Trustworthy AI Autonomy

M1-2: Explainability: Latent space visulization

Ding Zhao

Assistant Professor

Carnegie Mellon University

Carnegie Mellon University

Safe AI Lab @CMU

Contents

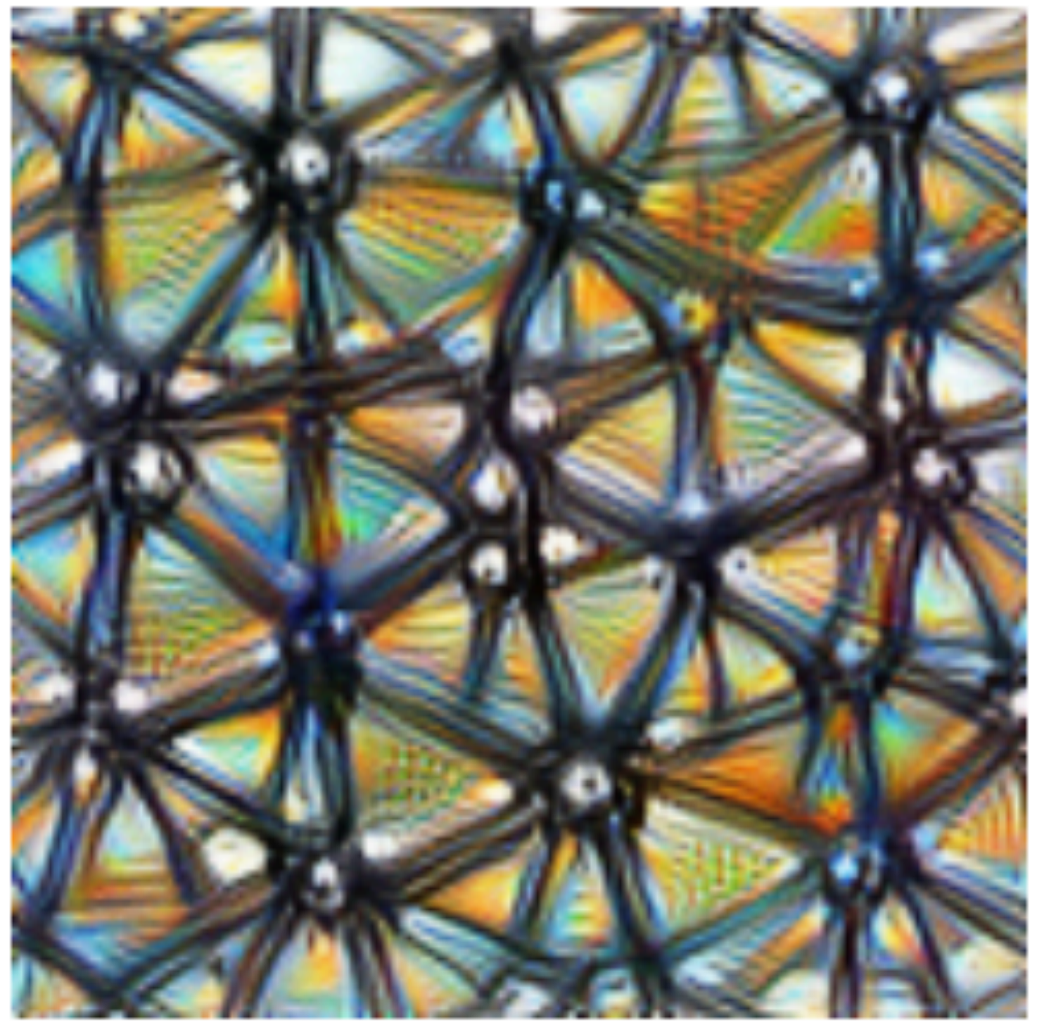
- Feature visualization
  - Data-driven methods
  - Optimization-based methods

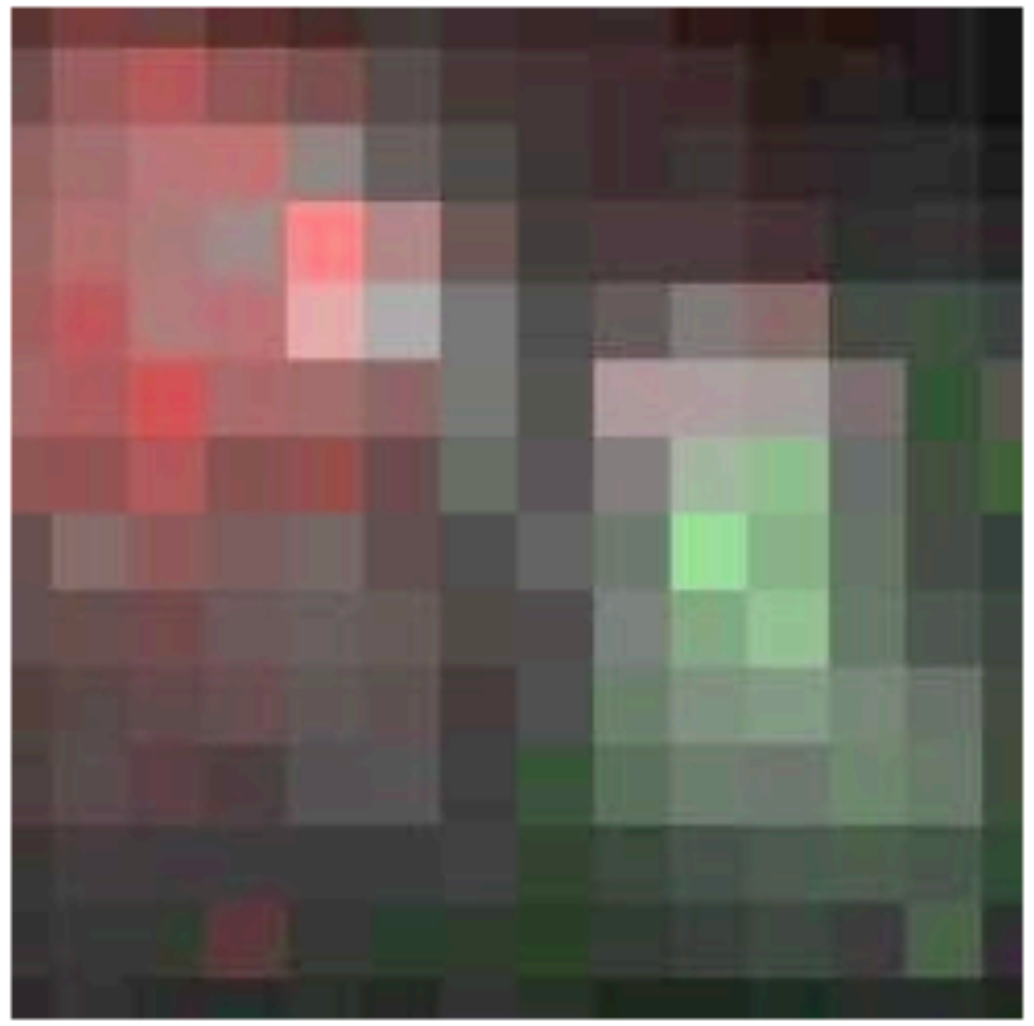
Why visualization

- Understand why deep neuron networks works
- Improve the design of neural networks
- A (surprise) link to adversarial machine learning and TAIAT (next lecture)

Feature visualization

- Different neurons are activated by different patterns in the input image
  - **Feature visualization:** answers questions about what a network—or parts of a network—are looking for by generating examples
  - **Attribution**: studies what part of an example is responsible for the network activating a particular way.

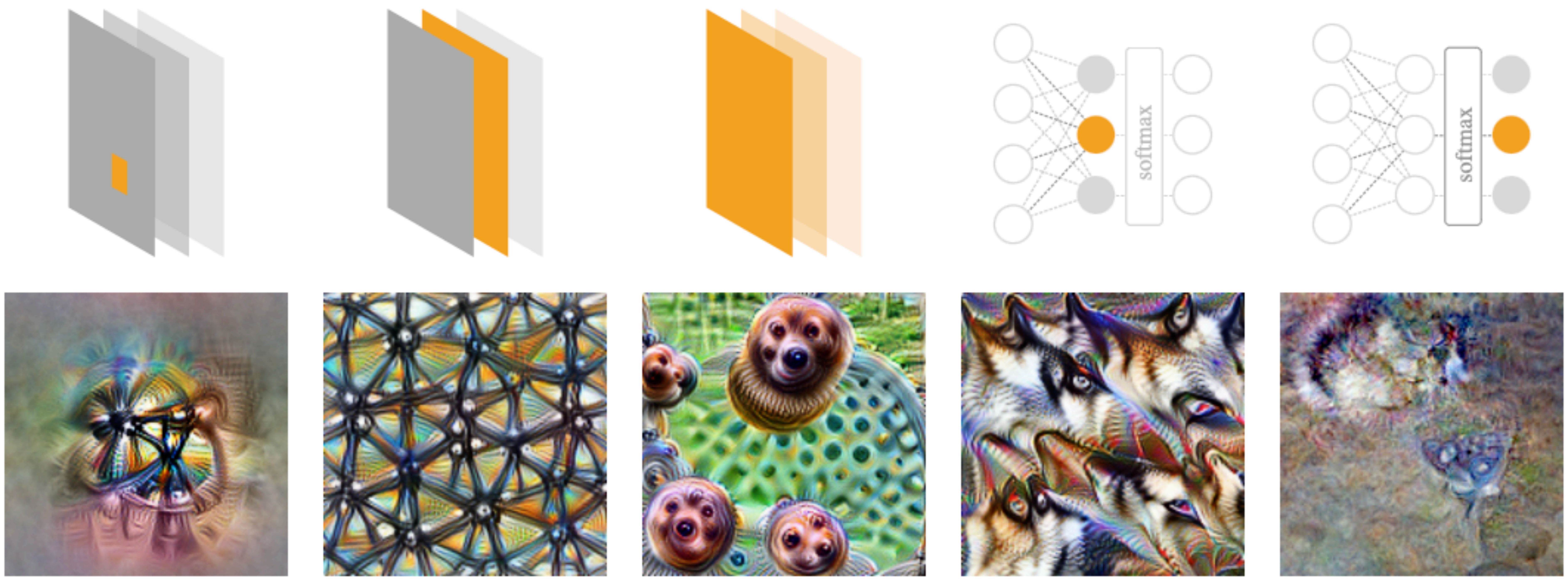


https://distill.pub/2017/feature-visualization/

Optimization objectives

Different **optimization objectives** show what different parts of a network are looking for.

- **n** layer index
- **x,y** spatial position
- **z** channel index
- **k** class index



Neuron	Channel	Layer/DeepDream	Class Logits	Class Probability
$\mathrm{layer}_n[x,y,z]$	$\mathrm{layer}_n[:,:,z]$	$\mathrm{layer}_n[:,:,:]^2$	`pre_softmax[k]`	`softmax[k]`

https://distill.pub/2017/feature-visualization/

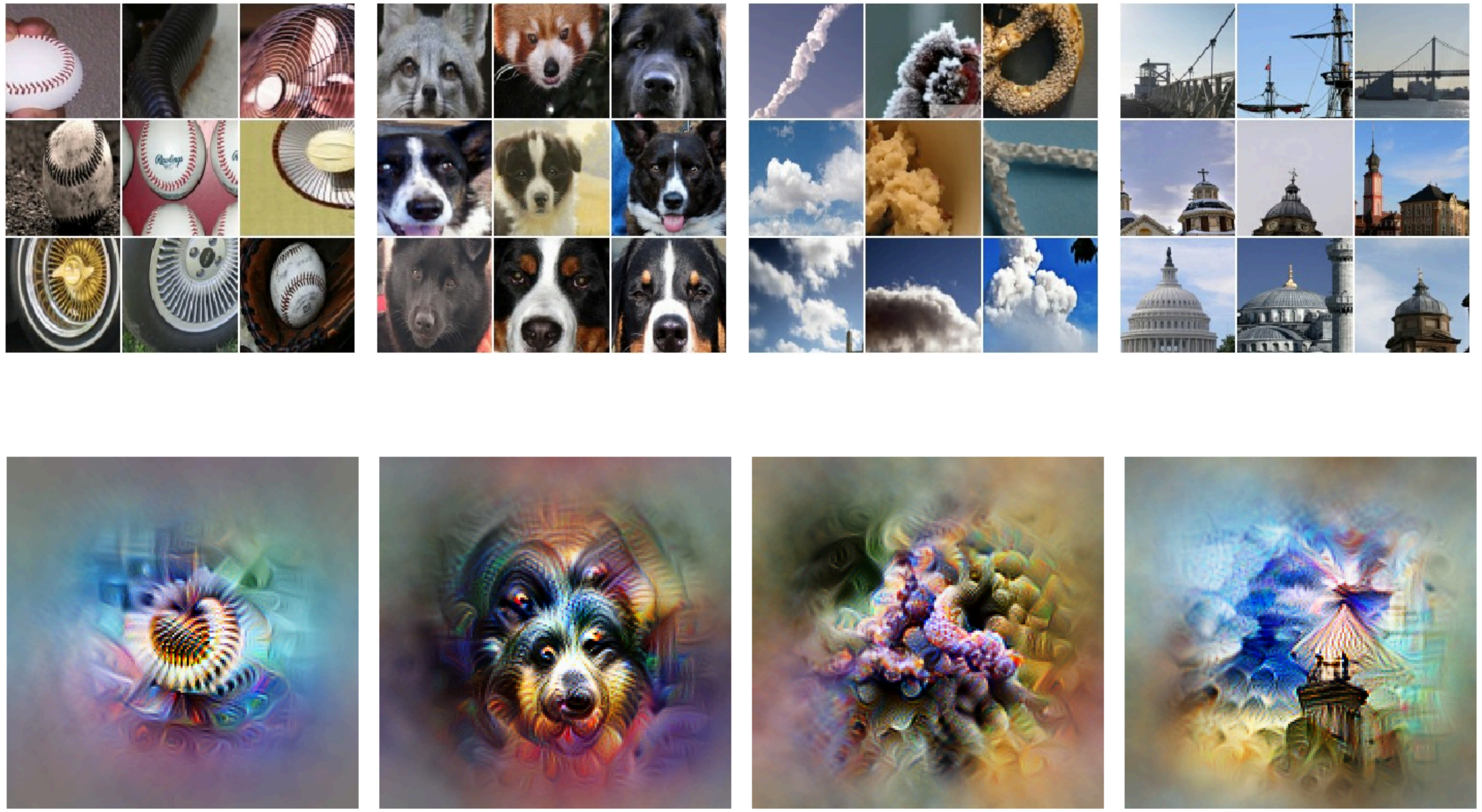
Feature visualization Approaches

- Dataset examples

  show us what neurons respond to in practice

- Optimization

  isolates the causes of behavior from mere correlations

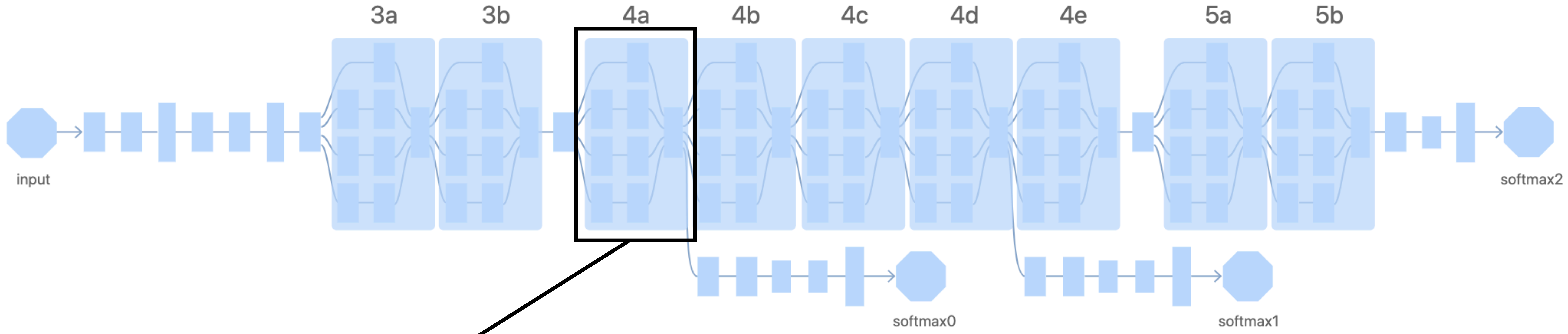


Neuron 1 Neuron 2 Neuron 3 Neuron 4

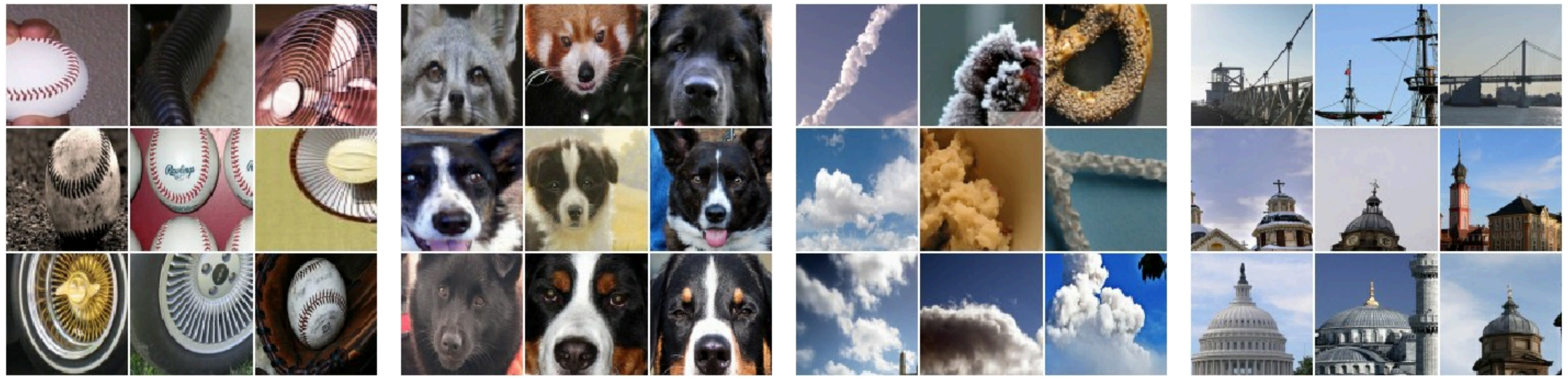
Credits: https://distill.pub/2017/feature-visualization/

Feature visualization

- Few samples that excite certain neurons in Layer 4a of GoogLeNet



GoogLeNet architecture



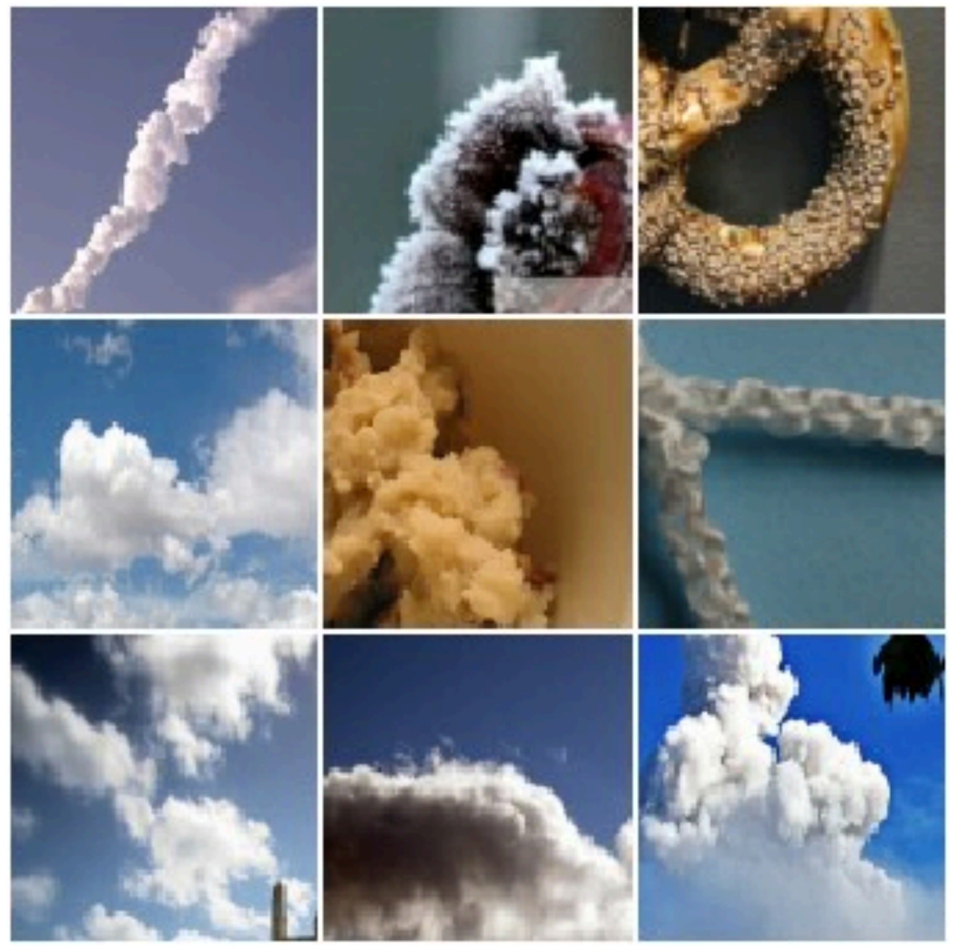
Credits: https://distill.pub/2017/feature-visualization/

Feature visualization

- Unclear which features of the image the network really emphasizes

Mixed 4a, neural #6
Baseball or stripes?

Mixed 4a, #240
Faces or snouts?



Mixed 4a, #453
Cloud or fluffy?

Mixed 4a, #492
Building or sky?

Credits: https://distill.pub/2017/feature-visualization/

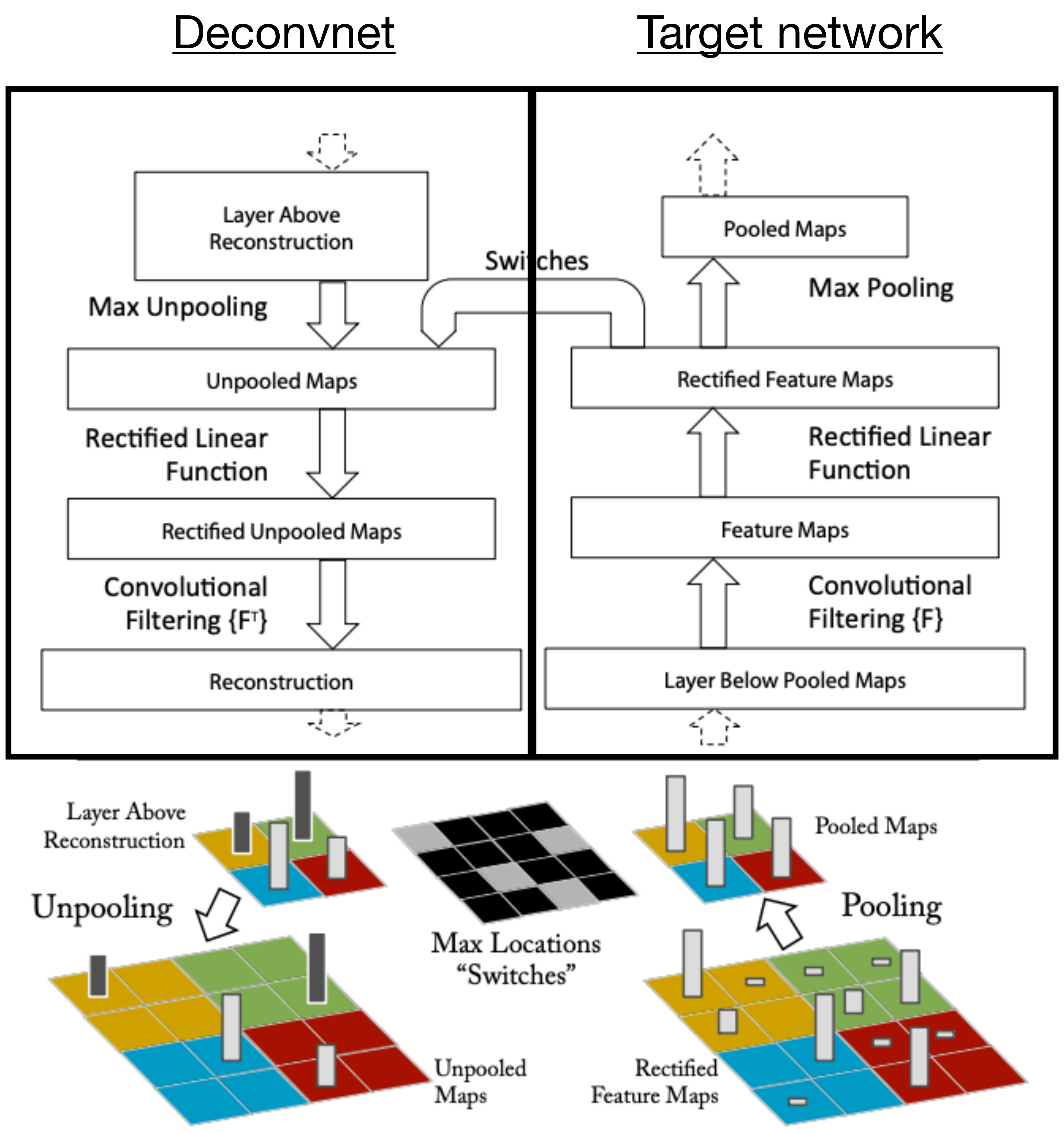
Deconvnet

- Deconvnet visualizes using input samples

  - Train backward-looking net
    to visualize the learned feature
    of the target network

  - Propagate the activations of single layer,
    to reconstruct the input image

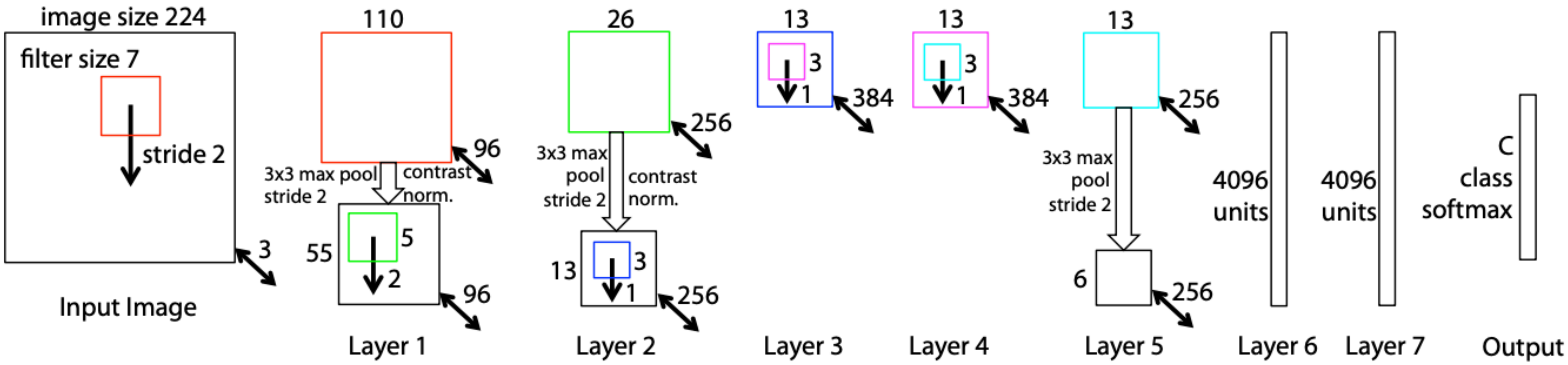
- Deconvnet allows to visualize what feature activates specific layer



Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." European conference on computer vision. Springer, Cham, 2014.

Deconvnet

- Suppose we have simplified AlexNet (8 layer conv net) as our target:

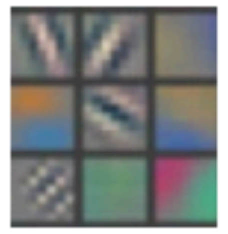


Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." European conference on computer vision. Springer, Cham, 2014.

Deconvnet of a single neuron: Layer 1

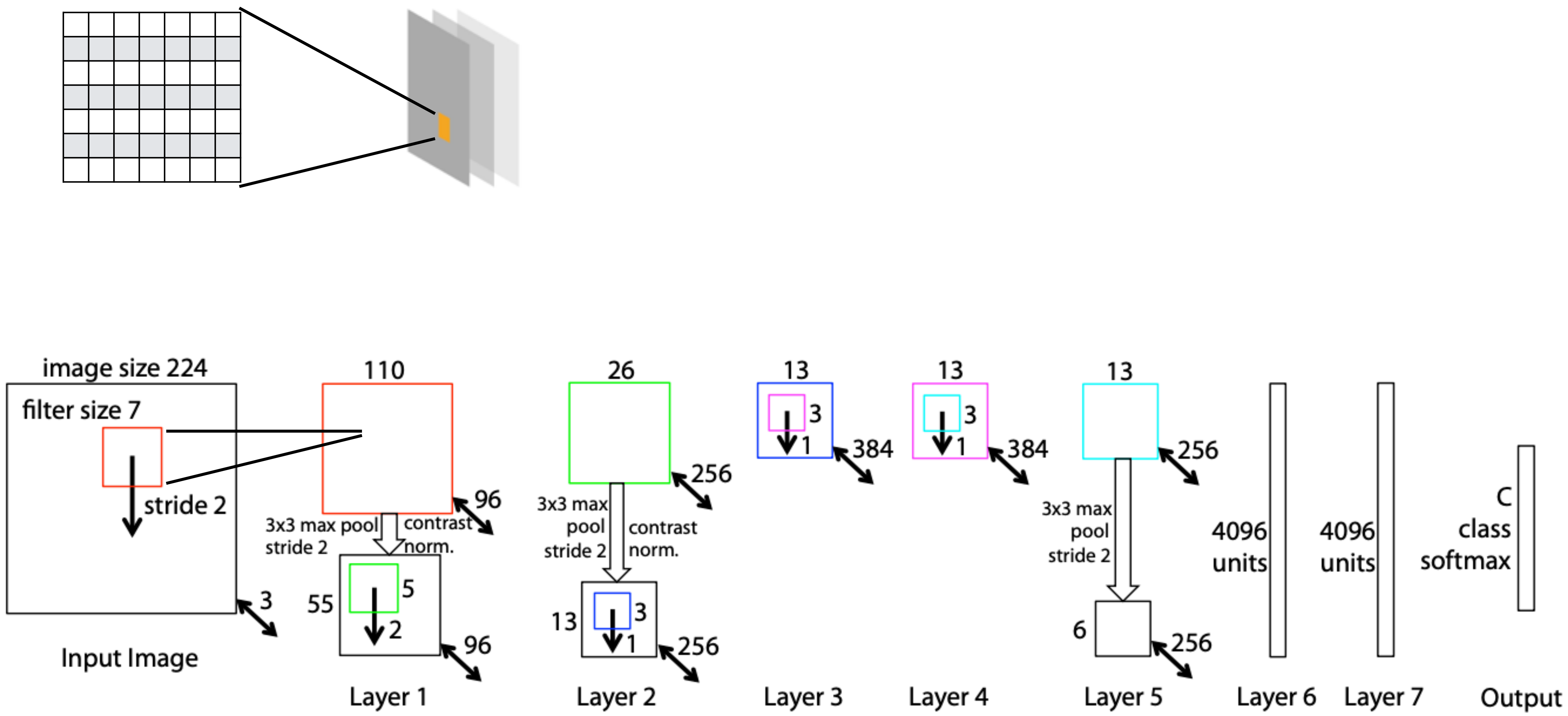
- Initial layers learn basic shapes

Filters



Layer 1

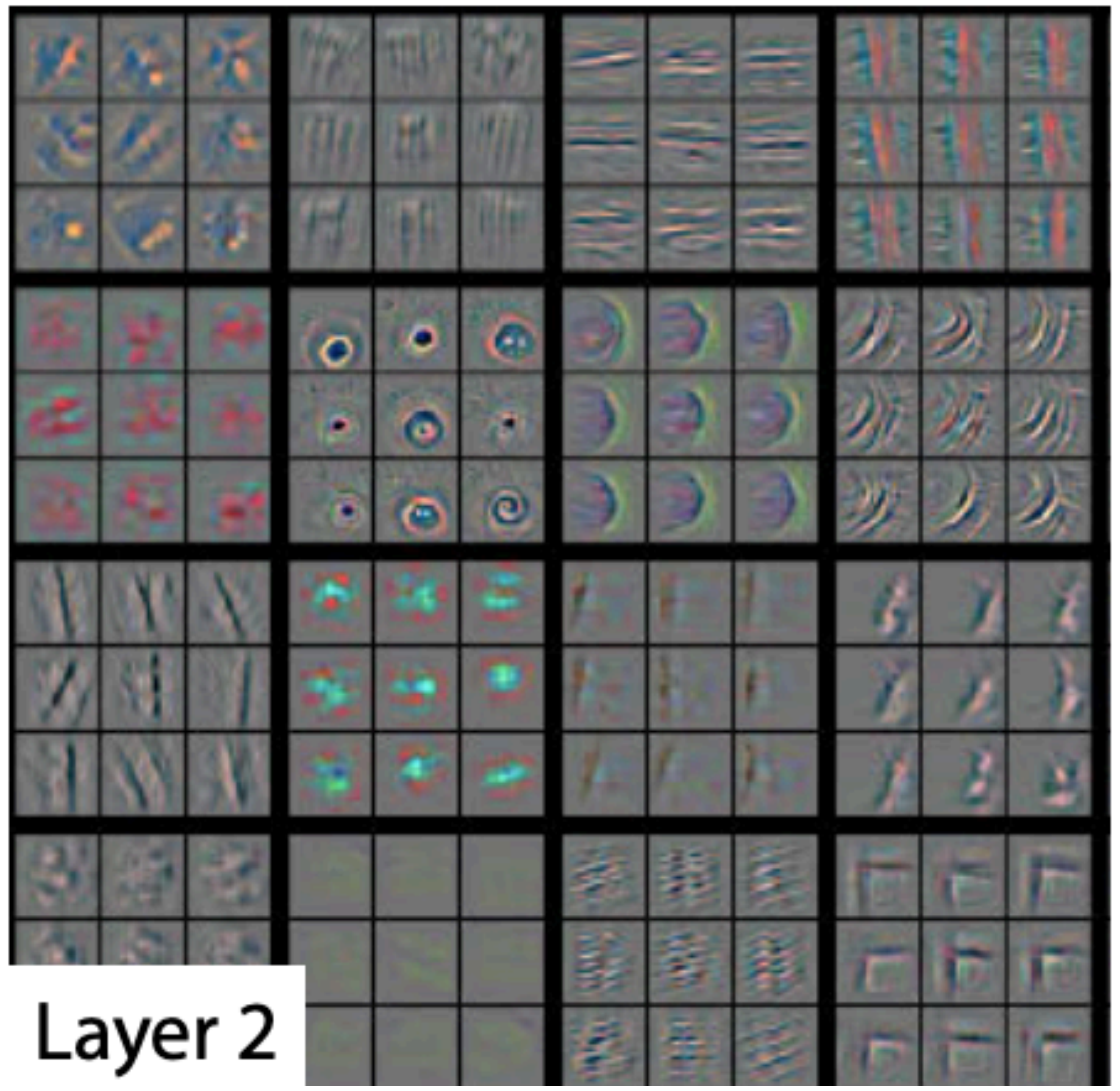
Corresponding image patches

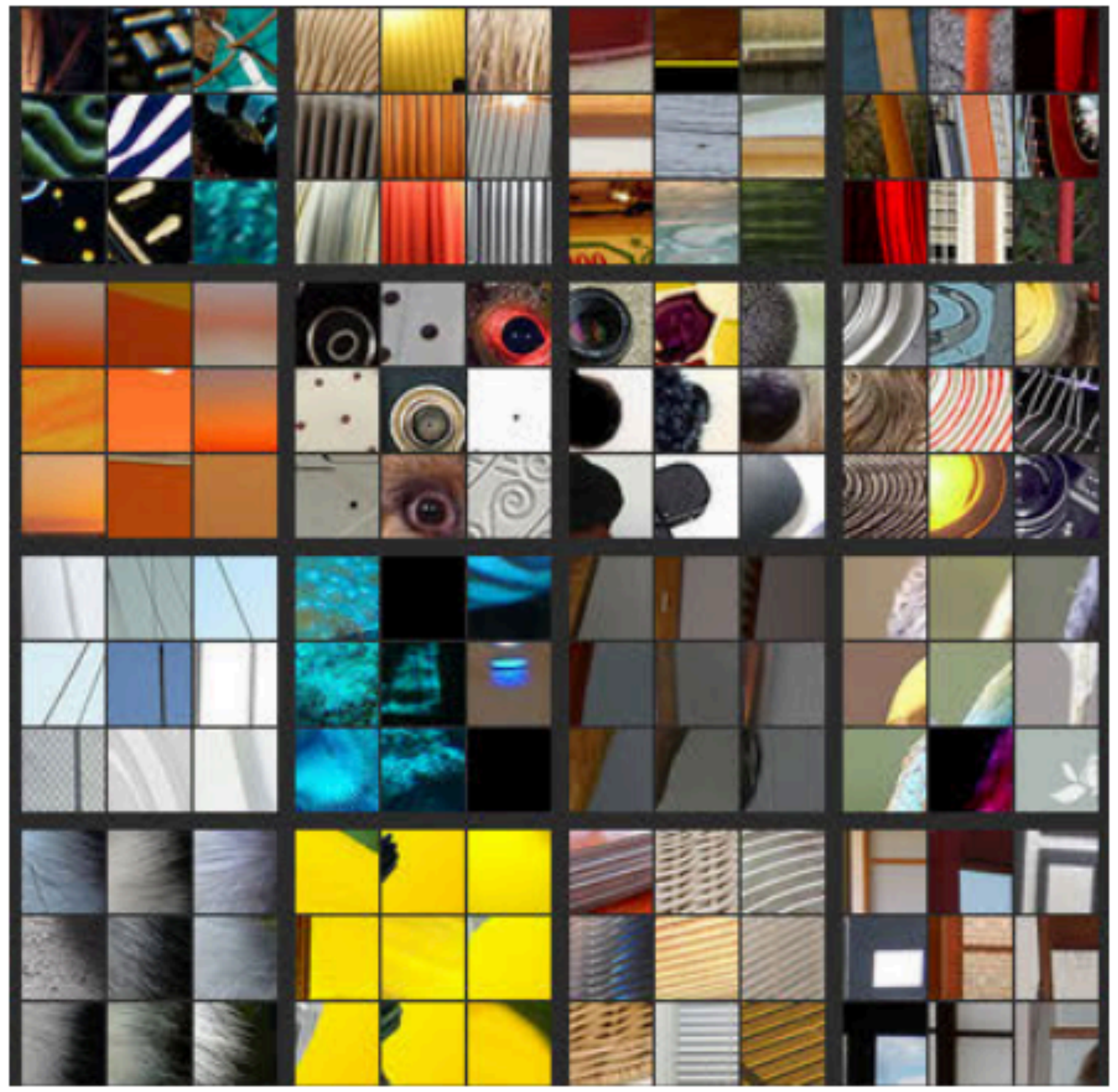


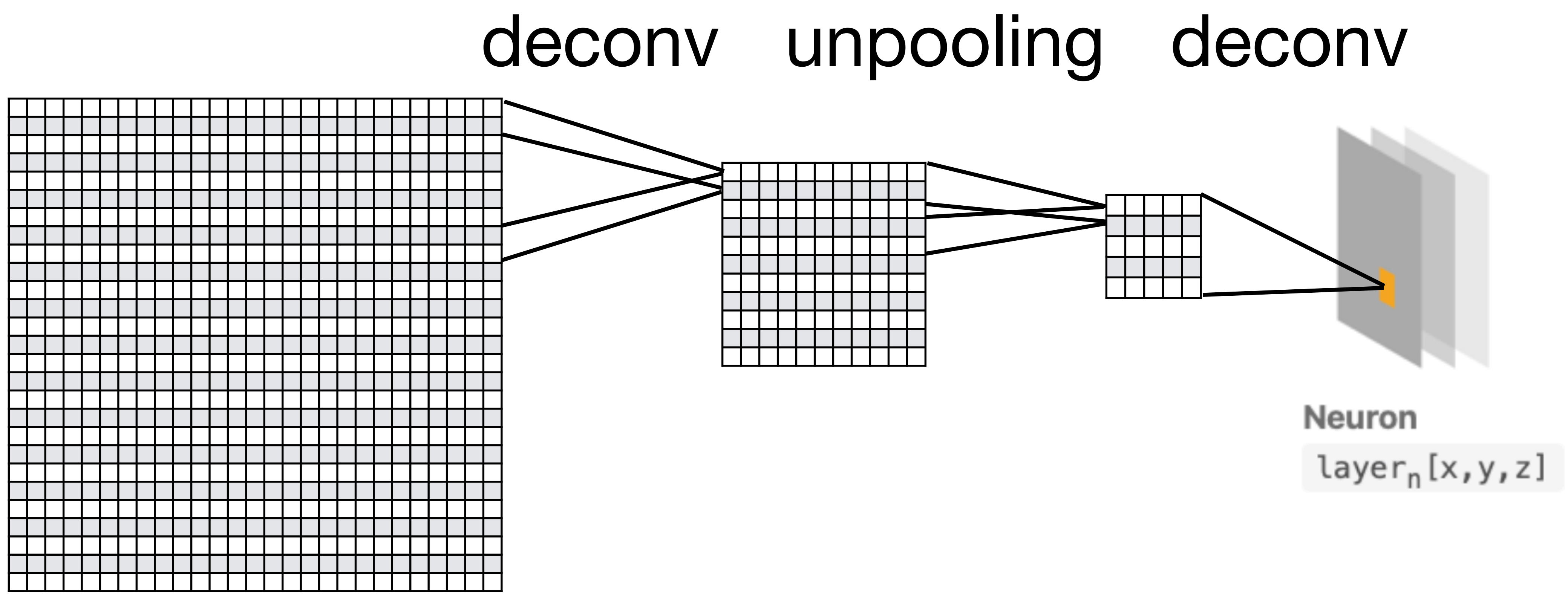
Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." European conference on computer vision. Springer, Cham, 2014.

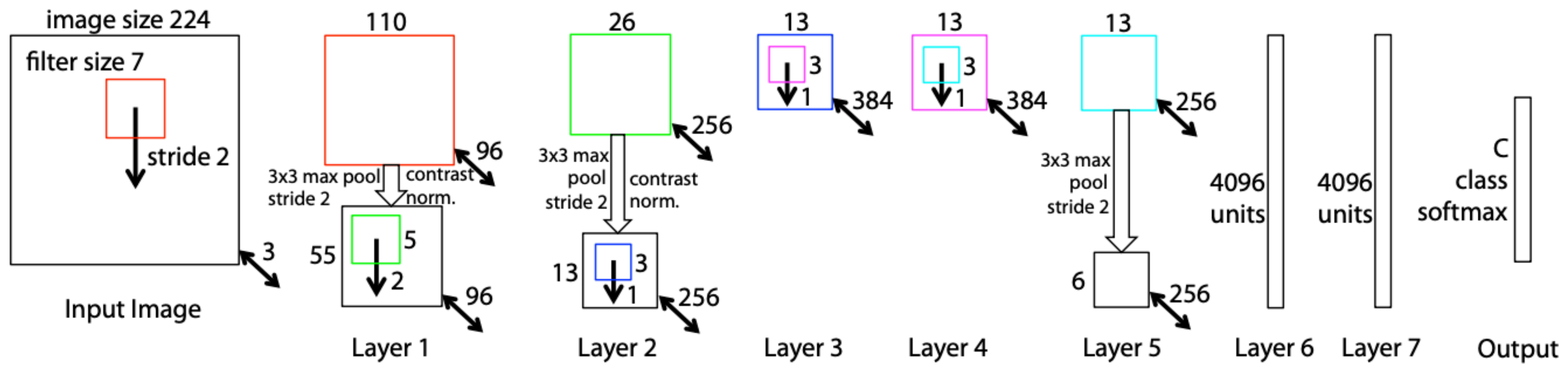
Deconvnet of a single neuron: Layer 2

Projection to the pixel space Corresponding image patches





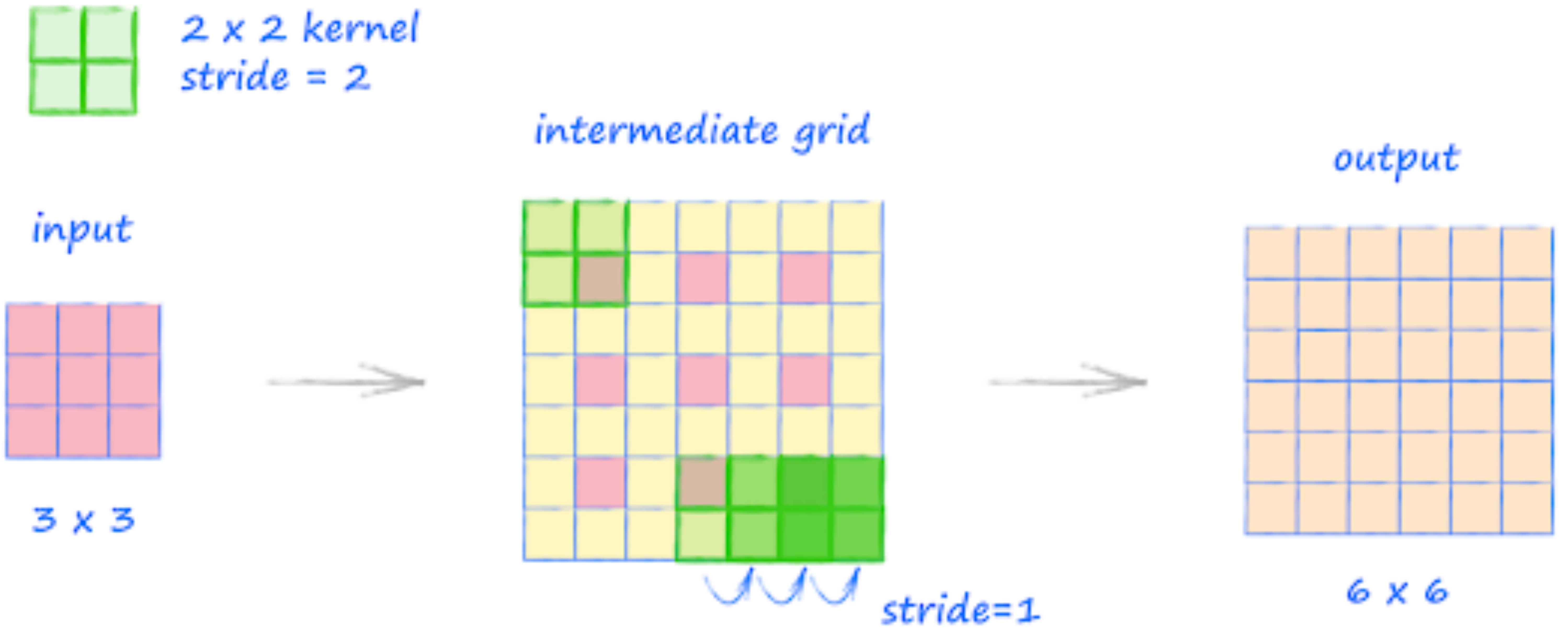




Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." European conference on computer vision. Springer, Cham, 2014.

Deconvolution operations

- Transpose convolution: expanding the input with intermediate grid



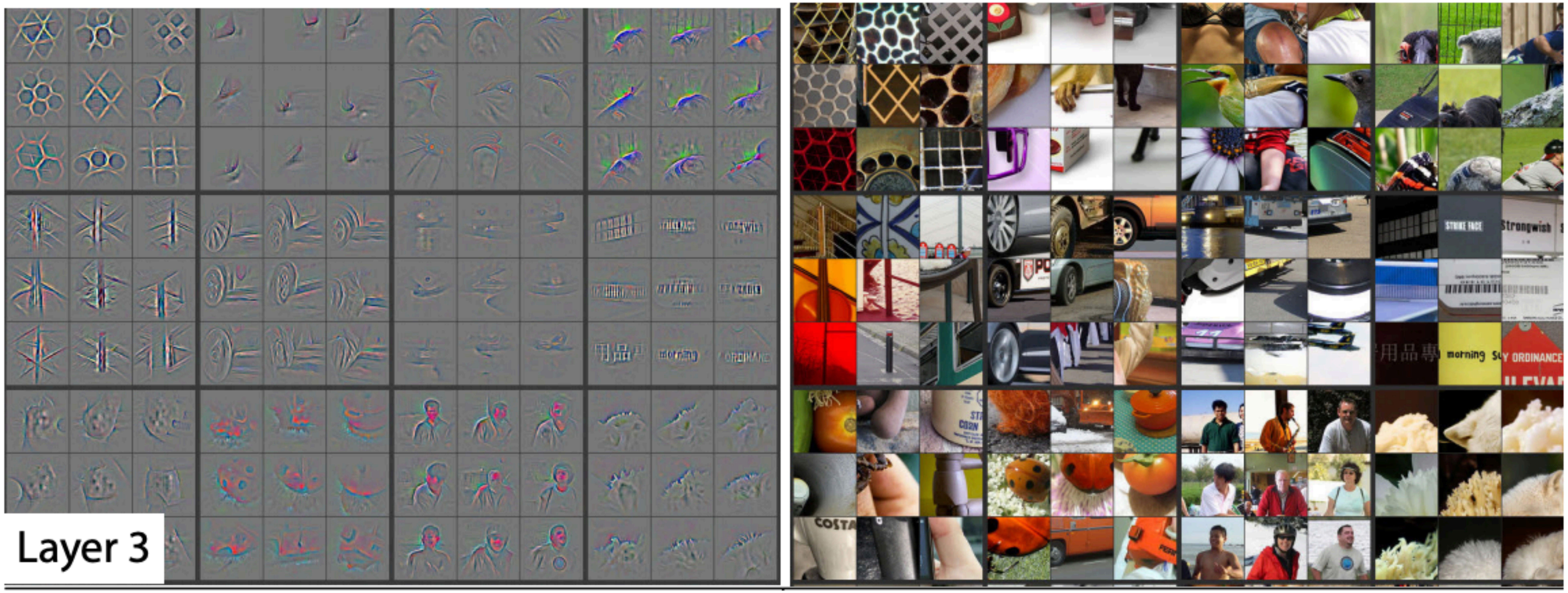
Output of transpose convolution:

$$\text{output size} = (\text{input size} - 1) * \text{stride} - 2 * \text{padding} + (\text{kernel size} - 1) + 1$$

Credits: http://makeyourownneuralnetwork.blogspot.com/2020/02/calculating-output-size-of-convolutions.html

Deconvnet

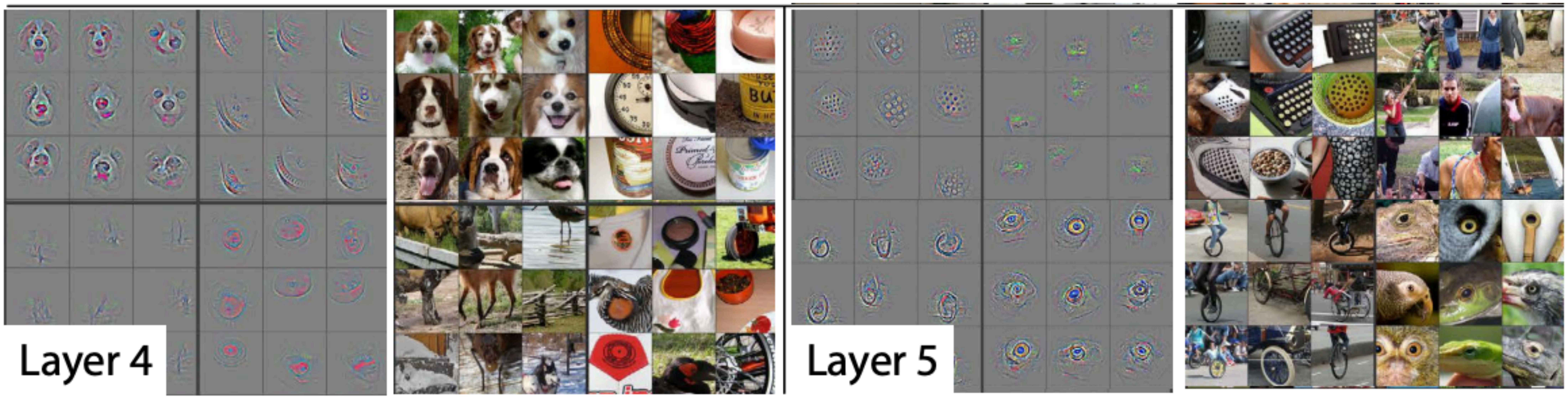
- Deeper layers create more complex features from the basic features



Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." European conference on computer vision. Springer, Cham, 2014.

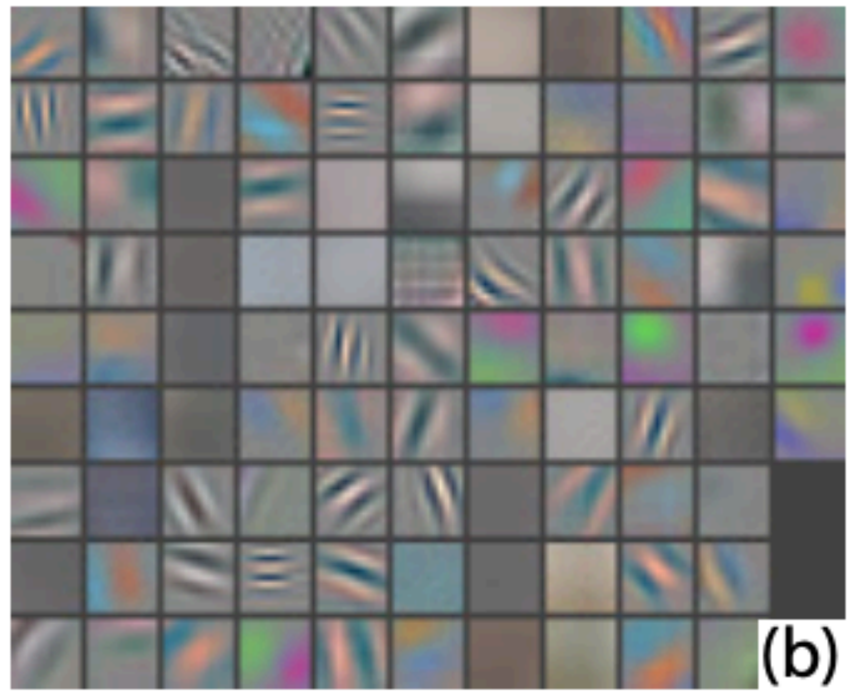
Deconvnet

- Final layers identify informative complex features for final prediction

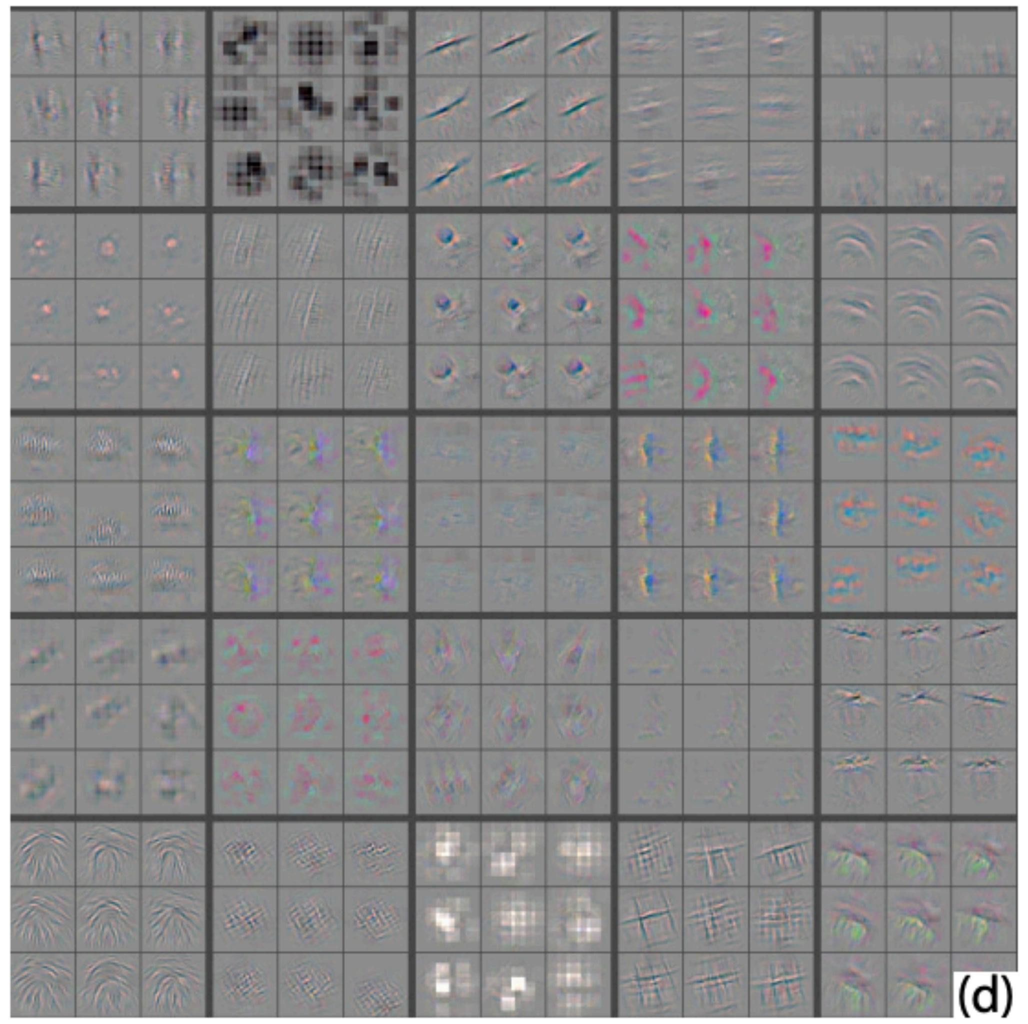


Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." European conference on computer vision. Springer, Cham, 2014.

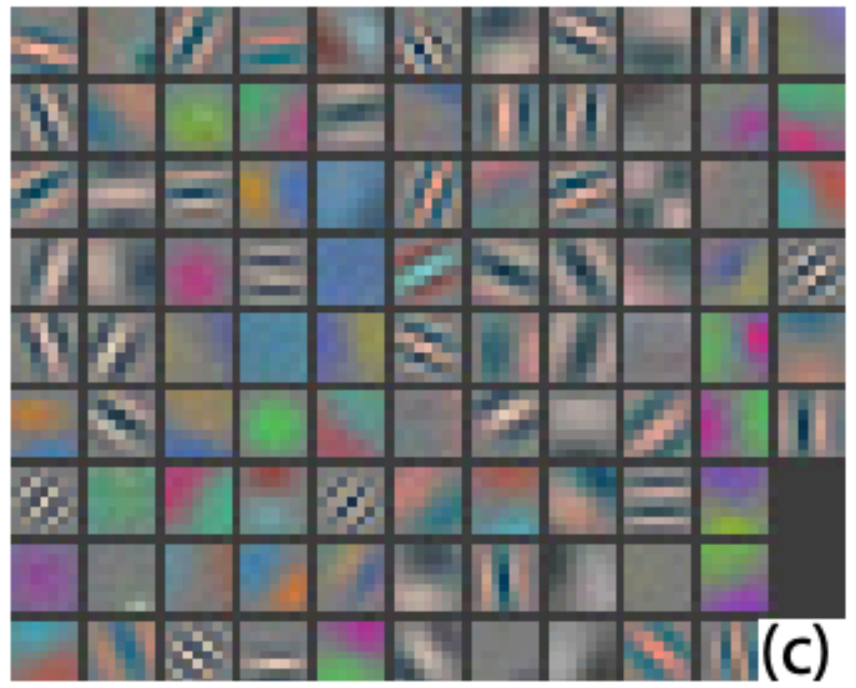
DeconvNet helps design better NN



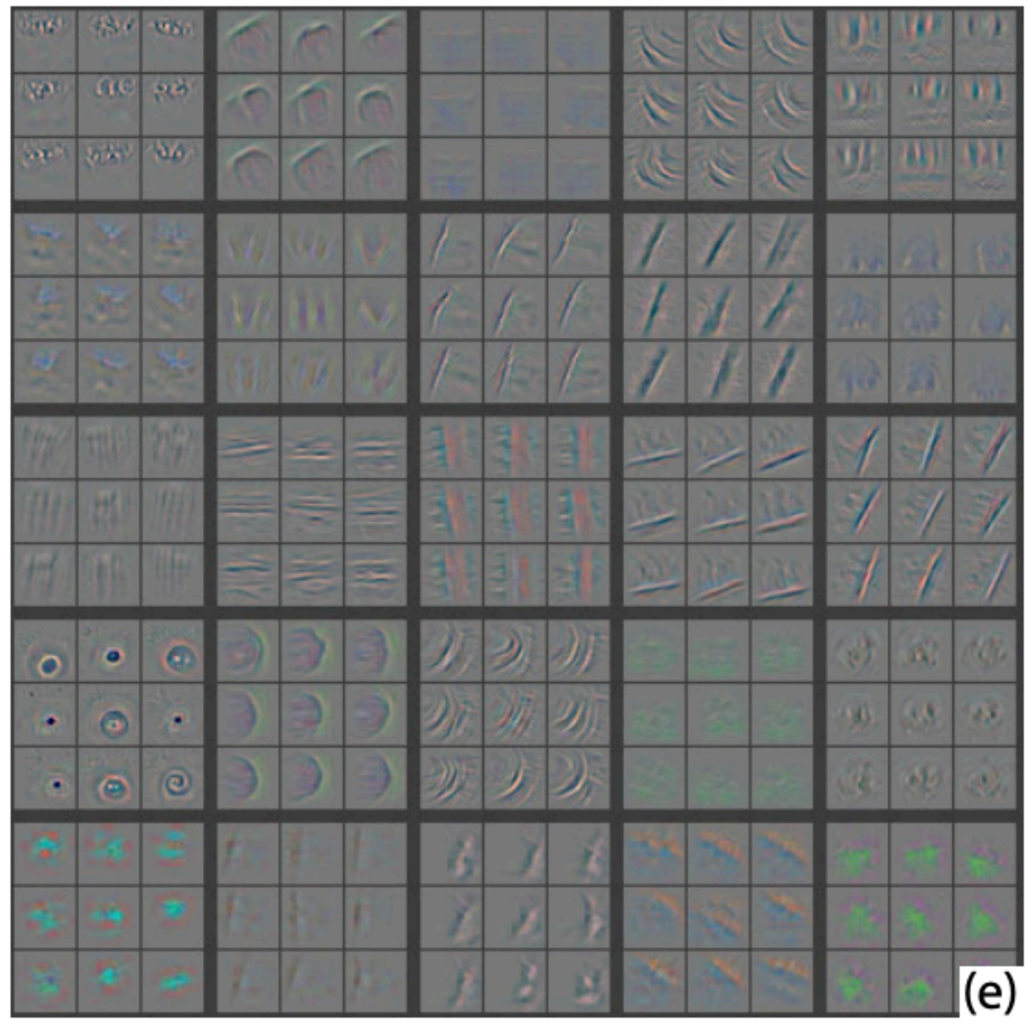
Layer 1 of AlexNet
(some dead features still exist)



Layer 2 of AlexNet



Layer 1 of the improved AlexNet
(fewer filters with dead features)



Layer 2 of the improved AlexNet (sharper features)

Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." European conference on computer vision. Springer, Cham, 2014.

Feature visualization

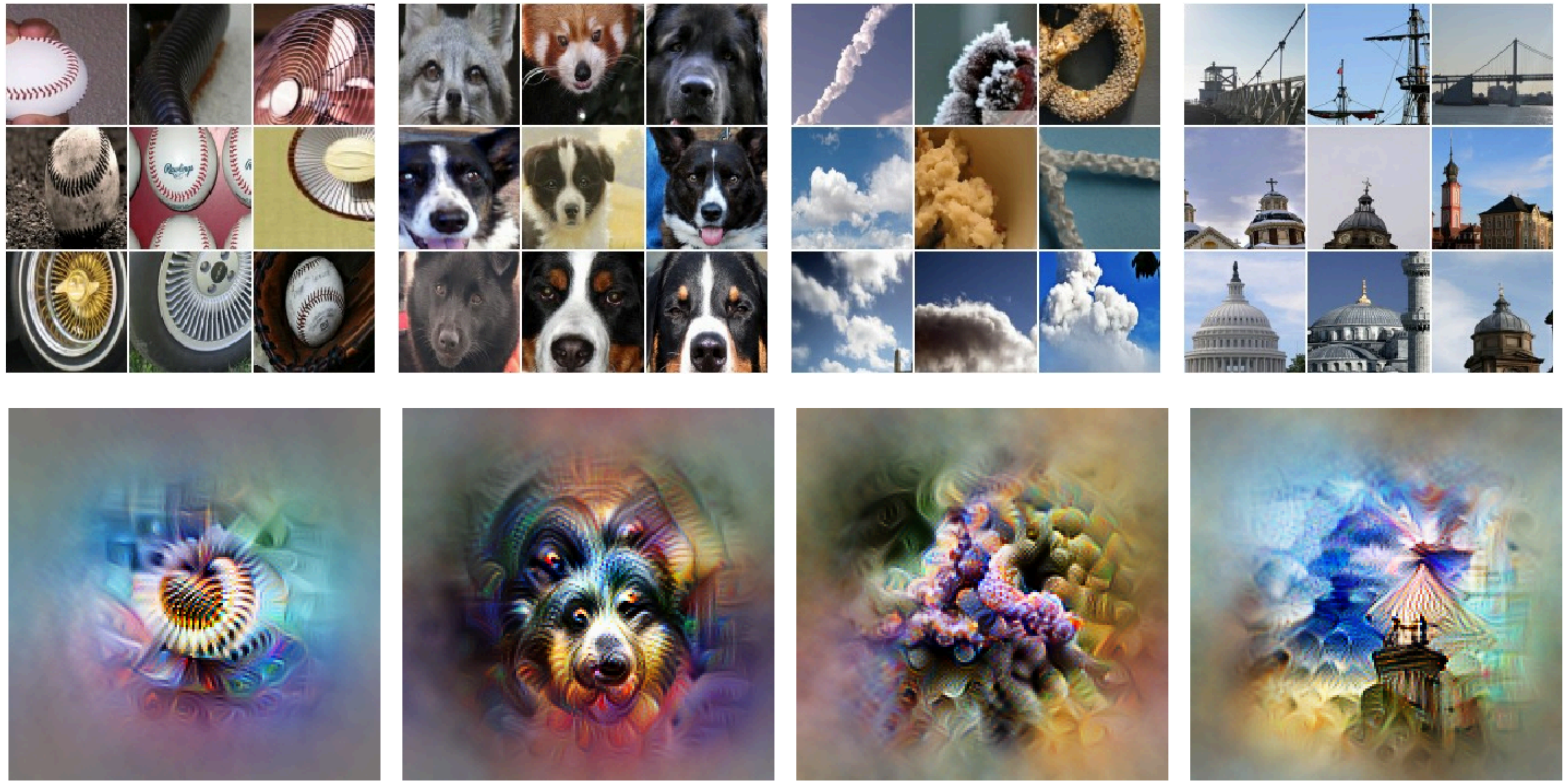
- Approaches:

  - Dataset examples

    show us what neurons respond to in practice

  - Optimization

    isolates the causes of behavior from mere correlations



Neuron 1 Neuron 2 Neuron 3 Neuron 4

Credits: https://distill.pub/2017/feature-visualization/

Gradient optimization-based feature visualization

- Gradient-based method does not need input samples:

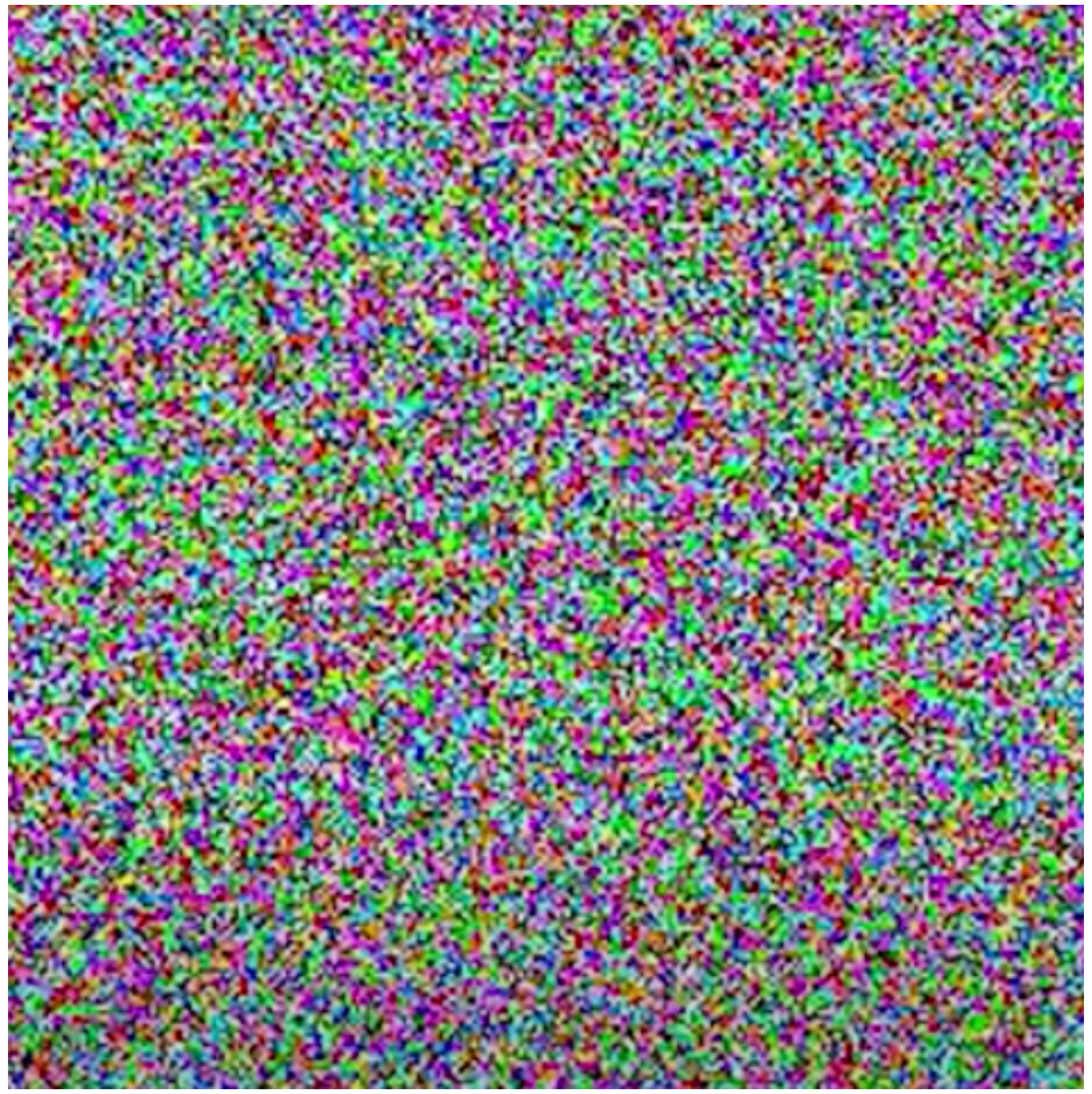
  - Randomly generate input image

  - Pick neurons to activate

  - Pass the random image into the network

  - Perform gradient* descent to activate the selected neurons

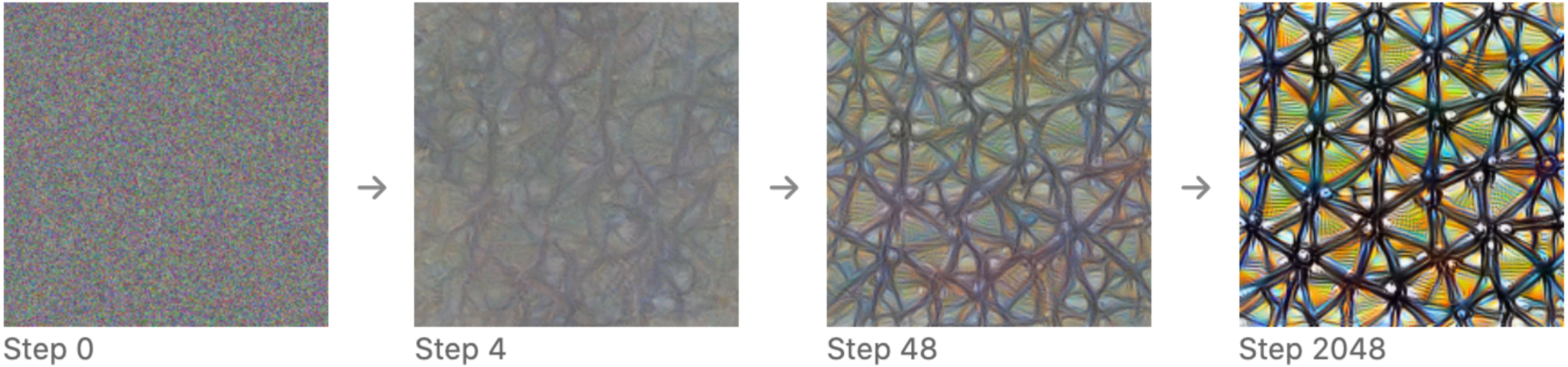
  *gradient here is computed w.r.t. to input noise, not network parameter



Credits: https://distill.pub/2017/feature-visualization/

Gradient optimization-based feature visualization

- The generated image improves with gradient descent steps



Credits: https://distill.pub/2017/feature-visualization/

Gradient optimization-based feature visualization

- Multiple neurons can be selected at once

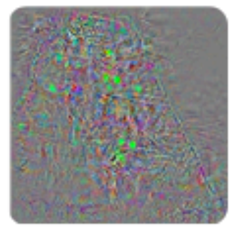
Credits: https://distill.pub/2017/feature-visualization/

Regularization

- Few selected regularization research. Again, regularization on noise input. The NN is fixed.

Weak Regularization avoids misleading correlations, but is less connected to real use.

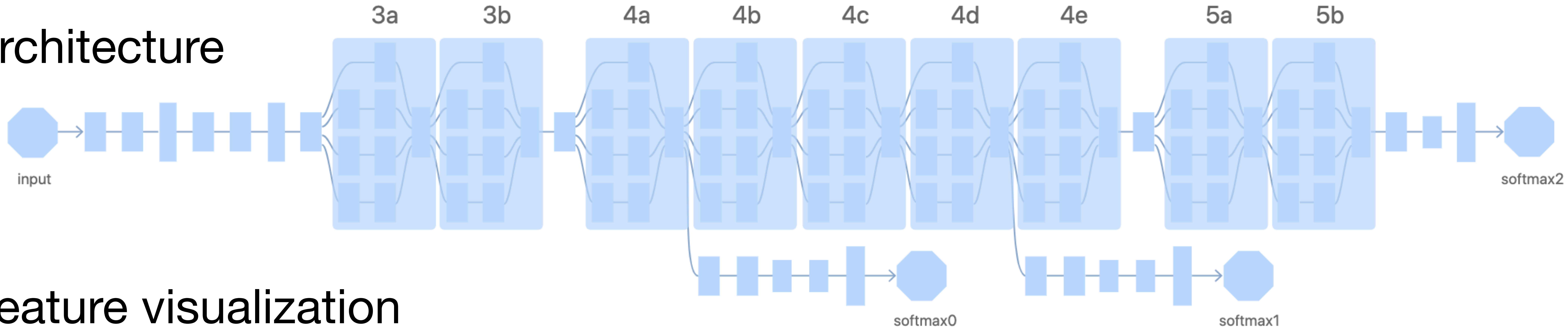
Strong Regularization gives more realistic examples at risk of misleading correlations.

		Unregularized	Frequency Penalization	Transformation Robustness	Learned Prior	Dataset Examples
	**Erhan, *et al.*, 2009** [3] Introduced core idea. Minimal regularization.	■	□	□	□	□
	**Szegedy, *et al.*, 2013** [11] Adversarial examples. Visualizes with dataset examples.	■	□	□	□	■
	**Mahendran & Vedaldi, 2015** [7] Introduces total variation regularizer. Reconstructs input from representation.	□	■	□	□	□
	**Nguyen, *et al.*, 2015** [14] Explores counterexamples. Introduces image blurring.	□	■	□	□	□
	**Mordvintsev, *et al.*, 2015** [4] Introduced jitter & multi-scale. Explored GMM priors for classes.	□	□	■	■	□
	**Øygard, *et al.*, 2015** [15] Introduces gradient blurring. (Also uses jitter.)	□	■	■	□	□

Credits: https://distill.pub/2017/feature-visualization/

GoogLeNet

- Architecture



- Feature visualization

Edges (layer conv2d0) | Textures (layer mixed3a) | Patterns (layer mixed4a) | Parts (layers mixed4b & mixed4c) | Objects (layers mixed4d & mixed4e)

[1] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

DeepDream (nightmare): CNN enhanced image patterns

Layer/DeepDream

$\texttt{layer}_n\texttt{[:,:,:]}^2$

https://www.youtube.com/watch?v=FFjGdNl5Sso

How DeepDream is constructed

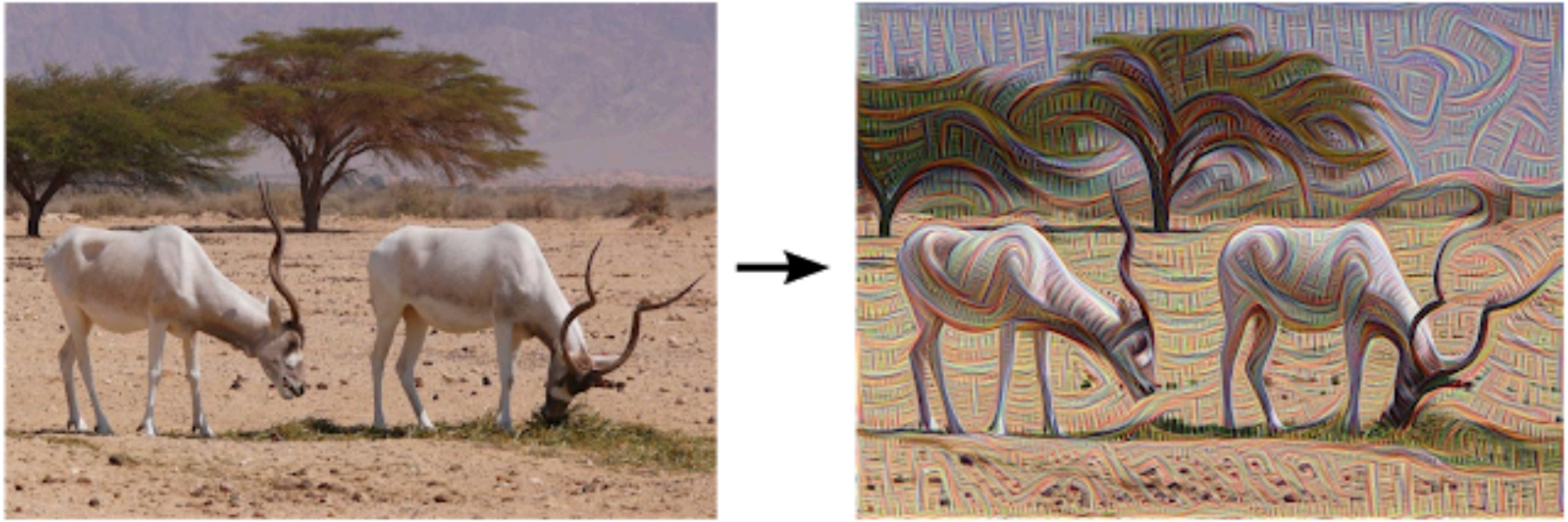
- "Whatever you see there, I want more of it!"
    - Enhance deeper layer
    - Enhance multiple channels
    - Things go wild when going deeper

How DeepDream is constructed

Layer/DeepDream

`layer`$_n$`[:,:,:]`^{2}

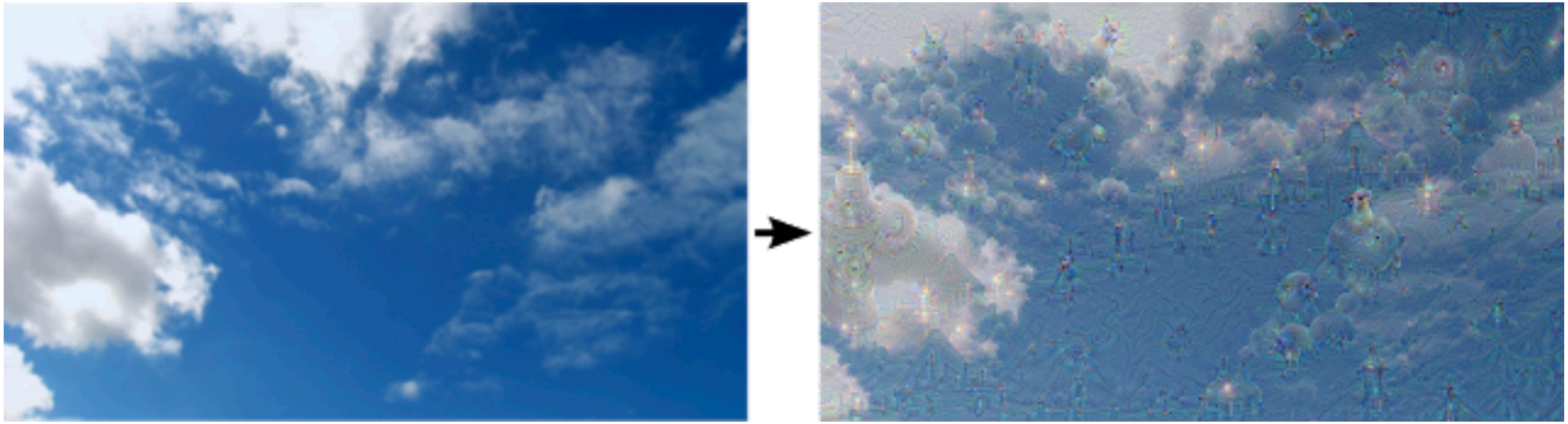
- “Whatever you see there, I want more of it!”
  - Lower layer



How DeepDream is constructed

Layer/DeepDream
$\texttt{layer}_n[:,:,:]^2$

- “Whatever you see there, I want more of it!”
    - Deeper layer (this one is for birds)



How DeepDream is constructed

- "Whatever you see there, I want more of it!"
    - Enhance deeper layer
    - Enhance multiple channels

Hands-on time: a neural network playground

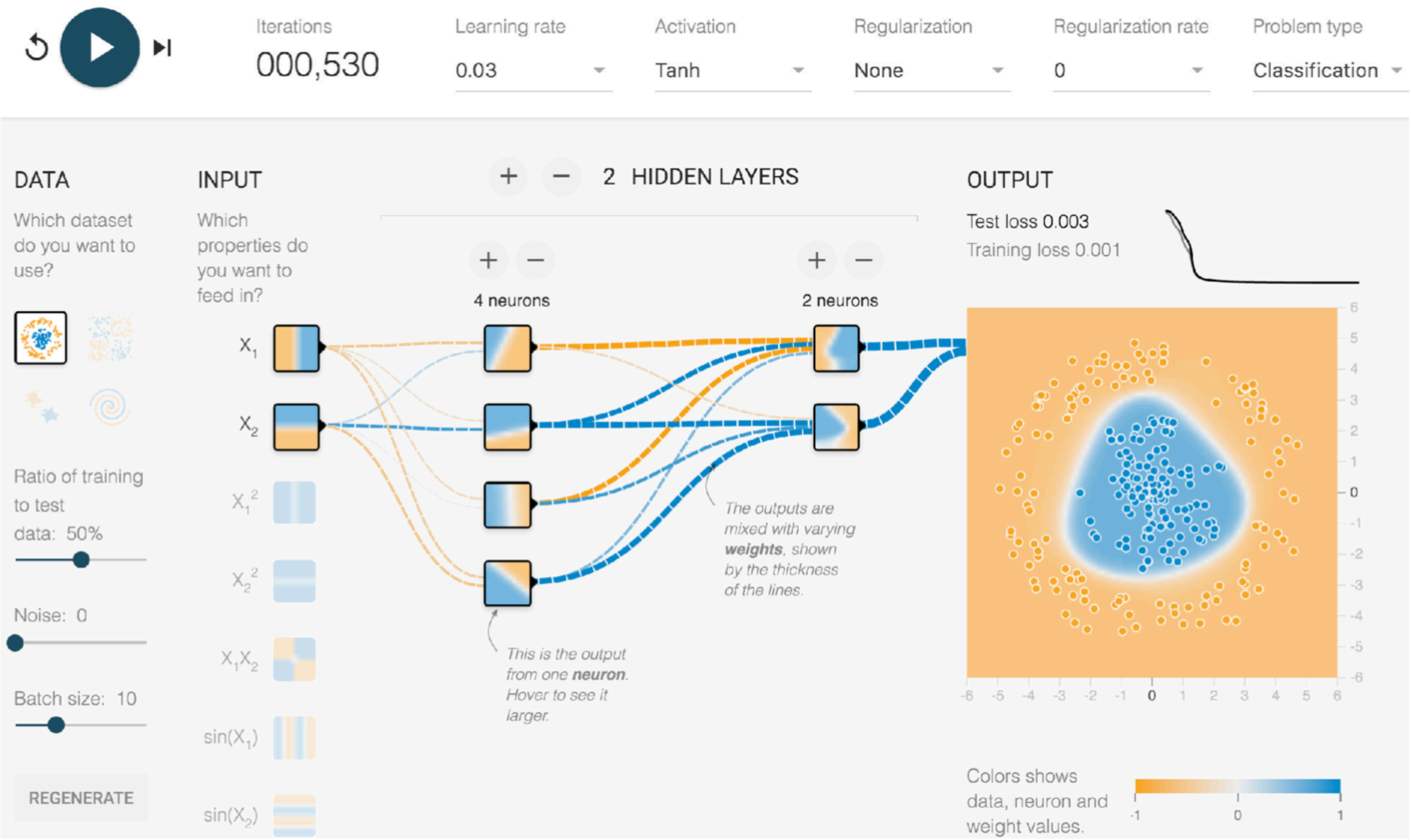


Figure 1: TensorFlow Playground. This network is, roughly speaking, classifying data based on distance to the origin. Curves show weight parameters, with thickness denoting absolute magnitude and color indicating sign. The feature heatmaps for each unit show how the classification function (large heatmap at right) is built from input features, then near-linear combinations of these features, and finally more complex features. At upper right is a graph showing loss over time. At left are possible features; x_1 and x_2 are highlighted, while other mathematical combinations are faded to indicate they should not be used by the network.

Credit: A Neural Network Playground

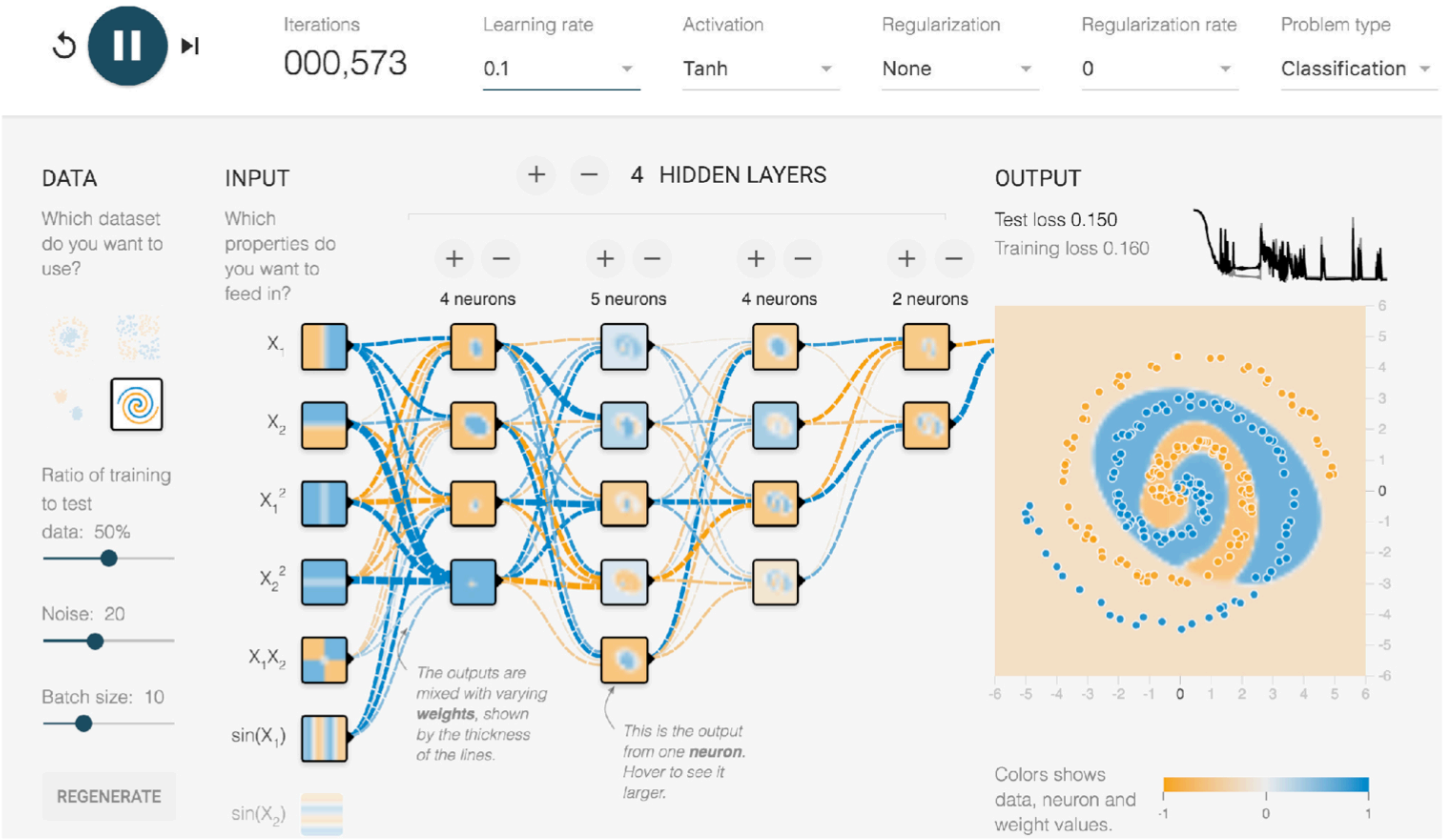


Figure 2: A complex configuration of TensorFlow Playground, in which a user is attempting to find hyper-parameters that will allow the classification of spiral data. Many possible feature combinations have been activated.

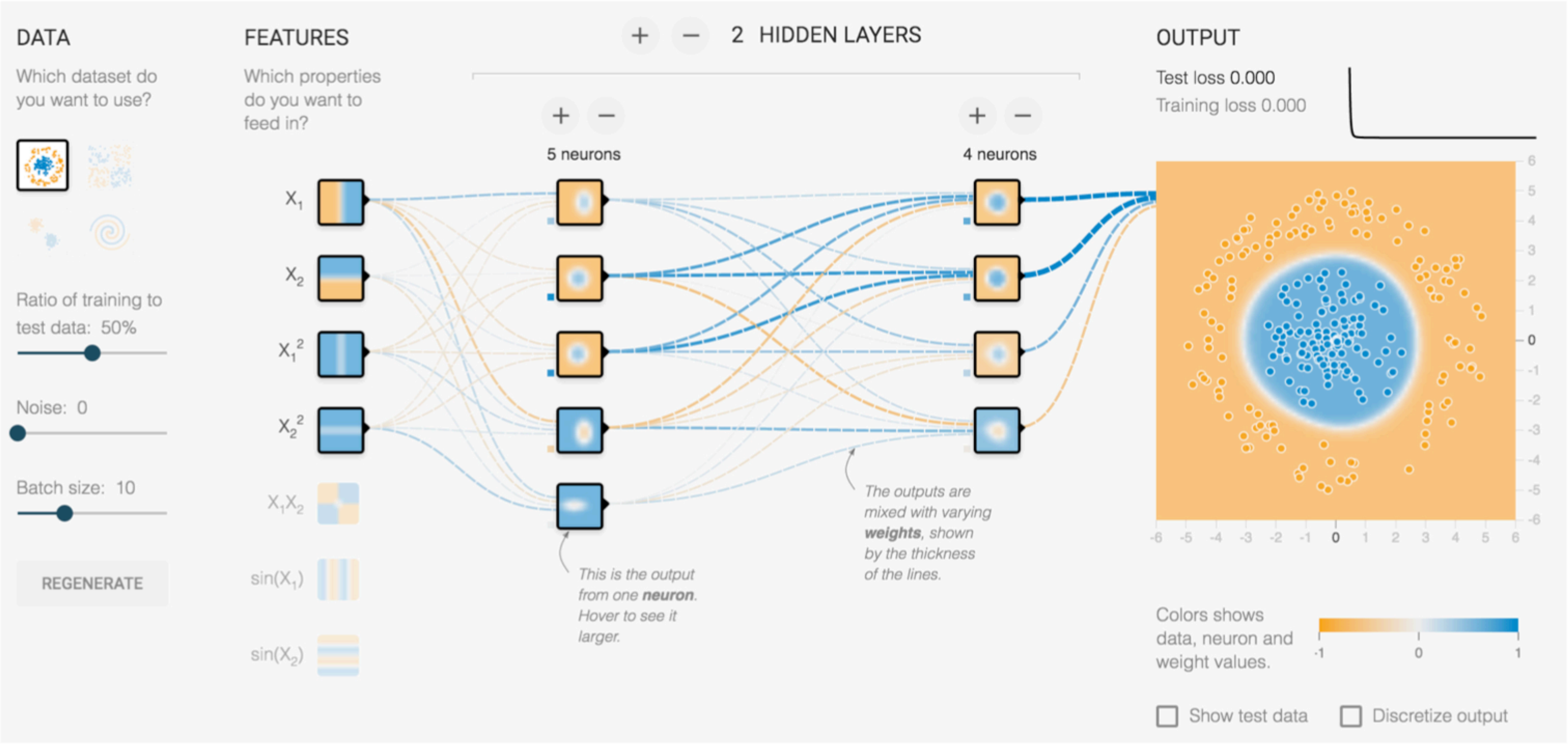


Figure 3: A network architecture with redundant layers and units. Several units in the first hidden layer have already essentially learned to classify the data, as seen by inspecting the in-network activation visualizations.

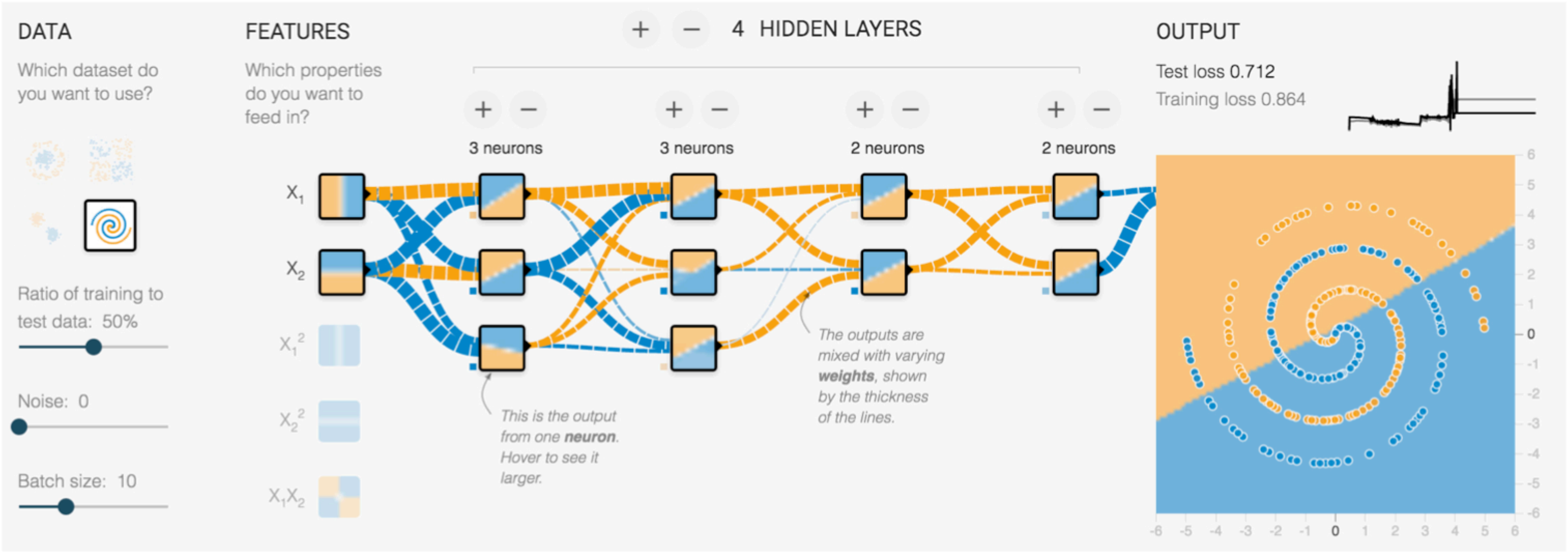


Figure 4: This network has completely failed to classify the data, even after many epochs. The high-contrast activation visualizations and thick weight connections hint at a systemic problem. This diagram was the result of setting the learning rate to the maximum speed.

Summary

- Feature visualization: studying what makes neurons activated
  - Data-driven approach
  - Optimization-based approach
- Next: Adversarial examples

Worth Reading

- **Classic paper on deconvolution**
  Matthew Zeiler and Rob Fergus, Visualizing and understanding convolutional networks. European conference on computer vision. Springer, 2014.
  https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53

- **Optimization-based feature visualization**
  Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature Visualization.
  https://distill.pub/2017/feature-visualization

- **A neural network playground**
  Smilkov, Daniel, Shan Carter, D. Sculley, Fernanda B. Viégas, and Martin Wattenberg. Direct-manipulation visualization of deep networks. arXiv 2017.
  https://arxiv.org/abs/1708.03788