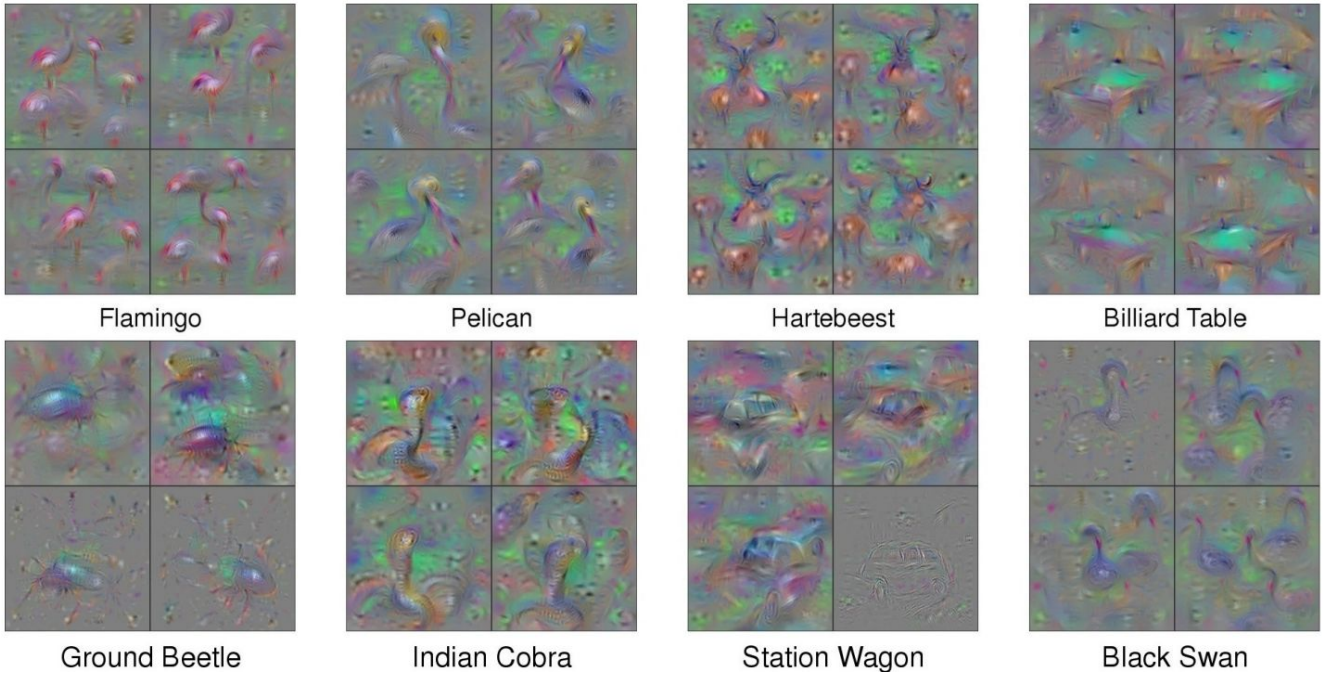
- Gradiente ascendente

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} (a_i(\mathbf{x}) - R_\theta(\mathbf{x})) \tag{1}$$

$$\mathbf{x} \leftarrow r_\theta \left(\mathbf{x} + \eta \frac{\partial a_i}{\partial \mathbf{x}} \right) \tag{2}$$

Espaço latente

- Gradiente ascendente



Espaço latente

- Gradiente ascendente
  - As diferentes “faces” de um neurônio

Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks

Anh Nguyen ANGUYEN8@UWYO.EDU
University of Wyoming

Jason Yosinski YOSINSKI@CS.CORNELL.EDU
Cornell University

Jeff Clune JEFFCLUNE@UWYO.EDU
University of Wyoming

Abstract

We can better understand deep neural networks by identifying which features each of their neurons have learned to detect. To do so, researchers have created Deep Visualization techniques including *activation maximization*, which synthetically generates inputs (e.g. images) that maximally activate each neuron. A limitation of current techniques is that they assume each neuron detects only one type of feature, but we know that neurons can be *multifaceted*, in that they fire in response to many different types of features: for example, a grocery store class neuron must activate either for rows of produce or for a storefront. Previous activation maximization techniques constructed images without regard for the multiple different facets of a neuron, creating inappropriate mixes of colors, parts of objects, scales, orientations, etc. Here, we introduce an algorithm that explicitly uncovers the multiple facets of each neuron by producing a synthetic visualization of each of the types of images that activate a neuron. We also introduce regularization methods that produce state-of-the-art results in terms of the interpretability of images obtained by activation maximization. By separately synthesizing each type of image a neuron fires in response to, the visualizations have more appropriate colors and coherent global structure. Multifaceted feature visualization thus provides a clearer and more comprehensive description of the role of each neuron.

Reconstructions of multiple feature types (facets) recognized by the same “grocery store” neuron

Corresponding example training set images recognized by the same neuron as in the "grocery store" class

Figure 1. **Top:** Visualizations of 8 types of images (feature facets) that activate the same “grocery store” class neuron. **Bottom:** Example training set images that activate the same neuron, and resemble the corresponding synthetic image in the top panel.

1. Introduction

Recently, deep neural networks (DNNs) have demonstrated state-of-the-art—and sometimes human-competitive—results on many pattern recognition tasks, especially vision classification problems (Krizhevsky et al., 2012; Szegedy

Espaço latente

- Gradiente ascendente
  - As diferentes “faces” de um neurônio

Espaço latente

- Gradiente ascendente
  - Modelo generativo

Synthesizing the preferred inputs for neurons in neural networks via deep generator networks

Anh Nguyen
anguyen8@uwyo.edu

Alexey Dosovitskiy
dosovits@cs.uni-freiburg.de

Jason Yosinski
jason@geometric.ai

Thomas Brox
brox@cs.uni-freiburg.de

Jeff Clune
jeffclune@uwyo.edu

Abstract

Deep neural networks (DNNs) have demonstrated state-of-the-art results on many pattern recognition tasks, especially vision classification problems. Understanding the inner workings of such computational brains is both fascinating basic science that is interesting in its own right—similar to why we study the human brain—and will enable researchers to further improve DNNs. One path to understanding how a neural network functions internally is to study what each of its neurons has learned to detect. One such method is called *activation maximization* (AM), which synthesizes an input (e.g. an image) that highly activates a neuron. Here we dramatically improve the qualitative state of the art of activation maximization by harnessing a powerful, learned prior: a deep generator network (DGN). The algorithm (1) generates qualitatively state-of-the-art synthetic images that look almost real, (2) reveals the features learned by each neuron in an interpretable way, (3) generalizes well to new datasets and somewhat well to different network architectures without requiring the prior to be relearned, and (4) can be considered as a high-quality generative method (in this case, by generating novel, creative, interesting, recognizable images).

1 Introduction and Related Work

Understanding how the human brain works has been a long-standing quest in human history. Neuroscientists have discovered neurons in human brains that selectively fire in response to specific, abstract concepts such as Halle Berry or Bill Clinton, shedding light on the question of whether learned neural codes are local vs. distributed [1]. These neurons were identified by finding the *preferred stimuli* (here, images) that highly excite a specific neuron, which was accomplished by showing subjects many different images while recording a target neuron's activation. Such neurons are multifaceted: for example, the "Halle Berry neuron" responds to very different stimuli related to the actress—from pictures of her face, to pictures of her in costume, to the word "Halle Berry" printed as text [1].

Espaço latente

- Gradiente ascendente
  - Modelo generativo

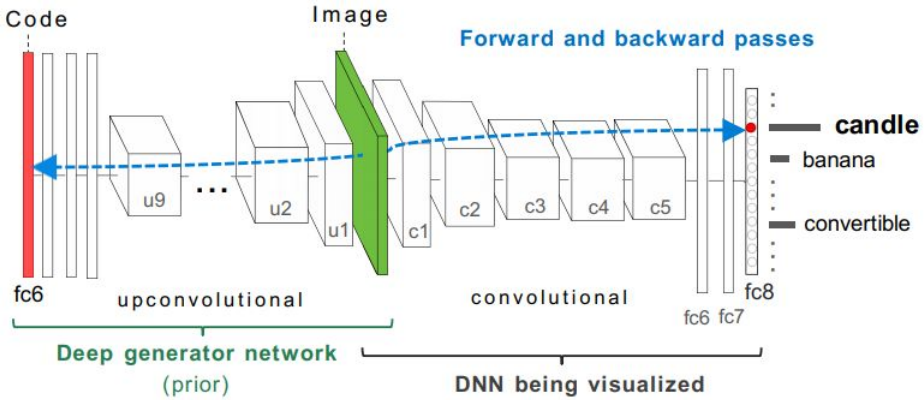


Figure 2: To synthesize a preferred input for a target neuron h (e.g. the "candle" class output neuron), we optimize the hidden code input (red bar) of a deep image generator network (DGN) to produce an image that highly activates h. In the example shown, the DGN is a network trained to invert the feature representations of layer fc6 of CaffeNet. The target DNN being visualized can be a different network (with a different architecture and or trained on different data). The gradient information (blue-dashed line) flows from the layer containing h in the target DNN (here, layer fc8) all the way through the image back to the input code layer of the DGN. Note that both the DGN and target DNN being visualized have fixed parameters, and optimization only changes the DGN input code (red).

Espaço latente

- Gradiente ascendente
  - Modelo generativo

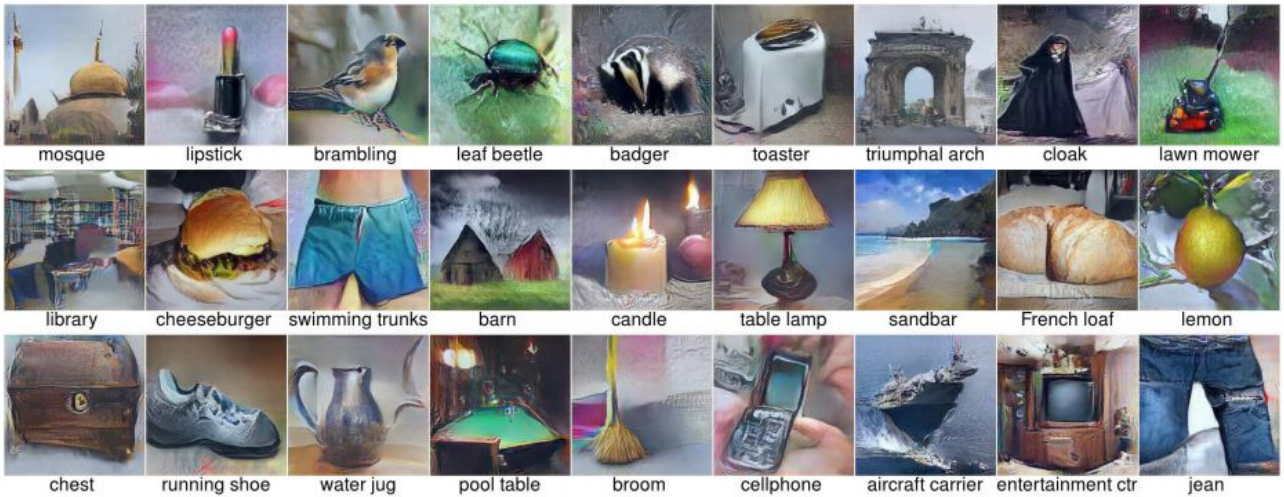
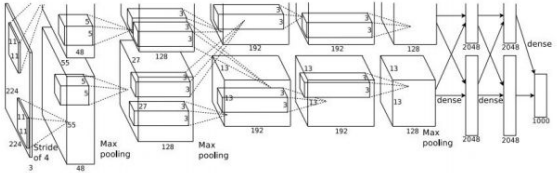
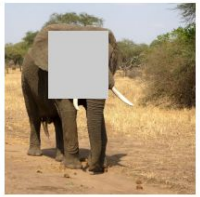


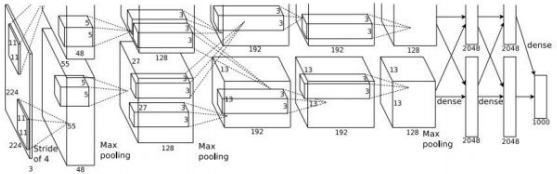
Figure 1: Images synthesized from scratch to highly activate output neurons in the CaffeNet deep neural network, which has learned to classify different types of ImageNet images. More examples showcasing the sample quality in comparison with real images are in Fig. S23.

Mapas de saliência

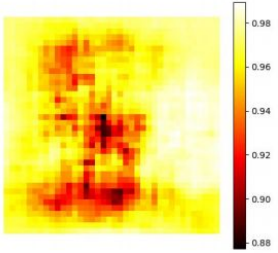
- Oclusão



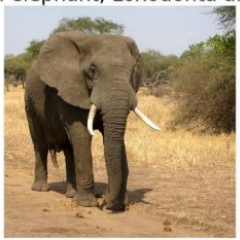


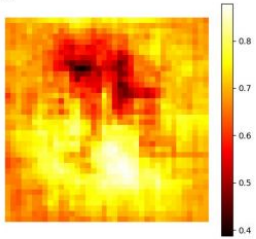


schooner

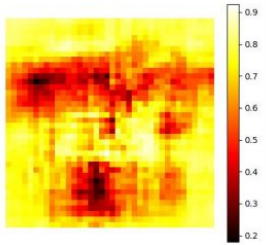


African elephant, Loxodonta africana





go-kart



Mapas de saliência

- Gradiente

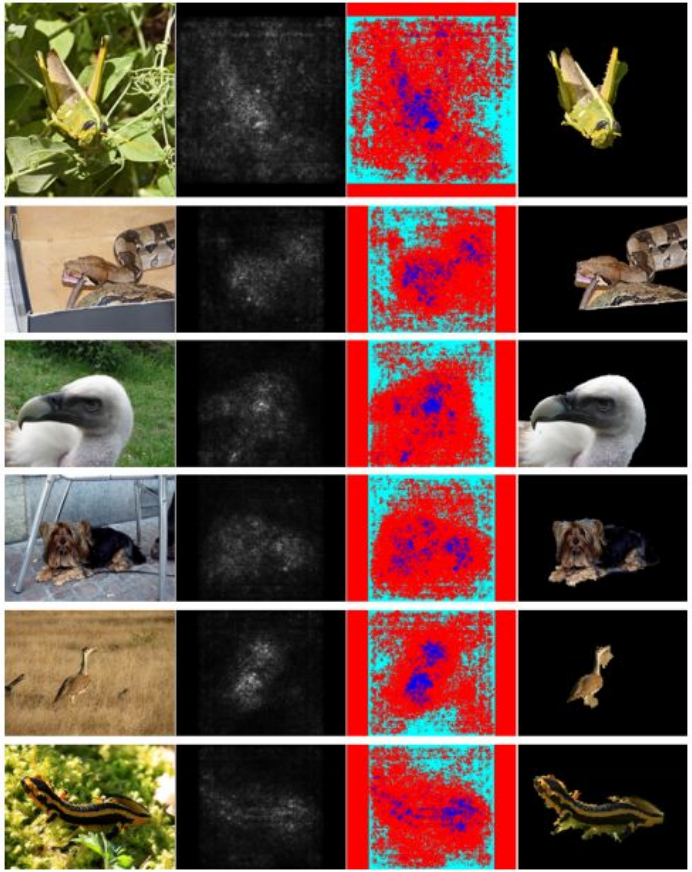


Figure 3: **Weakly supervised object segmentation using ConvNets (Sect. 3.2).** *Left:* images from the test set of ILSVRC-2013. *Left-middle:* the corresponding saliency maps for the top-1 predicted class. *Right-middle:* thresholded saliency maps: blue shows the areas used to compute the foreground colour model, cyan – background colour model, pixels shown in red are not used for colour model estimation. *Right:* the resulting foreground segmentation masks.

Mapas de saliência

- “Deconvoluções”

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.

est 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

Mapas de saliência

- “Deconvoluções”

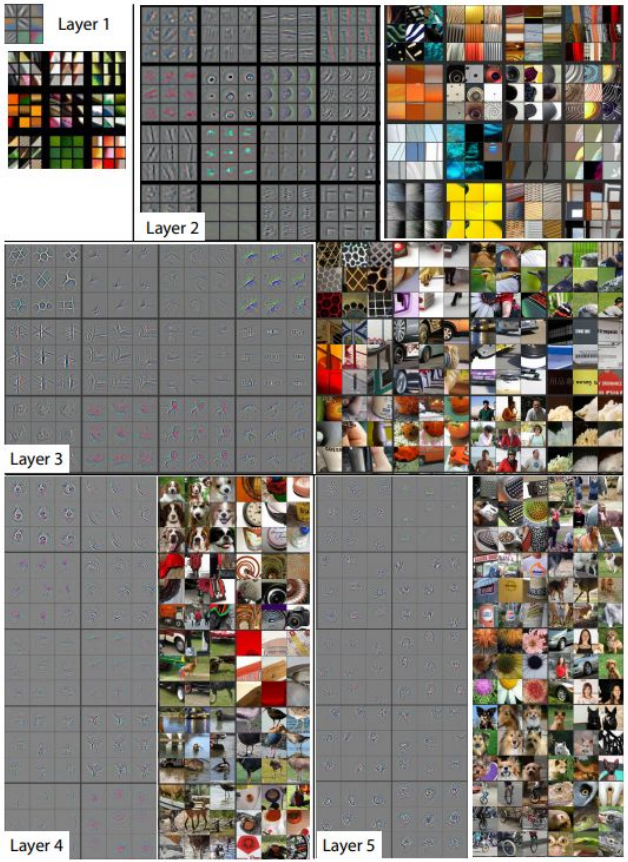
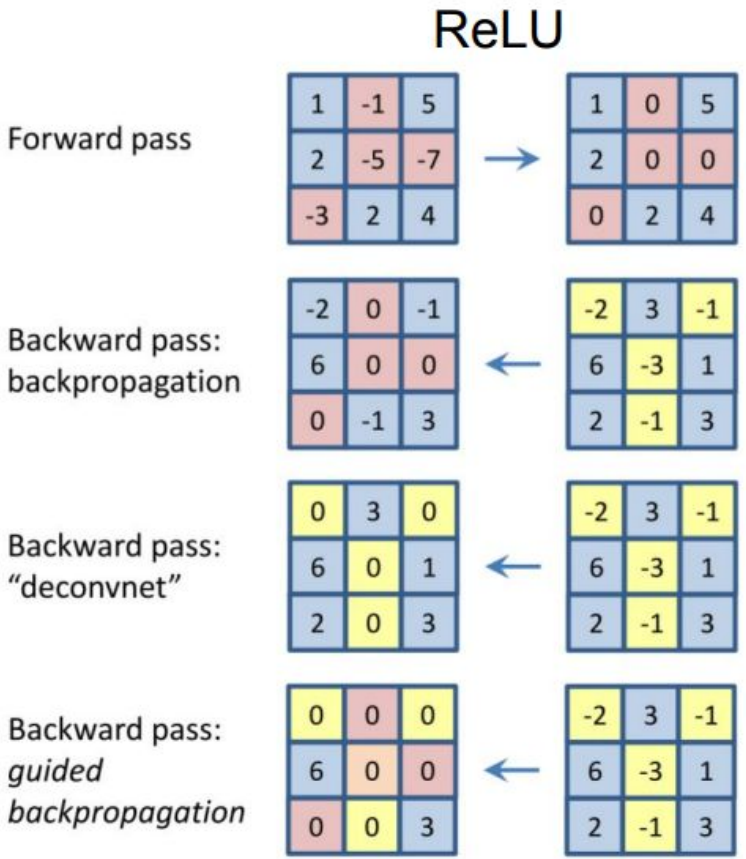


Figure 2. Visualization of features in a fully trained model. For layers 2-5 we show the top 9 activations in a random subset of feature maps across the validation data, projected down to pixel space using our deconvolutional network approach. Our reconstructions are *not* samples from the model: they 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.

Mapas de saliência

- “Deconvoluções”
  - Guided backprop



Mapas de saliência

- Sanity Checks for Saliency Maps (2018)

Sanity Checks for Saliency Maps

Julius Adebayo[*], **Justin Gilmer**[♯], **Michael Muelly**[♯], **Ian Goodfellow**[♯], **Moritz Hardt**[♯†], **Been Kim**[♯]
juliusad@mit.edu, {gilmer,muelly,goodfellow,mrtz,beenkim}@google.com
[♯]Google Brain
[†]University of California Berkeley

Abstract

Saliency methods have emerged as a popular tool to highlight features in an input deemed relevant for the prediction of a learned model. Several saliency methods have been proposed, often guided by visual appeal on image data. In this work, we propose an actionable methodology to evaluate what kinds of explanations a given method can and cannot provide. We find that reliance, solely, on visual assessment can be misleading. Through extensive experiments we show that some existing saliency methods are independent both of the model and of the data generating process. Consequently, methods that fail the proposed tests are inadequate for tasks that are sensitive to either data or model, such as, finding outliers in the data, explaining the relationship between inputs and outputs that the model learned, and debugging the model. We interpret our findings through an analogy with edge detection in images, a technique that requires neither training data nor model. Theory in the case of a linear model and a single-layer convolutional neural network supports our experimental findings[2].

Mapas de saliência

- Sanity Checks for Saliency Maps (2018)

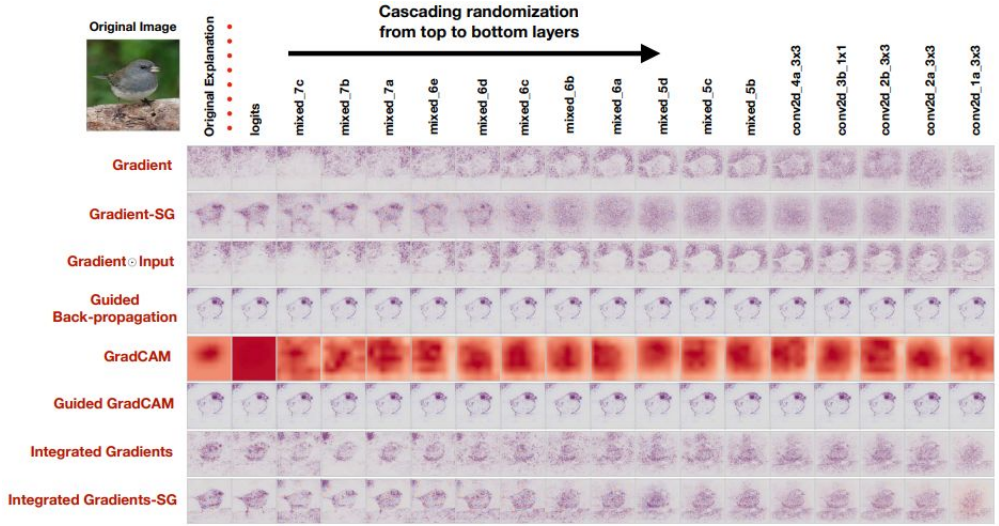


Figure 2: **Cascading randomization on Inception v3 (ImageNet).** Figure shows the original explanations (first column) for the Junco bird. Progression from left to right indicates complete randomization of network weights (and other trainable variables) up to that 'block' inclusive. We show images for 17 blocks of randomization. Coordinate (Gradient, mixed_7b) shows the gradient explanation for the network in which the top layers starting from Logits up to mixed_7b have been reinitialized. The last column corresponds to a network with completely reinitialized weights.

T-CAV

- **T**esting with **C**oncept **A**ctivation **V**ectors

Interpretability Beyond Feature Attribution:
Quantitative Testing with Concept Activation Vectors (TCAV)

Been Kim Martin Wattenberg Justin Gilmer Carrie Cai James Wexler
Fernanda Viegas Rory Sayres

Abstract

The interpretation of deep learning models is a challenge due to their size, complexity, and often opaque internal state. In addition, many systems, such as image classifiers, operate on low-level features rather than high-level concepts. To address these challenges, we introduce Concept Activation Vectors (CAVs), which provide an interpretation of a neural net's internal state in terms of human-friendly concepts. The key idea is to view the high-dimensional internal state of a neural net as an aid, not an obstacle. We show how to use CAVs as part of a technique, Testing with CAVs (TCAV), that uses directional derivatives to quantify the degree to which a user-defined concept is important to a classification result–for example, how sensitive a prediction of zebra is to the presence of stripes. Using the domain of image classification as a testing ground, we describe how CAVs may be used to explore hypotheses and generate insights for a standard image classification network as well as a medical application.

A key difficulty, however, is that most ML models operate on features, such as pixel values, that do not correspond to high-level concepts that humans easily understand. Furthermore, a model's internal values (e.g., neural activations) can seem incomprehensible. We can express this difficulty mathematically, viewing the state of an ML model as a vector space E_m spanned by basis vectors e_m which correspond to data such as input features and neural activations. Humans work in a different vector space E_h spanned by implicit vectors e_h corresponding to an unknown set of human-interpretable concepts.

From this standpoint, an "interpretation" of an ML model can be seen as function $g : E_m \to E_h$. When g is linear, we call it a **linear interpretability**. In general, an interpretability function g need not be perfect (Doshi-Velez, 2017); it may fail to explain some aspects of its input domain E_m and it will unavoidably not cover all possible human concepts in E_h.

In this work, the high-level concepts of E_h are defined using sets of example input data for the ML model under inspection. For instance, to define concept 'curly', a set of hairstyles and texture images can be used. Note the concepts

T-CAV

- **T**esting with **C**oncept **A**ctivation **V**ectors

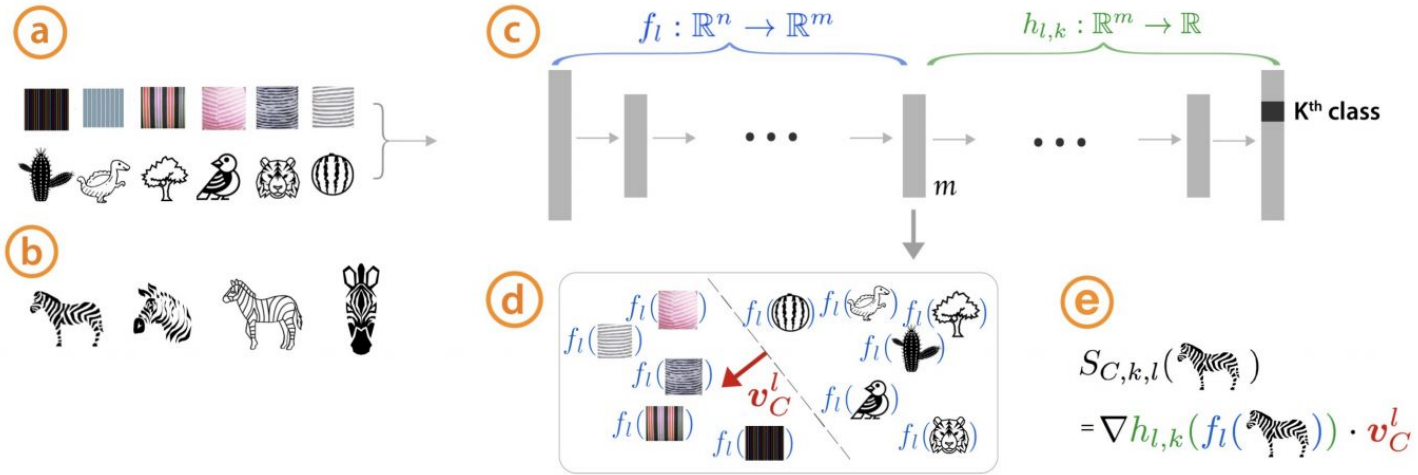


Figure 1. **Testing with Concept Activation Vectors:** Given a user-defined set of examples for a concept (e.g., 'striped'), and random examples ⓐ, labeled training-data examples for the studied class (zebras) ⓑ, and a trained network ⓒ, TCAV can quantify the model's sensitivity to the concept for that class. CAVs are learned by training a linear classifier to distinguish between the activations produced by a concept's examples and examples in any layer ⓓ. The CAV is the vector orthogonal to the classification boundary (v_C^l, red arrow). For the class of interest (zebras), TCAV uses the directional derivative $S_{C,k,l}(\boldsymbol{x})$ to quantify conceptual sensitivity ⓔ.

Pesquisa em interpretabilidade

Explaining Explanations: An Overview of Interpretability of Machine Learning

Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter and Lalana Kagal
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
{lgilpin, davidbau, bzy, abajwa, specter, lkagal}@ mit.edu

***Abstract*—There has recently been a surge of work in explanatory artificial intelligence (XAI). This research area tackles the important problem that complex machines and algorithms often cannot provide insights into their behavior and thought processes. XAI allows users and parts of the internal system to be more transparent, providing explanations of their decisions in some level of detail. These explanations are important to ensure algorithmic fairness, identify potential bias/problems in the training data, and to ensure that the algorithms perform as expected. However, explanations produced by these systems is neither standardized nor systematically assessed. In an effort to create best practices and identify open challenges, we describe foundational concepts of explainability and show how they can be used to classify existing literature. We discuss why current approaches to explanatory methods especially for deep neural networks are insufficient. Finally, based on our survey, we conclude with suggested future research directions for explanatory artificial intelligence.**

As a first step towards creating explanation mechanisms, there is a new line of research in interpretability, loosely defined as the science of comprehending what a model did (or might have done). Interpretable models and learning methods show great promise; examples include visual cues to find the "focus" of deep neural networks in image recognition and proxy methods to simplify the output of complex systems. However, there is ample room for improvement, since identifying dominant classifiers and simplifying the problem space does not solve all possible problems associated with understanding opaque models.

We take the stance that interpretability alone is insufficient. In order for humans to trust black-box methods, we need *explainability* – models that are able to summarize the reasons for neural network behavior, gain the trust of users, or produce insights about the causes of their decisions. While interpretabil-

Pesquisa em interpretabilidade

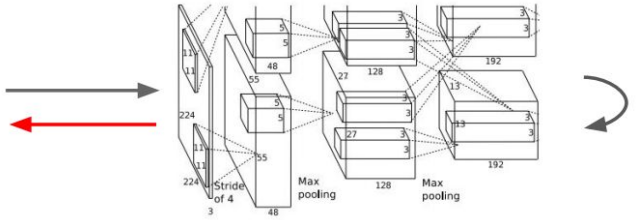
Towards A Rigorous Science of Interpretable Machine Learning

Finale Doshi-Velez* and Been Kim*

From autonomous cars and adaptive email-filters to predictive policing systems, machine learning (ML) systems are increasingly ubiquitous; they outperform humans on specific tasks [Mnih et al., 2013, Silver et al., 2016, Hamill, 2017] and often guide processes of human understanding and decisions [Carton et al., 2016, Doshi-Velez et al., 2014]. The deployment of ML systems in complex applications has led to a surge of interest in systems optimized not only for expected task performance but also other important criteria such as safety [Otte, 2013, Amodei et al., 2016, Varshney and Alemzadeh, 2016], nondiscrimination [Bostrom and Yudkowsky, 2014, Ruggieri et al., 2010, Hardt et al., 2016], avoiding technical debt [Sculley et al., 2015], or providing the right to explanation [Goodman and Flaxman, 2016]. For ML systems to be used safely, satisfying these auxiliary criteria is critical. However, unlike measures of performance such as accuracy, these criteria often cannot be completely quantified. For example, we might not be able to enumerate all unit tests required for the safe operation of a semi-autonomous car or all confounds that might cause a credit scoring system to be discriminatory. In such cases, a popular fallback is the criterion of *interpretability*: if the system can *explain* its reasoning, we then can verify whether that reasoning is sound with respect to these auxiliary criteria.

DeepDream

- Ampliar os features existentes no input



Choose an image and a layer in a CNN; repeat:

1. Forward: compute activations at chosen layer
2. Set gradient of chosen layer *equal to its activation*
3. Backward: Compute gradient on image
4. Update image

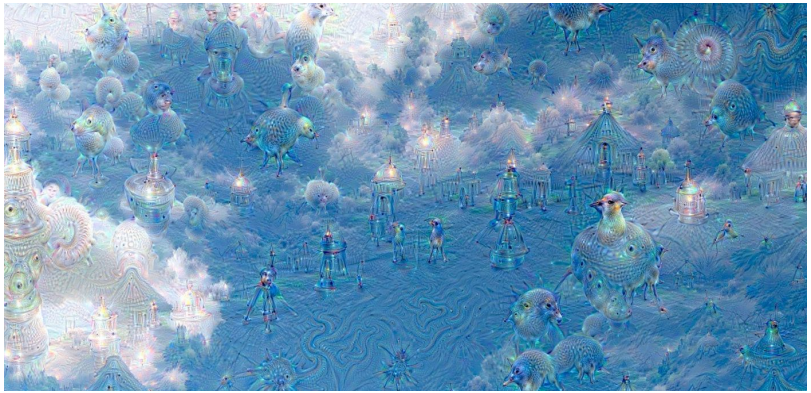
Equivalent to:

$$I^* = \arg\max_I \sum_i f_i(I)^2$$

Mordvintsev, Olah, and Tyka, "Inceptionism: Going Deeper into Neural Networks", Google Research Blog. Images are licensed under CC-BY 4.0

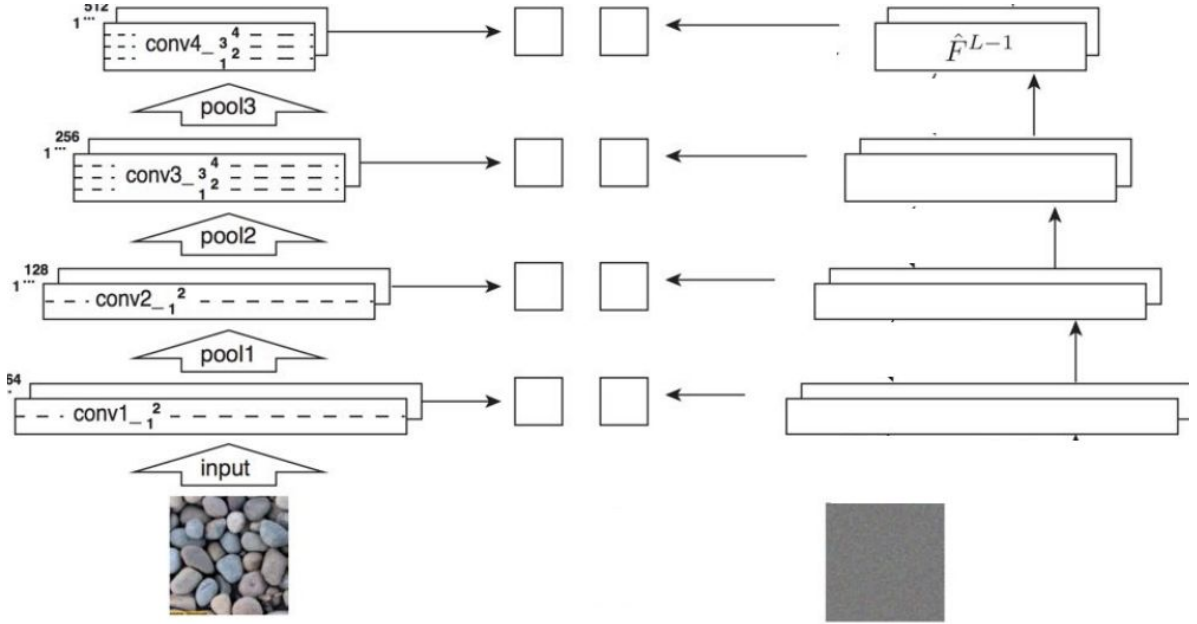
DeepDream

- Ampliar os features existentes no input



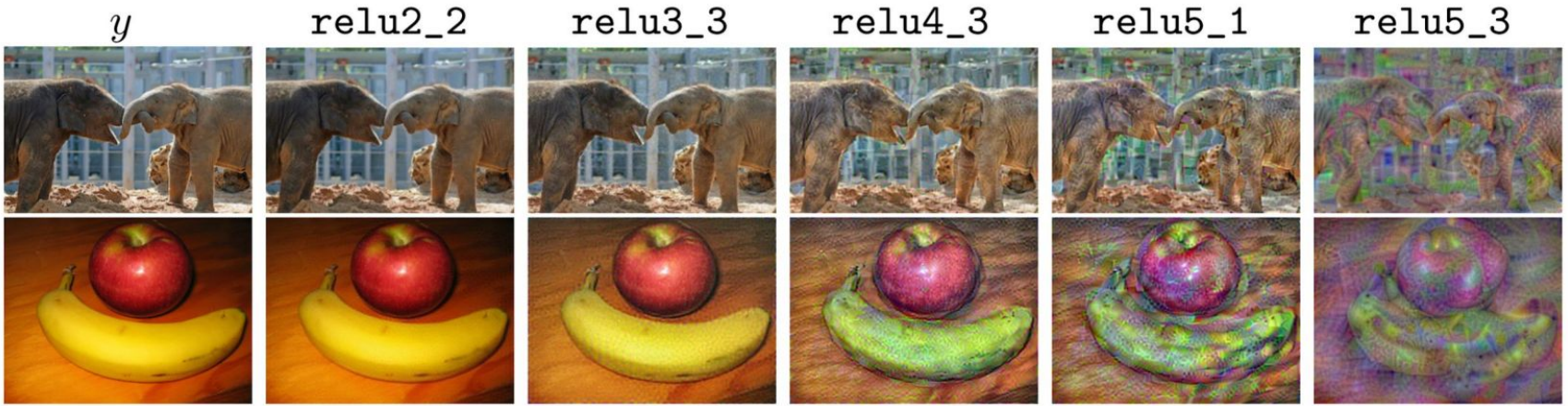
Style Transfer

- Reconstrução a partir dos features (Feature Inversion)



Style Transfer

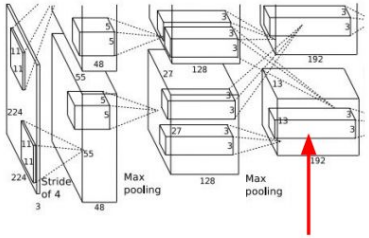
- Reconstrução a partir dos features (Feature Inversion)
  - conteúdo da imagem preservado

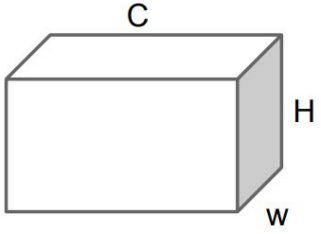


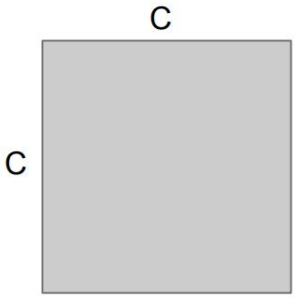
Style Transfer

- Síntese de textura: Matriz Gram

This image is in the public domain.







Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

Outer product of two C-dimensional vectors gives C x C matrix measuring co-occurrence

Average over all HW pairs of vectors, giving **Gram matrix** of shape C x C

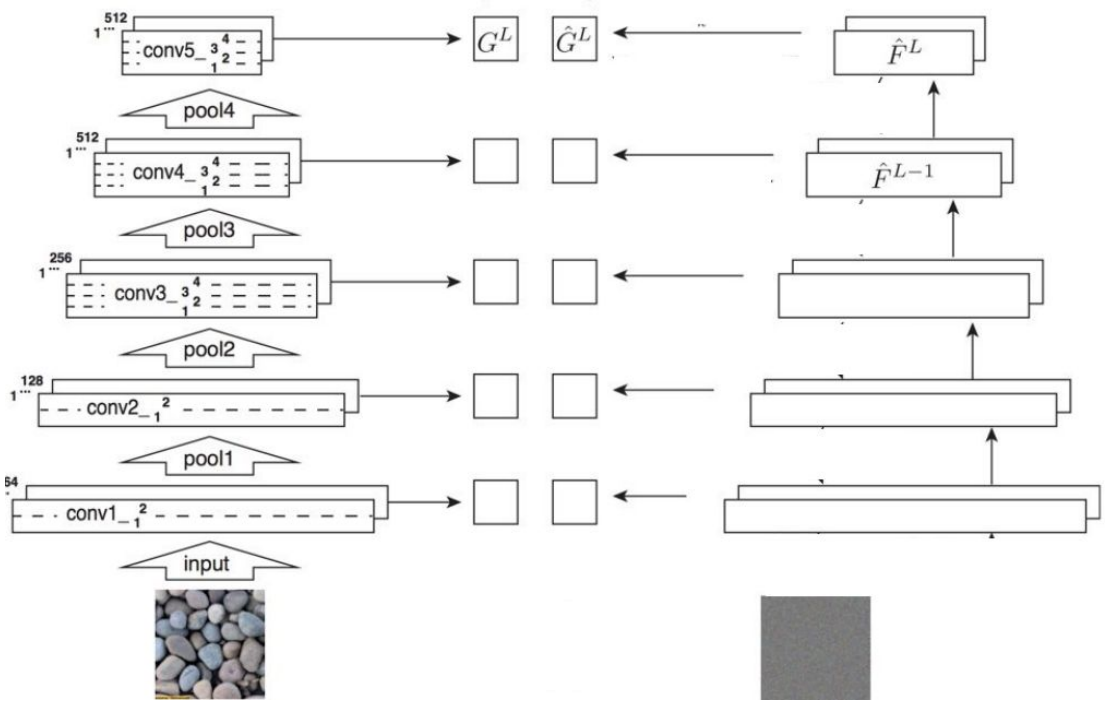
Efficient to compute; reshape features from

C x H x W to =C x HW

then compute $G = FF^T$

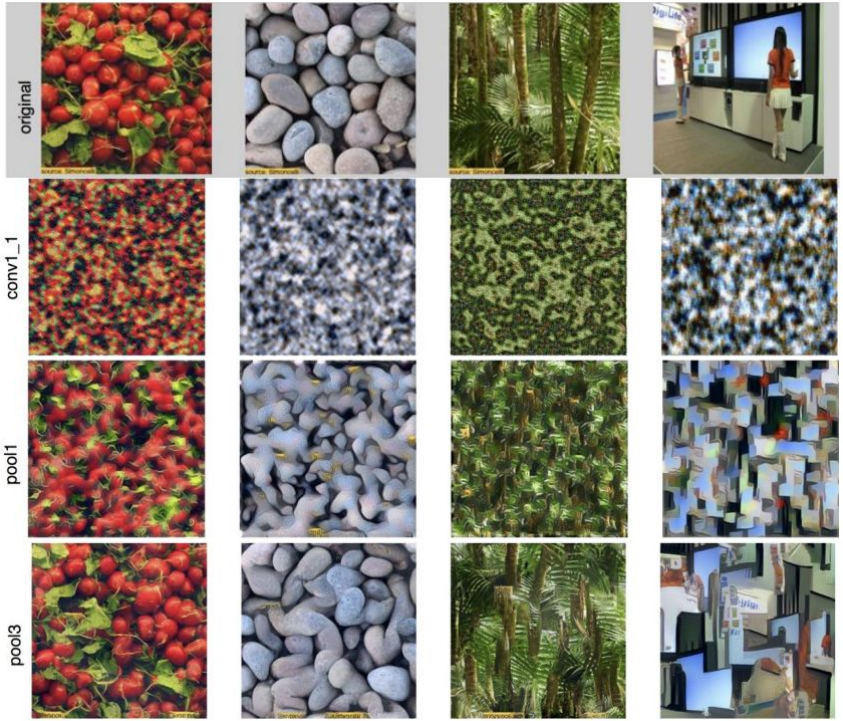
Style Transfer

- Síntese de textura: Matriz Gram



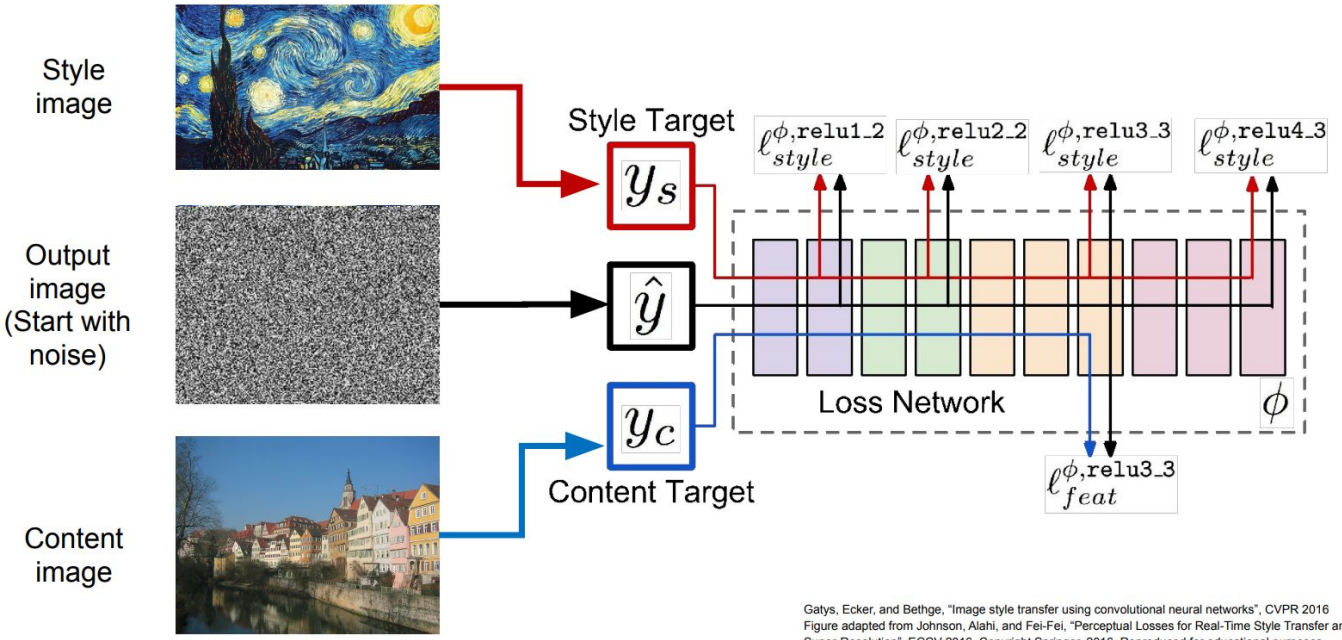
Style Transfer

- Síntese de textura: Matriz Gram



Style Transfer

- Textura + conteúdo



Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016
Figure adapted from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.

Style Transfer

- Textura + conteúdo

Johnson, Alahi, and Fei-Fei, “Perceptual Losses for Real-Time Style Transfer and Super-Resolution”, ECCV 2016

Ulyanov et al, “Texture Networks: Feed-forward Synthesis of Textures and Stylized Images”, ICML 2016
Ulyanov et al, “Instance Normalization: The Missing Ingredient for Fast Stylization”, arXiv 2016

Dumoulin, Shlens, and Kudlur, “A Learned Representation for Artistic Style”, ICLR 2017.