Why human-level Performance?

<Comparing to human-level performance>

never surpasses
Some theoretical limit (Base error)

Bayse Optimal
Error
|
best possible
error

actually slows down

Slow down ↑

↑ human
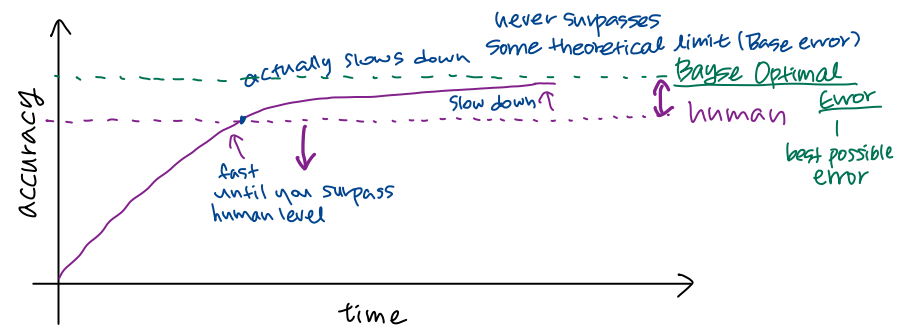
↑ fast
until you surpass
human level

$X \to Y$
audio   transcript
image   cat (0/1)

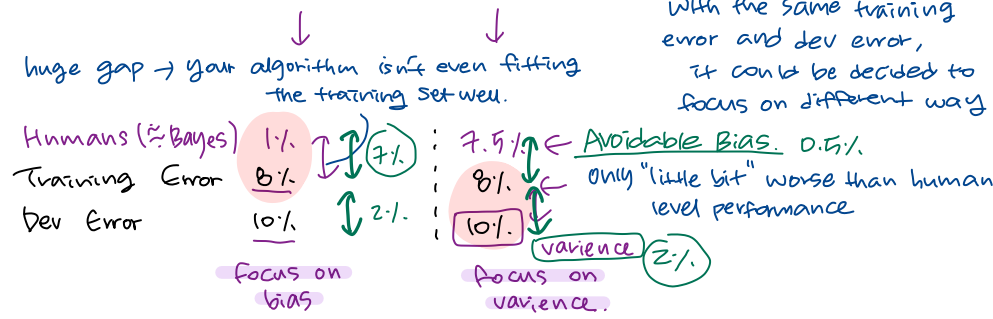<Why compare to human-level Performance>

Humans are <u>quite good at a lots of tasks</u>. So long as ML is worse than humans, you can:

- Get labled data from humans (x, y)
- Gain insight from manual error analysis:
  why did a person get this right?
- Better analysis of <u>bias / variance</u>.

ML algorithms tend to be <u>really good at kind of replicant tasks</u>

Avoidable Bias

<Cat Classification Example>

huge gap → your algorithm isn't even fitting the training set well.

With the Same training error and dev error, It could be decided to focus on different way

Humans (≈ Bayes)	1%	7.5% ← Avoidable Bias. 0.5%
Training Error	8%	8% ← Only "little bit" worse than human level performance
Dev Error	10%	10%

↕ 7%

↕ 2%

focus on bias

variance ↕ 2%

focus on variance.

Human level error as a proxy for Bayes error. (ok.. (computer vision))

by definition, human level error is worse than Bayes error
∴ error smaller than Bayes error doesn't exists.

(Avoidable Bias: Some bias / Some minimum level of error that you just cannot get below which is that if Bayes error 7.5%.. You don't actually want to get that level of error..)

Understanding human-level performance

< Human-level error as a proxy for Bayes error >
gives us a way of estimating Bayes error

Medical image classification example:

radiology image

Suppose:

  (a) Typical human   3% error
→ (b) **Typical doctor**   1% error
  (c) Experienced doctor   0.7% error
→ (d) **Team of Experienced doctor**   0.5% error ←

Bayes error ≤ 0.5%

if you want a proxy / an estimate for Bayes error
→ giving (d) can achieve 0.5 percent error

What is "human level error?"

'What your purpose is'
ex) - Surpass single human: (b)
    - proxy for Bayes error: (d)

< Error analysis Example >

whether 4% or 4.5% error it is clearly bigger than 1% error

Human (proxy for Bayes error)   1% / 0.7% / 0.5%

Avoidable Bias ↕
Training error   ↕ (4% 4.5%) ✓   5%

Variance ↕
Dev error   ↕ 1%   6%

"Bias"

further away from human level performance → easier to choose bias or variance.

1% / 0.7% / 0.5%
↕ 0%   0.5%
1%
↕ (4%)
5%

"variance"

really matters that you use..

0.7% / 0.5%
↕ (0.2%) ←
0.7%   0.0%
↕ (0.1%) ←
0.8%

< Summary of bias / variance with human level performance >

0%
↕
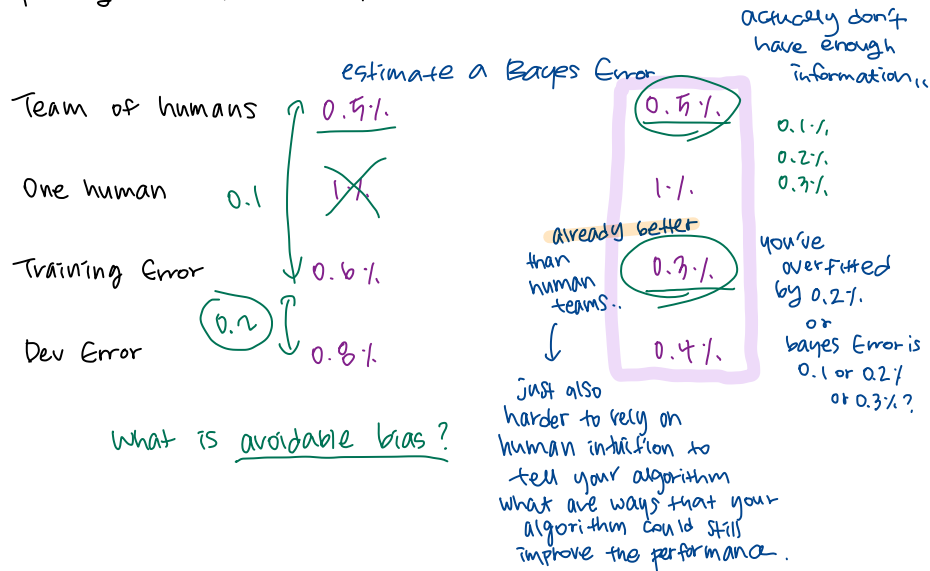"Bias" ↕   Human-level error (Prox for Bayes error)   ↕ "Avoidable bias"
↓   Training error

↕ "variance"

Dev error

Bayes error ≠ 0
nothing can surpass it

Bayes error ≈ 0

noisy data (speech).. → Bayes error ⇒ can use avoidable bias, variance to choose what to focus.

Surpassing Human-Level Performance

⟨Surpassing Human-Level Performance⟩

estimate a Bayes Error

Team of humans ↗ 0.5%

One human 0.1 ✗ 1%

Training Error ↓ 0.6%

Dev Error (0.2) ↕ 0.8%

What is _avoidable bias_?

actually don't have enough information...

(0.5%) 0.1% 0.2% 0.3%

already better
than human teams..
(0.3%)

you've overfitted by 0.2%
or bayes Error is 0.1 or 0.2%
or 0.3%?

(0.4%)

↓ just also harder to rely on human intuition to tell your algorithm what are ways that your algorithm could still improve the performance.

⟨Problems where ML significantly surpasses human-level performance⟩

- Online advertising
- Product recommendations
- Logistics (Predicting transit time)
- Loan approvals

Structure Data
Not natural perception (ex. Computer vision, speech recognition, natural language processing tasks..)
Lots of Data

⟹ human is much better at natural perception field.

can surpass human in some cases, but it has been a bit harder ∴ human tend to be very good at these sort of natural perception

- Speech recognition
- Some image recognition
- Medical
  - ECGs, Skin cancer, ...

Improving Your model Performance

⟨The two fundamental assumptions of supervised learning⟩

1. You can fit the training set pretty well.
   Low Avoidable Bias

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

- training bigger network
- training longer

- regularization
- getting more train data.

⟨Reducing (avoidable) bias and variance⟩

Human-level
↕ Avoidable bias →
Training error
↓ Variance →
Dev error

{ Train bigger model
Train longer/better optimization algorithms
  - add momentum, RMSprop, Adam..
NN architecture/hyperparameters search   RNN CNN

include everything from changing the activation function to changing the number of layers or hidden units..

{ More data
Regularization
  - L2, dropout, data augmentation
NN architecture/hyperparameters search