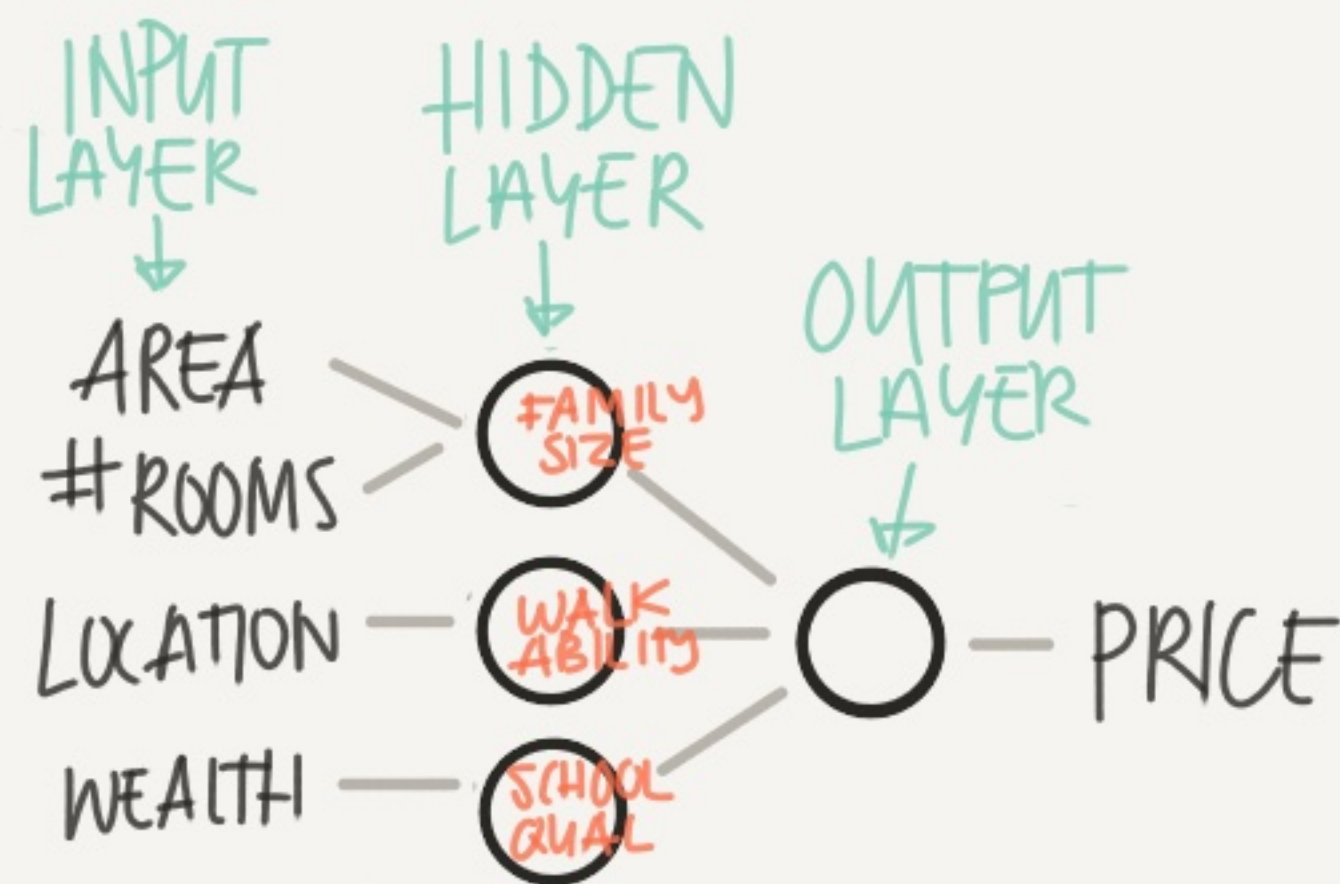
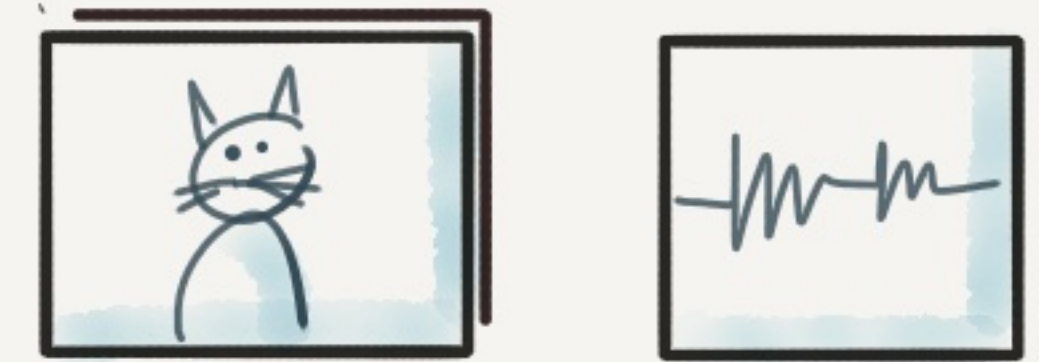
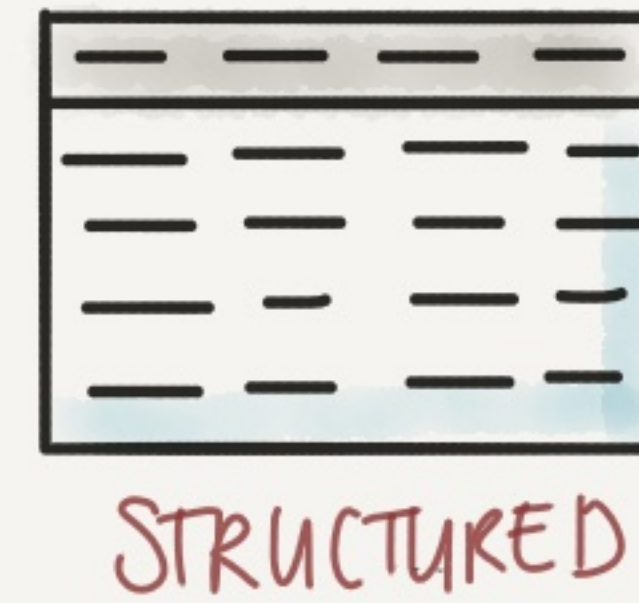


INTRO TO DEEP LEARNING



NNs CAN DEAL WITH BOTH STRUCTURED & UNSTRUCTURED DATA



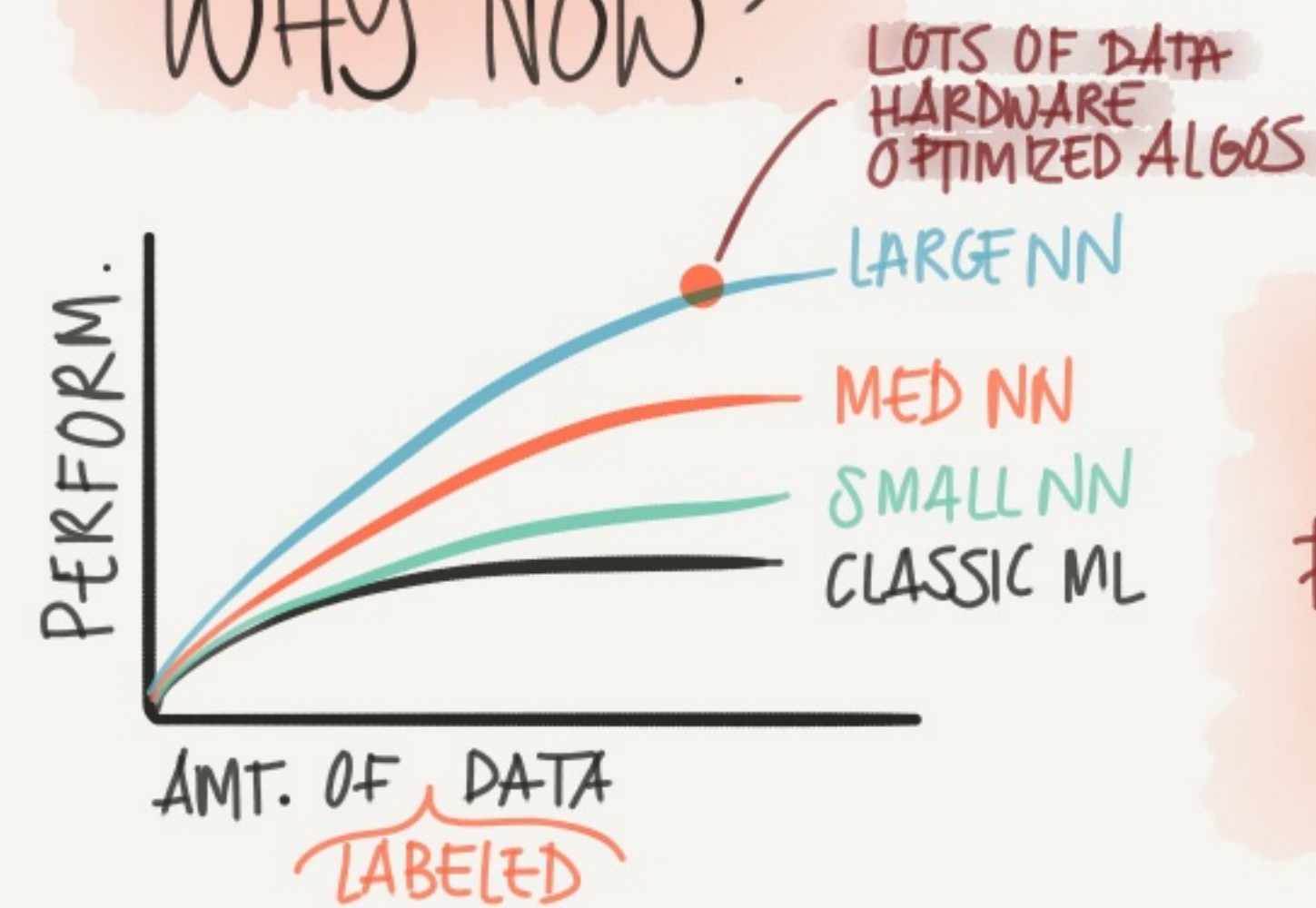
"THE QUICK BROWN FOX"
UNSTRUCTURED

HUMANS ARE GOOD AT THIS

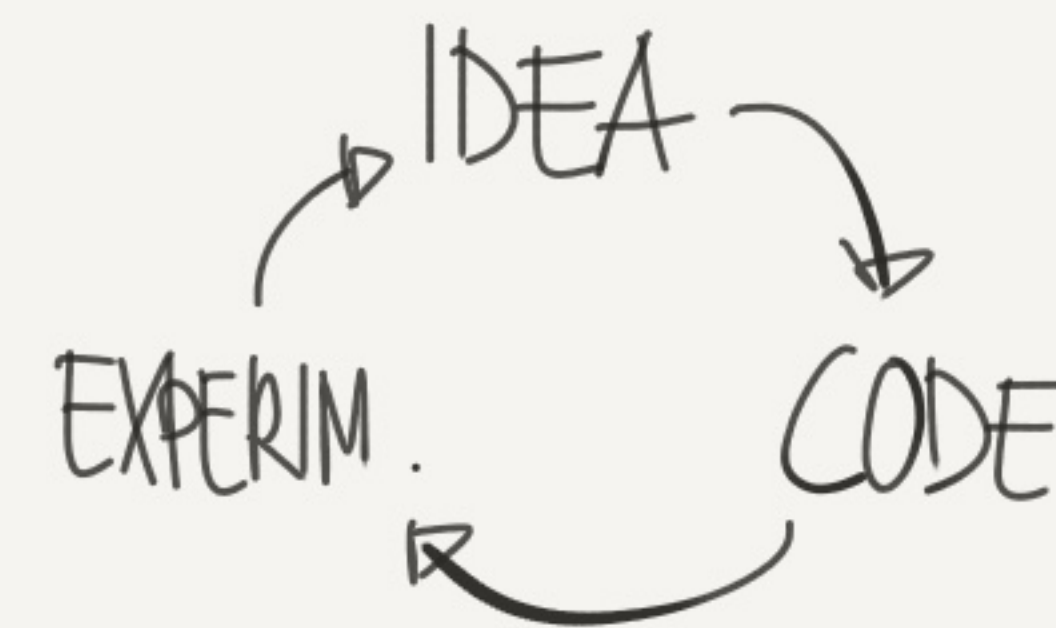
SUPERVISED LEARNING

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

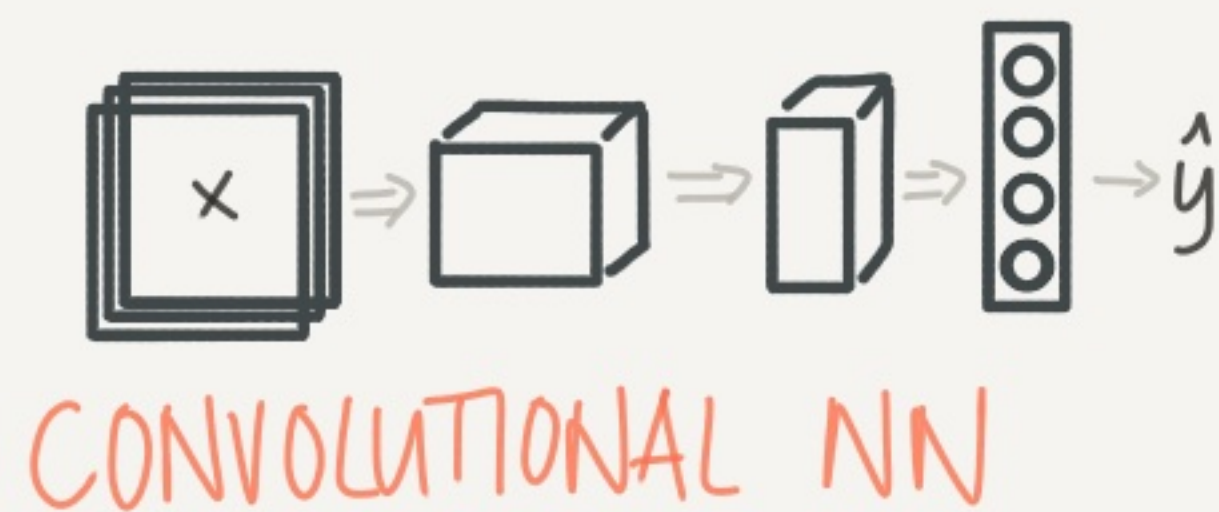
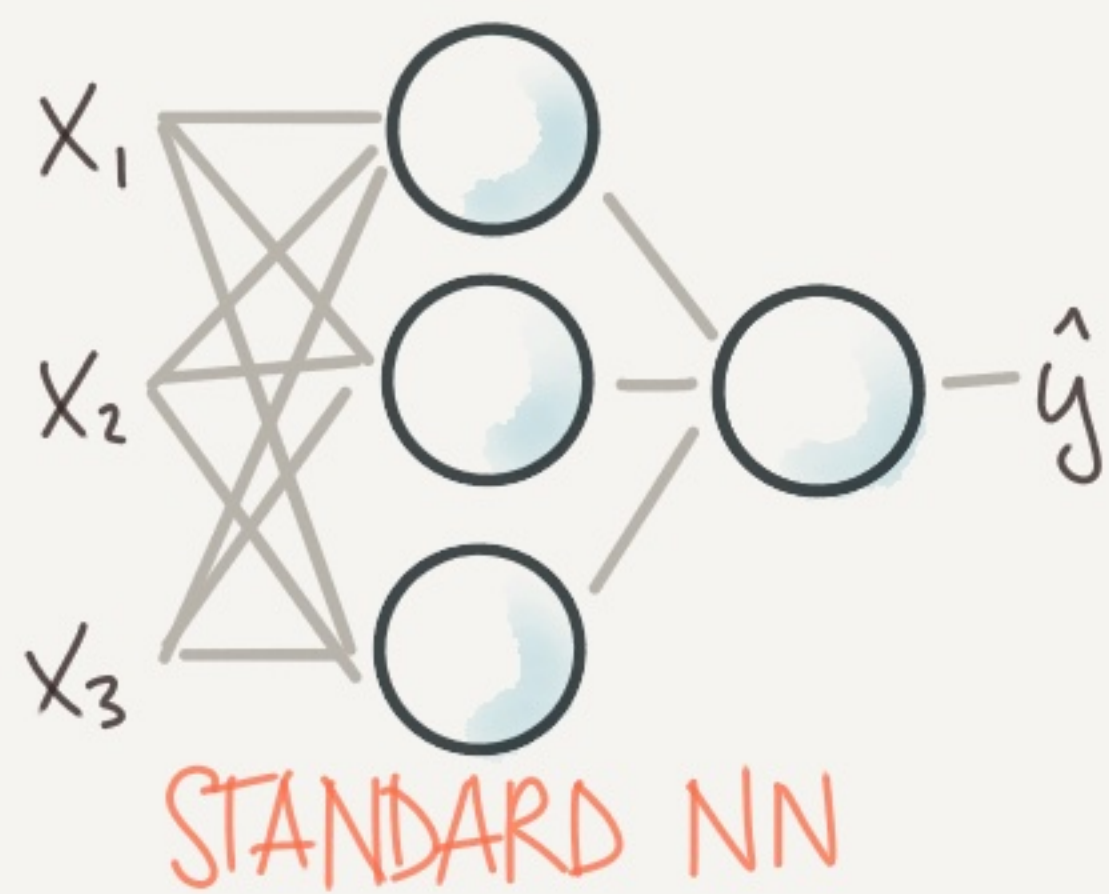
WHY NOW?



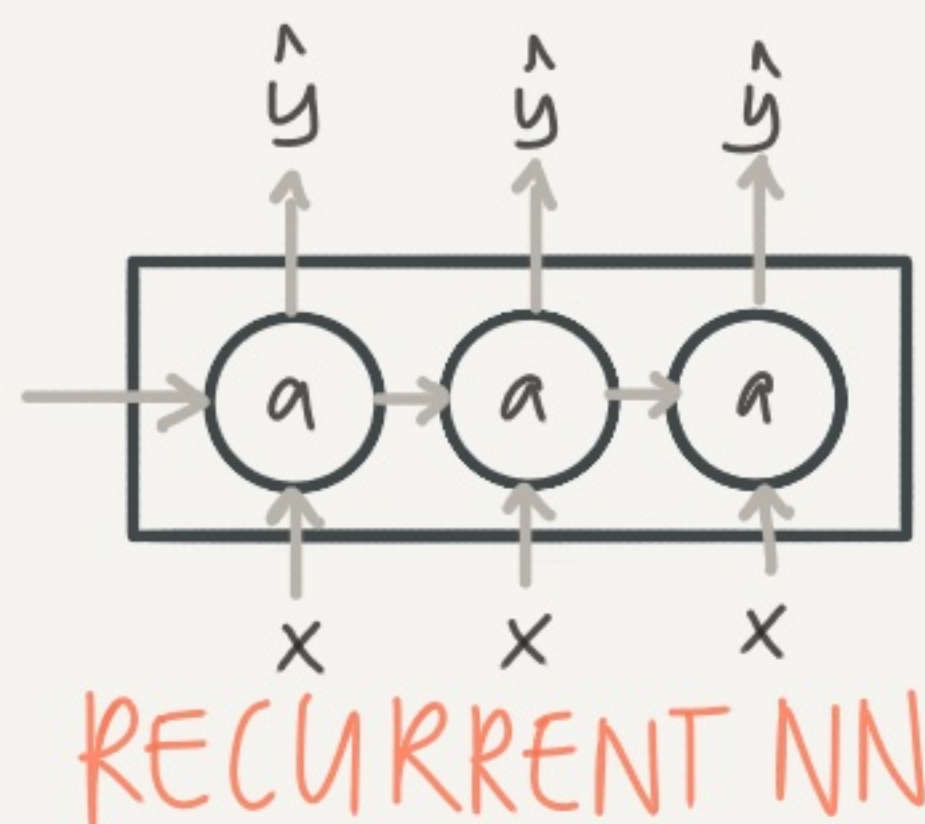
ONE OF THE BIG BREAKTHROUGHS HAS BEEN MOVING FROM SIGMOID TO RELU FOR FASTER GRADIENT DESCENT



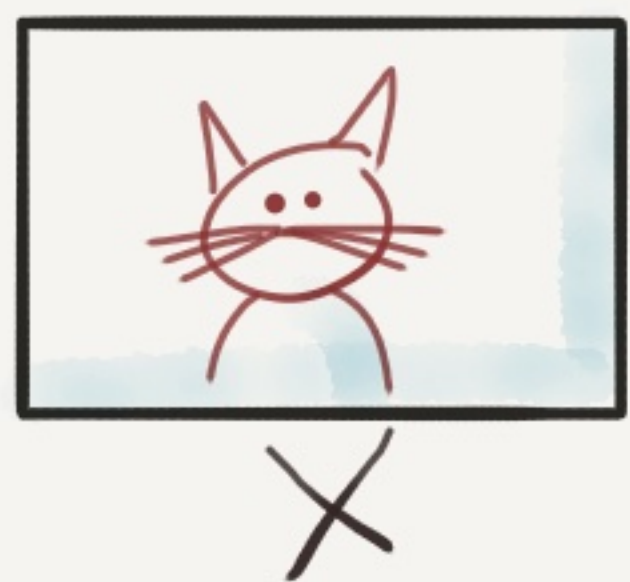
FASTER COMPUTATION IS IMPORTANT TO SPEED UP THE ITERATIVE PROCESS



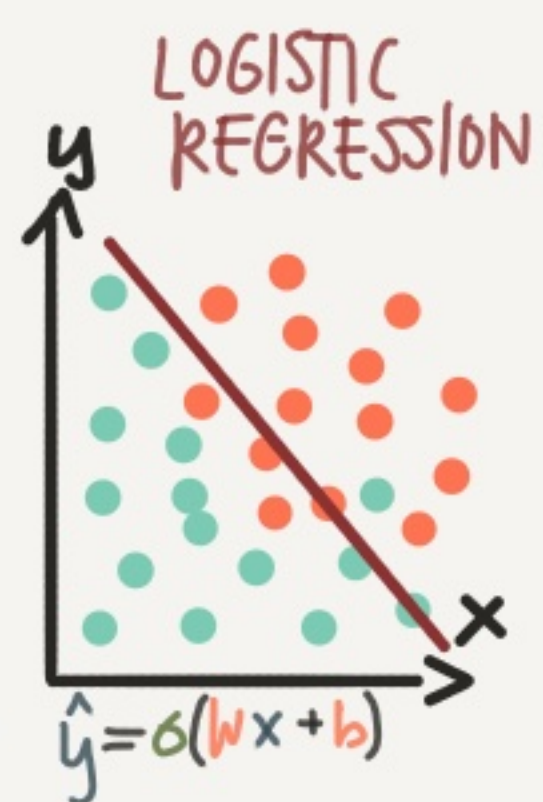
NETWORK ARCHITECTURES



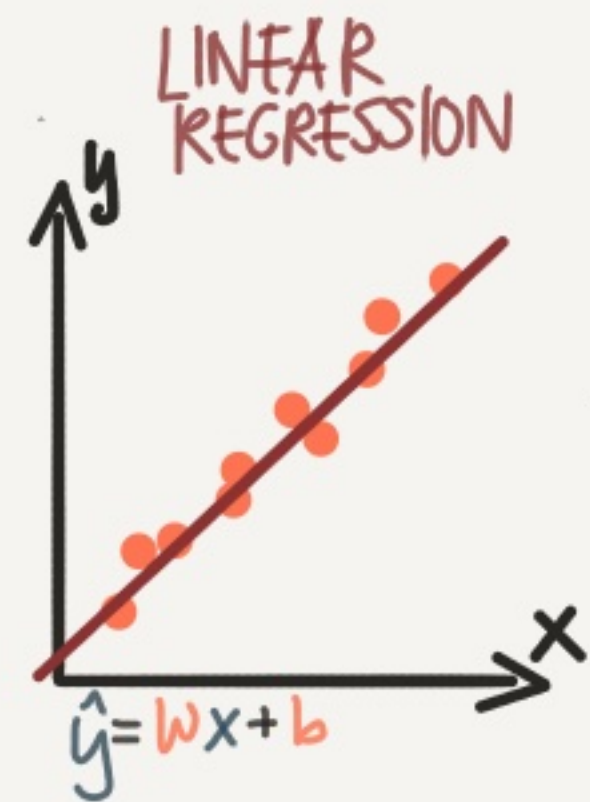
BINARY CLASSIFICATION



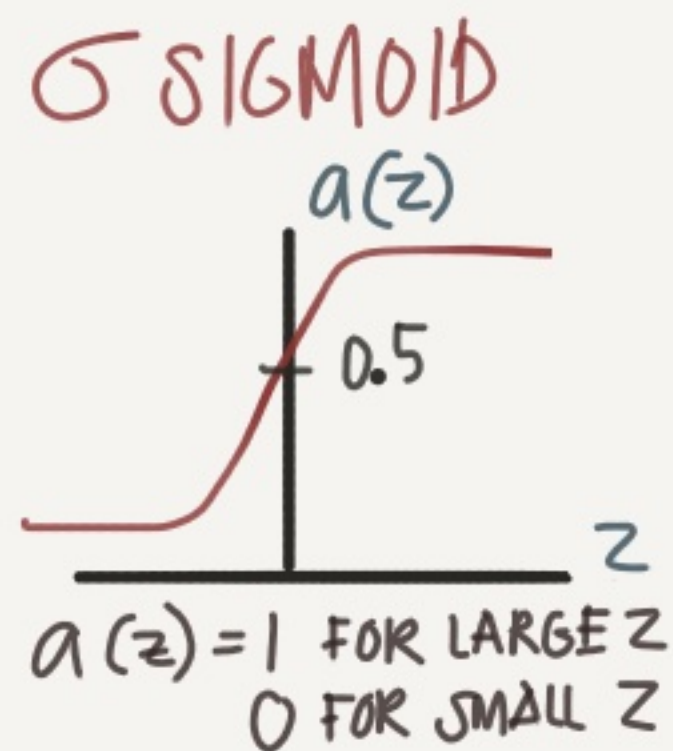
→
 1: CAT
 0: NOT CAT
 Y



=



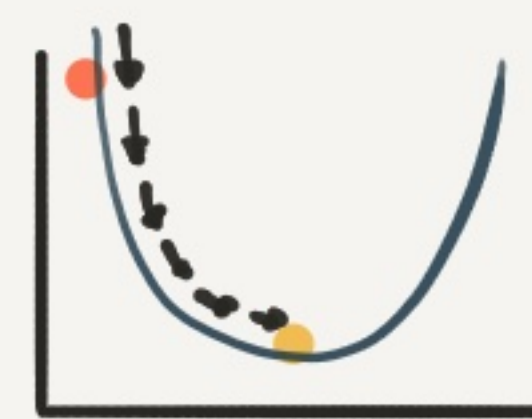
+



LOGISTIC REGRESSION

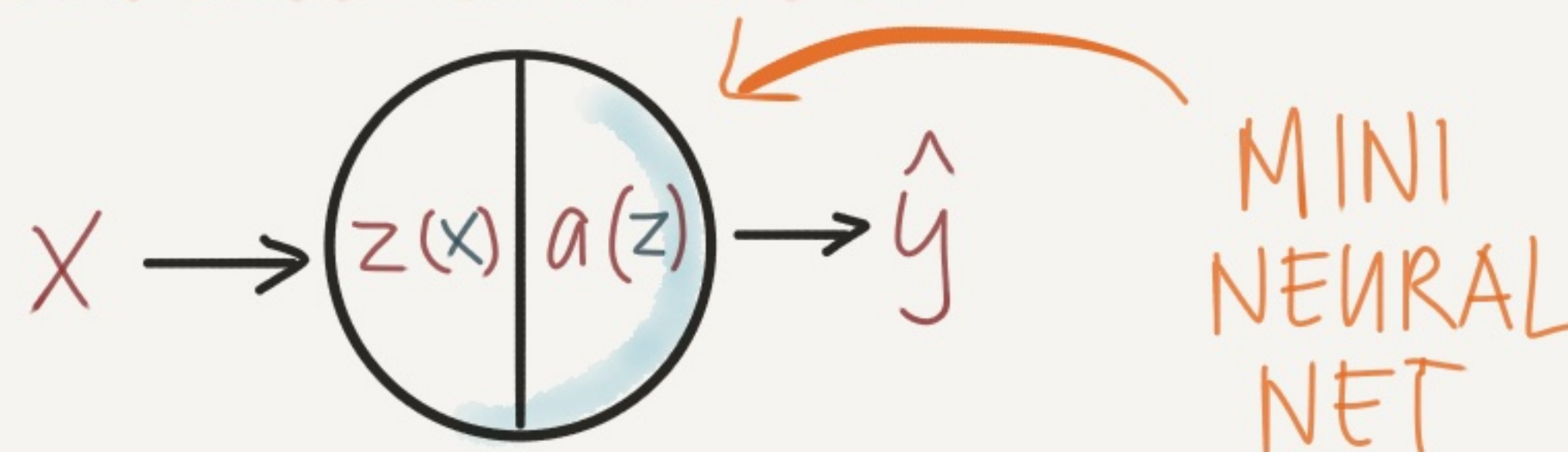
AS A NEURAL NET

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

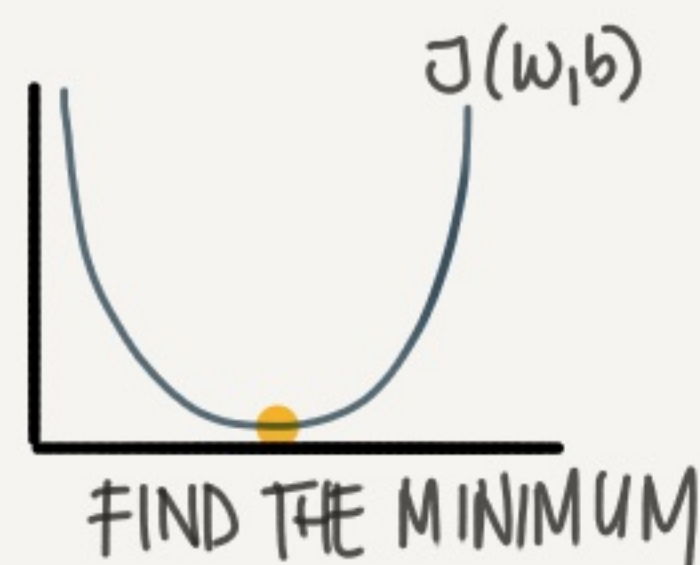
THE TASK IS TO LEARN w & b BUT HOW?

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

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

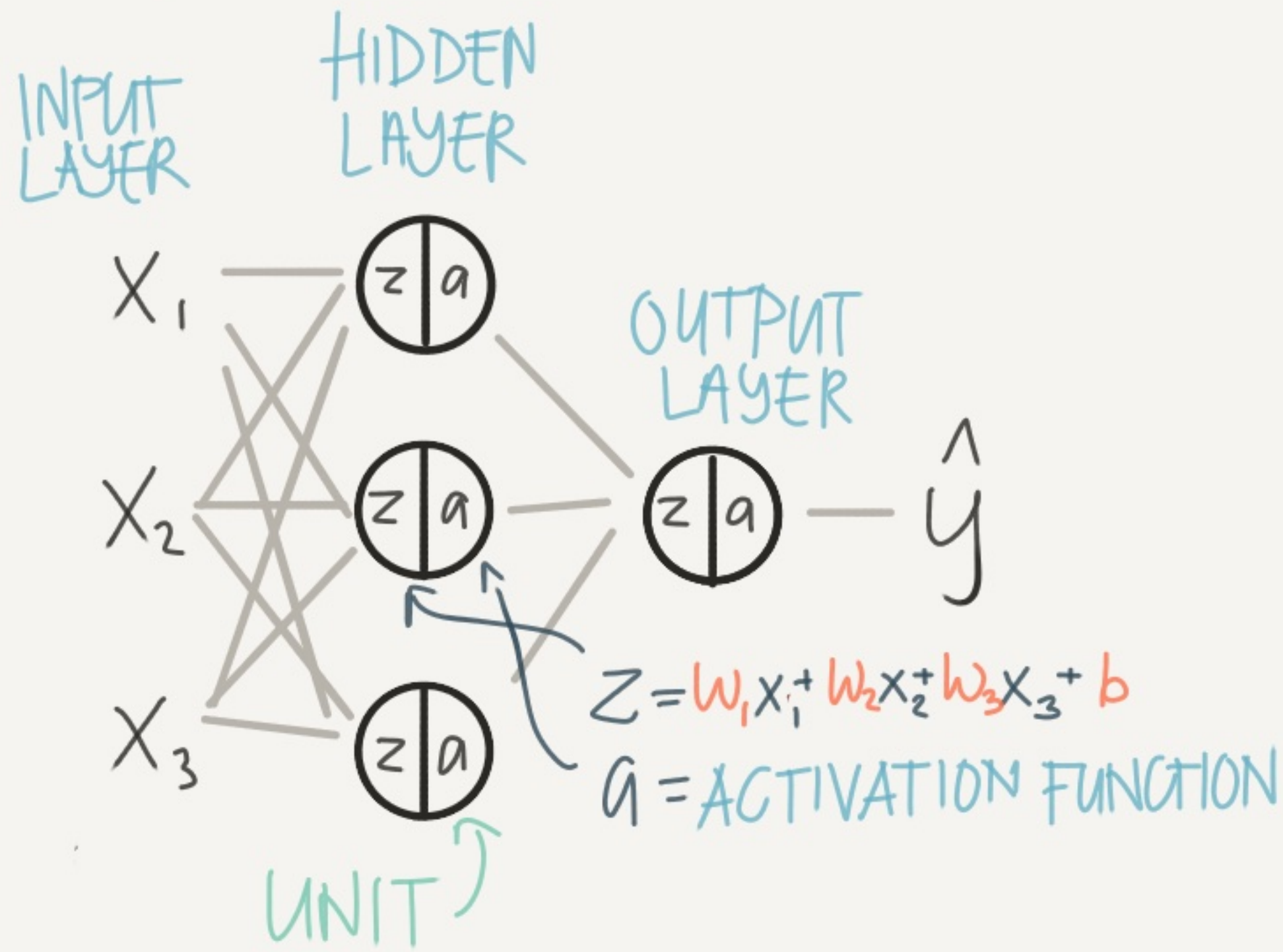
$$\text{COST} = J(w, b) = \frac{1}{m} \sum_{i=1}^m \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$$

COST = LOSS FOR THE ENTIRE DATASET



SHALLOW NEURAL NETS

2 LAYER NEURAL NET



WHY ACTIVATION FUNCTIONS?

EX. WITH NO ACTIVATION - $a = z$

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

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

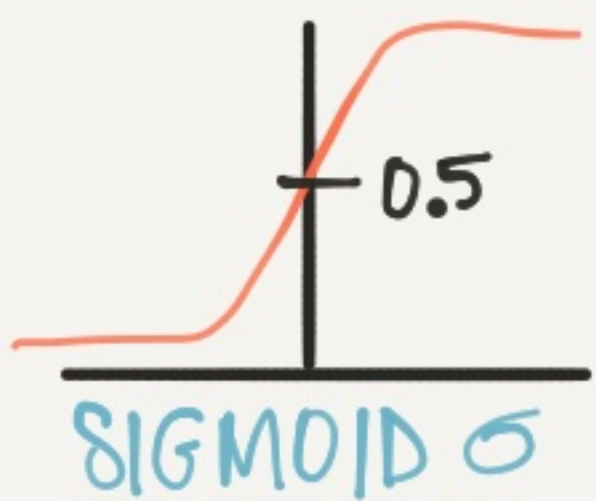
PLUG IN $a^{[1]}$

$$a^{[2]} = w^{[2]}(w^{[1]}x + b^{[1]}) + b^{[2]}$$

$$= \underbrace{w^{[2]}w^{[1]}}_{w'}x + \underbrace{w^{[2]}b^{[1]} + b^{[2]}}_{b'}$$

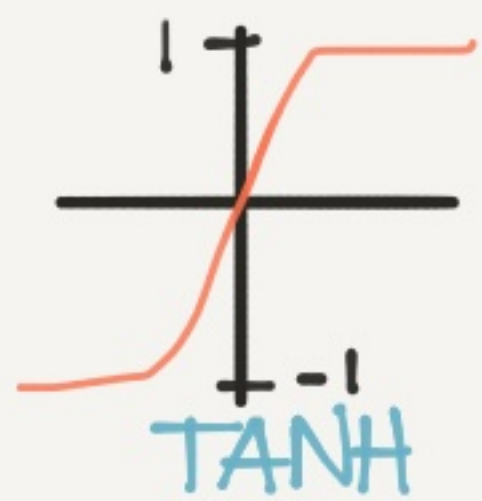
← LINEAR FUNCTION

ACTIVATION FUNCTIONS

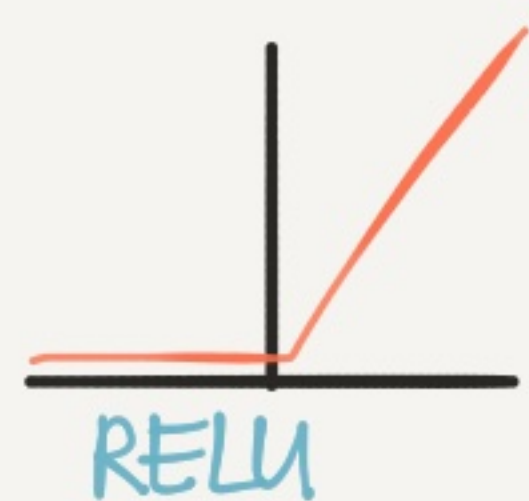


BINARY CLASSIFIER
- ONLY USED FOR OUTPUT LAYER

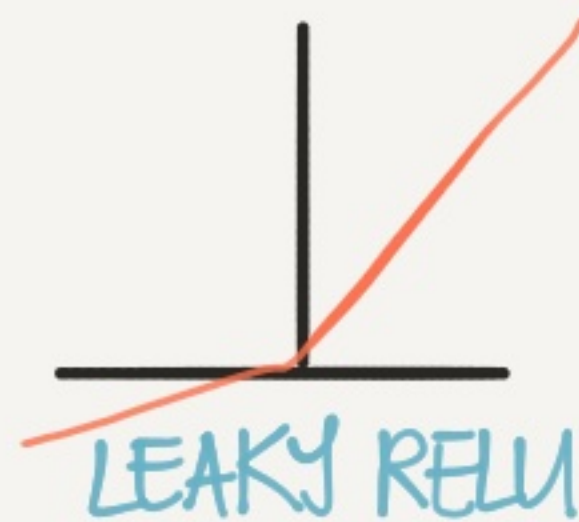
SLOW GRAD DESCENT SINCE SLOPE IS SMALL FOR LARGE/SMALL VAL



NORMALIZED
⇒ 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 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 $\text{RAND} * 0.01$

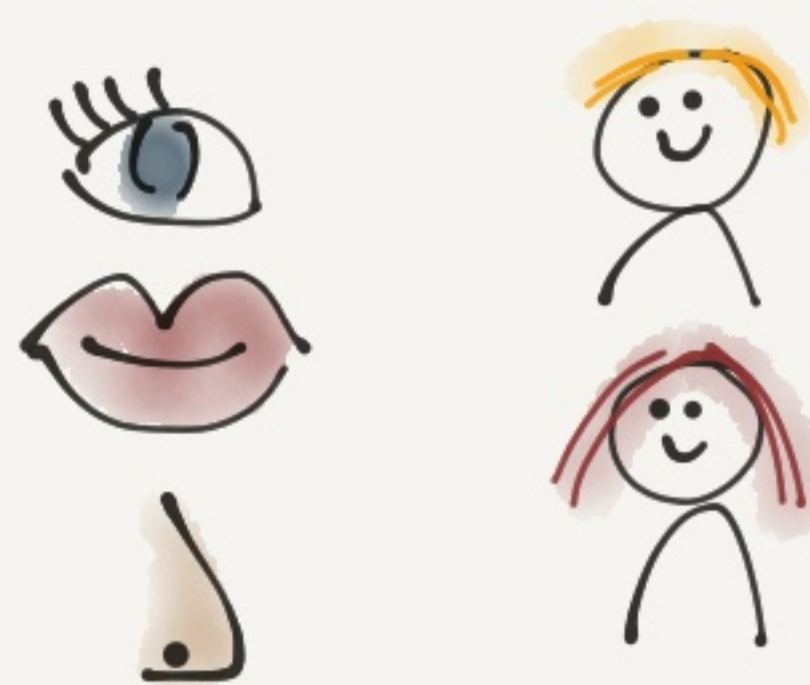
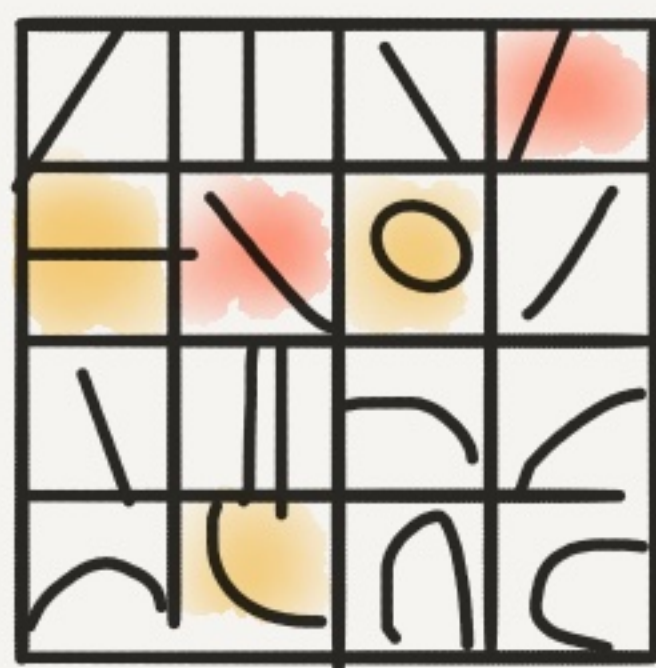
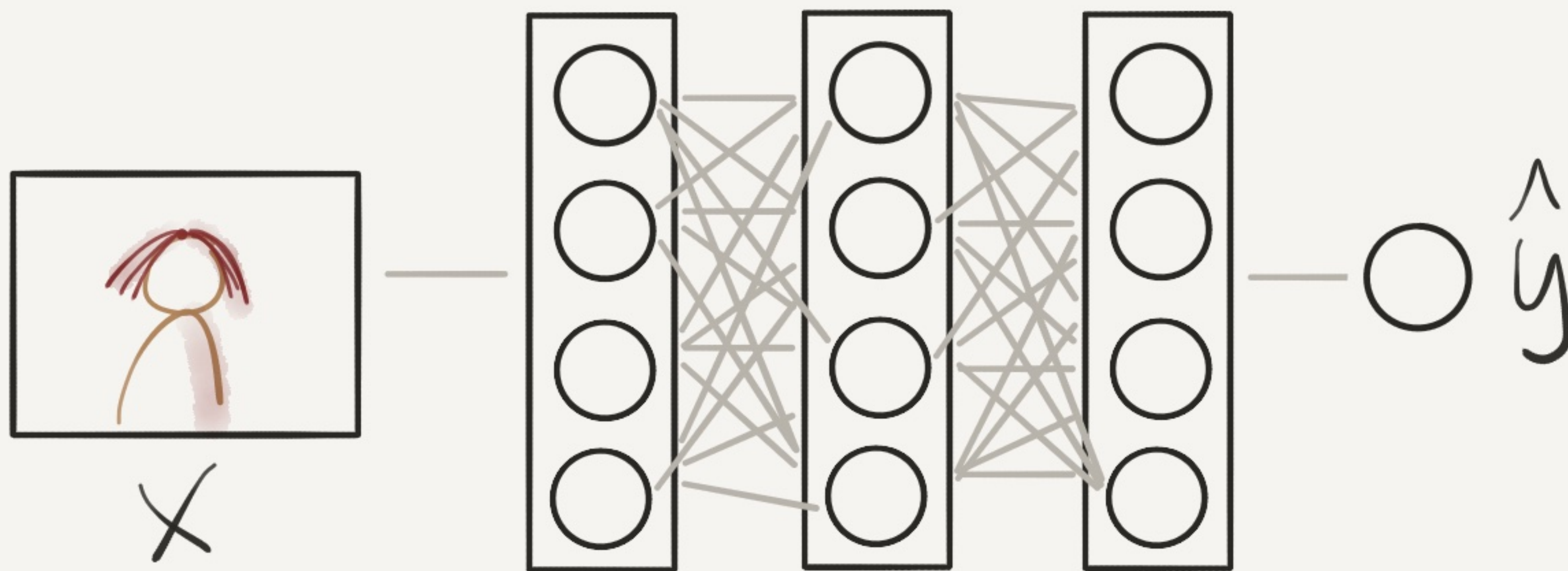
WE COULD JUST AS WELL HAVE SKIPPED THE WHOLE NEURAL NET & USED LIN. REGR.

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?



LOW LEVEL AUDIO WAVE FEATURES
↑ ↓ PITCH

PHONEMES
C A T

WORDS

SENTENCES

VERY DATA HUNGRY

NEED ^{LOTS OF} COMPUTER POWER

ALWAYS VECTORIZE
VECTOR MULT. CHEAPER THAN FOR LOOPS
COMPUTE ON GPUS

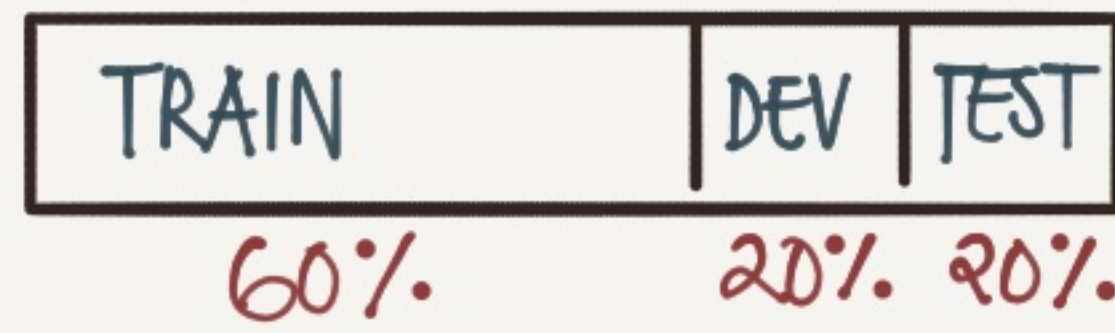
LOTS OF HYPERPARAMS

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

SETTING UP YOUR ML APP

CLASSIC ML

100 - 10000 SAMPLES



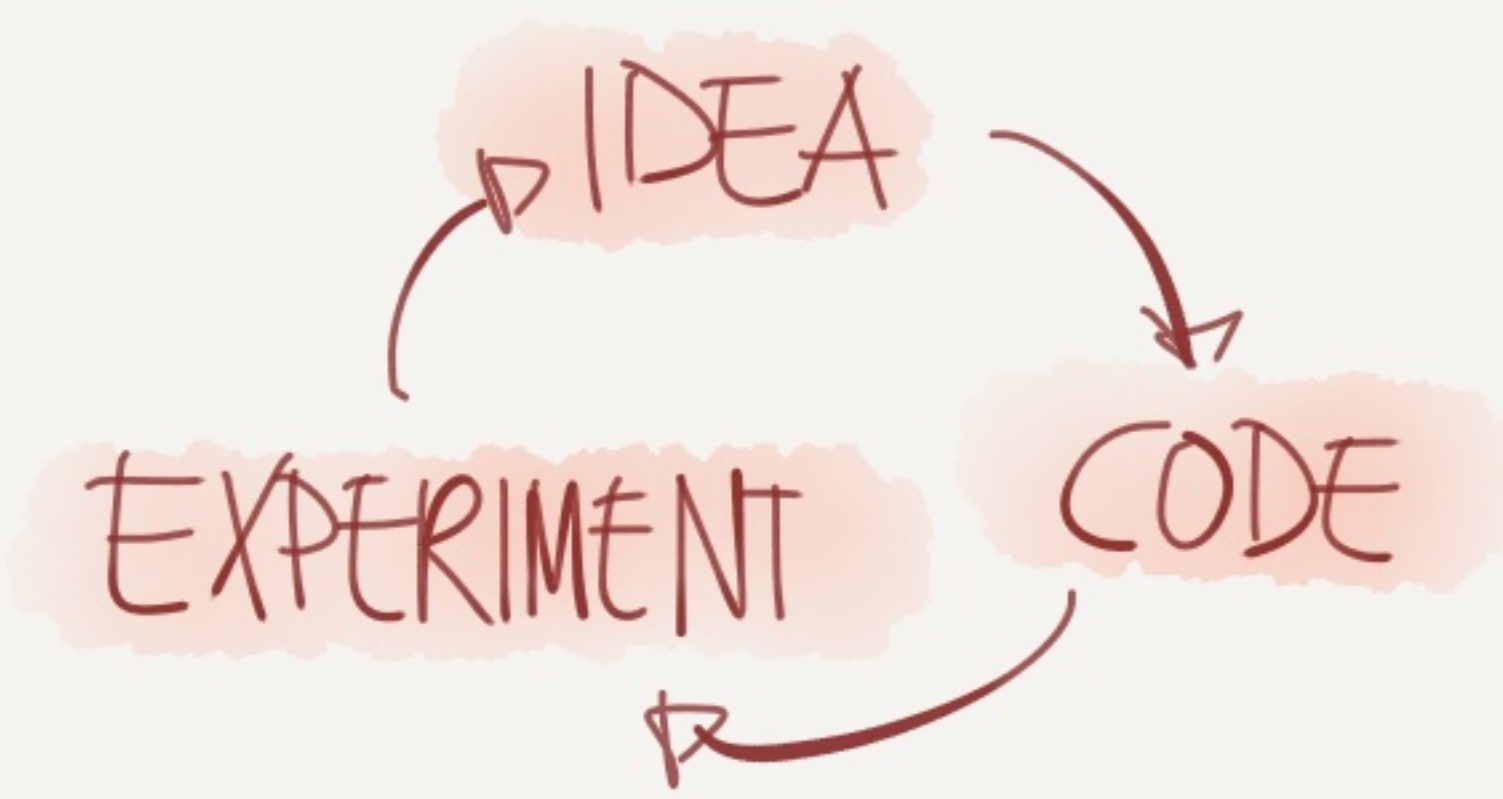
ALL FROM SAME PLACE
DISTRIBUTION

DEEP LEARNING

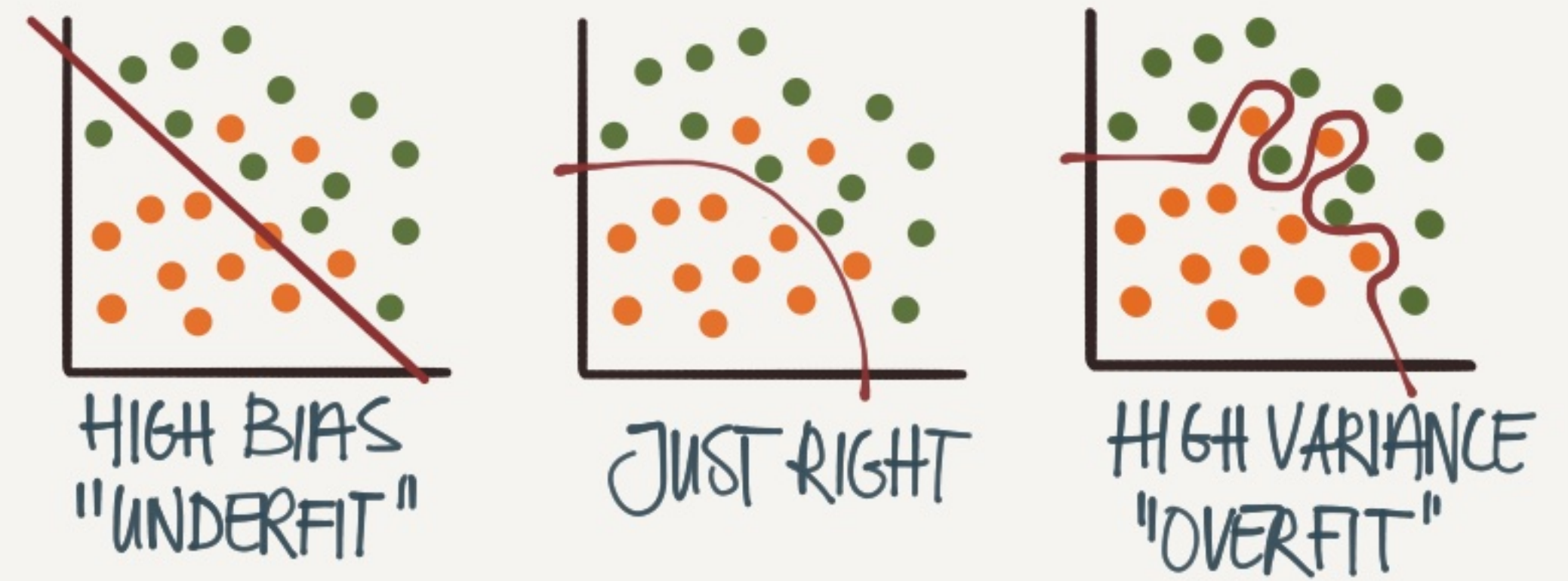
1M SAMPLES



TIP
DEV & TEST SHOULD COME FROM SAME DISTRIBUTION



BIAS/VARIANCE

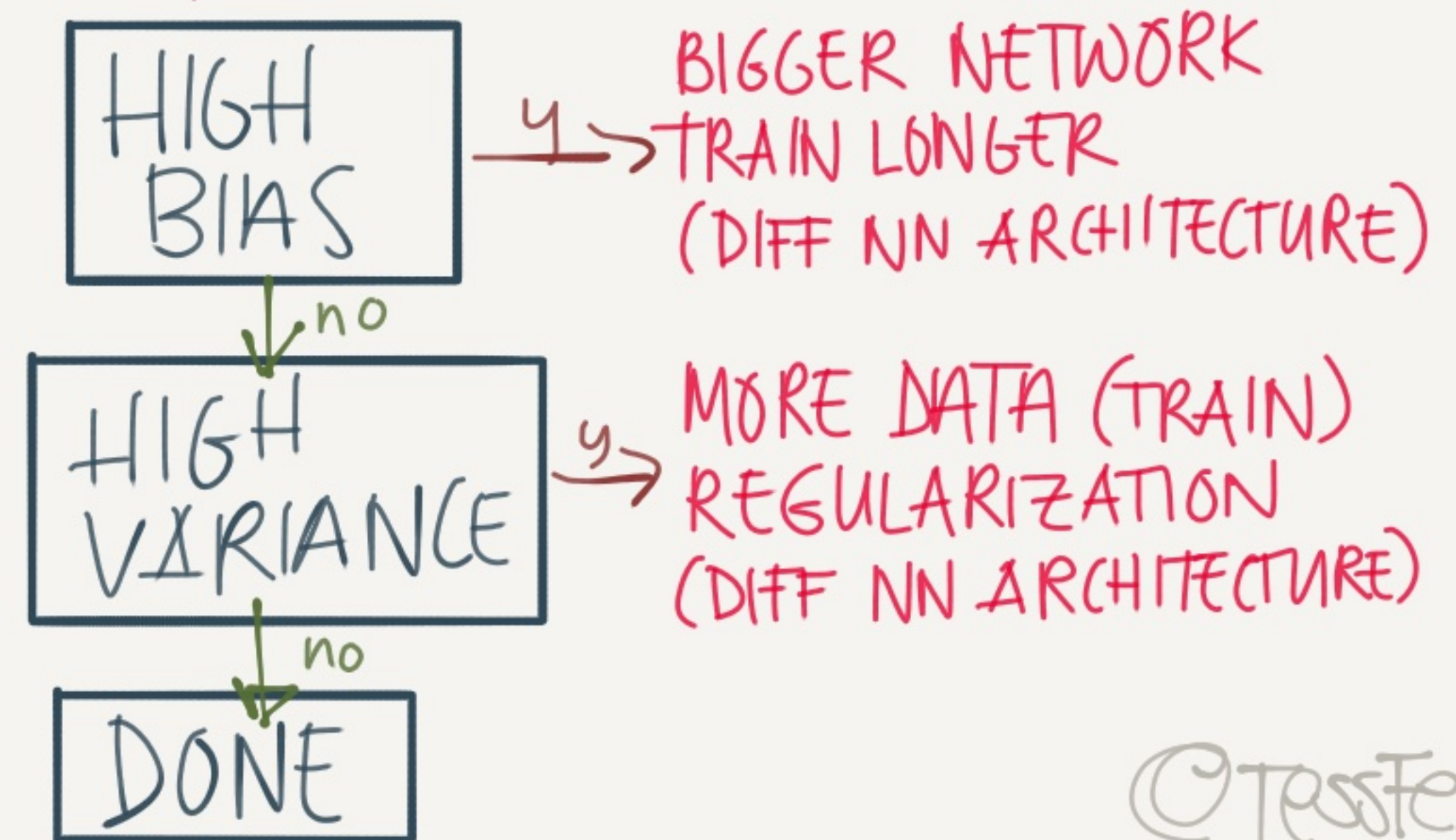


	ERROR			
TRAIN	1%	15%	15%	0.5%
TEST	11%	16%	30%	1%

HIGH VARIANCE HIGH BIAS HIGH BIAS & VARIANCE LOW BIAS & VARIANCE

ASSUMING HUMANS GET 0% ERROR

THE ML RECIPE



REGULARIZATION

PREVENTING OVERFITTING

L2 REGULARIZATION

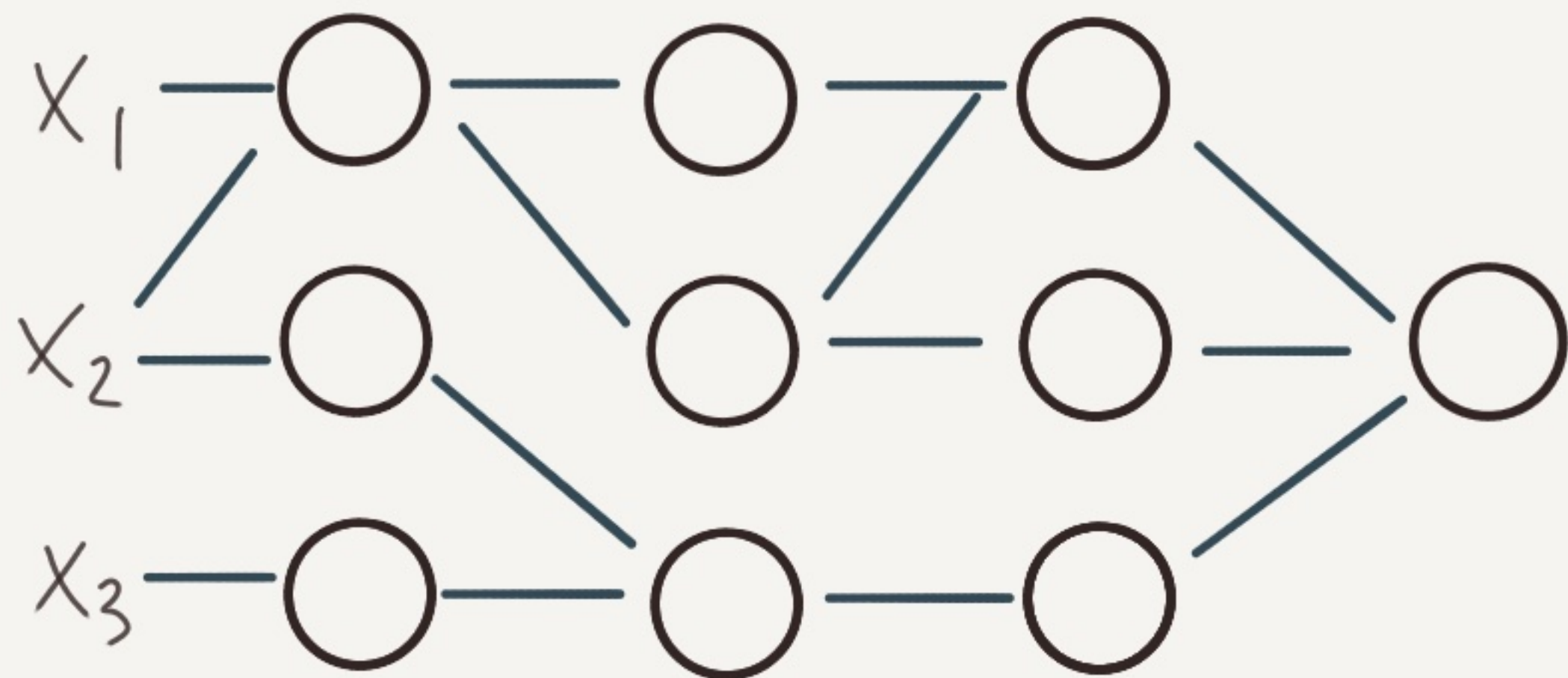
$$\text{COST: } J(w, b) = \frac{1}{m} \sum_{i=1}^m d(\hat{y}_i, y_i) + \frac{\lambda}{2m} \|w\|_2^2$$

← EUCLIDEAN NORM

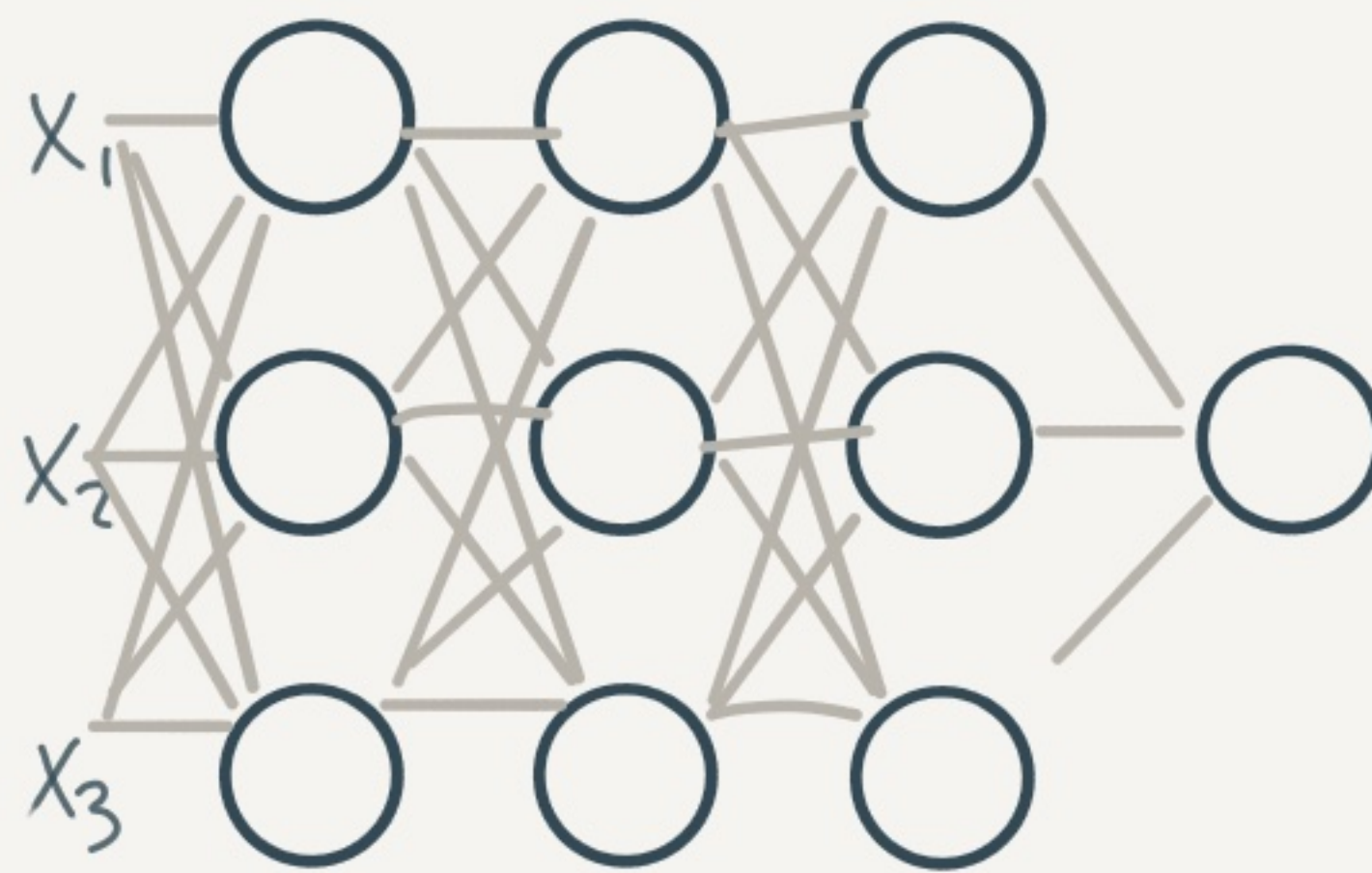
L1 REGULARIZATION

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

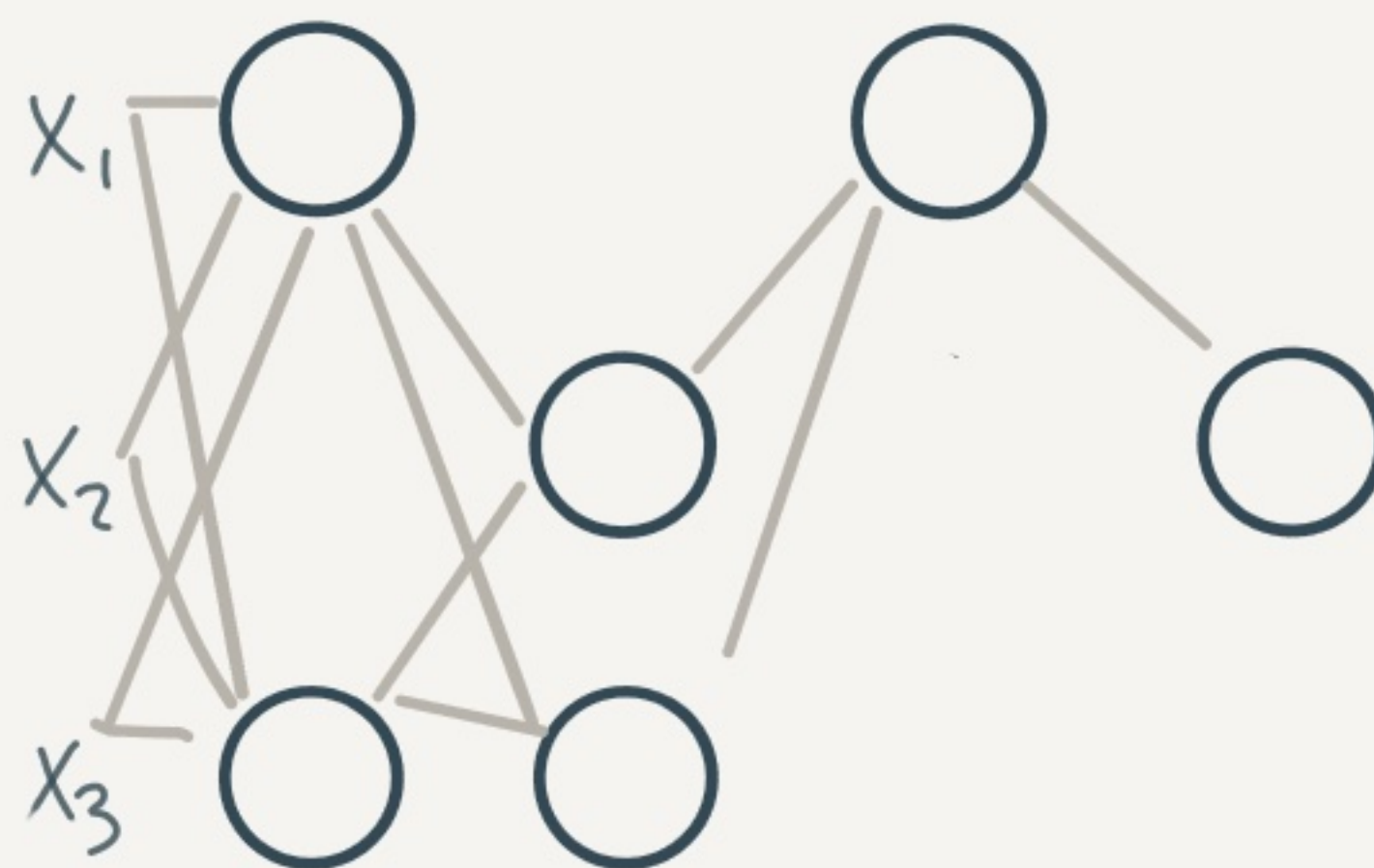
BOTH PENALIZE LARGE WEIGHTS ⇒
SOME WILL BE CLOSE TO 0 ⇒
SIMPLER NETWORKS



DROPOUT



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

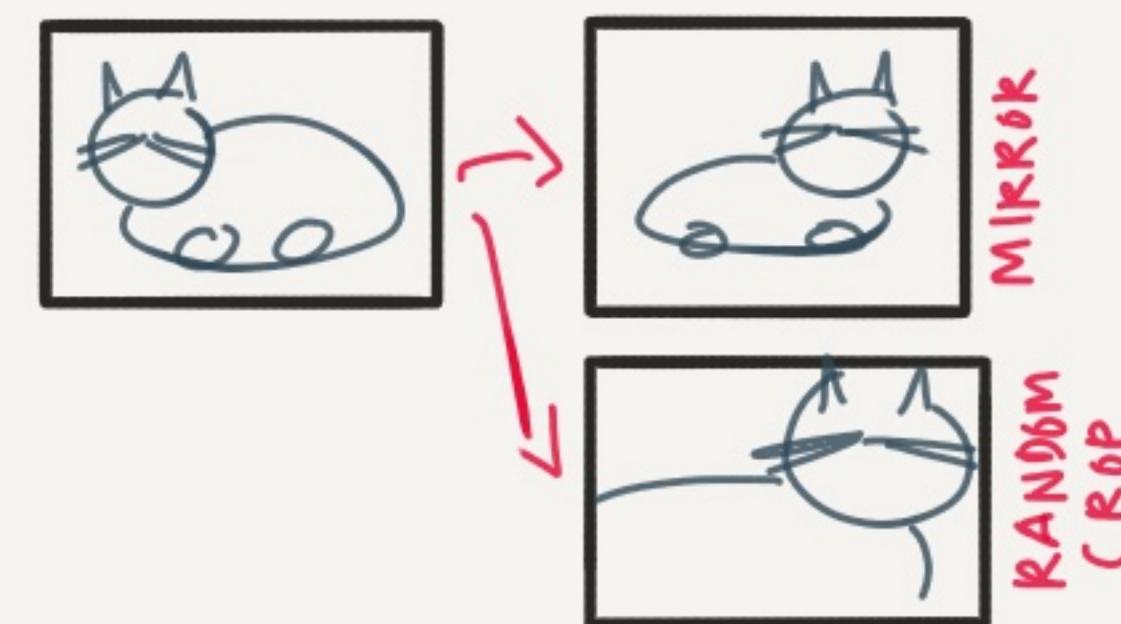


WE GET SIMPLER NETWORKS
& LESS CHANCE TO RELY ON
SINGLE FEATURES

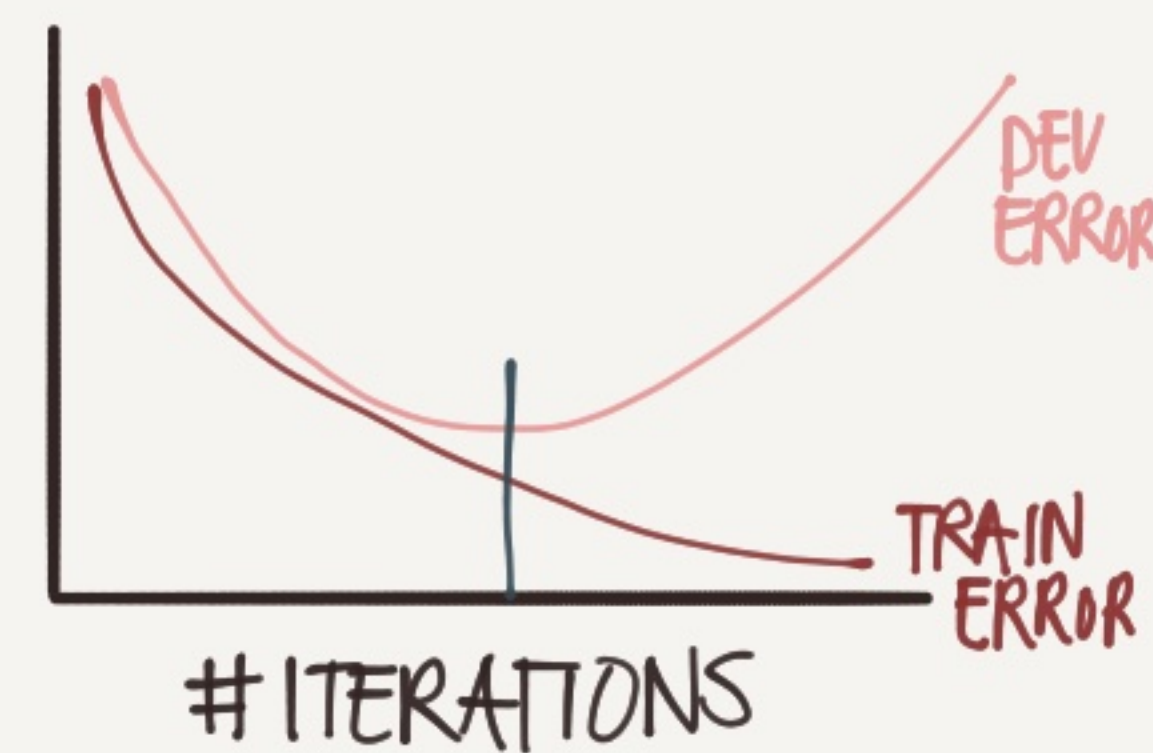
OTHER REGULARIZATION TECHNIQUES

DATA AUGMENTATION

GENERATE NEW PICS FROM EXISTING



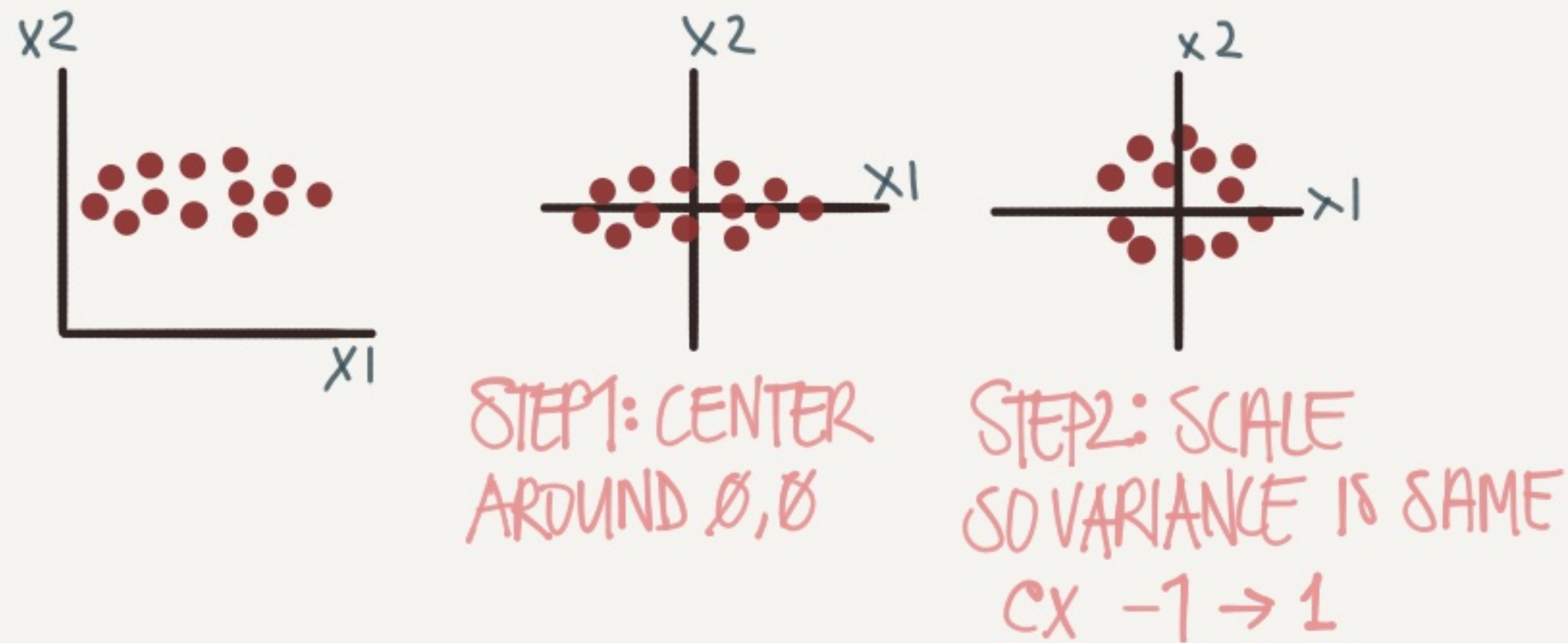
EARLY STOPPING



PROBLEM: AFFECTS BOTH
BIAS & VARIANCE

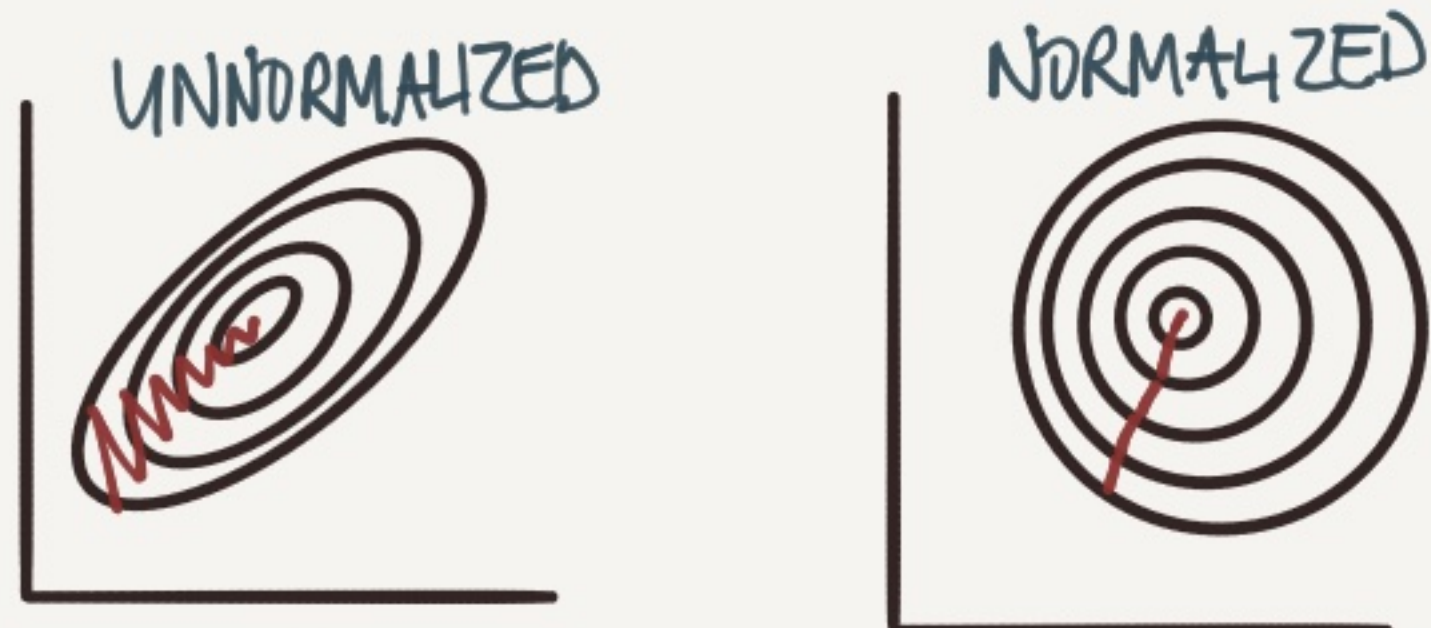
OPTIMIZING TRAINING

NORMALIZING INPUTS



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 NW (L LAYERS)

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

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

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

#inputs

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

SETS THE VARIANCE

GRADIENT CHECKING

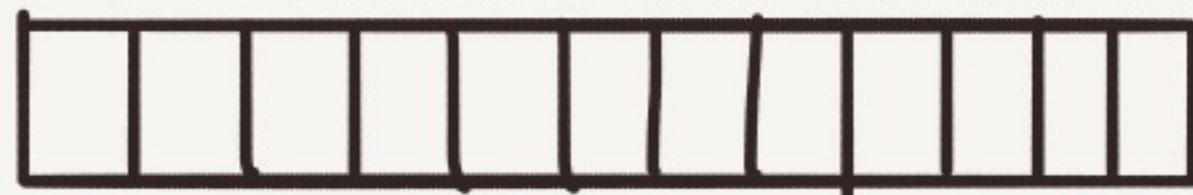
IF YOUR COST DOES NOT DECREASE ON EACH ITER YOU MAY HAVE A BACKPROP BUG.

GRADIENT CHECKING APPROXIMATES THE GRADIENTS SO YOU CAN VERIFY CALC.

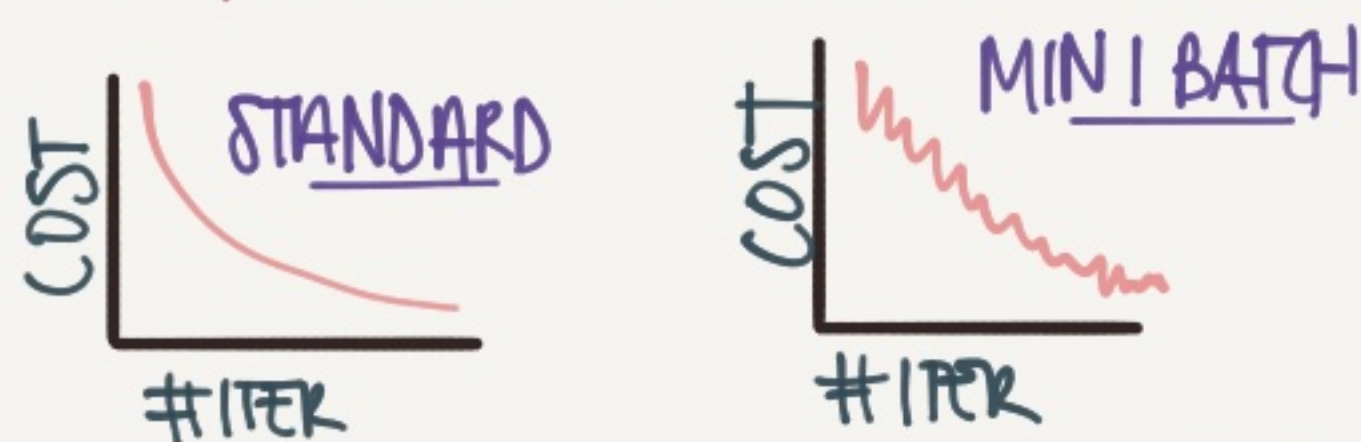
NOTE ONLY USE WHEN DEBUGGING SINCE IT'S SLOW

OPTIMIZATION ALGORITHMS

MINI-BATCH GRAD. DESCENT



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



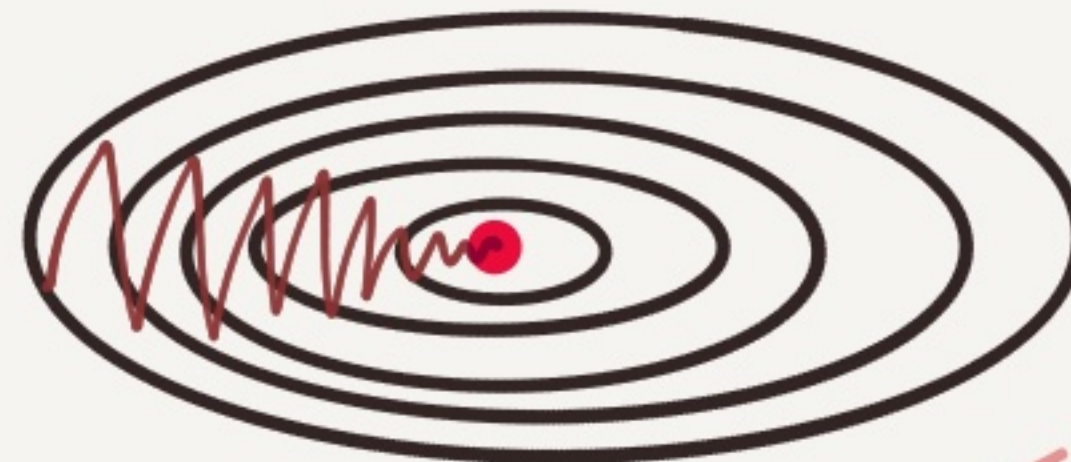
CHOOSING THE MINIBATCH SIZE

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



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

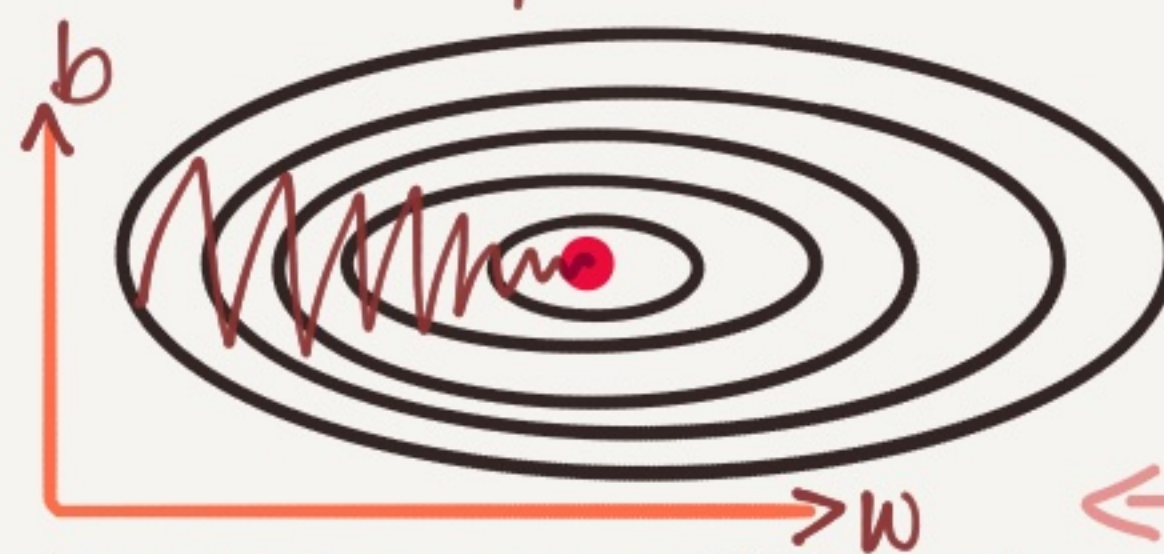


↑ SLOWER LEARNING
 ↓ FASTER LEARNING

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



↑ SLOWER LEARNING
 ↓ FASTER LEARNING

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

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

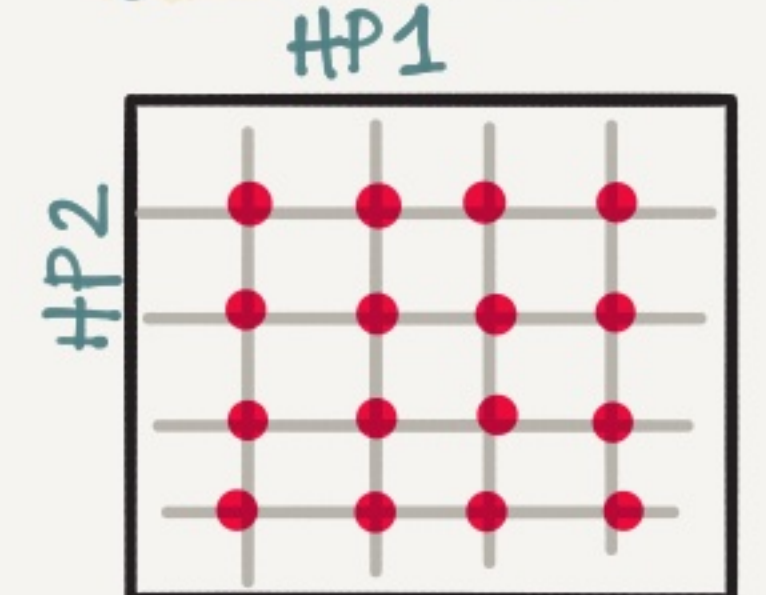
HYPERPARAM TUNING

WHICH HYPERPARAMS ARE MOST IMPORTANT?

- α LEARNING RATE
- # HIDDEN UNITS
- MINIBATCH SIZE
- β MOMENTUM: TURN=0.9
- # LAYERS
- LEARNING RATE DECAY
- β₁=0.9 β₂=0.999 ε=10⁻⁸ (ADAM)

TESTING VALUES

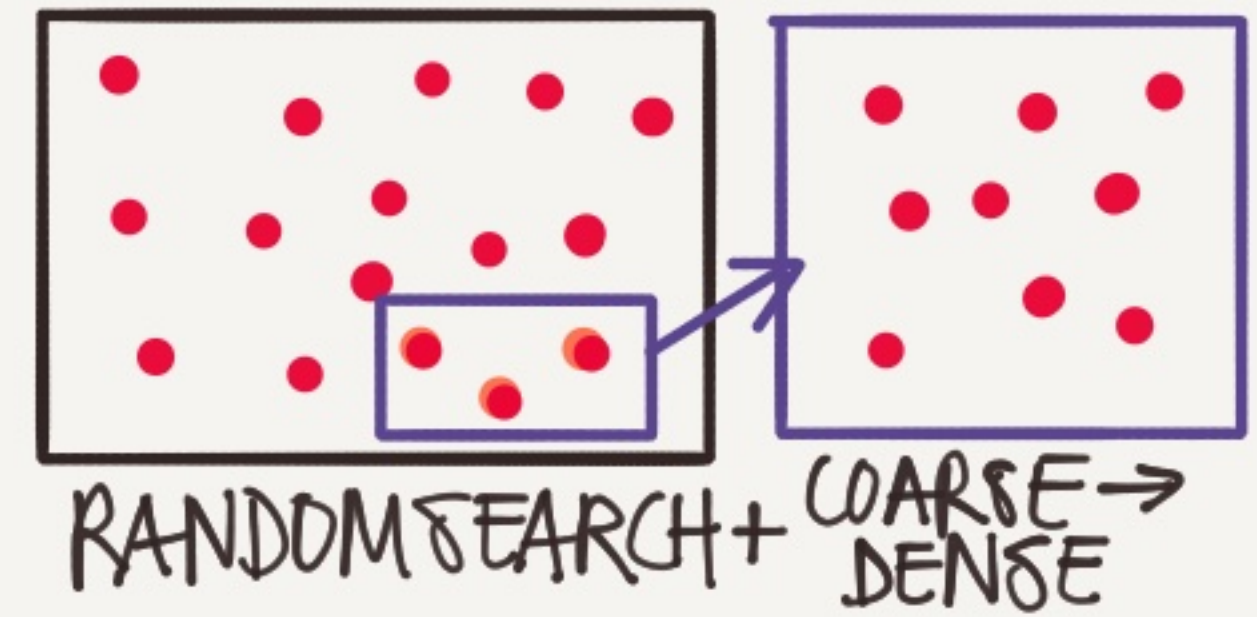
CLASSIC ML



PROBLEM: ONE ITERATION TAKES A LONG TIME & IN 16 60'S WE HAVE ONLY TRIED 4α - BUT 4 DIFF ε

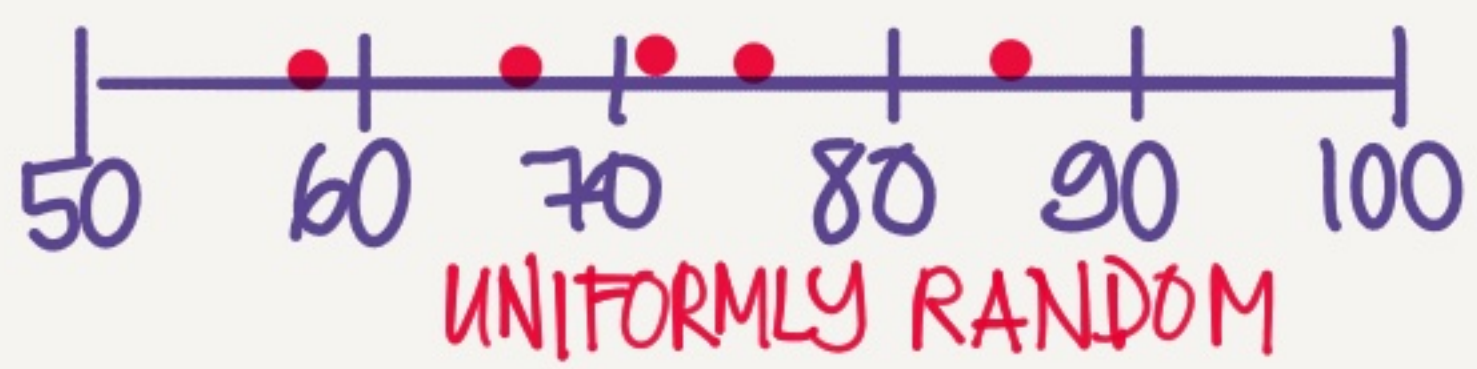
NOT AS IMPORTANT

GRID SEARCH SOLUTION

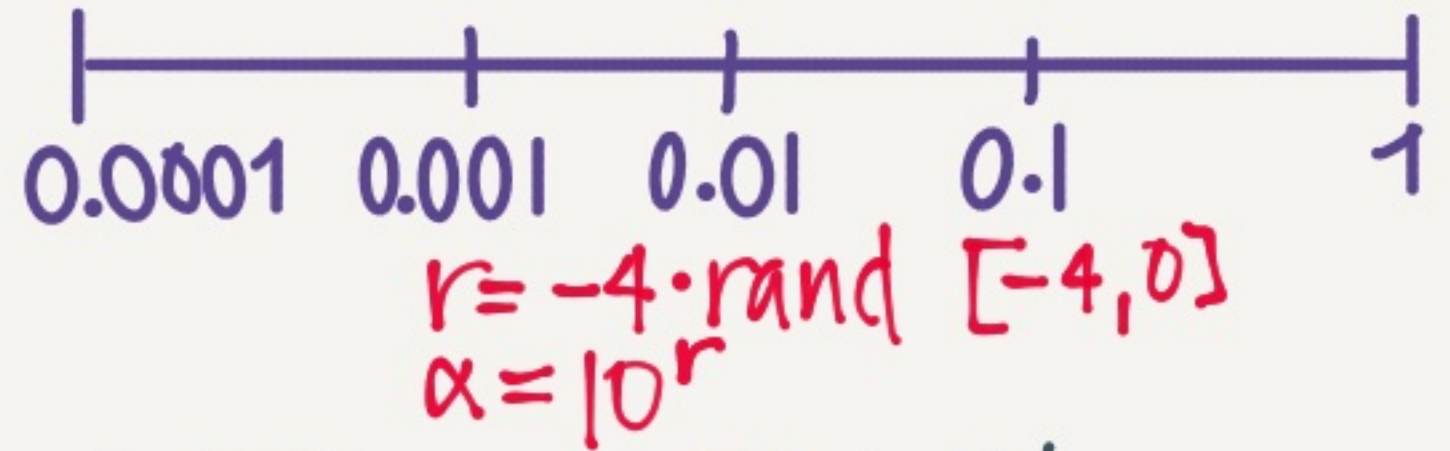


USE AN APPROPRIATE SCALE

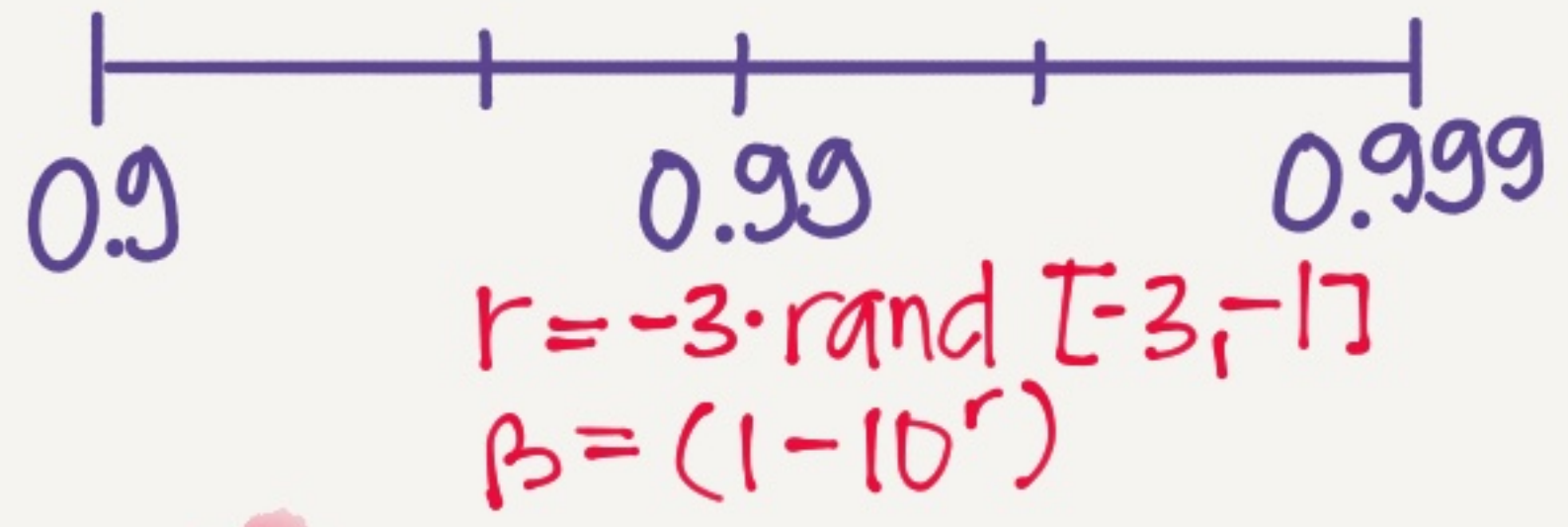
HIDDEN UNITS



α LEARNING RATE



β EXP WEIGHT AVE



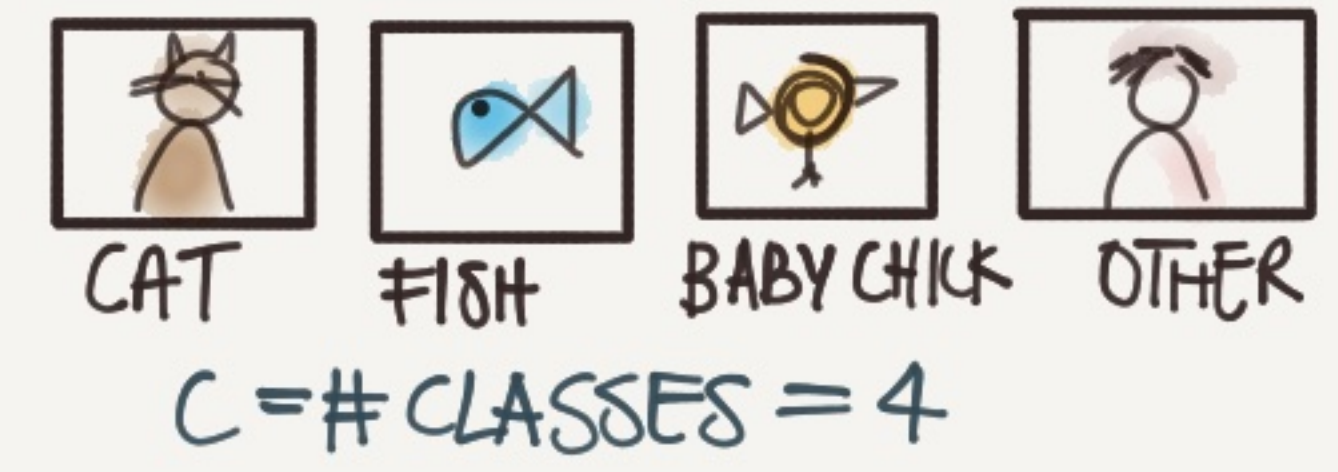
TIP RE-EVALUATE YOUR HYP. PARAMS EVERY FEW MONTHS

MISC. EXTRAS

BATCH NORMALIZATION

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

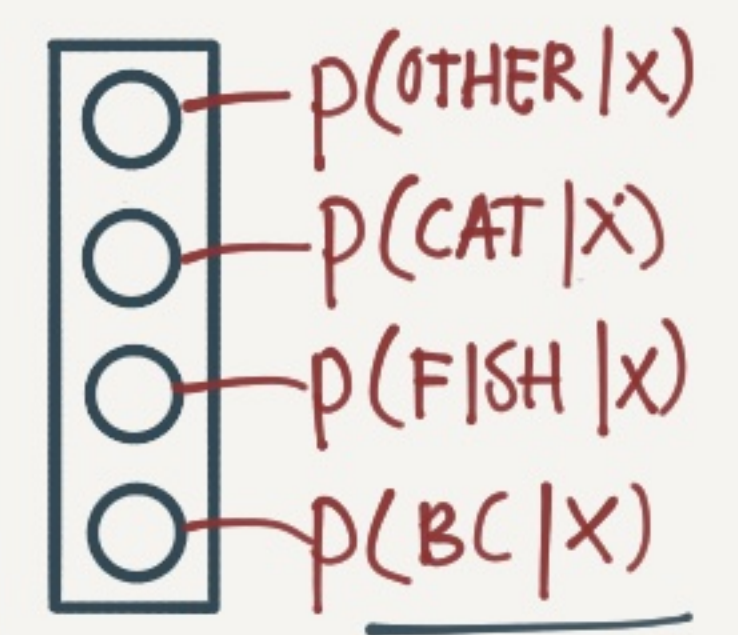
MULTICLASS CLASSIFIC.



SOFTMAX ACTIVATION

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

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



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

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

PANDA VS CAVIAR

BABY SIT ONE MODEL & TUNE

GOOD IF YOU HAVE LOTS OF SPARE COMP POWER

SPAWN LOTS OF MODELS W DIFF HP

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

IMPROVING DEEP NEURAL NETS COURSE

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 \leq 100ms

ACCURACY = OPTIMIZING
RUNTIME = SATISFICING

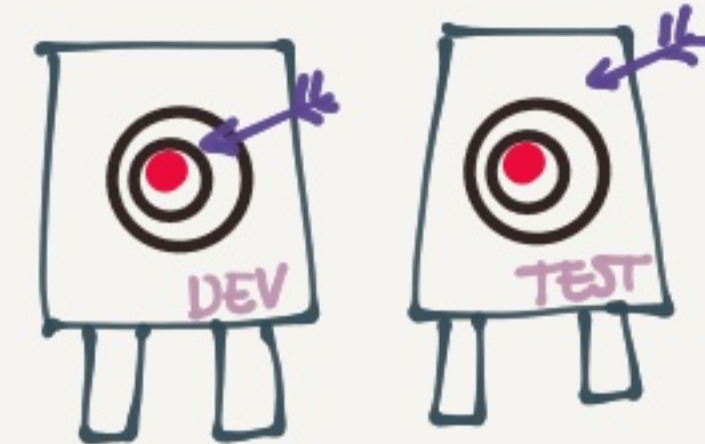
SELECTING YOUR DEV/TEST SETS

DATA

US
UK
EUR/AFR
S.AM
INDIA
CHINA
AUST.

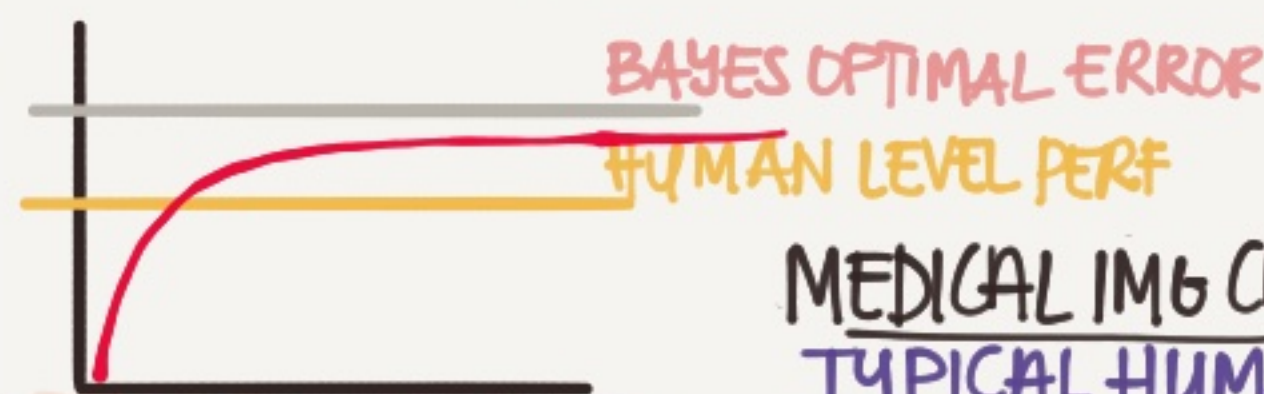
OPTION 1:

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

BLURRY
AVOIDABLE BIAS
VARIANCE
FOCUS ON BIAS
FOCUS ON VARIANCE

HUMAN } TRAIN BIGGER NETW.
| AVOIDABLE BIAS } TRAIN LONGER / BETTER OPT. (RMSProp, ALGOs, ADAM)
TRAIN } CHANGE NN ARCH OR HYPERPARAMS
| VARIANCE } MORE DATA (TRAIN)
DEV } REGULARIZATION
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 OVERTFIT 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?

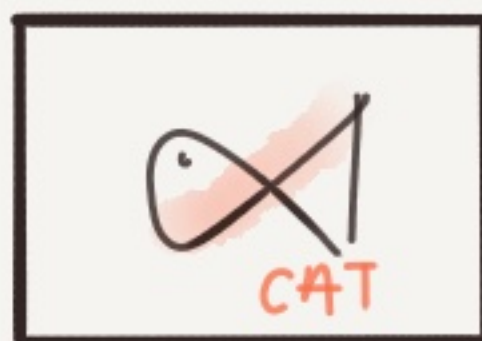
1. PICK 100 MIS-LABELED
2. COUNT ERROR REASONS

	DOG	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. 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 INCORR LABELED 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

©TessFerrandez

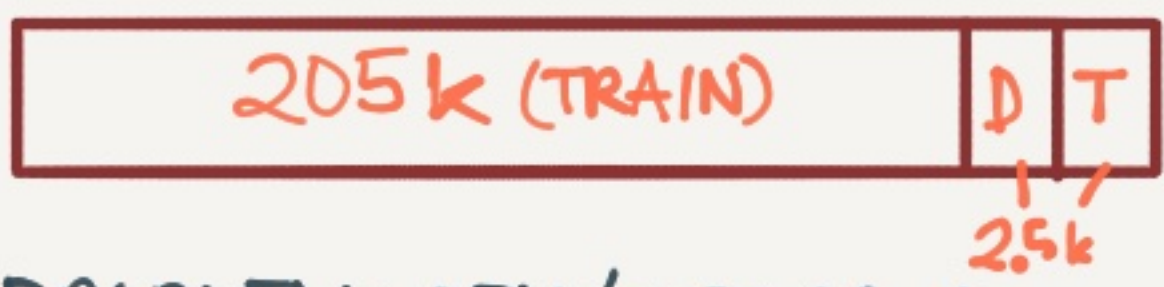
TRAIN vs DEV/TEST MISMATCH

AVAILABLE DATA

- 200k PRO CAT PICS FROM INTERNET
 - 10k BLURRY CAT PICS FROM APP
- WHAT WE CARE ABT**

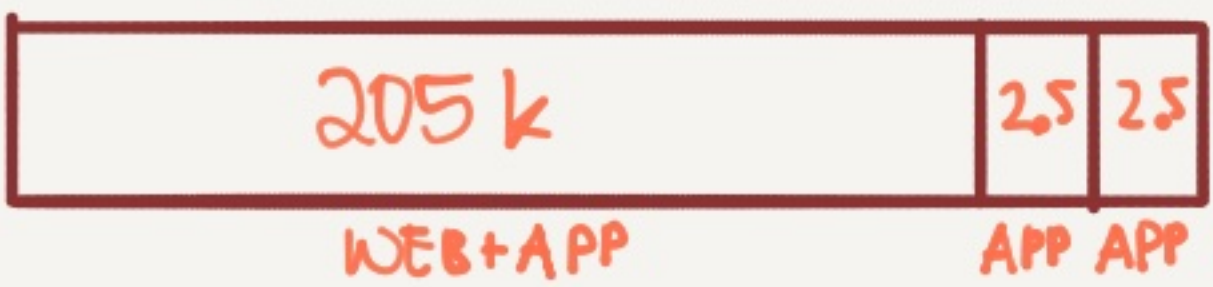
HOW DO WE SPLIT → TRAIN/DEV/TEST?

OPTION 1: SHUFFLE ALL



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

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

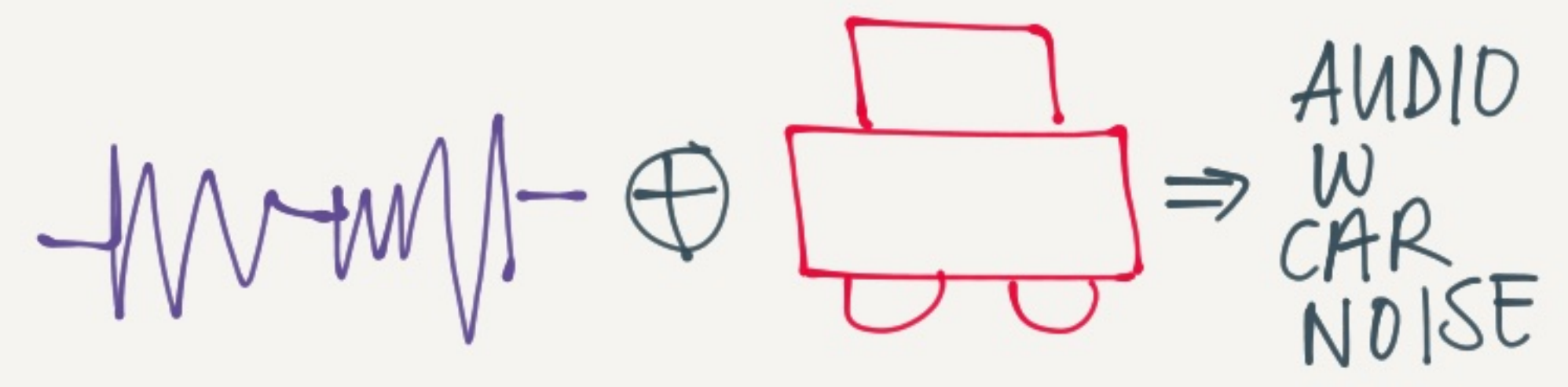


	A	B	C	D
TRAIN	1%	1%	10%	10%
TRAIN-DEV	9%	15%	11%	11%
DEV	10%	10%	12%	20%
	VARIANCE	TRAIN/DEV MISMATCH	BIAS	BIAS+ DATA MISMATCH

ADDRESSING DATA MISMATCH

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

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

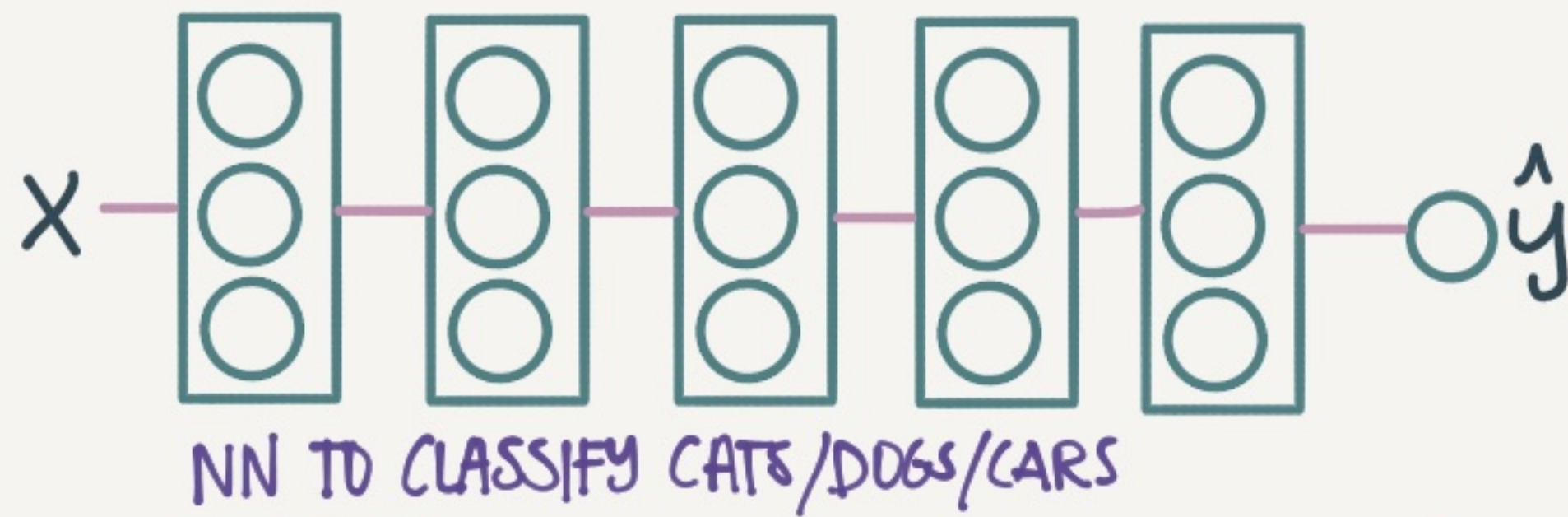


NOTE BE CAREFUL. IF YOU ONLY HAVE 1 HR OF CAR NOISE & APPLY IT TO 10K HR SPEECH YOU MAY OVERFIT TO THE CAR NOISE

EXTENDED LEARNING

TRANSFER LEARNING

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



OPTION 1: YOU ONLY HAVE A FEW RADIOLOGY IMAGES

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

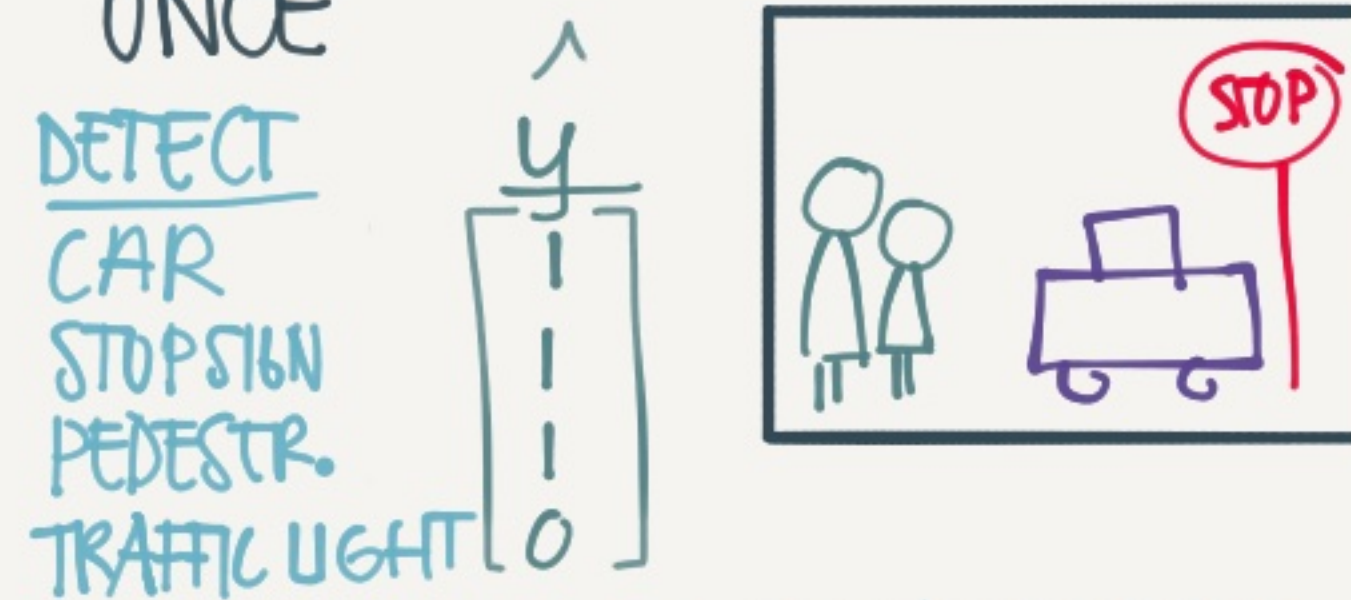
OPTION 2: YOU HAVE LOTS OF RADIOLOGY IMG.

SOLUTION: INIT WITH WEIGHTS FROM CAT NN RETRAIN ALL LAYERS

THIS IS MICROSOFT CUSTOM VISION

MULTITASK LEARNING

TRAINING ON MULT. TASKS AT ONCE



UNLIKE SOFMAX. MANY THINGS CAN BE TRUE

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

SUMMING OVER ALL OUTP OPTIONS

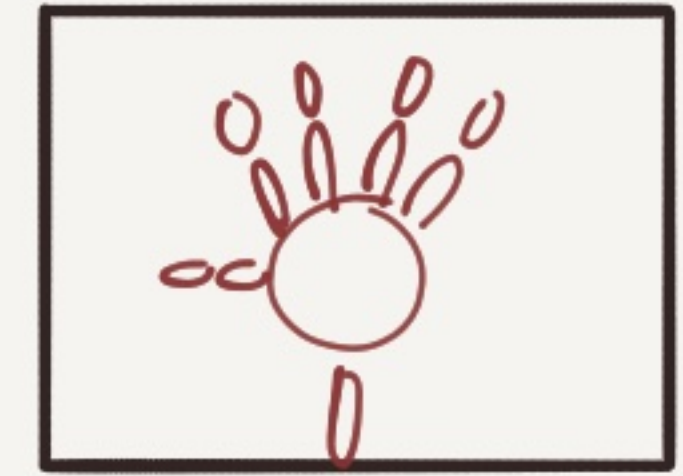
WE COULD HAVE JUST TRAINED 4 NNS INSTEAD BUT.. MT LEARNING MAKES SENSE WHEN

- 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:


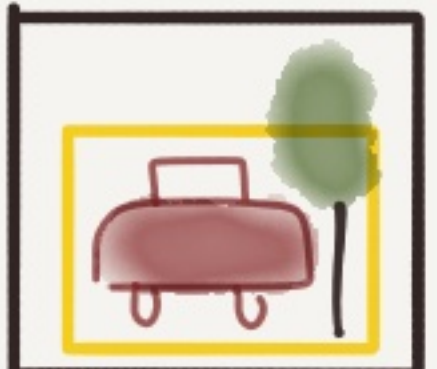
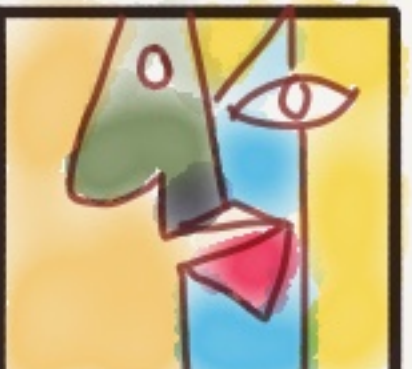
- NEEDS LARGE AMTS OF DATA (X → Y) ^{LARGELY}
- EXCLUDES POTENTIALLY USEFUL HAND-MADE COMPONENTS

©tessterrandez

CONVOLUTION

FUNDAMENTALS

COMPUTER VISION

<p>IMAGE CLASSIFICATION</p>  <p>CAT OR NOT-CAT</p>	<p>OBJECT DETECTION</p>  <p>WHERE IS THE CAR?</p>	<p>NEURAL STYLE TRANSFER</p>  <p>PAINT ME LIKE PICASSO</p>
---	--	---

PROBLEM: IMAGES CAN BE BIG
 $1000 \times 1000 \times 3$ (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

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

CONVOLUTION

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

INPUT 6x6 IMAGE

$$3 + 1 + 2 + 0 + 0 + 0 - 1 - 8 - 2 = -5$$

(3x3)

1	0	-1
1	0	-1
1	0	-1

CONVOLUTION
 FILTER 3x3

-5	-4	0	8
-16	-2	2	3
0	-2	-4	-7
-3	-2	-3	-16

OUTPUT 4x4 IMAGE

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

INPUT 6x6 IMAGE

1	0	-1
1	0	-1
1	0	-1

VERTICAL EDGE DETECTOR
 FILTER 3x3

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

OUTPUT 4x4 IMAGE

DETECTED EDGE IN THE MIDDLE

THIS IS LIKE ADDING AN 'INSTA' FILTER THAT JUST SHOWS OUTLINES

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

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	7	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

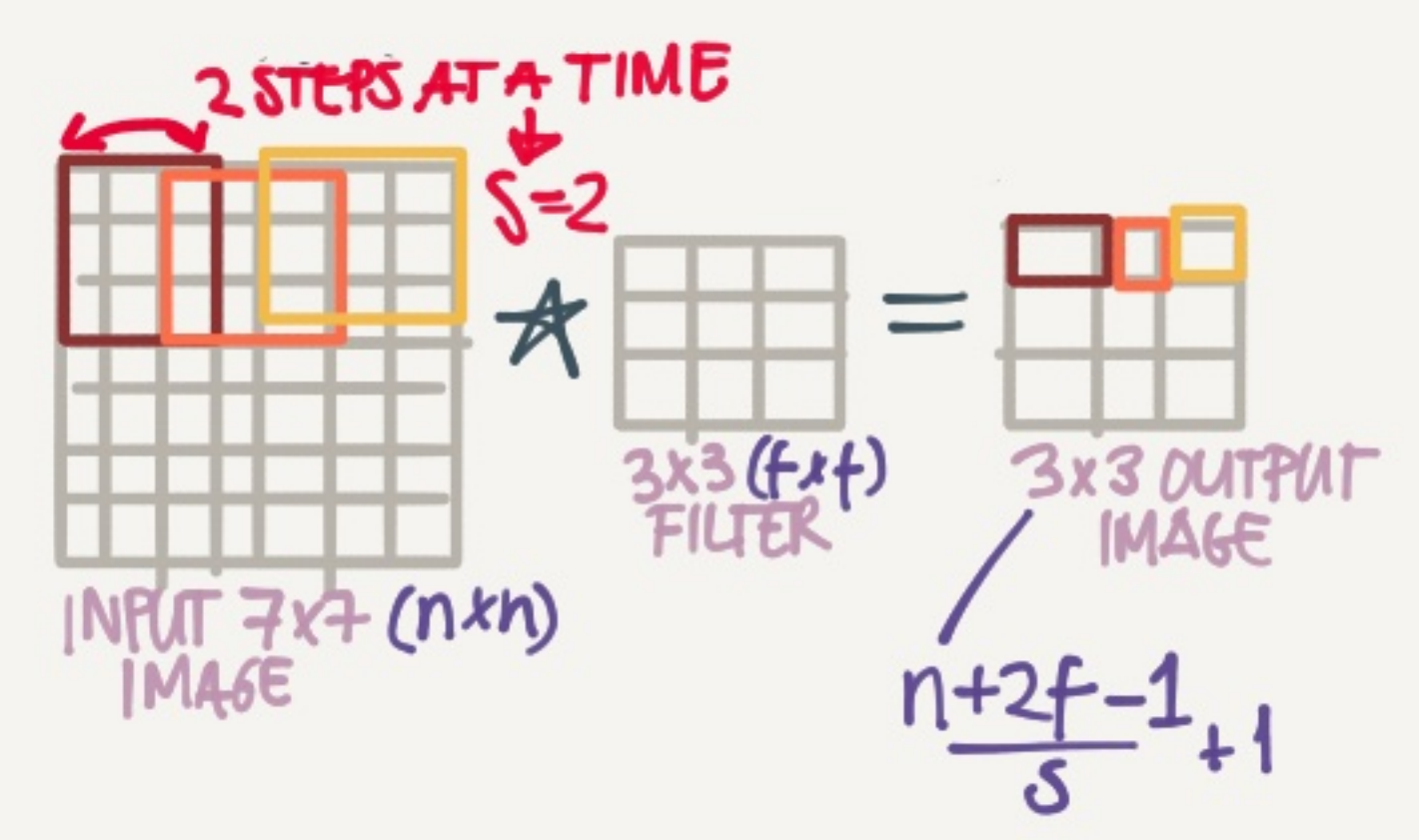
TWO COMMONLY USED PADDING OPTIONS
 (HOW MUCH TO PAD)

'VALID' $\Rightarrow P=0$ NO PADDING
 'SAME' $\Rightarrow P = \frac{f-1}{2}$ OUTPUT SIZE = INPUT SIZE
↑ FILTER SIZE

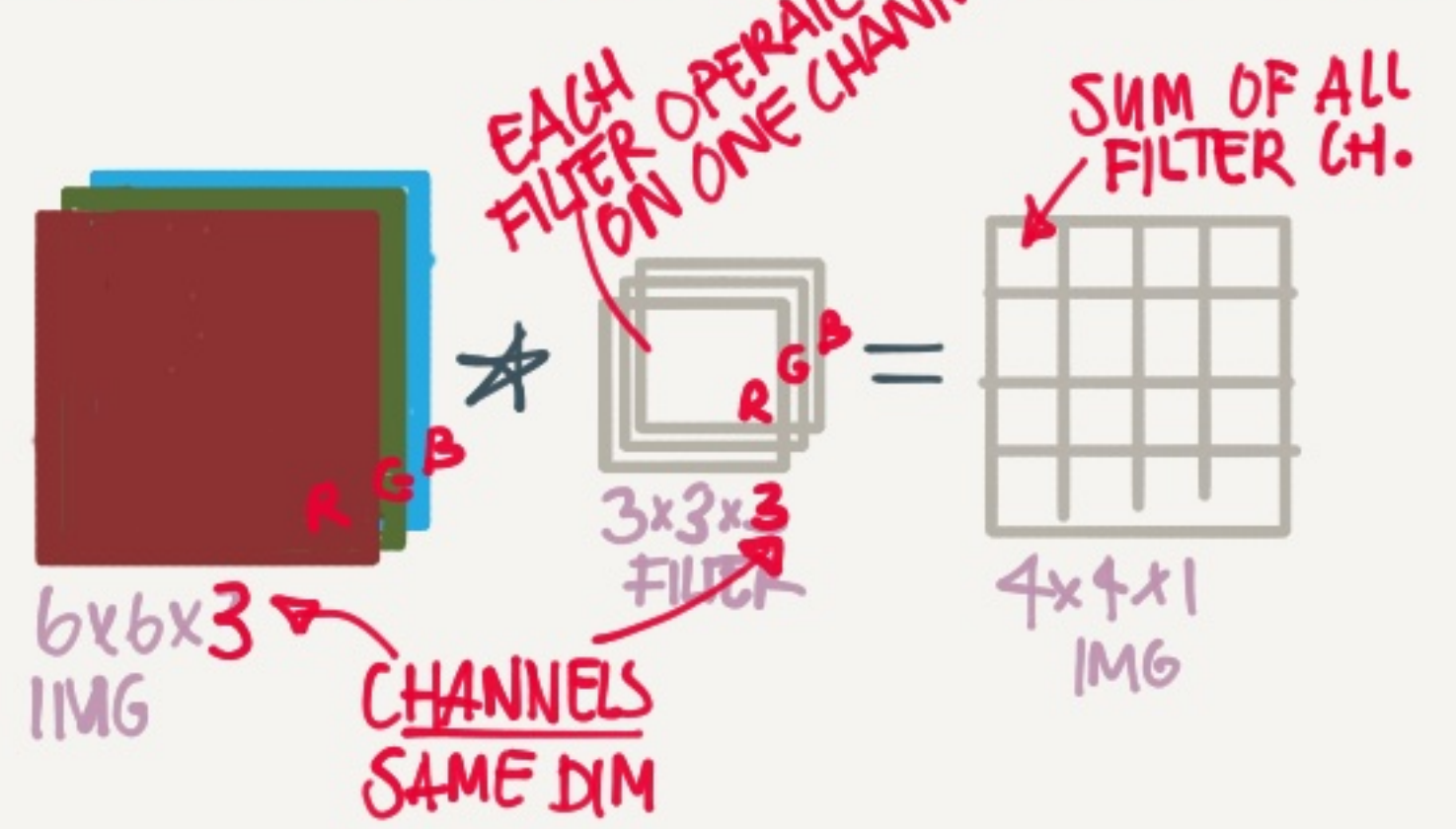
NOTE ALL CONVOLUTION IDEAS CAN BE APPLIED TO 1D AS WELL LIKE EKG SIGNALS • AND 3D LIKE CT-SCANS

STRIDE

WHAT PACE YOU SCAN WITH



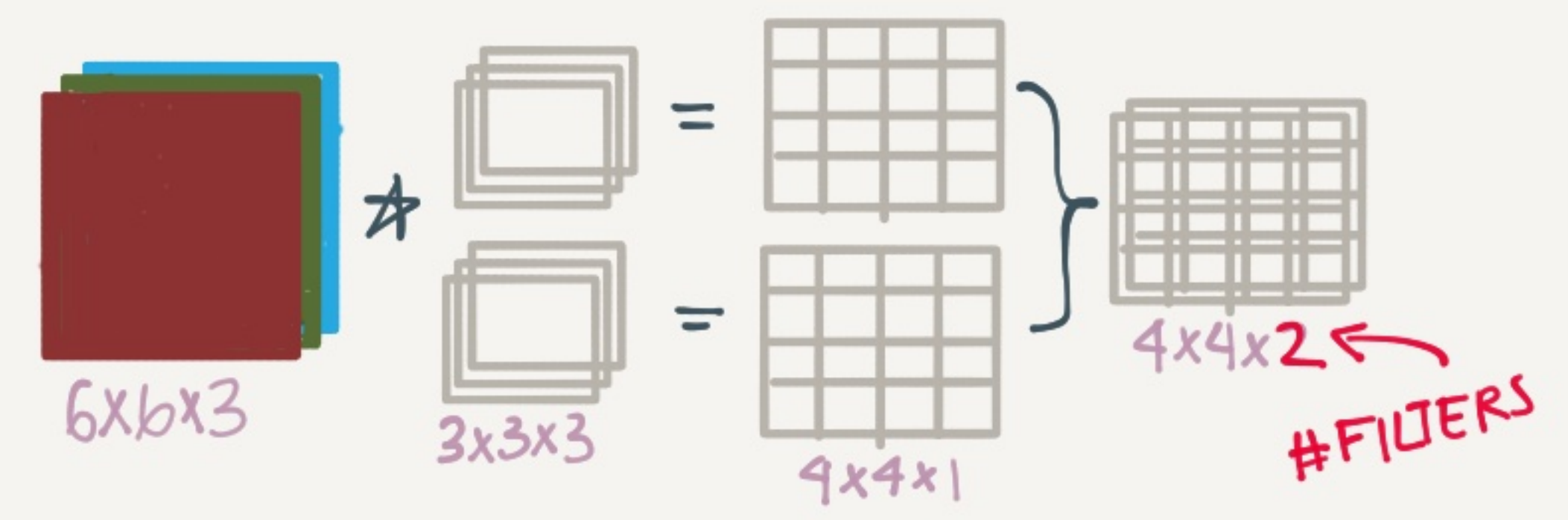
CONVOLUTIONS OVER VOLUMES



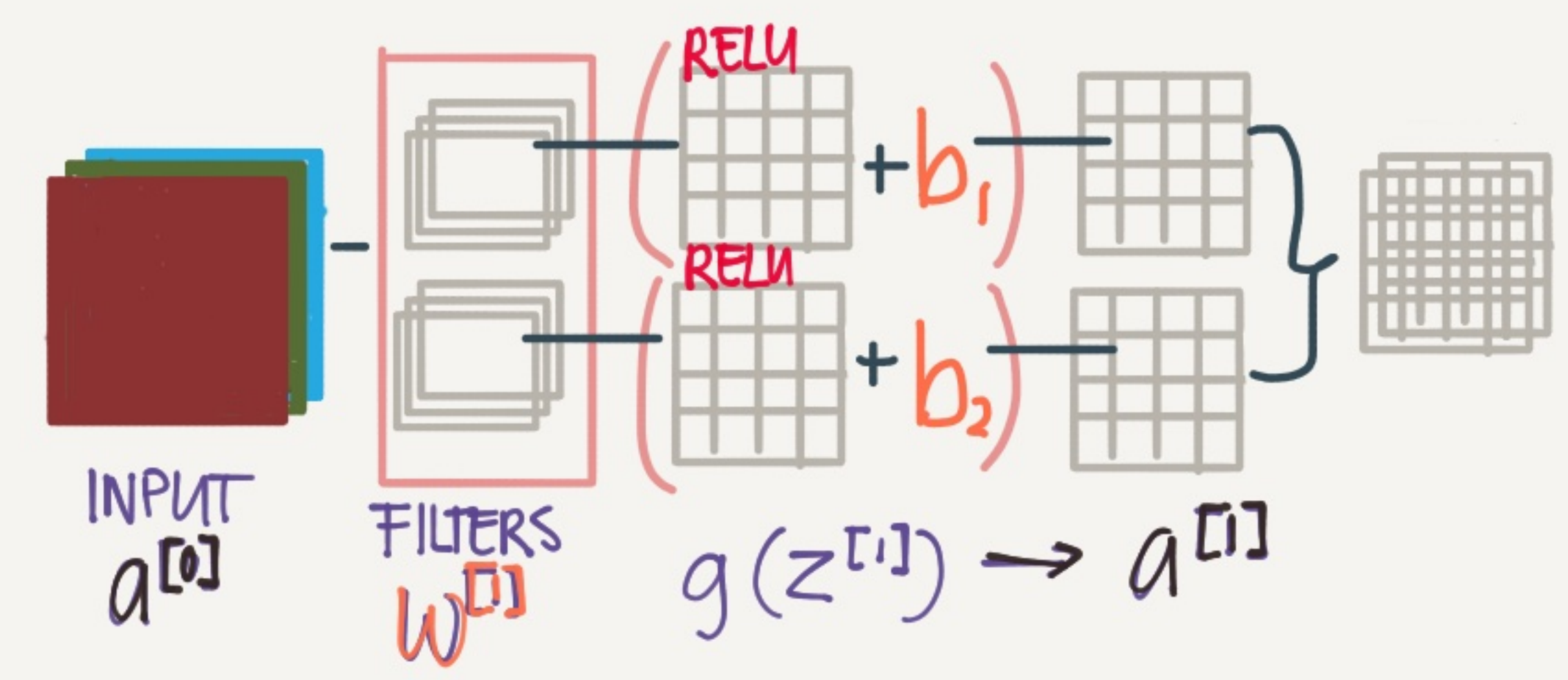
THIS ALLOWS US TO DETECT FEATURES IN COLOR IMAGES FOR EXAMPLE
 MAYBE WE WANT TO FIND ALL EDGES OR MAYBE ORANGE BLOBS

MULTIPLE FILTERS

DETECTING MULTIPLE FEATURES AT A TIME



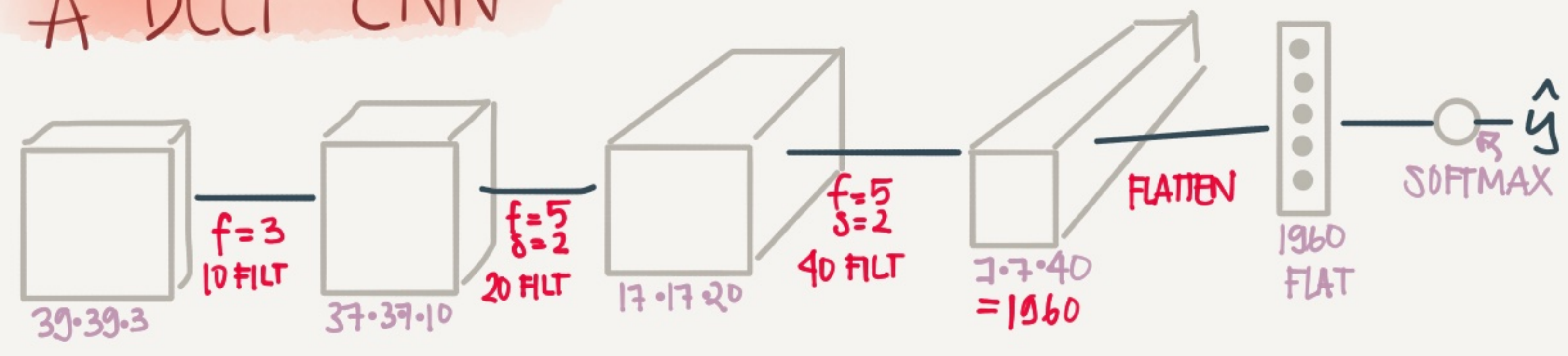
ONE CONV. NET LAYER



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

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

A DEEP CNN

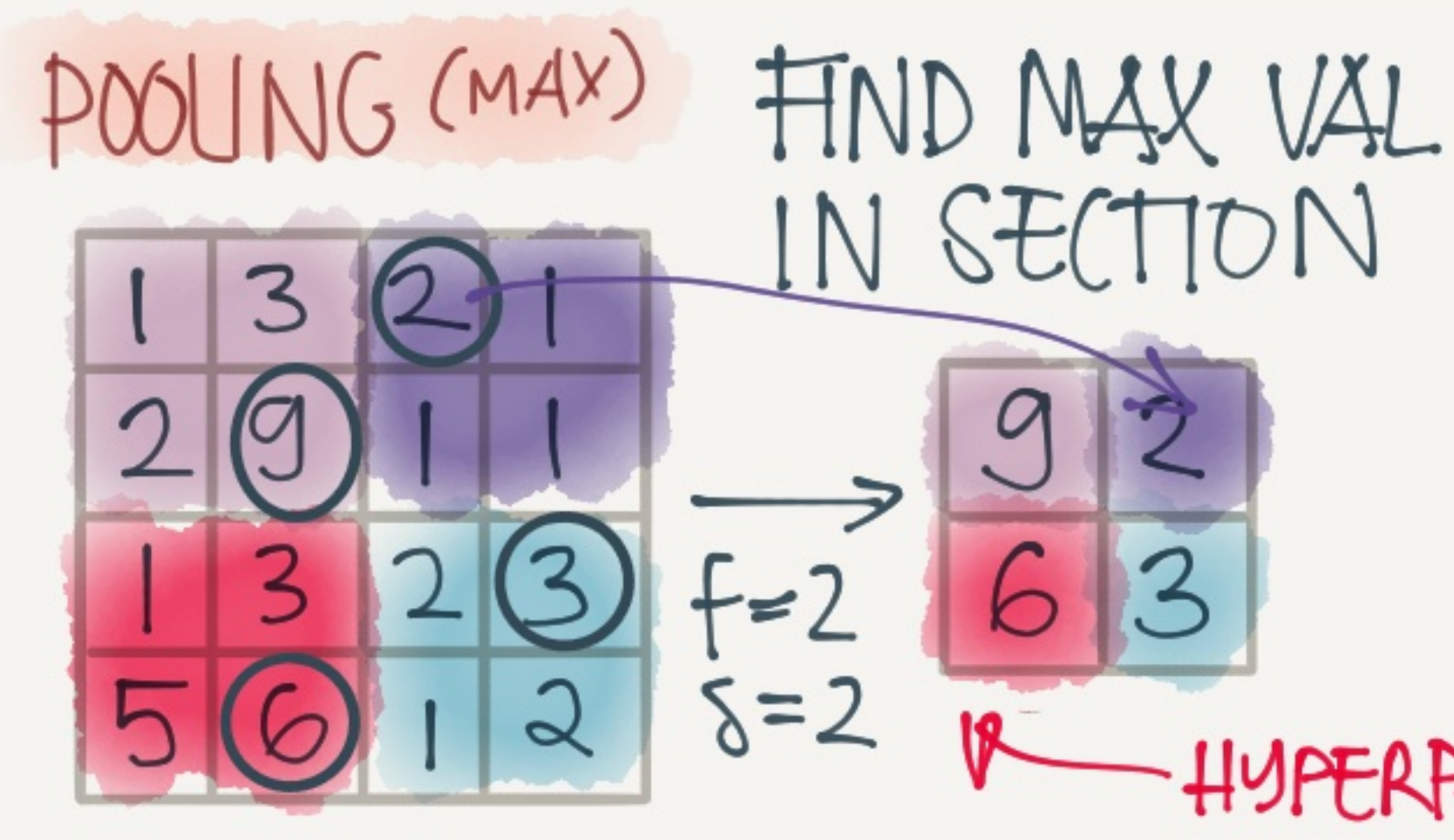


A LOT OF THE WORK IS FIGURING OUT HYPERPARAMS = #FILTERS, STRIDE, PADDING ETC

TYPICALLY SIZE \rightarrow TREND DOWN
FILTERS \rightarrow TREND UP

TYPICAL CONV.NET LAYERS

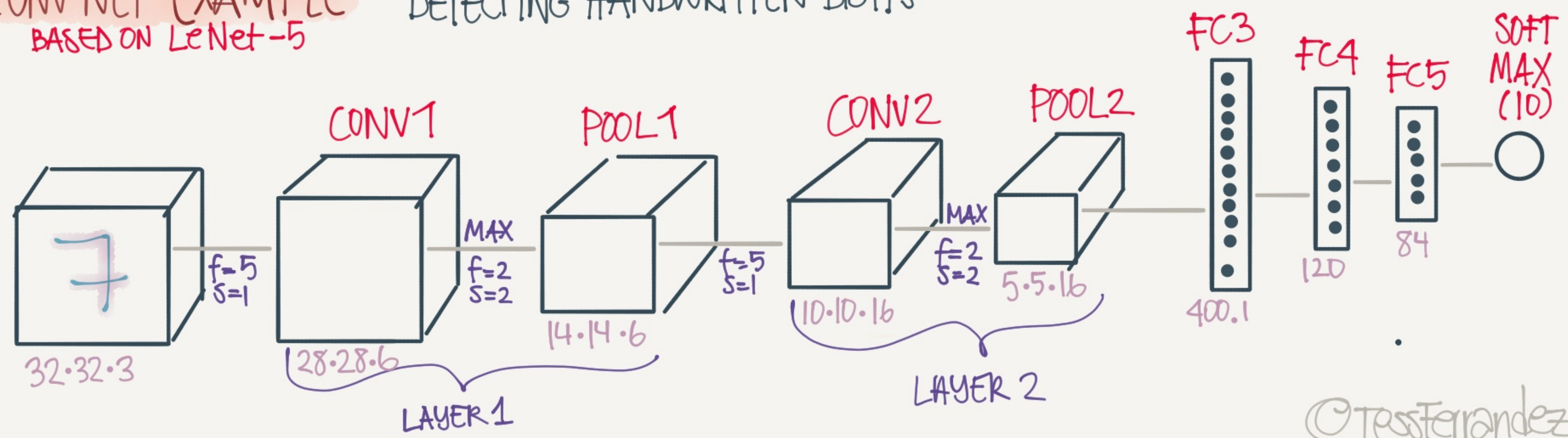
- CONVOLUTION
- POOLING
- FULLY CONNECTED



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

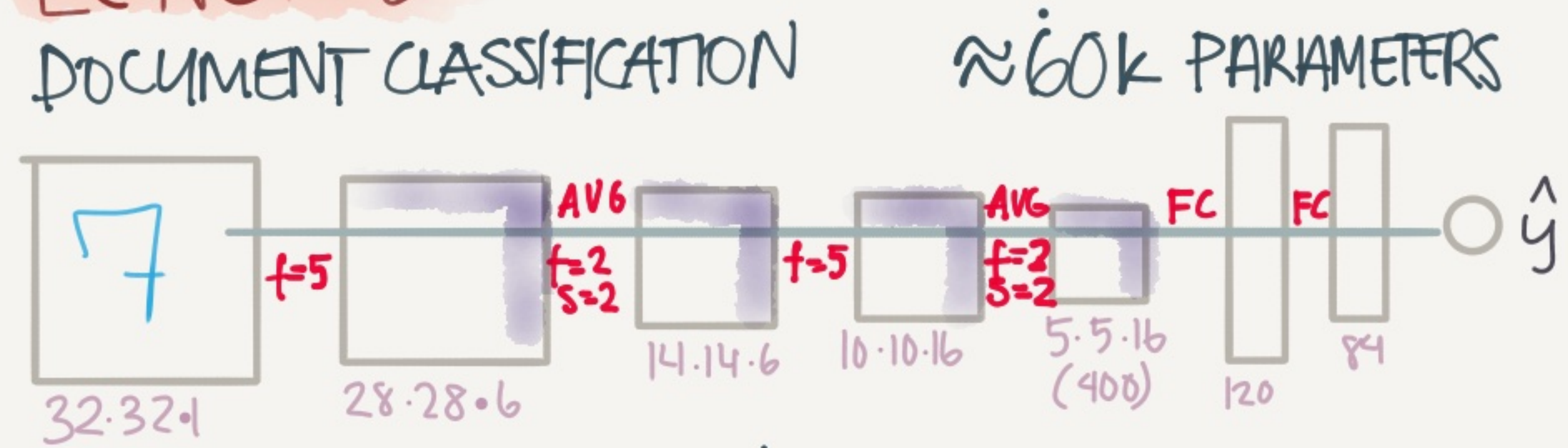
CONV NET EXAMPLE BASED ON LeNet-5

DETECTING HANDWRITTEN DIGITS



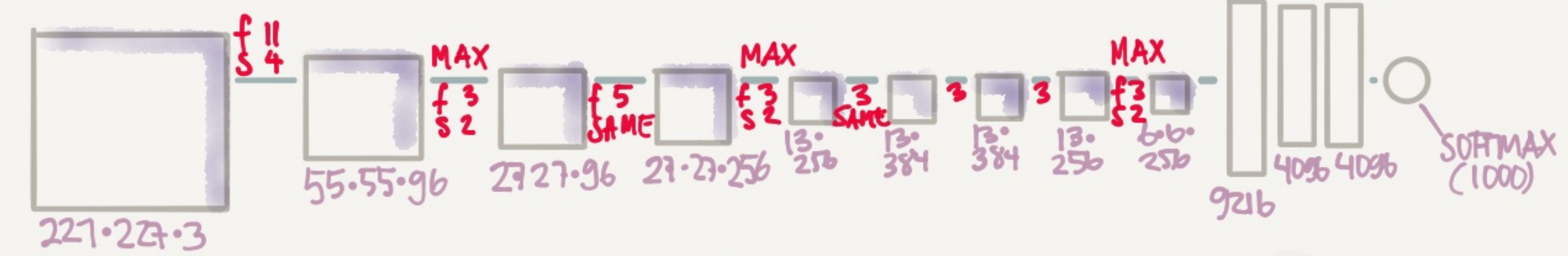
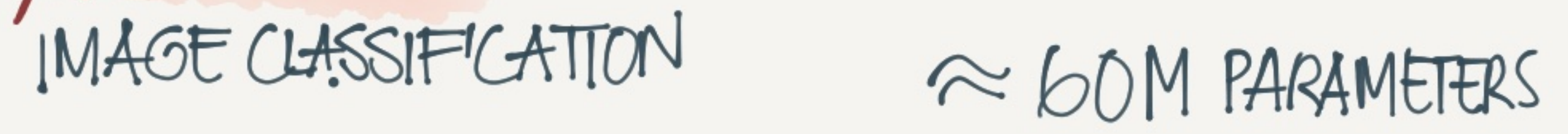
CLASSIC CONV. NETS

LeNet-5



- 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

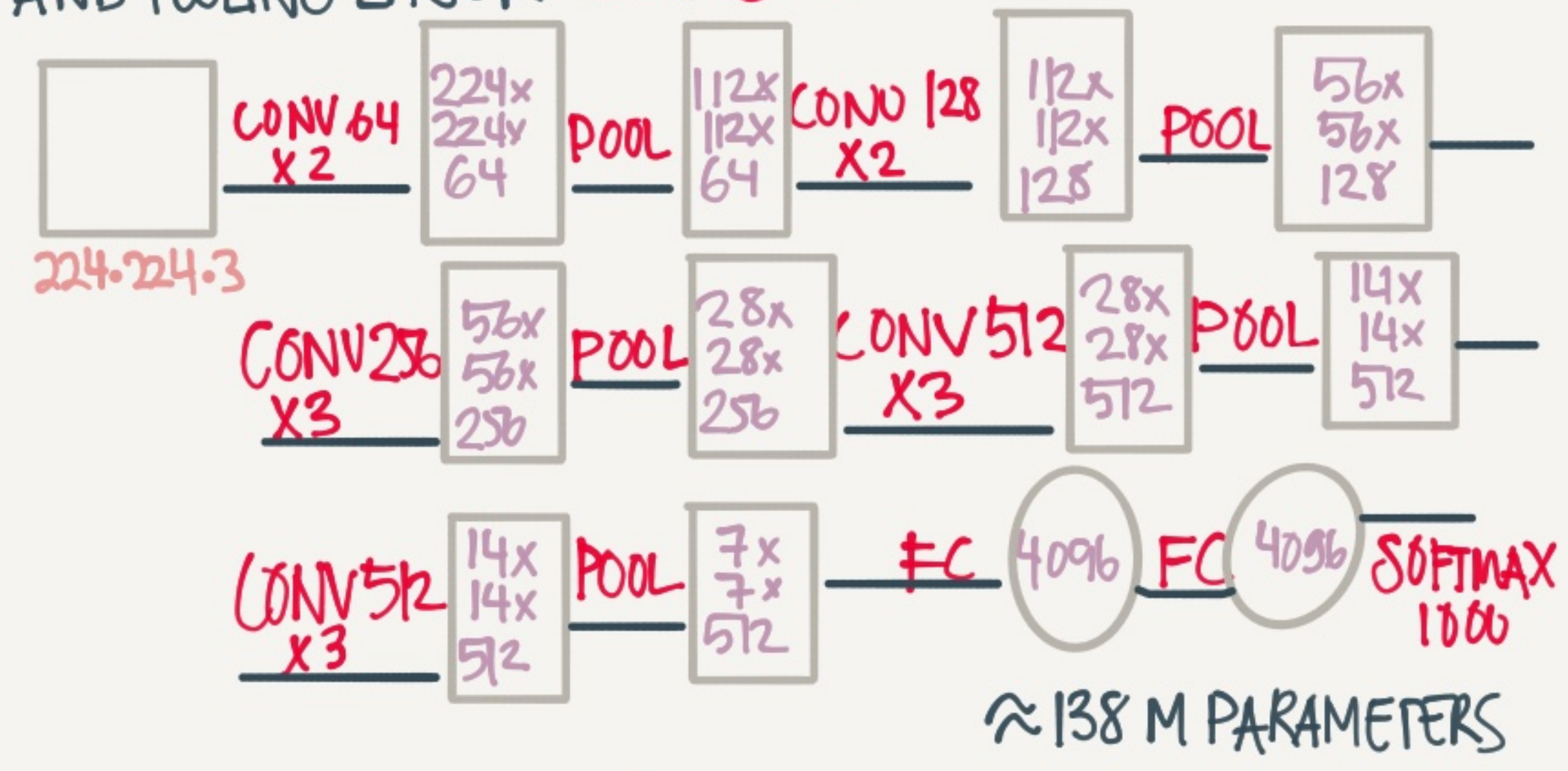
AlexNet



- 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
 $f=3 \times 3$ $s=1$ $p=SAME$
 AND POOLING LAYER 2×2 $s=2$



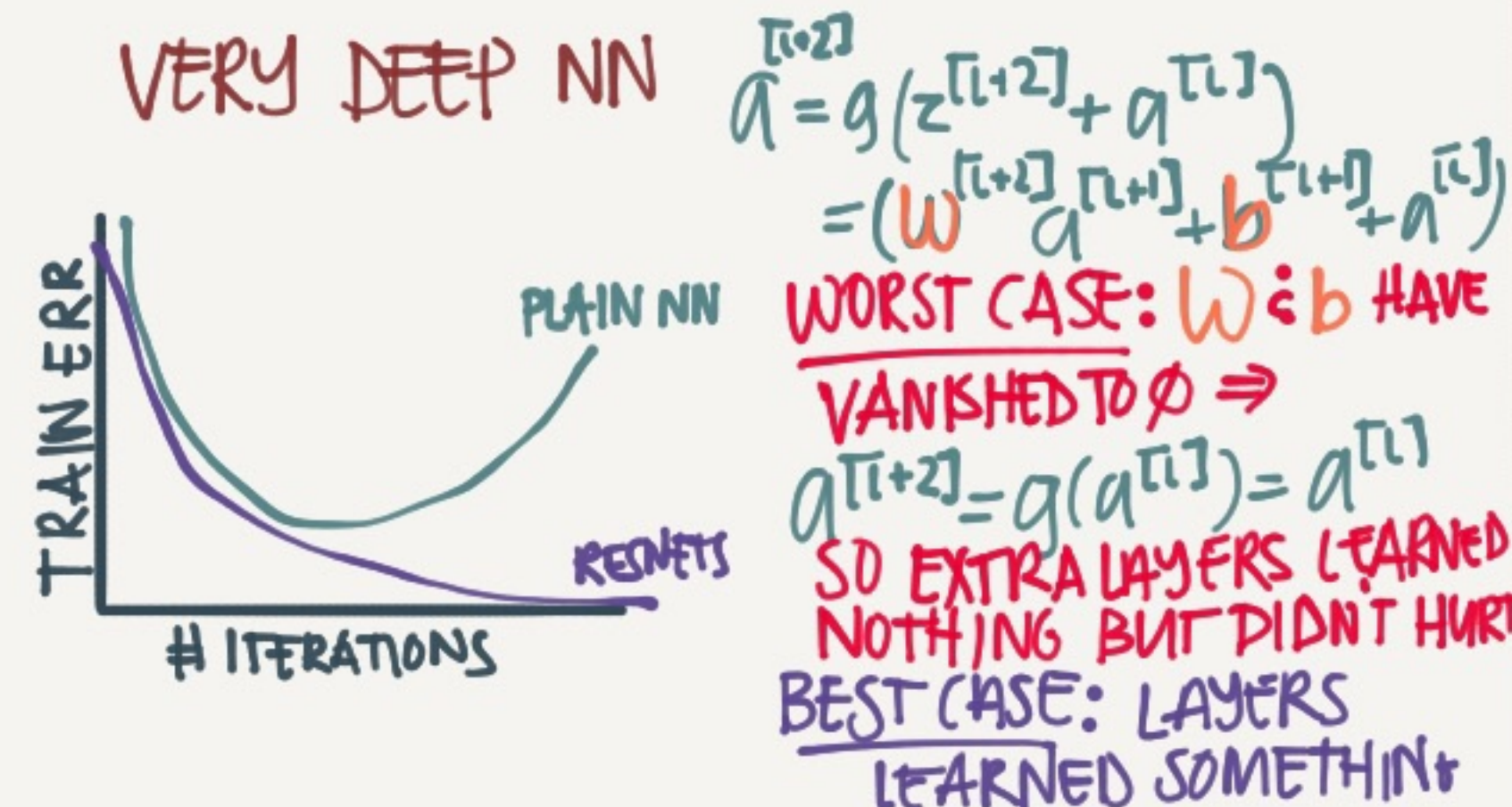
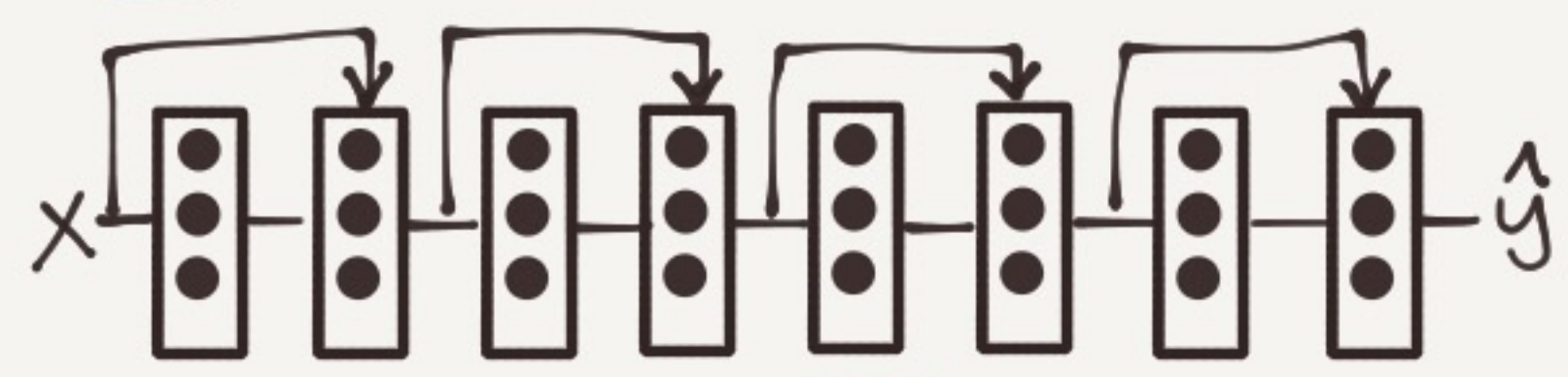
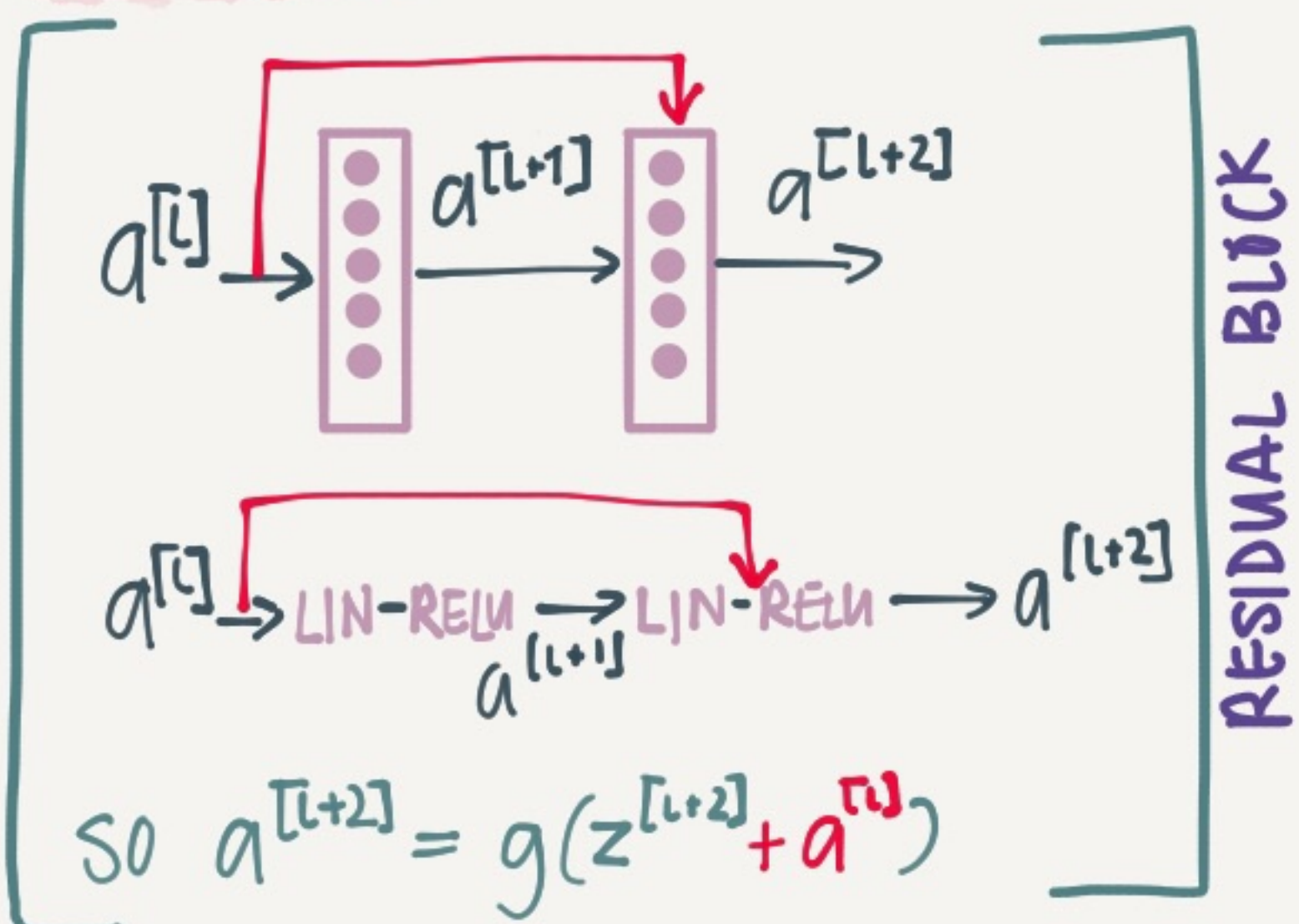
- VERY DEEP
- EASY ARCHITECTURE
- # FILTERS DOUBLE 64, 128, 256, 512

SPECIAL NETWORKS

ResNets

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

SOLUTION: RESIDUAL NETS



NETWORK IN NETWORK (1x1 CONVOLUTION)

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

 \star

2

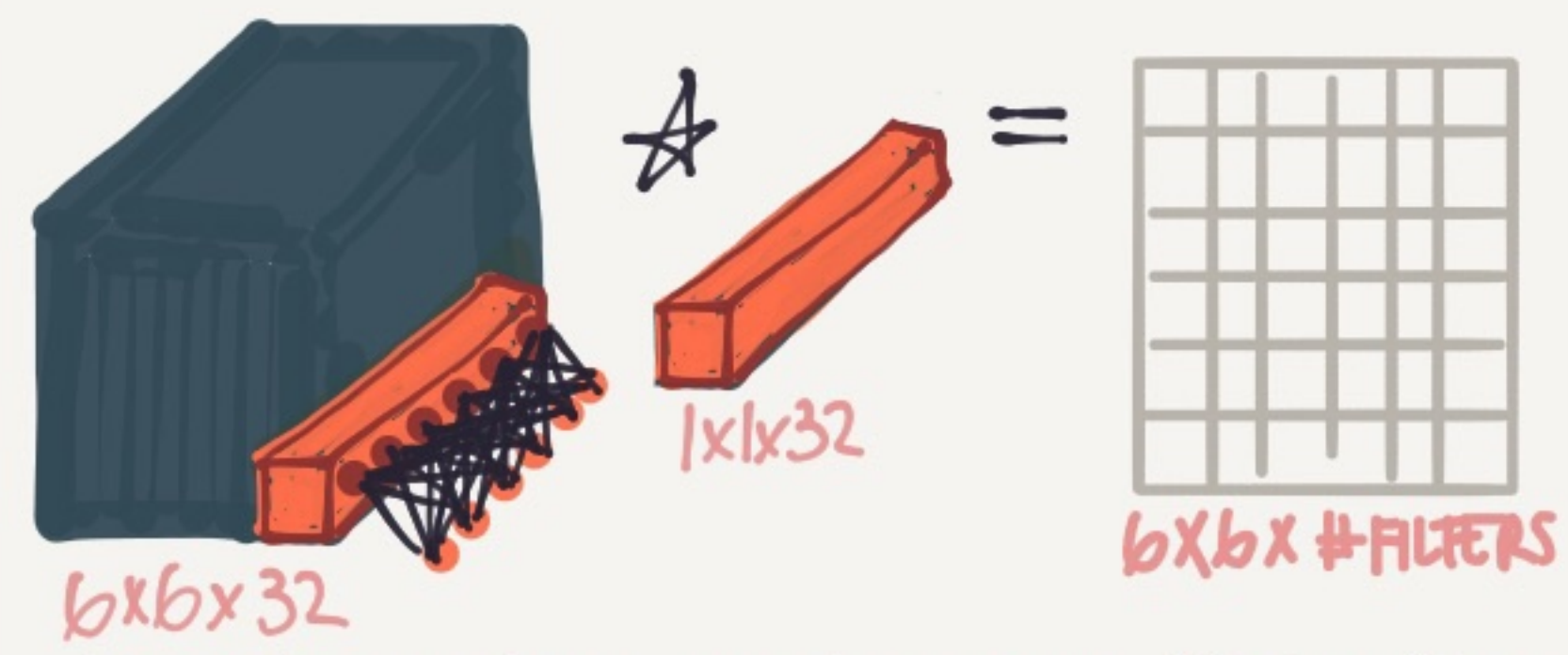
 $=$

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

1x1 CONVOLUTION

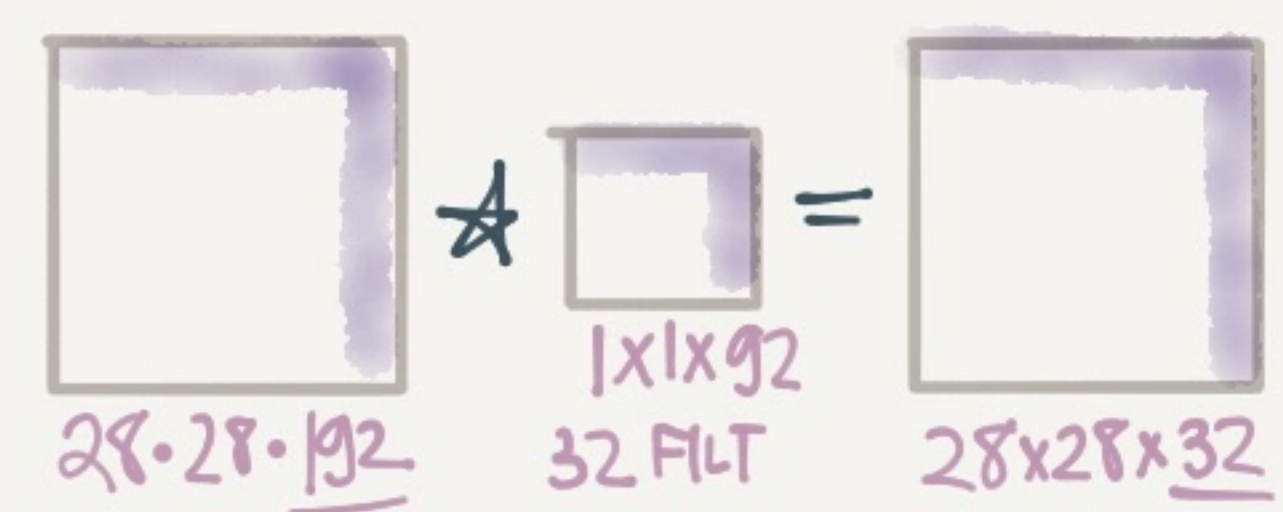
IT SEEMS PRETTY USELESS, BUT IT ACTUALLY SERVES 2 PURPOSES

1. NETWORK IN A NETWORK



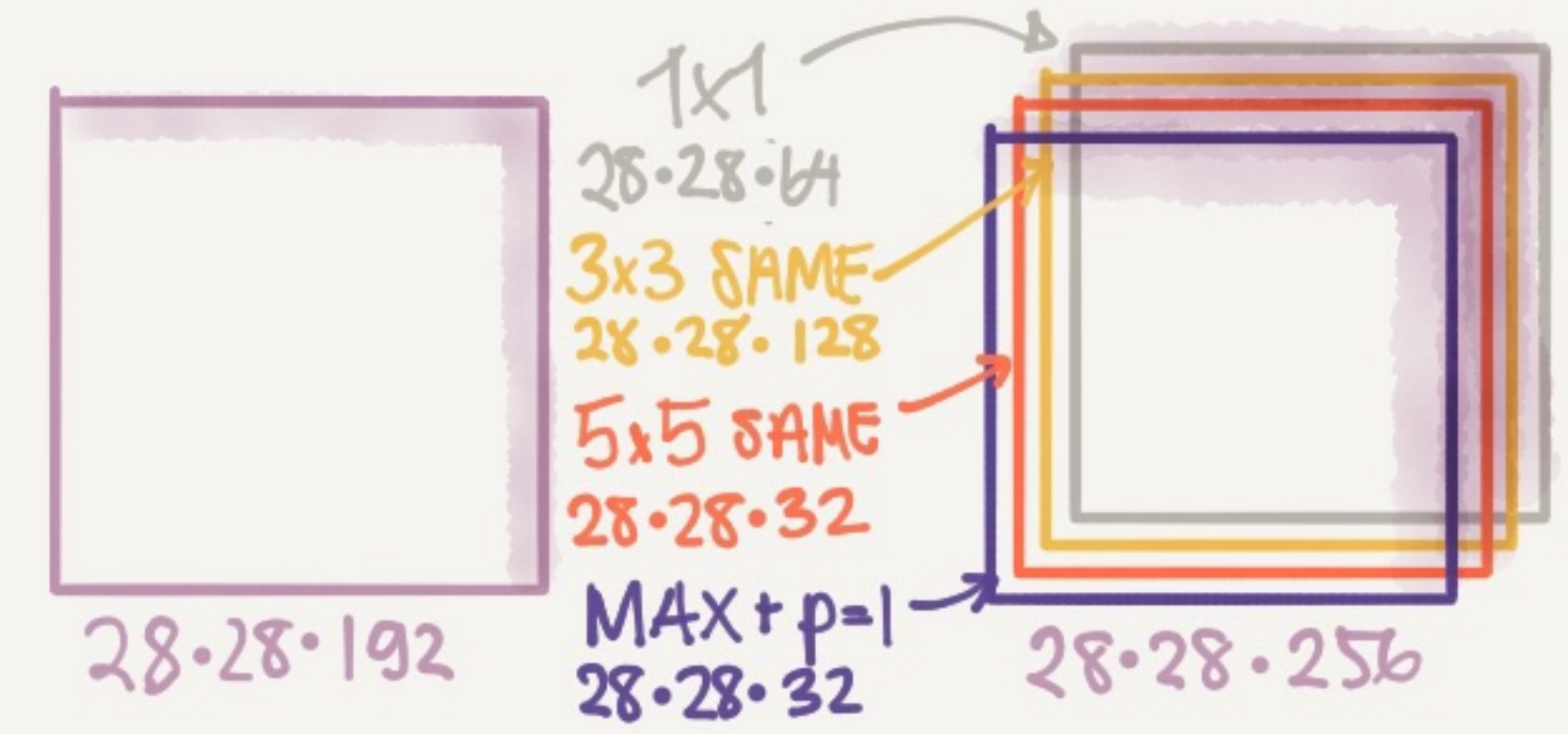
LEARNS COMPLEX, NON-LINEAR RELATIONSHIPS ABOUT A SLICE OF A VOLUME

2. REDUCING # CHANNELS



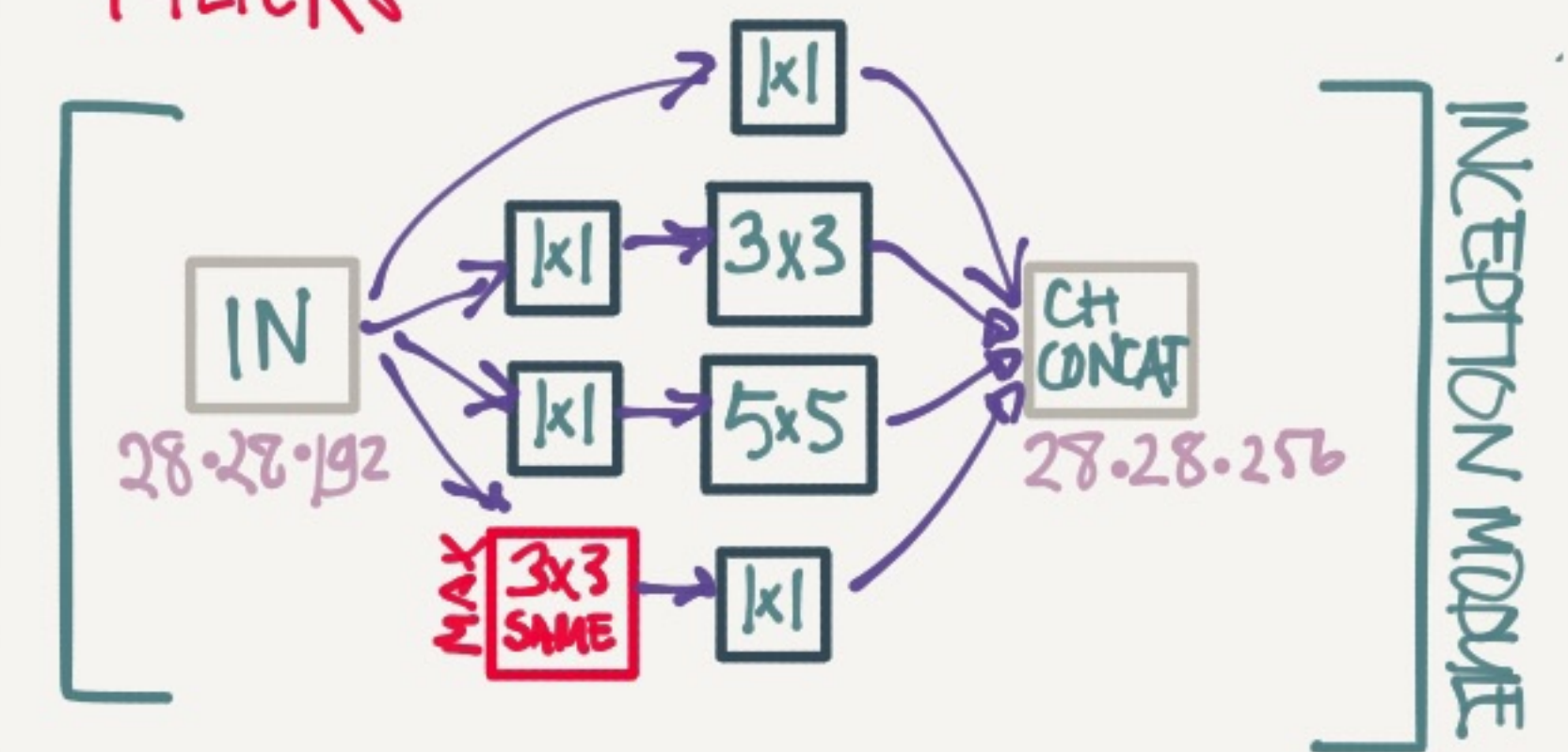
INCEPTION NETWORKS

INSTEAD OF CHOOSING A 1x1, 3x3, 5x5 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

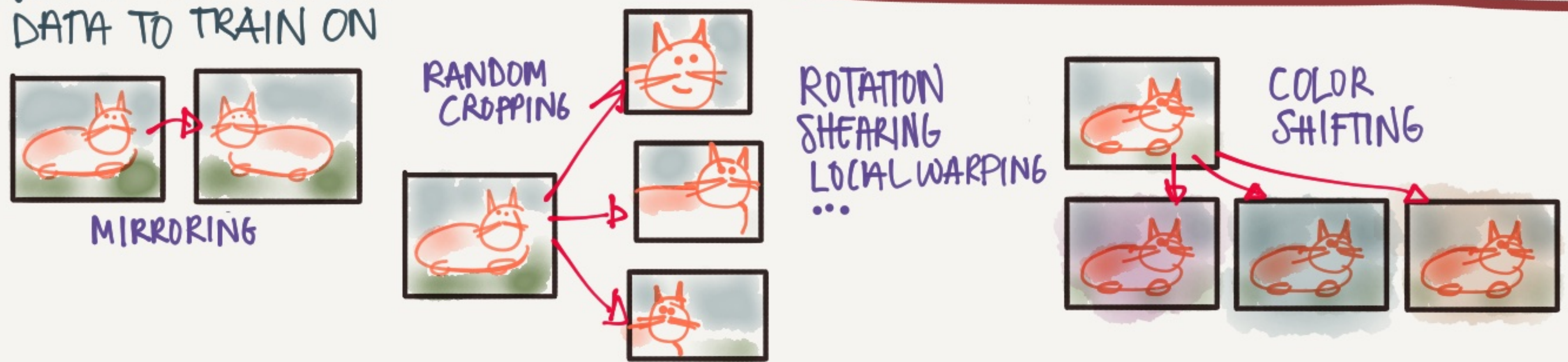
PRACTICAL ADVICE

USE OPEN SOURCE IMPLEMENTATIONS

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

DATA AUGMENTATION

WE ALMOST ALWAYS NEED MORE DATA TO TRAIN ON

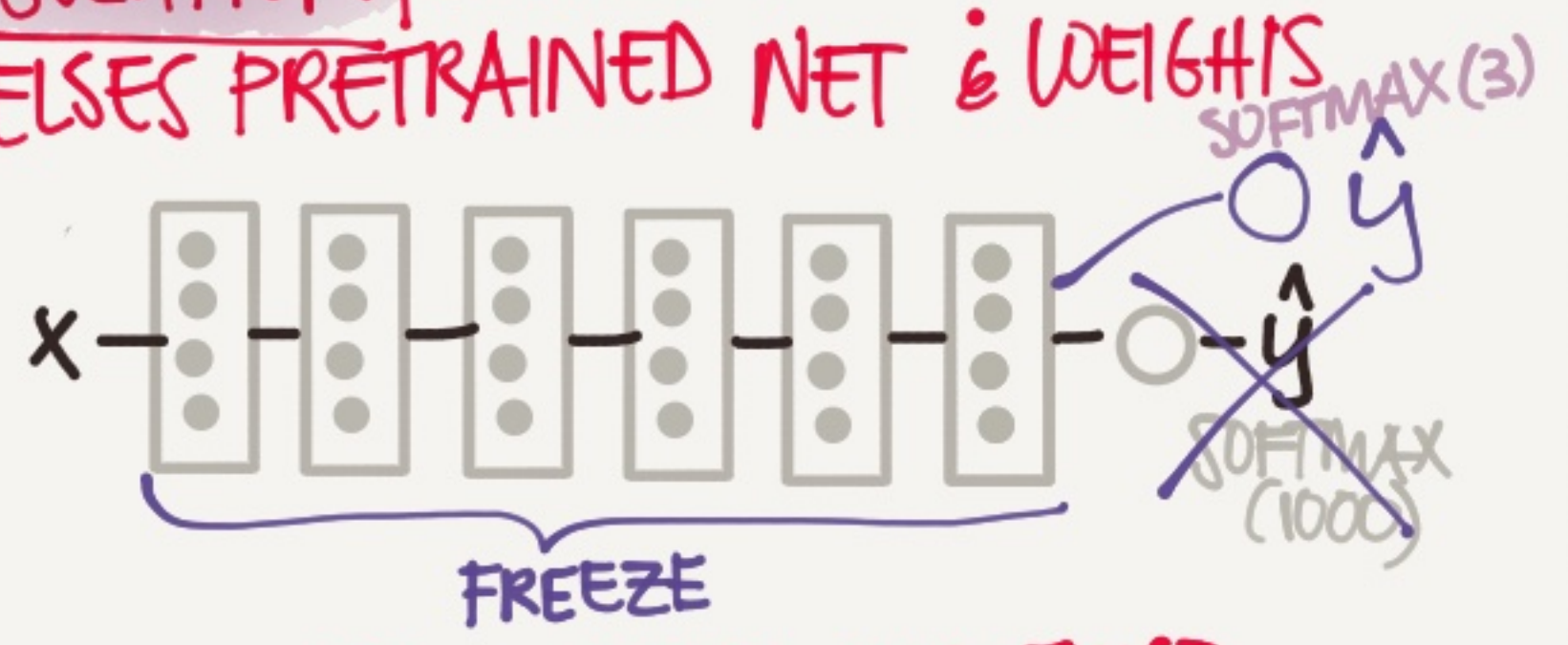


TRANSFER LEARNING



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

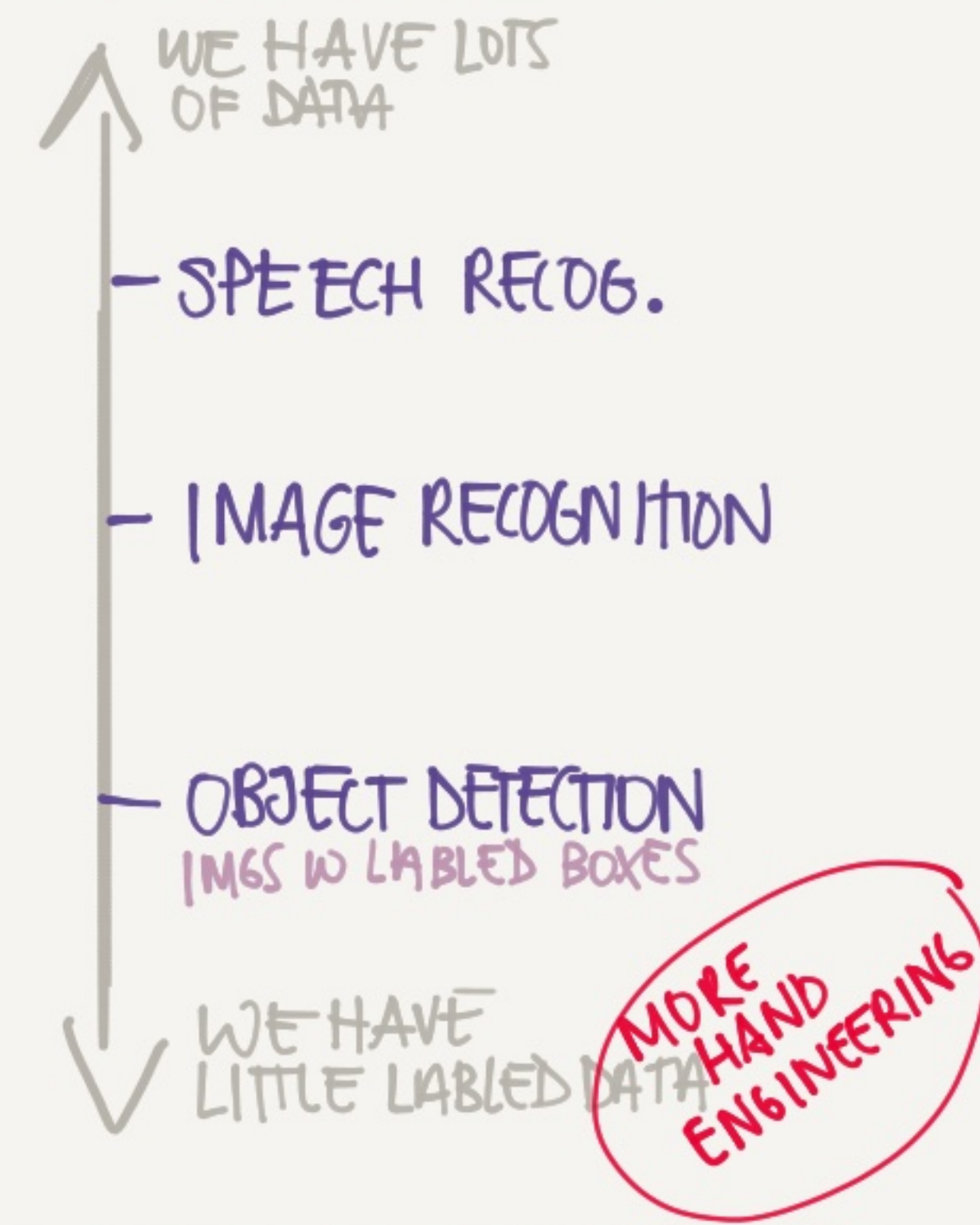
SOLUTION DOWNLOAD SOMEONE ELSE'S PRETRAINED NET & WEIGHTS



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

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

STATE OF COMPUTER VISION

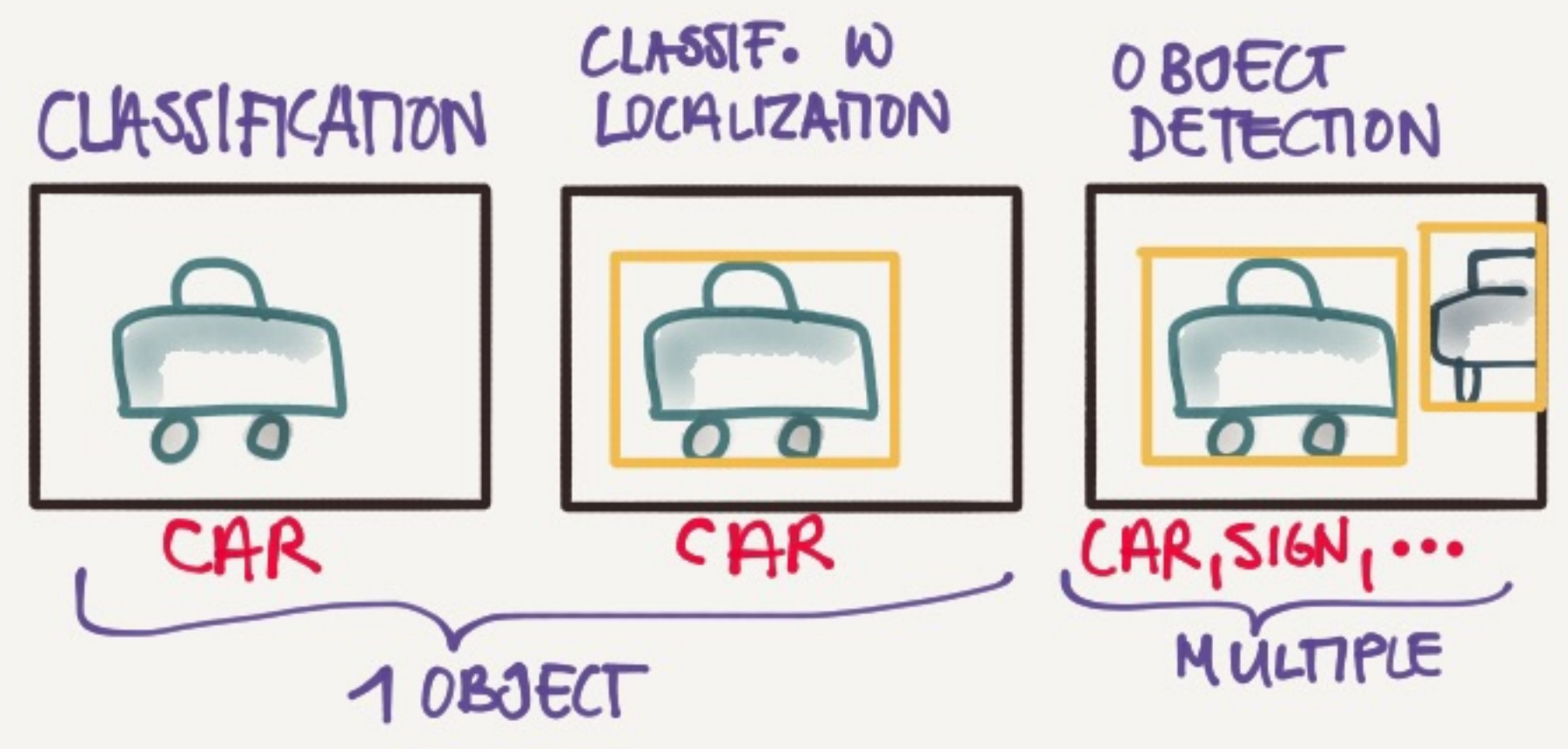


TIPS FOR DOING WELL ON BENCHMARKS/COMPETITIONS

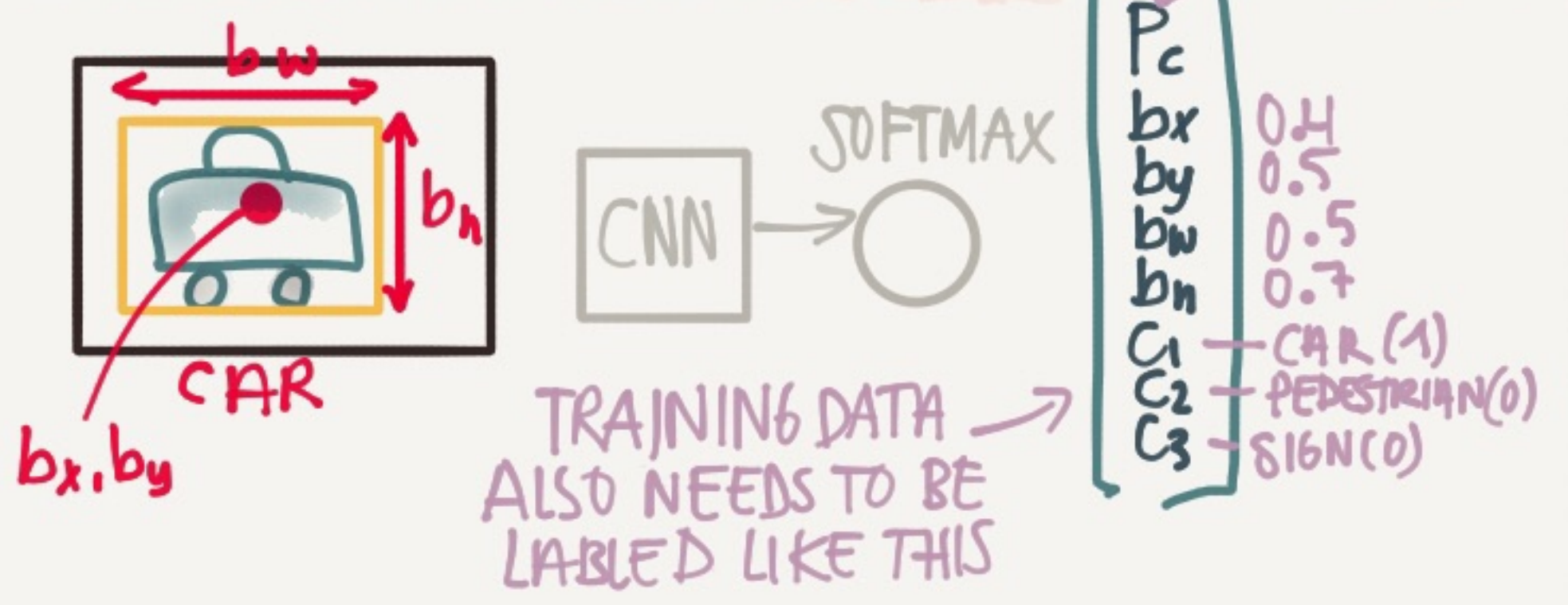
- * ENSEMBLING. AVG OUTPUTS FROM MULT NN
- * MULTI-CROP AT TEST TIME. AVG OUTPUTS FROM MULTIPLE CROPS OF THE IMAGE

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

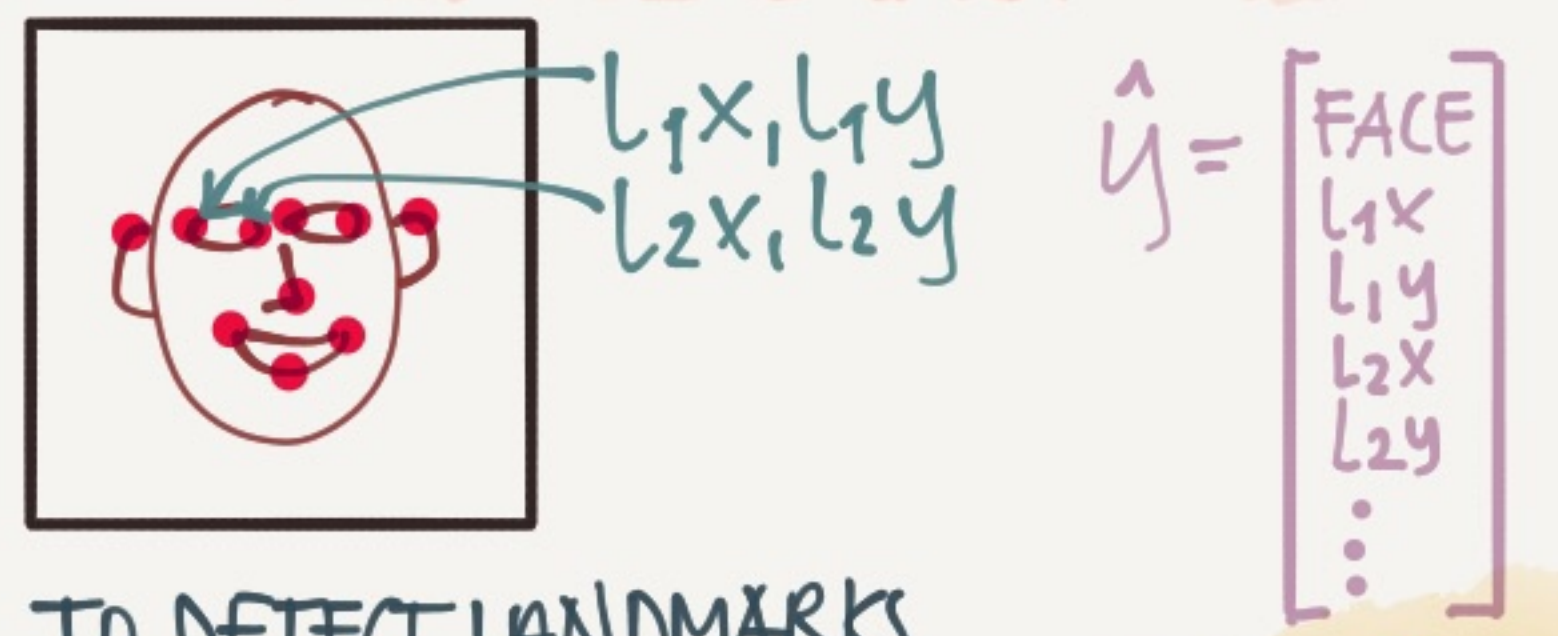
DETECTION ALGORITHMS



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.

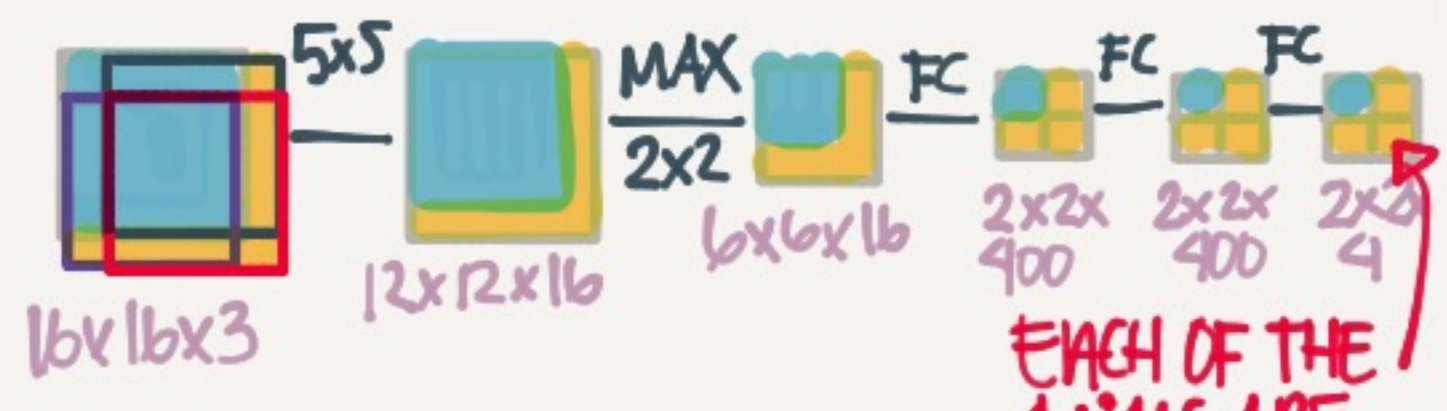
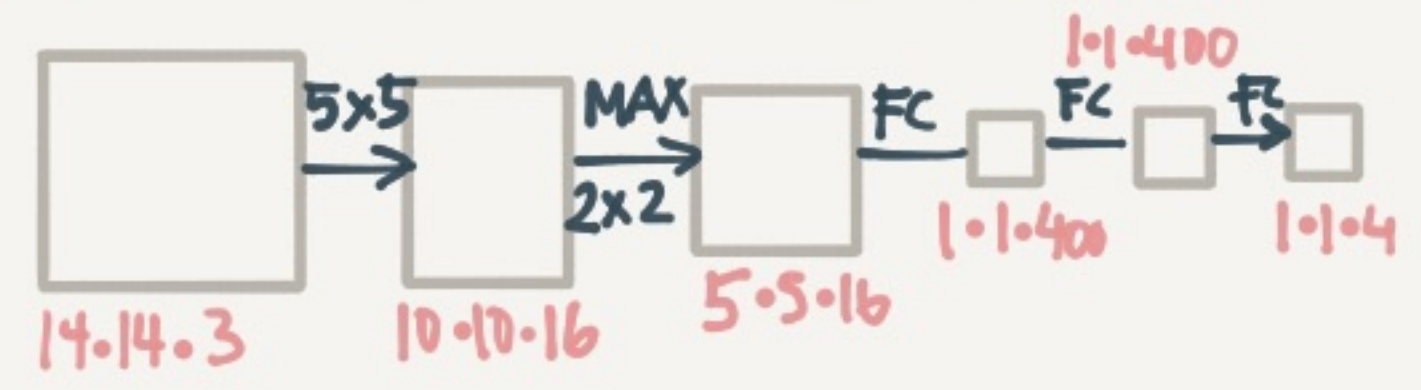
SLIDING WINDOWS DETECTION



1. CREATE TIGHTLY CROPPED IMG OF CARS (LOTS)
2. SLIDE A WINDOW OVER THE IMG. & CLASSIFY THIS WINDOW (CAR/NO) AGAINST YOUR OTHER CARS
3. REPEAT WITH SLIGHTLY LARGER WINDOW SIZE

PROBLEM: VERY EXPENSIVE (TO COMPUTE)

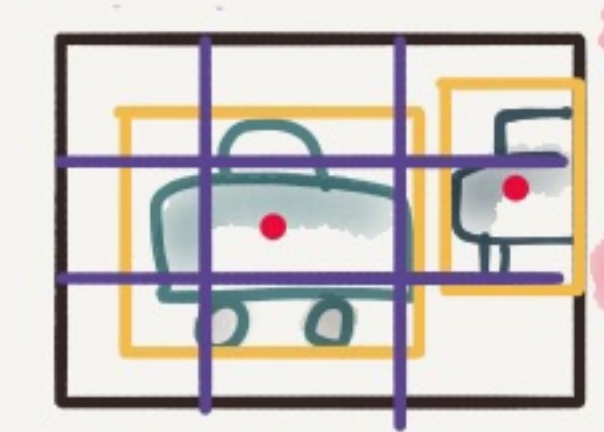
SINCE ADJ WINDOWS SHARE A LOT OF THE COMPUTATIONS WE CAN DO THIS MUCH CHEAPER W CONVOLUTIONS



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

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

YOLO: You Only Look Once



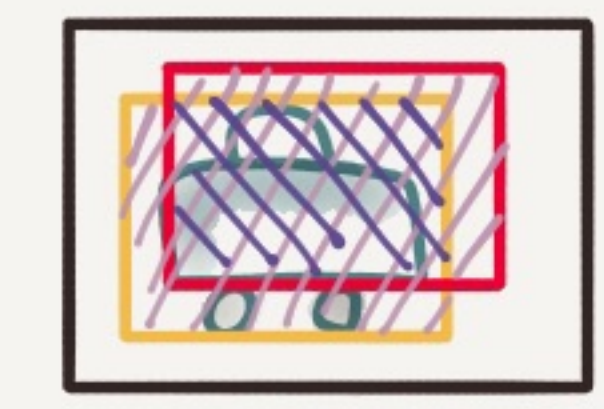
1. SPLIT IMG INTO X(9) GRID CELLS
2. FOR EACH CELL, SAY IF IT CONTAINS CAR + BOUNDING BOX (IF CELL CONTAINS THE MID POINT)

IN PRACTICE WE MIGHT HAVE A 19x19 GRID

$x \times y = 3 \times 3 \times 8$

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 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



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

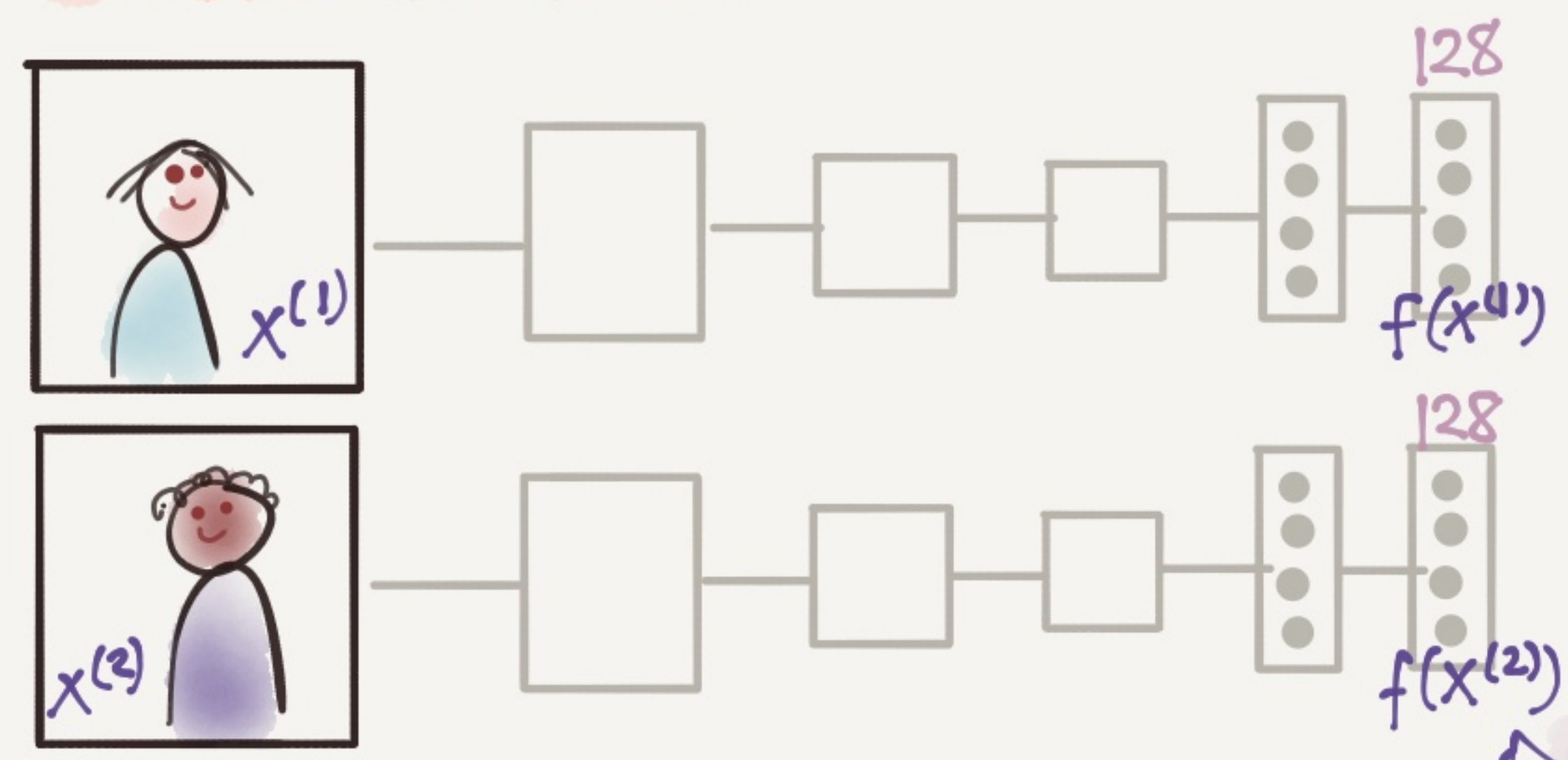
SOLUTION LEARN A SIMILARITY FUNCTION

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

BUT HOW DO YOU LEARN THIS?

SIAMESE NETWORK

DeepFace



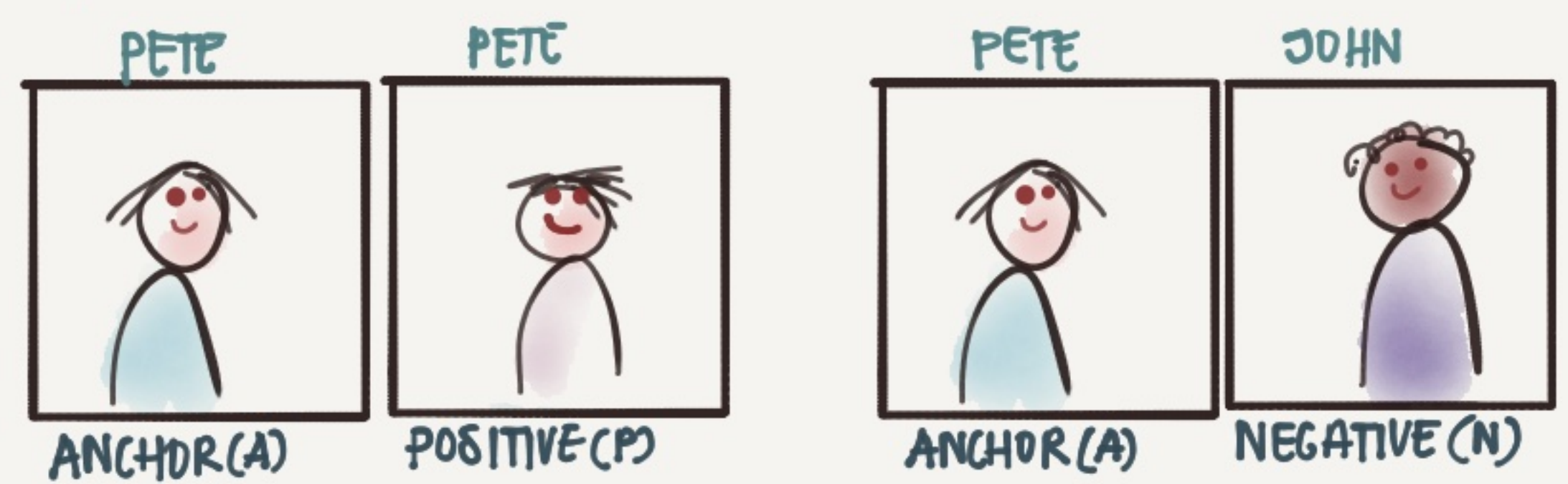
$$d(x^{(1)}, x^{(2)}) = \|f(x^{(1)}) - f(x^{(2)})\|_2^2$$

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



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

BUT WE WANT A GOOD MARGIN, SO...

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

HOW DO WE CHOOSE TRIPLETS TO TRAIN ON?

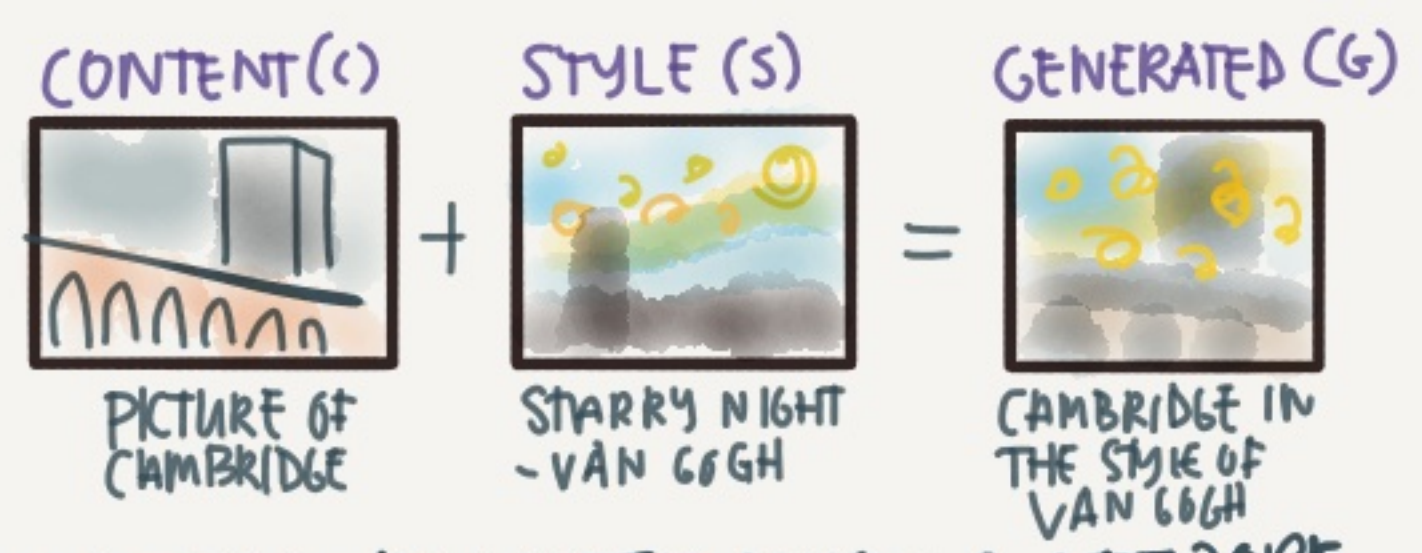
- IF A/P ARE VERY SIMILAR, & A/N ARE VERY DIFFERENT TRAINING IS VERY EASY.

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

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

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 (PARTS) ACTIVATED EACH UNIT MOST



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

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

↑
CONTENT
HOW SIMILAR ARE C & G

↑
STYLE
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

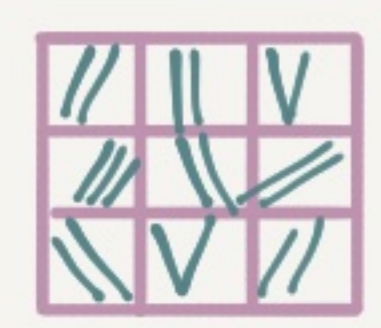
HOW DO WE TELL IF THEY ARE SIMILAR?

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

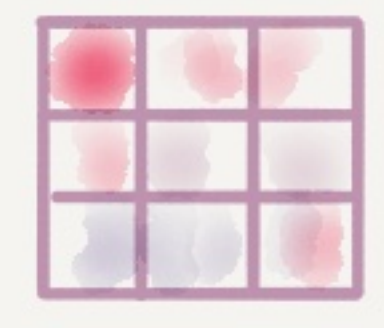
CAPTURING THE STYLE

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

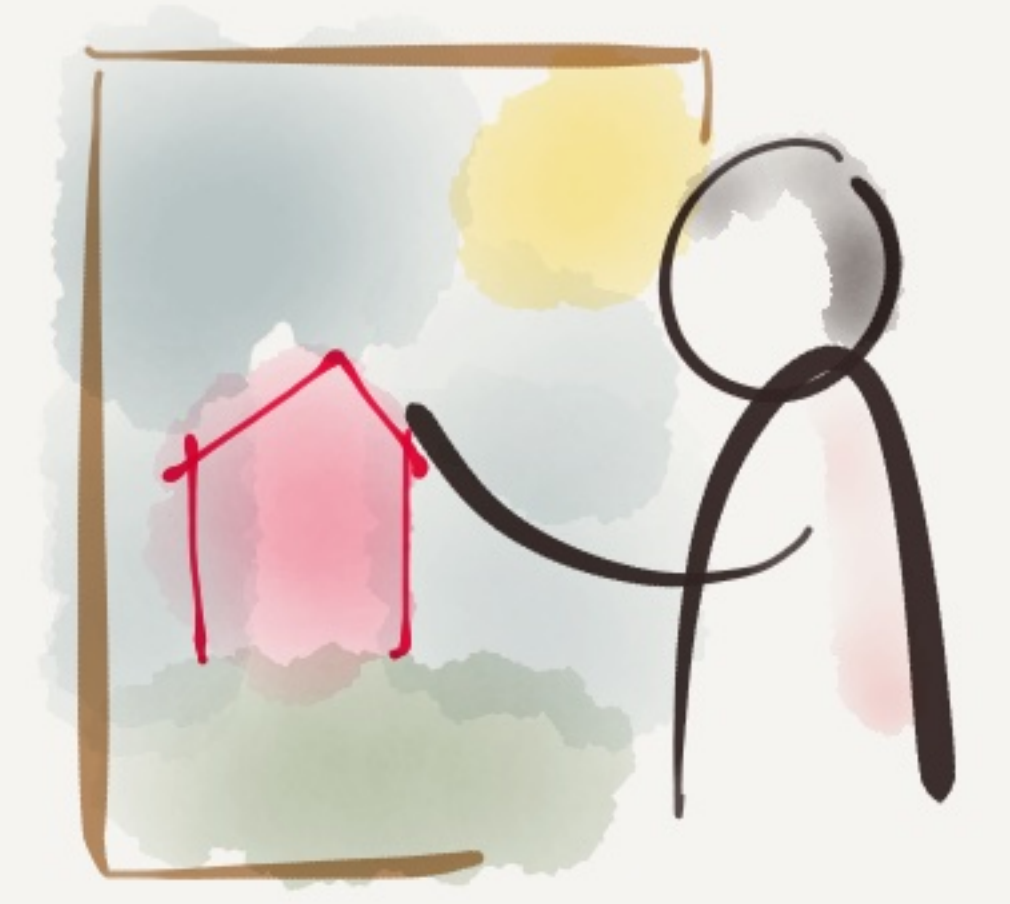
$$G_{KK'} = \sum_{i=1}^{n_H} \sum_{j=1}^{n_W} a_{ijk} \cdot a_{ijk'}$$

THE STYLE COST FUNCTION

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

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
	THE QUICK BROWN FOX JUMPED...	SPEECH RECOGNITION
		MUSIC GENERATION
THERE IS NOTHING TO LIKE IN THIS MOVIE		SENTIMENT CLASSIFICATION
AGCCCTGTG AGGAACTAG	AGCCCTGTG AGGAACTAG	DNA SEQUENCE ANALYSIS
Vouslez-vous chanter avec moi?	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

$X =$ HARRY POTTER AND HERMIONE GRANGER $T_x = 9$ (9 words)
 $x^{(1)}$ $x^{(2)}$...
 GRANGER INVENTED A NEW SPELL

$y =$

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

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

HOW DO WE REPRESENT WORDS?

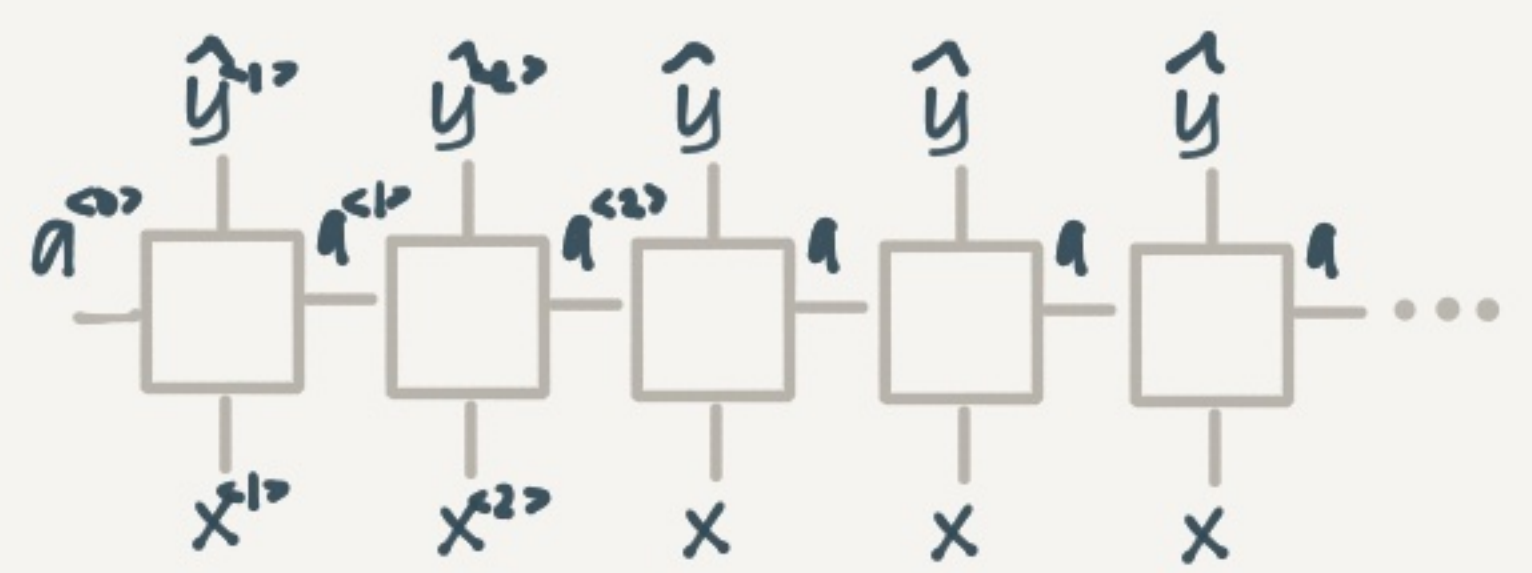
CREATE A VOCABULARY (EG 10K MOST COMMON WORDS IN YOUR TEXTS • OR DOWNLOAD EXISTING)

EACH WORD IS A **ONE-HOT VECTOR**
 $\begin{bmatrix} a \\ aaron \\ and \\ Harry \\ Potter \\ zulu \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 367 \\ 4075 \\ 6830 \\ 10000 \end{matrix}$
 $HARRY = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ (index 4075)

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.

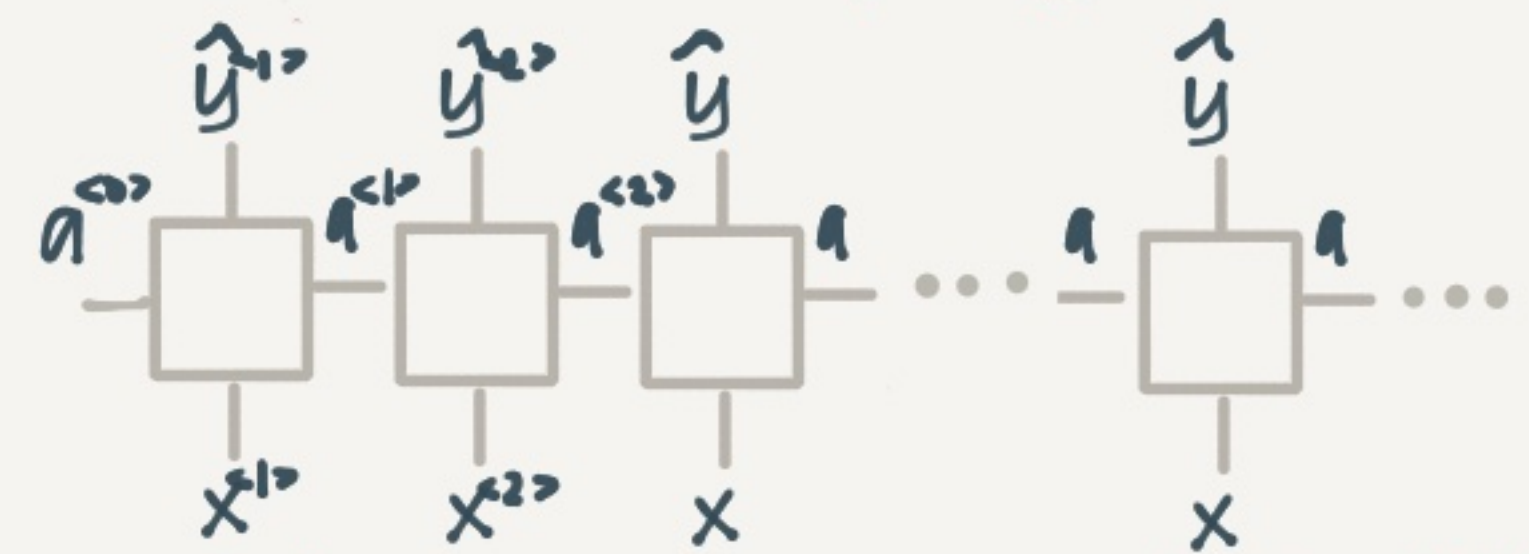
$$a^{(i)} = g_1(W_a [a^{(i-1)} x^{(i)}] + b_a) \text{ TANH/RELU}$$

$$\hat{y}^{(i)} = g_2(W_{y_a} a^{(i)} + b_y) \text{ SIGMOID}$$

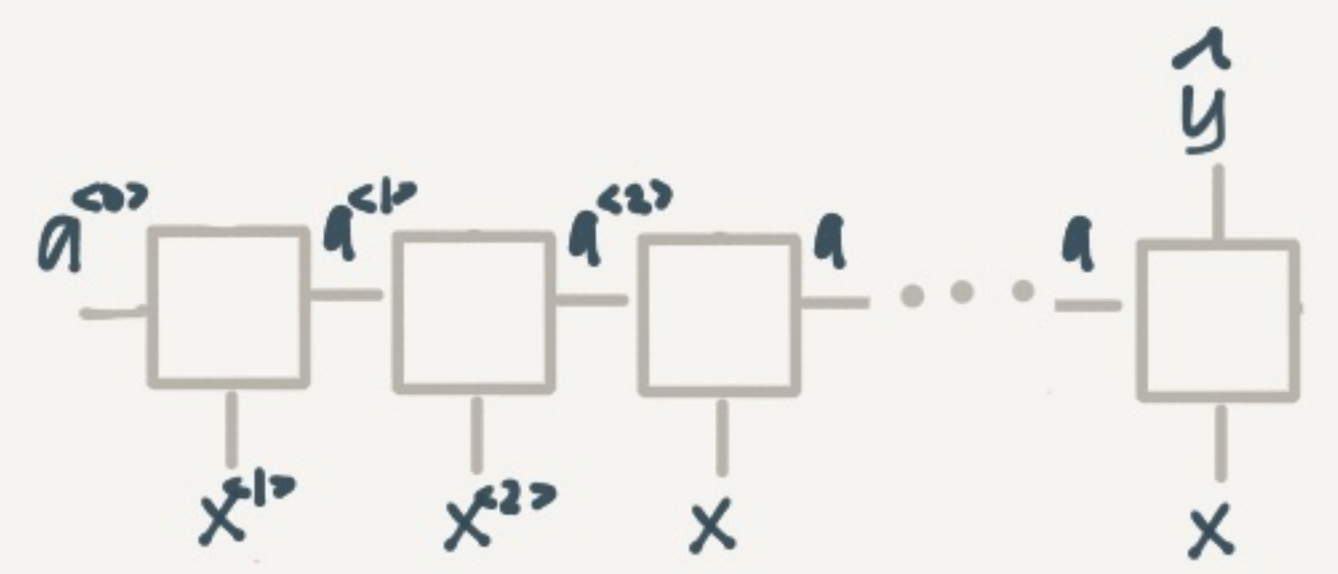
THE SAME W & b ARE USED IN ALL TIME STEPS
 THE LOSS WE OPTIMIZE IS THE SUM OF $\mathcal{L}(\hat{y}, y)$ FROM 1-T

DIFFERENT TYPES OF RNN

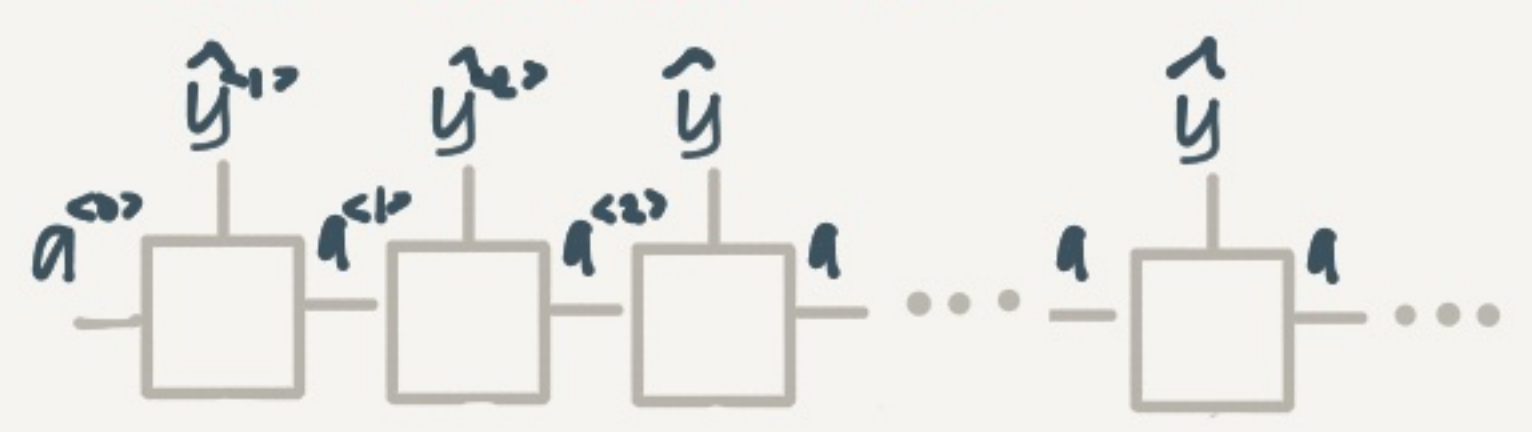
MANY-TO-MANY $T_x = T_y$



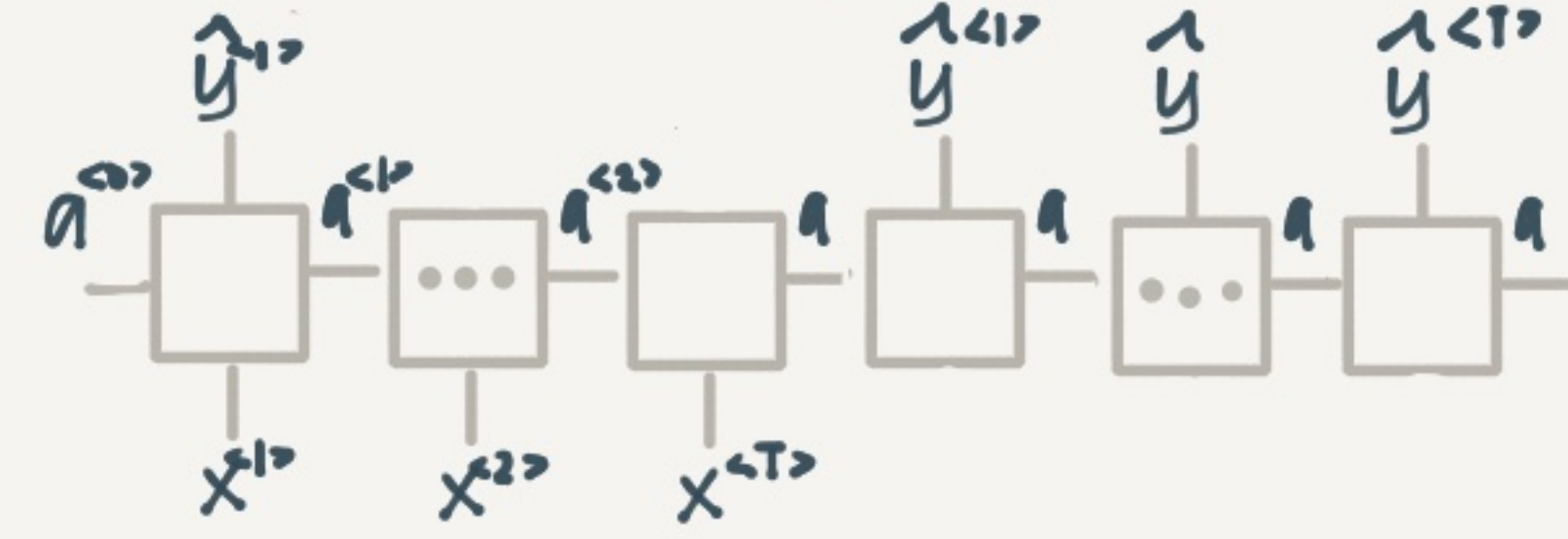
MANY-TO-ONE EX. SENTIMENT ANALYSIS



ONE-TO-MANY • MUSIC GENERATION



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



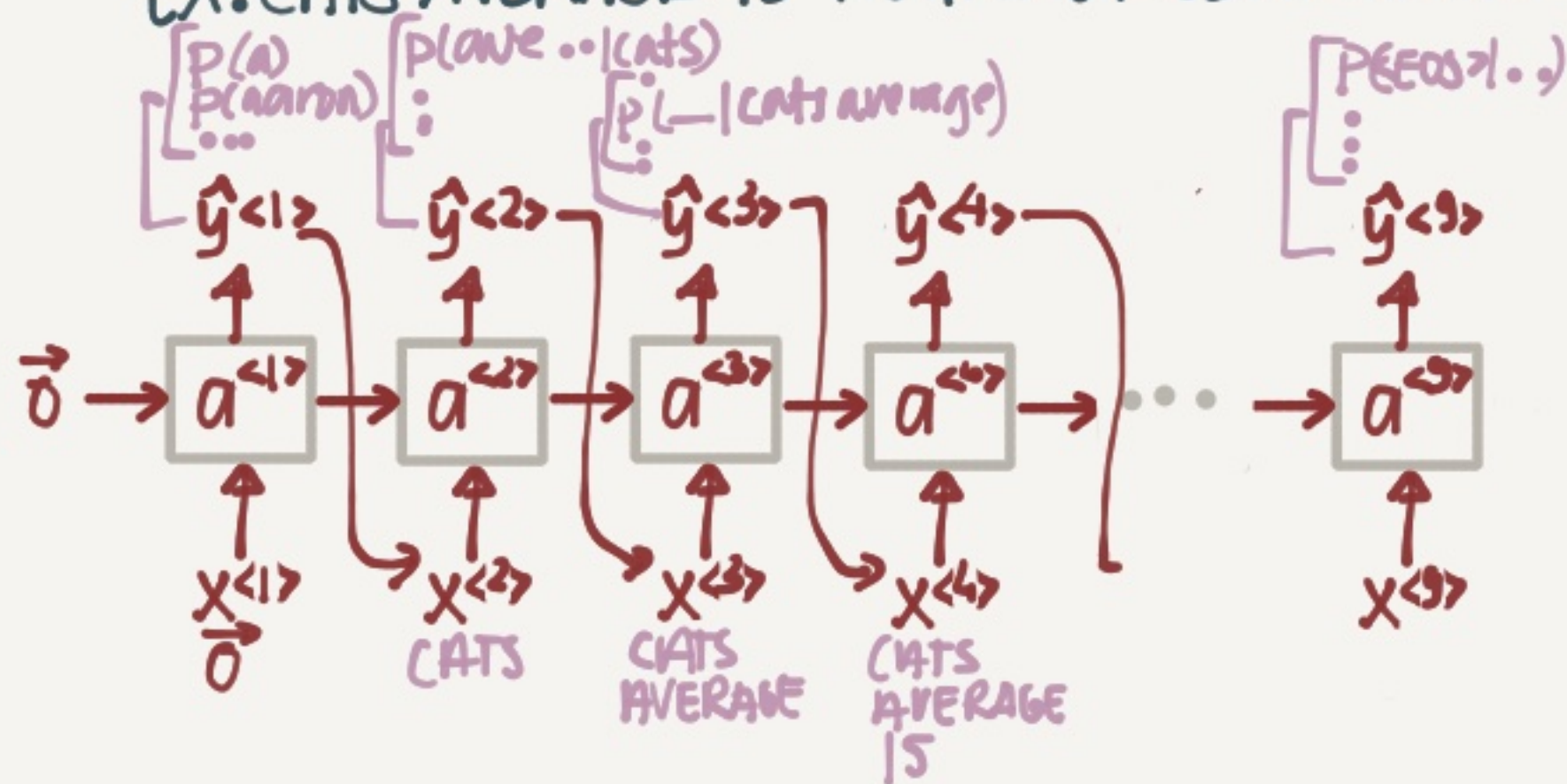
MORE ON RNNs

LANGUAGE MODELLING

HOW DO YOU KNOW IF SOMEONE SAID THE APPLE AND PEAR SALAD OR THE APPLE AND APPLE 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 ONE 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 CHARACTER LEVEL 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 BUT ~~BLIP~~ WAS FULL THE ~~CATS~~ WHO ALREADY ATE ... ~~WERE~~ FULL

NEED TO REMEMBER SING/PLURAL FOR A LONG TIME

SINCE LONG SENTENCE \Rightarrow 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



GRU

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

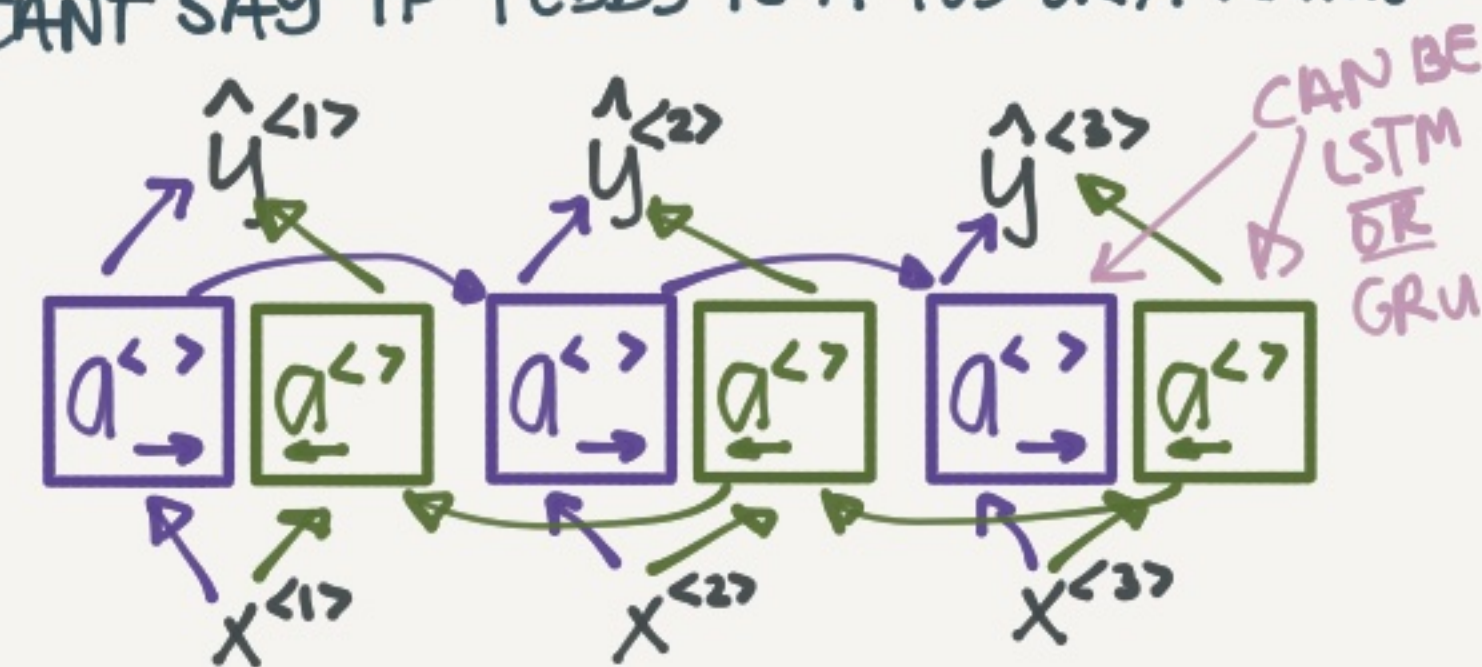
LONG SHORT TERM MEMORY (LSTM)

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

BI-DIRECTIONAL RNNs (BRNN)

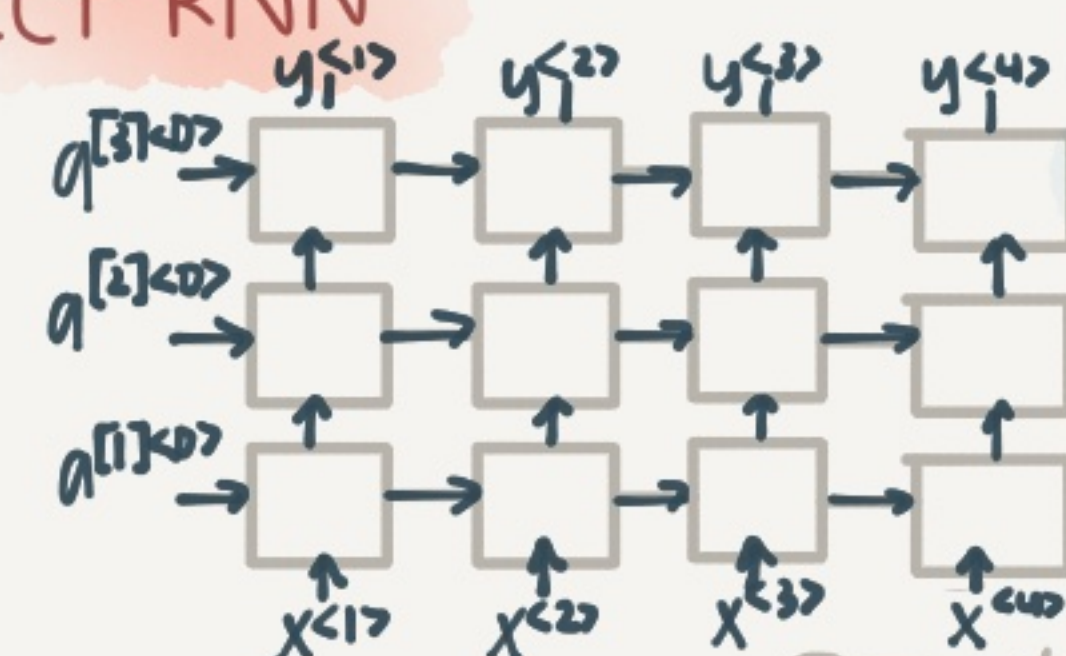
HE SAID, 'TEDDY BEARS ARE ON SALE' HE SAID, 'TEDDY ROOSEVELT WAS A GREAT PRESIDENT'

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 REC

DEEP RNN



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

@Tessferandez

SEQUENCE MODELS COURSE

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 _____

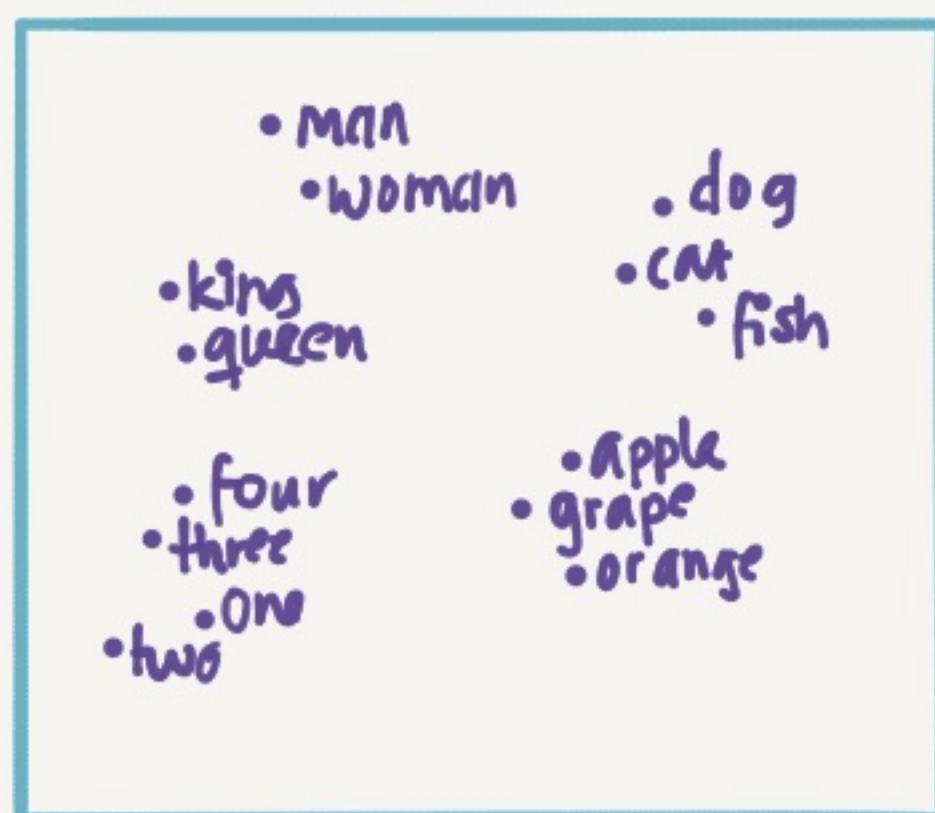
SOLUTION: CREATE A MATRIX OF FEATURES TO DESCRIBE THE WORDS

WORD EMBEDDINGS

	MAN	WOMAN	KING	QUEEN	APPLE	ORANGE
GENDER	1	0	0	0	0	0
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}

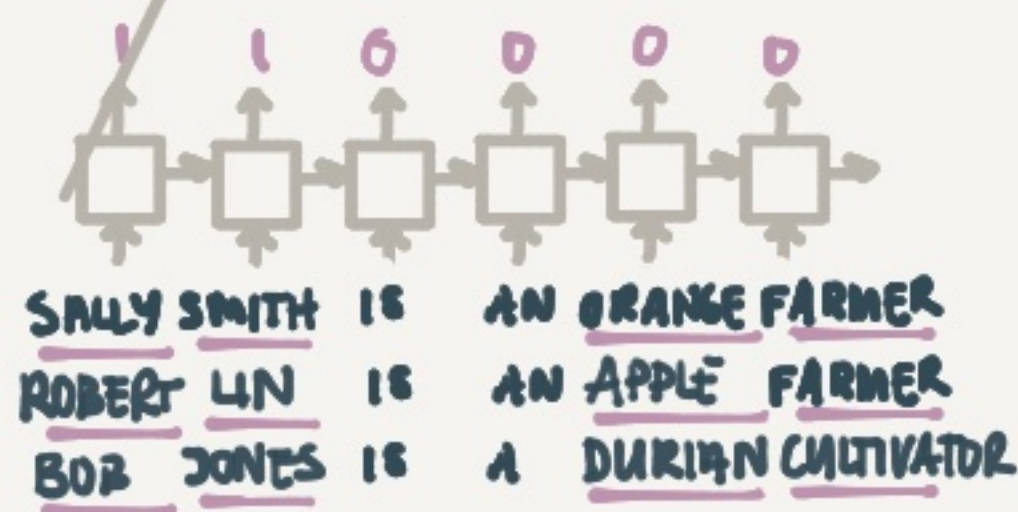
IN REALITY • THE FEATURES ARE LEARNED & NOT AS STRAIGHTFWD AS GENDER/AGE



t-SNE VISUAL REPRESENT OF 3000 WORD EMBEDDINGS

USING WORD EMBEDDINGS

EX. NAME/ENTITY RECOGN



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

- APPLE \sim ORANGE \Rightarrow 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 ?

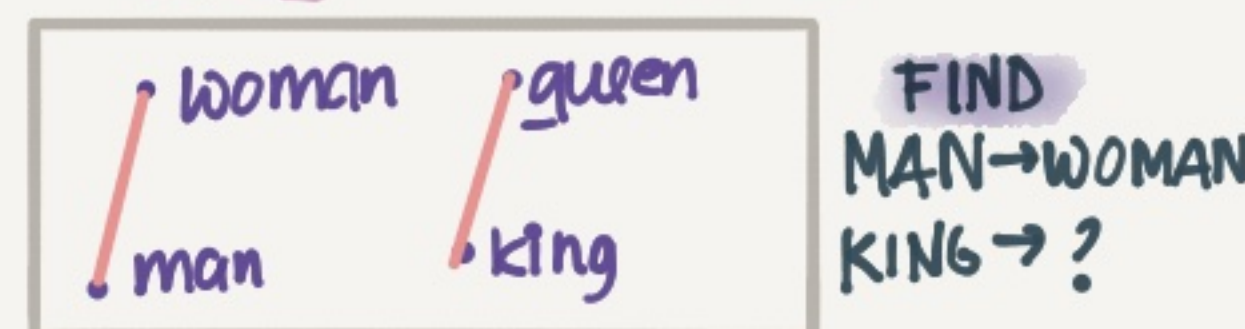
$E =$ EMBEDDING MATRIX

	MAN	WOMAN	KING	QUEEN
GENDER	1	0	0	0
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(w): ARG. MAX SIM($e_w, e_{king} - e_{man} + e_{woman}$)

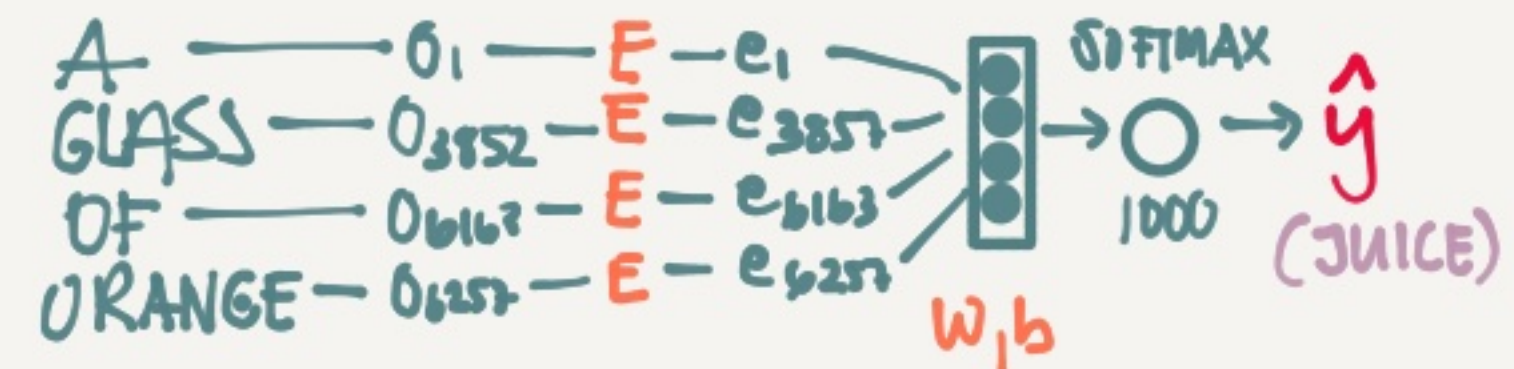
$SIM(u, v) = \frac{u \cdot v}{\|u\|_2 \|v\|_2}$ COSINE SIMILARITY

LEARNING WORD EMBEDDINGS

HOW DO WE LEARN THE EMBEDDING MATRIX E?

IDEA 1: USING A NEURAL LANG MODEL

I WANT A GLASS OF ORANGE \hat{y}



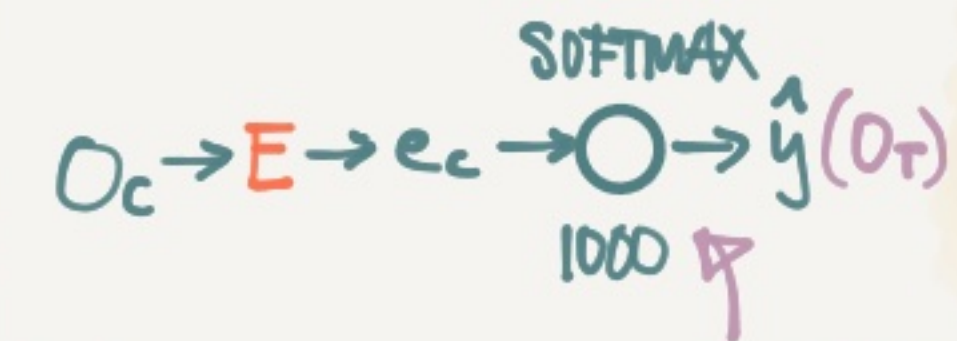
WE CAN USE DIFFERENT CONTEXTS THAN THE LAST 4 WORDS

- LAST 4 WORDS
- 4 WORDS LEFT+RIGHT
- LAST 1 WORD
- NEARBY 1 WORD \leftarrow 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
...	...



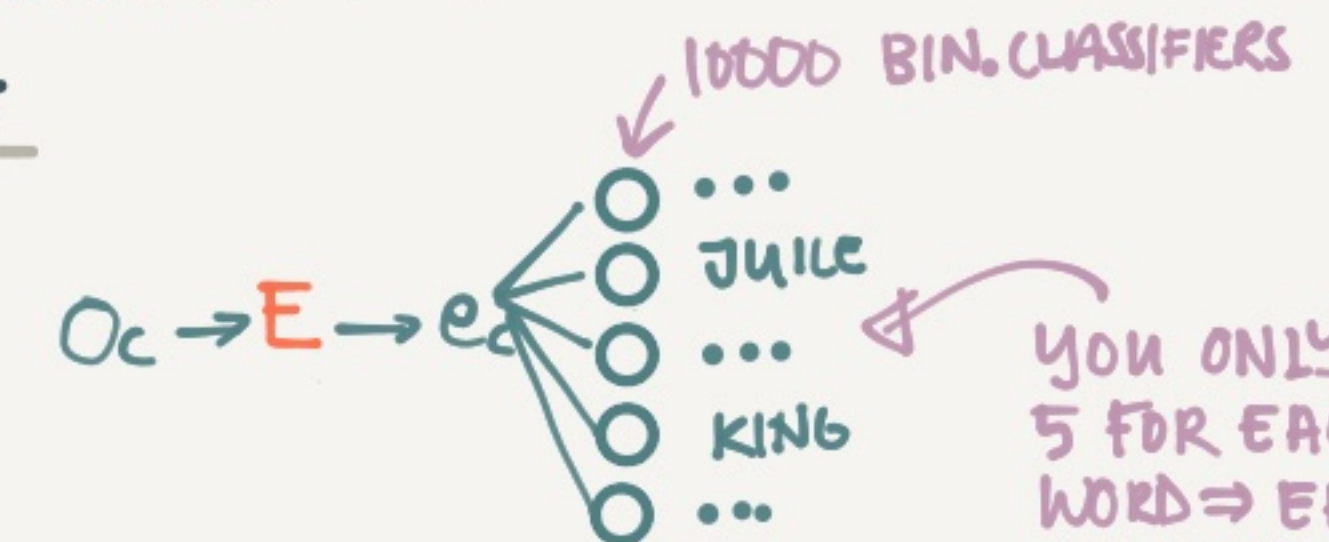
NOTE: WHILE THIS SIMPLE NN PREDICTS O_t OUR REAL GOAL IS TO LEARN E

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
FRANGE	BOOK	0
ORANGE	THE	0
ORANGE	OF	0



YOU ONLY TRAIN 5 FOR EACH CONTEXT WORD \Rightarrow EFFICIENT TO TRAIN

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

WORD EMBEDDINGS CONTINUED...

GLOVE WORD VECTORS

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

$$\text{MINIMIZE } \sum_{i=1}^{10k} \sum_{j=1}^{10k} f(x_{ij}) (\theta_i^T e_j + b_i + b_j - \log x_{ij})^2$$

IF NO CONTEXT (BUT HELPS WEIGHING VERY FREQ WORDS (THE, OF...) & VERY INFREQ (PURJAN))

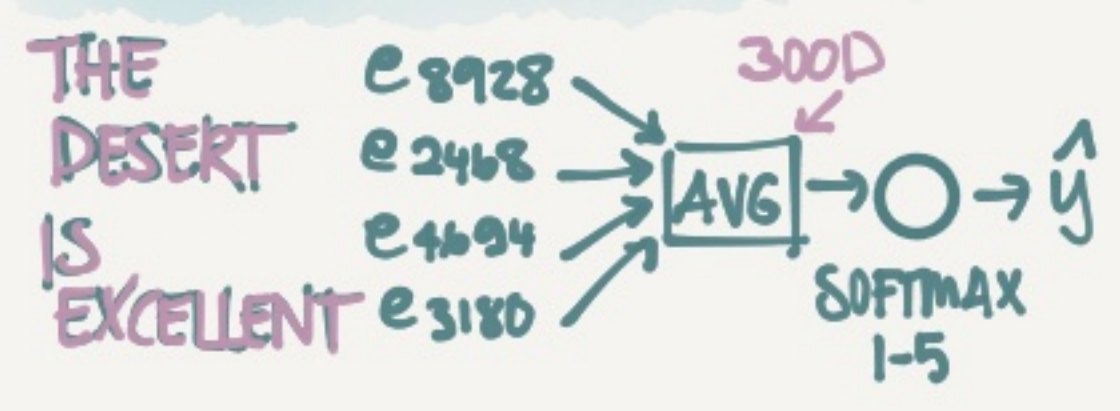
EVERYTHING LED UP TO THIS VERY SIMPLE ALGORITHM

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

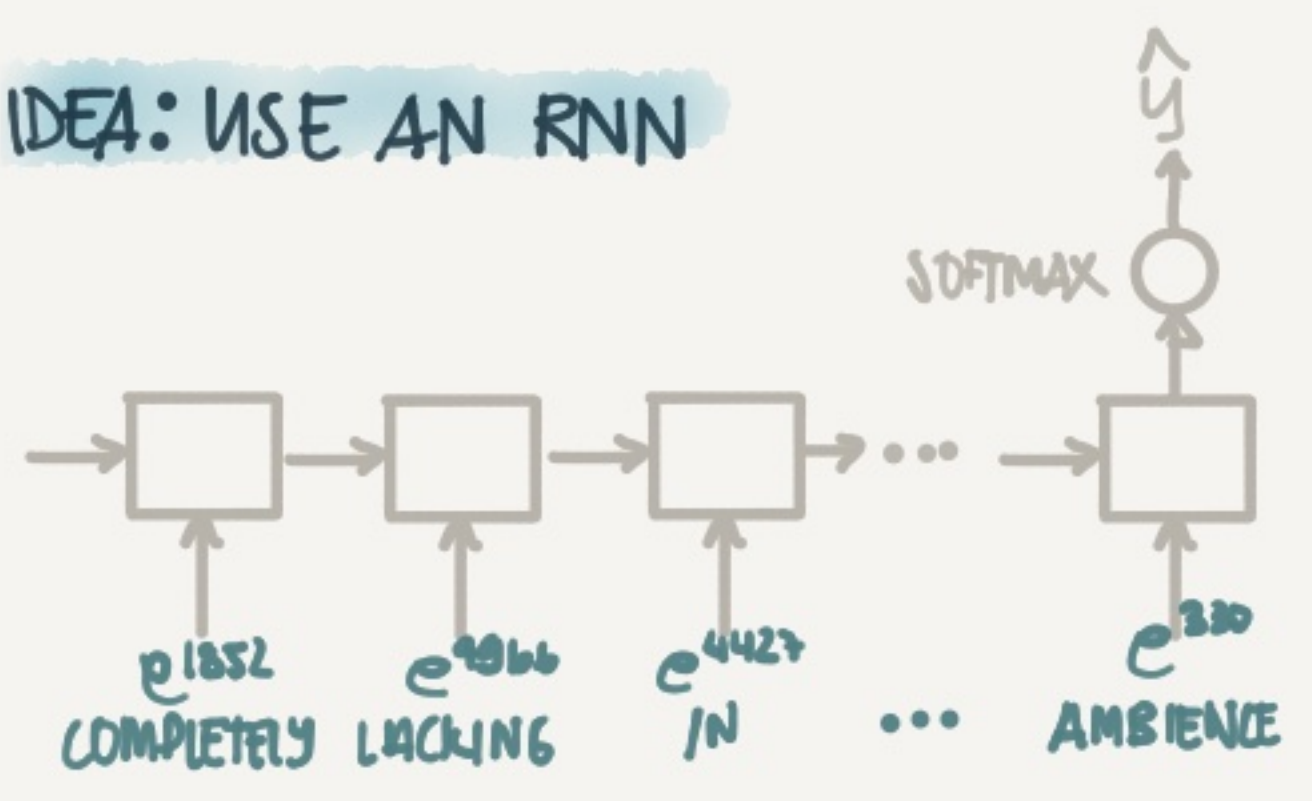
IDEA: SIMPLE CLASSIFICATION



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



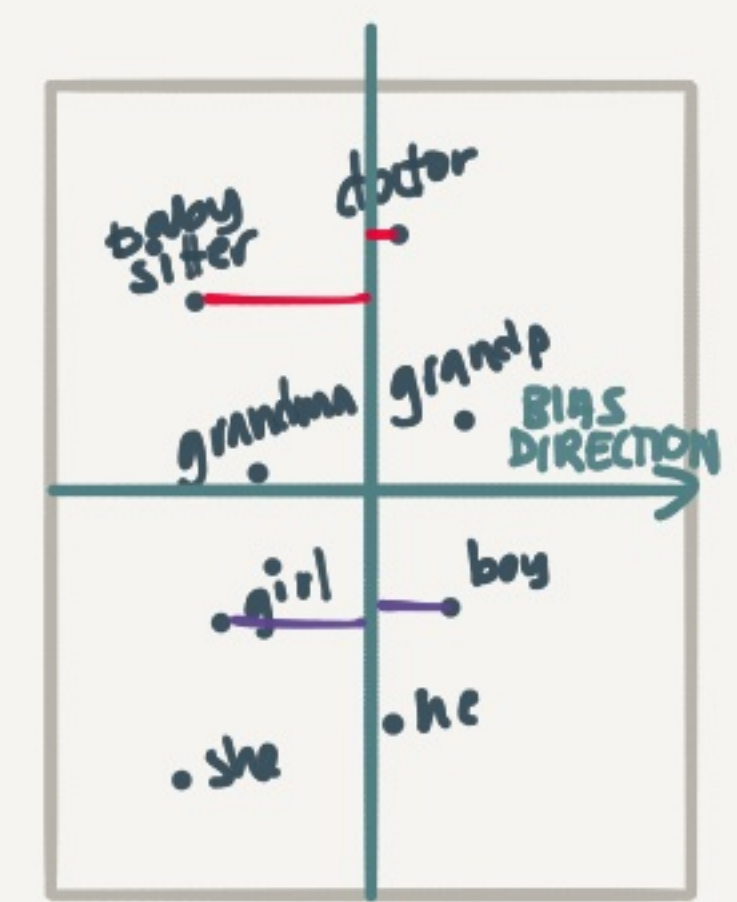
THIS CAN NOW TAKE INTO ACCOUNT THAT COMPLETELY LACKING NEGATES THE WORD GOOD

ELIMINATING BIAS IN WORD EMBEDDINGS

MAN IS TO COMPUTER PROGRAMMER AS WOMAN IS TO HOME MAKER

SOMETIMES THE TEXT CONTAINS & 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



- IDENTIFY BIAS DIRECTION
 $\{ e_{he} \rightarrow e_{she}$
 $\{ e_{male} \rightarrow e_{female}$
- NEUTRALIZE
 FOR EVERY WORD THAT IS NOT DEFINITIONAL (GIRL, BOY, HE, SHE...) PROJECT TO GET RID OF BIAS
- EQUALIZE PAIRS
 THE ONLY DIFF BETWEEN EX GIRL/BOY SHOULD BE GENDER

HOW DO YOU KNOW WHICH WORDS TO NEUTRALIZE?

DOCTOR, BEARD, SEWING MACHINE?

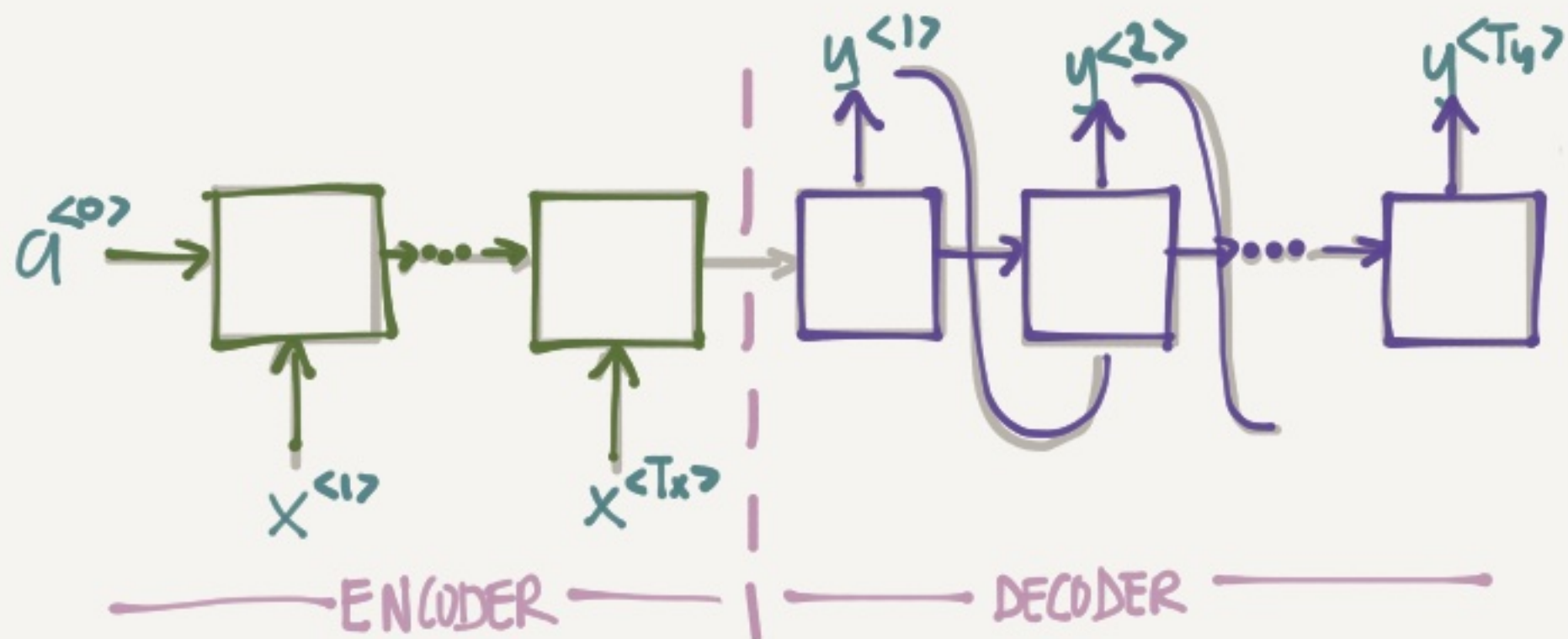
A: BY TRAINING A CLASSIFIER TO FIND OUT IF A WORD IS DEFINITIONAL

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

SEQUENCE 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?

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

ENGLISH FRENCH

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

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

IDEA: USE GREEDY SEARCH

- PICK THE WORD WITH THE BEST PROBABILITY
- REPEAT UNTIL $\langle \text{EOD} \rangle$

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

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



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

(IN SEPTEMBER, JANE IS, JANE VISITS)

- REPEAT TIL DONE

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

OVERFLOWS

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

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

INSTEAD WE CAN OPTIMIZE FOR THIS

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

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^*
ALSO: 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 $P_{y^*} = P(y^* | x) \leq P_{\hat{y}} = P(\hat{y} | x)$

IF $P_{y^*} > P_{\hat{y}}$:

BEAM PICKED THE WRONG ONE TRY A HIGHER B

ELSE:

THE RNN PICKED THE WRONG PROBS - SO FOCUS ON THE RNN

SEQUENCE TO SEQUENCE

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

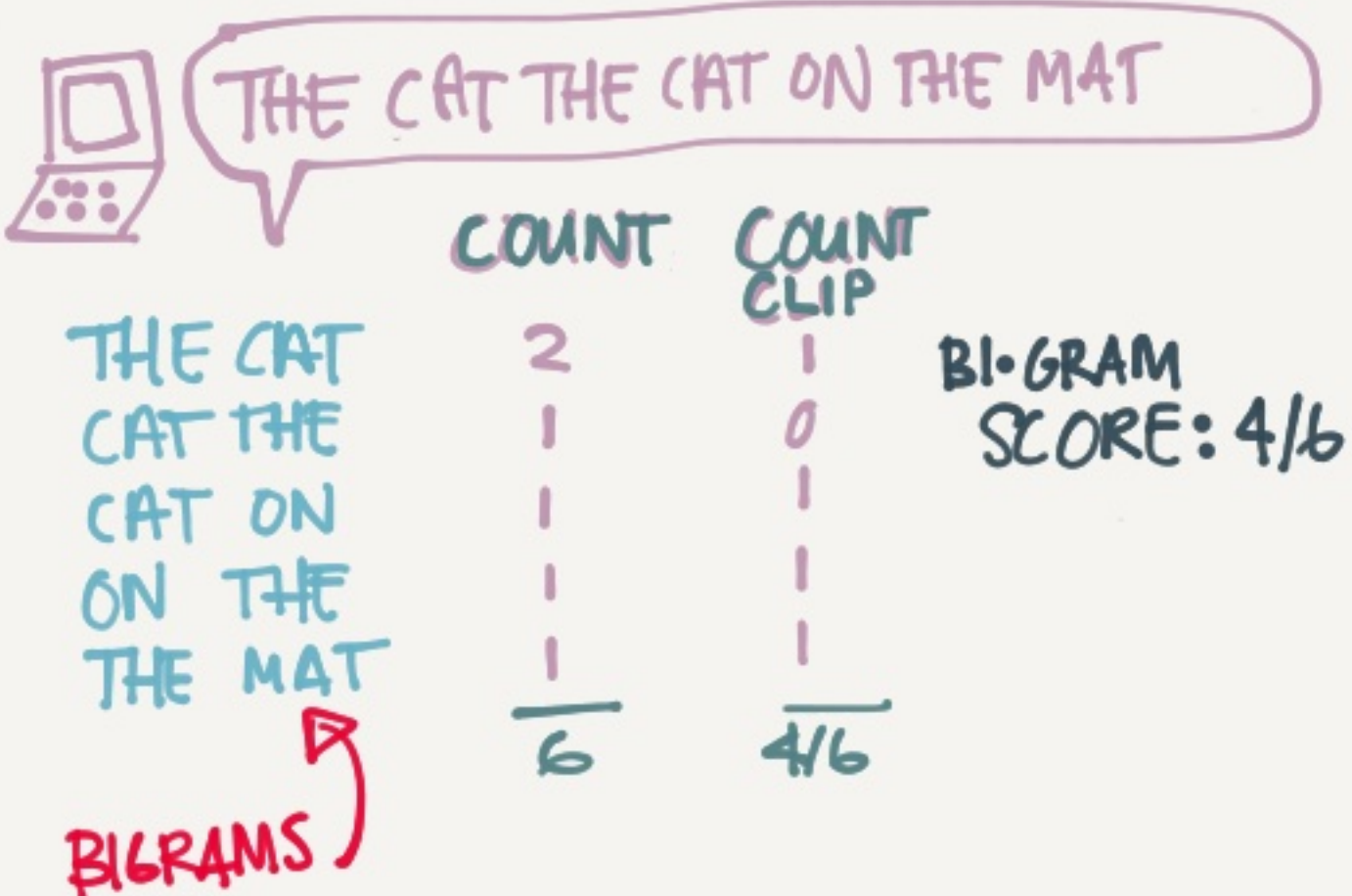
HOW DO YOU EVALUATE THE MACHINE TRANSLATION WHEN MULTIPLE ARE RIGHT?

BLEU SCORE

IDEA: CHECK IF THE WORDS ^{MR} 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}



COMBINED BLEU SCORE

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

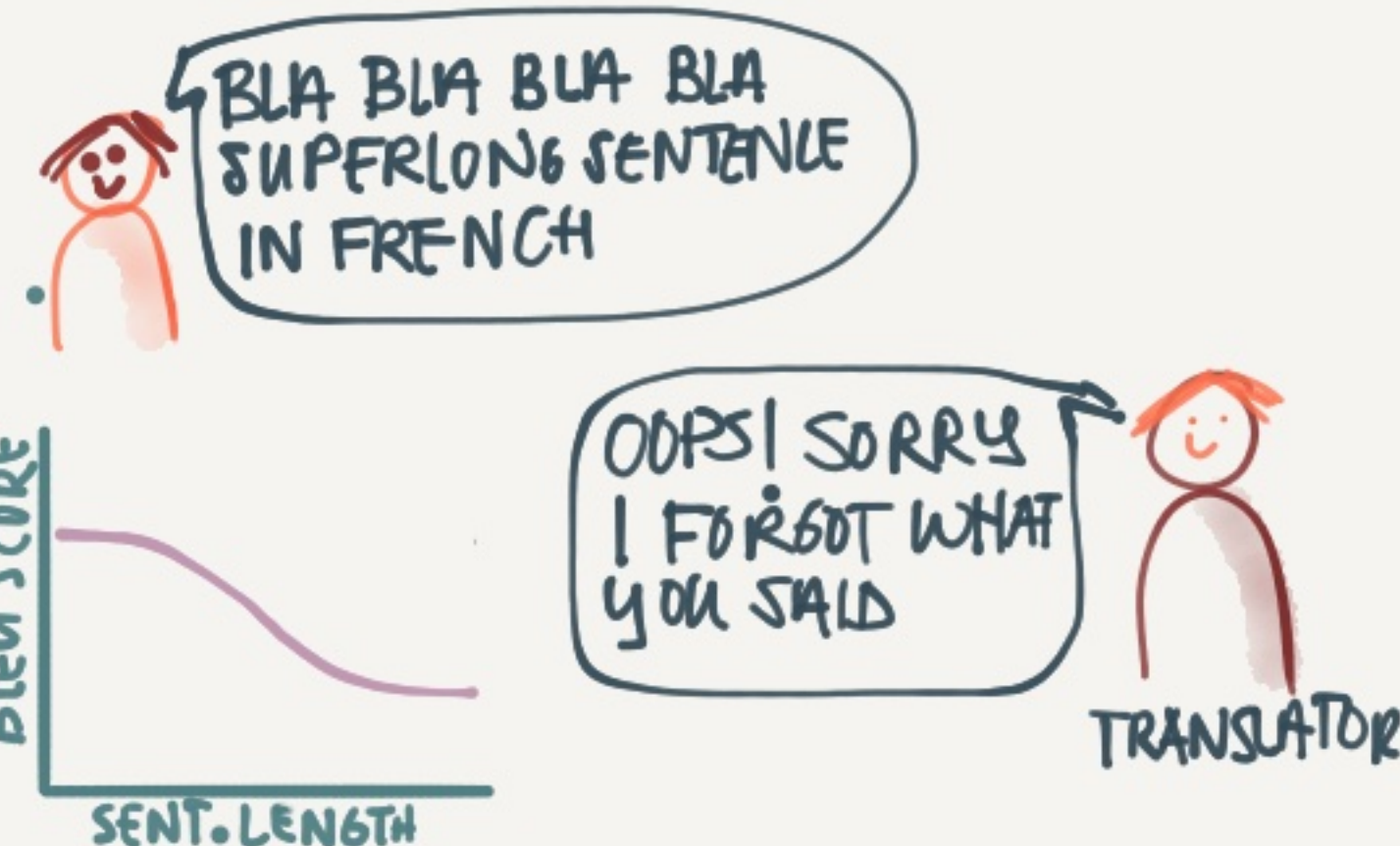
P_1 = SCORE SINGLE WORD
 P_2 = SCORE BIGRAMS
 ...

BP = BREVITY PENALTY

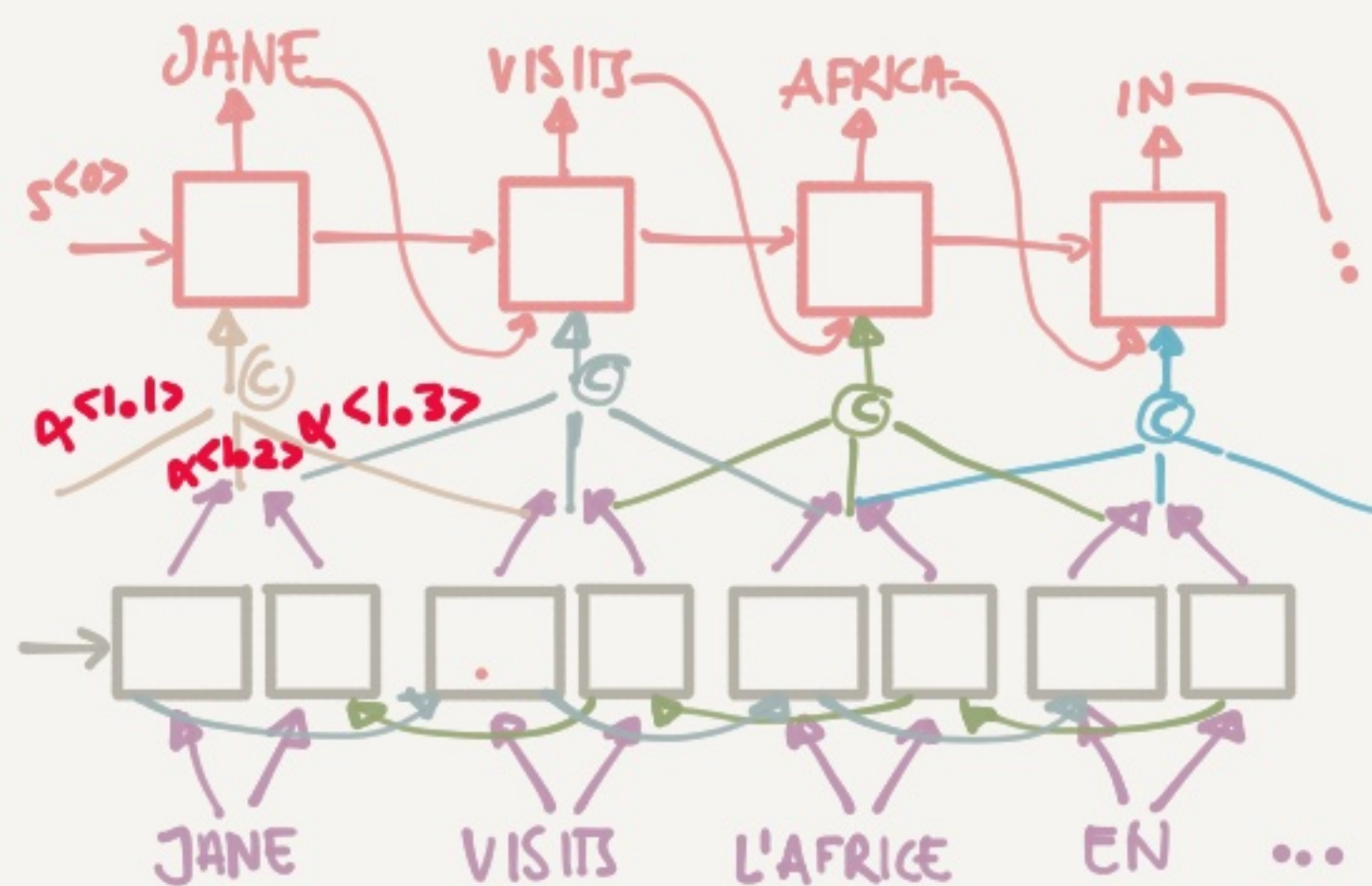
PENALIZES SENTENCES SHORTER THAN THE TARGET

A USEFUL SINGLE NUMBER EVAL METRIC

ATTENTION MODEL



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

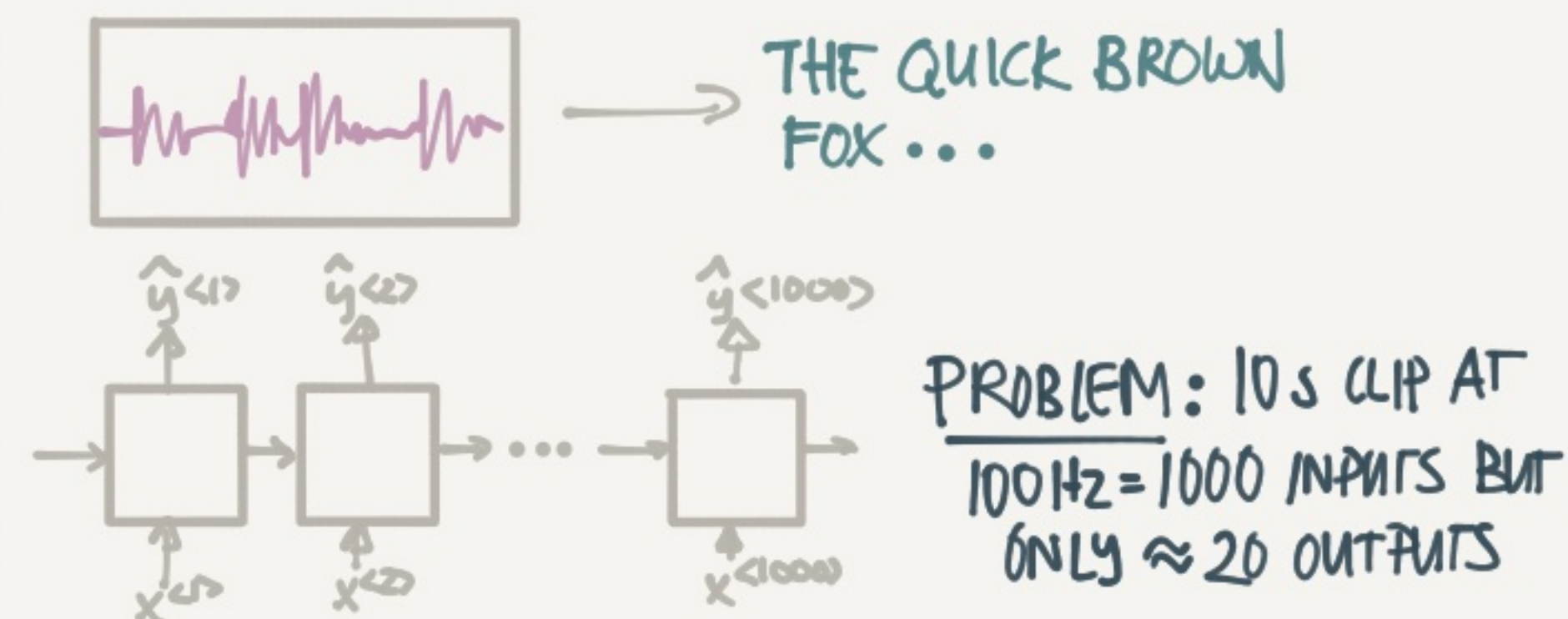


$\alpha^{<t, t'>} =$ HOW MUCH ATTENTION $y^{<t>}$ SHOULD PAY TO $a^{<t'>}$

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

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

