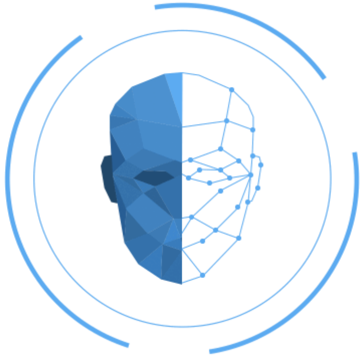
2018 Fall 创新创业实践课–编程之美

Introduction to Machine Learning

王浩宇
北京邮电大学 计算机学院
haoyuwang@bupt.edu.cn
sei.pku.edu.cn/~wanghy11

Outline

- What is Machine Learning and Why U Should Care?
- Easiest Introduction to Machine Learning
- Deep Learning and CNN
- Face Recognition
- Language Translation and RNN



机器学习　自然语言处理　深度学习（Deep Learning）

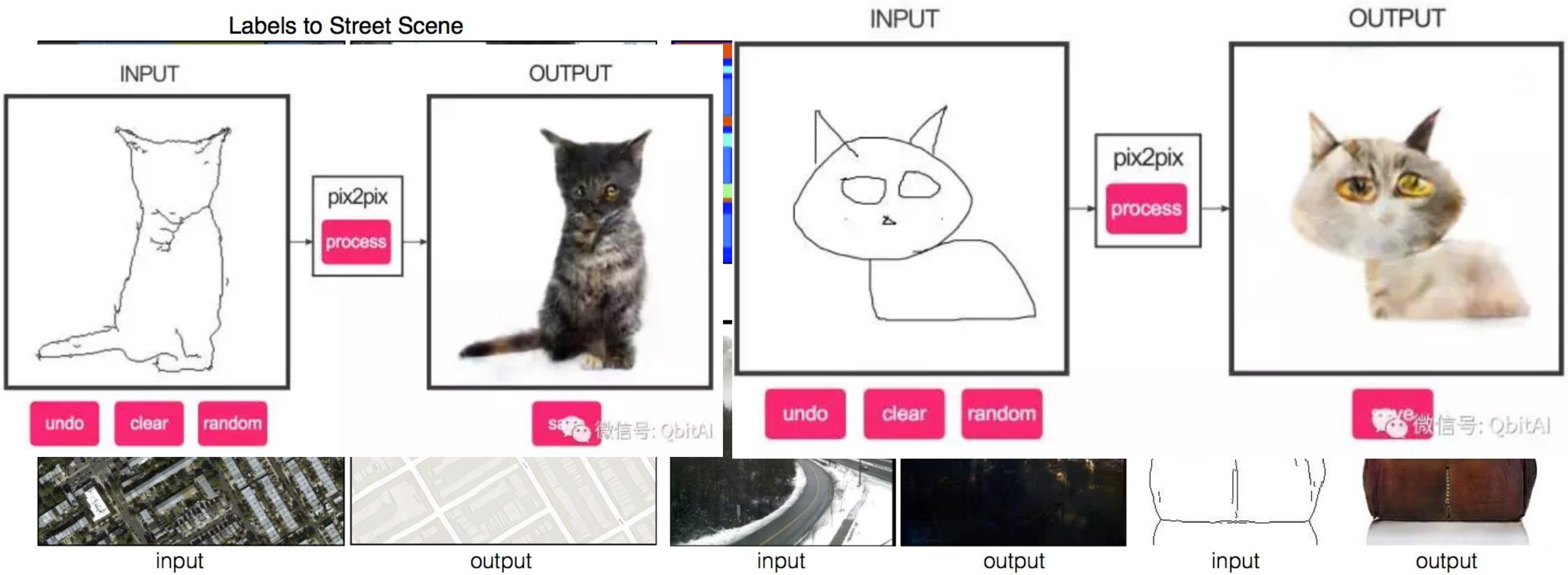
深度学习应用在哪些领域让你觉得「我去，这也能行！」？

深度学习被用在很多奇特的点上面，比如有的结合NLP去创作诗词文章；有的用来识别图片的物体、

https://www.zhihu.com/question/47563637

Crazy Machine Learning

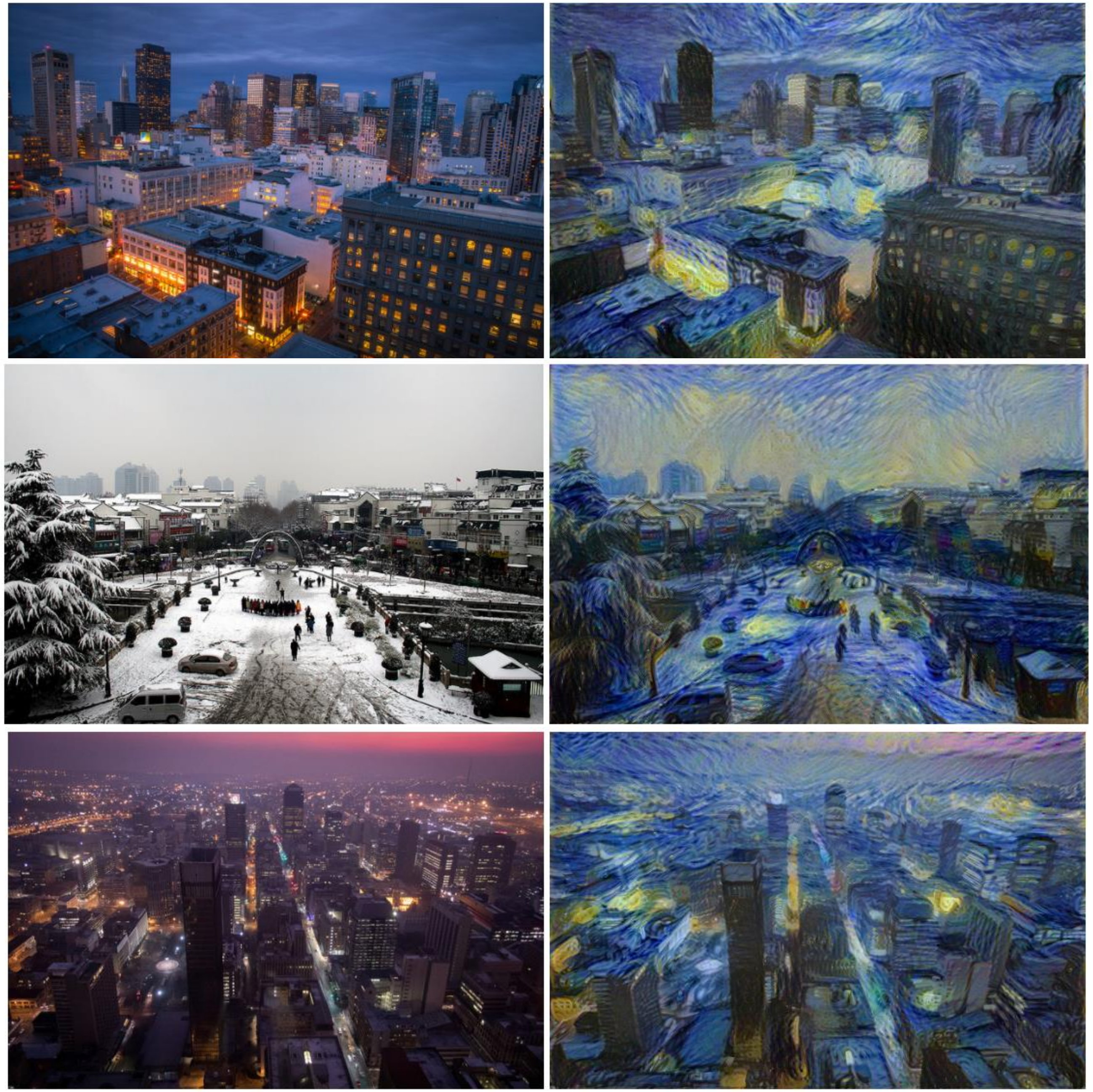
- Image-to-image translation with conditional adversarial nets
- https://github.com/phillipi/pix2pix



Crazy Machine Learning

- Style transfer

https://github.com/fzliu/style-transfer



Crazy Machine Learning

■ Deep Photo Transfer

https://github.com/luanfujun/deep-photo-styletransfer

Crazy Machine Learning

- **PySC2 - StarCraft II Learning Environment**

https://github.com/deepmind/pysc2

Crazy Machine Learning

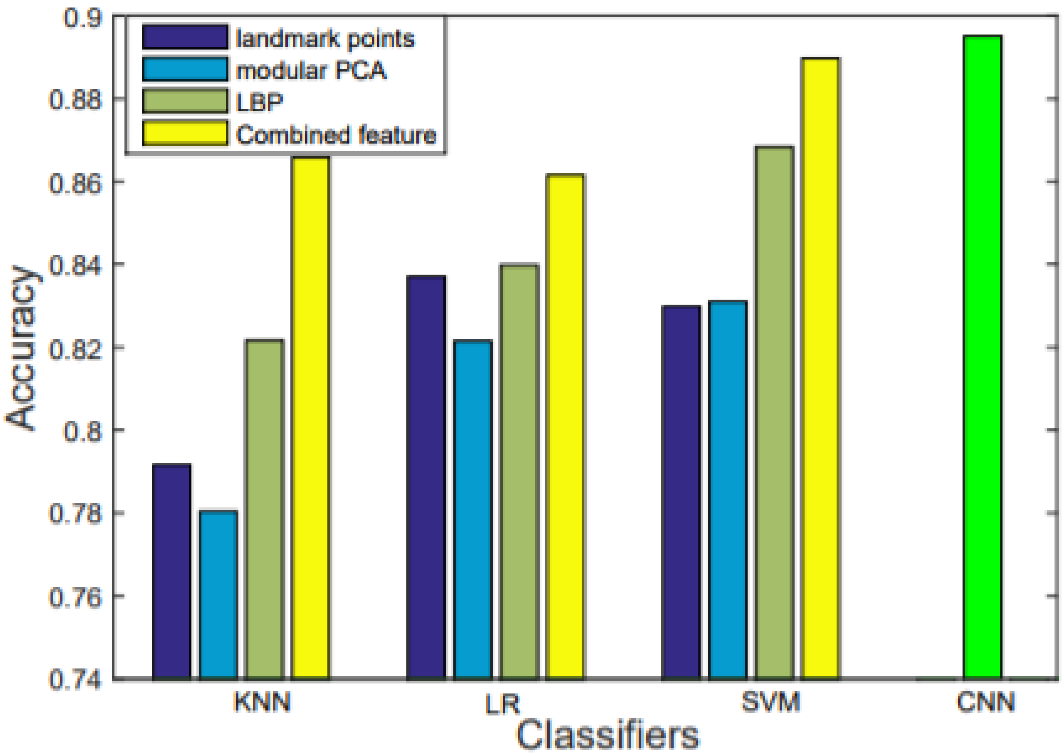
■ Automated Inference on Criminality using Face Images

https://arxiv.org/pdf/1611.04135v1.pdf

(a) Three samples in criminal ID photo set S_c.

(b) Three samples in non-criminal ID photo set S_n

Figure 1. Sample ID photos in our data set.

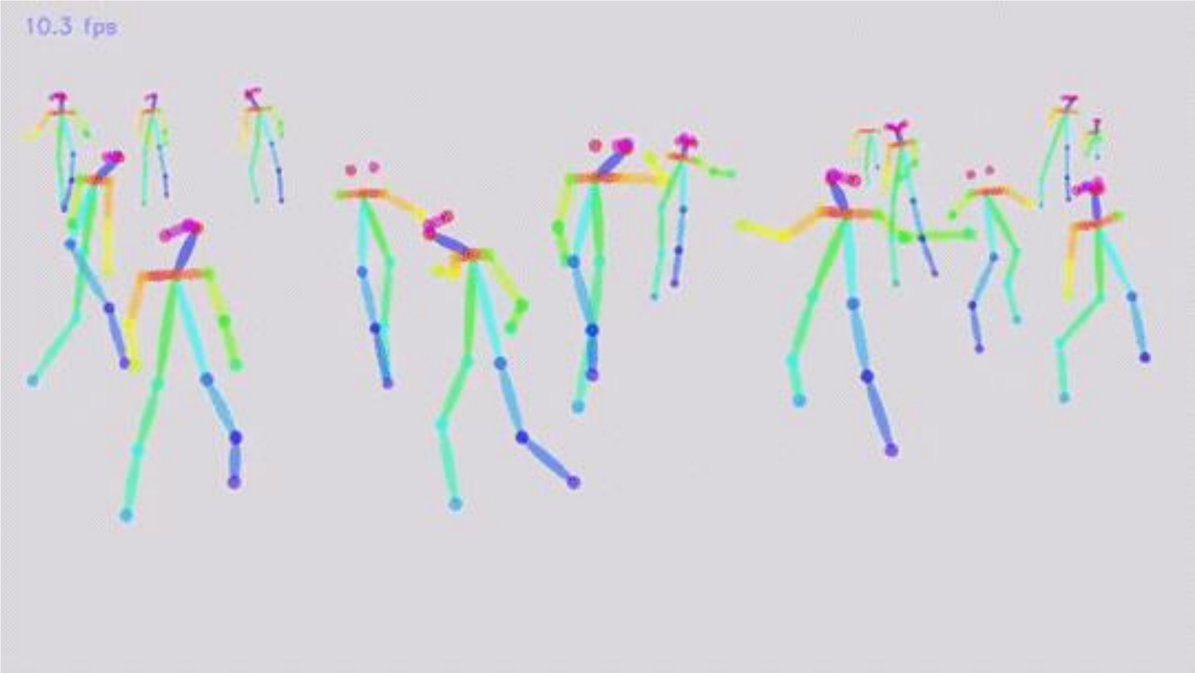


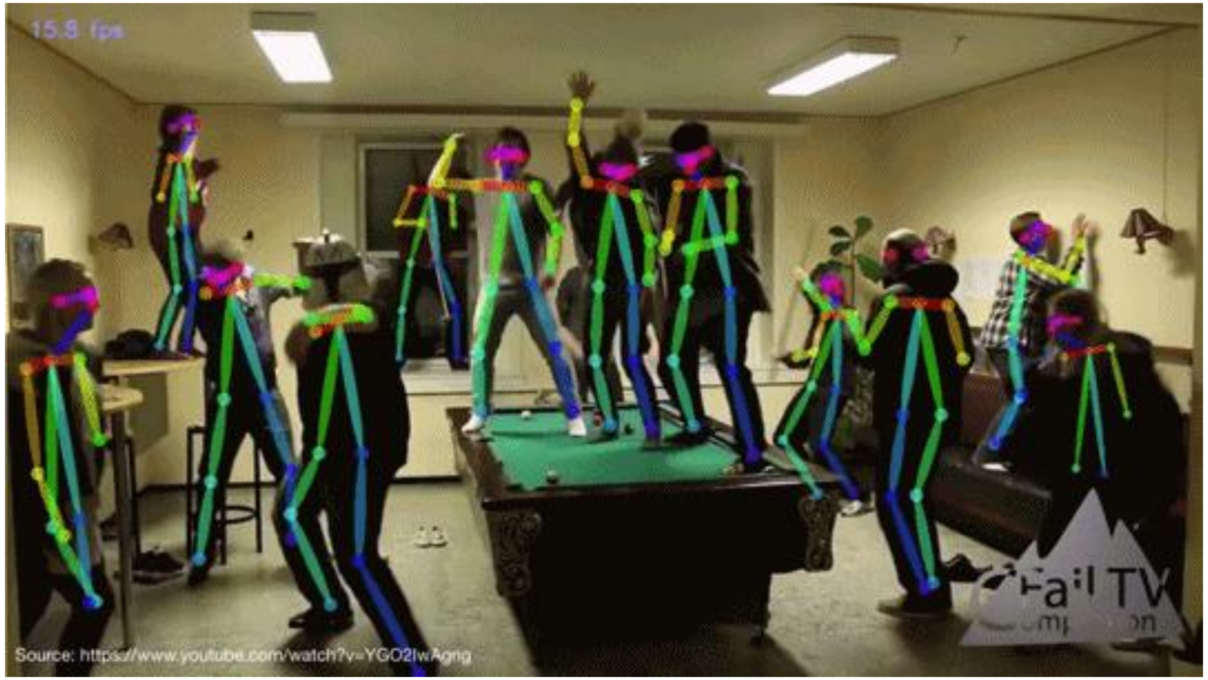
Crazy Machine Learning

- **Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields**

https://arxiv.org/abs/1611.08050 CVPR 2017

https://github.com/ZheC/Realtime_Multi-Person_Pose_Estimation





Crazy Machine Learning

- DeepWarp: Photorealistic Image Resynthesis for Gaze Manipulation

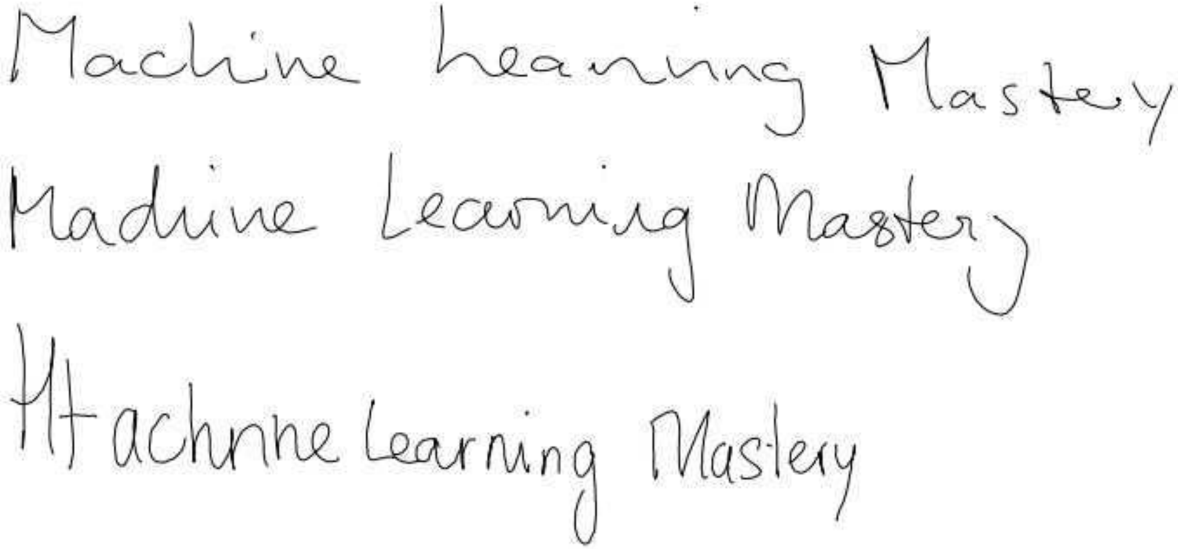
http://sites.skoltech.ru/compvision/projects/deepwarp/files/deepwarp_eccv2016.pdf

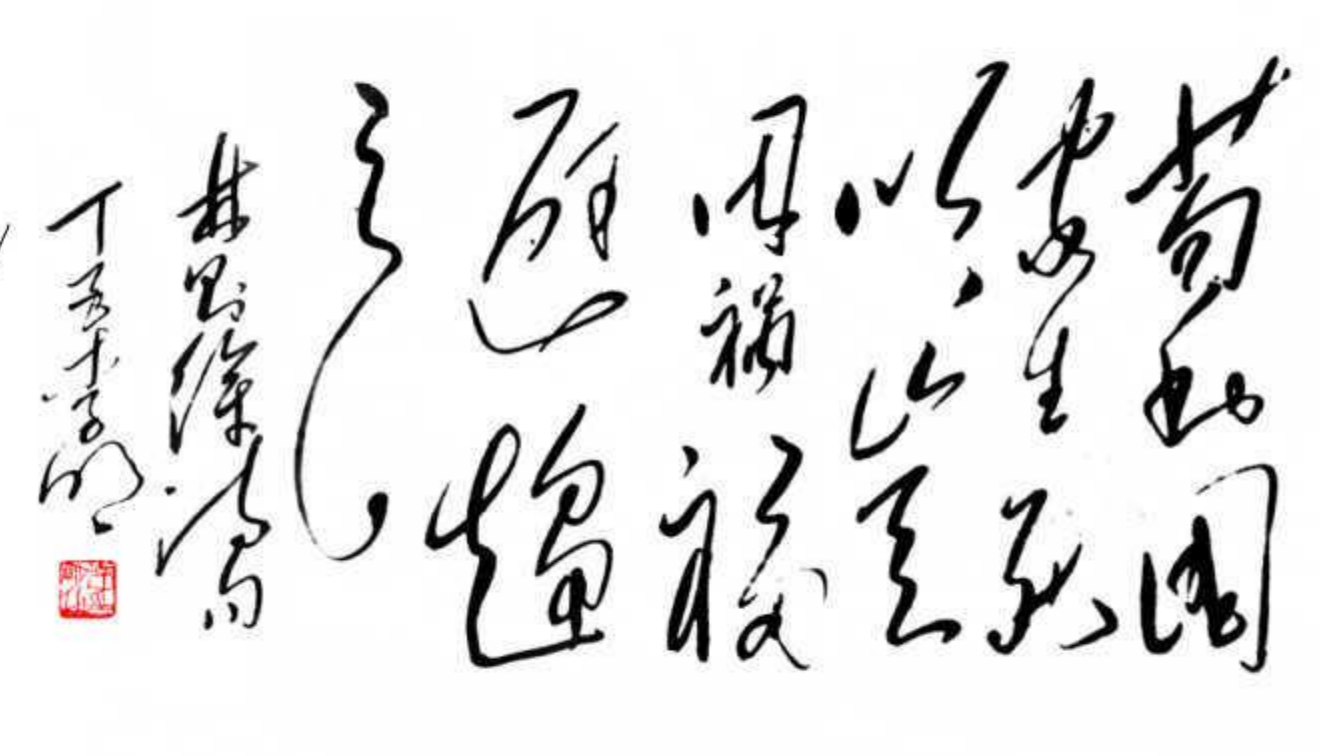
http://163.172.78.19/

Crazy Machine Learning

- **Handwriting Generation**

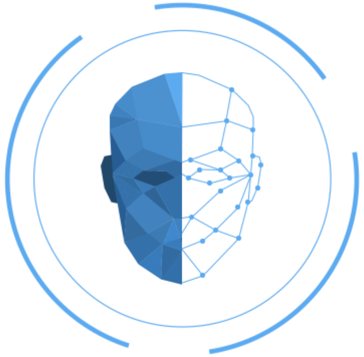
https://arxiv.org/pdf/1308.0850v5.pdf





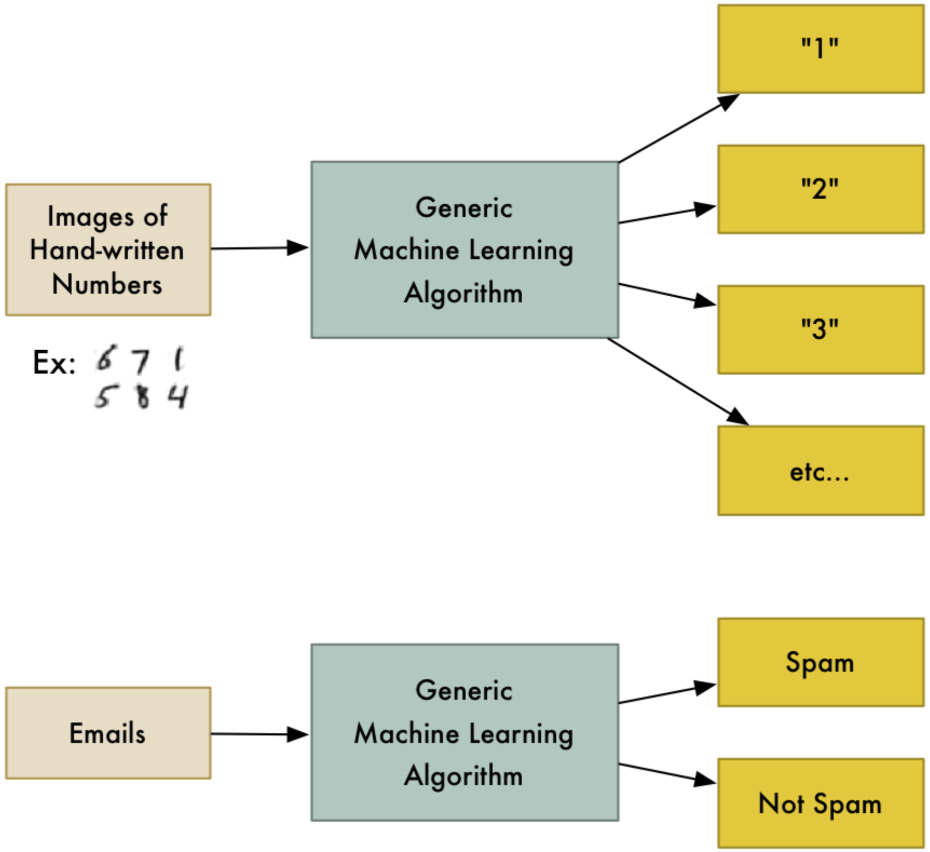
Outline

- What is Machine Learning and Why U Should Care?
- Easiest Introduction to Machine Learning
- Deep Learning and CNN
- Face Recognition
- Language Translation and RNN



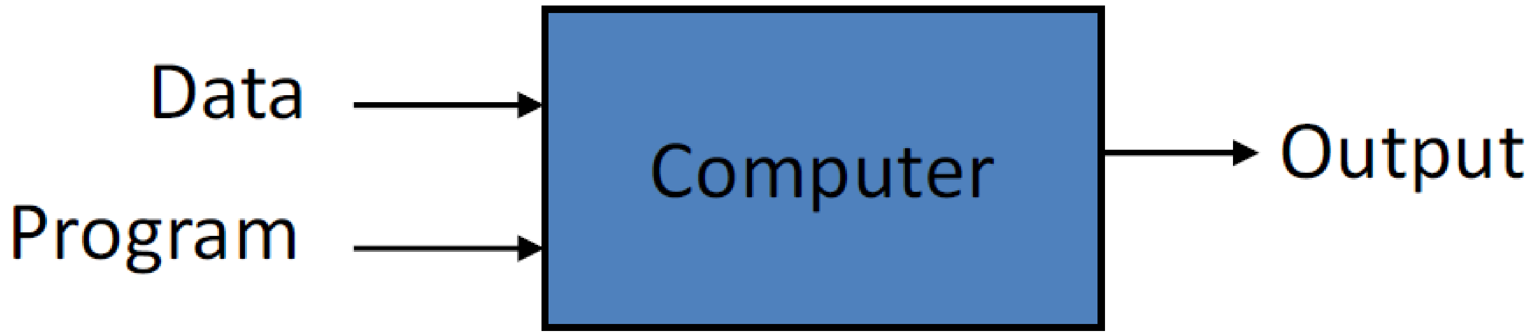
What is machine learning?

- Machine learning could **tell you something interesting about a set of data without you having to write any custom code** specific to the problem.
- Instead of writing code, you feed data to the generic algorithm and it builds its own logic based on the data

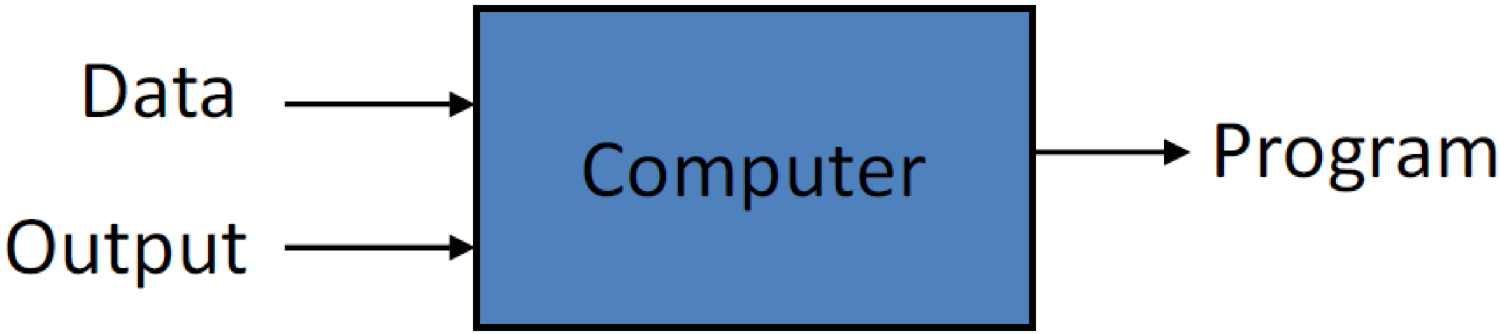


Traditional Programming VS. Machine Learning

Traditional Programming



Machine Learning

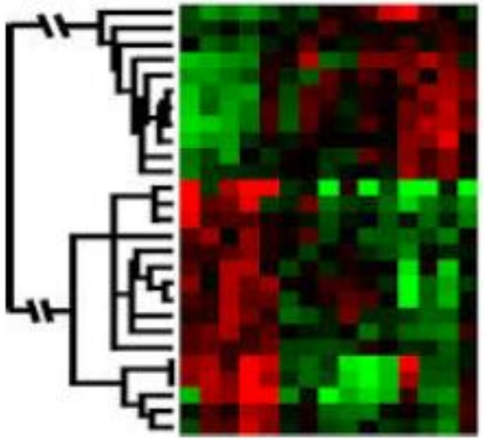


Why do we use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans can't explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)



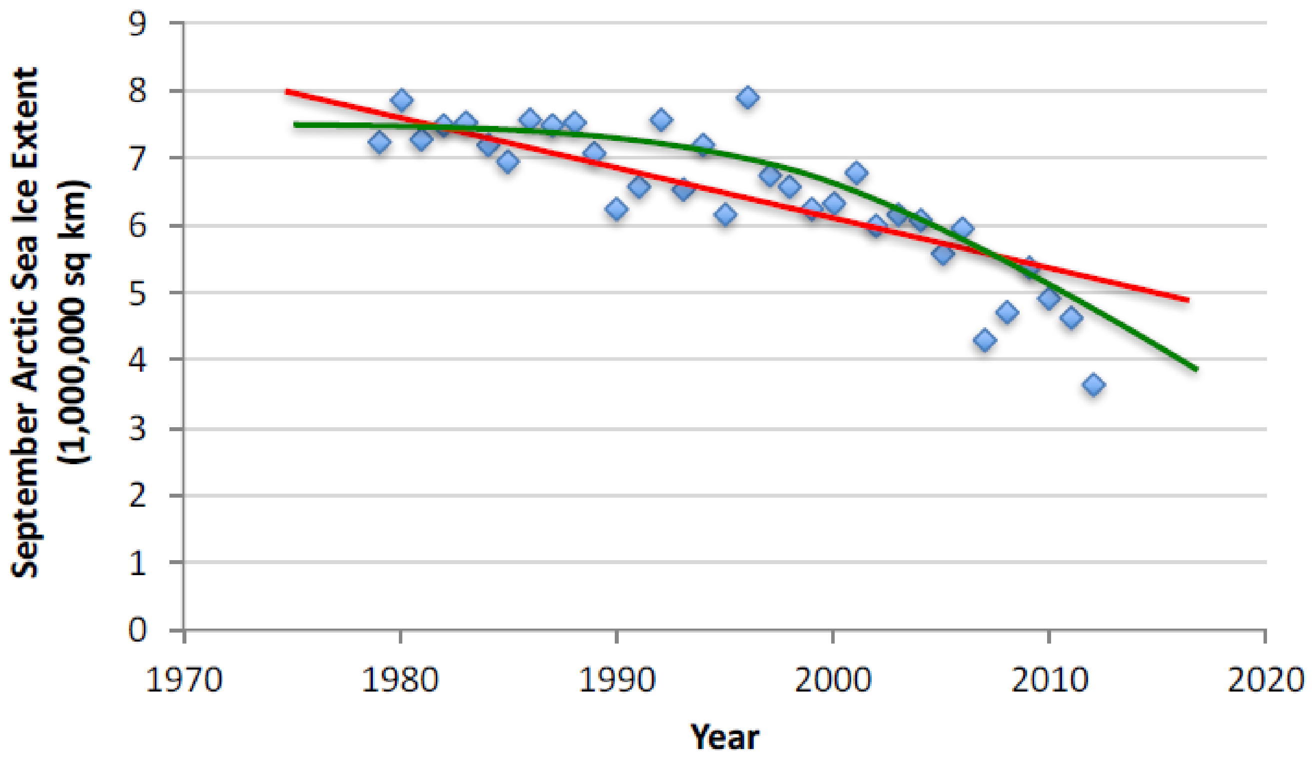



Types of Learning

- **Supervised learning（有监督学习）**
  - Given: training data + desired outputs (labels)
- **Unsupervised learning（无监督学习）**
  - Given: training data (without desired outputs)
- **Semi-supervised learning（半监督学习）**
  - Given: training data + a few desired outputs
- **Reinforcement learning（强化学习）**
  - Rewards from sequence of actions

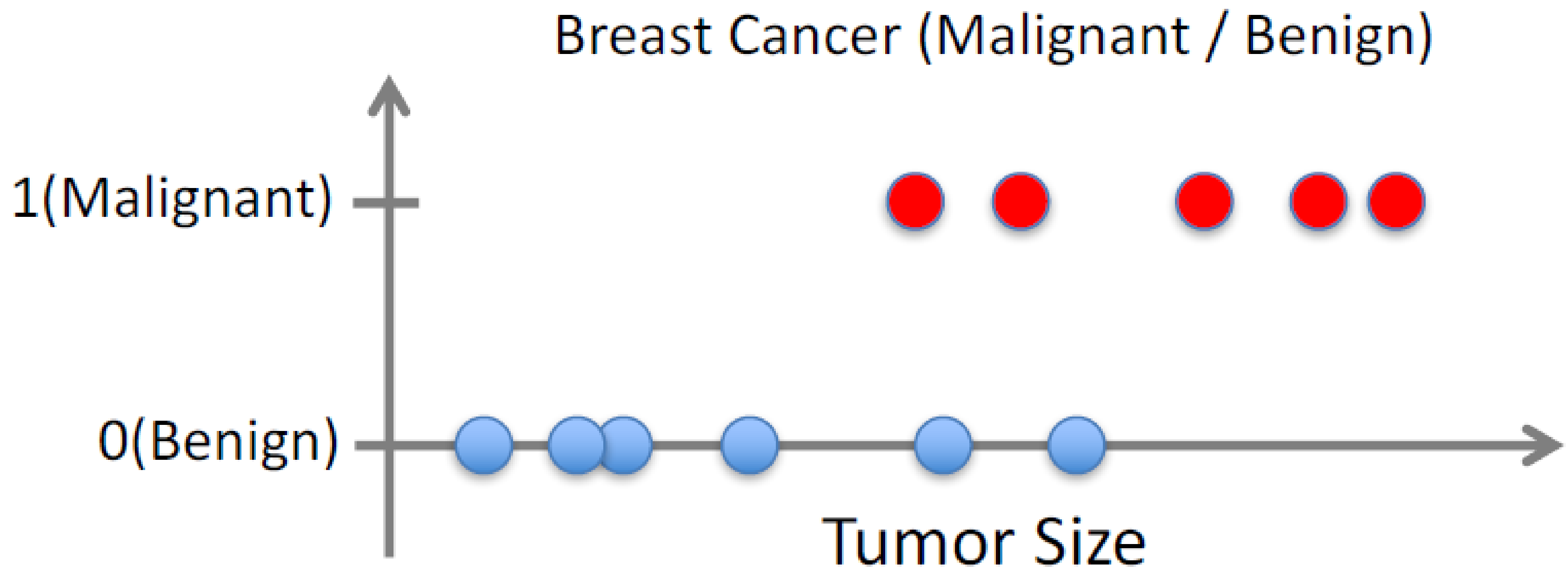
Supervised Learning: Regression

- Given (x1, y1), (x2, y2), ..., (xn, yn)
- Learn a function f(x) to predict y given x
  - y is real-valued == regression



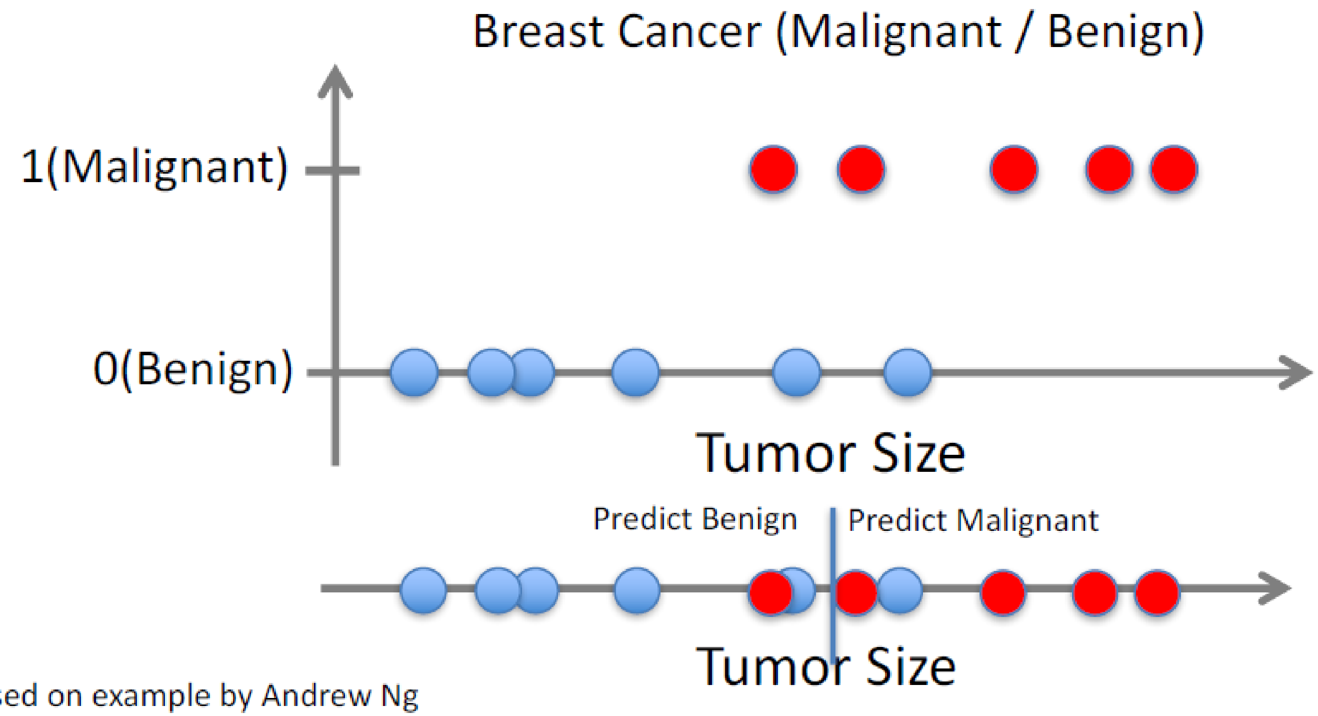
Supervised Learning: Classification

- Given (x1, y1), (x2, y2), ..., (xn, yn)
- Learn a function f(x) to predict y given x
  - y is categorical == classification



Supervised Learning: Classification

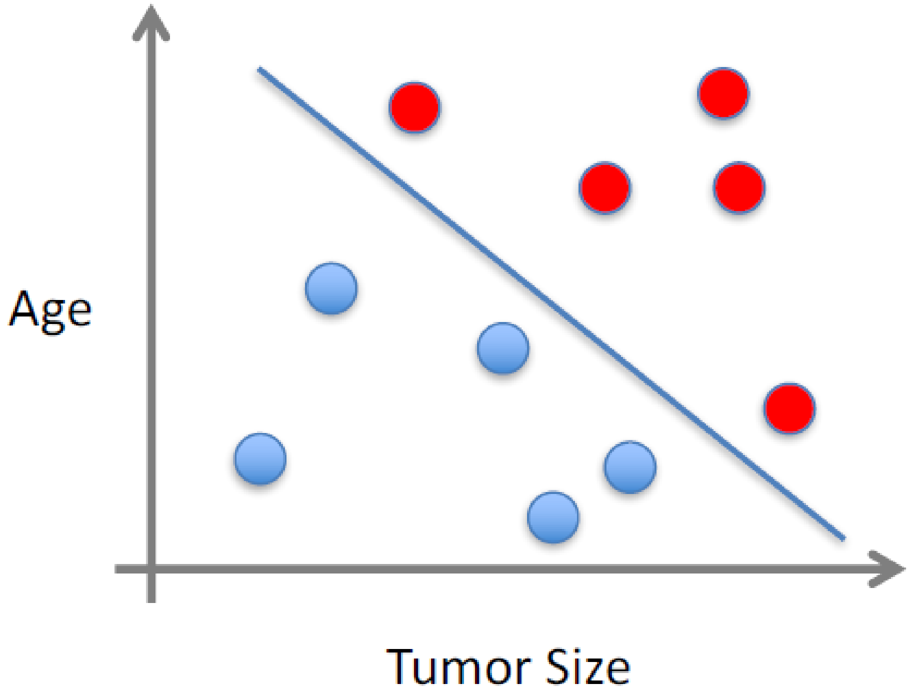
- Given (x1, y1), (x2, y2), ..., (xn, yn)
- Learn a function f(x) to predict y given x
  - y is categorical == classification



Based on example by Andrew Ng

Supervised Learning

- x can be multi-dimensional
  - – Each dimension corresponds to an attribute

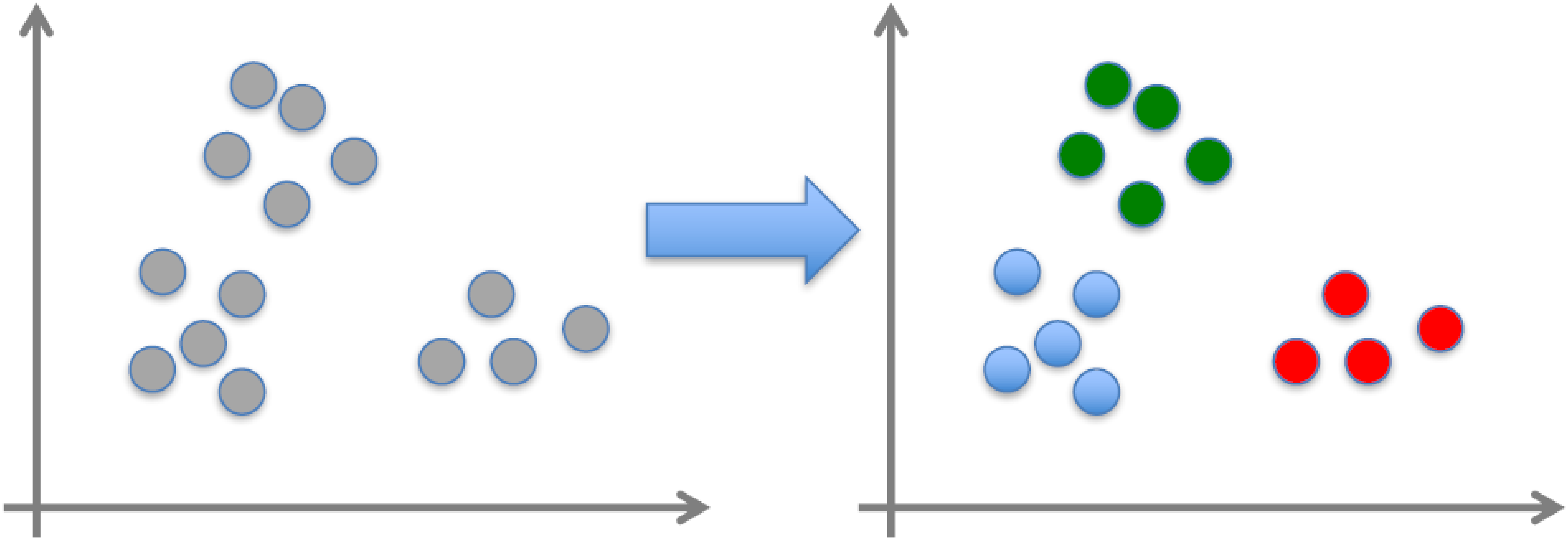


- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape

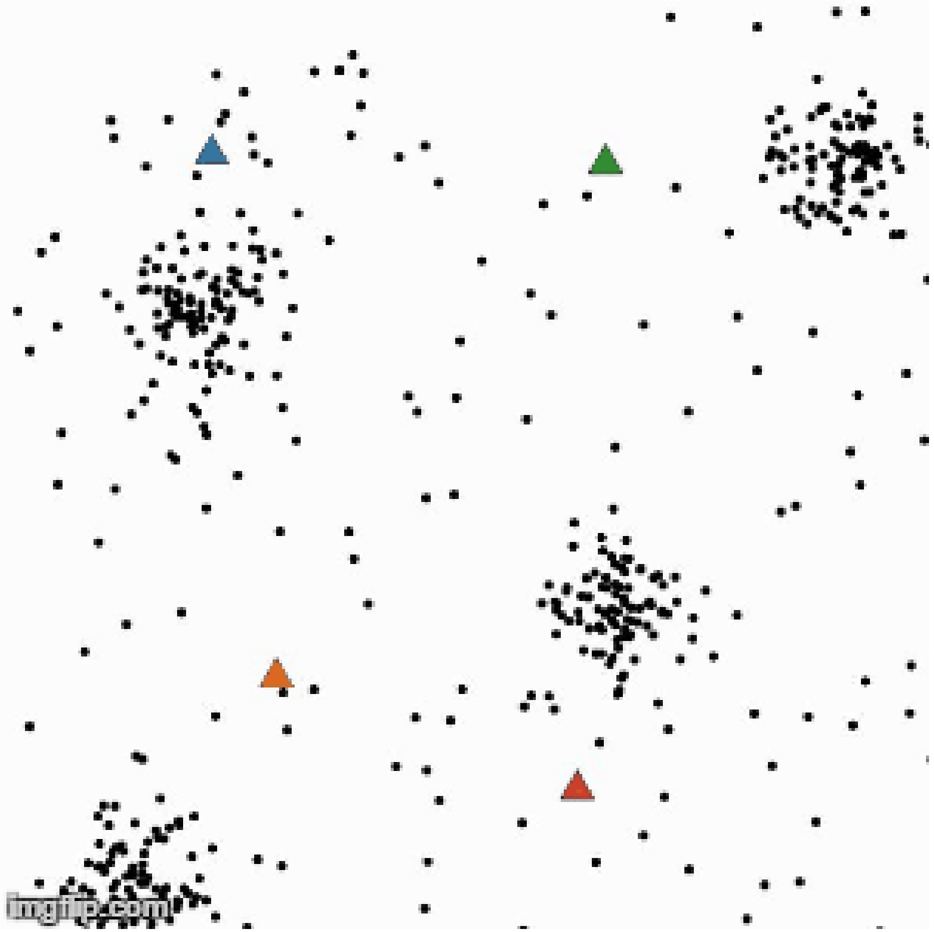
…

Unsupervised Learning

- **Given x1, x2, ..., xn (without labels)**
- **Output hidden structure behind the x's**
  - E.g., clustering



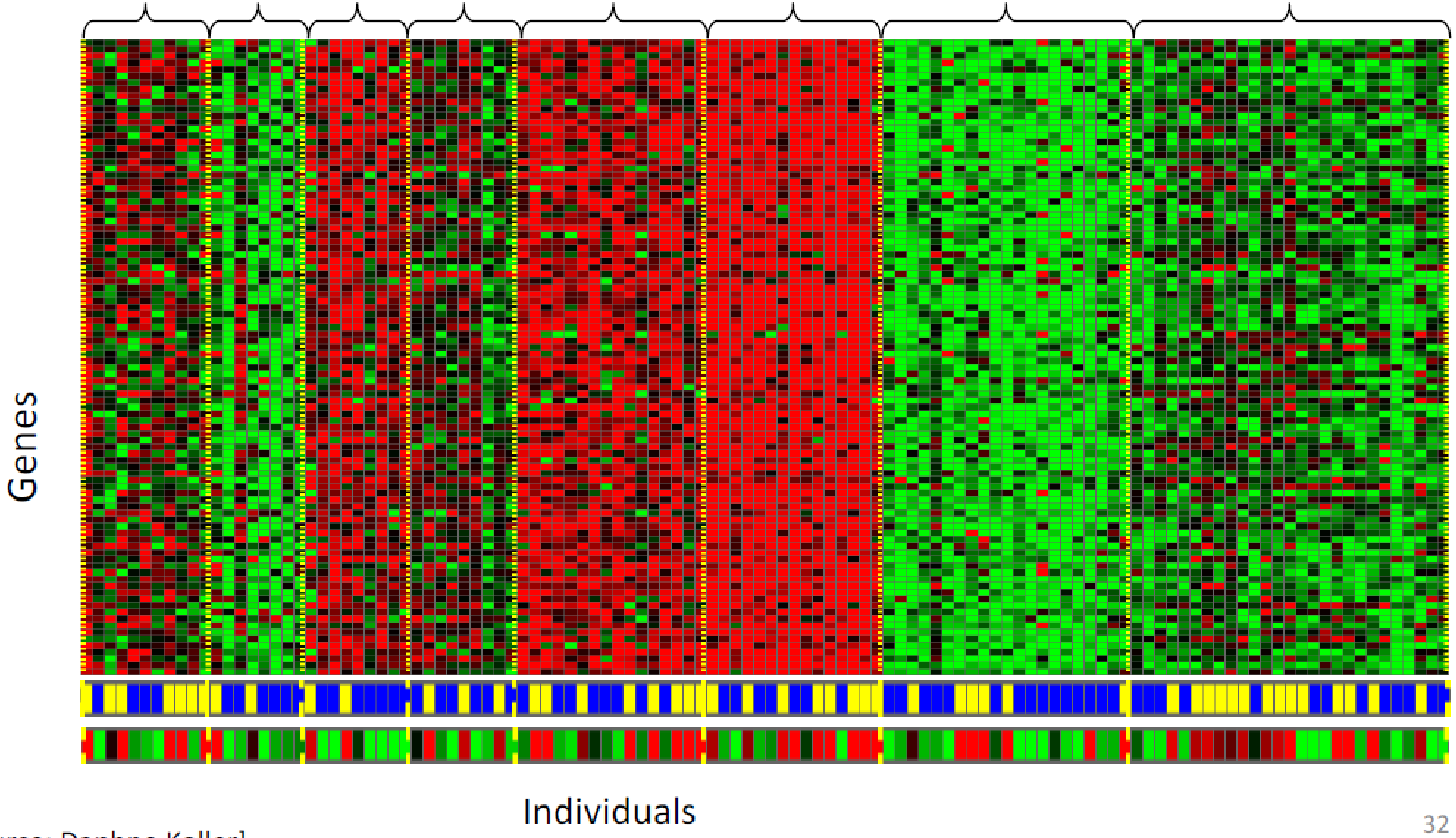
Unsupervised Learning —— K-means Clustering



- 根据预先设定的分类数目，在样本空间随机选择相应数目的点做为起始聚类中心点
- 将空间中到每个起始中心点距离最近的点作为一个集合，完成第一次聚类
- 获得第一次聚类集合所有点的平均值做为新的中心点，进行第二次聚类
- 直到得到的聚类中心点不再变化或达到尝试的上限，则完成了聚类过程

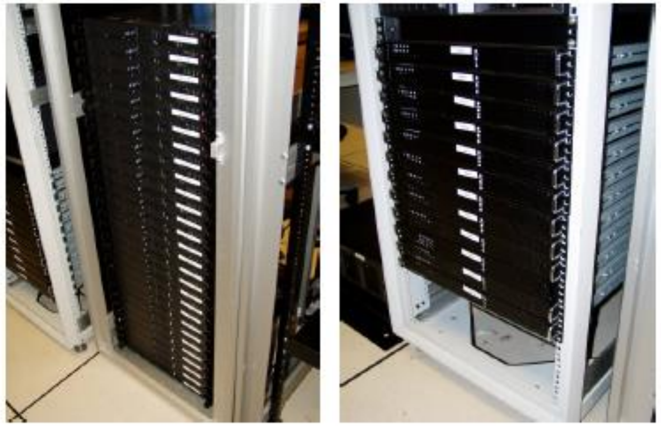
Unsupervised Learning

- Genomics application: group individuals by genetic similarity

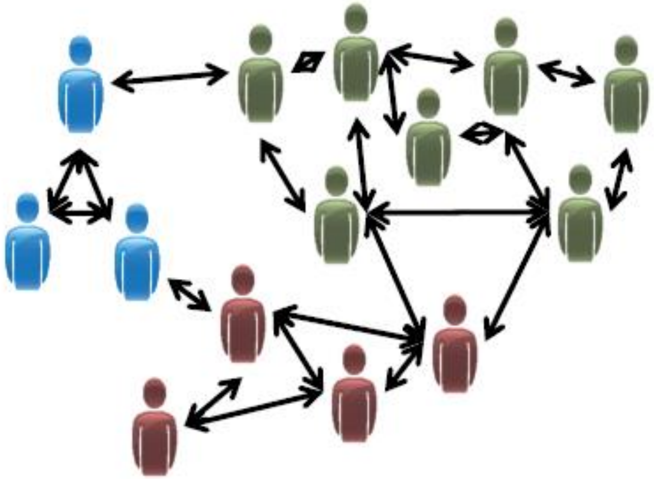


[Source: Daphne Koller]

Unsupervised Learning



Organize computing clusters



Social network analysis

Market segmentation

Astronomical data analysis

Reinforcement Learning（强化学习）

- 可以用电子游戏来理解强化学习，电子游戏也是强化学习算法中应用最广泛的一个领域。在经典电子游戏中，有以下几类对象：
  - 代理（agent，即智能体），可自由移动，对应玩家；
  - 动作，由代理做出，包括向上移动和出售物品等；
  - 奖励，由代理获得，包括金币和杀死其他玩家等；
  - 环境，指代理所处的地图或房间等；
  - 状态，指代理的当前状态，如位于地图中某个特定方块或房间中某个角落；
  - 目标，指代理目标为获得尽可能多的奖励；

这些对象是强化学习的组成部分，在强化学习中，设置好环境后，我们能通过逐个状态来指导代理，当代理做出正确动作时会得到奖励。

Reinforcement Learning（强化学习）

- **强化学习的基本工作原理**
  - 在每个状态下，代理会对所有可能动作（上下左右）进行计算和评估，并选择能获得最多奖励的动作
  - 进行若干步后，小鼠会熟悉这个迷宫
- **如何确定哪个动作会得到最佳结果?**
  - 策略学习、Q-learning等
  - 强化学习中并没有大量的原始已知输入数据,机器需要在变化的环境中通过大量的多次的试错学习,再根据某种规则找到产生最佳结果的最佳路径,从而做出最佳决策

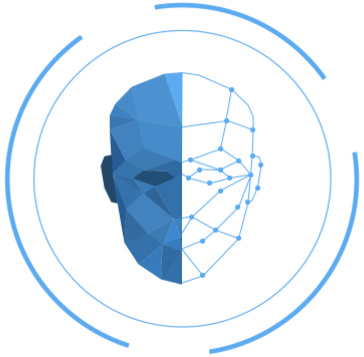
Reinforcement Learning

- 强化学习的大部分应用都在电子游戏方面
- 最新的强化学习算法在经典和现代游戏中取得了很不错的效果

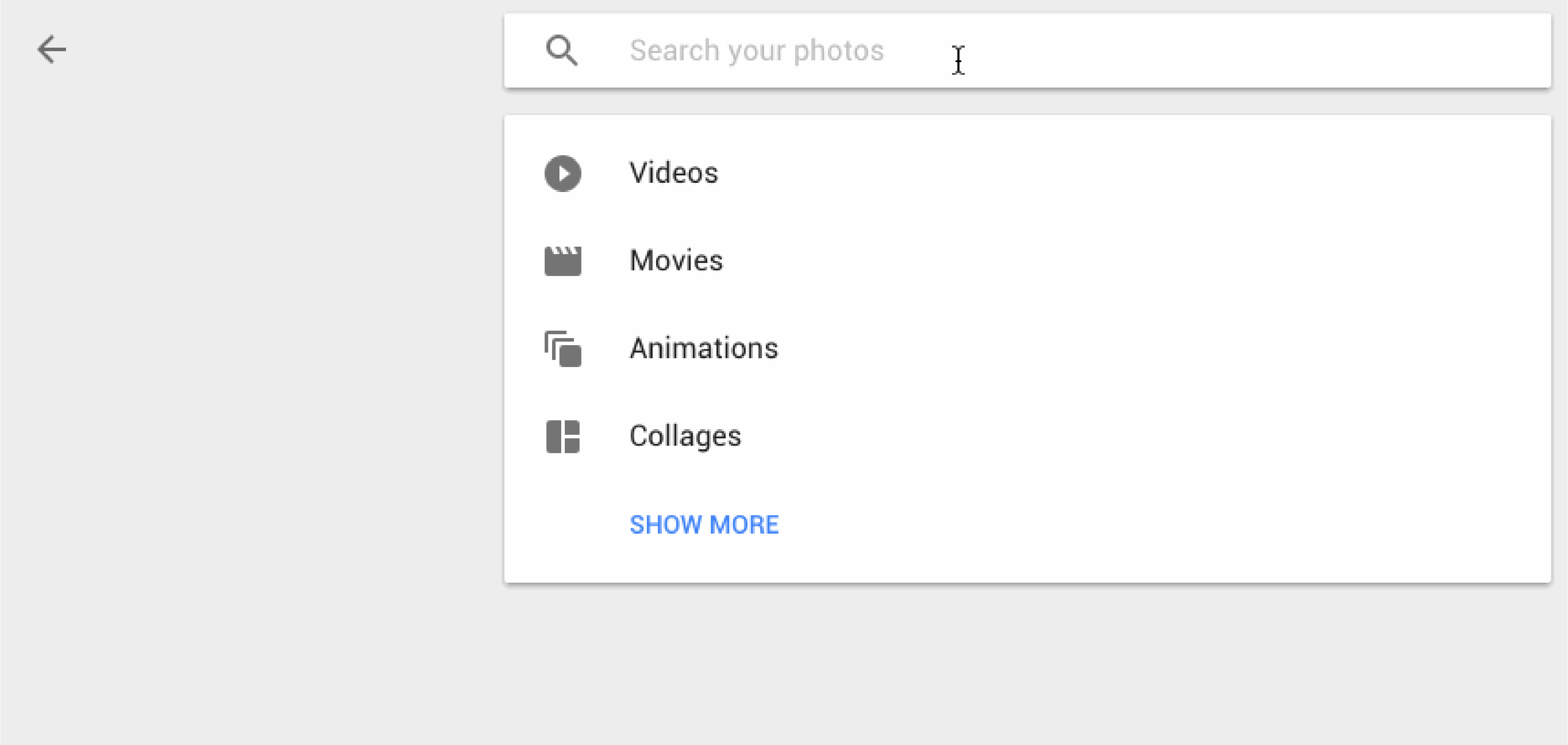
https://www.youtube.com/watch?v=4cgWya-wjgY

Outline

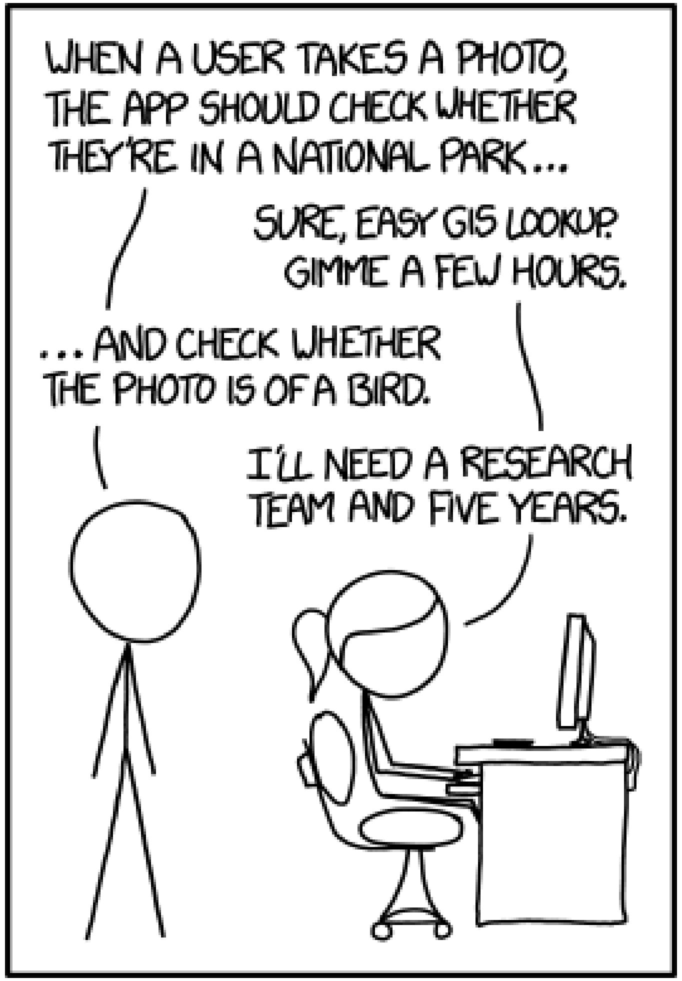
- What is Machine Learning and Why U Should Care?
- Easiest Introduction to Machine Learning
- Deep Learning and CNN
- Face Recognition
- Language Translation and RNN



Search Your Own Photo by Description



Recognizing Objects with Deep Learning

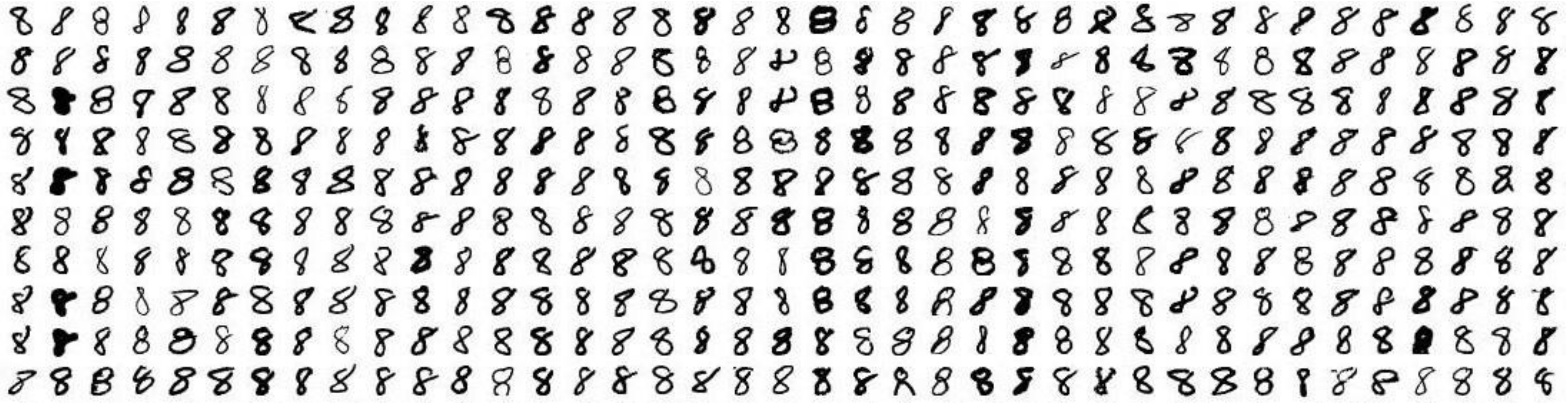


IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

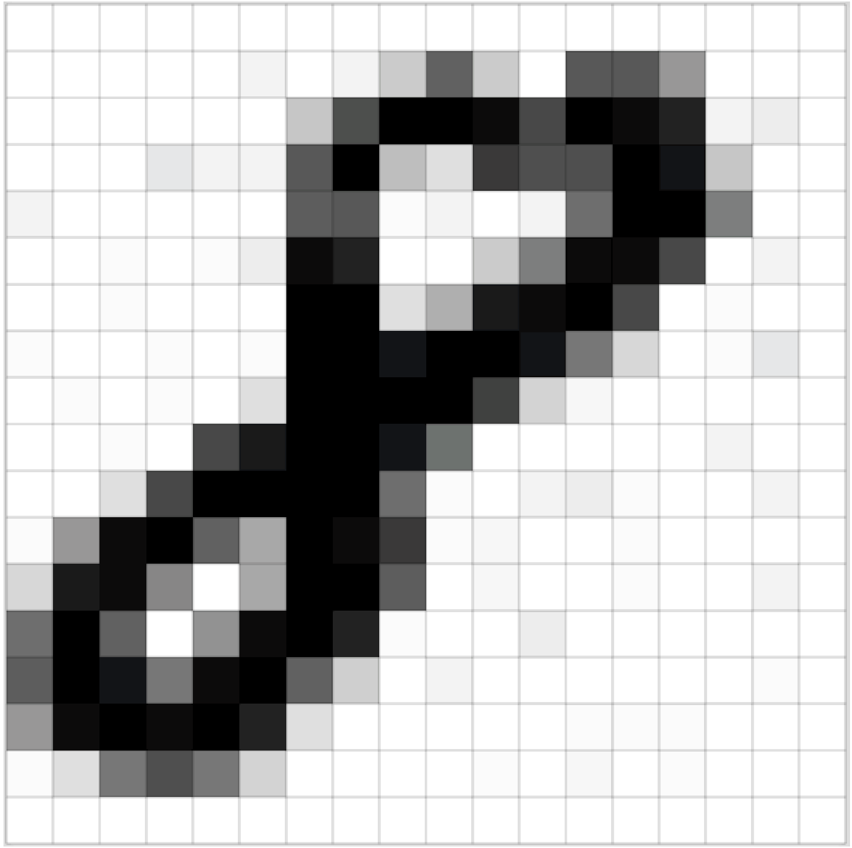
- 3-year-old child can recognize a photo of a bird
- but figuring out **how to make a computer recognize objects** has puzzled the very best computer scientists for over 50 years

Starting Simple - recognize handwritten text

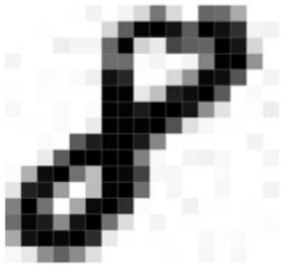
- same generic algorithms can be reused with different data to solve different problems
- we’ll only try to recognize one letter — the numeral “8”



If you think about it, everything is just numbers



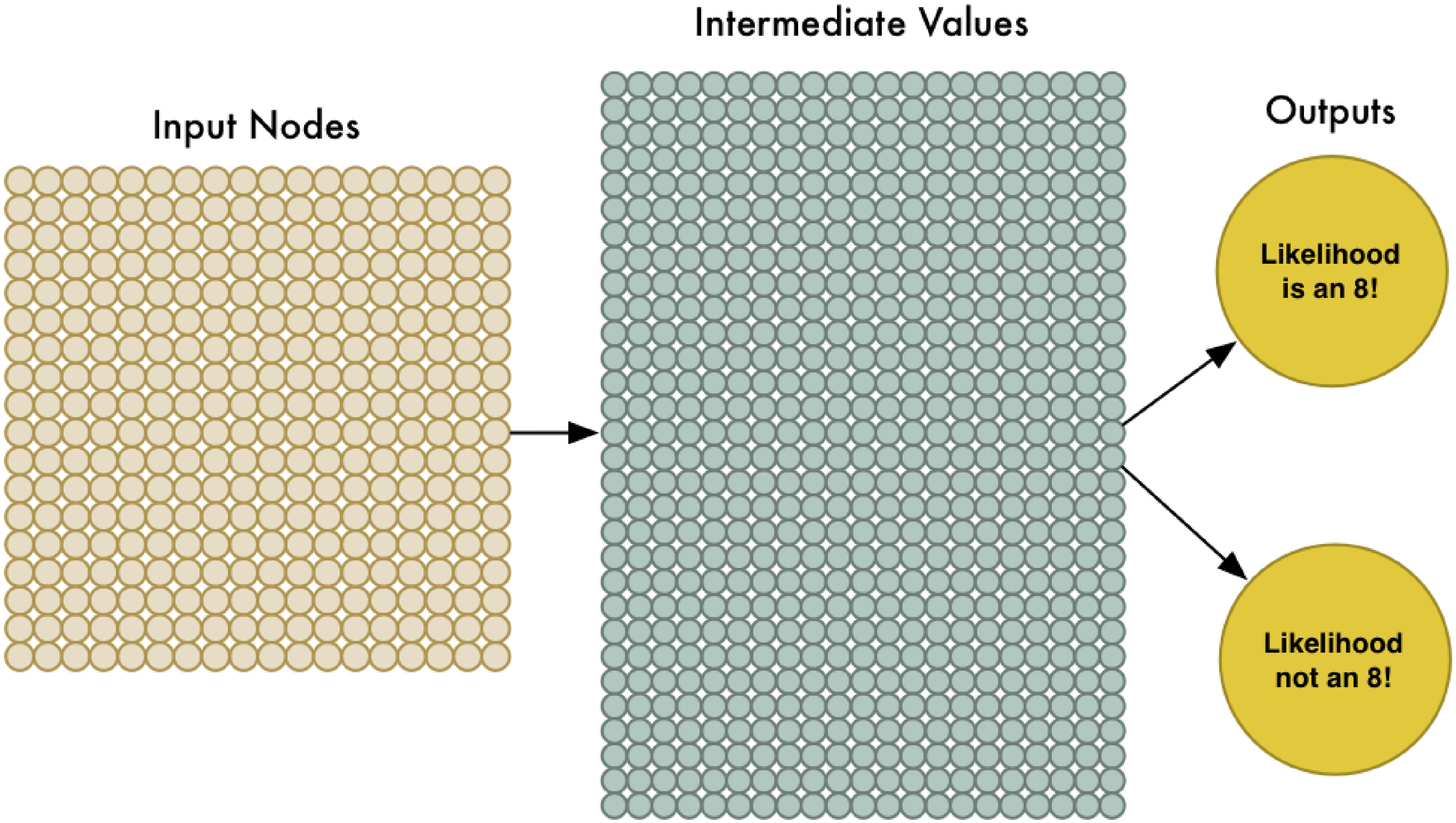
- Color images have three channels – red, green and blue
- grayscale images only have one channel
- The value of each pixel in the matrix will range from 0 to 255



=

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 12, 0, 11, 39, 137, 37, 0, 152, 147, 84, 0, 0, 0, 0,
0, 1, 0, 0, 0, 41, 160, 250, 255, 235, 162, 255, 238, 206, 11, 13, 0, 0, 0, 0, 16, 9, 9, 150, 251, 45, 21, 184, 159, 154, 2
55, 233, 40, 0, 0, 10, 0, 0, 0, 0, 0, 145, 146, 3, 10, 0, 11, 124, 253, 255, 107, 0, 0, 0, 0, 3, 0, 4, 15, 236, 216, 0, 0,
38, 109, 247, 240, 169, 0, 11, 0, 1, 0, 2, 0, 0, 0, 253, 253, 23, 62, 224, 241, 255, 164, 0, 5, 0, 0, 6, 0, 0, 4, 0, 3, 252
, 250, 228, 255, 255, 234, 112, 28, 0, 2, 17, 0, 0, 2, 1, 4, 0, 21, 255, 253, 251, 255, 172, 31, 8, 0, 1, 0, 0, 0, 0, 0, 4,
 0, 163, 225, 251, 255, 229, 120, 0, 0, 0, 0, 0, 11, 0, 0, 0, 0, 21, 162, 255, 255, 254, 255, 126, 6, 0, 10, 14, 6, 0, 0, 9
, 0, 3, 79, 242, 255, 141, 66, 255, 245, 189, 7, 8, 0, 0, 5, 0, 0, 0, 0, 26, 221, 237, 98, 0, 67, 251, 255, 144, 0, 8, 0, 0
, 7, 0, 0, 11, 0, 125, 255, 141, 0, 87, 244, 255, 208, 3, 0, 0, 13, 0, 1, 0, 1, 0, 0, 145, 248, 228, 116, 235, 255, 141, 34
, 0, 11, 0, 1, 0, 0, 0, 1, 3, 0, 85, 237, 253, 246, 255, 210, 21, 1, 0, 1, 0, 0, 6, 2, 4, 0, 0, 0, 6, 23, 112, 157, 114, 32
, 0, 0, 0, 0, 2, 0, 8, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Starting Simple - recognize handwritten text



Training

- When we feed in an "8", we'll tell it the probability the image is an "8" is 100% and the probability it's not an "8" is 0%

Welcome to the world of (late 1980's-era) image recognition!

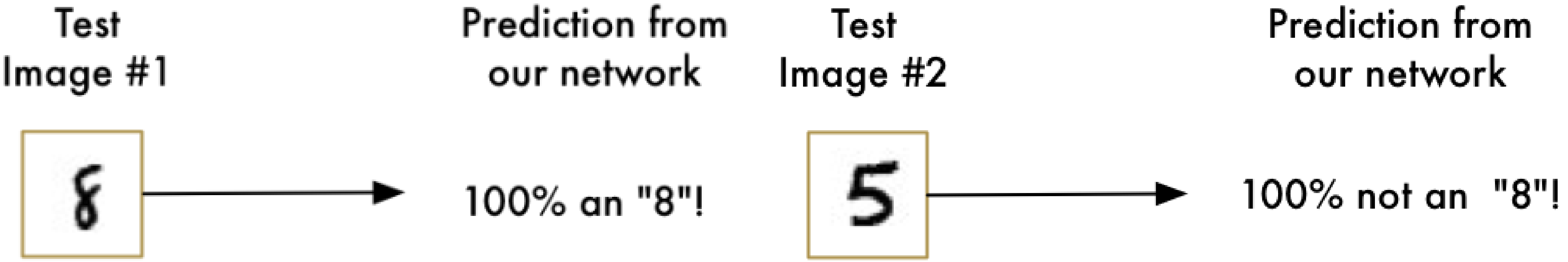
Simply feeding pixels into a neural network actually worked to build image recognition!

Machine learning is magic! …right?

呵呵，当然，不会，这么，简单…

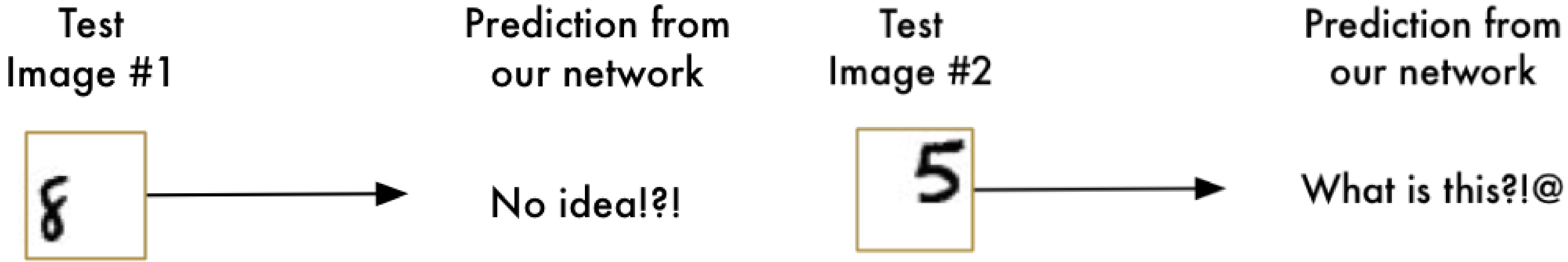
Good News

- good news is that our “8” recognizer really does work well on simple images where **the letter is right in the middle** of the image



However…

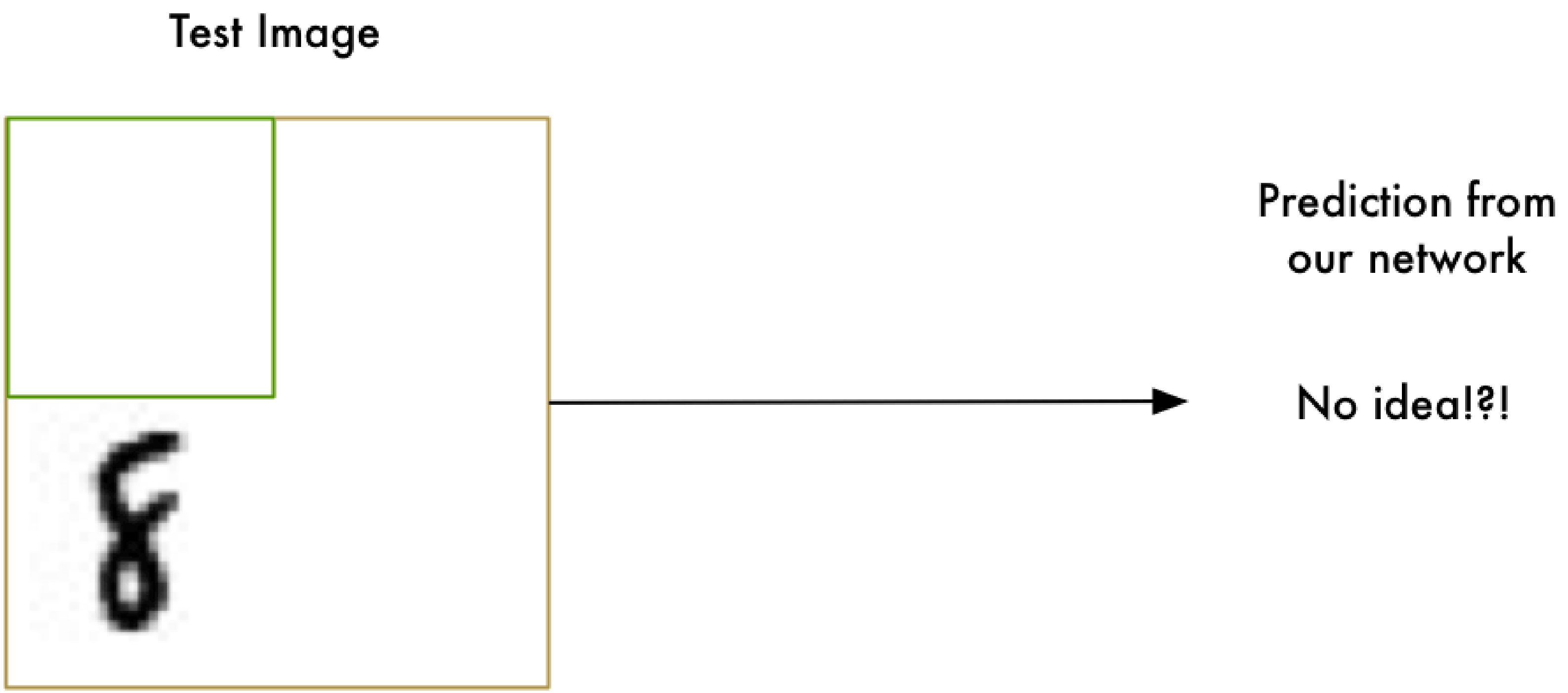
- Our “8” recognizer *totally fails* to work when the letter isn’t perfectly centered in the image.
- Just the slightest position change ruins everything



how to make our neural network work in cases where the “8” isn’t perfectly centered?

Brute Force Idea #1: Searching with a Sliding Window

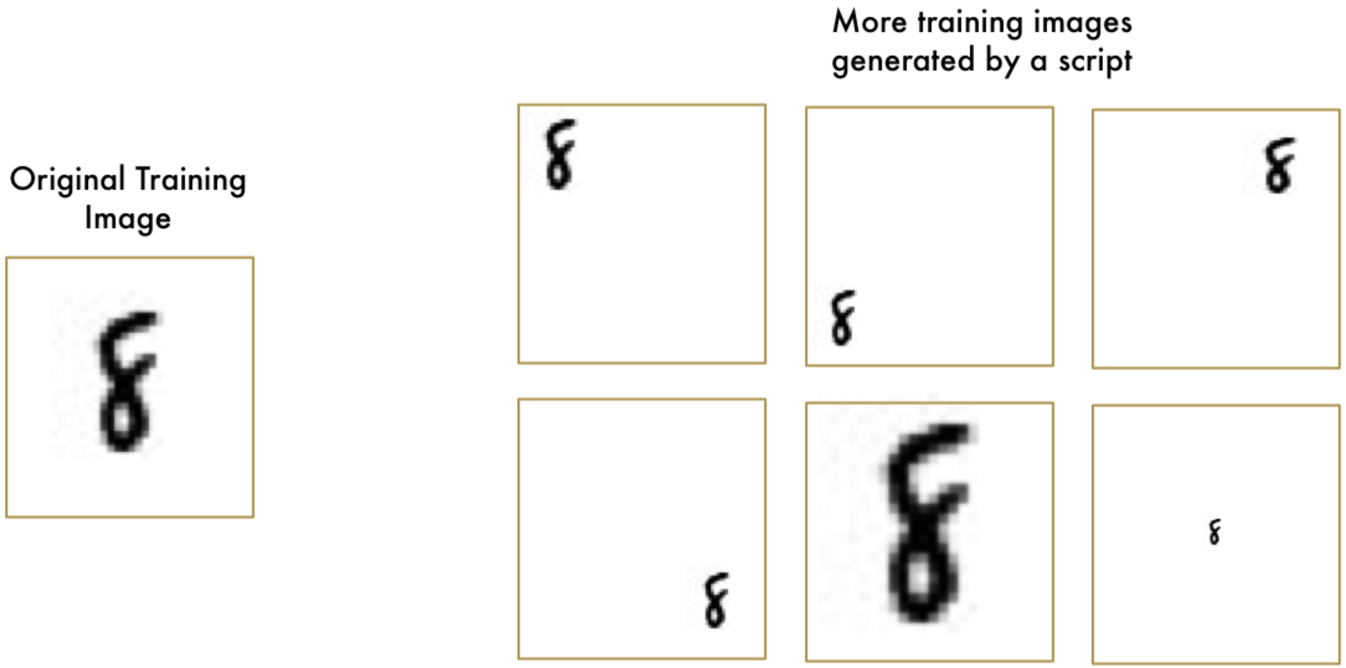
- What if we just scan all around the image for possible "8"s in smaller sections, one section at a time, until we find one?



You have to check the same image over and over looking for objects of different sizes

Brute Force Idea #2: More data and a Deep Neural Net

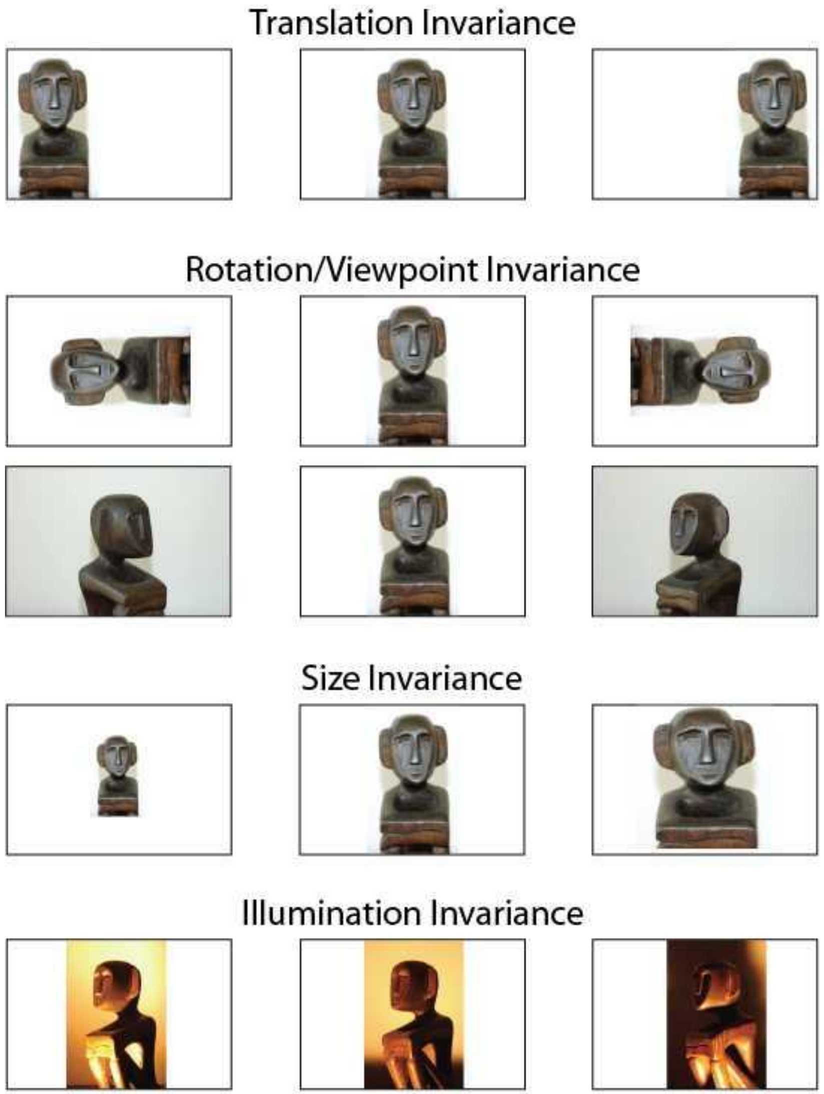
- What if we train it with more data, including "8"s in all different positions and sizes all around the image?
- just write a script to generate new images with the "8"s in all kinds of different positions in the image



The Solution is Convolution（卷积）

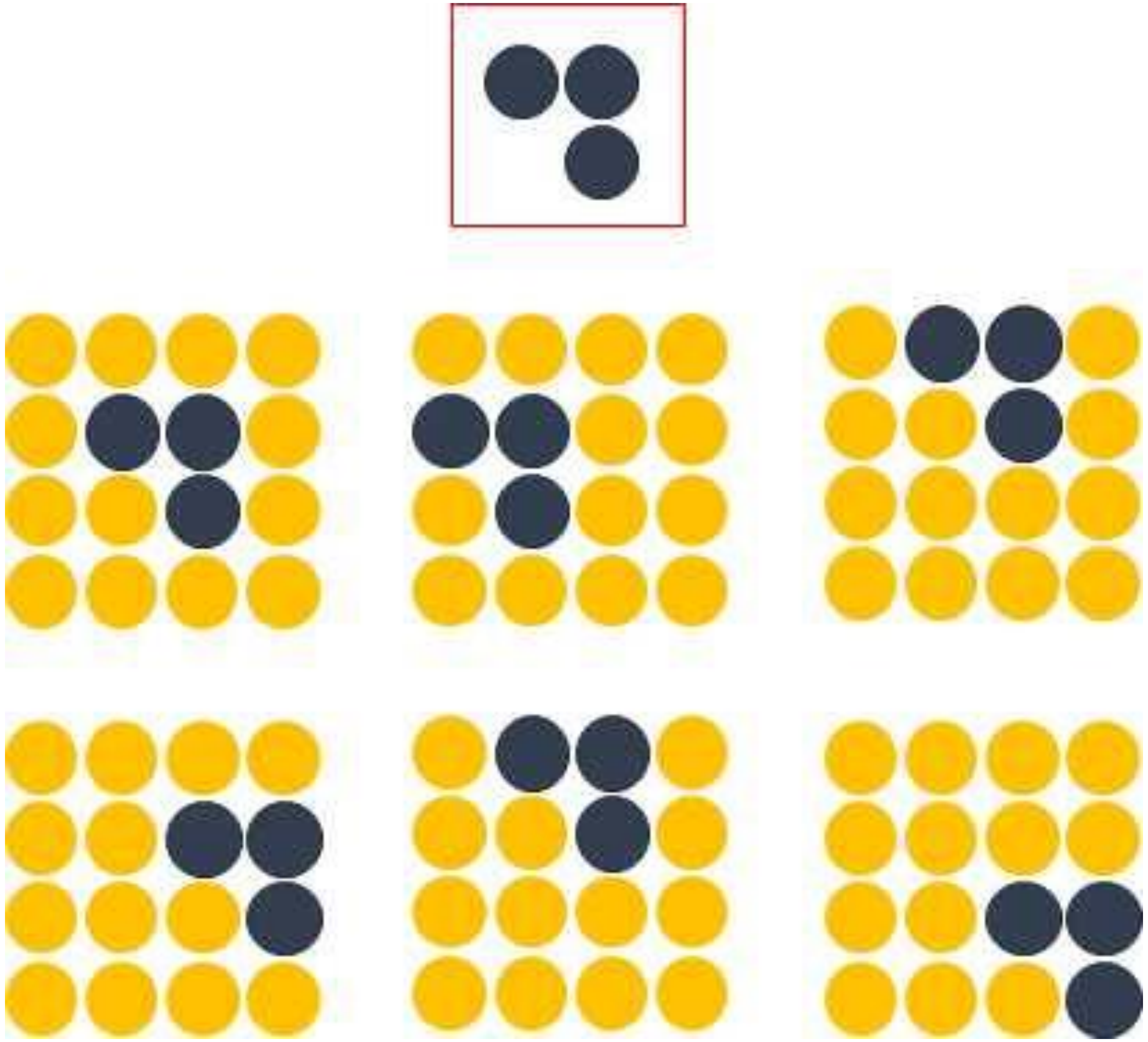
- As a human, you intuitively know that pictures have a *hierarchy* or *structure*. Consider this picture:

Translation invariance （平移不变性）



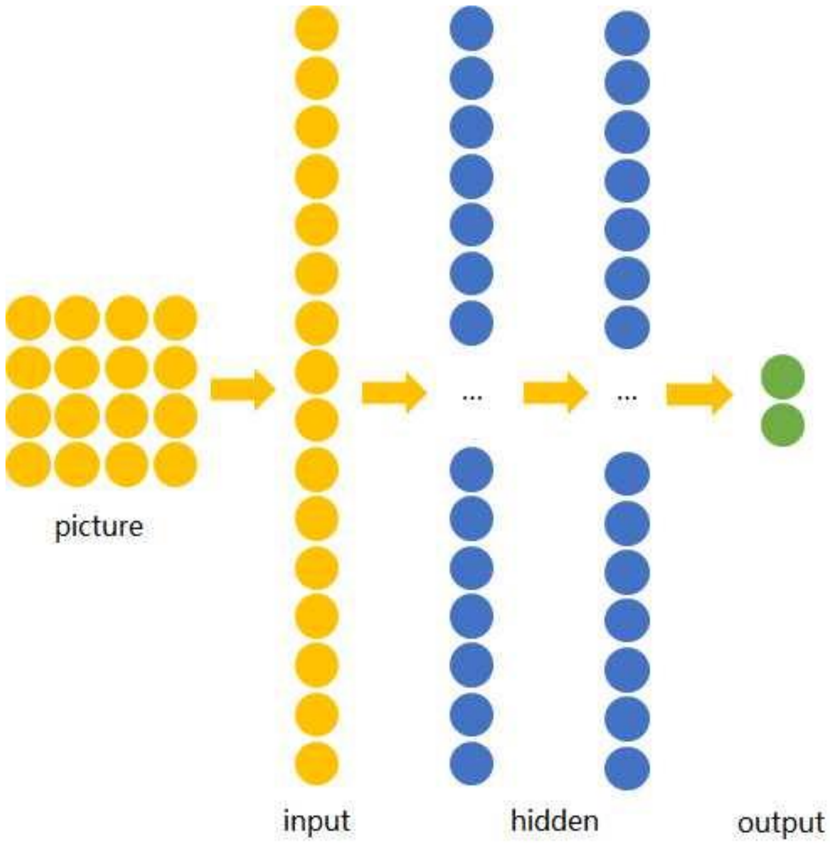
Matt Krause
mattkrau.se

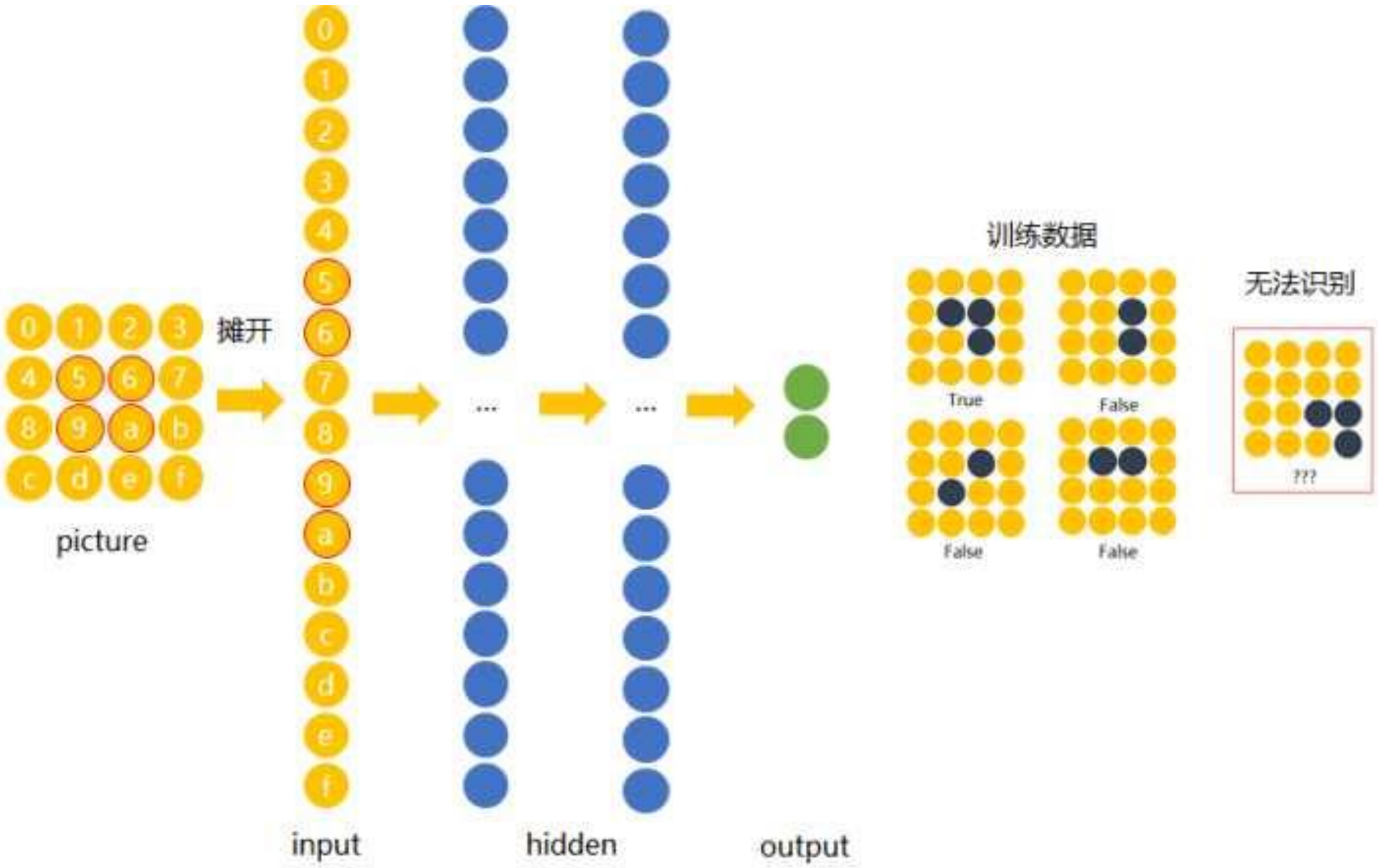
图像识别 前馈神经网络



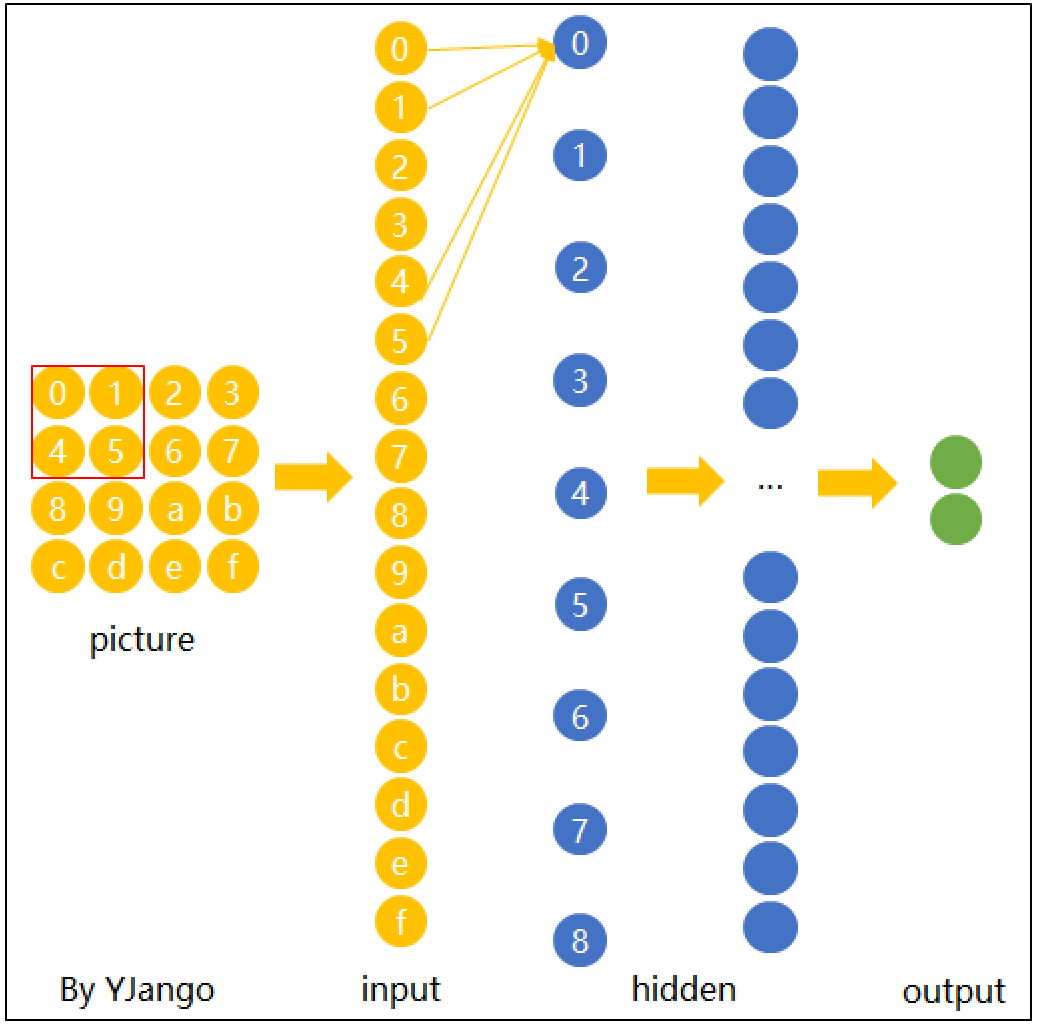
在长宽为4*4的图片中
识别是否有“横折”

图像识别 前馈神经网络


picture
...
...
input
hidden
output


0 1 2 3
4 5 6 7
8 9 a b
c d e f
picture
摊开
0
1
2
3
4
5
6
7
8
9
a
b
c
d
e
f
...
...
input
hidden
output
训练数据
True
False
False
False
无法识别
???

卷积神经网络



- 让权重在不同位置共享
- 选择一个局部区域，用这个局部区域扫描整张图片
- Filter, kernel, 或feature detector
  核，滤波器

$$\begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} \quad (1)$$

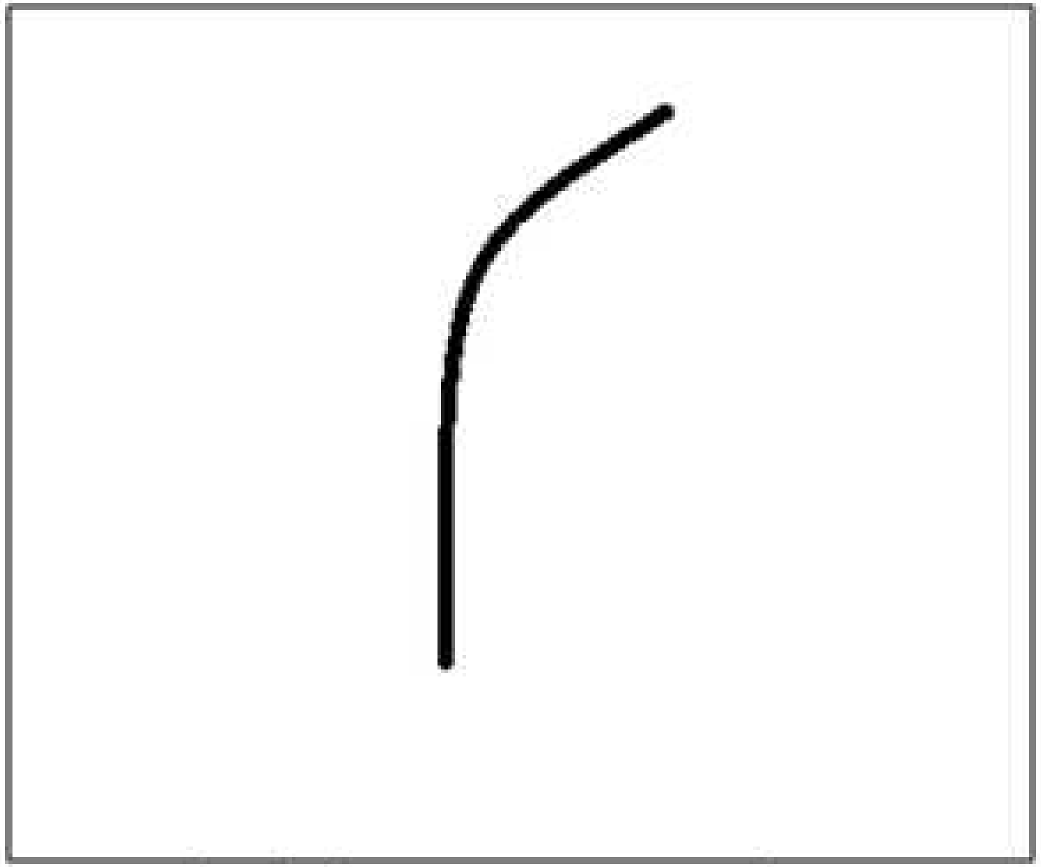
$$y_0 = x_0 * w_1 + x_1 * w_2 + x_4 * w_3 + x_5 * w_4 + b_0$$

$$y_0 = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_4 \\ x_5 \end{bmatrix} + b_0 \quad (2)$$

滤波器跟卷积神经网络有什么关系?

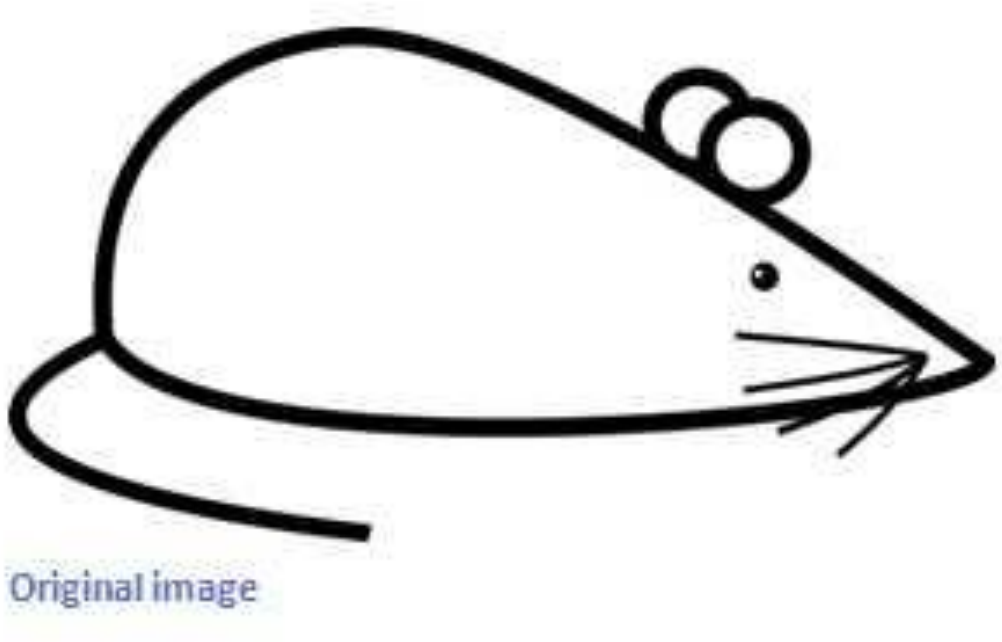
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

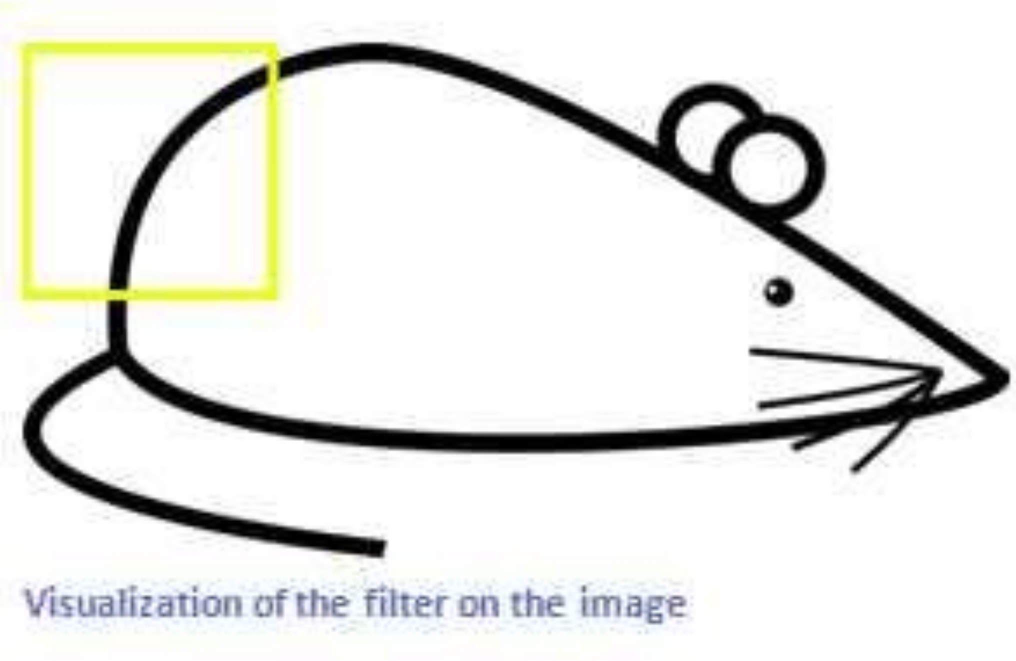
Pixel representation of filter



Visualization of a curve detector filter

滤波器跟卷积神经网络有什么关系?





滤波器跟卷积神经网络有什么关系?

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

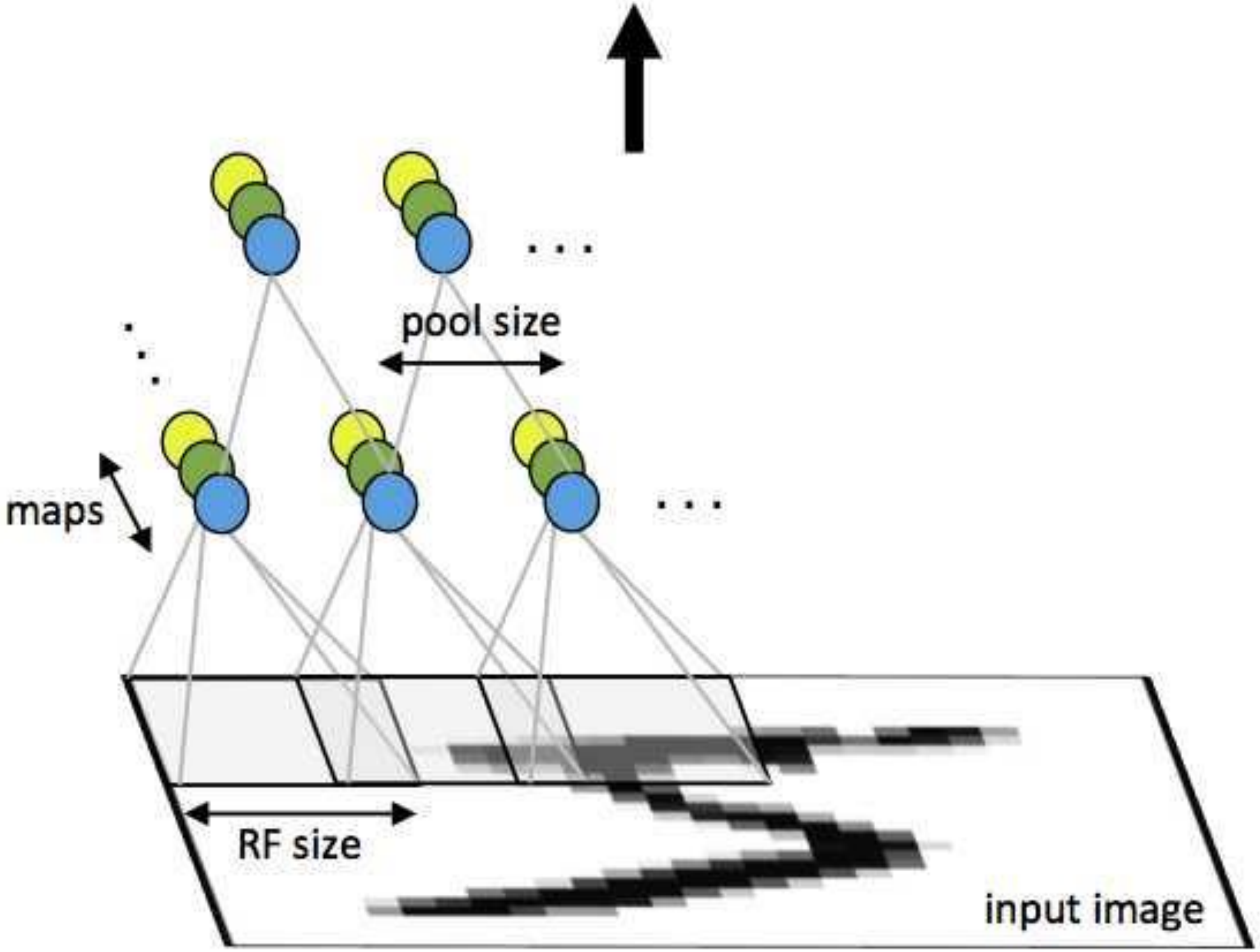
*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

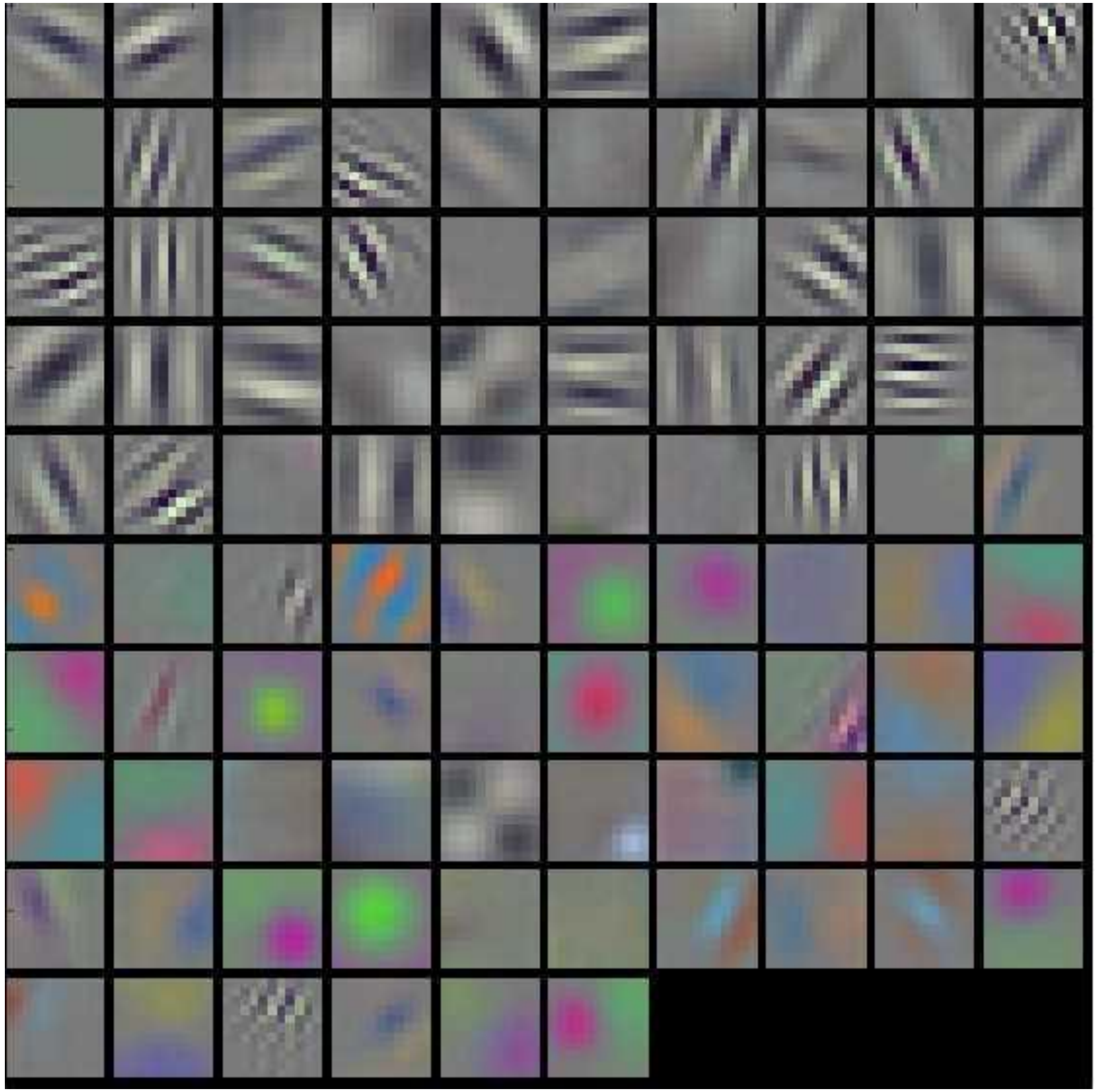
Pixel representation of filter

Multiplication and Summation = 0

滤波器跟卷积神经网络有什么关系?



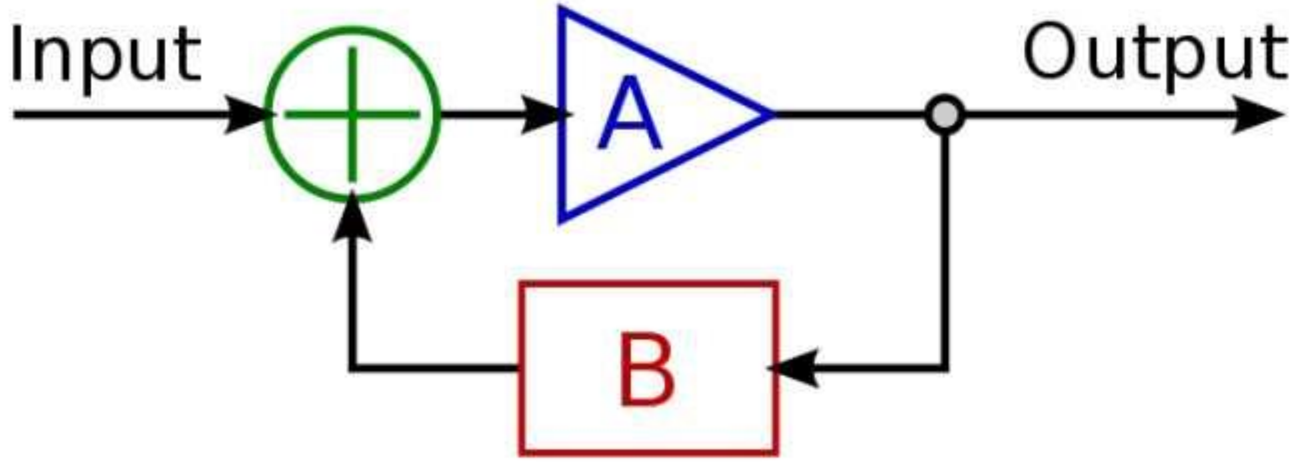
滤波器跟卷积神经网络有什么关系?



- 训练CNN在相当意义上是在训练每一个卷积层的滤波器。
- 让这些滤波器组对特定的模式有高的激活，以达到CNN网络的分类/检测等目的。
- 卷积神经网络的第一个卷积层的滤波器用来检测低阶特征，比如边、角、曲线等。
- 随着卷积层的增加，对应滤波器检测的特征就更加复杂（理性情况下，也是我们想要的情况）

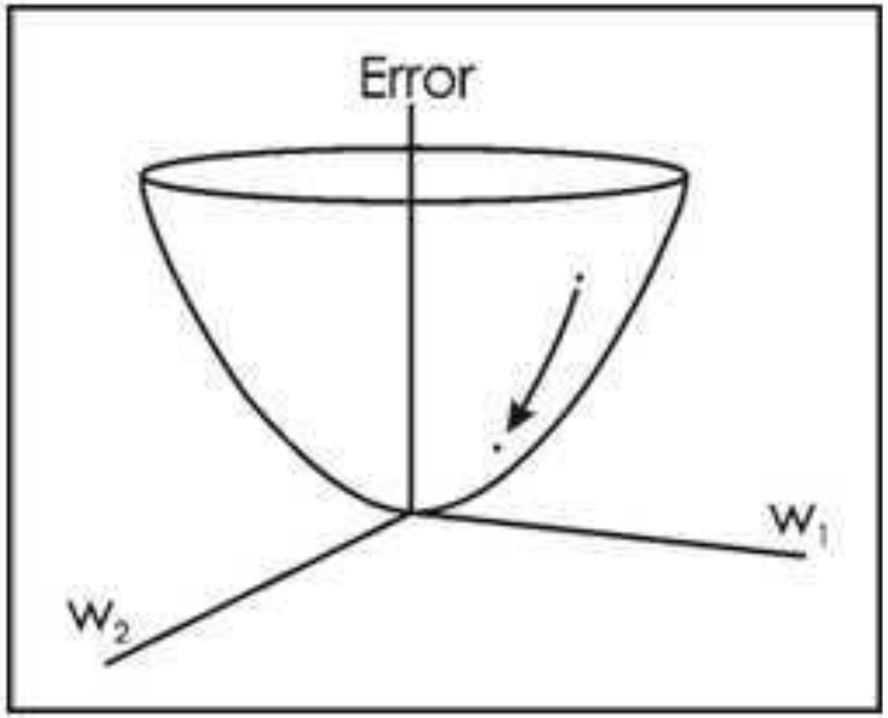
训练滤波器

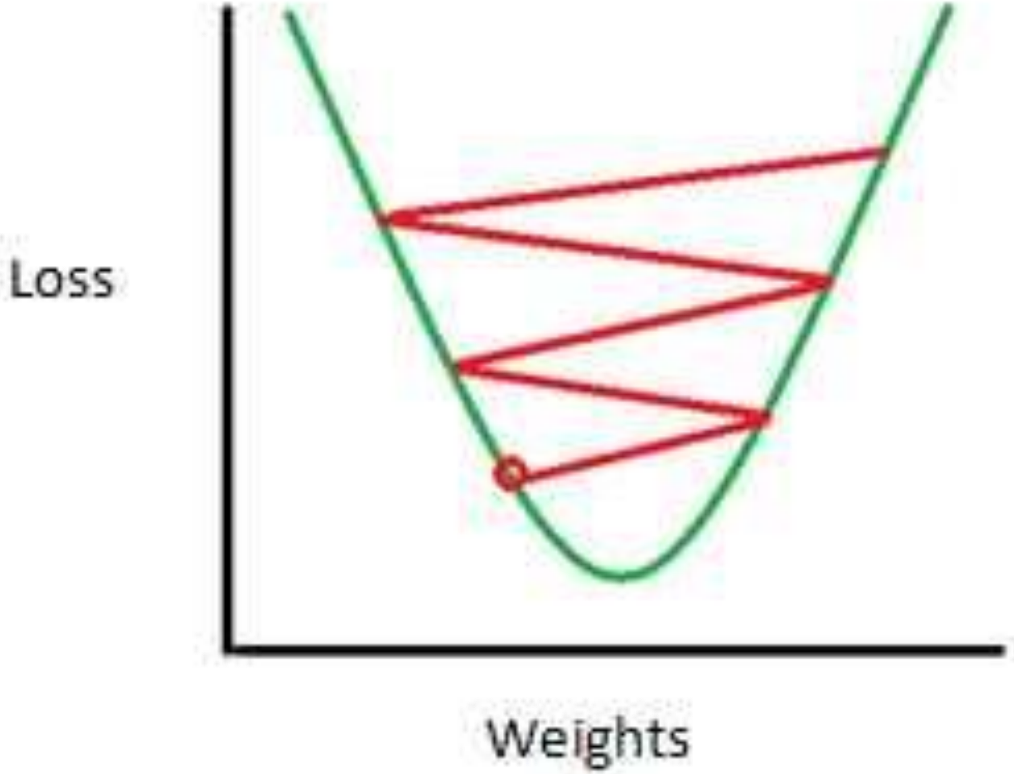
- 构建卷积神经网络的任务就在于构建这些滤波器
- 改变滤波器的值，使其能够识别特定的特征



训练滤波器

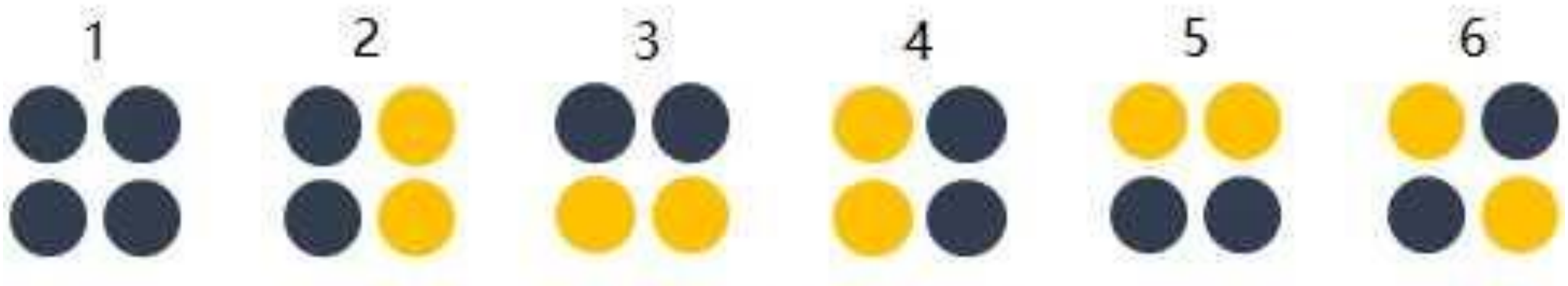
- **定义损失函数，反向传输**





让损失函数的值反馈给整个卷积神经网络，以修改各个滤波器的权重，使得损失值最小

Feature Map 再次卷积



2x2的几个形状的例子

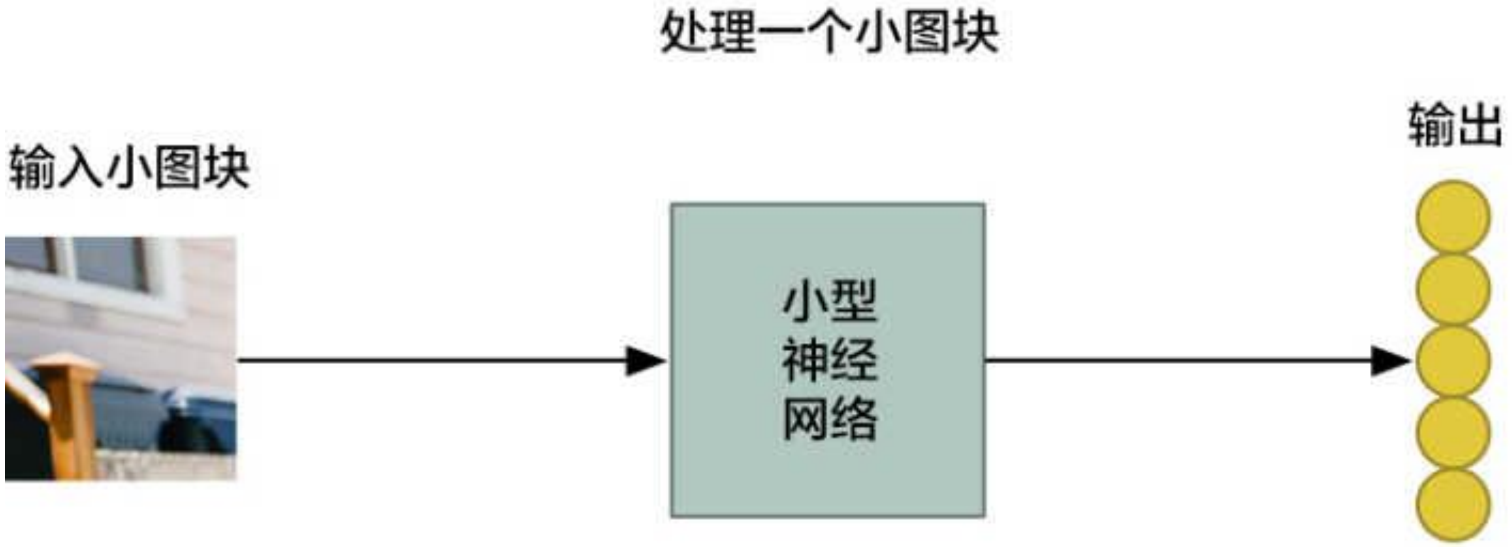
由上面6个形状为“零件”形成的“大”形状

卷积的全过程

- 把图片分解成部分重合的小图块

卷积的全过程

- 把每个小图块输入到小型神经网络中

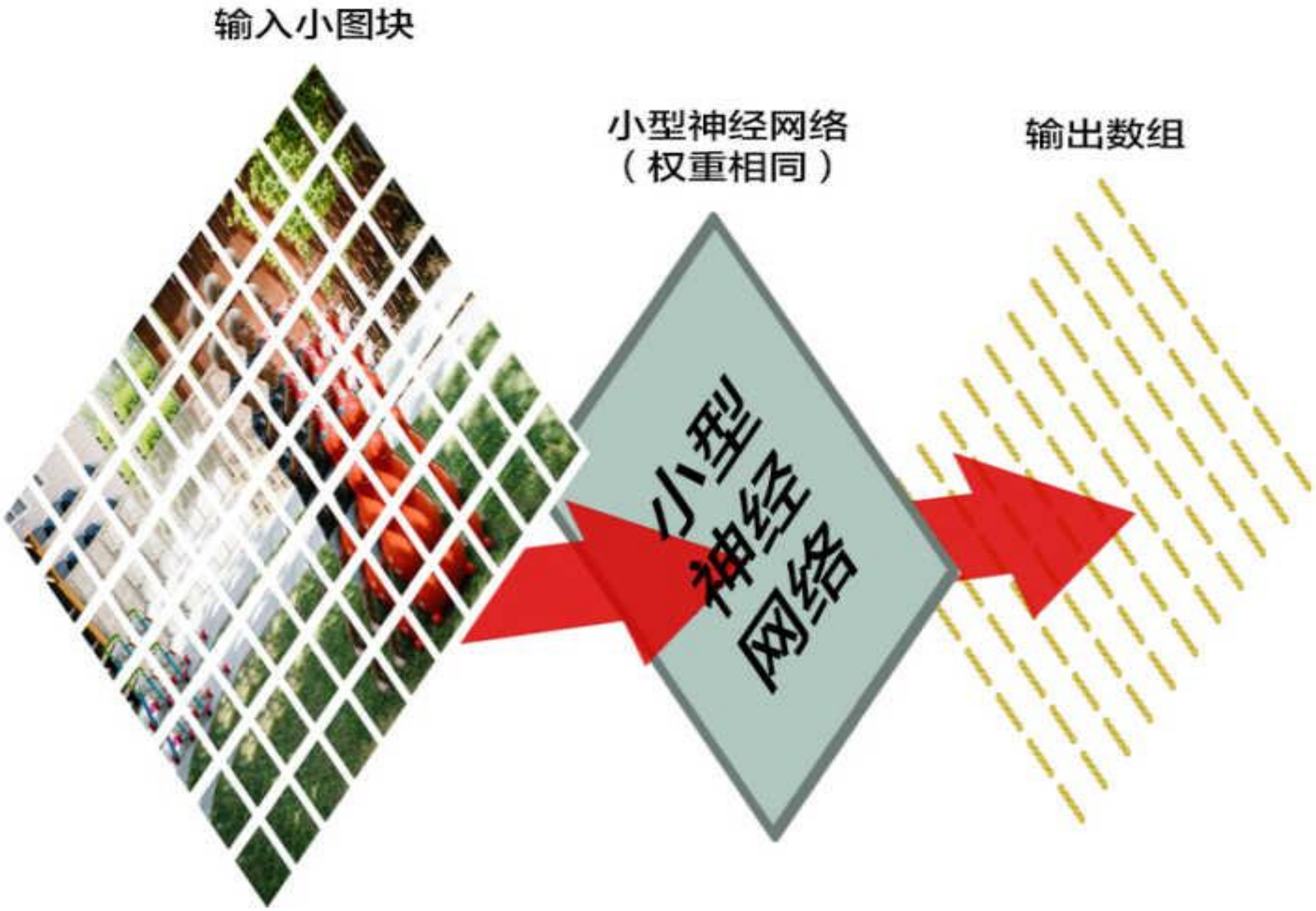


$$y_0 = x_0 * w_1 + x_1 * w_2 + x_4 * w_3 + x_5 * w_4 + b_0$$

$$y_0 = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_4 \\ x_5 \end{bmatrix} + b_0 \qquad (2)$$

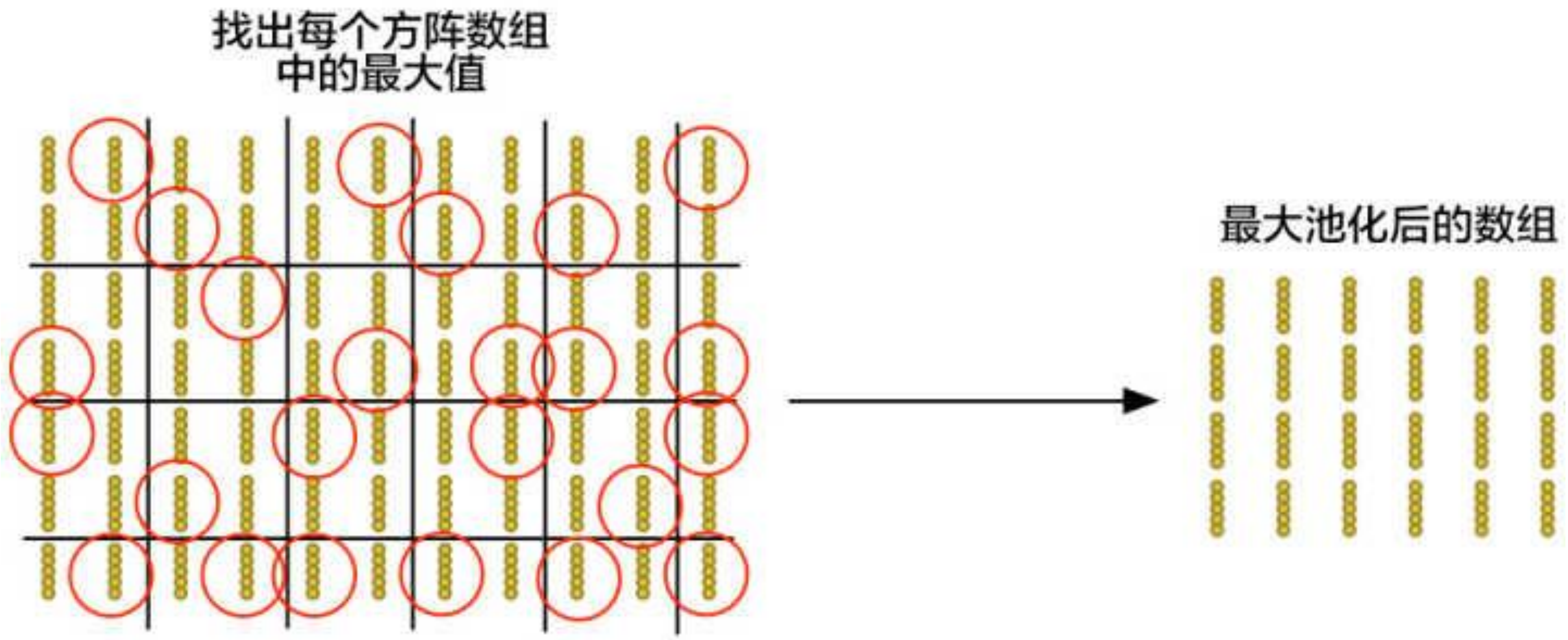
卷积的全过程

- 把每一个小图块的结果都保存到一个新的数组当中



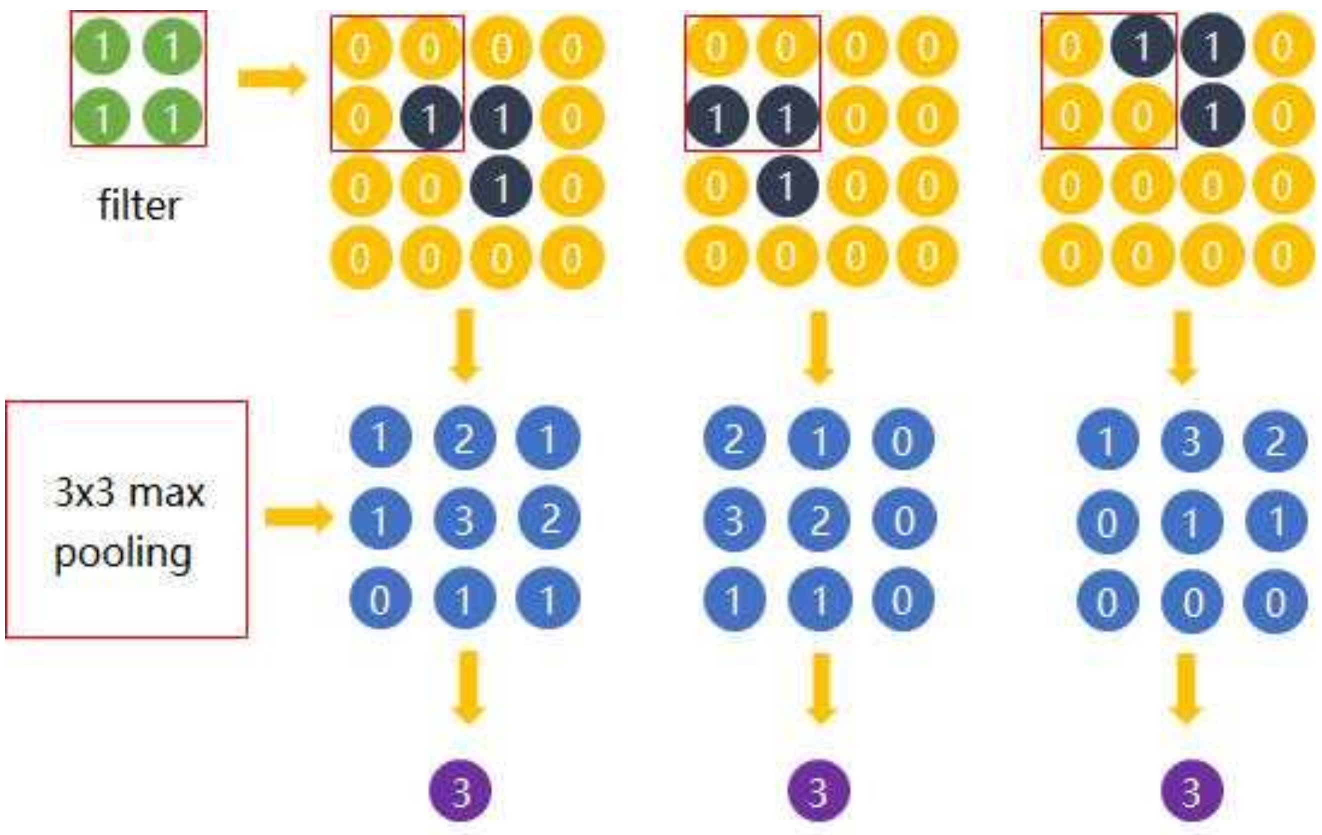
卷积的全过程

- Max pooling 缩减像素采样



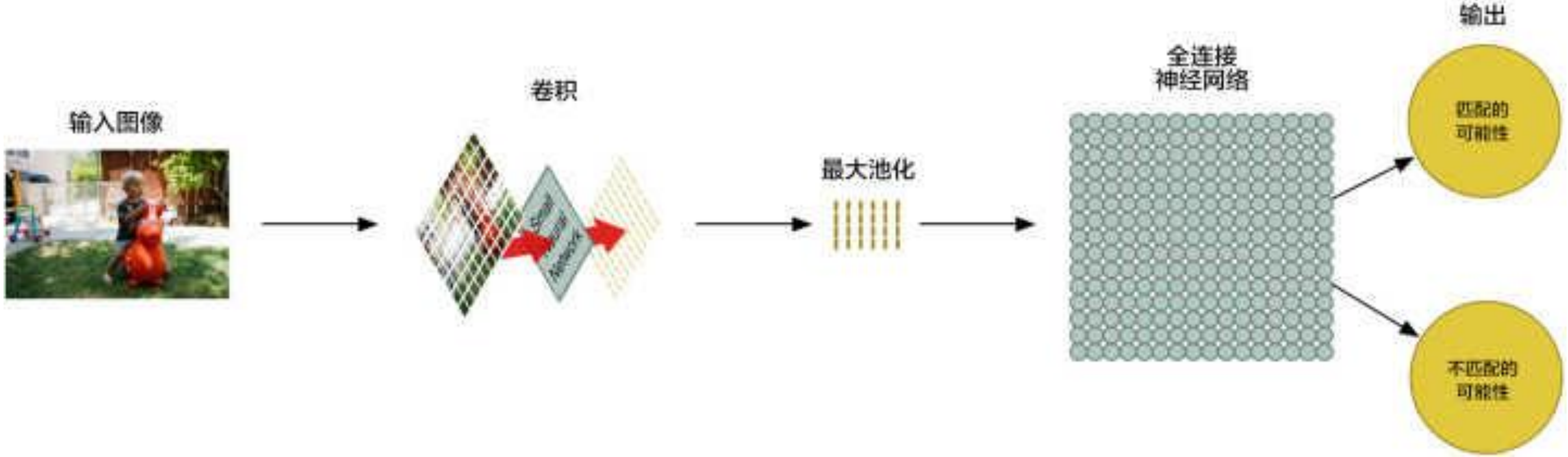
卷积的全过程

- Max Pooling的主要功能是降维，却不会损坏识别结果
- 卷积后的feature map 中有对于识别物体不必要的冗余信息



全连接层（前馈网络）

- **当抓取到足以用来识别图片的特征后，进行分类**



可以把这些步骤任意组合、堆叠多次，来解决真实世界中的问题！

大型神经网络

2012年多伦多大学构造了一个超大型卷积神经网络，有9层，共65万个神经元，6千万个参数。网络的输入是图片，输出是1000个类，比如小虫、美洲豹、救生船等等。

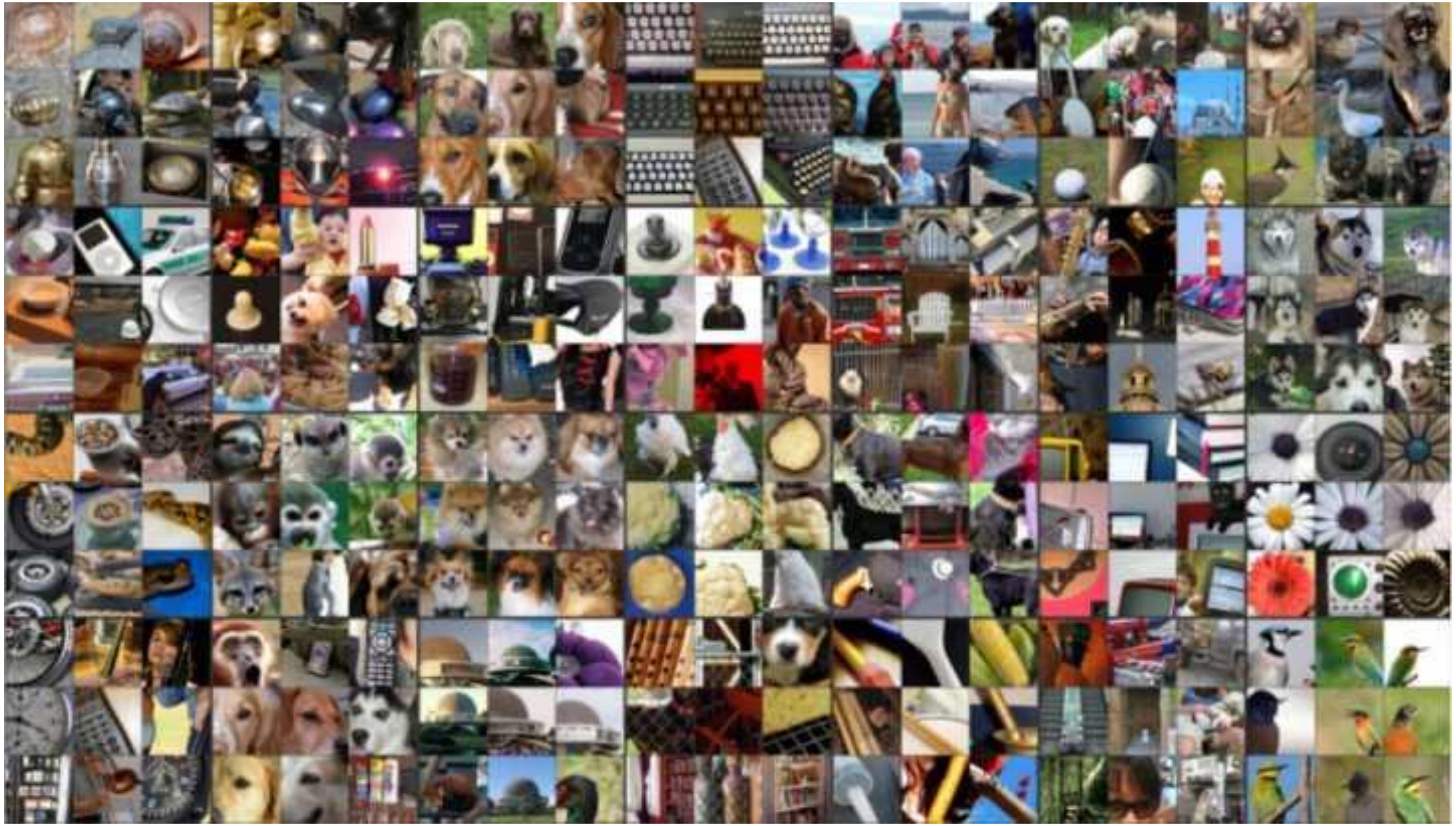
■ 第一层神经元主要负责识别颜色和简单纹理

- 第二层的一些神经元可以识别更加细化的纹理，比如布纹、刻度、叶纹

- 第三层的一些神经元负责感受黑夜里的黄色烛光、鸡蛋黄、高光

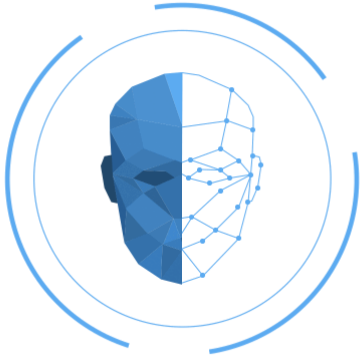
- 第四层的一些神经元负责识别萌狗的脸、七星瓢虫和一堆圆形物体的存在

- 第五层的一些神经元可以识别出花、圆形屋顶、键盘、鸟、黑眼圈动物



Outline

- What is Machine Learning and Why U Should Care?
- Easiest Introduction to Machine Learning
- Deep Learning and CNN
- Face Recognition
- Language Translation and RNN

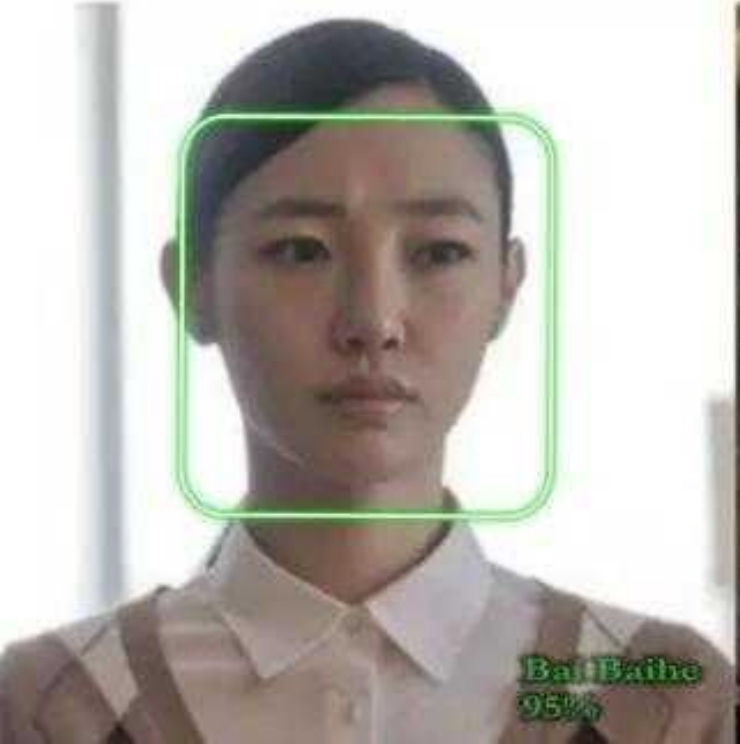


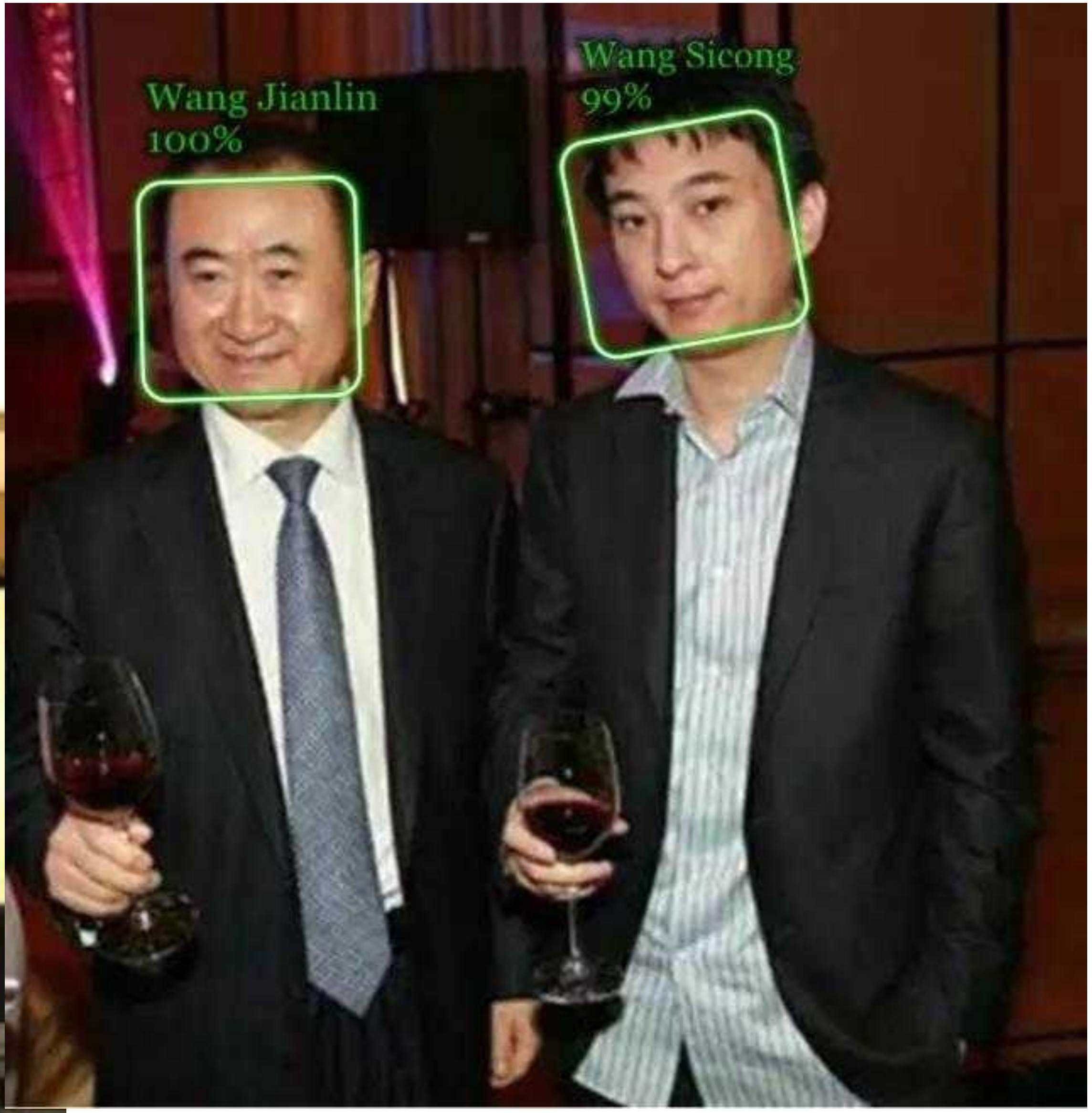
还在为如何有效区分张馨予、张雨琦、张歆艺、张予曦、张艺馨……而惆怅么?

还在好奇为什么刘翔退役之后居然改行做手机评测么?

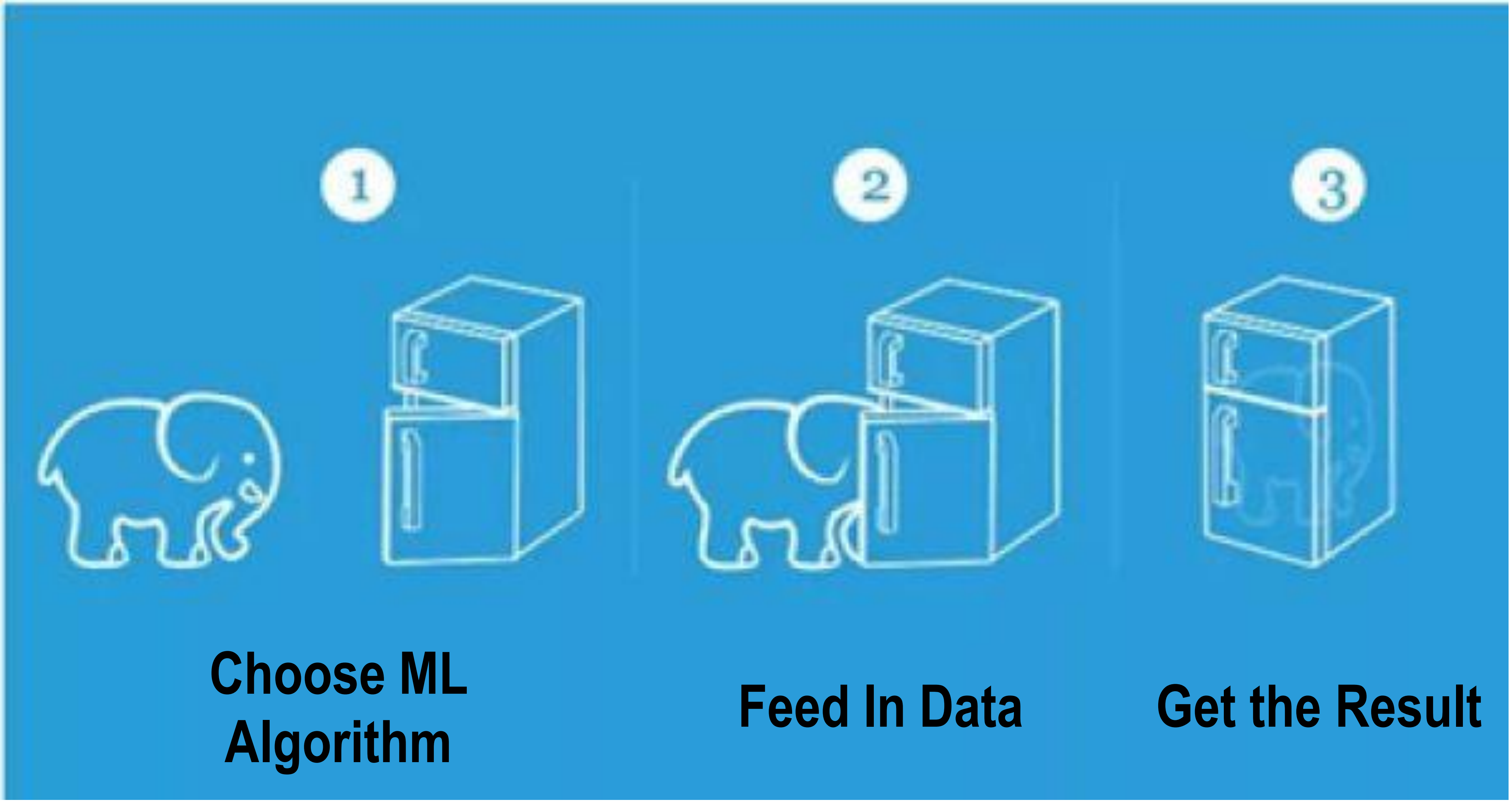
王珞丹、白百何傻傻分不清楚?

Amazon Rekognition





General Steps to Solve Problem Using Machine Learning

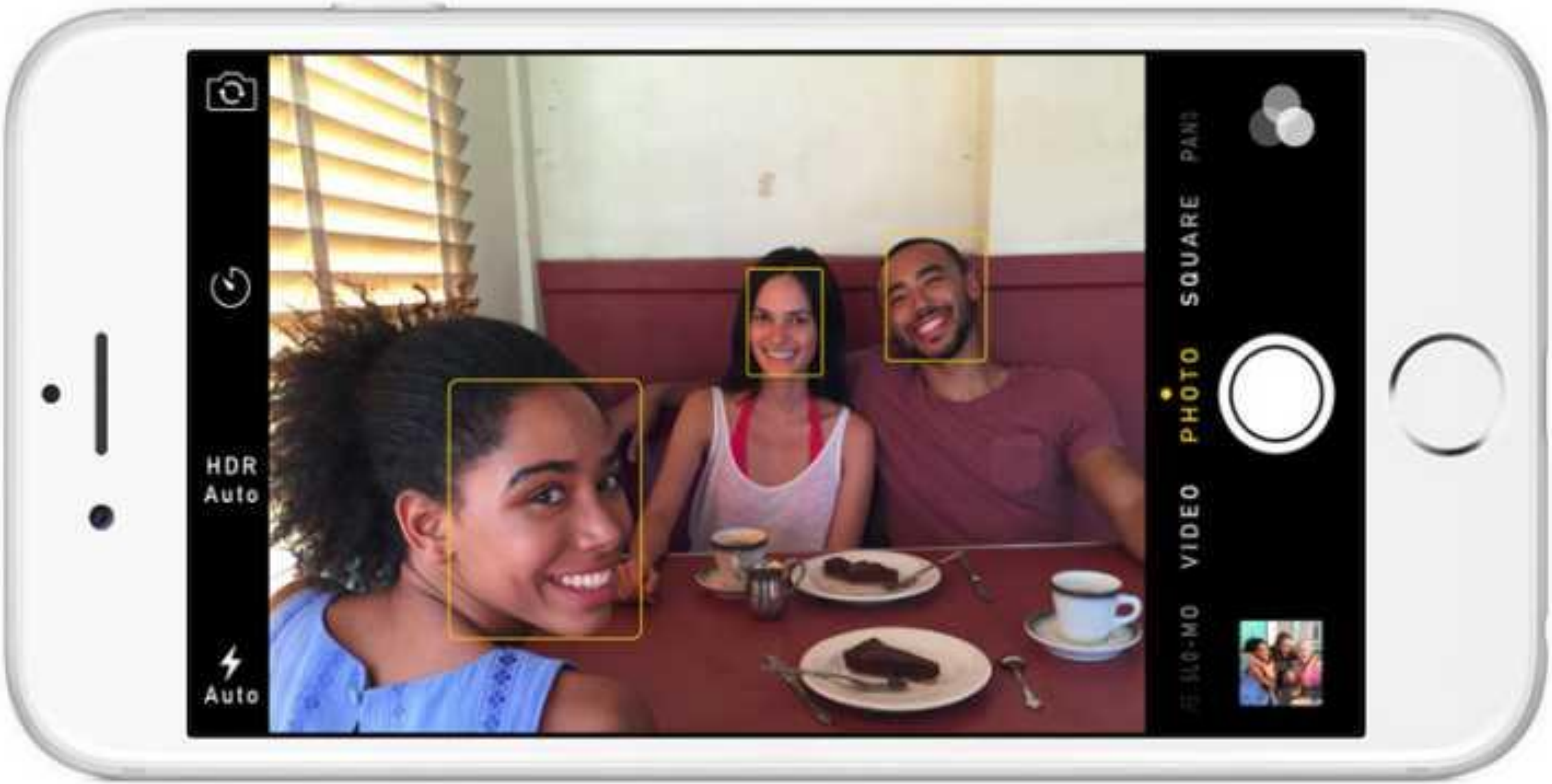


Face Recognition is a series of several related problems

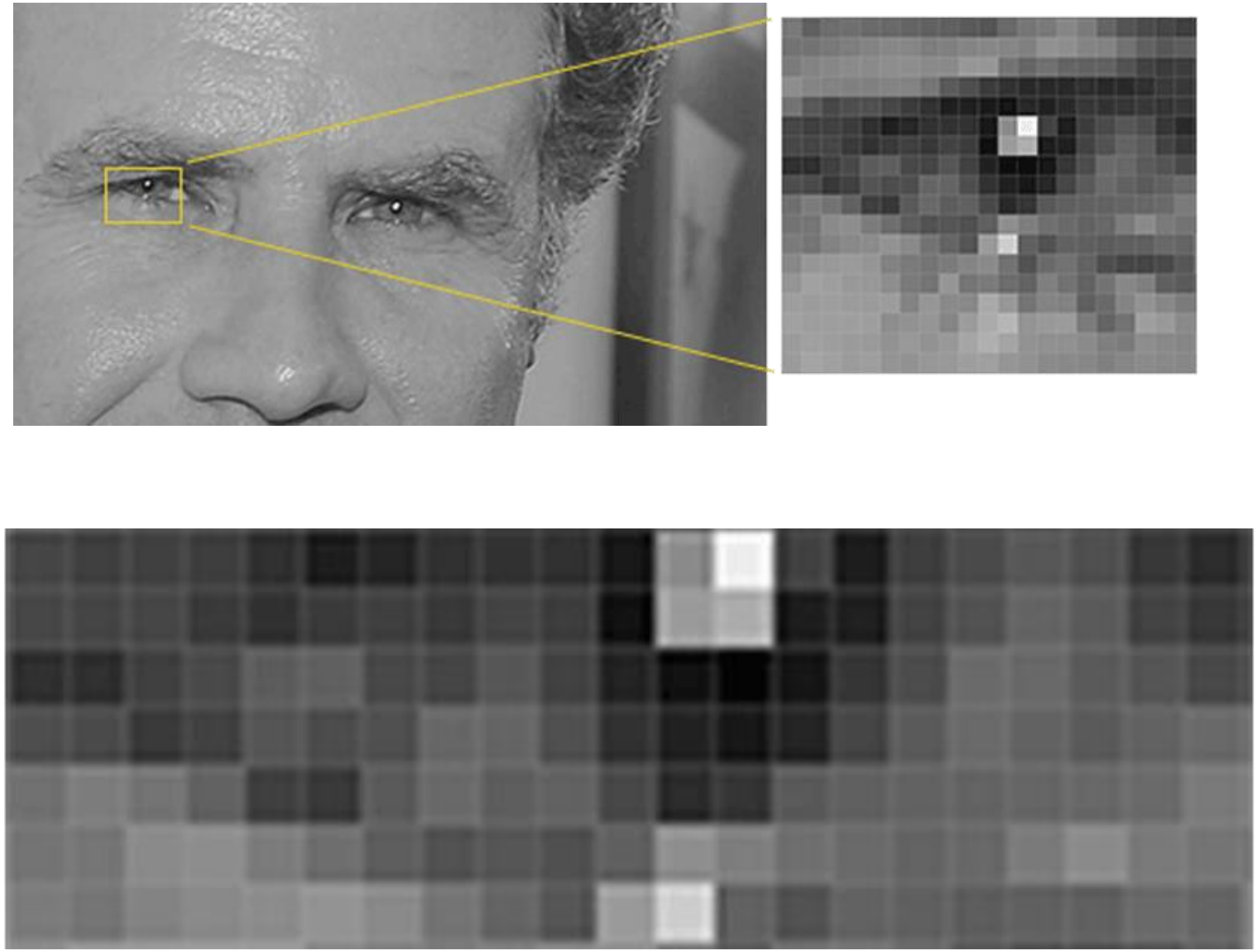
- First, look at a picture and **find all the faces** in it
- Second, **focus on each face** and be able to understand that even if a face is turned in a weird direction or in bad lighting, it is still the same person.
- Third, be able to **pick out unique features** of the face that you can use to tell it apart from other people— like how big the eyes are, how long the face is, etc.
- Finally, **compare the unique features** of that face to all the people you already know to determine the person’s name.

Step 1: Finding all the Faces

Face detection is a great feature for cameras



Histogram of Oriented Gradients Approach（方向梯度直方图）



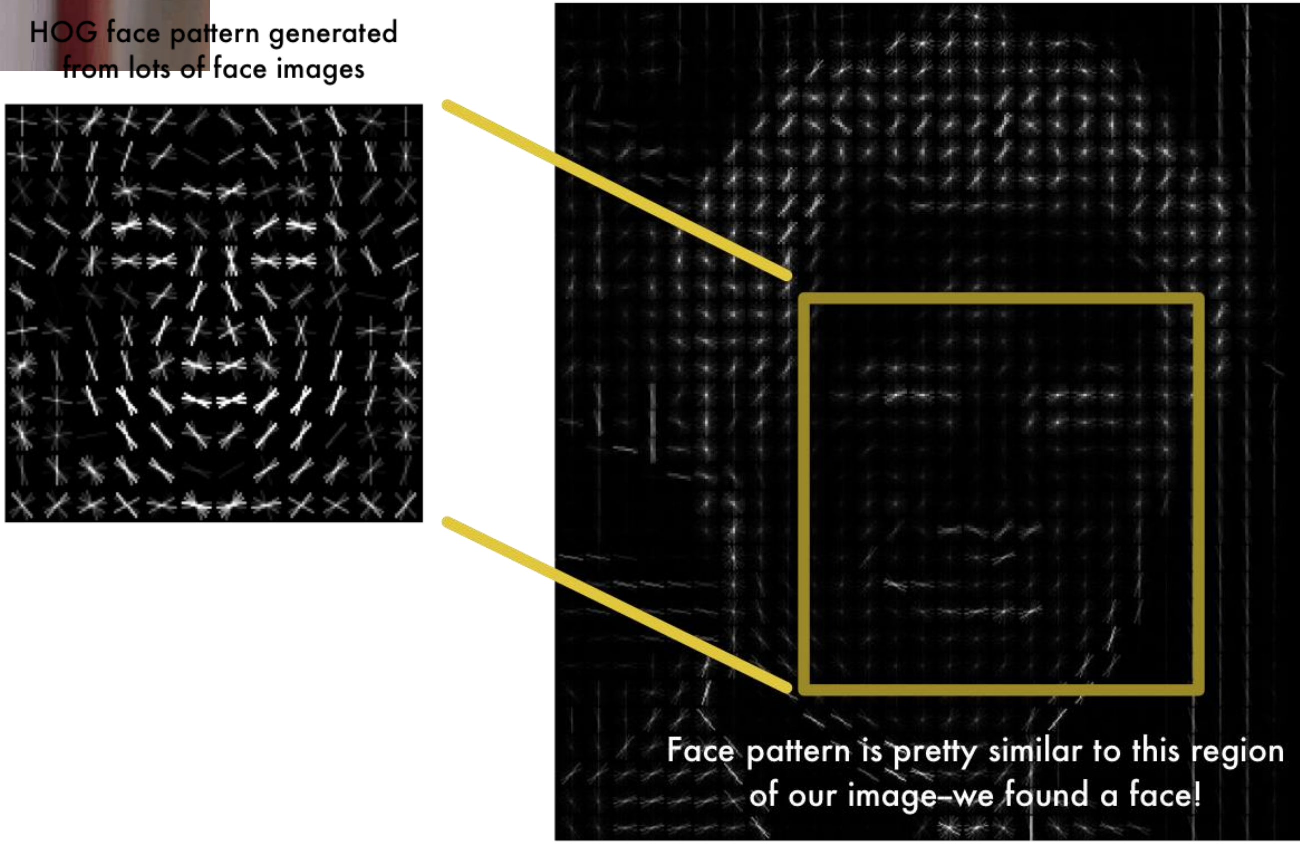
Histogram of Oriented Gradients Approach

- These arrows are called *gradients* and they show the flow from light to dark across the entire image

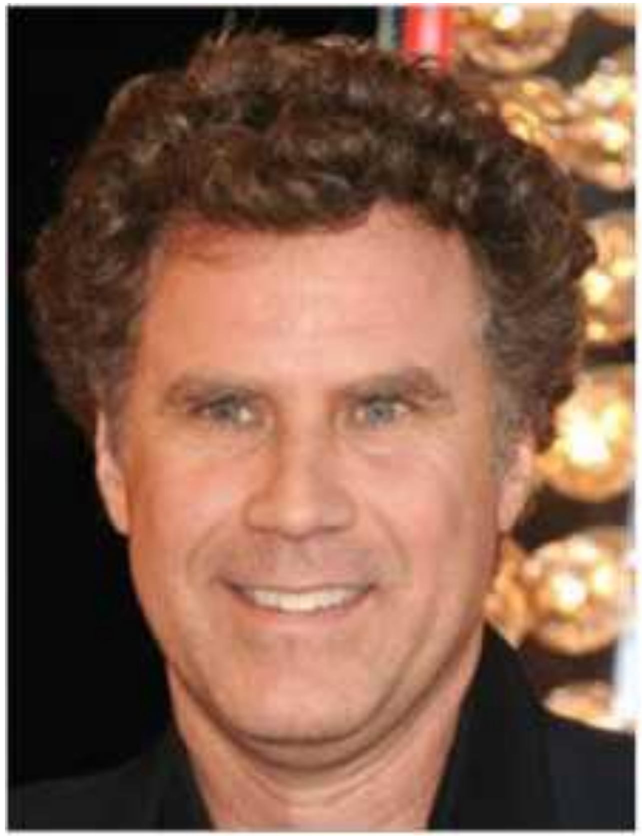
see the basic flow of lightness/darkness at a higher level

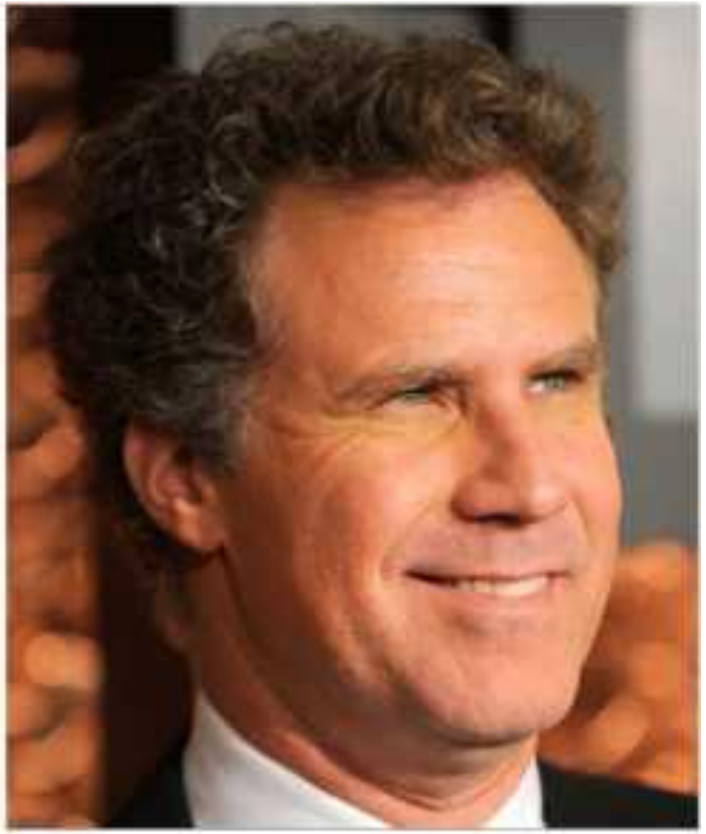
HOG face pattern generated from lots of face images

HOG version of our image



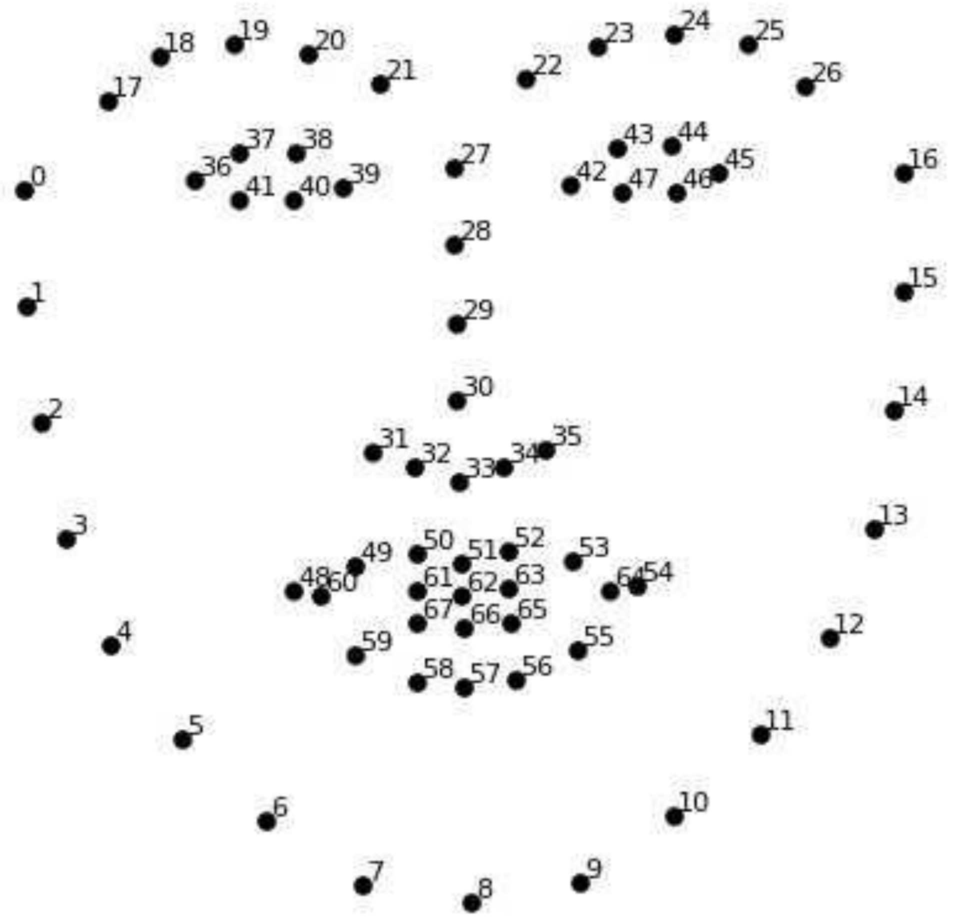
Step2: Posing and Projecting Faces 脸部的不同姿势



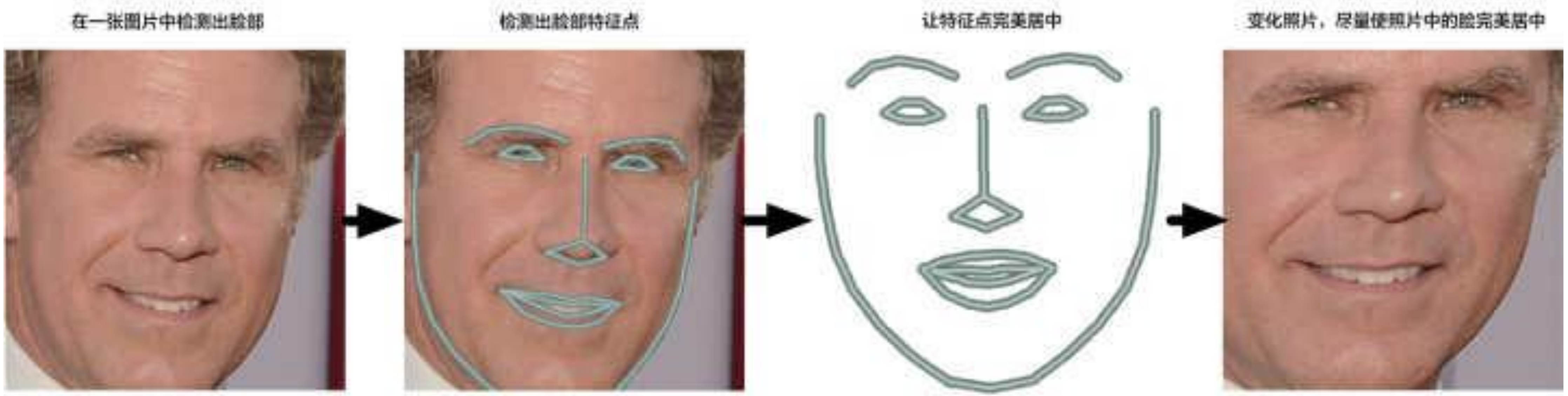


对于电脑来说，面朝不同方向的同一张脸，是不同的东西

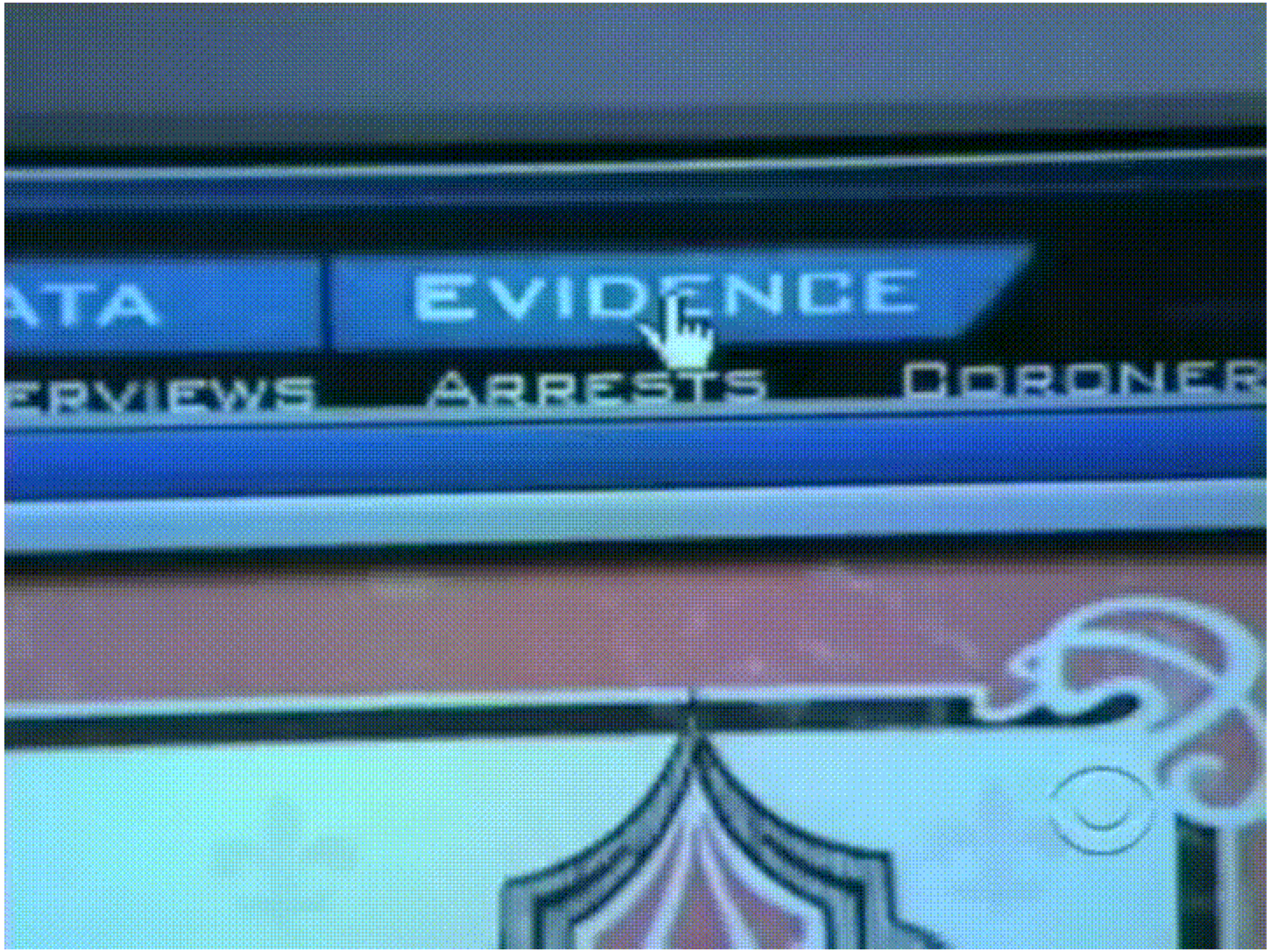
Face landmark estimation （面部特征点估计算法）



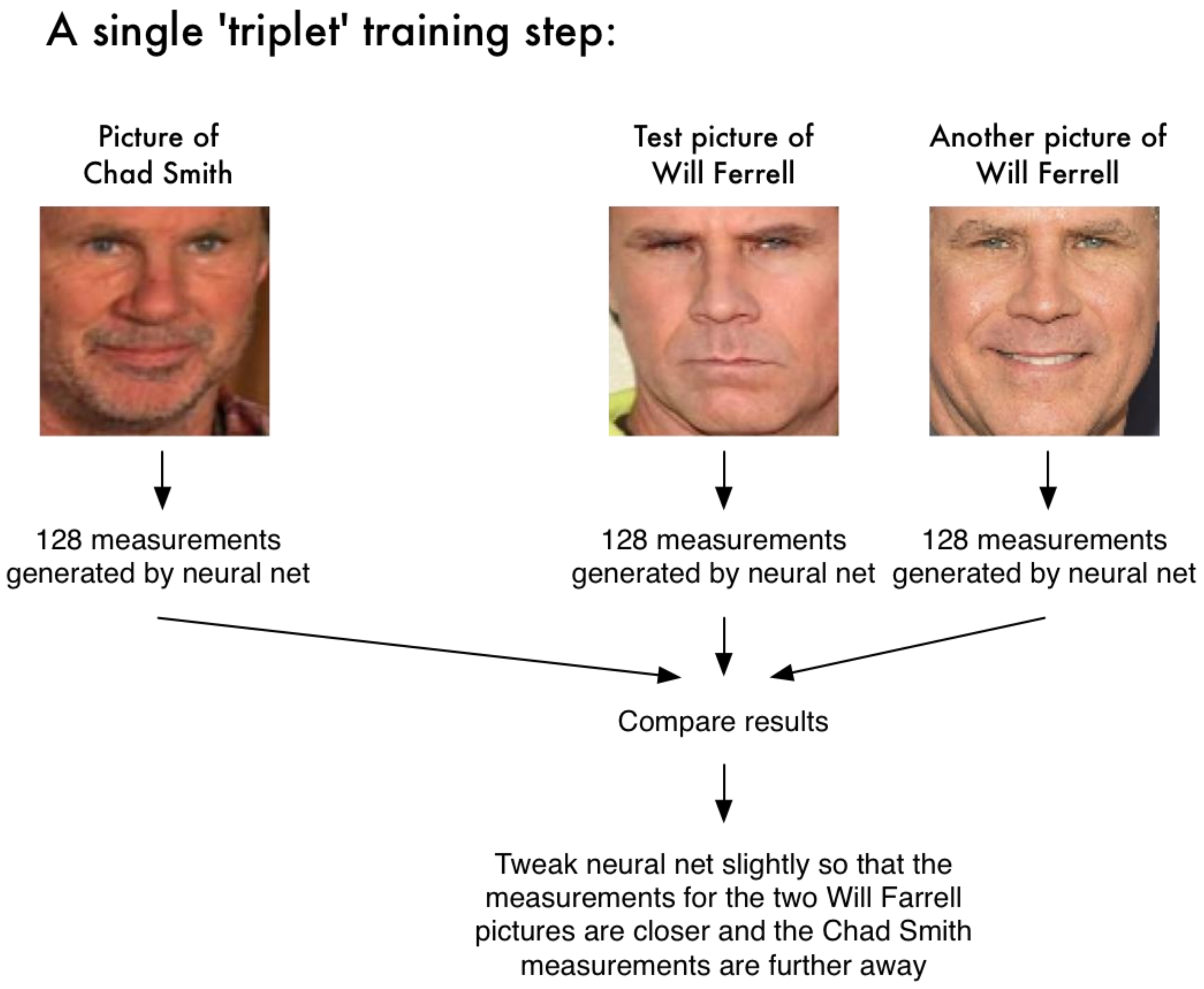
- 算法的基本思路是找到 68 个人脸上普遍存在的特定点
  - 包括下巴的顶部、每只眼睛的外部轮廓、每条眉毛的内部轮廓等。


在一张图片中检测出脸部
检测出脸部特征点
让特征点完美居中
变化照片，尽量使照片中的脸完美居中

Step 3: Encoding Faces



The most reliable way to measure a face

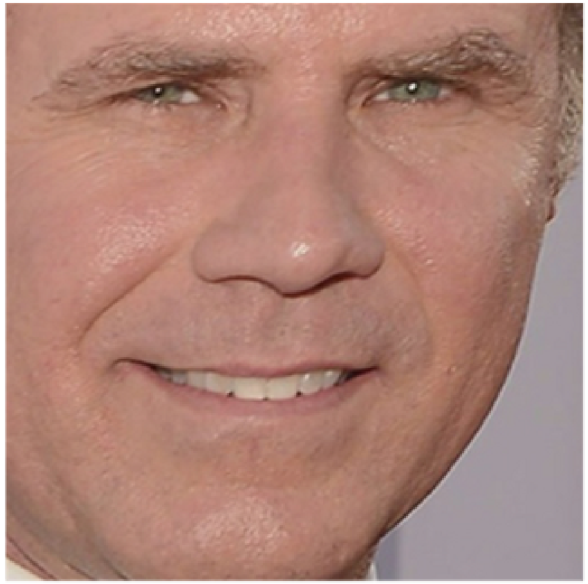


- The training process works by looking at 3 face images at a time:
  - Load a training face image of a known person
  - Load another picture of the same known person
  - Load a picture of a totally different person

Encoding our face image

- This process of training a convolutional neural network to output face embeddings requires a lot of data and computer power.
- But once the network has been trained, it can generate measurements for any face, even ones it has never seen before!

Input Image



128 Measurements Generated from Image

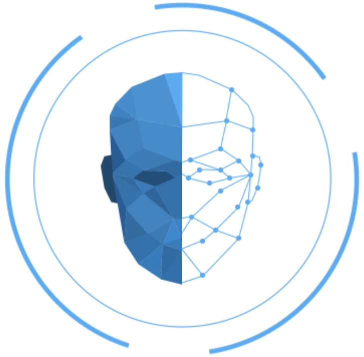
0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.0072777755558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128

Finding the person's name from the encoding

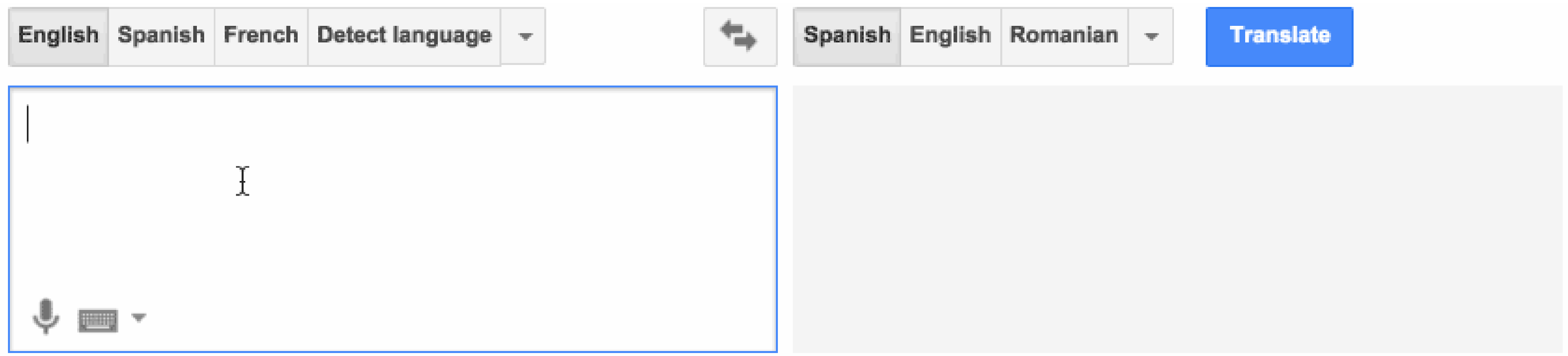
- Using any basic machine learning classification algorithm
- No fancy deep learning tricks are needed. We'll use a simple linear SVM, but lots of classification algorithms could work.

Outline

- What is Machine Learning and Why U Should Care?
- Easiest Introduction to Machine Learning
- Deep Learning and CNN
- Face Recognition
- Language Translation and RNN

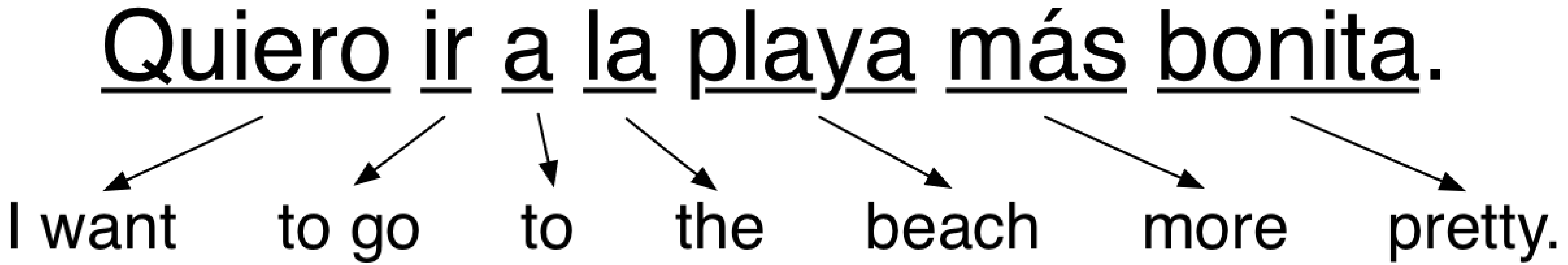


Deep Learning has totally rewritten our approach to machine translation



So how do we program a computer to translate human language?

The simplest approach is to replace every word in a sentence with the translated word in the target language



But the results are bad because it ignores grammar and context

Adding language-specific rules to improve the results

Unfortunately this only worked for simple, plainly-structured documents like weather reports

Making Computers Translate Better Using Statistics

- 建造一个基于统计数据的翻译系统需要大量的训练数据，其中完全相同的文本被翻译成至少两种语言。这种双重翻译的文本称为**平行语料库。**

幸运的是，已经有许多文本被同时翻译为两种语言。例如，欧洲议会将其诉讼程序翻译成了 21 种语言

英语	西班牙语
Resumption of the session I declare resumed the session of the European Parliament adjourned on Friday 17 December 1999, and I would like once again to wish you a happy new year in the hope that you enjoyed a pleasant festive period.	Reanudación del período de sesiones Declaro reanudado el período de sesiones del Parlamento Europeo, interrumpido el viernes 17 de diciembre pasado, y reitero a Sus Señorías mi deseo de que hayan tenido unas buenas vacaciones.

Thinking in Probabilities （统计翻译系统）

- statistical translation systems don’t try to generate one exact translation
- **generate thousands of possible translations**
- then they **rank those translations** by likely each is to be correct

- **Step1: Break original sentence into chunks**

Quiero ir a la playa más bonita.

Step 2: Find all possible translations for each chunk

- finding all the ways humans have translated those same chunks of words in our training data

Quiero ir a la playa más bonita.

Quiero	ir	a	la playa	más bonita
- I want	- to go	- to	- the beach	- more pretty
- I love	- to work	- at	- the seaside	- most pretty
- I like	- to run	- per	- the open space	- more lovely
- I try	- to appear			- most lovely
- I mean	- to be on			- more tidy
	- to be			- most tidy
	- to leave			
	- to pass away			
	- to forget			

Step 3: Generate all possible sentences and find the most likely one

- 仅仅把第二步中我们列出的词块译法组合起来，我们就可以生成将近 2500 句不同的句子。下面是一些例子：

I love | to leave | at | the seaside | more tidy.

I mean | to be on | to | the open space | most lovely.

I like | to be |on | per the seaside | more lovely.

I mean | to go | to | the open space | most tidy.

真实世界中，因为有不同的语序和词块分解方法，所以有更多可能的词块组合：

I try | to run | at | the prettiest | open space.

I want | to run | per | the more tidy | open space.

I mean | to forget | at | the tidiest | beach.

I try | to go | per | the more tidy | seaside.

扫描所有句子，找到那句听起来「最像人话」的

- 将每个句子与来自英语书籍和新闻故事的数百万个真实句子进行比较

I try | to leave | per | the most lovely | open space.

很可能没有人用英语写过这样的句子, 则会给这个译法设定一个低概率得分

I want | to go | to | the prettiest | beach.

这个句子和训练集中的句子很类似，所以它将获得一个高概率的得分

在尝试过所有可能的句子之后，我们会选择那个既包含了最有可能的词块译法，又与真实英语表达最相似的句子

Statistical Machine Translation was a Huge Milestone

- Statistical machine translation systems perform much better than rule-based systems if you give them enough training data.

"Every time I fire a linguist, my accuracy goes up."

— *Frederick Jelinek*

However…

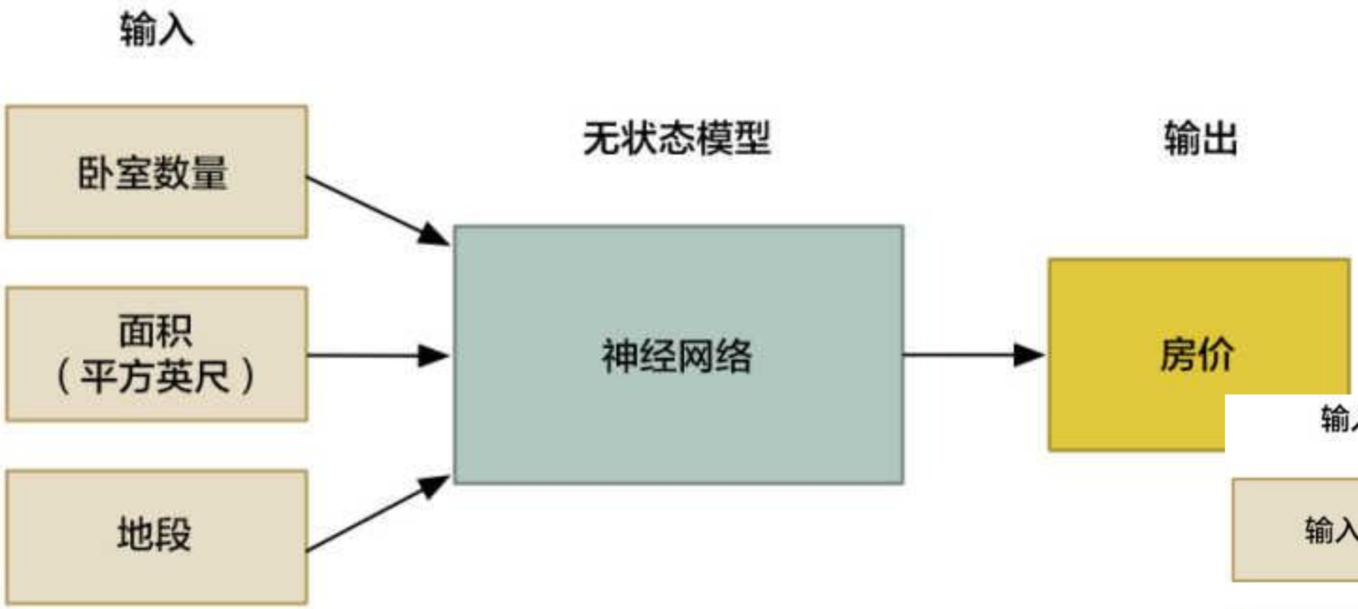
- Statistical machine translation systems work well, but they are complicated to build and maintain
- Every new pair of languages you want to translate requires experts to tweak and tune a new multi-step translation pipeline.

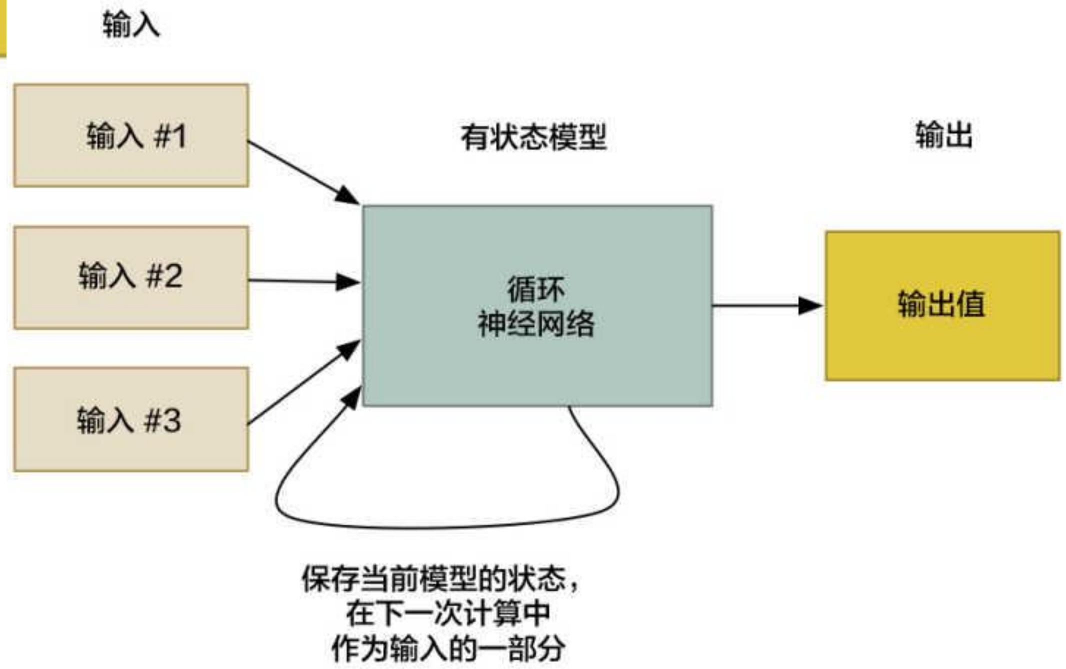
Making Computers Translate Better — Without all those Expensive People

- machine translation is a black box system that learns how to translate by itself— just by looking at training data

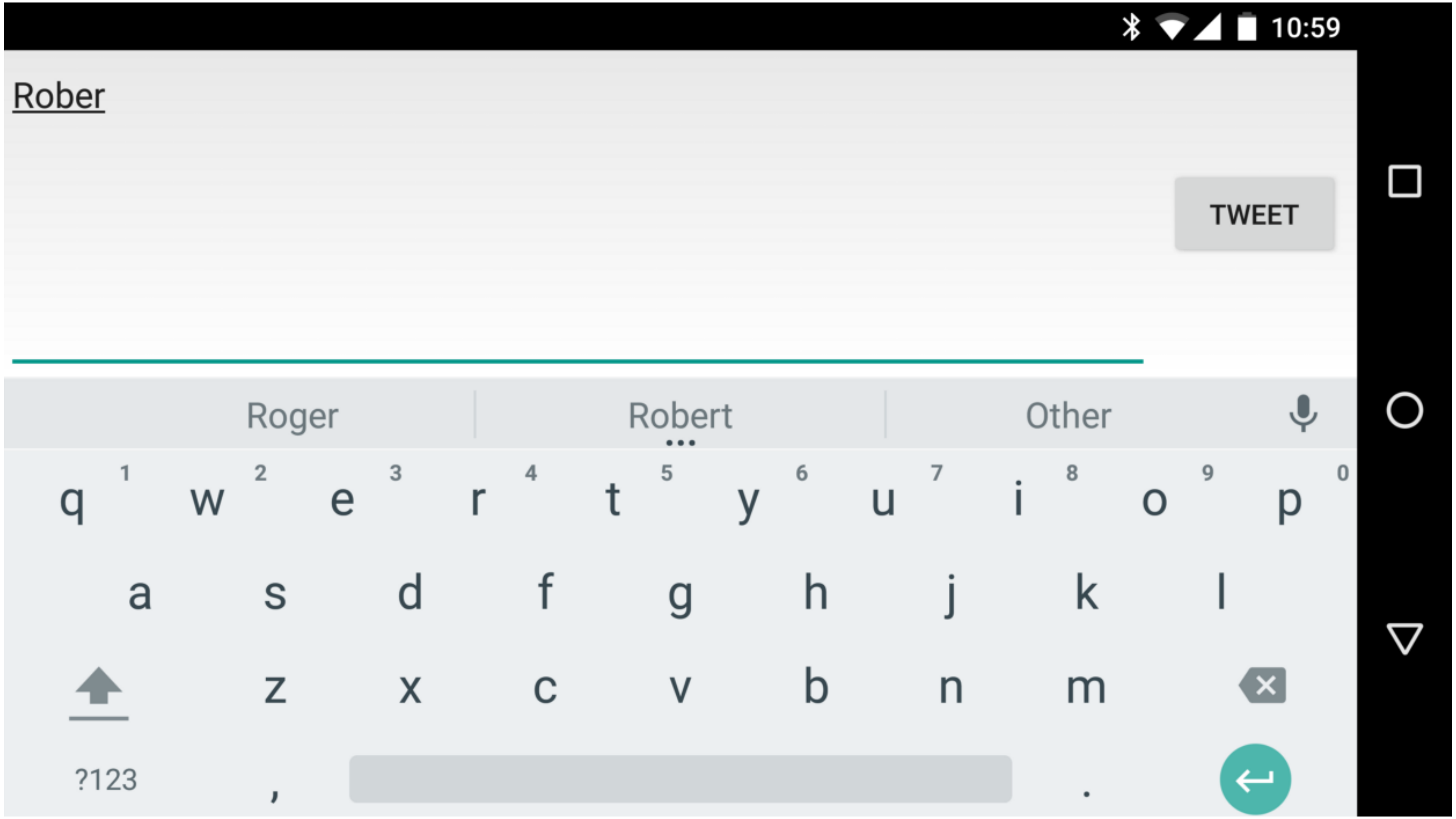
Two big ideas make this possible—*recurrent neural networks* *（循环神经网络）* and *encodings* *（编码）*. By combining these two ideas in a clever way, we can build a self-learning translation system

Recurrent Neural Networks





One cool use might be auto-predict for a mobile phone keyboard



自动生成故事。。。

- http://karpathy.github.io/2015/05/21/rnn-effectiveness/
- https://github.com/karpathy/char-rnn

Encodings

- 把面部特征转换为一系列测量值的想法就是编码的例子
  - 获取原始数据（面部图片），并将其转换为了代表这张脸的一系列测量值（编码）
- 可以使用神经网络，让它自动从面部生成测量值。找出哪些测量值能够区分两个相似的人，计算机在这方面比我们做得更好

我们可以用句子做同样的事情!

- 我们可以把任何一个句子表达成一系列独特的编码：

输入句子

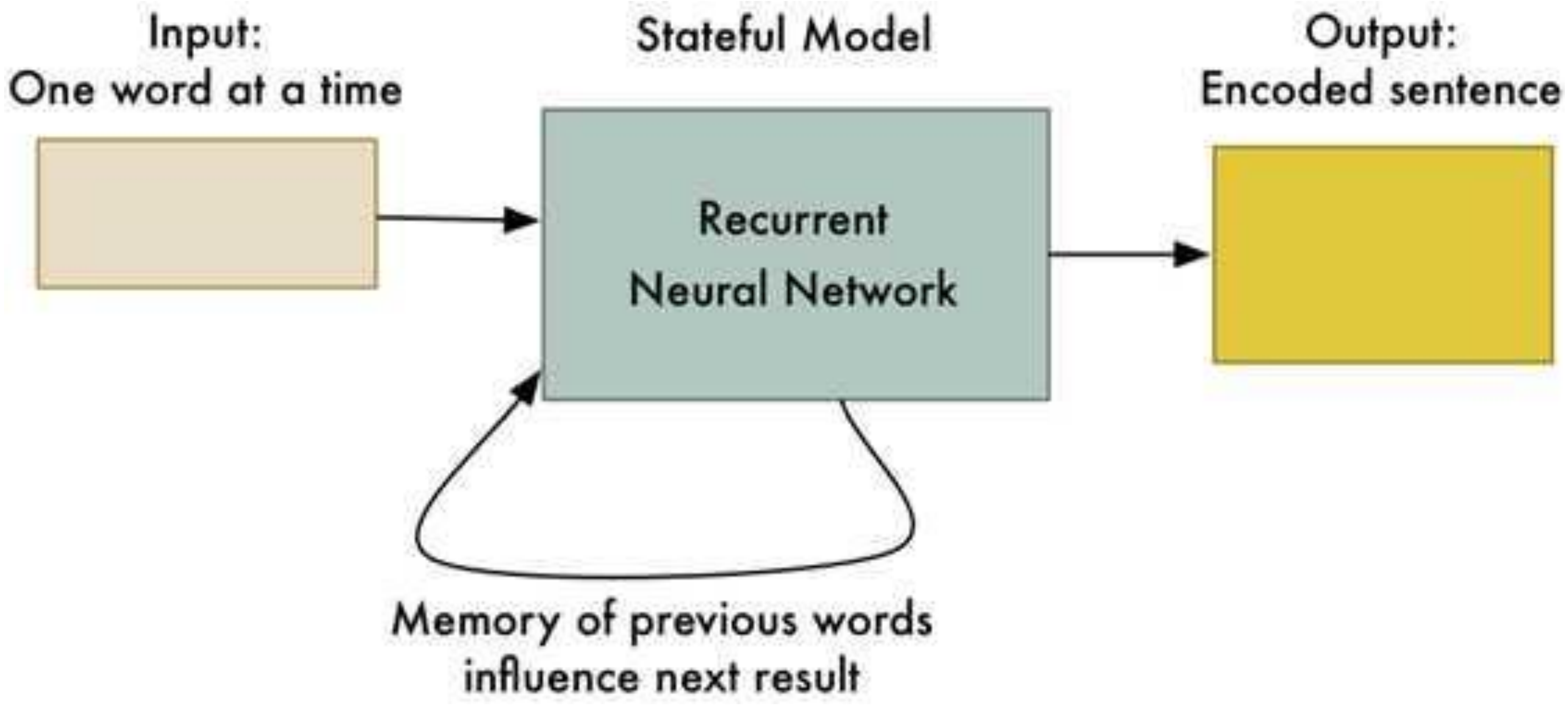
"Machine Learning is Fun!" →

从句子中生成的测量值

0.0974960848689O8	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030609439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.0072777755558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311582777	-0.040977258235216	-0.082041338086128

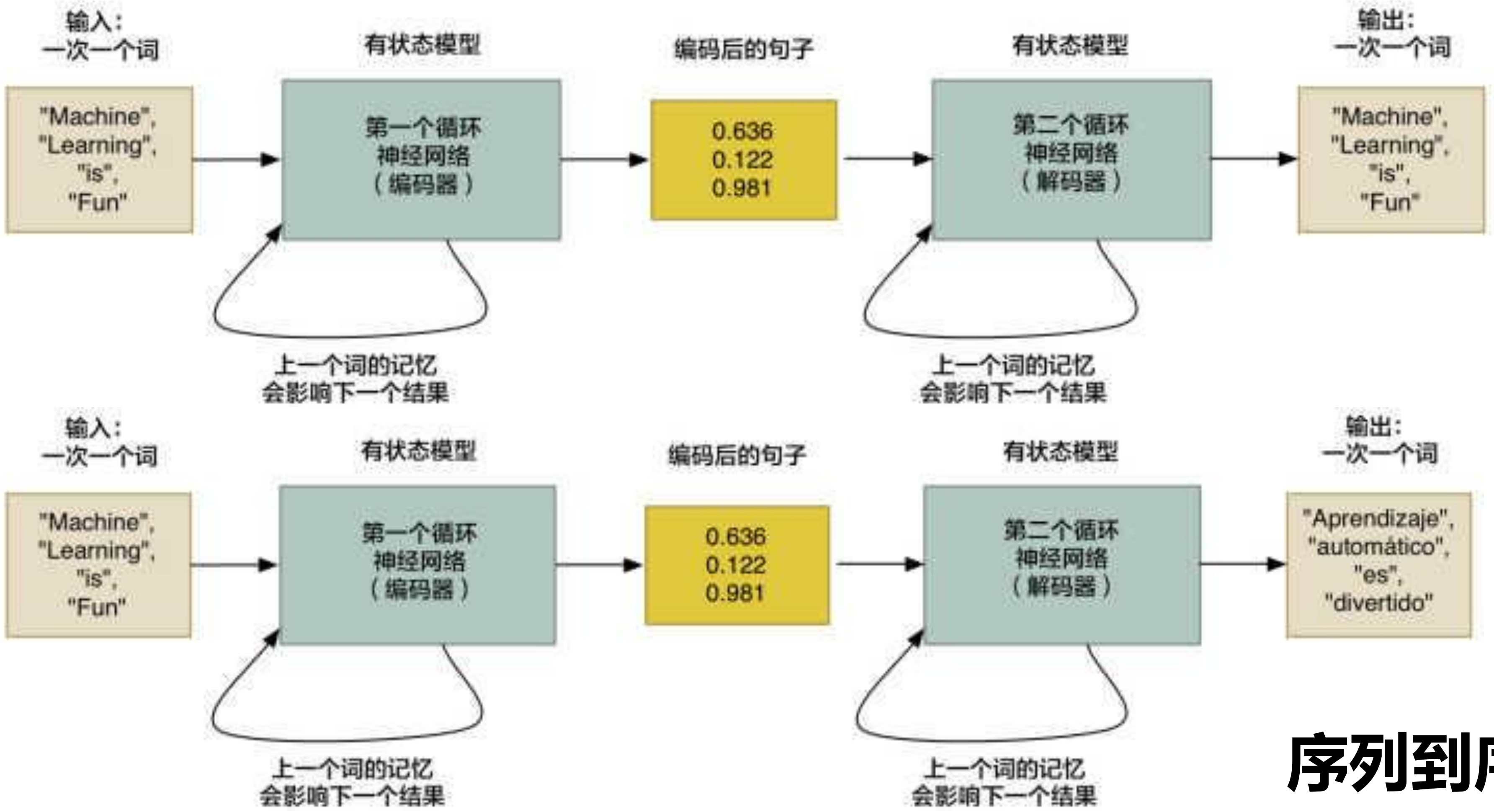
对句子编码

- 将句子输入到 RNN 中，一次一个词
- 最后一个词处理之后的最终结果，就将是表示整个句子的数值



How does that help us?

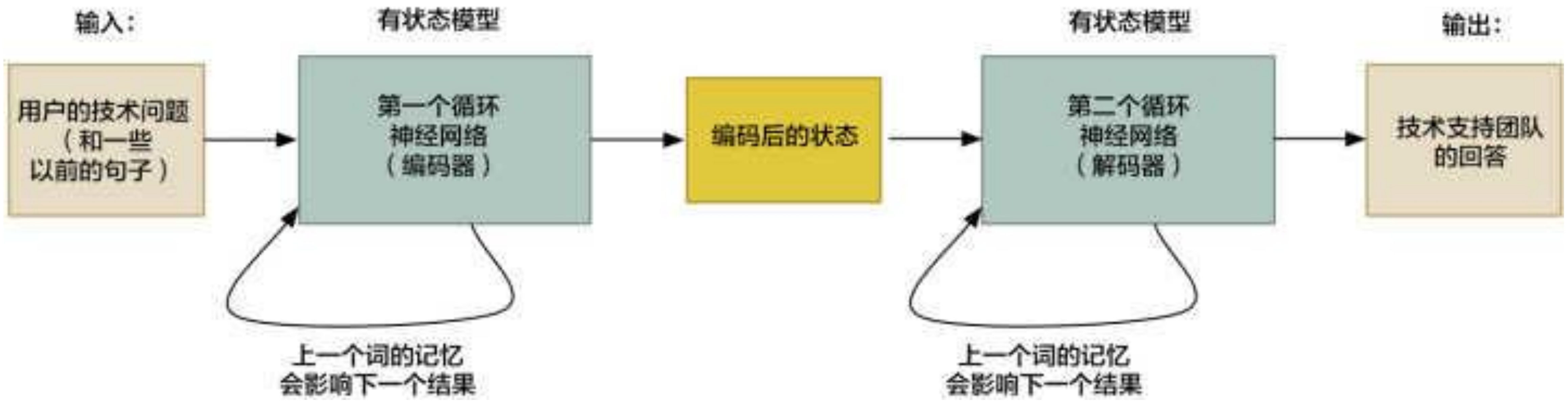
- We use a RNN to encode a sentence into a set of unique numbers



序列到序列模型

序列到序列模型的无穷力量

- 可以使用序列到序列模型来建造 AI 机器人



【教程】从零开始动手实现微信聊天机器人

http://www.bilibili.com/video/av16505671/

Further Reading

- **Andrew Ng's Machine Learning Course**

  https://www.coursera.org/learn/machine-learning

- **机器学习 周志华**