Twitter Thread by Radek Osmulski

Radek Osmulski
@radekosmulski

THREAD: How is it possible to train a well-performing, advanced Computer Vision model ■■ ■■■ ■■■? ■

At the heart of this lies the most important technique in modern deep learning - transfer learning.

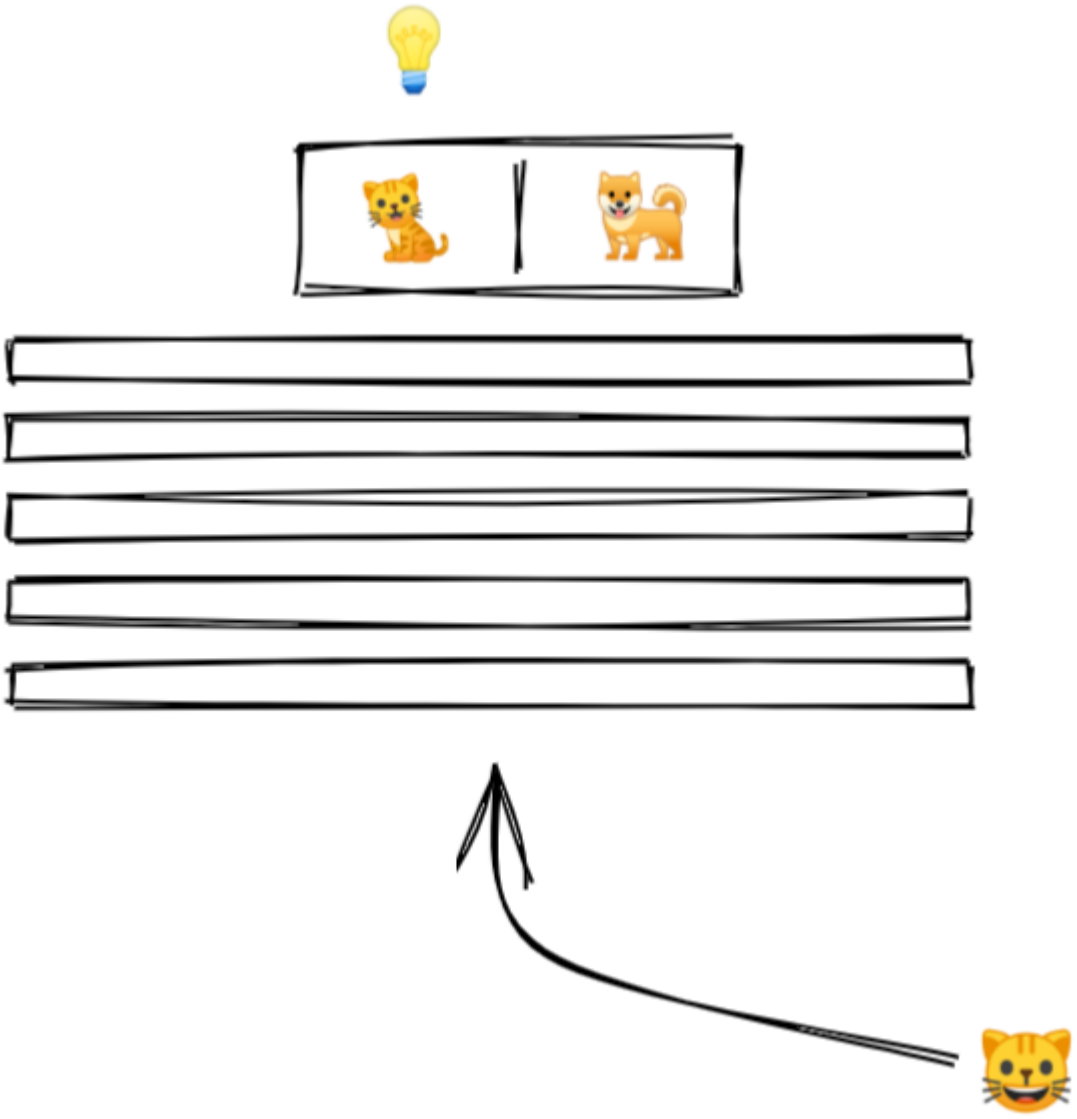
Let's analyze how it

> THREAD: Can you start learning cutting-edge deep learning without specialized hardware? \U0001f916
>
> In this thread, we will train an advanced Computer Vision model on a challenging dataset. \U0001f415\U0001f408 Training completes in 25 minutes on my 3yrs old Ryzen 5 CPU.
>
> Let me show you how...
>
> — Radek Osmulski (@radekosmulski) February 11, 2021

2/ For starters, let's look at what a neural network (NN for short) does.

An NN is like a stack of pancakes, with computation flowing up when we make predictions.

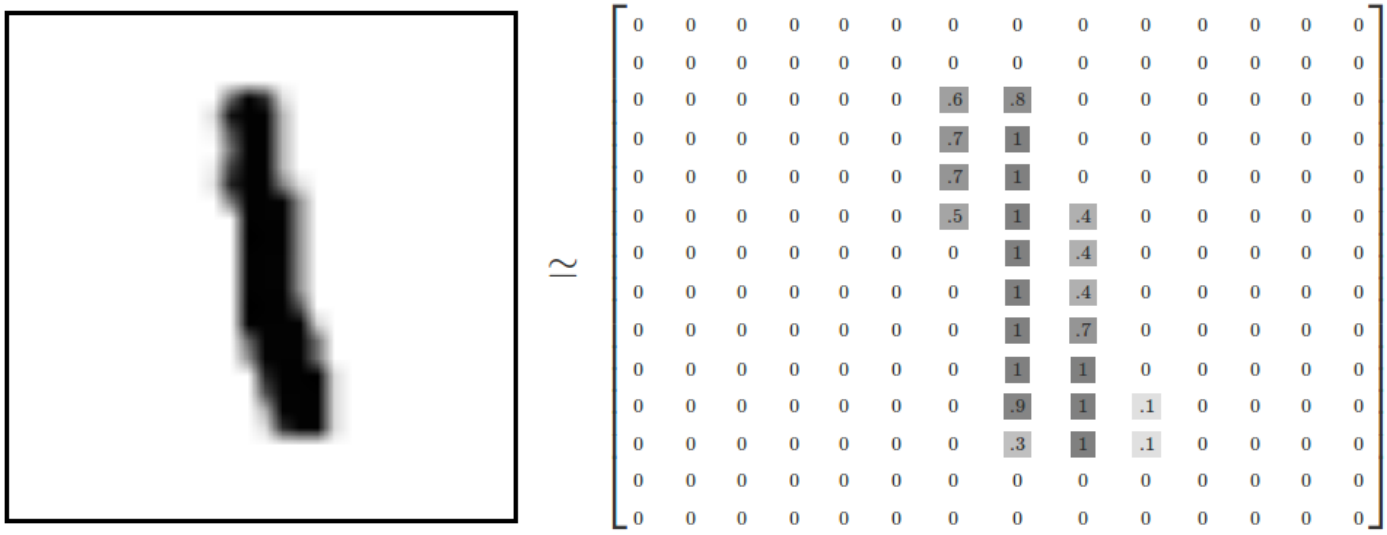
How does it all work?



3/ We show an image to our model.

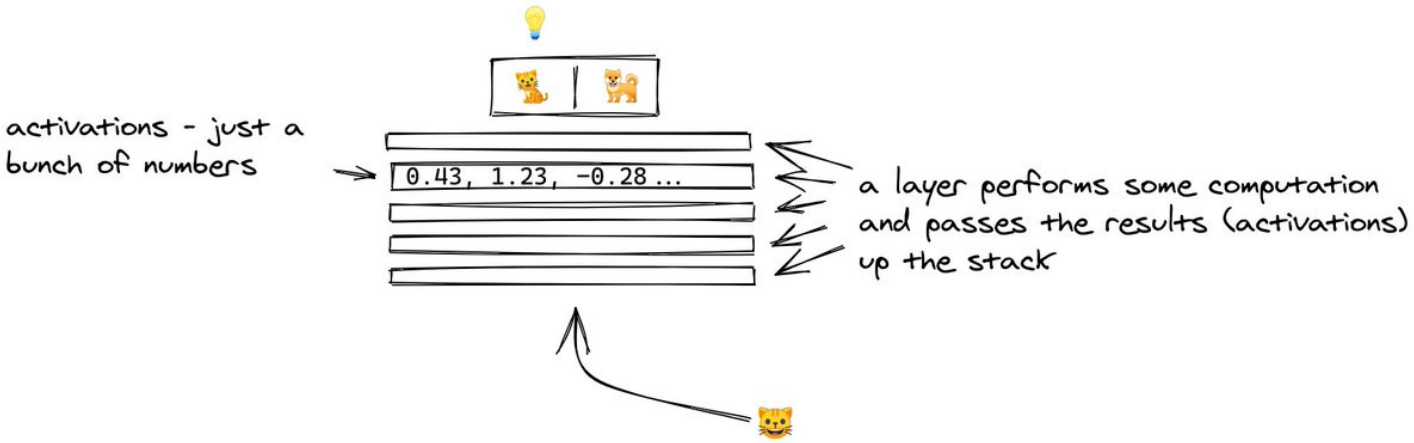
An image is a collection of pixels. Each pixel is just a bunch of numbers describing its color.

Here is what it might look like for a black and white image



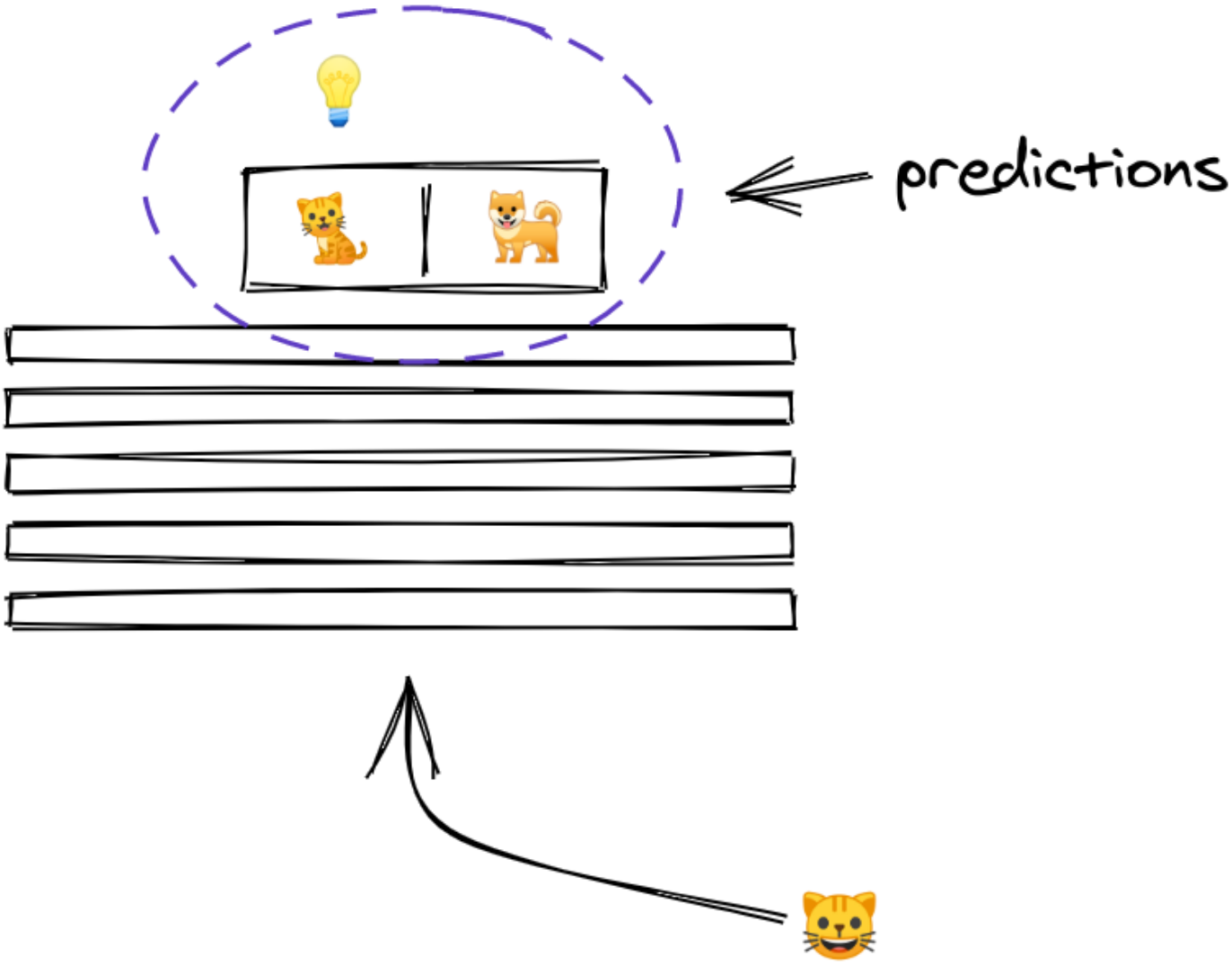
4/ The picture goes into the layer at the bottom.

Each layer performs computation on the image, transforming it and passing it upwards.



5/ By the time the image reaches the uppermost layer, it has been transformed to the point that it now consists of two numbers only.

The outputs of a layer are called activations, and the outputs of the last layer have a special meaning... they are the predictions!



6/ For a NN distinguishing between cats and dogs, when presented with an image of a cat we want the ■■■ neuron to light up!

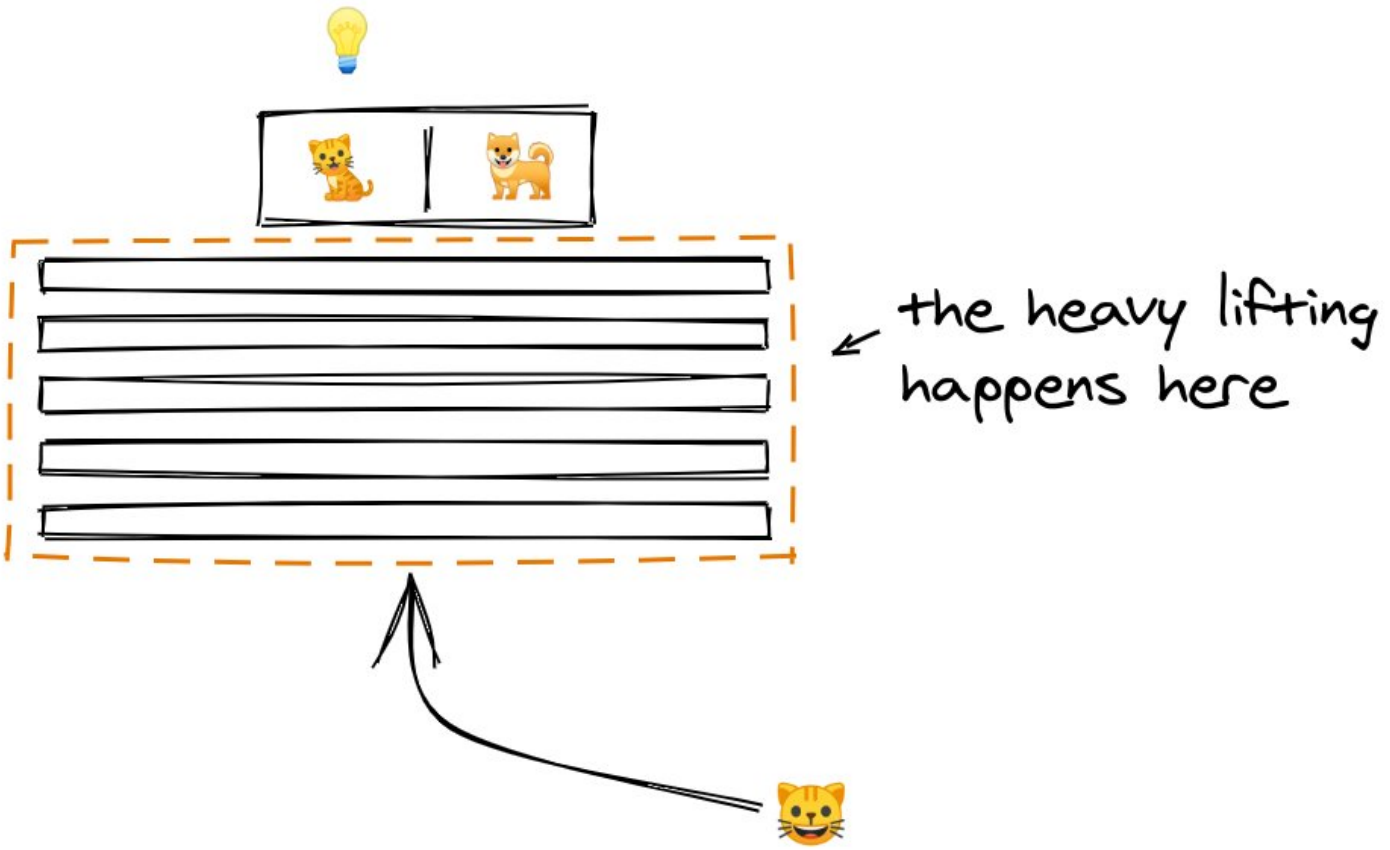
We would like for it to have a high value, and for other activations in the last layer to be small...

So far so good! But what about transfer learning?

7/ Consider the lower levels of our stack of pancakes! This is where the bulk of the computation happens.

We know that these layers evolve during training to become feature detectors.

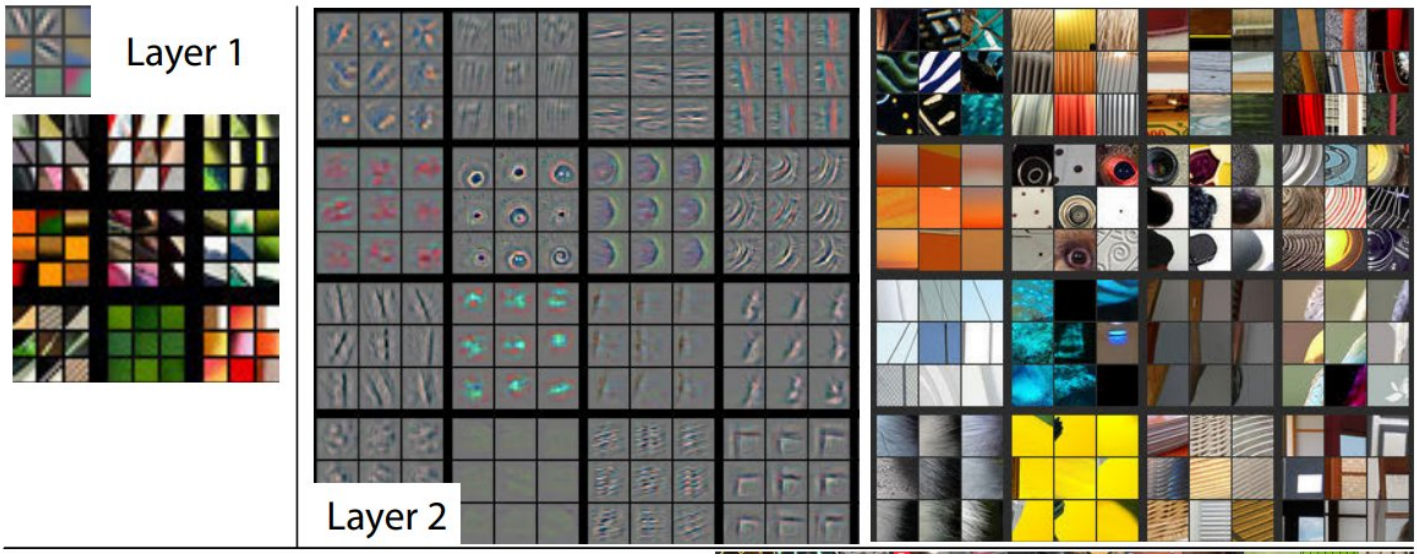
What do we mean by that?



8/ One layer may have tiny sliding windows that are good at detecting lines.

A layer above might have windows that construct shapes from these lines.

We might have a window light up when it sees a square, another when it sees a colorful blob.



9/ As we move up the stack, the features that windows can detect become more complex, building on the work of the layers below.

Maybe one sliding window will combine lines and detect text... maybe another one will learn to detect faces.

Does all of this sound like a hard task?

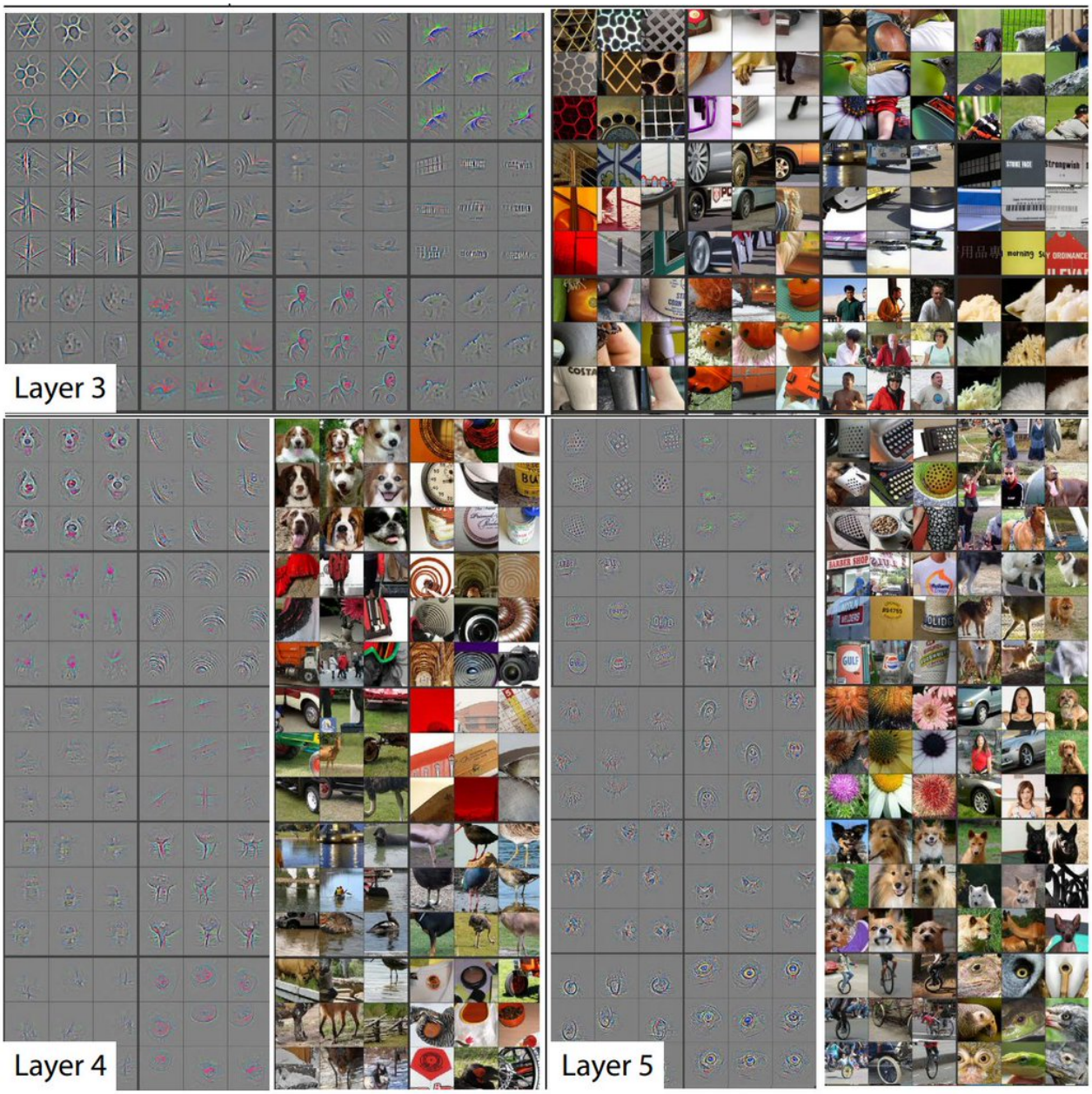


Figure 2. Visualization of features in a fully trained model. For layers 2-5 we show the top 9 activations in a random subset of feature maps across the validation data, projected down to pixel space using our deconvolutional network approach. Our reconstructions are *not* samples from the model: they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

10/ Absolutely! A network needs to see a lot of pictures to learn all of that.

But, presumably, once we detect all these lower-level features, we can combine them in a plethora of interesting ways? ■

11/ We can take all the lines, and the blobs, and the faces, or whatever the lower layers of the network can see, and combine them to predict cats and dogs!

Or trains, planes, and ships. Or blood cell boundaries. Or aneurysms in x-rays. The possibilities are endless!

12/ This is precisely what transfer learning is!

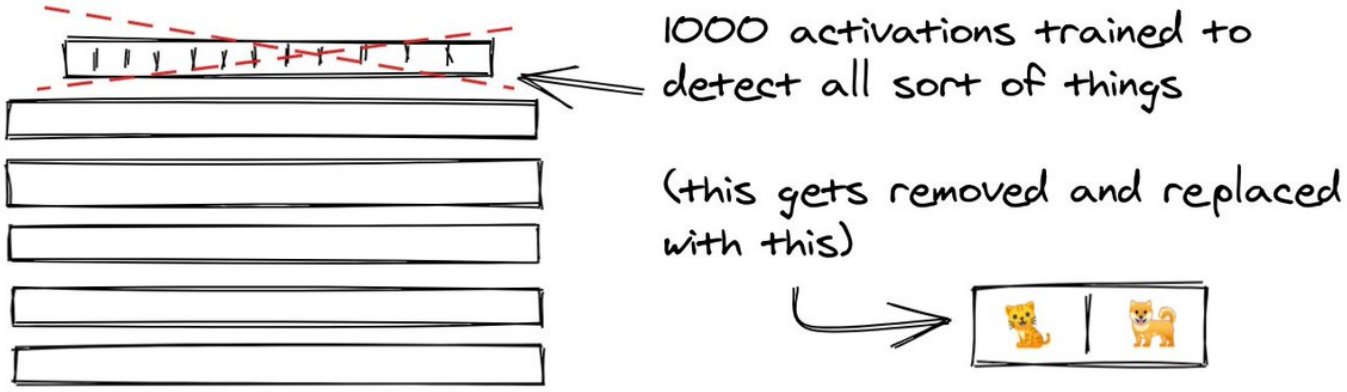
We let researchers, large corporations, spend millions of dollars to train very complex models.

And then we get to build on top of their work! ■

But so much for the theory. How does it all work in practice?

13/ In our example, we took a pretrained model that was trained on a subset of Imagenet consisting of 1.2 million images across 1000 classes!

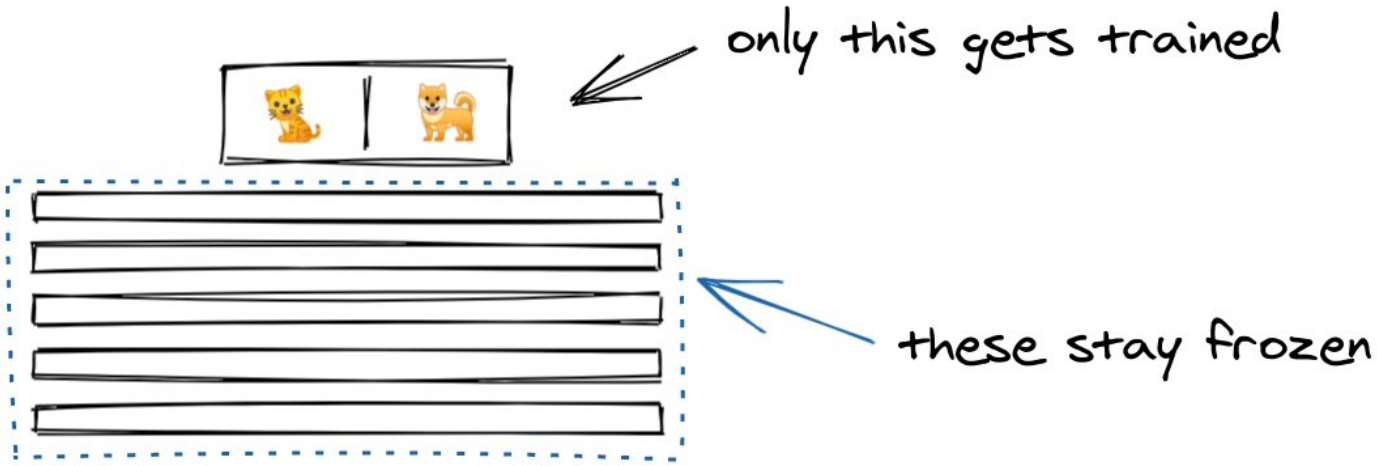
The @fastdotai framework downloaded the model for us and removed the top of it (the part responsible for predicting 1 of 1000 classes).



14/ It created a new head for our model, one tailored to the classes in the new dataset.

During training, we kept nearly the entire model frozen, and only trained the uppermost part, making use of all the lower level features that were being detected.

Ingenius! ■



15/ The concept of transfer learning, of utilizing a model trained on one task to perform another one, applies to other scenarios as well, including NLP (models that act on text).

We will hopefully get a chance to explore all of them ■

16/ I plan to explain all the concepts in modern AI in a similar fashion, assuming people find this useful ■

If you enjoyed this thread, let me know please and help me reach others who might also be interested ■■

And the visualizations of what the layers can detect?

17/ They come from this seminal paper - Visualizing and Understanding Convolutional Networks https://t.co/c3DMSTMc4T

Next stop - deciphering how it all works in code and finding ways to further improve our model!

Stay tuned for more ■