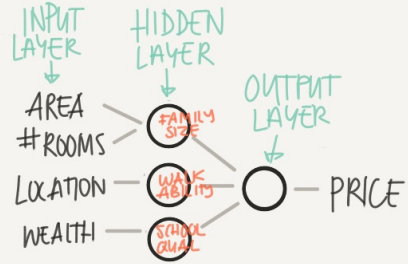
INTRO TO DEEP LEARNING



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

NNs CAN DEAL WITH BOTH STRUCTURED & UNSTRUCTURED DATA

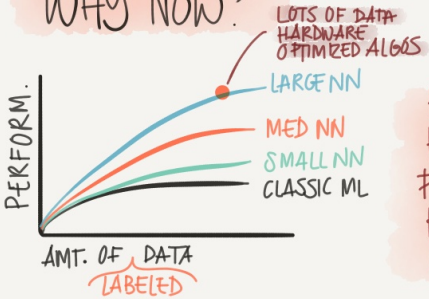
STRUCTURED

"THE QUICK BROWN FOX"

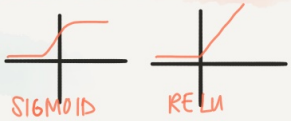
UNSTRUCTURED

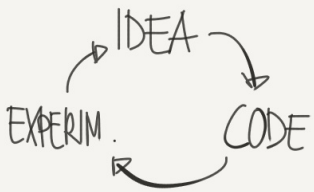
HUMANS ARE GOOD AT THIS

WHY NOW?



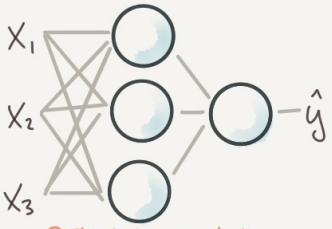
ONE OF THE BIG BREAKTHROUGHS HAS BEEN MOVING FROM SIGMOID TO RELU FOR FASTER GRADIENT DESCENT



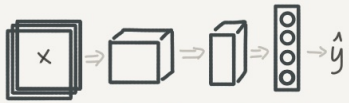


FASTER COMPUTATION IS IMPORTANT TO SPEED UP THE ITERATIVE PROCESS

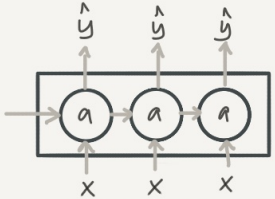
NETWORK ARCHITECTURES



STANDARD NN



CONVOLUTIONAL NN



RECURRENT NN

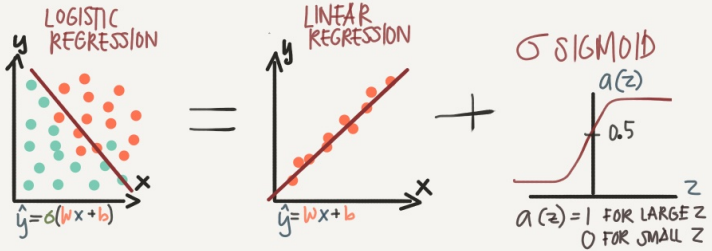
LOGISTIC REGRESSION

AS A NEURAL NET

BINARY CLASSIFICATION

1 : CAT
0 : NOT CAT

x y



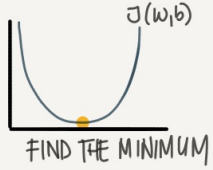
THE TASK IS TO LEARN w & b BUT HOW?

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

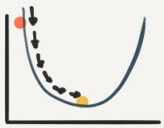
$$\text{LOSS} = \mathcal{L}(\hat{y}, y)$$

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

COST = LOSS FOR THE ENTIRE DATASET



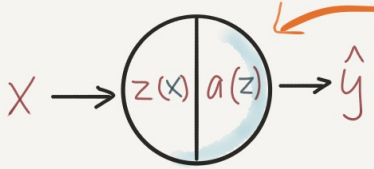
FINDING THE MINIMUM WITH GRADIENT DESCENT



1. FIND THE DOWNHILL DIRECTION (USING DERIVATIVES)
2. WALK (UPDATE w & b) AT A α LEARNING RATE

REPEAT UNTIL YOU REACH BOTTOM (CONVERGE)

PUTTING IT ALL TOGETHER



MINI NEURAL NET

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

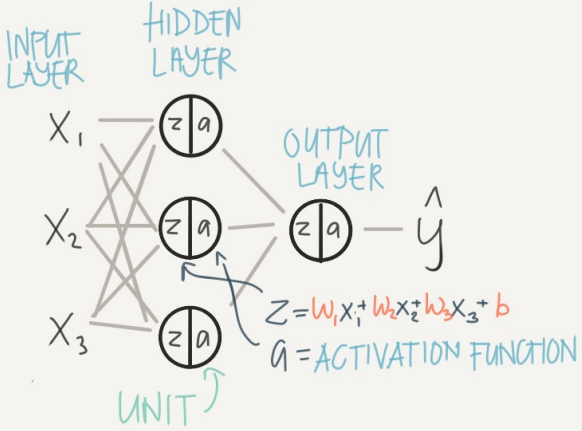
$$\hat{y} = a(z) = \sigma\,\text{SIGMOID}(z)$$

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

REPEAT UNTIL IT CONVERGES

SHALLOW NEURAL NETS

2 LAYER NEURAL NET



ACTIVATION FUNCTIONS

- SIGMOID σ: BINARY CLASSIFIER - ONLY USED FOR OUTPUT LAYER. SLOW GRAD DESCENT SINCE SLOPE IS SMALL FOR LARGE/SMALL VAL
- TANH: NORMALIZED ⇒ GRADIENT DESCENT IS FASTER
- RELU: DEFAULT CHOICE FOR ACTIVATION. SLOPE = 1/0
- LEAKY RELU: AVOIDS UNDEF SLOPE AT 0 BUT RARELY USED IN PRACTICE

WHY ACTIVATION FUNCTIONS?

EX. WITH NO ACTIVATION - $a = z$

$$a^{[1]} = z^{[1]} = w^{[1]}x + b^{[1]} \quad \text{LAYER 1}$$
$$a^{[2]} = z^{[2]} = w^{[2]}a^{[1]} + b^{[2]} \quad \text{LAYER 2}$$

PLUG IN $a^{[1]}$

$$a^{[2]} = w^{[2]}(w^{[1]}x + b^{[1]}) + b^{[2]}$$
$$= w^{[2]}w^{[1]}x + w^{[2]}b^{[1]} + b^{[2]}$$
$$w'x + b' \leftarrow \text{LINEAR FUNCTION}$$

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

INITIALIZING W+B

WHAT IF: INIT TO 0

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

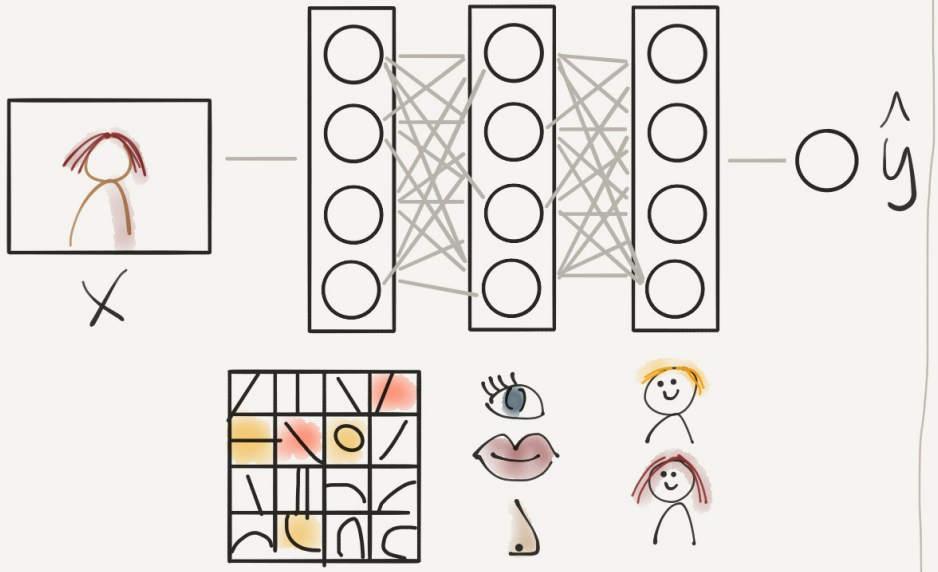
SOLUTION: RANDOM INIT

BUT ALSO WANT THEM SMALL SO RAND * 0.01 ← HYPERPARAM

DEEP
NEURAL NETS

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

WHY DEEP NEURAL NETS?



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
- # ITERATIONS
- # HIDDEN LAYERS
- # HIDDEN UNITS
- CHOICE OF ACTIVATION
- MOMENTUM
- MINI-BATCH SIZE
- REGULARIZATION

LOW LEVEL AUDIO WAVE FEATURES ↗↘ PITCH — PHONEMES — WORDS — SENTENCES

C A T

SETTING UP YOUR ML APP

CLASSIC ML

100 - 10000 SAMPLES

TRAIN	DEV	TEST
60%	20%	20%

ALL FROM SAME PLACE → DISTRIBUTION

DEEP LEARNING

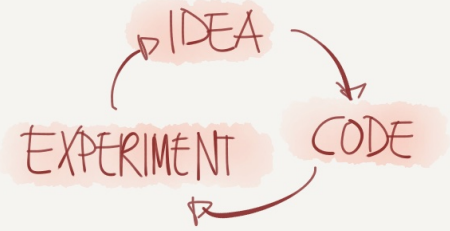
1M SAMPLES

TRAIN	D	T
98%	1%	1%

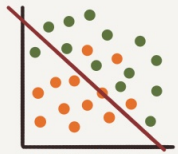
EX: TRAIN  PRO CAT PICS FROM INTERNET

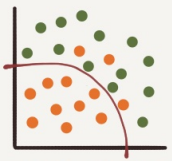
DEV/TEST  BLURRY CAT PICS FROM APP

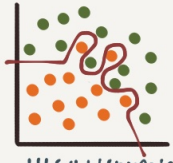
TIP
DEV & TEST SHOULD COME FROM SAME DISTRIBUTION



BIAS/VARIANCE


HIGH BIAS "UNDERFIT"

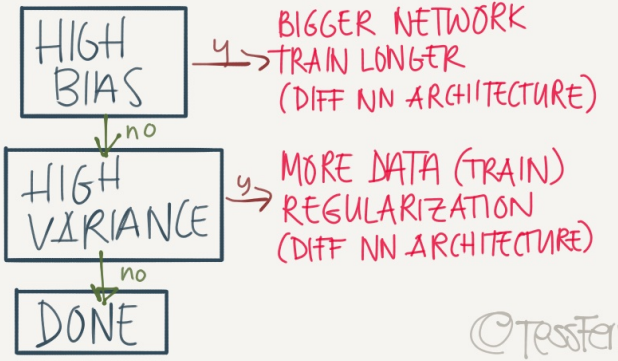

JUST RIGHT


HIGH VARIANCE "OVERFIT"

	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

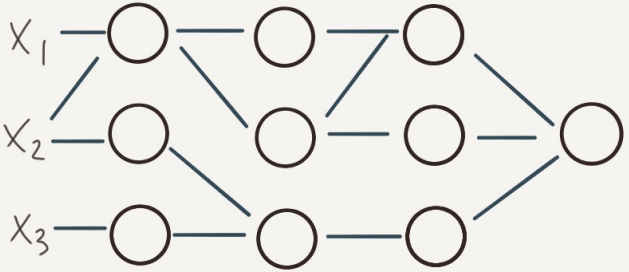
$$\text{COST: } J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y},y) + \frac{\lambda}{2m}\|w\|_2^2$$

EUCLIDEAN NORM

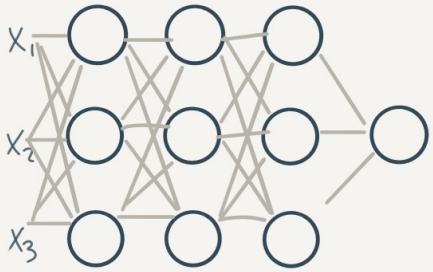
L1 REGULARIZATION

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

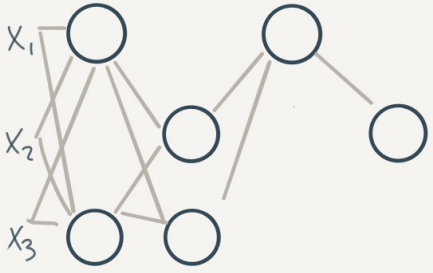
BOTH PENALIZE LARGE WEIGHTS ⇒ SOME WILL BE CLOSE TO Ø ⇒ SIMPLER NETWORKS



DROPOUT



FOR EACH ITERATION & SAMPLE SOME NODES ARE RANDOMLY DROPPED (BASED ON KEEP_PROB)

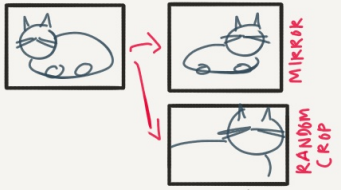


WE GET SIMPLER NWs & LESS CHANCE TO RELY ON SINGLE FEATURES

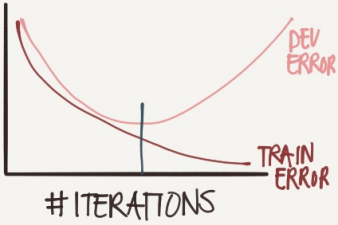
OTHER REGULARIZATION TECHNIQUES

DATA AUGMENTATION

GENERATE NEW PICS FROM EXISTING



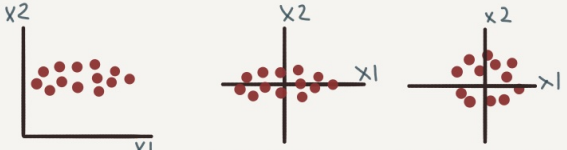
EARLY STOPPING



PROBLEM: AFFECTS BOTH BIAS & VARIANCE

OPTIMIZING TRAINING

NORMALIZING INPUTS



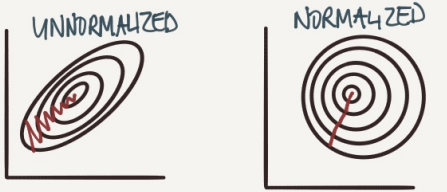
STEP1: CENTER AROUND 0,0

STEP2: SCALE SO VARIANCE IS SAME (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)

$$\hat{y} = W^{[L-1]} W^{[L-2]} \cdots W^{[1]} X + b$$

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

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

IN BOTH CASES GRADIENT DESCENT TAKES A VERY LONG TIME

PARTIAL SOLUTION: CHOOSE INITIAL VALUES CAREFULLY

$w^{[l]} = \text{rand} * \sqrt{\frac{2}{n^{[l-1]}}}$ (FOR RELU)

$n^{[l-1]}$ = #inputs

XAVIER $\sqrt{\frac{1}{n^{[l-1]}}}$ (FOR TANH)

SETS THE VARIANCE

GRADIENT CHECKING

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

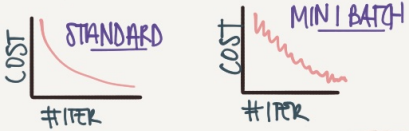
GRADIENT CHECKING APPROXIMATES THE GRADIENTS SO YOU CAN VERIFY CALC.

NOTE ONLY USE WHEN DEBUGGING SINCE IT'S SLOW

OPTIMIZATION ALGORITHMS

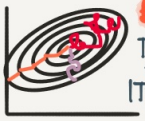
MINI-BATCH GRAD. DESCENT

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



CHOOSING THE MINIBATCH SIZE

SIZE = m → BATCH GRAD DESC.
SIZE = 1 → STOCHASTIC GRAD.DESC



BATCH — TOO LONG PER ITERATION
MIDDLE
STOCHASTIC — LOSE ALMOST ALL SPEED FROM VECTORIZATION

- SHORT ITER.
- USES VECT.

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 ↕ 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.

$$S_{dw} = \beta S_{dw} + (1-\beta)\, dw^2$$
$$S_{db} = \beta S_{db} + (1-\beta)\, db^2$$
$$w = w - \alpha \frac{dw}{\sqrt{S_{dw}}} \qquad b = b - \alpha \frac{db}{\sqrt{S_{db}}}$$

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: $\alpha = \frac{1}{1 + \text{DECAYRATE} \cdot \text{EPOCH}} \alpha_0$

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

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

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

OPTION 5: DISCRETE STAIRCASE

OPTION 6: MANUAL

EPOCH = 1 PASS THROUGH THE DATA

IMPROVING DEEP NEURAL NETS · COURSERA

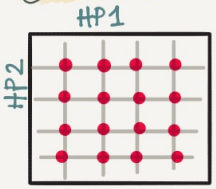
HYPERPARAM TUNING

WHICH HYPERPARAMS ARE MOST IMPORTANT?

- α LEARNING RATE
- # #HIDDEN UNITS
- MINIBATCH SIZE
- β MOMENTUM TERM = 0.9
- # LAYERS
- LEARNING RATE DECAY
- $\beta_1 = 0.9$ $\beta_2 = 0.999$ $\varepsilon = 10^{-8}$ (ADAM)

TESTING VALUES

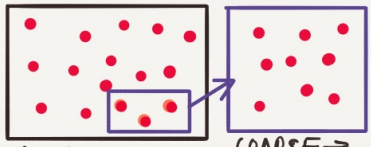
CLASSIC ML



GRID SEARCH

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

SOLUTION



RANDOM SEARCH + COARSE → DENSE

USE AN APPROPRIATE SCALE

HIDDEN UNITS

UNIFORMLY RANDOM

α LEARNING RATE

$r = -4 \cdot \text{rand}[-4, 0]$

$\alpha = 10^r$

β EXP WEIGHT AVE

$r = -3 \cdot \text{rand}[-3, -1]$

$\beta = (1 - 10^r)$

TIP

RE-EVALUATE YOUR HYP-PARAMS EVERY FEW MONTHS

PANDA VS CAVIAR

MY PANDA IS ACTUALLY A MISCLASSIFIED CAT BECAUSE I CAN'T DRAW PANDAS

BABYSIT ONE MODEL & TUNE

GOOD IF YOU HAVE LOTS OF SPARE COMP POWER

SPAWN LOTS OF MODELS W DIFF HP

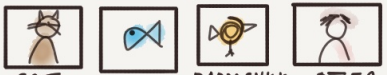
MISC. EXTRAS

BATCH NORMALIZATION

NORMALIZE LAYER OUTPUT

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

MULTICLASS CLASSIFIC.



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

SOFTMAX ACTIVATION

$$t = e^{(z^{[l]})}$$

$$a^{[l]} = \frac{t}{\sum t_i}$$

SUM: 1

EX: $z^{[l]} = \begin{bmatrix} 5 \\ 2 \\ -1 \\ 3 \end{bmatrix}$ $t = \begin{bmatrix} e^5 \\ e^2 \\ e^{-1} \\ e^3 \end{bmatrix} = \begin{bmatrix} 148.4 \\ 7.4 \\ 0.4 \\ 20.1 \end{bmatrix}$ 176.3

$\Rightarrow a^{[l]} = \frac{t}{176.3} = \begin{bmatrix} 0.842 \\ 0.042 \\ 0.02 \\ 0.114 \end{bmatrix}$ 11.4% PROB IT'S A BABY CHICK

IMPROVING DEEP NEURAL NETS · COURSERA

STRUCTURING YOUR ML PROJECTS

SETTING YOUR GOAL

★ GOAL SHOULD BE A SINGLE #

	PRECISION	RECALL
A	95%	90%
B	98%	85%

IS A OR B BEST?

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

A IS BEST

F1 = HARMONIC MEAN BETW. RECALL & PRECISION

★ DEFINE OPTIMIZING VS SATISFICING METRICS

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

MAXIMIZE ACC.
GIVEN TIME ≤ 100ms

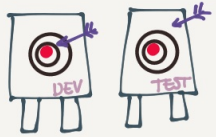
ACCURACY = OPTIMIZING
RUNTIME = SATISFICING

SELECTING YOUR DEV/TEST SETS

DATA

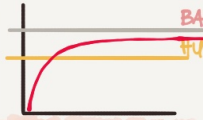
US
UK
~~EURDAE~~
S.AM
INDIA
CHINA
AUST.

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 IMG CLASS
TYPICAL HUMAN 3%
TYPICAL DOCTOR 1%
EXPERIENCED DR. 0.7%
TEAM OF EXP DRS. 0.5%
↑ HUMAN LEV PERF (PROXY FOR BAYES)

1. OFTEN CLOSE TO BAYES
2. A HUMAN CAN NO LONGER HELP IMPROVE (INSIGHTS)
3. DIFFICULT TO ANALYSE BIAS/VARIANCE

CAT CLASSIFICATION

	A	B (BLURRY)
HUMAN	1%	7.5%
TRAIN ERR	8%	8%
DEV ERR	10%	10%
	FOCUS ON BIAS	FOCUS ON VARIANCE

B: ← AVOIDABLE BIAS, ← VARIANCE

HUMAN
| AVOIDABLE BIAS
- TRAIN BIGGER NETW.
- TRAIN LONGER / BETTER OPT. (RMSProp & ADAM)
- CHANGE NN ARCH OR HYPERPARAMS

TRAIN
| VARIANCE
- MORE DATA (TRAIN)
- REGULARIZATION
- NN ARCHITECTURE

DEV

	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

STRUCTURING ML PROJECTS • COURSERA

ERROR ANALYSIS

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

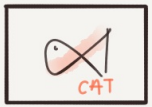
1. PICK 100 MIS-LABLED
2. COUNT ERROR REASONS

	DOG	BLURRY	INSTA FILTER	BIG CAT	...
1	1		1		
2				1	
3		1			
...					
100			1		
	5	...			

5% OF ALL ERRORS

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

YOU FIND SOME INCORR. LABLED 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 INCORR LABLED AS MICE)

ADD EXTRA COL. 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

STRUCTURING ML PROJECTS • COURSERA

TRAIN vs DEV/TEST MISMATCH

AVAILABLE DATA

200 k PRO CAT PICS FROM INTERNET

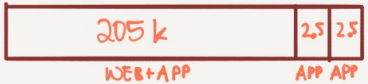
10 k BLURRY CAT PICS FROM APP (WHAT WE CARE ABT)

HOW DO WE SPLIT → TRAIN/DEV/TEST?

OPTION 1: SHUFFLE ALL

PROBLEM: DEV/TEST IS NOW MOSTLY WEB IMG (NOT REPRES. OF END SCENARIO)

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



BIAS & VARIANCE W MISMATCHED TRAIN/DEV

HUMANS ~0%

TRAIN 1%

DEV ERR 10%

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

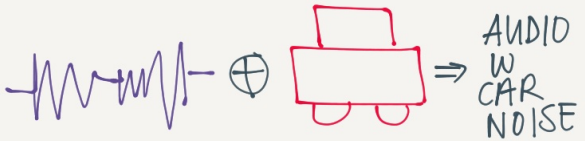
A: CREATE A TRAIN-DEV SET THAT WE DON'T TRAIN ON

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

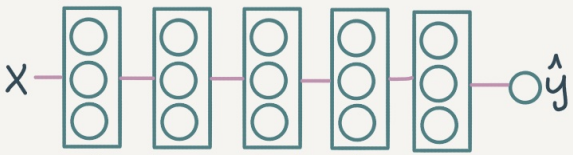


NOTE BE CAREFUL • IF YOU ONLY HAVE 1 HR OF CAR NOISE & APPLY IT TO 10K HR SPEECH YOU MAY OVERFIT TO THE CAR NOISE

EXTENDED LEARNING

TRANSFER LEARNING

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



NN TO CLASSIFY CATS/DOGS/CARS

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 IMG.

SOLUTION: INIT WITH WEIGHTS FROM CAT NN
RETRAIN ALL LAYERS

TESS NOTES

THIS IS MICROSOFT CUSTOM VISION

MULTI TASK LEARNING

TRAINING ON MULT. TASKS AT ONCE

DETECT
CAR
STOP SIGN
PEDESTR.
TRAFFIC LIGHT

$\hat{y}$

UNLIKE SOFMAX · MANY THINGS CAN BE TRUE

$$\text{COST: } J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{4} \mathcal{L}(\hat{y}_j^{(i)}, y_j^{(i)})$$

SUMMING OVER ALL OUTP OPTIONS

WE COULD HAVE JUST TRAINED 4 NNS INSTEAD BUT... MT LEARNING MAKES SENSE WHEN

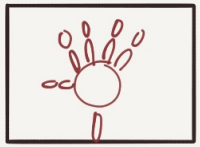
A. THE LEARNING DATA YOU HAVE FOR THE DIFF TASKS IS QUITE SIMILAR – & THE AMOUNTS (EG. 1K CARS, 1K STOP SIGNS)

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 CHILDS HAND TELL ME THE AGE OF THE CHILD



TYPICAL SOLN:

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

END-TO-END

RADIOLOGY IMG → CHILD AGE

PROS:

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

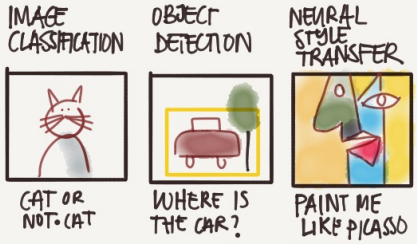
CONS:

- NEEDS LARGE AMTS OF LABELED DATA (X→Y)
- EXCLUDES POTENTIALLY USEFUL HAND-MADE COMPONENTS

CONVOLUTION
FUNDAMENTALS

CONVOLUTIONAL NEURAL NETS · COURSERA

COMPUTER VISION



PROBLEM: IMAGES CAN BE BIG

1000 × 1000 × 3 (RGB) = 3M

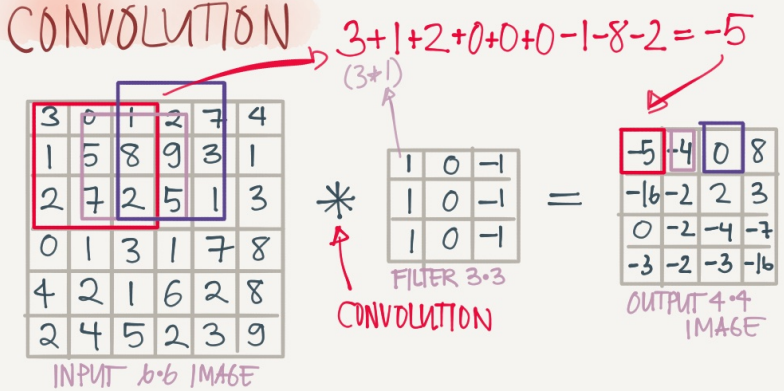
WITH 1000 HIDDEN UNITS WE NEED 3M * 1000 = 3B PARAMS

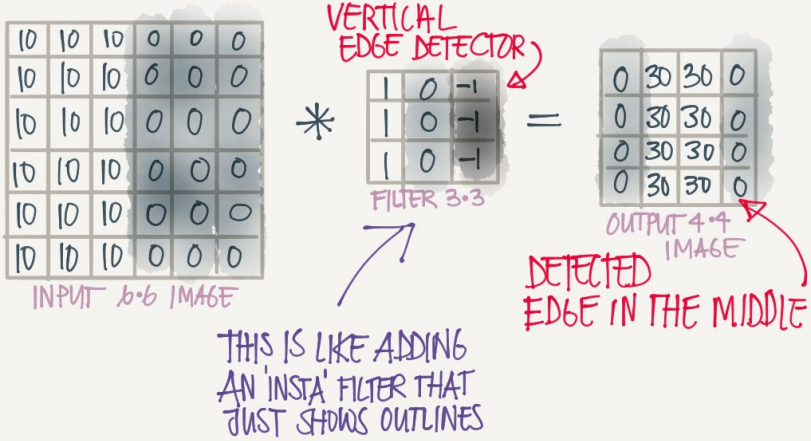
SOLUTION: USE CONVOLUTIONS

IT'S LIKE SCANNING OVER YOUR IMG WITH A MAGNIFYING GLASS OR FILTER

TESS NOTE: ALSO SOLVES THE PROBLEM THAT THE CAT IS NOT ALWAYS IN THE SAME LOCATION IN THE IMG

CONVOLUTION





WE 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

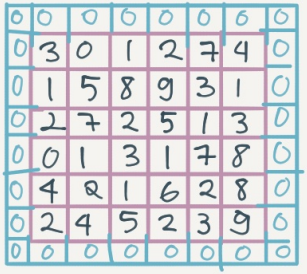
w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

CONVOLUTIONAL NEURAL NETS · COURSERA

PADDING

PROBLEM: IMAGES SHRINK
6x6 → 3x3 → 4x4
PROBLEM: EDGES GET LESS 'LOVE'

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



TWO COMMONLY USED PADDING OPTIONS
(HOW MUCH TO PAD)

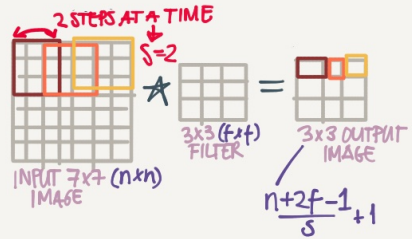
'VALID' ⇒ P = 0 — NO PADDING

'SAME' ⇒ P = $\frac{f-1}{2}$ — OUTPUT SIZE = INPUT SIZE

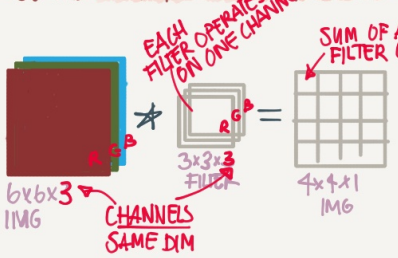
FILTER SIZE

STRIDE

WHAT PACE YOU SCAN WITH



CONVOLUTIONS OVER VOLUMES



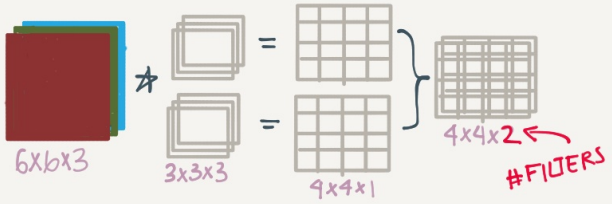
THIS ALLOWS US TO DETECT FEATURES IN COLOR IMAGES FOR EXAMPLE

MAYBE WE WANT TO FIND ALL EDGES OR MAYBE ORANGE BLOBS

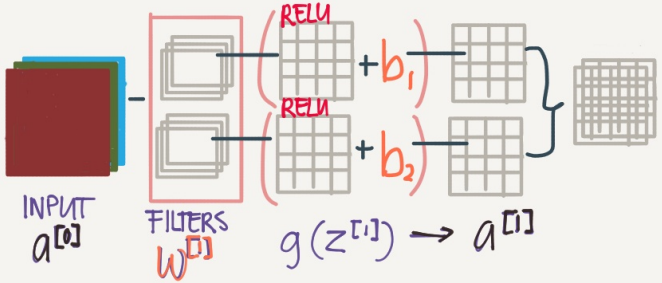
NOTE: ALL CONVOLUTION IDEAS CAN BE APPLIED TO 1D AS WELL LIKE EKG SIGNALS · AND 3D LIKE CT-SCANS

MULTIPLE FILTERS

DETECTING MULTIPLE FEATURES AT A TIME



ONE CONV. NET LAYER

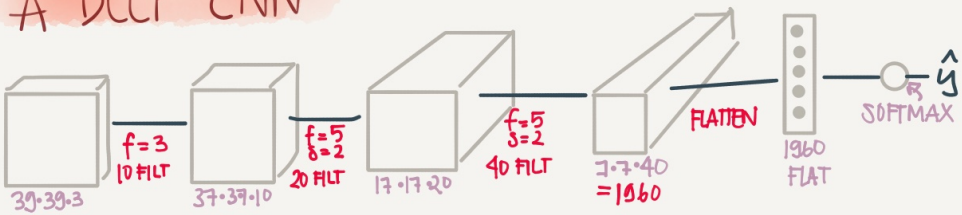


NOTE IT DOESN'T MATTER HOW BIG THE INPUT IS - THE LEARNABLE PARAMS W & b ONLY DEPEND ON THE # OF FILTERS AND THEIR SIZES.

$W = 3 \cdot 3 \cdot 3 \cdot 2 = 54$
$b = 2$
} 56 PARAMS TO LEARN

CONVOLUTIONAL NEURAL NETS · COURSERA

A DEEP CNN



A LOT OF THE WORK IS FIGURING OUT HYPERPARAMS
= #FILTERS, STRIDE, PADDING ETC

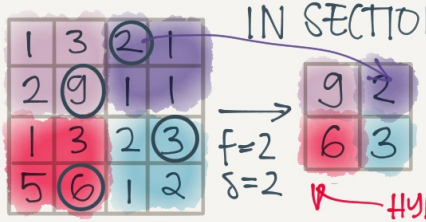
TYPICALLY SIZE → TREND DOWN
FILTERS → TREND UP

TYPICAL CONV.NET LAYERS

CONVOLUTION
POOLING
FULLY CONNECTED

POOLING (MAX)

FIND MAX VAL IN SECTION

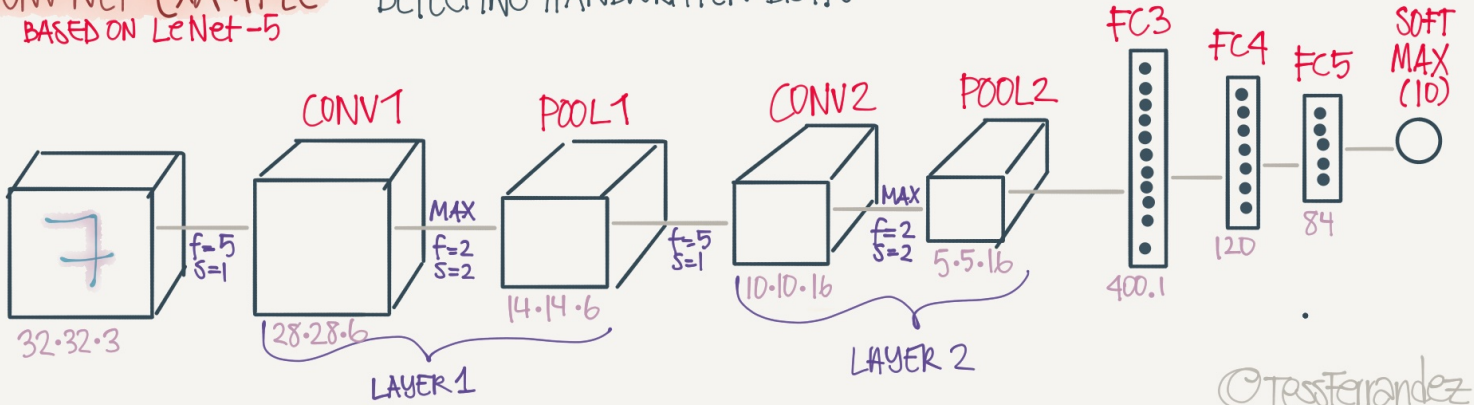


- REDUCES SIZE OF REPRES.
- SPEEDS UP COMPUTATION
- MAKES SOME OF THE DETECTED FEAT. MORE ROBUST

CONV NET EXAMPLE

BASED ON LeNet-5

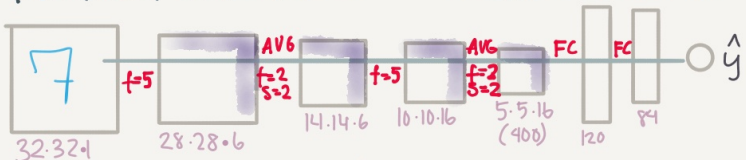
DETECTING HANDWRITTEN DIGITS



CLASSIC CONV. NETS

LeNet-5

DOCUMENT CLASSIFICATION ≈60k PARAMETERS



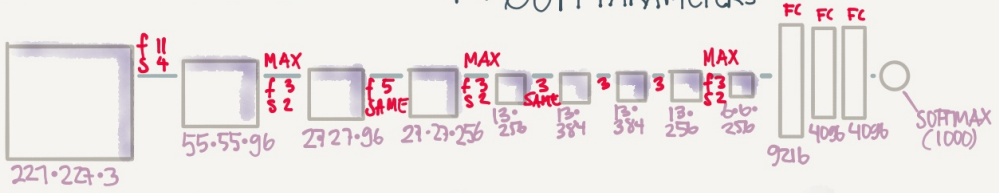
TRENDS: HEIGHT/WIDTH GO DOWN
CHANNELS GO UP

COMMON PATTERN: A COUPLE OF CONV(1+)/POOL 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

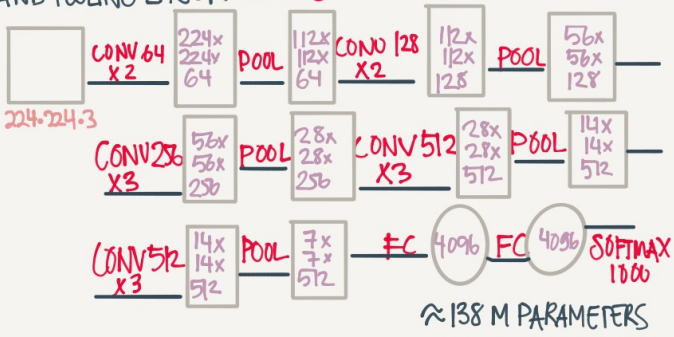
IMAGE CLASSIFICATION ≈ 60M PARAMETERS



- 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=3x3 S=1 P=SAME
AND POOLING LAYER 2x2 S=2



≈138 M PARAMETERS

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

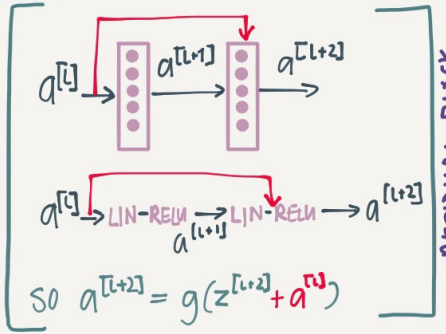
CONVOLUTIONAL NEURAL NETS · COURSERA

SPECIAL NETWORKS

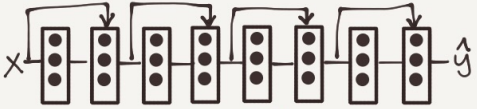
ResNets

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

SOLUTION: RESIDUAL NETS



So $a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$

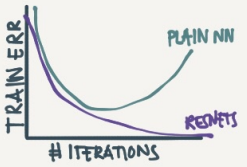


VERY DEEP NN

$a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$
$= (w^{[l+2]} a^{[l+1]} + b^{[l+1]} + a^{[l]})$

WORST CASE: w & b HAVE VANISHED TO $\emptyset$ ⇒ $a^{[l+2]} = g(a^{[l]}) = a^{[l]}$
SO EXTRA LAYERS LEARNED NOTHING BUT DIDN'T HURT

BEST CASE: LAYERS LEARNED SOMETHING



NETWORK IN NETWORK (1×1 CONVOLUTION)

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

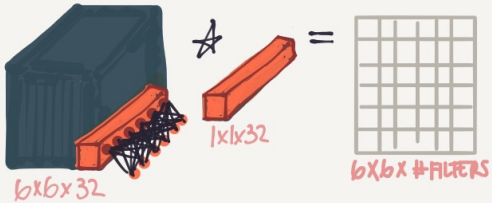
* [2] =

12	10	6	4
8	2	18	10
10	16	4	8
0	6	12	2

1×1 CONVOLUTION

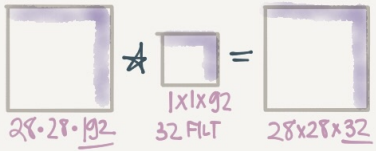
IT SEEMS PRETTY USELESS, BUT IT ACTUALLY SERVES 2 PURPOSES

1. NETWORK IN A NETWORK



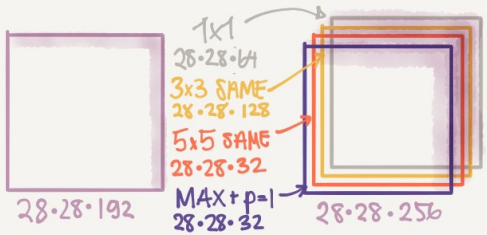
LEARNS COMPLEX, NON-LINEAR RELATIONSHIPS ABOUT A SLICE OF A VOLUME

2. REDUCING # CHANNELS



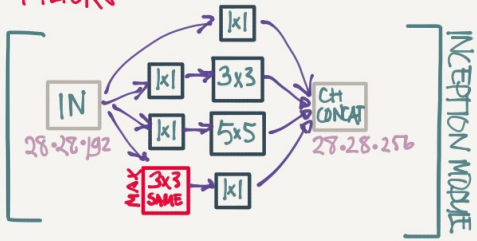
INCEPTION NETWORKS

INSTEAD OF CHOOSING A 1×1, 3×3, 5×5 OR A POOLING LAYER - CHOOSE ALL



PROBLEM: VERY EXPENSIVE TO COMPUTE

SOLUTION: SHRINK THE #CHANNELS W A 1×1 CONV BEFORE APPLYING ALL THE FILTERS



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

INCEPTION THE MOVIE

CONVOLUTIONAL NEURAL NETS · COURSERA

PRACTICAL ADVICE

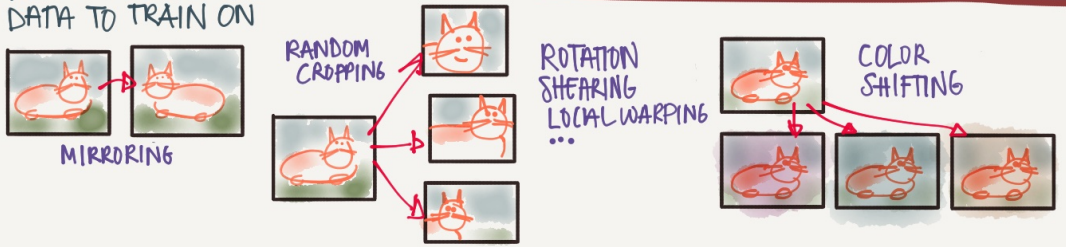
USE OPEN SOURCE IMPLEMENTATIONS

SOME OF THE PAPERS ARE HARD TO IMPLEMENT FROM SCRATCH - USING OS YOU CAN REUSE OTHER PPLS WORK

DON'T FORGET TO CONTRIBUTE

DATA AUGMENTATION

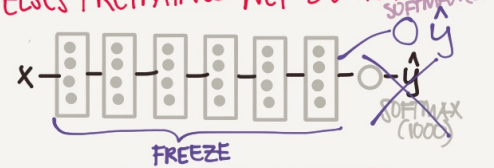
WE ALMOST ALWAYS NEED MORE DATA TO TRAIN ON



TRANSFER LEARNING

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

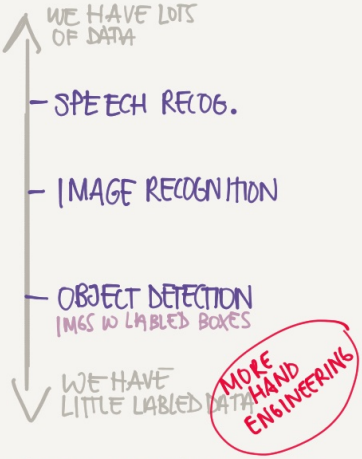
SOLUTION DOWNLOAD SOMEONE ELSES PRETRAINED NET & WEIGHTS



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

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

STATE OF COMPUTER VISION



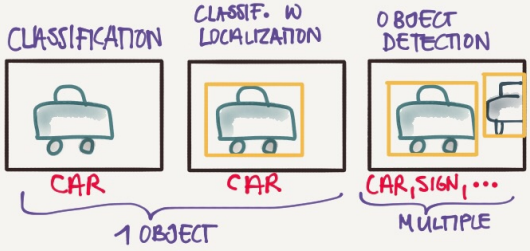
TIPS FOR DOING WELL ON BENCHMARKS/COMPETITIONS

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

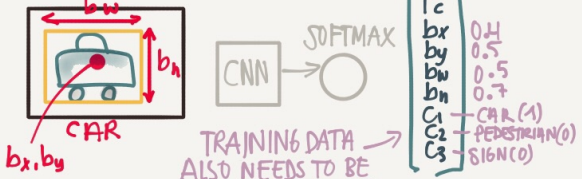
IN PRACTICE THEY ARE NOT USED IN PRODUCTION BECAUSE THEY ARE COMPUTE & MEM EXPENSIVE

CONVOLUTIONAL NEURAL NETS · COURSERA

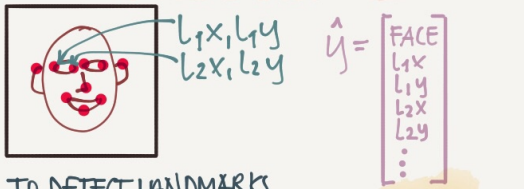
DETECTION ALGORITHMS



OBJECT LOCALIZATION



LANDMARK DETECTION



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

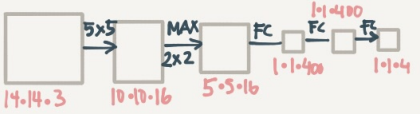
USED FOR SENTIMENT ANALYSIS & FOR EFFECTS LIKE PLACING CROWN ON HEAD ETC.

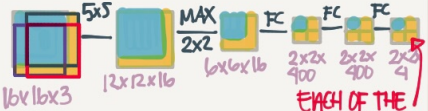
SLIDING WINDOWS DETECTION

1. CREATE TIGHTLY CROPPED IMG OF CARS (1'S)
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 W CONVOLUTIONS

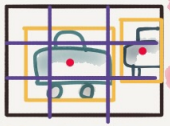




EACH OF THE 4 VALS ARE RESULTS FOR EACH OF THE 4 WINDOWS

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

YOLO: You Only Look Once



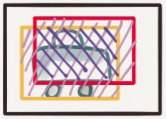
1. SPLIT IMG INTO X(9) GRID CELLS
2. FOR EACH CELL, SAY IF IT CONTAINS CAR + BOUNDING BOX (IF CELL CONTAINS THE MID POINT)

SAME AS OBJECT LOCALIZATION

IN PRACTICE WE MIGHT HAVE A 19×19 GRID

$X \in \hat{y} = 3 \times 3 \times 8$

HOW DO YOU KNOW HOW GOOD IT IS?



HOW GOOD IS THE RED SQUARE?

$IOU = \frac{\text{SIZE OF } \square}{\text{SIZE OF } \square}$

INTERSECTION OVER UNION

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

WHAT IF MULTIPLE SQUARES CLAIM THE SAME CAR?

NON-MAX SUPRESSION

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

CONVOLUTIONAL NEURAL NETS · COURSERA

FACE RECOGNITION

FACE VERIFICATION

IS THIS PETE?

99% ACC ⇒ PRETTY GOOD

FACE RECOGNITION

WHO IS THIS? (OUT OF K PERSONS)

IF K = 100 NEED MUCH HIGHER THAN 99%

ONE-SHOT LEARNING

NEED TO BE ABLE TO RECOGNIZE A PERSON EVEN THOUGH YOU ONLY HAVE ONE SAMPLE IN YOUR DB.

YOU CAN'T TRAIN A CNN WITH A SOFTMAX (EACH PERSON) BECAUSE

(A) YOU DON'T HAVE ENOUGH SAMPLES

(B) IF A NEW PERSON JOINS YOU NEED TO RETRAIN THE NETWORK

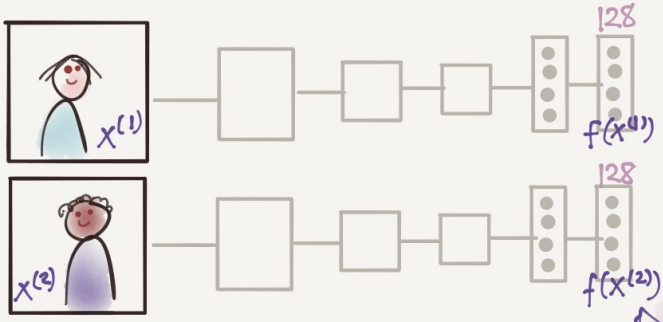
SOLUTION LEARN A SIMILARITY FUNCTION

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

BUT HOW DO YOU LEARN THIS?

SIAMESE NETWORK

DeepFace



LEARN THE PARAMS OF THE NN SUCH THAT

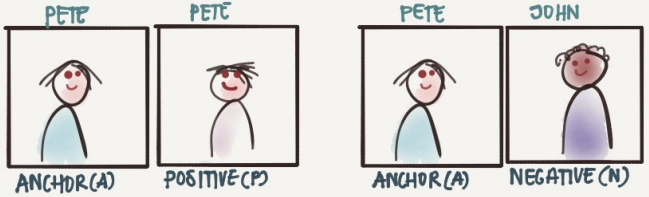
- IF $x^{(i)}, x^{(j)}$ ARE THE SAME PERSON · $d(x^i, x^j) \Rightarrow$ SMALL
- IF $x^{(i)}, x^{(j)}$ ARE DIFFERENT PEOPLE · $d(x^i, x^j) \Rightarrow$ LARGE

$$d(x^{(1)}, x^{(2)}) = \| f(x^{(1)}) - f(x^{(2)}) \|_2^2$$

WE CAN ACCOMPLISH THIS WITH THE TRIPLET LOSS FUNCTION

TRIPLET LOSS

FaceNet



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

BUT WE WANT A GOOD MARGIN, SO...

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

TIP

PRECOMPUTE ENCODINGS FOR PPL IN YOUR DB, SO YOU DONT HAVE TO SAVE IMAGES & COMPUTE ENCODINGS AT RUN-TIME

HOW DO WE CHOOSE TRIPLETS TO TRAIN ON?

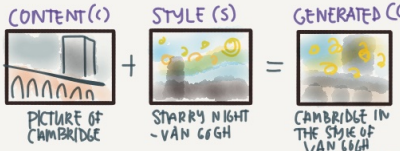
- IF A/P ARE VERY SIMILAR, & A/N ARE VERY DIFFERENT TRAINING IS VERY EASY.

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

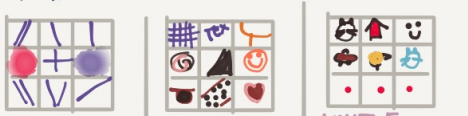
SOME BIG COMPANIES HAVE ALREADY TRAINED NETWORKS ON LARGE AMTS OF PHOTOS SO YOU MAY JUST WANT TO REUSE THEIR WEIGHTS

CONVOLUTIONAL NEURAL NETS · COURSERA

NEURAL STYLE TRANSFER



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



LAYER 1 · SIMPLE FEATURES LIKE LINES/COLORS

LAYER 3 · MORE COMPLEX FEATURES · BIRD LEGS, HONEY COMB PATTERNS

LAYER 5 · EVEN MORE COMPLEX · MAYBE ONE NODE IS A CAT DETECTOR

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

IDEA:

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

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

HOW SIMILAR ARE C & G — 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 ⇒ COPIES LARGER FEATURES
- LET $a^{[l](C)}$ & $a^{[l](G)}$ BE THE ACTIVATIONS
- IF $a^{[l](C)}$ & $a^{[l](G)}$ ARE SIMILAR THEY HAVE SIMILAR CONTENT
  BECAUSE THEY BOTH TRIGGER THE SAME HIDDEN UNITS

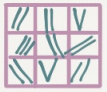
HOW DO WE TELL IF THEY ARE SIMILAR?

$$J_{CONTENT}(C,G) = \frac{1}{2} \| a^{[l](C)} - a^{[l](G)} \|^2$$

CAPTURING THE STYLE

USING THE STYLE IMG 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

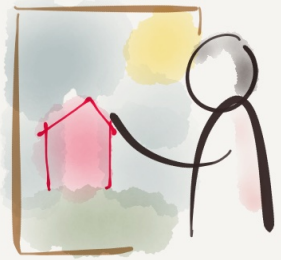
$$G_{kk'} = \sum_{i=1}^{n_H} \sum_{j=1}^{n_W} a_{ijk} \cdot a_{ijk'}$$

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 MODELS · COURSERA

SEQUENCE PROBLEMS

IN	OUT	PURPOSE
	THE QUICK BROWN FOX JUMPED...	SPEECH RECOGNITION
∅		MUSIC GENERATION
THERE IS NOTHING TO LIKE IN THIS MOVIE	★ ★ ★ ★ ★	SENTIMENT CLASSIFICATION
AGCCCCTGTG AGGAACTAG	AGCCCCTGTG AGGAACTAG	DNA SEQUENCE ANALYSIS
Voulez-vous chanter avec moi?	Do you want to sing with me?	MACHINE TRANSLATION
	RUNNING	VIDEO ACTIVITY RECOGNITION
Yesterday Harry Potter met Hermoine Granger	Yesterday Harry Potter met Hermoine Granger	NAME ENTITY RECOGNITION

NAME ENTITY RECOGNITION

X = HARRY ($x^{<1>}$) POTTER ($x^{<2>}$) AND HERMOINE ... $T_x = 9$ (9 WORDS)

GRANGER INVENTED A NEW SPELL

y = 1 ($y^{<1>}$) 1 ($y^{<2>}$) 0 ... 1 $T_y = T_x$

1 0 0 0 0

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)

a	1
aaron	2
and	367
harry	4075
potter	6830
zulu	10000

EACH WORD IS A **ONE-HOT** VECTOR

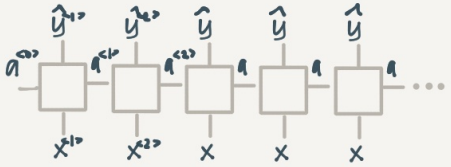
HARRY = [0 ... 1 (4075) ... 0]

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.

$$a^{<1>} = g_1(W_a[a^{<0>}, x^{<1>}] + b_a) \quad \text{TANH/RELU}$$

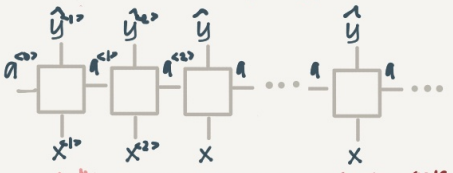
$$\hat{y}^{<1>} = g_2(W_{ya}\, a^{<1>} + b_y) \quad \text{SIGMOID}$$

THE SAME W & b ARE USED IN ALL TIME STEPS

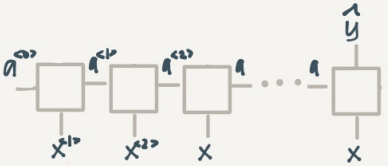
THE LOSS WE OPTIMIZE IS THE SUM OF $\mathcal{L}(\hat{y}, y)$ FROM 1–T

DIFFERENT TYPES OF RNN

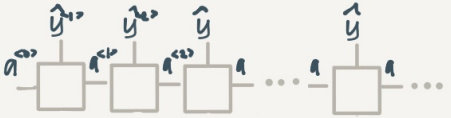
MANY-TO-MANY $T_x = T_y$



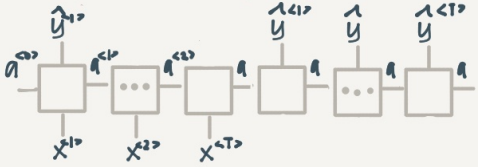
MANY-TO-ONE EX. SENTIMENT ANALYSIS



ONE-TO-MANY · MUSIC GENERATION



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



MORE ON RNNs

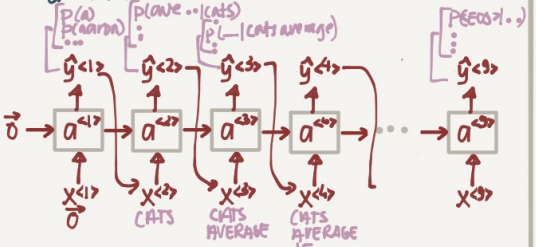
SEQUENCE MODELS · COURSERA

LANGUAGE MODELLING

HOW DO YOU KNOW IF SOMEONE SAID
THE APPLE AND PAIR SALAD OR
THE APPLE AND PEAR SALAD?

THE PURPOSE OF A LANG. MODEL IS TO CALCULATE THE PROBABILITIES

EX. CATS AVERAGE 15 HOURS OF SLEEP A DAY



SO GIVEN: CATS AVERAGE 15 WHAT IS THE PROB. THE NEXT WORD IS HOURS?

SAMPLING SENTENCES

1. TRAIN ON ALL HARRY POTTER BOOKS.
2. RANDOMLY SELECT A WORD (ON OF THE TOP WORDS) (EX. THE)
3. PASS THIS INTO THE NEXT TIMESTAMP AND SAMPLE A NEW WORD
4. REPEAT UNTIL X WORDS OR YOU REACHED <EOS>

CAN DO AT CHARACTERLEVEL AS WELL

YAY! YOU ARE NOW YOUR OWN J.K. ROWLING

VANISHING GRADIENTS

THE CAT, WHO ALREADY ATE APPLES AND ORANGES AND A FEW MORE THINGS BLA BLA WAS FULL
THE CATS, WHO ALREADY ATE ... WERE FULL
...

NEED TO REMEMBER SING/PLURAL FOR A LONG TIME

SINCE LONG SENTENCE ⇒ DEEP RNN WE GET THE VANISHING GRADIENTS PROB WE HAVE IN STANDARD NNs - I.E THE GRADIENTS FOR CAT/CATS HAVE LITTLE OR NO EFFECT ON WAS/WERE.

NOTE SOMETIMES YOU SEE EXPLODING GRAD (AS OVERFLOW NaN) BUT THIS IS EASILY FIXED WITH GRADIENT CLIPPING

GATED RECURRENT UNIT GRU

HELPS RECALL IF CAT WAS SING. OR PLURAL

THE GRU ACTS AS A MEMORY — AT EVERY TIMESTEP IT CALCULATES A NEW $\tilde{c}$ TO STORE AND A GATE Γ_u DECIDES TO UPDATE c TO $\tilde{c}$ OR NOT

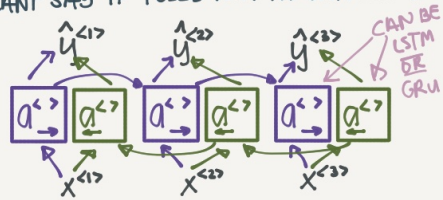
LONG SHORT TERM MEMORY (LSTM)

THE LSTM IS A VARIATION ON THE SAME THEME AS GRU BUT WITH AN ADDITIONAL Γ_F FORGET GATE

BI-DIRECTIONAL RNNs (BRNN)

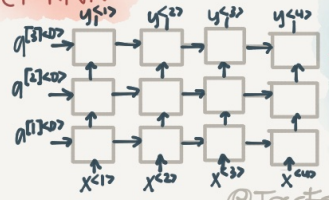
HE SAID, 'TEDDY BEARS ARE ON SALE'
HE SAID, 'TEDDY ROOSEVELT WAS A GREAT PRESIDENT'

PROBLEM: WITHOUT LOOKING FORWARD WE CANT SAY IF TEDDY IS A TOY OR A NAME



ONE DISADVANTAGE IS THAT YOU NEED THE FULL SENTENCE BEFORE YOU BEGIN - SO NOT SUITABLE FOR LIVE SPEECH RECO

DEEP RNN



SINCE THEY ARE ALREADY TEMORALLY DEEP THEY USUALLY DON'T HAVE A LOT OF LAYERS

SEQUENCE MODELS · COURSERA

NLP & WORD EMBEDDINGS

MAN IS TO WOMAN AS KING IS TO QUEEN

PROBLEM: THE ONE-HOT REPR O_{4856} OF APPLE HAS NO INFO ABOUT ITS RELATIONSHIP TO O_{6257} ORANGE

I WANT A GLASS OF ORANGE ____
I WANT A GLASS OF APPLE ____

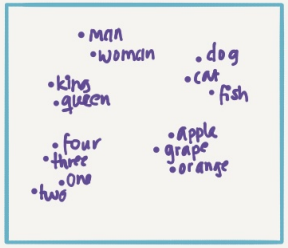
SOLUTION: CREATE A MATRIX OF FEATURES TO DESCRIBE THE WORDS

WORD EMBEDDINGS

	MAN 5391	WOMAN 9853	KING 4914	QUEEN 7157	APPLE 456	ORANGE 6257
GENDER	-1	1	-0.95	0.97	0.00	0.01
ROYAL	0.01	0.02	0.93	0.95	-0.01	0.00
AGE	0.03	0.02	0.7	0.69	0.03	-0.02
FOOD	0.04	0.01	0.02	0.01	0.95	0.97
⋮						

e_{5391}

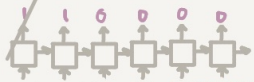
IN REALITY · THE FEATURES ARE LEARNED & NOT AS STRAIGHTFWD AS GENDER/AGE



t-SNE VISUAL REPRESENT OF 300D WORD EMBEDDINGS

USING WORD EMBEDDINGS

EX. NAME/ENTITY RECOGN



SALLY SMITH	IS	AN ORANGE FARMER
ROBERT LIN	IS	AN APPLE FARMER
BOB JONES	IS	A DURIAN CULTIVATOR

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

- APPLE ~ ORANGE ⇒ APPLE FARMER = PERSON
- USING WORD EMBEDDINGS TRAINED ON LOTS OF TEXT WE ALSO GET EMB. FOR MORE UNCOMMON WORDS (DURIAN, CULTIVATOR)

EX. MAN IS TO WOMAN AS KING IS TO ?

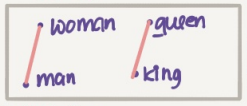
E = EMBEDDING MATRIX

	MAN 5391	WOMAN 9853	KING 4914	QUEEN 7157
GENDER	-1	1	-0.95	0.97
ROYAL	0.01	0.02	0.93	0.95
AGE	0.03	0.02	0.7	0.69
FOOD	0.04	0.01	0.02	0.01

e_{man}

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

$\begin{bmatrix}-2\\0\\0\\0\end{bmatrix}$ VERY SIMILAR $\begin{bmatrix}-2\\0\\0\\0\end{bmatrix}$



FIND MAN→WOMAN KING→ ?

$FIND(w): \arg\max_w SIM(e_{w}, e_{KING} - e_{MAN} + e_{WOMAN})$

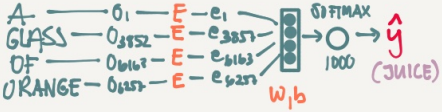
$SIM(u,v) = \frac{u^T v}{\|u\|_2 \|v\|_2}$ COSINE SIMILARITY

LEARNING WORD EMBEDDINGS

HOW DO WE LEARN THE EMBEDDING MATRIX E?

IDEA 1: USING A NEURAL LANG MODEL

I WANT A GLASS OF ORANGE ŷ



WE CAN USE DIFFERENT CONTEXTS THAN THE LAST 4 WORDS

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

RANDOM WITHIN EX 5 WORDS

IDEA 2: SKIP-GRAMS 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	MY
...	...

$O_c \to E \to e_c \to$ SOFTMAX (1000) $\to \hat{y}(O_T)$

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

1. PICK A CONTEXT/TARGET PAIR AS A POSITIVE EXAMPLE
2. PICK A FEW NEG EXAMPLES CONTEXT + RANDOM

CONTEXT	WORD	TARGET
ORANGE	JUICE	1
ORANGE	KING	0
ORANGE	BOOK	0
ORANGE	THE	0
ORANGE	OF	0

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

$O_c \to E \to e_c \to$ 10000 BIN. CLASSIFIERS (... JUICE ... KING ...)

YOU ONLY TRAIN 5 FOR EACH CONTEXT WORD ⇒ EFFICIENT TO TRAIN

SEQUENCE MODELS · COURSERA

WORD EMBEDDINGS CONTINUED...

GloVe WORD VECTORS

X_{ij} = #TIMES WORD i (TARGET t) APPEARS IN THE CONTEXT OF j (CONTEXT c)
(HOW RELATED THEY ARE)

$$\text{MINIMIZE} \sum_{i=1}^{10k} \sum_{j=1}^{10k} f(X_{ij})(\theta_i^T e_j + b_i + b_j' - \log X_{ij})^2$$

$f(X_{ij})$: 0 IF NO CONTEXT
(ALSO HELPS WEIGHING VERY FREQ WORDS (THE, OF...) & VERY INFREQUENT (DURIAN))

EVERYTHING LED UP TO THIS VERY SIMPLE ALGORITHM

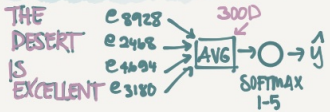
SENTIMENT CLASSIFICATION

X	y
THE DESSERT IS EXCELLENT	★★★★
SERVICE WAS QUITE SLOW	★★
GOOD FOR A QUICK MEAL BUT NOTHING SPECIAL	★★★
COMPLETELY LACKING IN GOOD TASTE GOOD SERVICE AND GOOD AMBIENCE	★

PROBLEM: YOU MAY NOT HAVE A LARGE DATASET

BUT YOU CAN USE AN EMBEDDING MATRIX E THAT IS ALREADY PRE-TRAINED

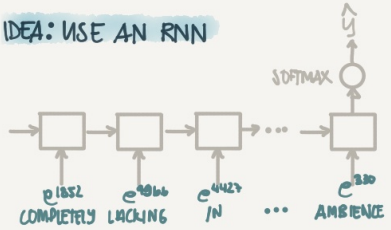
IDEA: SIMPLE CLASSIFICATION



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 +++ REVIEW

IDEA: USE AN RNN



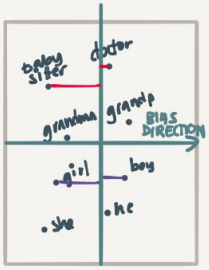
THIS CAN NOW TAKE INTO ACCOUNT THAT COMPLETELY LACKING NEGATES THE WORD GOOD

ELIMINATING BIAS IN WORD EMBEDDINGS

MAN IS TO COMPUTER PROGRAMMER AS WOMAN IS TO HOME MAKER

SOMETIMES THE TEXT CONTAINS & ALGOS 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
   $e_{he} \rightarrow e_{she}$
   $e_{male} \rightarrow e_{female}$
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/BOY 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

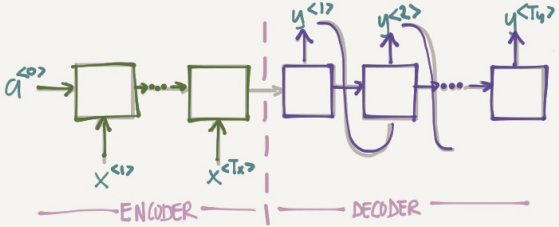
TURNS OUT THE # OF PAIRS IS FAIRLY SMALL SO YOU CAN EVE HAND PICK THEM

SEQUENCE MODELS · COURSERA

SEQUENCE TO SEQUENCE

BASIC MODELS

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



 → THIS IS A CAT ON A CHAIR

CNN → RNN

HOW DO YOU PICK THE MOST LIKELY SENTENCE?

$$P(y^{\langle 1\rangle}, \dots, y^{\langle Ty\rangle} | x)$$ (ENGLISH / FRENCH)

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

$$\text{ARG MAX } P(y^{\langle 1\rangle}, \dots, y^{\langle Ty\rangle} | x)$$

IDEA: USE GREEDY SEARCH

1. PICK THE WORD WITH THE BEST PROBABILITY
2. REPEAT UNTIL <EOS>

WITH THIS WE COULD GET
- JANE IS GOING TO BE VISITING AFRICA THIS SEPTEMBER

INSTEAD OF
- JANE IS VISITING AFRICA THIS SEPTEMBER

SOLUTION: OPTIMISE 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 W B PAIRS
   $P(y^{\langle 1\rangle}, y^{\langle 2\rangle} | x) = P(y^{\langle 1\rangle} | x)\, P(y^{\langle 2\rangle} | x, y^{\langle 1\rangle})$

(IN SEPTEMBER, JANE IS, JANE VISITS)

NOTE: KEEP TRACK OF THE P FOR THE SENTENCES OF EACH LENGTH - AFTER X ITER (X MAX WORDS) PICK THE BEST

3. REPEAT TIL DONE

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

PROBLEM: MULTIPLYING PROBABILITIES (0<P<1) RESULTS IN A VERY SMALL NUMBER → OVERFLOWS

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

$$\frac{1}{Ty^{\alpha}} \sum_{t=1}^{Ty} \log P(y^{\langle t\rangle} | x, y^{\langle 1\rangle}, \dots, y^{\langle t-1\rangle})$$

HOW DO WE PICK B?

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

IN PROD YOU MIGHT SE B=10.
100 IS PROBABLY A BIT TOO HIGH - BUT ITS DOMAIN DEPENDENT

ERROR ANALYSIS IN BEAM S.

HUMAN: JANE VISITS AFRICA IN SEPT... y^*
ALGO: JANE VISITED AFRICA LAST SEPTEMBER $\hat{y}$

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

LET THE RNN GIVE $Py^* = P(y^*, x)$ & $P\hat{y} = P(\hat{y}, x)$

IF $Py^* > P\hat{y}$:
BEAM PICKED THE WRONG ONE
TRY A HIGHER B

ELSE:
THE RNN PICKED THE WRONG PROBS · SO FOCUS ON THE RNN

SEQUENCE MODELS · COURSERA

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 (MT) APPEAR IN THE REAL TRANSLATION

THE THE THE THE THE THE THE
SCORE: 7/7

IDEA: ONLY GIVE CREDIT FOR A WORD THE MAX # TIMES IT APPEARS IN A TARGET SENTENCE
SCORE: 2/7 ← COUNT CLIP

THE CAT THE CAT ON THE MAT

	COUNT	COUNT CLIP
THE CAT	2	1
CAT THE	1	0
CAT ON	1	1
ON THE	1	1
THE MAT	1	1
BIGRAMS	6	4/6

BI-GRAM SCORE: 4/6

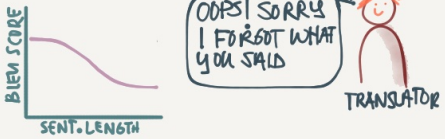
COMBINED BLEU SCORE

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

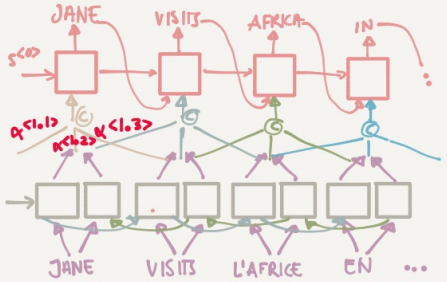
P_1 = SCORE SINGLE WORD
P_2 = SCORE BIGRAMS
...
BP = BREVITY PENALTY ← PENALIZES SENTENCES SHORTER THAN THE TARGET

↑ A USEFUL SINGLE NUMBER EVAL METRIC

ATTENTION MODEL



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

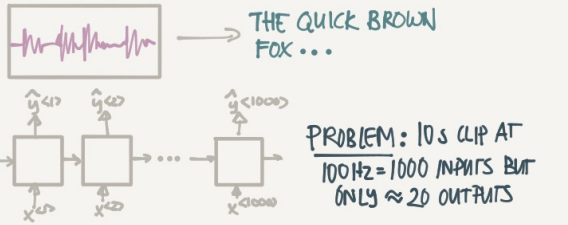


$\alpha^{<t,t'>}$ = HOW MUCH ATTENTION $y^{<t>}$ SHOULD PAY TO $a^{<t'>}$

$c^{<2>} = \sum_{t'} \alpha^{<2,t'>} \cdot a^{<t'>} \quad | \quad \sum_{t'} \alpha^{<1,t'>} = 1$

α IS CALCULATED USING A SMALL NEURAL NETWORK

SPEECH RECOGNITION



SOLUTION USE CTC COST (CONNECTION TEMPORAL CLASSIFICATION)

ttt_h_eee___u___qqq__ → COLLAPSE REPEATED CHARS NOT SEP BY BLANK

TRIGGER WORD DETECTION