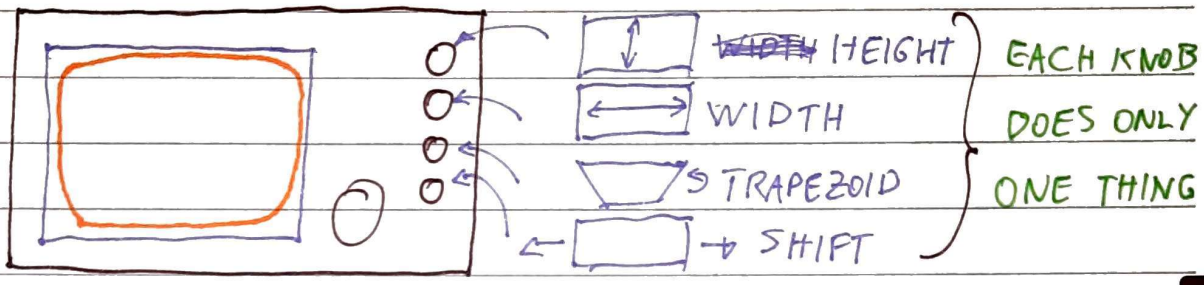
DEEP LEARNING SPECIALIZATION
COURSE 3/5

STRUCTURING MACHINE LEARNING PROJECTS

→ ORTHOGONALIZATION

* EXAMPLE: TV TUNING



* EACH HYPERPARAMETER "TUNES" ONLY ONE THING

* CHAIN OF ASSUMPTIONS IN M.L.

- FIT TRAINING SET WELL ON COST FUNCTION
  → TUNE: BIGGER NETWORK, OPTIMIZATION ALGORITHM...
- FIT DEV SET WELL ON COST FUNCTION
  → TUNE: REGULARIZATION, BIGGER TRAIN SET, ...
- FIT TEST SET WELL ON COST FUNCTION
  → TUNE: BIGGER DEV SET, ...
- PERFORMS WELL IN REAL WORLD
  → TUNE: CHANGE DEV SET, CHANGE COST FUNCTION...

* EARLY STOPPING IS NO GOOD FOR ORTHOGONALIZATION

→ SINGLE NUMBER EVALUATION METRIC

* EXAMPLE: TWO PERFORMANCE METRICS

CLASSIFIER	PRECISION	RECALL	F1 SCORE
A	95%	90%	92,4% ← F1 BETTER
B	98%	85%	91,0%

- PRECISION: OF EXAMPLES RECOGNIZED AS A CAT, WHAT PERCENTAGE ACTUALLY ARE CATS?
- RECALL: WHAT % OF ACTUAL CATS ARE CORRECTLY RECOGNIZED?
- "PRECISION-RECALL" TRADE-OFF
- IT'S HARDER TO PICK A CLASSIFIER BASED ON TWO METRICS: A DOES BETTER AT RECALL, B AT PRECISION
- COMBINE PRECISION AND RECALL: "F1 SCORE"
- F1 SCORE: "AVERAGE" OF PRECISION AND RECALL

$$F_1 = \frac{2}{\frac{1}{\text{PRECISION}} + \frac{1}{\text{RECALL}}}$$

← HARMONIC MEAN OF PRECISION AND RECALL

WELL-DEFINED DEV SET + SINGLE NUMBER EVALUATION METRIC

SPEED UP TRAINING

- IF YOU HAVE A LOT OF ERROR DATA, COMPUTE AVERAGE

→ SATISFICING AND OPTIMIZING METRICS

* ANOTHER CLASSIFICATION EXAMPLE

	OPTIMIZING	SATISFICING
CLASSIFIER	ACCURACY	RUNNING TIME
A	90%	80 ms
B	92%	95 ms
C	95%	1500 ms

EXAMPLE: ACCURACY AS OPTIMIZING METRIC, RUNNING TIME AS SATISFICING METRIC

CONDITIONS: MAXIMUM ACCURACY SUBJECT TO RUNNING TIME $\leq$ 100 ms.

BETTER CHOICE FOR CONDITIONS: B

IF YOU HAVE N METRICS: 1 - OPTIMIZING
N-1 - SATISFICING

→ DEV/TEST SETS SELECTION

* EXAMPLE: CAT CLASSIFICATION IN DIFFERENT REGIONS

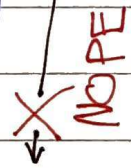
RANDOM SHUFFLE!

- US
- UK
- OTHER EUROPE
- SOUTH AMERICA

DEV

- INDIA
- CHINA
- OTHER ASIA
- AUSTRALIA

TEST



- DON'T GET DATA FROM DIFFERENT DISTRIBUTIONS FOR DEV & TEST SETS
- GET DATA THAT REFLECTS REAL DATA YOU EXPECT
- MIX ALL DATA AND MAKE A RANDOM SHUFFLE INTO DEV/TEST SETS

→ WHEN TO CHANGE DEV/TEST SETS AND METRICS?

✳ CAT CLASSIFIER APP

- METRIC: CLASSIFICATION ERROR

  ALGORITHM A: 3% ERROR → SHOWS PORN PICS

  ALGORITHM B: 5% ERROR → DOESN'T SHOW PORN

- METRIC + DEV: PREFER A, YOUR USERS: PREFER B
- MODIFYING ERROR CALCULATION

$$\text{ERROR: } \cancel{\frac{1}{m_{dev}}} \frac{1}{\sum w^{(i)}} \sum_{i=1}^{m_{dev}} I\left(y_{PRED}^{(i)} \neq y^{(i)}\right) \cdot w^{(i)}$$

$$w^{(i)} = \begin{cases} 1, & \text{IF } x^{(i)} \text{ ISN'T PORN} \\ 10, & \text{IF } x^{(i)} \text{ IS PORN} \end{cases}$$

✳ ORTHOGONALIZATION FOR CAT PICTURES: ANTI-PORN

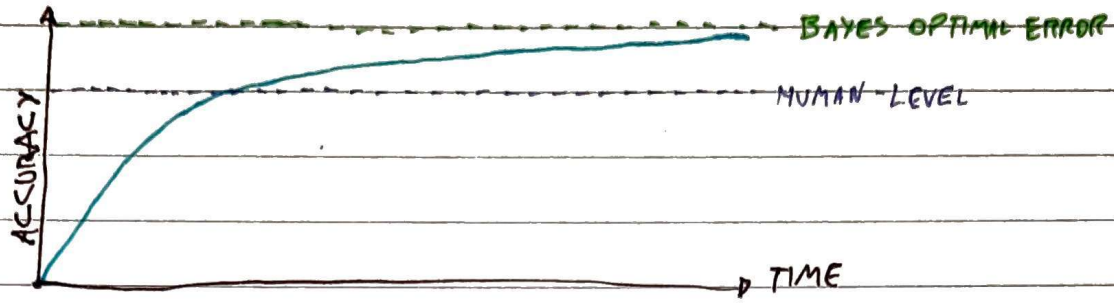
- DEFINE A METRIC TO EVALUATE CLASSIFIERS (PLACE TARGET)
- WORRY SEPARATELY ABOUT HOW TO DO WELL ON THIS METRIC (AIM/SHOOT AT TARGET)

$$J = \frac{1}{\sum w^{(i)}} \sum_{i=1}^{m} w^{(i)} \cdot L\left(\hat{y}^{(i)}, y^{(i)}\right)$$

✳ ANOTHER EXAMPLE: RESOLUTION OF IMAGES DIFFERING ON DEV AND TEST SETS

- IF DOING WELL ON YOUR METRIC + DEV/TEST SET DOESN'T CORRESPOND TO DOING WELL ON YOUR APPLICATION, CHANGE YOUR METRIC AND/OR DEV/TEST SETS.

→ WHY HUMAN-LEVEL PERFORMANCE



BAYES OPTIMAL ERROR: BEST POSSIBLE ERROR

* WHY COMPARE TO HUMAN-LEVEL PERFORMANCE?

- HUMANS ARE QUITE GOOD AT LOT OF TASKS. SO LONG AS ML IS WORSE THAN HUMANS, YOU CAN:
  - → GET LABELED DATA FROM HUMANS (X, Y)
  - → GAIN INSIGHT FROM MANUAL ERROR ANALYSIS: WHY DID A PERSON GET THIS RIGHT?
  - → BETTER ANALYSIS OF BIAS/VARIANCE

→ AVOIDABLE BIAS

	AVOIDABLE BIAS: 7%	AVOIDABLE BIAS: 0,5%
HUMAN ERROR:	1% ↕ HUGE GAP	7,5%
TRAINING ERROR:	8%	8% ↕ VARIANCE
DEV ERROR:	10%	10%
	FOCUS ON BIAS	FOCUS ON VARIANCE

- HUMAN-LEVEL ERROR AS A PROXY FOR BAYES ERROR

- AVOIDABLE BIAS: DIFFERENCE BETWEEN BAYES ERROR AND TRAINING ERROR

→ UNDERSTANDING HUMAN-LEVEL PERFORMANCE

* HUMAN-LEVEL AS A PROXY FOR BAYES ERROR: MEDICAL IMAGE CLASSIFICATION EXAMPLE

SUPPOSE:

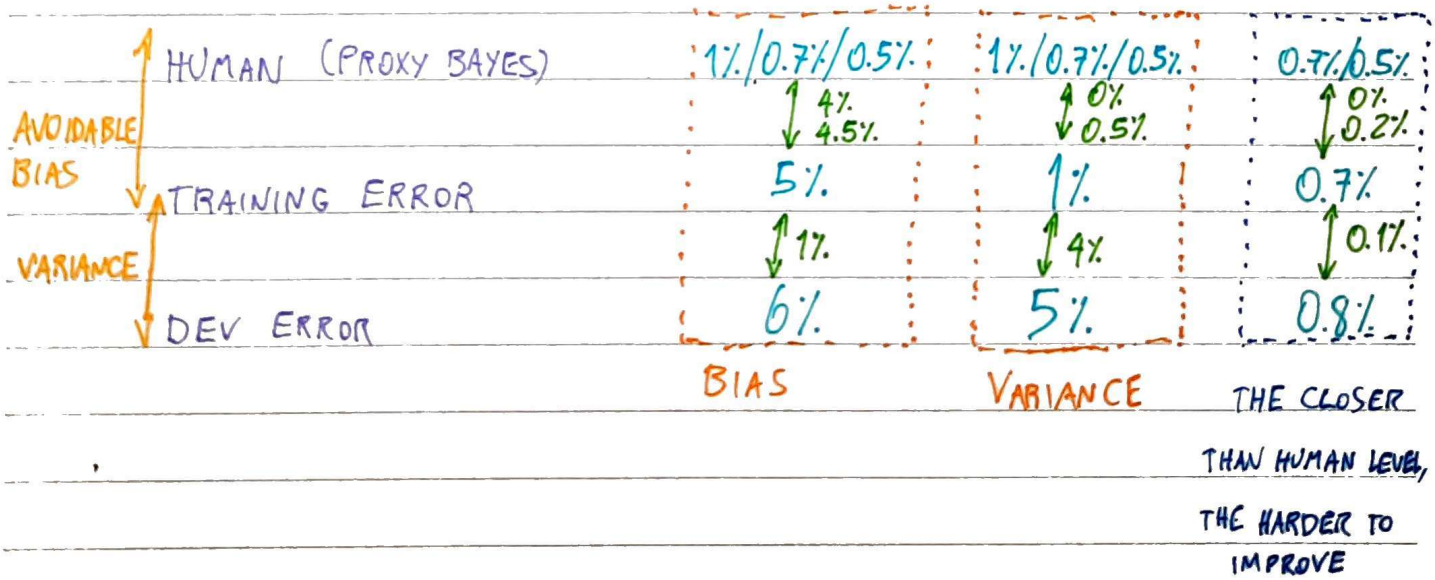
- TYPICAL HUMAN - 3% ERROR
- TYPICAL DOCTOR - 1% ERROR
- EXPERIENCED DOCTOR - 0.7% ERROR
- TEAM OF EXPERIENCED DOCTORS - 0.5% ERROR

WHICH ~~SHOULD~~ WE PICK FOR "HUMAN-LEVEL" ERROR?

TEAM OF EXPERIENCED DOCTORS. (BAYES-ERROR $\leq$ 0.5%)

FOR RESEARCH PAPER OR DEPLOY SYSTEM, TYPICAL DOCTOR ERROR COULD BE USED.

* ERROR ANALYSIS EXAMPLE



→ SURPASSING HUMAN-LEVEL PERFORMANCE

* EXAMPLE

TEAM OF HUMANS	0,5%		0,5%	- WHAT IS THE AVOIDABLE BIAS?
ONE HUMAN	~~1%~~	0,1%	~~1%~~	
TRAINING ERROR	0,6%		0,3%	
DEV ERROR	0,8%	0,2%	0,4%	

- TRAINING ERROR < TEAM ERROR: NOT ENOUGH INFORMATION ABOUT WHAT'S THE REAL BAYES ERROR
- PROGRESS CAN STILL BE MADE, BUT IT'S LESS CLEAR

* PROBLEMS WHERE ML SIGNIFICANTLY SURRASSES HUMAN-LEVEL PERFORMANCE

- ONLINE ADVERTISING
- PRODUCT RECOMENDATIONS
- LOGISTICS (PREDICTING TRANSIT TIME)
- LOAN APPROVALS

↑

STRUCTURED DATA, NON-NATURAL PERCEPTION, LOTS OF DATA

- HARDER TO SURPASS HUMAN-LEVEL PERFORMANCE ON NATURAL PERCEPTION TASKS
- IN SOME SPEECH RECOGNITION, SOME IMAGE RECOGNITION AND MEDICAL RECOGNITION, WE STILL CAN SURPASS HUMAN-LEVEL PERFORMANCE, BUT IT'S HARDER

→ IMPROVING MODEL PERFORMANCE

* THE TWO FUNDAMENTAL ASSUMPTIONS OF SUPERVISED LEARNING

- YOU CAN FIT THE TRAINING SET PRETTY WELL
  → ACCEPTABLE AVOIDABLE BIAS
- THE TRAINING SET PERFORMANCE GENERALIZES PRETTY WELL TO THE DEV/TEST SETS
  → VARIANCE

* REDUCING AVOIDABLE BIAS AND VARIANCE

AVOIDABLE BIAS:

- TRAIN BIGGER NETWORK/MODEL
- TRAIN LONGER/BETTER OPTIMIZATION ALGORITHMS
- NN ARCHITECTURE/HYPERPARAMETERS SEARCH

VARIANCE:

- MORE DATA
- REGULARIZATION
- NN ARCHITECTURE/HYPERPARAMETERS SEARCH

→ CARRYING OUT ERROR ANALYSIS

* EXAMPLE CAT CLASSIFIER

- 90% ACCURACY, 10% ERROR

SHOULD YOU TRY TO MAKE YOUR CAT CLASSIFIER DO BETTER ON DOGS?

ERROR ANALYSIS:

→ GET ~100 MISLABELED DEV SET EXAMPLES.

→ COUNT UP HOW MANY ARE DOGS

IF 5/100: 5% ⟶ 9.5% ERROR. DON'T TRY.

IF 50/100: 50% ⟶ 5% ERROR. PERHAPS!

* EVALUATE MULTIPLE IDEAS IN PARALLEL

- FIX PICTURES OF DOGS RECOGNIZED AS CATS
- FIX GREAT CATS (LIONS, PANTHERS, ETC...) BEING MISRECOGNIZED
- IMPROVE PERFORMANCE ON BLURRY IMAGES
- IMPROVE PERFORMANCES ON IMAGES WITH INSTAGRAM FILTER

IMAGE	DOG	GREAT CATS	BLURRY	FILTER	COMMENTS
1	✓			✓	PITBULL
2			✓		
3		✓	✓		RAINY DAY AT ZOO
⋮	⋮	⋮	⋮	⋮	
% TOTAL	8%	43%	61%	12%	

FOCUS SHOULD BE ON THIS PROBLEMS

→ CLEANING UP INCORRECTLY LABELED DATA

* DEEP LEARNING ALGORITHMS ARE QUITE ROBUST TO RANDOM ERRORS IN TRAINING SET, BUT NOT ROBUST TO SYSTEMATIC ERRORS
* ERROR ANALYSIS
  - IN ORDER TO DETERMINE IF ITS WORTH TO CHECK MISLABELED DATA ON DEV/TEST SETS, WE CHECK THREE METRICS:

OVERALL SET ERROR	10%	2%
ERRORS DUE INCORRECT LABELS	0,6%	0,6%
ERRORS DUE TO OTHER CAUSES	9,4%	1,4%
	↑ IT'S NOT WORTH	↑ GREATER IMPACT. WORTH IT.

* CORRECTING INCORRECT DEV/TEST SET EXAMPLES
  - APPLY SAME PROCESS TO YOUR DEV AND TEST SETS TO MAKE SURE THEY CONTINUE TO COME FROM THE SAME DISTRIBUTION
  - CONSIDER EXAMINING EXAMPLES YOUR ALGORITHM GOT RIGHT AS WELL AS ONE IT GOT WRONG
  - TRAIN AND DEV/TEST DATA MAY NOW COME FROM SLIGHTLY DIFFERENT DISTRIBUTIONS

→ BUILD YOUR FIRST SYSTEM QUICKLY, THEN ITERATE

* SETUP DEV/TEST SET AND METRIC
* BUILD INITIAL SYSTEM QUICKLY
* USE BIAS/VARIANCE ANALYSIS AND ERROR ANALYSIS TO PRIORITIZE NEXT STEPS

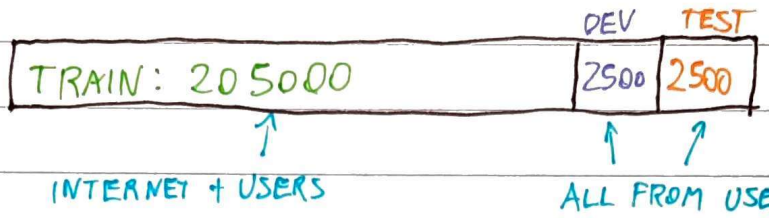
TRAINING AND TEST SETS ON DIFFERENT DISTRIBUTIONS

*EXAMPLE: CAT RECOGNITION APP

- TRAINING SET: HIGH-RES IMAGES FROM WEB (200K EXAMPLES)
- TEST SET: LOW-RES BLURRY IMAGES FROM USERS (10K EXAMPLES)

OPTION 1:

OPTION 2:

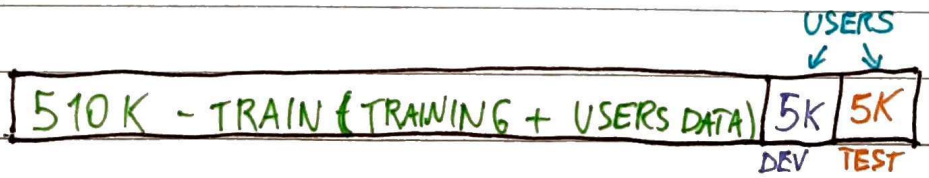


- OPTION 1 NOT RECOMENDED: IN RANDOM SHUFFLE, MOST OF IMAGES OF DEV/TEST SET WILL BE INTERNET IMAGES: THIS DOESN'T REPRESENT REAL TARGET
- OPTION 2 RECOMENDED, BUT MOST OF TRAIN SET WILL NOT BE FROM SAME DISTRIBUTION OF DEV AND TEST SETS

*EXAMPLE: CAR SPEECH RECOGNITION

- 500K EXAMPLES FOR TRAINING, 20K COLLECTED FROM APP

OPTIONS:



→ BIAS AND VARIANCE WITH MISMATCHED DATA DISTRIBUTIONS

* CAT CLASSIFIER EXAMPLE

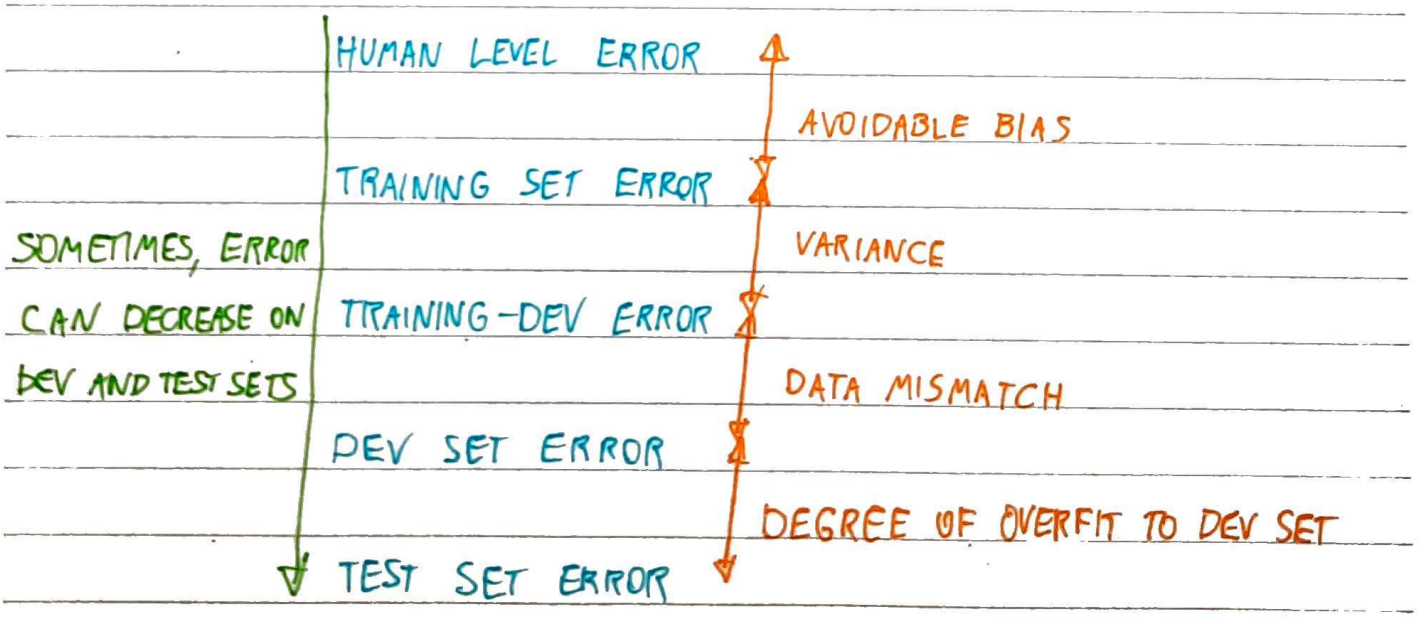
- ASSUME HUMANS GET ≈ 0% ERROR.

TRAINING ERROR	1%	} HOW DO WE KNOW IF IT'S HIGH
DEV ERROR	10%	} VARIANCE OR DATA MISMATCH?

CREATE TRAINING-DEV SET: SAME DISTRIBUTION AS TRAINING SET, BUT NOT USED FOR TRAINING

TRAINING ERROR	1%	1%	10%	10%
TRAINING-DEV ERROR	9%	1,5%	11%	11%
DEV ERROR	10%	10%	12%	20%
DIAGNOSIS	VARIANCE	DATA MISMATCH	AVOIDABLE BIAS	AVOIDABLE BIAS + DATA MISMATCH

* CHECKING METRICS



→ ADDRESSING DATA MISMATCH

* CARRY OUT MANUAL ERROR ANALYSIS TO TRY TO UNDERSTAND DIFFERENCE BETWEEN TRAINING AND DEV/TEST SETS
* MAKE TRAINING DATA MORE SIMILAR; OR COLLECT MORE DATA SIMILAR TO DEV/TEST SETS

* EXAMPLE: SPEECH RECOGNITION IN A CAR
  - RECORDING FAR FROM MIC + ADD CAR NOISE
  - ARTIFICIAL DATA SYNTHESIS

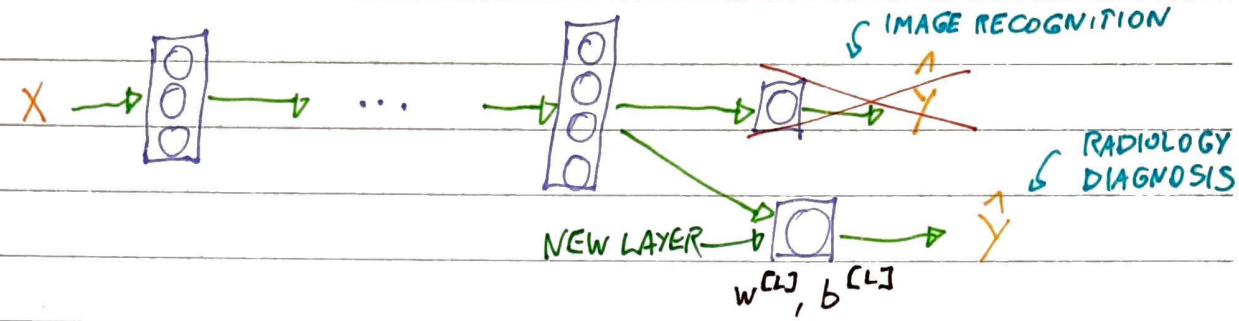
CAUTION:

1000 HOURS OF QUIET AUDIO + 1 HOUR OF CAR NOISE REPEATED ← SYSTEM CAN OVERFIT TO THIS 1 HOUR OF NOISE

- WHEN USING ARTIFICIAL DATA SYNTHESIS, BE CAREFUL TO NOT USE A REPEATED SMALL SUBSET OF DATA: IT CAN CAUSE AN OVERFIT

→ TRANSFER LEARNING

* USE PRE-TRAINED N.N. ON A DIFFERENT TASK, CHANGING THE OUTPUT
* EXAMPLE: IMAGE RECOGNITION TO RADIOLOGY DIAGNOSIS



- IF YOU HAVE A FEW RADIOLOGY DATA, YOU ONLY NEED TO TRAIN COEFICIENTS FOR NEW LAYER(S). IF THERE'S A LOT OF DATA: RETRAIN ENTIRE NETWORK.

IMAGE RECOGNITION —FINE TUNING→ RADIOLOGY DIAGNOSIS

* WHEN TRANSFER LEARNING MAKES SENSE?

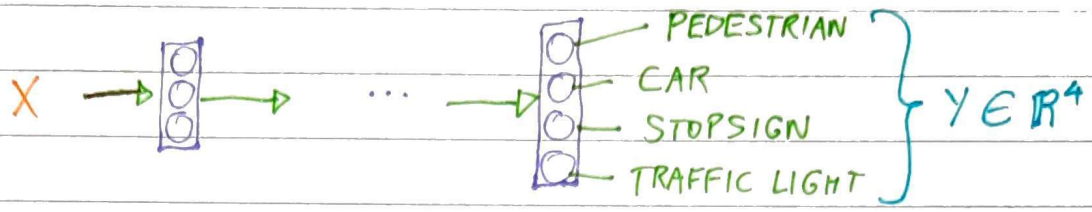
ASSUMING TRANSFER LEARNING FROM TASK A TO TASK B:

- TASK A AND B HAVE THE SAME INPUT X.
- YOU HAVE A LOT MORE DATA FROM TASK A THAN B.
- LOW LEVEL FEATURES FROM A COULD BE HELPFUL FOR LEARNING B.

-DMULTI-TASK LEARNING

* EXAMPLE: IMAGE RECOGNITION FOR AUTONOMOUS CAR

- DETECT PEDESTRIANS, CARS, STOP SIGNS, TRAFFIC LIGHTS...
- MULTIPLE OUTPUTS



COST FUNCTION: $\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{4} L(\hat{y}_j^{(i)}, y_j^{(i)})$ → USUAL LOGISTIC LOSS

- UNLIKE SOFTMAX REGRESSION, ONE IMAGE CAN HAVE MULTIPLE LABELS
- Y TRAINING LABELS CAN HAVE "DONT-CARES"

$$Y = \begin{bmatrix} 1 & 0 & 0 & ? \\ 0 & 1 & ? & 0 \\ ? & 0 & 1 & 1 \\ ? & 0 & 0 & 1 \end{bmatrix} \quad \ldots$$

SUM OF COST FUNCTION SHOULD IGNORE DON'T CARES

* WHEN MULTITASK LEARNING MAKES SENSE?

- TRAINING ON A SET OF TASKS THAT COULD BENEFIT FROM HAVING SHARED LOWER-LEVEL FEATURES
- USUALLY: AMOUNT OF DATA YOU HAVE FOR EACH TASK IS QUITE SIMILAR
- CAN TRAIN A BIG ENOUGH NEURAL NETWORK TO DO WELL ON ALL THE TASKS.

D-3-16

→ END-TO-END DEEP LEARNING

* EXAMPLE: SPEECH RECOGNIZING

OLD PIPELINE:

AUDIO —MFCC→ FEATURES —M.L.→ PHONEMES → WORDS → TRANSCRIPT (?)

END-TO-END:

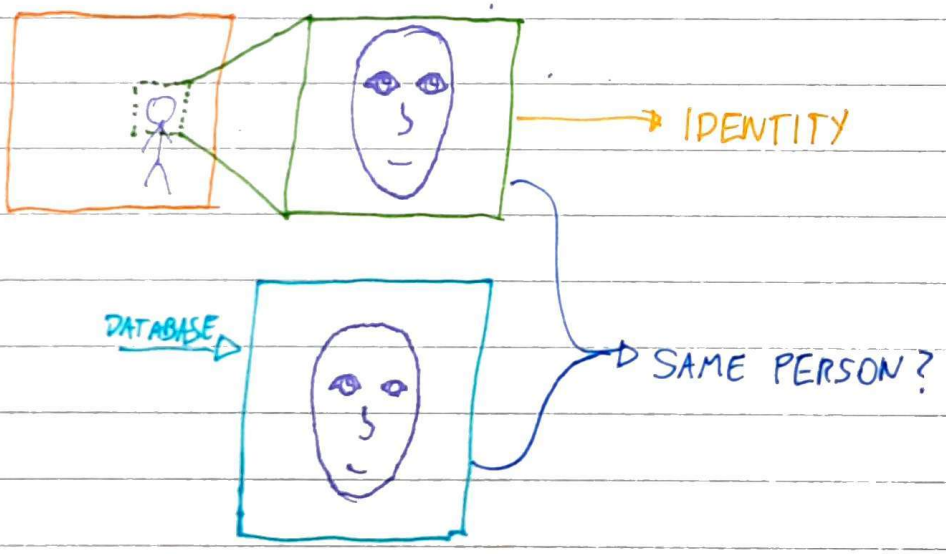
AUDIO → TRANSCRIPT

INTERMEDIATE APPROACH:

AUDIO → PHONEMES → TRANSCRIPT

* EXAMPLE: FACE RECOGNITION

- DOESN'T USE END-TO-END: SEPARATE APPROACH STILL WORKS BETTER



* MORE EXAMPLES

- MACHINE TRANSLATION
- ESTIMATE CHILD'S AGE BY RADIOLOGY IMAGE

→ WHETHER TO USE END-TO-END DEEP LEARNING

* PROS AND CONS OF END-TO-END DEEP LEARNING

PROS:

- LETS THE DATA SPEAK
- LESS HAND-DESIGNING OF COMPONENTS NEEDED

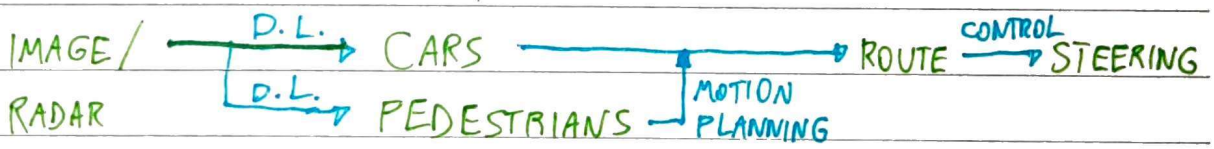
CONS:

- MAY NEED LARGE AMOUNT OF DATA
- EXCLUDES POTENTIALLY USEFUL HAND-DESIGNED COMPONENTS

* APPLYING END-TO-END IN DEEP LEARNING

KEY QUESTION: DO YOU HAVE SUFFICIENT DATA TO LEARN A FUNCTION OF THE COMPLEXITY NEEDED TO MAP X TO Y?

* EXAMPLE: AUTONOMOUS CAR



- USE DEEP LEARNING TO LEARN INDIVIDUAL COMPONENTS
- CAREFULLY CHOOSE X→Y DEPENDING WHAT TASKS YOU CAN GET DATA FOR