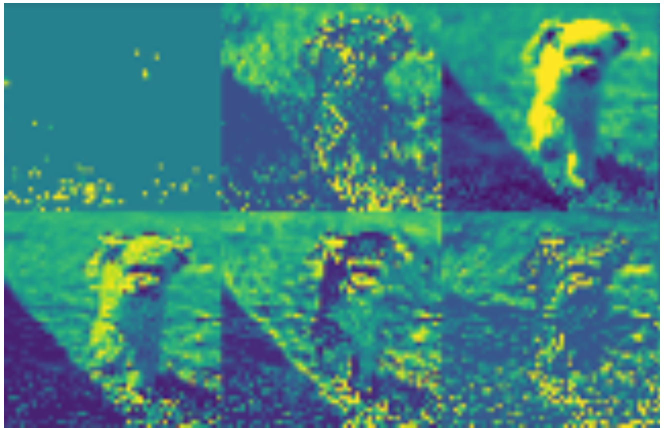
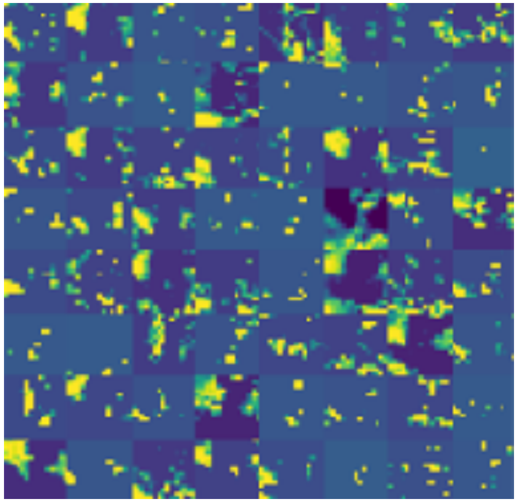
Visualizing Intermediate Activations

- **Result:** Information Distillation Pipeline
  - Deeper layers have more abstract triggers
  - Deeper activations are increasingly sparse
  - Early layers are texture and edge detectors
  - Notion of "High Level Abstraction," has biological motivation



Early Activations are larger but not as numerous

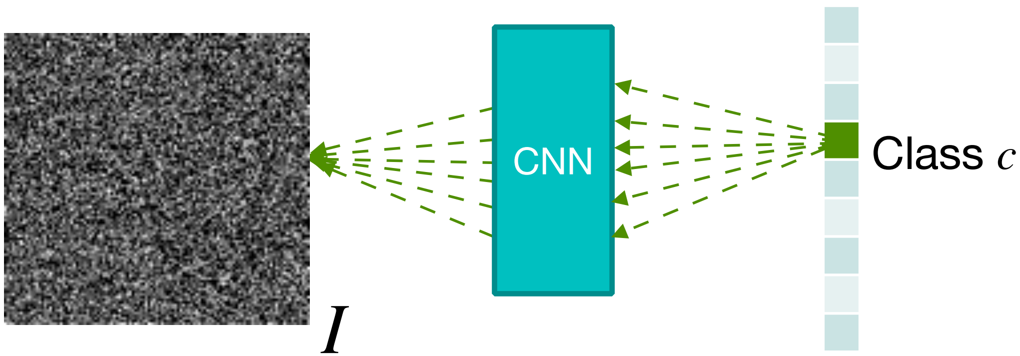
Later Activations are smaller and more numerous



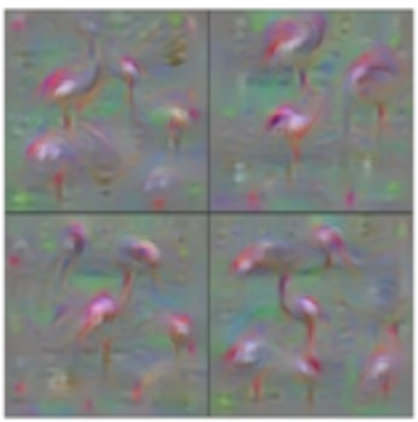
Visualizing Filters: Class Neuron

Method Two

- **Idea:** What Maximally Activates a Class Output?
    - Gradient Ascent in the Input Space



$$I \leftarrow I + \eta \nabla f_c(I)$$



Flamingo

where c is a specific neuron in output layer

f is the neural network function

I is the input image, init to zeros (or random)

∇ is the gradient of f_c w.r.t I

CNN weights stay unchanged

http://cs231n.github.io/understanding-cnn/

Visualizing Filters: Maximal Activations

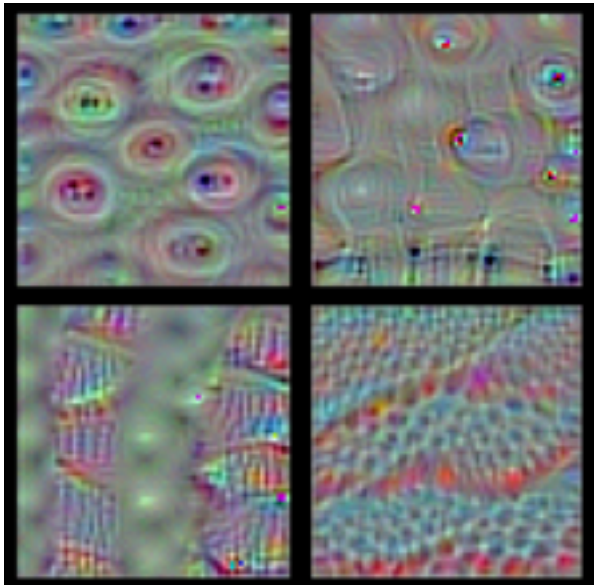
- **Idea:** What Maximally Activates a **Filter**?
  - **Again**: Gradient Ascent in the Input Space

$$I \leftarrow I + \eta \sum_{i,j} \nabla f_n(I)_{i,j}$$

"trick" use norm of gradient

where n is a specific **filter** in a layer

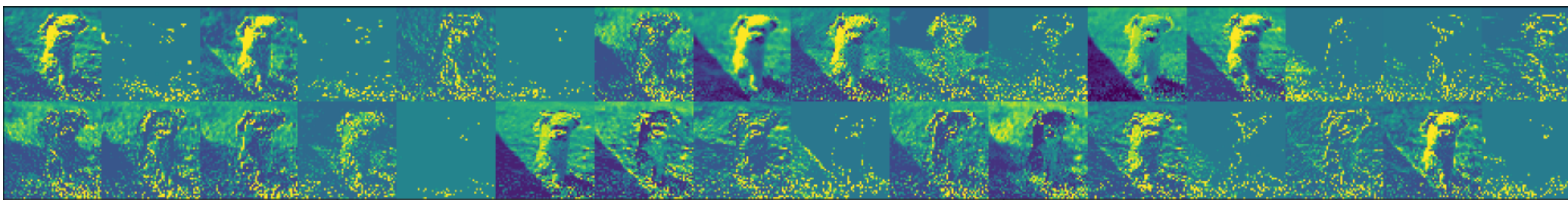
f is the function to n^{th} filter in layer

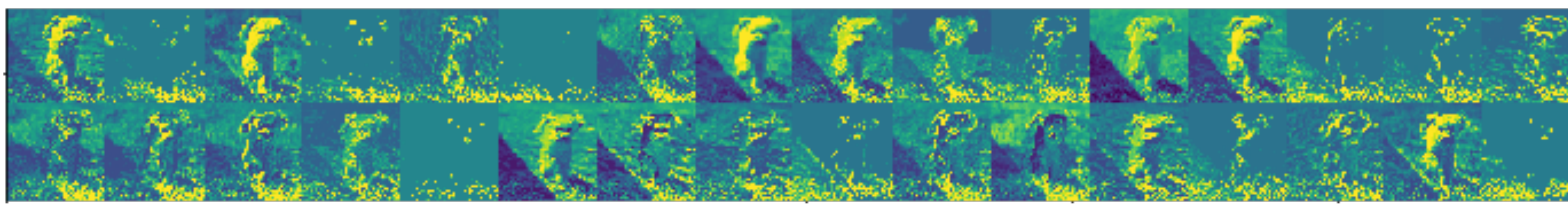


Visualizing ConvNets

Part One: Filter Activations

Part Two: Image Gradients





Follow Along: 04 LectureVisualizingConvnets.ipynb
activation-demo

Class Activation Mapping (CAM)

- **Idea:** What areas of the image contributed most to the classification result?
- Also, for each class, what areas of the image exhibit features of that class?
- Use change in output, w.r.t. final conv layer

normalize by h x w of A

final layer output in response to image I
c is class of interest

$$\alpha_k^c = \frac{1}{|A_k^{(L)}|} \sum_{i,j} \frac{\partial f_c(I)}{\partial A_{i,j,k}^{(L)}}$$

final convolutional layer, L, activations
for row, column, channel

gradient weight for channel k and class c in layer L
k in $1 \ldots K$ activations in final layer

Ramprasath et al., **Grad-CAM:** Visual Explanations from Deep Networks via Gradient-based Localization

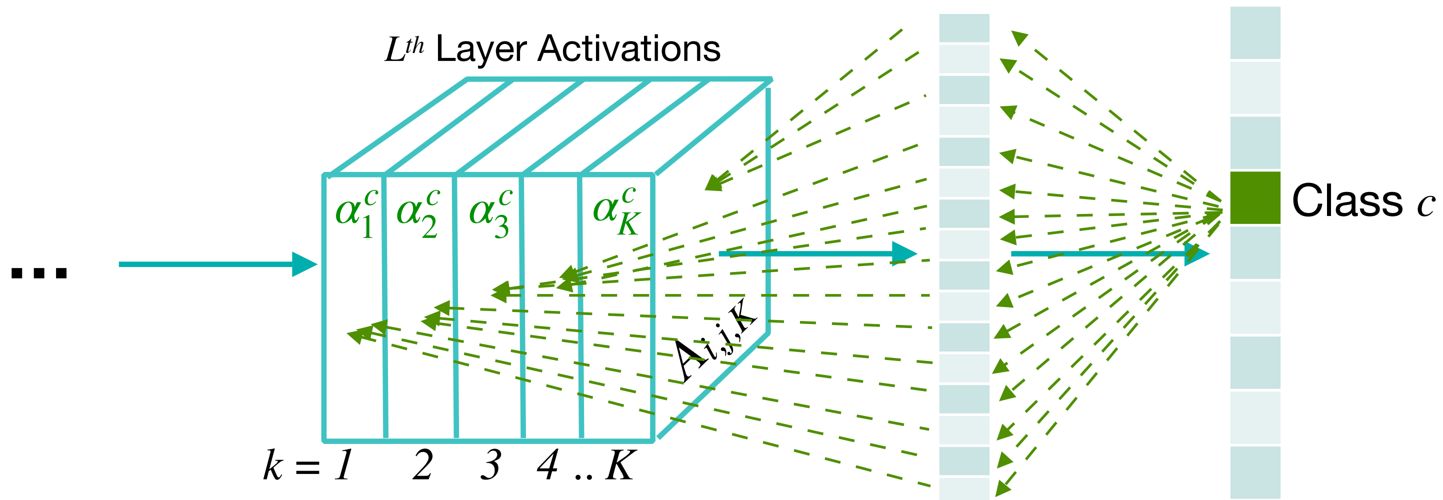
Class Activation Mapping (CAM)

$$\alpha_k^c = \frac{1}{|A_k^{(L)}|} \sum_{i,j} \frac{\partial f_c(I)}{\partial A_{i,j,k}^{(L)}}$$

final layer output in response to image I
c is class of interest

final convolutional layer, L, activations
for row, column, channel

gradient weight for channel k and class c in layer L
k in $1 \ldots K$ activations in final layer



Sensitivity of Class to Activations

Ramprasath et al., **Grad-CAM:** Visual Explanations from Deep Networks via Gradient-based Localization

Class Activation Mapping (CAM)

$$\alpha_k^c = \frac{1}{|I \times J|} \sum_{i,j} \frac{\partial f_c(I)}{\partial A_{i,j,k}^{(L)}}$$

final layer output in response to image I
c is class of interest

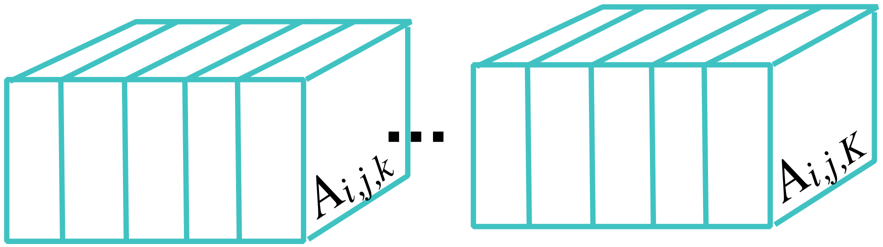
final convolutional layer, L, activations for row, column, channel

gradient weight for channel k and class c in layer L
k in $1 \ldots K$ activations in final layer

Heatmap, S, is the **weighted sum** of final layer activations:

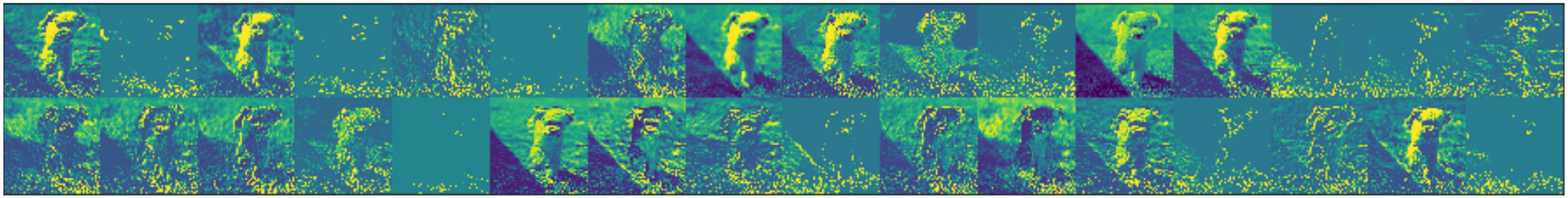
$$S_{i,j} = \frac{1}{S_{max}} \sum_k \phi(\alpha_k^c A_{i,j,k}^{(L)})$$

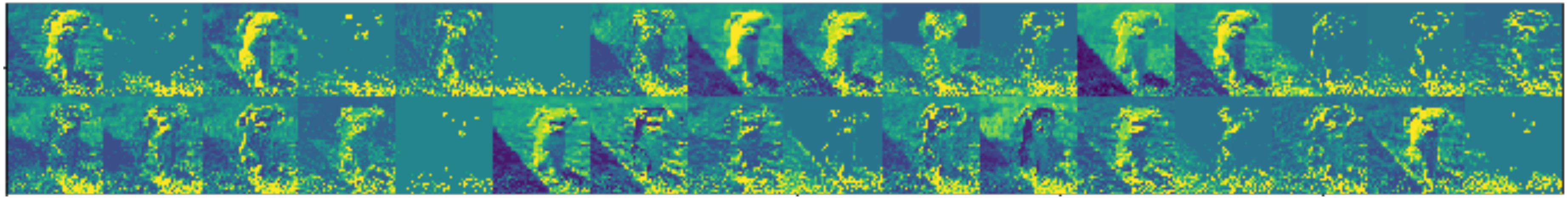
relu activation



Visualizing ConvNets

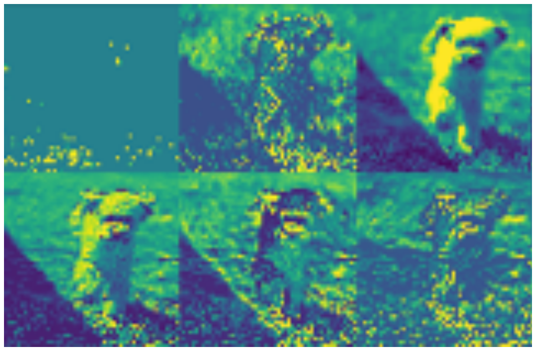
Part Three: Grad-CAM



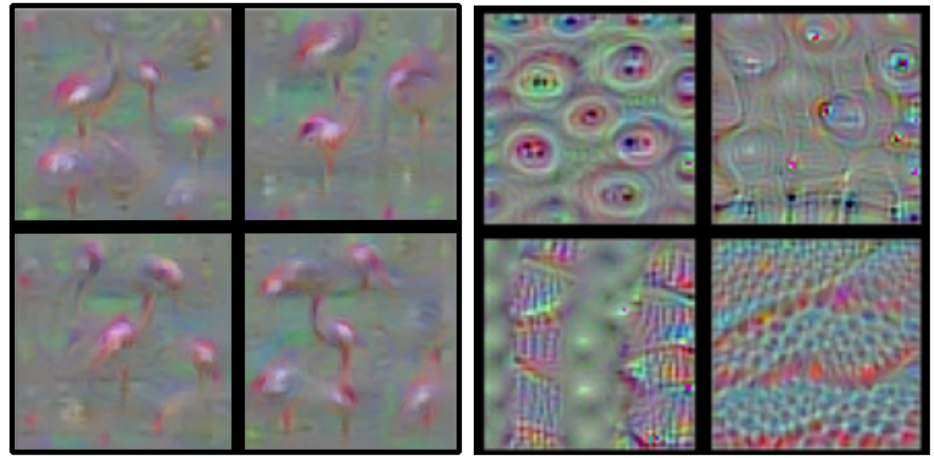


`Follow Along: 04 LectureVisualizingConvnets.ipynb`
`activation-demo`

Review: our visualization toolset

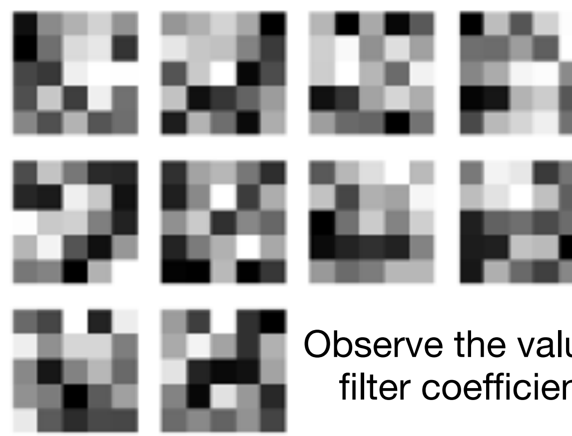


Visualize Activation
in response to input image



Visualize input maximized
to activate a certain class of filter

Use final convolutional layer to
see most influential part of input



Observe the value of
filter coefficients

Circuits and Features

We believe that neural networks consist of meaningful, understandable features. Early layers contain features like edge or curve detectors, while later layers have features like floppy ear detectors or wheel detectors. The community is divided on whether this is true. While many researchers treat the existence of meaningful neurons as an almost trivial fact — there's even a small literature studying them [15, 2, 16, 17, 4, 18, 19] — many others are deeply skeptical and believe that past cases of neurons that seemed to track meaningful latent variables were mistaken [20, 21, 22, 23, 24] . [3] Nevertheless, thousands of hours of studying individual neurons have led us to believe the typical case is that neurons (or in some cases, other directions in the vector space of neuron activations) are understandable.

Cammarata, et al., "Thread: Circuits", Distill, 2020.

Why Visualize Trained CNN Architectures?

From OpenAI: Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, Shan Carter

Many important transition points in the history of science have been moments when science

SCHWANN'S CLAIMS ABOUT CELLS

Claim 1

The cell is the unit of structure, physiology, and organization in living things.

Claim 2

The cell retains a dual existence as a distinct entity and a building block in the construction of organisms.

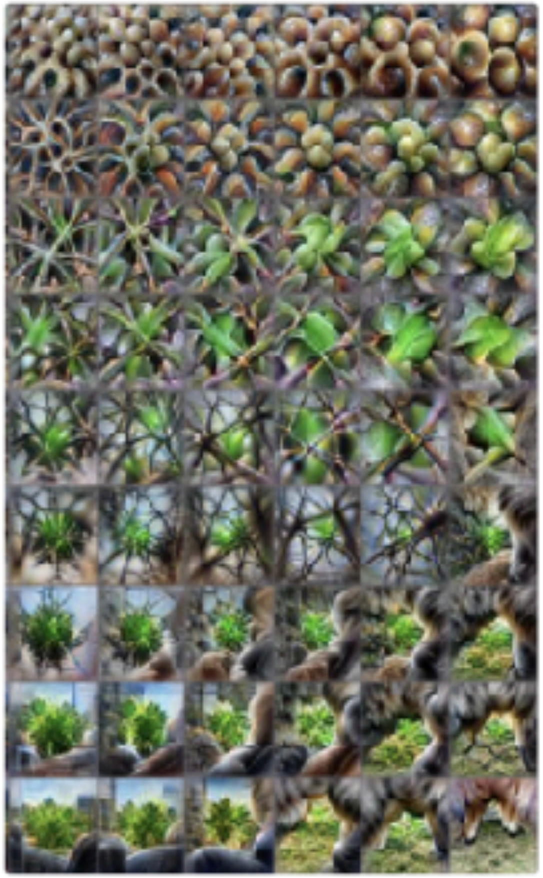
Claim 3

Cells form by free-cell formation, similar to the formation of crystals.

The famous examples of this phenomenon happened at a very large scale, but it can also be the more modest shift of a small research community realizing they can now study their topic in a finer grained level of detail.

https://distill.pub/2020/circuits/zoom-in/

Speculative Claims for Circuits



THREE SPECULATIVE CLAIMS ABOUT NEURAL NETWORKS

Claim 1: Features

Features are the fundamental unit of neural networks.

They correspond to directions. [1] These features can be rigorously studied and understood.

Claim 2: Circuits

Features are connected by weights, forming circuits. [2]

These circuits can also be rigorously studied and understood.

Claim 3: Universality

Analogous features and circuits form across models and tasks.

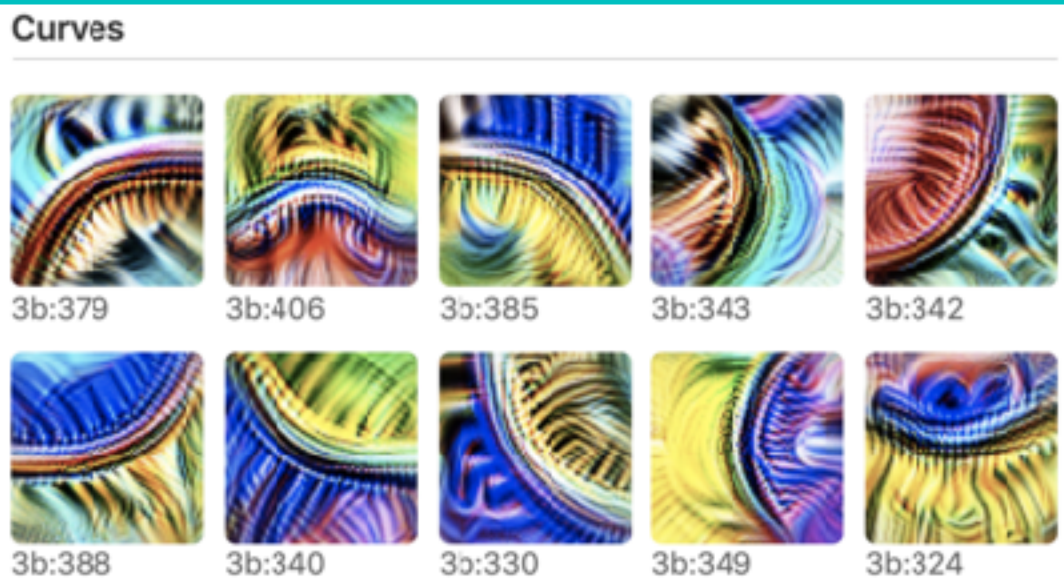
Left: An activation atlas [13] visualizing part of the space neural network features can represent.

https://distill.pub/2020/circuits/zoom-in/

Building Blocks: Features

- *Features are fundamental units of neural network. Features are how we describe what an activation in a network does.*
- They must be discovered, typically by:
  - Extensive visualization of excitations and filter weights (*forward analysis*)
  - Analysis of synthetic examples and dataset examples (*forward and backward analysis*)
  - Through similarity to other features. *e.g.*, rotations or scaling of a given feature (*parallel analysis*)
  - Through downstream features which *naturally* depend on the given feature working (*backward analysis*)
- With assumption of what **feature** is, a **circuit** can be implemented (even by hand) that nearly identically follows the assumed functionality

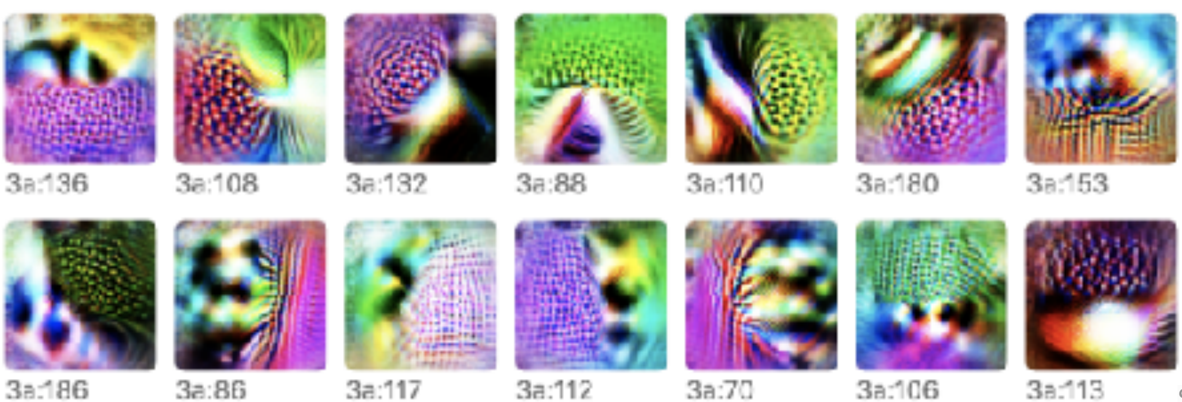
Examples of Discovered Features

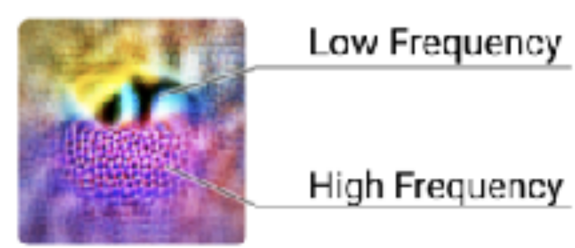


Hypothesized feature group (part of circuit)

Downstream features

High to Low Frequency Transition: perhaps good at finding blurred versus area in focus





Shubert, et al., “Hi-Lo Freq. Detectors”, 2021.

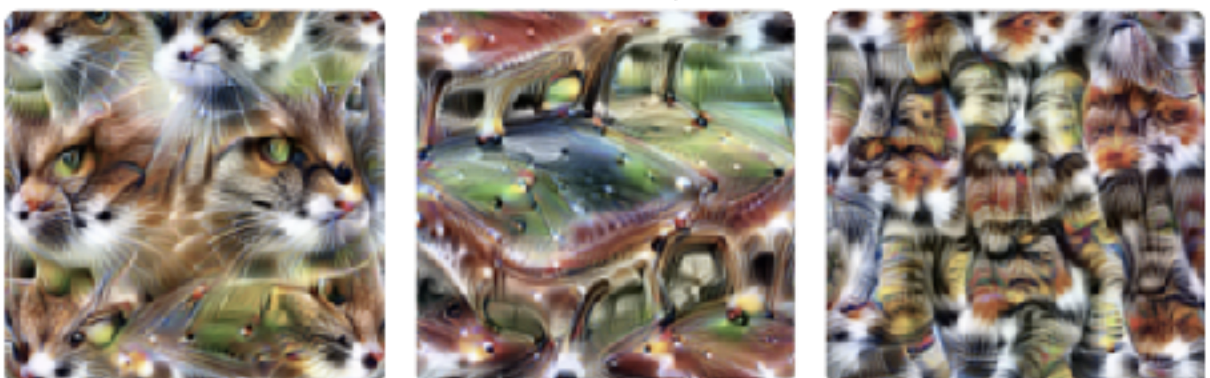
More Examples: Higher Level Features

Pose Invariant Dog-head Detection

Neuron 4b:409

Dataset examples for neuron 4b:409

Polysemantic Neurons: things that become coupled…



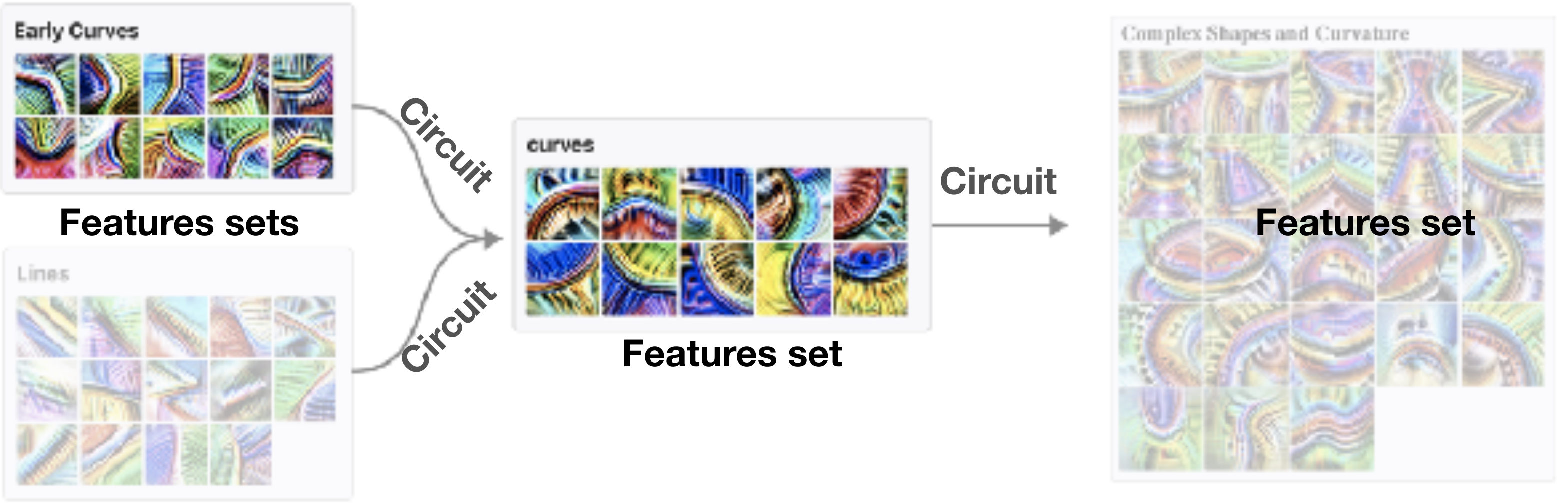
4e:55 is a polysemantic neuron which responds to cat faces, fronts of cars, and cat legs. It was discussed in more depth in Feature Visualization [4].

The existence of these neurons is likely one of the main criticism of network features.

Why do these exist?

From Features to Circuits

- *Features are connected by weights, forming circuits*
- *"All neurons in our network are formed from linear combinations of neurons in the previous layer, followed by ReLU. If we can understand the features in both layers, shouldn't we also be able to understand the connections between them?"*
- *"Once you understand what features they're connecting together… You can literally read meaningful algorithms off of the weights."*



https://microscope.openai.com/models/inceptionv1/