Economics 2355: Deep Visualization

Melissa Dell

Harvard University

February 2021

Outline

CNN Visualization

Most Basic Visualization Approaches

Gradient Based Ascent

Deep Dream

Visualization

We are going to talk briefly about visualization:

- It helps build intuition for what CNNs are doing
- Computer vision researchers have phenomenal visualization skills (check out *Distill*; economists definitely have a lot to learn from them)
- We will later return to visualization in the context of NLP

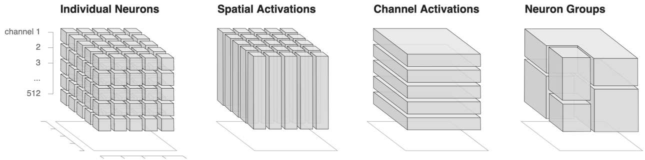
Visualization Questions

1. What does the network detect?
2. How does the network develop its interpretation of images?

We know how to interpret the inputs (values for the red, green, and blue color channels for every pixel in the image) and the outputs (i.e. class labels and probabilities, coordinates if detecting layouts). Thinking about the hidden layers is harder.

Interpreting the Hidden Layers

Recall that we can think of the weights learned by each layer as a three dimensional cube:



In addition to naturally slicing a hidden layer's cube of activations into neurons, spatial locations, or channels, we can also consider more arbitrary groupings of locations and channels.

Olah, Chris, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. "The building blocks of interpretability." Distill 3, no. 3 (2018).

We can use visualization to better understand what different neurons, spatial positions, channels, or more arbitrary regions of the network look for in an image.

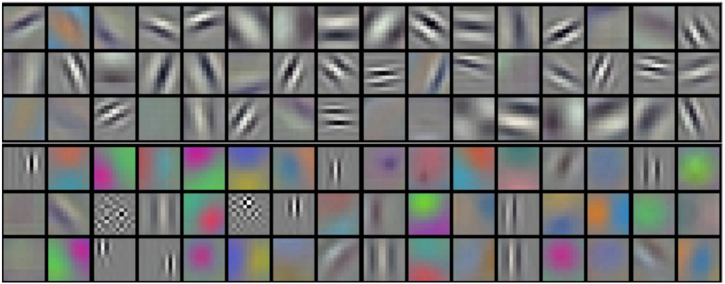
How to Visualize

- It is hard to make sense of the raw building blocks of the hidden layers: high-dimensional numerical vectors
- We want to symbolize what the network learns. Since the network is learning visual representations, it makes the most sense to summarize this using visual representations
- If we were studying audio, we would want to understand what the network is doing using audio "visualizations". When studying text, visualizations like word clouds are common

Outline

Visualizing Weights from First Layer

The following image is from the AlexNet paper and visualizes the filters applied to the input images



Krizhevsky et al., 2012

It is easy to visualize weights from the first layer because they have three channels, like the input images

Harder to Think About Hidden Layers

As humans we visualize things in 3 channels. We don't know how to think in 256 channel space

layer 1 weights

16 x 3 x 7 x 7

layer 2 weights

20 x 16 x 7 x 7

layer 3 weights

20 x 20 x 7 x 7

Stanford CS 231n

L2 Distance in Features Space is Meaningful

An insight, also from the AlexNet paper

Krizhevsky et al., 2012

Images near each other in the last hidden layer (L2 distance between their features vectors) convey similar concepts, even when the pixel distance of the original images is not close

Dimensionality Reduction Allows us to Visualize this Distance

- Our final features vector in a CNN model is a high dimensional object
- Can we project that to two dimensions, in a way that meaningfully preserves the semantic concepts that the high dimensional space captures

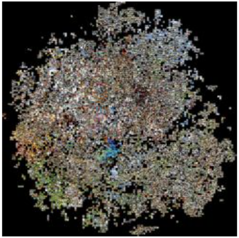
Dimensionality Reduction

- We will see a demo that uses a method called tSNE to project the 4096-dimensional fc7 CNN features vectors from an ImageNet classifier into 2D space (in practice, there is now a more modern method called UMAP that is preferred for dimensionality reduction)
- We can visualize the original images at their location in this projected two dimensional space

Visualization in projected fc7 space

1000x1000 (240KB)

4000x4000 (3.2MB)



6000x6000 (6.6MB)

https://cs.stanford.edu/people/karpathy/cnnembed/

Visualization in projected fc7 space

https://cs.stanford.edu/people/karpathy/cnnembed/

Outline

Inside the black box

- When we train a network, we choose the weights to minimize a cost function
- Suppose instead that we've already trained the weights. Instead, we want to train the input image to maximize certain neurons in the network
- For example, you could input an image of a cat, see which neurons light up the most, and change the input image to maximize those neurons

Gradient Based Ascent

We can visualize CNN features using a method called gradient based ascent. The intuition is to generate an image that maximally activates a neuron value, $f(I)$

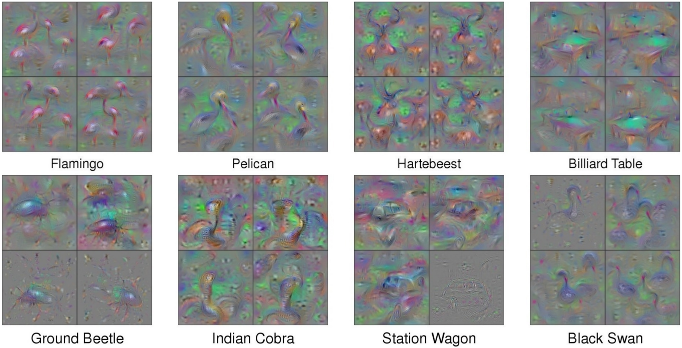
$$I^* = argmax_I f(I) + R(I) \tag{1}$$

$R(I)$ is a regularization term (i.e. the L2 norm of an image). In practice, there are also some additional tweaks during training: Gaussian blur, clip pixels with small values and gradients to 0

Gradient Based Ascent

- Start with an image initialized to zeroes. Pass the image through the CNN to compute the current values
- Then backprop to get the gradient of the neuron value with respect to the image pixels
- Update the image, then repeat

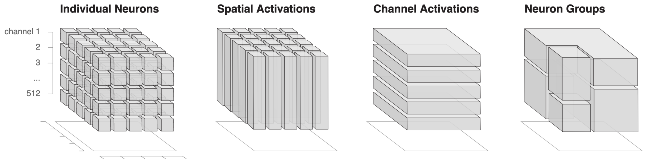
Gradient Based Ascent



Yosinski, Jason, Jeff Clune, Thomas Fuchs, and Hod Lipson. "Understanding neural networks through deep visualization." In In ICML Workshop on Deep Learning 2015.

Interpretability

Subsequent work has maximized not just individual neurons, but also spatial locations, channels, or more arbitrary regions.



In addition to naturally slicing a hidden layer's cube of activations into neurons, spatial locations, or channels, we can also consider more arbitrary groupings of locations and channels.

Olah, Chris, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. "The building blocks of interpretability." Distill 3, no. 3 (2018).

I highly recommend checking out the interactive visualizations in "The Building Blocks of Interpretability." It integrates abstractions for visualization with rich interfaces, marrying the interpretability lit and the human-computer interaction lit

Outline

Deep Dream Algorithm

Another fun and related visualization tool is Google Deep Dream

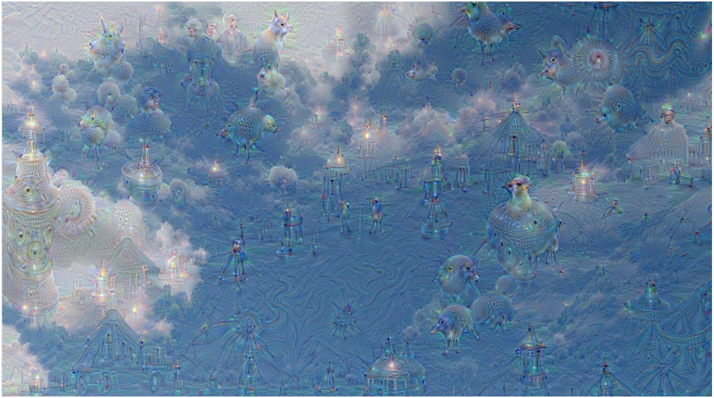
- Compute (chosen) activations at chosen layer
- Set gradient of chosen neurons equal to their activations
- Compute gradient on image
- Update image

Amplifies existing features that were detected by the network in the image

Deep Dream

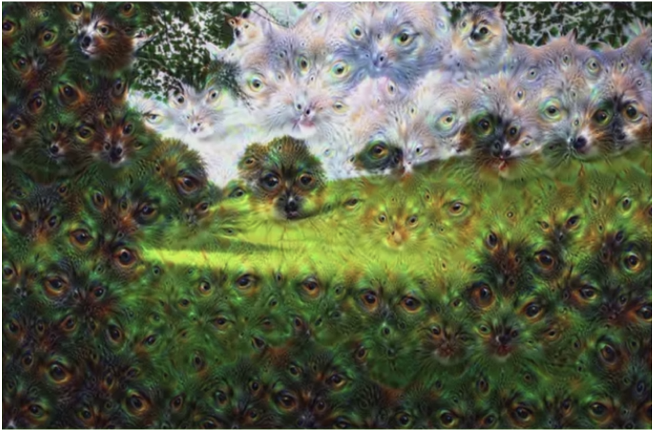
https://github.com/google/deepdream

Deep Dream



https://github.com/google/deepdream

Different layers will hallucinate different features



Deep Dream by Computerphile https://www.youtube.com/watch?v=BsSmBPmPeYQ

Different layers will hallucinate different features

Deep Dream by Computerphile `https://www.youtube.com/watch?v=BsSmBPmPeYQ`

Funny Animals

"Admiral Dog!"

"The Pig-Snail"

"The Camel-Bird"

"The Dog-Fish"

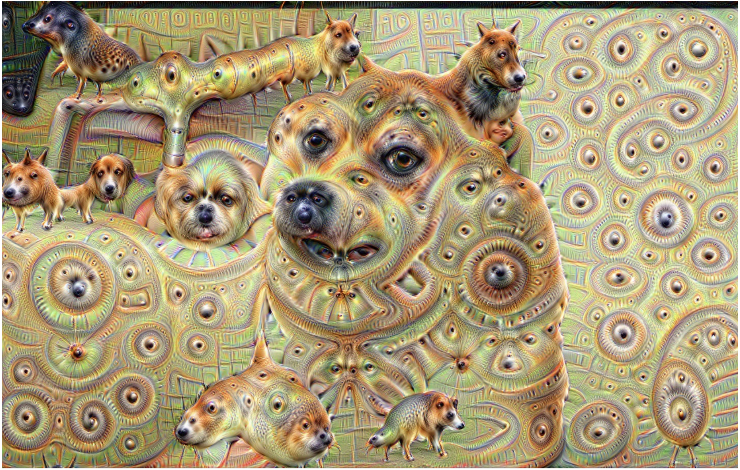
https://github.com/google/deepdream

Something like 120 of the 1,000 classes in ImageNet are different dog breeds, so it is not surprising that Deep Dream hallucinates dogs

Deep Dream Dogs

https://github.com/google/deepdream

Deep Dream Dogs



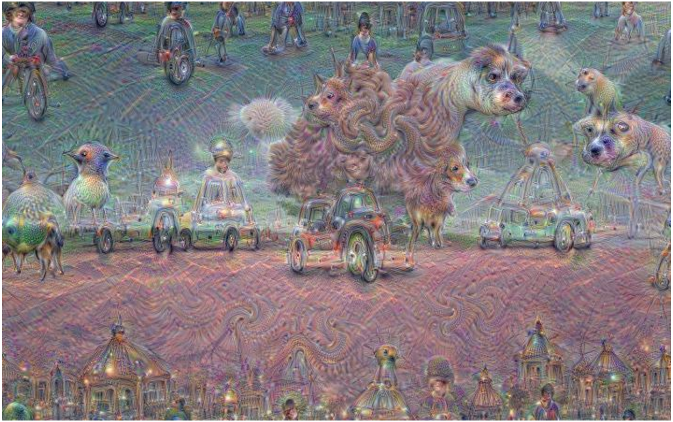
https://github.com/google/deepdream

Adding multi-scale processing can lead to some really amazing images.

Deep Dream

https://github.com/google/deepdream

Deep Dream



https://github.com/google/deepdream

Deep Dream on MIT Places

https://github.com/google/deepdream

Deep Dream on MIT Places

https://github.com/google/deepdream