2 Fundamental Assumptions of Supervised Learning

1> You can fit the training set pretty well.
→ Small Avoidable Bias

2> The training set performance generalizes pretty well to the dev/test set.
→ Small Variance

Human-Level
Training Error — Avoidable Bias
- Bigger Model
- Train Longer
- Better Optimization Algorithms
- NN Architecture, Hyperparameter Search

Dev. Error — Variance
- More Data
- Regularization
- NN Architecture, Hyperparameter Search

Error Analysis

Look at dev examples to evaluate ideas
1> Set ~100 mislabeled dev set examples
2> Count up how many are dogs — 5% not worth it / 50% worth it!

Evaluate multiple ideas in parallel

Incorrectly Labeled Data

	Error Type 1	Error Type 2	...	Error Type n
Sample 1				
Sample 2				
⋮				
Sample 100				
% of Total	8%	50%		20%

↑ most worthwhile "Prioritize"

Cleaning Up Mislabeled Samples

1) DL Algorithms are quite robust to random errors in the training set if dataset is sufficiently large (rather than systematic errors)

2) Error Analysis

Look at 3 numbers

Overall Dev Set Error	10%	2%
Errors Due to incorrect labels	0.6%	0.6%
Errors Due to other causes	9.4%	1.4%

Goal of Dev set is to help you select between two different models.

Dev set and Test Set should come from the same distribution.

Correct mislabeled data in both Dev and Test set.

But not necessarily training data.

Build Your First System Fast Then Iterate

Speech Recognition Example

→ Noisy Background

→ Accented Speech

→ Children's Speech

1. Set up dev/test set and metric
2. Build initial system quickly
3. Use Bias/Variance Analysis & Error Analysis to prioritize next steps

Quick & Dirty → Error Analysis → Iterate

Training & Testing on different distributions

Ex) Data from Webpage Data from App

Training: Web + App

Dev: App Only

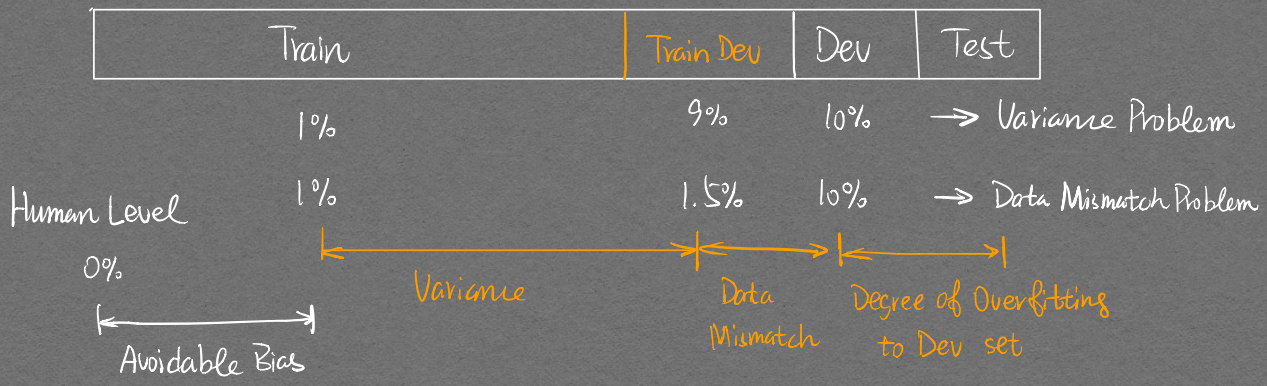
Test: App Only

Target is Right

Ex 2

Train	Dev / Test
Smart Speaker Control	Part of Speech Activated
Voice keyboard	Revere Mirror
Purchased Data	
Part of Speech Activated	
Revere Mirror	

Training Error 1%
Dev Error 10%
} 9% — Variance Problem
Different Distribution Data Mismatch

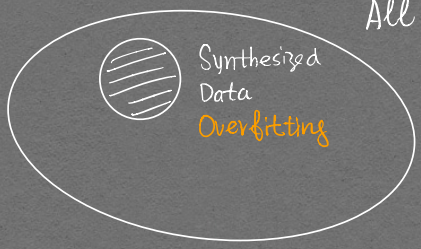


	General Speech Recognition	Revere Mirror Speech Data
Human Level	4%	6%
Error on examples trained on	7%	6%
Error on examples not trained on	10%	6%

Addressing Data Mismatch

1. Error Analysis to understand difference between training and dev/test
2. Make Data more similar / Collect more similar data to dev/test

→ Artificial Data Synthesis



Transfer Learning

Low-level Features are perhaps useful for different tasks

Lines	Phonemes
Dots	Noises
Colors	
Image Recognition	Speech Recognition

Take out the last layer

Randomly Initialize Weights Matrixs in the last layer $W^{(L)}, b^{(L)}$

Pretraining → Finetuning → New Model

A lot of Data	Small Data

When transfer learning makes sense

1) Task A & B has the same input X
2) A lot more data for A than B
3) Shared low-level features

Multi-task Learning

loss $\hat{y}^{(i)}$ (4,1) $\rightarrow \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{4} \mathcal{L}(\hat{y}_j^{(i)}, y_j^{(i)})$ Multilabels

When Multitask learning makes sense

1) Shared low-level features
2) Amount of data for each task is similar
3) Train a big neural network is feasible

End-to-end Deep Learning

Traditional Pipeline with muliple stages

X (audio) —MFCC→ Features —ML→ Phonemes → word → y (transcript)

End-to-end Deep Learning (skip intermediate stages)

X (audio) ⟶ y (transcript)

Pros

→ Let the data speak
→ Less hand designing of components

Cons

→ Requires large amount of data
→ Exclude potential useful hand-designed components

knowledge ← Data
knowledge ← Hand-designed Features