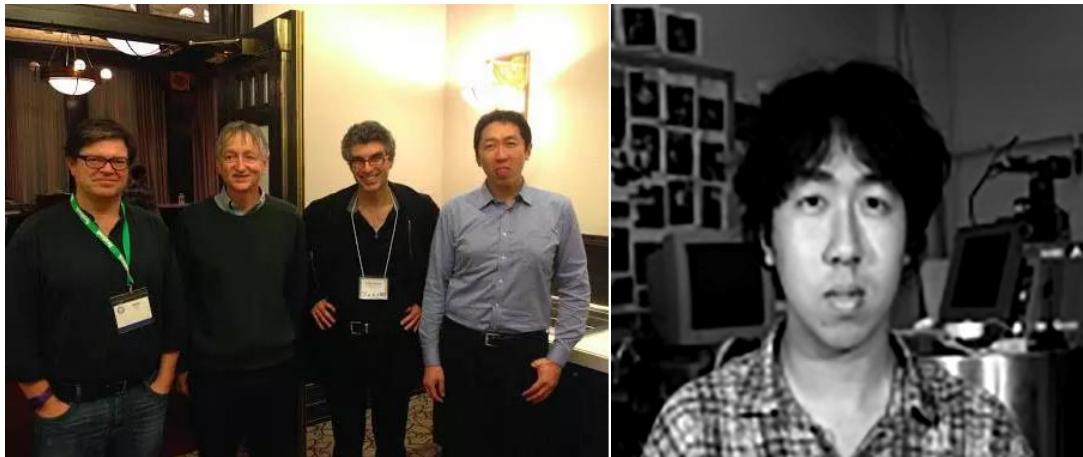


生物统计学： 生物信息中的概率统计模型

2020年秋



Yann LeCun
Geoffrey Hinton
Yoshua Bengio
Andrew Ng

有关信息

- 授课教师：宁康
 - Email: ningkang@hust.edu.cn
 - Office: 华中科技大学东十一楼504室
 - Phone: 87793041, 18627968927
- 课程网页
 - <http://www.microbioinformatics.org/teach/#>
 - QQ群: 182996651



2020生物统计学



扫一扫二维码，加入群聊。



课程安排

- 生物背景和课程简介
- 传统生物统计学及其应用
- 生物统计学和生物大数据挖掘
 - Hidden Markov Model (HMM)及其应用
 - Markov Chain
 - HMM理论
 - HMM和基因识别 (Topic I)
 - HMM和序列比对 (Topic II)
 - 进化树的概率模型 (Topic III)
 - Motif finding中的概率模型 (Topic IV)
 - EM algorithm
 - Markov Chain Monte Carlo (MCMC)
 - 基因表达数据分析 (Topic V)
 - 聚类分析-Mixture model
 - Classification-Lasso Based variable selection
 - 基因网络推断 (Topic VI)
 - Bayesian网络
 - Gaussian Graphical Model
 - 基因网络分析 (Topic VII)
 - Network clustering
 - Network Motif
 - Markov random field (MRF)
 - Dimension reduction及其应用 (Topic VIII)
- 面向生物大数据挖掘的深度学习

研究对象：
生物序列，
进化树，
生物网络，
基因表达
...

方法：
生物计算与生物统计

第10章：面向生物大数据挖掘的深度学习

- 深度学习是什么
- 深度学习的基本方法（以及神经网络算法）
- 生物大数据的深度学习方法
- 生物大数据深度学习的应用案例

Part I

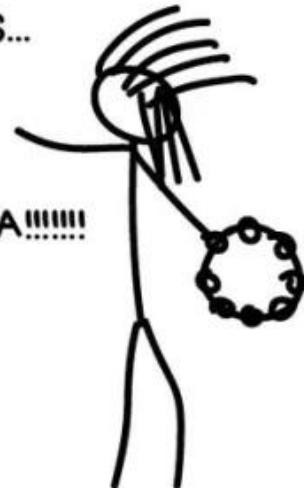
深度学习是什么

$$\begin{aligned}\frac{d\mathcal{L}}{dW} &= \sum_i \frac{d\mathcal{L}}{da_i} \frac{da_i}{dz_i} \frac{dz_i}{dW} \\&= \frac{1}{N} \sum_i - \left(\frac{y_i}{a_i} - \frac{1-y_i}{1-a_i} \right) \cdot \frac{\exp(-z)}{(1+\exp(-z))^2} \cdot x_i \\&\quad \cancel{\frac{1}{N} \sum_i - \left(\frac{y_i - a_i}{a_i(1-a_i)} \right) \cdot a_i(1-a_i) \cdot x_i} \\&= \frac{1}{N} \sum_i -(y_i - a_i) \cdot x_i\end{aligned}$$

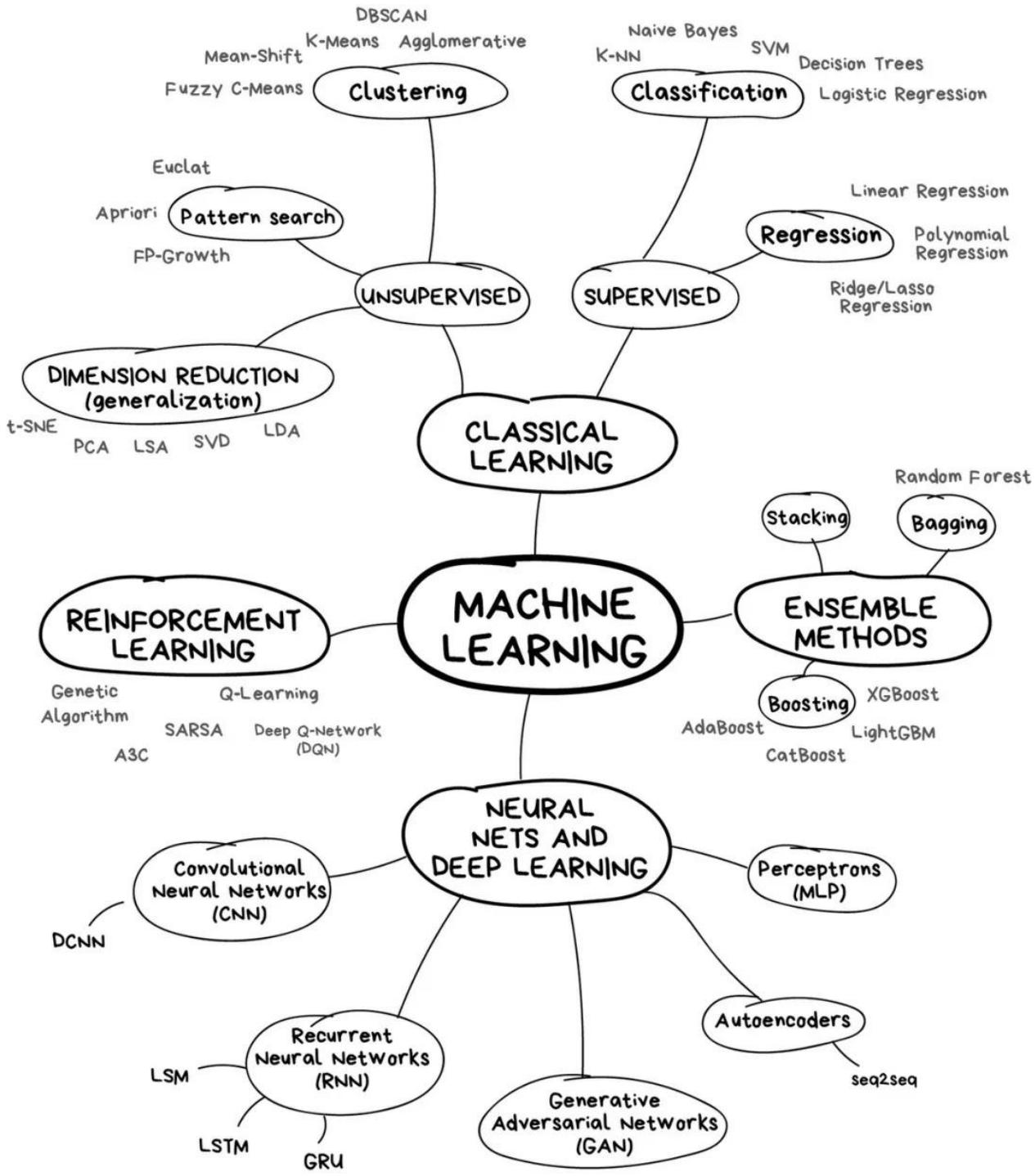
AS WE CAN SEE HERE,
THIS IS OBVIOUS!



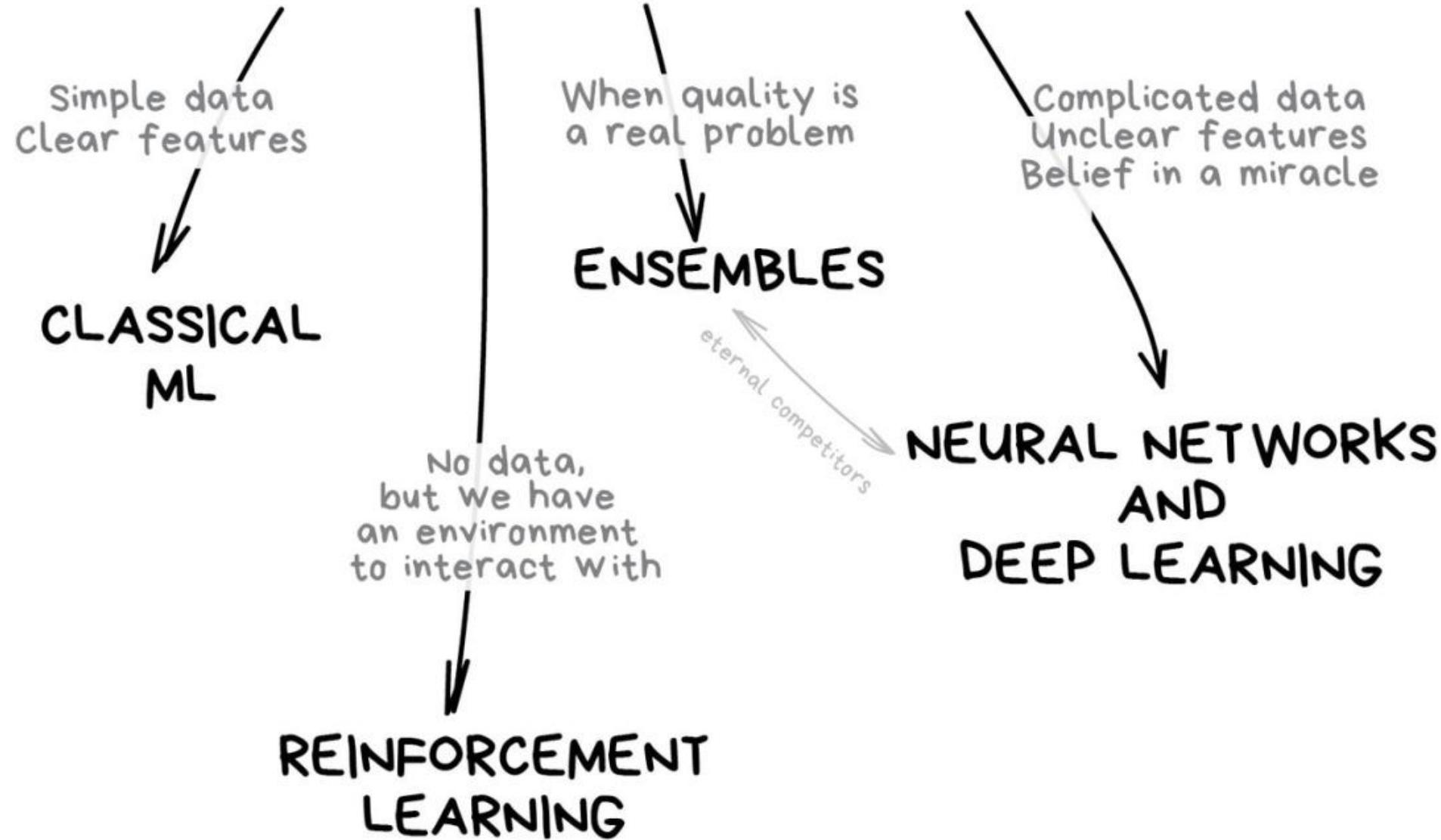
PROGRAMMERS ARE PROGRAMMING!
DATASCIENCE!
PROFESSION OF FUTURE!
IN THE NEXT FIVE YEARS...
EXPONENTIAL GROWTH!!!
SMART MACHINES!
A-A-A-A-A-A-A-A-A-AAA!!!!!!



TWO TYPES OF ARTICLES ABOUT MACHINE LEARNING



THE MAIN TYPES OF MACHINE LEARNING



CLASSICAL MACHINE LEARNING

Data is pre-categorized or numerical

SUPERVISED

Predict a category

Predict a number

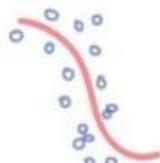
CLASSIFICATION

«Divide the socks by color»



REGRESSION

«Divide the ties by length»



Data is not labeled in any way

UNSUPERVISED

Divide by similarity

CLUSTERING

«Split up similar clothing into stacks»



Identify sequences

ASSOCIATION

«Find what clothes I often wear together»



DIMENSION REDUCTION (generalization)

«Make the best outfits from the given clothes»



hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

«KITTY»

13 times

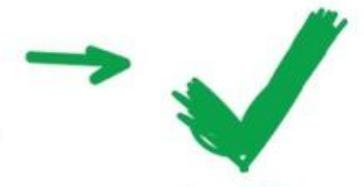
672 times

THE SIMPLEST SPAM-FILTER

(used until 2010)

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

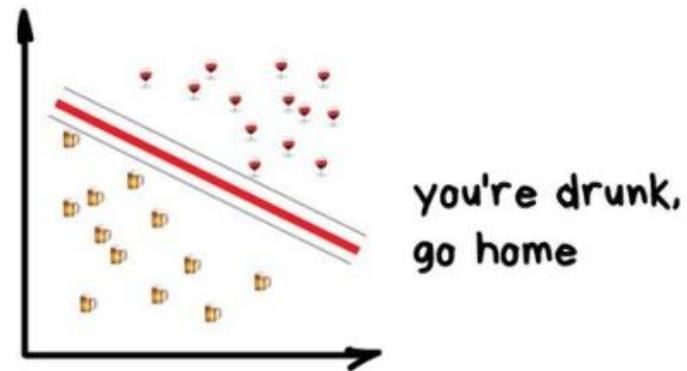
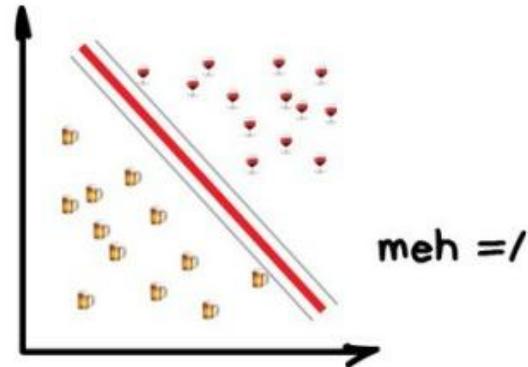
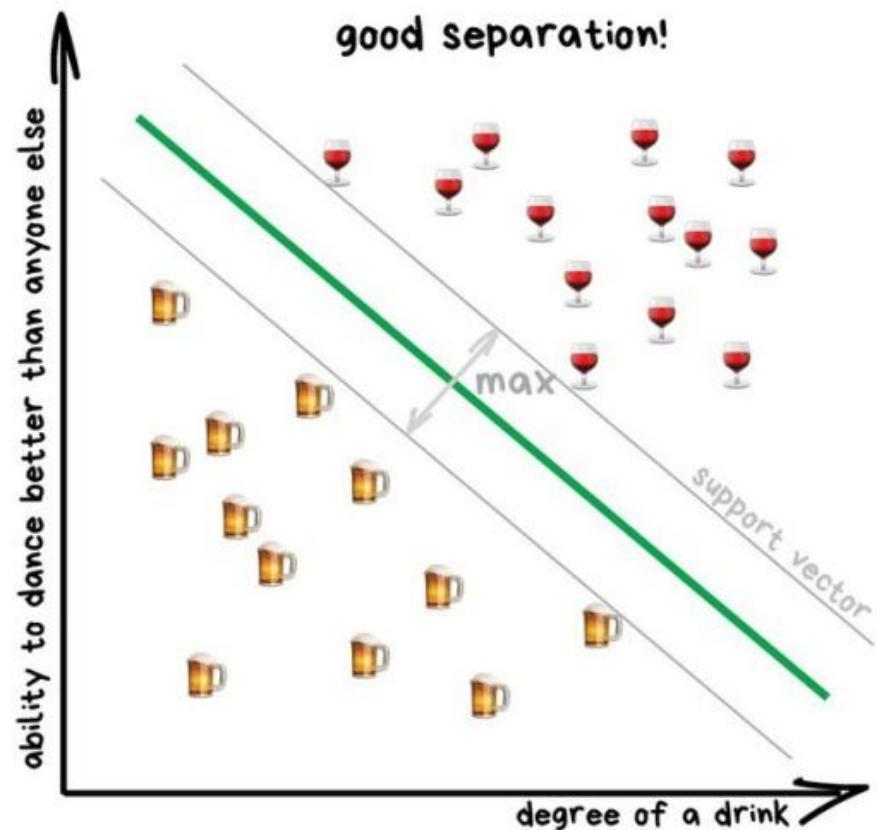
BAYES' THEOREM



NOT SPAM

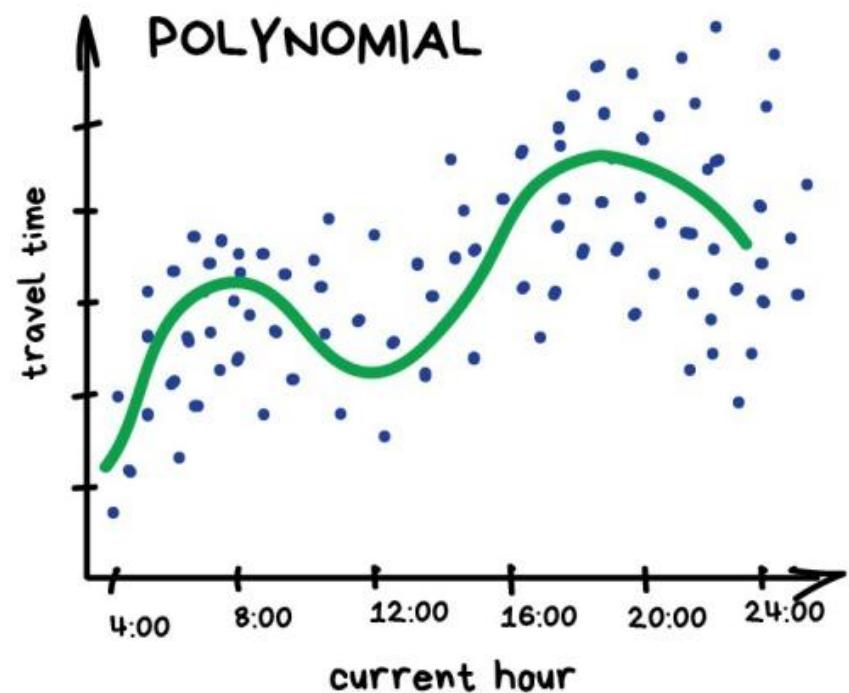
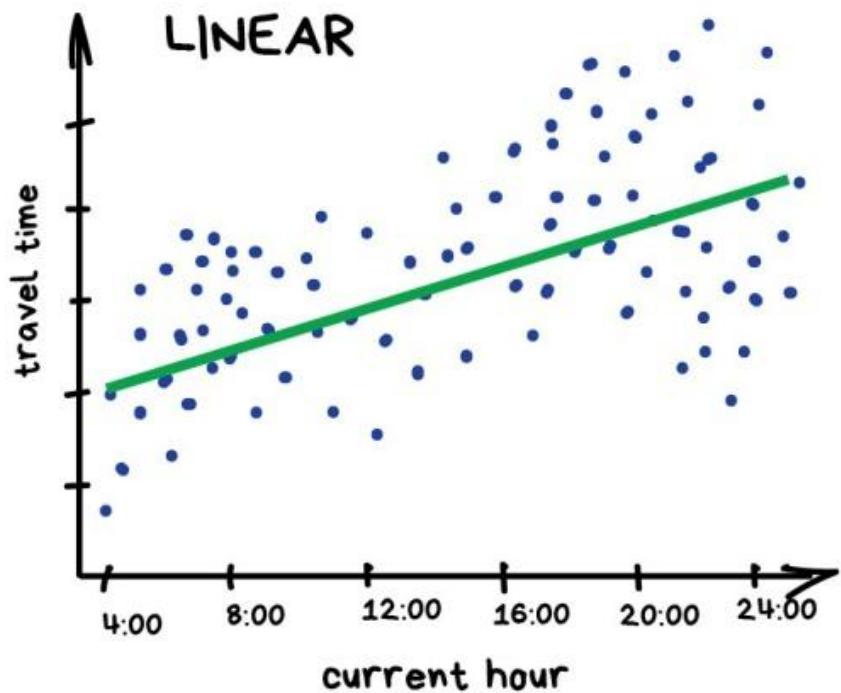
NAIVE BAYES

SEPARATE TYPES OF ALCOHOL



SUPPORT VECTOR MACHINE

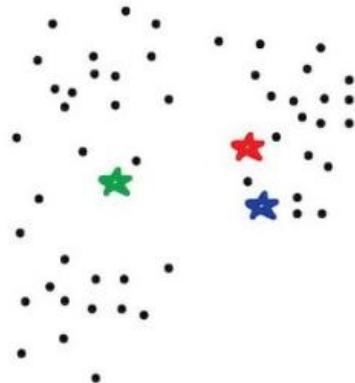
PREDICT TRAFFIC JAMS



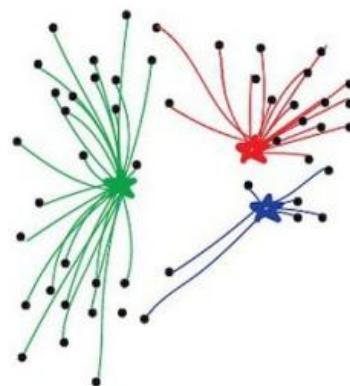
REGRESSION

PUT KEBAB KIOSKS IN THE OPTIMAL WAY

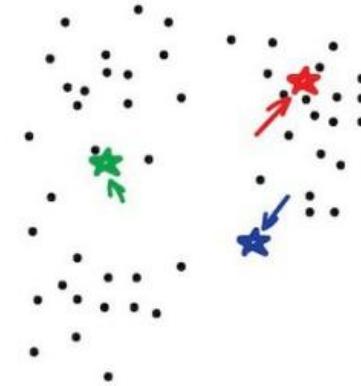
(also illustrating the K-means method)



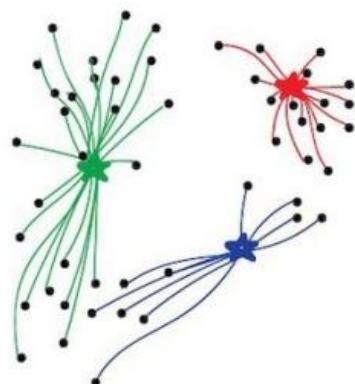
1. Put kebab kiosks in random places in city



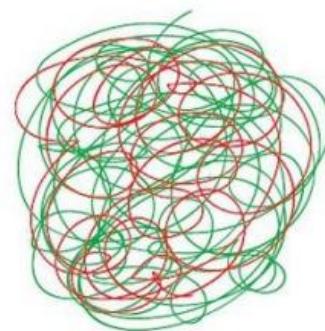
2. Watch how buyers choose the nearest one



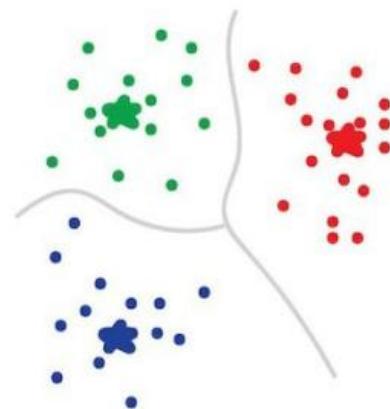
3. Move kiosks closer to the centers of their popularity



4. Watch and move again

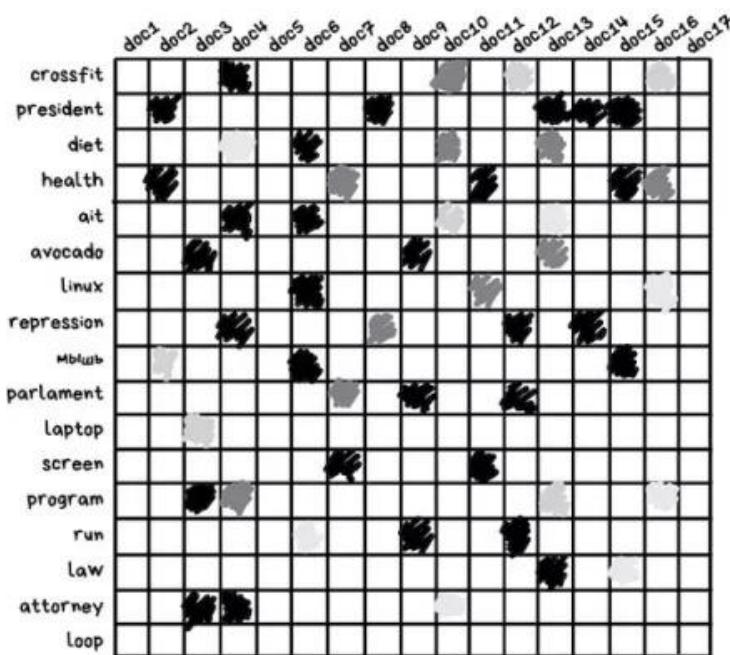


5. Repeat a million times

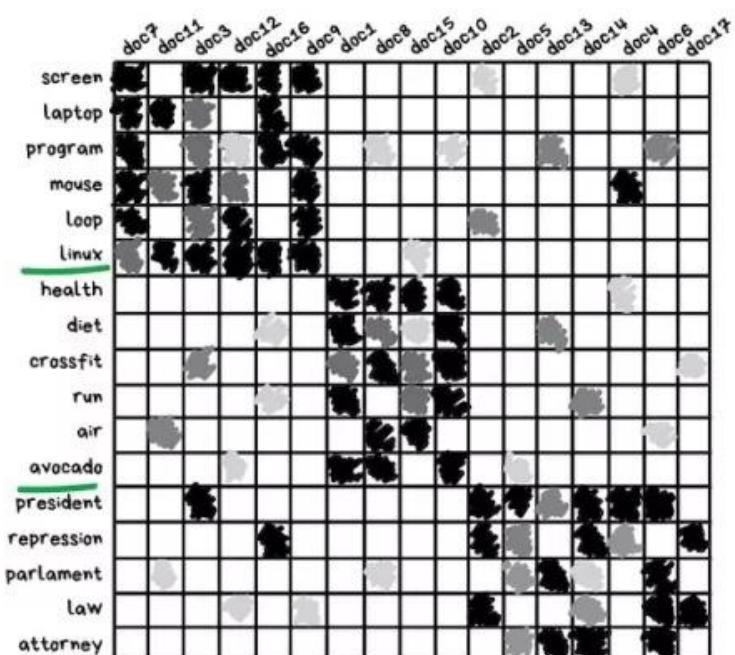


6. Done!
You're god of kebabs!

SEPARATE DOCUMENTS BY TOPIC



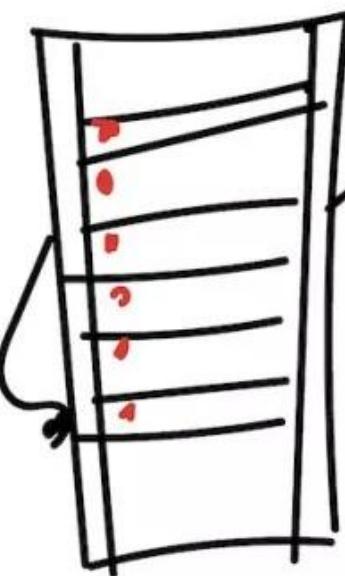
→
SVD
2. Transform



1. Build a matrix of how often each word can be found in each document
(black - more often)

3. Get visual topic clusters.
Even if the words haven't met together

LATENT SEMANTIC ANALYSIS (LSA)



THAT MEATBAG BOUGHT A SOFA



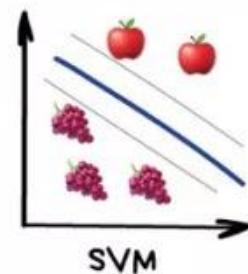
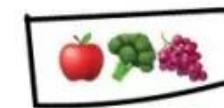
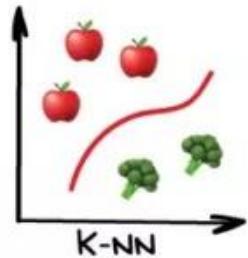
PROBABLY, HE LOVE SOFAS!!!



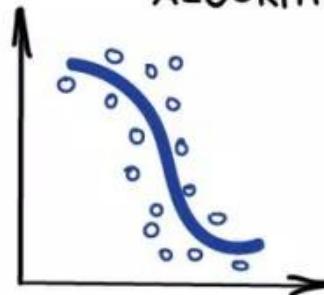
RECOMMEND HIM 148 MORE SOFAS



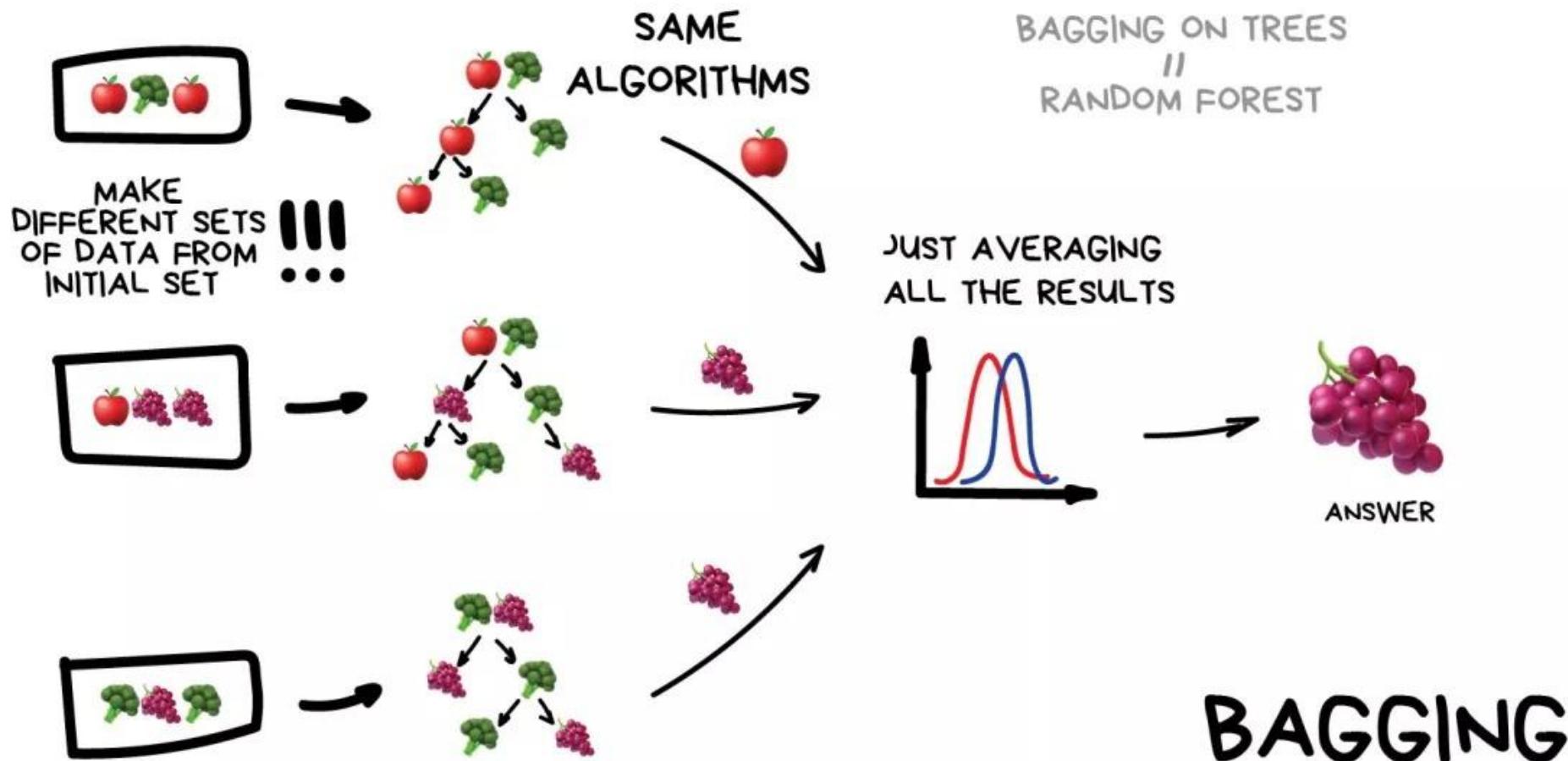
DIFFERENT ALGORITHMS

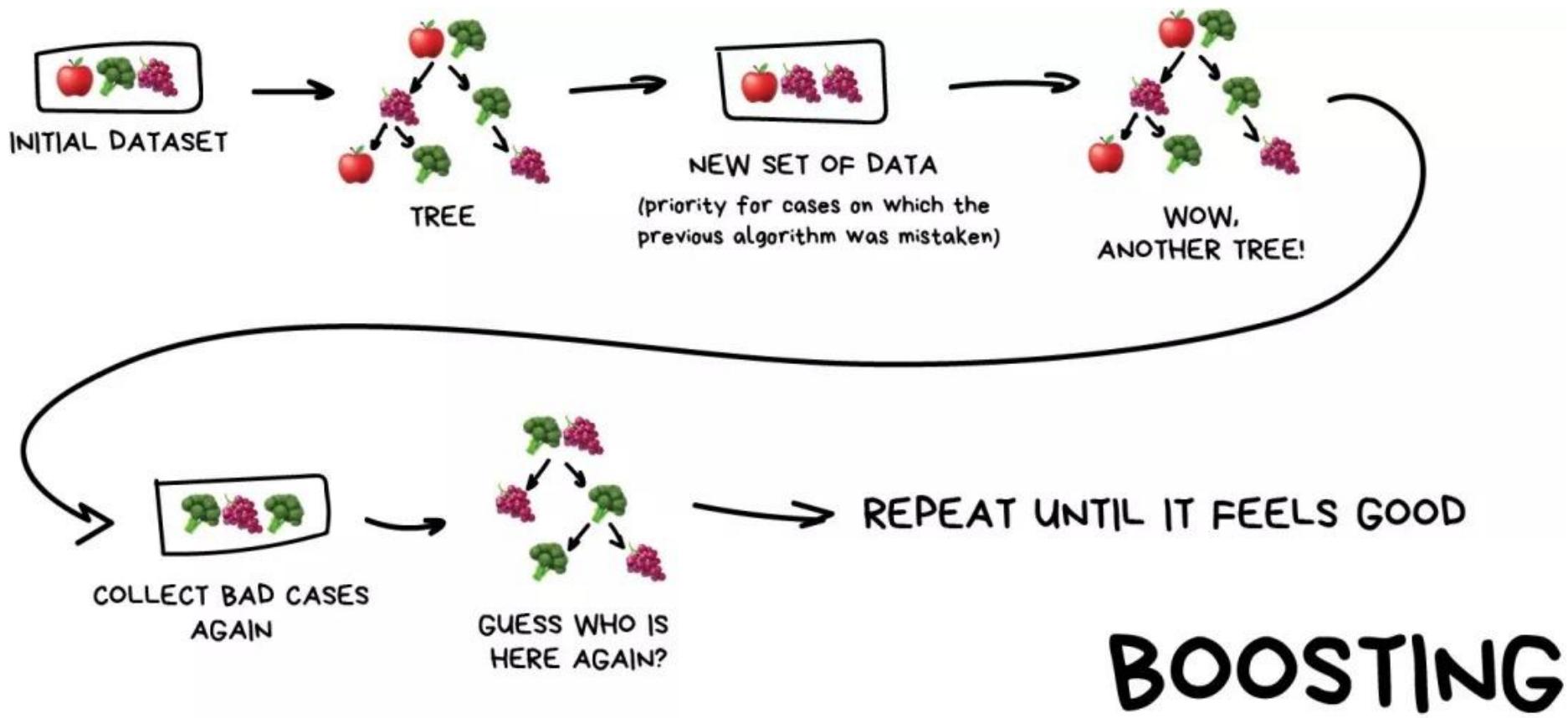


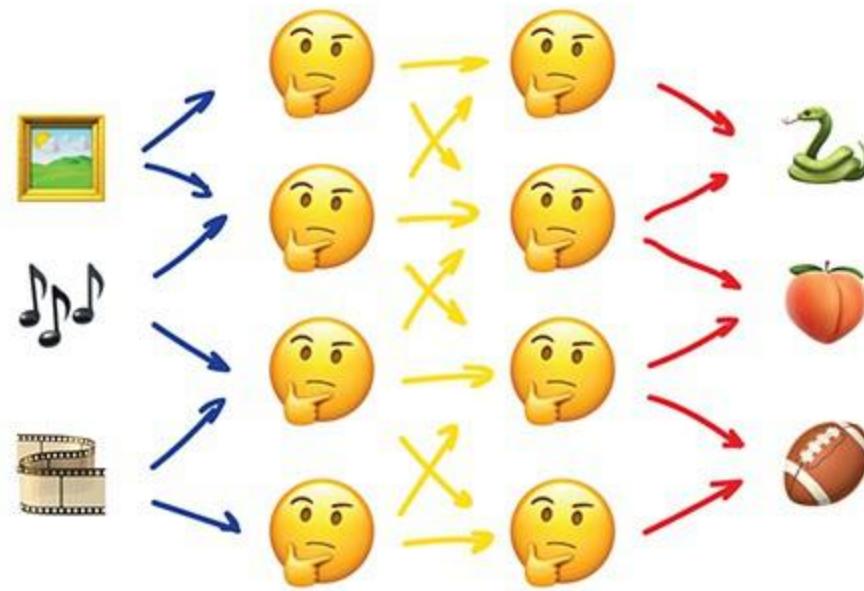
FINAL DECISION ALGORITHM



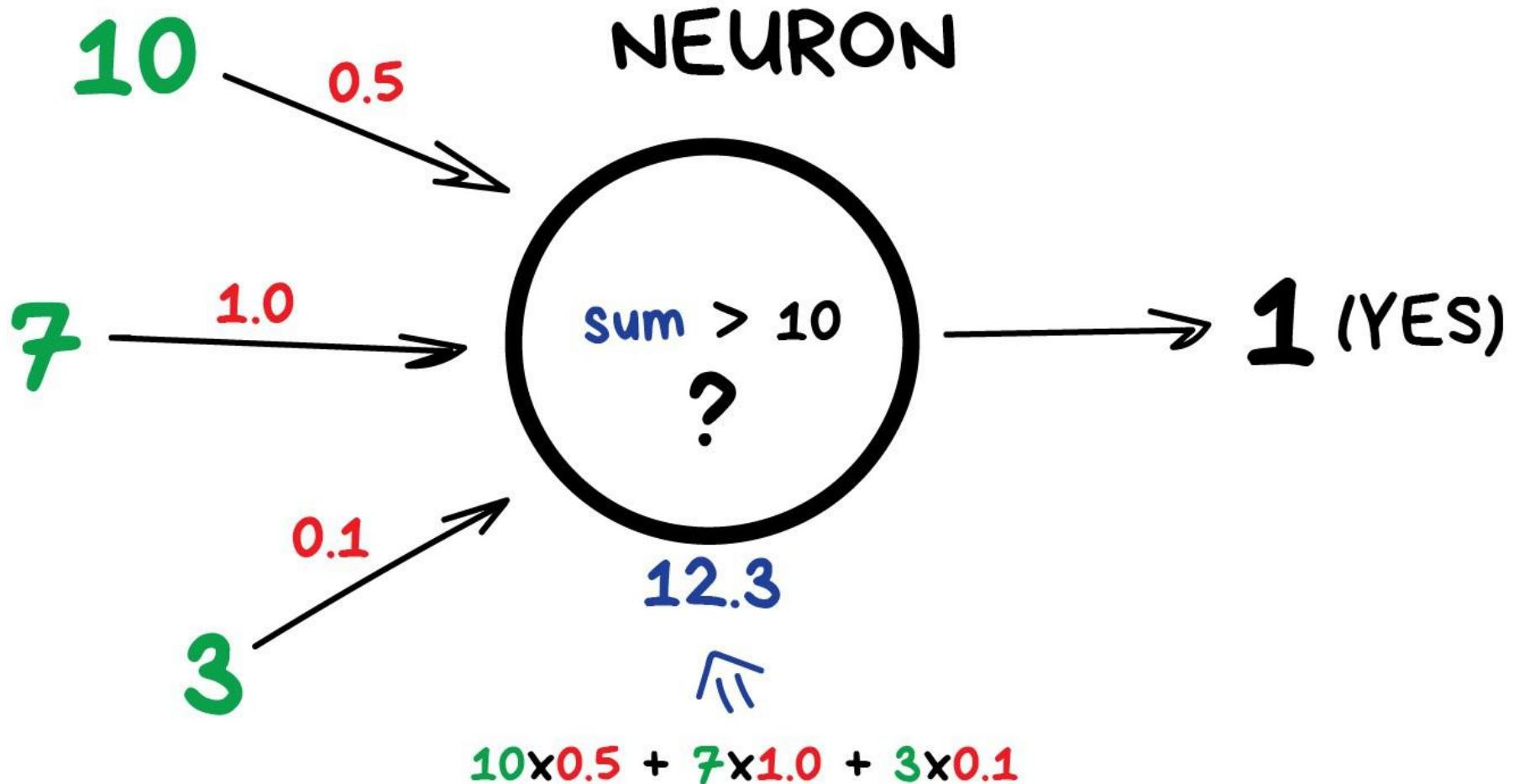
STACKING

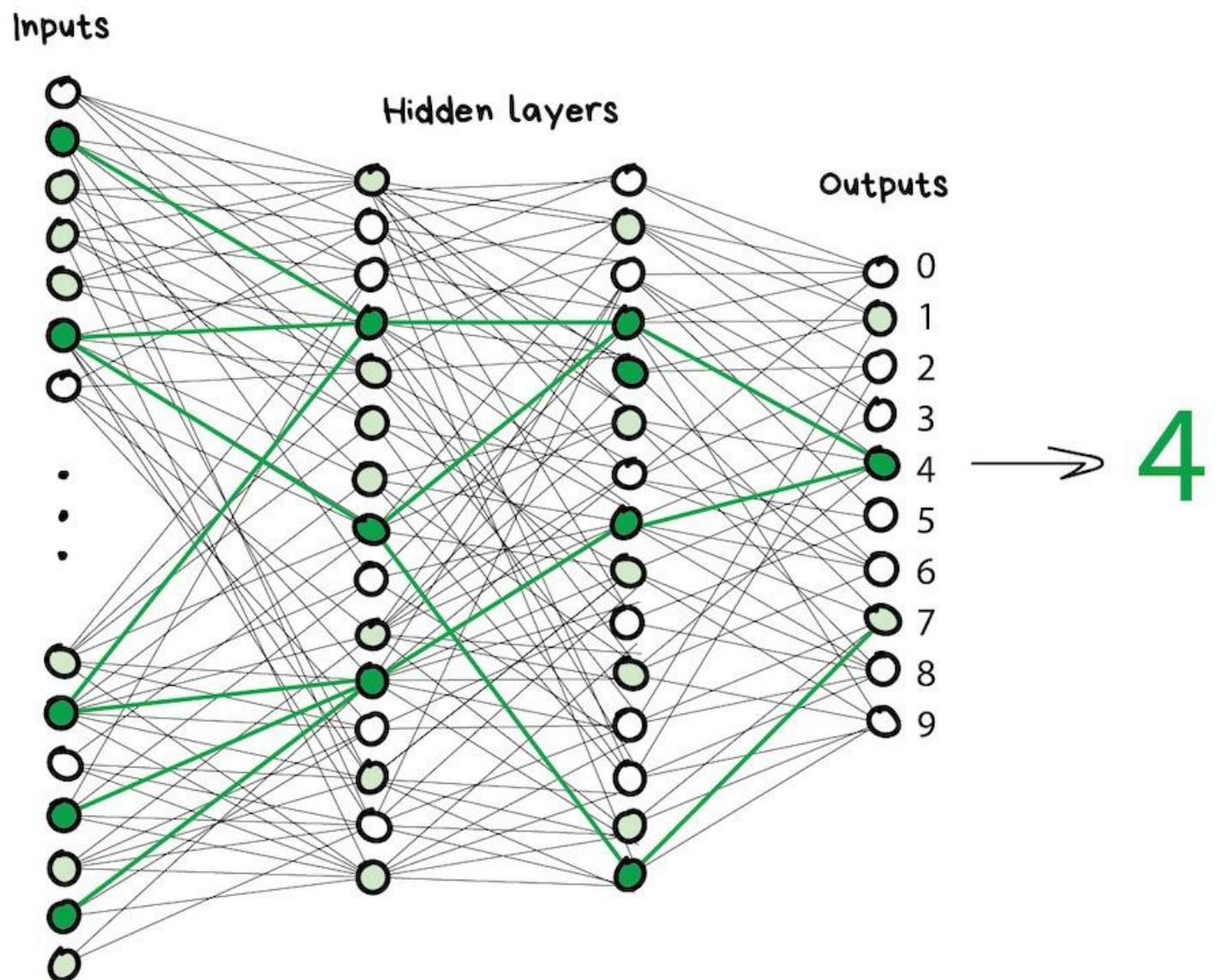
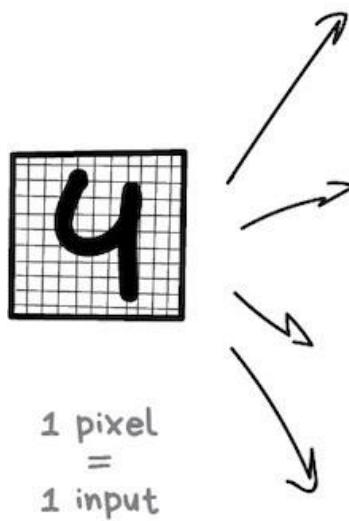




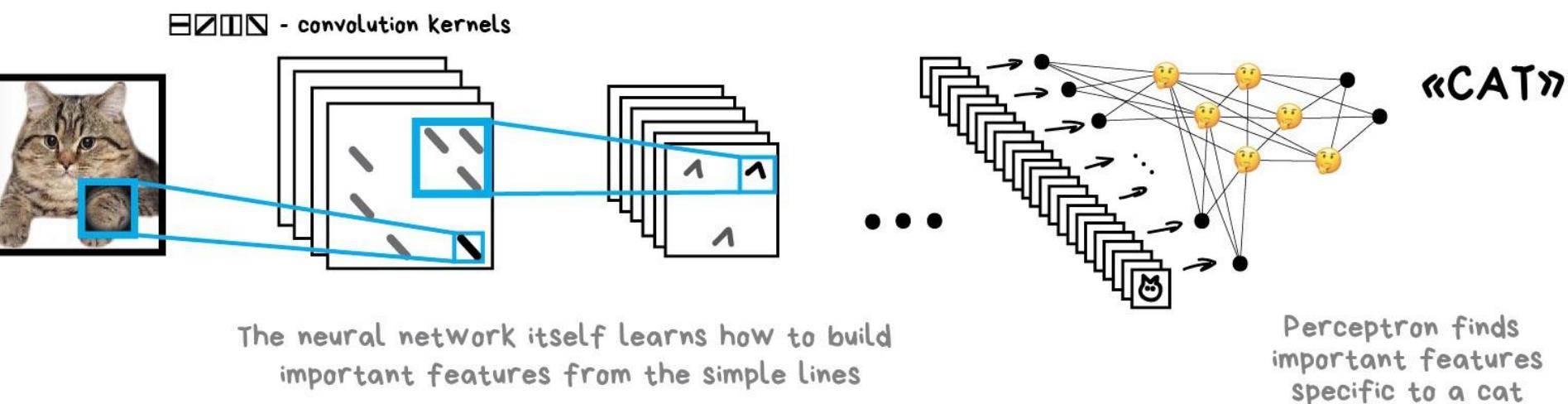
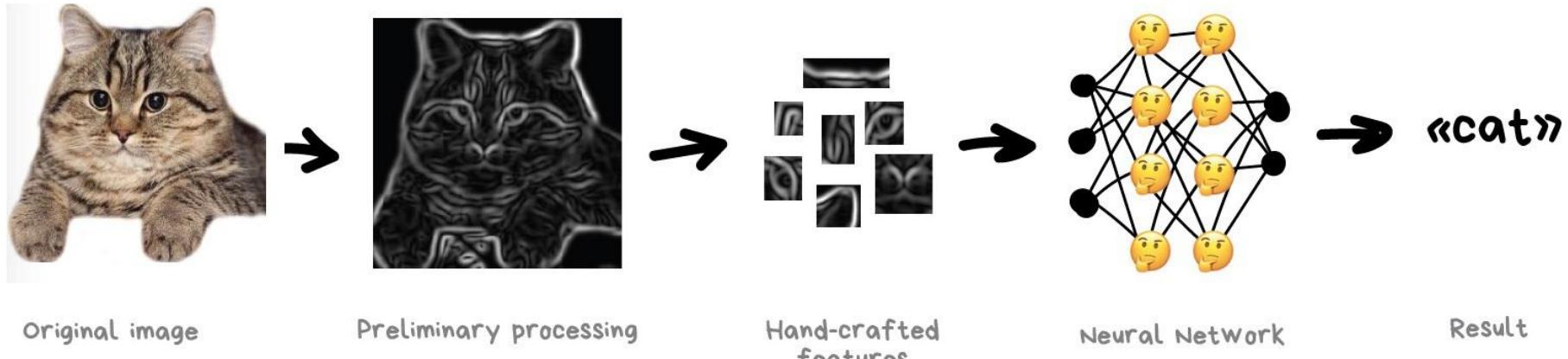


Neural Networks

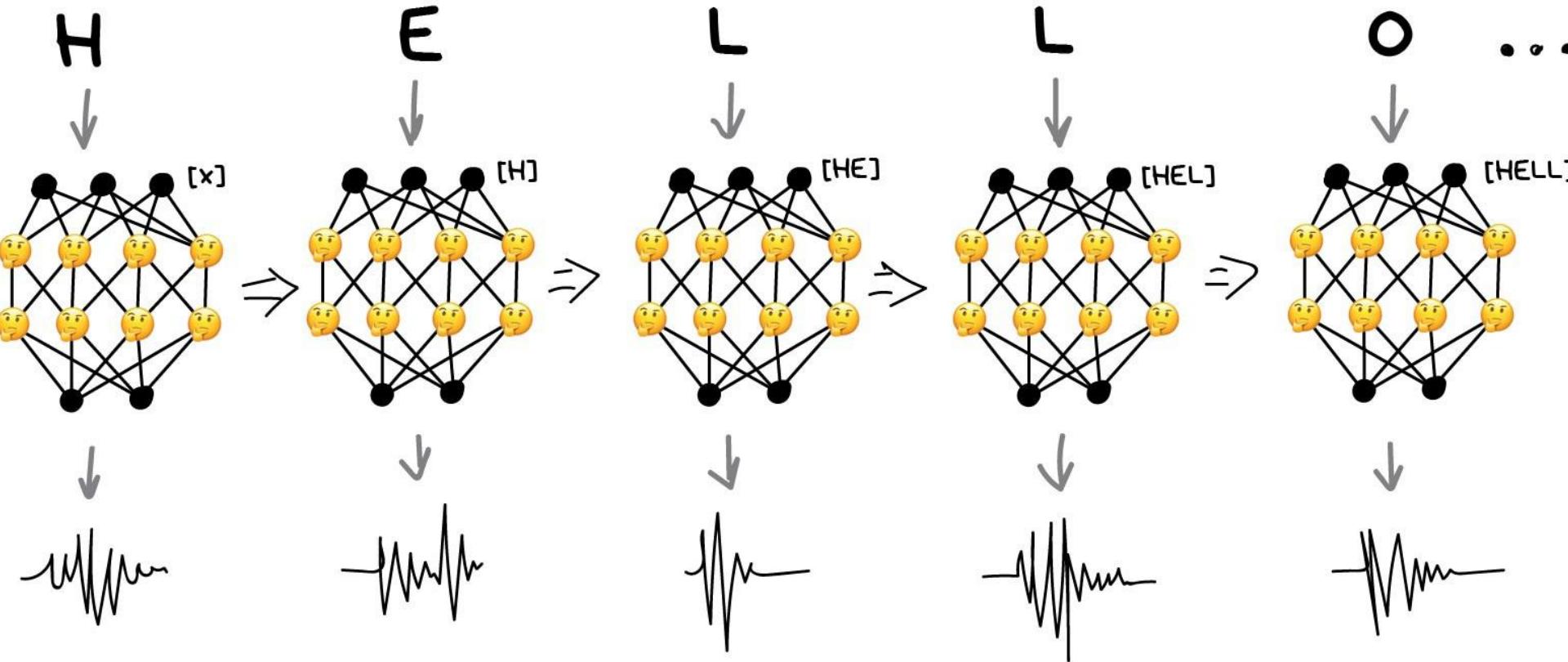




MULTILAYER PERCEPTRON (MLP)



CONVOLUTIONAL NEURAL NETWORK (CNN)

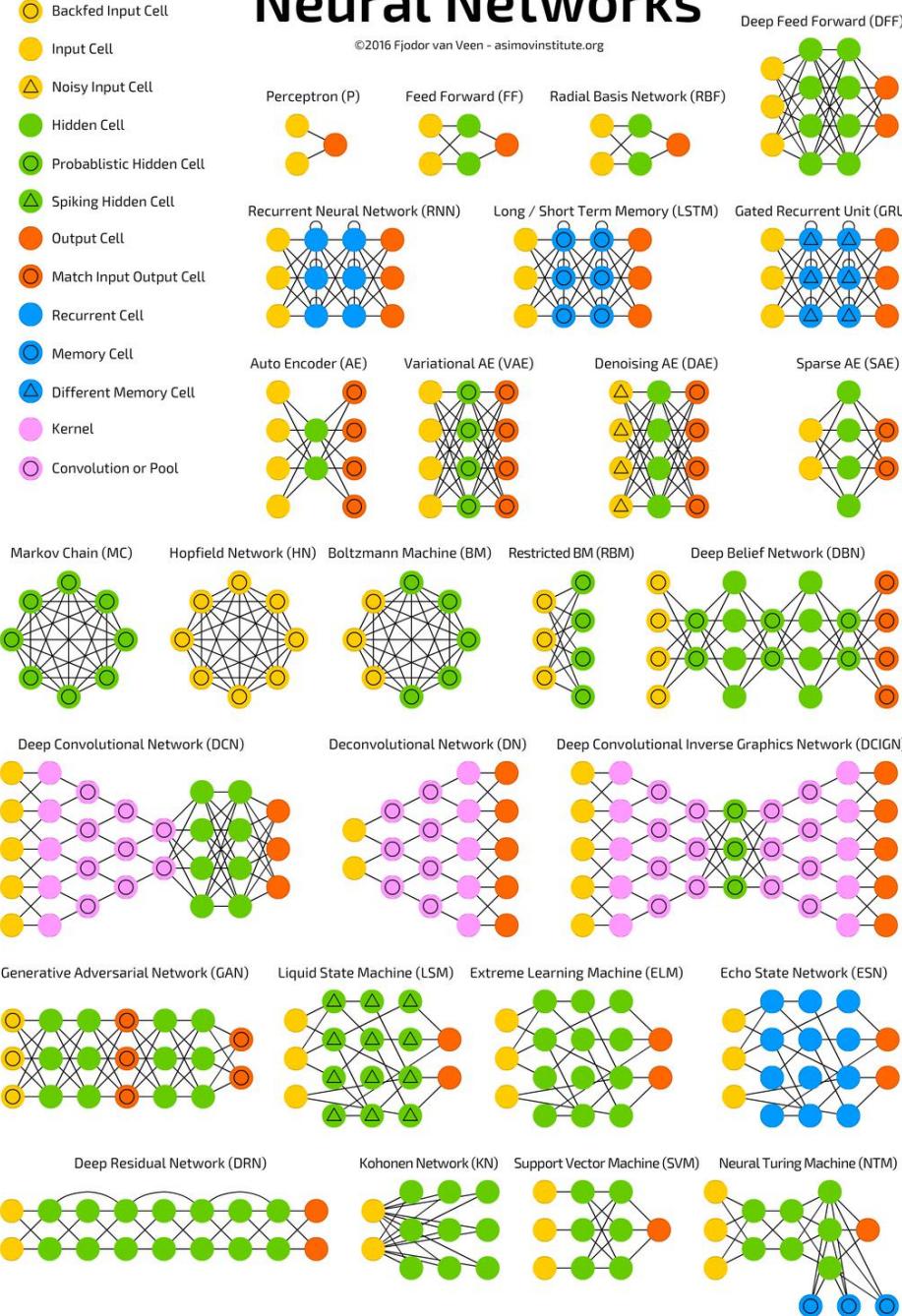


RECURRENT NEURAL NETWORK (RNN)

A mostly complete chart of
Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

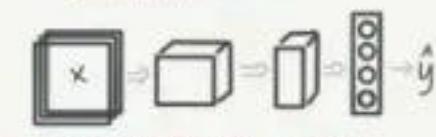
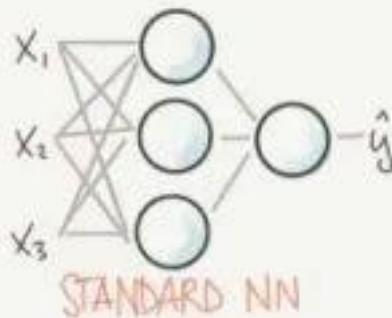
- Backfed Input Cell
- Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- △ Different Memory Cell
- Kernel
- Convolution or Pool



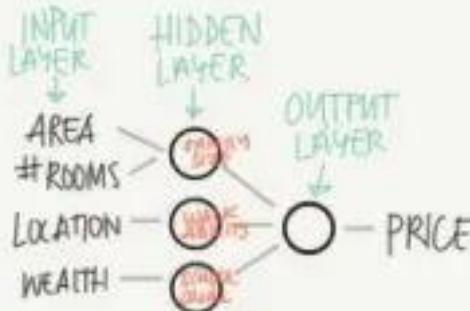
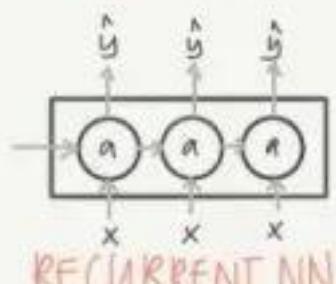
INTRO TO DEEP LEARNING

SUPERVISED LEARNING

INPUT: X	OUTPUT: y	NN TYPE
HOME FEATURES	PRICE	STANDARD NN
AD+USER INFO	WILL CLICK ON AD (0/1)	
IMAGE	OBJECT (1..1000)	CONV. NN (CNN)
AUDIO	TEXT TRANSCRIPT	RECURRENT NN (RNN)
ENGLISH	CHINESE	
IMAGE/RADAR	POS OF OTHER CARS	CUSTOM/HYBRID



NETWORK ARCHITECTURES



NNs CAN DEAL WITH BOTH
STRUCTURED & UNSTRUCTURED DATA



STRUCTURED



"THE QUICK BROWN FOX"
UNSTRUCTURED

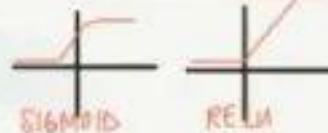


HUMANS ARE GOOD
AT THIS

WHY NOW?



ONE OF THE
BIG BREAKTHROUGHS
HAS BEEN MOVING
FROM SIGMOID TO
RELU FOR FASTER
GRADIENT DESCENT

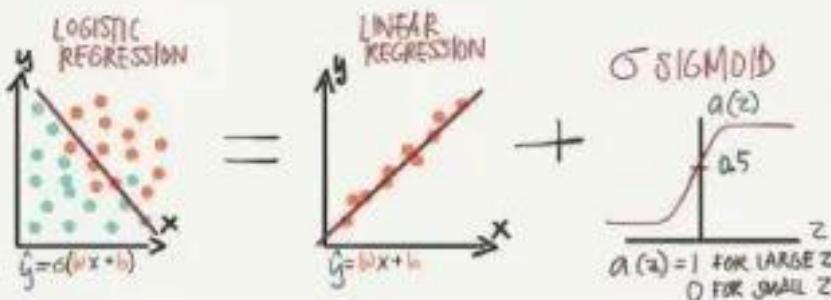


FASTER COMPUTATION
IS IMPORTANT TO SPEED UP
THE ITERATIVE PROCESS

BINARY CLASSIFICATION



\rightarrow
1: CAT
0: NOT CAT
 y



THE TASK IS TO LEARN w & b BUT HOW?

A: OPTIMIZE HOW GOOD THE GUESS IS BY MINIMIZING THE DIFF BETWEEN GUESS (\hat{y}) AND TRUTH (y)

$$\text{LOSS} = L(\hat{y}, y)$$

$$\text{COST} = J(w, b) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)})$$

COST = LOSS FOR THE ENTIRE DATASET

LOGISTIC REGRESSION

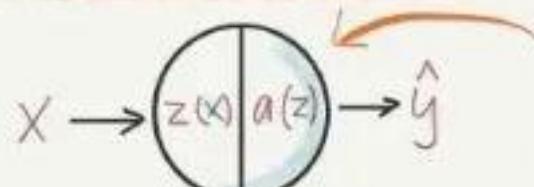
AS A NEURAL NET

FINDING THE MINIMUM WITH GRADIENT DESCENT

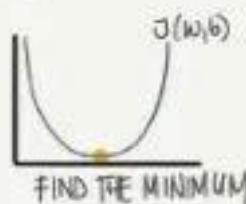


1. FIND THE DOWNTURN DIRECTION (USING DERIVATIVES)
 2. WALK (UPDATE w & b) AT A LEARNING RATE
- REPEAT UNTIL YOU REACH BOTTOM (CONVERGE)

PUTTING IT ALL TOGETHER



MINI NEURAL NET



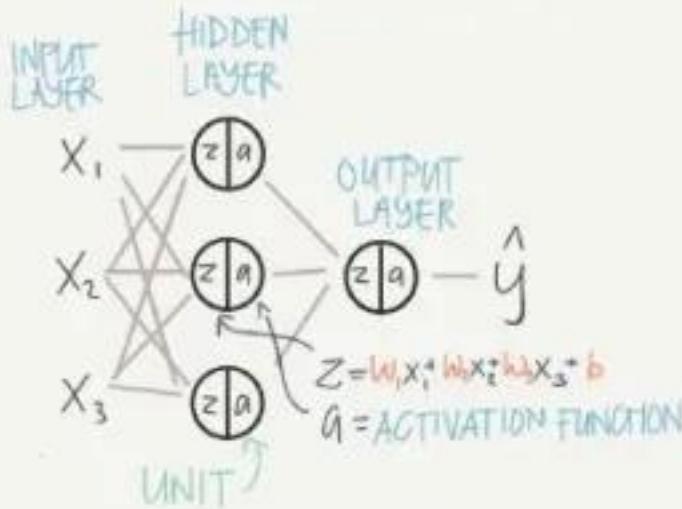
$$z(x) = wx + b$$

$$\hat{y} = a(z) = \text{SIGMOID}(z)$$

1. FORWARD PROPAGATION • CALCULATE \hat{y}
2. BACKWARD PROPAGATION • GRADIENT DESCENT + UPDATE w & b

REPEAT UNTIL IT CONVERGES

2 LAYER NEURAL NET



ACTIVATION FUNCTIONS



BINARY CLASSIFIER
- ONLY USED FOR
OUTPUT LAYER

SLOW GRAD
DESCENT SINCE
SLOPE IS SMALL
FOR LARGE/SMALL VAL

NORMALIZED
 \Rightarrow GRADIENT
DESCENT IS
FASTER

DEFAULT
CHOICE FOR
ACTIVATION
SLOPE = 1/0

AVOIDS UNDEF
SLOPE AT 0
BUT RARELY
USED IN PRACTICE

INITIALIZING $W+b$

WHAT IF: INIT TO \emptyset

THIS WILL CAUSE ALL THE UNITS
TO BE THE SAME AND LEARN
EXACTLY THE SAME FEATURES

SOLUTION: RANDOM INIT
BUT ALSO WANT THEM
SMALL SO RAND $\times 0.01$

PERPARAM
@Sorjhiel
teststefanutz

SHALLOW NEURAL NETS

WHY ACTIVATION FUNCTIONS?

EX. WITH NO ACTIVATION - $a = z$

$$\begin{aligned} a^{[1]} &= z^{[1]} = W^{[1]} x + b^{[1]} \\ a^{[2]} &= z^{[2]} = W^{[2]} a^{[1]} + b^{[2]} \end{aligned}$$

LAYER 1
LAYER 2

PLUG IN $a^{[1]}$

$$\begin{aligned} a^{[2]} &= W^{[2]}(W^{[1]} x + b^{[1]}) + b^{[2]} \\ &= W^{[2]} W^{[1]} x + W^{[2]} b^{[1]} + b^{[2]} \\ &= W' x + b' \end{aligned}$$

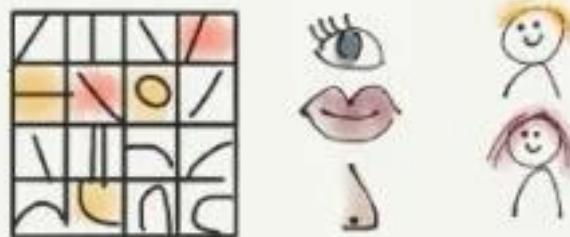
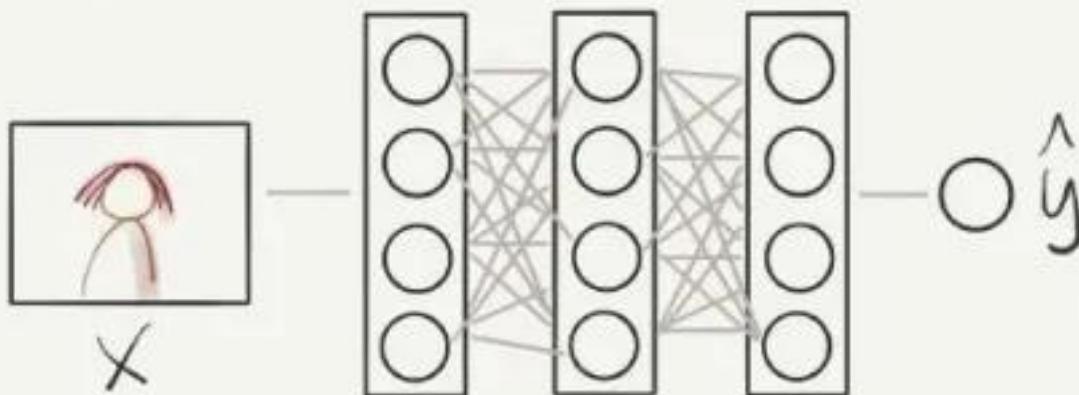
LINEAR
FUNCTION

WE COULD JUST
AS WELL HAVE
SKIPPED THE WHOLE
NEURAL NET &
USED LIN. REGR.

DEEP NEURAL NETS

WHY DEEP NEURAL NETS?

THERE ARE FUNCTIONS A
SMALL DEEP NET CAN COMPUTE
THAT SHALLOW NETS NEED EXP.
MORE UNITS TO COMP.



LOW LEVEL
AUDIO WAVE
FEATURES
↑ PITCH

— PHONEMES — WORDS — SENTENCES

C A T

VERY DATA HUNGRY

NEED ^{LOTS OF} COMPUTER
POWER

ALWAYS VECTORIZE
VECTOR MULT. CHEAPER THAN FOR LOOPS

COMPUTE ON GPUs

LOTS OF HYPERPARAMS

LEARNING RATE α # HIDDEN UNITS
ITERATIONS CHOICE OF ACTIVATION
HIDDEN LAYERS MOMENTUM

MINI-BATCH SIZE
REGULARIZATION

SETTING UP YOUR ML APP

CLASSIC ML

100 - 1000 SAMPLES

TRAIN	DEV	TEST
60%	20%	20%

ALL FROM SAME PLACE
DISTRIBUTION

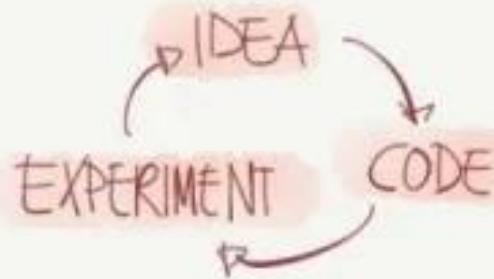
DEEP LEARNING

1M SAMPLES

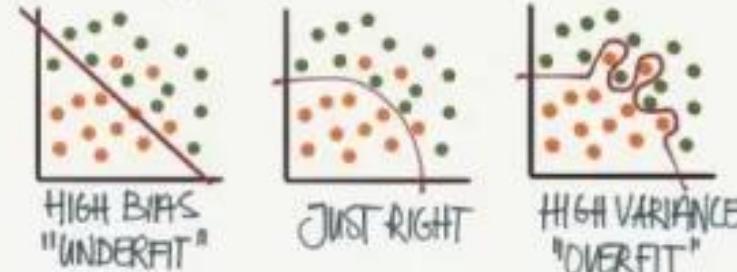
TRAIN	D	T
98%	1%	1%



 **TIP**
DEV & TEST SHOULD COME
FROM SAME DISTRIBUTION

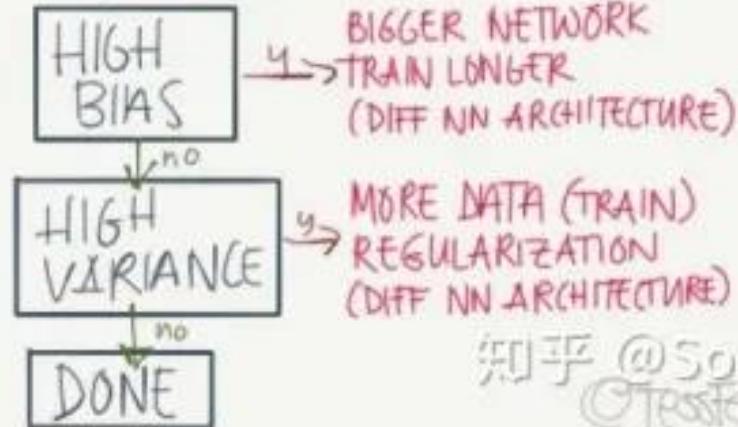


BIAS/VARIANCE



	ERROR				
TRAIN	1%	15%	15%	0.5%	
TEST	11%	16%	30%	1%	
	HIGH VARIANCE	HIGH BIAS	HIGH BIAS & VARIANCE	LOW BIAS & VARIANCE	
	ASSUMING HUMANS GET 0% ERROR				

THE ML RECIPE



REGULARIZATION

PREVENTING OVERFITTING

L2 REGULARIZATION

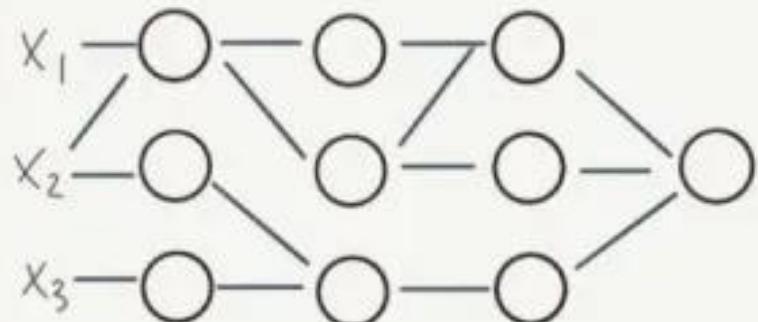
$$\text{COST: } J(w, b) = \frac{1}{m} \sum_{i=1}^m d(\hat{y}_i, y_i) + \frac{\lambda}{2m} \|w\|_2^2$$

EUCLIDEAN
NORM

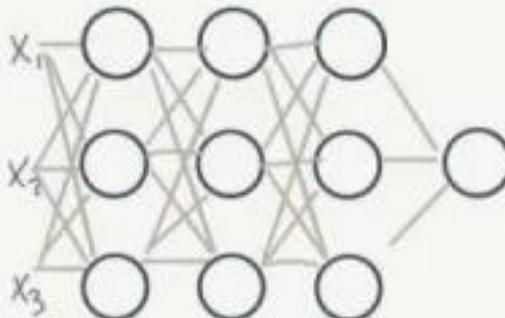
L1 REGULARIZATION

$$\text{COST: } J(w, b) = \frac{1}{m} \sum_{i=1}^m d(\hat{y}_i, y_i) + \frac{\lambda}{m} \|w\|_1$$

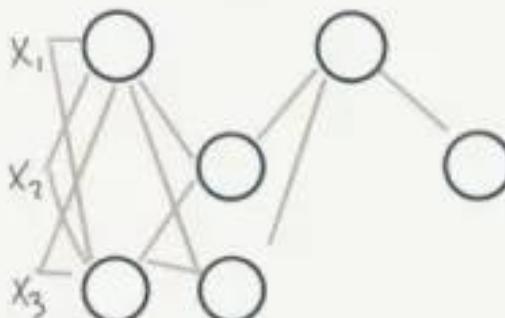
BOTH PENALIZE LARGE WEIGHTS \Rightarrow
SOME WILL BE CLOSE TO 0 \Rightarrow
SIMPLER NETWORKS



DROPOUT



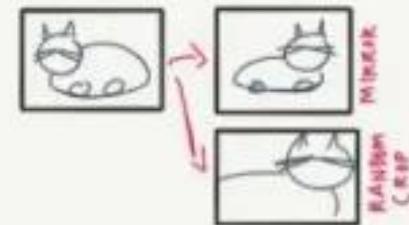
FOR EACH ITERATION & SAMPLE
SOME NODES ARE RANDOMLY
DROPPED (BASED IN KEEP-PROB)



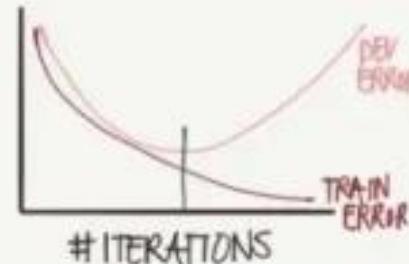
WE GET SIMPLER NWs
& LESS CHANCE TO RELY ON
SINGLE FEATURES

OTHER REGULARIZATION TECHNIQUES

DATA AUGMENTATION
GENERATE NEW PICS FROM EXISTIN



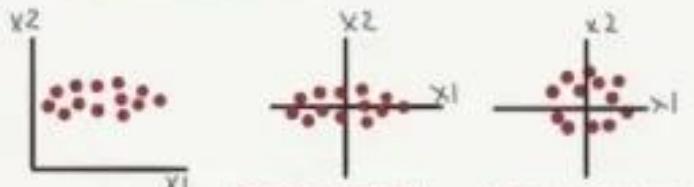
EARLY STOPPING



PROBLEM: AFFECTS BOTH
BIAS & VARIANCE

OPTIMIZING TRAINING

NORMALIZING INPUTS

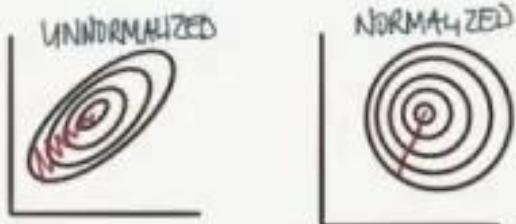


STEP 1: CENTER AROUND 0,0

STEP 2: SCALE SO VARIANCE IS SAME
 $c_x - 1 \rightarrow 1$

TIP
USE SAME AVG/VAR TO NORMALIZE DEV/TEST

WHY DO WE DO THIS?



IF WE NORMALIZE, WE CAN USE A MUCH LARGER LEARNING RATE α

DEALING WITH VANISHING/EXPLODING GRADIENTS

EX: DEEP NN (L LAYERS)

$$\hat{y} = W^{(L-1)} W^{(L-2)} \dots W^{(0)} x + b$$

IF $W = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \Rightarrow 0.5^{-1} \Rightarrow$ VANISHING
OR $W = \begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix} \Rightarrow 1.5^{-1} \Rightarrow$ EXPLODING

IN BOTH CASES GRADIENT DESCENT TAKES A VERY LONG TIME

PARTIAL SOLUTION: CHOOSE INITIAL VALUES CAREFULLY

$$W^{(l)} = \text{rand} * \sqrt{\frac{2}{n^{l+1}}} \quad (\text{FOR RELU})$$

$$\text{XAVIER } \sqrt{\frac{1}{n^{l+1}}} \quad (\text{FOR TANH})$$

SETS THE VARIANCE

GRADIENT CHECKING

IF YOUR COST DOES NOT DECREASE ON EACH ITER YOU MAY HAVE A BACKPROP BUG.

GRADIENT CHECKING APPROXIMATES THE GRADIENTS SO YOU CAN VERIFY CALC.

NOTE ONLY USE WHEN DEBUGGING SINCE IT'S SLOW

OPTIMIZATION ALGORITHMS

MINI-BATCH GRAD. DESCENT



SPLIT YOUR DATA INTO MINI-BATCHES
EACH DO GRAD DESCENT AFTER EACH BATCH
THIS WAY YOU CAN PROGRESS AFTER
JUST A SHORT WHILE



CHOOSING THE MINIBATCH SIZE

SIZE = $m \rightarrow$ BATCH GRAD DESC.
SIZE = 1 \rightarrow STOCHASTIC GRAD DESC.



TIP:
IF YOU HAVE < 2000 SAMPLES
USE SIZE = 2000
OTHERWISE, USE 64, 128, 256...
SO X+Y FITS IN CPU/GPU CACHE

GRADIENT DESCENT W. MOMENTUM



WE WANT TO REDUCE
OSCILLATION \uparrow SO WE GET TO THE
GOAL FASTER

SOLUTION: SMOOTH OUT THE
CURVE BY TAKING AN EXPONENTIALLY
WEIGHTED AVERAGE OF THE
DERIVATIVES (i.e. LAST ONE HAS MORE)
IMPORTANCE

RMSProp - ROOT MEAN SQUARED



NORMALIZE GRADIENT USING A MOVING AVG.

$$S_{dw} = \beta S_{dw} + (1-\beta) dw^2$$

$$S_{db} = \beta S_{db} + (1-\beta) db^2$$

$$w = w - \alpha \frac{dw}{\sqrt{S_{dw}}} \quad b = b - \alpha \frac{db}{\sqrt{S_{db}}}$$

ADAM OPTIMIZATION COMBO OF GD w/ MOMENTUM & RMSProp

LEARNING RATE DECAY

IDEA: USE A LARGE α IN THE
BEGINNING. THEN DECREASE AS
WE GET CLOSER TO GOAL

OPTION 1: $\alpha = \frac{1}{1 + \text{DECAY RATE} \cdot \text{EPOCH}}$

EXPOENTIAL: $\alpha = 0.95^{\text{EPOCH}} \alpha_0$

OPTION 3: $\alpha = \frac{k}{\sqrt{\text{EPOCH}}} \alpha_0$

OPTION 4: $\alpha = \frac{k}{\sqrt{\text{EPOCH}}} \alpha_0$

OPTION 5: DISCRETE STAIRCASE α_0

OPTION 6: MANUAL

EPOCH = 1 PASS THROUGH THE DATA

HYPERPARAM

TUNING

WHICH HYPERPARAMS ARE MOST IMPORTANT?

α LEARNING RATE

HIDDEN UNITS

MINIBATCH SIZE

β MOMENTUM TURN = 0.9

LAYERS

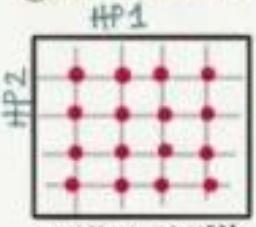
LEARNING RATE DECAY

$\beta_1 = 0.9$ $\beta_2 = 0.999$ $\epsilon = 10^{-8}$ (ADAM)



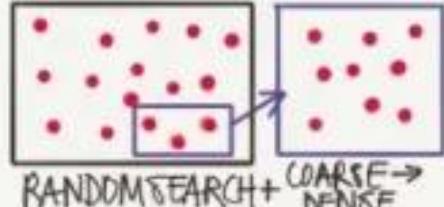
TESTING VALUES

CLASSIC ML



GRID SEARCH

SOLUTION



PROBLEM: ONE ITERATION TAKES A LONG TIME & IN 16 GO'S WE HAVE ONLY TRIED 4 α - BUT 4 DIFF ϵ ↪ NOT AS IMPORTANT

MY PANDA IS ACTUALLY A MISLABELLED CAT BECAUSE I ONLY PANDAS

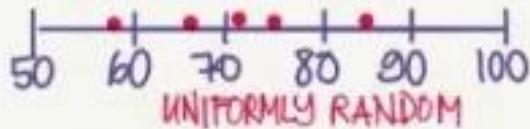
PANDA VS CAVIAR

BABY'S IT ONE MODEL & TUNE

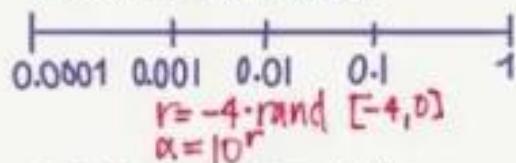
GOOD IF YOU HAVE LOTS OF SHARE COMP POWER SPINN LOTS OF MODELS IN DIFF HP

USE AN APPROPRIATE SCALE

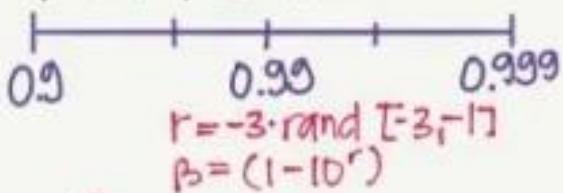
HIDDEN UNITS



α LEARNING RATE



β EXP WEIGHT AVE



TIP
RE-EVALUATE YOUR HYP. PARAMS EVERY FEW MONTHS

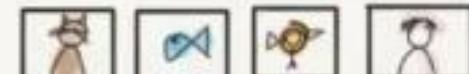
MISC. EXTRAS

BATCH NORMALIZATION

NORMALIZE LAYER OUTPUT

- SPEEDS UP TRAINING
- MAKES WEIGHTS DEEPER IN NOW MORE ROBUST (COVARIATE SHIFT)
- SLIGHT REGULARIZING EFFECT

MULTICLASS CLASSIFIC.

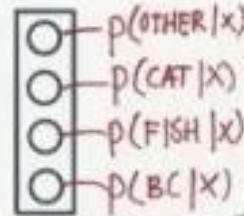


CAT FISH BABY CHICK OTHER

$$C = \# \text{ CLASSES} = 4$$

SOFTMAX ACTIVATION

$$t = e^{(z_i)} \quad a^{[i]} = \frac{t}{\sum t_i}$$



SUM: 1

$$\text{EX: } Z^{[i]} = \begin{bmatrix} 5 \\ 2 \\ -1 \\ 3 \end{bmatrix} \quad t = \begin{bmatrix} e^5 \\ e^2 \\ e^{-1} \\ e^3 \end{bmatrix} = \begin{bmatrix} 14.84 \\ 7.34 \\ 0.4 \\ 2.61 \end{bmatrix}$$

$$\Rightarrow a^{[i]} = \frac{t}{\sum t} = \begin{bmatrix} 0.892 \\ 0.042 \\ 0.002 \\ 0.064 \end{bmatrix} \rightarrow 11.4\% \text{ P(X=CAT)} \text{ P(X=BE)=2.61\%}$$

© Tessenderloer

STRUCTURING YOUR ML PROJECTS

SETTING YOUR GOAL

A GOAL SHOULD BE A SINGLE #

	PRECISION, RECALL	
A	95%	90% (90%)
B	98%	85%

IS A OR
B BEST?

	PRECISION, RECALL		F1
A	95%	90%	92.4% (92.4%)
B	98%	85%	91%

A IS
BEST

F1 = HARMONIC MEAN BETW.
RECALL & PRECISION

A DEFINE OPTIMIZING VS
SATISFYING METRICS

	ACCURACY	RUNTIME
A	90%	80ms
B	92%	95ms
C	95%	1500ms

MAXIMIZE ACC.
GIVEN TIME < 100ms

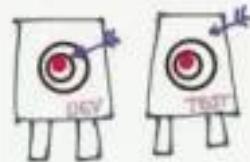
ACCURACY =
OPTIMIZING
RUNTIME =
SATISFYING

SELECTING YOUR DEV/TEST SETS

DATA

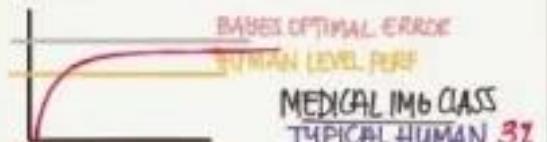
US
UK
EUROPE
S.AM
INDIA
CHINA
AUST.

OPTION 1:
DEV = UK, US, EUR
TEST = REST



IF DEV & TEST ARE DIFF
& WE OPTIMIZE FOR DEV
WE WILL MISS THE TEST-TARGET

HUMAN LEVEL PERF



WHY DOES ACC SLOW DOWN WHEN WE SURPASS HUMAN LEVEL PERF?
MEDICAL IMAGING CLASS
TYPICAL HUMAN 3%
TYPICAL DOCTOR 1%
EXPERIENCED DR. 0.7%
TEAM OF EXP DRs. 0.5%

HUMAN LEV PERF
(PROXY FOR BAYES)

- OFTEN CLOSE TO BAYES
- A HUMAN CAN NO LONGER HELP IMPROVE (INSIGHTS)
- DIFFICULT TO ANALYSE BIAS/VARIANCE

CAT CLASSIFICATION

	A	B
HUMAN	1%	7.5%
TRAIN ERR	8%	8%
DEV ERR	10%	10%

FOCUS ON BIAS
FOCUS ON VARIANCE

HUMAN TRAIN BIGGER NETW.
AVOIDABLE BIAS TRAIN LONGER/BETTER OPT. (RMSPROP)
TRAIN CHANGE NN ARCH OR HYPERPARAM
VARIANCE MORE DATA (TRAIN)
REGULARIZATION
DEV NN ARCHITECTURE

	A	B
HUMAN	0.5	0.5
TRAIN ERR	0.6	0.3
DEV ERR	0.8	0.4
AVOID. BIAS	0.1	?

AVOIDABLE BIAS
VARIANCE
DON'T KNOW IF WE OVERFIT OR IF WE'RE CLOSE TO BAYES

OPTIONS TO PROCEED ARE UNCLEAR

ERROR ANALYSIS

YOU HAVE 10% ERRORS, SOME ARE DOGS MIS-CLASSIFIED AS CATS. SHOULD YOU TRAIN ON MORE DOG PICS?

1. PICK 100 MIS-LABELED
2. COUNT ERROR REASONS

	Dog	Bunny	Insta filter	Cat	...
1	1			1	
2					1
3		1			
...					
100				1	
5	1	...			

5% OF ALL ERRORS

FOCUSING ON DOGS THE BEST WE CAN HOPE FOR IS 9.5% ERROR

YOU FIND SOME INCORR. LABELED DATA IN THE DEV SET. SHOULD YOU FIX IT?



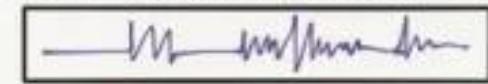
DL ALGORITHMS ARE PRETTY ROBUST TO RANDOM ERRORS. BUT NOT TO SYSTEMATIC ERR.
(EX. ALL WHITE CATS INCORRECTLY LABELED AS MICE)

ADD EXTRA CDL. IN ERROR ANALYSIS AND USE SAME CRITERIA

NOTE IF YOU FIX DEV YOU SHOULD FIX TEST AS WELL.

FOR NEW PROJ. BUILD 1ST SYSTEM QUICK & ITERATE

EX: SPEECH RECOGNITION



WHAT SHOULD YOU FOCUS ON?

NOISE
ACCENTS
FAR FROM MIKE

1. START QUICKLY DEV/TEST METRICS
2. GET TRAIN-SET
3. TRAIN
4. BIAS/VARIANCE ANAL
5. ERROR ANALYSIS
6. PRIORITIZE NEXT STEP

TRAIN vs DEV/TEST MISMATCH

AVAILABLE DATA

200k PRO CAT PICS FROM INTERNET

10k BLURRY CAT PICS FROM APP
WHAT WE CARE ABOUT

HOW DO WE SPLIT → TRAIN/DEV/TEST?

OPTION 1: SHUFFLE ALL

205k (TRAIN)

D T

25k

PROBLEM: DEV/TEST IS NOW
MOSTLY WEB/IMG (NOT REPR.
OF ENDSCENARIO)

SOLUTION: LET DEV/TEST COME
FROM APP. THEN SHUFFLE 5k
OF APP PICS TO WEB FOR TRAIN

205k

D T

WEB+APP

APP APP

BIAS & VARIANCE IN MISMATCHED TRAIN/DEV

HUMANS	~0%
TRAIN	1%
DEV ERR	10%

IS THIS DIFF
DUETO THE MODEL
NOT GENERALIZING
OR IS DEV DATA
MUCH HARDER

A: CREATE A TRAIN-DEV SET
THAT WE DONT TRAIN ON

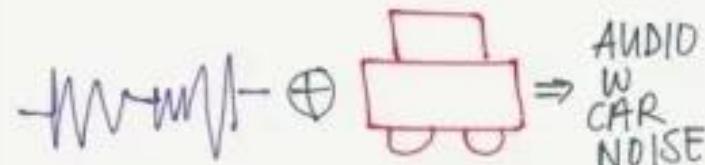
TRAIN	FD	D	T
-------	----	---	---

	A	B	C	D
TRAIN	1%	1%	10%	10%
TRAIN-DEV	9%	15%	11%	11%
DEV	10%	10%	12%	20%
VARIANCE	MISMATCH	TRAIN-DEV	BIAS	BIAS + DATA MISMATCH

ADDRESSING DATA MISMATCH

EX. CAR GPS - TRAINING DATA IS 10.000H OF GENERAL SPEECH DATA

1. CARRY OUT MANUAL ERROR ANALYSIS TO UNDERSTAND THE DIFFERENCE (EX NOISE, STREET NUMBERS)
2. TRY TO MAKE TRAIN MORE SIMILAR TO DEV OR GATHER MORE DEV-LIKE TRAIN-DATA



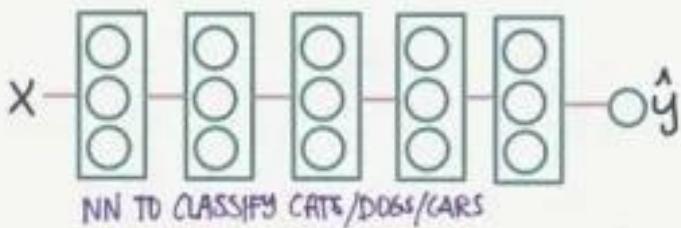
NOTE

BE CAREFUL. IF YOU ONLY HAVE 1 HR OF CAR NOISE & APPLY IT TO 10K HR SPEECH YOU MAY OVERFIT TO THE CAR NOISE

EXTENDED LEARNING

TRANSFER LEARNING

PROBLEM: YOU WANT TO CLASSIFY SOME MEDICAL IMB. YOU HAVE AN NN THAT CLASSIFIES CATS



OPTION 1: YOU ONLY HAVE A FEW RADIOLOGY IMAGES

SOLUTION: INIT W. WEIGHTS FROM CAT NN
ONLY RETRAIN LAST LAYER(S) ON RADIOLOGY IMAGES

OPTION 2: YOU HAVE LOTS OF RADIOLOGY IMB.

SOLUTION: INIT WITH WEIGHTS FROM CAT NN
RETRAIN ALL LAYERS

THIS IS MICROSOFT CUSTOM VISION

MULT TASK LEARNING

TRAINING ON MULT. TASKS AT ONCE

DETECT
CAR
STOP SIGN
PEDESTRIAN
TRAFFIC LIGHT



UNLIKE SOFTMAX - MANY THINGS CAN BE TRUE

$$\text{COST: } J(w, b) = \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^n p^{(i,j)}(y_i^{(j)}, y_j^{(i)})$$

SHANNON OVER ALL OUTP OPTIONS

WE COULD HAVE JUST TRAINED 4 NN'S INSTEAD BUT.. MT LEARNING MAKES SENSE WHEN

A. THE LEARNING DATA YOU HAVE FOR THE DIFF TASKS IS QUITE SIMILAR - & THE AMOUNTS (E.G. 1K CARS, 1K STOP SIGNS)

B. THE SUM OF THE DATA ALLOWS YOU TO TRAIN A BIG ENOUGH NN TO DO WELL ON ALL TASKS

IN REALITY TRANSFER LEARNING IS USED MORE OFTEN

END-TO-END LEARNING

FROM X-RAY OF CHILD'S HAND
TELL ME THE AGE OF THE CHILD



TYPICAL STEPS:

1. LOCATE BONES TO FIND LENGTHS USING ML
2. TRAIN MODEL TO PREDICT AGE BASED ON BONE LENGTH

END-TO-END

RADIOLOGY \longrightarrow CHILD AGE

PROS:

- LET'S THE DATA SPEAK (MAYBE IT FINDS RELATIONS WE'RE UNAWARE OF)
- LESS HAND-DESIGNING OF COMPONENTS NEEDED

CONS:

- NEEDS LARGE AMTS OF DATA ($X \rightarrow Y$)
- EXCLUDES POTENTIALLY USEFUL HAND-SHAPING INFORMATION

© Tessenderloher

CONVOLUTION

FUNDAMENTALS

COMPUTER VISION

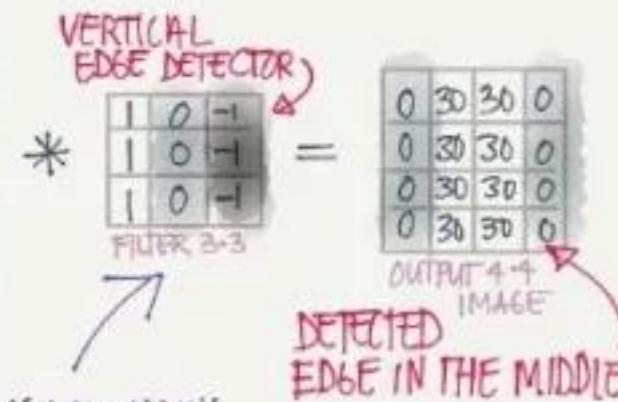
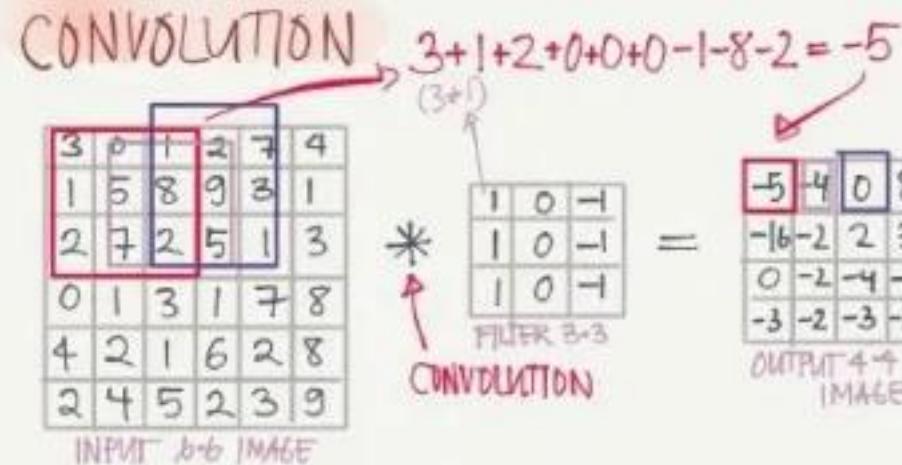
IMAGE CLASSIFICATION



OBJECT DETECTION

NEURAL
STYLE
TRANSFERPROBLEM: IMAGES CAN BE BIG

$$1000 \times 1000 \times 3 \text{ (RGB)} = 3M$$

WITH 1000 HIDDEN UNITS WE
NEED $3M \times 1000 = 3B$ PARAMSSOLUTION: USE CONVOLUTIONSIT'S LIKE SCANNING OVER YOUR
IMG WITH A MAGNIFYING GLASS
OR FILTERALSO SOLVES THE PROBLEM
THAT THE CAT IS NOT
ALWAYS IN THE SAME
LOCATION IN THE IMGTHIS IS LIKE ADDING
AN INSTA' FILTER THAT
JUST SHOWS OUTLINESWE COULD HARD-CODE FILTERS · JUST LIKE WE
CAN HARD-CODE HEURISTIC RULES ... BUT.... A MUCH BETTER
WAY IS TO TREAT THE FILTER# AS PARAMS
TO BE LEARNED

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

PADDING

PROBLEM: IMAGES SHRINK

$$6 \times 6 \rightarrow 3 \times 3 \rightarrow 4 \times 4$$

PROBLEM: EDGES GET LESS LOVE

SOLUTION: PAD w. A BORDER OF 0s BEFORE CONVOLVING

0	0	0	0	0	0	0	0
0	3	0	1	2	3	4	0
0	1	5	8	9	3	1	0
0	2	7	2	5	1	3	0
0	0	1	3	1	7	8	0
0	4	2	1	6	2	8	0
0	2	4	5	2	3	9	0
0	0	0	0	0	0	0	0

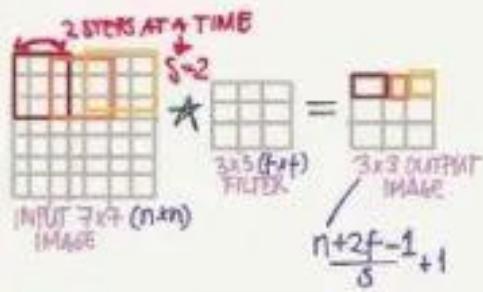
TWO COMMONLY USED
PADDING OPTIONS
(HOW MUCH TO PAD)

$$\begin{aligned} \text{'VALID'} &\Rightarrow P=0 & \text{NO PADDING} \\ \text{'SAME'} &\Rightarrow P=\frac{f-1}{2} & \text{OUTPUT SIZE = INPUT SIZE} \\ && \text{FILTER SIZE} \end{aligned}$$

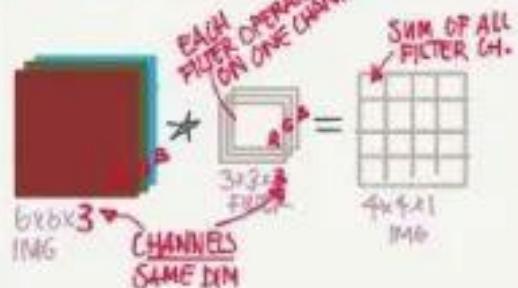
NOTE: ALL CONVOLUTION IDEAS CAN BE
APPLIED TO 1D AS WELL LIKE
ECG SIGNALS · AND 3D LIKE CT-SCANS

STRIDE

WHAT PACE YOU SCAN WITH



CONVOLUTIONS OVER VOLUMES

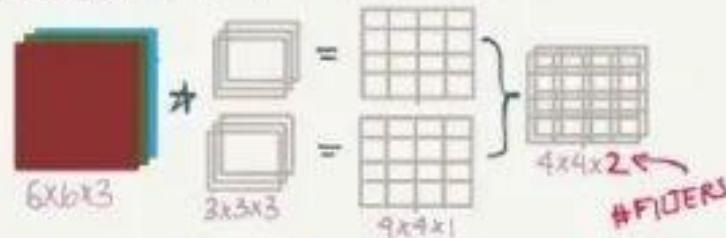


THIS ALLOWS US TO DETECT FEATURES
IN COLOR IMAGES FOR EXAMPLE

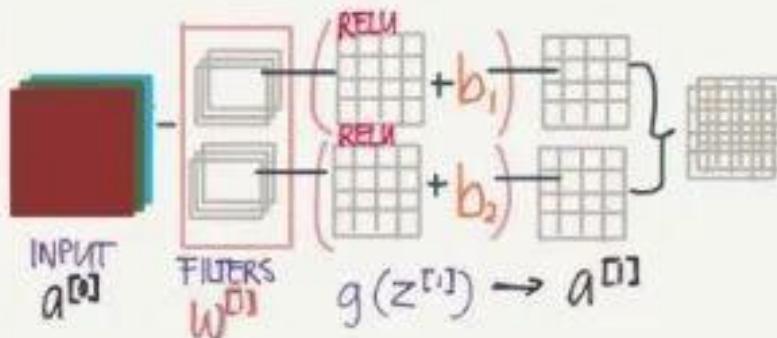
MAYBE WE WANT TO FIND ALL
EDGES OR MAYBE ORANGE BLOBS

MULTIPLE FILTERS

DETECTING MULTIPLE FEATURES AT A TIME



ONE CONV. NET LAYER

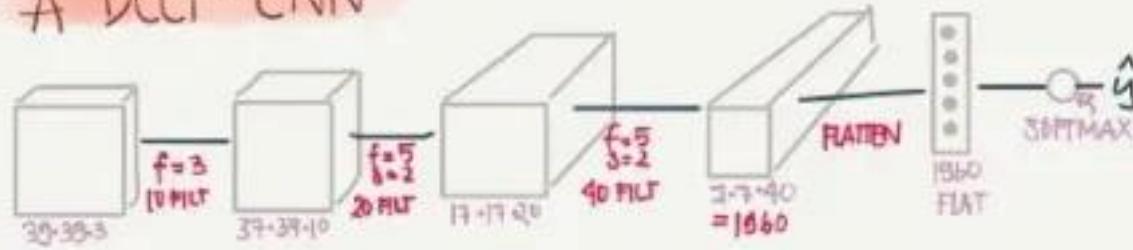


NOTE IT DOESN'T MATTER HOW BIG THE
INPUT IS - THE LEARNABLE PARAMS W & b
ONLY DEPEND ON THE # OF FILTERS
AND THEIR SIZES.

$$W = 3 \cdot 3 \cdot 3 \cdot 2 = 54 \quad \text{56 PARAMS}$$

$$b = 2 \quad \text{TO LEARN}$$

A DEEP CNN

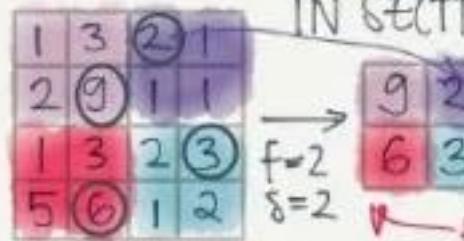


A LOT OF THE WORK IS FIGURING OUT HYPERPARAMS
 $= \# \text{FILTERS}, \text{STRIDE}, \text{PADDING} \text{ ETC}$
 TYPICALLY SIZE \rightarrow TREND DOWN
 $\# \text{FILTERS} \rightarrow$ TREND UP

TYPICAL CONVNET LAYERS

CONVOLUTION
POOLING
FULLY CONNECTED

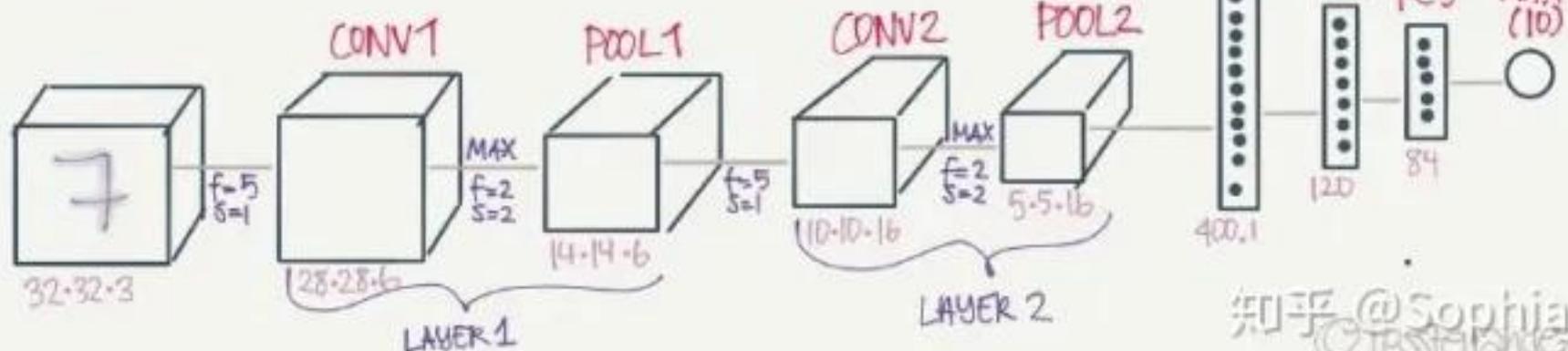
POOLING (MAX)
FIND MAX VAL IN SECTION



- * REDUCES SIZE OF REPRES.
- * SPEEDS UP COMPUTATION
- * MAKES SOME OF THE DETECTED FEAT. MORE ROBUST

CONV NET EXAMPLE BASED ON LeNet-5

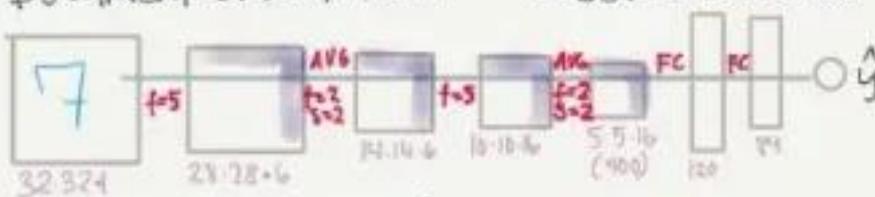
DETECTING HANDWRITTEN DIGITS



CLASSIC CONV. NETS

LeNet-5

DOCUMENT CLASSIFICATION



$\approx 60k$ PARAMETERS

TRENDS: HEIGHT/WIDTH GO DOWN
CHANNELS GO UP

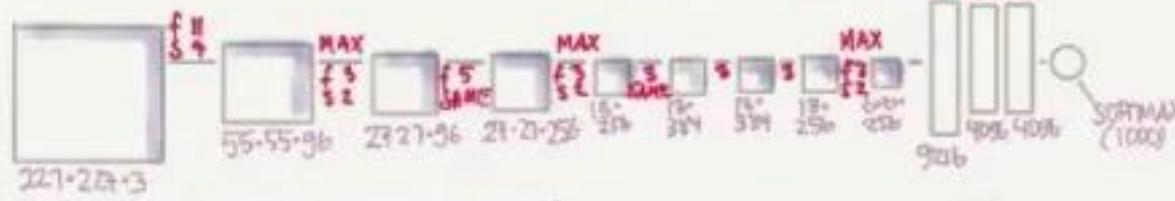
COMMON: A COUPLE OF CONV(P)POOL
PATTERN: LAYERS FOLLOWED BY A FEW FC

OLD STUFF: USED AVG POOLING INST. OF MAX
PADDING WAS NOT VERY COMMON
IT USED SIGMOID/TANH INST. OF RELU

AlexNet

IMAGE CLASSIFICATION

$\approx 60M$ PARAMETERS



- SIMILAR TO LeNet BUT MUCH BIGGER
- USES RELU
- THE NN THAT GOT RESEARCHERS INTERESTED IN VISION AGAIN

VGG-16

ALL CONV. LAYERS HAVE SAME PARAMS
 $f=3 \times 3$ $s=1$ $p=\text{SAME}$
AND POOLING LAYER 2×2 $s=2$



$\approx 138M$ PARAMETERS

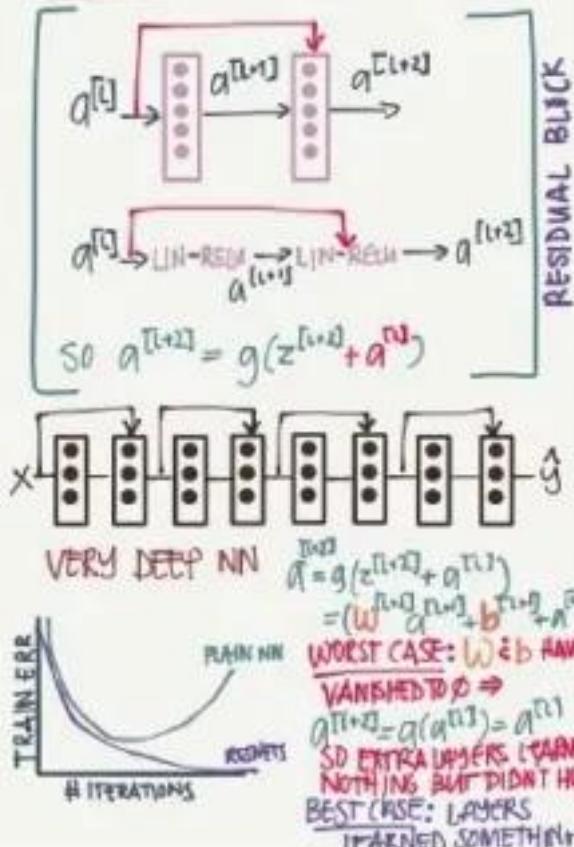
- VERY DEEP
- EASY ARCHITECTURE
- # FILTERS DOUBLE 64, 128, 256, 512

SPECIAL NETWORKS

ResNets

PROBLEM: DEEP NN OFTEN SUFFER PROBLEMS IN VANISHING OR EXPLODING GRADIENTS

SOLUTION: RESIDUAL NETS



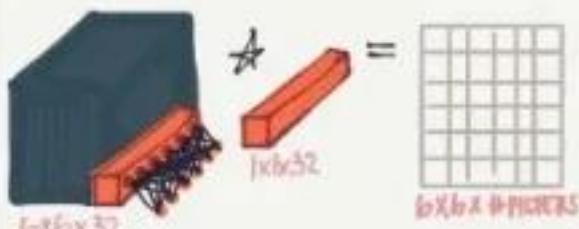
NETWORK IN NETWORK (1x1 CONVOLUTION)

$$\begin{array}{rrrr} 6 & 5 & 3 & 2 \\ 4 & 1 & 0 & 5 \\ 5 & 8 & 2 & 4 \\ 0 & 3 & 6 & 1 \end{array} \star 2 = \begin{array}{rrrr} 12 & 10 & 6 & 4 \\ 8 & 2 & 18 & 10 \\ 10 & 16 & 4 & 8 \\ 0 & 6 & 12 & 2 \end{array}$$

1x1 CONVOLUTION

IT SEEMS PRETTY USELESS, BUT IT ACTUALLY SERVES 2 PURPOSES

1. NETWORK IN A NETWORK



LEARNS COMPLEX, NON-LINEAR RELATIONSHIPS ABOUT A SLICE OF A VOLUME

2. REDUCING # CHANNELS

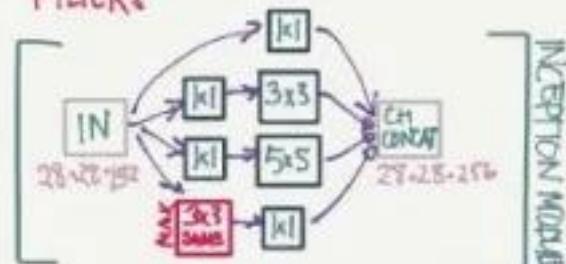
$$\begin{array}{r} 28 \cdot 28 \cdot 192 \\ \star \quad 1 \times 1 \times 92 \\ 32 \cdot \text{PMT} \end{array} = \begin{array}{r} 28 \cdot 28 \cdot 92 \end{array}$$

INCEPTION NETWORKS

INSTEAD OF CHOOSING A $1 \times 1, 3 \times 3, 5 \times 5$ OR A POOLING LAYER - CHOOSE ALL



PROBLEM: VERY EXPENSIVE TO COMPUTE
SOLUTION: SHRINK THE # CHANNELS W/ A 1x1 CONV BEFORE APPLYING ALL THE FILTERS



TO BUILD AN INCEPTION NETWORK YOU MAINLY STACK A BUNCH OF INCEPTION MODULES



INCEPTION
THE MOVIE

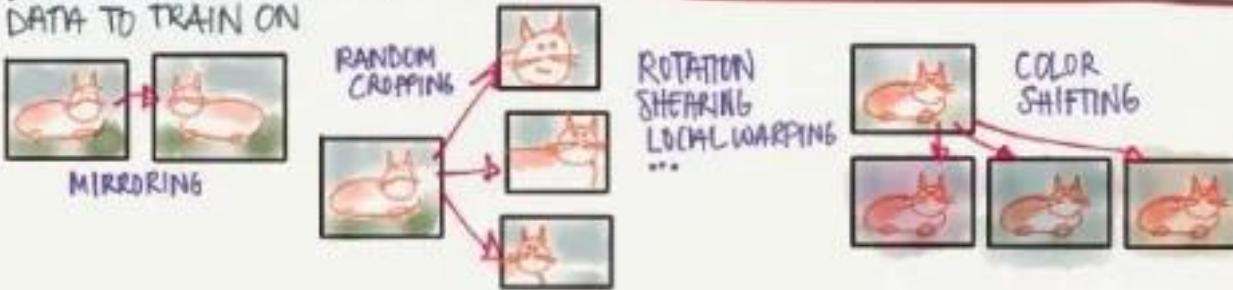
PRACTICAL ADVICE

USE OPEN SOURCE IMPLEMENTATIONS

SOME OF THE PAPERS ARE HARD TO IMPLEMENT FROM SCRATCH - USING OS YOU CAN REUSE OTHER PAPER'S WORK
DON'T FORGET TO CONTRIBUTE

DATA AUGMENTATION

WE ALMOST ALWAYS NEED MORE DATA TO TRAIN ON



TRANSFER LEARNING



I WANT TO TRAIN A CLASSIFIER FOR YOUR CATS BUT DON'T HAVE ENOUGH PICTURES

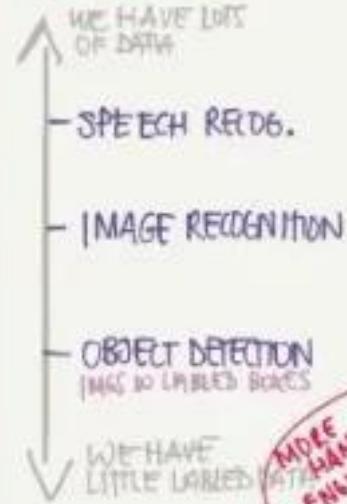
[SOLUTION] DOWNLOAD SOMEONE ELSE'S PRETRAINED NET & WEIGHTS



FREEZE THE PARAMS, AND JUST REPLACE THE SOFTMAX LAYER WITH YOUR OWN & TRAIN

IF YOU HAVE MORE PICS - RETRAIN A FEW OF THE LATER LAYERS (MAYBE INITIALIZING WITH THE PRETRAINED WEIGHTS)

STATE OF COMPUTER VISION

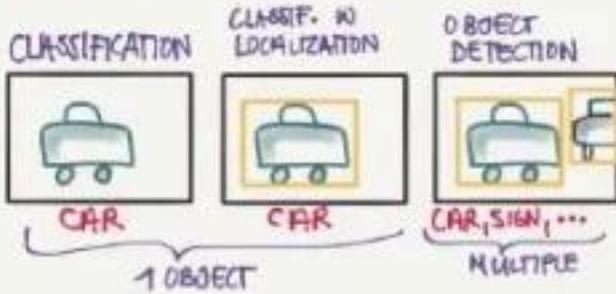


TIPS FOR DOING WELL ON BENCHMARKS/COMPETITIONS

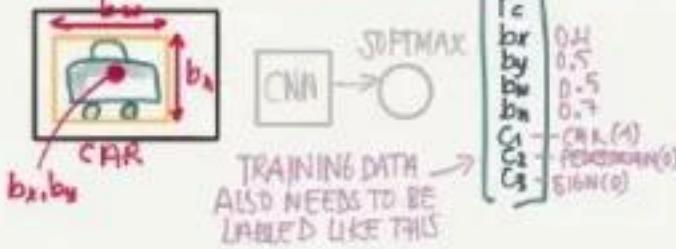
- * ENSEMBLING: AVG OUTPUTS FROM MULT NN
- * MULTI-CROP AT TEST TIME: AVG OUTPUTS FROM MULTIPLE CROPS OF THE IMAGE

IN PRACTICE THEY ARE NOT USED IN PRODUCTION BECAUSE THEY ARE COMPUT & MEM EXPENSIVE

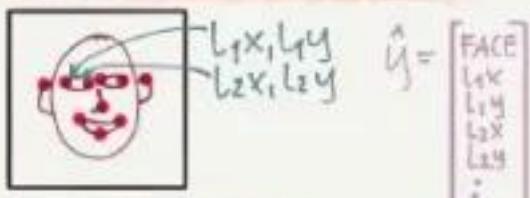
DETECTION ALGORITHMS



OBJECT LOCALIZATION



LANDMARK DETECTION



TO DETECT LANDMARKS IN THE FACE (CORNER OF MOUTH ETC) LABEL THE X, Y COORDS OF THE LANDMARK

USED FOR SENTIMENT ANALYSIS & FOR EFFECTS LIKE PLACING CROWN ON HEAD ETC.

SLIDING WINDOWS DETECTION



1. CREATE TIGHTLY CROPPED IMGS OF CARS (LOTS)
2. SLIDE A WINDOW OVER THE IMG. & CLASSIFY THIS WINDOW CAR (1/0) AGAINST YOUR OTHER CARS
3. REPEAT WITH SLIGHTLY LARGER WINDOW SIZE

PROBLEM: VERY EXPENSIVE (TO COMPUTE)

SINCE ADJ WINDOWS SHARE A LOT OF THE COMPUTATIONS WE CAN DO THIS MUCH CHEAPER IN CONVOLUTIONS



NOW WE JUST PASS THROUGH ONCE AND CALL ALL AT THE SAME TIME

END OF THE 4 VALS ARE RESULTS FOR EACH OF THE 4 WINDOWS

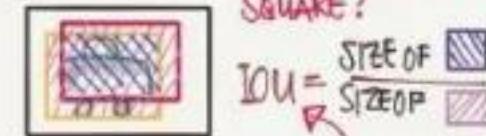
YOLO · You Only Look Once

1. SPLIT IMG INTO $X(g)$ GRID CELLS
2. FOR EACH CELL, SAY IF IT CONTAINS CAR + BOUNDING BOX (IF CELL CONTAINS SAME AS OBJECT LOCALIZATION)

$$X \times g = 3 \times 3 \times 8$$

HOW DO YOU KNOW HOW GOOD IT IS?

HOW GOOD IS THE RED SQUARE?



INTERSECTION OVER UNION

GENERALLY · IF $IoU \geq 0.5$ IT IS REGARDED AS CORRECT

WHAT IF MULTIPLE SQUARES CLAIM THE SAME CAR?

NON-MAX SUPPRESSION

IF TWO BOUNDING BOXES HAVE A HIGH IOU - PICK THE ONE W HIGHEST P_c - GET RID OF THE REST.

ANCHOR BOXES

ANCHOR BOXES LET YOU ENCODE MULTIPLE OBJECTS IN THE SAME SQUARE

FACE RECOGNITION

FACE
VERIFICATION



99% ACC \Rightarrow
PRETTY GOOD

FACE
RECOGNITION



IF $K = 100$ NEED
MUCH HIGHER THAN
99%

ONE SHOT LEARNING

NEED TO BE ABLE TO RECOGNISE
A PERSON EVEN THOUGH YOU ONLY
HAVE ONE SAMPLE IN YOUR DB.
YOU CAN'T TRAIN A CNN WITH
A SOFTMAX (EACH PERSON) BECAUSE

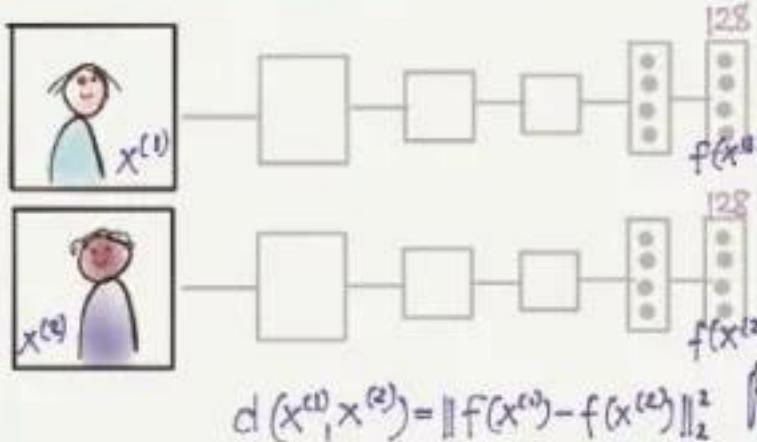
- YOU DON'T HAVE ENOUGH SAMPLES
- IF A NEW PERSON JOINS YOU
NEED TO RETRAIN THE NETWORK

SOLUTION LEARN A SIMILARITY
FUNCTION

$$d(\text{img1}, \text{img2}) = \text{degree of difference}$$

BUT HOW DO YOU LEARN THIS?

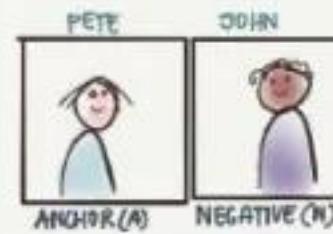
SIAMESE NETWORK DeepFace



LEARN THE PARAMS OF
THE NN SUCH THAT
 - IF $x^{(1)}, x^{(2)}$ ARE THE SAME
PERSON $\cdot d(x^{(1)}, x^{(2)}) \Rightarrow$ SMALL
 - IF $x^{(1)}, x^{(2)}$ ARE DIFFERENT
PEOPLE $\cdot d(x^{(1)}, x^{(2)}) \Rightarrow$ LARGE

WE CAN ACCOMPLISH
THIS WITH THE TRIPLET
LOSS FUNCTION

TRIPLET LOSS FaceNet



$$\text{WANT } \|f(A) - f(P)\|^2 \leq \|f(A) - f(N)\|^2 \Rightarrow d(A, P) - d(A, N) \leq 0$$

$d(A, P)$ $d(A, N)$

BUT WE WANT A GOOD MARGIN, SO...

$$d(A, P) - d(A, N) + \alpha \leq 0$$

HOW DO WE CHOOSE TRIPLETS
TO TRAIN ON?

- IF A/P ARE VERY SIMILAR, & A/N ARE VERY DIFFERENT
TRAINING IS VERY EASY.

SELECT A/N THAT ARE PRETTY SIMILAR TO TRAIN A GOOD NET

SOME BIG COMPANIES
HAVE ALREADY TRAINED
NETWORKS ON LARGE
AMTS OF PHOTOS SO
YOU MAY JUST
WANT TO REUSE
THEIR WEIGHTS

NEURAL STYLE TRANSFER



WE CAN VISUALIZE WHAT A NETWORK LEARNS BY LOOKING AT WHAT IMAGES (PATS) ACTIVATED EACH UNIT MOST



BUT HOW DOES THIS HELP US GENERATE AN IMAGE IN THE STYLE OF ANOTHER?

IDEA:

1. GENERATE A RANDOM IMAGE
2. OPTIMIZE THE COST FUNCTION

$$J(G) = \alpha J_{\text{CONTENT}}(C, G) + \beta J_{\text{STYLE}}(S, G)$$

α HOW SIMILAR ARE C & G
 β HOW SIMILAR ARE S & G

3. UPDATE EACH PIXEL

CONTENT COST FUNCTION

- USE A PRE-TRAINED CONVNET (EX VGG)
- SELECT A HIDDEN LAYER SOMEWHERE IN THE MIDDLE
LATER \Rightarrow COPIES LARGER FEATURES
- LET $a^{(1)(C)}$ & $a^{(1)(G)}$ BE THE ACTIVATIONS
- IF $a^{(1)(C)} \in a^{(1)(G)}$ ARE SIMILAR THEY HAVE SIMILAR CONTENT
BECAUSE THEY BOTH TRIGGER THE SAME HIDDEN UNITS

HOW DO WE TELL IF THEY ARE SIMILAR?

$$J_{\text{CONTENT}}(C, G) = \frac{1}{2} \| a^{(1)(C)} - a^{(1)(G)} \|_F^2$$

CAPTURING THE STYLE



USING THE STYLE IMAGE AND THE ACTIVATIONS IN A LAYER.
LOOK THROUGH THE ACTIVATIONS IN THE DIFFERENT CHANNELS TO SEE HOW CORRELATED THEY ARE

WHEN WE SEE PATTERNS LIKE THIS DO WE USUALLY SEE IT WITH PATCHES LIKE THESE?



STYLE MATRIX

CREATE A MATRIX OF HOW CORRELATED THE ACTIVATIONS ARE, FOR EACH POS (x, y) & CHANNEL PAIR (k, k') FOR THE STYLE IMG & GENERATED

$$G_{kk'} = \sum_{i=1}^n \sum_{j=1}^m a_{ijk} \cdot a_{ijk'}$$

THE STYLE COST FUNCTION

$$J(S, G) = \| G^{(S)} - G^{(G)} \|_F^2$$

FROBENIUS NORM

TO GET MORE VISUALLY PLEASING IMAGES IF YOU CALC $J(S, G)$ OVER MULTIPLE LAYERS



NEURAL STYLE TRANSFER



WE CAN VISUALIZE WHAT A NETWORK LEARNS BY LOOKING AT WHAT IMAGES (PARTS) ACTIVATED EACH UNIT MOST



BUT HOW DOES THIS HELP US GENERATE AN IMAGE IN THE STYLE OF ANOTHER?

IDEA:

1. GENERATE A RANDOM IMAGE
2. OPTIMIZE THE COST FUNCTION

$$J(G) = \alpha \underset{\text{CONTENT}}{\underset{C \in G}{\text{SIMILARITY}}} J(C, G) + \beta \underset{\text{STYLE}}{\underset{S \in G}{\text{SIMILARITY}}} J(S, G)$$

3. UPDATE EACH PIXEL

CONTENT COST FUNCTION

- USE A PRE-TRAINED CONVNET (EX VGG)
- SELECT A HIDDEN LAYER SOMEWHERE IN THE MIDDLE
LATER \Rightarrow COPIES LARGER FEATURES
- LET $a^{(l)(c)}$ & $a^{(l')(c')}$ BE THE ACTIVATIONS
- IF $a^{(l)(c)} \approx a^{(l')(c')}$ ARE SIMILAR THEY HAVE SIMILAR CONTENT
BECAUSE THEY BOTH TRIGGER THE SAME HIDDEN UNITS

HOW DO WE TELL IF THEY ARE SIMILAR?

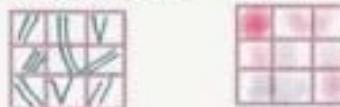
$$J_{\text{CONTENT}}(C, G) = \frac{1}{2} \| a^{(l)(c)} - a^{(l')(c')} \|_F^2$$

CAPTURING THE STYLE



USING THE STYLE IMAGE AND THE ACTIVATIONS IN A LAYER.
LOOK THROUGH THE ACTIVATIONS IN THE DIFFERENT CHANNELS TO SEE HOW CORRELATED THEY ARE

WHEN WE SEE PATTERNS LIKE THIS DO WE USUALLY SEE IT WITH PATCHES LIKE THESE?



STYLE MATRIX

CREATE A MATRIX OF HOW CORRELATED THE ACTIVATIONS ARE, FOR EACH POS (x, y) & CHANNEL PAIR (k, k') FOR THE STYLE IMG & GENERATED

$$G_{kk'} = \sum_{i=1}^n \sum_{j=1}^m a_{ijk} \cdot a_{ijk'}$$

THE STYLE COST FUNCTION

$$J(S, G) = \| G^{(S)} - G^{(G)} \|_F^2$$

FROBENIUS NORM

TO GET MORE VISUALLY PLEASING IMAGES IF YOU CALC $J(S, G)$ OVER MULTIPLE LAYERS



RECURRENT NEURAL NETWORKS

SEQUENCE PROBLEMS

IN	OUT	PURPOSE
Mr. Potato	THE QUICK BROWN FOX JUMPED...	SPEECH RECOGNITION
∅	♪ ♪ ♪ ♪ ♪ ♪	MUSIC GENERATION
THERE IS NOTHING TO SEE IN THIS MOVIE	★ ★ ★ ★	SENTIMENT CLASSIFICATION
AGCCCGCTTG AGGAACATG	AGCCCGCTTG AGGAACATG	DNA SEQUENCE ANALYSIS
You're voice changes and now?	Do you want to sing with me?	MACHINE TRANSLATION
🏃 🏃 🏃	RUNNING	VIDEO ACTIVITY RECOGNITION
Yesterday Harry Potter met Hermione Granger	Yesterday Harry Potter met Hermione Granger	NAME ENTITY RECOGNITION

NAME ENTITY RECOGNITION

$x = \text{HARRY POTTER AND HERMIONE}$ $T_x = 9$
 $x^{<1>} x^{<2>} \dots$ (9 words)

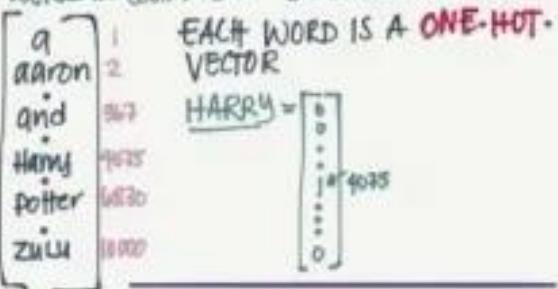
GRANGER INVENTED A NEW SPELL

$$y = \begin{matrix} 1 & 1 & 0 & 1 \\ y^{<1>} & y^{<2>} & \dots & T_y = T_x \\ 1 & 0 & 0 & 0 \end{matrix}$$

EXAMPLE OF A PROBLEM WHERE
EVERY $x^{<i>}$ HAS AN OUTPUT $y^{<i>}$

HOW DO WE REPRESENT WORDS?

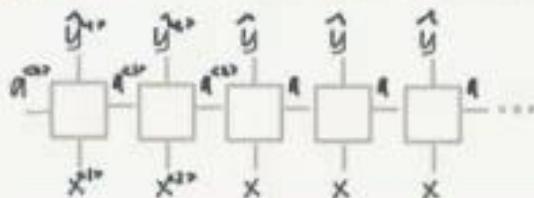
CREATE A VOCABULARY (EG 10K MOST COMMON WORDS IN YOUR TEXTS · OR DOWNLOAD EXISTING)



WE COULD USE A STANDARD NETWORK BUT...

- A INPUT & OUTPUTS CAN HAVE DIFFERENT LENGTHS IN DIFF EXAMPLES
- B WE DON'T SHARE FEATURES LEARNED ACROSS DIFFERENT POSITIONS

RECURRENT NEURAL NET (RNN)



PREVIOUS RESULTS ARE PASSED IN AS INPUTS SO WE GET CONTEXT.

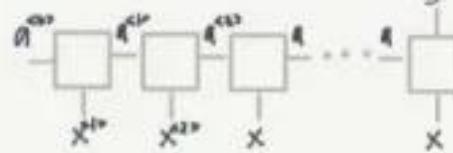
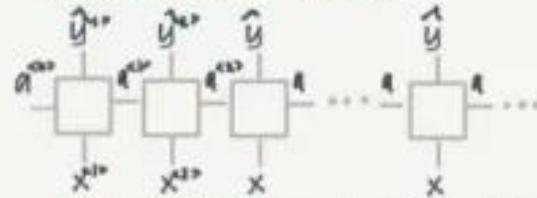
$$\begin{aligned} a^{<1>} &= g_1(W_a[x^{<0>} x^{<1>}] + b_a) \quad \text{TANH / RELU} \\ a^{<2>} &= g_2(W_a a^{<1>} + b_a) \quad \text{SIGMOID} \end{aligned}$$

THE SAME W & b ARE USED IN ALL TIME STEPS

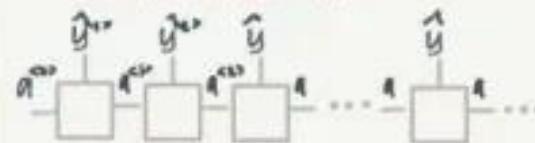
THE LOSS WE OPTIMIZE IS THE SUM OF $l(y_i, \hat{y}_i)$ FROM $1-T$

DIFFERENT TYPES OF RNN

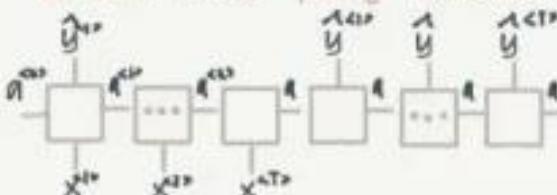
MANY-TO-MANY $T_x = T_y$



ONE-TO-MANY ↗ MUSIC GENERATION



MANY-TO-MANY $T_x \neq T_y$ ↗ TRANSLATION



NLP & WORD EMBEDDINGS

MAN IS TO WOMAN AS
KING IS TO QUEEN

PROBLEM: THE ONE-HOT REPR. OF
APPLE HAS NO INFO ABOUT ITS RELATIONSHIP
TO ORANGE

I WANT A GLASS OF ORANGE —
I WANT A GLASS OF APPLE —

SOLUTION: CREATE A MATRIX OF
FEATURES TO DESCRIBE THE WORDS

WORD EMBEDDINGS

	MAN	WOMAN	KING	QUEEN	APPLE	ORANGE
	53%	3833	494	3157	456	6294
GENDER	-1	1	-0.5	0.97	0.0	0.01
ROYAL	0.01	0.02	0.95	0.95	-0.01	0.00
AGE	0.03	0.02	0.7	0.369	0.016	-0.02
FOOD	0.04	0.01	0.02	0.01	0.95	0.91
...						
C	5351					

IN REALITY, THE FEATURES ARE
LEARNED & NOT AS STRAIGHTFORWARD
AS GENDER/AGE

	MAN	WOMAN	KING	QUEEN	APPLE	ORANGE
	• man	• woman	• dog	• cat	• apple	• orange
	• king	• queen	• cat	• fish	• grape	• orange
	• four	• three	• man	• king	• man	• orange
	• three	• one	• man	• king	• man	• orange
	• two	• one	• man	• king	• man	• orange

t-SNE
VISUAL
REPRESENT
OF 300D
WORD
EMBEDDINGS

USING WORD EMBEDDINGS

EX. NAME/ENTITY RECOGN



WITH WORD EMBEDDINGS WE
UNDERSTAND THAT AN ORANGE
FARMER IS A PERSON = SALLY
SMITH = NAME

- APPLE ~ ORANGE → PERSON
- USING WORD EMBEDDINGS TRAINED
ON LOTS OF TEXT WE ALSO GET EMB.
FOR MORE UNCOMMON WORDS
(CHILDREN, CHIEFTAIN)

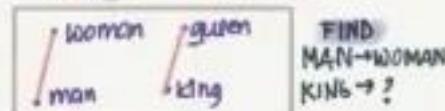
EX. MAN IS TO WOMAN AS
KING IS TO ?

E = MATRIX

Man	1	-0.5	0.97	0.0	0.01
Woman	0.01	0.02	0.95	0.95	-0.01
King	0.03	0.02	0.7	0.369	0.016
Queen	0.04	0.01	0.02	0.01	0.95
...					

e_{man} - e_{woman} = e_{king} - e_{queen}

$$\begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{EVERY } \rightarrow \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



FIND(w):

$$\text{ARG-MAX SIM}(e_w, e_{\text{king}} - e_{\text{man}} + e_{\text{woman}})$$

$$\text{SIM}(u, v) = \frac{u \cdot v}{\|u\| \|v\|}$$

COSINE
SIMILARITY

LEARNING WORD EMBEDDINGS

HOW DO WE LEARN THE EMBEDDING MATRIX E?

IDEA 1: USING A NEURAL LANG MODEL

I WANT A GLASS OF ORANGE \hat{y}



WE CAN HAVE DIFFERENT CONTEXTS THAN THE LAST 4 WORDS

- LAST 4 WORDS
- 4 WORDS LEFT+RIGHT
- LAST 1 WORD
- NEARBY 1 WORDS

SKIPGRAM

RANDOM WITHIN EX 5 WORDS

IDEA 2: SKIP-GRAM WORD2VEC

I WANT A GLASS OF ORANGE JUICE TO GO ALONG WITH MY CEREAL
PICK RANDOM CONTEXT/TARGET PAIRS (WITHIN EX 5 WORDS)

CONTEXT	TARGET
ORANGE	JUICE
ORANGE	GLASS
ORANGE	MILK
...	...

$$O_c \rightarrow E \rightarrow e_c \rightarrow \underset{\text{softmax}}{\circ} \rightarrow \hat{y}(O_t)$$

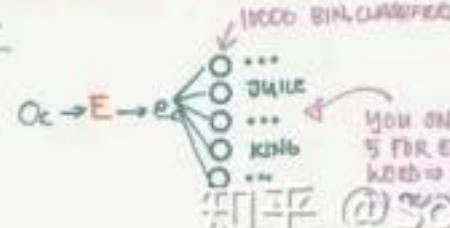
NOTE: WHILE THIS
SIMPLE NN PREDICTS O_T
OUR REAL GOAL IS TO
LEARN E

THIS IS VERY COMPUTATIONALLY EXPENSIVE
BUT WE CAN OPTIMIZE BY USING A HIERARCHICAL
SOFTMAX CLASSIFIER

IDEA: NEGATIVE SAMPLING

- PICK A CONTEXT/TARGET PAIR AS A POSITIVE EXAMPLE
- PICK A FEW NEG EXAMPLES (CONTEXT + RANDOM)

CONTEXT	WORD	TARGET
ORANGE	JUICE	1
ORANGE	KING	0
FRUIT	BARK	0
ORANGE	THE	0
ORANGE	OF	0



NOTE: SOMETIMES BY
CHANCE YOU PICK A
POS PAIR BUT IT DOESN'T
MATTER.

YOU ONLY TRAIN
5 FOR EACH CONTEXT
WORD = EFFICIENT

@Sophie
@TeachCraze

WORD EMBEDDINGS

CONTINUED...

GloVe WORD VECTORS

$X_{ij} = \text{# TIMES WORD } i \text{ APPEARS IN THE CONTEXT OF } j$
 (HOW RELATED THEY ARE)

$$\text{MINIMIZE } \sum_{i=1}^n \sum_{j=1}^m f(x_{ij})(\theta_i^T e_j + b_i + b_j - \log x_{ij})^2$$

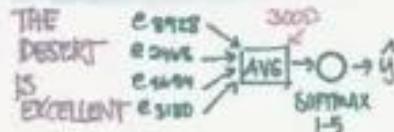
IF IT IS NOT COMMON
 (YOU CAN'T USE IT VERY FEW WORDS (THE, OF...))
 & IT'S INFREQUENT (CHARACTERS)

EVERYTHING LED UP TO THIS VERY SIMPLE ALGORITHM

SENTIMENT CLASSIFICATION

x	y
THE DESSERT IS EXCELLENT	1
SERVICE WAS SHITE SLOW	-1
GOOD FOR A QUICK MEAL BUT NOTHING SPECIAL	0.5
COMPLETELY LACKING IN GOOD TASTE, GOOD SERVICE AND GOOD AMBIENCE	-1
PROBLEM: YOU MAY NOT HAVE A LARGE DATASET BUT YOU CAN USE AN EMBEDDING MATRIX E THAT IS ALREADY PRE-TRAINED	

IDEA: SIMPLE CLASSIFICATION

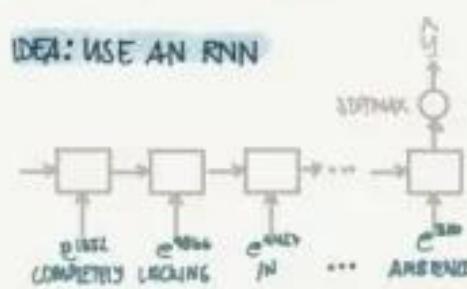


WORKS WELL FOR SHORT SENTENCES
 BUT DOESN'T TAKE ORDER INTO ACCOUNT

"COMPLETELY LACKING IN GOOD TASTE,
 GOOD SERVICE AND GOOD AMBIENCE"

THIS MAY BE SEEN AS A -1 REVIEWS

IDEA: USE AN RNN



THIS CAN NOW TAKE INTO ACCOUNT THAT COMPLETELY LACKING NEGATES THE WORD GOOD

ELIMINATING BIAS IN WORD EMBEDDINGS

MAN IS TO COMPUTER PROGRAMMER AS WOMAN IS TO HOME MAKER

SOMETIMES THE TEXT CONTAINS - C ALSO LEARN A GENDER, RACE, AGE... BIAS WE DON'T WANT OUR MODELS TO HAVE - EX. HIRING BASED ON GENDER, SENTENCING BASED ON RACE ETC.

ADDRESSING BIAS

1. IDENTIFY BIAS DIRECTION

$$\begin{cases} \text{man} \rightarrow \text{woman} \\ \text{male} \rightarrow \text{female} \end{cases}$$

2. NEUTRALIZE

FOR EVERY WORD THAT IS NOT DEFINITIONAL (GIRL, BOY, HE, SHE...) PROJECT TO GET RID OF BIAS

3. EQUALIZE PAIRS

THE ONLY DIFF BETWEEN EX GIRL/BM SHOULD BE GENDER

HOW DO YOU KNOW WHICH WORDS TO NEUTRALIZE?

DOCTOR, BEARD, SEWING MACHINE?

A: BY TRAINING A CLASSIFIER TO FIND OUT IF A WORD IS DEFINITIONAL

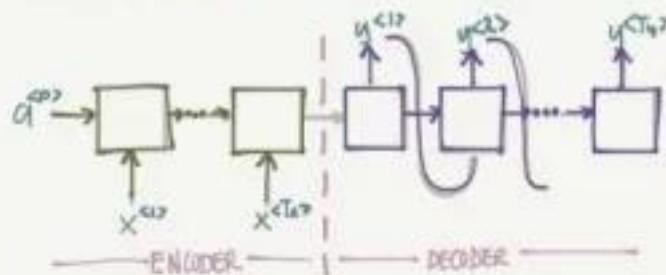
URNS OUT THE # OF PAIRS IS FAIRLY SMALL SO YOU CAN EVEN HAND PICK THEM

知乎 @Sophia
 @TeachCramz

SEQUENCE TO SEQUENCE

BASIC MODELS

JANE VISITE L'AFRIQUE → JANE IS VISITING AFRICA
EN SEPTEMBRE IN SEPTEMBER



→ THIS IS A CAT
ON A CHAIR.

CNN → RNN

HOW DO YOU PICK THE MOST LIKELY SENTENCE?

$$P(y^{<1}, \dots, y^{<t} | x)$$

WE DON'T WANT A RANDOMLY GENERATED SENTENCE
(WE WOULD SOMETIMES GET A GOOD, SOMETIMES BAD)
INSTEAD WE WANT TO MAXIMIZE

$$\text{ARG MAX } P(y^1, \dots, y^{<t} | x)$$

IDEA: USE GREEDY SEARCH

1. PICK THE WORD WITH THE BEST PROBABILITY
2. REPEAT UNTIL DEAD

WITH THIS WE COULD GET

- JANE IS GOING TO BE VISITING AFRICA
THIS SEPTEMBER

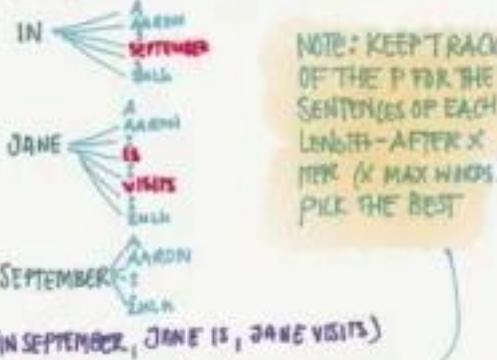
INSTEAD OF

- JANE IS VISITING AFRICA THIS SEPTEMBER

SOLUTION OPTIMIZE THE PROB OF THE WHOLE SENTENCE INSTEAD

BEAM SEARCH

1. PICK THE FIRST WORD
PICK THE B (EX: 3) BEST ALTERNATIVES
(IN, JANE, SEPTEMBER)
2. FOR EACH B WORDS PICK THE NEXT WORD
AND EVALUATE THE PAIRS TO END UP IN B PAIRS
 $P(y^{<1}, y^{<2} | x) = P(y^{<1} | x) P(y^{<2} | x, y^{<1})$



3. REPEAT TIL DONE

$$\text{ARG MAX } \prod_{i=1}^t P(y^{<i} | x, y^{<1}, \dots, y^{<i-1})$$

OVERFLOWS

PROBLEM: MULTIPLYING PROBABILITIES $(O(K^{T+1}))$
RESULTS IN A VERY SMALL NUMBER

PROBLEM II: IF WE OPTIMIZE FOR THE MULT
WE WILL PREFER SHORT SENTENCES. SINCE
EACH WORD WILL REDUCE PROB

INSTEAD WE CAN OPTIMIZE FOR THIS

$$\frac{1}{T_y} \sum_{t=1}^T \log(P(y^{<t} | x, y^{<1}, \dots, y^{<t-1}))$$

HOW DO WE PICK B?

LARGE B: BETTER RESULT, SLOWER
SMALL B: WORSE RESULT, BETTER

IN PRD YOU MIGHT SET B=10.
100 IS PROBABLY A BIT TOO HIGH -
BUT IT'S DOMAIN DEPENDENT

ERROR ANALYSIS IN BEAM S.

HUMAN: JANE VISITS AFRICA IN SEPT...
ALSO: JANE VISITED AFRICA LAST SEPTEMBER

HOW DO WE KNOW IF ITS OUR RNN
OR OUR BEAM SEARCH WE SHOULD
WORK ON?

LET THE RNN GIVE $P_{\text{RNN}} = P(y^1 | x) \cdot P_{\text{RNN}}^2 = P(y^1 | x) \cdot P_{\text{RNN}}^2$

IF $P_{\text{RNN}} > P_{\text{BS}}$:

BEAM PICKED THE WRONG ONE
TRY A HIGHER B

ELSE:

THE RNN PICKED THE WRONG
PREFERRED WORDS (BY THE RNN)
JUNIPER @Sophia
@TessCarranza

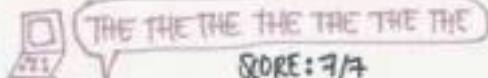
SEQUENCE TO SEQUENCE

FRENCH: LE CHAT EST SUR LE TAPIS
 HUMAN1: THE CAT IS ON THE MAT
 HUMAN2: THERE IS A CAT ON THE MAT

HOW DO YOU EVALUATE THE MACHINE TRANSLATION WHEN MULTIPLE ARE RIGHT?

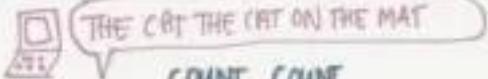
BLEU SCORE

IDEA: CHECK IF THE WORDS APPEAR IN THE REAL TRANSLATION



IDEA: ONLY GIVE CREDIT FOR A WORD THE MAX # TIMES IT APPEARS IN A TARGET SENTENCE

SCORE: 2/7 COUNT CLIP



	COUNT	COUNT CLIP	
THE	2	1	
CAT	1	0	
THE	1	1	
ON	1	1	
THE	1	1	
MAT	1	1	
BI-GRAMS	6	4/6	

COMBINED BLEU SCORE

$$\text{BP} \cdot \exp\left(\frac{1}{4} \sum_{n=1}^4 p_n\right)$$

p_i = SCORE SINGLE WORD
 p_1 = SCORE BIGRAMS

...
 BP = BREVITY PENALTY

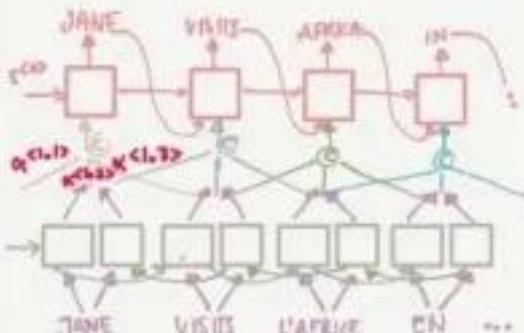
PENALIZES SENTENCES SHORTER THAN THE TARGET
 DR

A USEFUL SINGLE NUMBER EVAL METRIC

ATTENTION MODEL



SOLUTION: TRANSLATE A LITTLE AT A TIME USING ONLY PARTS OF THE SENTENCE AS CONTEXT

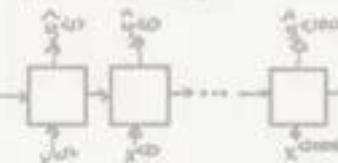


$$\alpha^{<t, t'} = \text{HOW MUCH ATTENTION } y^{t'} \text{ SHOULD PAY TO } x^{t'}$$

$$C^{<2>} = \sum_{t'} \alpha^{<2, t'} \cdot \alpha^{<t>} \quad \sum_t \alpha^{<1, t>} = 1$$

α IS CALCULATED USING A SMALL NEURAL NETWORK $\text{SET-1} \xrightarrow{\alpha^{<t, t'}} \text{SET-2} \xrightarrow{\alpha^{<t>}} \alpha^{<t>} = \frac{\exp(e^{<t, t'}))}{\sum_t \exp(e^{<t, t'}))}$

SPEECH RECOGNITION



PROBLEM: 10s CLIP AT 100Hz = 1000 INPUTS BUT ONLY ≈ 20 OUTPUTS

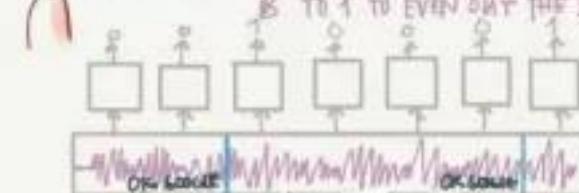
SOLUTION: USE CTC COST (CONNECTION TEMPORAL CLASSIFICATION)

m-i-h-a-n-t--u---g-g-i--g

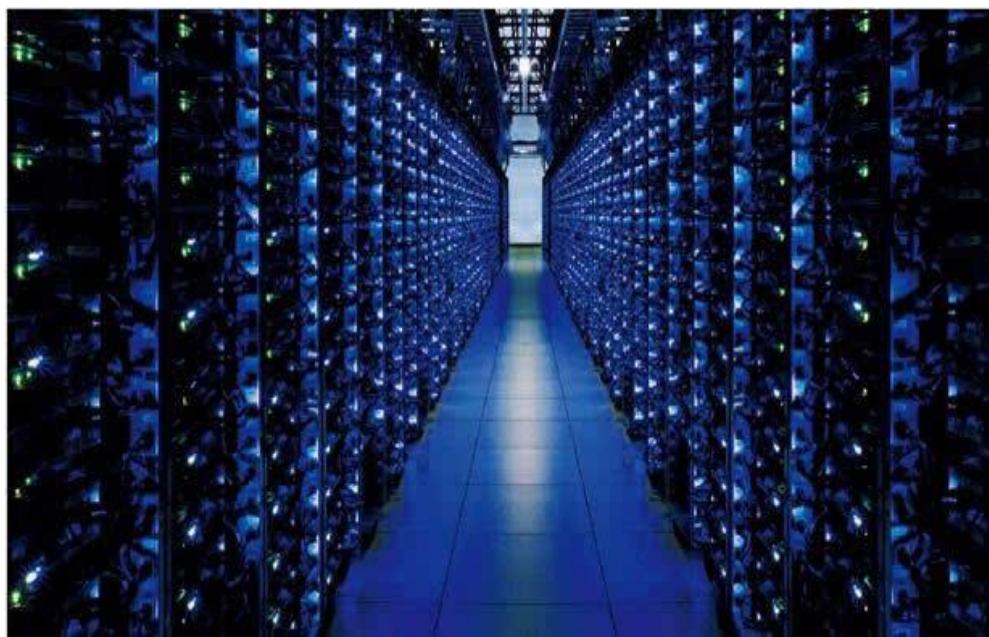
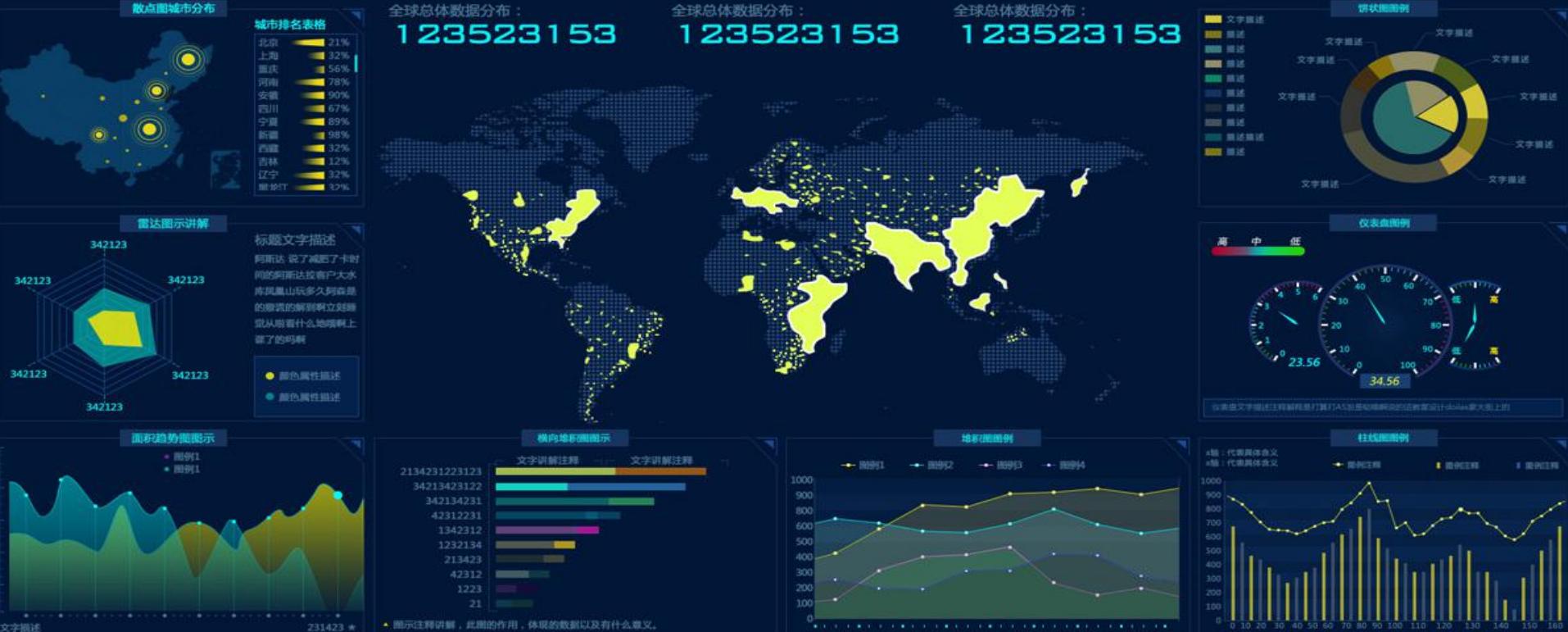
COLLAPSE REPEATED CHAR NOT SEP BY BUNK

TRIGGER WORD DETECTION

COULD SET A FEW SUBSEQUENT TO 1 TO EVEN OUT THE DS



OK GOOGLE @Sophia @TessCorrazez



什么是大数据？



大数据特征定义



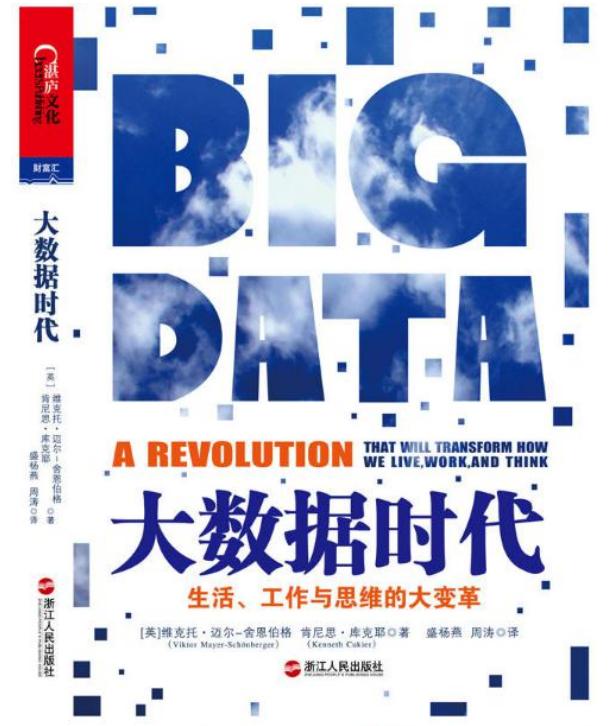
大数据时代要具备大数据思维

维克托·迈尔-舍恩伯格认为：

- 1-需要全部数据样本而不是抽样；
- 2-关注效率而不是精确度；
- 3-关注相关性而不是因果关系。

大数据并不在“大”，而在于“有用”。

价值含量、挖掘成本比数量更为重要。



大数据的价值所在？

- 如果把大数据比作一种产业，那么这种产业实现盈利的关键，在于提高对数据的“加工能力”，通过“加工”实现数据的“增值”。
- 未来在大数据领域最具有价值的是两种事物：



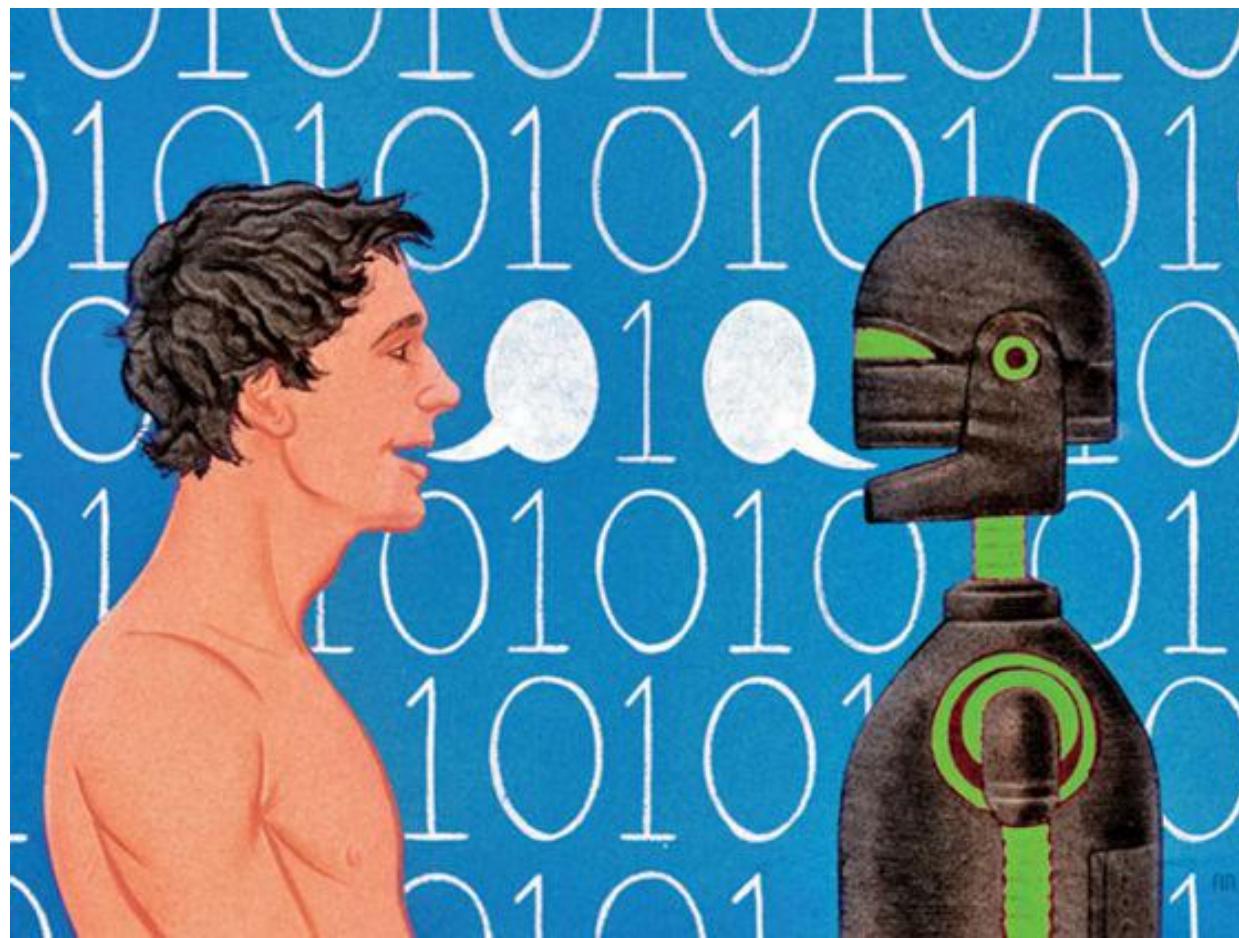
云计算和大数据的关系



云计算思想：把计算能力作为一种像水和电一样的公用事业提供给用户。

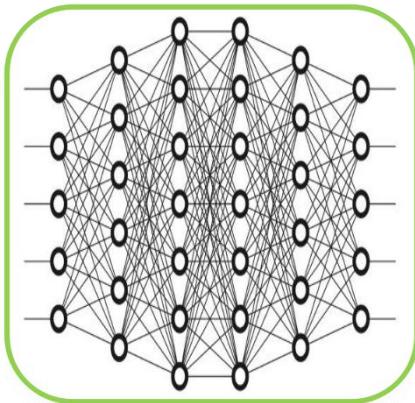
云计算充当了工业革命时期的发动机的角色，而大数据则是电。

机器学习、深度学习和人工智能





- 任何通过数据训练的学习算法的相关研究都属于机器学习。比如线性回归（Linear Regression）、K均值（K-means，基于原型的目标函数聚类方法）、决策树（Decision Trees，运用概率分析的一种图解法）、随机森林（Random Forest，运用概率分析的一种图解法）、PCA（Principal Component Analysis，主成分分析）、SVM（Support Vector Machine，支持向量机）以及ANN（Artificial Neural Networks，人工神经网络）。



- 深度学习的概念源于人工神经网络的研究，通过组合低层特征形成更加抽象的高层表示属性类别或特征，以发现数据的分布式特征表示。深度学习是机器学习研究中的一个新的领域，其动机在于建立、模拟人脑进行分析学习的神经网络，它模仿人脑的机制来解释数据，例如图像，声音和文本。



- 人工智能企图了解智能的实质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器，该领域的研究包括语音识别、图像识别、机器人、自然语言处理、智能搜索和专家系统等。

人工智能与机器学习、深度学习的关系

AI

人工智能(AI)：

让计算机能够象人一样思考

ML

机器学习(ML)：

提升计算机模拟人类思考能力的方法

DL

深度学习(DL)：

通过神经网络方式进行机器学习的方法

基础

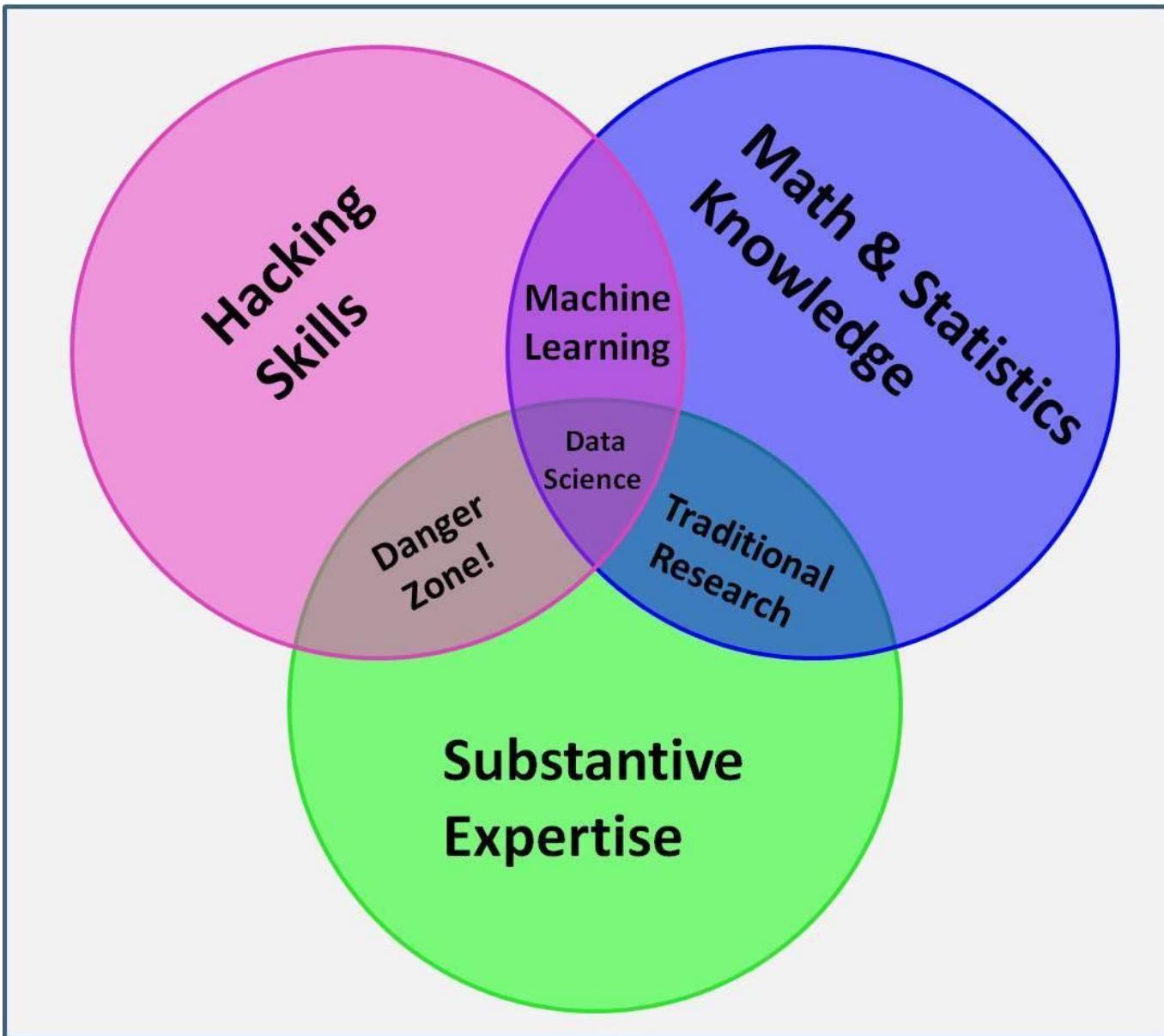
计算机

统计学

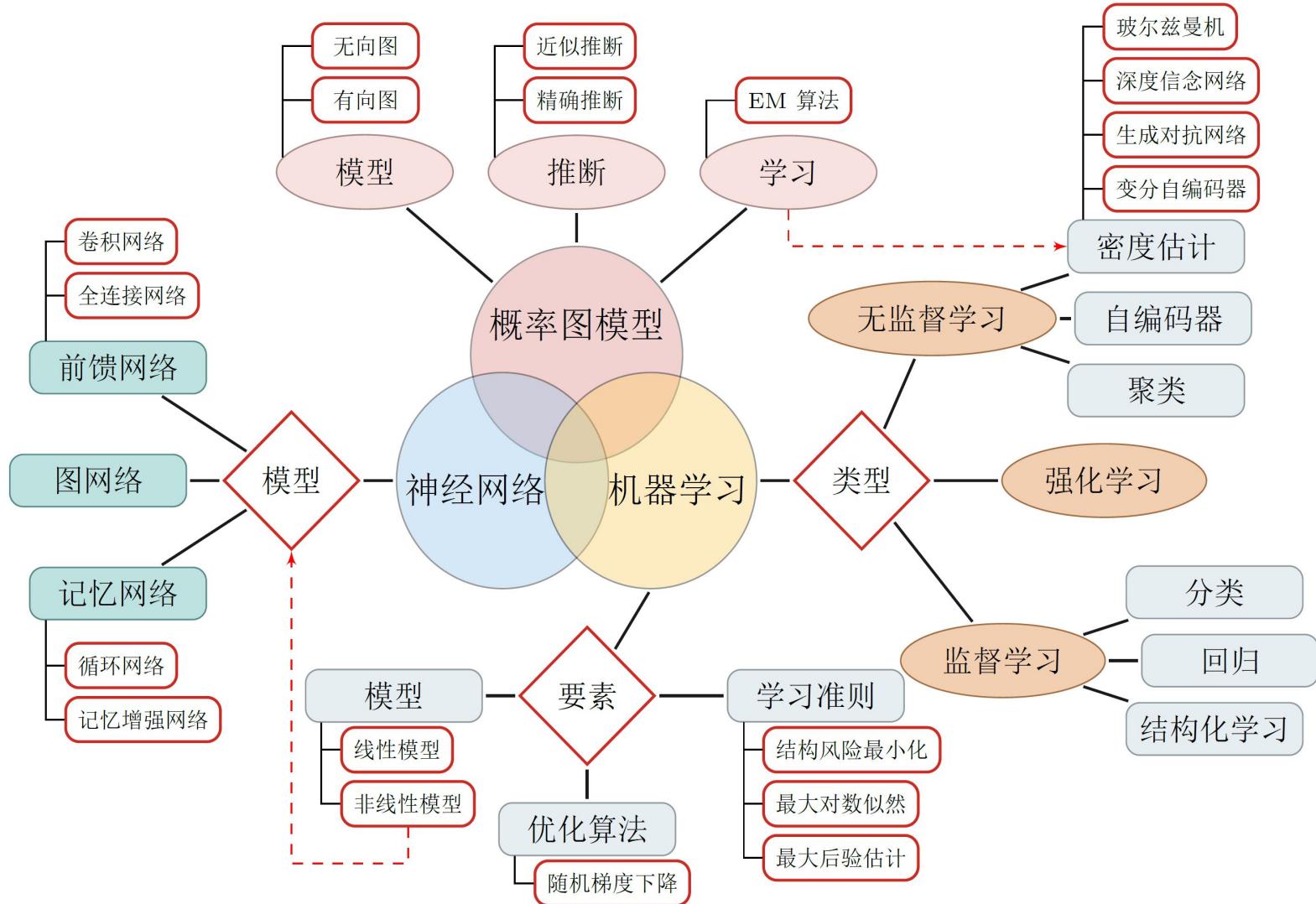
行业知识

知识工程

深度学习与数据科学



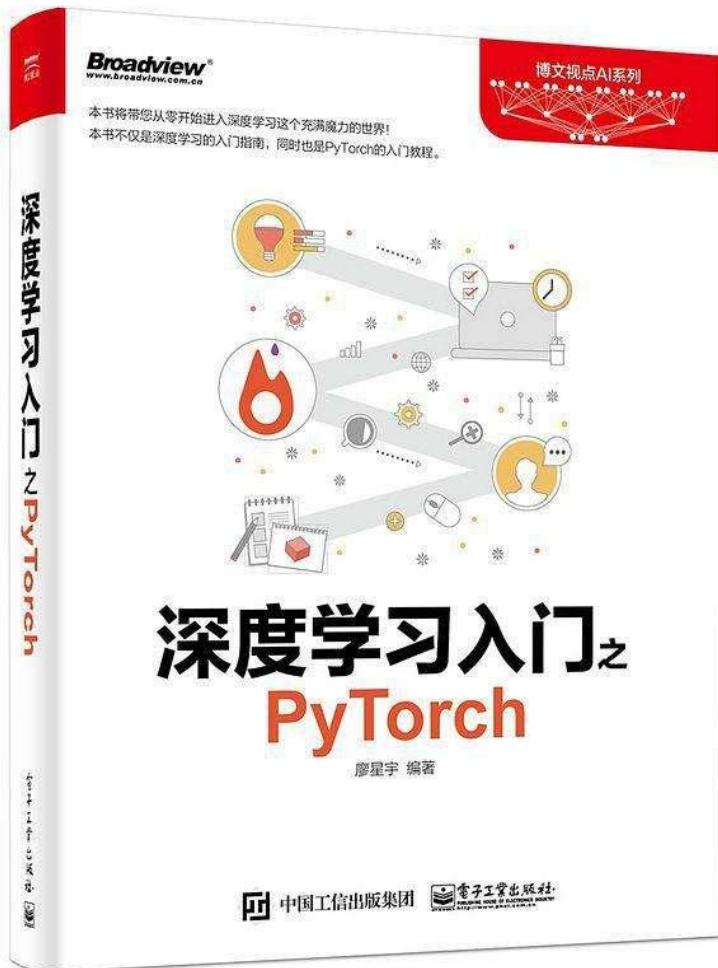
深度学习与数据科学



深度学习与数据科学

- 深度学习：一种基于无监督特征学习和特征层次结构的学习方法
- 可能的的名称：
 - 深度学习
 - 特征学习
 - 无监督特征学习

真假深度学习



真

The image shows a screenshot of a Java IDE (IntelliJ IDEA) with a dark theme. The main window displays a Java file named 'AiMain.java' containing the following code:

```
5  /*
6   * AI核心代码，估值1个亿
7   */
8 public class AiMain {
9     public static void main(String[] args) {
10        Scanner sc = new Scanner(System.in);
11        String str;
12        while (true) {
13            str = sc.next();
14            str = str.replace("吗", "");
15            str = str.replace("?", "!");
16            str = str.replace("？", "！");
17            System.out.println(str);
18        }
19    }
20 }
```

Below the code editor, there is a 'Run' toolbar with four tabs: 'AiMain' (selected), 'AiMain', 'AiMain', and 'AiMain'. The output window below shows the following text:
在吗?
在!
你好
你好
能听懂汉语吗?
能听懂汉语!
真的吗?
真的!

In the bottom right corner of the IDE interface, there is a watermark-like text: '头条 @宋忠平'.

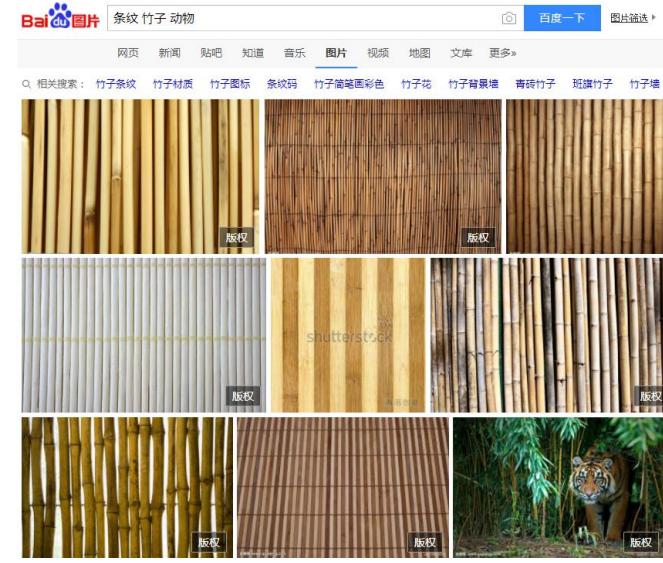
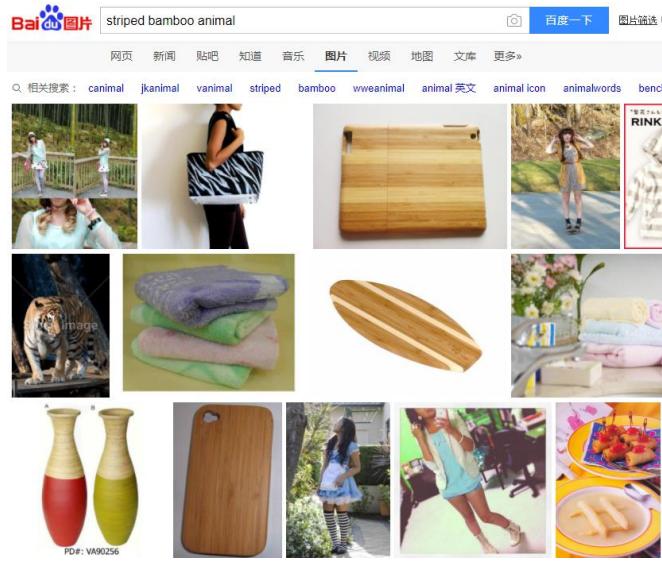
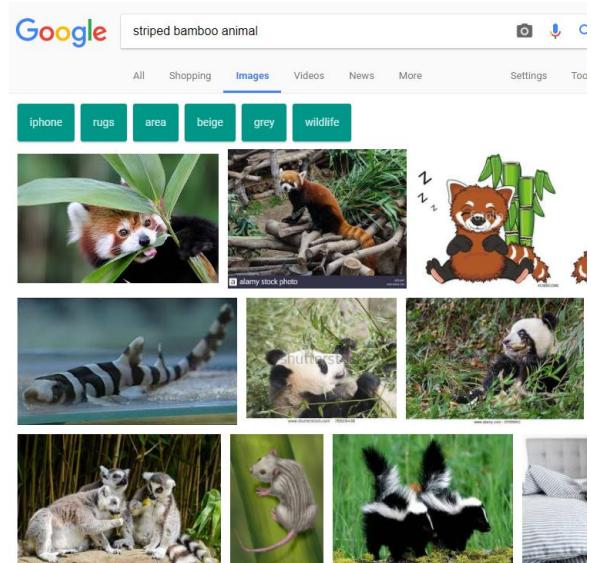
假

搜索引擎的区别

striped bamboo animal

striped bamboo animal

条纹 竹子 动物



翻译引擎和人类的较量



宁德市医院 NINGDE MUNICIPAL HOSPITAL 健康体检科

B 超室 2

B超室 B-Ultrasound Room

B super Room

检测到中文 英语 翻译 人工翻译 双语对照



14 DL

人有多大胆 地有多大产

No guts, no glory, man.

How bold a man is and how productive he is

检测到中文 英语 翻译 人工翻译 双语对照



舍不得孩子 套不着狼啊
You can't make an omelet without cracking a few eggs.

翻译引擎和人类的较量

Simple
Florida Georgia Line >

We've been there, it's safe to say it ain't our style
我们也曾有过这种处境，但我们可以肯定地说那不是我们的风格

It's just that simple, S-I-M-P-L-E
就是如此简单，鸡一安简，的安单

Simple as can be
尽可能的简单

It's just that simple, S-I-M-P-L-E
就是如此简单，鸡一安简，的安单

Simple as can be
尽可能的简单



翻译引擎的区别

biolake

37,300,000 Results Any time ▾

English (detected) ▾ Chinese Simplified ▾

biolake 比奥拉克
bǐ ào lā kè

Data from: Microsoft Translator

Google Translate

Text Documents

ENGLISH - DETECTED ENGLISH CHINESE SPANISH CHINESE (SIMPLIFIED) ENGLISH SPANISH

biolake × 光谷生物城
Guāng gǔ shēngwù chéng

Send feedback

biolake

检测到英语 → 中文 翻译 人工翻译

biolake × 生物钟
Guānggǔ shēngwù zhōng

您是不是要找: biolak 报错 拼音 双语对照

光谷生物城

检测到中文 → 英语 翻译 人工翻译

光谷生物城 × Biolake
Guānggǔ shēngwù chéng

报错 双语对照

翻译的区别（互有优劣）

它很难吃。

谷歌： It is hard to eat.

百度： It tastes terrible.

我要下班。

谷歌： I want to get off work from work.

百度： I'm going to get off work.

Baidu Translate interface showing a comparison between Google and Baidu's translation of the sentence "It is hard to eat." The Baidu version is labeled as "tastes terrible".

Google translation:

标题: Anger Boils Over at Charlottesville Council Meeting
text: Anger boiled over at the first Charlottesville City Council meeting since a white nationalist rally in the city descended into violent chaos, with some residents screaming and cursing at councilors Monday night and calling for their resignations.
Scores of people packed the council's chambers, and The Daily Progress reported Mayor Mike Signer was interrupted by shouting several times in the first few minutes of the meeting. As tensions escalated, the meeting was halted. Live video showed protesters standing on a dais with a sign that said, "Blood on your hands."

Baidu translation:

标题: 愤怒在夏洛茨维尔市议会会议上沸腾
正文: 愤怒爆发了第一次夏洛茨维尔市议会会议以来，在市一个白人民族主义者集会演变成暴力骚乱，一些居民的尖叫和咒骂议员星期一晚上呼吁辞职。许多人挤满了会议室，每日进展报告说，市长Mike Signer在会议的头几分钟被人多次叫喊。随着紧张局势升级，会议被叫停。现场视频显示，示威者站在一个标志说，“你手上的血。”
在与群众交谈后，议员Wes Bellamy说，安理会将放弃议程，并集中在群众的关注，该报报道。
嘲讽，大喊大叫的，还骂人，然后轮流解决委员会，一些挫折，领导人授予的8月12日的集会变成了暴力的许可证。其他人批评警方对这一事件的回应，该事件引来数百名白人民族主义者和其他反示威者。
阅读更多

Google Translate interface showing a comparison between Google and Baidu's translation of the sentence "I want to get off work from work." The Baidu version is labeled as "I'm going to get off work".

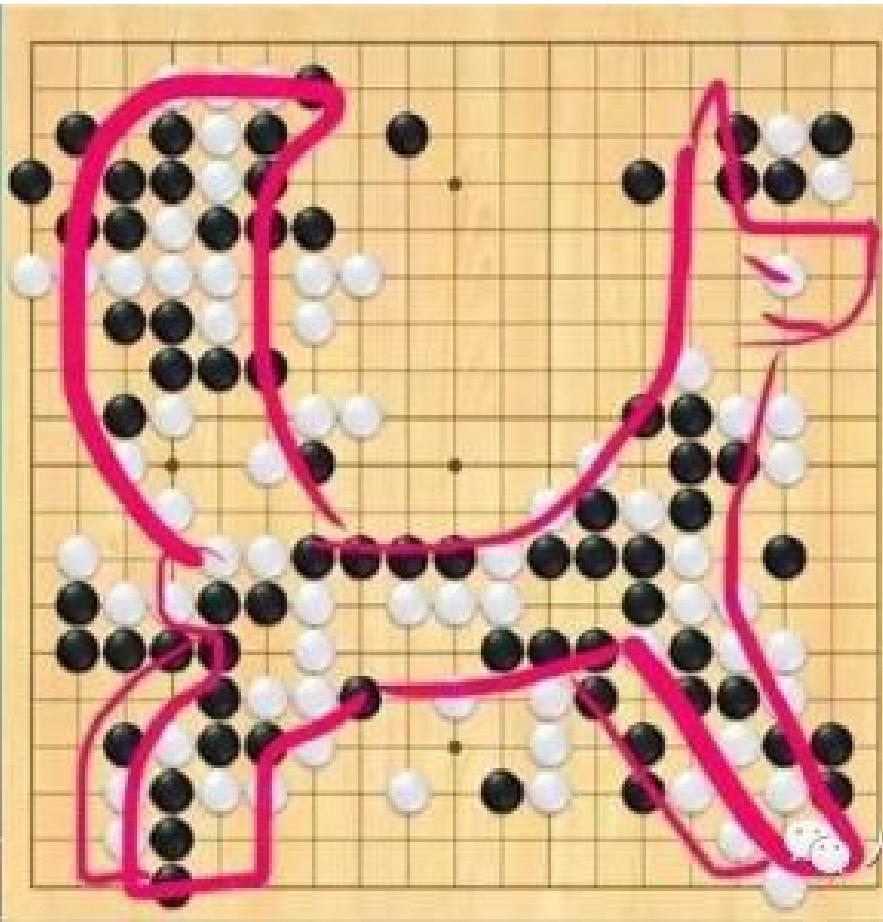
Google translation:

标题: Anger Boils Over at Charlottesville Council Meeting
text: Anger boiled over at the first Charlottesville City Council meeting since a white nationalist rally in the city descended into violent chaos, with some residents screaming and cursing at councilors Monday night and calling for their resignations.
Scores of people packed the council's chambers, and The Daily Progress reported Mayor Mike Signer was interrupted by shouting several times in the first few minutes of the meeting. As tensions escalated, the meeting was halted. Live video showed protesters standing on a dais with a sign that said, "Blood on your hands."
After talking with members of the crowd, Councilor Wes Bellamy said the council would drop its agenda and focus on the crowd's concerns, the newspaper reported.
Speakers, some yelling and hurling profanities, then took turns addressing the council, some expressing frustration that leaders had granted a permit for the Aug. 12 rally that had turned violent. Others criticized the police response to the event, which drew hundreds of white nationalists and other counter-protesters.

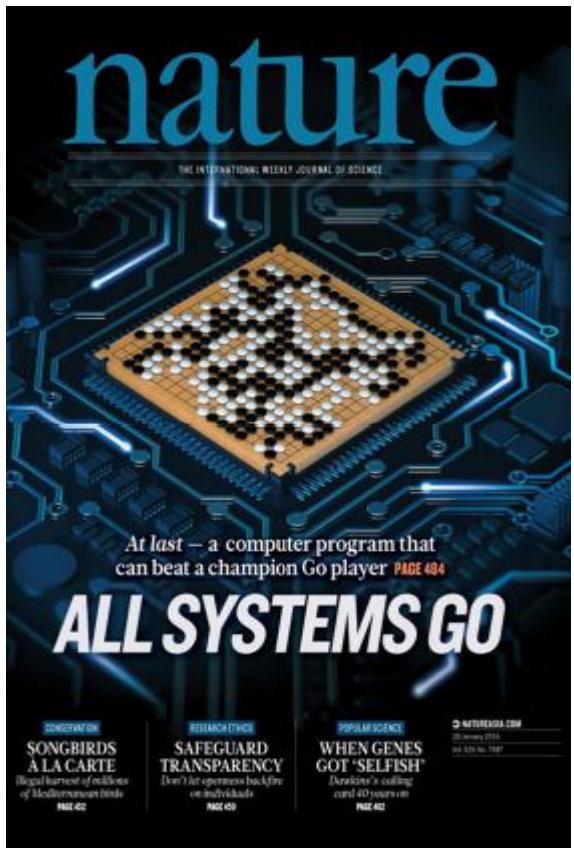
Baidu translation:

标题: 愤怒在夏洛茨维尔市议会会议上沸腾
文字: 在第一届夏洛茨维尔市议会会议上，愤怒者沸腾起来，因为白人民族主义集会在这个城市发生暴力混乱。一些居民星期一晚上在议员尖叫和骂声，并呼吁辞职。
几分钟的人们挤满了议会的众议院：“日报”报道说，市长迈克·派克（Mike Signer）在会议的前几分钟多次喊叫中断。随着紧张局势升级，会议停止了。现场视频显示抗议者站在一个戴着标志的表示说，“血在你手上。”
报纸报道，在与人群中的成员交谈之后，议员韦斯·贝拉米（Wes Bellamy）表示，该议会将放弃其议程，重点关注人群的关切。
发言人，一些大喊大叫和肆意的亵渎言论，然后轮到议会，有些人表示沮丧，领导人已经给了8月12日暴力许可。其他人批评警方对这一事件的回应，并提出了数百名白人民族主义者和其他反抗议者。

AlphaGo



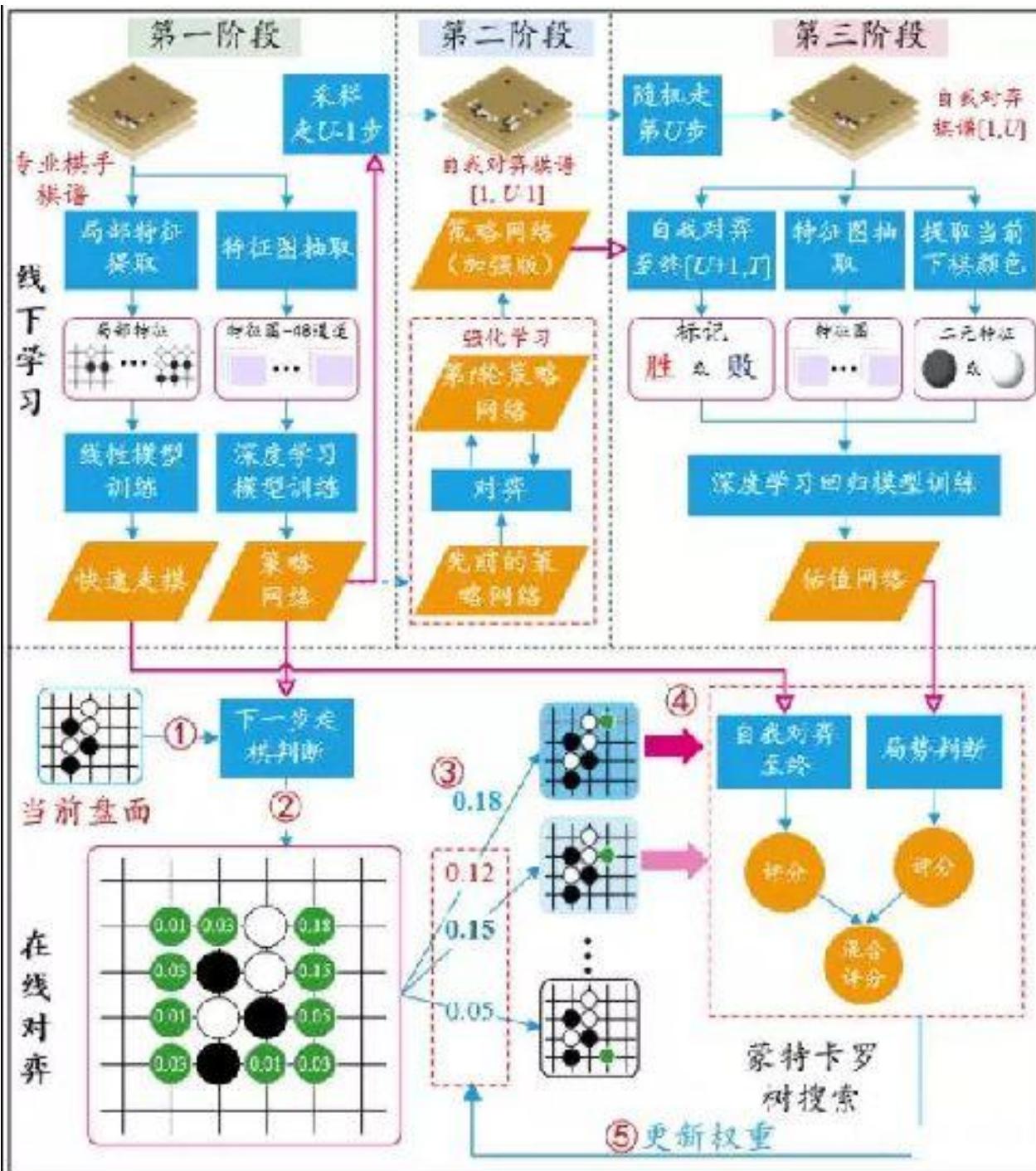
AlphaGo



ARTICLE

Mastering the game of Go with deep neural networks and tree search

David Silver*, Aja Huang*, Chris J. Maddison*, Arthur Guez*, Laurent Sifre*, George van den Driessche*, Julian Schrittwieser*, Ioannis Antonoglou*, Veda Pamarnehvelan*, Marc Lanctot*, Sander Dieleman*, Dominik Grewe*, John Nham*, Nat Kalgren-Hammer*, Ilya Sutskever*, Timothy Lillicrap*, Madeleine Leach*, Koray Karaoglu*, Thore Graepel* & Demis Hassabis*



AlphaGo vs. AlphaGo Zero



数据很重要！



算法更重要！



AlphaZero

Reinforcement Learning (强化学习)





Transfer Learning (迁移学习)

The international journal of science / 14 November 2019

nature

The cover of the November 14, 2019, issue of Nature magazine. The title "nature" is at the top in large white letters. Below it is a complex, glowing blue and purple interface that looks like a futuristic version of the StarCraft II game map. The interface features various icons, arrows, and energy fields. The word "GAME PLAN" is prominently displayed in large white letters on the left side of the interface. Below this, a subtitle reads "AI program learns to play *StarCraft II* to Grandmaster level". At the bottom of the cover, there are three news headlines: "Pharmaceuticals How to fit a drug factory inside a briefcase", "3D printing Nozzle extrudes multimaterial objects in a single run", and "Cancer imaging Tracer reveals metabolic nature of live lung tumours".

GAME
PLAN

AI program learns to play *StarCraft II* to Grandmaster level

Pharmaceuticals
How to fit a drug factory inside a briefcase

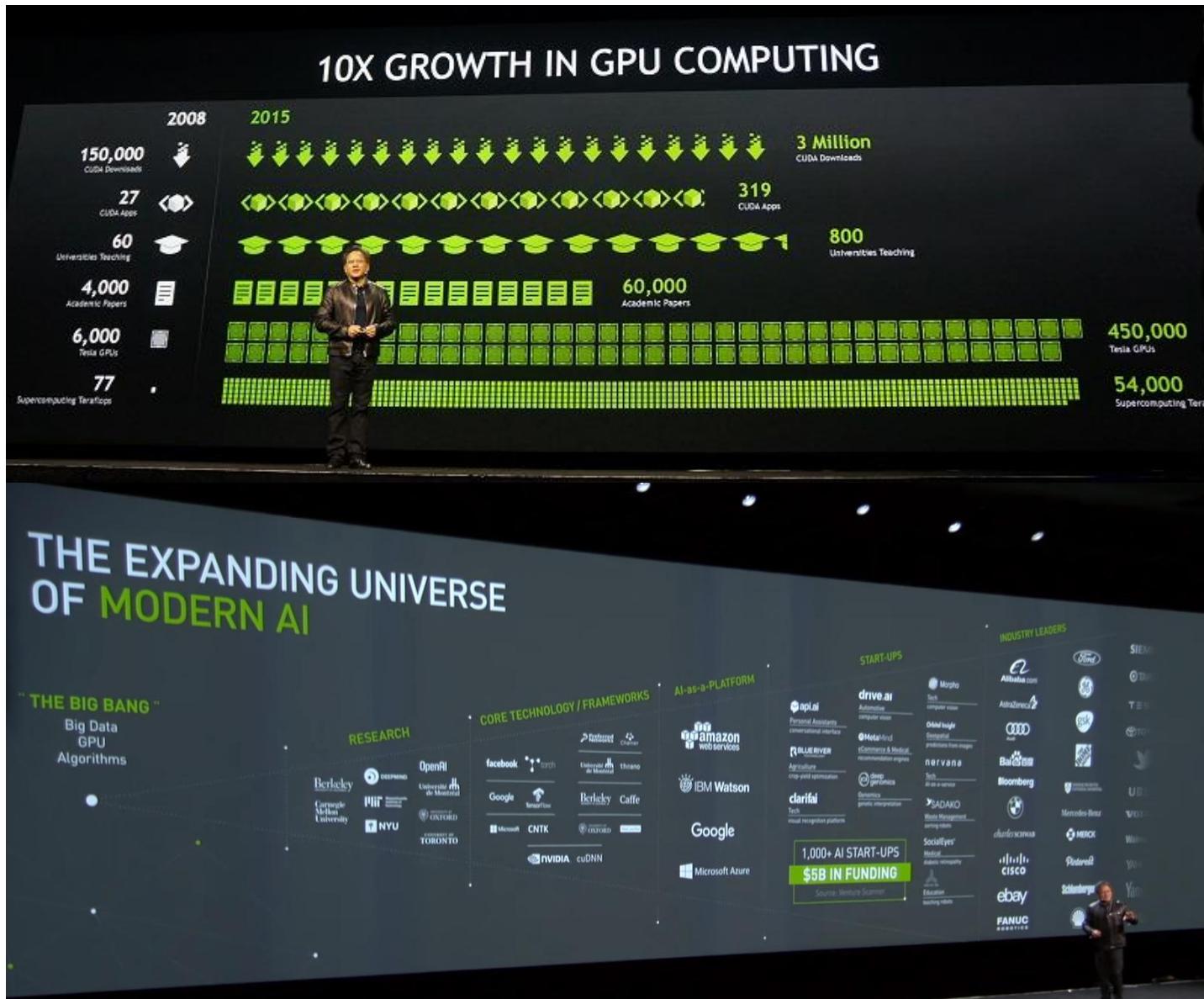
3D printing
Nozzle extrudes multimaterial objects in a single run

Cancer imaging
Tracer reveals metabolic nature of live lung tumours

Nat. 575, No. 7782
14 November 2019
9 770028083095

A standard linear barcode located at the bottom right of the cover.

Hardware: GPU, HPC...



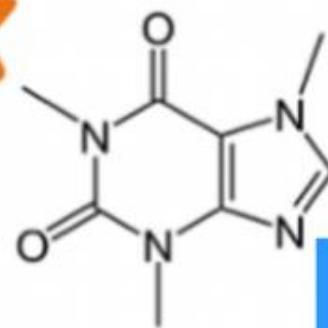
Software: TensorFlow, Caffe...



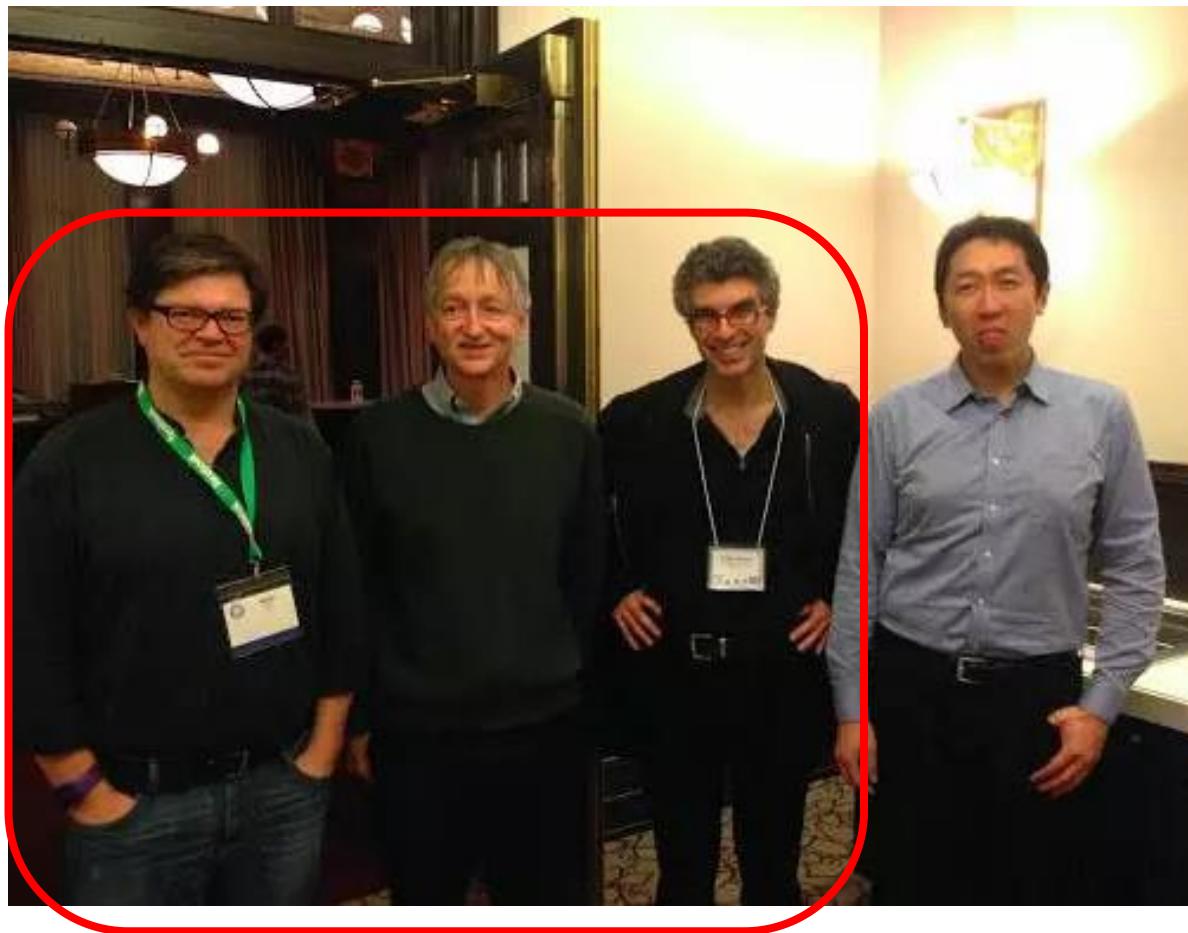
Caffe



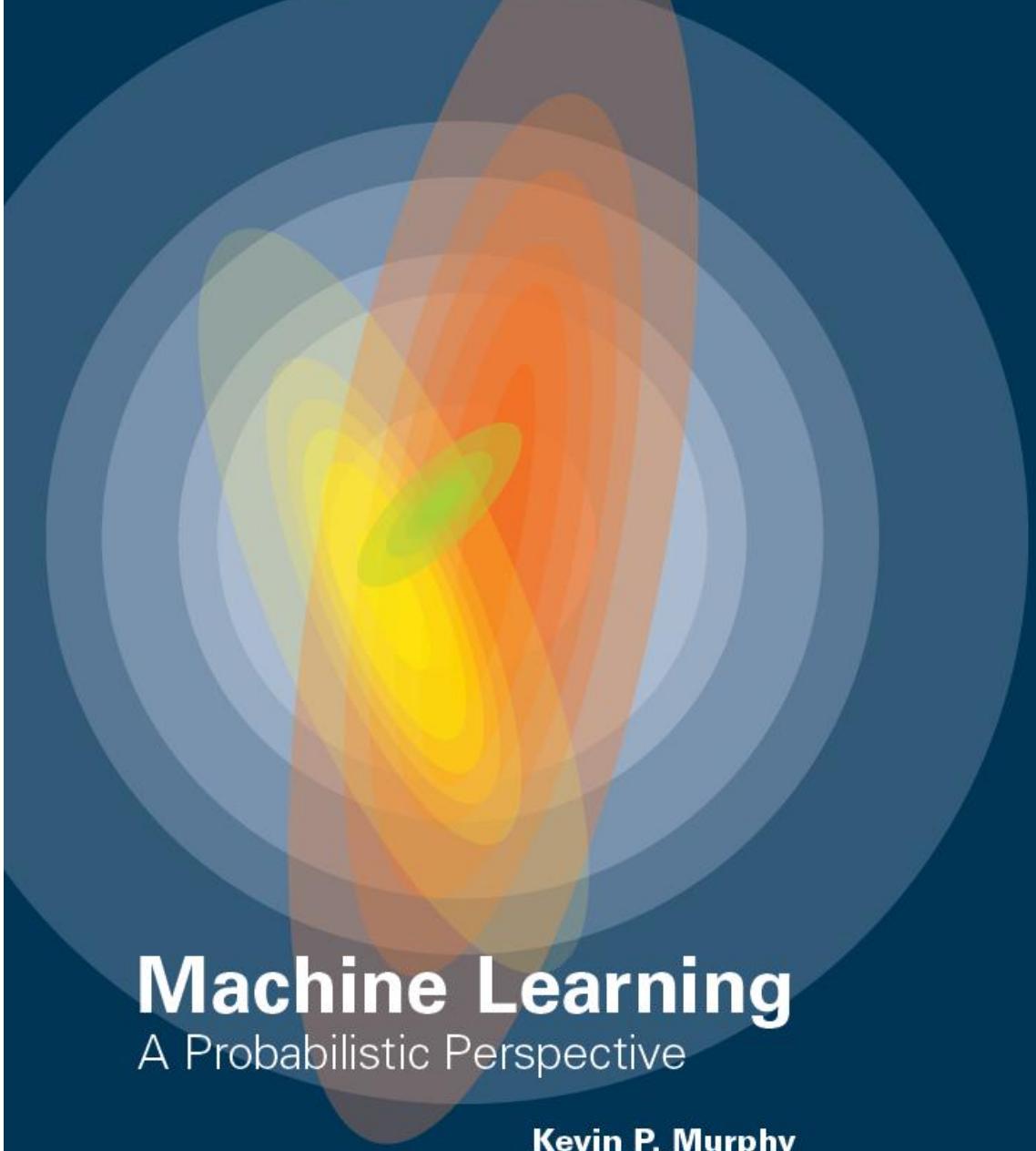
Keras



Turing Award, 2019



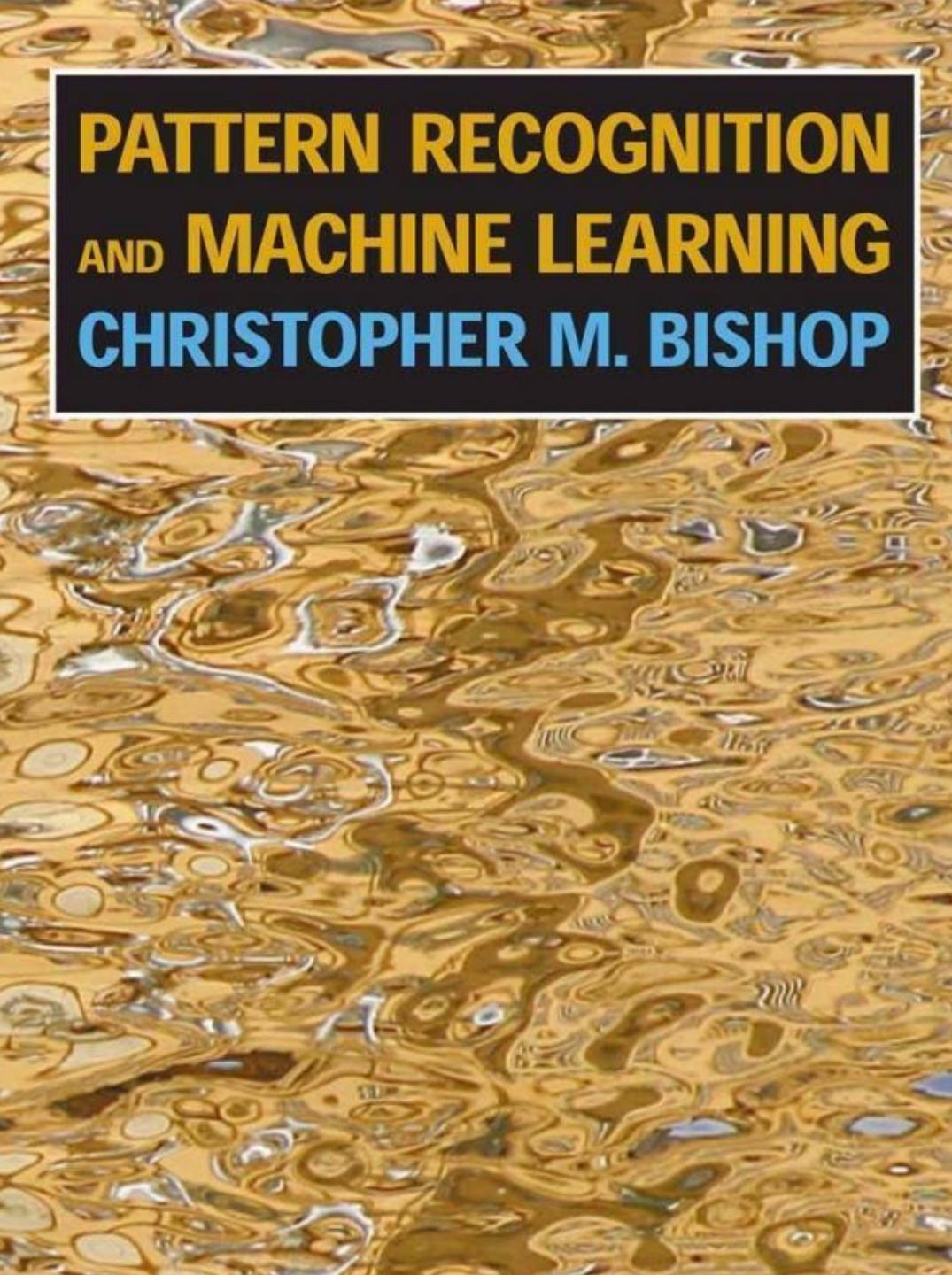
Yann LeCun
Geoffrey Hinton
Yoshua Bengio
Andrew Ng



Machine Learning

A Probabilistic Perspective

Kevin P. Murphy



PATTERN RECOGNITION
AND MACHINE LEARNING
CHRISTOPHER M. BISHOP

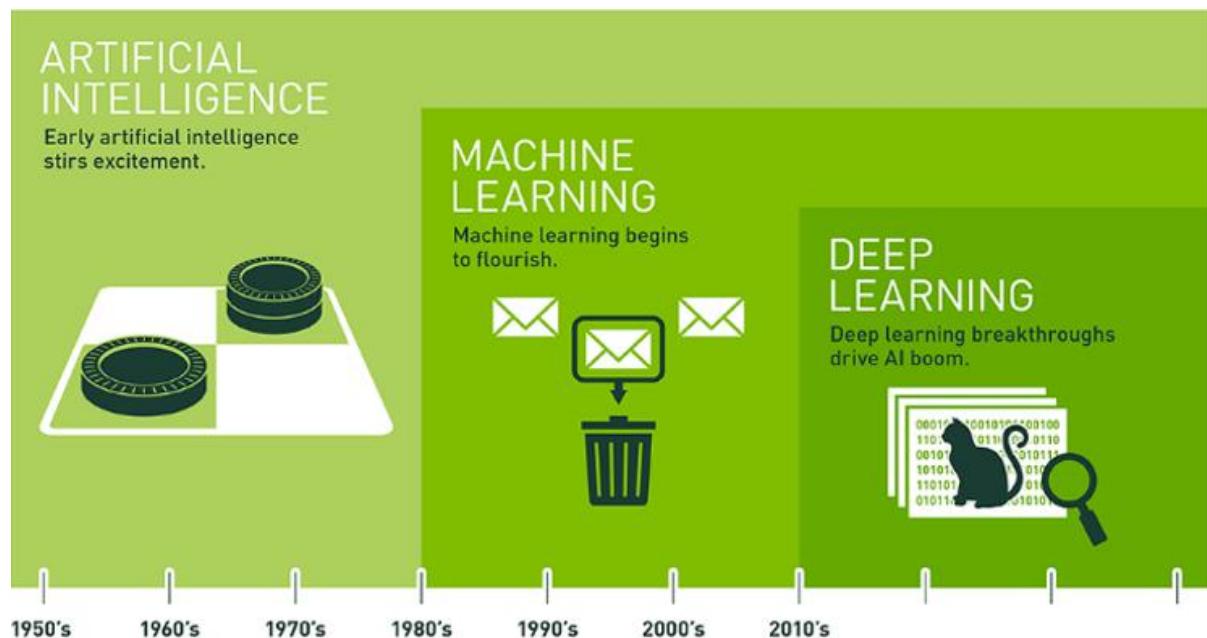
DEEP LEARNING

Ian Goodfellow, Yoshua Bengio,
and Aaron Courville



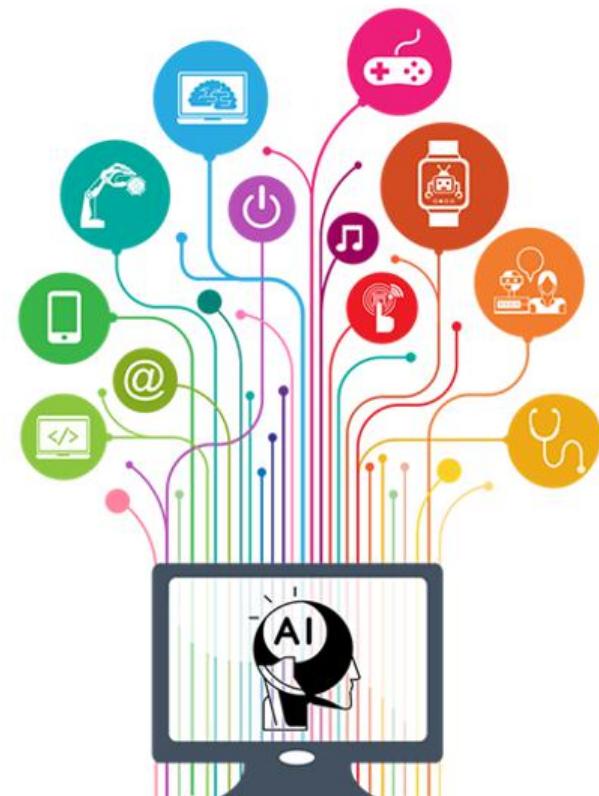
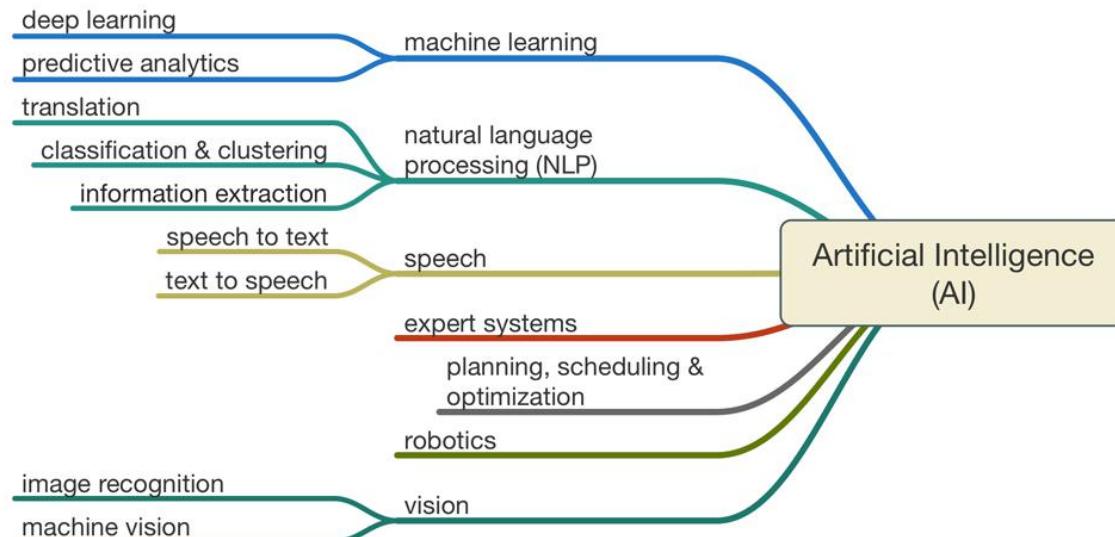
Introduction

- Artificial Intelligence
- Machine learning
- “Deep” learning



Artificial Intelligence

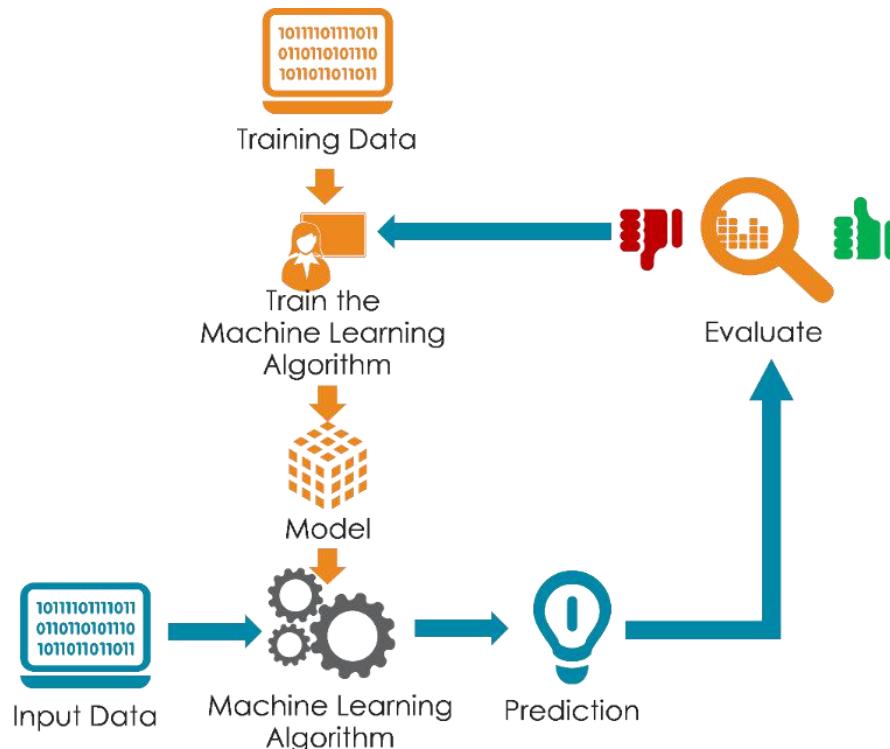
Artificial intelligence (AI, also machine intelligence, MI) is intelligence displayed by machines, in contrast with the natural intelligence (NI) displayed by humans and other animals.



Hatley, L. (2016). Presentation, New Designs for Learning: Games and Gamification
https://en.wikipedia.org/wiki/Applications_of_artificial_intelligence

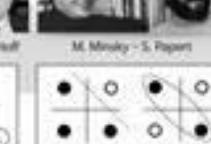
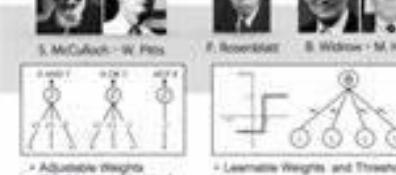
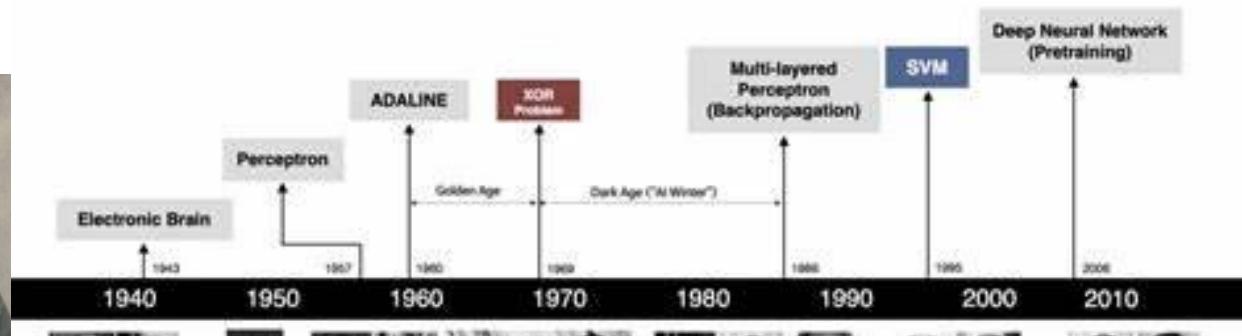
Machine Learning

Machine learning is the science of getting computers to act without being explicitly programmed.



General workflow of Machine Learning

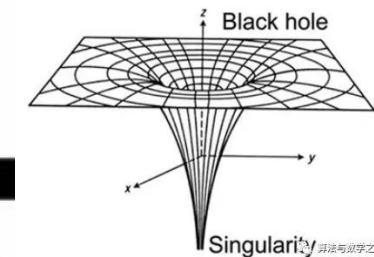
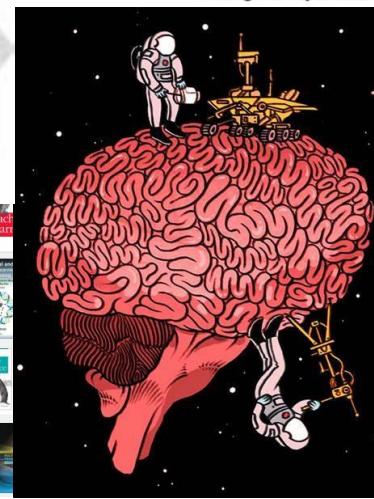
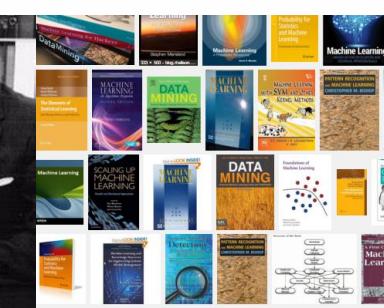
Machine Learning



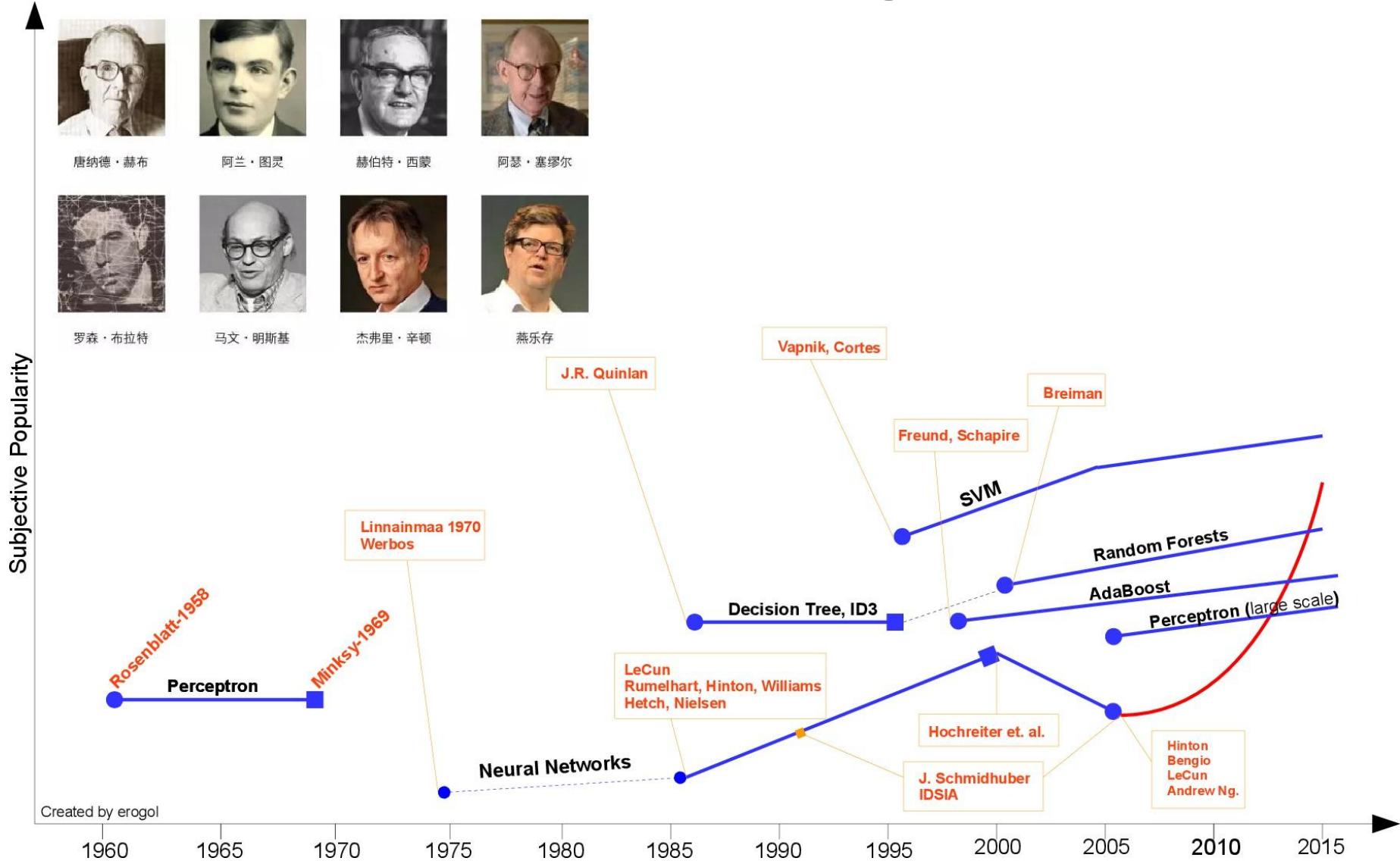
• Solution to non-linearly separable problems
• Big computation, local optima and overfitting

• Limitations of learning prior knowledge
• Kernel function, Human supervision

• Hierarchical Feature Learning



Machine Learning



机器学习 \approx 构建一个映射函数

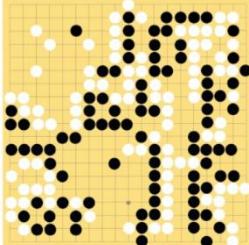
- 语音识别

$$f(\text{}) = \text{“你好”}$$

- 图像识别

$$f(\text{}) = \text{“9”}$$

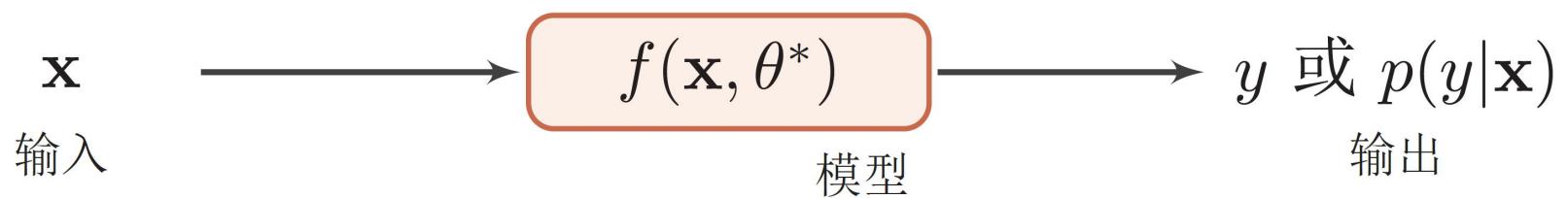
- 围棋

$$f(\text{}) = \text{“6-5”} \quad (\text{落子位置})$$

- 机器翻译

$$f(\text{“你好！”}) = \text{“Hello!”}$$

机器学习概览



机器学习的三要素

▶ 模型

- ▶ 线性方法: $f(\mathbf{x}, \theta) = \mathbf{w}^T \mathbf{x} + b$
- ▶ 广义线性方法: $f(\mathbf{x}, \theta) = \mathbf{w}^T \phi(\mathbf{x}) + b$
 - ▶ 如果 $\phi(\mathbf{x})$ 为可学习的非线性基函数, $f(\mathbf{x}, \theta)$ 就等价于神经网络。

▶ 学习准则

- ▶ 期望风险

$$\mathcal{R}(f) = \mathbb{E}_{(\mathbf{x}, y) \sim p(\mathbf{x}, y)} [\mathcal{L}(f(\mathbf{x}), y)],$$

▶ 优化

- ▶ 梯度下降

常见的机器学习类型

	监督学习	无监督学习	强化学习
训练样本	训练集 $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$	训练集 $\{\mathbf{x}^n\}_{n=1}^N$	智能体和环境交互的轨迹 τ 和累积奖励 G_τ
优化目标	$y = f(\mathbf{x})$ 或 $p(y \mathbf{x})$	$p(\mathbf{x})$ 或带隐变量 \mathbf{z} 的 $p(\mathbf{x} \mathbf{z})$	期望总回报 $\mathbb{E}_\tau[G_\tau]$
学习准则	期望风险最小化 最大似然估计	最大似然估计 最小重构错误	策略评估 策略改进

参数学习

- ▶ 期望风险未知，通过经验风险近似
- ▶ 训练数据： $\mathcal{D} = \{x^{(n)}, y^{(n)}\}, i \in [1, N]$

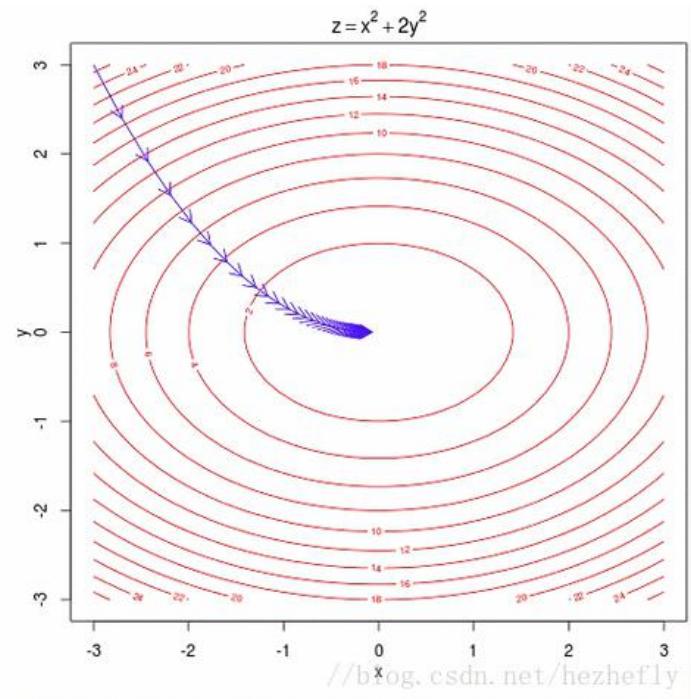
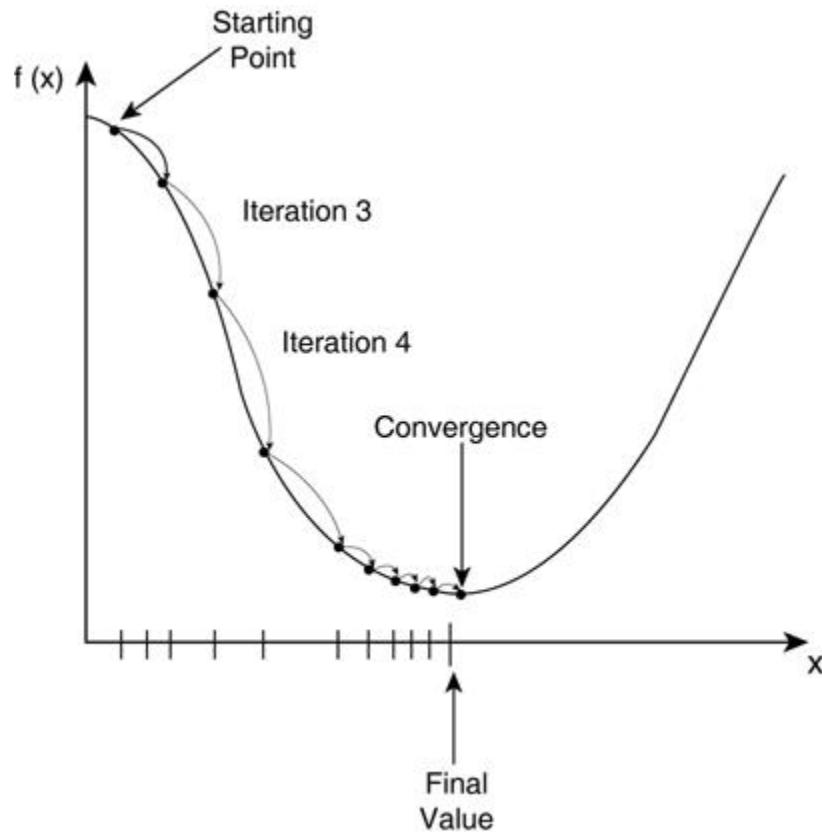
$$\mathcal{R}_{\mathcal{D}}^{emp}(\theta) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(y^{(n)}, f(x^{(n)}, \theta))$$

- ▶ 经验风险最小化
- ▶ 在选择合适的风险函数后，我们寻找一个参数 θ^* ，使得经验风险函数最小化。

$$\theta^* = \arg \min_{\theta} \mathcal{R}_{\mathcal{D}}^{emp}(\theta)$$

- ▶ 机器学习问题转化成一个优化问题

优化：梯度下降法



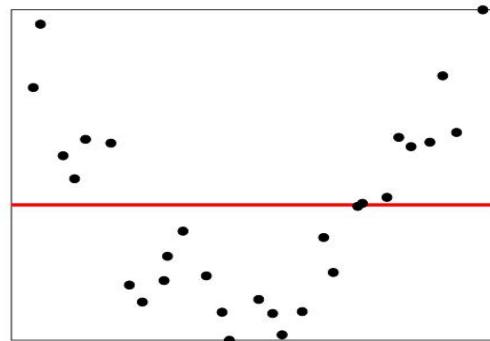
//blog.csdn.net/hezhefly

机器学习 = 优化?

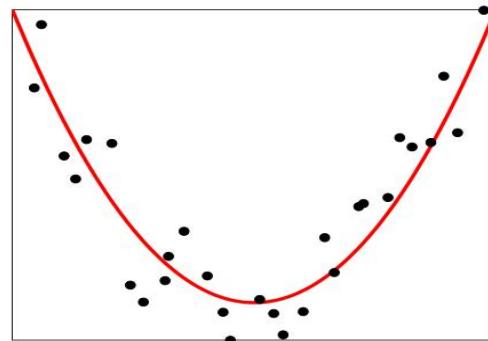
机器学习 = 优化?

NO!

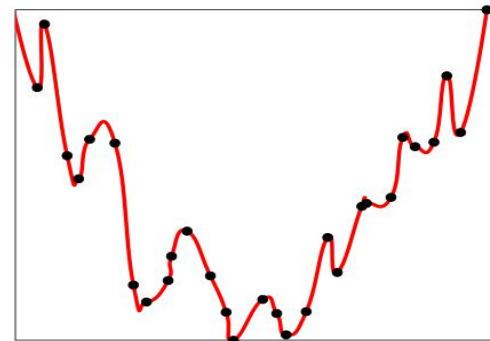
欠拟合



正常



过拟合



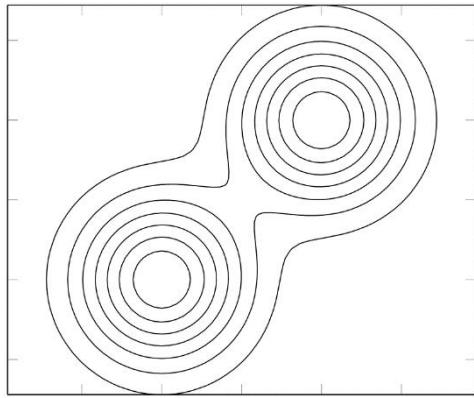
过拟合：**经验风险最小化原则**很容易导致模型在训练集上错误率很低，但是在未知数据上错误率很高。

泛化错误

期望风险

$$\mathcal{R}(f) = \mathbb{E}_{(\mathbf{x}, y) \sim p(\mathbf{x}, y)} [\mathcal{L}(f(\mathbf{x}), y)],$$

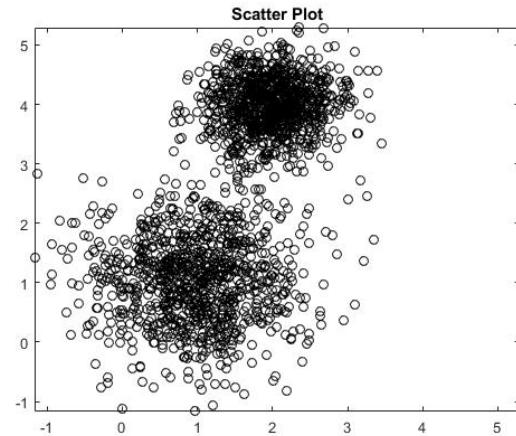
真实分布 p_r



≠

经验风险

$$\mathcal{R}_{\mathcal{D}}^{emp}(\theta) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(y^{(n)}, f(x^{(n)}, \theta))$$



$$\mathcal{G}_{\mathcal{D}}(f) = \mathcal{R}(f) - \mathcal{R}_{\mathcal{D}}^{emp}(f)$$

泛化错误

如何减少泛化错误？

优化

经验风险最小

正则化

降低模型复杂度



正则化 (regularization)

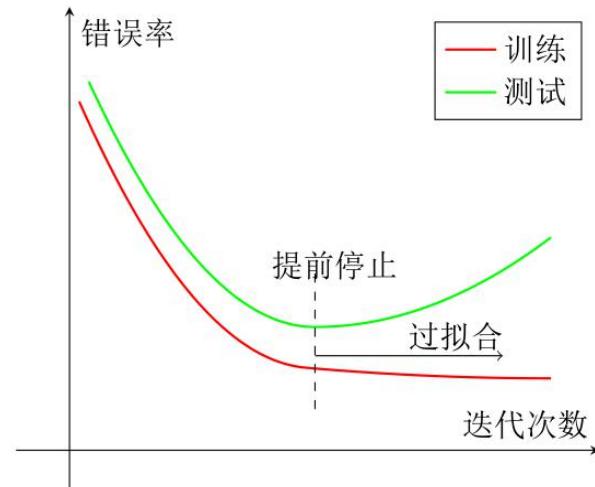
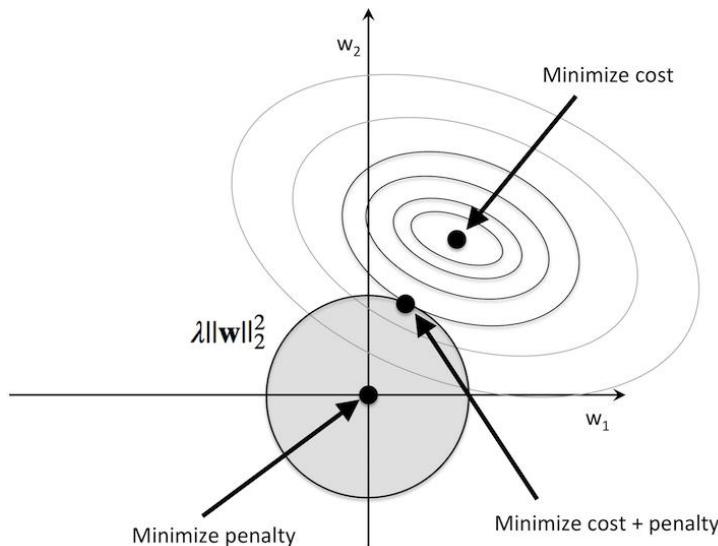
所有损害优化的方法都是正则化。

增加优化约束

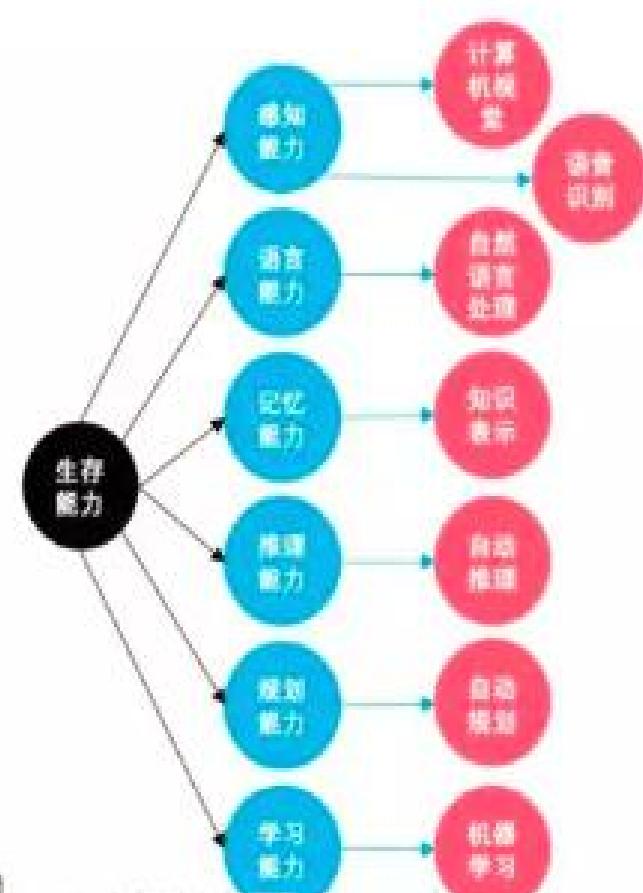
L1/L2约束、数据增强

干扰优化过程

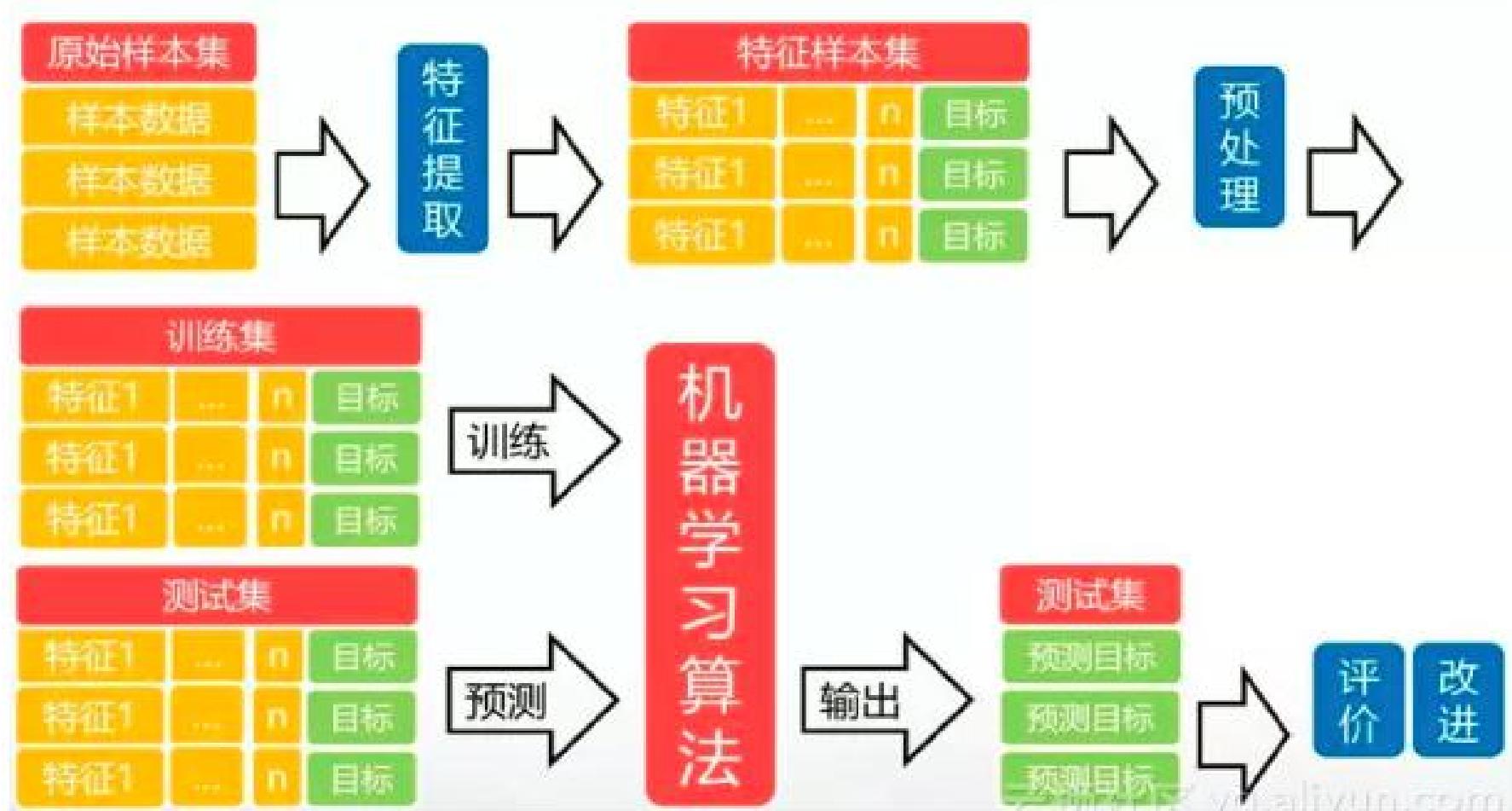
权重衰减、随机梯度下降、提前停止



拥抱人工智能从机器学习开始

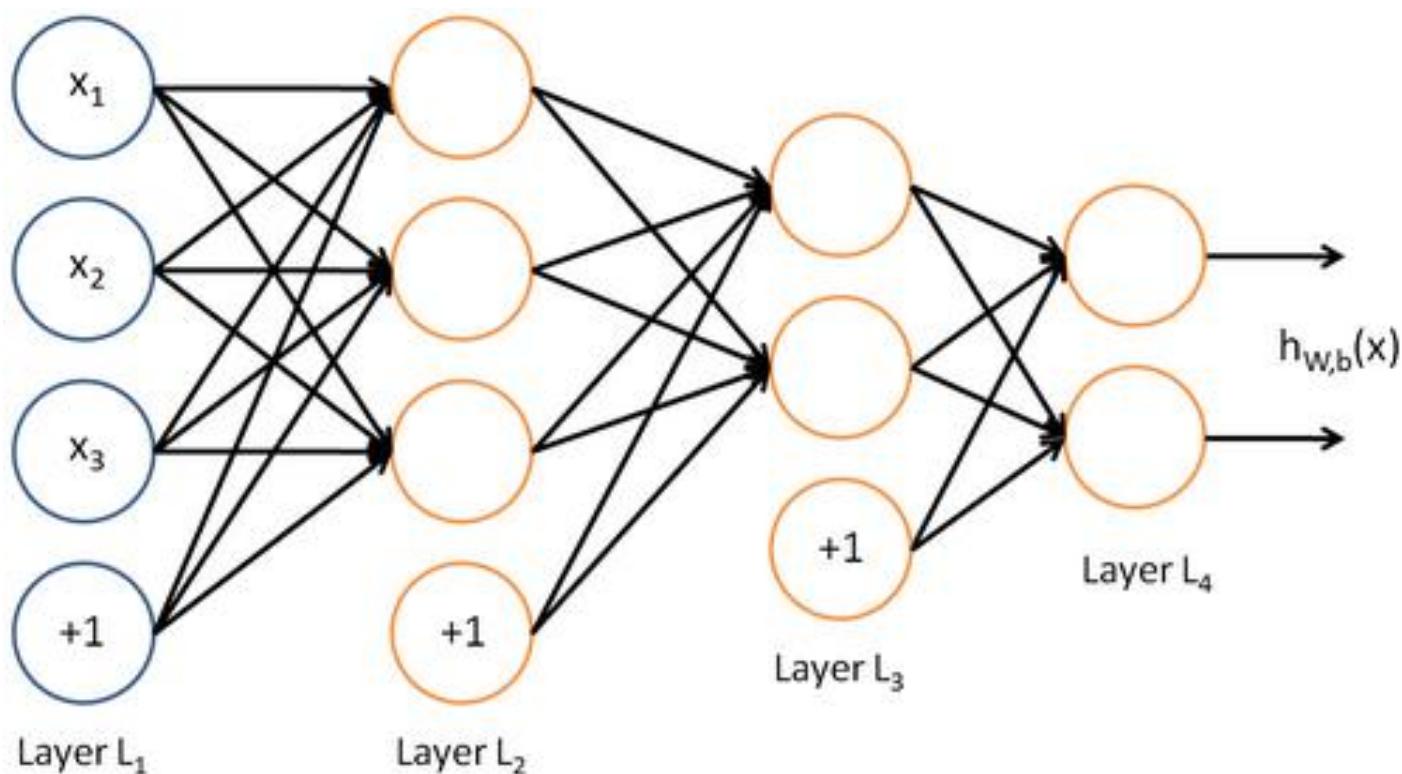


- 机器学习最大的特点是利用数据而不是指令来进行各种工作，其学习过程主要包括：数据的特征提取、数据预处理、训练模型、测试模型、模型评估改进等几部分。



机器学习算法是使计算机具有智能的关键

- 算法是通过使用已知的输入和输出以某种方式“训练”以对特定输入进行响应。代表着用系统的方法描述解决问题的策略机制。人工智能的发展离不开机器学习算法的不断进步。



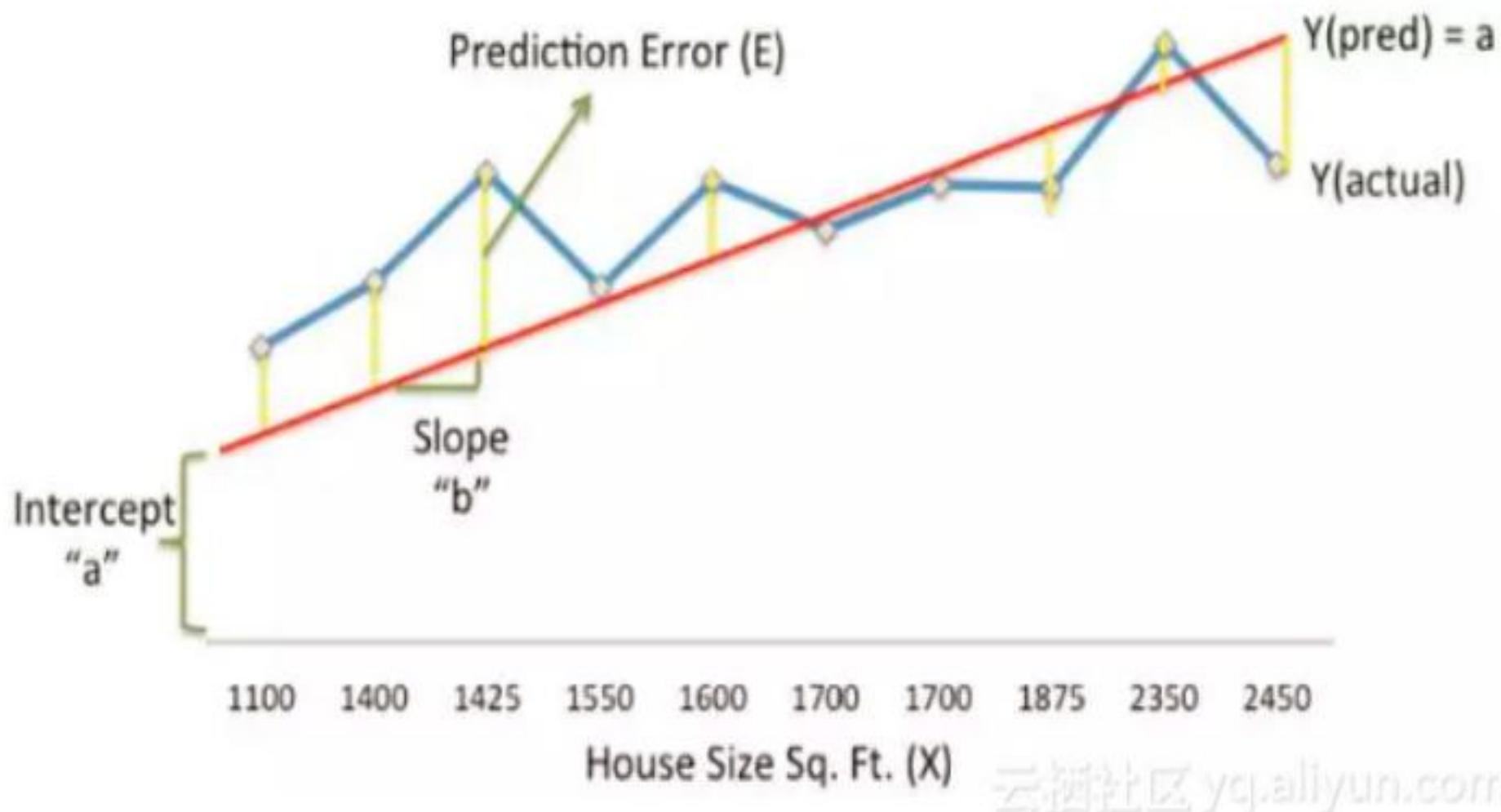
机器学习算法分类



1. 线性回归：找到一条直线来预测目标值

- 一个简单的场景：已知房屋价格与尺寸的历史数据，问面积为2000时，售价为多少？

House Size sq.ft (X)	1400	1600	1700	1875	1100	1550	2350	2450	1425	1710
House Price \$ (Y)	245,000	312,000	279,000	308,000	199,000	219,000	405,000	324,000	319,000	255,000

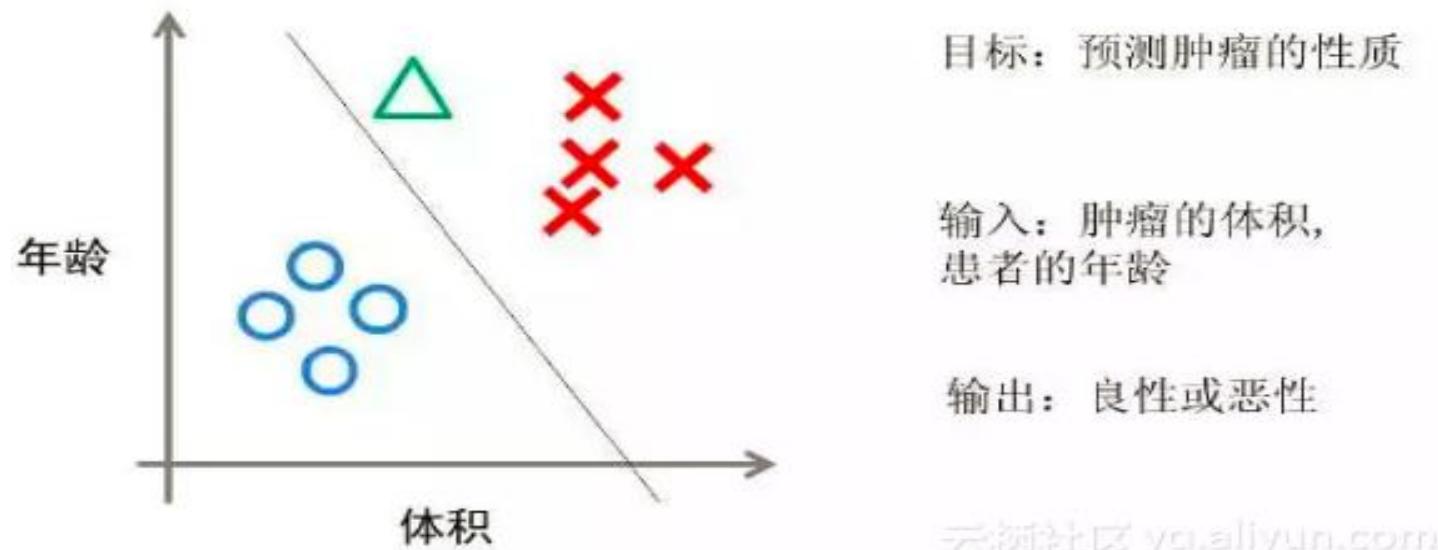


线性回归的应用

- **预测客户终生价值：** 基于老客户历史数据与客户生命周期的关联关系，建立线性回归模型，预测新客户的终生价值，进而开展针对性的活动。
- **机场客流量分布预测：** 以海量机场WiFi数据及安检登机值机数据，通过数据算法实现机场航站楼客流分析与预测。
- **货币基金资金流入流出预测：** 通过用户基本信息数据、用户申购赎回数据、收益率表和银行间拆借利率等信息，对用户的申购赎回数据的把握，精准预测未来每日的资金流入流出情况。
- **电影票房预测：** 依据历史票房数据、影评数据、舆情数据等互联网公众数据，对电影票房进行预测。

2. 逻辑回归：找到一条直线来分类数据

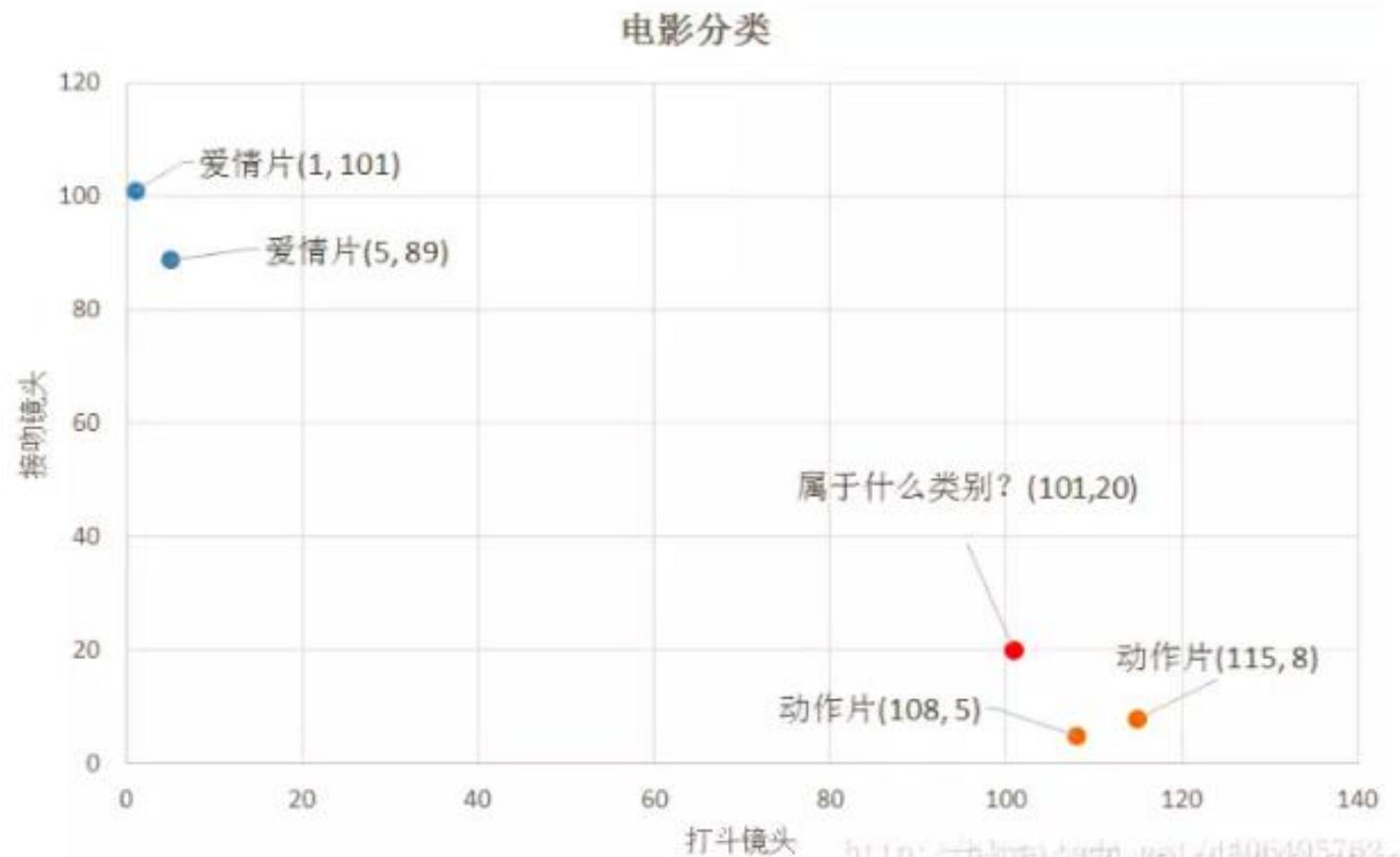
- 逻辑回归虽然名字叫回归，却是属于分类算法，是通过 Sigmoid函数将线性函数的结果映射到Sigmoid函数中，预估事件出现的概率并分类。



逻辑回归从直观上来说是画出了一条分类线。位于分类线一侧的数据，概率 >0.5 , 属于分类A；位于分类线另一侧的数据，概率 <0.5 , 属于分类B。

3. K-近邻：用距离度量最相邻的分类标签

- 一个简单的场景：已知一个电影中的打斗和接吻镜头数，判断它是属于爱情片还是动作片。当接吻镜头数较多时，根据经验我们判断它为爱情片。那么计算机如何进行判别呢？



4. 朴素贝叶斯：选择后验概率最大的类为分类标签

- 一个简单的场景：一号碗(C1)有30颗水果糖和10颗巧克力糖，二号碗(C2)有水果糖和巧克力糖各20颗。现在随机选择一个碗，从中摸出一颗糖，发现是水果糖。
- 问这颗水果糖(x)最有可能来自哪个碗？

$$P(\omega_j | x) = \frac{p(x | \omega_j) P(\omega_j)}{p(x)}$$

似然函数 $p(x | \omega_j)$ 和 先验概率 $P(\omega_j)$ 是分子的组成部分。
后验概率 $P(\omega_j | x)$ 是整个表达式的结果。
证据因子 (evidence) $p(x)$ 是分母的组成部分，可以通过以下公式计算：

$$p(x) = \sum_{j=1}^2 p(x | \omega_j) P(\omega_j)$$

例如上面的例子中： $P(X)$: 水果糖的概率为 $5/8$

$P(X|C1)$: 一号碗中水果糖的概率为 $3/4$

$P(X|C2)$: 二号碗中水果糖的概率为 $2/4$

$P(C1)=P(C2)$: 两个碗被选中的概率相同，为 $1/2$

则水果糖来自一号碗的概率为：

$$P(C1|X) = P(X|C1)P(C1)/P(X) = (3/4)(1/2)/(5/8) = 3/5$$

水果糖来自二号碗的概率为：

$$P(C2|X) = P(X|C2)P(C2)/P(X) = (2/4)(1/2)/(5/8) = 2/5$$

$$P(C1|X) > P(C2|X)$$

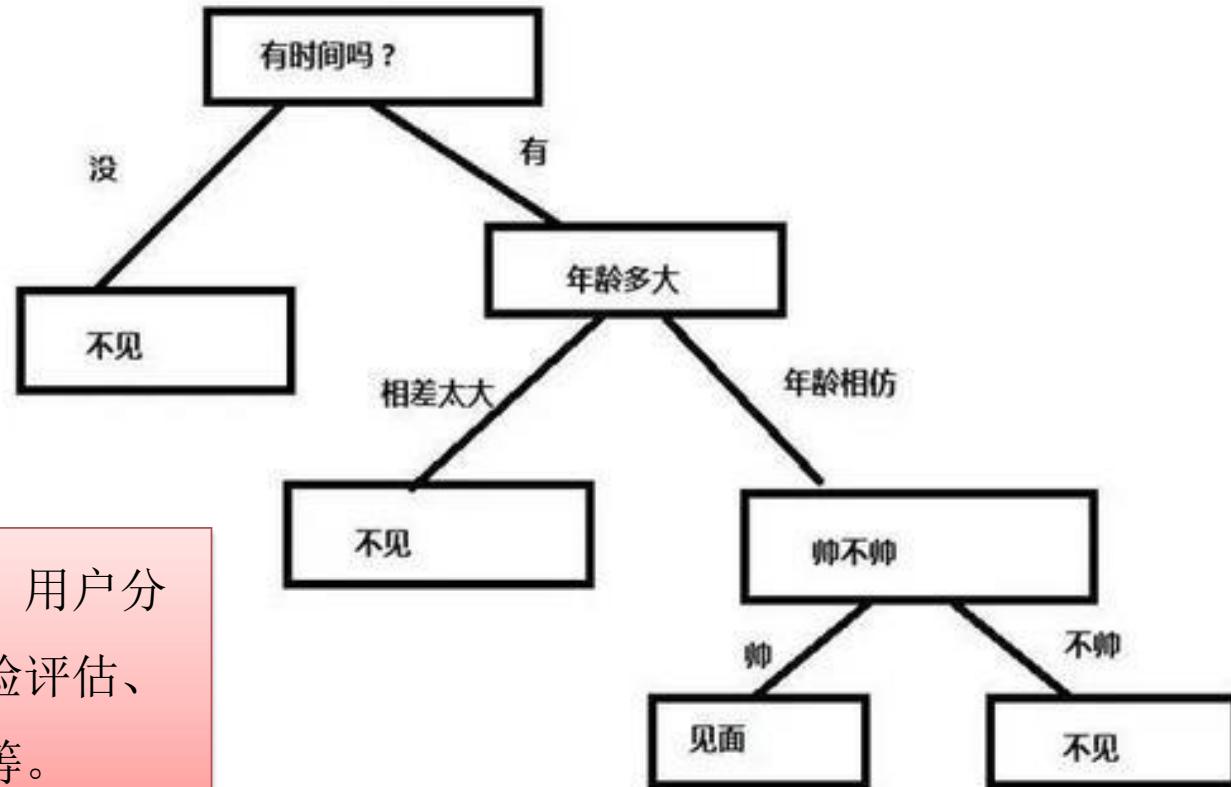
因此这颗糖最有可能来自一号碗。

朴素贝叶斯的主要应用有文本分类、垃圾文本过滤，情感判别，多分类实时预测等。

5. 决策树：构造熵值下降最快的分类树

- 一个简单的场景：

相亲时，可能首先检测相亲对方是否有时间。如果有，则考虑进一步接触，再观察其是否有上进心，如果没有，直接Say Goodbye。如果有，则在看帅不帅，帅的可以列入候选名单。

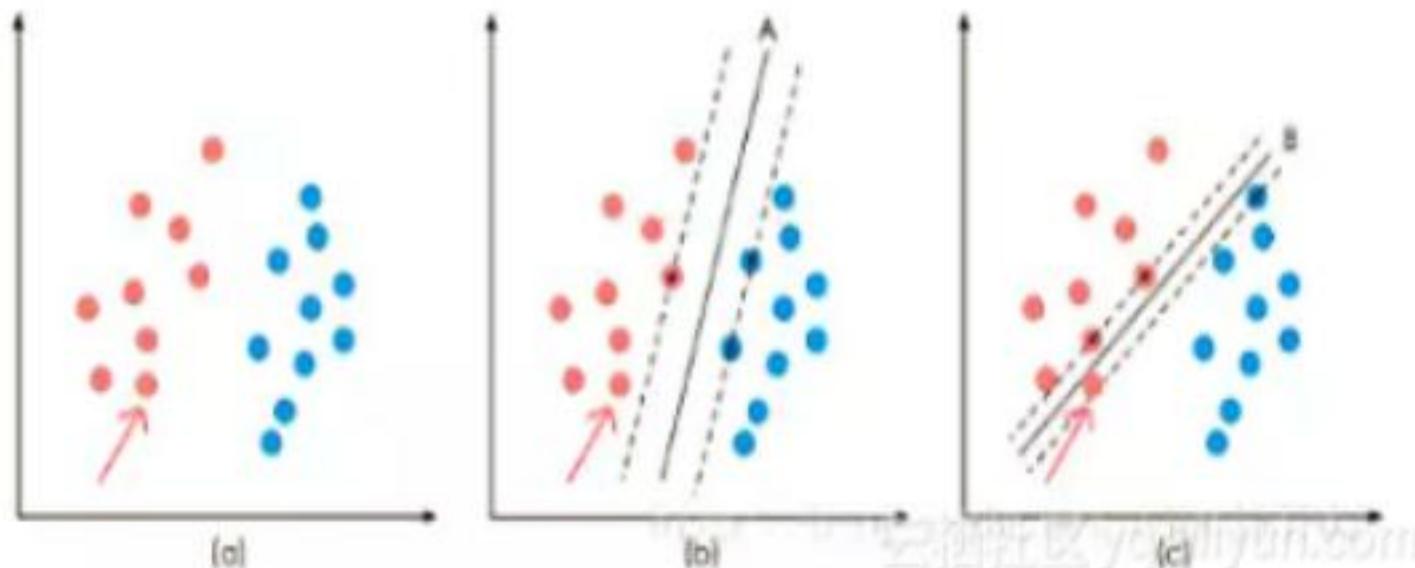


决策树可以应用于：用户分级评估、贷款风险评估、选股、投标决策等。

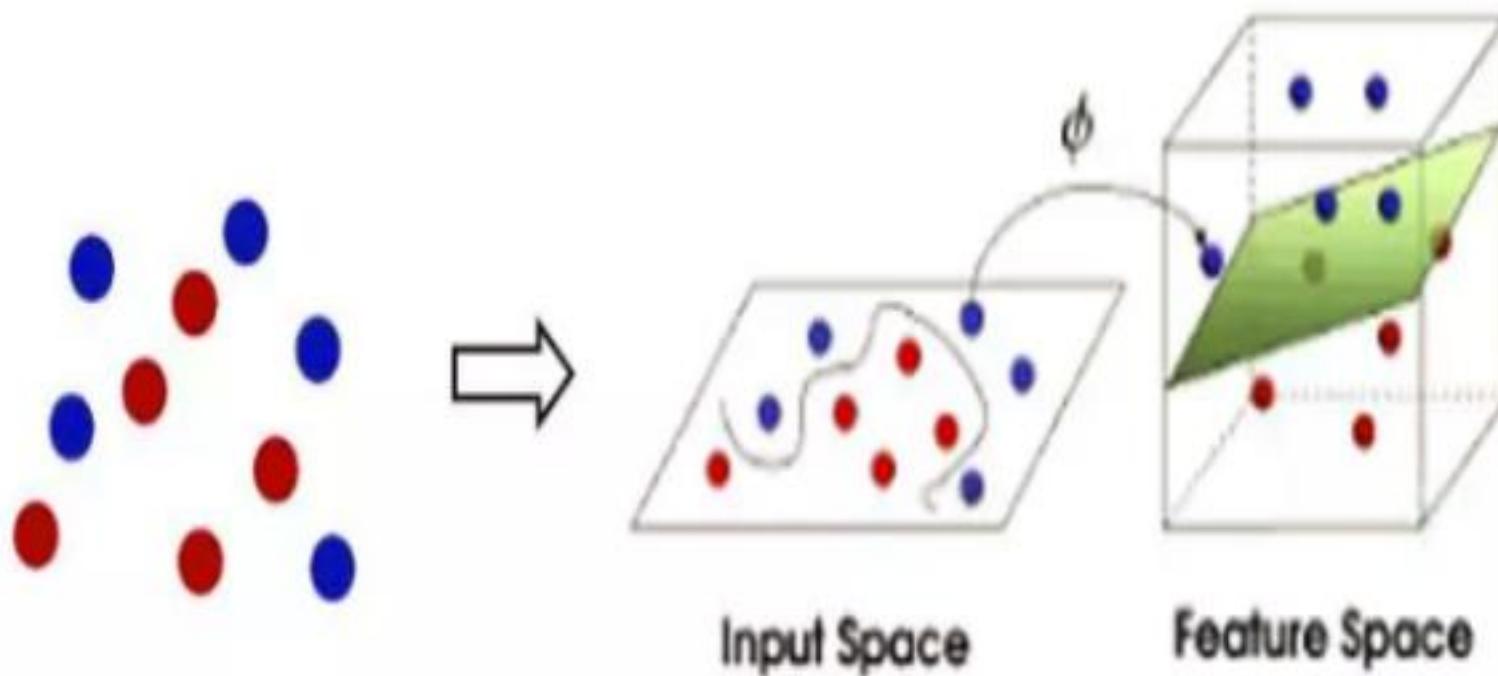
6. 支持向量机 (SVM)：构造超平面，分类非线性数据

一个简单的场景：

要求用一根线将不同颜色的球分开，要求尽量在放更多球之后，仍然适用。A、B两条线都可以满足条件。再继续增加球，线A仍可以将球很好的分开，而线B则不可以。



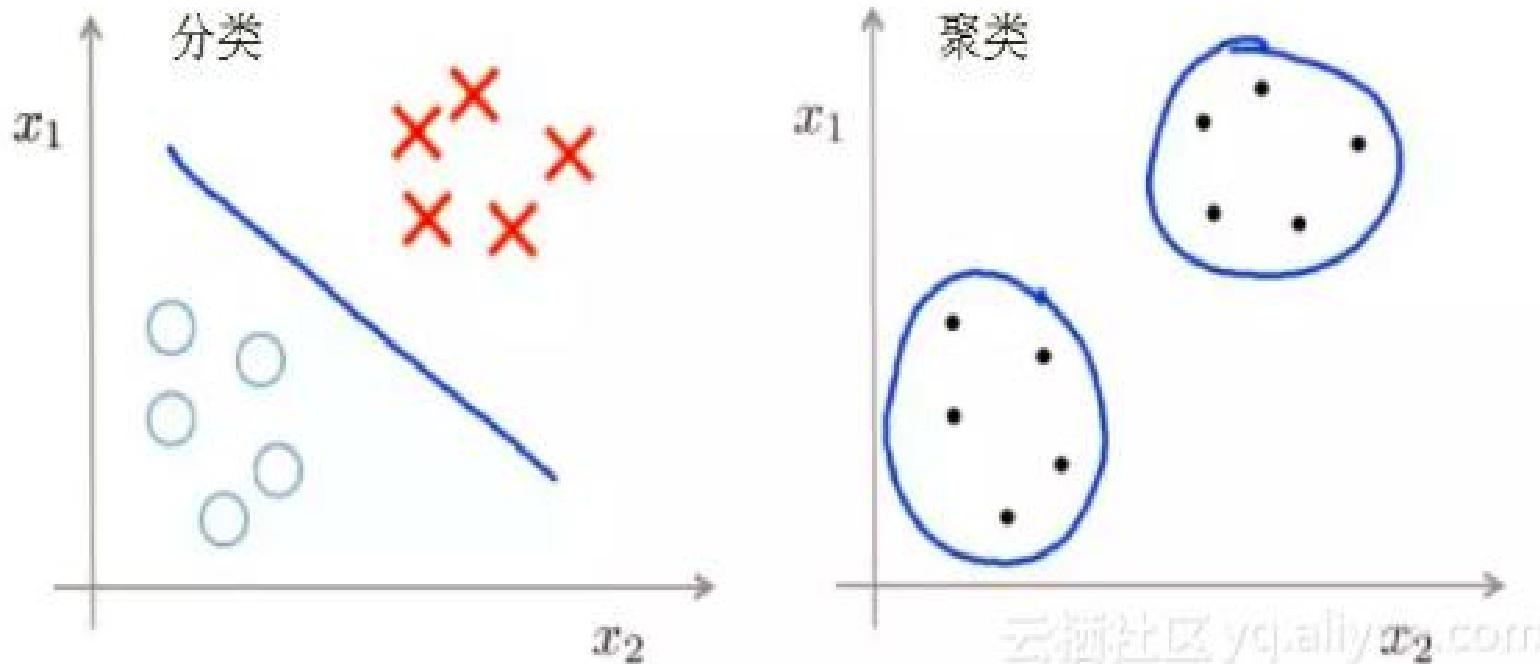
进一步增加难度，当球没有明确的分界线，用一条直线已经无法将球分开，该怎么解决？



SVM 可应用于垃圾邮件识别、手写识别、文本分类、选股等。

7. K-means: 计算质心，聚类无标签数据

- 在上面介绍的分类算法中，需要被分类的数据集已经有标记，例如数据集已经标记为○或者×，通过学习出假设函数对这两类数据进行划分。而对于没有标记的数据集，希望能有一种算法能够自动的将相同元素分为紧密关系的子集或簇，这就是聚类算法。

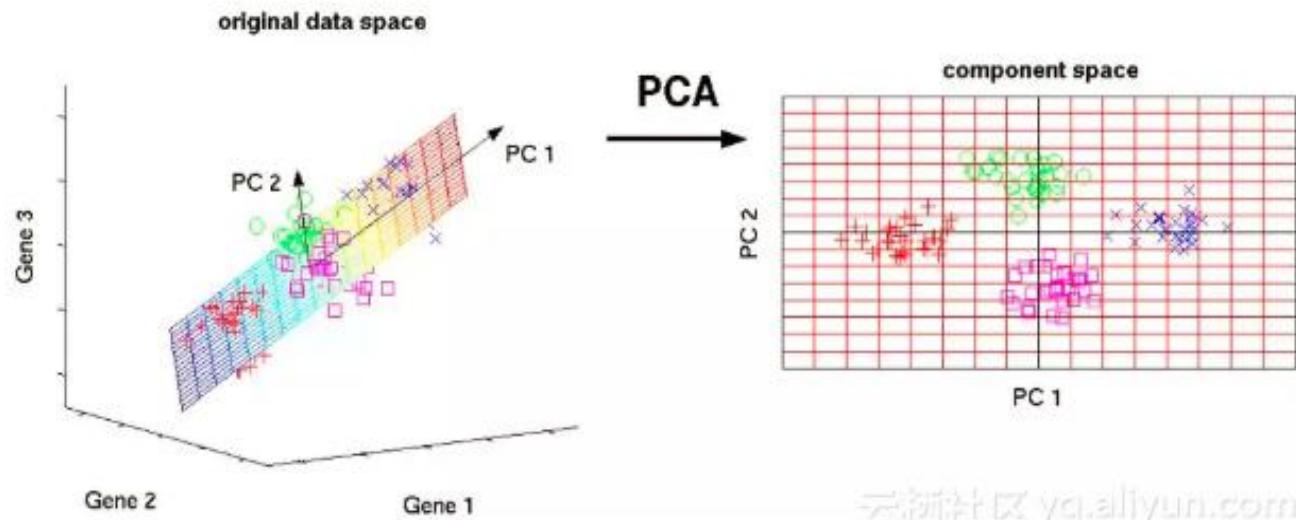


• 8. 关联分析：挖掘啤酒与尿布（频繁项集）的关联规则

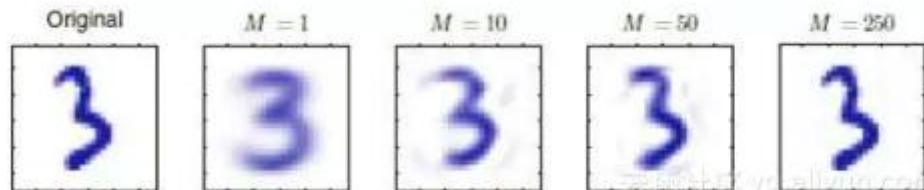
- 算法中几个相关的概念：
 - **频繁项集：**在数据库中大量频繁出现的数据集合。例如购物单数据中{'啤酒'}、{'尿布'}、{'啤酒', '尿布'}出现的次数都比较多。
 - **关联规则：**由集合 A，可以在某置信度下推出集合 B。即如果 A 发生了，那么 B 也很有可能会发生。例如购买了{'尿布'}的人很可能会购买{'啤酒'}。
 - **支持度：**指某频繁项集在整个数据集中的比例。假设数据集有 10 条记录，包含{'啤酒', '尿布'}的有 5 条记录，那么{'啤酒', '尿布'}的支持度就是 $5/10 = 0.5$ 。
 - **置信度：**有关联规则如{'尿布'} -> {'啤酒'}，它的置信度为 {'尿布'} -> {'啤酒'}
假设{'尿布', '啤酒'}的支持度为 0.45，{'尿布'}的支持度为 0.5，则 {'尿布'} -> {'啤酒'} 的置信度为 $0.45 / 0.5 = 0.9$ 。

9. PCA降维：减少数据维度，降低数据复杂度

- 降维是指将原高维空间中的数据点映射到低维度的空间中。因为高维特征的数目巨大，距离计算困难，分类器的性能会随着特征数的增加而下降；减少高维的冗余信息所造成的误差，可以提高识别的精度。



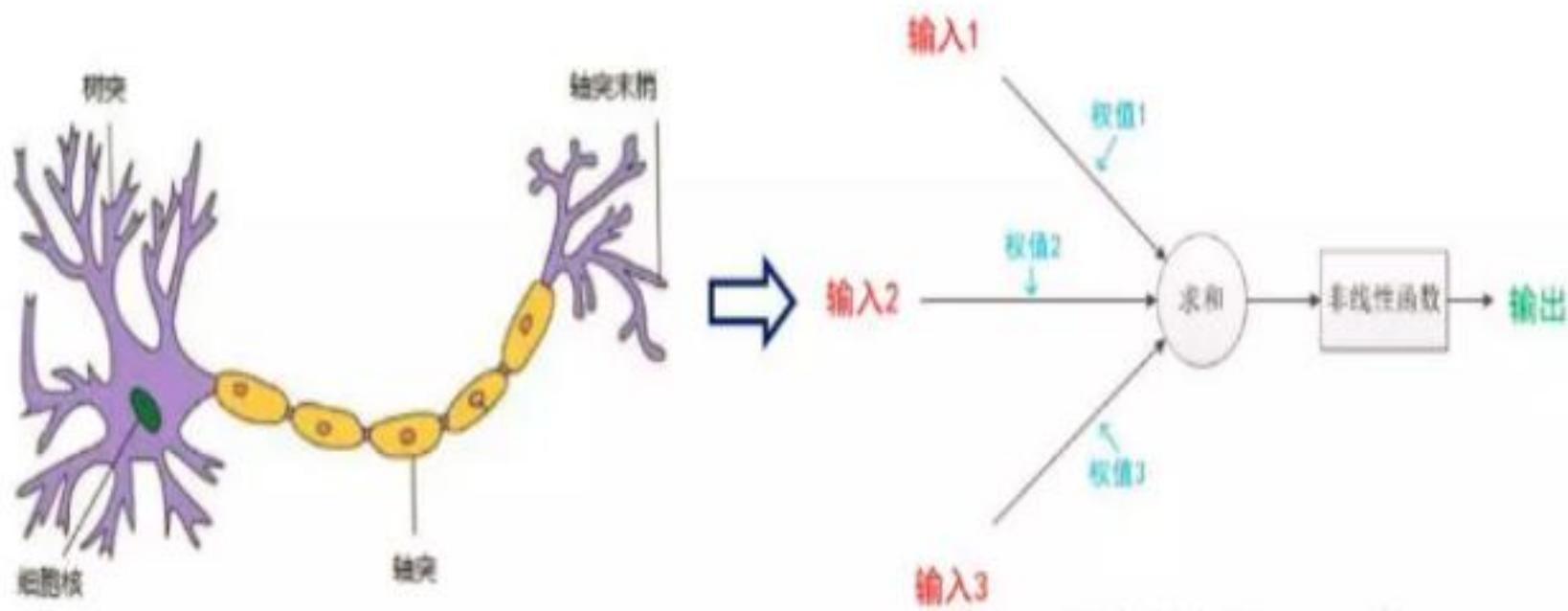
云栖社区 yq.aliyun.com



云栖社区 yq.aliyun.com

10. 人工神经网络：逐层抽象，逼近任意函数

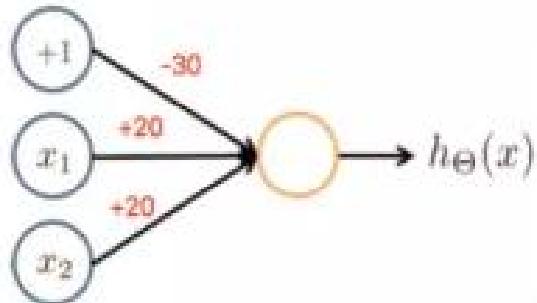
- 前面介绍了九种传统的机器学习算法，现在介绍一下深度学习的基础：人工神经网络。它是模拟人脑神经网络而设计的模型，由多个节点（人工神经元）相互联结而成，可以用来对数据之间的复杂关系进行建模。



- 例如利用单层神经网络实现逻辑与门和同或门

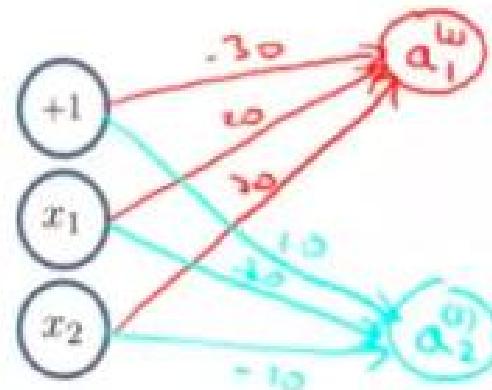
与门AND

$$h_{\Theta}(x) = g(-30 + 20x_1 + 20x_2)$$



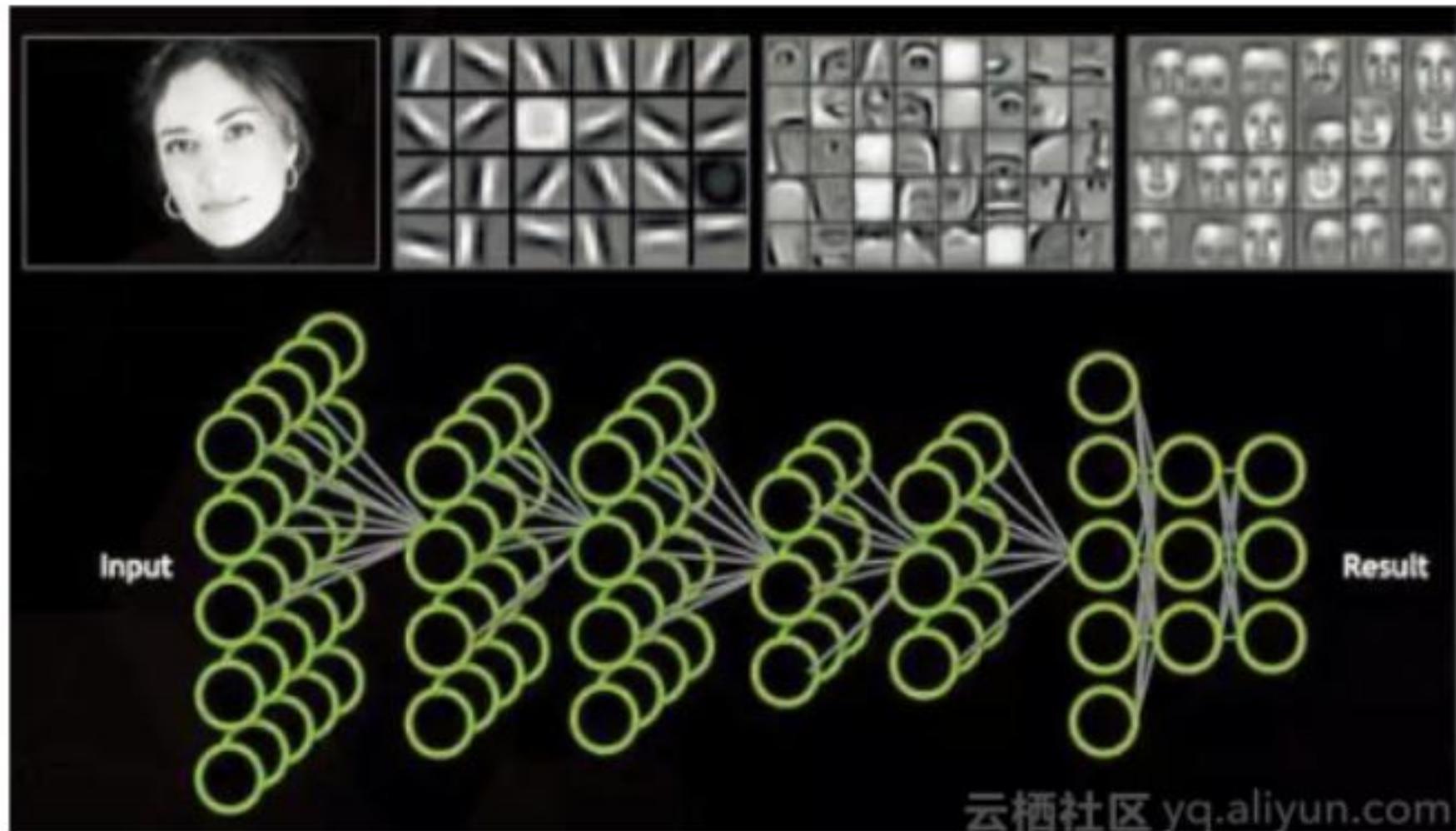
x1	x2	$h_{\Theta}(x)$
0	0	$g(-30) \approx 0$
0	1	$g(-10) \approx 0$
1	0	$g(-10) \approx 0$
1	1	$g(10) \approx 1$

同或门XNOR



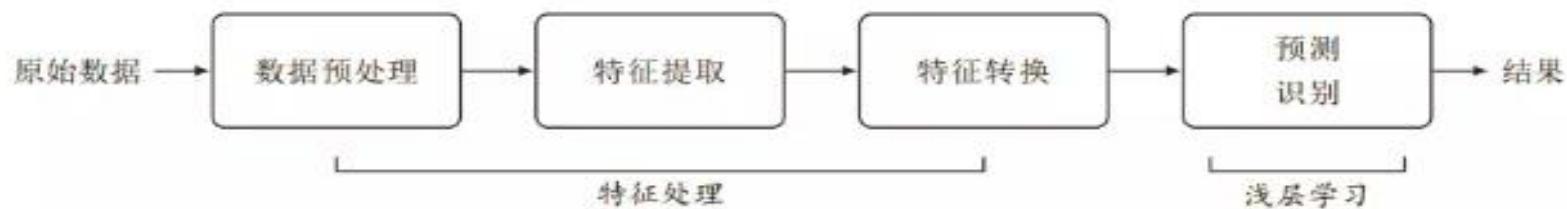
x1	x2	$a_1(2)$	$a_2(2)$	$h_{\Theta}(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

- 多层神经网络的每一层神经元学习到的是前一层神经元值的更抽象的表示，通过抽取更抽象的特征来对事物进行区分，从而获得更好的区分与分类能力。

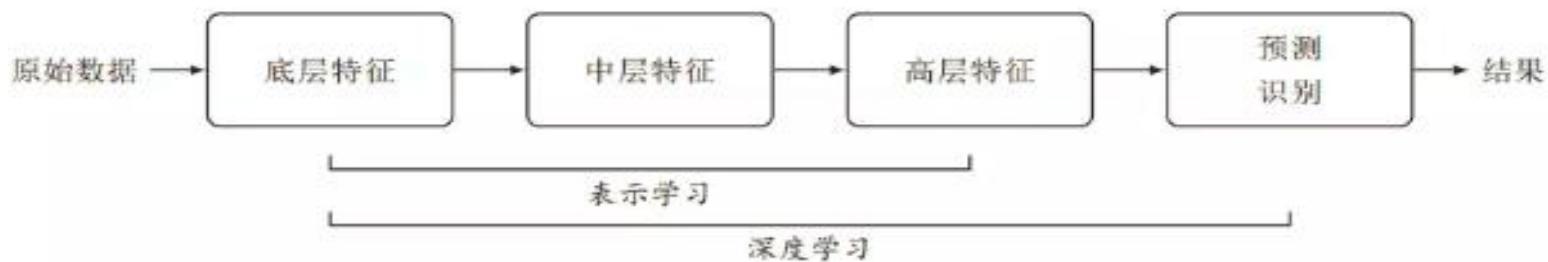


11. 深度学习：赋予人工智能以璀璨的未来

- 深度学习就是一种基于对数据进行表征学习的方法，使用多层网络，能够学习抽象概念，同时融入自我学习，逐步从大量的样本中逐层抽象出相关的概念，然后做出理解，最终做出判断和决策。通过构建具有一定“深度”的模型，可以让模型来自动学习好的特征表示（从底层特征，到中层特征，再到高层特征），从而最终提升预测或识别的准确性。



传统机器学习的数据处理流程。



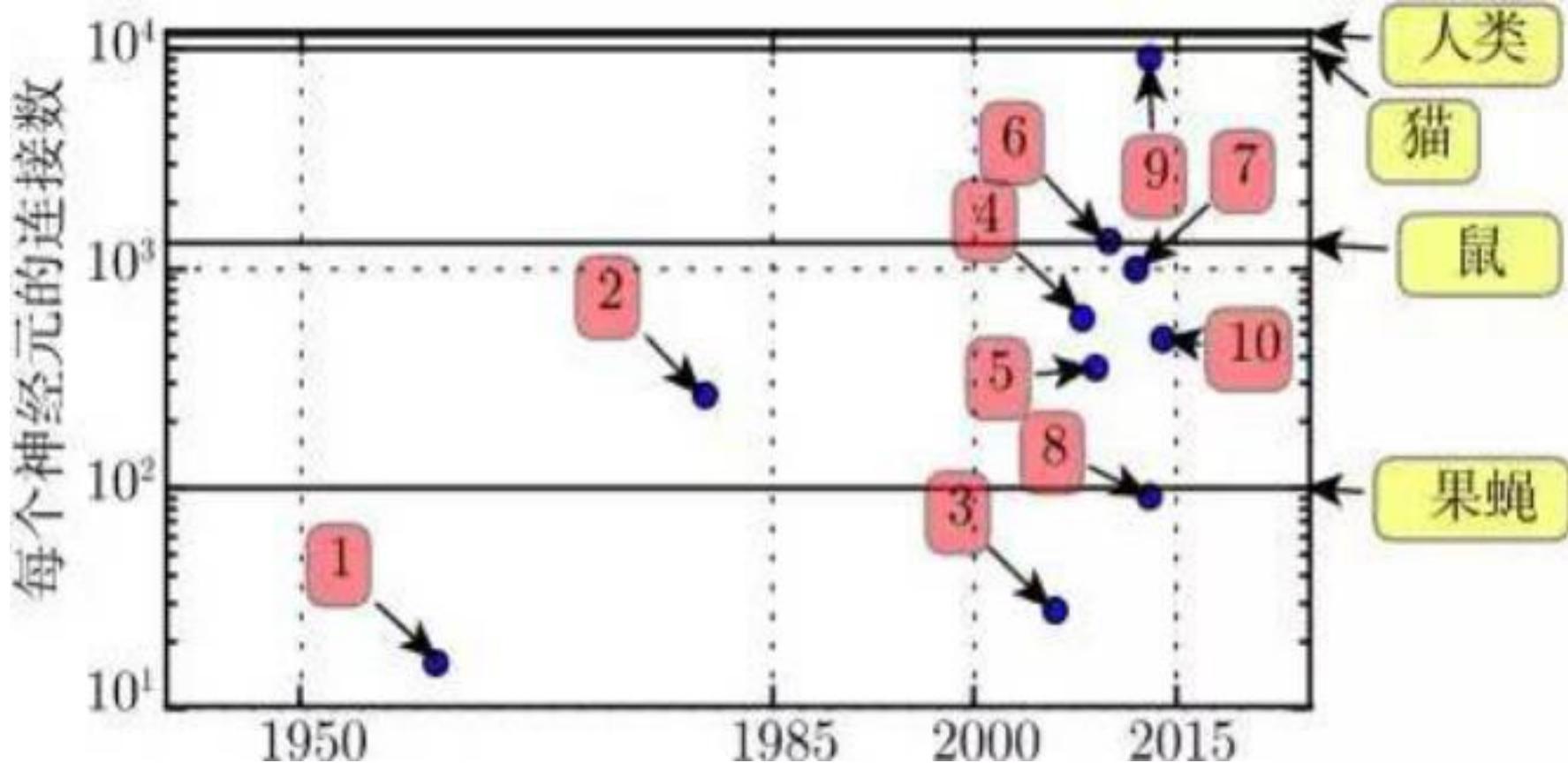
深度学习的数据处理流程。

深度学习的历史变迁：

深度学习经历了三次浪潮：

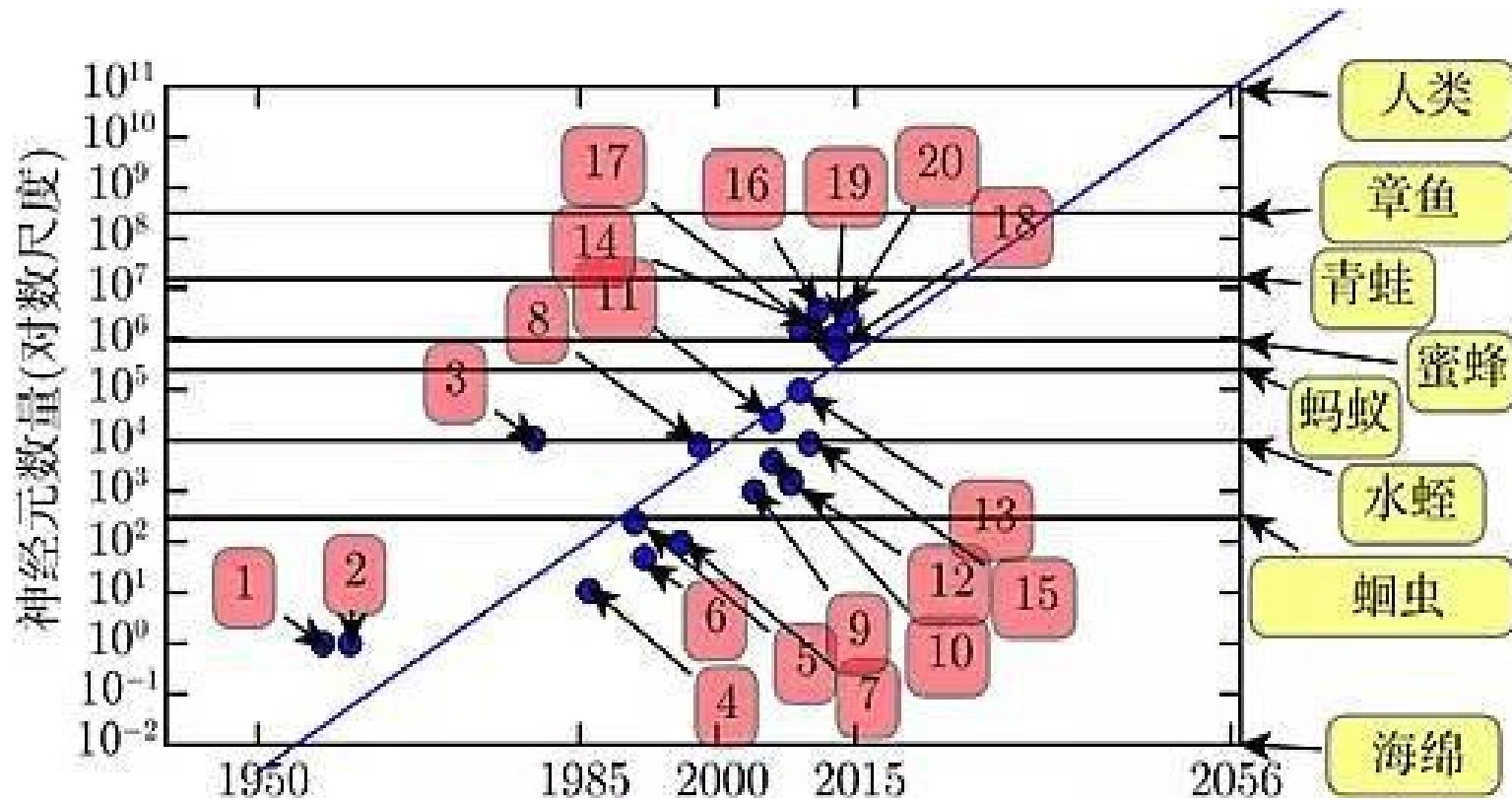
- 20世纪40年代～60年代，深度学习的雏形出现在控制论中；
 - 20世纪80年代～90年代，深度学习表现为联结主义；
 - 2006年以后，正式以深度学习之名复兴。
-
- 第一次浪潮：以感知机和线性模型为代表
不能解决与或问题
 - 第二次浪潮：以多层感知机和BP模型为代表
以统计学为基础，应用核函数和图模型的支持向量机算法（SVM算法）等各種浅层有监督的机器学习模型广泛应用，且深度神经网络不可训练
 - 第三次浪潮：以无监督学习为代表。
解决了深层神经网络的计算能力问题；解决了深度神经网络后向误差反馈梯度消失的问题。

与日俱增的每个神经元的连接数



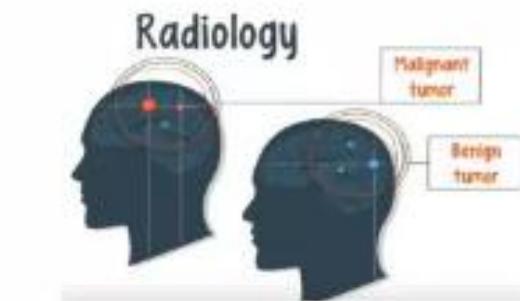
- 最初，人工神经网络中神经元之间的连接数受限于硬件能力。而现在，神经元之间的连接数大多是出于设计考虑。一些人工神经网络中每个神经元的连接数与猫一样多，并且对于其他神经网络来说，每个神经元的连接数与较小哺乳动物（如小鼠）一样多，这种情况是非常普遍的。甚至人类大脑每个神经元的连接数也没有过高的数量。
- 1. 自适应线性单元 (Widrow and Hoff, 1960)； 2. 神经认知机 (Fukushima, 1980)； 3. GPU 加速卷积网络 (Chellapilla et al., 2006)； 4. 深度玻尔兹曼机 (Salakhutdinov and Hinton, 2009a)； 5. 无监督卷积网络 (Jarrett et al., 2009b)； 6. GPU- 加速多层感知机 (Ciresan et al., 2010)； 7. 分布式自编码器 (Le et al., 2012)； 8. Multi-GPU 卷积网络 (Krizhevsky et al., 2012a)； 9. COTS HPC 无监督卷积网络 (Coates et al., 2013)； 10. GoogLeNet (Szegedy et al., 2014a)

与日俱增的神经网络规模



- 自从引入隐藏单元，人工神经网络的规模大约每 2.4 年翻一倍。
- 1. 感知机 (Rosenblatt, 1958, 1962)；2. 自适应线性单元 (Widrow and Hoff, 1960)；3. 神经认知机 (Fukushima, 1980)；4. 早期后向传播网络 (Rumelhart et al., 1986b)；5. 用于语音识别的循环神经网络 (Robinson and Fallside, 1991)；6. 用于语音识别的多层感知机 (Bengio et al., 1991)；7. 均匀场 sigmoid 信念网络 (Saul et al., 1996)；8. LeNet5 (LeCun et al., 1998c)；9. 回声状态网络 (Jaeger and Haas, 2004)；10. 深度信念网络 (Hinton et al., 2006a)；11. GPU- 加速卷积网络 (Chellapilla et al., 2006)；12. 深度玻尔兹曼机 (Salakhutdinov and Hinton, 2009a)；13. GPU 加速深度信念网络 (Raina et al., 2009a)；14. 无监督卷积网络 (Jarrett et al., 2009b)；15. GPU- 加速多层感知机 (Ciresan et al., 2010)；16. OMP-1 网络 (Coates and Ng, 2011)；17. 分布式自编码器 (Le et al., 2012)；18. MultiGPU 卷积网络 (Krizhevsky et al., 2012a)；19. COTS HPC 无监督卷积网络 (Coates et al., 2013)；20. GoogLeNet (Szegedy et al., 2014a)

- 目前深度学习的应用十分广泛，例如图像识别、语音识别、机器翻译、自动驾驶、金融风控、智能机器人等。



Machine Learning in biology

↓
Supervised Unsupervised

$x \rightarrow y$

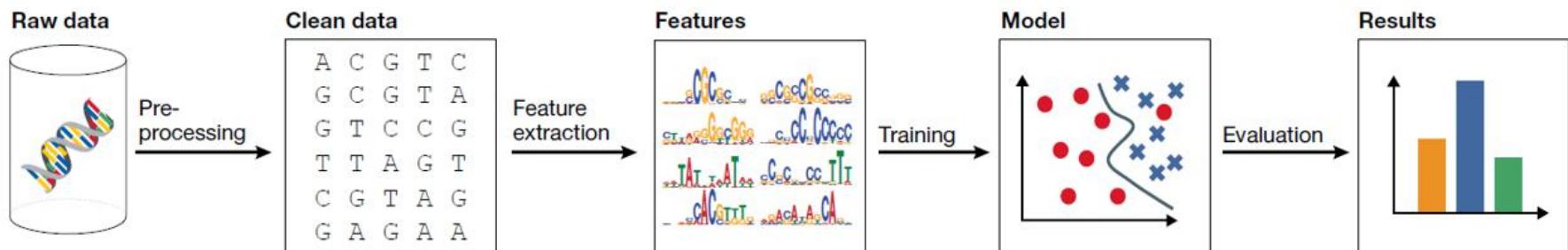


x



- Linear regression
 - Logistic regression
 - Random Forest
 - SVM
 - ...
- PCA
 - Factor analysis
 - Clustering
 - Outlier detection
 - ...

Machine learning algorithm: supervised and unsupervised



An exemplification of Machine learning in biology : classification model

Deep learning

Deep learning is a part of machine learning.

Deep:

Complex Model : Multi-layer characters, Many parameters (**Curse of dimensionality**)

Training data & Testing data : a big volume of data

Adjustment: A single model construction could cost one week

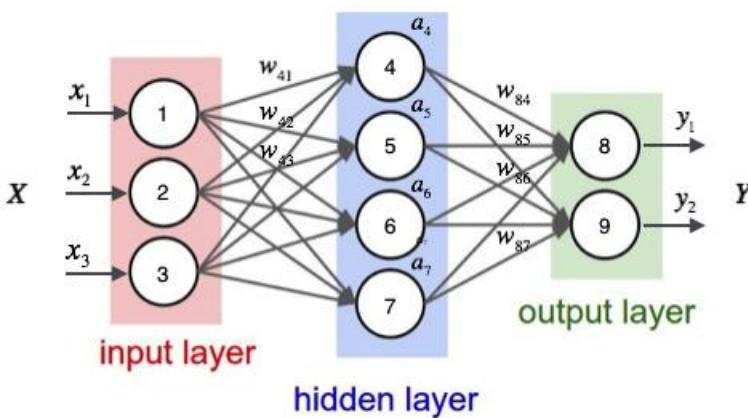
几乎所有深度学习算法都可以被描述为一个简单配方：

特定数据集、代价函数、优化过程和模型

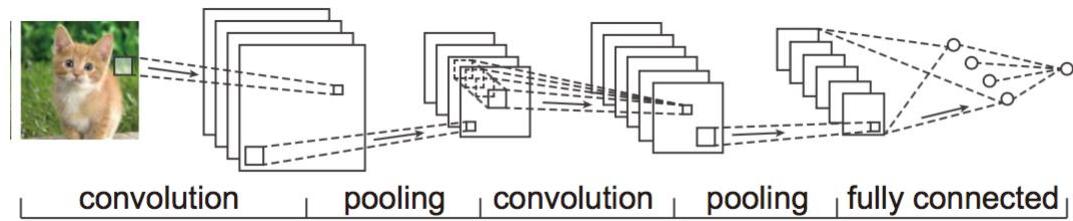


Deep learning

Now, neural network and convolution neural network models that work best



Neural Network

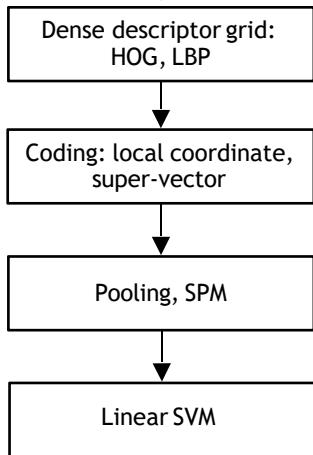


Convolution Neural Network

IMAGENET Large Scale Visual Recognition Challenge

Year 2010

NEC-UIUC

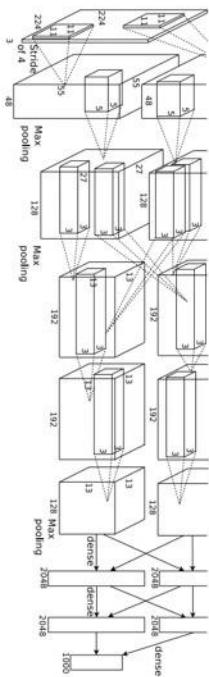


[Lin CVPR 2011]

Lion image by Swissfrog is licensed under CC BY 3.0

Year 2012

SuperVision



[Krizhevsky, NIPS 2012]

Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

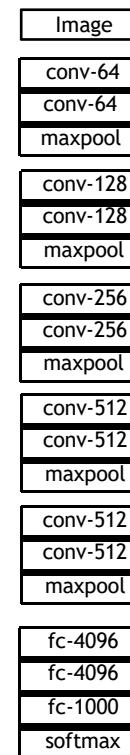
Year 2014

GoogLeNet



[Szegedy arxiv 2014]

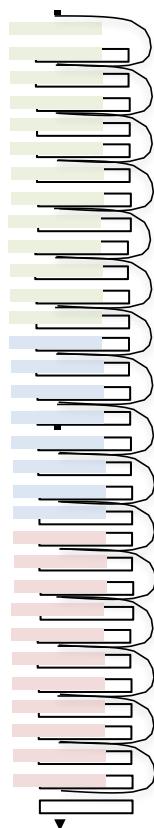
VGG



[Simonyan arxiv 2014]

Year 2015

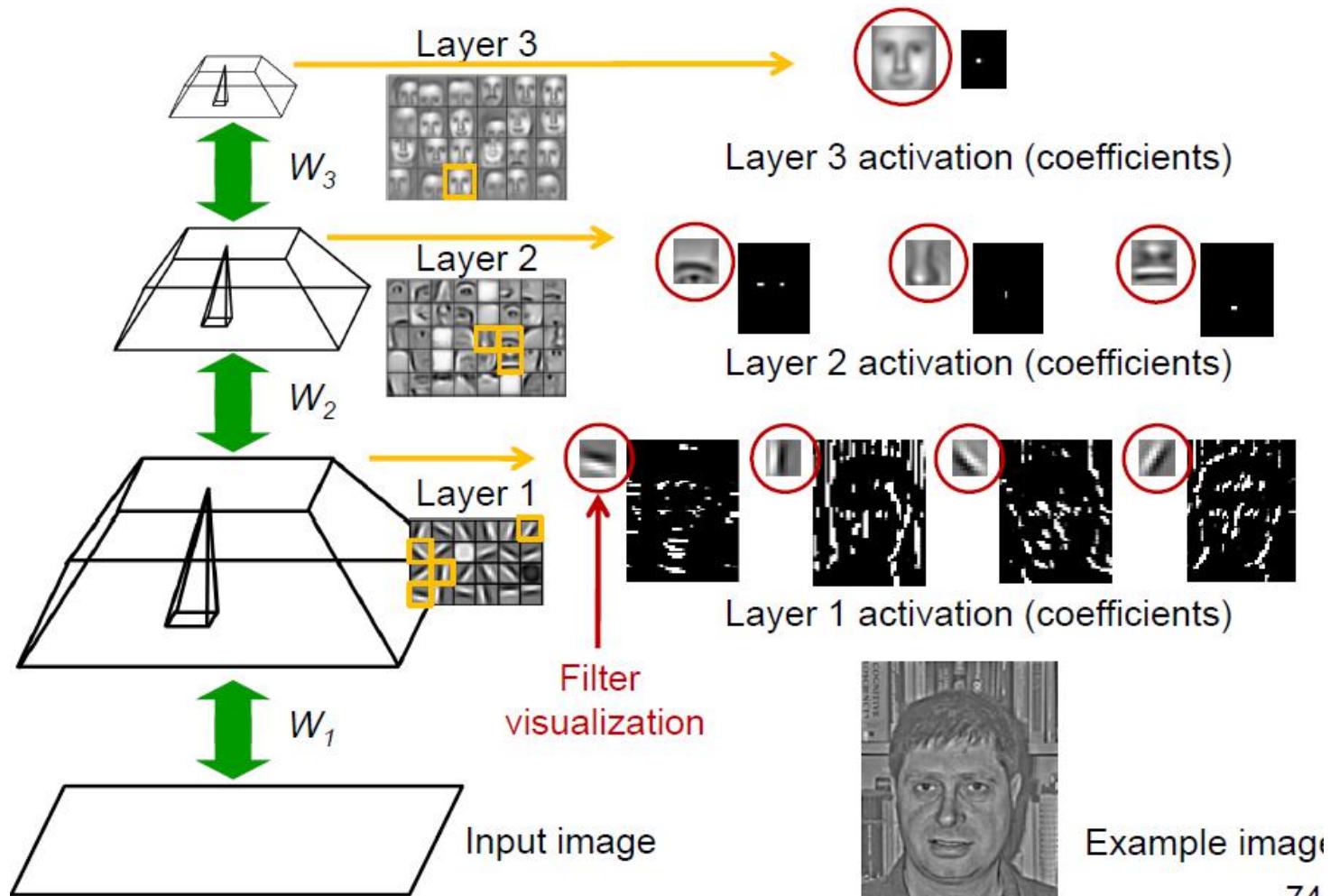
MSRA



[He ICCV 2015]

深度学习的应用

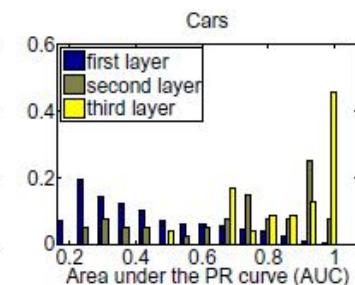
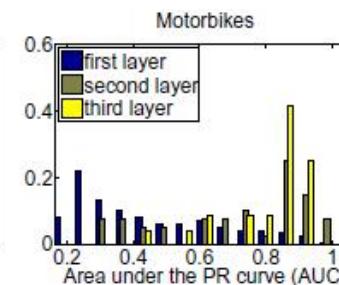
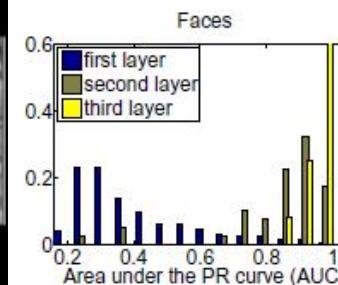
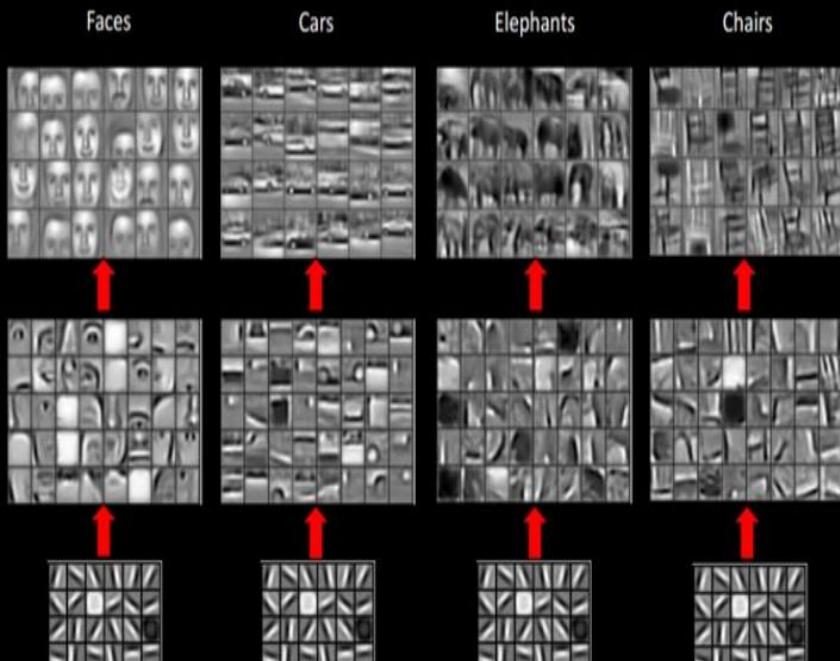
- 深度学习在图像识别上的应用



深度学习的应用

- 深度学习在图像识别上的应用

Features learned from training on different object classes.

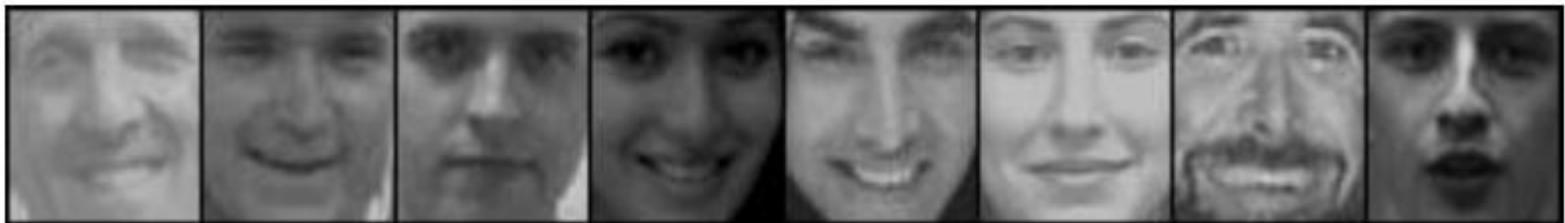


Features	Faces	Motorbikes	Cars
First layer	0.39±0.17	0.44±0.21	0.43±0.19
Second layer	0.86±0.13	0.69±0.22	0.72±0.23
Third layer	0.95±0.03	0.81±0.13	0.87±0.15

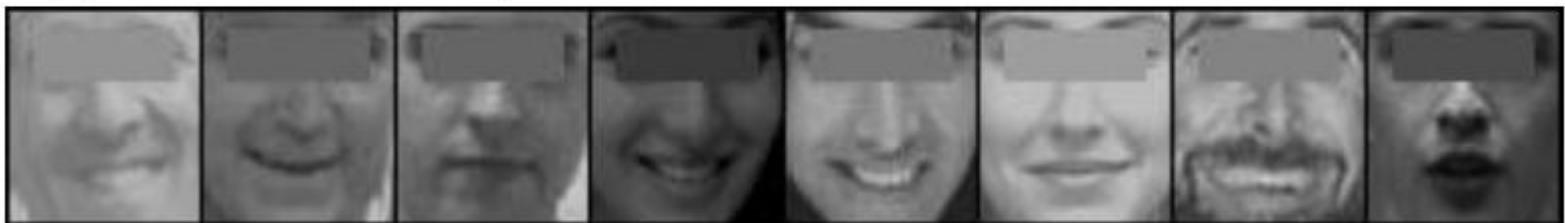
深度学习的应用

- 深度学习在图像识别上的应用

originals



Type 1 occlusion: eyes

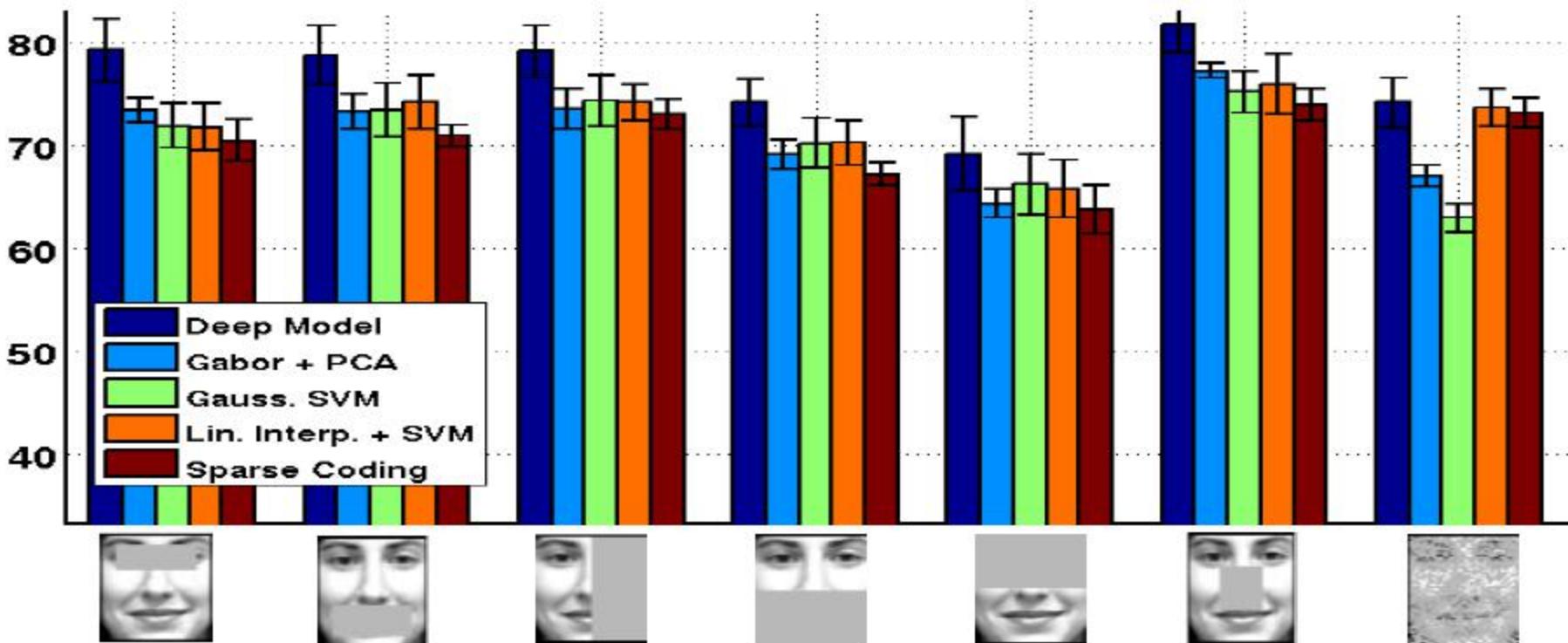


Restored images



深度学习的应用

- 深度学习在图像识别上的应用



深度学习的应用

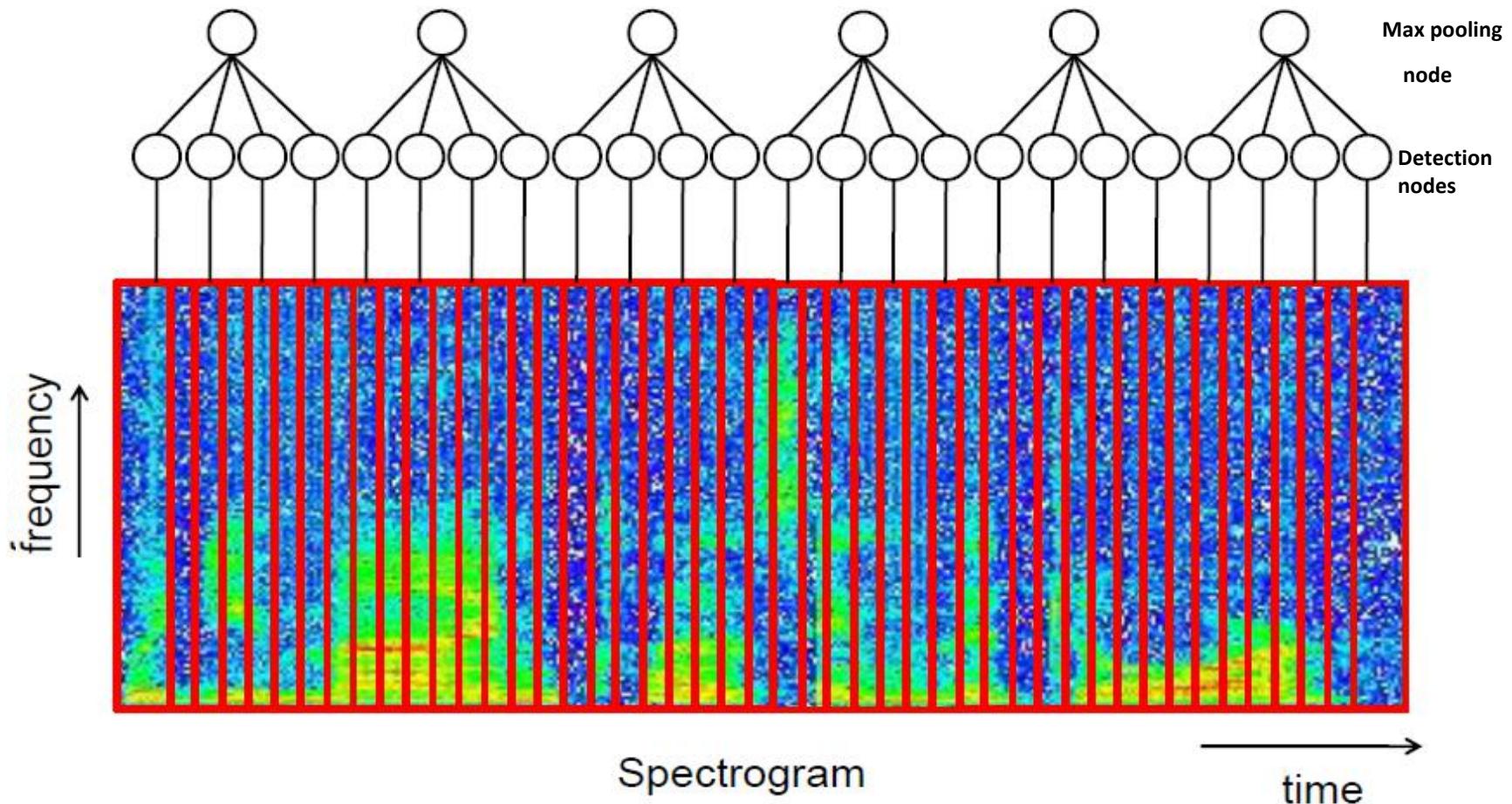
- 深度学习在图像识别上的应用

< Caltech 256 >

# of training images	30	60
Griffin et al. [2]	34.10	-
vanGemert et al., PAMI 2010	27.17	-
ScSPM [Yang et al., CVPR 2009]	34.02	40.14
LLC [Wang et al., CVPR 2010]	41.19	47.68
Sparse CRBM [Sohn et al., ICCV 2011]	42.05	47.94

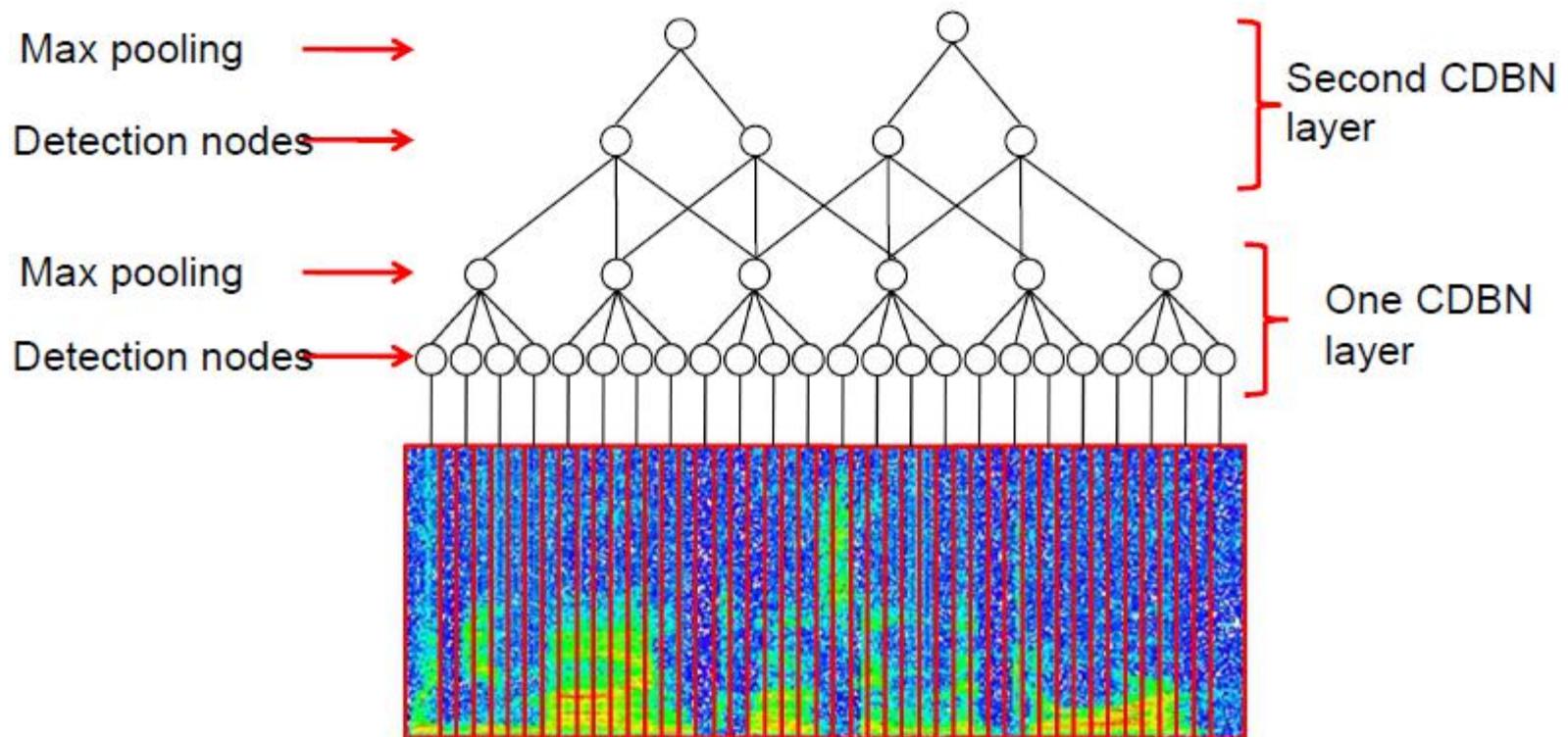
深度学习的应用

- 深度学习在音频识别上的应用



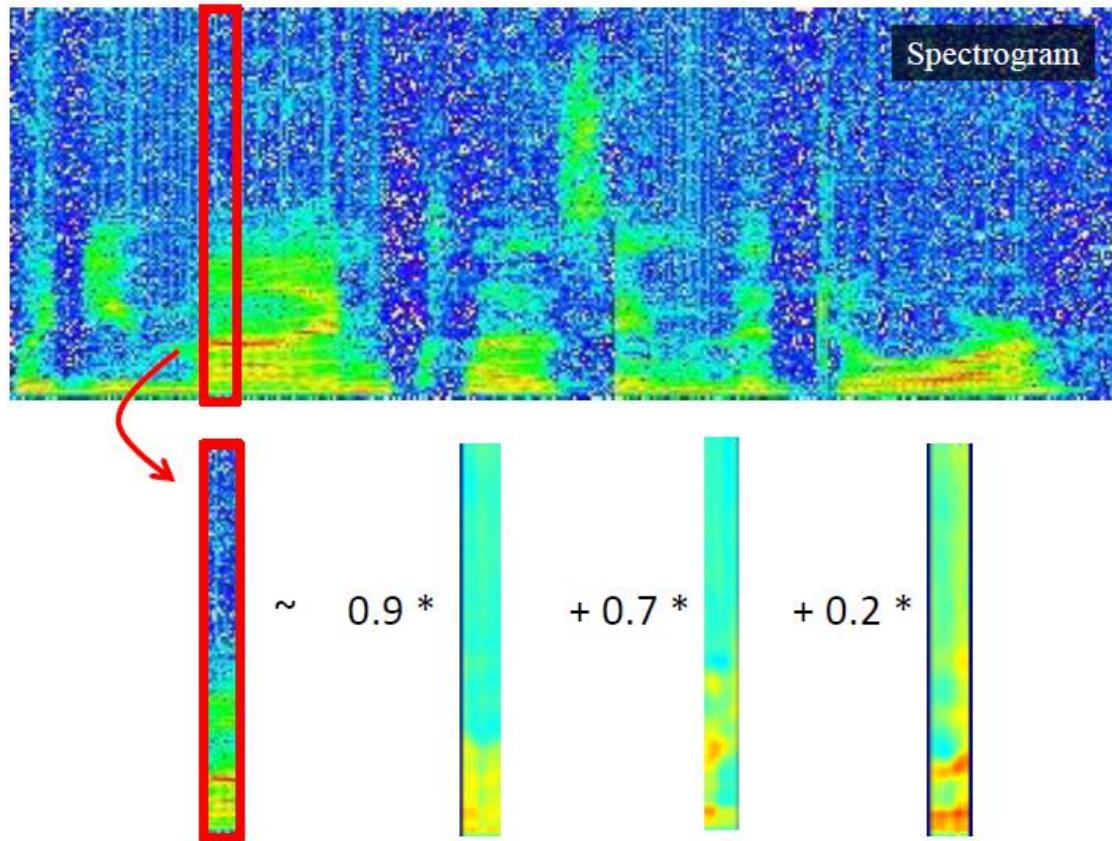
深度学习的应用

- 深度学习在音频识别上的应用



深度学习的应用

- 深度学习在音频识别上的应用



[Lee, Largman, Pham, Ng, NIPS 2009]

深度学习的应用

- 深度学习在音频识别上的应用
- Speaker identification

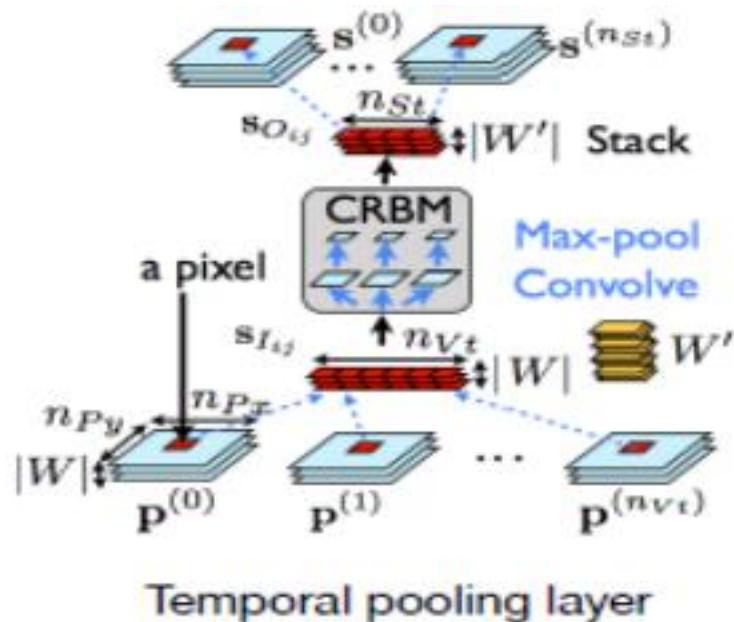
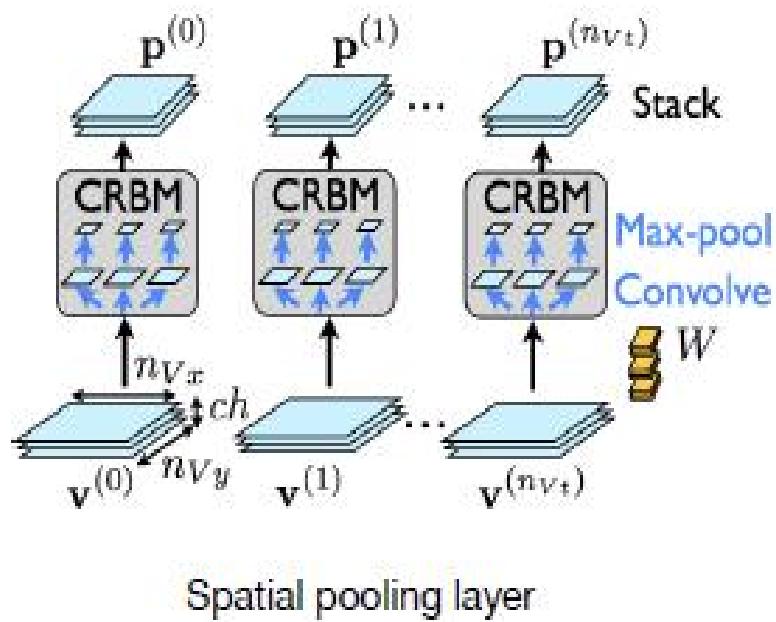
TIMIT Speaker identification	Accuracy
Prior art (Reynolds, 1995)	99.7%
Convolutional DBN	100.0%

- Phone classification

TIMIT Phone classification	Accuracy
Clarkson et al. (1999)	77.6%
Petrov et al. (2007)	78.6%
Sha & Saul (2006)	78.9%
Yu et al. (2009)	79.2%
Convolutional DBN	80.3%
Transformation-invariant RBM (Sohn et al., ICML 2012)	81.5%

深度学习的应用

- 深度学习在视频识别上的应用



深度学习的应用

- 深度学习在视频识别上的应用

Video Activity recognition (Hollywood 2 benchmark)



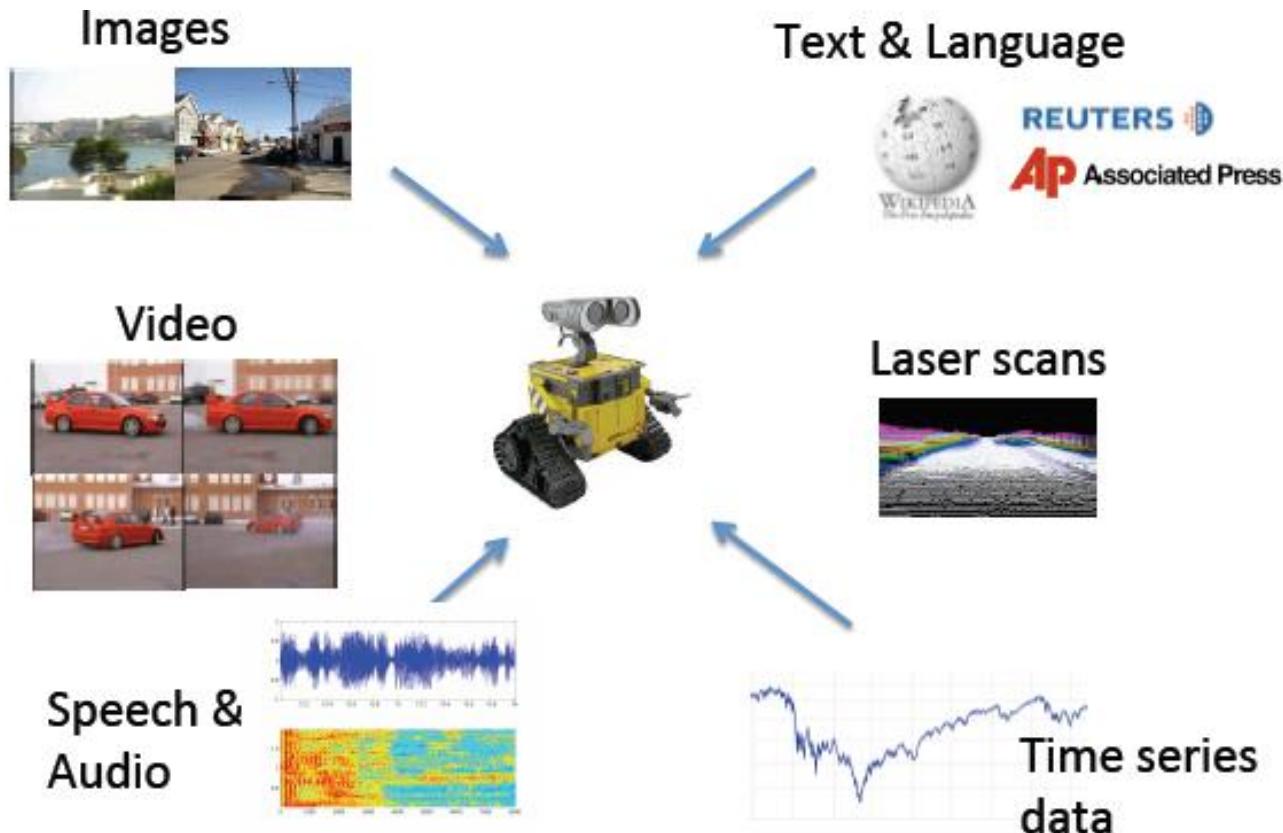
Method	Accuracy
Hessian + ESURF [Williems et al 2008]	38%
Harris3D + HOG/HOF [Laptev et al 2003, 2004]	45%
Cuboids + HOG/HOF [Dollar et al 2005, Laptev 2004]	46%
Hessian + HOG/HOF [Laptev 2004, Williems et al 2008]	46%
Dense + HOG / HOF [Laptev 2004]	47%
Cuboids + HOG3D [Klaser 2008, Dollar et al 2005]	46%
Unsupervised feature learning (our method)	52%



Unsupervised feature learning significantly improves
on the previous state-of-the-art.

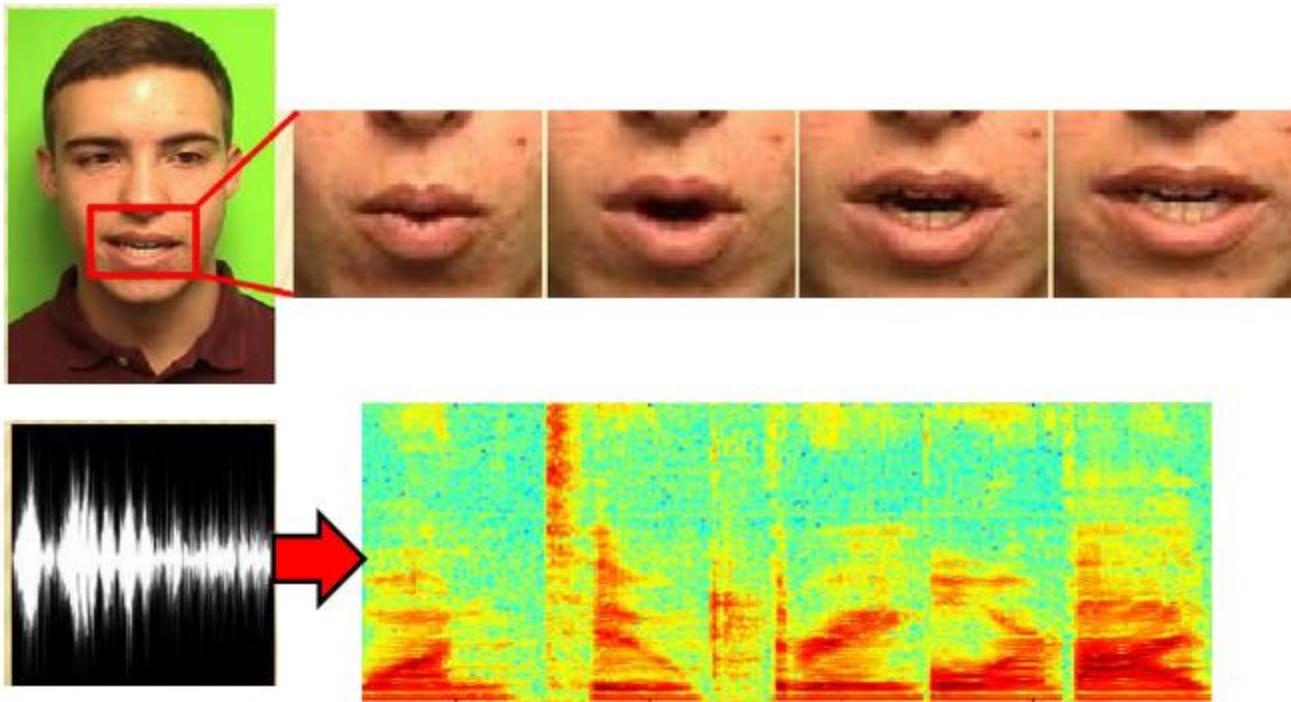
深度学习的应用

- 深度学习在多模态学习中的应用



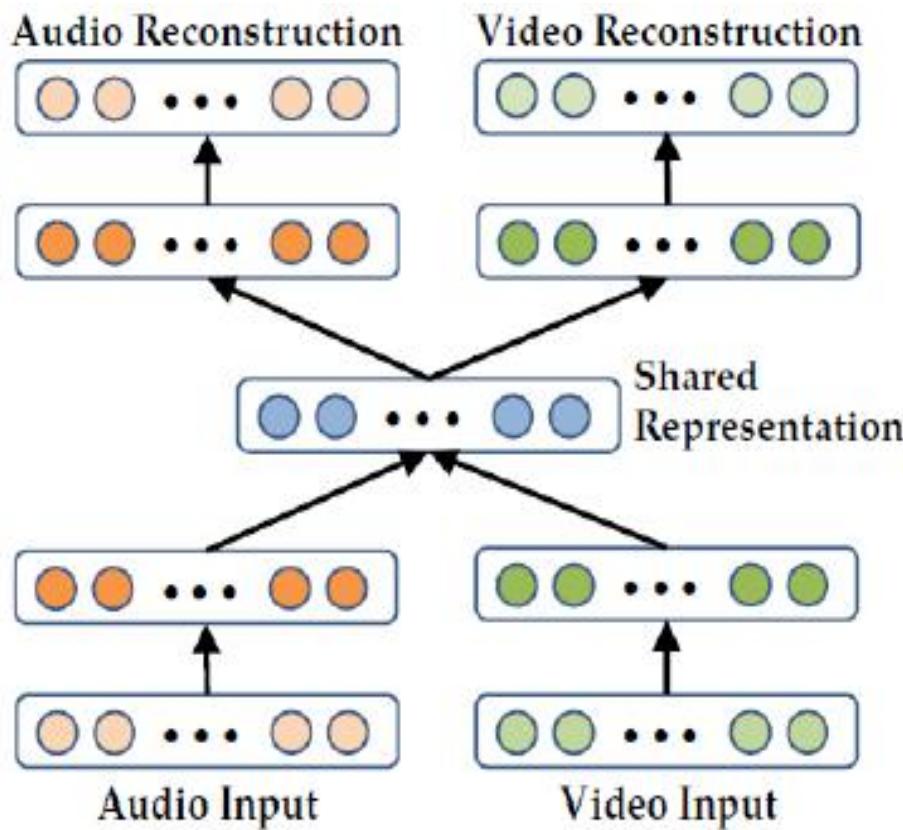
深度学习的应用

- 深度学习在多模态学习中的应用



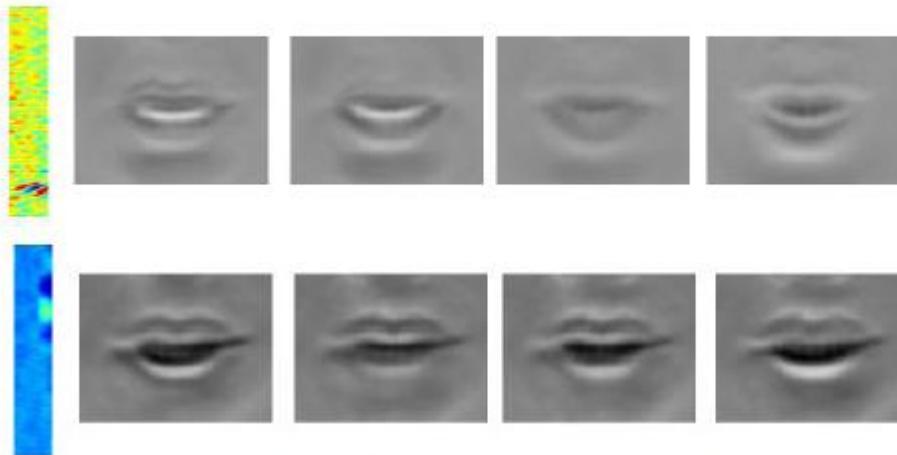
深度学习的应用

- 深度学习在多模态学习中的应用



深度学习的应用

- 深度学习在多模态学习中的应用
 - Visualization of learned filters



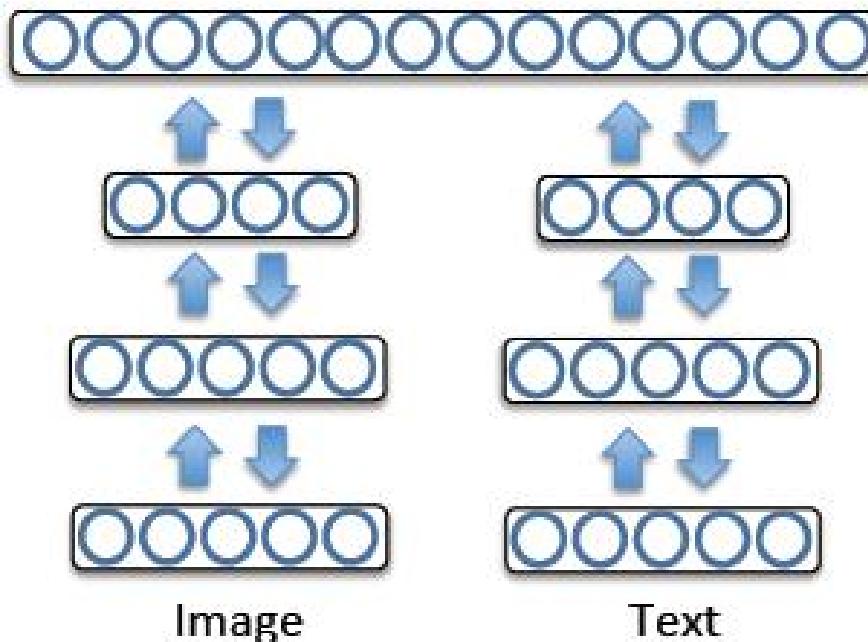
Audio(spectrogram) and Video features learned over 100ms windows

- Results: AVLetters Lip reading dataset

Method	Accuracy
Prior art (Zhao et al., 2009)	58.9%
Multimodal deep autoencoder (Ngiam et al., 2011)	65.8%

深度学习的应用

- 深度学习在多模态学习中的应用

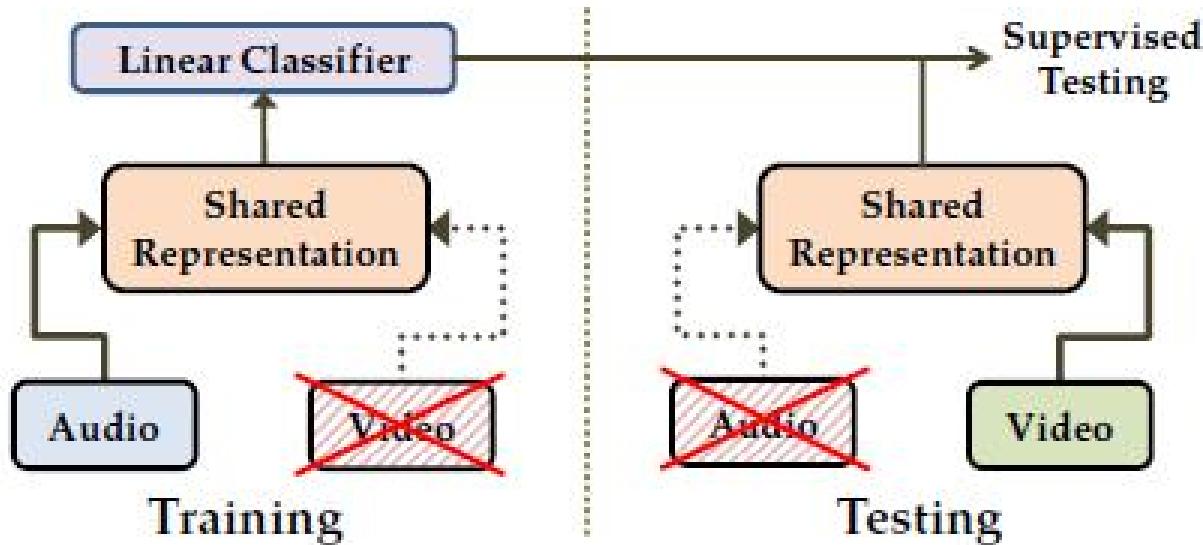


- Multimodal Inputs (images + text), 38 classes.

Learning Algorithm	Mean Average Precision
Image-text SVM	0.475
Image-text LDA	0.492
Multimodal DBN	0.566

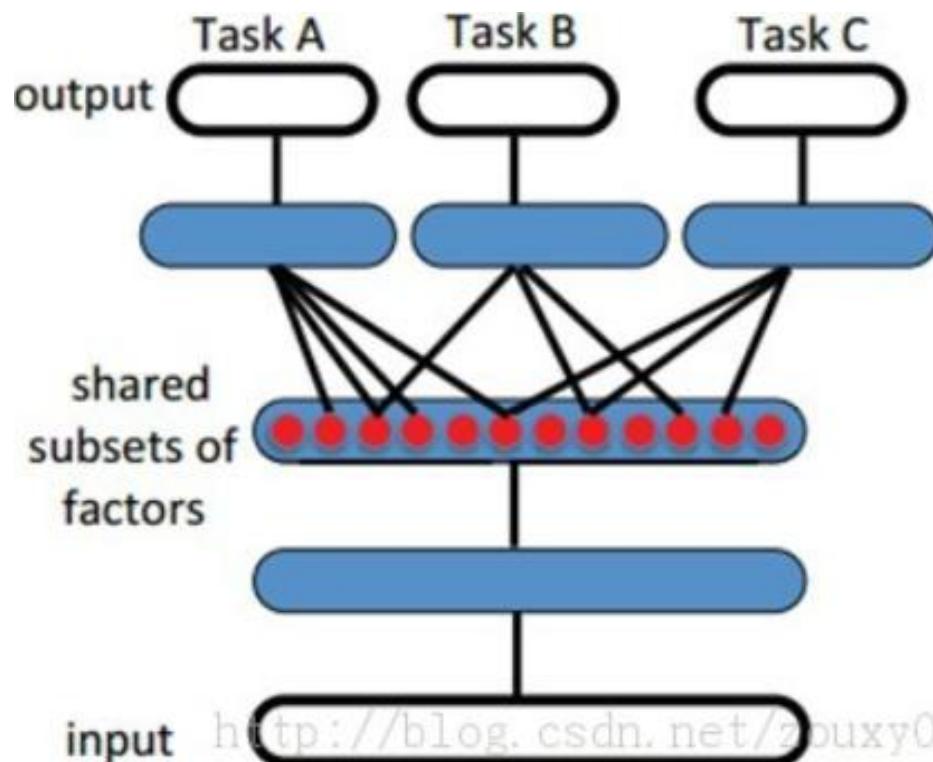
深度学习的应用

- 深度学习在多模态学习中的应用



深度学习的应用

- 深度学习在多任务学习中的应用



深度学习的应用

- 深度学习在多任务学习中的应用

- ✓ 在深度学习模型中，对于相关任务的联合学习，往往取得较好的特征表达；
- ✓ 多任务联合学习，能够增强损失函数的作用效能；

比如：单独进行人脸检测会比较难（光照、遮挡等因素），但是当人脸检测与人脸识别这两个相关的任务联合学习时，人脸检测的难度反而降低了。

深度学习的应用

- 基于深度学习的迁移学习应用

Background Knowledge

Millions of unlabeled images



Some labeled images



Bicycle



Dolphin



Elephant



Tractor

Learn to Transfer
Knowledge



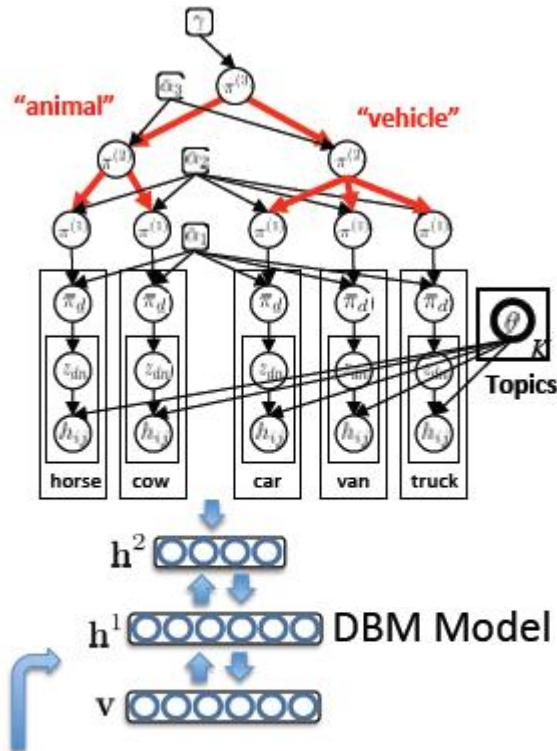
Learn novel concept
from one example

Test:
What is this?

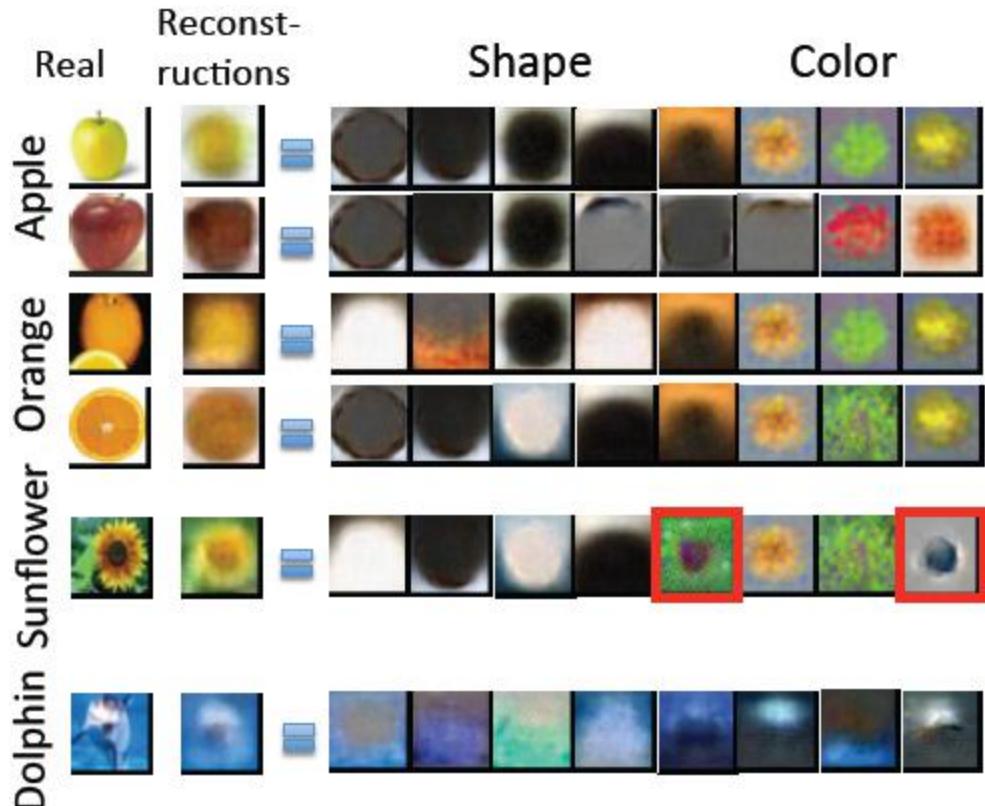


深度学习的应用

- 基于深度学习的迁移学习应用

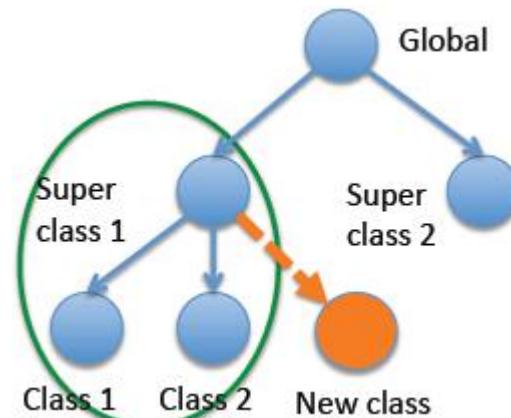
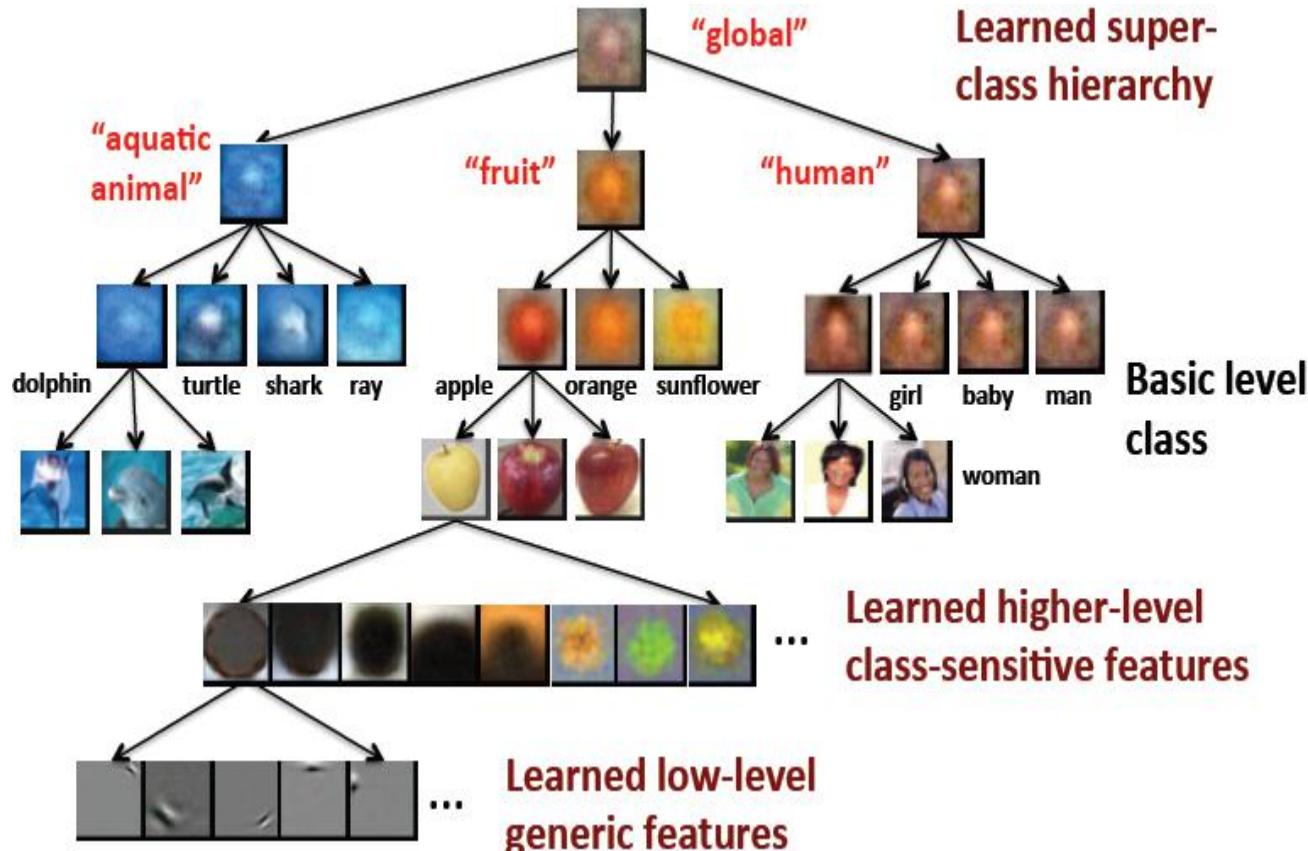


Low-level features:
replace GIST, SIFT



深度学习的应用

- 基于深度学习的迁移学习应用



深度学习的应用

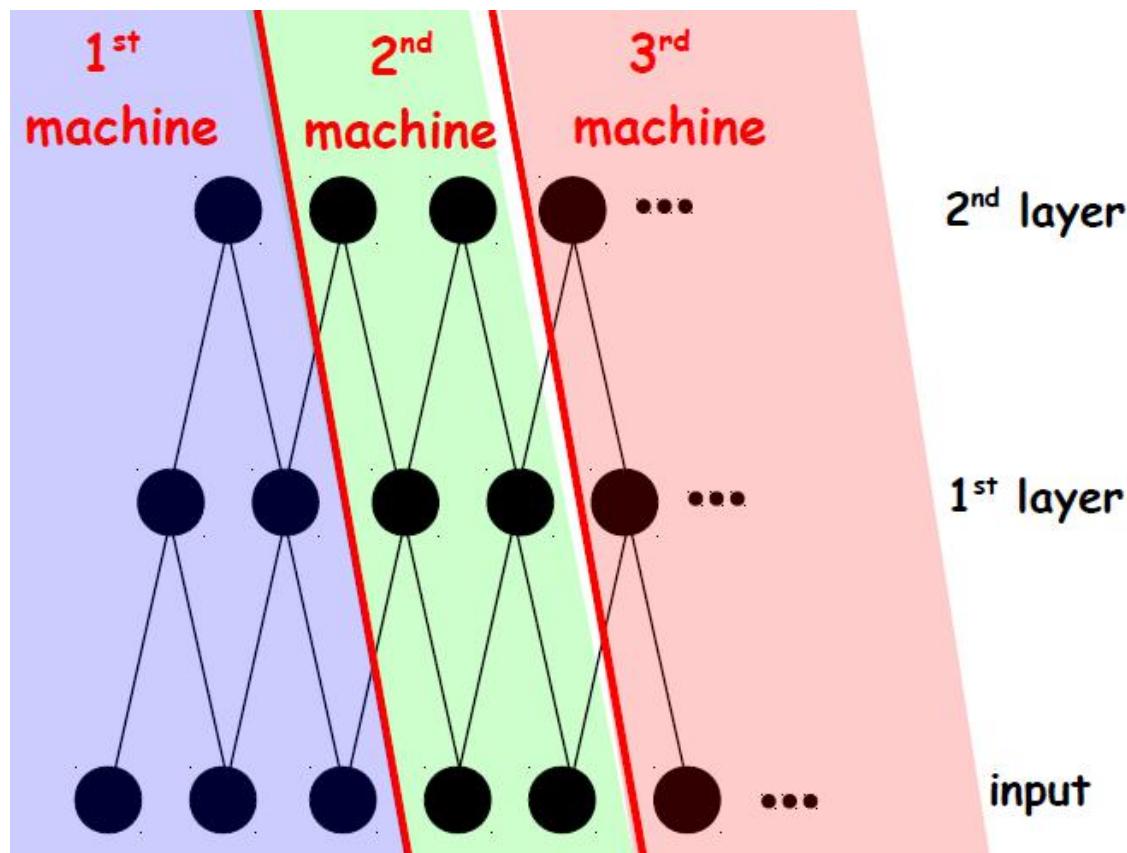
- 深度学习在大尺度数据集上的应用

□ 大尺度数据集：

- ✓ 样本总数 $>100M$,
- ✓ 类别总数 $>10K$,
- ✓ 特征维度 $>10K$

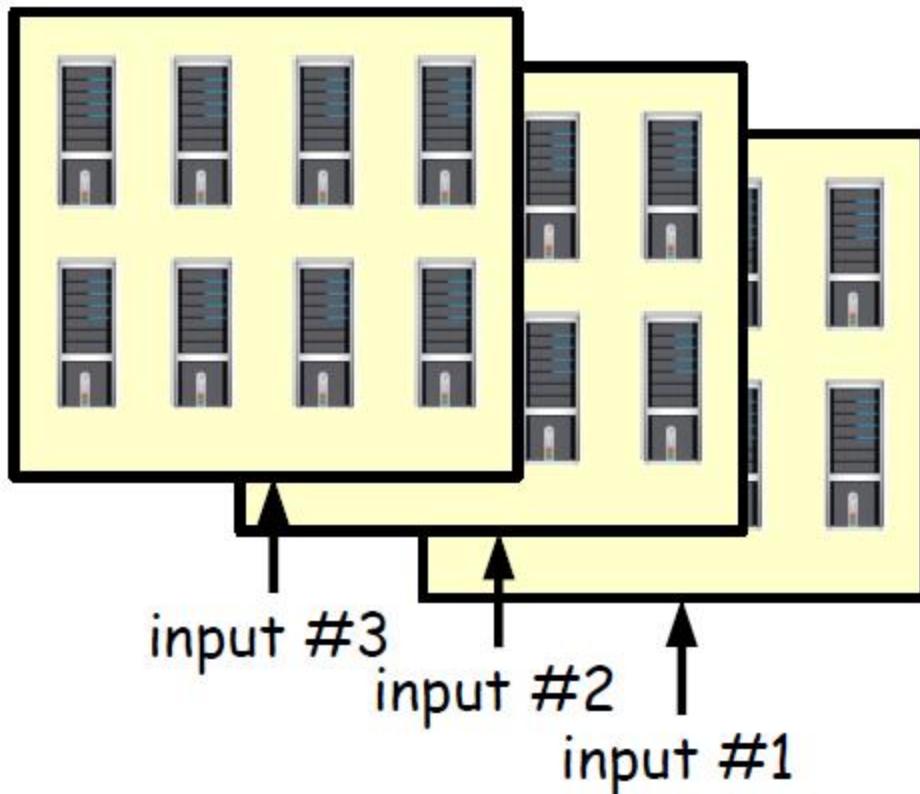
深度学习的应用

- 深度学习在大尺度数据集上的应用



深度学习的应用

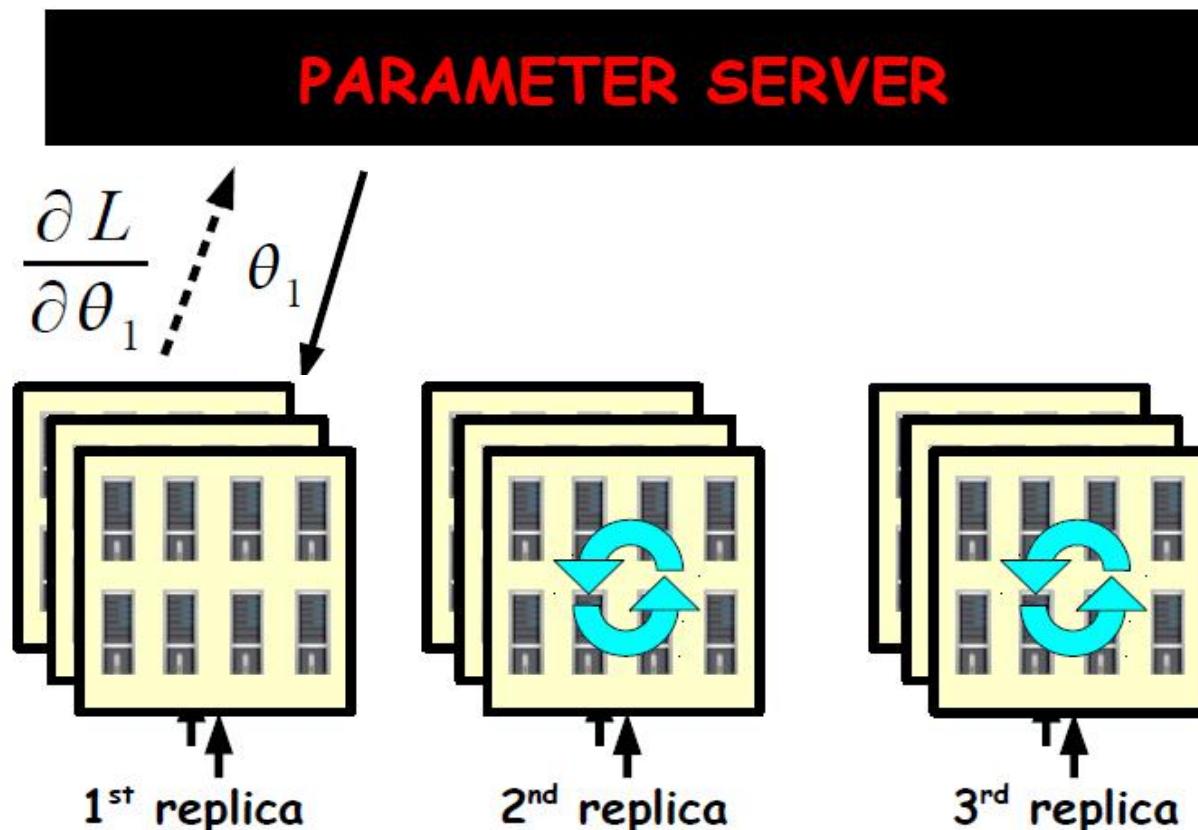
- 深度学习在大尺度数据集上的应用



MODEL
PARALLELISM
+
DATA
PARALLELISM

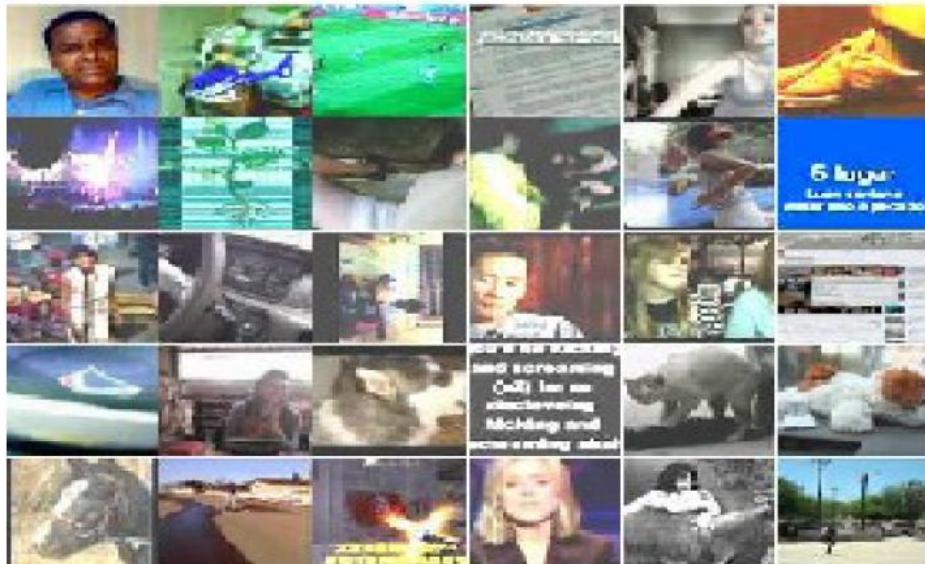
深度学习的应用

- 深度学习在大尺度数据集上的应用



深度学习的应用

- 深度学习在大尺度数据集上的应用



IMAGENET v.2011 (16M images, 20K categories)

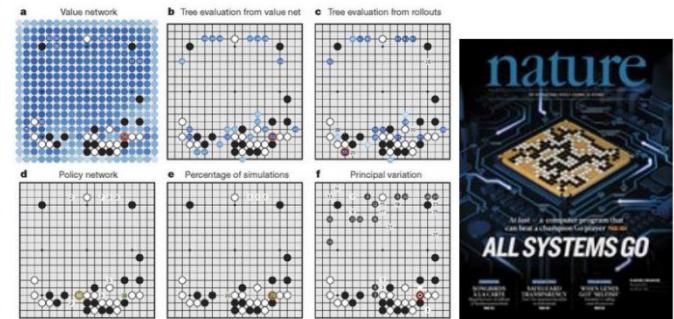
METHOD	ACCURACY %
Weston & Bengio 2011	9.3
Linear Classifier on deep features	13.1
Deep Net (from random)	13.6
Deep Net (from unsup.)	15.8

深度学习的应用

- 深度学习的State-of-the-art

Images	
CIFAR Object classification	Accuracy
Prior art (Ciresan et al., 2011)	80.5%
Stanford Feature learning	82.0%
NORB Object classification	
Prior art (Scherer et al., 2010)	94.4%
Stanford Feature learning	95.0%
Video	
Hollywood2 Classification	Accuracy
Prior art (Laptev et al., 2004)	48%
Stanford Feature learning	53%
KTH	Accuracy
Prior art (Wang et al., 2010)	92.1%
Stanford Feature learning	93.9%
YouTube	
Prior art (Liu et al., 2009)	71.2%
Stanford Feature learning	75.8%
UCF	
Prior art (Wang et al., 2010)	85.6%
Stanford Feature learning	86.5%
Text/NLP	
Paraphrase detection	Accuracy
Prior art (Das & Smith, 2009)	76.1%
Stanford Feature learning	76.4%
Sentiment (MR/MPQA data)	
Prior art (Nakagawa et al., 2010)	77.3%
Stanford Feature learning	77.7%
Multimodal (audio/video)	
AVLetters Lip reading	Accuracy
Prior art (Zhao et al., 2009)	58.9%
Stanford Feature learning	65.8%
Other unsupervised feature learning records:	
Pedestrian detection (Yann LeCun)	
Speech recognition (Geoff Hinton)	
PASCAL VOC object classification (Kai Yu)	

AlphaGo



The input to the policy network is a $19 \times 19 \times 48$ image stack consisting of 48 feature planes. The first hidden layer zero pads the input into a 23×23 image, then convolves k filters of kernel size 5×5 with stride 1 with the input image and applies a rectifier nonlinearity. Each of the subsequent hidden layers 2 to 12 zero pads the respective previous hidden layer into a 21×21 image, then convolves k filters of kernel size 3×3 with stride 1, again followed by a rectifier nonlinearity. The final layer convolves 1 filter of kernel size 1×1 with stride 1, with a different bias for each position, and applies a softmax function. The match version of AlphaGo used $k = 192$ filters; Fig. 2b and Extended Data Table 3 additionally show the results of training with $k = 128, 256$ and 384 filters.

policy network:

[$19 \times 19 \times 48$] Input

CONV1: 192 5×5 filters , stride 1, pad 2 => [$19 \times 19 \times 192$]

CONV2..12: 192 3×3 filters, stride 1, pad 1 => [$19 \times 19 \times 192$]

CONV: 1 1×1 filter, stride 1, pad 0 => [19×19] (*probability map of promising moves*)

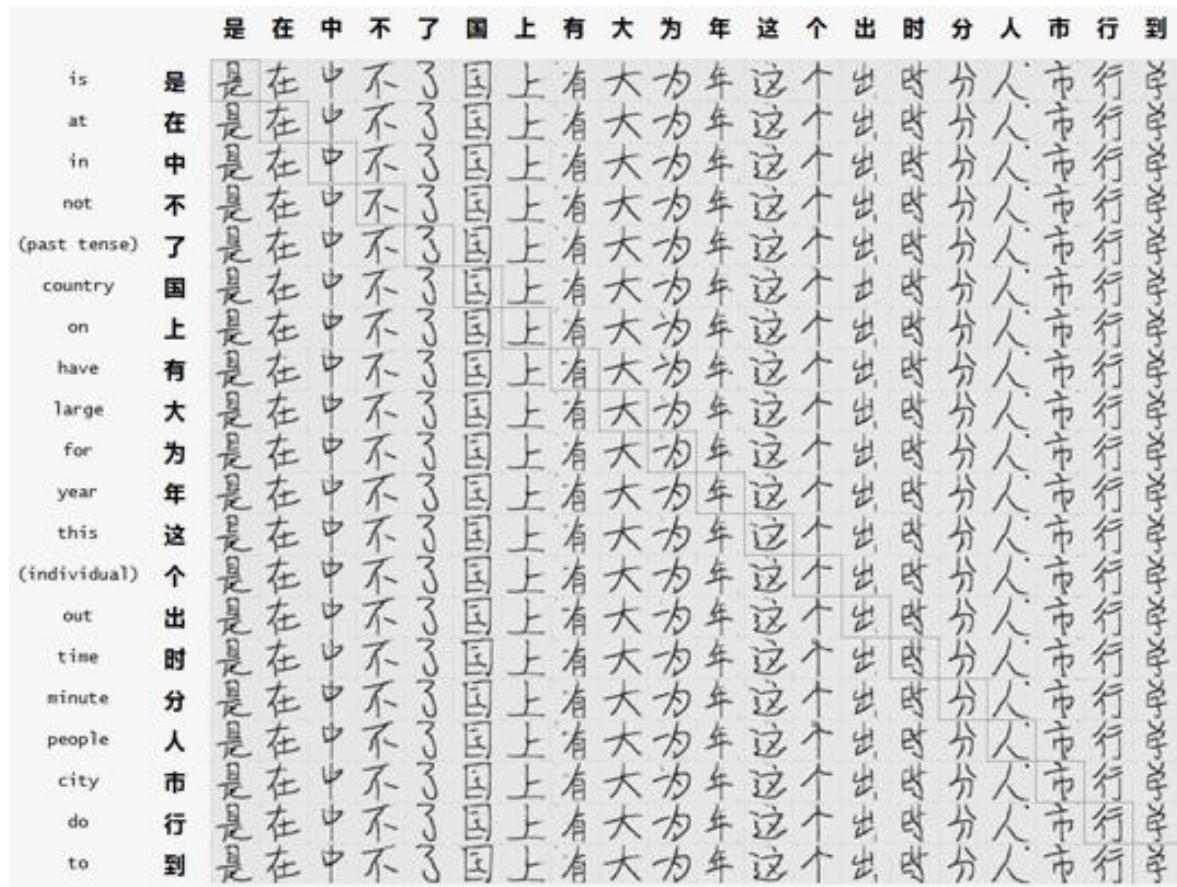
- 单机版：48 个CPU 和8 块GPU
- 走子速度：3 毫秒-2 微秒

Mask RCNN



Figure 4. More results of **Mask R-CNN** on COCO test images, using ResNet-101-FPN and running at 5 fps, with 35.7 mask AP (Table 1).

图像生成



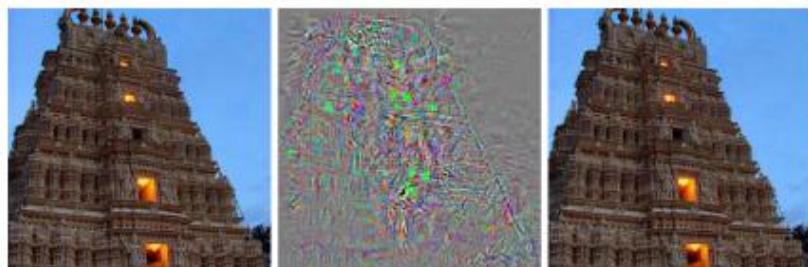
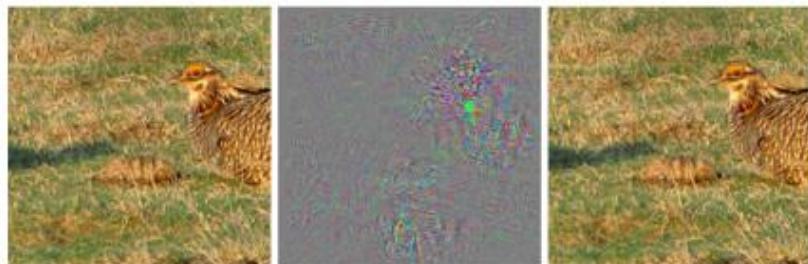
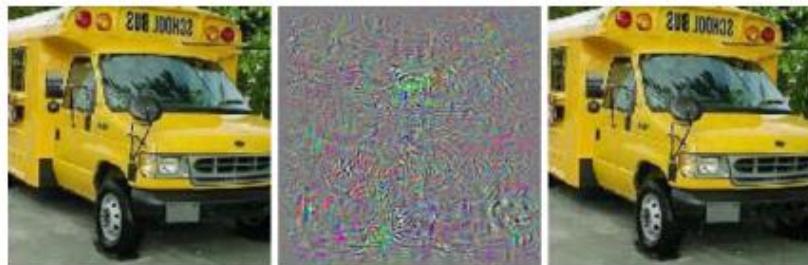
Deep Dream



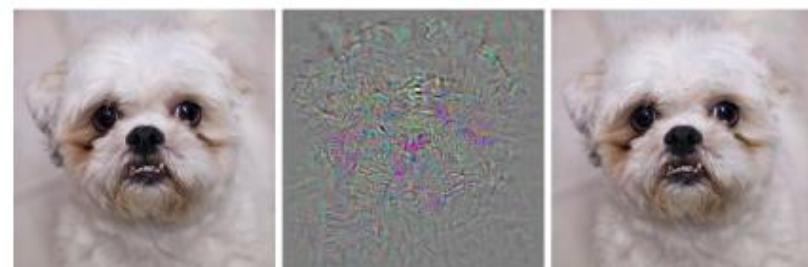
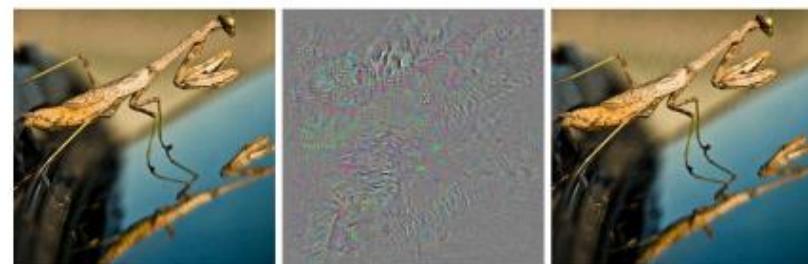
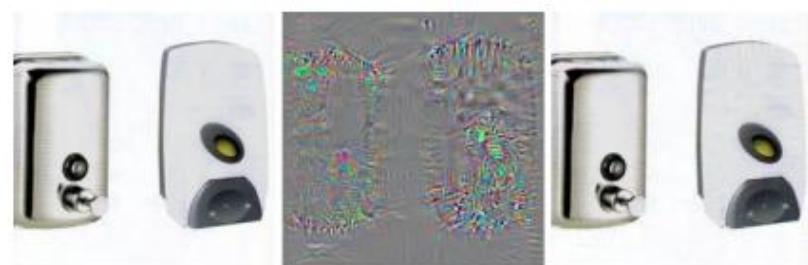
画风迁移



对抗样本



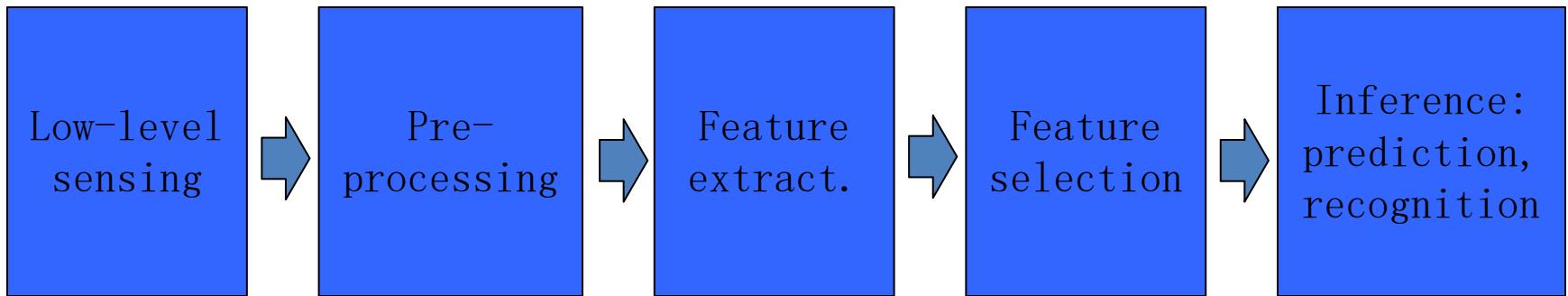
(a)



(b)

动 机

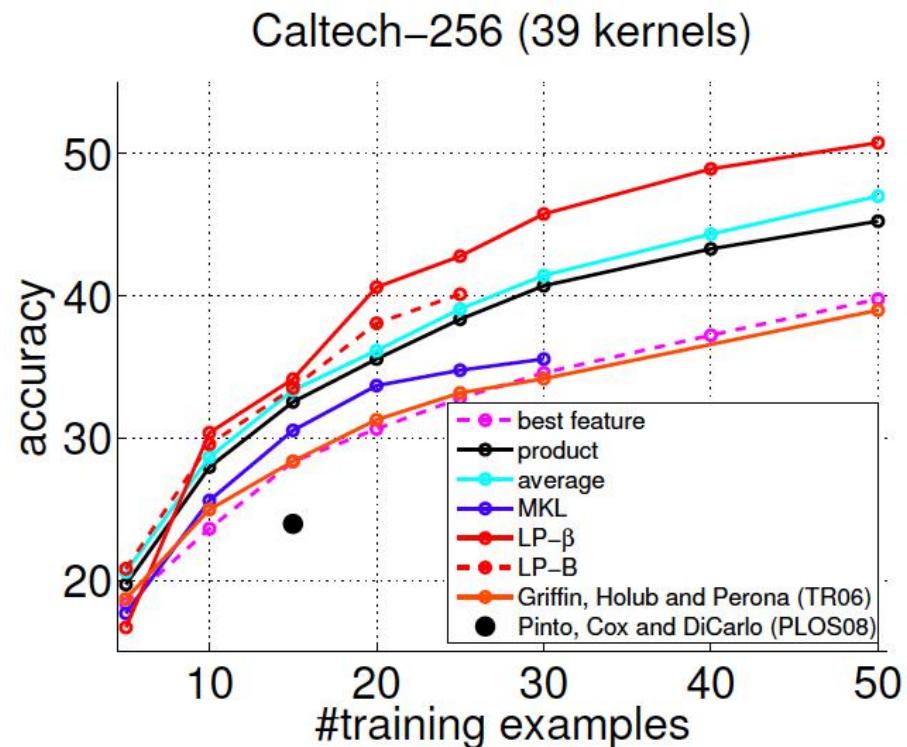
传统的模式识别方法：



- 良好的特征表达，对最终算法的准确性起了非常关键的作用；
- 识别系统主要的计算和测试工作耗时主要集中在特征提取部分；
- 特征的样式目前一般都是人工设计的，靠人工提取特征。

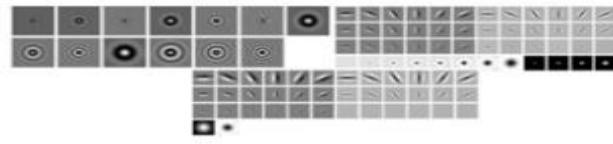
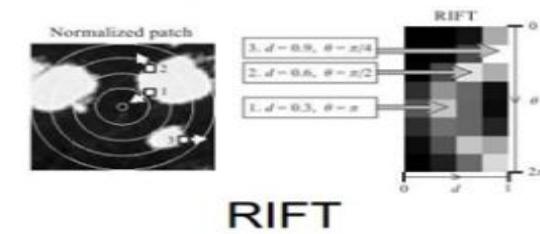
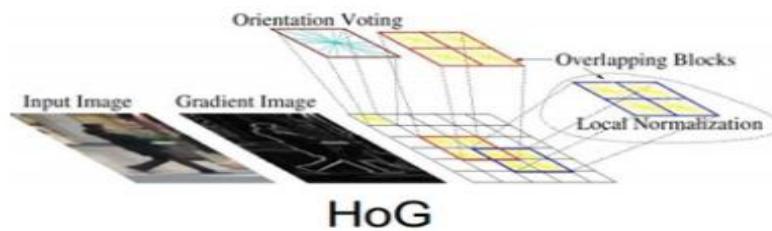
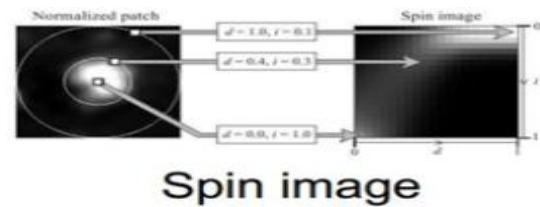
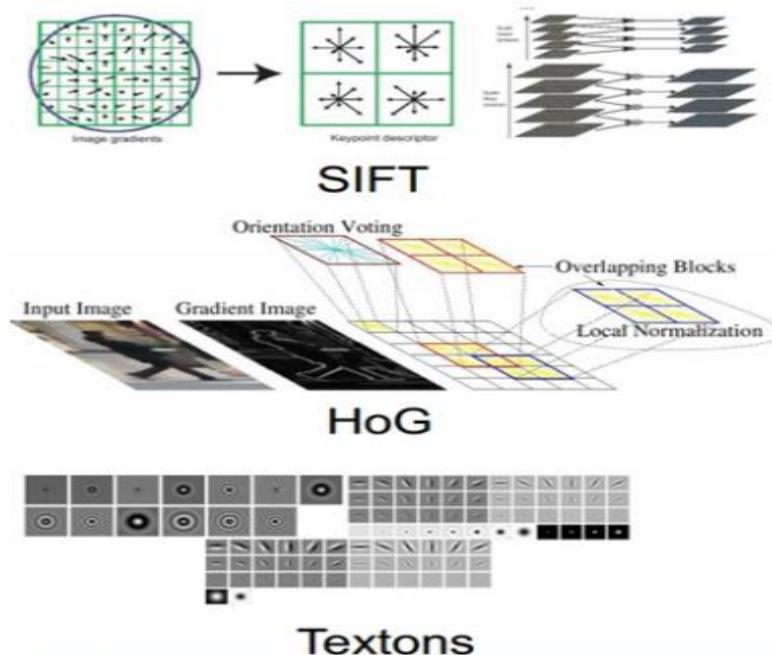
动机——为什么要自动学习特征

- 实验: LP- β Multiple Kernel Learning
 - Gehler and Nowozin, On Feature Combination for Multiclass Object Classification, ICCV' 09
 - 采用39个不同的特征
 - PHOG, SIFT, V1S+, Region Cov. Etc.
 - 在普通特征上MKL表现有限
- 结论: 特征更重要

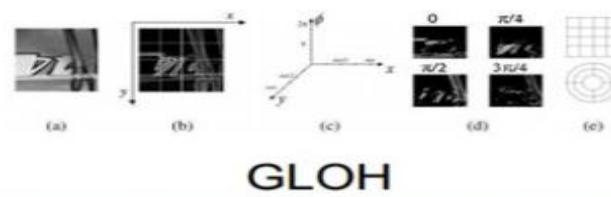


动机——为什么要自动学习特征

- 机器学习中，获得好的特征是识别成功的关键
- 目前存在大量人工设计的特征，不同研究对象特征不同，特征具有多样性，如：SIFT，HOG，LBP等
- 手工选取特征费时费力，需要启发式专业知识，很大程度上靠经验和运气
- 是否能自动地学习特征？



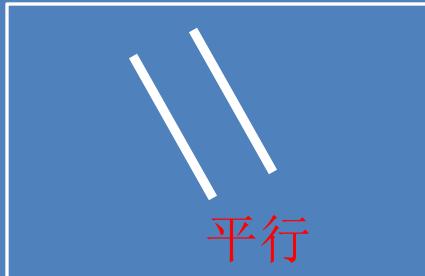
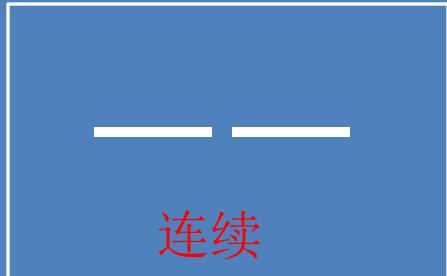
Textons



动 机——为什么要自动学习特征

- 中层特征

- ✓ 中层信号:



“Tokens” from Vision by D.Marr:

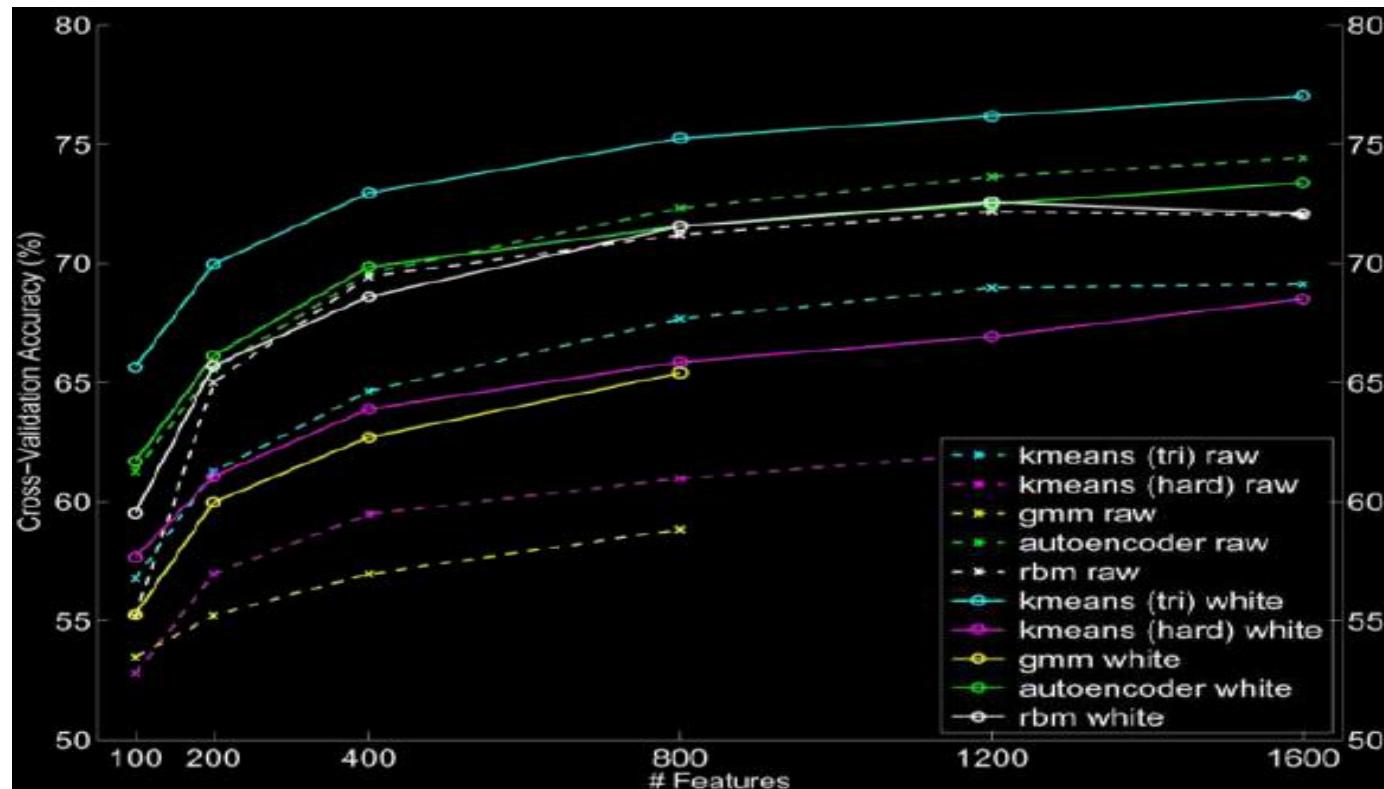


- ✓ 物体部件:



- 他们对于人工而言是十分困难的，那么如何学习呢？

动机——为什么要自动学习特征

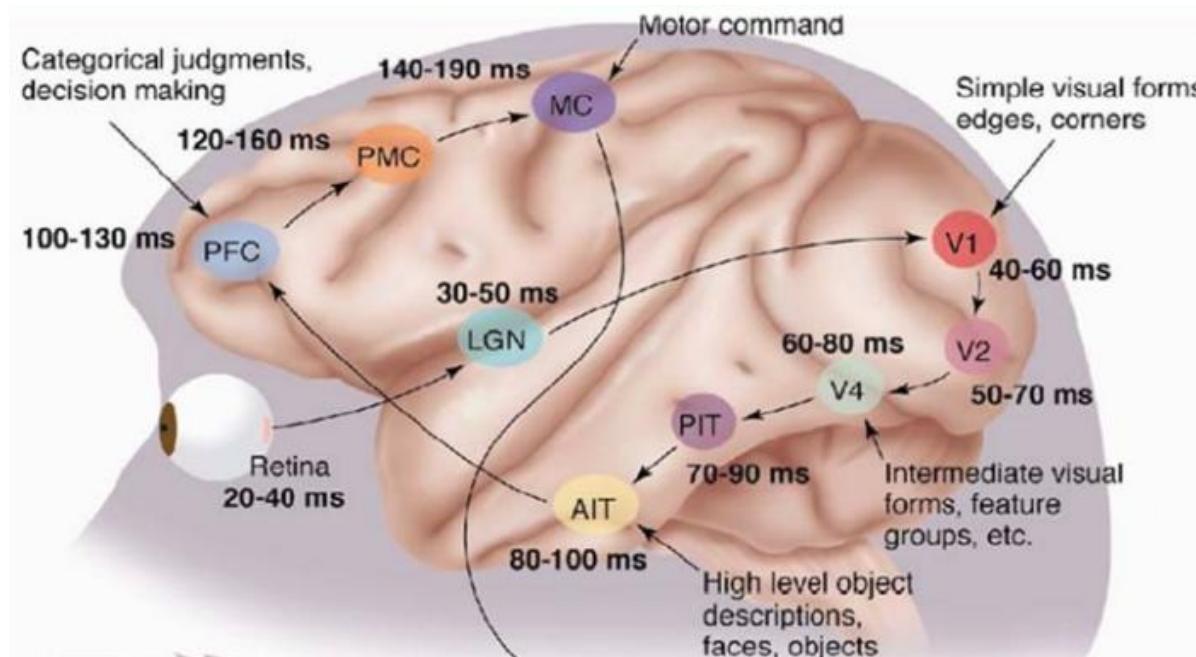


- 一般而言，特征越多，给出信息就越多，识别准确性会得到提升；
- 但特征多，计算复杂度增加，探索的空间大，可以用来训练的数据在每个特征上就会稀疏。
- **结论：不一定特征越多越好！需要有多少个特征，需要学习确定。**

动机——为什么采用层次网络结构

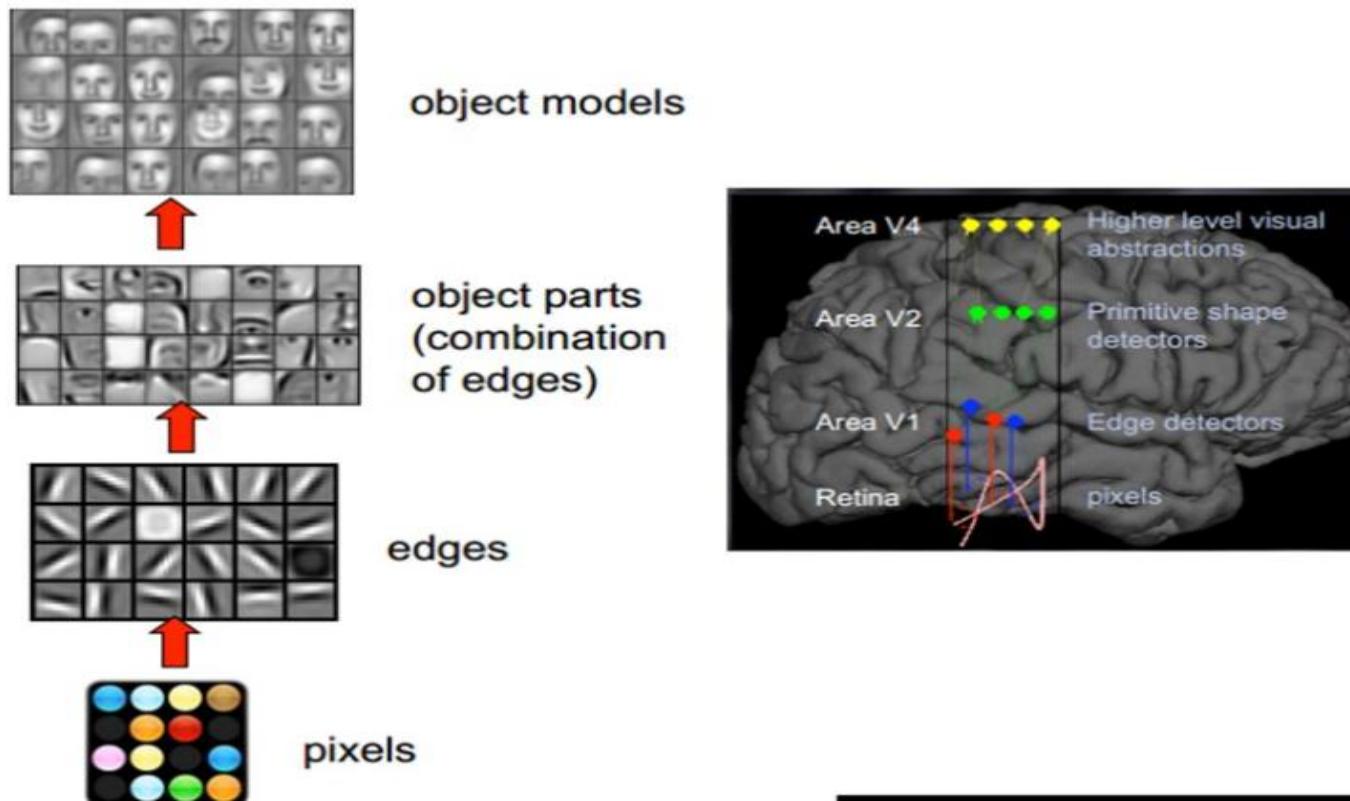
- 人脑视觉机理

- ✓ 1981年的诺贝尔医学奖获得者 David Hubel 和 Torsten Wiesel 发现了视觉系统的信息处理机制
- ✓ 发现了一种被称为“方向选择性细胞”的神经元细胞，当瞳孔发现了眼前的物体的边缘，而且这个边缘指向某个方向时，这种神经元细胞就会活跃



动 机——为什么采用层次网络结构

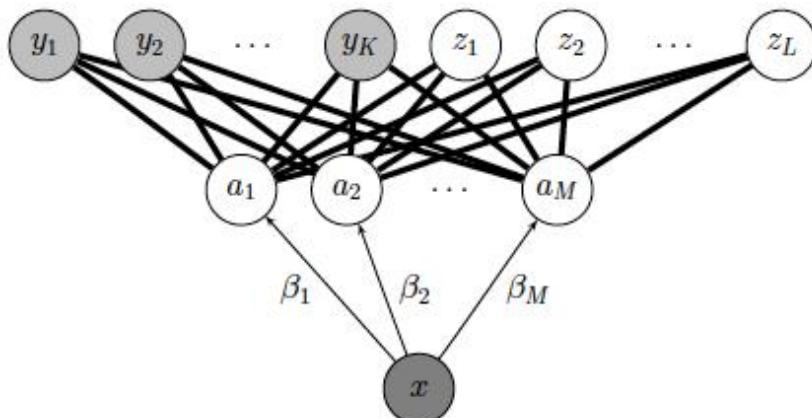
- 人脑视觉机理
 - ✓ 人的视觉系统的信息处理是分级的
 - ✓ 高层的特征是低层特征的组合，从低层到高层的特征表示越来越抽象，越来越能表现语义或者意图
 - ✓ 抽象层面越高，存在的可能猜测就越少，就越利于分类



动 机——为什么采用层次网络结构

- 视觉的层次性
- ✓ 属性学习，类别作为属性的一种组合映射

Lampert et al. CVPR' 09



otter

black:	yes
white:	no
brown:	yes
stripes:	no
water:	yes
eats fish:	yes



polar bear

black:	no
white:	yes
brown:	no
stripes:	no
water:	yes
eats fish:	yes



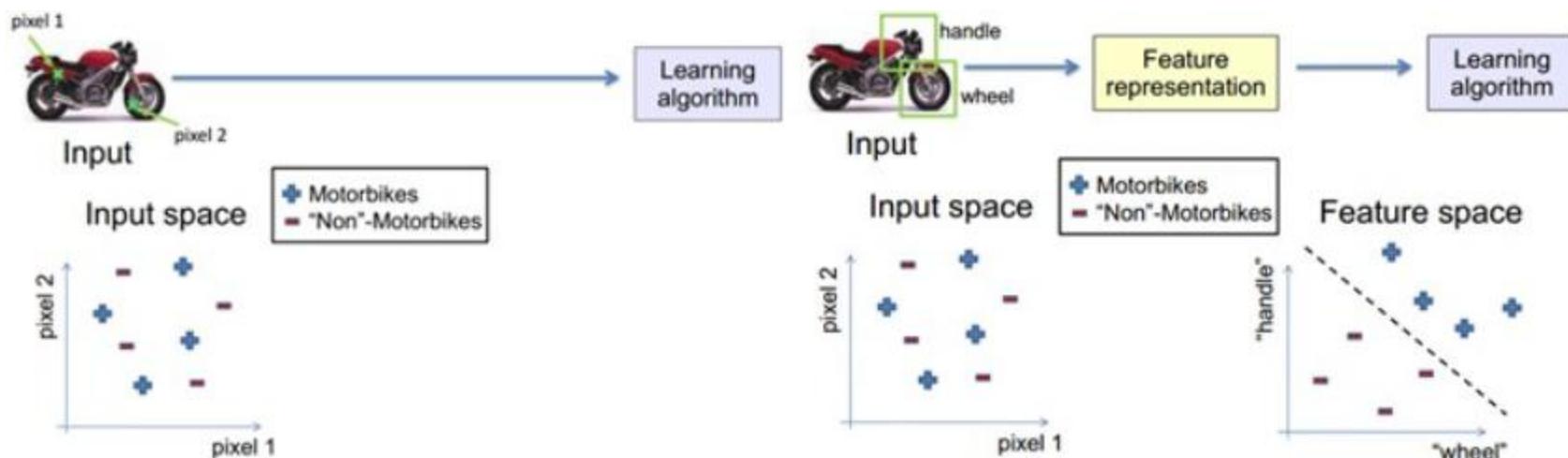
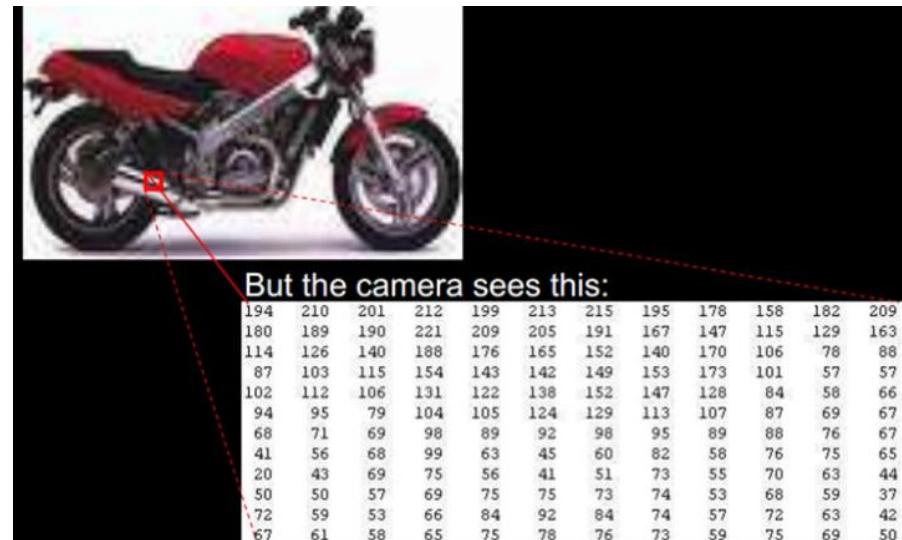
zebra

black:	yes
white:	yes
brown:	no
stripes:	yes
water:	no
eats fish:	no



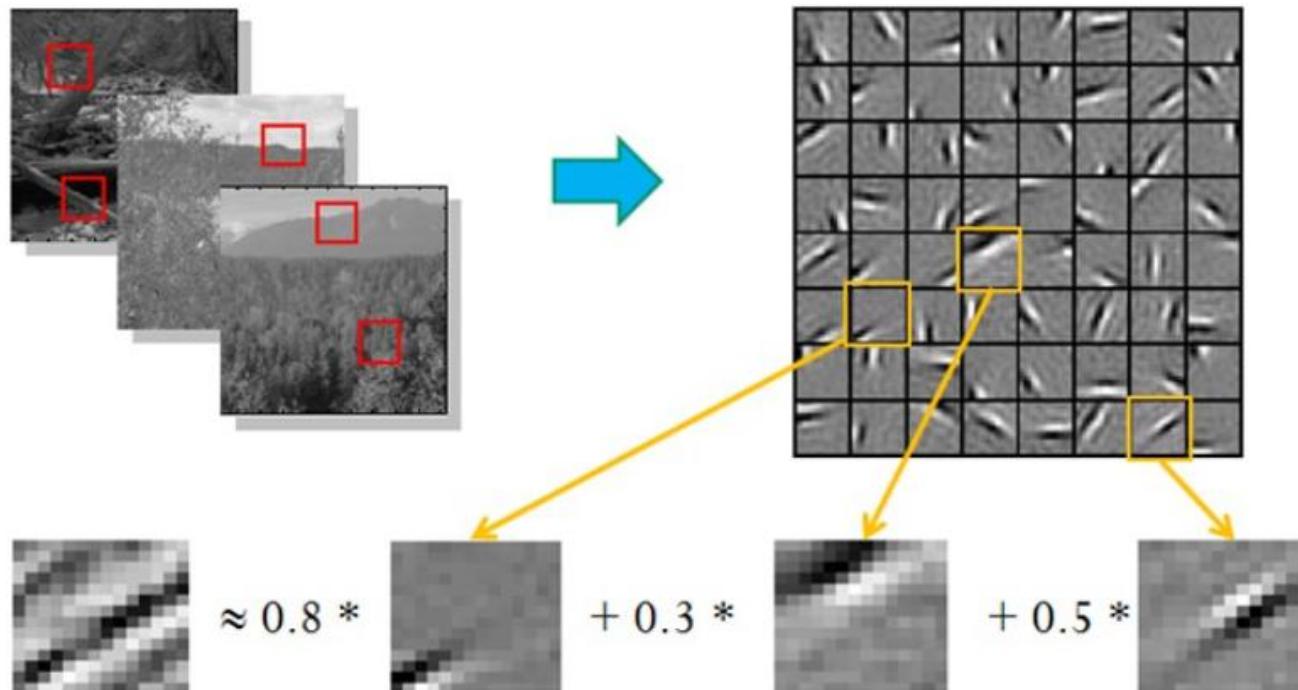
动机——为什么采用层次网络结构

- 特征表示的粒度
- ✓ 具有结构性（或者语义）的高层特征对于分类更有意义



动 机——为什么采用层次网络结构

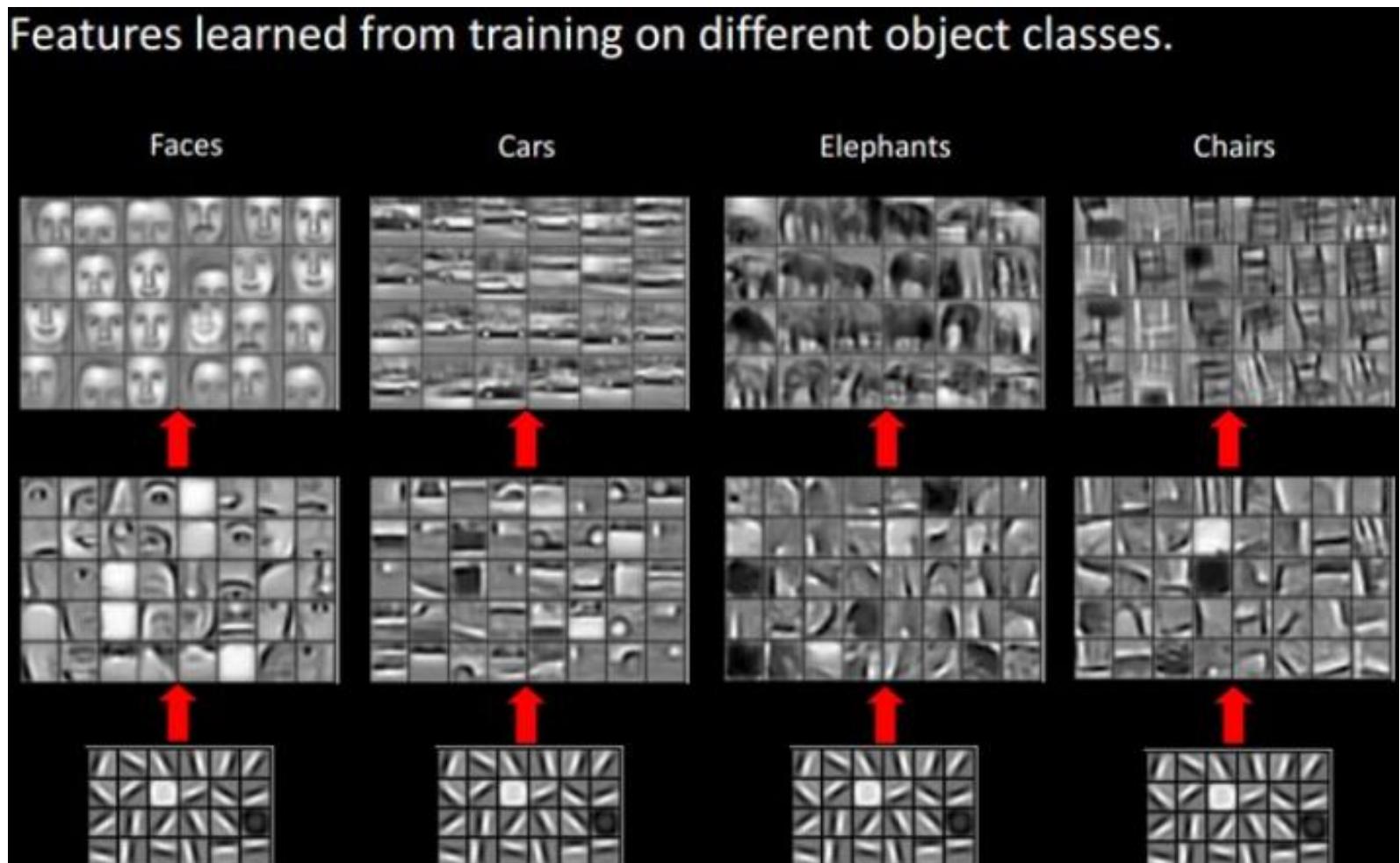
- 初级（浅层）特征表示



$[a_1, \dots, a_{64}] = [0, 0, \dots, 0, \mathbf{0.8}, 0, \dots, 0, \mathbf{0.3}, 0, \dots, 0, \mathbf{0.5}, 0]$
(feature representation)

动 机——为什么采用层次网络结构

- 结构性特征表示



动 机——为什么采用层次网络结构

- 浅层学习的局限

- ✓ 人工神经网络（BP算法）

- 虽被称作多层感知机，但实际是一种只含有一层隐层节点的浅层模型

- ✓ SVM、Boosting、最大熵方法（如LR, Logistic Regression）

- 带有一层隐层节点（如SVM、Boosting），或没有隐层节点（如LR）的浅层模型

局限性：有限样本和计算单元情况下对复杂函数的表示能力有限，针对复杂分类问题其泛化能力受限。

深度学习

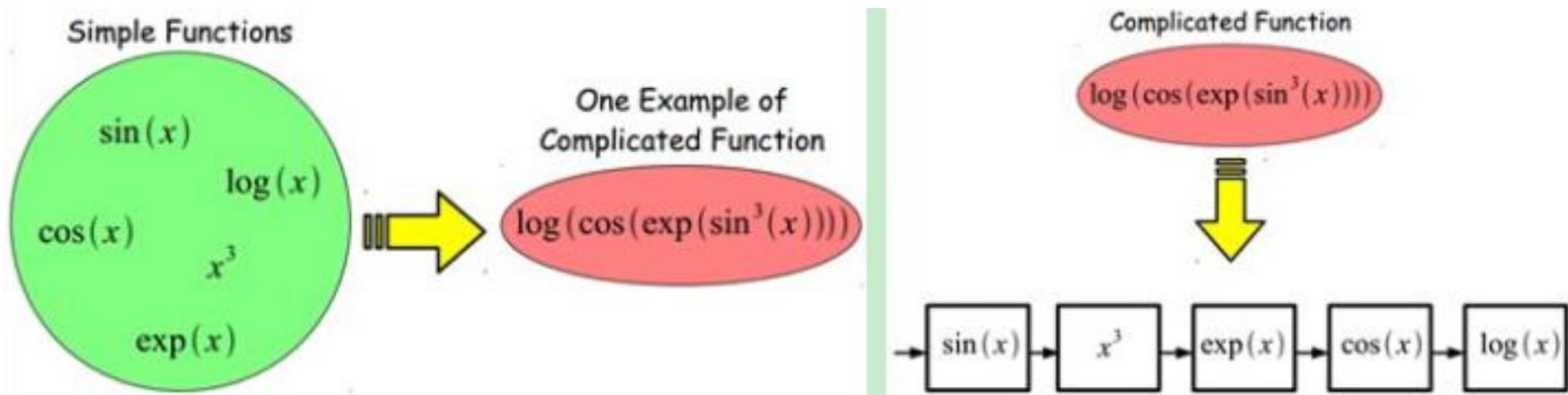
- 2006年，加拿大多伦多大学教授、机器学习领域的泰斗Geoffrey Hinton在《科学》上发表论文提出深度学习主要观点：
 - 1) 多隐层的人工神经网络具有优异的特征学习能力，学习得到的特征对数据有更本质的刻画，从而有利于可视化或分类；
 - 2) 深度神经网络在训练上的难度，可以通过“逐层初始化”（layer-wise pre-training）来有效克服，逐层初始化可通过无监督学习实现的。

深度学习

- **本质：**通过构建多隐层的模型和海量训练数据（可为无标签数据），来学习更有用的特征，从而最终提升分类或预测的准确性。“深度模型”是手段，“特征学习”是目的。
- **与浅层学习区别：**
 - 1) 强调了模型结构的深度，通常有5-10多层的隐层节点；
 - 2) 明确突出了特征学习的重要性，通过逐层特征变换，将样本在原空间的特征表示变换到一个新特征空间，从而使分类或预测更加容易。与人工规则构造特征的方法相比，利用大数据来学习特征，更能够刻画数据的丰富内在信息。

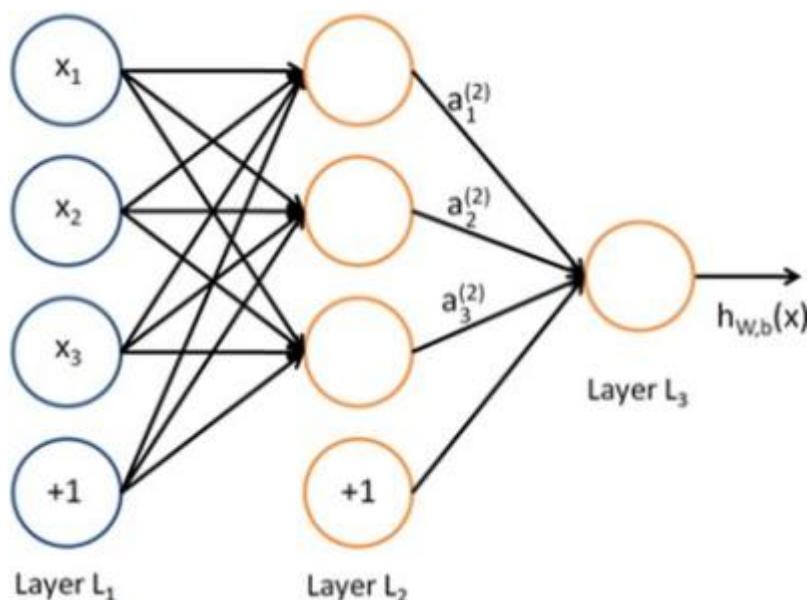
深度学习

- 好处：可通过学习一种深层非线性网络结构，实现复杂函数逼近，表征输入数据分布式表示。

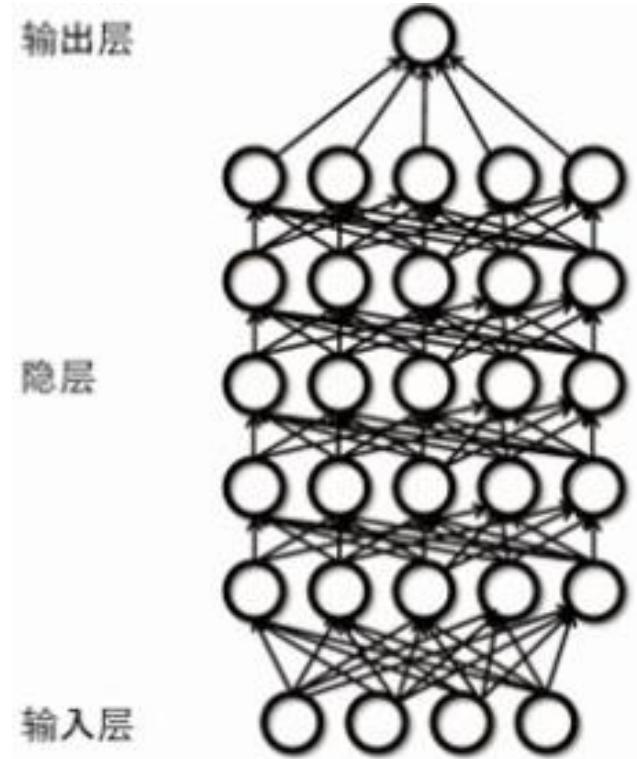


深度学习 vs. 神经网络

神经网络 :



深度学习:



含多个隐层的深度学习模型

深度学习 vs. 神经网络

相同点：二者均采用分层结构，系统包括输入层、隐层（多层）、输出层组成的多层网络，只有相邻层节点之间有连接，同一层以及跨层节点之间相互无连接，每一层可以看作是一个logistic 回归模型。

不同点：

神经网络：采用BP算法调整参数，即采用迭代算法来训练整个网络。随机设定初值，计算当前网络的输出，然后根据当前输出和样本真实标签之间的差去改变前面各层的参数，直到收敛；

深度学习：采用逐层训练机制。采用该机制的原因在于如果采用BP机制，对于一个deep network（7层以上），残差传播到最前面的层将变得很小，出现所谓的gradient diffusion（梯度扩散）。

深度学习 vs. 神经网络

- 神经网络的局限性：
 - 1) 比较容易过拟合，参数比较难调整，而且需要不少技巧；
 - 2) 训练速度比较慢，在层次比较少（小于等于3）的情况下效果并不比其它方法更优；

Deep learning

Yoshua Bengio:

Science is NOT a battle, it is a collaboration. We all build on each other's ideas. Science is an act of love, not war. Love for the beauty in the world that surrounds us and love to share and build something together. That makes science a highly satisfying activity, emotionally speaking!

科学不是战争而是合作，任何学科的发展从来都不是一条路走到黑，而是同行之间互相学习、互相借鉴、博采众长、相得益彰，站在巨人的肩膀上不断前行。机器学习的研究也是一样，**你死我活那是邪教，开放包容才是正道。**

Part II

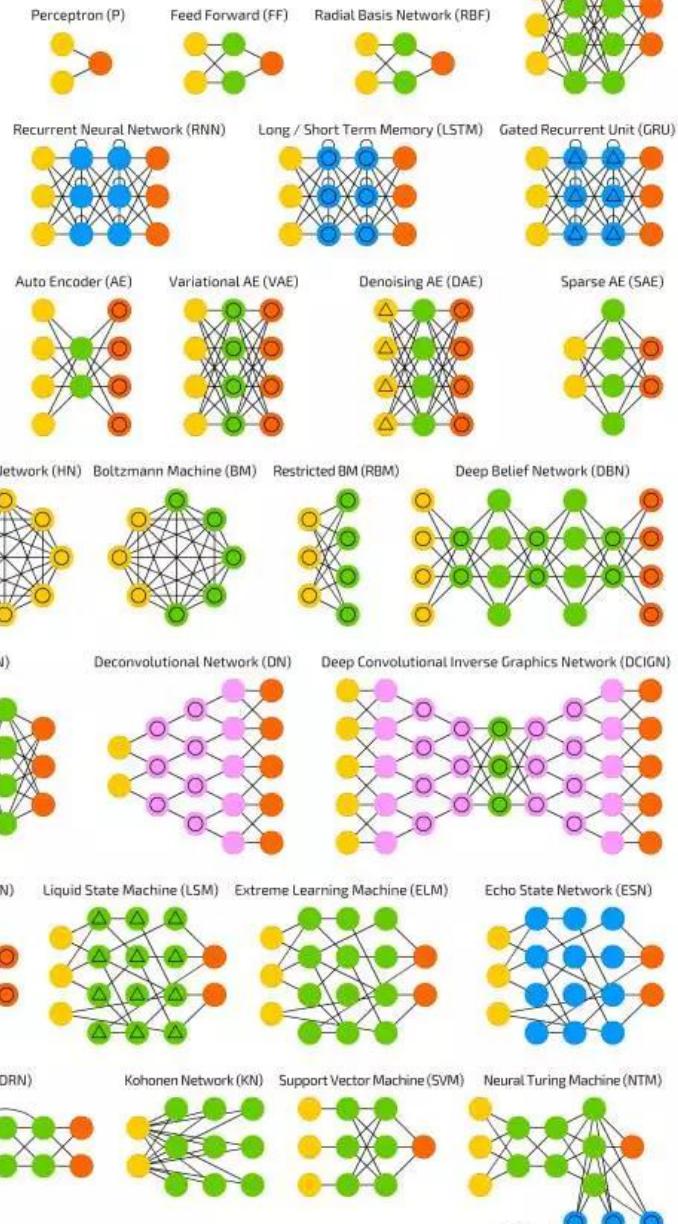
深度学习的基本方法
(以及神经网络算法)

Neural network

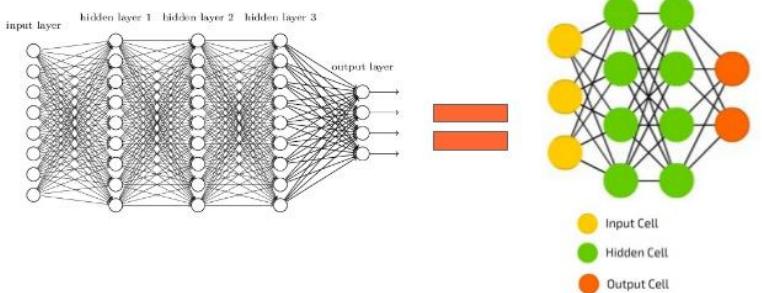
A mostly complete chart of Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

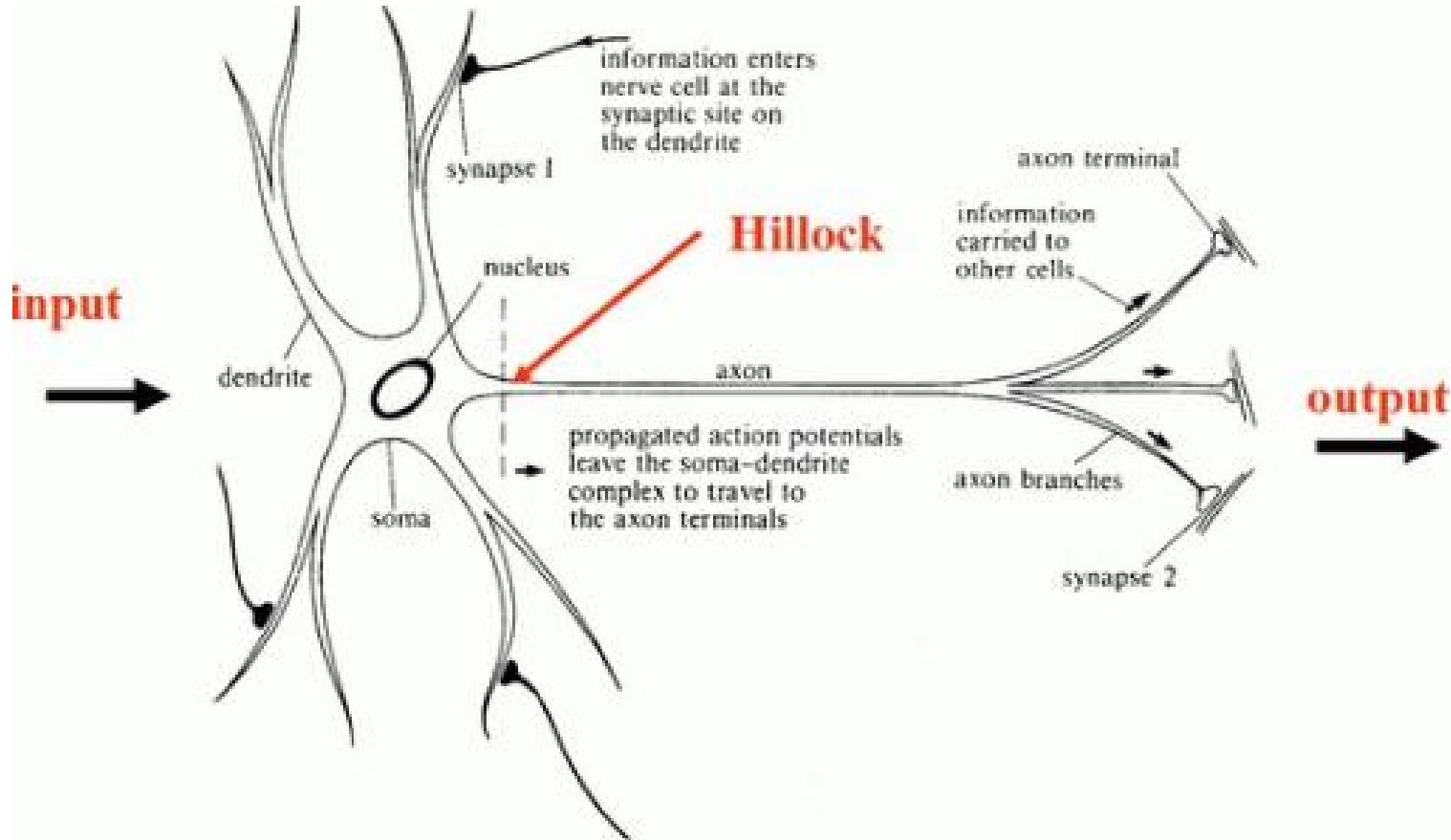
- Backfed Input Cell
- Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- Different Memory Cell
- Kernel
- Convolution or Pool



Deep Neural Network (DNN)



Neural network

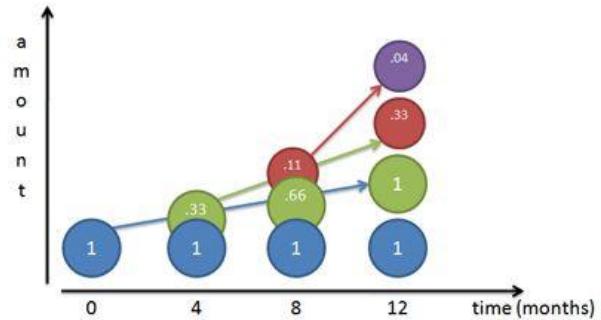
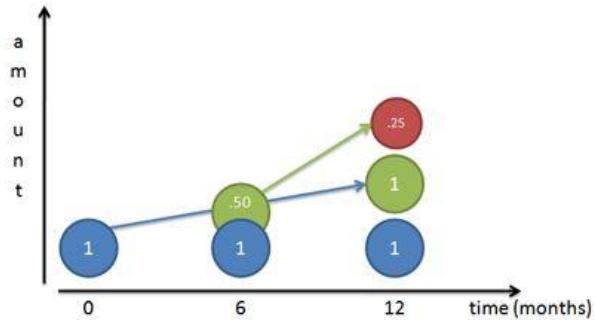
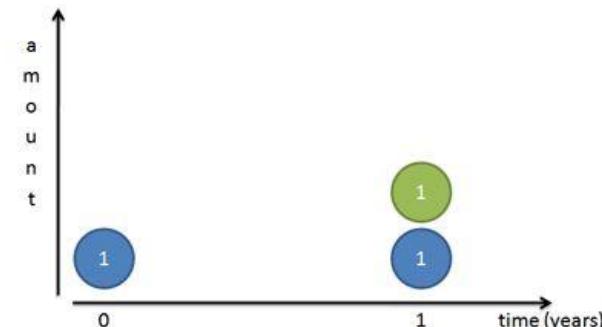


Neural network

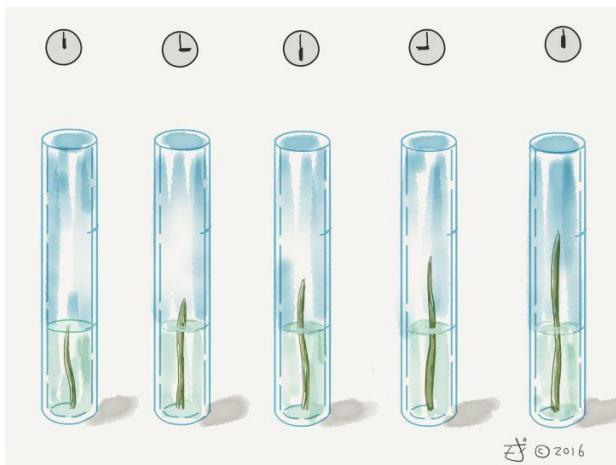
- 自然对数e
- 逻辑回归
- 神经网络
- 反向传播算法(BP算法)
- 示例

Neural network

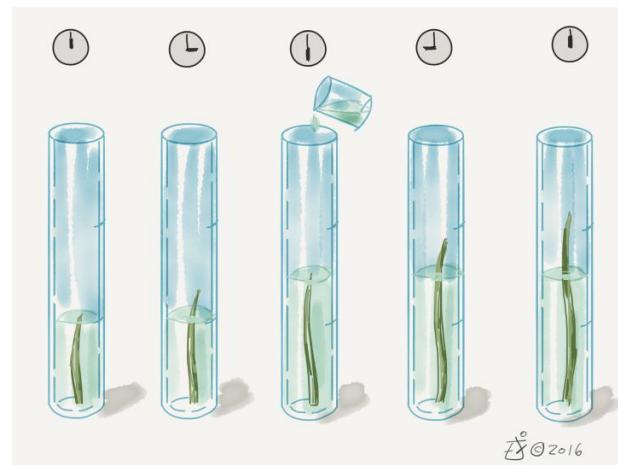
自然对数e



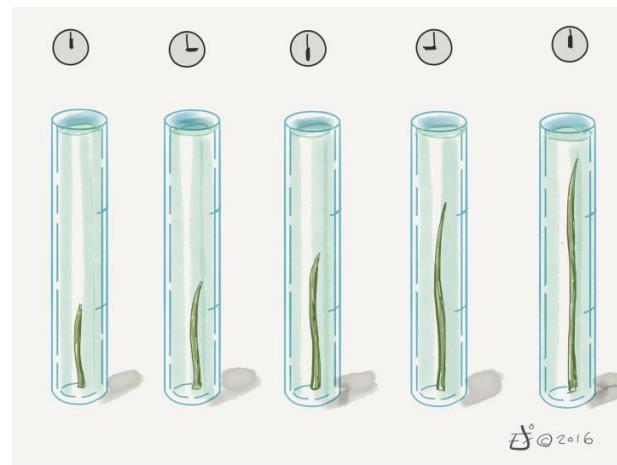
1元存1年，在年利率100%下，无论怎么利滚利，其余额总有一个上限，这个天花板就是e



有一种草齐头泡水半天变2倍



中间加一次水能长到2.25倍



加足水最多能长到2.718...倍

Neural network

自然对数e



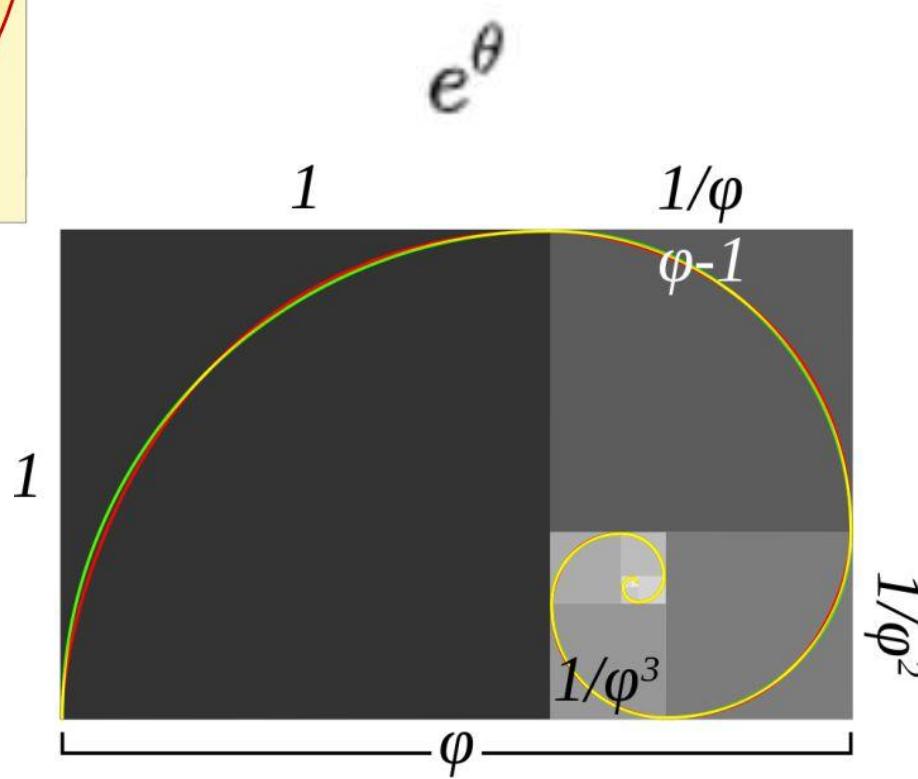
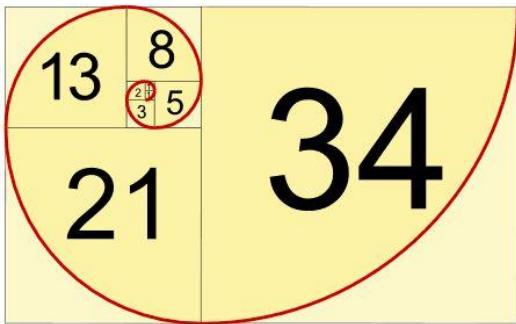
$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \frac{1}{6!} + \frac{1}{7!} + \dots = 1 + 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{24} + \frac{1}{120} + \dots \approx 2.71828$$

$$e = (1 + 1/x)^x \approx 2.71828$$



Neural network

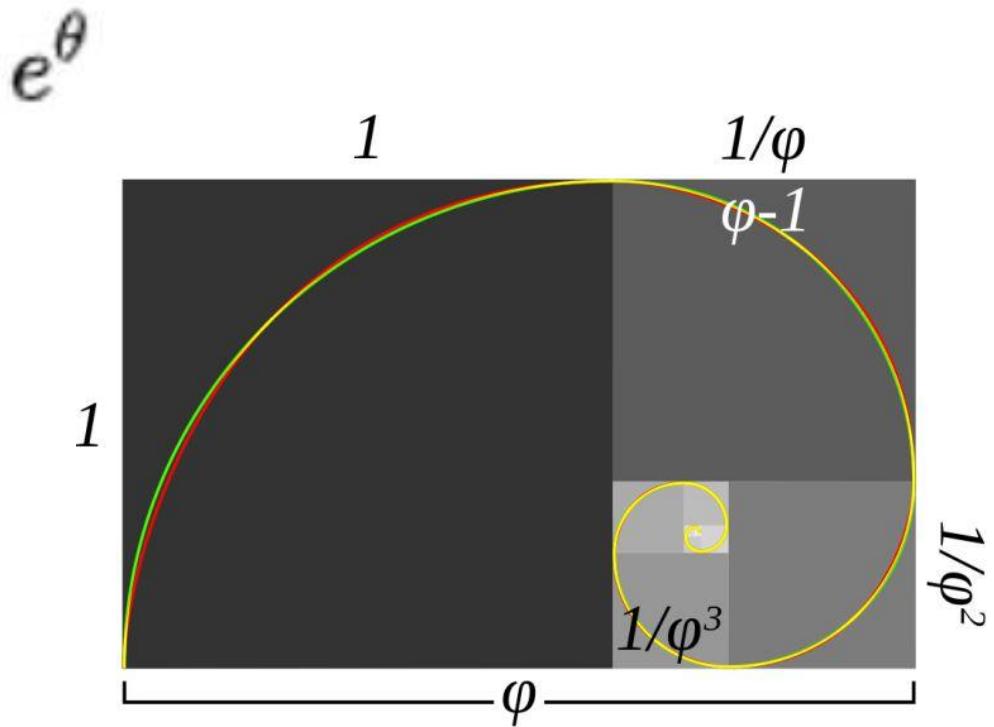
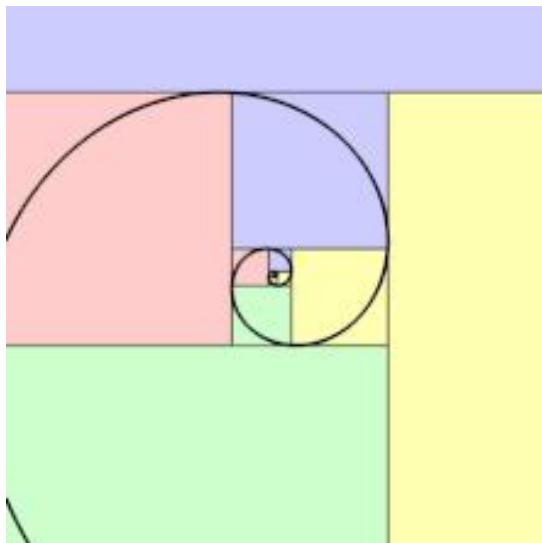
自然对数e vs. 斐波拉切数列



斐波那契螺线仅仅是对一种叫黄金螺线（Golden spiral）的近似，黄金螺线是一种内涵黄金分割比例的对数螺线。
红色的才是黄金曲线，绿色的是“假黄金螺线”（斐波那契螺线），近似却不重合。

Neural network

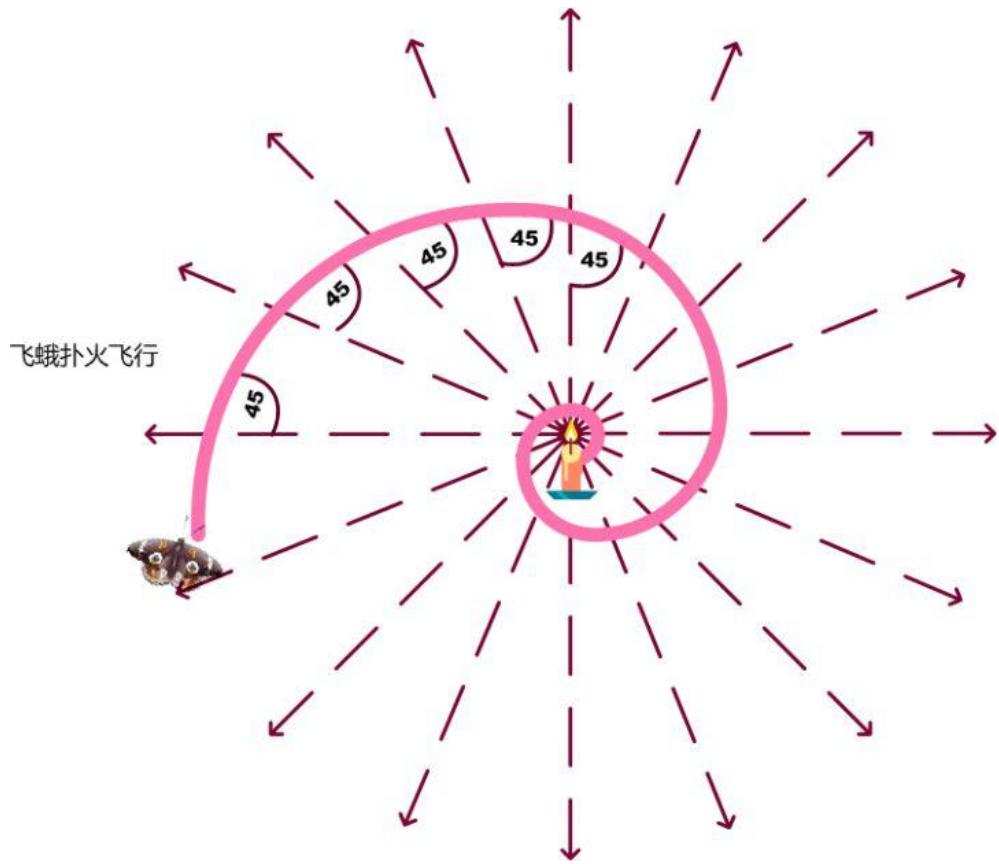
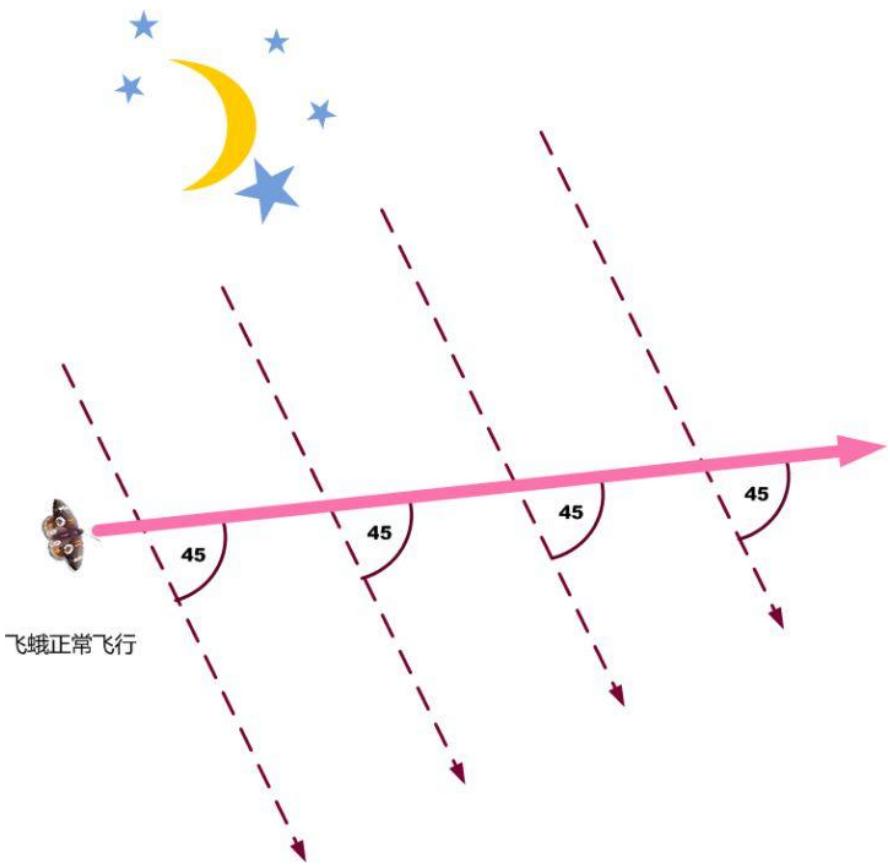
自然对数e & 黄金分割



$$\varphi = 1.6180339887498948482\dots$$

Neural network

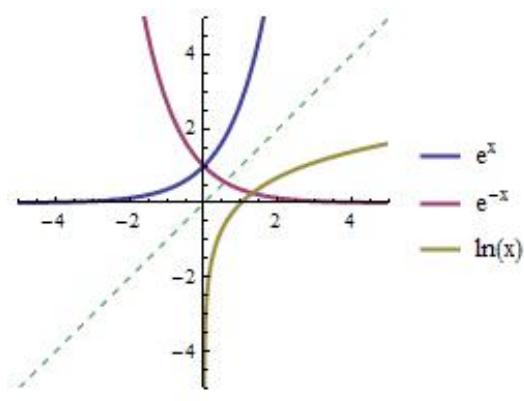
自然对数e



Neural network

自然对数e

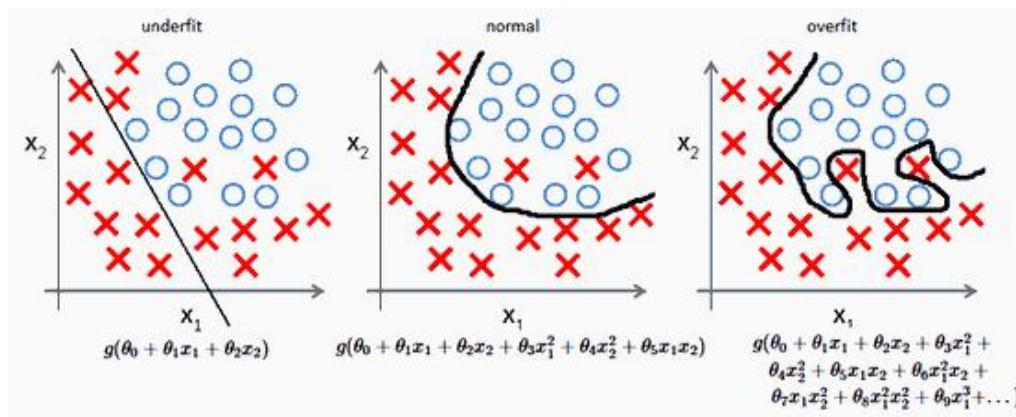
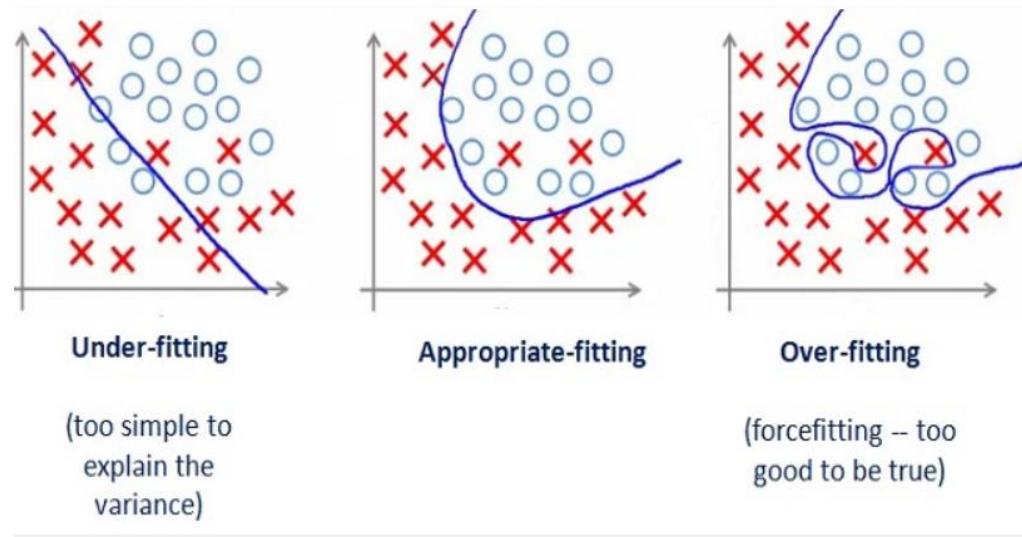
$$\begin{array}{c} \text{幂(power)} \rightarrow y = e^x \xleftarrow[\text{底数(base number)}]{\text{指数(exponent)}} \\[10pt] \text{对数(logarithm)} \rightarrow x = \log_e y = \ln(y) \xrightarrow{\text{真数(正数)}} \text{自然对数(natural logarithm)} \end{array}$$



e^x 和 e^{-x} 的图形是对称的； $\ln(x)$ 是 e^x 的逆函数，它们呈45度对称。

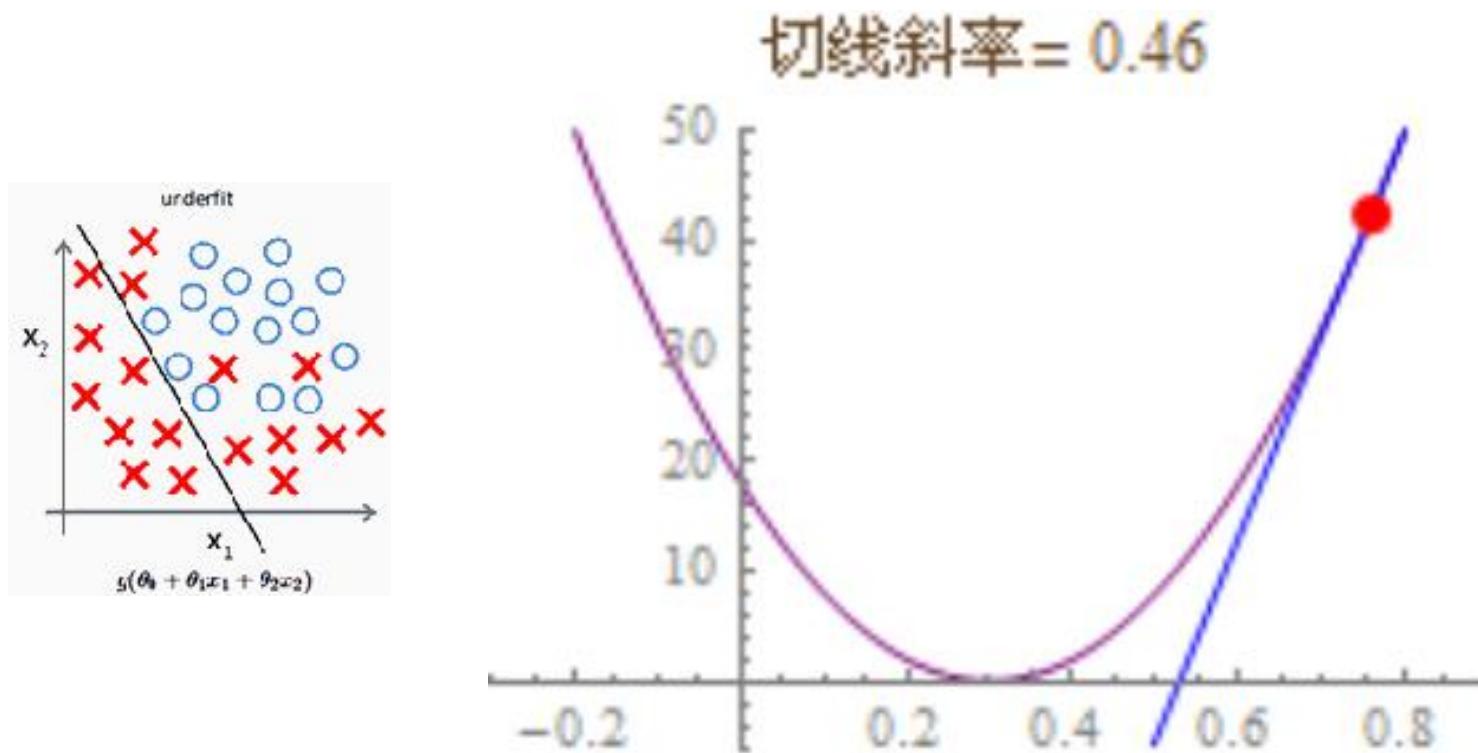
Neural network

逻辑回归



Neural network

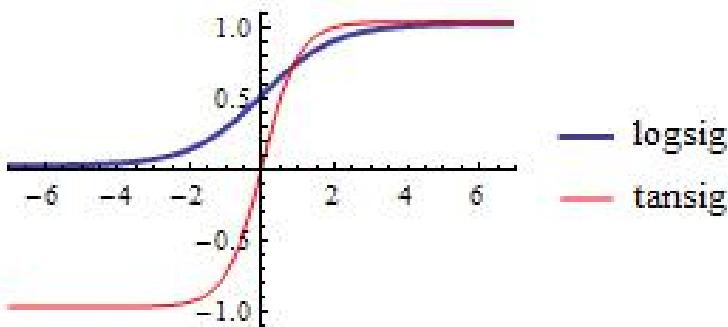
逻辑回归



切线每次旋转的幅度叫做学习率(Learning Rate)，加大学习率会加快拟合速度，但是如果调得太高会导致切线旋转过度而无法收敛。[学习率其实是个预先设置好的参数，不会每次变化，不过可以影响每次变化的幅度。]

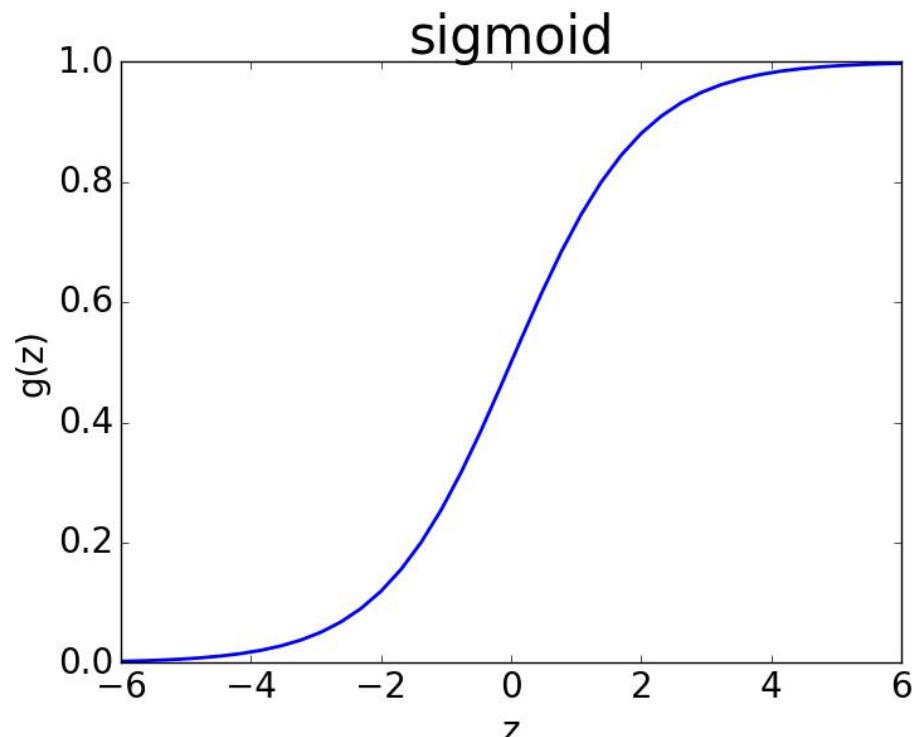
Neural network

逻辑回归(Logistic Regression, 逻辑斯谛函数)



$$y = \text{logsig}(x) = \frac{1}{1 + e^{-x}}$$

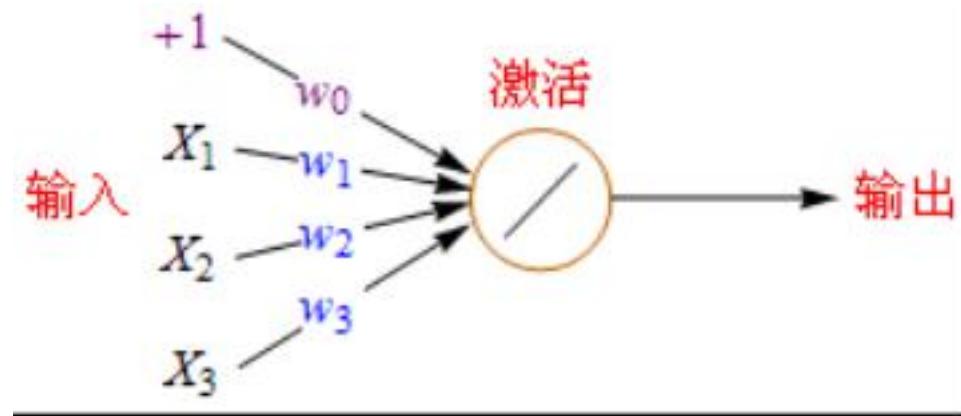
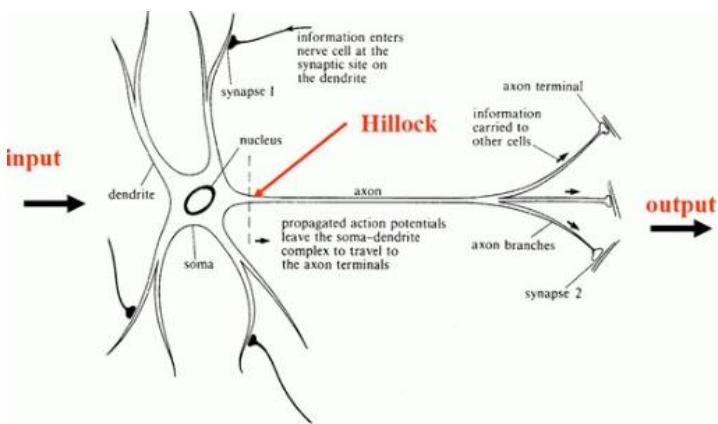
$$y = \text{tansig}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



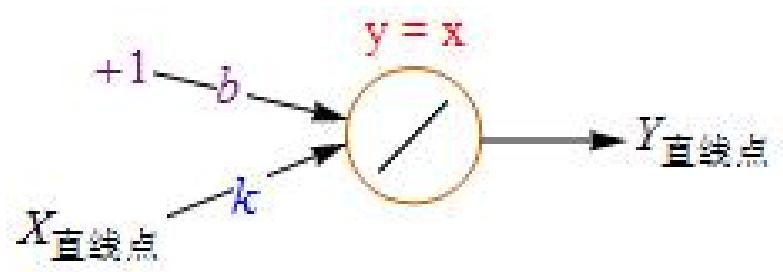
y的阈值处于 $(-\infty, +\infty)$ ，此时不能很好的给出属于某一类的概率，因为概率的范围是 $[0,1]$ ，我们需要一个更好的映射函数，能够将分类的结果很好的映射成为 $[0,1]$ 之间的概率，并且这个函数能够具有很好的可微分性。

Neural network

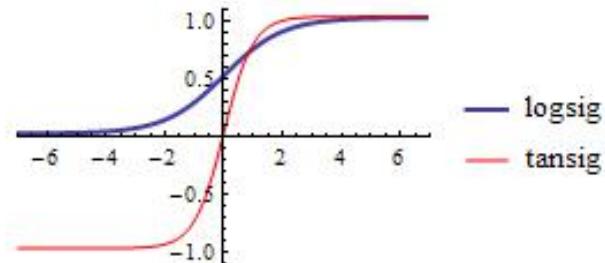
逻辑回归



Purelin激活函数



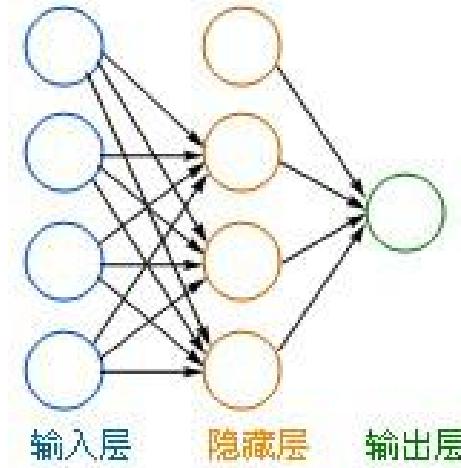
Sigmoid激活函数



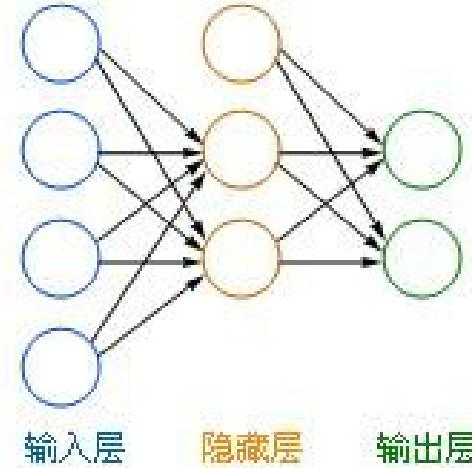
Neural network

神经网络

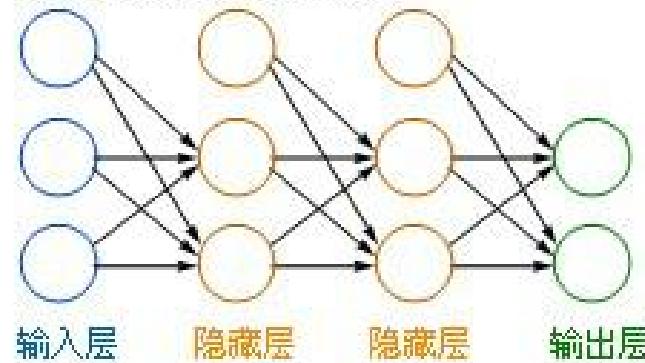
典型的三层网络



三层网络，输出层有两个节点

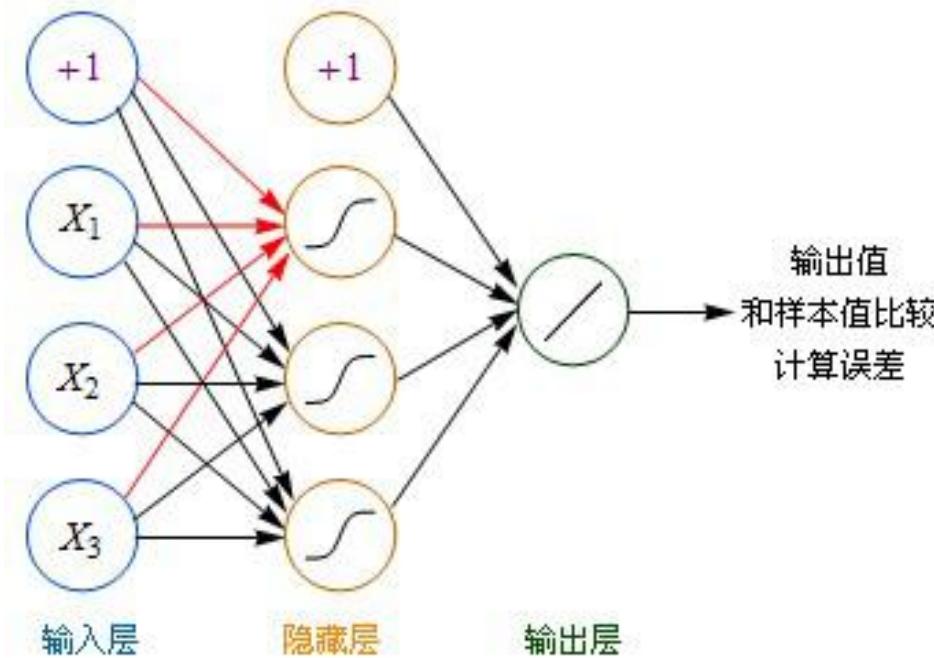


四层网络，包含两个隐藏



Neural network

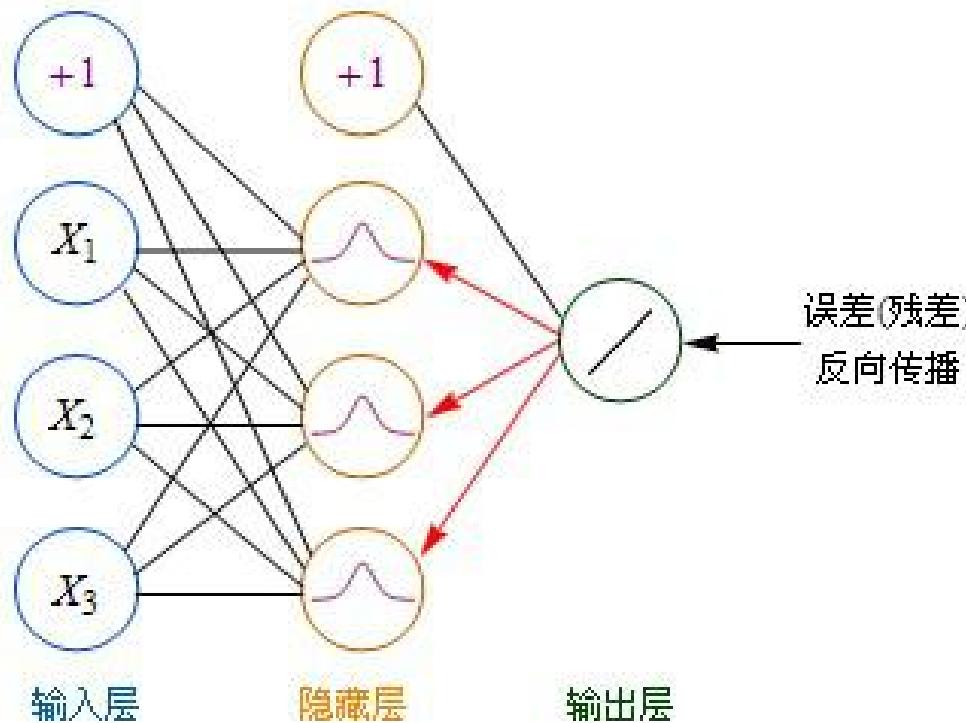
神经网络



- 隐藏层用都是用Sigmoid作激活函数，而输出层用的是Purelin。这是因为Purelin可以保持之前任意范围的数值缩放，便于和样本值作比较，而Sigmoid的数值范围只能在0~1之间。
- 起初输入层的数值通过网络计算分别传播到隐藏层，再以相同的方式传播到输出层，最终的输出值和样本值作比较，计算出误差，这个过程叫前向传播(Forward Propagation)。

Neural network

反向传播算法(BP算法)

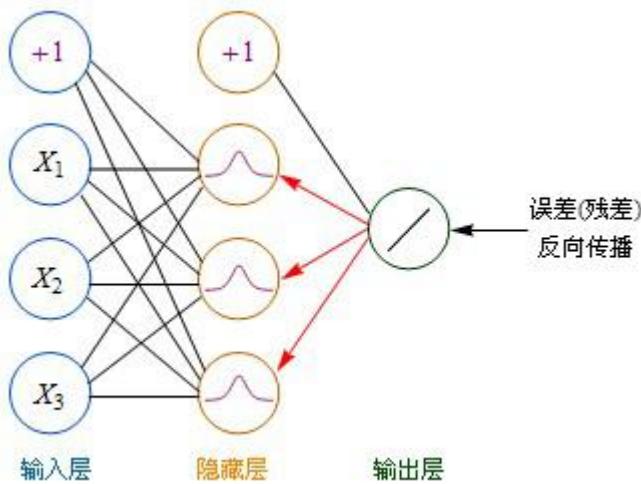


利用前向传播最后输出的结果来计算误差的偏导数，再用这个偏导数和前面的隐藏层进行加权求和，如此一层一层的向后传下去，直到输入层(不计算输入层)，最后利用每个节点求出的偏导数来更新权重

Neural network

反向传播算法(BP算法)

前馈神经网络(FeedForward Neural Network), 也叫BP神经网络(Back Propagation Neural Network)。



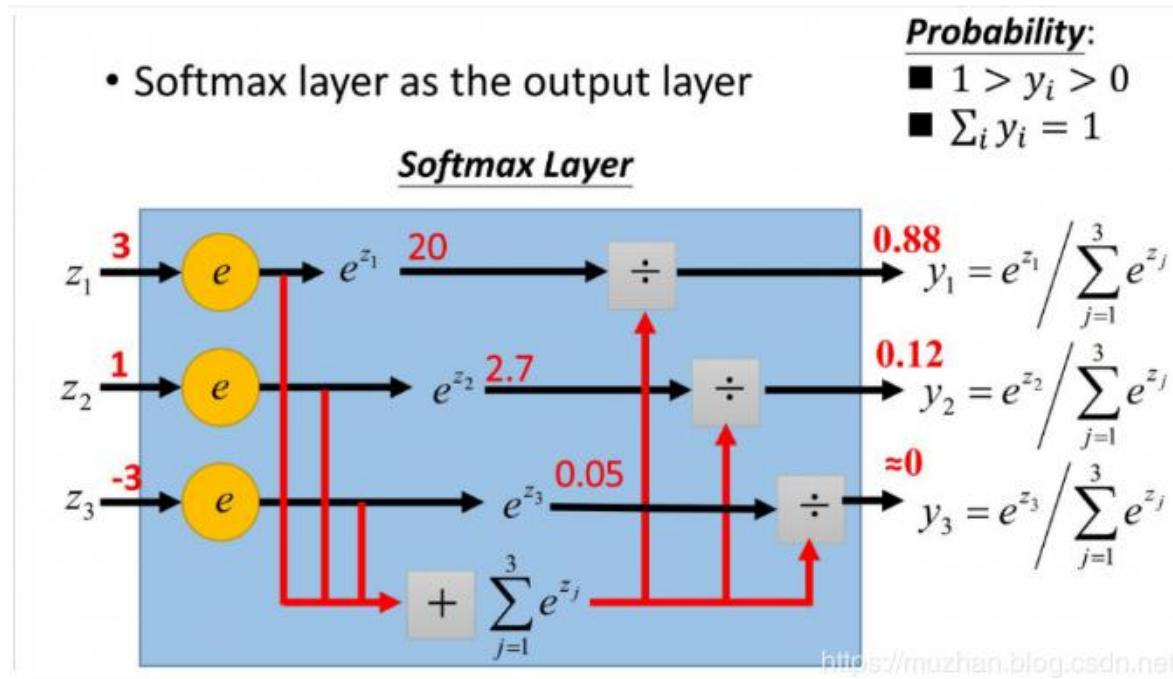
如果输出层用Purelin作激活函数, Purelin的导数是1, 输出层→隐藏层: 残差 = -(输出值-样本值)

如果用Sigmoid(logsig)作激活函数, 那么: Sigmoid导数 = Sigmoid*(1-Sigmoid)
输出层→隐藏层: 残差 = -(Sigmoid输出值-样本值) * Sigmoid*(1-Sigmoid) = -(输出值-样本值)
输出值(1-输出值)
隐藏层→隐藏层: 残差 = (右层每个节点的残差加权求和)* 当前节点的Sigmoid*(1-当前节点的Sigmoid)

Neural network

softmax分类器

$$\text{softmax}(x_0) = \frac{e^{x_0}}{e^{x_0} + e^{x_1} + e^{x_2}}$$



softmax对神经元的输出信号进行加工，输出为分类的概率值。

Neural network

示例

图例：

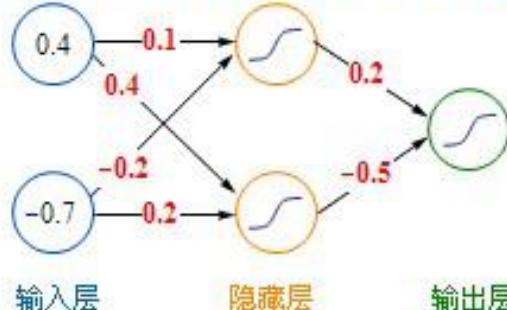
- logsig激活函数
- 残差(误差偏导数)
- 加权求和

训练集的数据，首先对第一行进行处理

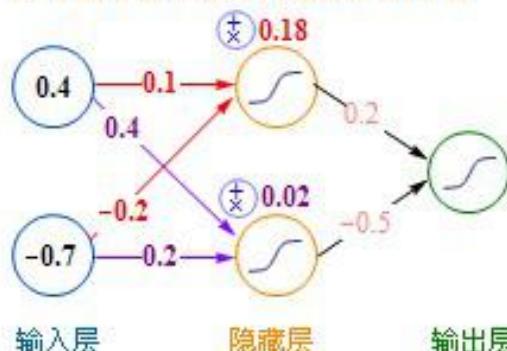
X1	X2	样本值
0.4	-0.7	0.1
0.3	-0.5	0.05
0.6	0.1	0.3
0.2	0.4	0.25
0.1	-0.2	0.12

为了适应输出层的logsig变换，X1,X2的数值范围规定在至0~1之间

1. 每个节点之间的初始权重一般是随机生成的



2. 对输入层节点进行加权求和计算



$$0.4 \times 0.1 + (-0.7) \times (-0.2) = 0.18$$

$$0.4 \times 0.4 + (-0.7) \times (0.2) = 0.02$$

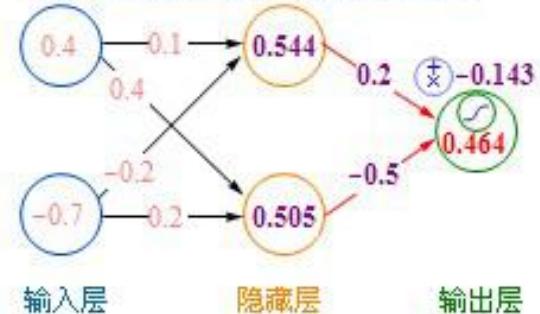
3. 执行 Sigmoid 激活



$$\text{logsig}(0.18) = \frac{1}{1+e^{-0.18}} = 0.544$$

$$\text{logsig}(0.02) = \frac{1}{1+e^{-0.02}} = 0.505$$

4. 用相同的方法计算出输出层的值



$$0.544 \times 0.2 + 0.505 \times (-0.5) = -0.143$$

$$\text{logsig}(-0.143) = \frac{1}{1+e^{0.143}} = 0.464$$

Neural network

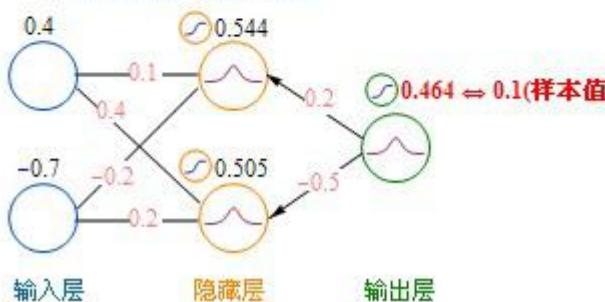
示例

5. 计算误差，误差接近0时收敛

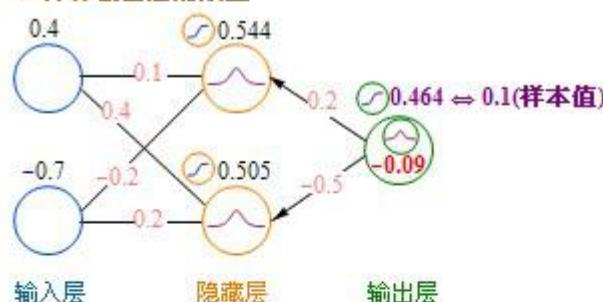


$$\text{误差} = (0.464 - 0.1)^2 = 0.132 \quad (\text{误差} < 0.0001, \text{可以收敛})$$

6. 开始准备误差的反向传播



7. 计算输出层的残差



8. 输出层节点的残差加权求和

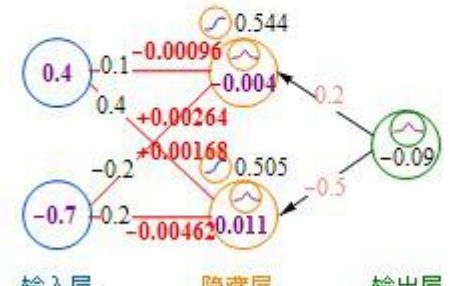


9. 继续求隐藏层的残差



$$\text{残差2} = 0.045 \times 0.505 \times (1 - 0.505) = 0.011$$

10. 开始准备更新第一层权重，设学习率为0.6



$$0.4 \times 0.011 \times 0.6 = 0.0044 \times 0.6 = 0.00264$$
$$-0.7 \times (-0.004) \times 0.6 = 0.0028 \times 0.6 = 0.00168$$
$$-0.7 \times 0.011 \times 0.6 = -0.0077 \times 0.6 = -0.00462$$

Neural network

示例

11. 更新第一层权重



$$\begin{aligned}0.1 - 0.00096 &= 0.09904 \\0.4 + 0.00264 &= 0.40264 \\-0.2 + 0.00168 &= -0.19832 \\0.2 - 0.00462 &= 0.19538\end{aligned}$$

12. 使用前面两步的方法，计算出后两层的权重



$$\begin{aligned}0.544 \times (-0.09) \times 0.6 &= -0.04896 \times 0.6 = -0.029376 \\0.505 \times (-0.09) \times 0.6 &= -0.04545 \times 0.6 = -0.02727 \\0.2 - 0.029376 &= 0.170624 \\-0.5 - 0.02727 &= -0.52727\end{aligned}$$

利用更新之后的权重，对训练集的每一条数据
反复进行前面1~12步的计算，直到最后收敛

■ Step 15 :

With the updated weights [V] and [W], error is calculated again and next training set is taken and the error will then get adjusted.

■ Step 16 :

Iterations are carried out till we get the error less than the tolerance.

■ Step 17 :

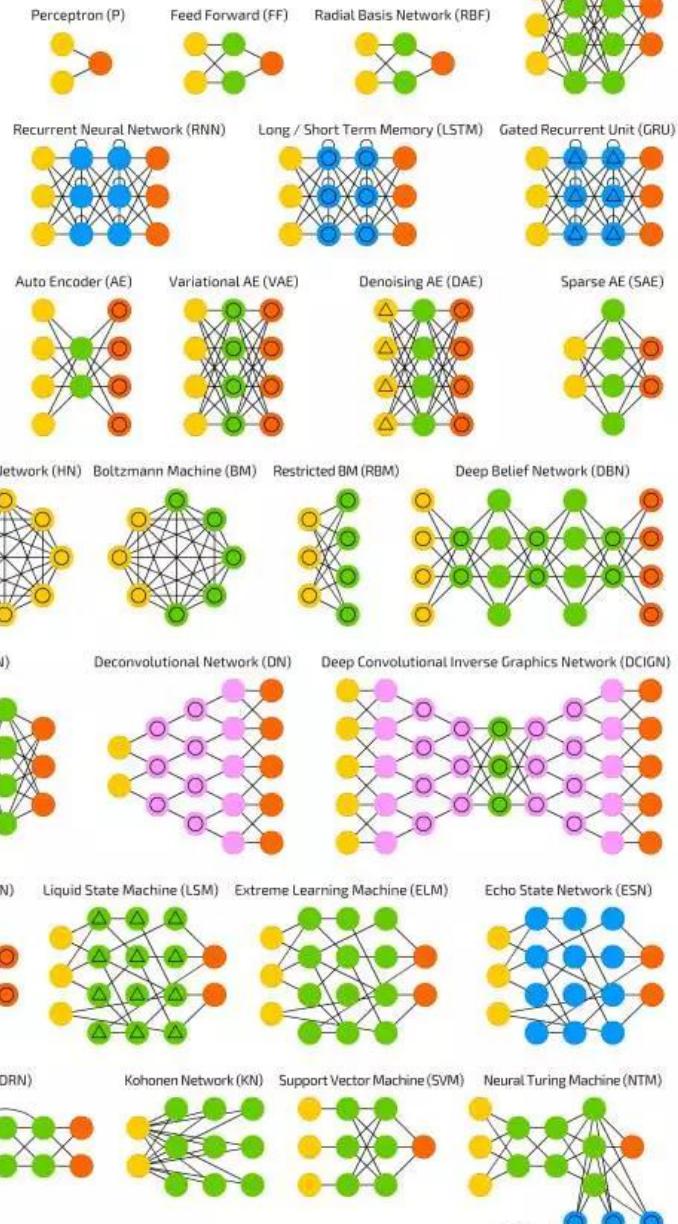
Once the weights are adjusted the network is ready for inferencing new objects .

Neural network

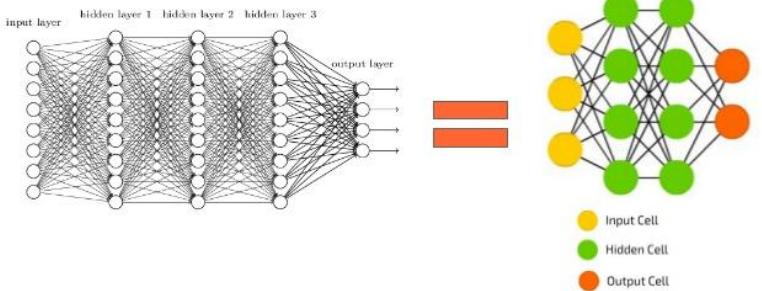
A mostly complete chart of Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

- Backfed Input Cell
- Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- Different Memory Cell
- Kernel
- Convolution or Pool

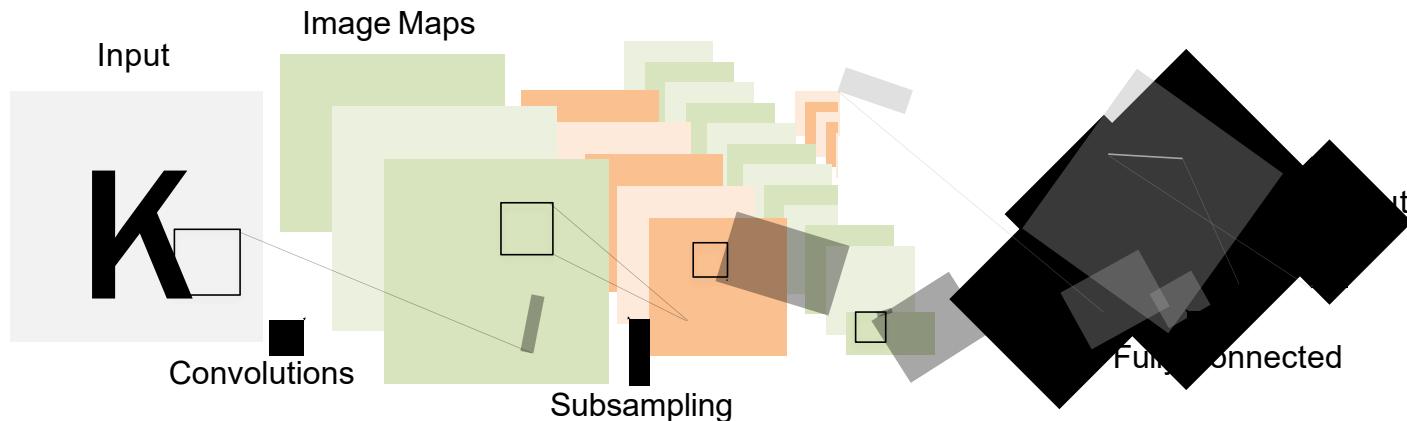


Deep Neural Network (DNN)



1998

LeCun et al.



of transistors



10^6

pentium® II

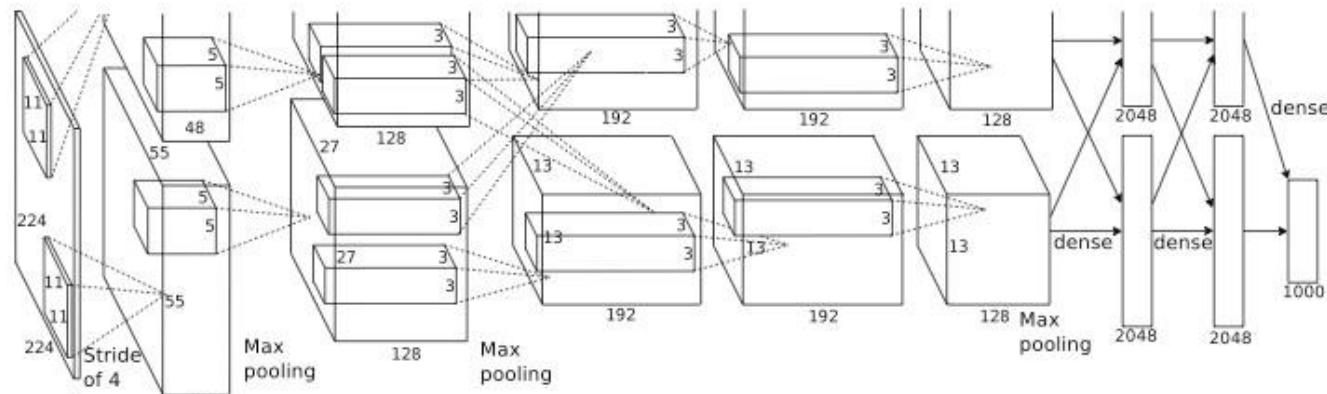
of pixels used in training

10^7



2012

Krizhevsky et al.



of transistors



10^9

GPUs



of pixels used in training

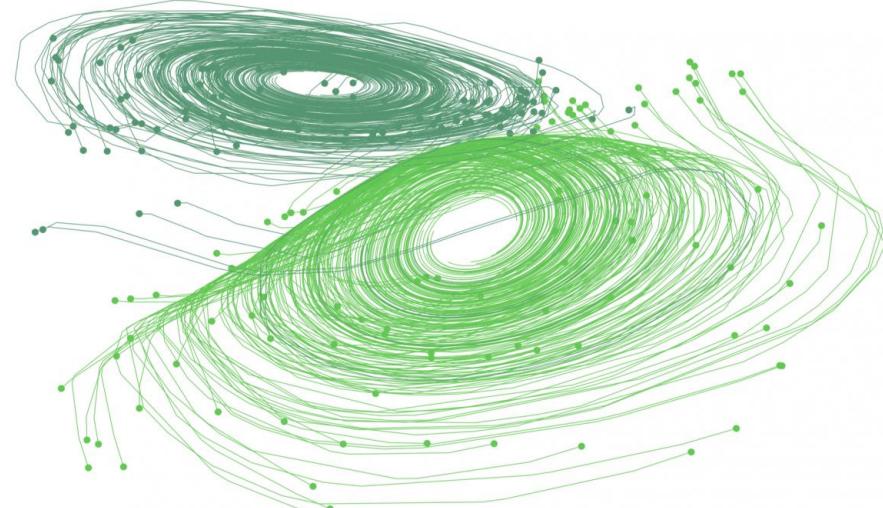
10^{14}



Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012.
Reproduced with permission.

Convolutional Neural Networks (CNN) were NOT invented overnight!

Neural network



Advertisement

Intelligent Machines

A radical new neural network design could overcome big challenges in AI

Researchers borrowed equations from calculus to redesign the core machinery of deep learning so it can model continuous processes like changes in health.

MIT Technology Review

EmTech Asia

22 – 23 January 2019 | Singapore

Jonah Myerberg
Desktop Metal

Samantha Payne
Open Bionics

BOOK NOW!

深度学习的基本过程



深度学习建模（模型训练）流程



Data-Driven Approach

1. Collect a dataset of images and labels
2. Use Machine Learning to train a classifier
3. Evaluate the classifier on new images

Example training set

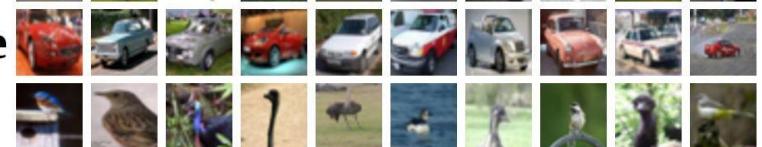
```
def train(images, labels):  
    # Machine learning!  
    return model
```

```
def predict(model, test_images):  
    # Use model to predict labels  
    return test_labels
```

airplane



automobile



bird



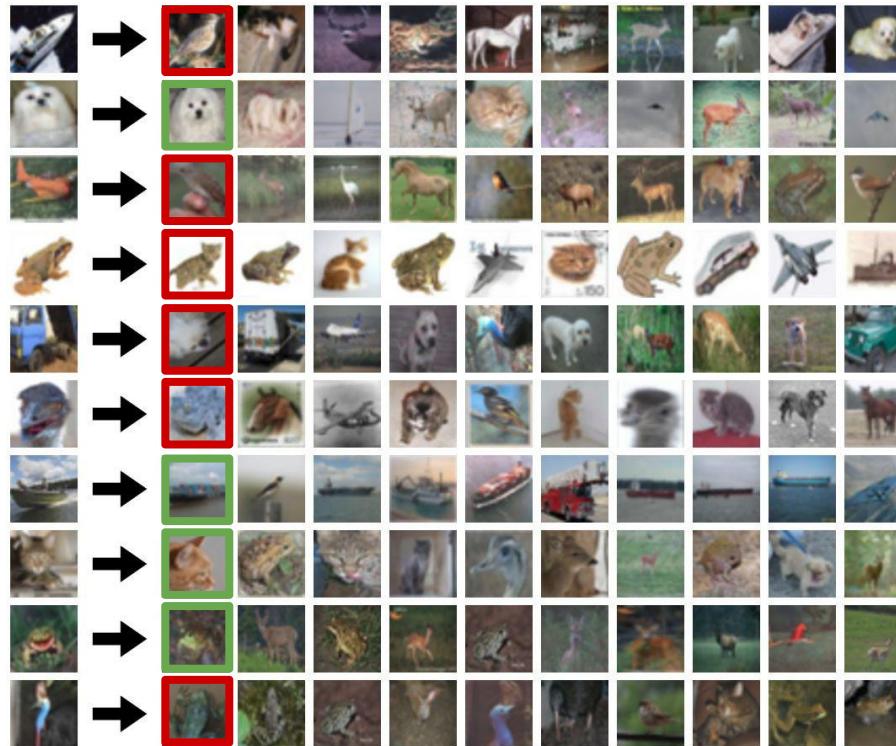
cat



deer



What does this look like?



Parametric Approach: Linear Classifier

3072x1

Image



Array of **32x32x3** numbers
(3072 numbers total)

$$f(x, W) = Wx$$

10x1 10x3072

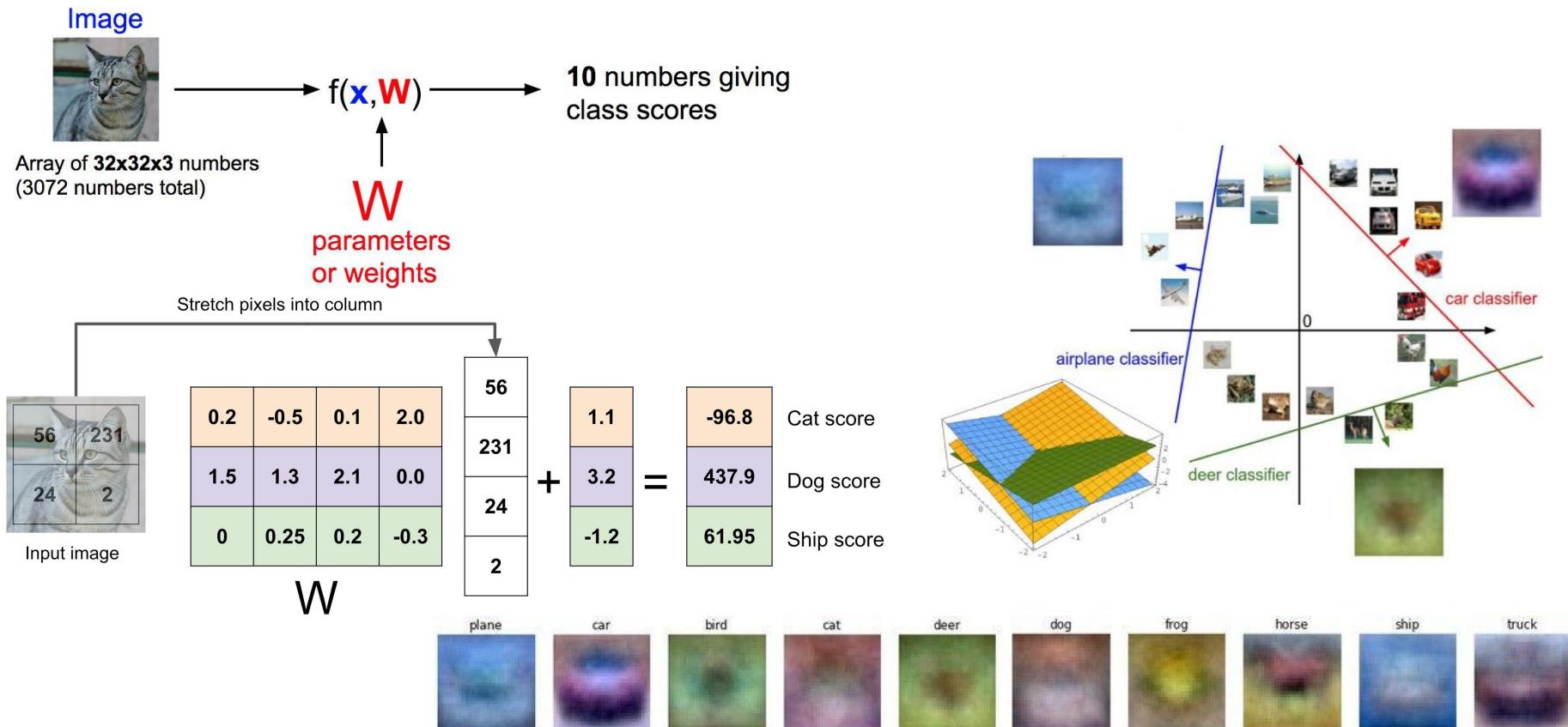
W

parameters
or weights

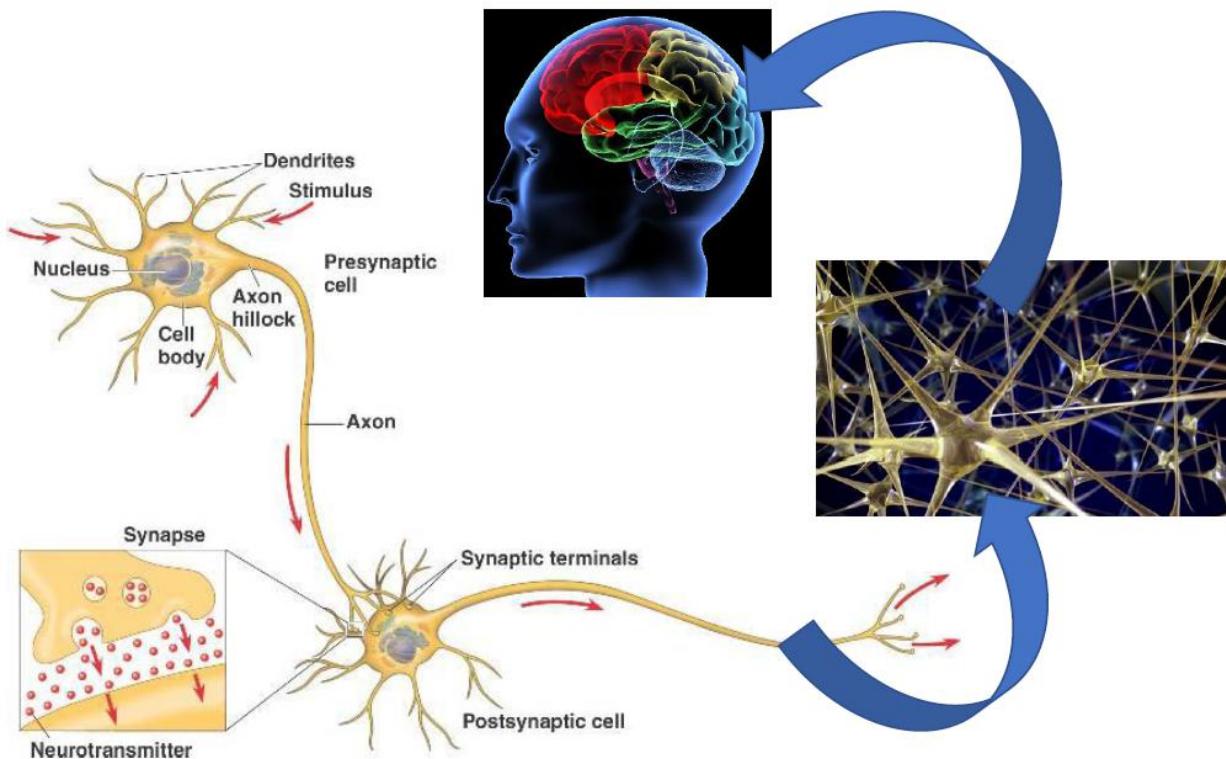
10 numbers giving
class scores

Parametric Approach: Linear Classifier

$$f(x, W) = Wx + b$$



Neural Network introduction



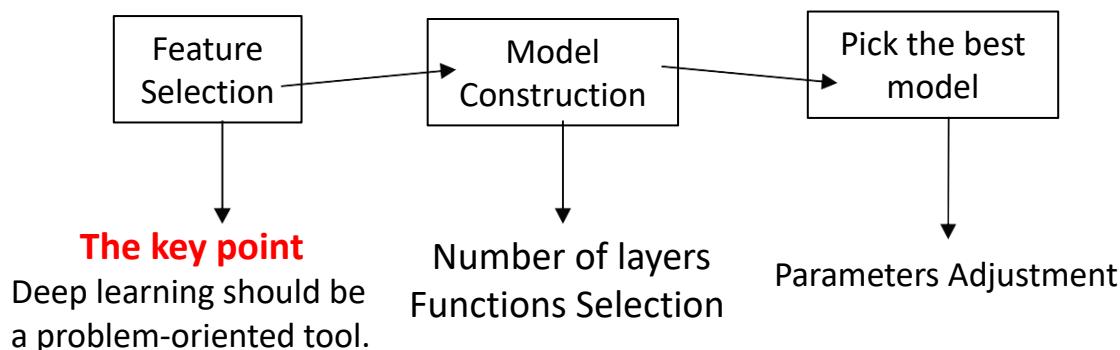
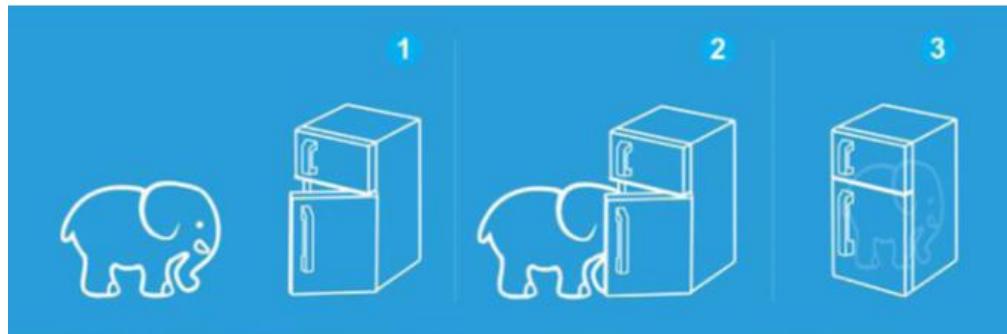
Neural network model is inspired by human
neuron

Each neuron is a judgement unit

Hubel & Wiesel, 1959

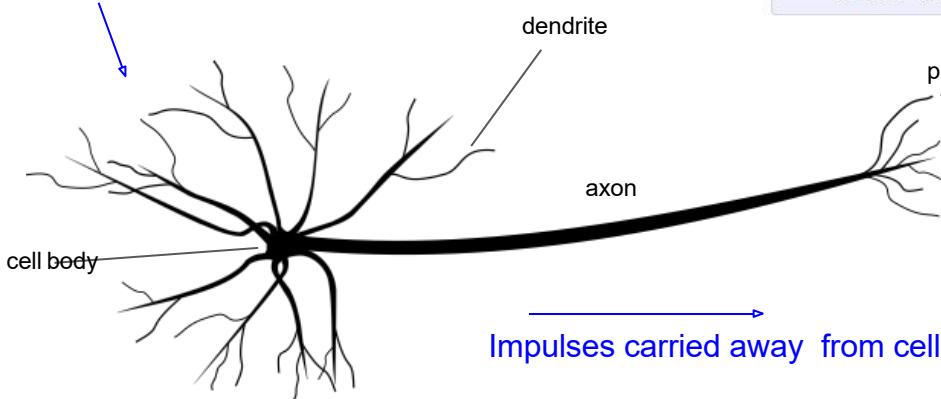
Neural Network workflow

Neural network is so simple.....

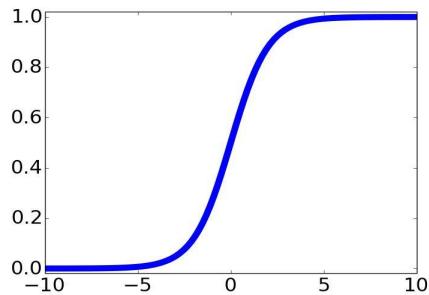


Neuron Network

Impulses carried toward cell body



This image by Felipe Perucho
is licensed under CC-BY3.0

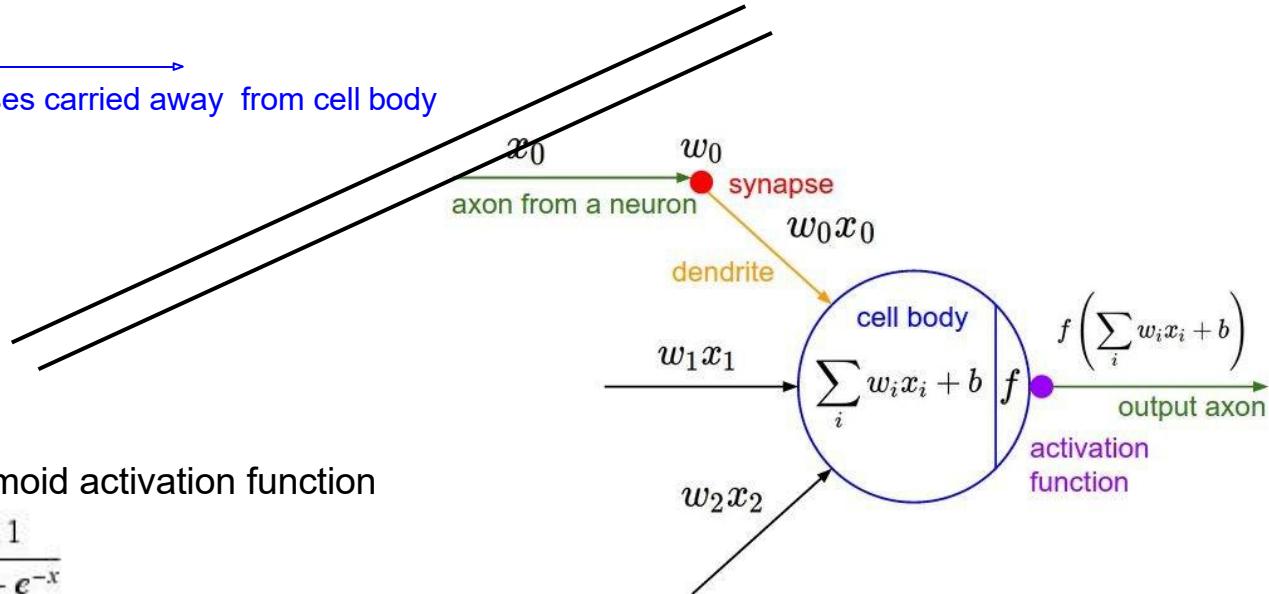


sigmoid activation function

$$\frac{1}{1 + e^{-x}}$$

```
class Neuron:  
    # ...  
    def neuron_tick(inputs):  
        """ assume inputs and weights are 1-D numpy arrays and bias is a number """  
        cell_body_sum = np.sum(inputs * self.weights) + self.bias  
        firing_rate = 1.0 / (1.0 + math.exp(-cell_body_sum)) # sigmoid activation function  
        return firing_rate
```

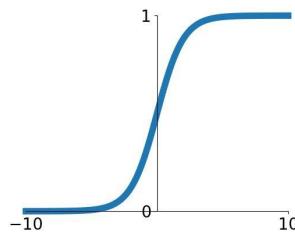
Impulses carried away from cell body



Activation Functions

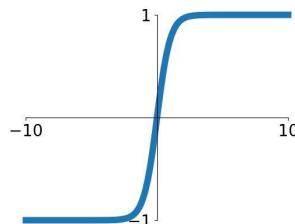
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



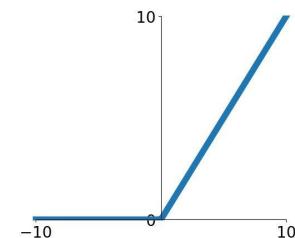
tanh

$$\tanh(x)$$



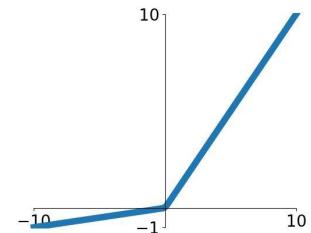
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

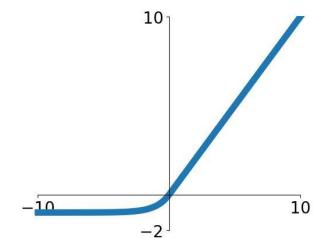


Maxout

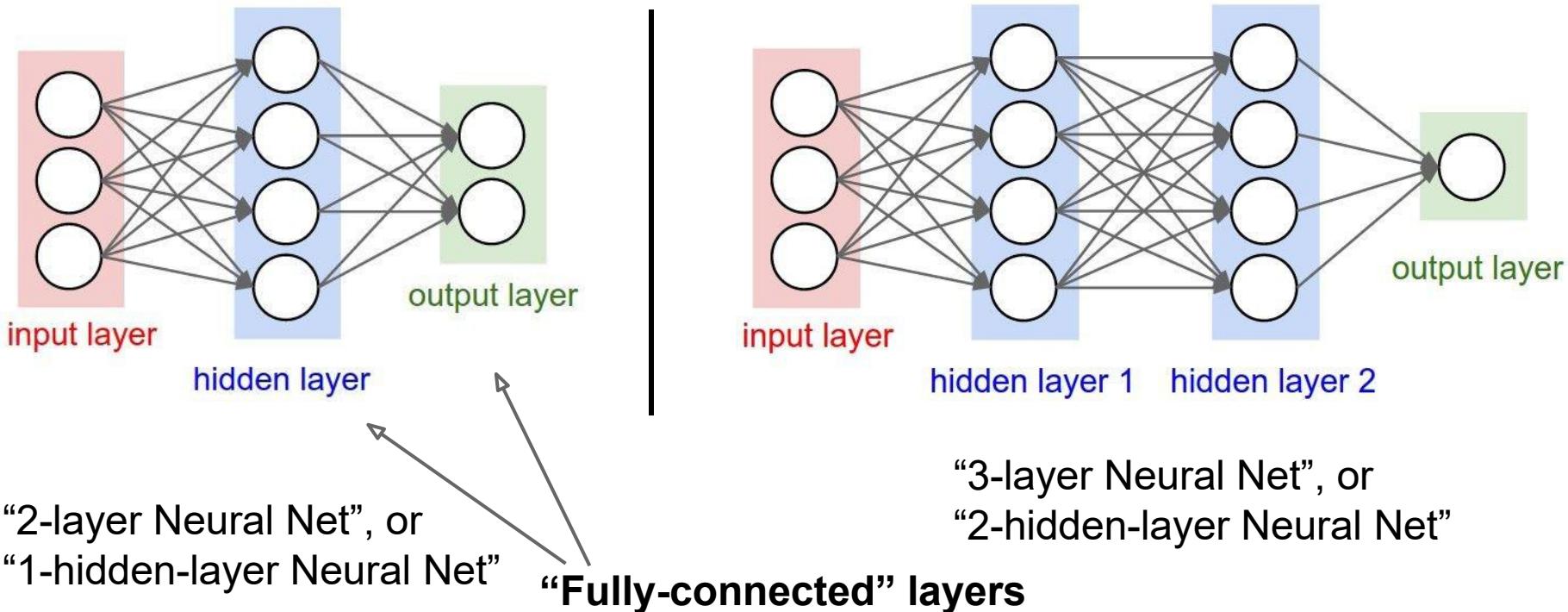
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Neural networks: Architectures



Convolutional Neural Networks

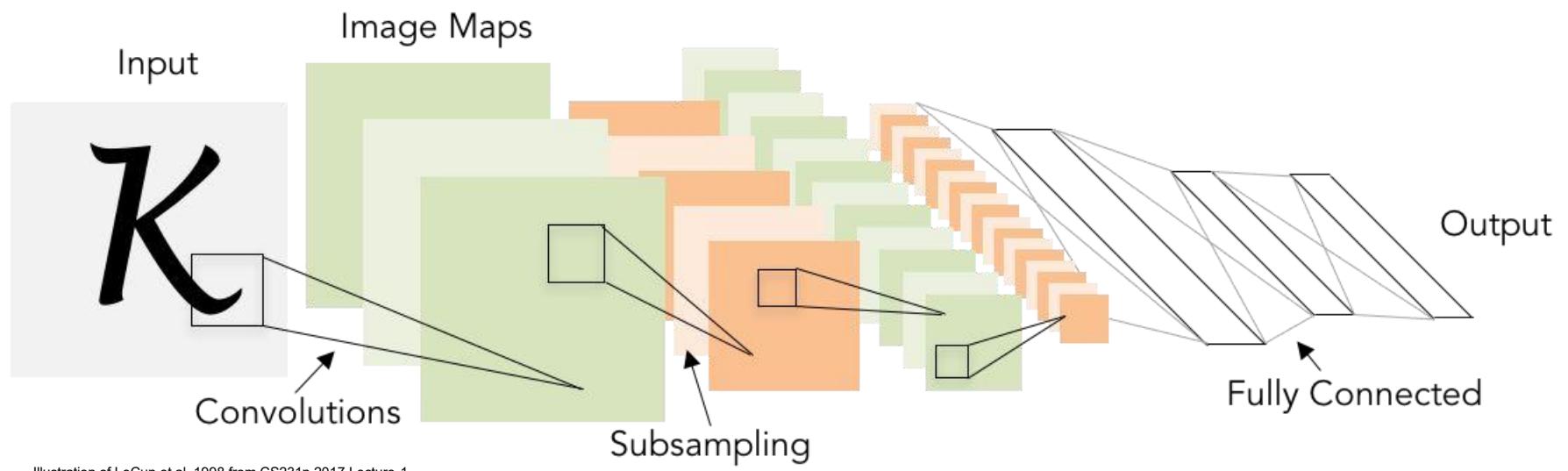
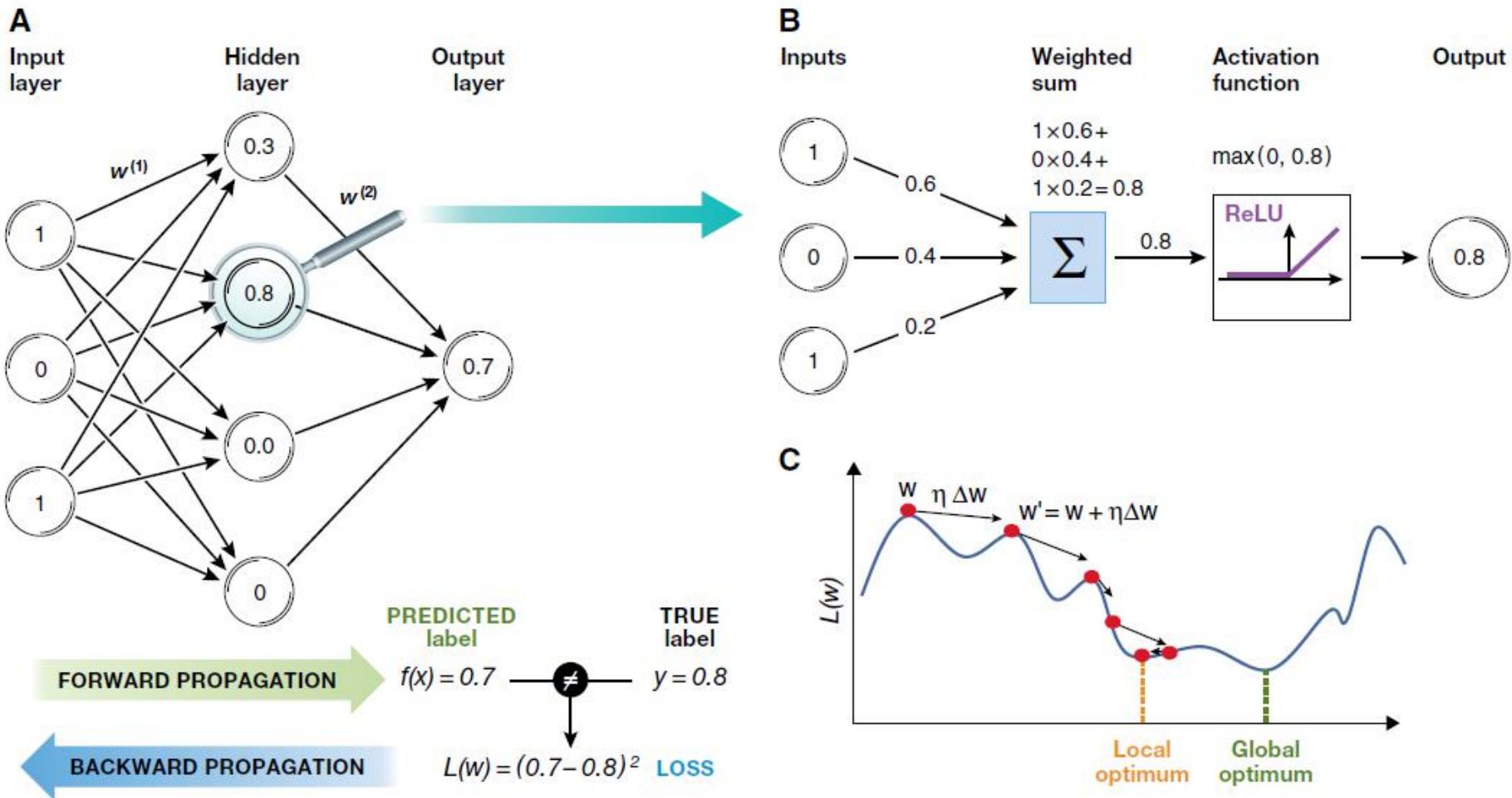


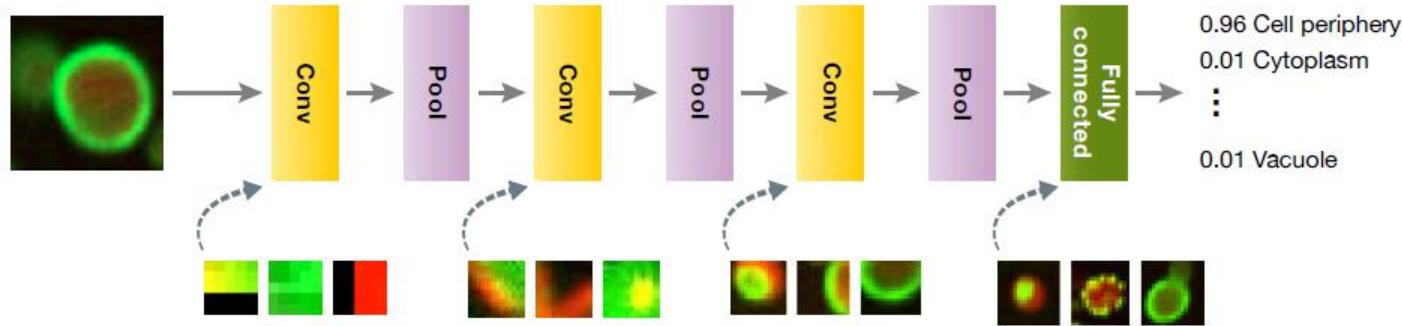
Illustration of LeCun et al. 1998 from CS231n 2017 Lecture 1

Artificial Neural Network



Convolution Neural Network

Convolution neural network is similar to neural network, it is a good approach to process figure



Convolution and pooling process : edge features extraction



Convolution Neural Network

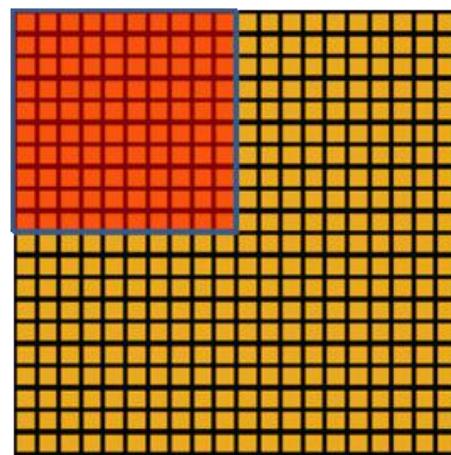
1 <small>x1</small>	1 <small>x0</small>	1 <small>x1</small>	0	0
0 <small>x0</small>	1 <small>x1</small>	1 <small>x0</small>	1	0
0 <small>x1</small>	0 <small>x0</small>	1 <small>x1</small>	1	1
0	0	1	1	0
0	1	1	0	0

Image

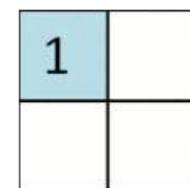
4		

Convolved
Feature

Convolution Neural Network

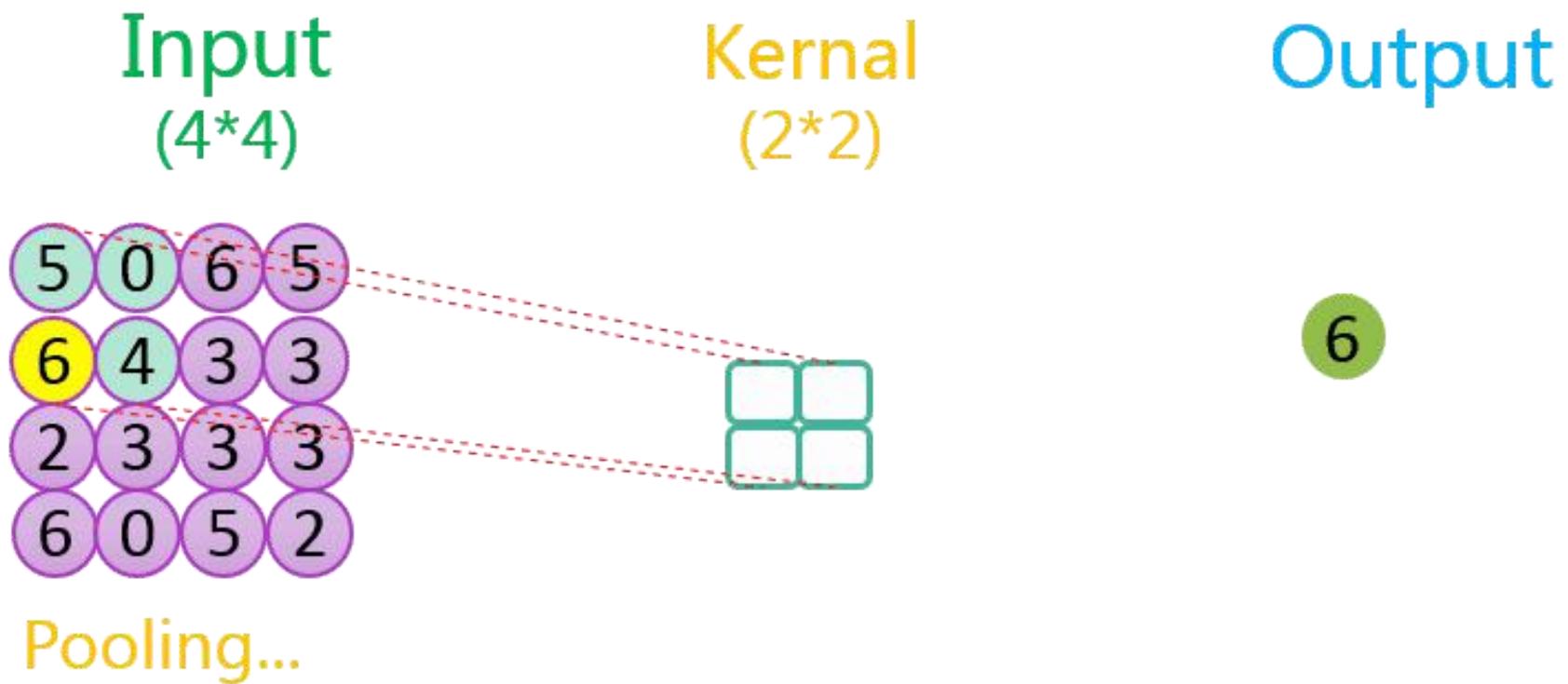


Convolved
feature

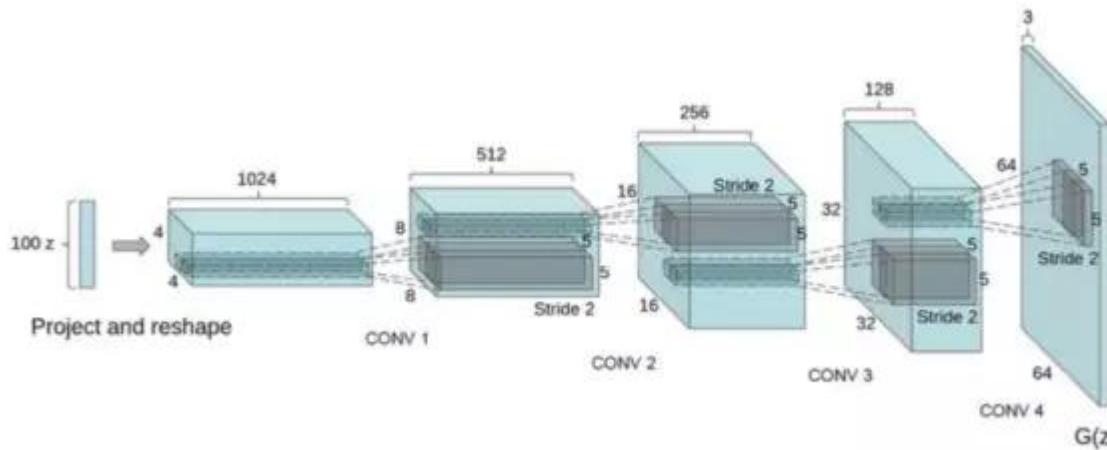
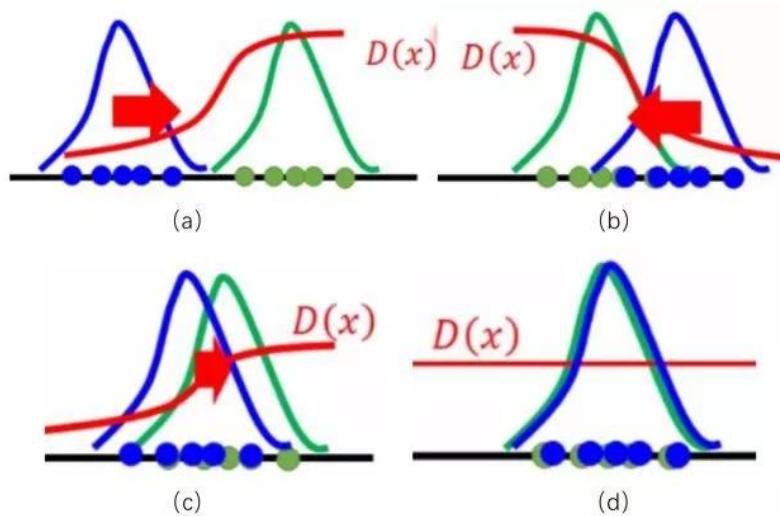


Pooled
feature

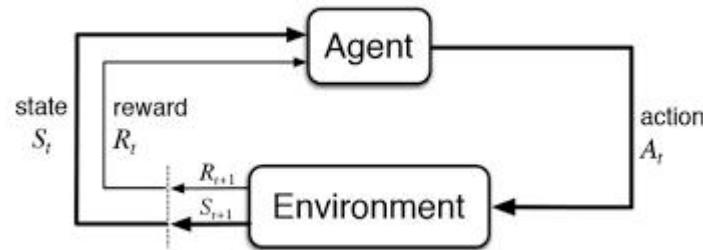
Convolution Neural Network



Generative adversarial network, GAN

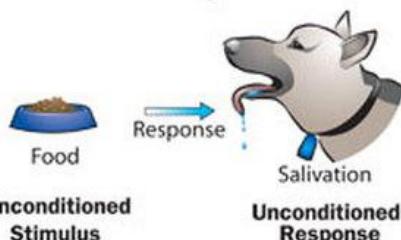


Reinforcement learning



How Dog Training Works

1. Before Conditioning



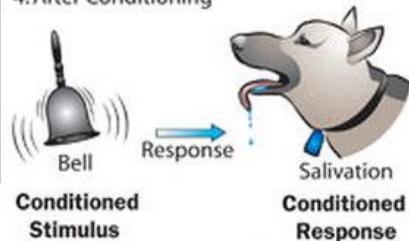
2. Before Conditioning



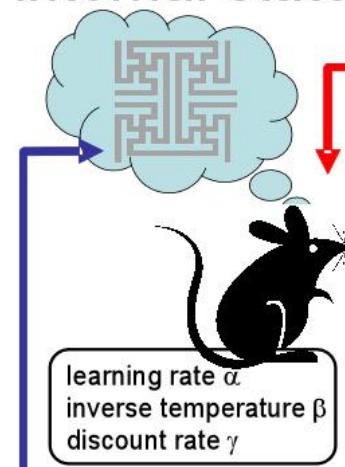
3. During Conditioning



4. After Conditioning



internal state



reward

environment

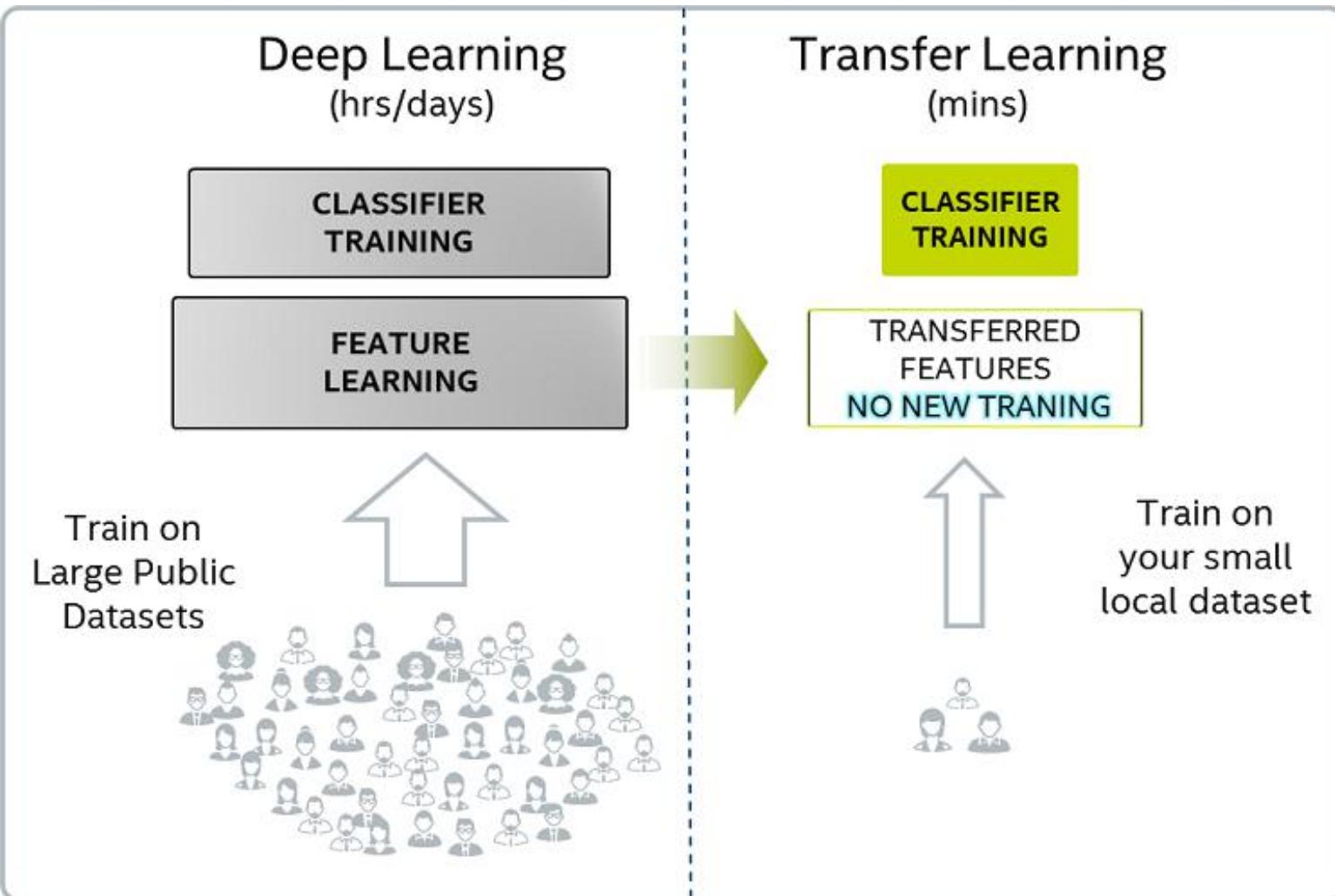
action



learning rate α
inverse temperature β
discount rate γ

observation

Transfer learning



深度学习的特点

深度学习常用算法介绍

深度学习常用框架介绍

深度学习

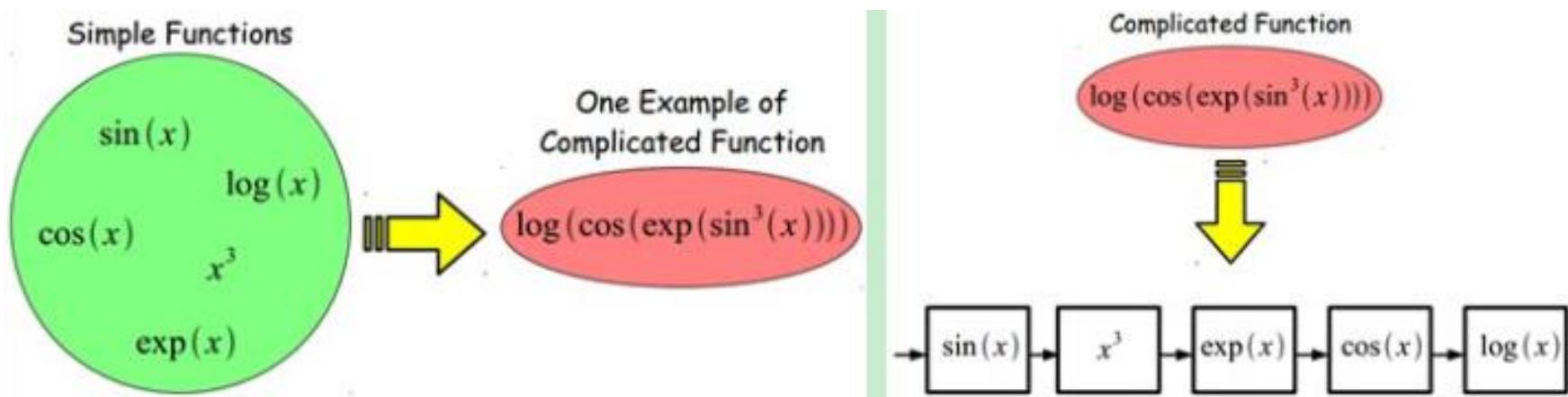
- 2006年，加拿大多伦多大学教授、机器学习领域的泰斗Geoffrey Hinton在《科学》上发表论文提出深度学习主要观点：
 - 1) 多隐层的人工神经网络具有优异的特征学习能力，学习得到的特征对数据有更本质的刻画，从而有利于可视化或分类；
 - 2) 深度神经网络在训练上的难度，可以通过“逐层初始化”（layer-wise pre-training）来有效克服，逐层初始化可通过无监督学习实现的。

深度学习

- **本质：**通过构建多隐层的模型和海量训练数据（可为无标签数据），来学习更有用的特征，从而最终提升分类或预测的准确性。“深度模型”是手段，“特征学习”是目的。
- **与浅层学习区别：**
 - 1) 强调了模型结构的深度，通常有5-10多层的隐层节点；
 - 2) 明确突出了特征学习的重要性，通过逐层特征变换，将样本在原空间的特征表示变换到一个新特征空间，从而使分类或预测更加容易。与人工规则构造特征的方法相比，利用大数据来学习特征，更能够刻画数据的丰富内在信息。

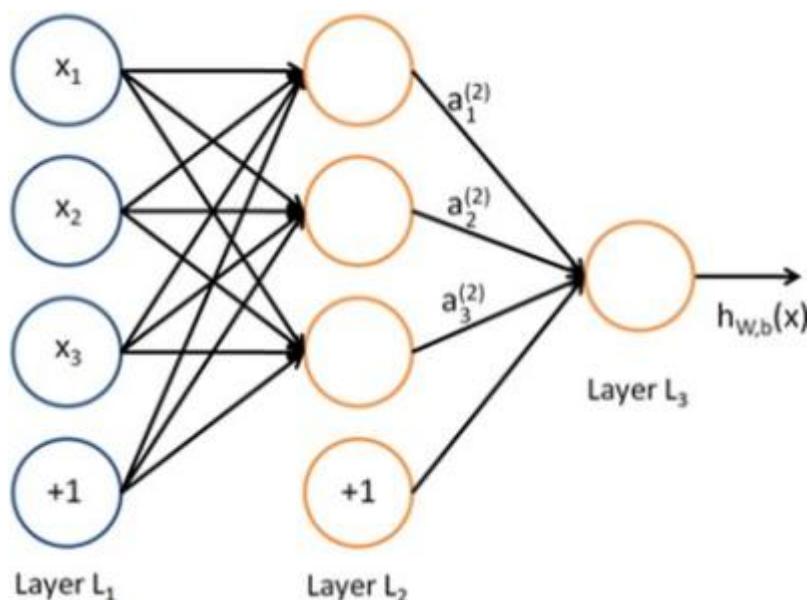
深度学习

- 好处：可通过学习一种深层非线性网络结构，实现复杂函数逼近，表征输入数据分布式表示。



深度学习 vs. 神经网络

神经网络 :



深度学习:



含多个隐层的深度学习模型

深度学习 vs. 神经网络

相同点：二者均采用分层结构，系统包括输入层、隐层（多层）、输出层组成的多层网络，只有相邻层节点之间有连接，同一层以及跨层节点之间相互无连接，每一层可以看作是一个logistic 回归模型。

不同点：

神经网络：采用BP算法调整参数，即采用迭代算法来训练整个网络。随机设定初值，计算当前网络的输出，然后根据当前输出和样本真实标签之间的差去改变前面各层的参数，直到收敛；

深度学习：采用逐层训练机制。采用该机制的原因在于如果采用BP机制，对于一个deep network（7层以上），残差传播到最前面的层将变得很小，出现所谓的gradient diffusion（梯度扩散）。

深度学习 vs. 神经网络

- 神经网络的局限性：
 - 1) 比较容易过拟合，参数比较难调整，而且需要不少技巧；
 - 2) 训练速度比较慢，在层次比较少（小于等于3）的情况下效果并不比其它方法更优；

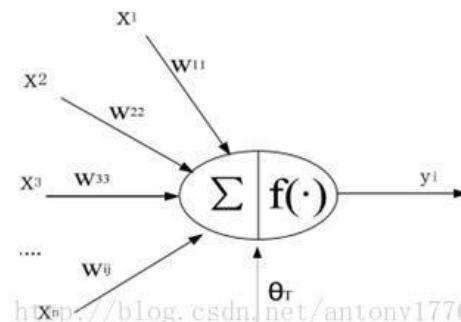
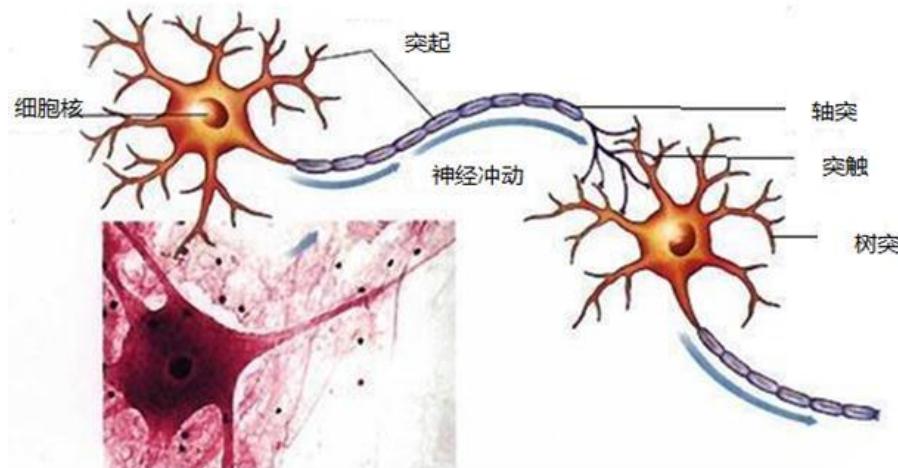
深度学习的特点

深度学习常用算法介绍

深度学习常用框架介绍

人工神经网络 (ANN)

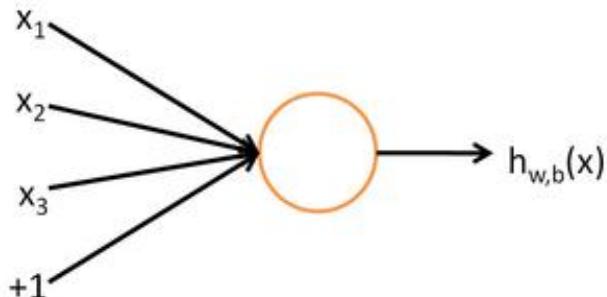
- 人工神经网络 (Artificial Neural Networks) 是一种模仿生物神经网络行为特征，进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度，通过调整内部大量节点（神经元）之间相互连接的权重，从而达到处理信息的目的。



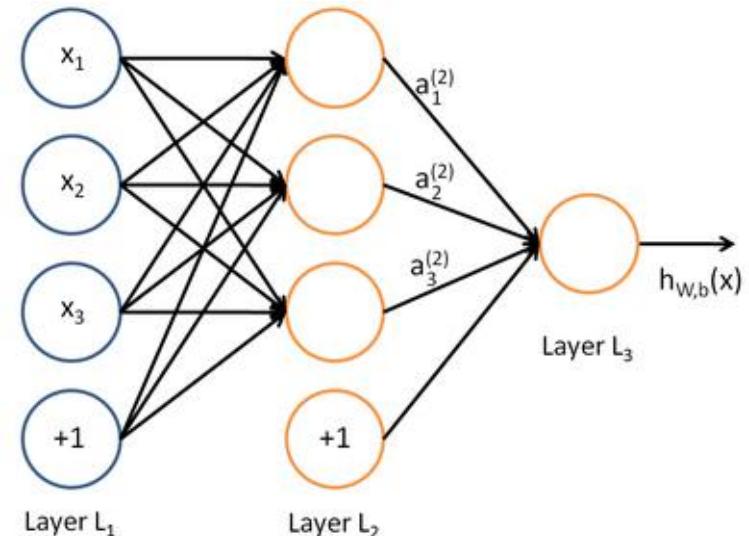
<http://blog.csdn.net/antony1776>

人工神经网络 (ANN)

● 神经网络



$$h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^3 W_i x_i + b)$$



$$a_1^{(2)} = f(W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + W_{13}^{(1)} x_3 + b_1^{(1)})$$

$$a_2^{(2)} = f(W_{21}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{23}^{(1)} x_3 + b_2^{(1)})$$

$$a_3^{(2)} = f(W_{31}^{(1)} x_1 + W_{32}^{(1)} x_2 + W_{33}^{(1)} x_3 + b_3^{(1)})$$

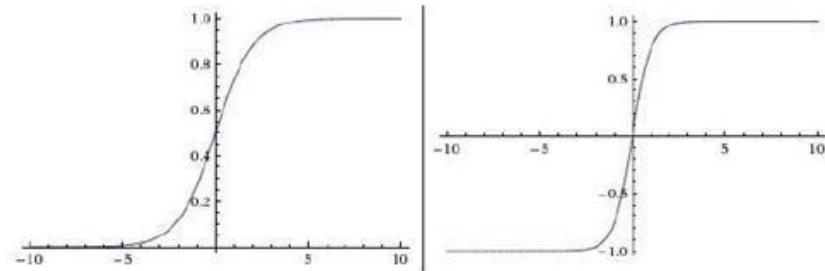
$$h_{W,b}(x) = a_1^{(3)} = f(W_{11}^{(2)} a_1^{(2)} + W_{12}^{(2)} a_2^{(2)} + W_{13}^{(2)} a_3^{(2)} + b_1^{(2)})$$

人工神经网络 (ANN)

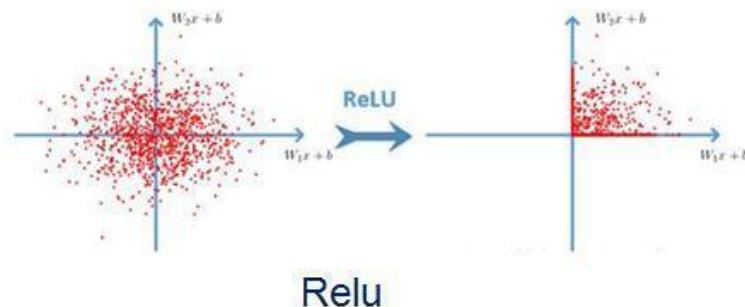
- 人工神经网络的重要概念：

- 1 权值矩阵：相当于神经网络的记忆！在训练的过程中，动态调整和适应。

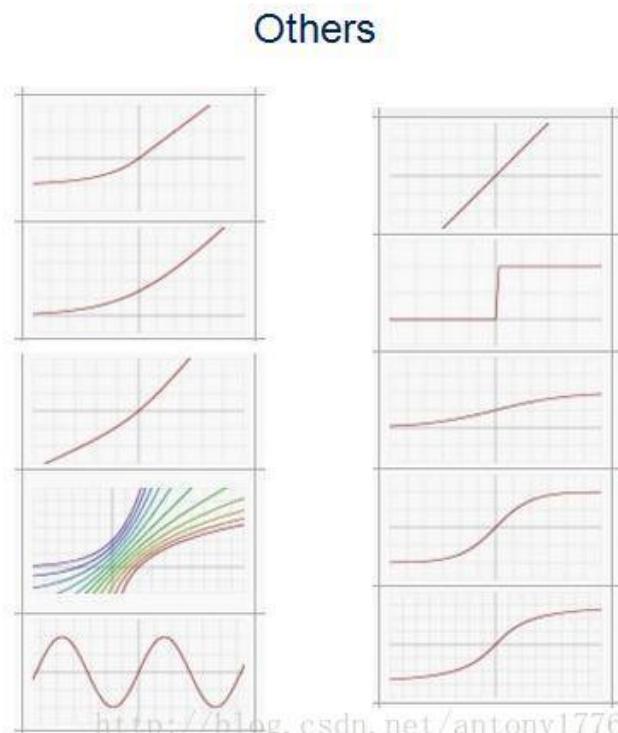
- 2 激励函数：



Sigmoid



ReLU



人工神经网络 (ANN)

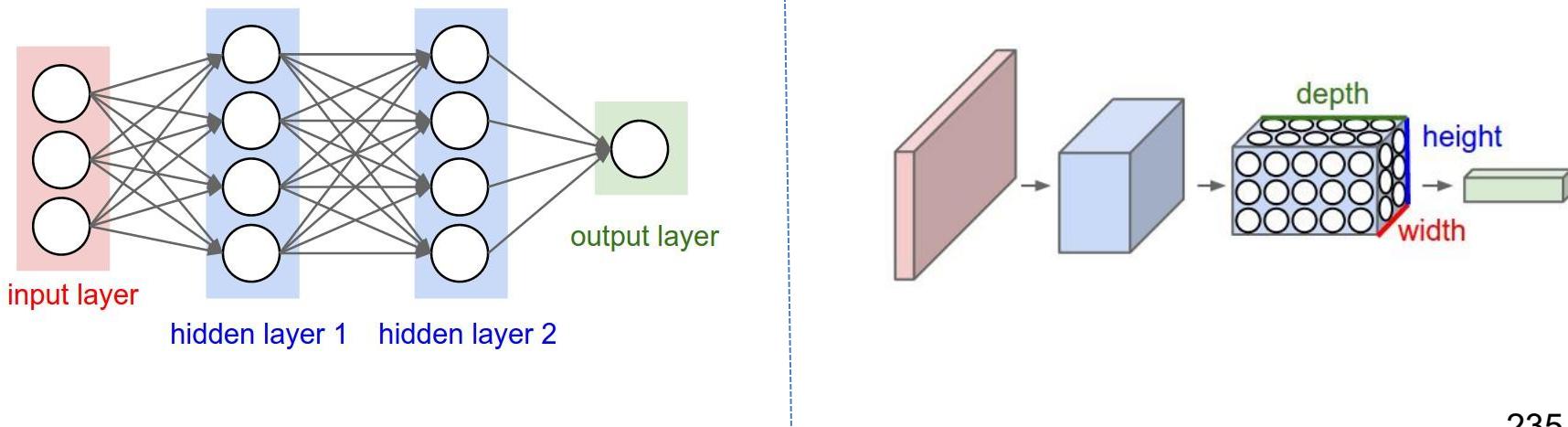
- 人工神经网络的重要概念：

激励函数很重要，无论是对建立神经网络的模型，还是理解神经网络。首先要了解，它有以下几个影响：

- 1 如何能更好的求解目标函数的极值！——高等数学中求解函数极值的知识！可微，单调！
- 2 如何提升训练效率，让梯度的优化方法更稳定；
- 3 权值的初始值，不影响训练结果！

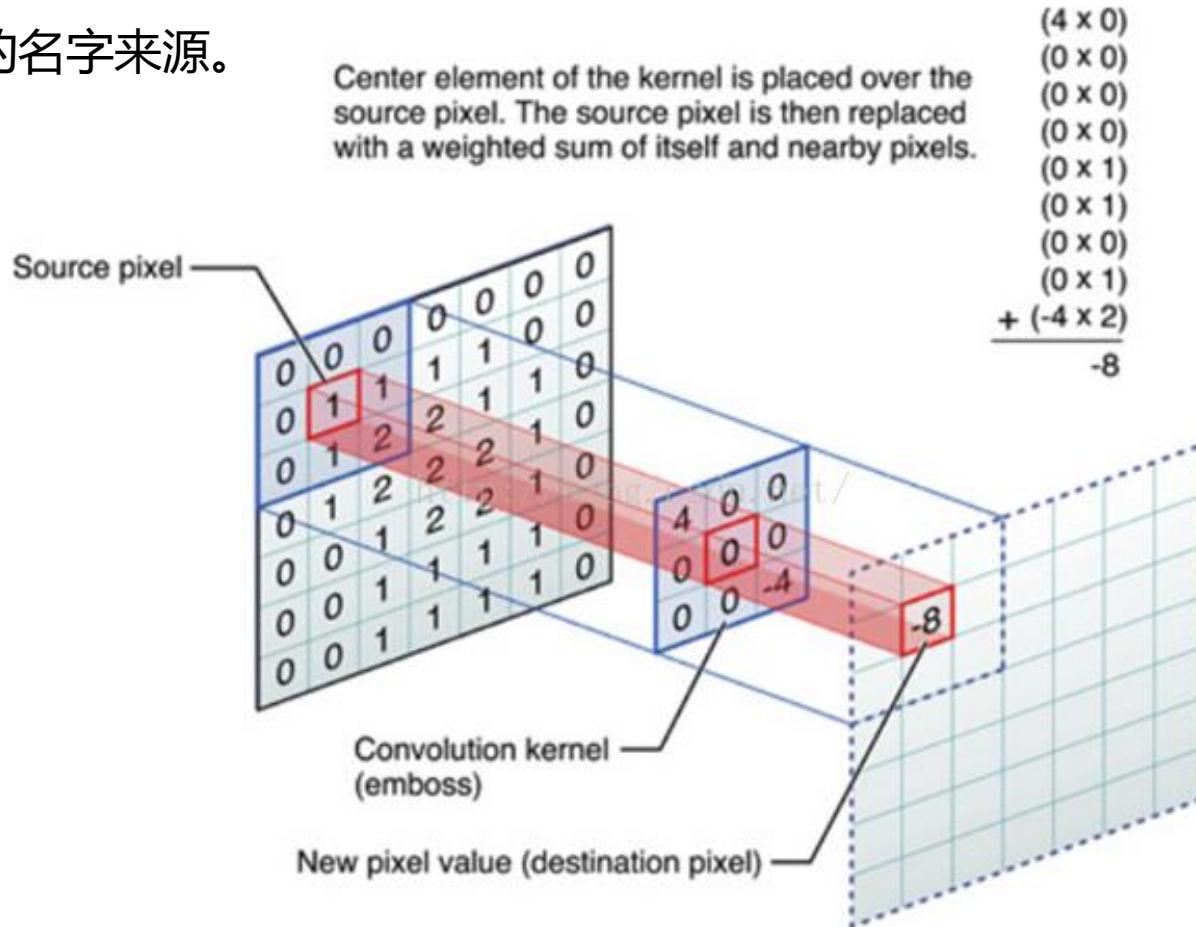
卷积神经网络 (CNN)

- 卷积神经网络 (Convolutional Neural Networks / CNNs / ConvNets) 与普通神经网络非常相似，它们都由具有可学习的权重和偏置常量(biases)的神经元组成。每个神经元都接收一些输入，并做一些点积计算，输出是每个分类的分数，普通神经网络里的一些计算技巧到这里依旧适用。
- 与普通神经网络不同之处：卷积神经网络默认输入是图像，可以让我们把特定的性质编码入网络结构，使是我们的前馈函数更加有效率，并减少了大量参数。



卷积神经网络 (CNN)

- 卷积操作：对图像（不同的数据窗口数据）和滤波矩阵（一组固定的权重：因为每个神经元的权重固定，所以又可以看做一个恒定的滤波器filter）做内积（逐个元素相乘再求和）的操作就是所谓的『卷积』操作，也是卷积神经网络的名字来源。



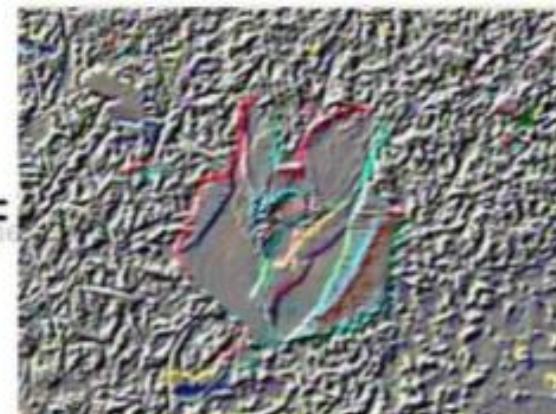
卷积神经网络 (CNN)



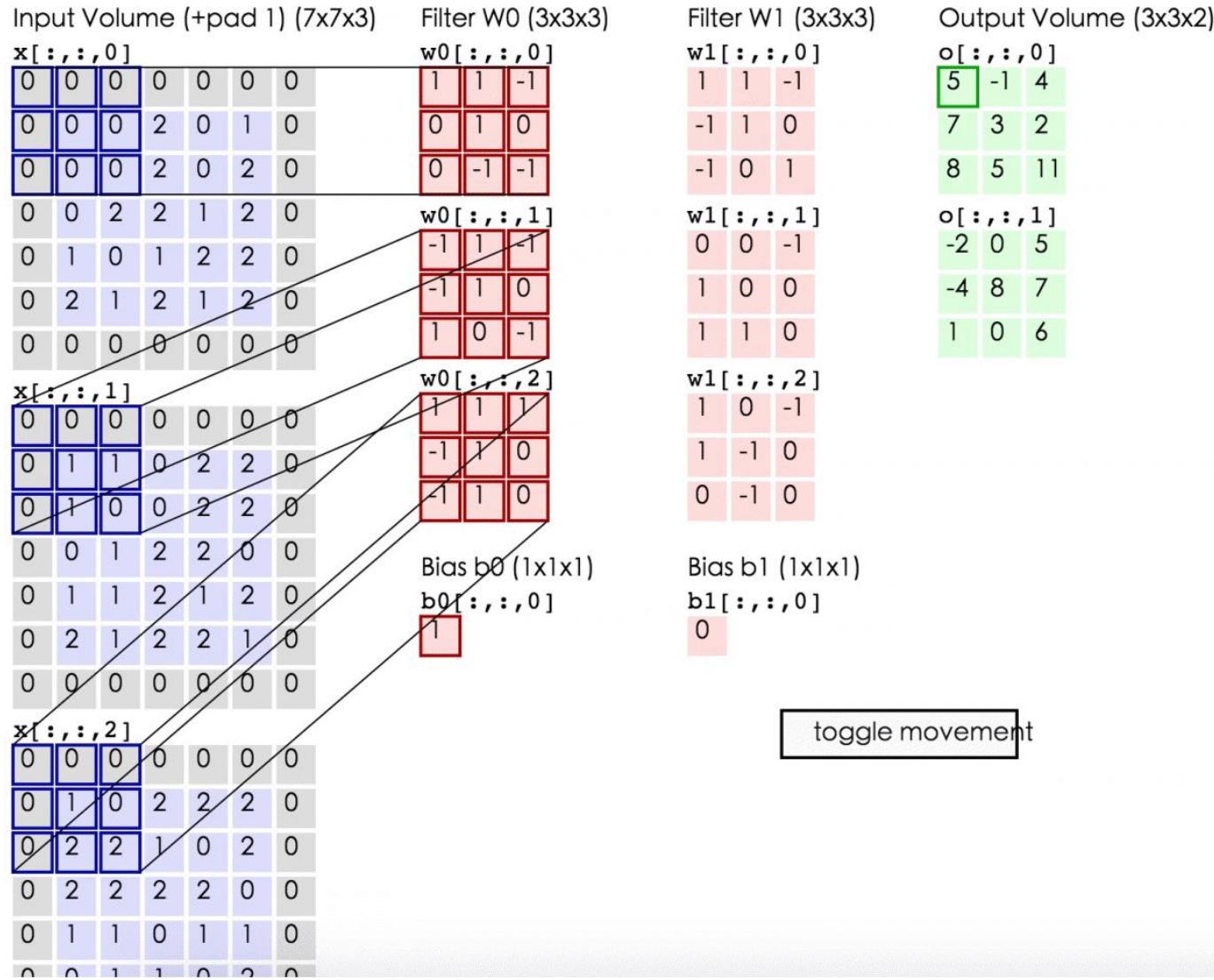
$$\ast \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} =$$



$$\ast \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} =$$



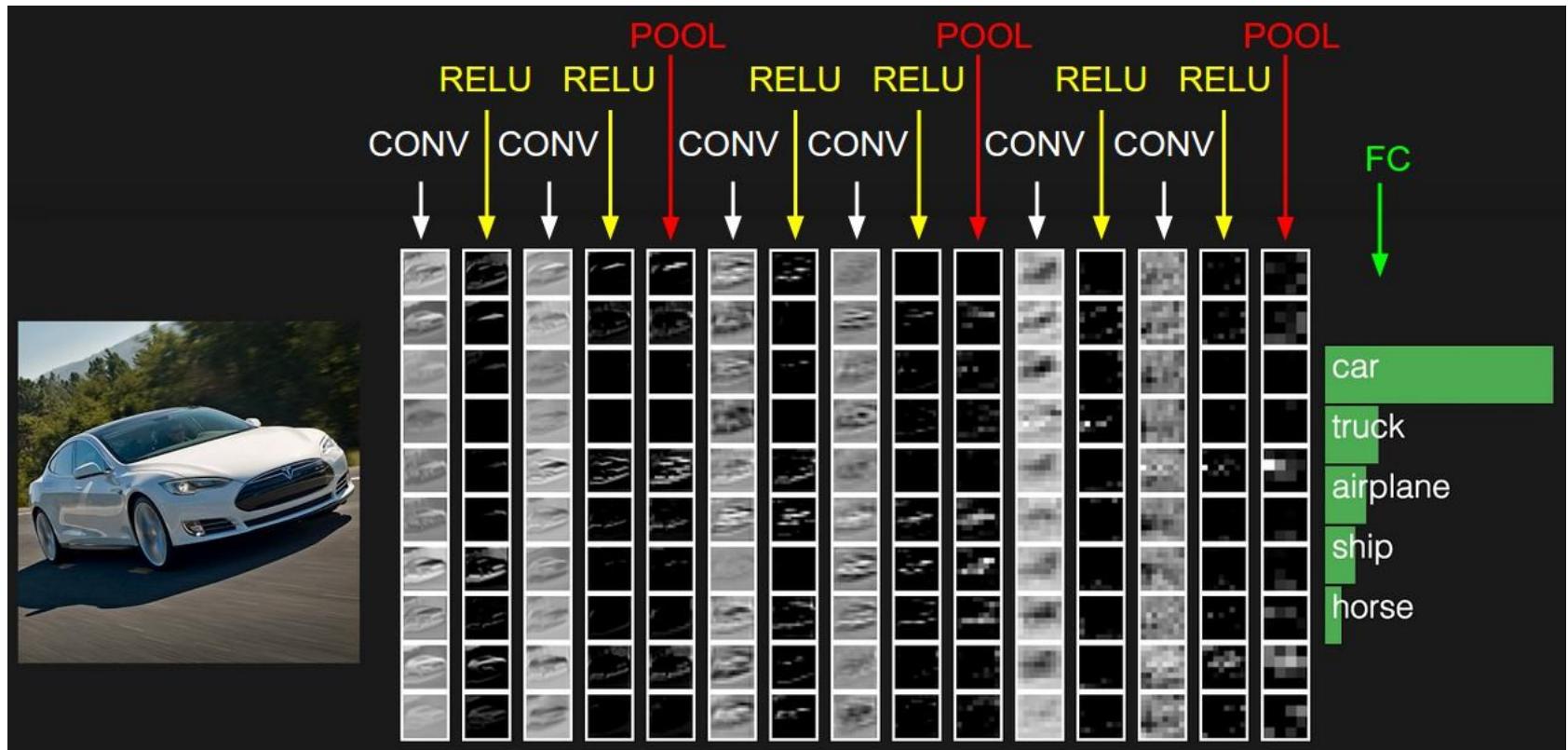
卷积神经网络 (CNN)



卷积神经网络 (CNN)

- 卷积层 (Convolutional layer) , 卷积神经网路中每层卷积层由若干卷积单元组成，每个卷积单元的参数都是通过反向传播算法优化得到的。卷积运算的目的是提取输入的不同特征，第一层卷积层可能只能提取一些低级的特征如边缘、线条和角等层级，更多层的网络能从低级特征中迭代提取更复杂的特征。
- 线性整流层 (Rectified Linear Units layer, ReLU layer) , 这一层神经的激励函数 (Activation function) 使用线性整流 (Rectified Linear Units, ReLU) $f(x)=\max(0,x)$ 。
- 池化层 (Pooling layer) , 通常在卷积层之后会得到维度很大的特征，将特征切成几个区域，取其最大值或平均值，得到新的、维度较小的特征。
- 全连接层 (Fully-Connected layer) , 把所有局部特征结合变成全局特征，用来计算最后每一类的得分。

卷积神经网络 (CNN)

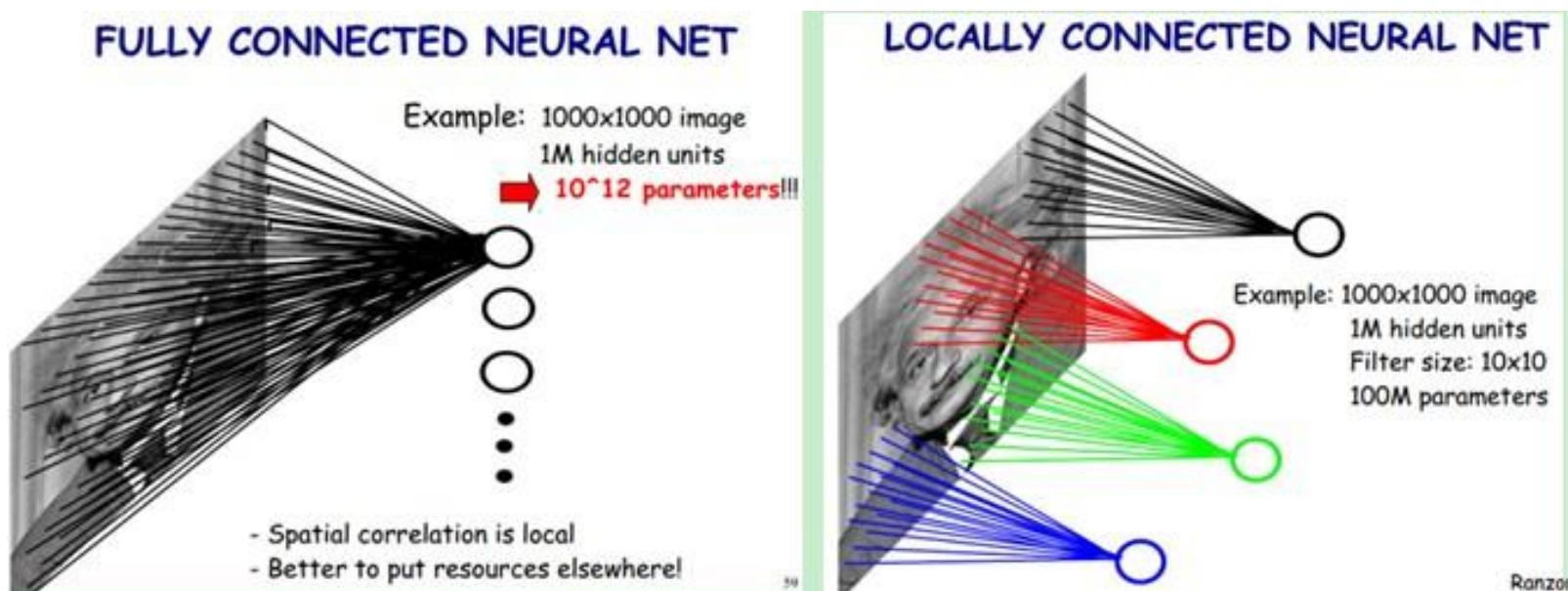


卷积神经网络 (CNN)

- 在图像处理中，往往把图像表示为像素的向量，比如一个 1000×1000 的图像，可以表示为一个1000000的向量。在上一节中提到的神经网络中，如果隐含层数目与输入层一样，即也是1000000时，那么输入层到隐含层的参数数据为 $1000000 \times 1000000 = 10^{12}$ ，这样就太多了，基本没法训练。所以图像处理要想练成神经网络大法，必先减少参数加快速度。就跟辟邪剑谱似的，普通人练得很挫，一旦自宫后内力变强剑法变快，就变的很牛了。

卷积神经网络 (CNN)

- 卷积神经网络有两种神器可以降低参数数目，第一种神器叫做局部感知。
- 在下方右图中，假如每个神经元只和 10×10 个像素值相连，那么权值数据为 1000000×100 个参数，减少为原来的万分之一。而那 10×10 个像素值对应的 10×10 个参数，其实就相当于卷积操作。

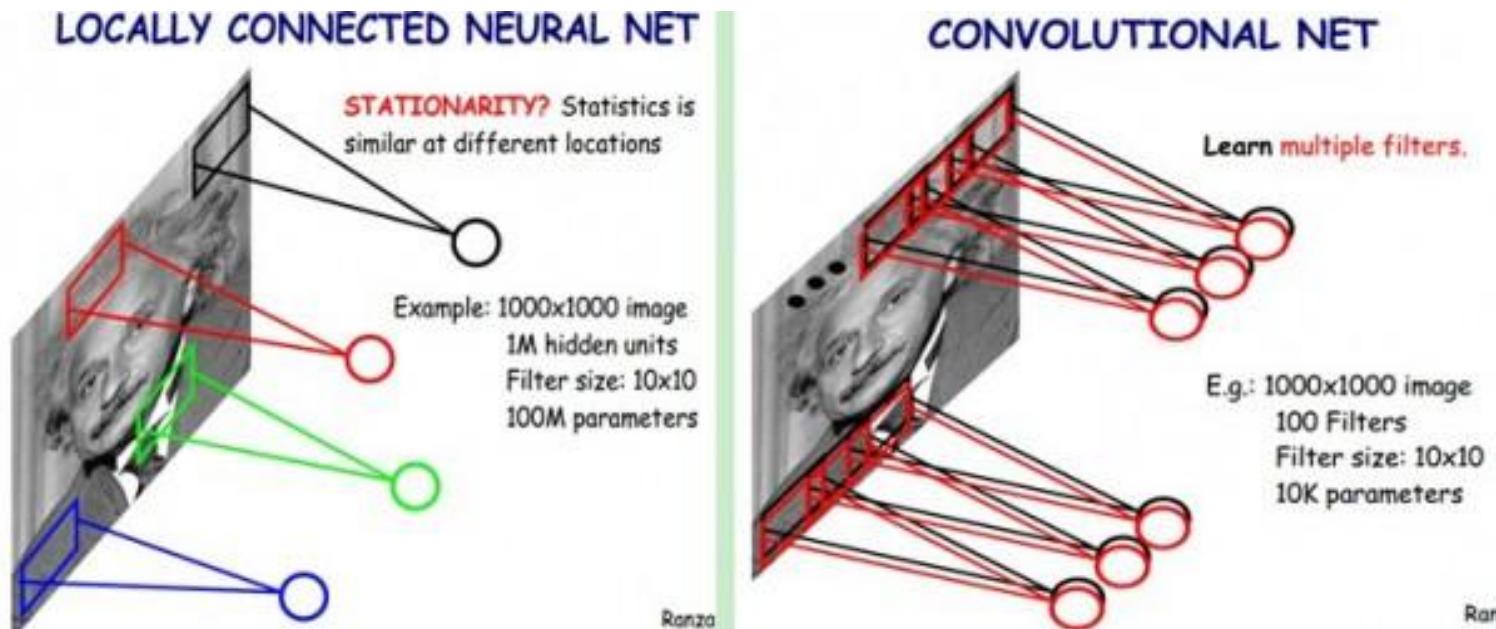


卷积神经网络 (CNN)

- 这样的话参数仍然过多，那么就启动第二级神器，即权值共享。在上面的局部连接中，每个神经元都对应100个参数，一共1000000个神经元，如果这1000000个神经元的100个参数都是相等的，那么参数数目就变为100了。
- 怎么理解权值共享呢？我们可以将这100个参数（也就是卷积操作）看成是提取特征的方式，该方式与位置无关。这其中隐含的原理则是：图像的一部分的统计特性与其他部分是一样的。这也意味着我们在这一部分学习的特征也能用在另一部分上，所以对于这个图像上的所有位置，我们都能使用同样的学习特征。

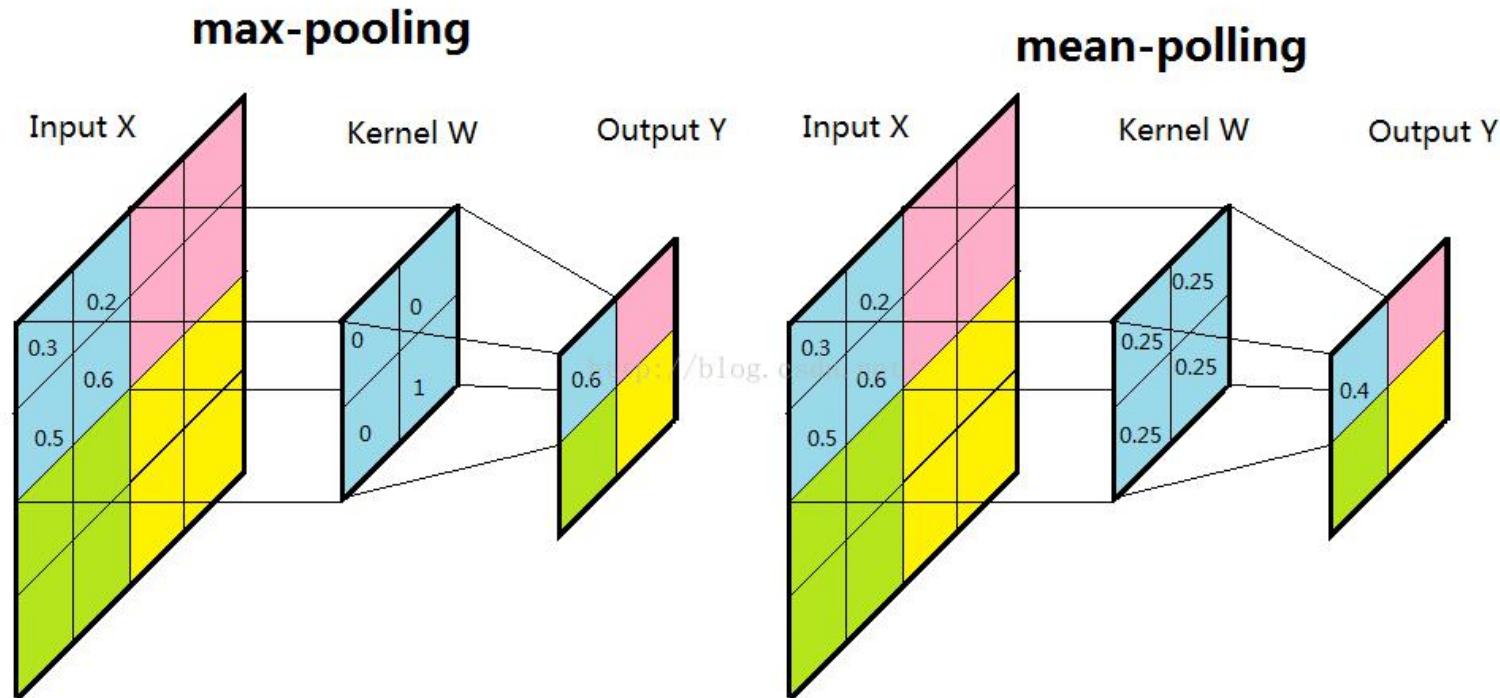
卷积神经网络 (CNN)

- 上面所述只有100个参数时，表明只有1个 $100*100$ 的卷积核，显然，特征提取是不充分的，我们可以添加多个卷积核，比如32个卷积核，可以学习32种特征。在有多个卷积核时，如下图所示：



卷积神经网络 (CNN)

- 池化，也称作下采样，可以实现降维。常用有最大值池化和均值池化。

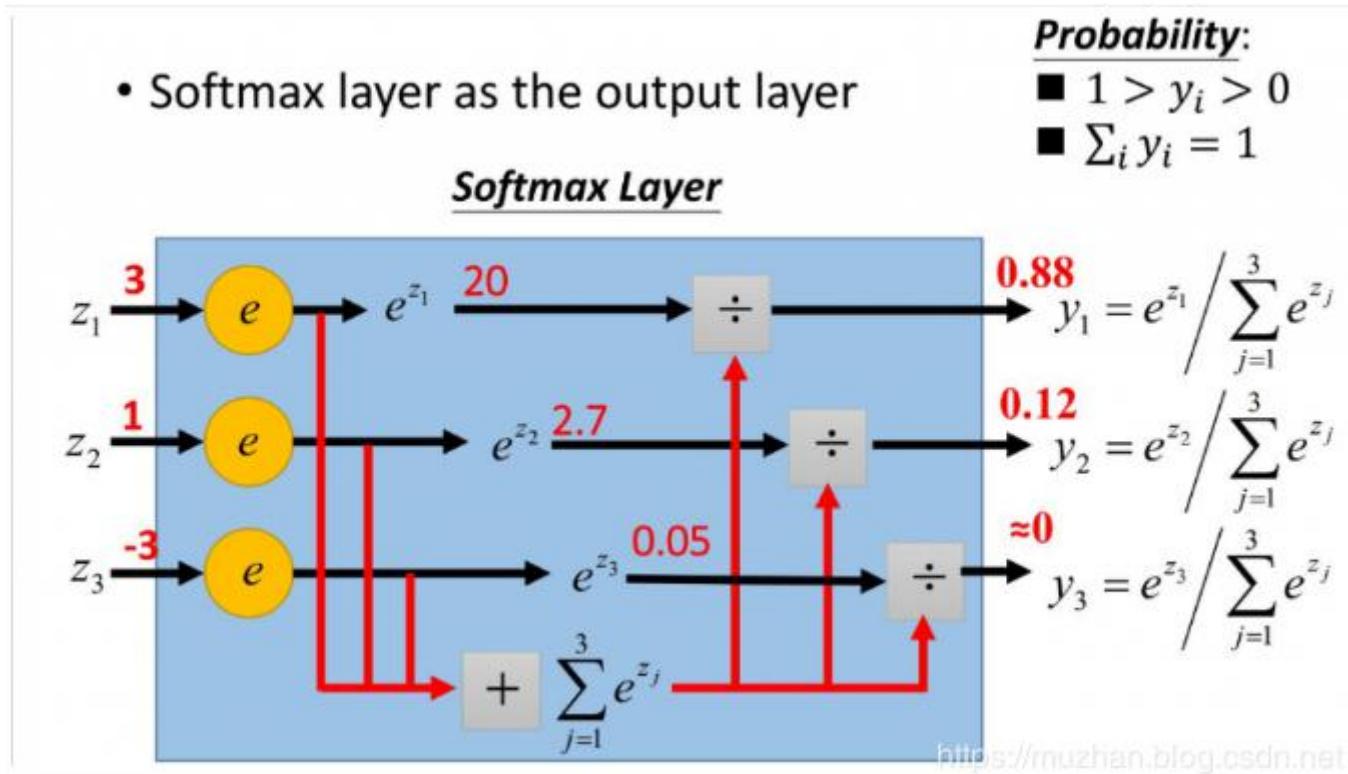


卷积神经网络 (CNN)

- 全连接层：连接所有的特征，将输出值送给分类器（如softmax分类器），最终得出识别结果。

softmax分类器

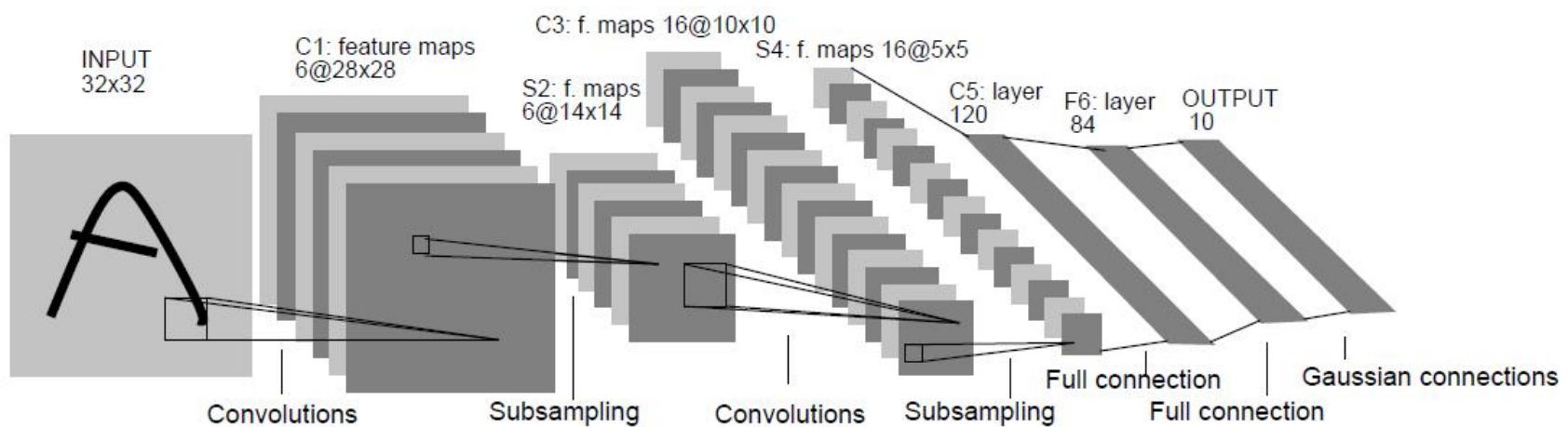
$$\text{softmax}(x_0) = \frac{e^{x_0}}{e^{x_0} + e^{x_1} + e^{x_2}}$$



softmax对神经元的输出信号进行加工，输出为分类的概率值。

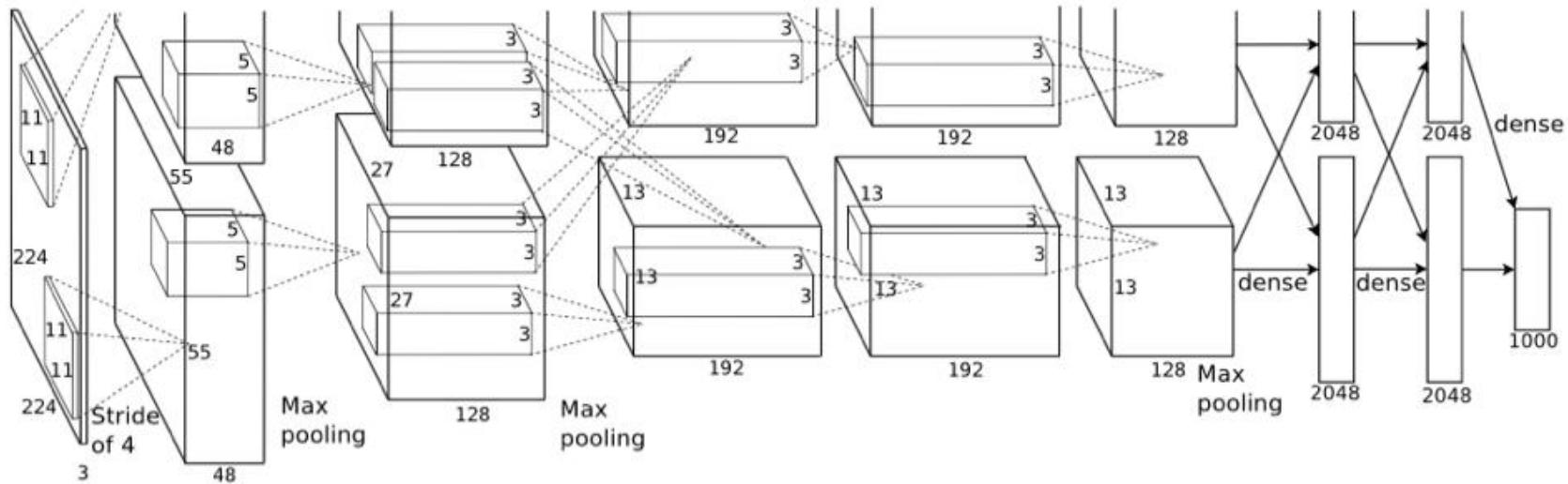
常见网络模型

● LeNet



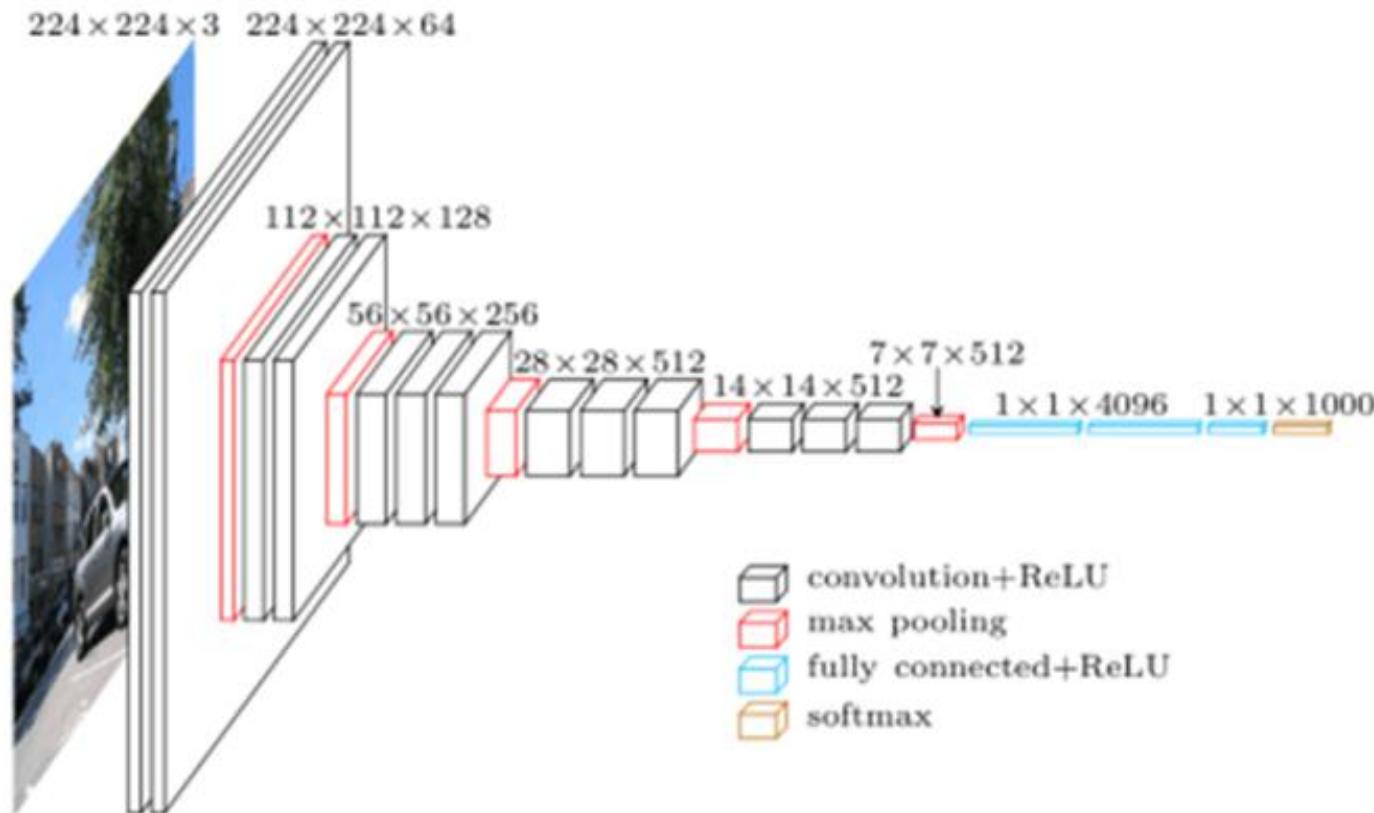
常见网络模型

● AlexNet



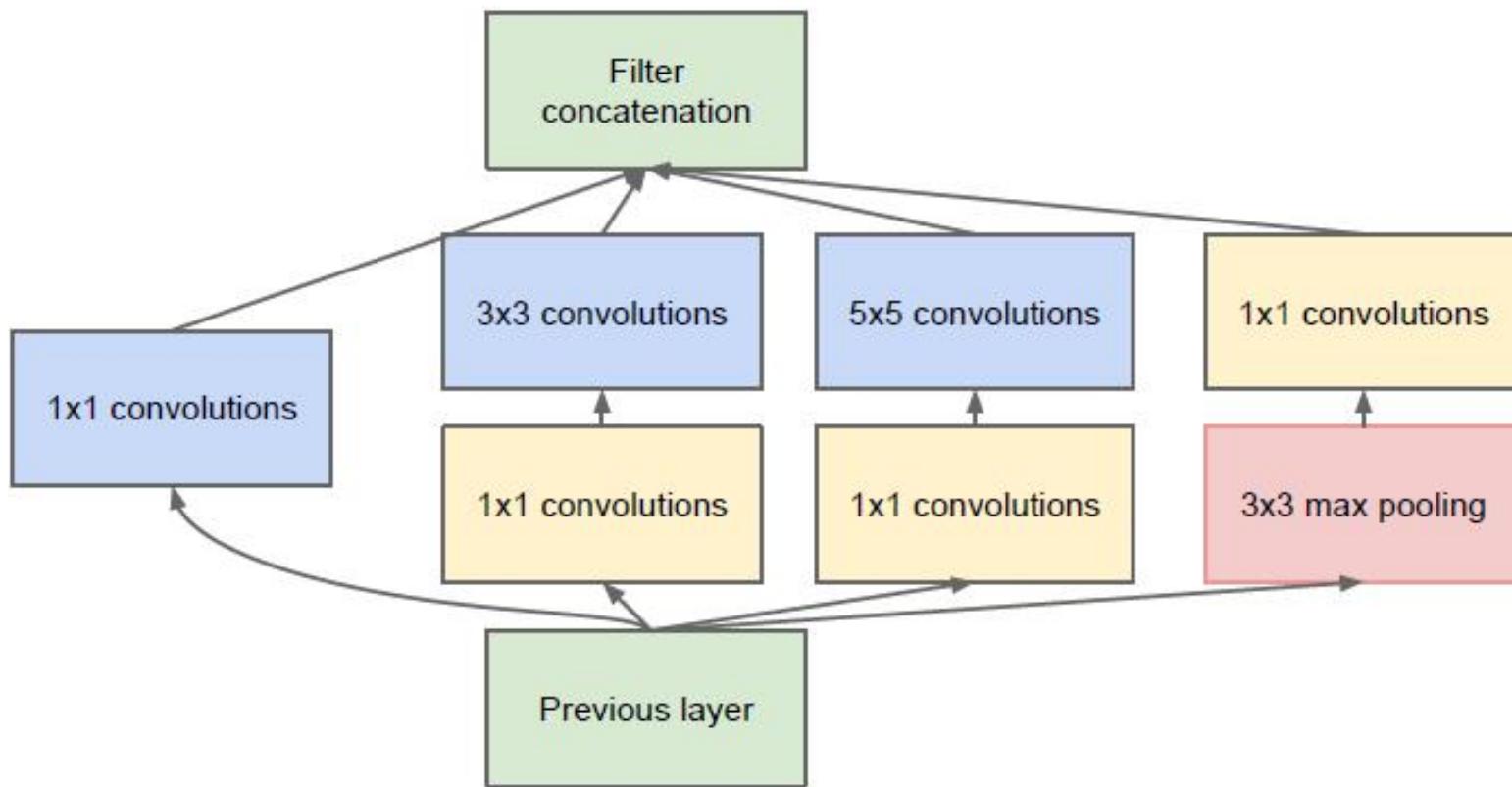
常见网络模型

● VGG16



常见网络模型

- GoogleNet (InceptionV4)



常见网络模型



CNN模型比
较.doc

● 比较

模型名	AlexNet	VGG	GoogLeNet	ResNet
初入江湖	2012	2014	2014	2015
层数	8	19	22	152
Top-5错误	16.4%	7.3%	6.7%	3.57%
Data Augmentation	+	+	+	+
Inception(NIN)	-	-	+	-
卷积层数	5	16	21	151
卷积核大小	11,5,3	3	7,1,3,5	7,1,3,5
全连接层数	3	3	1	1
全连接层大小	4096,4096,1000	4096,4096,1000	1000	1000
Dropout	+	+	+	+
Local Response Normalization	+	-	+	-
Batch Normalization	-	-	-	+

其他深度学习算法

- 自动编码器 (AutoEncoder)
- 稀疏编码 (Sparse Coding)
- 限制玻尔兹曼机 (RBM)

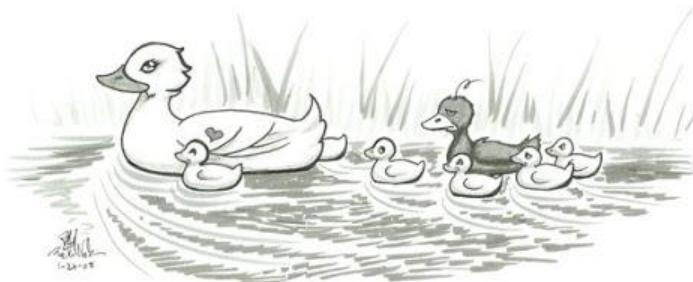
常用的定理

- 没有免费午餐定理（No Free Lunch Theorem, NFL）
 - 对于基于迭代的最优化算法，不存在某种算法对所有问题（有限的搜索空间内）都有效。如果一个算法对某些问题有效，那么它一定在另外一些问题上比纯随机搜索算法更差。



常用的定理

- 丑小鸭定理(Ugly Duckling Theorem)
 - 丑小鸭与白天鹅之间的区别和两只白天鹅之间的区别一样大.



常用的定理

- 奥卡姆剃刀原理(Occam's Razor)
 - 如无必要，勿增实体



深度学习的特点

深度学习常用算法介绍

深度学习常用框架介绍

开源框架概述

- 深度学习研究的热潮持续高涨，各种开源深度学习框架也层出不穷，其中包括TensorFlow、Caffe、Keras、CNTK、Torch7、MXNet、Leaf、Theano、DeepLearning4、Lasagne、Neon等等。下图是各个开源框架在GitHub上的数据统计（2017年初）。

框架	机构	支持语言	Stars	Forks	Contributors
TensorFlow	Google	Python/C++/Go/...	41628	19339	568
Caffe	BVLC	C++/Python	14956	9282	221
Keras	fchollet	Python	10727	3575	322
CNTK	Microsoft	C++	9063	2144	100
MXNet	DMLC	Python/C++/R/...	7393	2745	241
Torch7	Facebook	Lua	6111	1784	113
Theano	U. Montreal	Python	5352	1868	271
Deeplearning4J	Deeplearning4J	Java/Scala	5053	1927	101
Leaf	AutumnAI	Rust	4562	216	14
Lasagne	Lasagne	Python	2749	761	55
Neon	NervanaSystems	Python	2633	573	52

开源框架概述

- Google、Microsoft、Facebook等巨头都参与了这场深度学习框架大战，此外，还有毕业于伯克利大学的贾扬清主导开发的Caffe，蒙特利尔大学Lisa Lab团队开发的Theano，以及其他个人或商业组织贡献的框架。下表是主流深度学习框架在各个维度的评分。

	模型设计	接 口	部 署	性 能	架构设计	总体评分
TensorFlow	80	80	90	90	100	88
Caffe	60	60	90	80	70	72

续表

	模型设计	接 口	部 署	性 能	架构设计	总体评分
CNTK	50	50	70	100	60	66
Theano	80	70	40	50	50	58
Torch	90	70	60	70	90	76
MXNet	70	100	80	80	90	84
DeepLearning4J	60	70	80	80	70	72

TensorFlow

- TensorFlow最初是由研究人员和Google Brain团队针对机器学习和深度神经网络进行研究所开发的，目前开源之后可以在几乎各种领域适用。
- TensorFlow灵活的架构可以部署在一个或多个CPU、GPU的台式以及服务器中，或者使用单一的API应用在移动设备中。



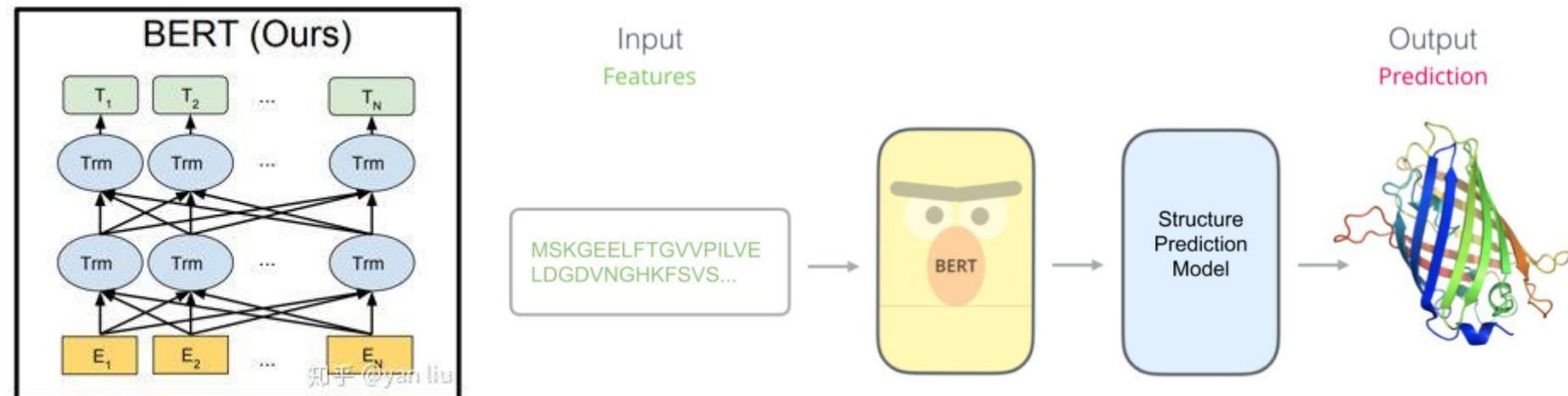
Torch

- Torch是一个有大量机器学习算法支持的科学计算框架，其诞生已经有十年之久，但是真正起势得益于Facebook开源了大量Torch的深度学习模块和扩展。Torch另外一个特殊之处是采用了编程语言Lua(该语言曾被用来开发视频游戏)。
- PyTorch是基于Torch的衍生，支持Python语言，实现了机器学习框架 Torch 在 Python 语言环境的执行。



Bert

- 使用Transformer的结构将已经走向瓶颈期的Word2Vec带向了一个新的方向，并再一次炒火了《Attention is All you Need》这篇论文；
- 11个NLP任务的精度大幅提升足以震惊整个深度学习领域；
- 无私的开源了多种语言的源码和模型，具有非常高的商业价值；
- 迁移学习又一次胜利，而且这次是在NLP领域的大胜，狂胜。



Caffe

- Caffe由加州大学伯克利的PHD贾扬清开发，全称Convolutional Architecture for Fast Feature Embedding，是一个清晰而高效的开源深度学习框架，目前由伯克利视觉学中心（Berkeley Vision and Learning Center, BVLC）进行维护。（贾扬清曾就职于MSRA、NEC、Google Brain，他也是TensorFlow的作者之一，目前任职于Facebook FAIR实验室。）
- Caffe2脸书 (Facebook) 出品，为生产环境设计，提供在各种平台（包括移动设备）的运行。



Theano

- 2008年诞生于蒙特利尔理工学院,Theano派生出了大量深度学习Python软件包，最著名的包括Blocks和Keras。Theano的核心是一个数学表达式的编译器，它知道如何获取你的结构。并使之成为一个使用numpy、高效本地库的高效代码，如BLAS和本地代码（C++）在CPU或GPU上尽可能快地运行。它是为深度学习中处理大型神经网络算法所需的计算而专门设计的，是这类库的首创之一（发展始于2007年），被认为是深度学习研究和开发的行业标准。



Deeplearning4j

- Deeplearning4j是 “for Java” 的深度学习框架，也是首个商用级别的深度学习开源库。Deeplearning4j由创业公司Skymind于2014年6月发布，使用Deeplearning4j的不乏埃森哲、雪弗兰、博斯咨询和IBM等明星企业。DeepLearning4j是一个面向生产环境和商业应用的高成熟度深度学习开源库，可与Hadoop和Spark集成，即插即用，方便开发者在APP中快速集成深度学习功能。

MXNet

- 出自CXXNet、Minerva、Purine等项目的开发者之手，主要用C++编写。MXNet 强调提高内存使用的效率，甚至能在智能手机上运行诸如图像识别等任务。

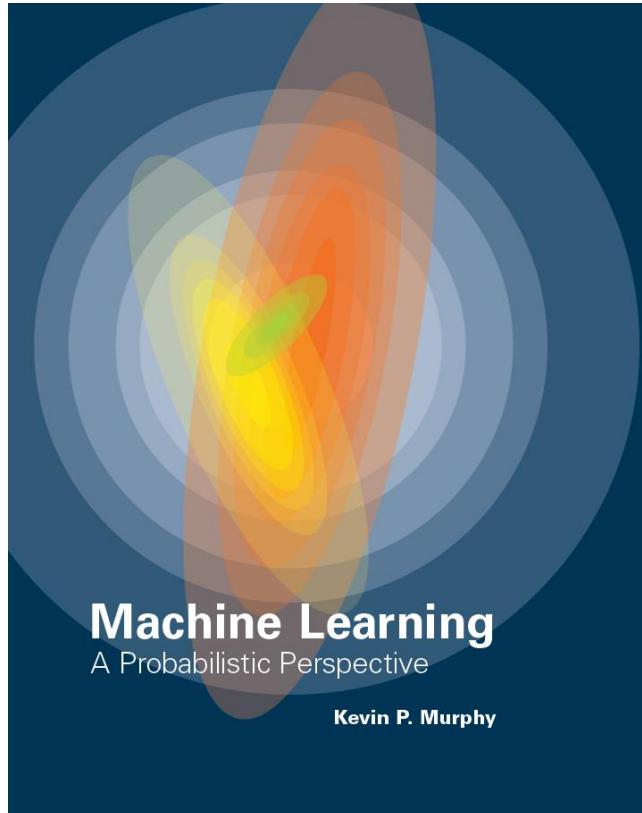
mxnet

CNTK

- CNTK (Computational Network Toolkit) 是微软研究院 (MSR) 开源的深度学习框架。它最早由start the deep learning craze的演讲人创建，目前已经发展成一个通用的、跨平台的深度学习系统，在语音识别领域的使用尤其广泛。

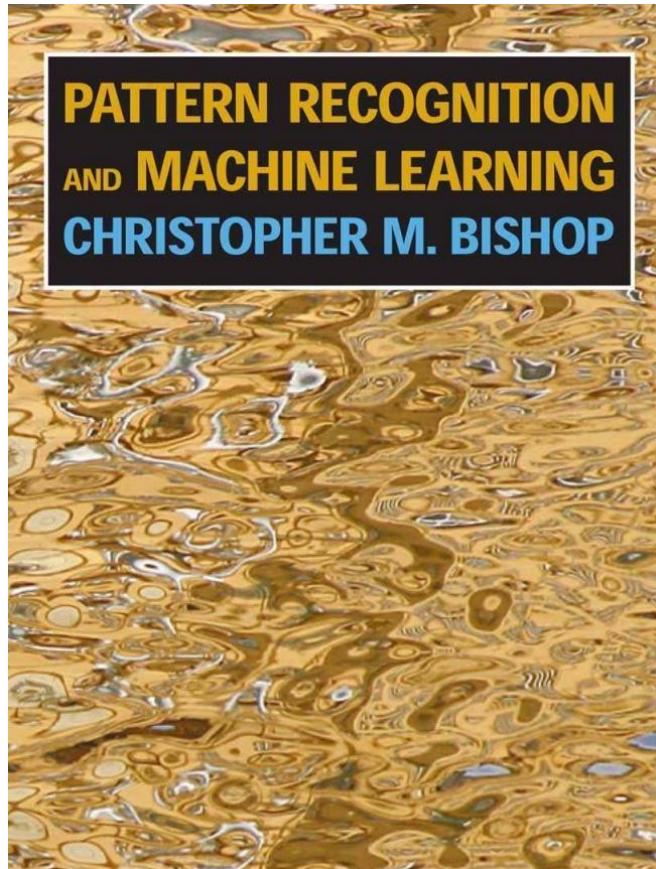


References

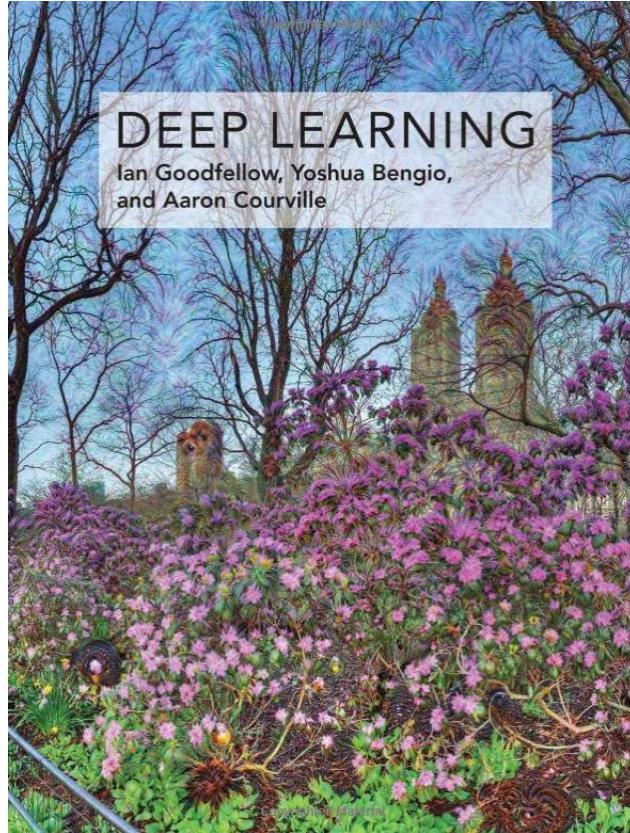


<https://github.com/probml/pyprobml>

References



References



<https://github.com/exacity/deeplearningbook-chinese>

Part III

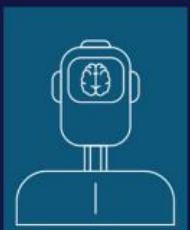
生物大数据的深度学习方法

AI in a nutshell

LEVELS OF ARTIFICIAL INTELLIGENCE



ARTIFICIAL
NARROW
INTELLIGENCE



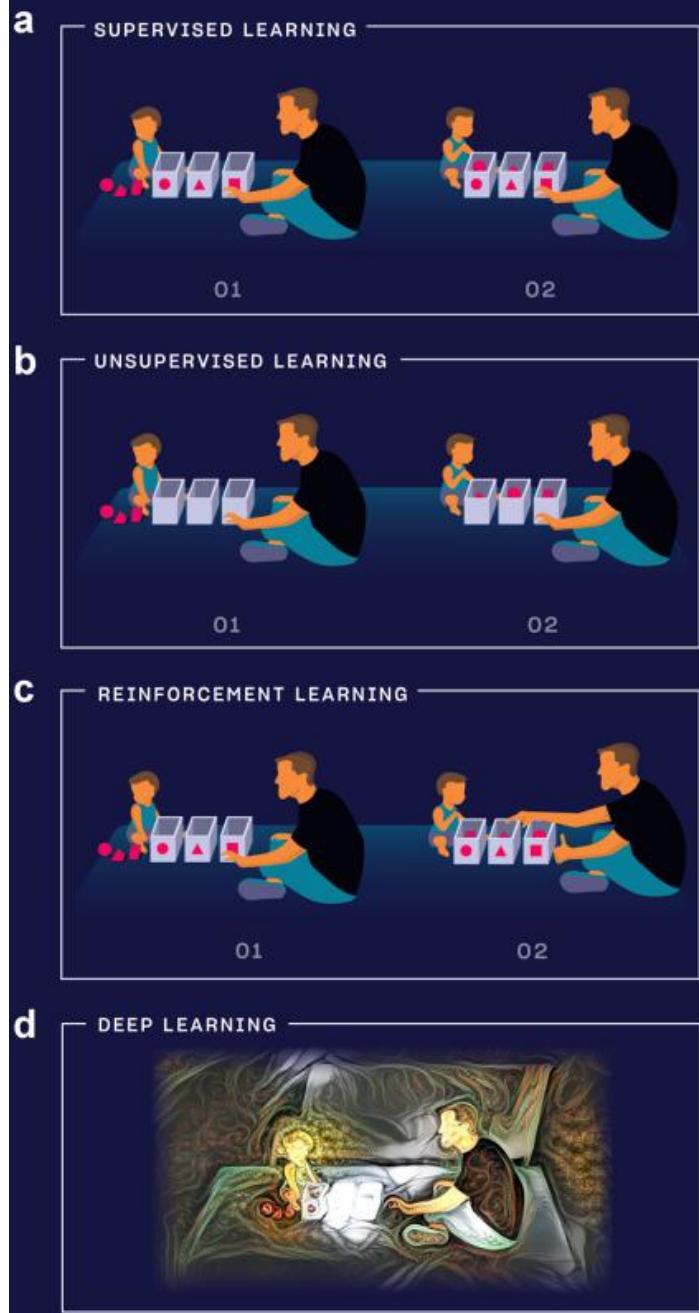
ARTIFICIAL
GENERAL
INTELLIGENCE



ARTIFICIAL
SUPER
INTELLIGENCE

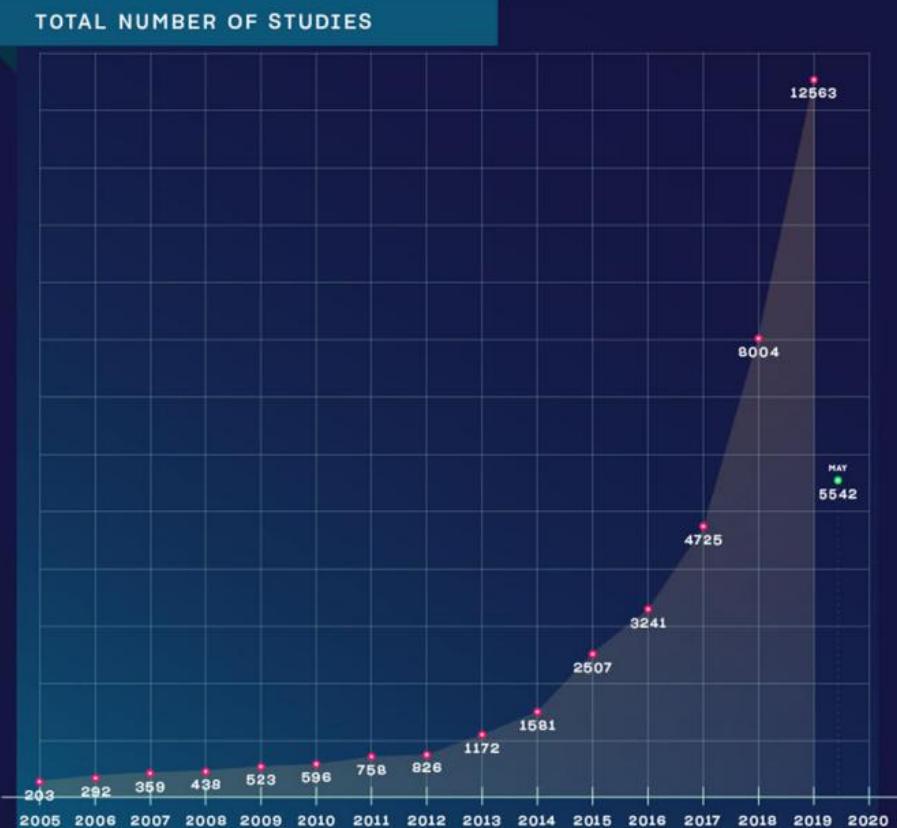


● : THE IDEAL A.I.



Machine Learning in biology

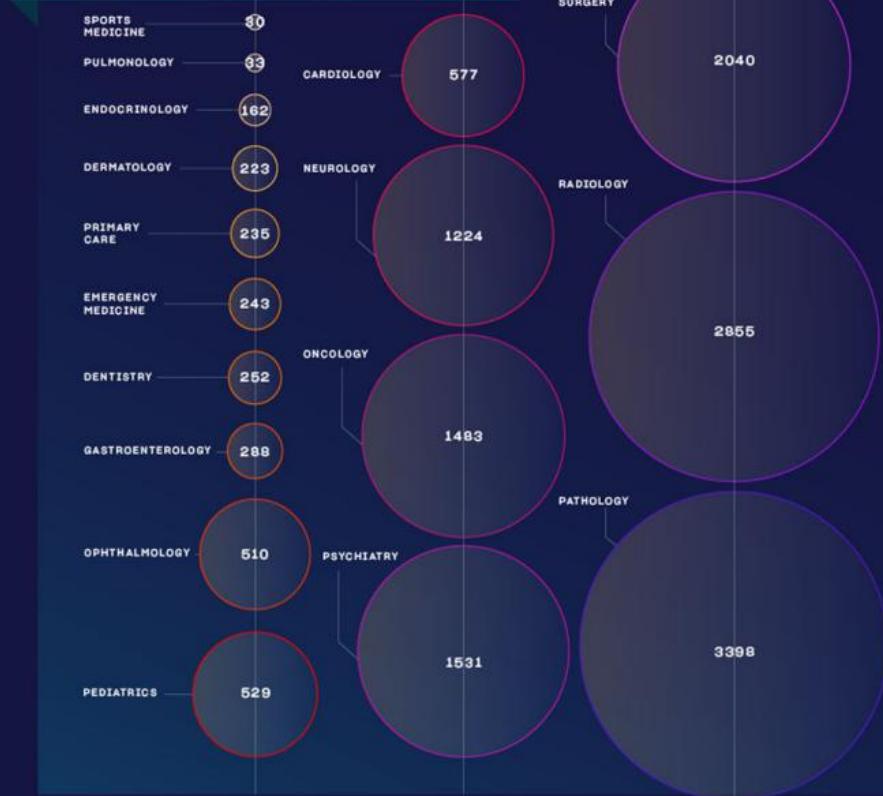
a



b

MACHINE AND DEEP LEARNING STUDIES ON PUBMED.COM

STUDIES PER SPECIALTY



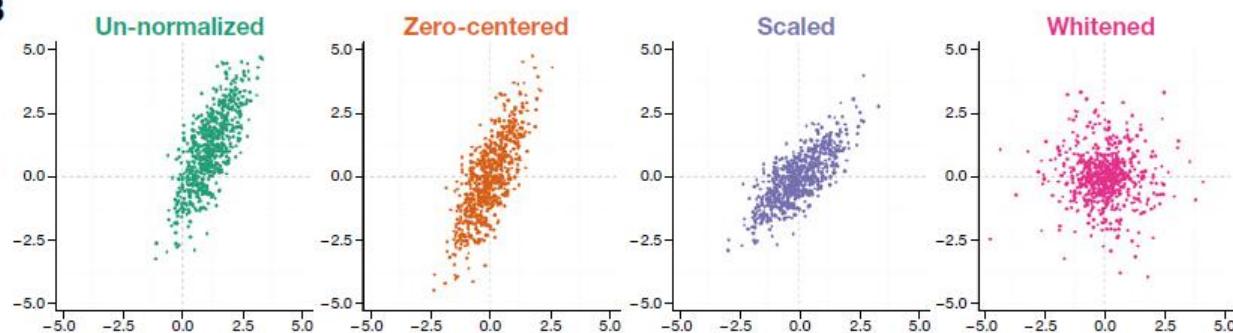
Feature Selection

Collection → Partition → Normalization
Data label

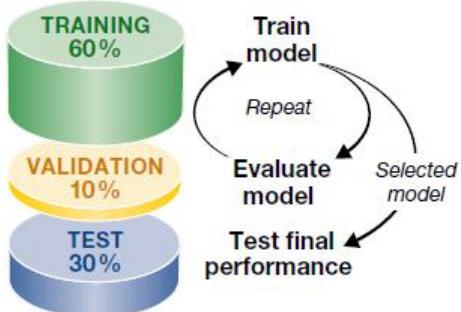
A

A	G	T	C	A	G	C
1	0	0	0	0	1	0
0	1	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
1	0	0	0	0	1	0
0	1	0	0	0	0	1

B



C



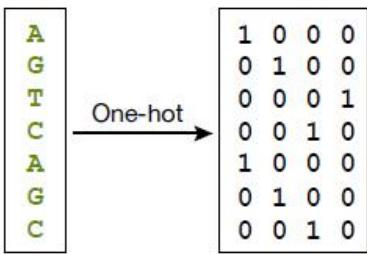
(A) Data One-hot Coding

(B) Different data normalization methods

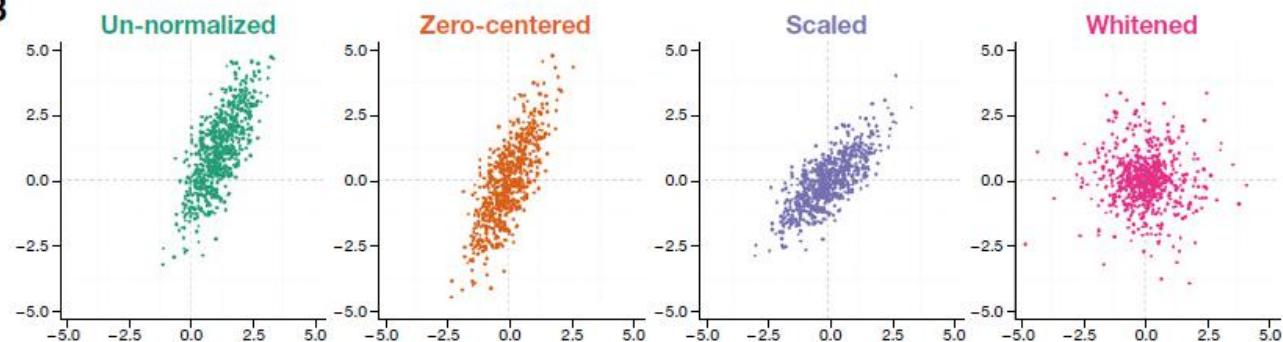
(C) Training data, validation data (adjust model structure), Test data

Data normalization

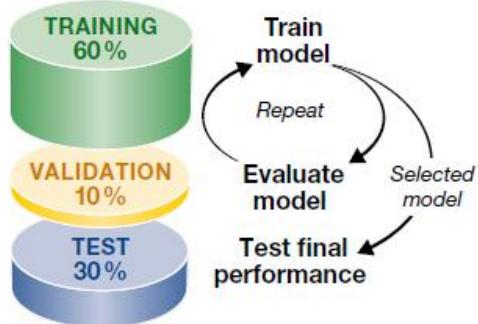
A



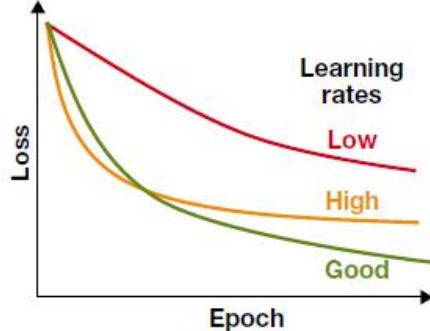
B



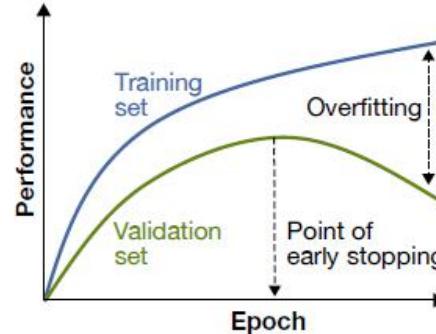
C



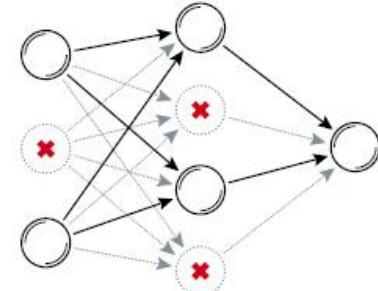
D



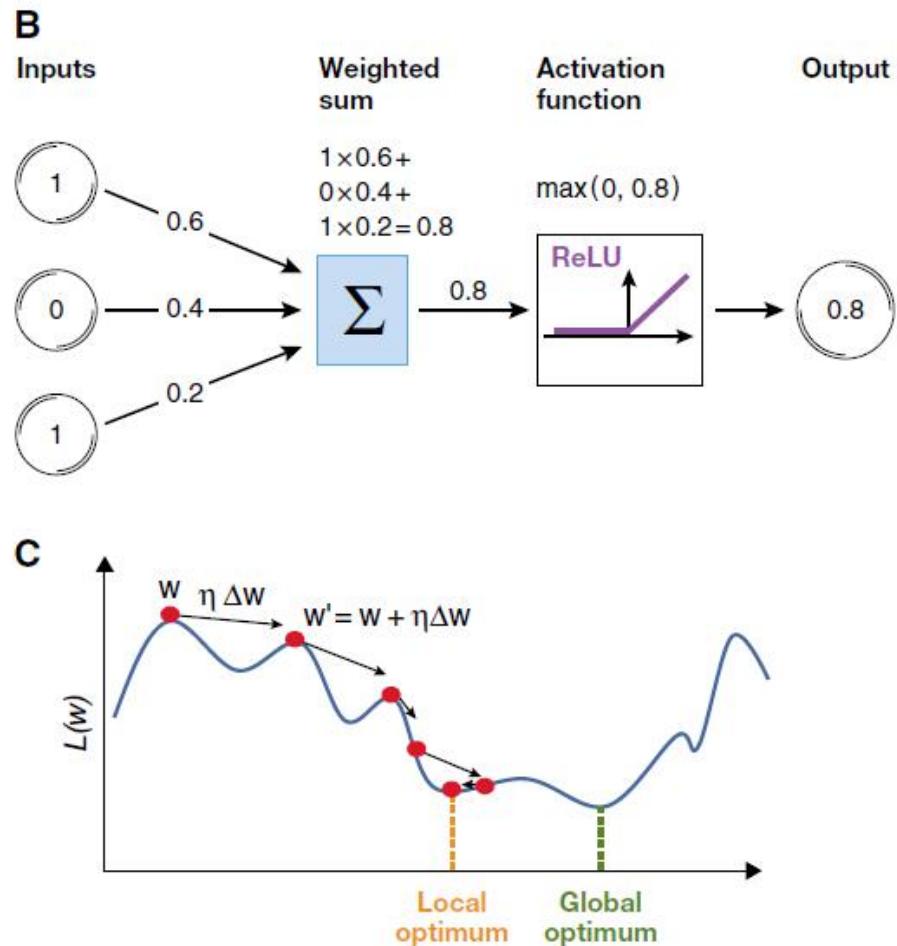
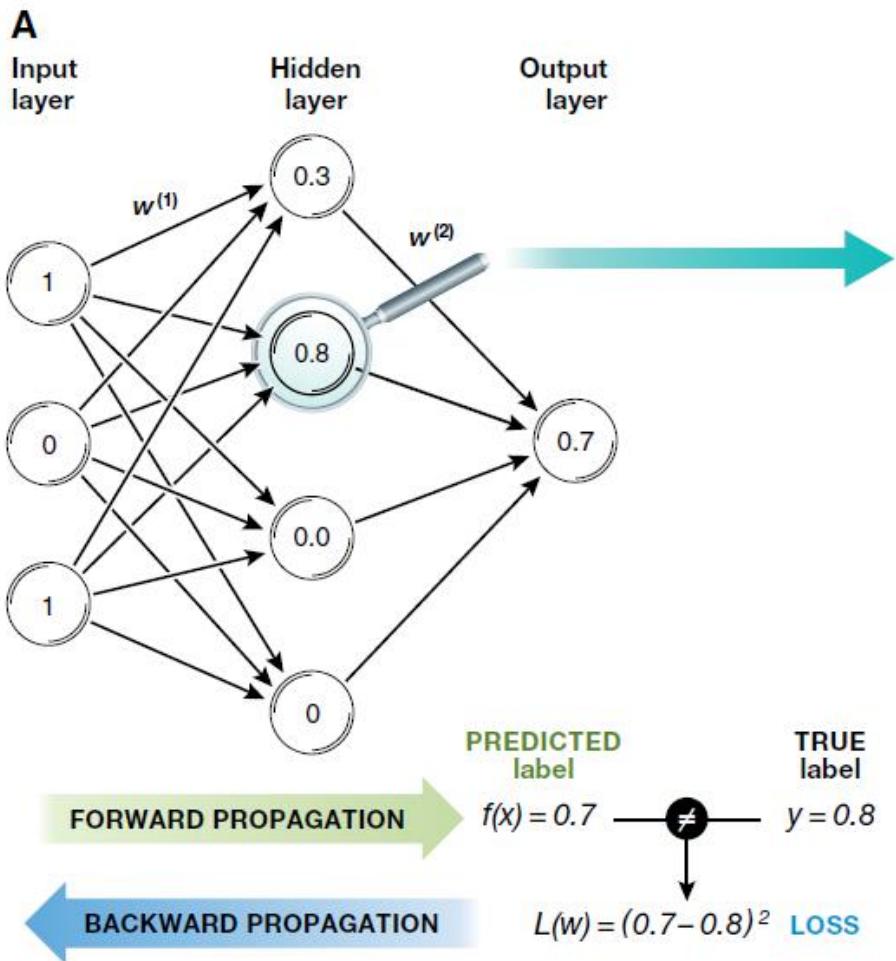
E



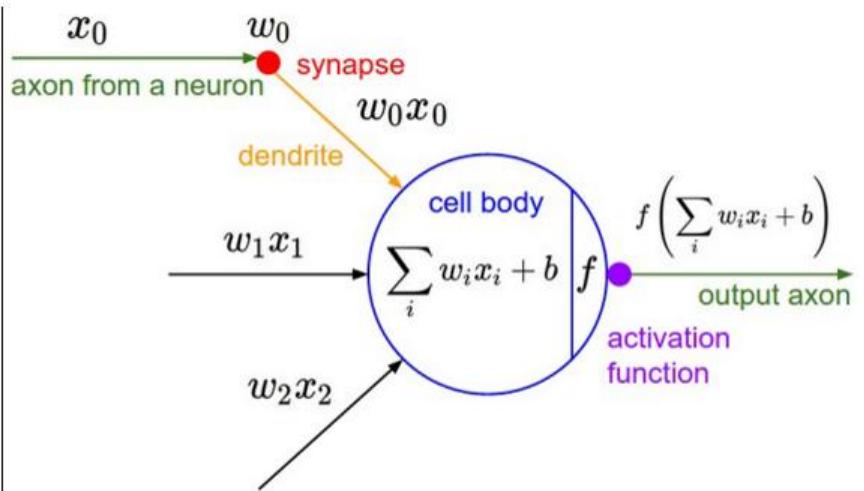
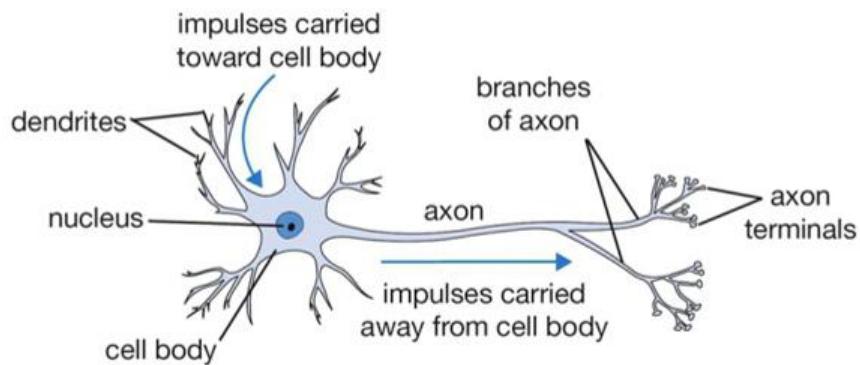
F



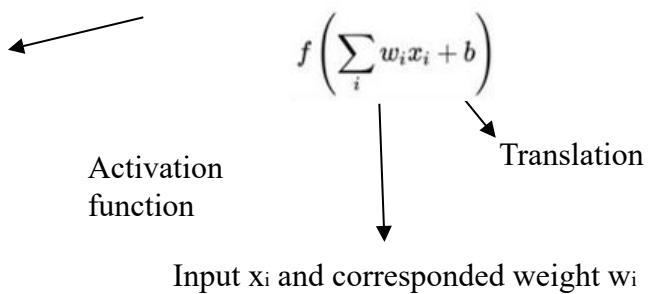
Artificial Neural Network



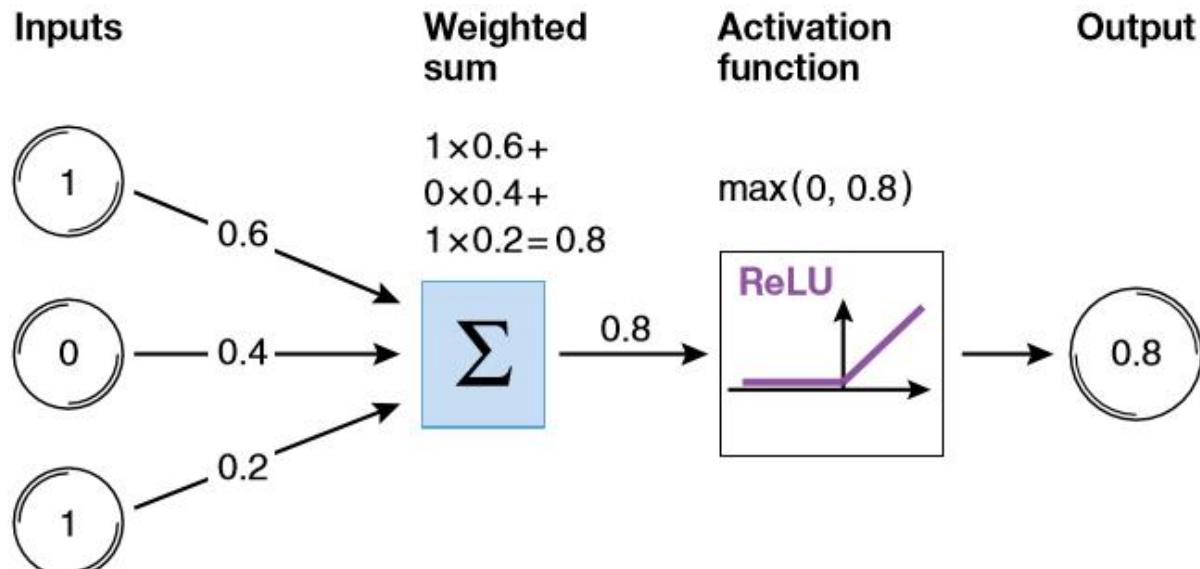
Model Construction



A Cartoon drawing of a biological neuron(left) and its mathematical model(right)

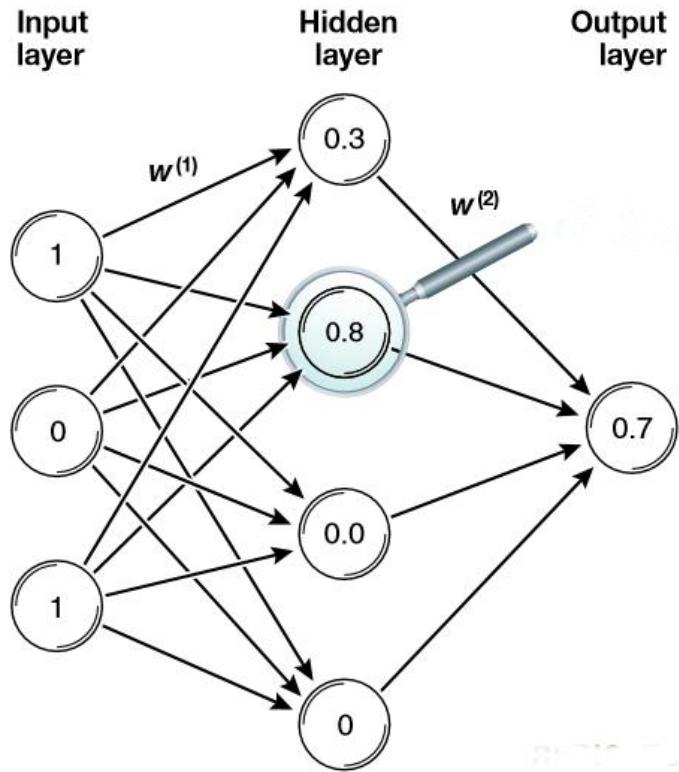


Model Construction



An example for a “transmission of neural signal”

Model Construction



Model construction:

Input label: 1, 0, 1

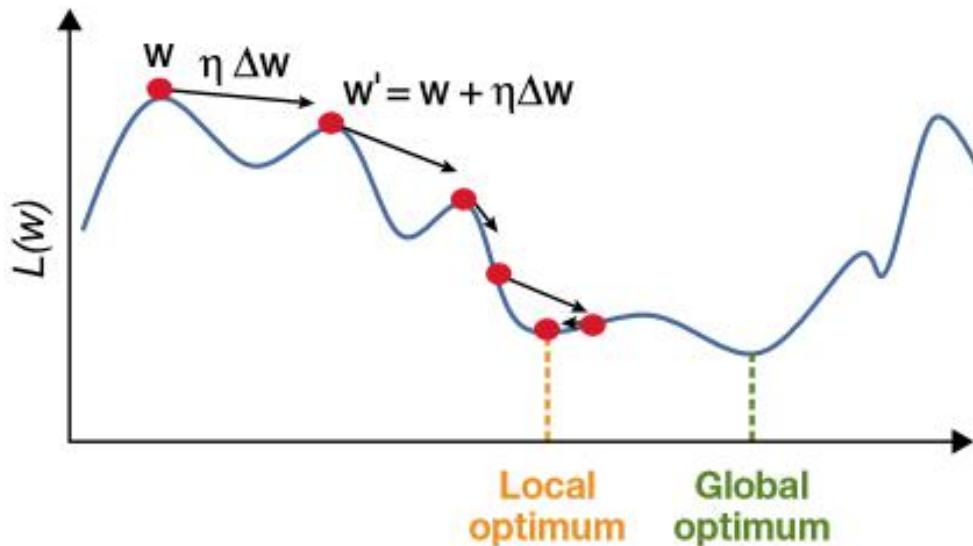
Predicted label: 0.7

Ture data label: 0.8

Neuron network construction
process

Pick the best model

loss function: $L(w) = (\text{predicted label} - \text{true label})^2$

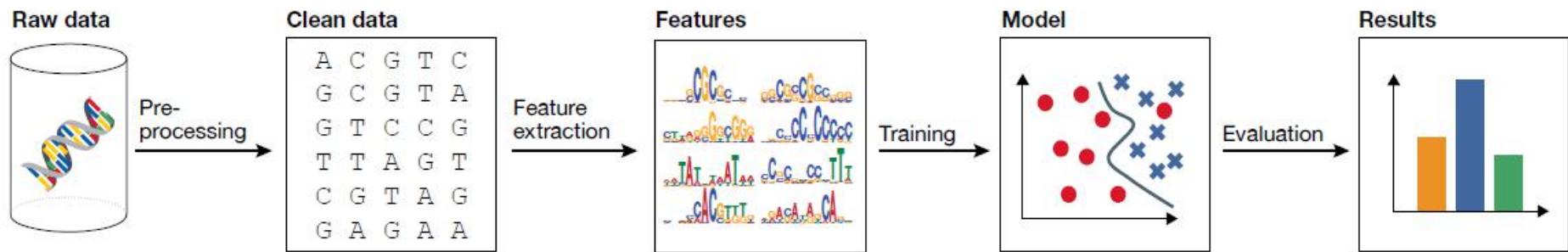


Many parameters are adjusted to find the global minimum
then update the neuron weight

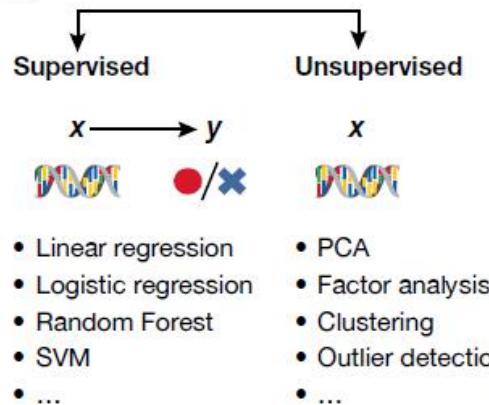
Predicted label: 0.8 → True label: 0.8

Machine learning and representation learning

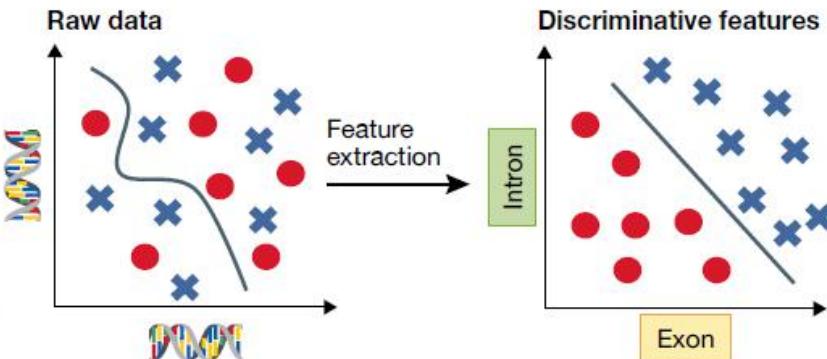
A



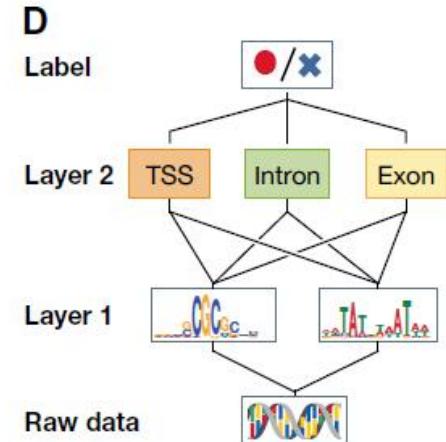
B



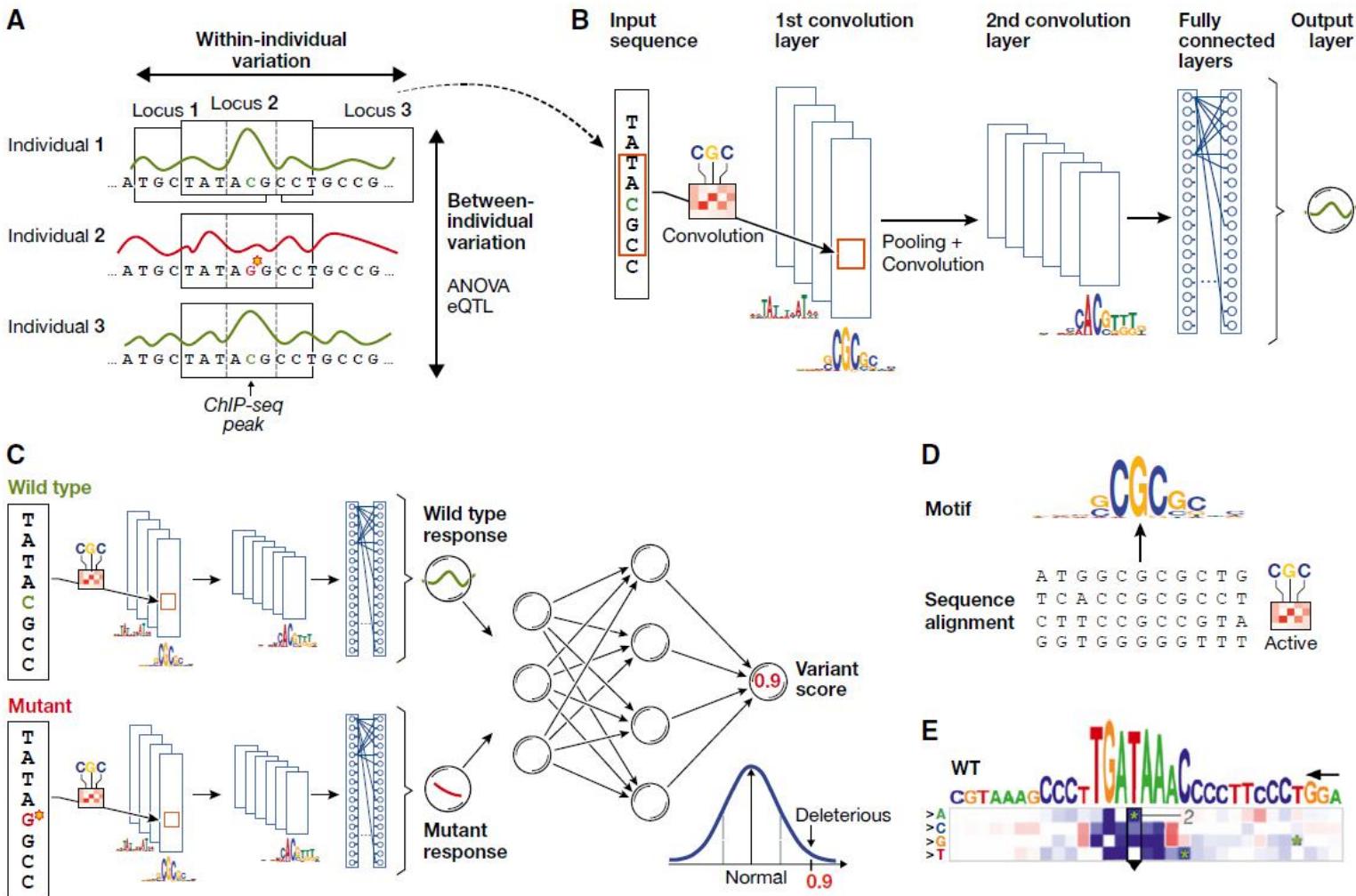
C



D



Neural Network for DNA sequences



Adapted from "Deep learning for computational biology. Molecular Systems Biology, 2016."

Part IV

生物大数据深度学习的应用案例

INPUT

PROCESSING

APPLICATION

DNA sequences

RNA

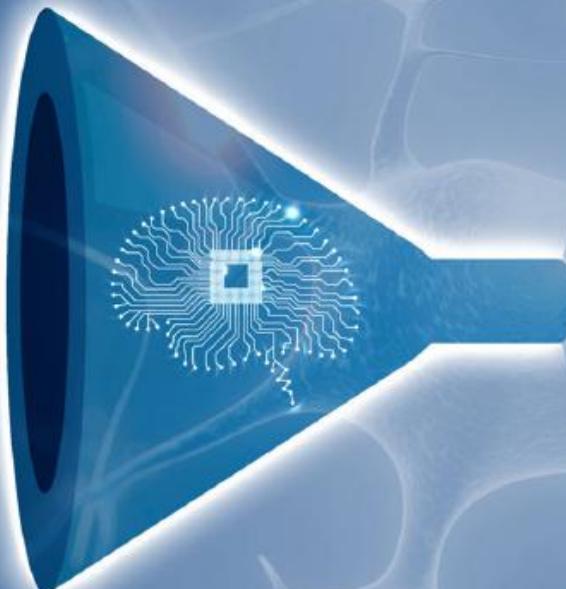
MicroRNA

Gene expressions

Gene alleles

Molecule compounds

Protein structures



Disease
stratification



Cancer
diagnostic



Gene
variations



Drug
design

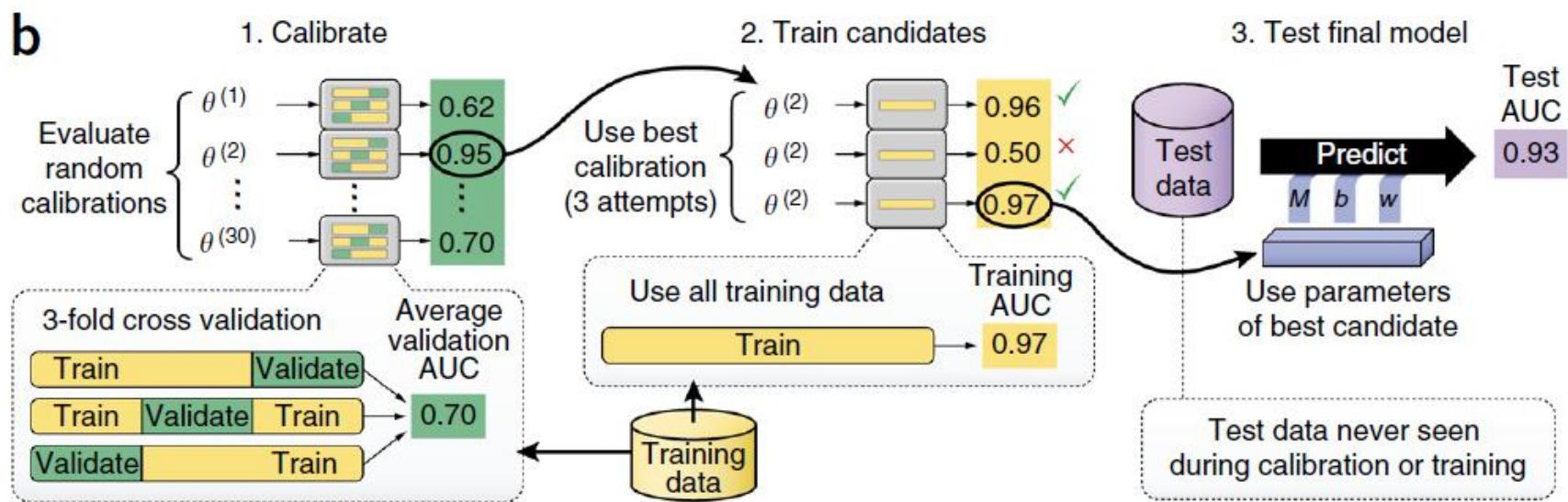
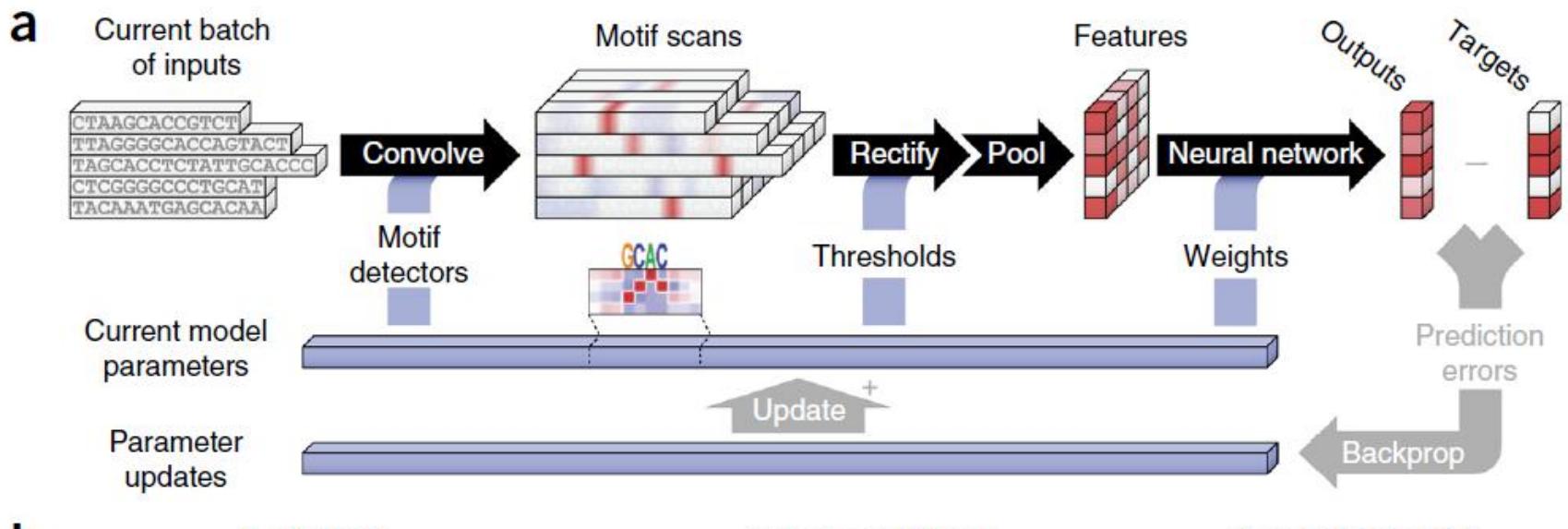


RNA
Splicing



Protein structure
and interaction

DNA- and RNA-binding protein prediction

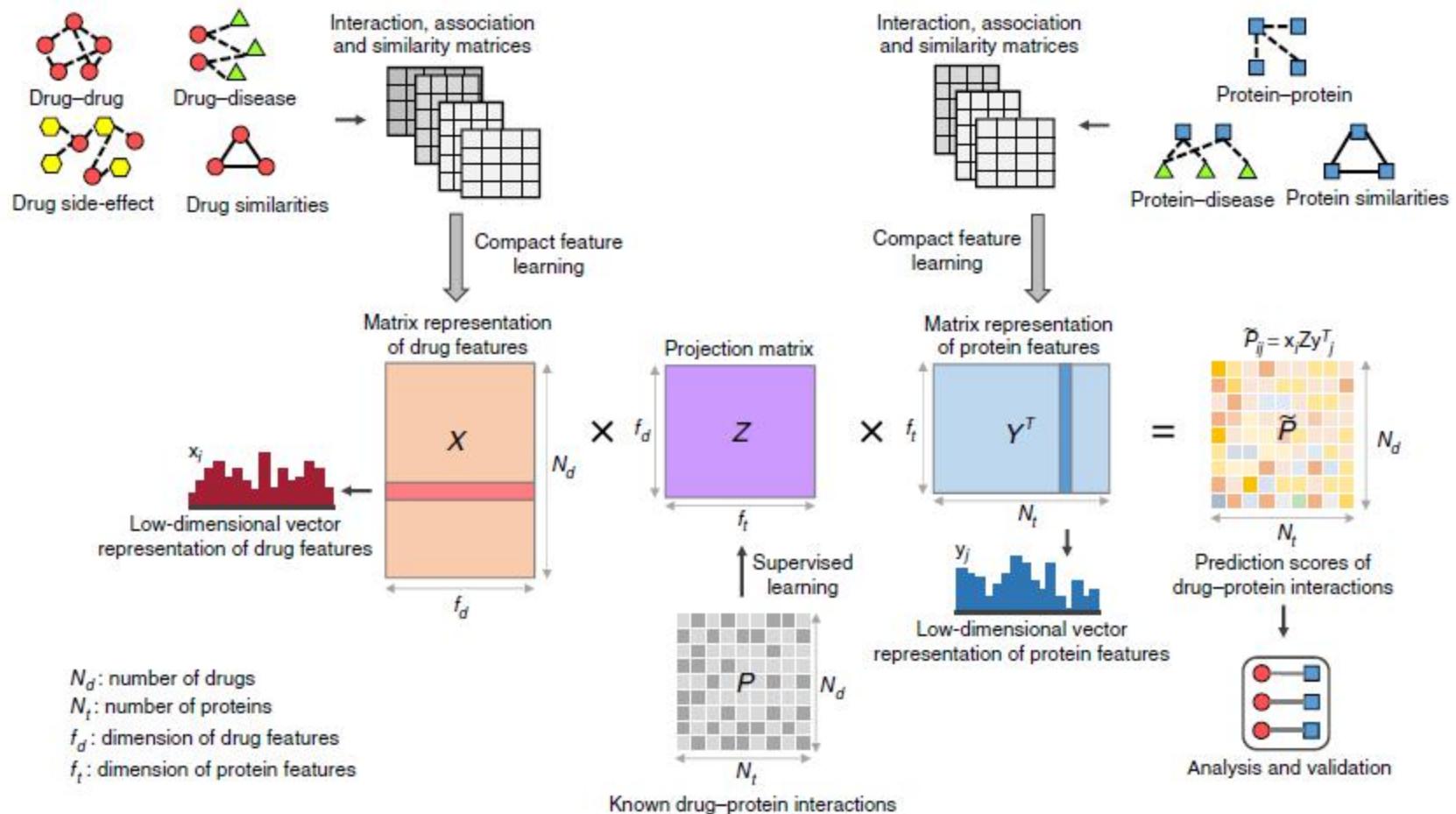


Adapted from "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nature Biotechnology, 2015."

DeepBind总结

- **CNN**算法自动学习了模版**M**的参数，这个模版就是能够决定蛋白质是否**Binding**的特征！
- 而**CNN**网络就是在前端自动总结出可能的模版，在后端根据这些模版在序列中出现与否而进行判断；
- **CNN**算法巧妙地将统计归纳和**MLP**的学习功能相结合，减少了**MLP**中要学习的参数；

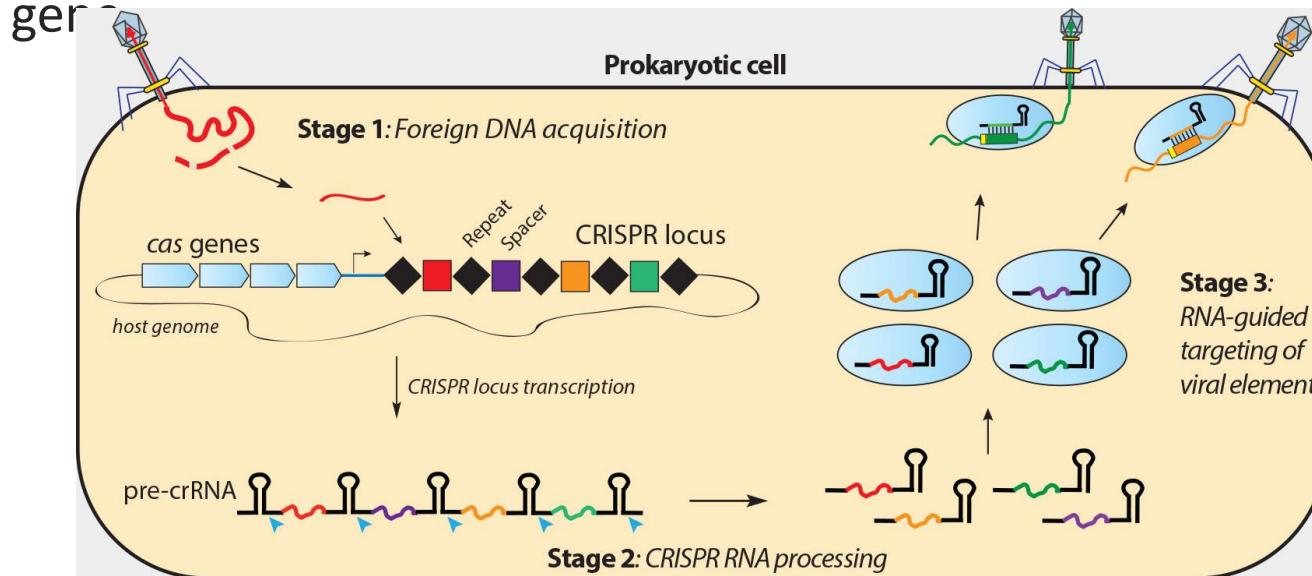
Drug-target interaction prediction



Adapted from “A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. Nature Communications, 2017”

What is Cas?

- Cas (CRISPR-associated system) genes
bacteria immune system: cleave the foreign
gen



CRISPR-Cas system workflow: foreign DNA acquisition → CRISPR RNA processing → RNA-guided targeting of viral element

The existing method for Cas genes prediction

- The existing method for Cas genes prediction

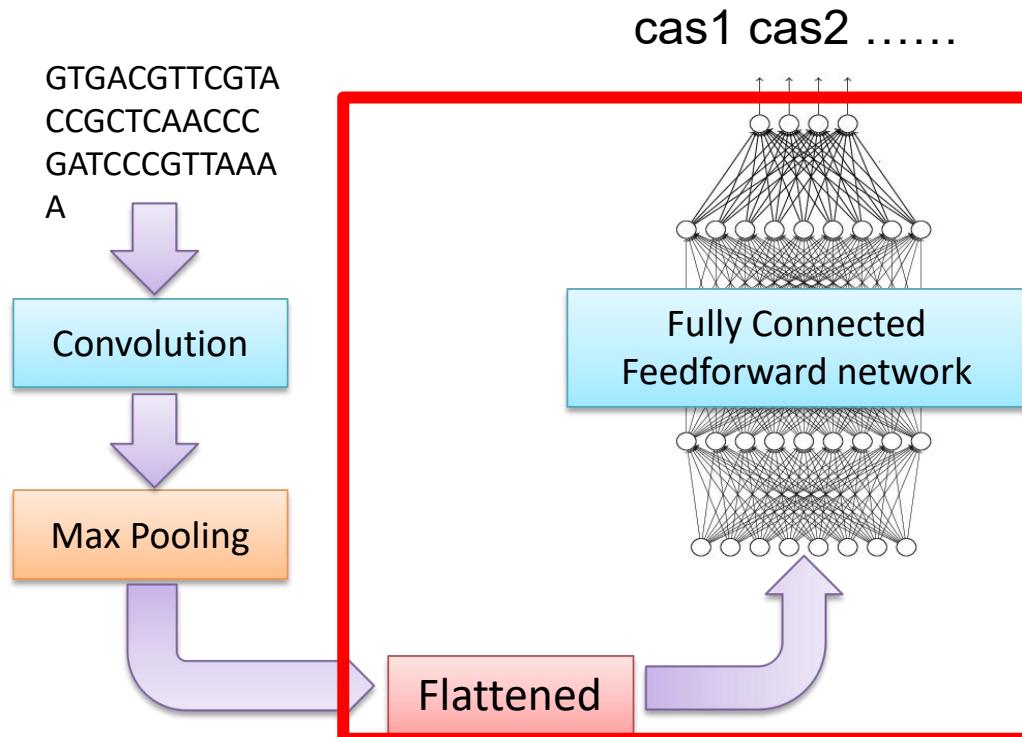
Existing tools	Description
HMMCAS	A web tool for Cas protein identification and domain annotation
CRISPRone	A website tool to predict CRISPR arrays of repeat-spacer units, and cas genes
BLAST	A website tool to find regions of local similarity between sequences
CD-search	A tool allows the conserved domain annotation for large sets of protein queries
Custom HMMs	HMMs deposited in the TIGRFAMs and Pfam protein family databases for known protein families

- Limitations

- Scan for known Cas proteins.
- Identify or predict based on the “best hits” of homology searches against existing databases.
- Work well for detecting known and highly conserved types of Cas genes but may fail to detect novel Cas genes.

The whole neural network to predict Cas genes

Mainly based on Python Tensorflow package



Data preparation

10 core family, 75 family, 7500 Cas genes in total

Core family	Structural features
Cas1	N-terminal β stranded domain and catalytic C-terminal α-helical domain
Cas2	RuvC-like nuclease domains
Cas3	Helicase and HD domain
Cas4	RecB-like nuclease homolog with three-cysteine C-terminal cluster
Cas5	RRM (ferredoxin) fold, RAMP superfamily
Cas6	Double RRM (ferredoxin) fold, RAMP superfamily
Cas7	RRM (ferredoxin) fold with subdomains, RAMP superfamily
Cas8	Subunit of Cascade complex, involved in PAM recognition
Cas9	RuvC-like (RNase H fold) and HNH nuclease domains
Cas10	Two domains homologous to Palm domain polymerases and cyclases

Data preparation

one-hot encoding, the 4 nucleotides as binary vectors A=[1, 0, 0, 0], T=[0, 1, 0, 0], G=[0, 0, 1, 0] and C=[0, 0, 0, 1].

- Raw data format:

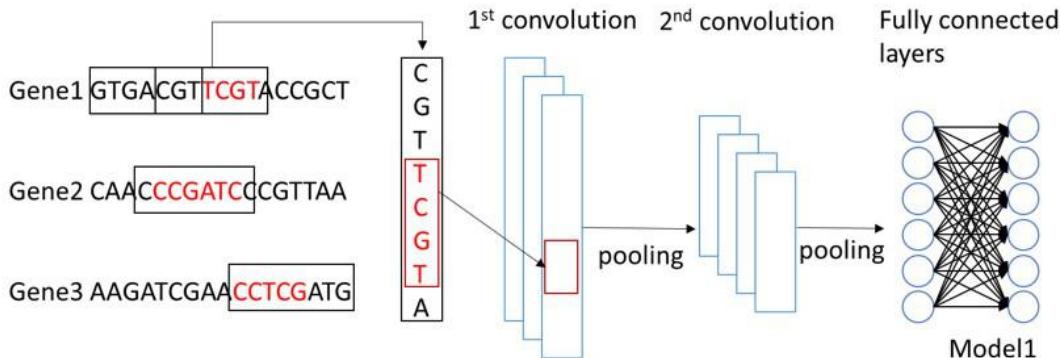
>CP001154.1_cds_AC074068.1_1072
GTGGAGCTGCTGGTCGCCCGGACCTCAAAACCATCCTGCATTCCGGCGGGCCAGCCTTATGCCCTGGAATACTGCCG
>CP010338.1_cds_AMC82843.1_3024
ATGCGTACCGAAGGCCATGAACCCGCAACTCATCGAGGCCATAATCGGCTGCCTTGCACGACATTGGCAAACCGTCCA
>CP002567.1_cds_AFI63574.1_1345
ATGAGCAAGATTCCCGGAGTAAACACCAGAATTGAAATCGTGGTGCAGGATAGGCTGACGTTCTGTATGC
>CP018217.1_cds_APG74590.1_777
ATGTCATGGAGATATGTTGCACTAACCTGTAGGATAACGTATAACCAACAATAGTATAGTTGTTCAAACCTTCGA
>CP012396.1_cds_AOC85297.1_899
GTGACGTTCTGCTACCGCTCAACCCGATCCGTTAAAAGATCGAACCTCGATGATCTCCCTCAGTAGCGTCAAATGACGT



- After change to one_hot:

Sequence label, save in an individual file

Convolution Neural Networks model



- Design a neural network that will classify cas proteins.
- Given a cas gene sequence, we want our neural net to let us know which cas gene that it belongs to.
- The sequence is made up of A, T, C, G, and fasta format:
- the neural net will have **xxx inputs**, each one representing a particular sequence and a hidden layer consisting of a number of neurons (more on this later) all feeding their output into just **one neuron in the output layer**.

A convolutional layer

A CNN is a neural network with some convolutional layers (and some other layers). A convolutional layer has a number of filters that does convolutional operation.

These are the network parameters to be learned

1	0	0	0
0	1	0	0
0	0	0	1
1	0	0	0
:	:	:	:
0	0	1	0

80 x 4

1	-1	-1	-1
-1	1	-1	-1
-1	-1	-1	1
1	-1	-1	-1

Filter 1

-1	1	-1	-1
-1	-1	1	-1
-1	-1	-1	1
1	-1	-1	-1

Filter 2

.....

Each filter detects a small pattern (4 x 4).

A convolutional layer

stride=1

1	0	0	0
0	1	0	0
0	0	0	1
1	0	0	0
.	.	.	.
.	.	.	.
0	0	1	0

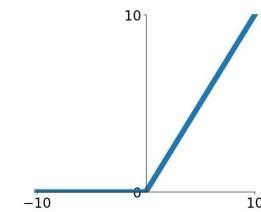
80 x 4

Dot
product

4

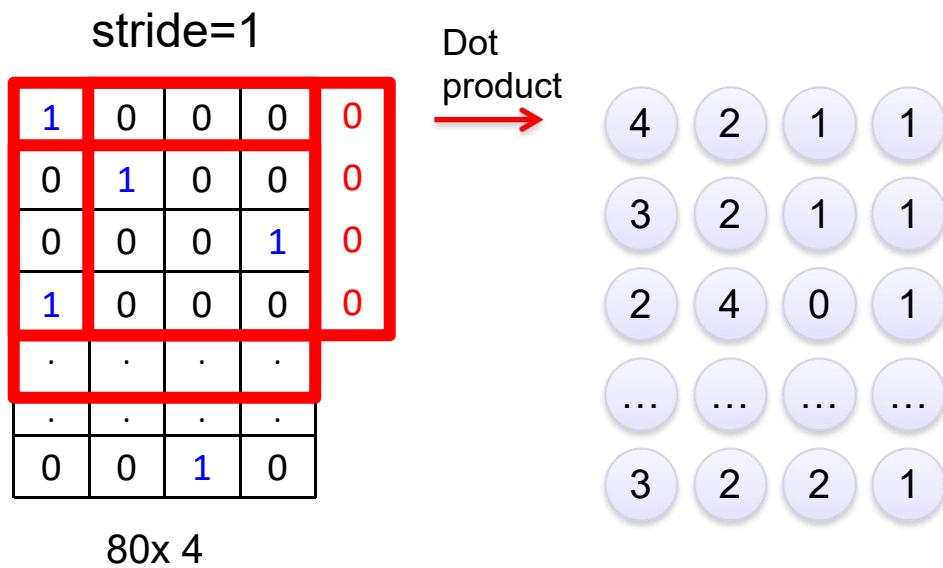
1	-1	-1	-1
-1	1	-1	-1
-1	-1	-1	1
1	-1	-1	-1

Filter 1



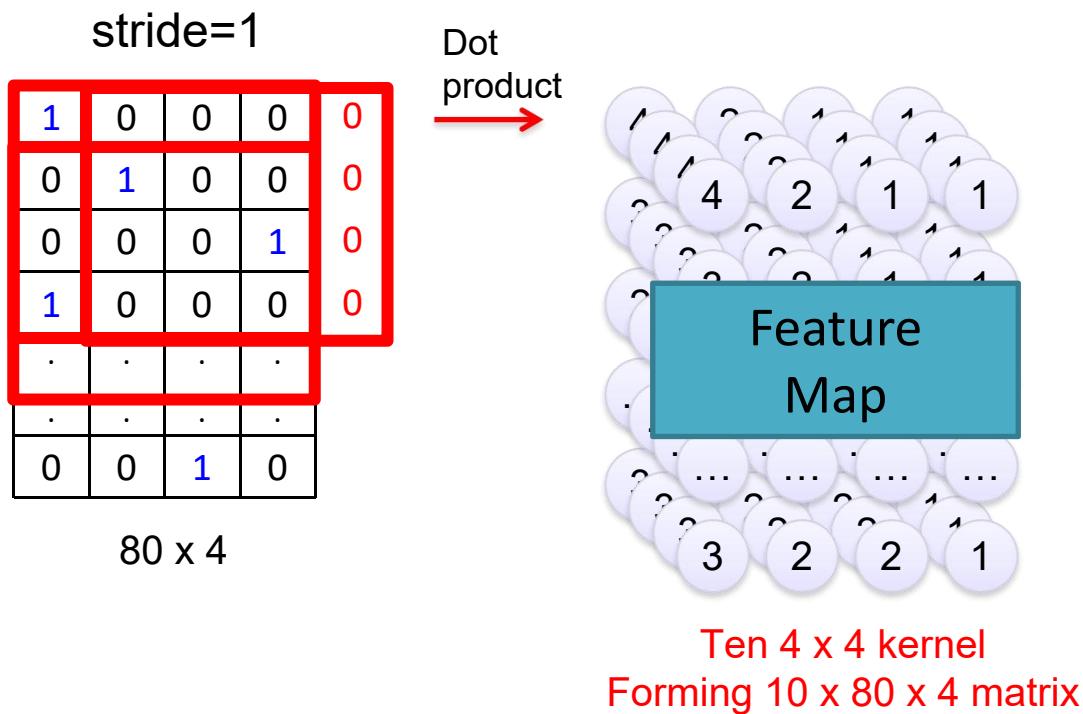
$$\text{ReLU}(x) = \max(0, x)$$

A convolutional layer



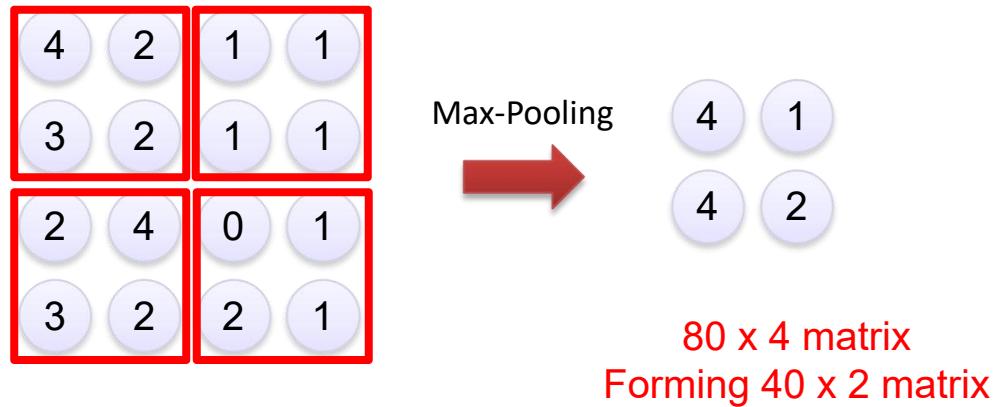
Repeat this for each filter

A convolutional layer

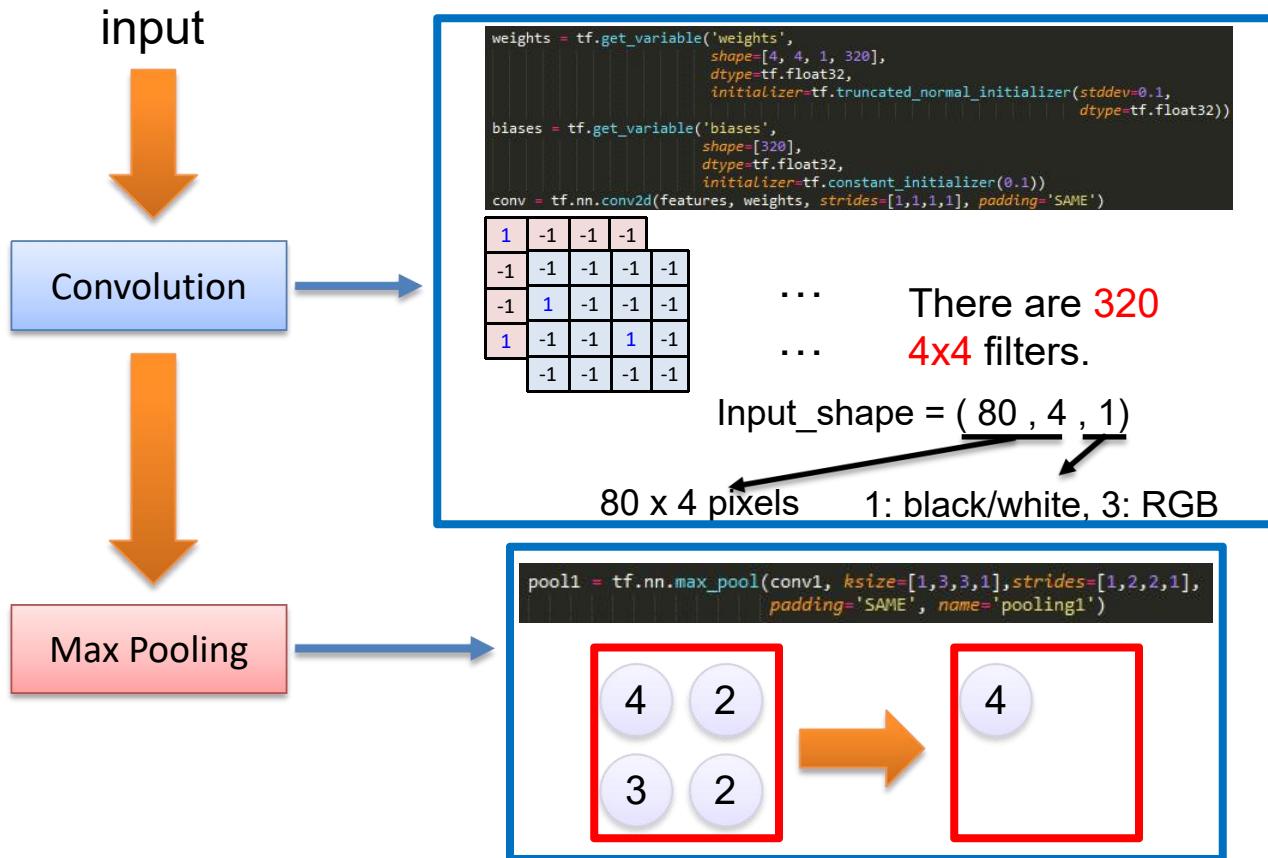


A convolutional layer

$$p_{nfi} = \max_{|k| < P/2} (a_{nf,i+k})$$



CNN model construction in Tensorflow



Max Pooling

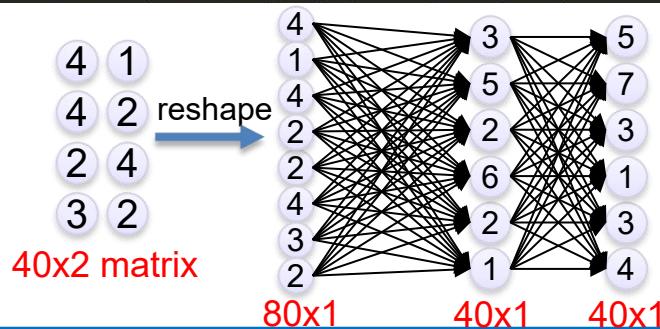


Fully Connected
Feedforward
network

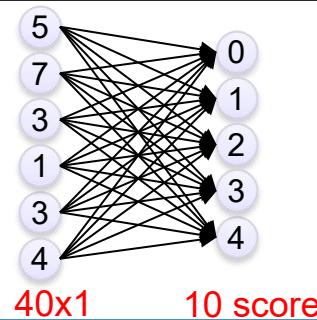


softmax

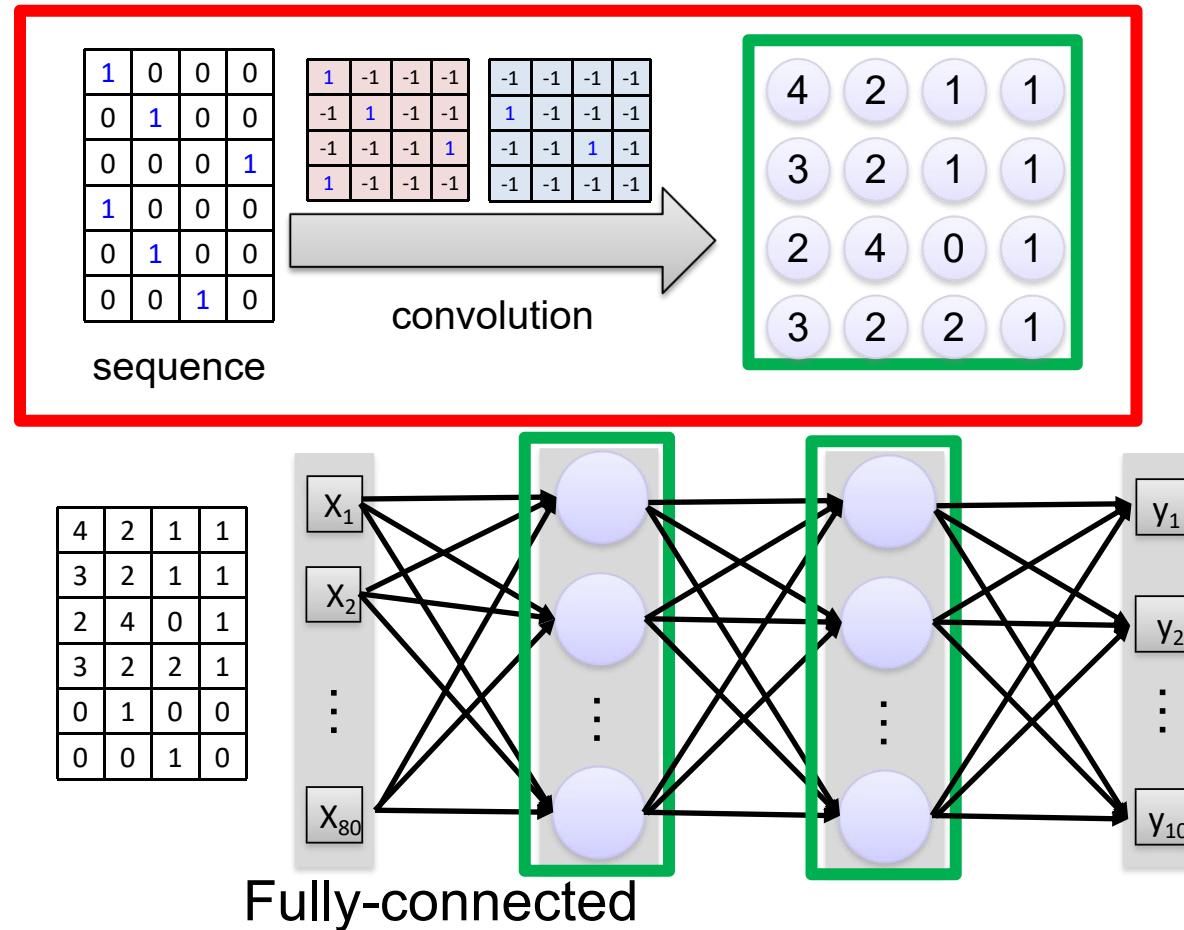
```
with tf.variable_scope('local3') as scope:  
    reshape = tf.reshape(pool1, shape=[batch_size, -1])  
    dim = reshape.get_shape()[1].value  
    weights = tf.get_variable('weights',  
        shape=[dim, 40],  
        dtype=tf.float32,  
        initializer=tf.truncated_normal_initializer(stddev=0.005,dtype=tf.float32))  
    biases = tf.get_variable('biases',  
        shape=[40],  
        dtype=tf.float32,  
        initializer=tf.constant_initializer(0.1))  
    local3 = tf.nn.relu(tf.matmul(reshape, weights) + biases, name=scope.name)
```



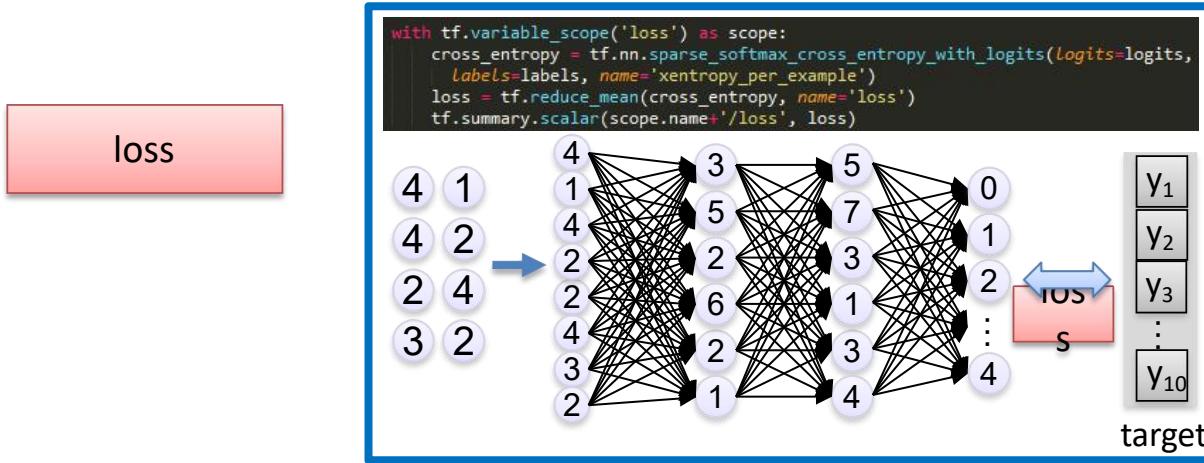
```
with tf.variable_scope('softmax_linear') as scope:  
    weights = tf.get_variable('softmax_linear',  
        shape=[40, n_classes],  
        dtype=tf.float32,  
        initializer=tf.truncated_normal_initializer(stddev=0.005,dtype=tf.float32))  
    biases = tf.get_variable('biases',  
        shape=[n_classes],  
        dtype=tf.float32,  
        initializer=tf.constant_initializer(0.1))  
    softmax_linear = tf.add(tf.matmul(local3, weights), biases, name='softmax_linear')
```



Convolution v.s. Fully Connected



Pick the best model



Randomly initialize network parameters

Pick the 1st batch

$L' = l_1 + l_2 + \dots$

Update parameters once

Pick the 2nd batch

$L'' = l_1 + l_2 + \dots$

Update parameters once

...

Until all mini-batches have been picked

training

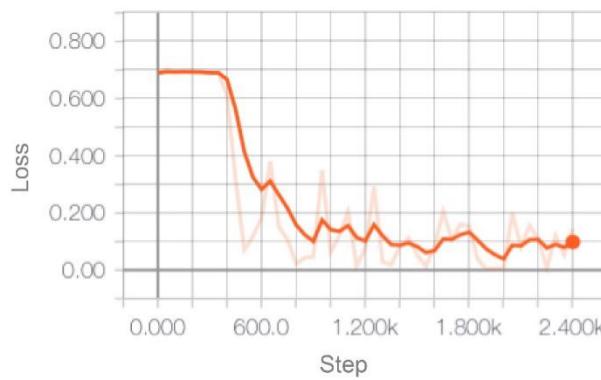
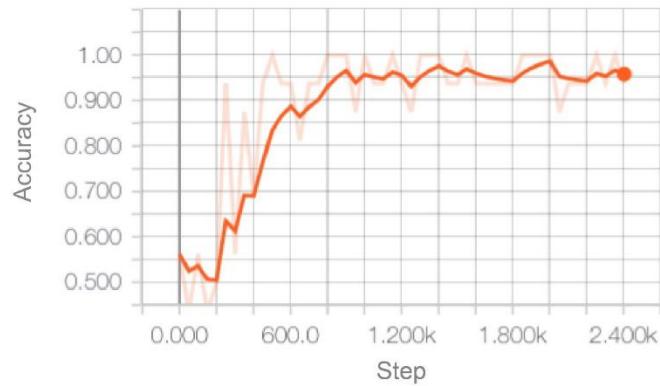
```
N_CLASSES = 2
seq_W = 20 # resize the seq, if the input seq is too large, training will be very slow.
seq_H = 80
BATCH_SIZE = 16
CAPACITY = 2000
MAX_STEP = 10000 # with current parameters, it is suggested to use MAX_STEP>10k
learning_rate = 0.0001 # with current parameters, it is suggested to use learning rate<0.0001
n = 1601
```

Output:

```
Step 0, train loss = 0.69, train accuracy = 62.50%
Step 50, train loss = 0.69, train accuracy = 56.25%
Step 100, train loss = 0.69, train accuracy = 43.75%
Step 150, train loss = 0.69, train accuracy = 56.25%
Step 200, train loss = 0.68, train accuracy = 93.75%
Step 250, train loss = 0.68, train accuracy = 43.75%
Step 300, train loss = 0.57, train accuracy = 68.75%
Step 350, train loss = 0.45, train accuracy = 68.75%
Step 400, train loss = 0.36, train accuracy = 75.00%
Step 450, train loss = 0.41, train accuracy = 87.50%
```

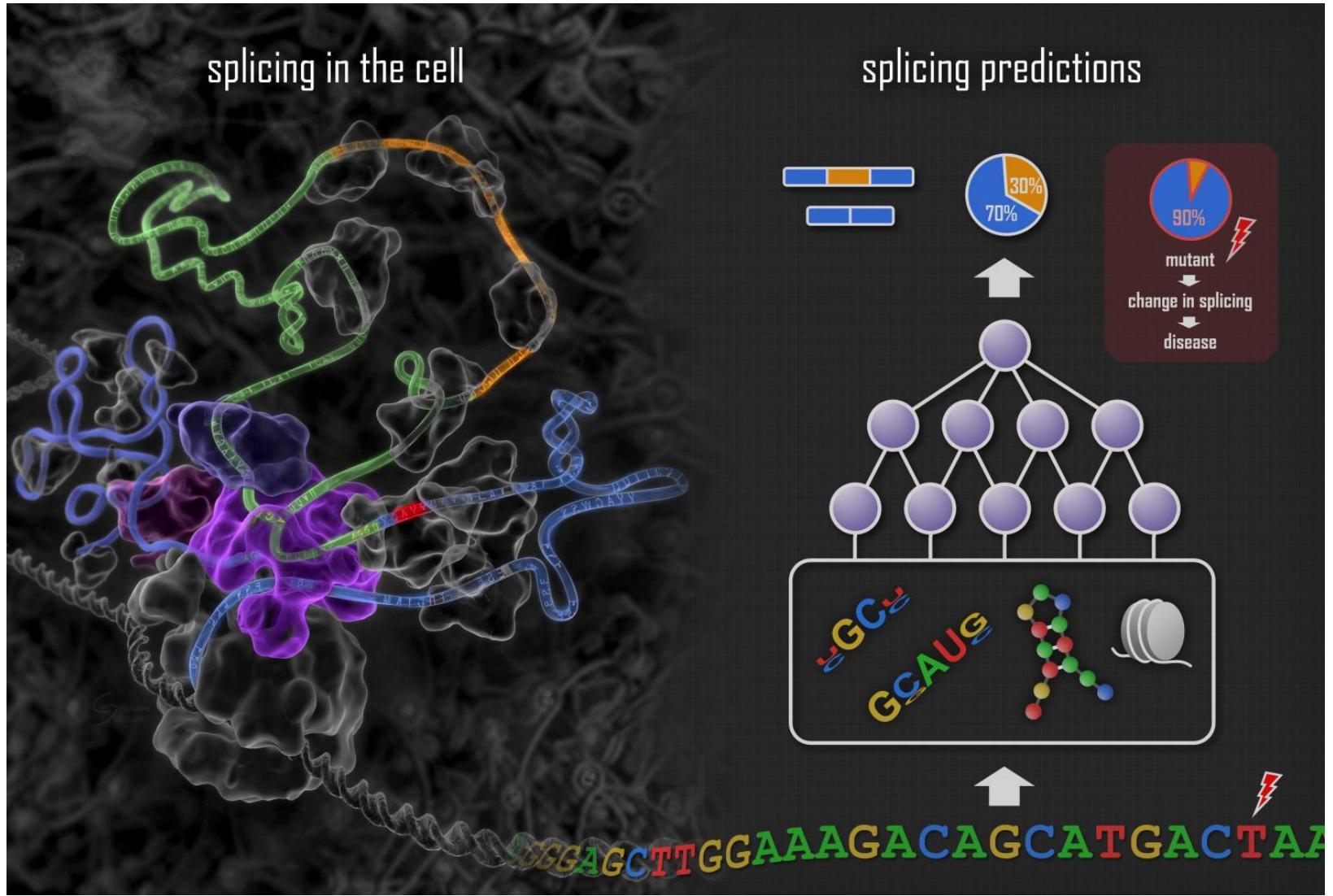
Training Module1

- Data: 5000 Cas9 sequences with domain sites and 5000 without domain sites

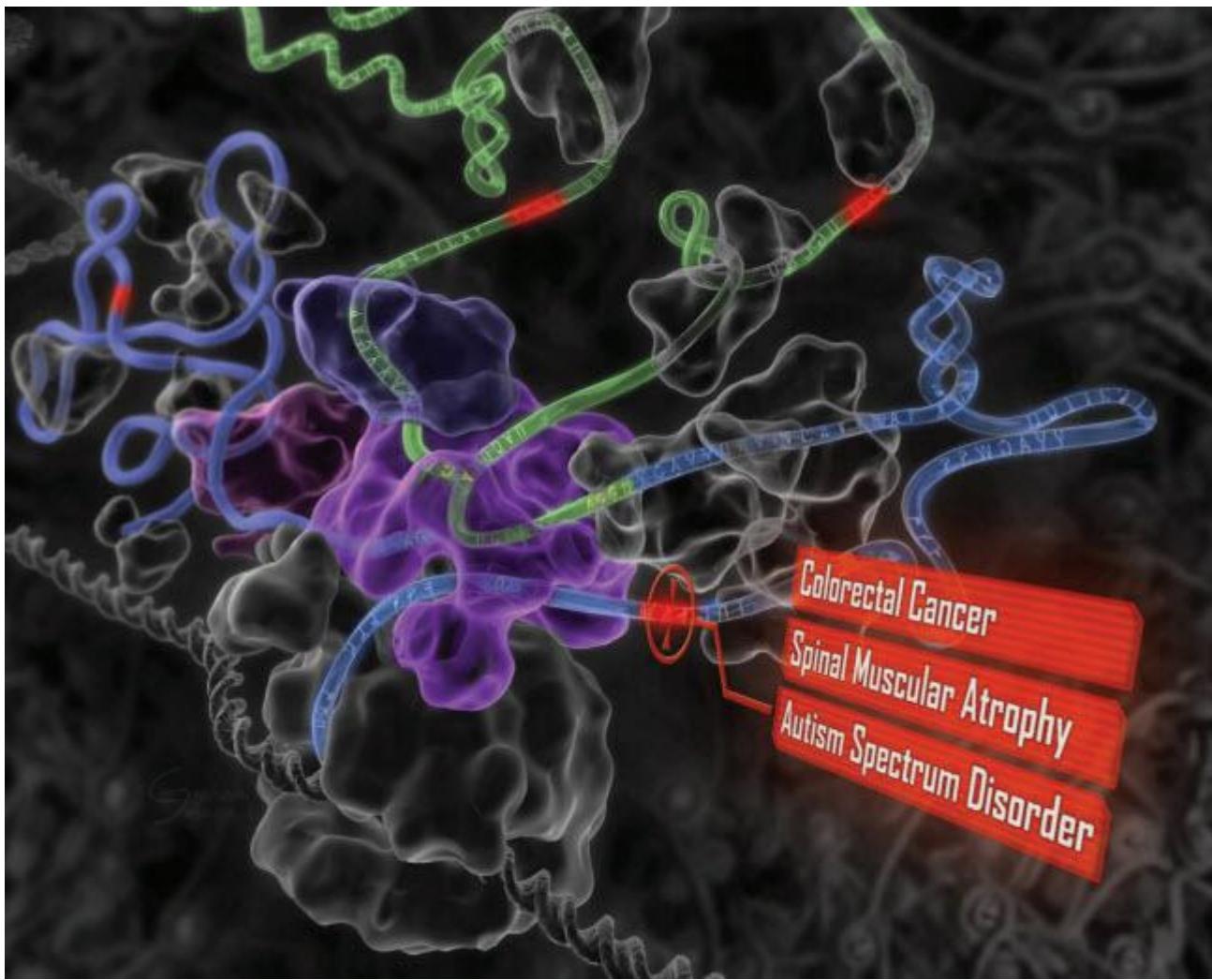


Accuracy: 0.80225

New variant discovery

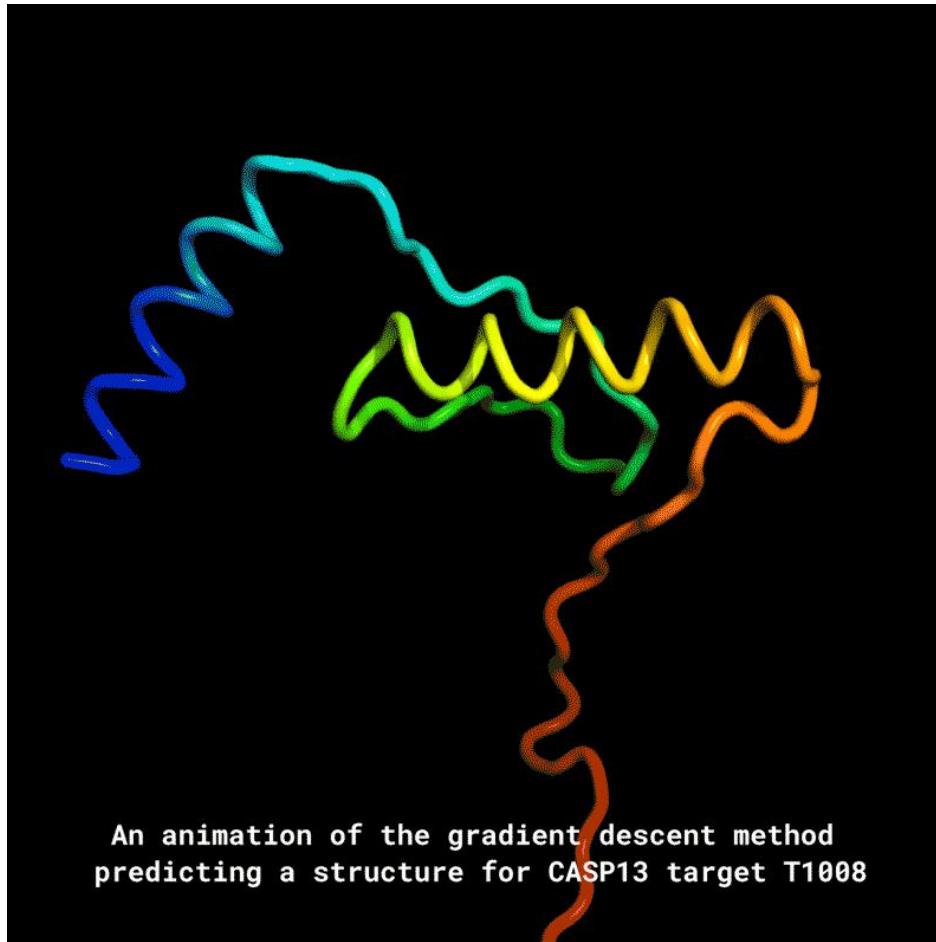


New variant discovery



CASP

(Critical Assessment of Techniques for Protein Structure Prediction)



CASP

(Critical Assessment of Techniques for Protein Structure Prediction)



I-TASSER
Protein Structure & Function Predictions

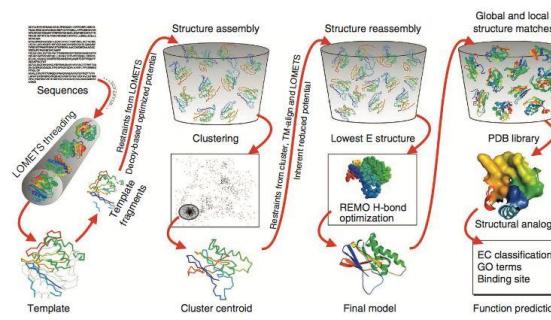
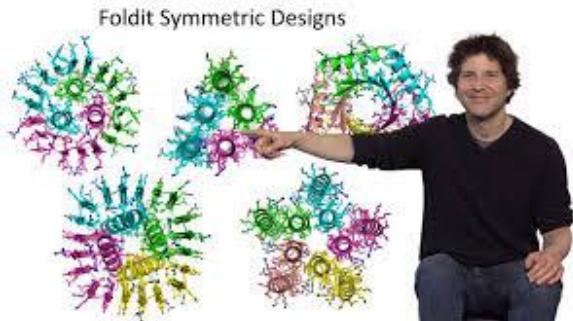


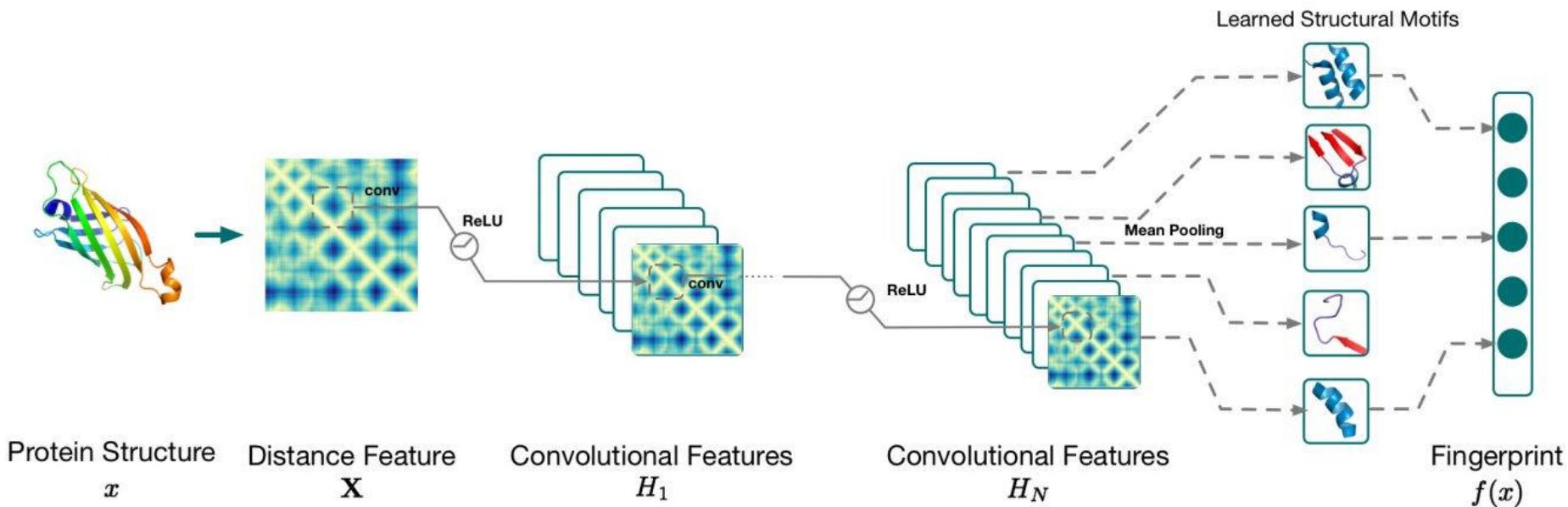
Figure 1 | A schematic representation of the I-TASSER protocol for protein structure and function predictions. The protein chains are colored from blue at the N-terminus to red at the C-terminus. 駿波



CASP

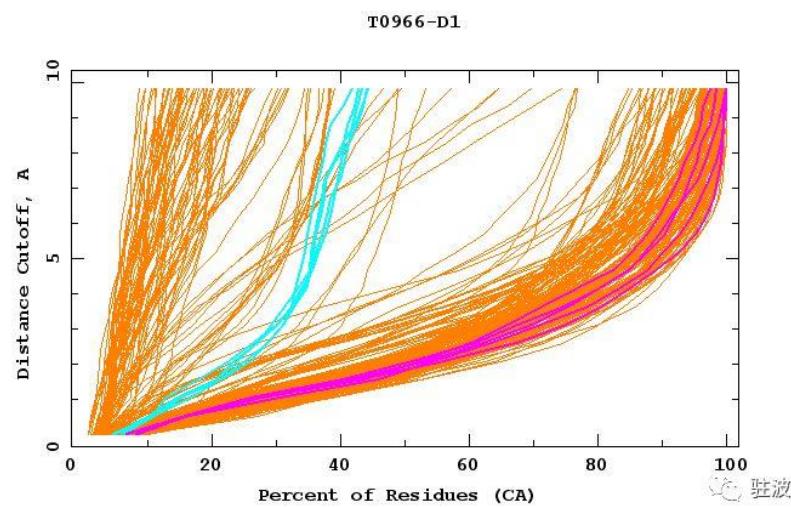
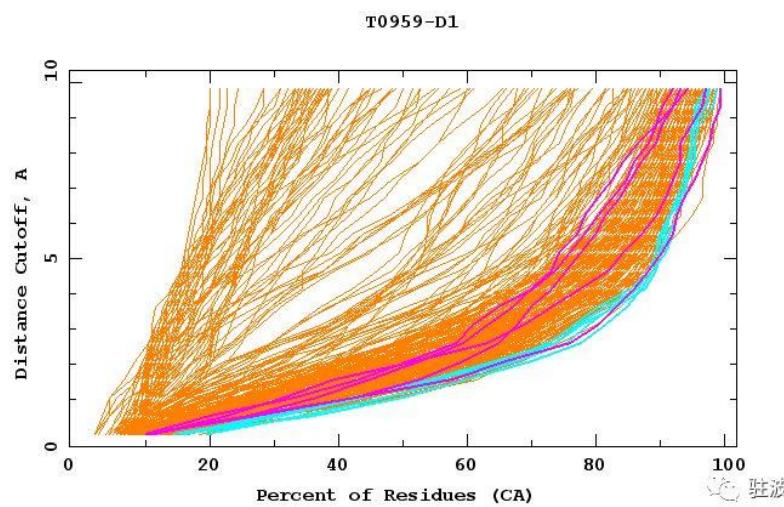
(Critical Assessment of Techniques for Protein Structure Prediction)

DeepFold: overall winner of CASP18



CASP

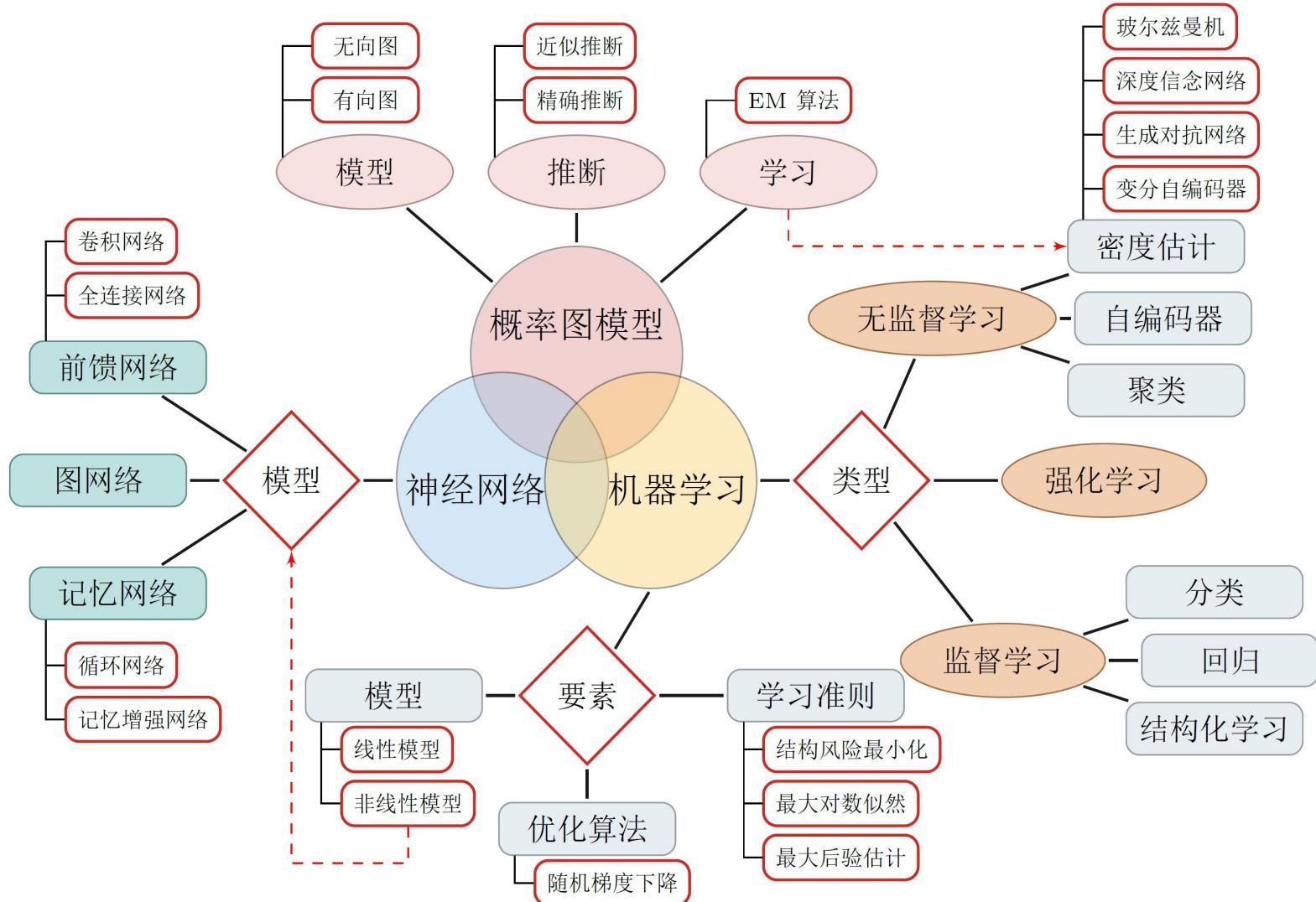
(Critical Assessment of Techniques for Protein Structure Prediction)



Recap (知识点总结)

- 深度学习：
 - 概念和区别
 - 模型和方法
 - 应用领域
- 生物大数据的深度学习：
 - 序列和特征
 - 特征学习和模型构建
 - 应用案例

汇总



推荐课程

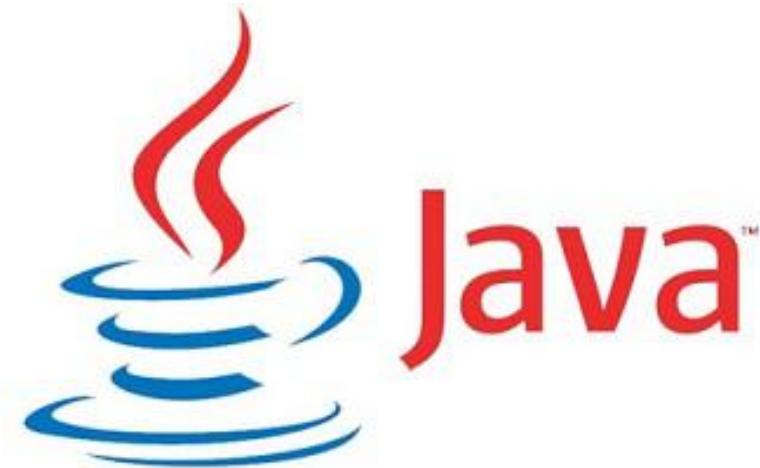
- 斯坦福大学CS224d: Deep Learning for Natural Language Processing
 - <http://cs224d.stanford.edu/>
 - Richard Socher 主要讲解自然语言处理领域的各种深度学习模型
- 斯坦福大学CS231n: Convolutional Neural Networks for Visual Recognition
 - <http://cs231n.stanford.edu/>
 - Fei-Fei Li Andrej Karpathy 主要讲解CNN、RNN在图像领域的应用
- 加州大学伯克利分校 CS 294: Deep Reinforcement Learning
 - <http://rail.eecs.berkeley.edu/deeprlcourse/>

推荐材料

- 林轩田 《机器学习基石》 《机器学习技法》
 - <https://www.csie.ntu.edu.tw/~htlin/mooc/>
- 李宏毅 《1天搞懂深度学习》
 - http://speech.ee.ntu.edu.tw/~tlkagk/slides/Tutorial_HYLee_Deep.pptx
- 李宏毅 《Generative Adversarial Network (GAN)》
 - http://speech.ee.ntu.edu.tw/~tlkagk/slides/Tutorial_HYLee_GAN.pptx

编程语言的选择





R 与 Python 语言的区别



学习难度大

入门简单

命令式编程

适合处理大量数据

统计功能强大

功能强大

Nov 2018	Nov 2017	Change	Programming Language	Ratings	Change
1	1		Java	16.746%	+3.51%
2	2		C	14.396%	+5.10%
3	3		C++	8.282%	+2.94%
4	4		Python	7.683%	+3.20%
5	7	▲	Visual Basic .NET	6.490%	+3.58%
6	5	▼	C#	3.952%	+0.94%
7	6	▼	JavaScript	2.655%	-0.32%
8	8		PHP	2.376%	+0.48%
9	-	▲	SQL	1.844%	+1.84%
10	14	▲	Go	1.495%	-0.07%
11	19	▲	Objective-C	1.476%	+0.06%
12	20	▲	Swift	1.455%	+0.07%
13	9	▼	Delphi/Object Pascal	1.423%	-0.32%
14	11	▼	R	1.407%	-0.20%
15	10	▼	Assembly language	1.108%	-0.61%
16	13	▼	Ruby	1.091%	-0.50%
17	12	▼	MATLAB	1.030%	-0.57%
18	15	▼	Perl	1.001%	-0.56%
19	18	▼	PL/SQL	1.000%	-0.45%
20	17	▼	Visual Basic	0.854%	-0.63%

Python在线学习推荐----

RUNOOB.COM

首页 HTML CSS JAVASCRIPT JQUERY BOOTSTRAP SQL

Python 基础教程

Python 基础教程

Python 简介

Python 环境搭建

Python 中文编码

Python 基础语法

Python 变量类型

Python 运算符

Python 条件语句

Python 循环语句

Python While 循环语句

Python for 循环语句

Python 循环嵌套

Python break 语句

Python continue 语句

Python pass 语句

Python Number(数字)

Python 字符串

Python 列表(List)

Python 元组

Python 字典(Dictionary)

Python 日期和时间

Python 函数

Python 模块

Python 基础教程



Python是一种高级的、解释型的、面向对象的编程语言，由Guido van Rossum设计，像Perl语言一样易于使用。本教程主要针对初学者，介绍Python 3.X版本的语法和基本用法。

谁适合阅读本教程？

本教程适合想从零开始学习Python编程语言的开发人员。

学习本教程前你需要了解

在继续本教程之前，你应该了解一些基本的计算机编程概念。

执行Python程序

对于大多数程序语言，第一个入门编程代码便是"Hello World"。

实例(Python 2.0+)

```
#!/usr/bin/python
print "Hello, World!"
```

[运行实例 »](#)



Anaconda: 初学Python、入门机器学习的首选

NumPy

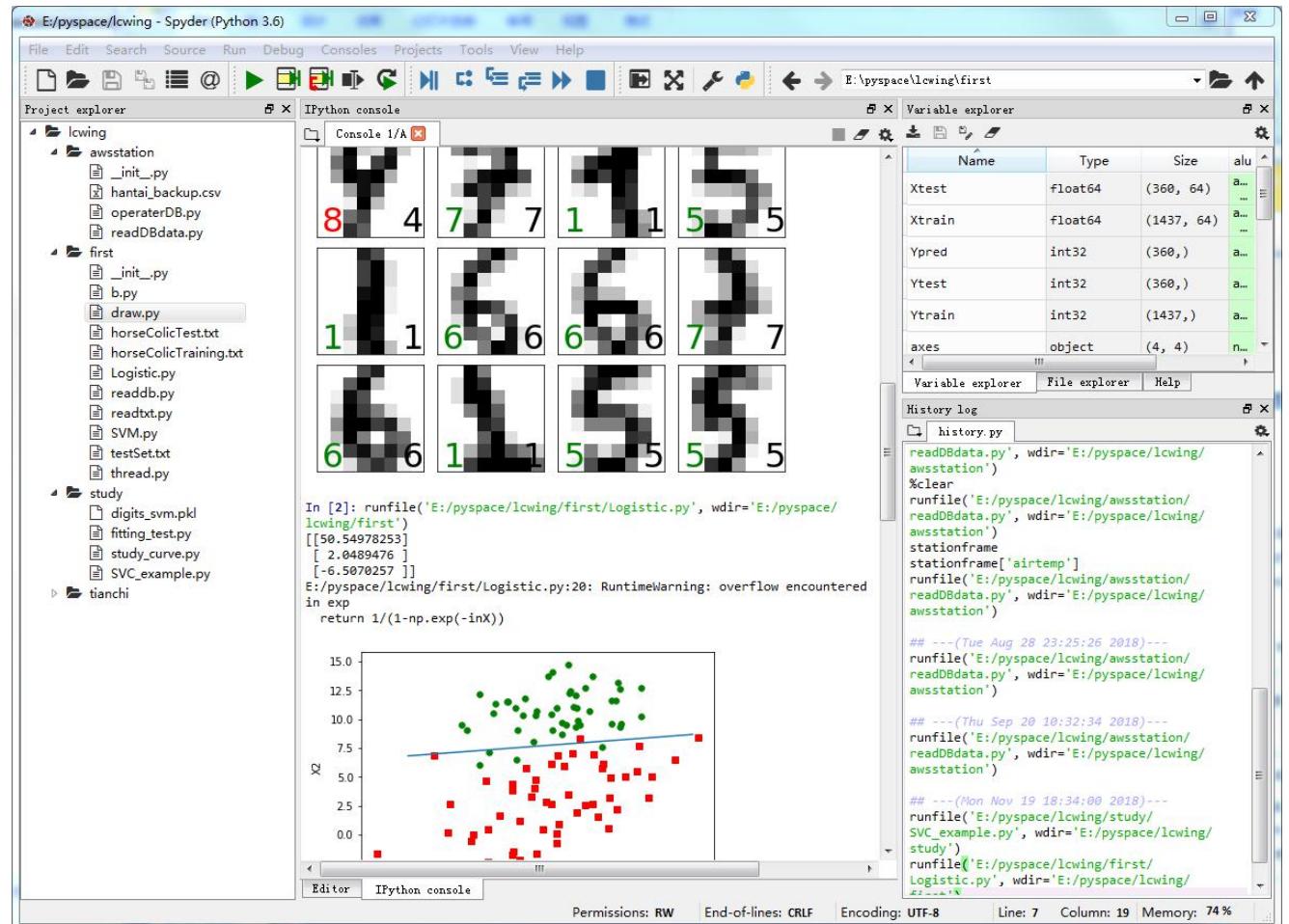
Scipy

Matplotlib

Pandas

Scikit-Learn

TensorFlow



References

- Angermueller, C., et al., Deep learning for computational biology. *Molecular systems biology*, 2016. 12(7): p. 878.
- Jurtz, V.I., et al., An introduction to deep learning on biological sequence data: examples and solutions. *Bioinformatics*, 2017. 33(22): p. 3685-3690.
- Wang, S., et al., Protein secondary structure prediction using deep convolutional neural fields. *Scientific reports*, 2016. 6.
- Angermueller, C., et al., DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning. *Genome biology*, 2017. 18(1): p. 67.
- Yuan, L., et al., Applications of Deep Learning in Biological and Medical Data Analysis. *PROGRESS IN BIOCHEMISTRY AND BIOPHYSICS*, 2016. 43(5): p. 472-483.
- Chai, G., et al., HMMCAS: a web tool for the identification and domain annotations of Cas proteins. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2017.

深度学习的问题。 . .

?

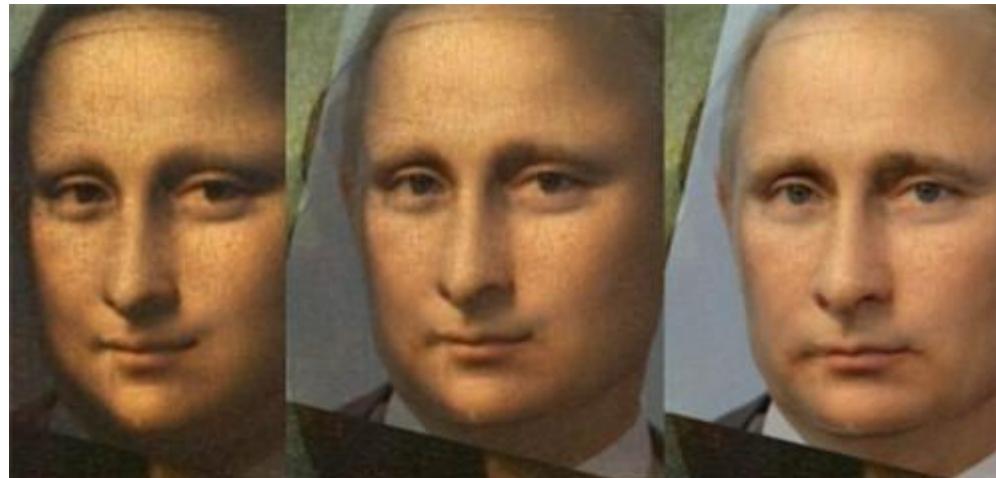


Feature extraction? Or D&C

Case 1

为何对基因的分析结果难以达到共识？“其中一个原因是遗传变异太常见。一个典型的基因包含了数以百计的变异，有些变异可能对健康没有任何影响。有些疾病可能是多个基因变异相互作用的结果，这种作用可能经常被算法所遗漏”，威康信托基金会桑格研究所遗传学家 Matthew Hurles 说。Hurles 目前正引领一个关于破解发育障碍的项目，该项目分析了 1400 个家庭中未确诊重症儿的外显子。他说，“即使当一个单一变异能解释疾病，也还需要进行统计调查、分析数据以及开展临床试验，以提供最终的确诊。”

深度学习的问题。。。。



Majority vote? Or rely on expert?

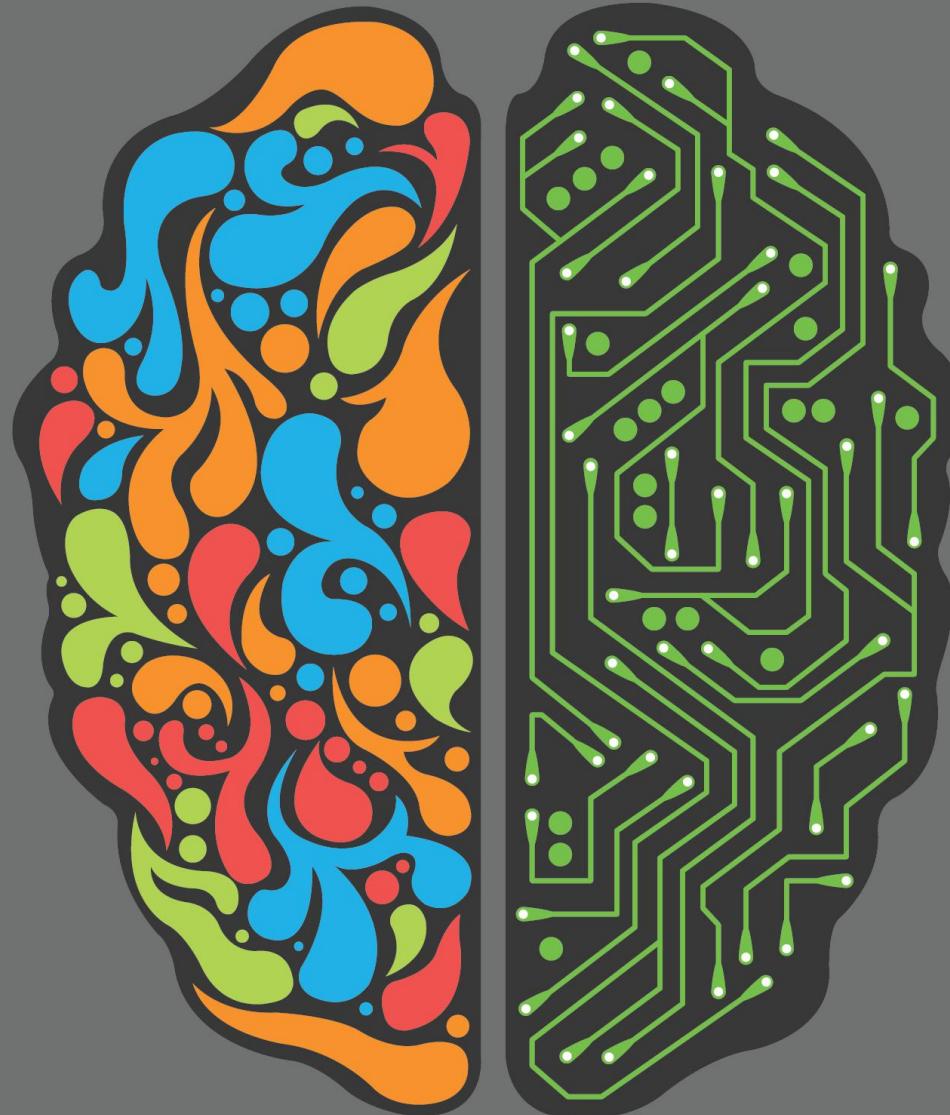
Case 1

一名40岁女子从小被诊断为重型地中海贫血及血小板过多症，后来并被切除脾臟及长期服用抑制血小板功能的药物，进一步做全外显子基因体定序，才发现真正的致病机转是因细胞内质网相关的基因突变引起「先天性红血球生成异常性贫血」，输血治疗地中海贫血反受其害。

Case 2

2007年，美国最大的分子诊断公司Quest Diagnostics 子公司——Athena Diagnostics 给患儿Christian Millare 进行了SCN1A基因突变检测。2008年5月1日，两岁的Christian疾病发作并不幸去世。此后死亡患儿母亲Amy Williams 通过咨询专家和查阅发表的文献，再结合Christian的病历，她坚信儿子的生命不应该这样过早的结束。8年后，Williams对Quest 诊断公司和Athena诊断公司提出了诉讼。

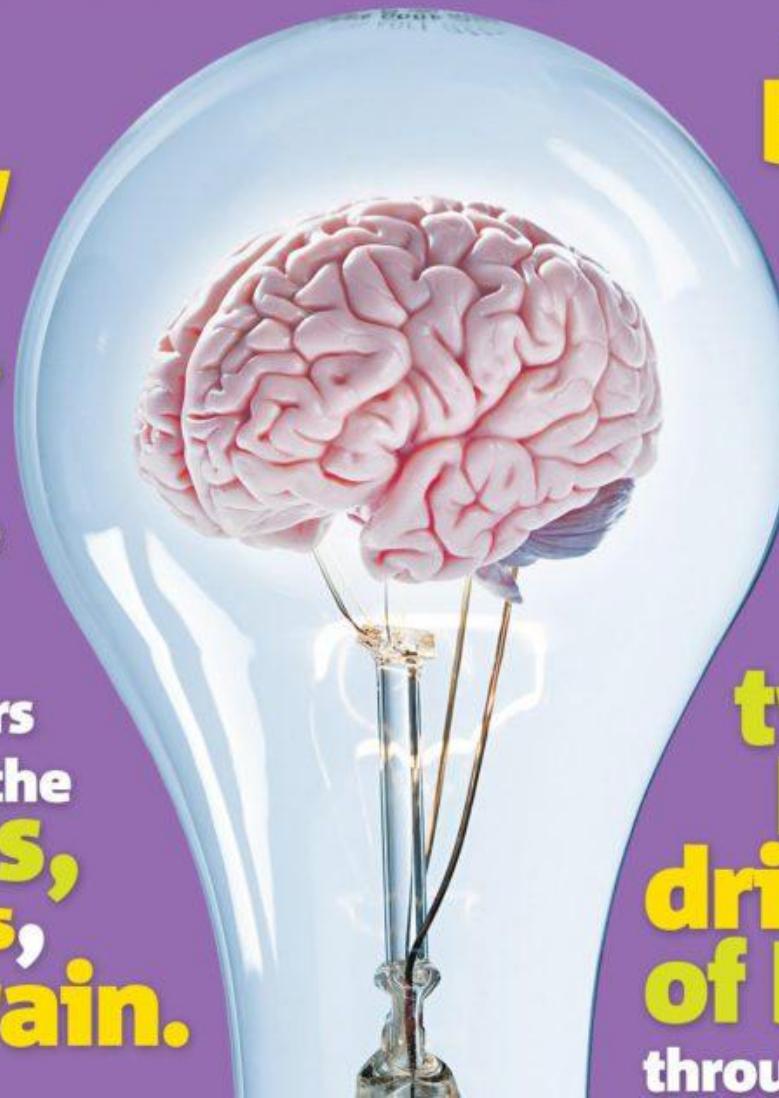
深度学习的问题：脑科学。。



深度学习的问题：脑科学。。。

① Your brain generates enough electricity to power a light bulb.

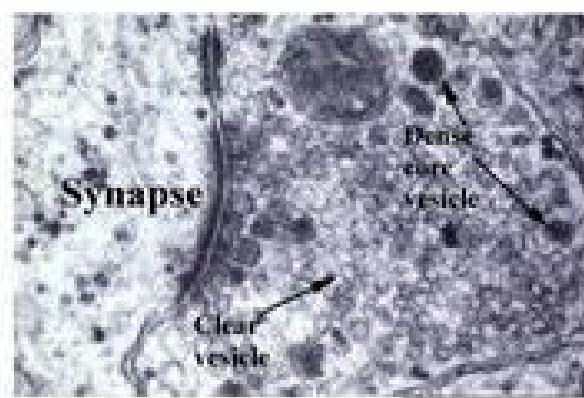
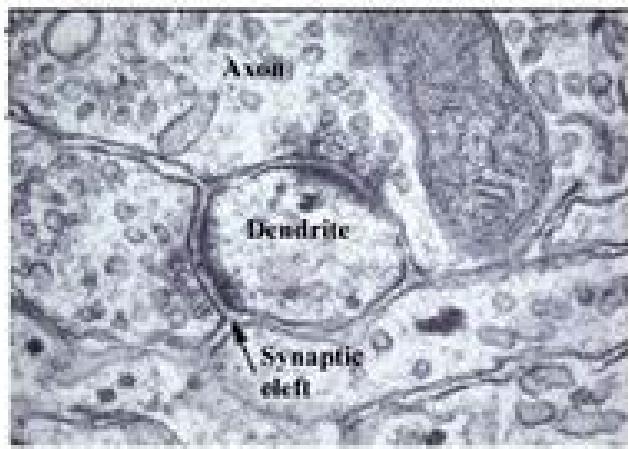
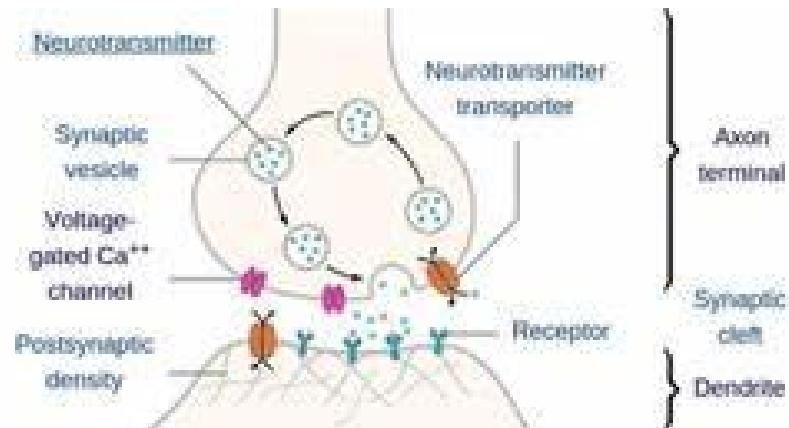
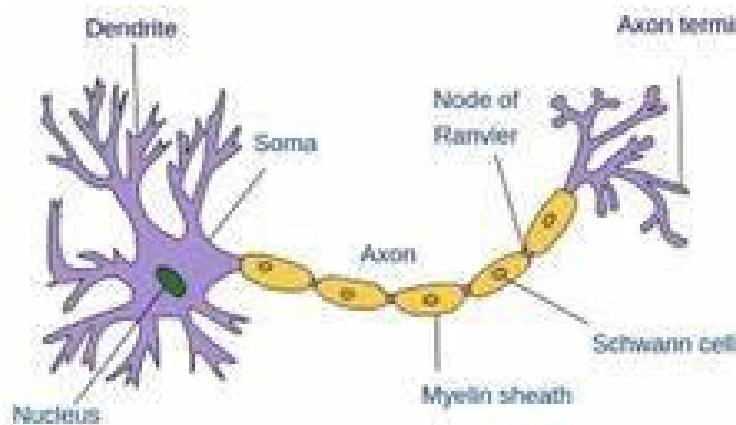
② It would take close to 3,000 years to count the neurons, or nerve cells, in your brain.



④ Exercise can make your brain work better.

⑤ Each minute, about 750 millilitres – or two and a bit fizzy drinks cans – of blood travels through the brain.

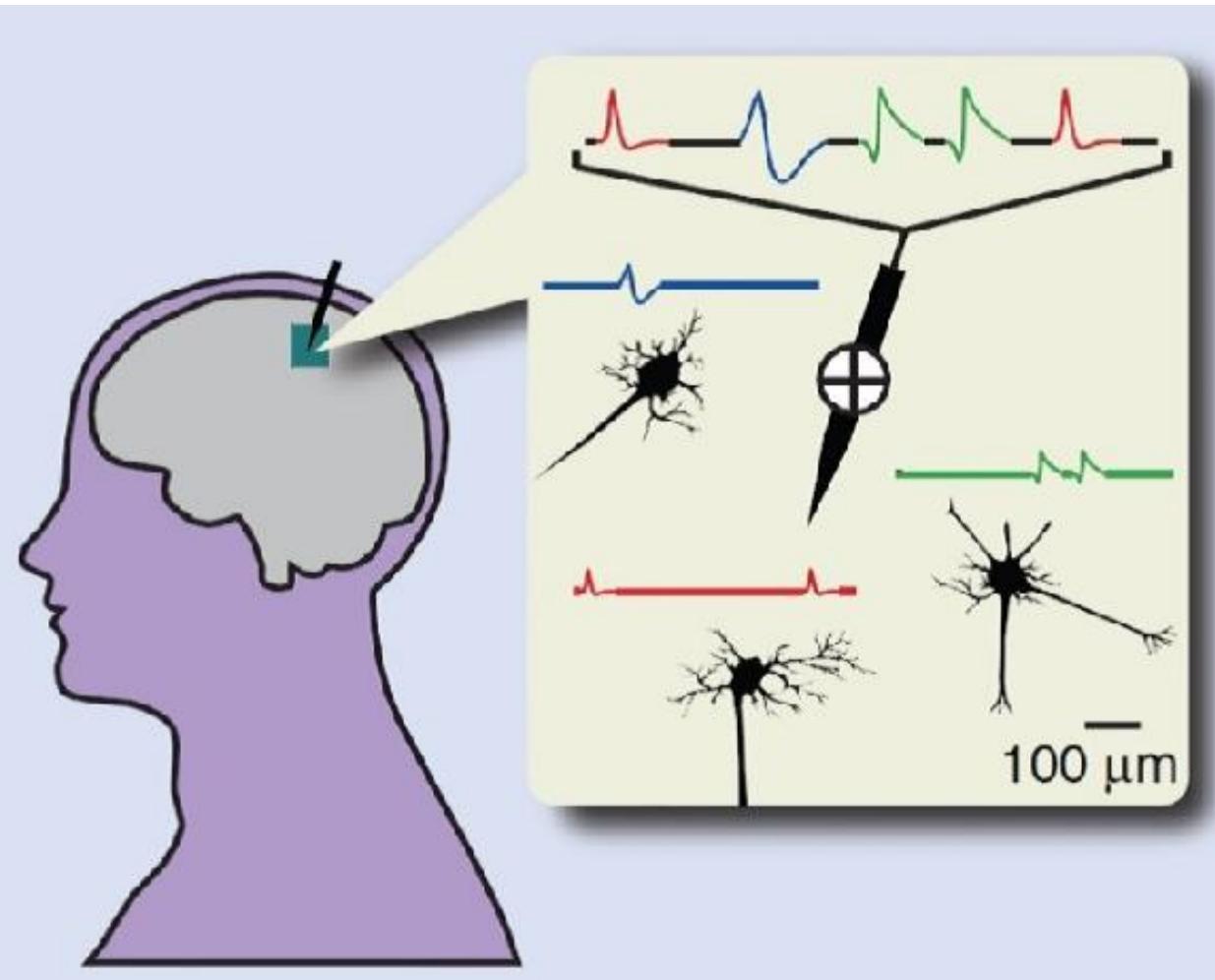
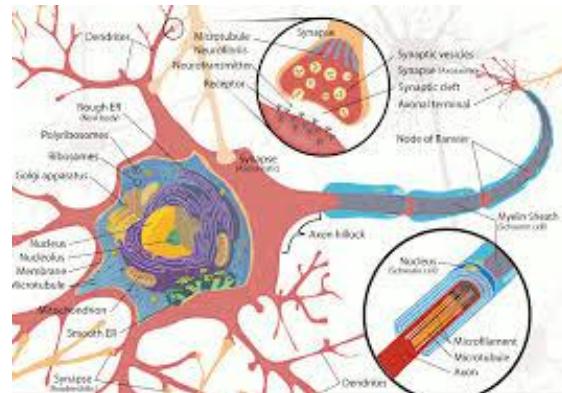
深度学习的问题：脑科学。。。



The Brain vs. Deep Learning vs. Singularity

深度学习的问题：脑科学。。。

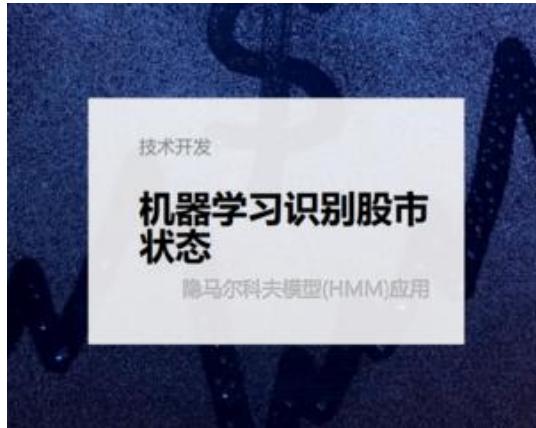
We are actually HMM (or Deep learning) animal...



深度学习的问题。 。 。

More problems that you can think of?

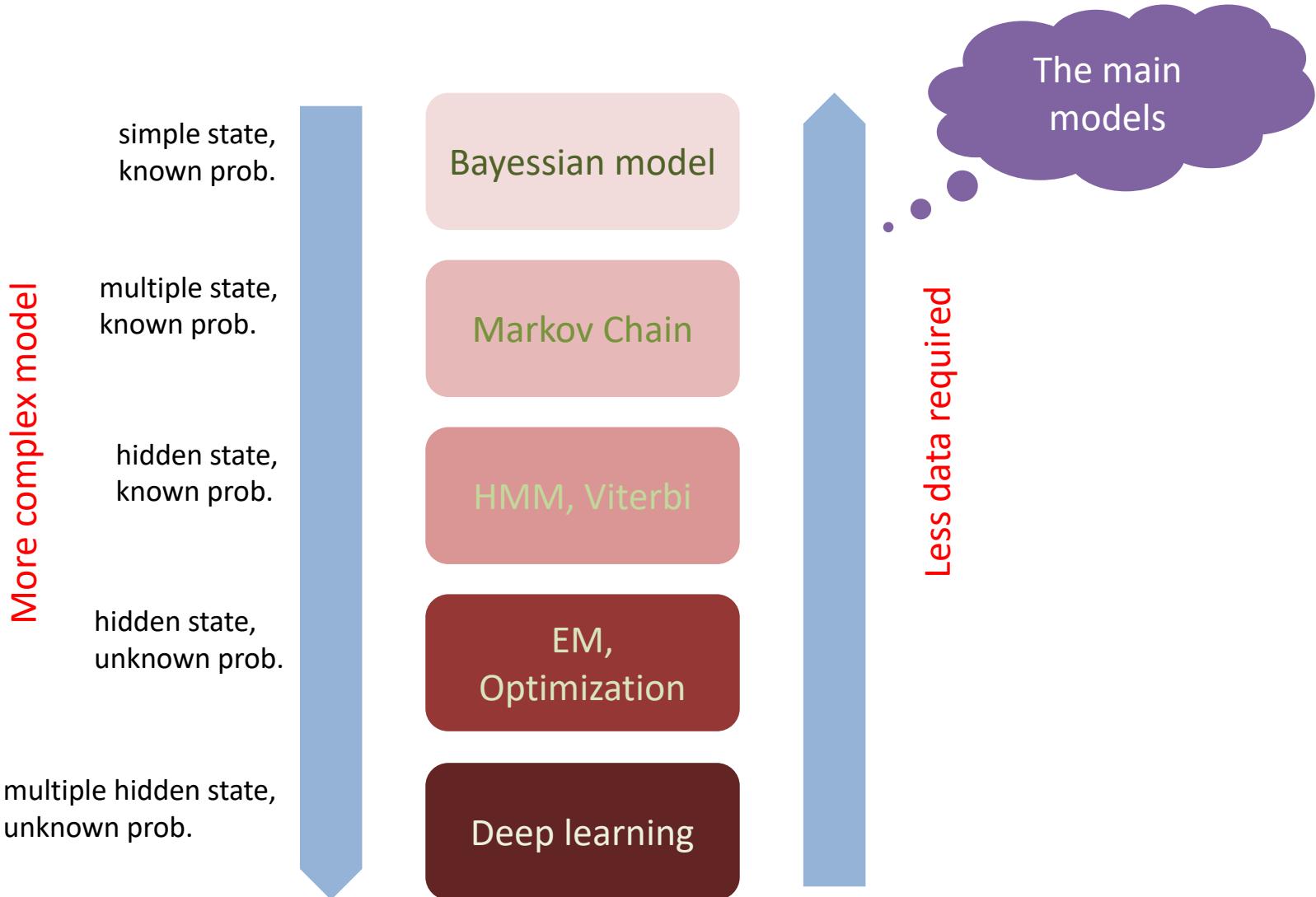
不建议用简单的深度学习来解决的问题。。。。



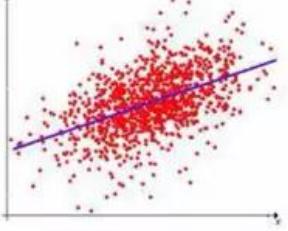
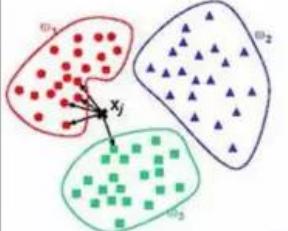
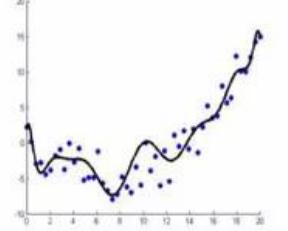
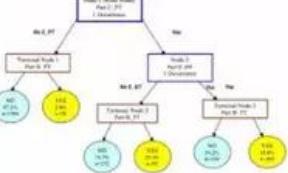
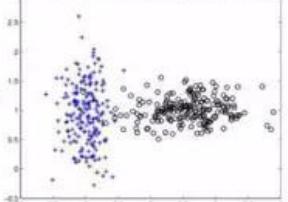
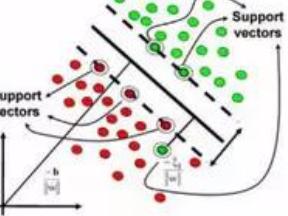
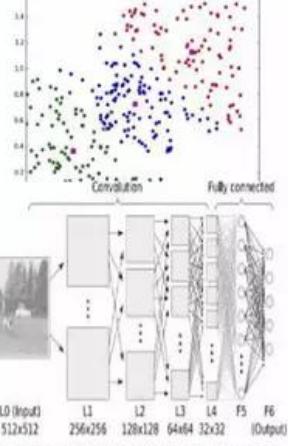
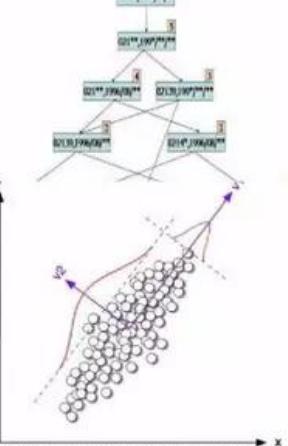
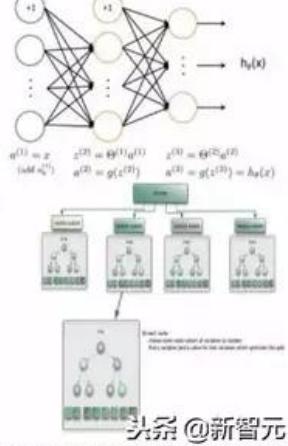
Statistical modeling

Distribution, Modeling, Prediction

Statistical modeling



Statistical modeling

回归算法	基于实例的算法	正则化方法
		
决策树学习	贝叶斯方法	基于核的算法
		
聚类算法	关联规则学习	人工神经网络
		

课程安排

- 生物背景和课程简介
- 传统生物统计学及其应用
- 生物统计学和生物大数据挖掘
 - Hidden Markov Model (HMM)及其应用
 - Markov Chain
 - HMM理论
 - HMM和基因识别 (Topic I)
 - HMM和序列比对 (Topic II)
 - 进化树的概率模型 (Topic III)
 - Motif finding中的概率模型 (Topic IV)
 - EM algorithm
 - Markov Chain Monte Carlo (MCMC)
 - 基因表达数据分析 (Topic V)
 - 聚类分析-Mixture model
 - Classification-Lasso Based variable selection
 - 基因网络推断 (Topic VI)
 - Bayesian网络
 - Gaussian Graphical Model
 - 基因网络分析 (Topic VII)
 - Network clustering
 - Network Motif
 - Markov random field (MRF)
 - Dimension reduction及其应用 (Topic VIII)
- 面向生物大数据挖掘的深度学习

研究对象：
生物序列，
进化树，
生物网络，
基因表达
...

方法：
生物计算与生物统计

Good luck in biostatistics!

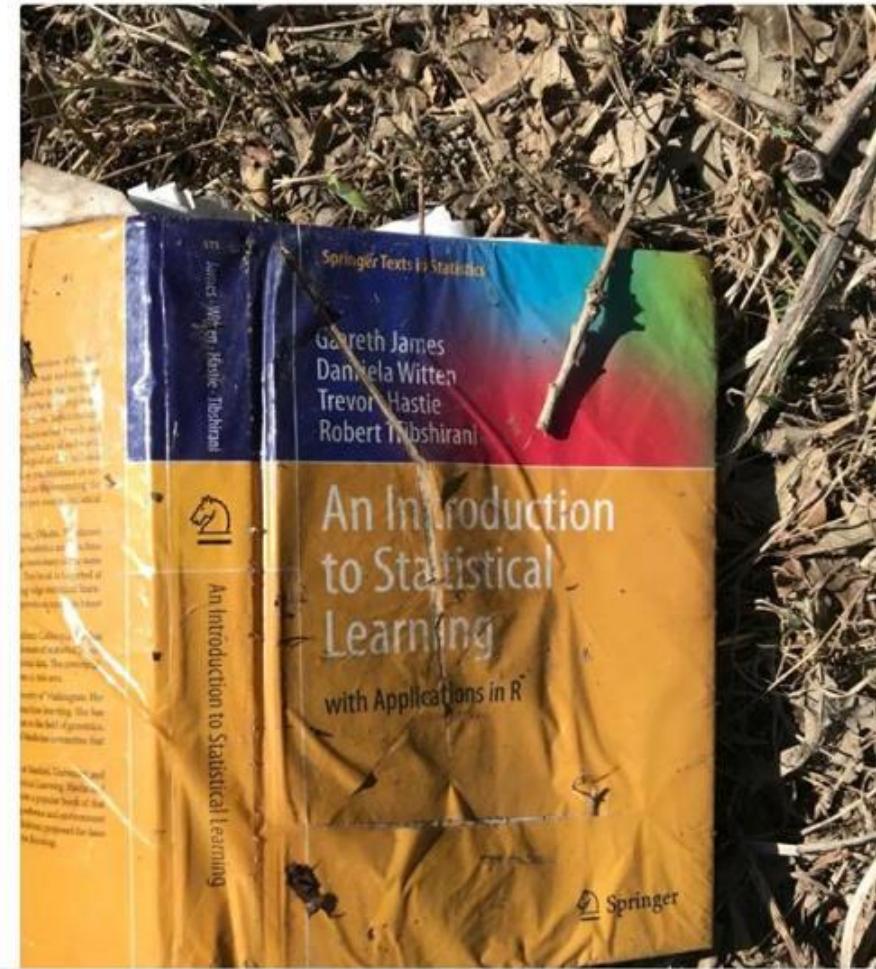


p < 0.05

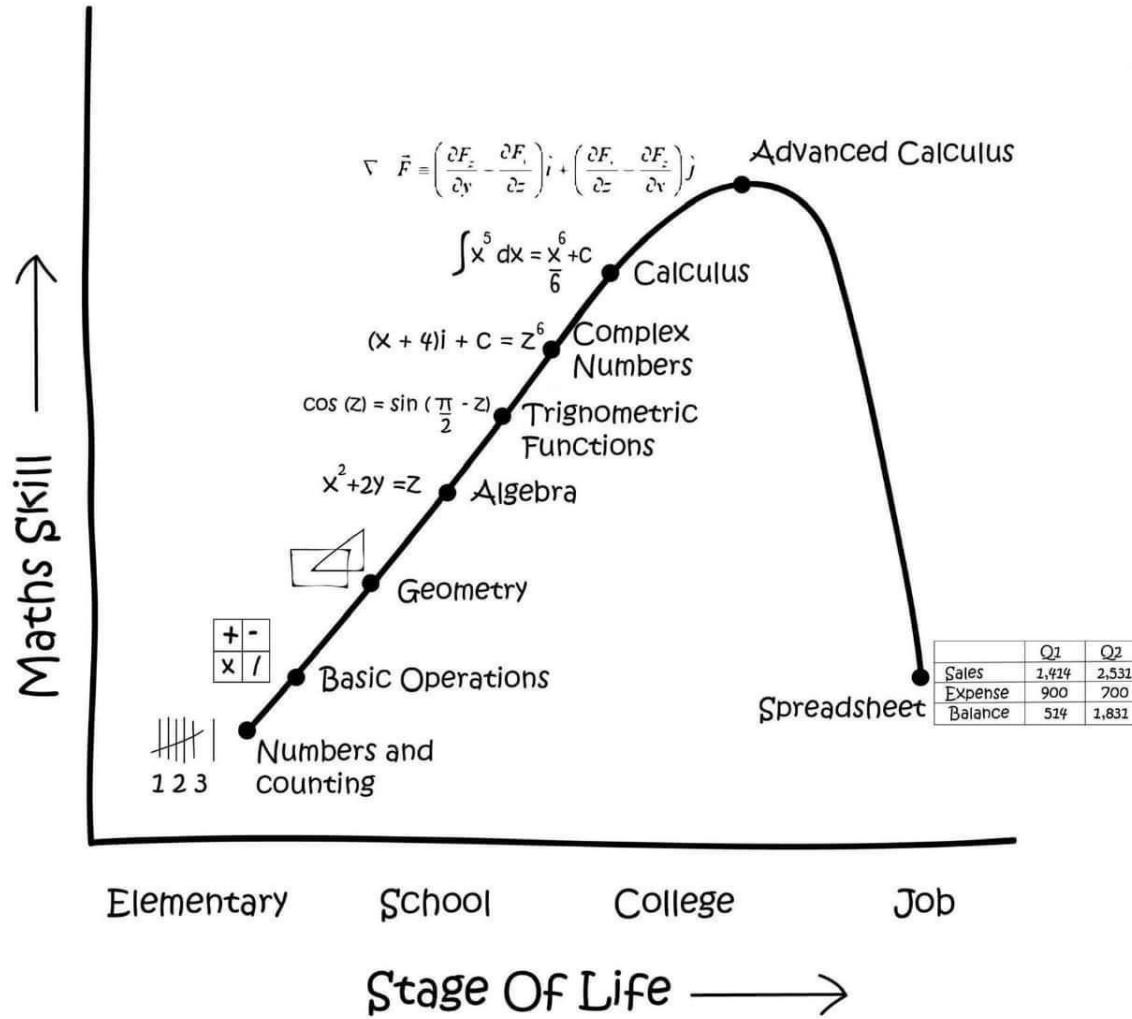


Noah Williams
@Bellmanequation

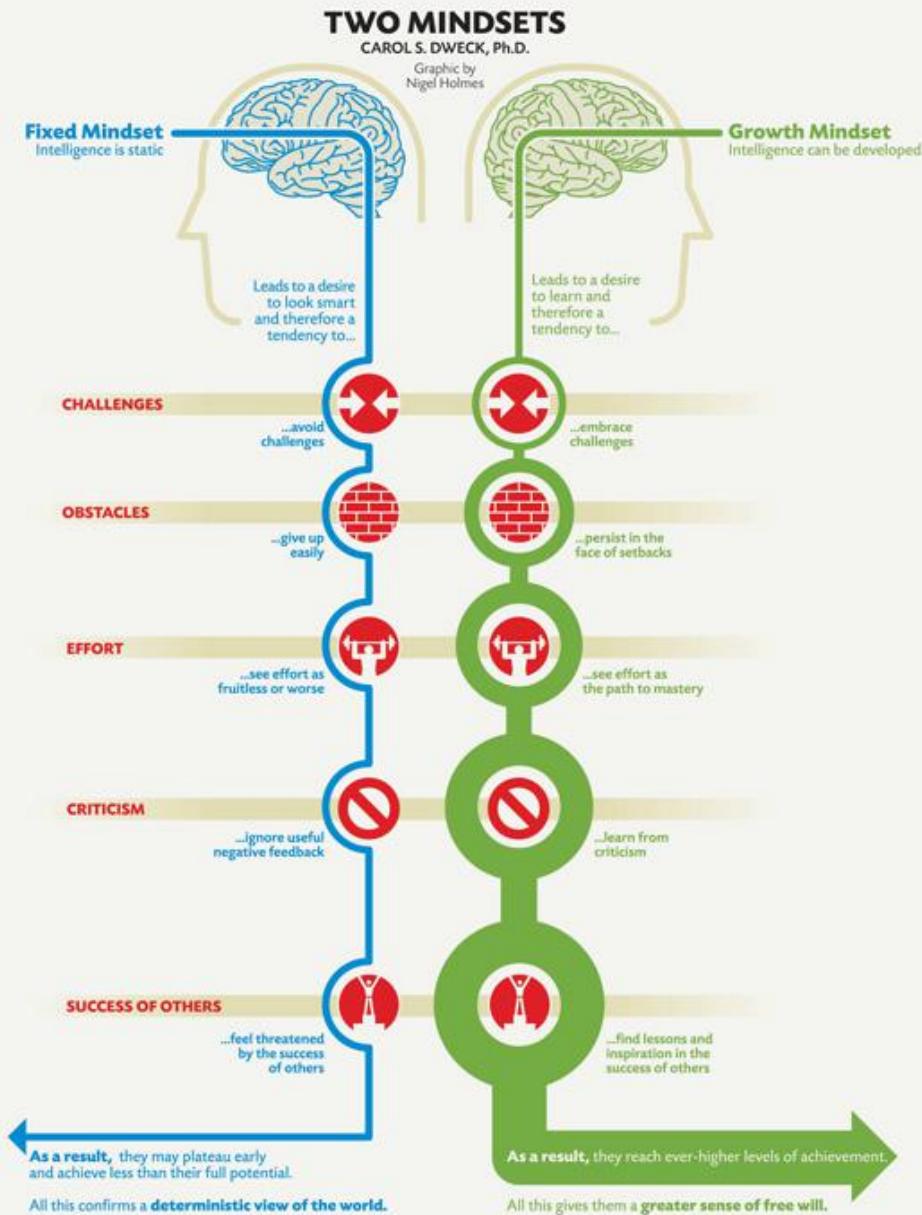
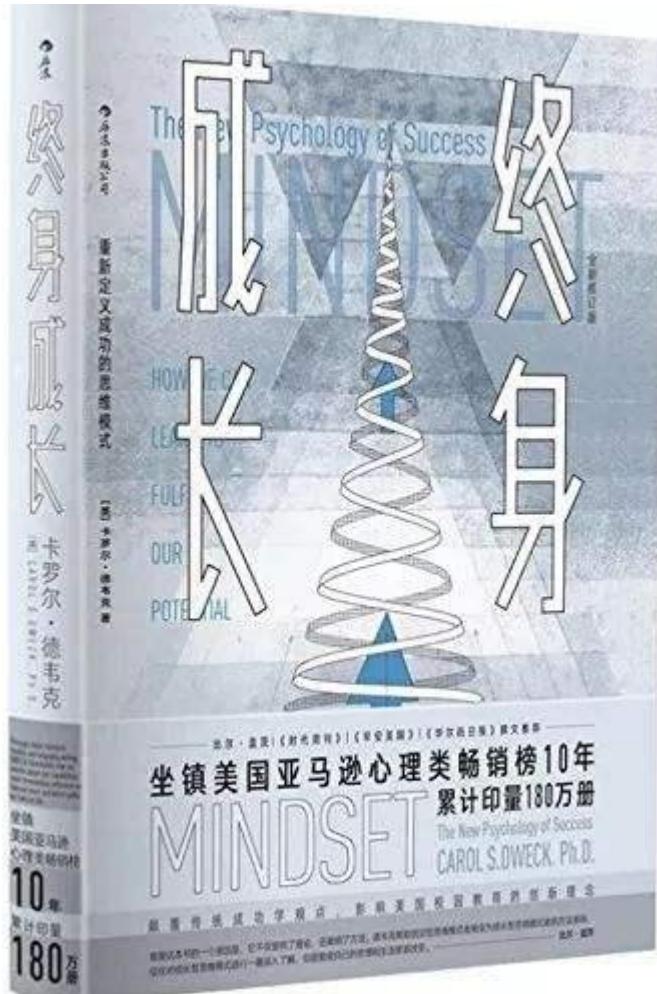
Found on the roadside:
@SpringerStats yellow book,
beside empty vodka bottles &
cigarette packs



Maybe you will not face these many statistical problems...



But you have the knowledge



Statistical modeling

More to learn:

- **Fast Fourier Transformation:**
<https://zhuanlan.zhihu.com/p/19763358>
- **Causality analysis:**
<https://zhuanlan.zhihu.com/p/33860572>
- ...

Statistical modeling

More to learn:

- 生物统计与生物信息的区别与联系? (<https://www.zhihu.com/question/30284194>)
 - 生物学的生物信息学：首先是讲进化，做序列比对，blast，构建进化树；然后，讲基因功能，基因功能富集分析等。侧重的是，将进化中的生物学原理，参数如何设置，软件如何使用，如何将这些软件应用到生物学问题。
 - 计算机系的生物信息学：首先讲的也是进化，但是，讲的是算法设计；然后讲了很多马尔科夫链，HMM模型，序列比对中的blosum62矩阵是怎么来的，如何加快计算效率，如何降低空间存储等等。侧重的是，如何设计合理的算法，给生物学的人使用。
 - 数学系的系统生物学：讲了回归中的penalty function，然后我学了LASSO回归；后来学了SVM中的核函数；还有Gibbs sampling，MCMC；还有时间序列中的Granger因果推断等等。
 - 生统对数学要求更高，生信基本不需要。国内生统大多是挂XXX卖XXX
- Yoshua Bengio:
 - 你死我活那是邪教，开放包容才是正道。
- What do you think?

Biostatistics in the market...

[公司文件] 加强与国内大学合作，吸纳全球优秀人才，共同推动中国基础研究
——任总与中国科学技术大学包信和校长座谈的讲话

2018-12-13 17:59 ④ 6799 113

只看楼主

总裁办电子邮件

电邮讲话【2018】128号

签发人：任正非

加强与国内大学合作，吸纳全球优秀人才，共同推动中国基础研究

——任总与中国科学技术大学包信和校长座谈的讲话

2018年11月19日

在高校学科设置上，我特别支持你们重视**统计学**。计算机科学不仅仅是技术，还应该以**统计学**为基础。大数据需要**统计学**，信息科学需要**统计学**，生命科学也需要**统计学**。国家要搞人工智能，更要重视**统计学**。**统计学**不是一个纯粹的学科，而是每一个学科都要以**统计学**为基础。

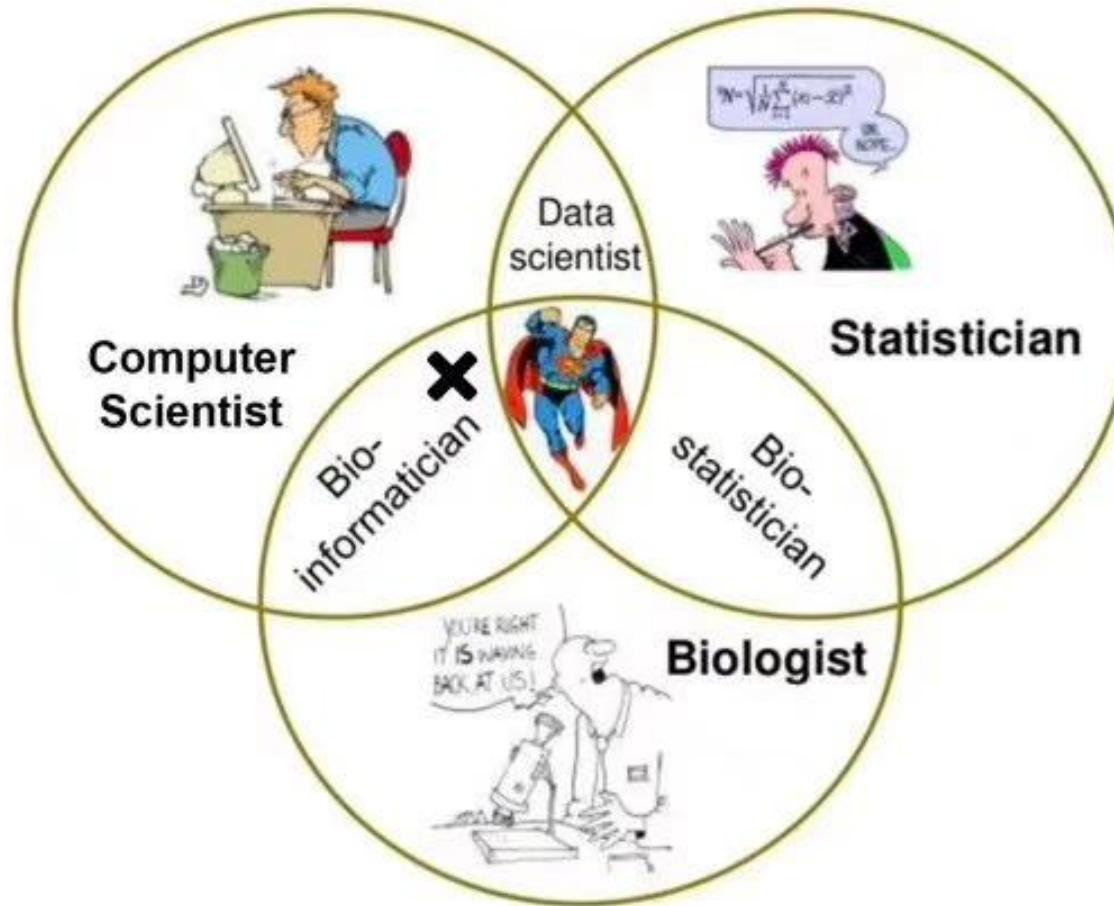


Or



Good luck in final exam!

Superman/Wonder woman



Good luck in your life!

谢谢大家！



- I thank you from the bottom of my heart
- That's so kind of you
- I am very thankful
- You are great
- I'm really grateful
- Thanks a million



Please accept my deepest thanks •



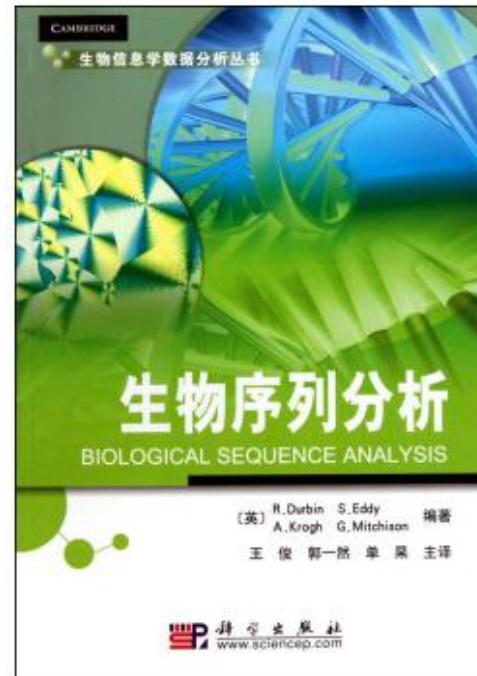
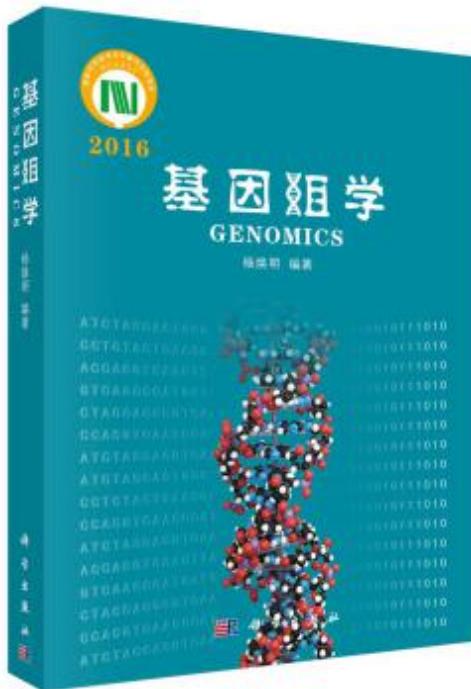
- I'm in your debt •
- You are the best •
- I owe you one •
- Thank you so much •
- I really appreciate it •

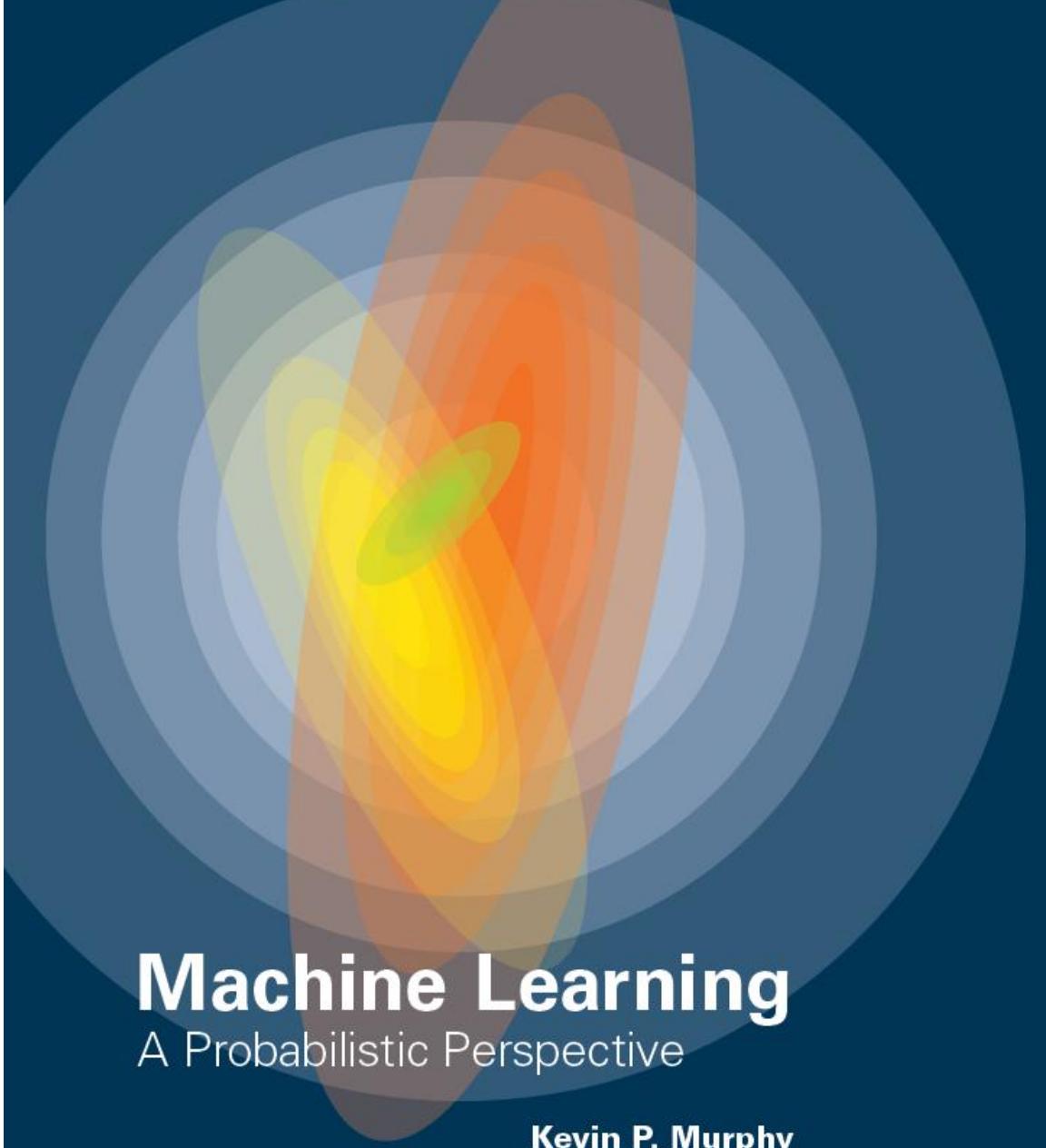


希望大家和我一起，
终身学习！

dakujem grazzi tahanan
efcharisto obrigada havala
obrigada
Thank You
blagodarya tack
Merci dekui dekui
gracias dzieki multumesc
Danke dank grazie
koszi kiitos obrigado

References

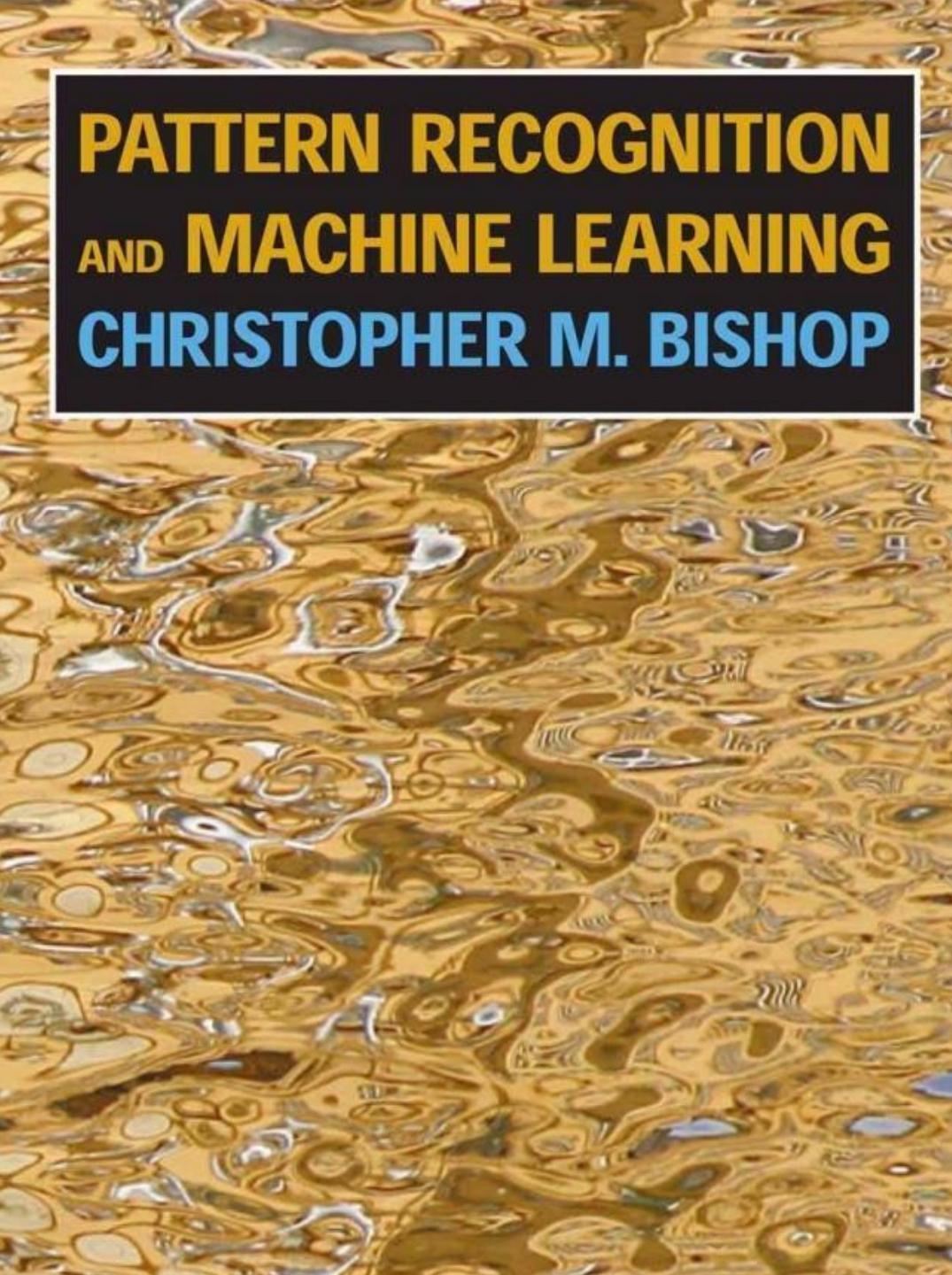




Machine Learning

A Probabilistic Perspective

Kevin P. Murphy



PATTERN RECOGNITION
AND MACHINE LEARNING
CHRISTOPHER M. BISHOP

DEEP LEARNING

Ian Goodfellow, Yoshua Bengio,
and Aaron Courville



Slides credits

- 生物信息学研究方法概述：北京大学生物信息中心
- 生物统计学：卜东波@中国科学院计算技术研究所，邓明华@北京大学
- 神经网络与深度学习：邱锡鹏@复旦大学
- Introduction to Computational Biology and Bioinformatics: Xiaole Shirley Liu Lab@Harvard University
- Combinatorial Methods in Computation Biology: Ken Sung Lab@NUS
- Deep Learning in the Life Sciences: MIT

