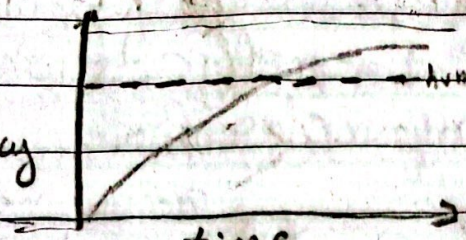
→ why human level performance



• progress relatively fast as you approach human performance after surpassing human then prog. accuracy is slow & the performance never passes Theoretical limit called The Bayesian optimal error

• as long as ML is worse Than humans you can:
① get labeled data from humans ② gain insights from manual error analysis
③ Better analysis of Bias/Variance

• The distance between Bayesian error & the training error called The avoidable bias & the distance between training error & Deverror is Variance.

Humans ≃ bayesian 7.5%
train error 8% } Variance 2%
dev error 10%

*(Course 3)(week2) error analysis

→ Carrying out error analysis

• if learning algorithm is not up to The level of human, then directly identifying the mistakes The algorithm is making will give you insights on what to do next & this process is called error analysis

→ cleaning up incorrectly labelled Data.
• If error have significant impact on evaluating alg. using development set then we should spend time fixing mislabeling

→ Build your first system quickly, then iterate
① setup dev/test set & metric
② Build initial system quickly.
③ use Bias/Variance analysis & error analysis to prioritize next step

→ mismatched training & dev/test data
→ training & testing on diff. distributions

Data from web
200000

Data from mobile app.
10000

→ shuffle

X option1

train	dev	test
205,000	2,500	2500

} with shuffle

option 2

web & mob	dev	test
205000	mob 2,5	2,5

} without

→ Bias & Variance with mismatch data distributions
• we define training-dev set: same distribution as training set but not used for training.

train	train-dev	dev	test

training error	1%	} Variance
train-dev error	9%	
dev error	10%	} data mismatch
test error	10%	} degree of overfitting

PAGE
DATE

→ Addressing data mismatch

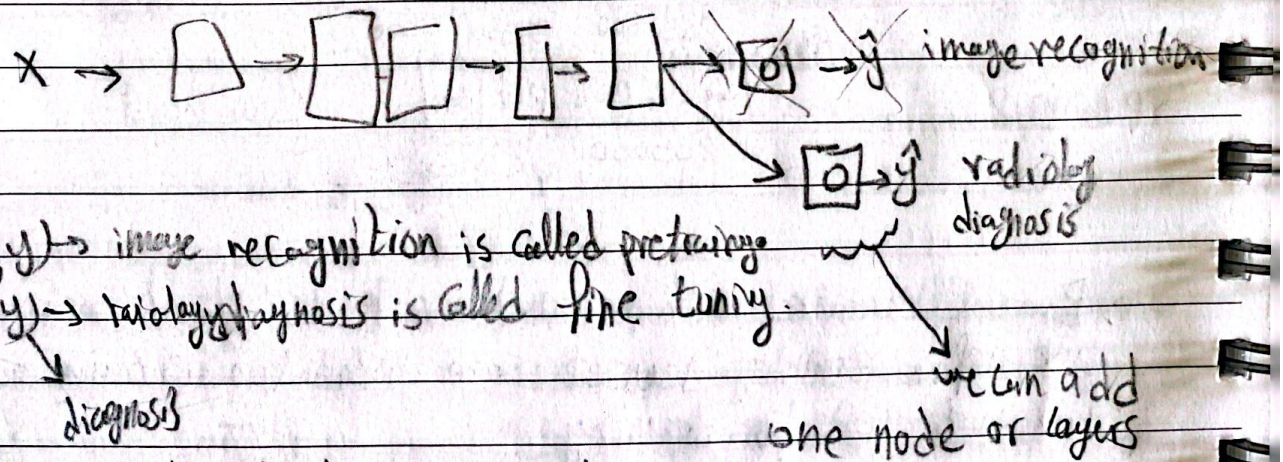
- if training set has different distribution than the dev. test set & error analysis may show data inconsistency issues we can do the following

① Addressing data mismatch:

- Carry out manual error analysis to try to understand the diff. between training & dev/test sets.
- Make training data similar (or) collect more data similar to dev/test sets

→ transfer learning

- one of the powerful ideas in deep learning is that to take knowledge the neural network has already learned one task & apply this knowledge to separate task



(x, y) → image recognition is called pretraining

(x, y) → radiology diagnosis is called fine tuning.

x: radio y: diagnosis

- When transfer learning makes sense

if we want transfer from task A → tranfer knowledge B

task A & b have the same input (x) → images or voice

- transfer learning is useful when task b has less data & increase learning speed

RND Dr Mo PAGE DATE

→ multi tasks learning

- in transfer learning it is sequential process, you learn from task A & transfer to task B but in multi task neural network can perform multiple tasks simultaneously & each task helps the remaining tasks
- rather than a seprate model for classifing scene, segmentation, detection we do it in one mod
- when multi-task learning makes sense
  - ✓ trainy on set of tasks that could benefit from having shared lower level features
  - ✓ train a big nn to do well on all tasks

→ what is end to end deep learning

- it is Data processing systems or learning systems that sometimes require multiple processing steps & end to en takes those multiple steps & rearrange them using sing neural network.

eg: X (audio) → MFCC → features → ML → phonemes → words → transc

in end to end learning involves training network

audio → transcript

- acc / evaluate / valad