Course 3 : Structuring machine learning projects.

Content :

Week 1 : ML Strategy (1)

Week 2 : ML Strategy (2)

Coursera

Deep learning specialization

How to improve deep learning system?

1. collect more data
2. collect more diverse training set
3. Train algorithm longer with gradient descent
4. Try different optimization algorithm (Adam)
5. Try bigger network
6. Try smaller net work → hmm interesting
7. Try dropout
8. Add L2 Regularization
9. Change network architecture → ... etc.
   - activation fn
   - # of hidden units

fit training set well on cost fnc
- bigger net work
- better optimization algorithm, Adam

fit dev set well on cost fnc
- regularization
- bigger training set

fit test set well on cost fnc
- bigger dev set

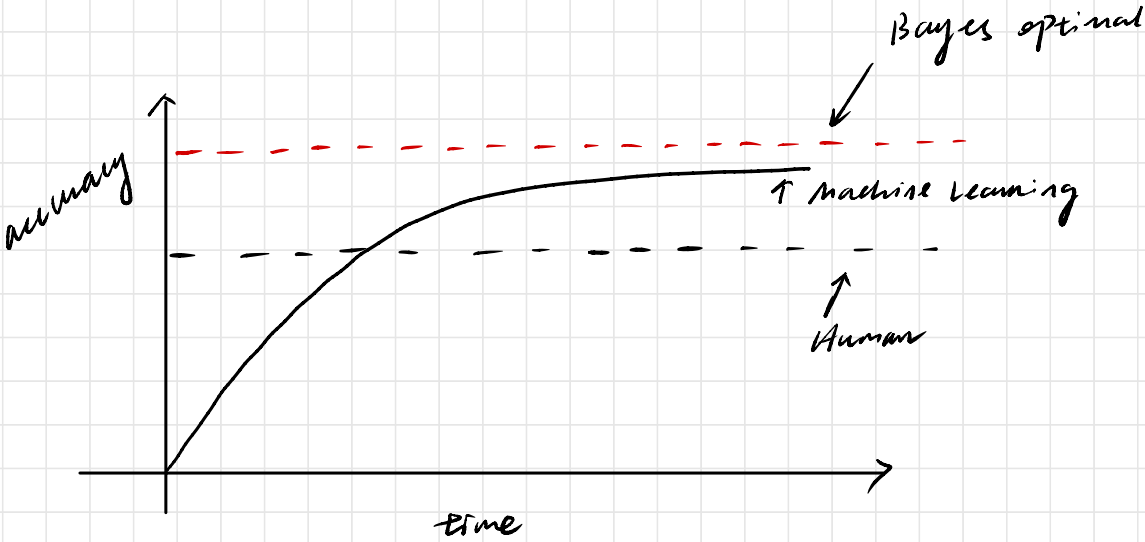
perform well in real world
- change dev set or cost fnc.

dev & test → should come from same distribution

Size of dev & test sets

old rule of thumb 70/30

98% train 1% dev 1% test



Avoidable bias: difference b/t the training set & the human level error

① bias reduction technique: training bigger N.N. or running the training set longer.

② Variance reduction technique: regularization or have a bigger training set.

m Human level error is a proxy for Bayes error.

m if difference b/t Human & Train >

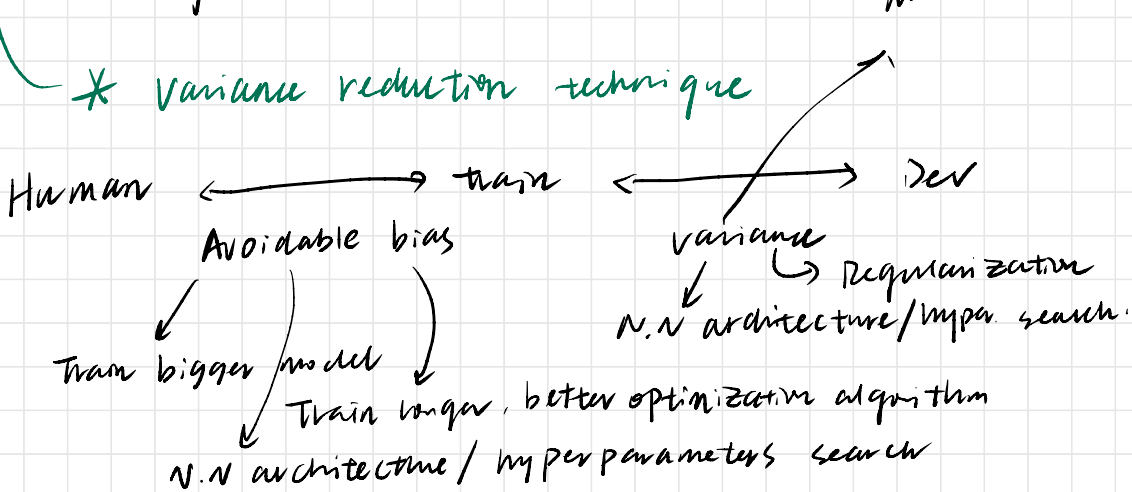
difference b/t Train & dev.

* bias reduction technique

m if difference b/t Train & dev >

difference b/t Human & Train

* Variance reduction technique



Data mismatch Addressing

* perform manual error analysis.

  should never be done on test set to avoid overfitting

* making training data / collect data similar to dev. & test sets.

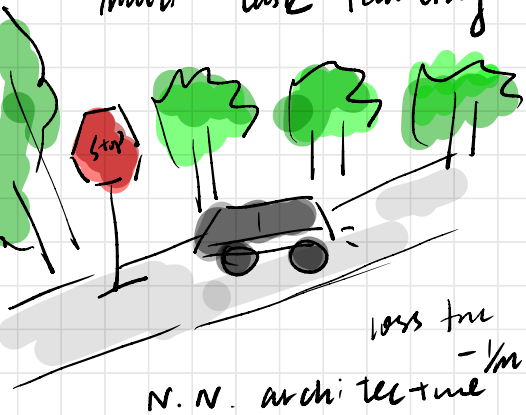
  artificial data synthesis.

Transfer learning

↓

Using N.N for another application.

multi-task learning



$y^{(i)} = \begin{bmatrix} 0 \\ 1 \leftarrow \text{cars} \\ 1 \leftarrow \text{road signs STOP} \\ 0 \end{bmatrix}$ ← pedestrians

↑ Traffic lights

loss fnc $-\frac{1}{m}\sum_{i}\sum_{j}\left(y_j^{(i)}\log\left(\hat{y}_j^{(i)}\right)+\left(1-y_j^{(i)}\right)\log\left(1-\hat{y}_j^{(i)}\right)\right)$

N.N. architecture