



## 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 Algorithms are the ones which fit the following requirements



**Infinitely flexible function**

means a network which can solve any problem given right parameters

**All-purpose parameter fitting**

need to find params for a specific problem

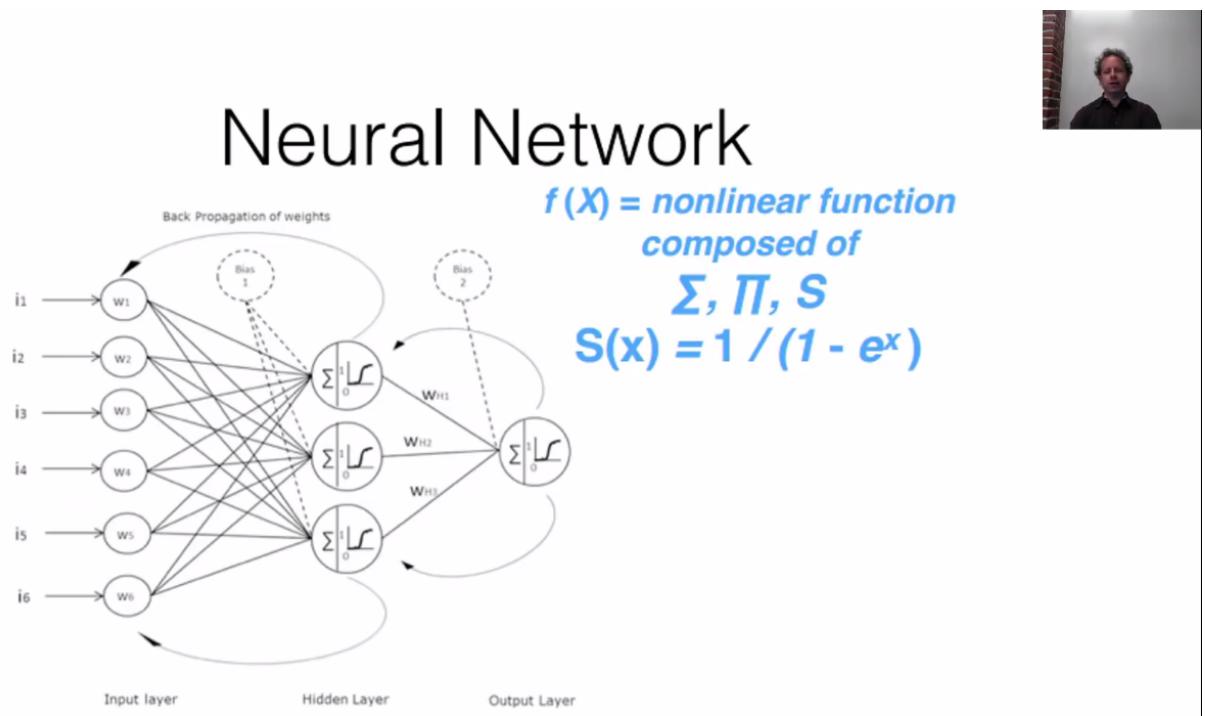
**Fast and scalable**

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=Ijqkc70LenI>



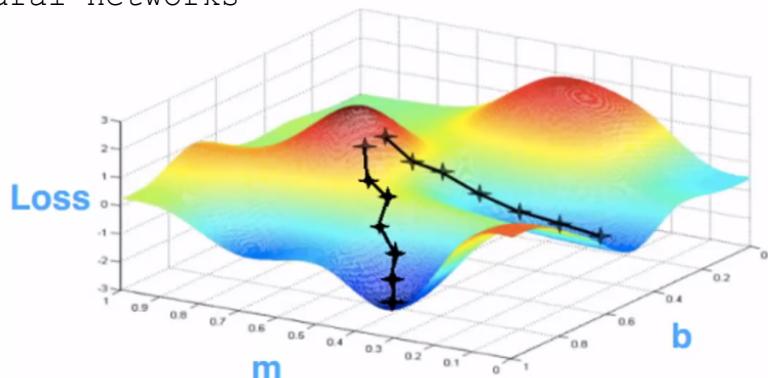
Even this neural network with single hidden layer does support universal approximation theorem but it requires exponential number of params, so not fast and scalable.

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



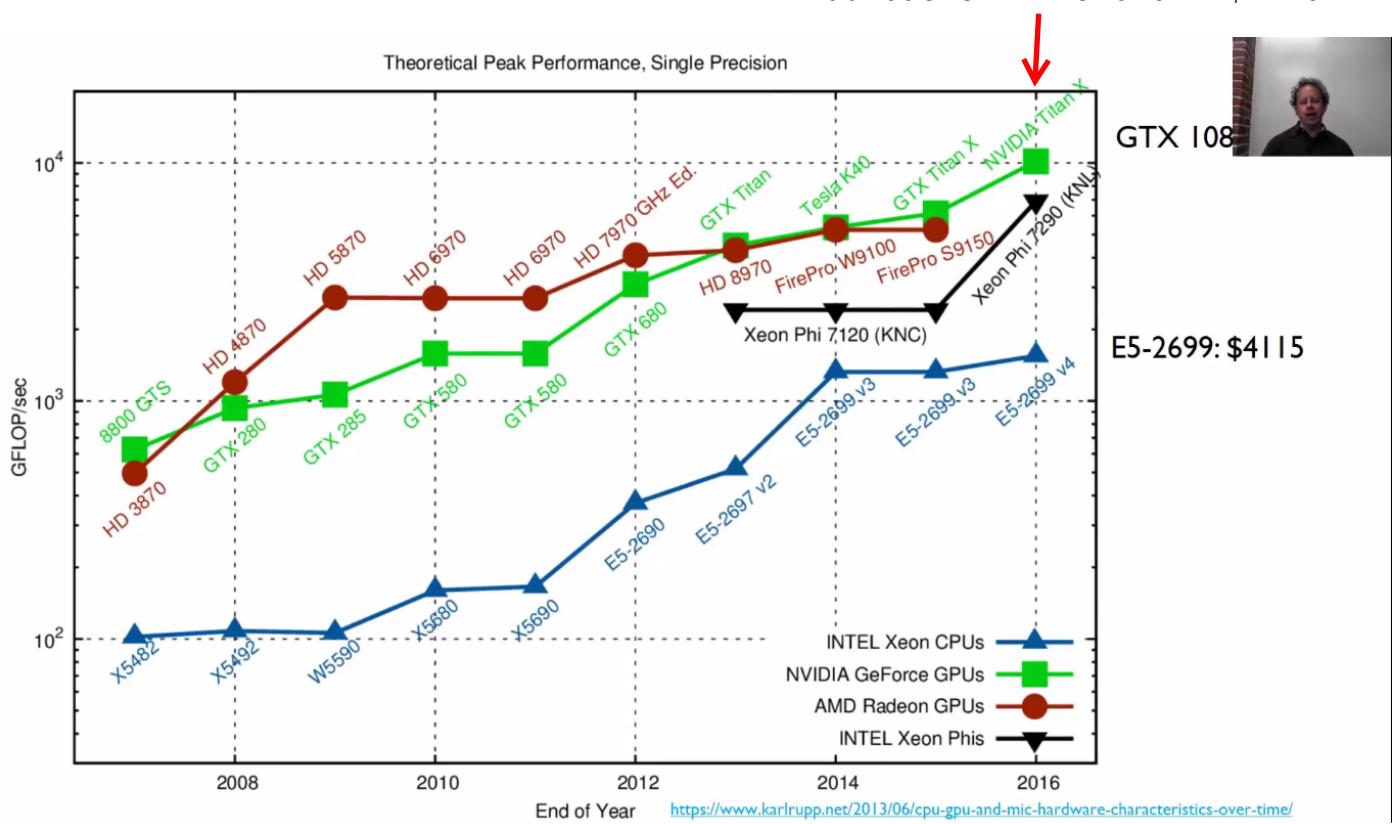
# Gradient Descent

GD is an excellent way for fitting  $f(x) = \text{nonlinear function of } x$  parameters to neural networks



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



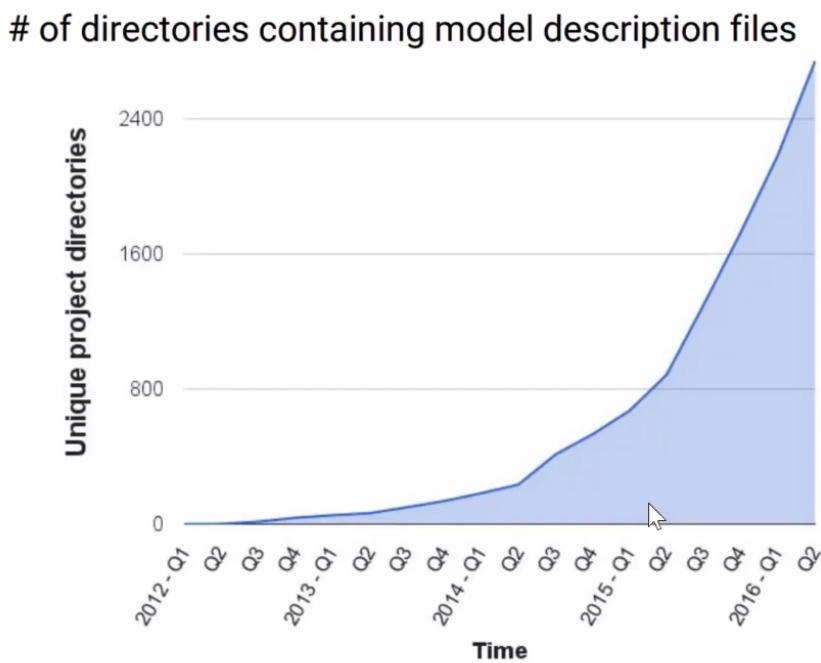
red, green: GPU

On log scale, GPU ten times fast than CPU

Blue: CPU

**Examples where DL is being applied**

## Growing Use of Deep Learning at Google



**Across many products/areas:**

Android  
Apps  
drug discovery  
Gmail  
Image understanding  
Maps  
Natural language understanding  
Photos  
Robotics research  
Speech  
Translation  
YouTube  
... many others ...





## Official Gmail Blog

News, tips and tricks from Google's Gmail team and friends.

### Computer, respond to me in Inbox by Gmail

Posted: Tuesday, November 03, 2015

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.

Posted by Béálint 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 with them](#).

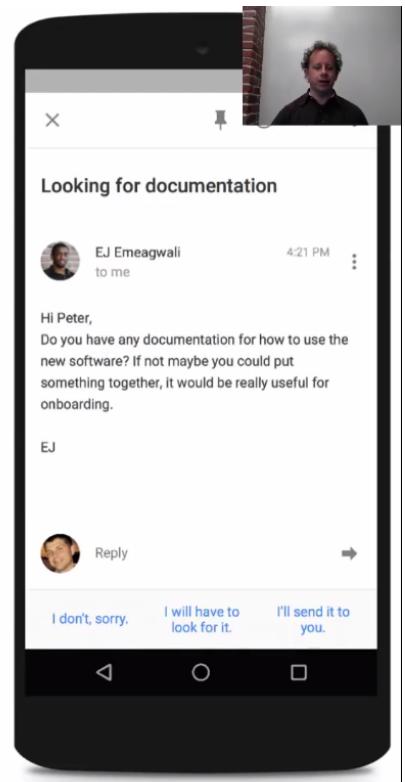
But when you're checking email on your phone, some of them. What if there was a way to quickly reply, prepare a few responses on your behalf and present them to you, one tap away?

Well, starting later this week, Inbox will do just that with Smart Reply.

I don't, sorry.

I will have to look for it.

I'll send it to you.



The screenshot shows the homepage of the Skype Translator Preview website. At the top, there's a navigation bar with links for 'Downloads', 'Business', 'Rates', and 'Help'. On the right side of the header, there are 'Sign in' and 'Join us' buttons, along with a small profile picture of a man. Below the header, a large banner features a woman in a red dress and a man in a white shirt having a video call on a laptop, with a smaller video window showing a woman's face. The banner text reads 'Welcome to Skype Translator Preview' and 'Now including two additional spoken languages — German & French — and 50 IM languages'. A green button labeled 'Download now' with a downward arrow is visible. The background of the page shows a brick wall.

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.



Cornell University Library

arXiv.org > cs > arXiv:1603.01768

Computer Science > Computer Vision and Pattern Recognition

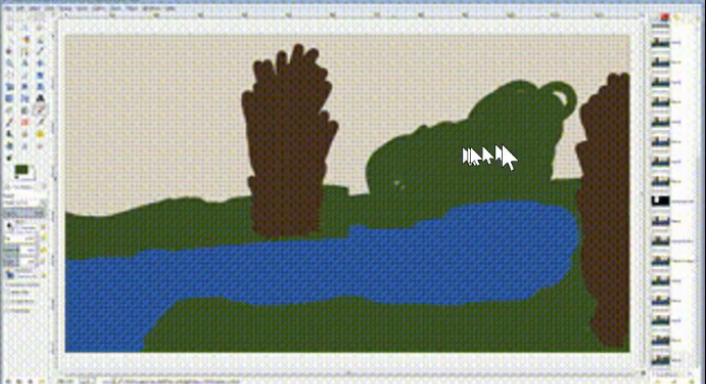
**Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artworks**

Alex J. Champandard

(Submitted on 5 Mar 2016)

Convolutional neural networks (CNNs) have proven highly effective at image synthesis and style transfer. For most users, however, using them as tools can be a challenging task due to their unpredictable behavior that goes against common intuitions. This paper introduces a novel concept to augment such generative architectures with semantic annotations, either by manually authoring pixel labels or using existing solutions for semantic segmentation. The result is a content-aware generative algorithm that offers meaningful control over the outcome. Thus, we increase the quality of images generated by avoiding common glitches, make the results look significantly more plausible, and extend the functional range of these algorithms---whether for portraits or landscapes, etc. Applications include semantic style transfer and turning doodles with few colors into masterful paintings!

Subjects: Computer Vision and Pattern Recognition (cs.CV)  
Cite as: arXiv:1603.01768 [cs.CV]  
(or arXiv:1603.01768v4 for this version)

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

### Neural Doodle Program

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

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

**CNN Money** Business Markets Tech Personal Finance Small Business Luxury

By Jillian Eugenics @jillianeugenics

## Could this computer save your life?

Meet the computer diagnosing cancer

Cancer is good at hiding.

It's so good that sometimes sick patients are sent home with a clean bill of health.

And screenings don't always help: A 2013 study by Oxford University found "no evidence" that screening programs are responsible for the decline in breast cancer, and a study by the Huntsman Cancer Institute last year found that colon cancer is missed in about 6% of colonoscopies.

A company is looking to change that margin of error by bringing a super-smart computer into the examination room.

"In one panel of scans that we looked at, when you look at the number of times that radiologists sent someone home with a clean bill of health, about 7% of the time that patient was ultimately found to have cancer," said John Zedlewski, a data scientist with Enlitic, a medical technology company.

**Most Popular**

- Sony launches \$50 Vue cord-cutting service
- 10 weirdest job interview questions
- Brands swoop in to buy, porn and, sucks before the trolls do

**Search for Jobs**

Millions of job openings!

Job title:   
Location:   
Search

Accounting   Engineering   Development  
Finance   Management   Media  
Marketing   Sales   See All Jobs

**Hot List**

- Firefighter free falls into retirement

## Deep Learning Hungry areas

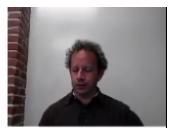


Convolution with this filter is finding edges of face.  
 Very low output values of convolution correspond to black while large values to white colour

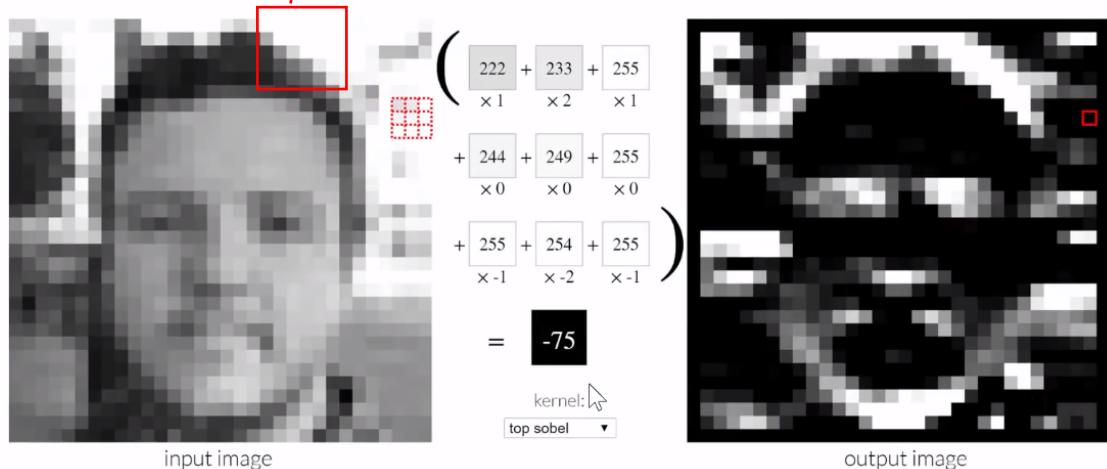
e.g., in this region, overall output will be high. so white color

<http://setosa.io/ev/image-kernels/>

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

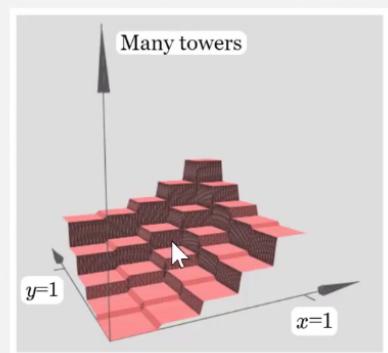
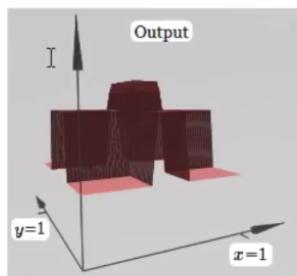
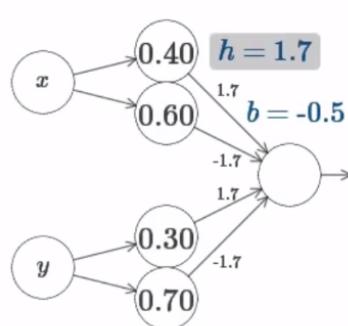
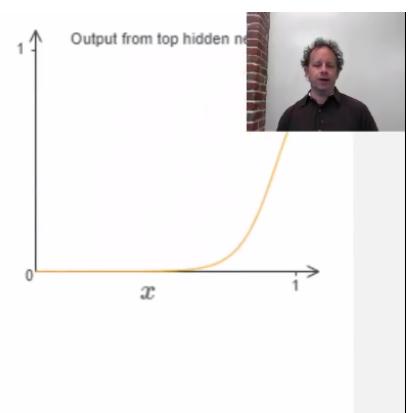
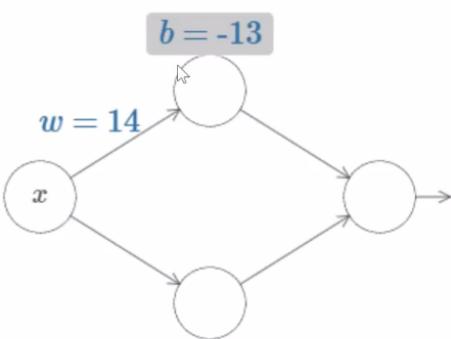


Below, for each 3x3 block of pixels in the image on the left, we multiply each pixel by the corresponding entry of the kernel and then take the sum. That sum becomes a new pixel in the image on the right. Hover over a pixel on either image to see how its value is computed.

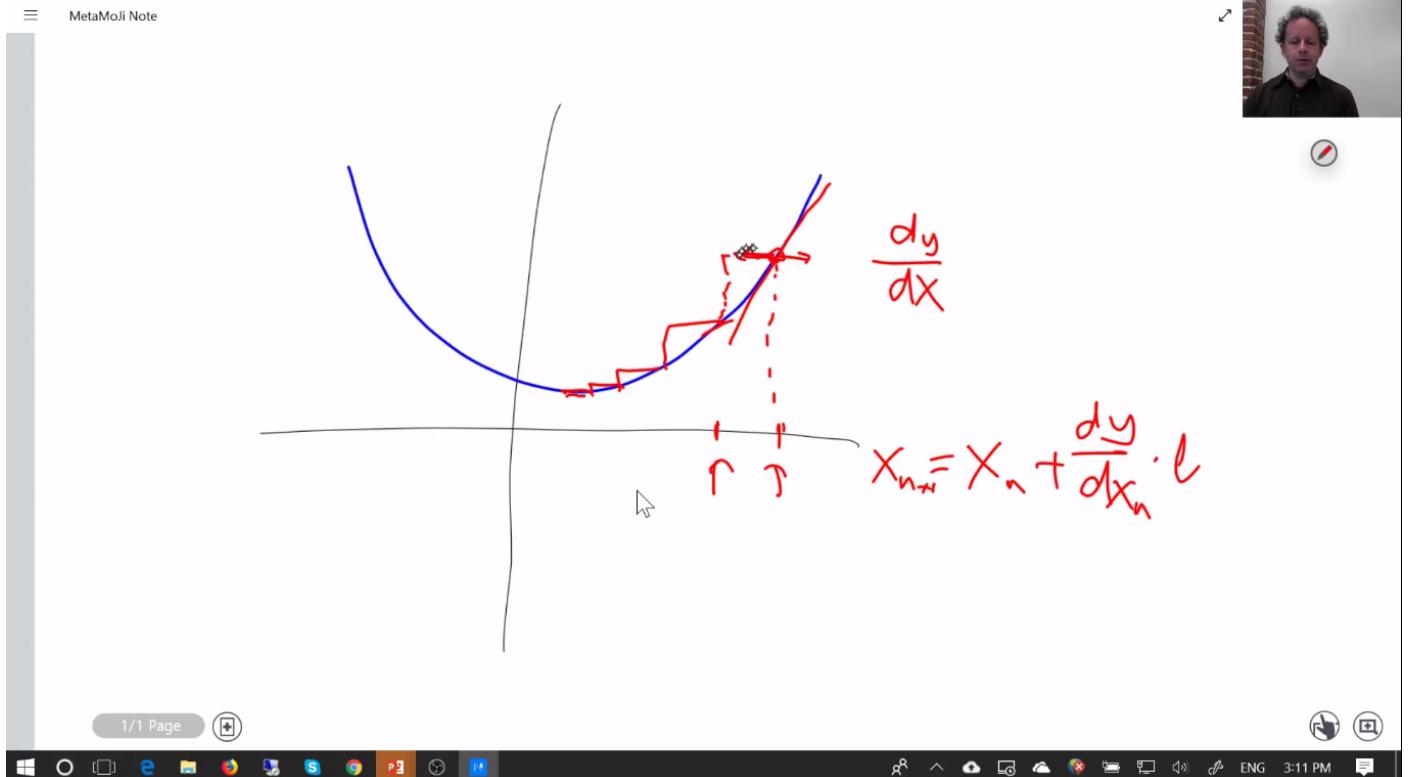


Covolutional filter are normally of 3x3 size  
 Covolution is a simple linear operation will be used in neural networks

<http://neuralnetworksanddeeplearning.com/chap4.html>



Non-linearity addition (in top fig also which is outputting sigmoid)  
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.

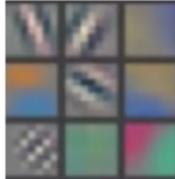
---

**Visualizing and Understanding Convolutional Networks**

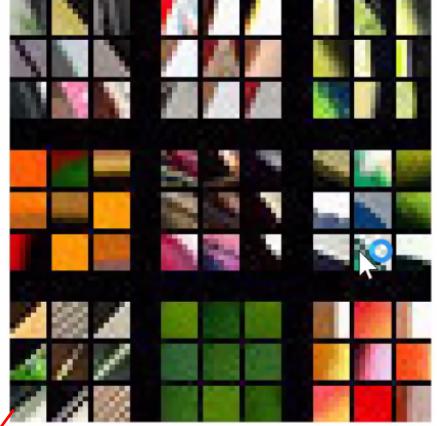


Matthew D. Zeiler  
Dept. of Computer Science, Courant Institute, New York University  
ZEILER@CS.NYU.EDU

Rob Fergus  
Dept. of Computer Science, Courant Institute, New York University  
FERGUS@CS.NYU.EDU



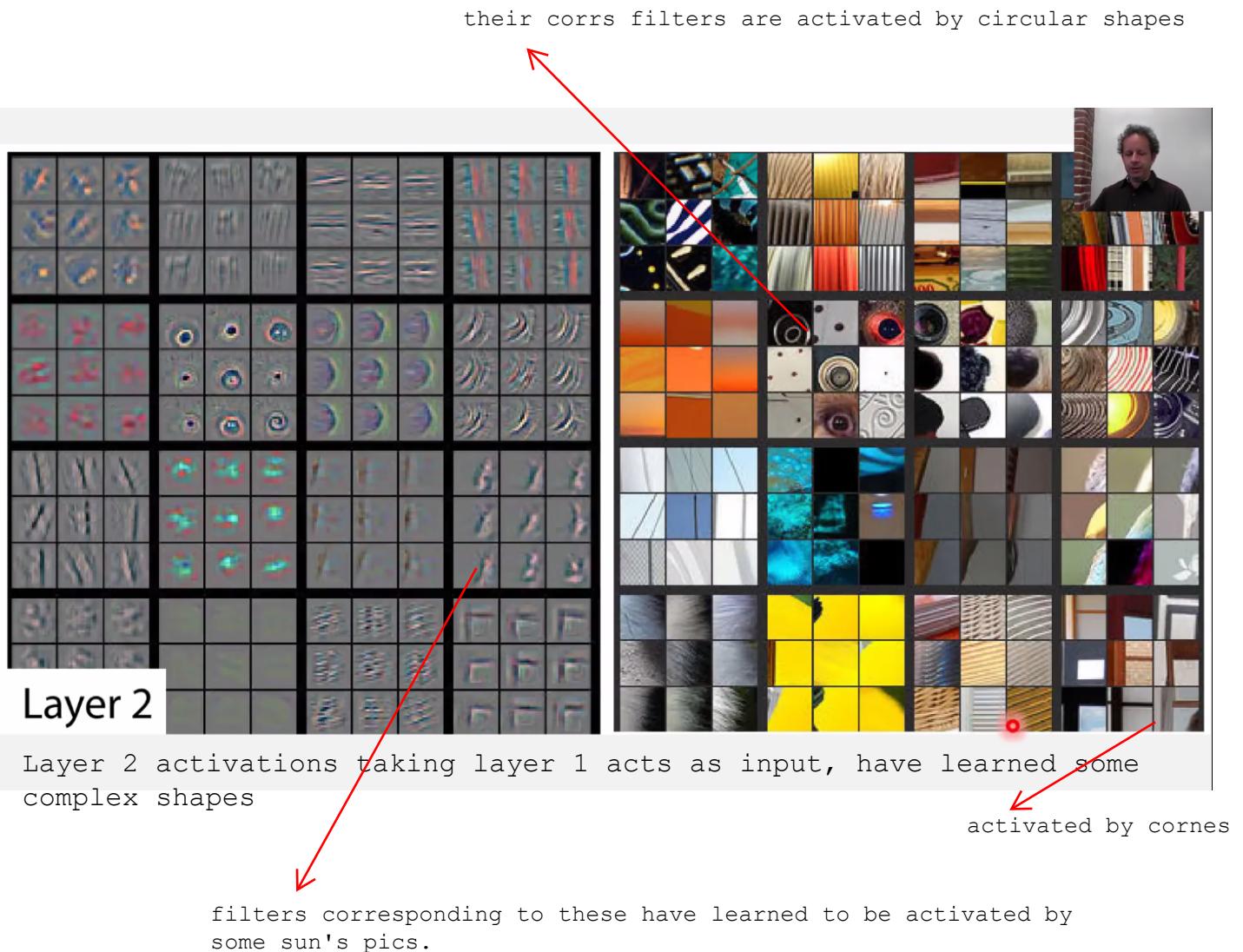
# Layer 1



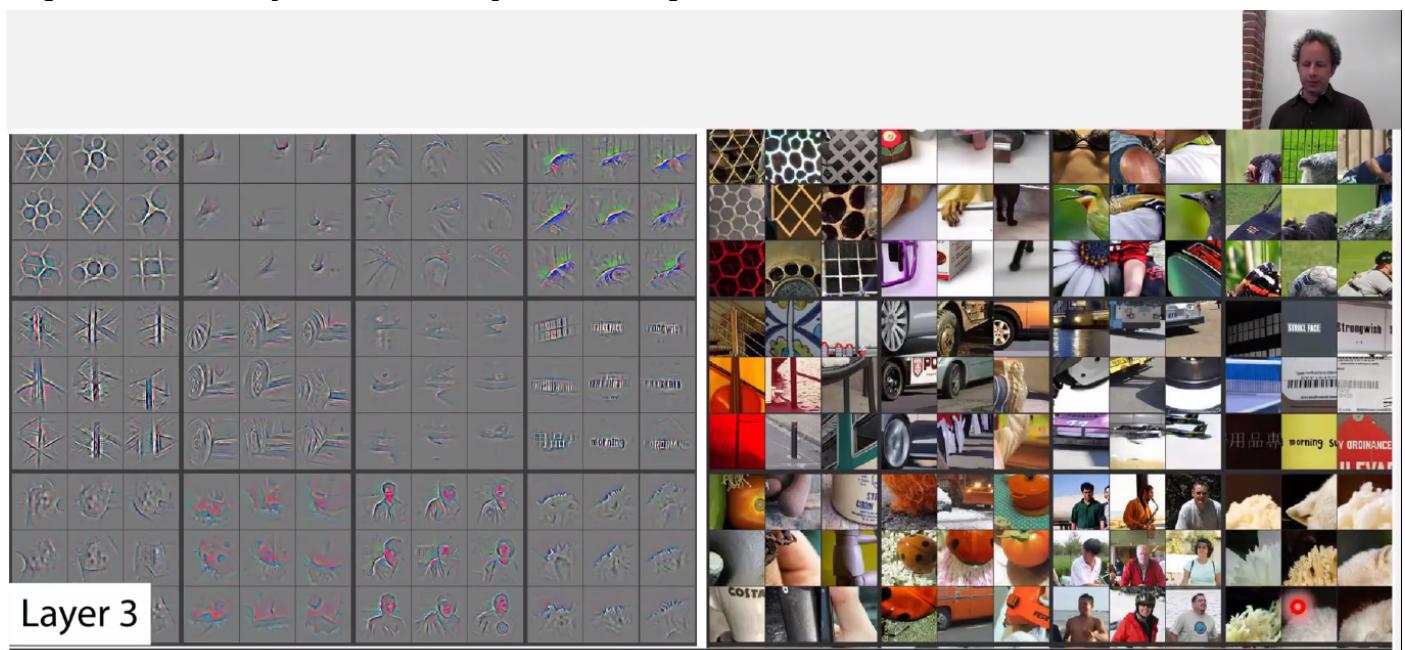
These are 9 examples of learned CNN filters of first layer by gradient descent i.e., each filter has learned 9 values some has learned diagonal lines

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

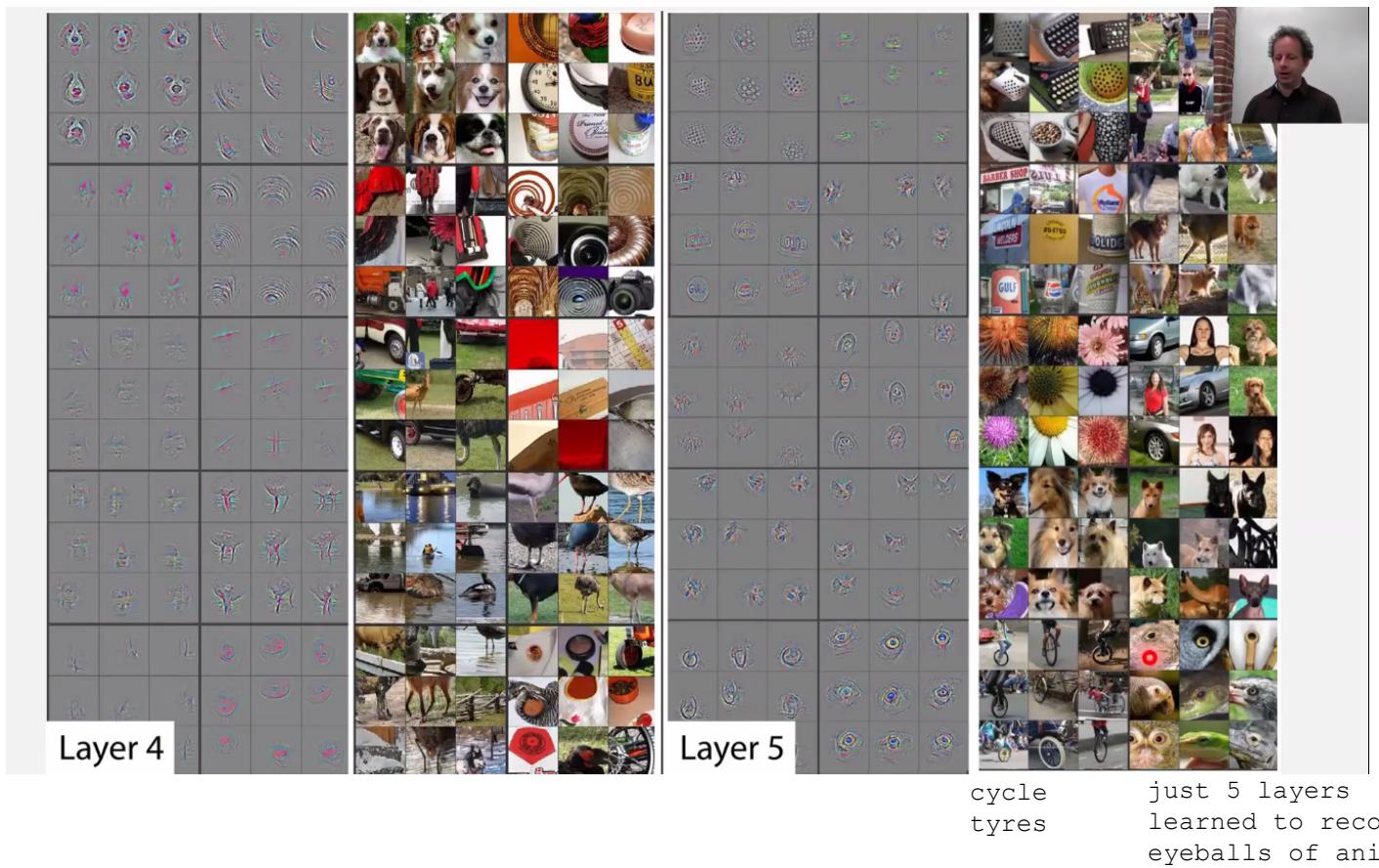
These are actual 9 photos which activate that filter.



layer 3 combining features of previous layers



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



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.

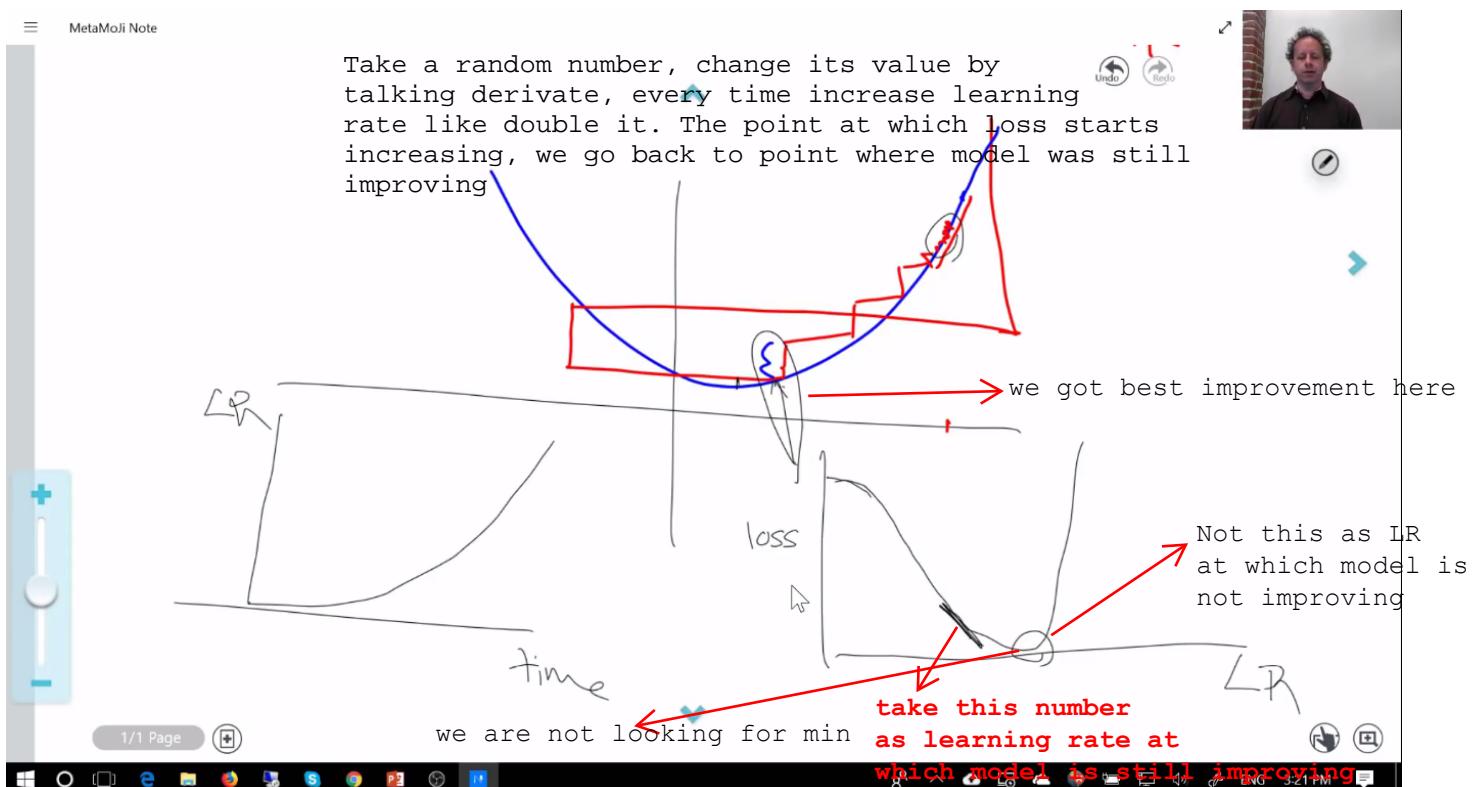
The method `learn.lr_find()` helps you find an optimal learning rate. It uses the technique developed in the 2015 paper [Cyclical Learning Rates for Training Neural Networks](#), where we simply keep increasing the learning rate from a very small value, until the loss starts decreasing. We can plot the 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]: lr_f=learn.lr_find()
```

This Paper's Idea:



point. That point could actually be where loss started to increase