🙂

Deep AI: Structuring Machine Learning Projects

Set Up Targets

Single Judgement Matric

- 一般来说，应该有一个单一数值作为评价指标会更好
- 如果存在多个指标，往往可以进行分解： 那些是需要满足的(satisfactory)，哪些是需要优化的(optimization)
- 如果有多个需要优化的指标，可以通过不同的均值方式将他们组合
- 除了通用的一些指标，我们可以通过给样本加权的方式，使得获得更加适合我们应用的评价指标

Train/Dev/Test Distributions

我们应该尽可能的保证训练，验证和测试集的分布一致。这样才能保证，我们训练的目标和实际运用（测试）的目标一致。

Choose a dev set and test set to **reflect data you expect to get in the future** and consider important to do well on

举例来说，如果有不同国家的猫的照片，将他们粗暴的按照国别分类，就会造成训练集，验证集和测试集的分布不一致

Size of Dev and Test Sets

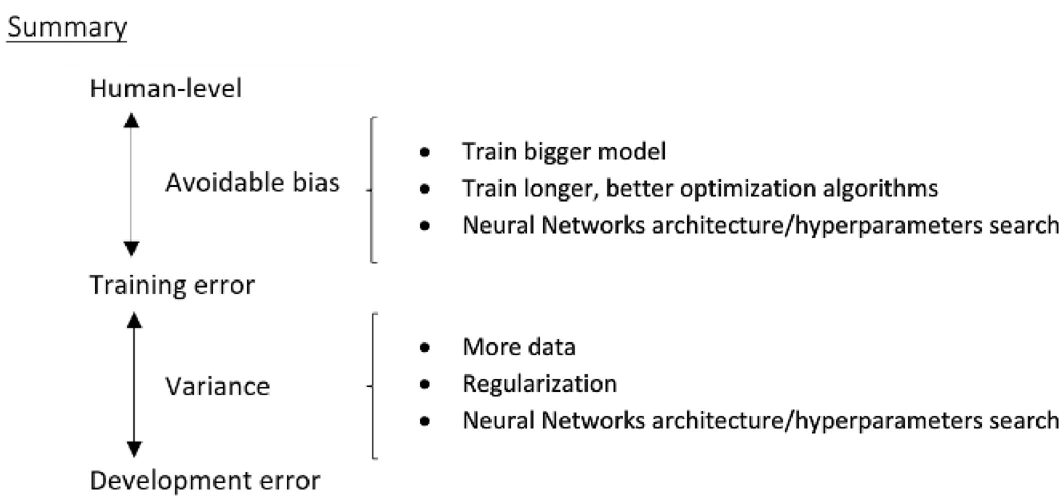
- Set your dev set to be big enough to detect differences in algorithm/models you're trying out.
- Set your test set to be big enough to give high confidence in the overall performance of your system.

也就是说，我们没有必要严格按照比例划分不同集合，如果数据量特别大，比如100000，我们完全可以使用90000训练集，然后使用5000验证集和5000测试集

Comparing to Human-level Performance

通常情况下，我们可以使用人类在某项任务上的表现，作为一种参考，作为对贝叶斯误差的估计。（但是对于有些任务，机器有很大可能比人类表现更好，这种情况另当别论）

然后通过与人类的比较，可以进行如下划分， 并且针对不同部分使用不同的策略



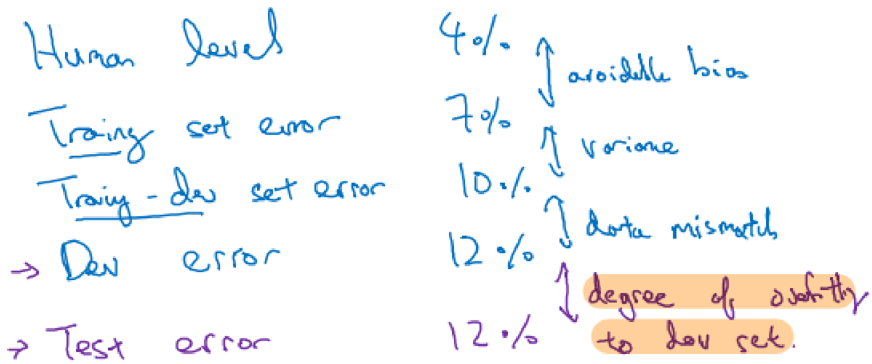
Error Analysis

Cleaning up Incorrectly Labeled Data

- DL algorithms are quite robust to random errors in the training set
- Apply same process to your dev and test sets to make sure they continue to come from the same distribution
- Train and dev/test data may now come from slightly different distributions

Mismatched Training and Dev/Test Set

If there is quite big different between training and dev/test set, it is a good idea to prepare a **training-dev** set, which is splited from the training set and have the same distribution as the training set.

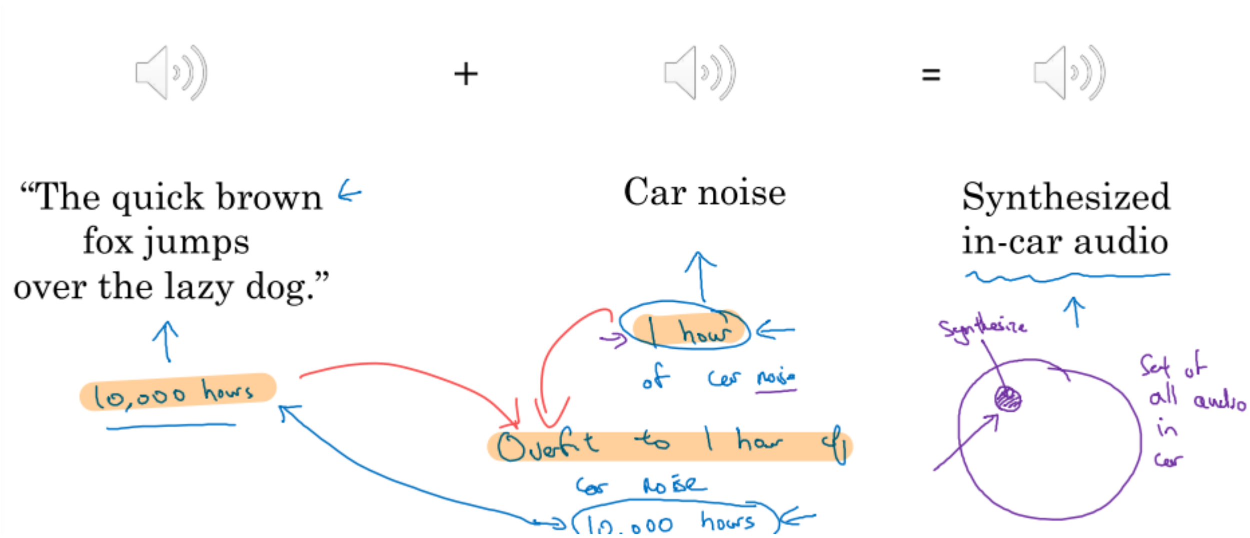


How to deal with data mismatch?

- 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

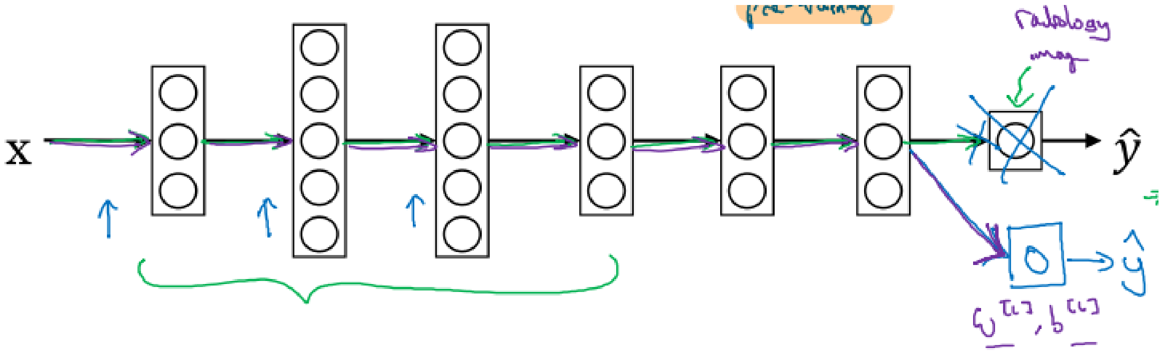
Attention:

If you want to add manual noisy/features to the training set to make the training data more similar, be careful for the overfitting



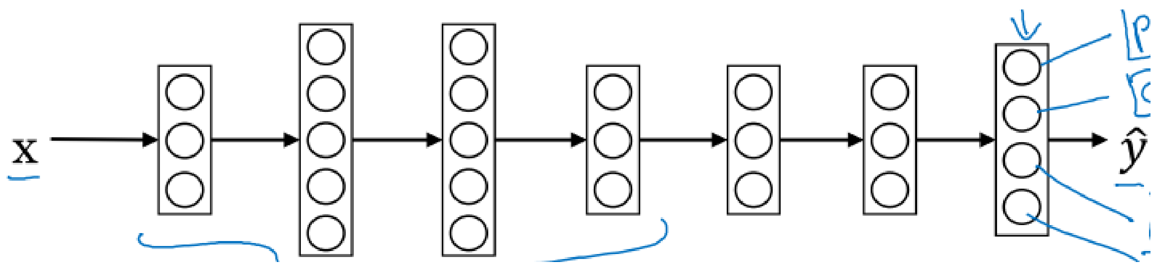
Transfer Learning and Multiple Tasks

Transfer Learning



- Task A and B have **the same input x**.
- You have **a lot more data** for Task A than Task B.
- **Low level features** from A could be helpful for learning B.

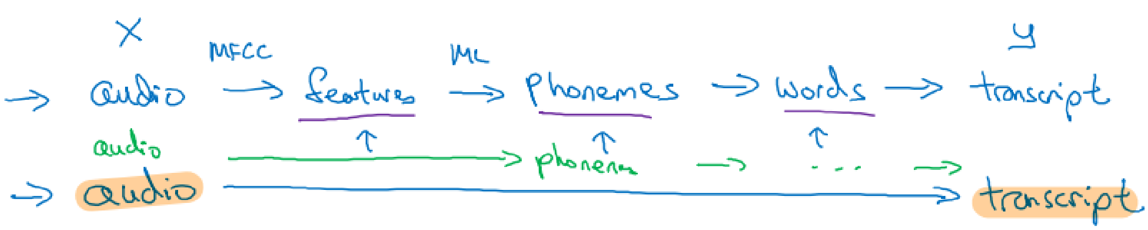
Multi-Task Learning



Multi-Task learning can be used when some labels are missing. If it is the case, when calculating error, do not take the missing labels into consideration

- Training on a set of tasks that could benefit from having shared lower-level features
- 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

End-to-End Deep Learning



Do you have sufficient data to learn a function of the complexity needed to map x to y

Pros:

- let the data speak
- less hand-designing of components needed

Cons:

- Many need large amount of data
- Excludes potentially useful hand-designed components