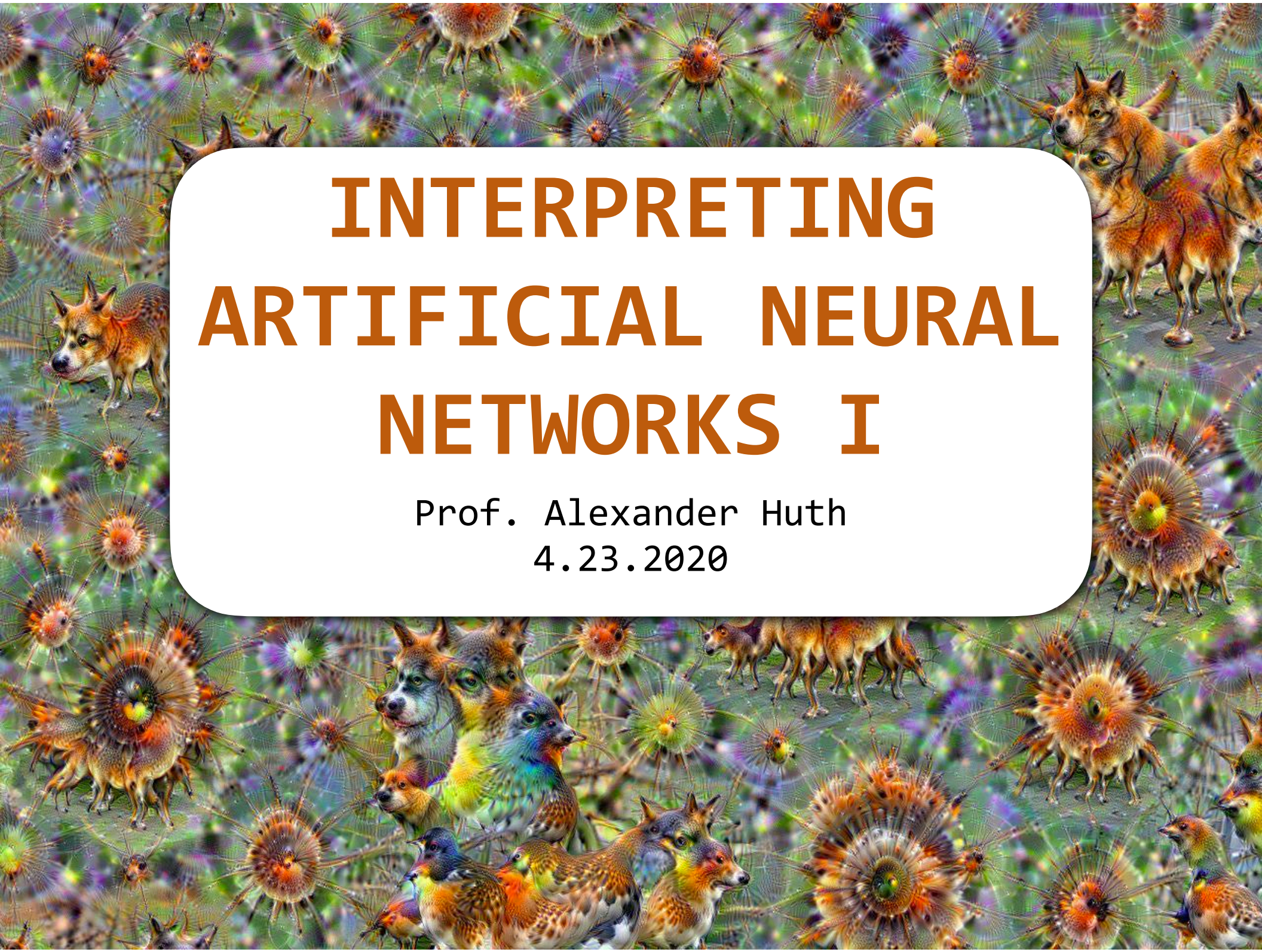


INTERPRETING ARTIFICIAL NEURAL NETWORKS I

Prof. Alexander Huth

4.23.2020

PROJECT WRITE-UP

* Write-up due **May 5** (before class)
* Should be a PDF **<= 4 pages** including figures but *not* including references (i.e. references can appear on page 5+)
* Should discuss:
  * **Motivation** (what is problem, why are you doing what you are doing)
  * **Background** (related things in the literature, preferably >= 5 references)
  * **Methods** (what you did, & why) *<– most important*
  * **Results** (what you found) *<– least important*
  * **Individual contributions** (what each person in group was responsible for, including ideation, execution, & writing)

PROJECT PRESENTATION

* Presentations will occur during class periods **May 5 & 7**
* ~6 presentations per class (assignments will be randomized & posted on Canvas)
* Each presentation should be 8-9 minutes, with 2-3 minutes for questions from other students
* Presentations should cover (at least) **Motivation**, **Methods**, & **Results**
* **Every member of the group** must participate in the presentation
* You may submit a recorded presentation (with same parameters as above) instead of doing it live if you strongly prefer

RECAP

* Artificial neural networks
  * Convolutional neural networks (e.g. ImageNet)
  * Recurrent neural networks (& LSTMs)
* System construction approach
  * Build neural networks that solve a task, ask whether it solves that task similarly to how humans solve that task

INTERPRETING ANNS

* For system construction to be useful, do we also need to **understand** the representations in the artificial networks?

* Is our **new problem** (understanding the representations in *artificial* networks) easier than our **old problem** (understanding representations in *biological* networks)?

  * Or are we just *“replacing one thing we don’t understand with another thing we don’t understand”*?

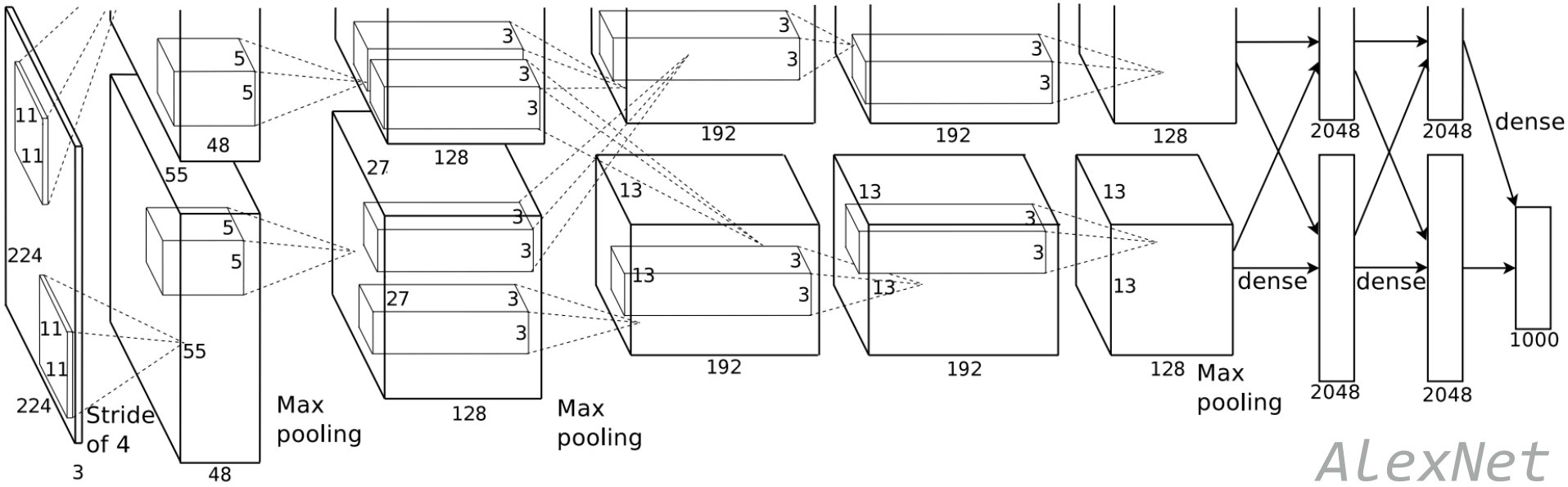
INTERPRETING ANNS

* My opinion: the new problem is **easier** than the old problem
  * The brain is a **classic black box**: we see what goes in & (partially) what comes out, but must infer computation
  * The ANN is an **open gray box**: we see exactly how computation occurs, even if we don’t understand it

So let’s try to understand it!

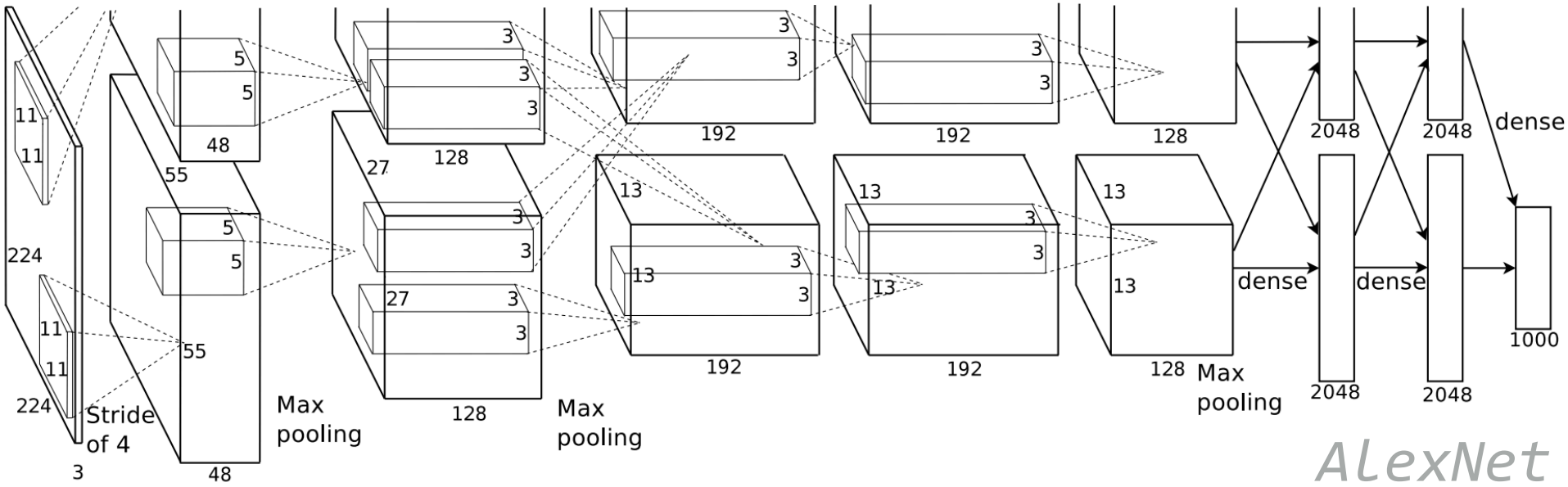
INTERPRETING ANNS

* Given a trained artificial neural network (e.g. a CNN trained to classify images), can we interpret the representations within that network?



INTERPRETING ANNS

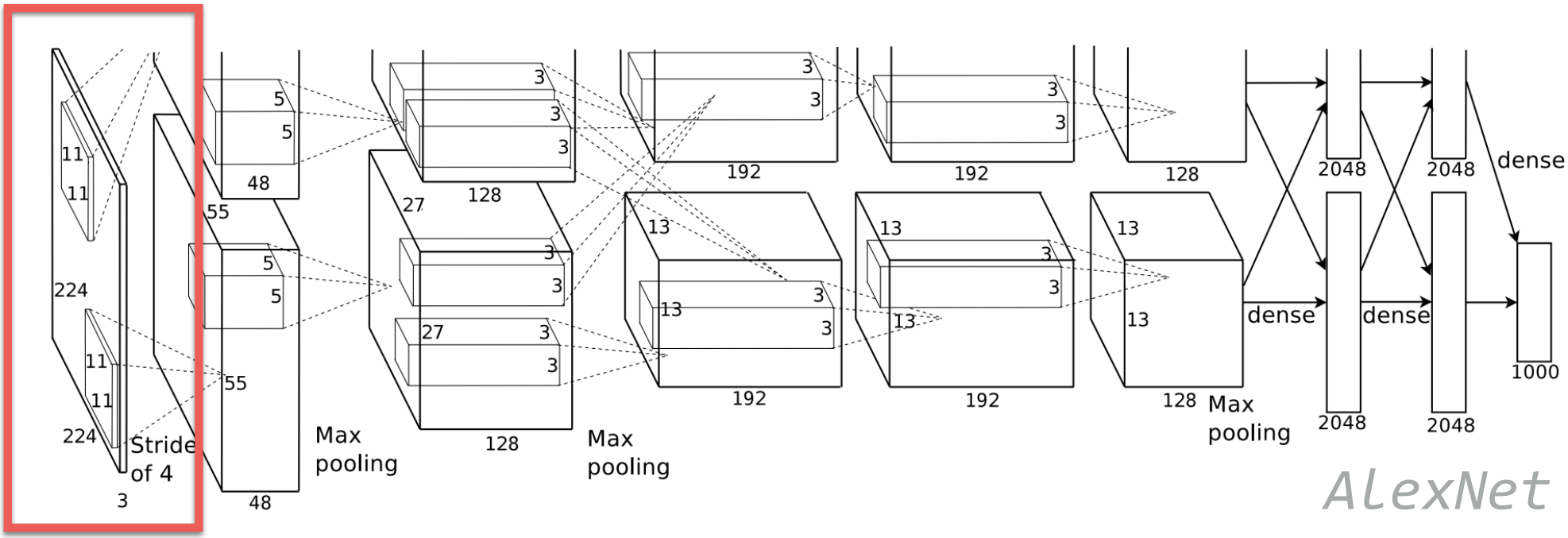
* Simpler: given a network, can we understand what *any one unit* in the network is representing?



AlexNet

INTERPRETING ANNS

* One particular layer is very easy to visualize: the first

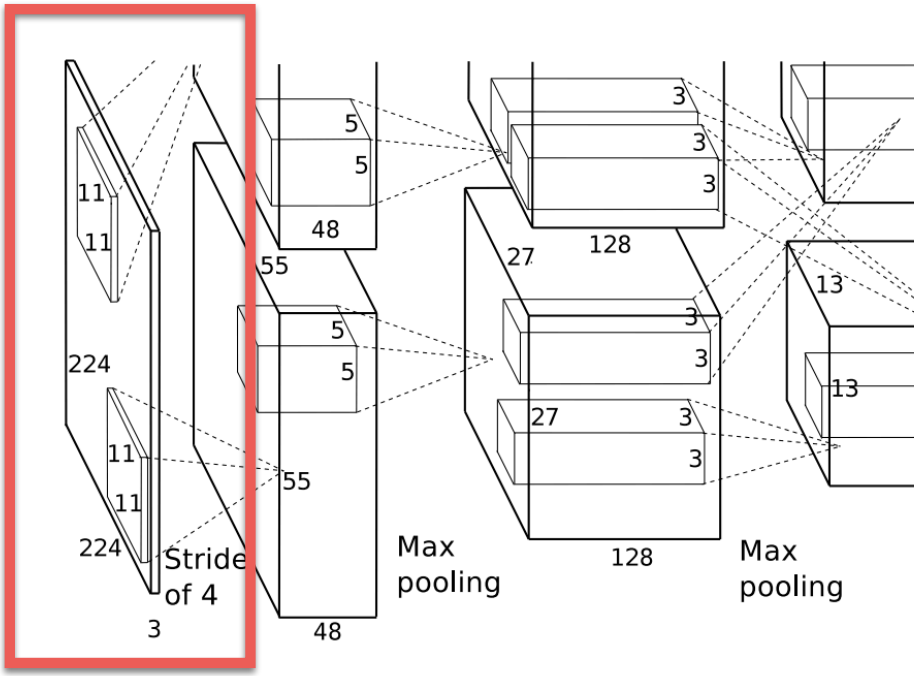


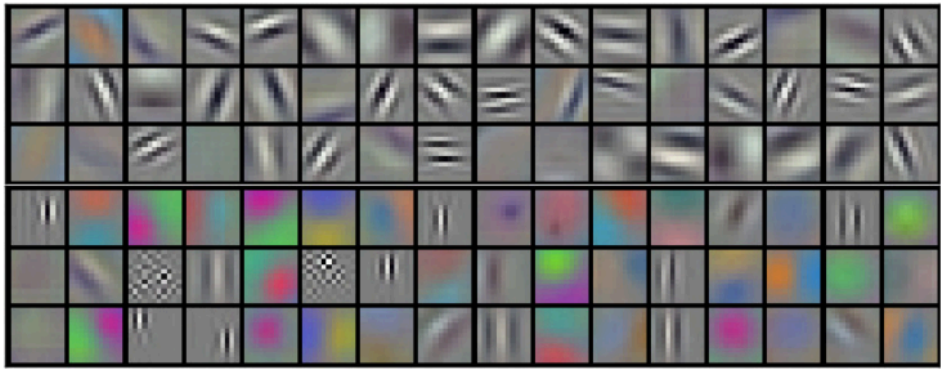
AlexNet

INTERPRETING ANNS

* One particular layer is very easy to visualize: the first

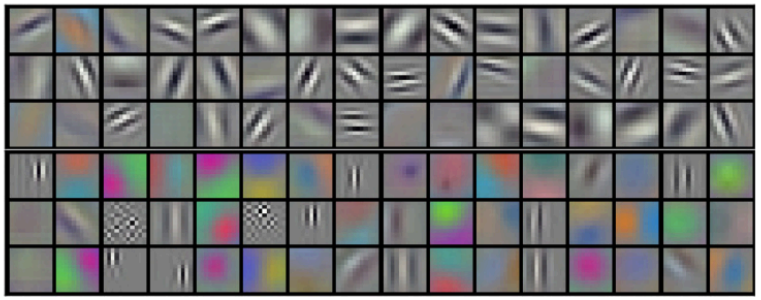
* Features learned by 1st layer of AlexNet



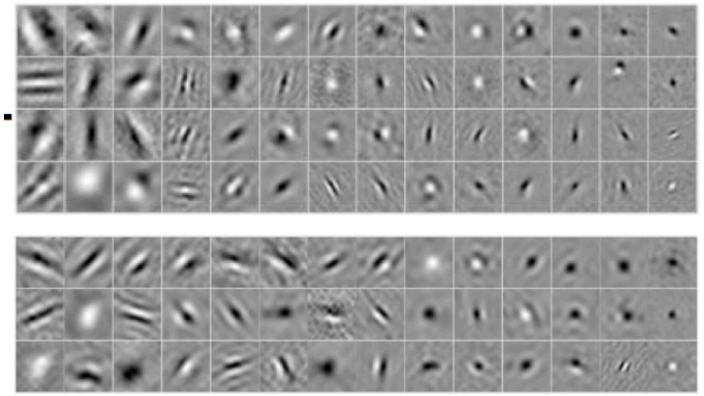


RECALL: ALEXNET

* Features learned by 1st layer of AlexNet



* Receptive fields of V1 neurons from a macaque



INTERPRETING ANNS

* What about units in **other layers** than the first?
  * We can easily view their weights, but the weights do not form images
  * Because these are weights on *activations* of units from earlier layers, not inputs

INTERPRETING ANNS

* Currently we have two approaches to understanding unit representations:
  * **Feature visualization (today):** what types of inputs drive unit *i* in the network? (*receptive field estimation*!)
  * **Attribution (next lecture):** what is it about input *X* that drives unit *i* in the network to respond (or not respond)?

FEATURE VISUALIZATION

* **Question:** what types of inputs drive unit i in the network?

* **Possible answer 1:**

  * take a large set of inputs $[X_1, X_2, \ldots, X_n]$

  * find response of unit i to each input, $[Y_{i,1}, Y_{i,2}, \ldots, Y_{i,n}]$

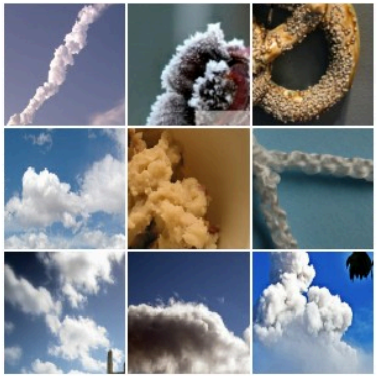
  * sort inputs by unit i response, then inspect the inputs with the largest (& smallest?) values

FEATURE VISUALIZATION

* **Example** units from layer *mixed4a* from GoogLeNet

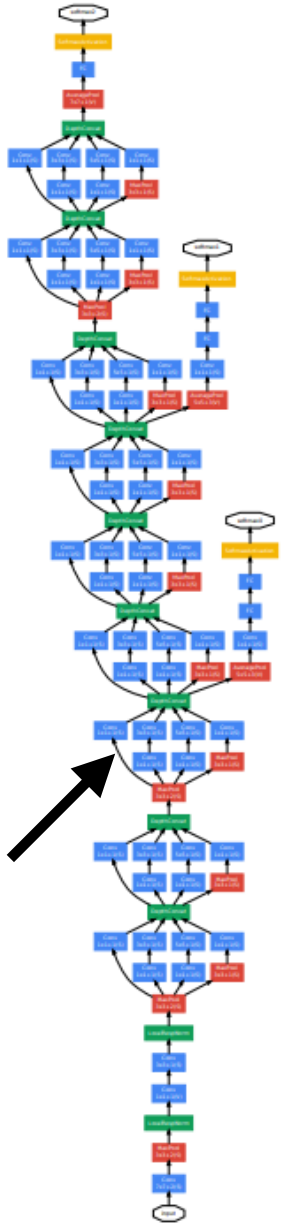
stripes?

animal faces?



fluffy things?

buildings w/ sky?



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

FEATURE VISUALIZATION

* **Question:** what types of inputs drive unit i in the network?

* **Possible answer 1:** (inspect inputs w/ greatest response)

* Does this work?

FEATURE VISUALIZATION

* **Question:** what types of inputs drive unit i in the network?

* **Possible answer 2:**

  * Use knowledge of network structure to create an input that maximizes Y_i, the activation of unit i [Erhan, 2009]

  * Formally: $x^* = \underset{x}{\operatorname{argmax}}\, Y_i(x)$

FEATURE VISUALIZATION

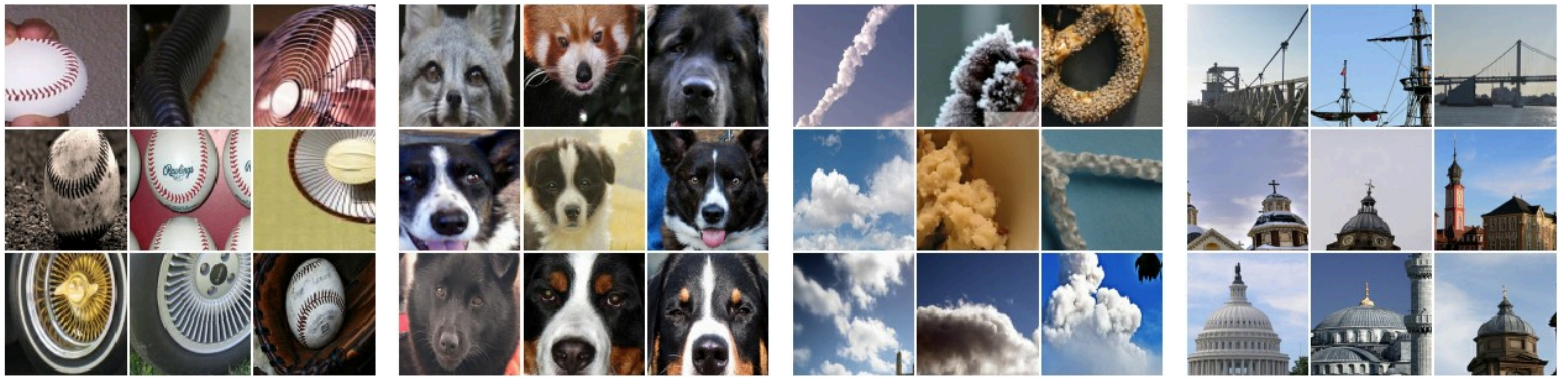
* Since we have *complete* & *exact* knowledge of how activation Y_i arises from the input, we can approximate this maximization using **gradient ascent**, i.e.

$$x^{(l+1)} = x^{(l)} + \epsilon \frac{\partial Y_i(x^{(l)})}{\partial x^{(l)}}$$

(In practice x is also regularized to ensure that it doesn't explode or do other silly things that real inputs can't do)

FEATURE VISUALIZATION

Best Inputs



Optimized Inputs

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

FEATURE VISUALIZATION

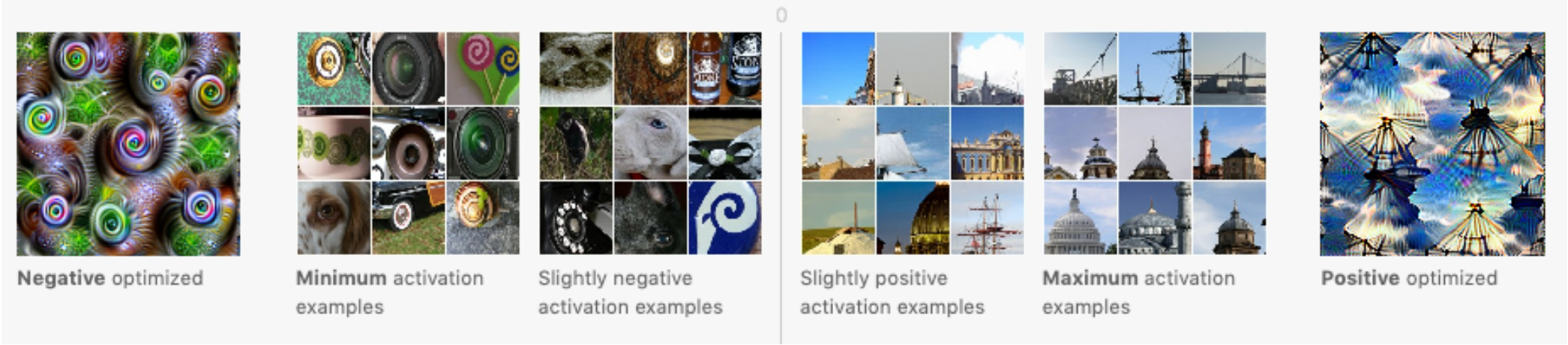
* **Question:** what types of inputs drive unit i in the network?

* **Possible answer 2:** (find an optimal input using gradient ascent)

* Does this work? Does it work better?

FEATURE VISUALIZATION

* Both of these techniques can also be used to explore negative (and intermediate) activations, potentially giving a more complete picture



it doesn't like spirals, round things?

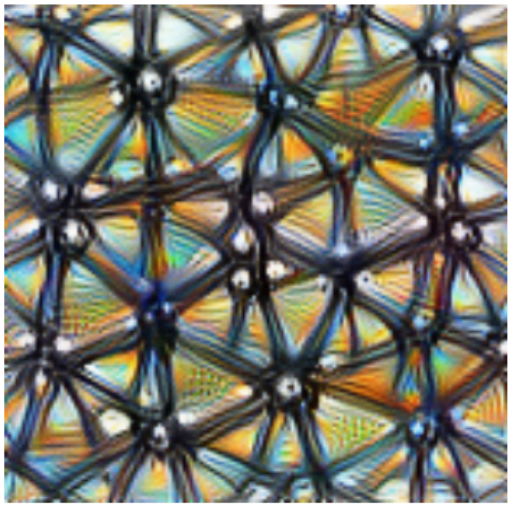
it likes pointy things in the sky?

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

FEATURE VISUALIZATION

* The optimization technique can also be used to optimize the activation of entire **channels** or **layers** in the network

One Unit



Whole Layer

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

FEATURE VISUALIZATION

* **Question:** what types of inputs drive unit *i* in the network?
  * **Possible answer 1:** (inspect inputs w/ greatest response)
  * **Possible answer 2:** (find an optimal input using gradient ascent)
* *Which works better? Do these techniques give satisfactory answers or interpretations? What would a satisfactory answer be?*
  * Let's discuss these questions in breakout rooms for ~10 minutes, then share thoughts as a group

NEXT TIME

* Attribution in neural networks

* –> Transformer networks (aka multi-head attention)