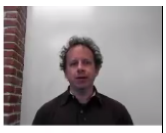
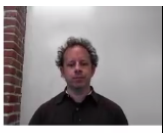


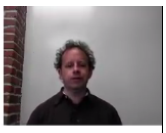
TOP-DOWN VS BOTTOM-UP

- Bottom-up: learn each building block you need, and eventually put them together
  - Hard to maintain motivation
  - Hard to know the "big picture"
  - Hard to know which pieces you'll actually need
- fast.ai: Get students using a neural net right away, getting results ASAP
  - Gradually peel back the layers, modify, look under the hood

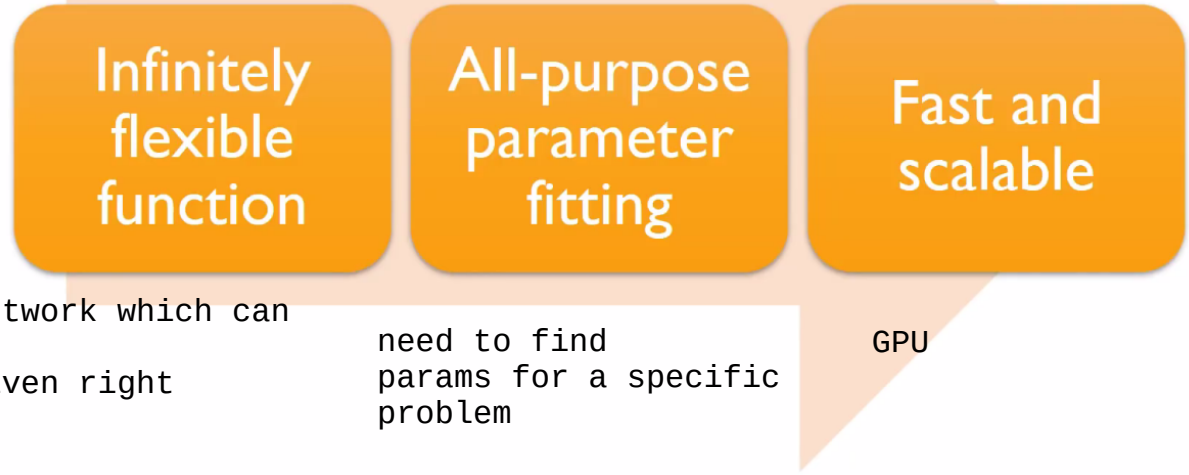


"I personally fell into the habit of watching the lectures too much and googling definitions / concepts / etc too much, without running the code. At first I thought that I should read the code quickly and then spend time researching the theory behind it…

"In retrospect, I should have spent the **majority of my time on the actual code in the notebooks** instead, in terms of running it and seeing that goes into it and what comes out of it"



SO Deep Learning Algoriths are the ones which fit the following requirements



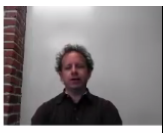
means a network which can solve any problem given right parameters

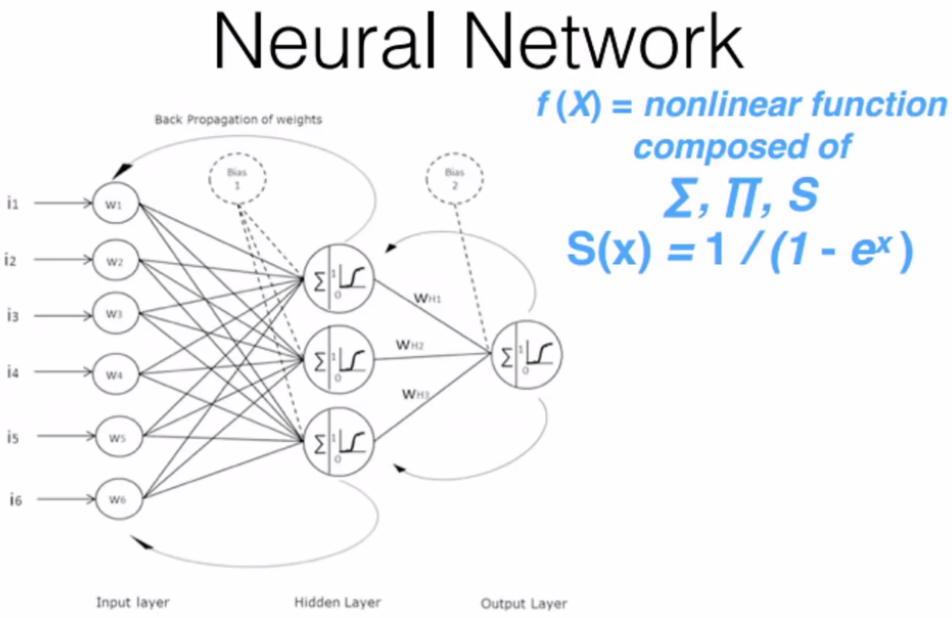
need to find params for a specific problem

GPU

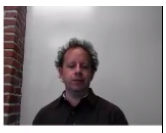
In a Neural Network, when a number of simple linear layers are in-dispersed with number of simple non-linear layers, when you get is called universal approximation theorem and it says that it can solve any problem to a very close accuracy given enough number of parameters. It has provably been shown to be infinitely flexible fn.

more at:
https://www.youtube.com/watch?v=Ijqkc7OLenI





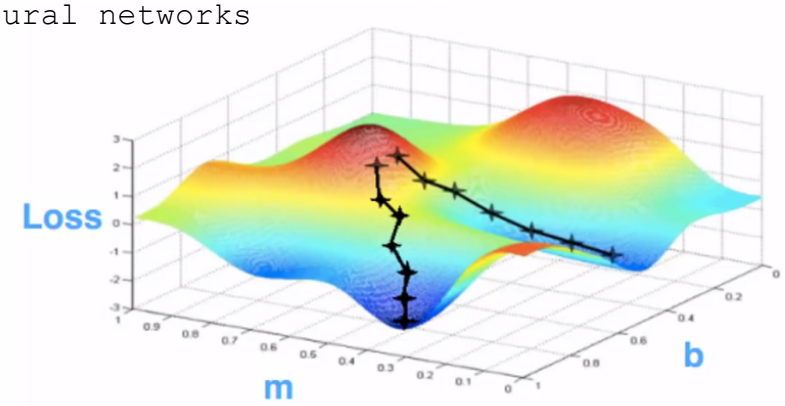
Even this neural network with single hidden layer does support universal approximation theoroem but it requires exponential number of params, so not fast and scalable.
But also been discovered that if we add more hidden layers to such networks, so linear scaling with more accuracy for complex problems and such huge networks with many hidden layers and more parameters than a single hidden layer this is called deep learning



Gradient Descent

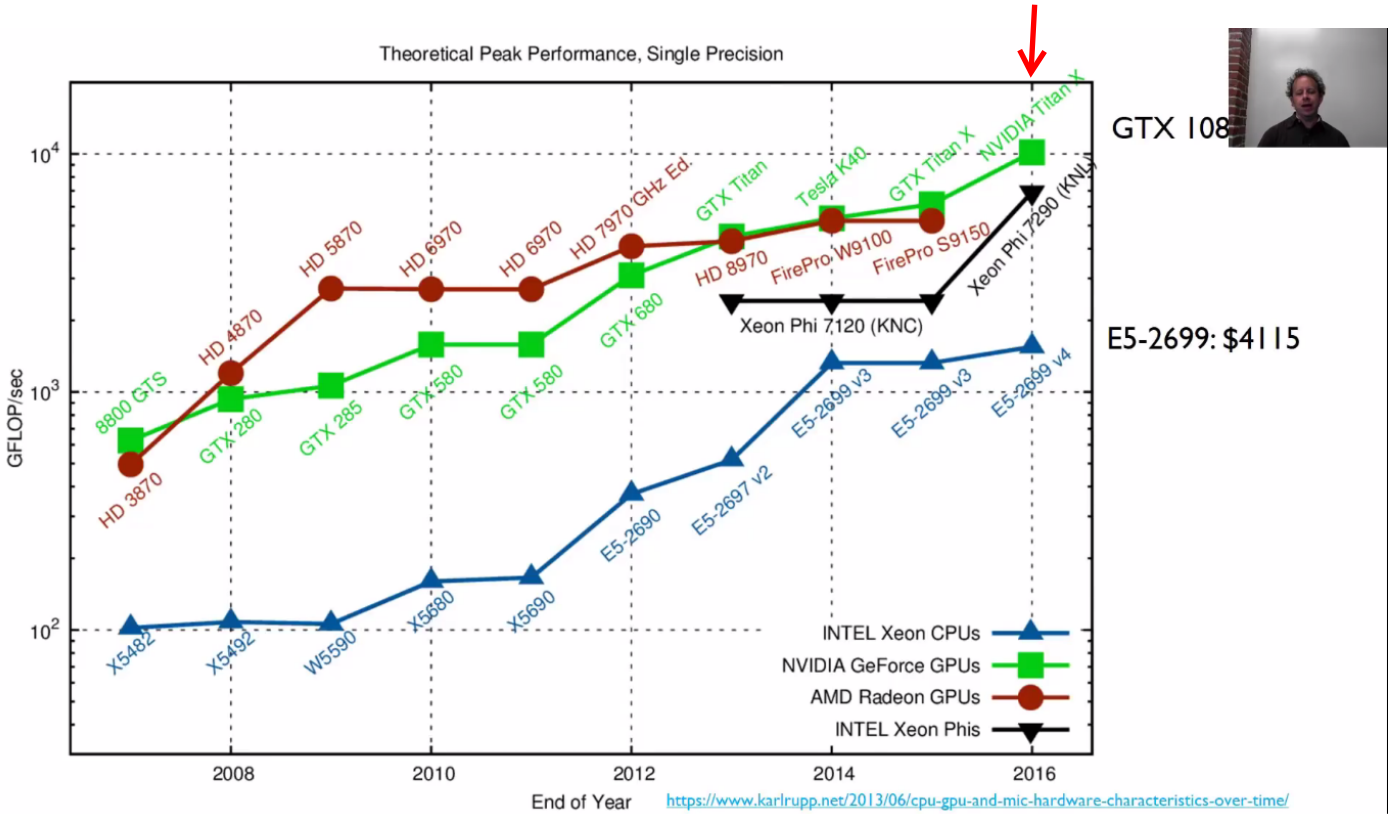
GD is an excellent way for fitting parameters to neural networks

f (x) = nonlinear function of x



We can compute these parameters very fast owing to GPUs

NVIDIA titan X 10 times fast than E5 CPU still costs around 700 bucks while CPU = $4115



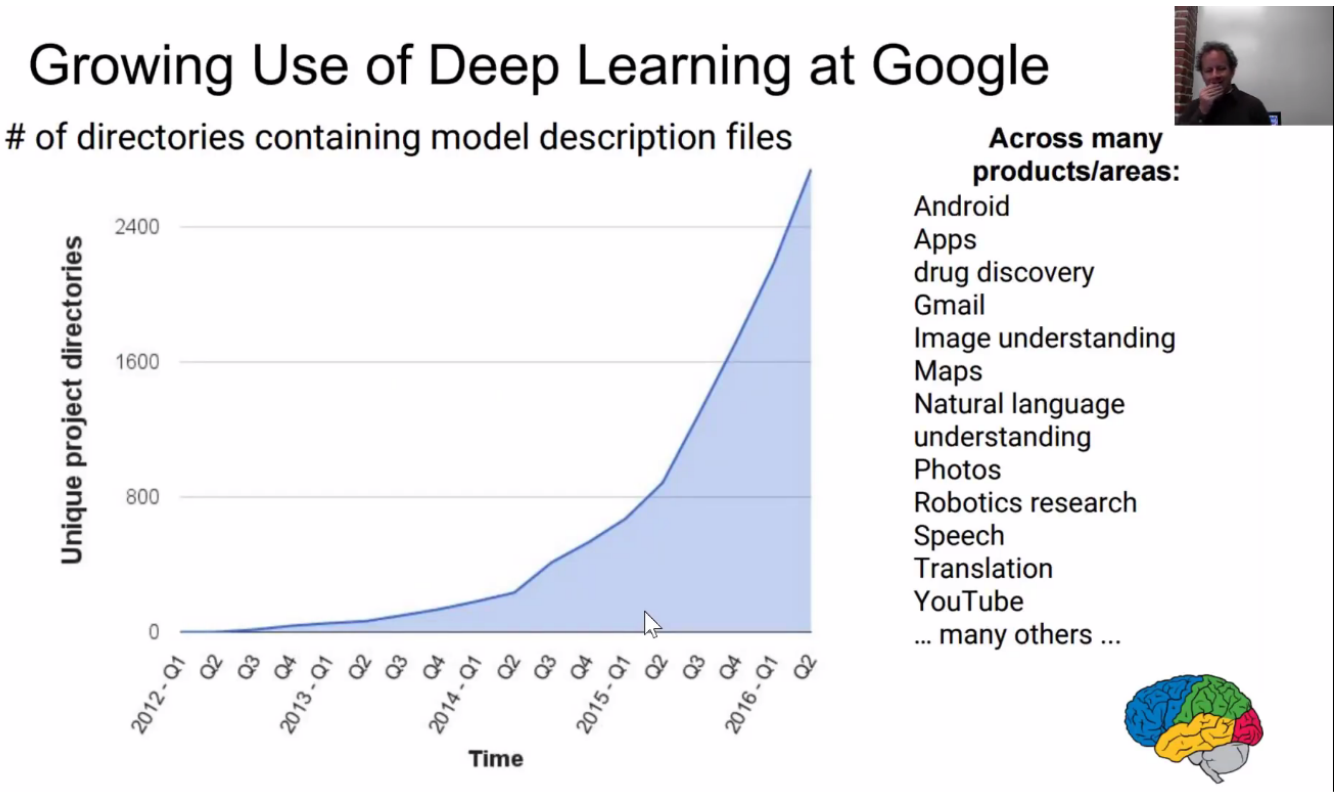
GTX 108

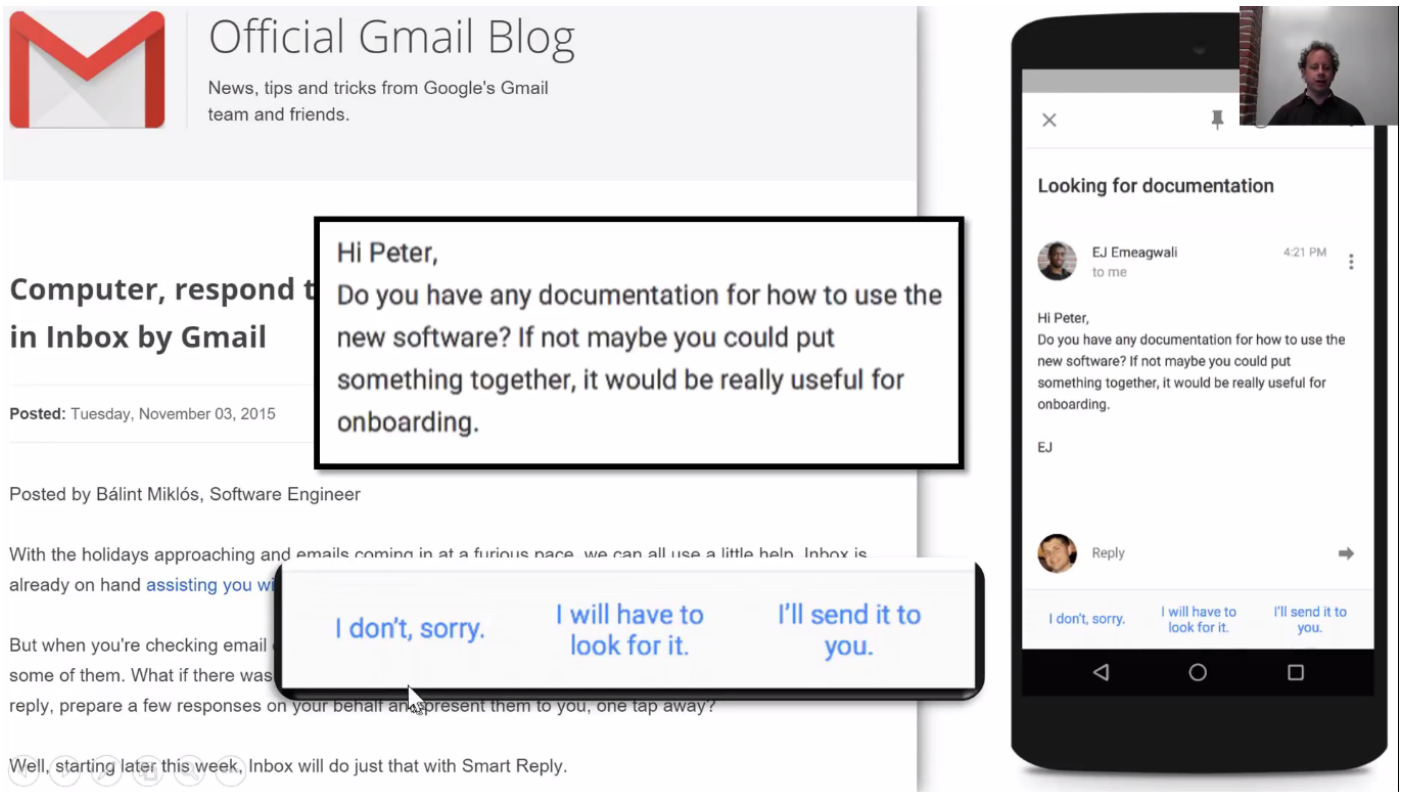
E5-2699: $4115

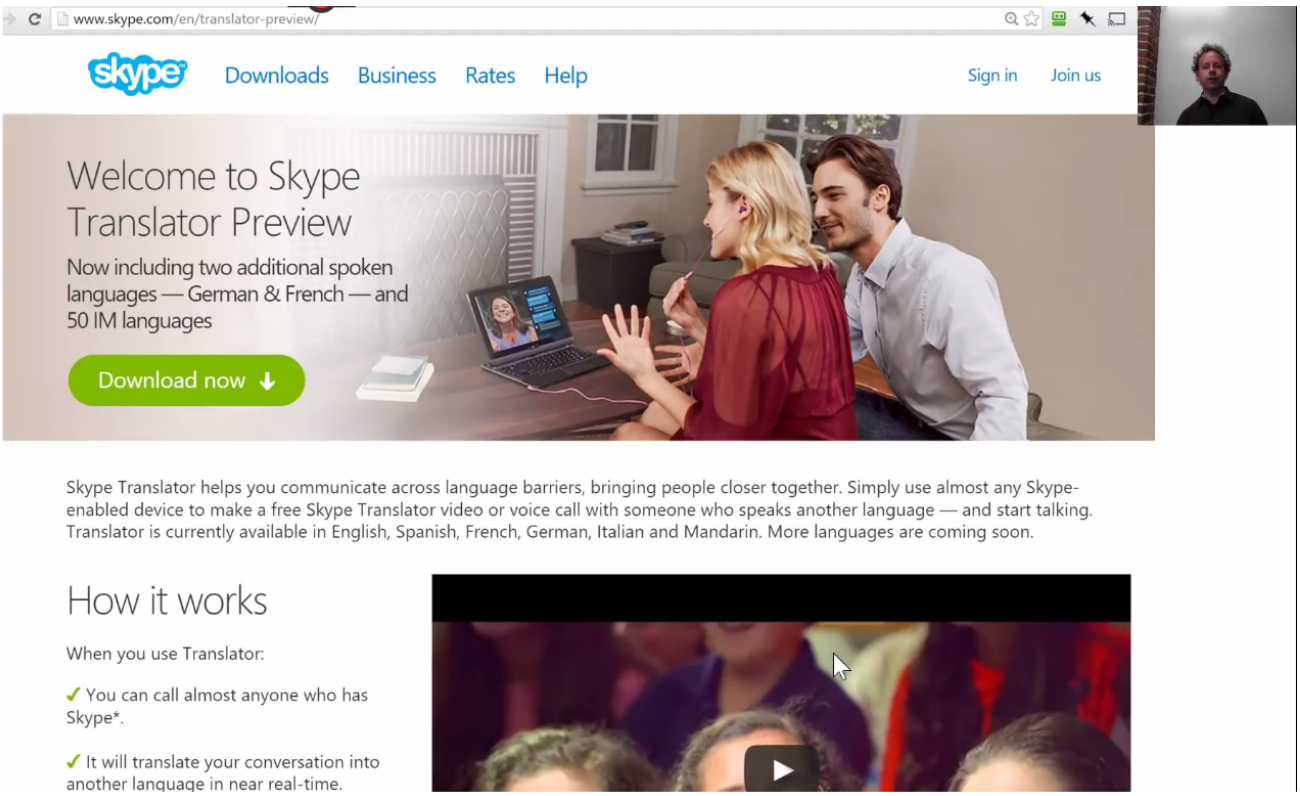
red, green: **GPU**
Blue: CPU

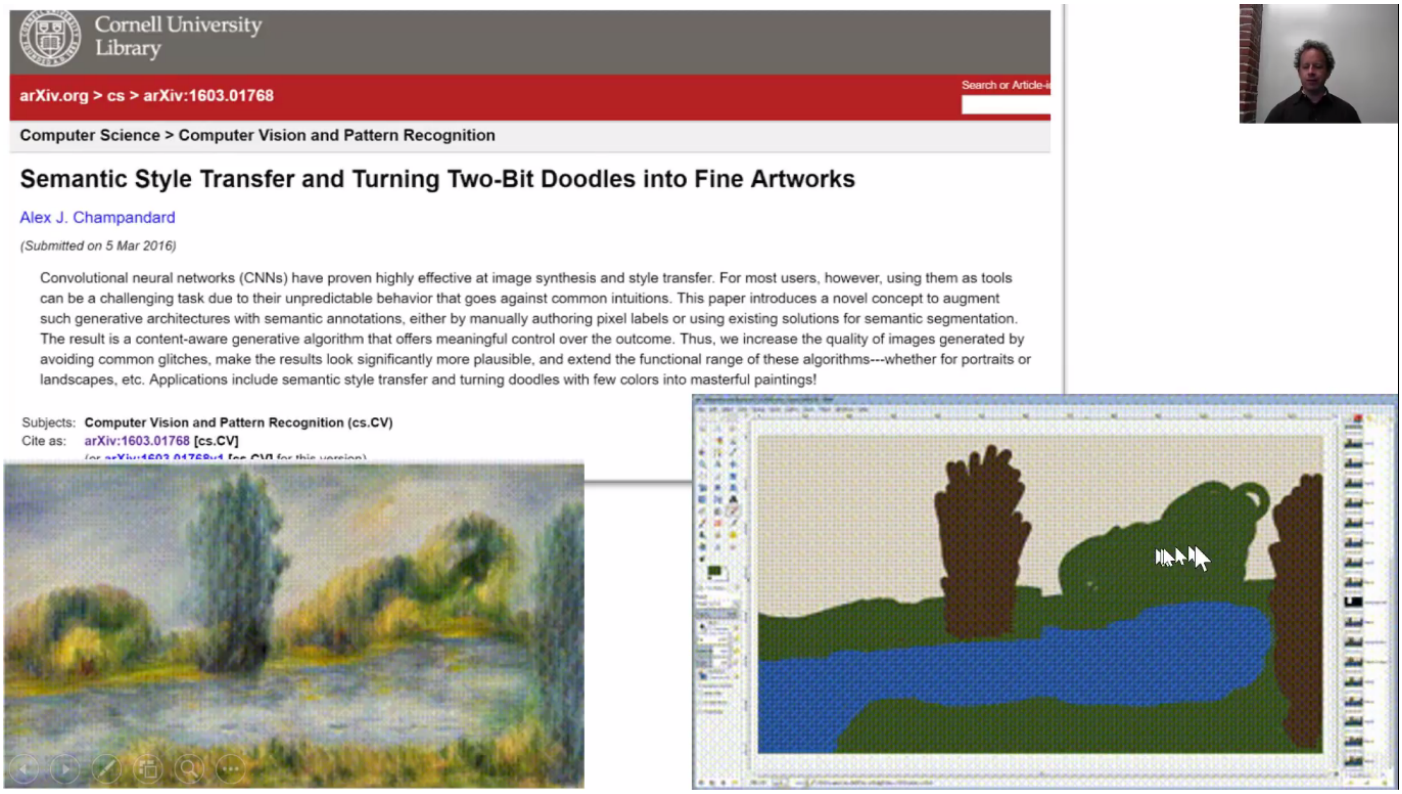
On log scale, GPU ten times fast then CPU

Examples where DL is being applied




Official Gmail Blog
News, tips and tricks from Google's Gmail team and friends.
Computer, respond t
in Inbox by Gmail
Posted: Tuesday, November 03, 2015
Posted by Bálint Miklós, Software Engineer
With the holidays approaching and emails coming in at a furious pace, we can all use a little help. Inbox is already on hand assisting you w
But when you're checking email
some of them. What if there was
reply, prepare a few responses on your behalf and present them to you, one tap away?
Well, starting later this week, Inbox will do just that with Smart Reply.
Hi Peter,
Do you have any documentation for how to use the new software? If not maybe you could put something together, it would be really useful for onboarding.
I don't, sorry.
I will have to look for it.
I'll send it to you.
Looking for documentation
EJ Emeagwali
to me
4:21 PM
Hi Peter,
Do you have any documentation for how to use the new software? If not maybe you could put something together, it would be really useful for onboarding.
EJ
Reply
I don't, sorry.
I will have to look for it.
I'll send it to you.


www.skype.com/en/translator-preview/
skype
Downloads
Business
Rates
Help
Sign in
Join us
Welcome to Skype Translator Preview
Now including two additional spoken languages — German & French — and 50 IM languages
Download now
Skype Translator helps you communicate across language barriers, bringing people closer together. Simply use almost any Skype-enabled device to make a free Skype Translator video or voice call with someone who speaks another language — and start talking. Translator is currently available in English, Spanish, French, German, Italian and Mandarin. More languages are coming soon.
How it works
When you use Translator:
✓ You can call almost anyone who has Skype*.
✓ It will translate your conversation into another language in near real-time.

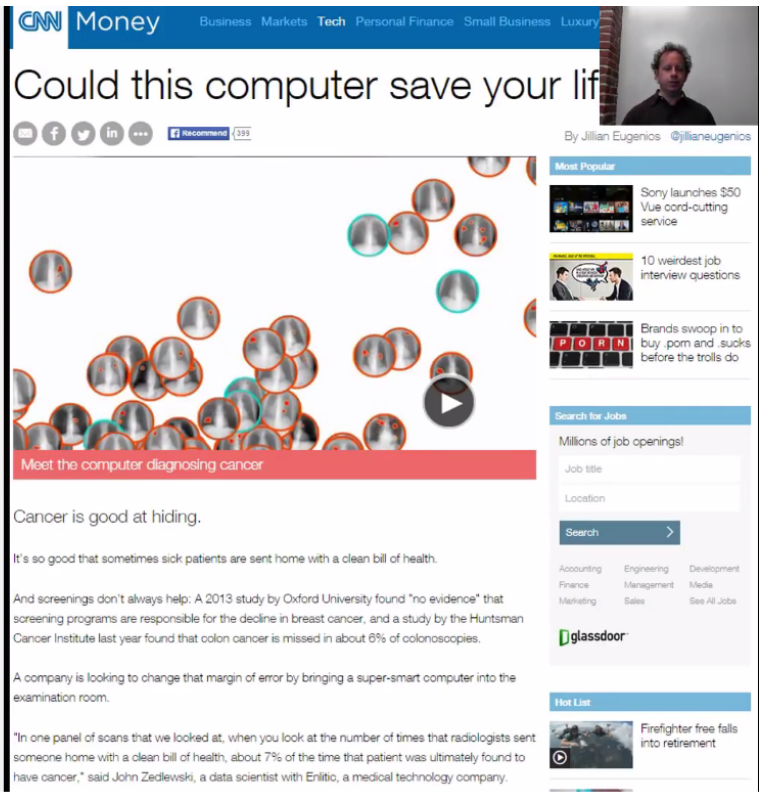


Draw this art by yourself and algo on left will convert your pic into an art

Neural Doodle Program

	False Positive Rate	False Negative Rate
Panel of 4 Human Radiologists	66.3%	7.0%
Enlitic Algorithm	47.5%	0.0%

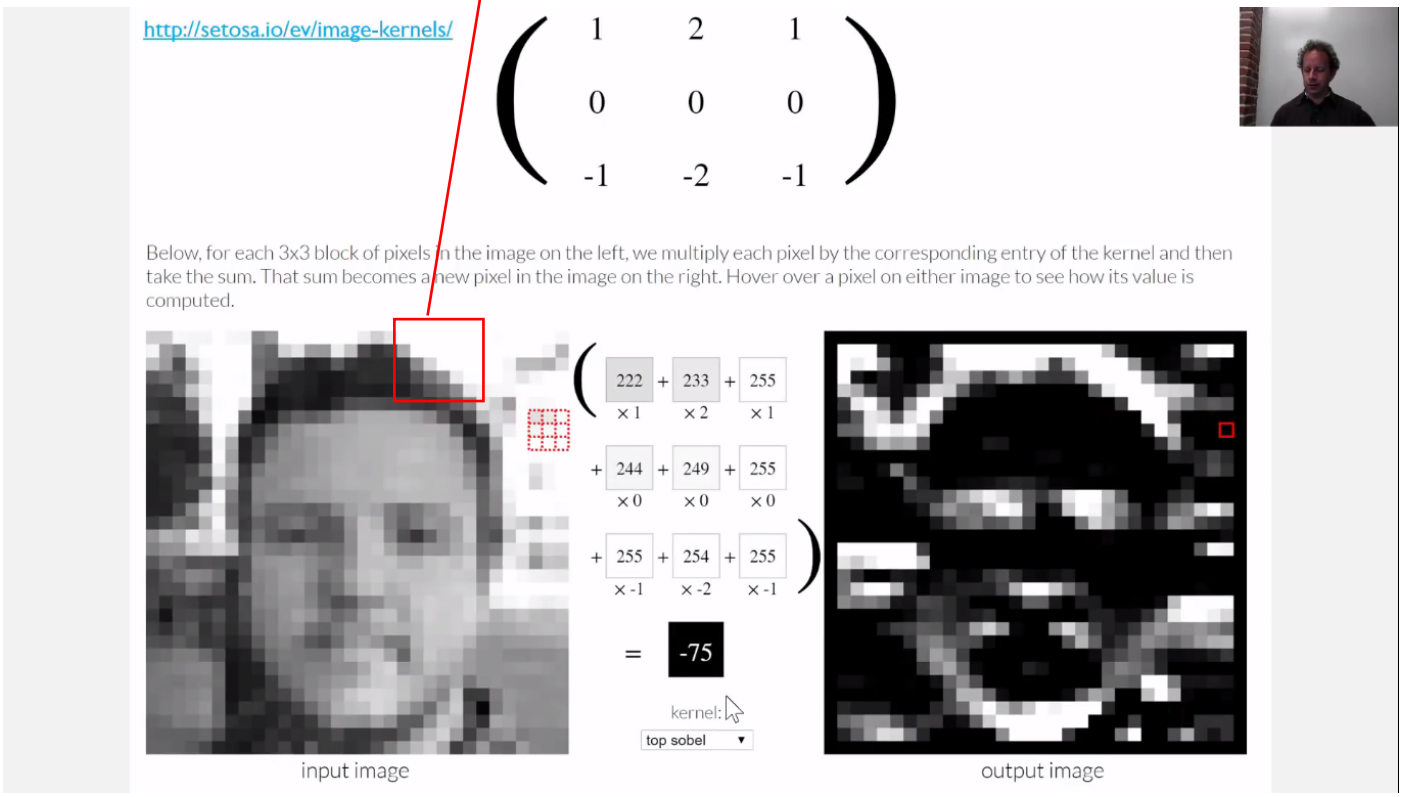
Jeremy & The friends (:P) company
Deep Learning yayyyyy!



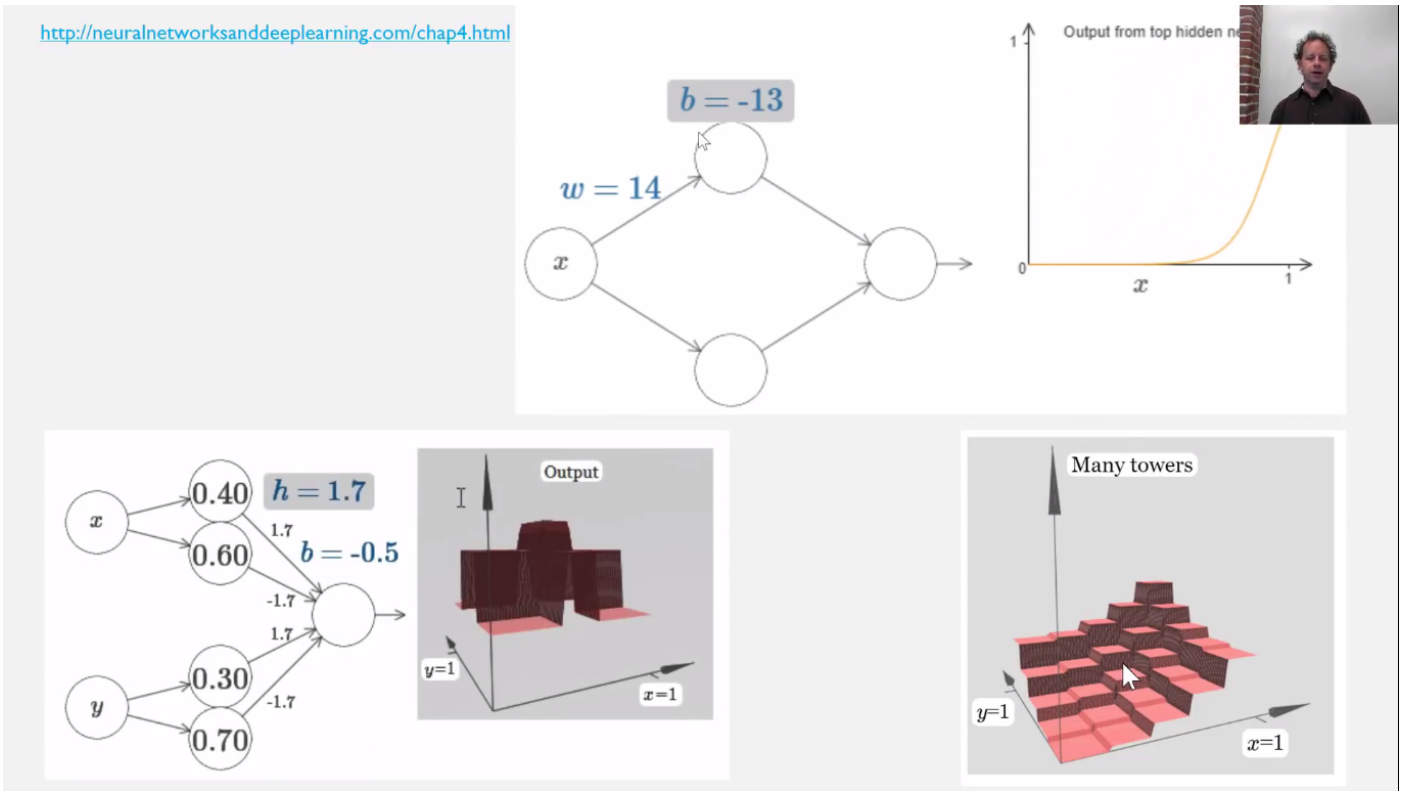
Deep Learning Hungry areas

Convolution with this filter is finding edges of face.
Very low ouput values of convolution correspond to black while large values to white colour

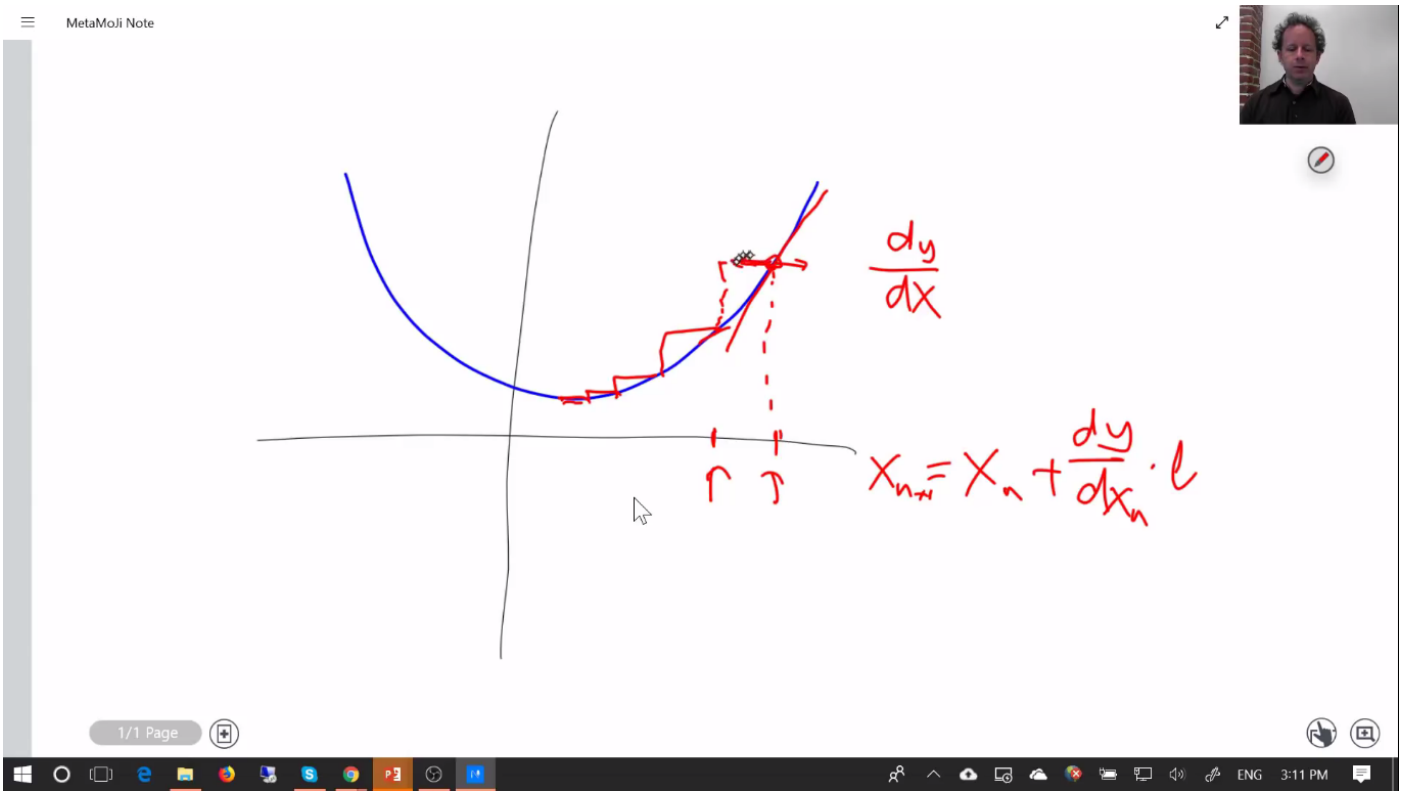
e.g., in this region, overall output will be high. so white color



Covolutional filter are normally of 3x3 size
Covolution is a simple linear operation will be used in neural networks

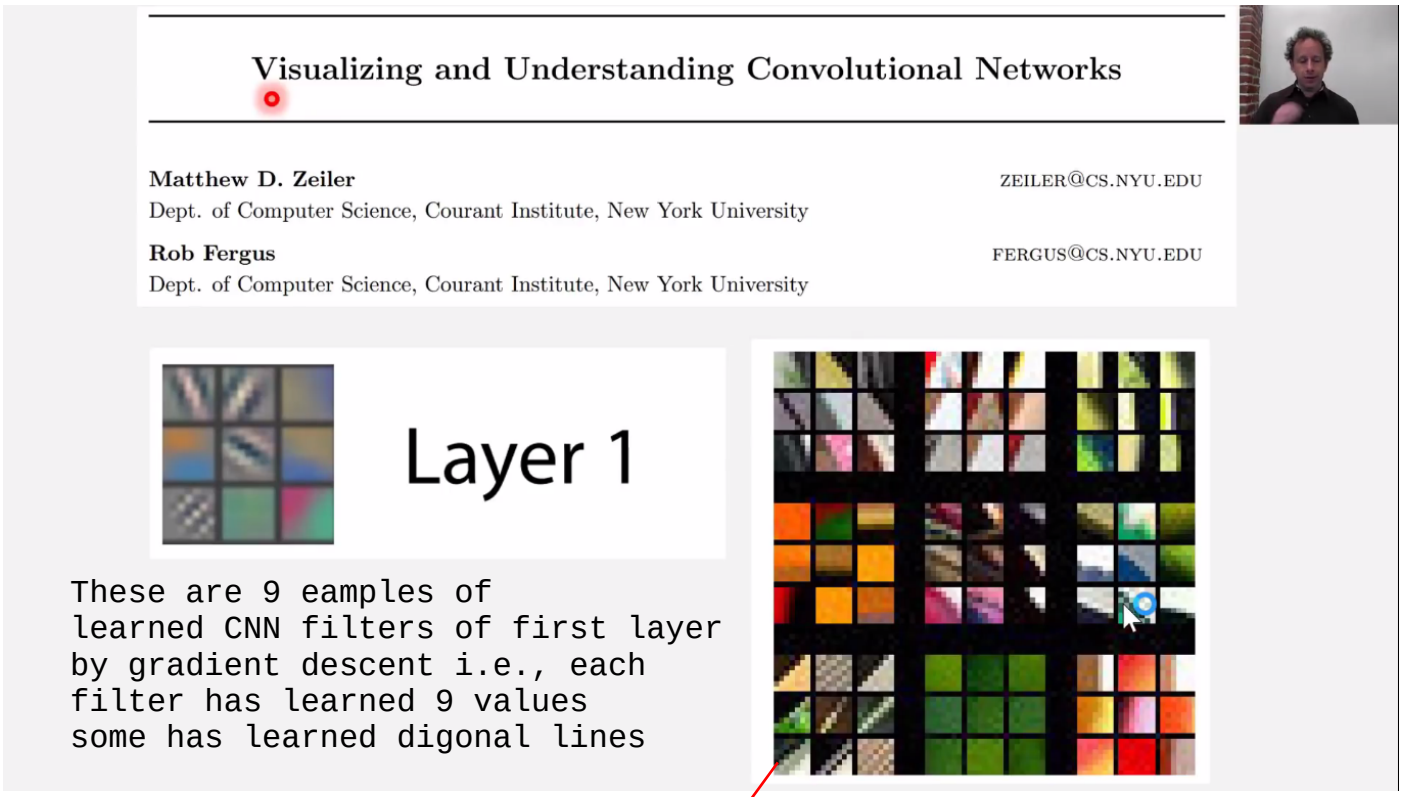


Non-linearity addition (in top fig also which is ouputing simgoid) with just change in values inside neurons
i.e., weights which are its params, complex towers are formed with neural networks



Finding params by Gradient Descent:
we want to find minimum of graph, lets take a random number, take its derivate which will point where graph is minimal
l = learning rate (jump size)

An attempt to draw what each layer learned.

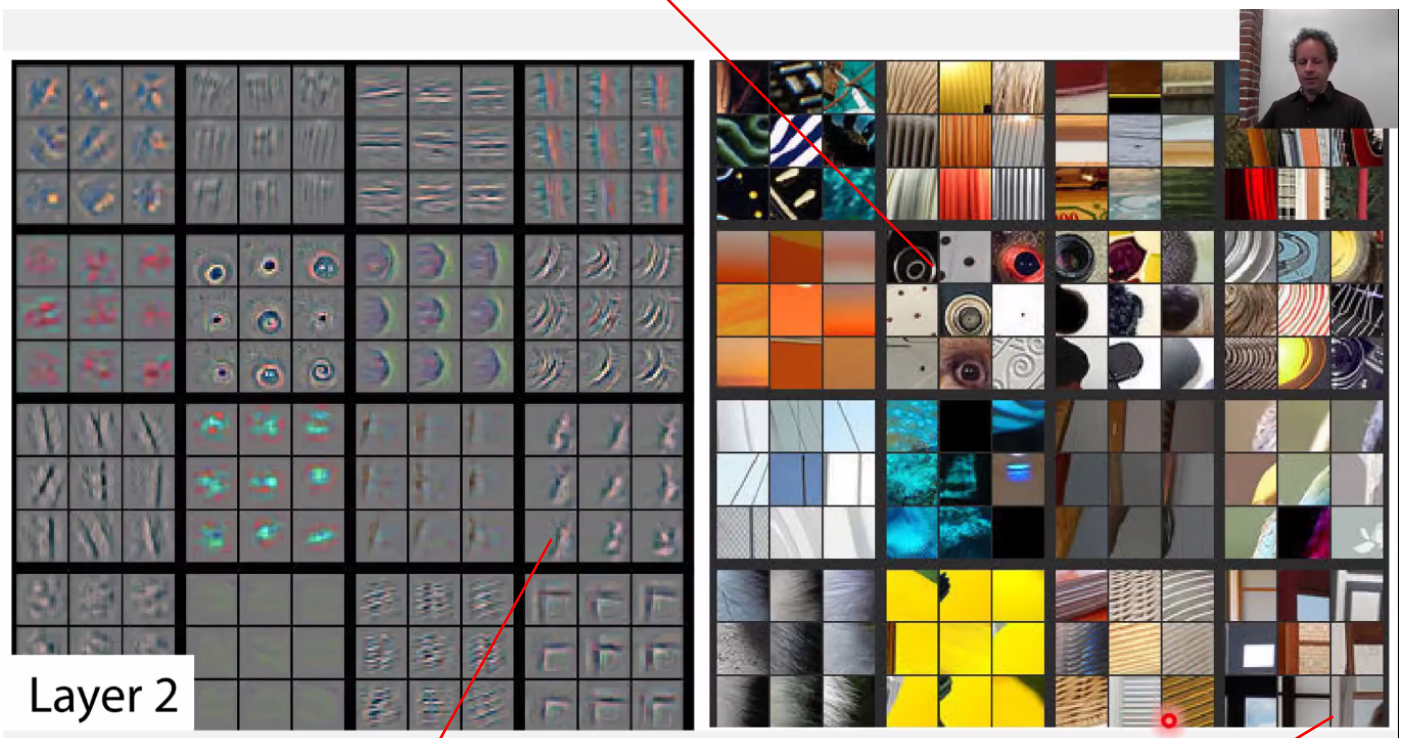


These are 9 eamples of learned CNN filters of first layer by gradient descent i.e., each filter has learned 9 values some has learned digonal lines

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

These are actual 9 photos which activate that filter.

their corrs filters are activated by circular shapes

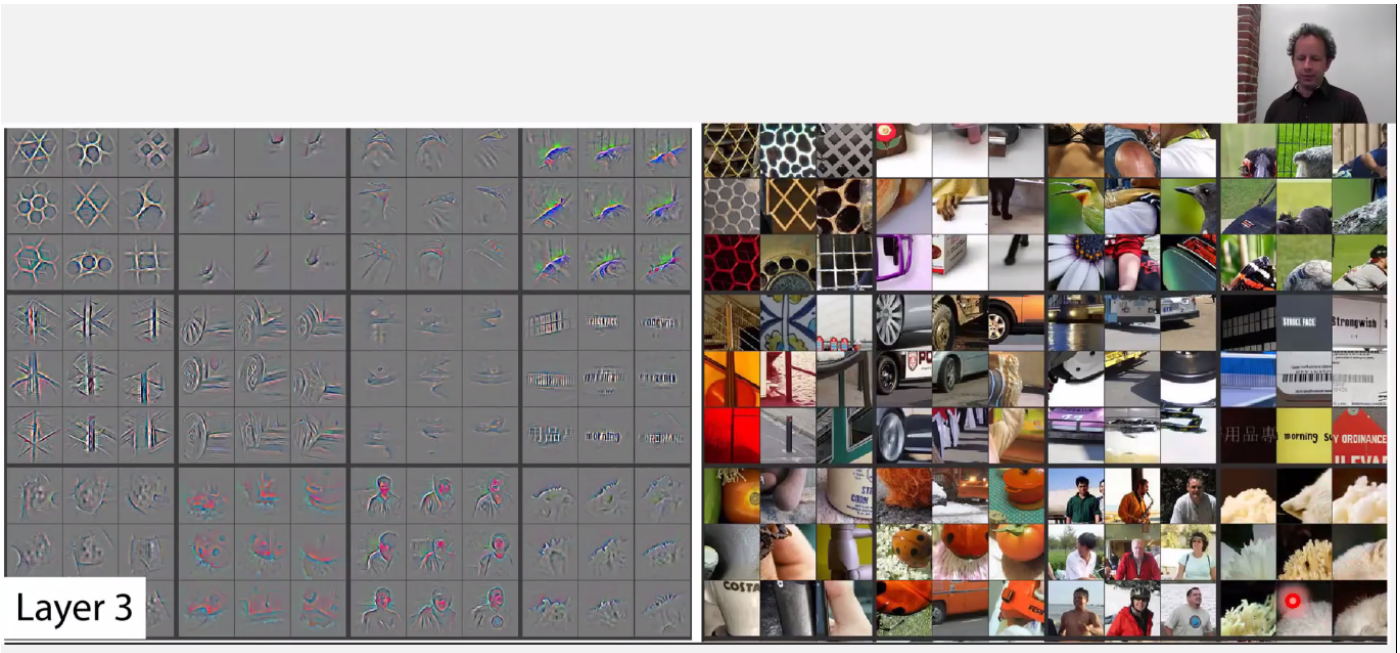


Layer 2 activations taking layer 1 acts as input, have learned some complex shapes

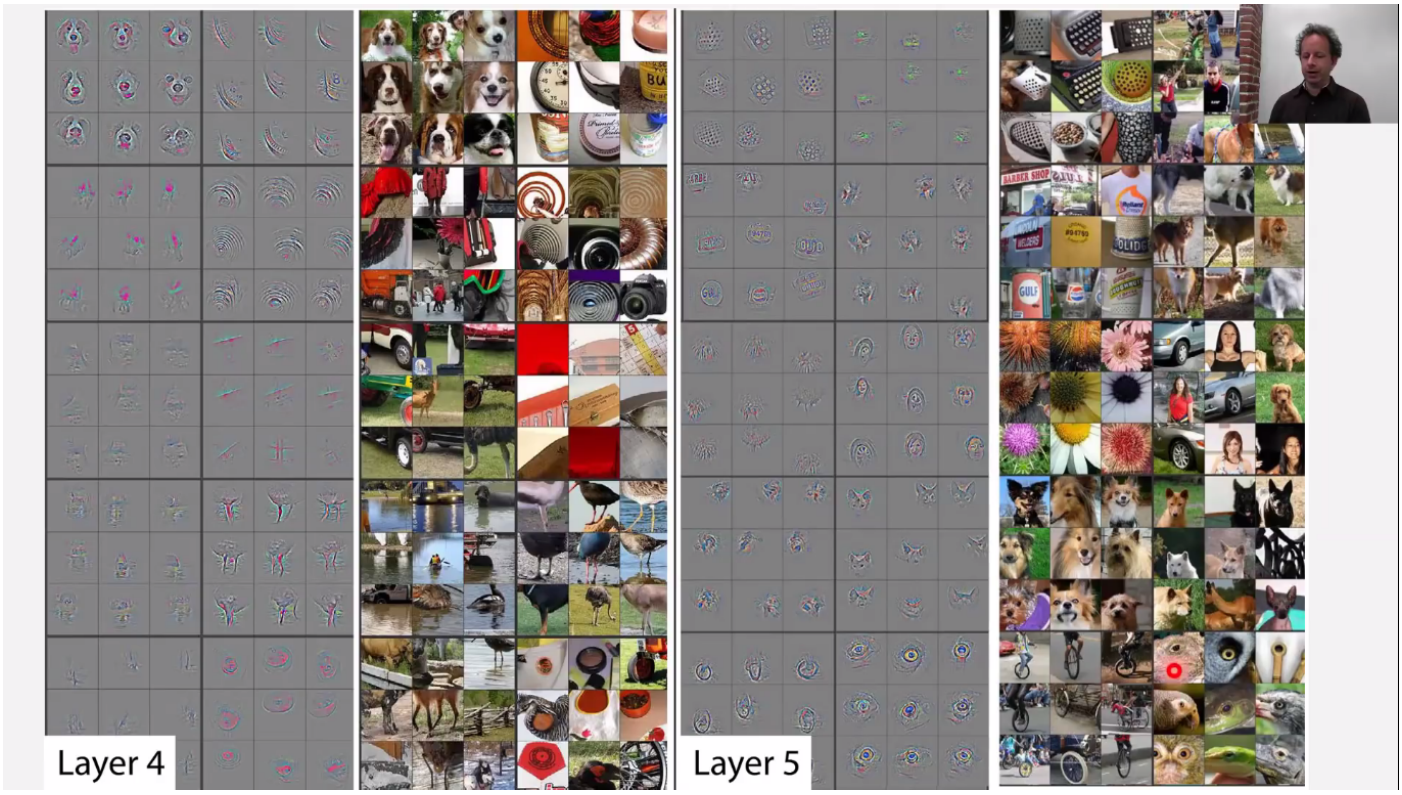
activated by cornes

filters corresponding to these have learned to be activated by some sun's pics.

layer 3 combining features of previous layers



one filter has learned to recogize the presence of human faces, one has learned to recognize presence of car tyres, text etc.



cycle
tyres

just 5 layers
learned to recognize
eyeballs of animals

All thanks to universal approximation theorem and use of multiple hidden layers

Choosing a learning rate

The *learning rate* determines how quickly or how slowly you want to update the *weights* (or *parameters*). Learning rate is one of the most difficult parameters to set, because it significantly affect model performance.

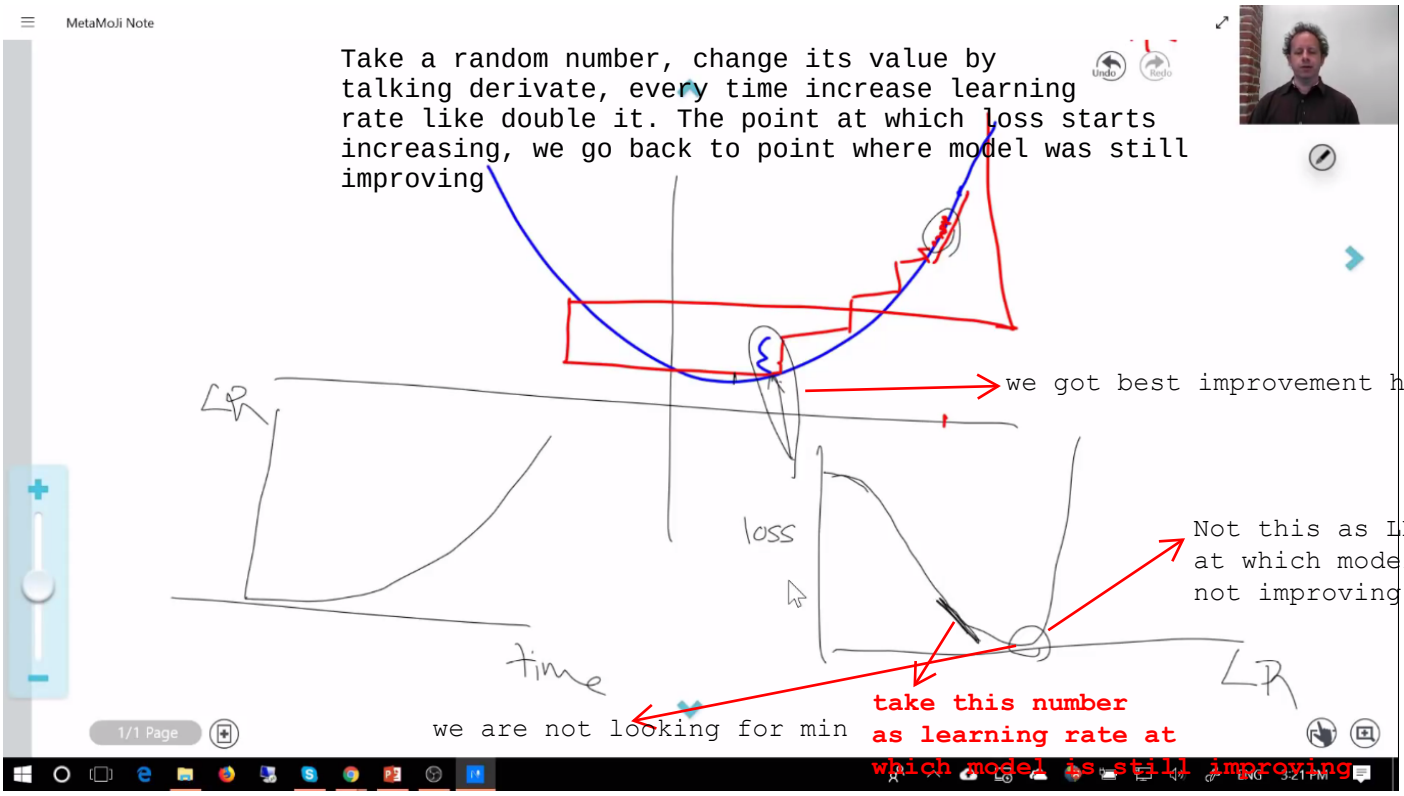
The method `learn.lr_find()` helps you find an optimal learning rate. It uses the technique developed in the 2015 paper Cyclical Learning Rates for Training Neural Networks, where we simply keep increasing the learning rate from a very small value, until the loss starts decreasing. We can plot the learning rate across batches to see what this looks like.

We first create a new learner, since we want to know how to set the learning rate for a new (untrained) model.

In [10]:
```
learn = ConvLearner.pretrained(arch, data, precompute=True)
```

In [12]:
```
lrf=learn.lr_find()
```

This Paper's Idea:



we are not looking for min point. That point could actually be where loss started to increase