deeplearning.ai

Introduction to ML strategy

Why ML Strategy?

Motivating example






90%

Ideas:

- Collect more data
- Collect more diverse training set
- Train algorithm longer with gradient descent
- Try Adam instead of gradient descent
- Try bigger network
- Try smaller network
- Try dropout
- Add L_2 regularization
- Network architecture
  - Activation functions
  - # hidden units
  - …

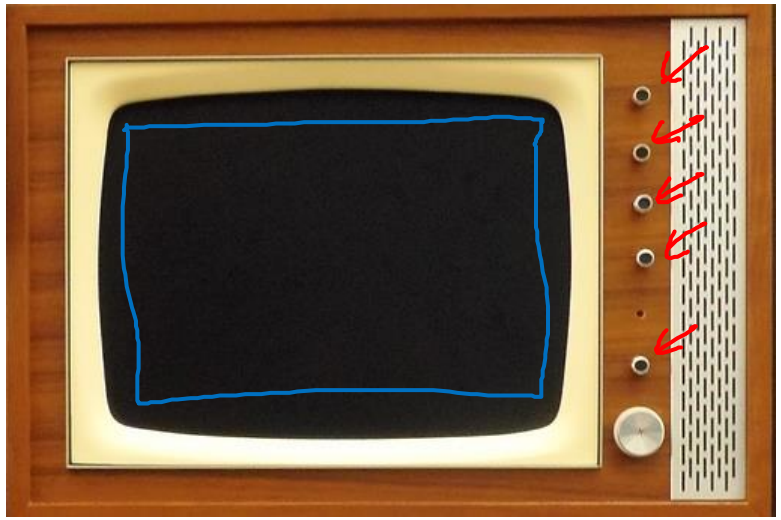
Andrew Ng

deeplearning.ai

Introduction to ML strategy

Orthogonalization

TV tuning example



Orthogonalization

0.1 x

+ 0.3 x

- 1.7 x

+ 0.8 x

+ ...

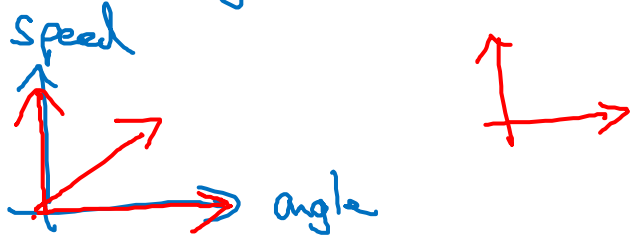
Car

→ Steering

→ {Acceleration, Braking}

→ 0.3 x angle − 0.8 speed

→ 2 x angle + 0.9 speed.



Andrew Ng

Chain of assumptions in ML

→ Fit training set well on cost function

(≈ human-level performance)

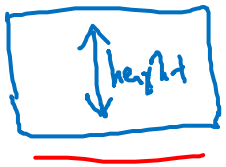
- Bigger network
- Adam
- ...

↓

→ Fit dev set well on cost function

- Regularization
- Bigger training set

early stopping



↓

→ Fit test set well on cost function

Bigger dev set

↓

→ Performs well in real world

(Happy cat pic app users.)

Change dev set or cost function

Orthogonalization

Orthogonalization or orthogonality is a system design property that assures that modifying an instruction or a component of an algorithm will not create or propagate side effects to other components of the system. It becomes easier to verify the algorithms independently from one another, it reduces testing and development time.

When a supervised learning system is design, these are the 4 assumptions that needs to be true and orthogonal.

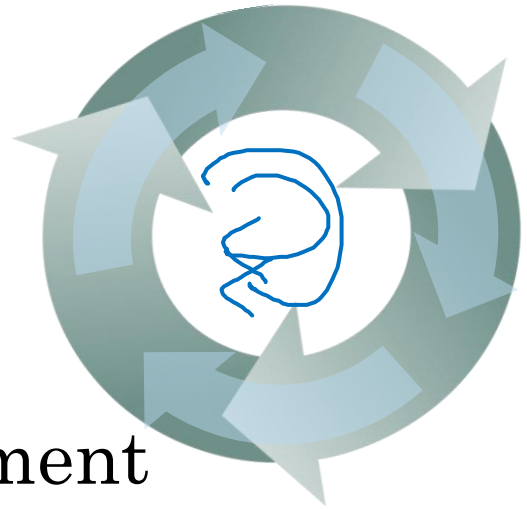
1. Fit training set well in cost function
   - If it doesn't fit well, the use of a bigger neural network or switching to a better optimization algorithm might help.
2. Fit development set well on cost function
   - If it doesn't fit well, regularization or using bigger training set might help.
3. Fit test set well on cost function
   - If it doesn't fit well, the use of a bigger development set might help
4. Performs well in real world
   - If it doesn't perform well, the development test set is not set correctly or the cost function is not evaluating the right thing.

deeplearning.ai

Setting up your goal

Single number evaluation metric

Using a single number evaluation metric



Of examples recognized as cat, what % actually are cats?

What % of actual cats are correctly recognized

Classifier	Precision	Recall
A	95%	90%
B	98%	85%

F1 score = "Average" of P and R.

$\left(\frac{2}{\frac{1}{P}+\frac{1}{R}}\right.$. "Harmonic mean")

Dev set + Single number evaluation metric

real

Speed up iterating

Another example

Algorithm	US	China	India	Other
A	3%	7%	5%	9%
B	5%	6%	5%	10%
C	2%	3%	4%	5%
D	5%	8%	7%	2%
E	4%	5%	2%	4%
F	7%	11%	8%	12%

Single number evaluation metric

To choose a classifier, a well-defined development set and an evaluation metric speed up the iteration process.

Example : Cat vs Non- cat

y = 1, cat image detected

Predict class $\hat{y}$ \ Actual class y	1	0
1	True positive	False positive
0	False negative	True negative

Precision

Of all the images we predicted y=1, what fraction of it have cats?

Precision (%) = $\frac{True\ positive}{Number\ of\ predicted\ positive} x\ 100 = \frac{True\ positive}{(True\ positive + False\ positive)}\ x\ 100$

Recall

Of all the images that actually have cats, what fraction of it did we correctly identifying have cats?

Recall (%) = $\frac{True\ positive}{Number\ of\ predicted\ actually\ positive} x\ 100 = \frac{True\ positive}{(True\ positive + True\ negative)}\ x\ 100$

Let's compare 2 classifiers A and B used to evaluate if there are cat images:

Classifier	Precision (p)	Recall (r)
A	95%	90%
B	98%	85%

In this case the evaluation metrics are precision and recall.

For classifier A, there is a 95% chance that there is a cat in the image and a 90% chance that it has correctly detected a cat. Whereas for classifier B there is a 98% chance that there is a cat in the image and a 85% chance that it has correctly detected a cat.

The problem with using precision/recall as the evaluation metric is that you are not sure which one is better since in this case, both of them have a good precision et recall. F1-score, a harmonic mean, combine both precision and recall.

$$\text{F1-Score} = \frac{2}{\frac{1}{p}+\frac{1}{r}}$$

Classifier	Precision (p)	Recall (r)	F1-Score
A	95%	90%	92.4 %
B	98%	85%	91.0%

Classifier A is a better choice. F1-Score is not the only evaluation metric that can be use, the average, for example, could also be an indicator of which classifier to use.

deeplearning.ai

Setting up your goal

Satisficing and optimizing metrics

Another cat classification example

optimizing ↓ ↓ satisficing

Classifier	Accuracy	Running time
A	90%	80ms
B	92%	95ms
C	95%	1,500ms

Cost = accuracy − 0.5 * running Time

Maximize accuracy

subject to runningTime ≤ 100 ms.

N metrics: 1 optimizing

N−1 satisficing

Wakewords / Trigger words

Alexa, OK Google,

Hey Siri, Nihaobaidu

你好百度

accuracy.

#false positive

Maximize accuracy.

s.t. ≤ 1 false positive

every 24 hours.

Andrew Ng

Satisficing and optimizing metric

There are different metrics to evaluate the performance of a classifier, they are called evaluation matrices. They can be categorized as satisficing and optimizing matrices. It is important to note that these evaluation matrices must be evaluated on a training set, a development set or on the test set.

Example: Cat vs Non-cat

Classifier	Accuracy	Running time
A	90%	80 ms
B	92%	95 ms
C	95%	1 500 ms

In this case, accuracy and running time are the evaluation matrices. Accuracy is the optimizing metric, because you want the classifier to correctly detect a cat image as accurately as possible. The running time which is set to be under 100 ms in this example, is the satisficing metric which mean that the metric has to meet expectation set.

The general rule is:

$$N_{metric}: \begin{cases} 1 & Optimizing\ metric \\ N_{metric} - 1 & Satisficing\ metric \end{cases}$$

deeplearning.ai

Setting up your goal

Train/dev/test distributions

Cat classification dev/test sets

development set, hold out cross validation set

Regions:

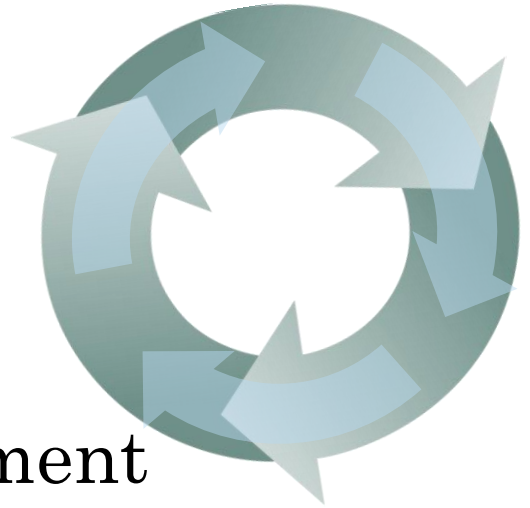
- US
- UK
- Other Europe
- South America
- India
- China
- Other Asia
- Australia

Dev

Test

Randomly shuffle into dev/test

dev set + metric



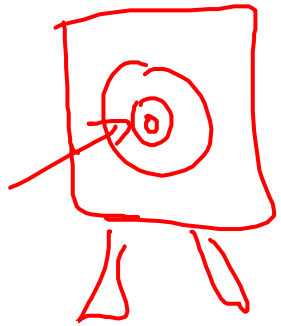
True story (details changed)

Optimizing on dev set on loan approvals for medium income zip codes

$x \rightarrow y$ (repay loan?)

Tested on low income zip codes

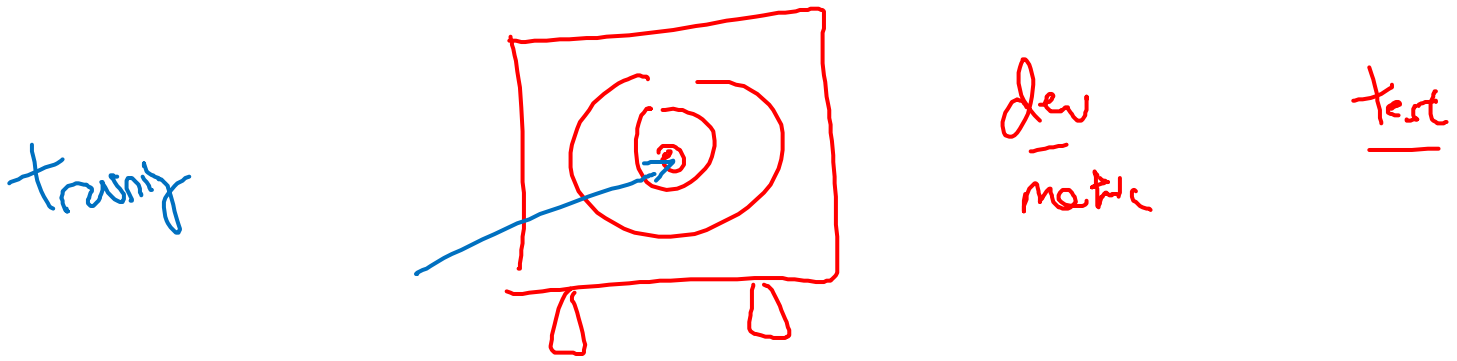
~3 month



Guideline

Same distribution

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



Training, development and test distributions

Setting up the training, development and test sets have a huge impact on productivity. It is important to choose the development and test sets from the same distribution and it must be taken randomly from all the data.

Guideline

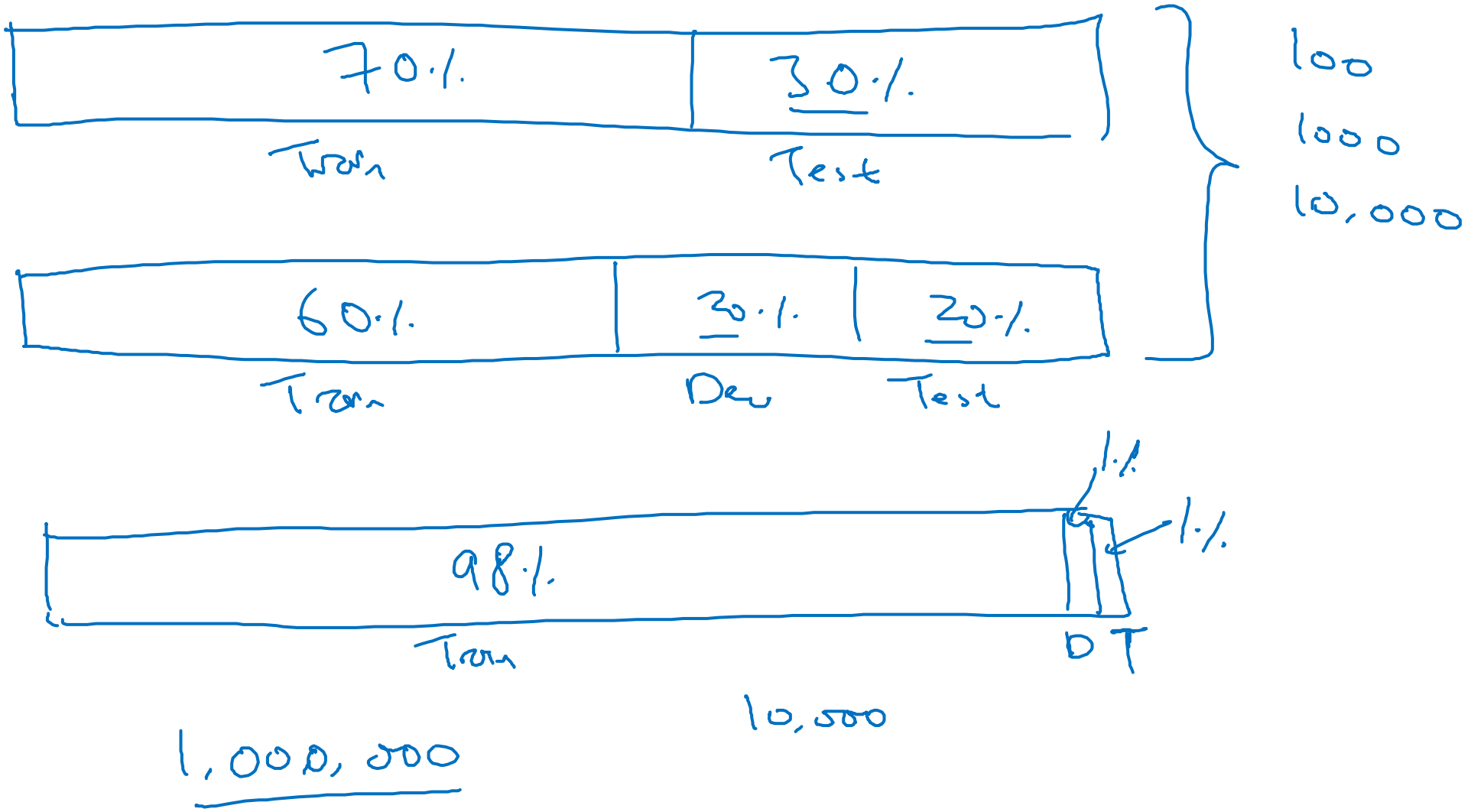
Choose a development set and test set to reflect data you expect to get in the future and consider important to do well.

deeplearning.ai

Setting up your goal

Size of dev and test sets

Old way of splitting data



Size of dev set

A B

Set your dev set to be big enough to detect differences in algorithm/models you're trying out.

A B

97% → 97.1 %

0.1%

100 : small

↳ 1%

1,000

10,000

100,000

0.01%

0.001%

Online advertising

Size of test set

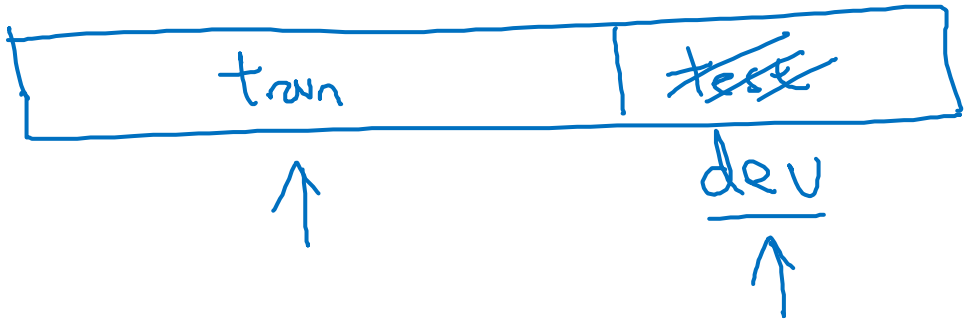
- Set your test set to be big enough to give high confidence in the overall performance of your system.

10,000

100,000

Train + Dev.

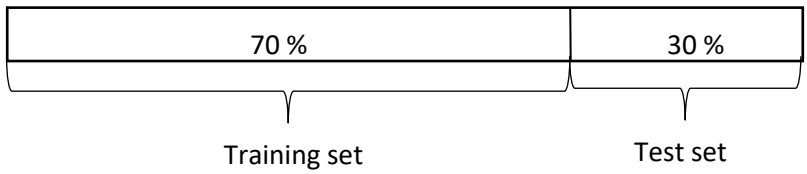
No test set.



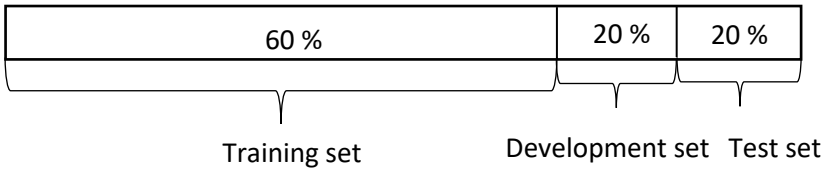
Size of the development and test sets

Old way of splitting data

We had smaller data set therefore we had to use a greater percentage of data to develop and test ideas and models.

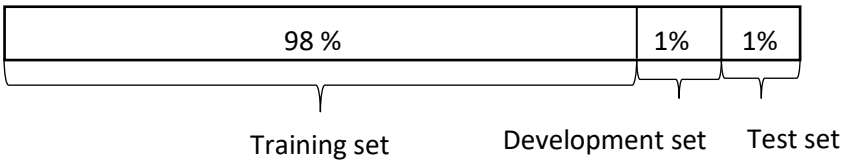


Or



Modern era – Big data

Now, because a large amount of data is available, we don't have to compromised as much and can use a greater portion to train the model.



Guidelines

- Set up the size of the test set to give a high confidence in the overall performance of the system.
- Test set helps evaluate the performance of the final classifier which could be less 30% of the whole data set.
- The development set has to be big enough to evaluate different ideas.

deeplearning.ai

Setting up
your goal

When to change dev/test sets and metrics

Cat dataset examples

Metric + Dev : Prefer A

Your users : Prefer B.

→ Metric: classification error

Algorithm A: 3% error → pornographic

✓ Algorithm B: 5% error

$$\text{Error}: \frac{1}{\sum w^{(i)}} \cancel{\frac{1}{m_{dev}}} \sum_{i=1}^{m_{dev}} w^{(i)} \mathbb{1}\{y_{pred}^{(i)} \neq y^{(i)}\}$$

$y_{pred}^{(i)}$: predicted value (0/1)

$$\rightarrow w^{(i)} = \begin{cases} 1 & \text{if } x^{(i)} \text{ is non-porn} \\ 10 & \text{if } x^{(i)} \text{ is porn} \end{cases}$$

Orthogonalization for cat pictures: anti-porn

1. So far we've only discussed how to define a metric to evaluate classifiers. ← Place target

2. Worry separately about how to do well on this metric. ← Aim/shoot at target

$$J = \frac{1}{\sum w^{(i)}} \sum_{i=1}^{m} w^{(i)} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$

Another example

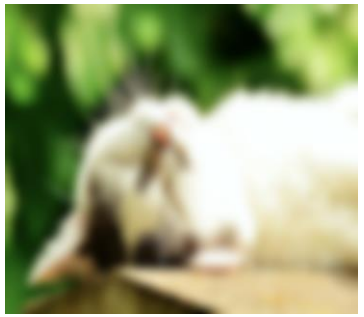
Algorithm A: 3% error

Algorithm B: 5% error

Dev/test



User images



If doing well on your metric + dev/test set does not correspond to doing well on your application, change your metric and/or dev/test set.

When to change development/test sets and metrics

Example: Cat vs Non-cat

A cat classifier tries to find a great amount of cat images to show to cat loving users. The evaluation metric used is a classification error.

Algorithm	Classification error [%]
A	3%
B	5%

It seems that Algorithm A is better than Algorithm B since there is only a 3% error, however for some reason, Algorithm A is letting through a lot of the pornographic images.

Algorithm B has 5% error thus it classifies fewer images but it doesn't have pornographic images. From a company's point of view, as well as from a user acceptance point of view, Algorithm B is actually a better algorithm. The evaluation metric fails to correctly rank order preferences between algorithms. The evaluation metric or the development set or test set should be changed.

The misclassification error metric can be written as a function as follow:

$$Error: \frac{1}{m_{dev}} \sum_{i=1}^{m_{dev}} \mathcal{L}\{(\hat{y}^{(i)} \neq y^{(i)}\}$$

This function counts up the number of misclassified examples.

The problem with this evaluation metric is that it treats pornographic vs non-pornographic images equally. On way to change this evaluation metric is to add the weight term $w^{(i)}$.

$$w^{(i)} = \begin{cases} 1 & if\ x^{(i)}\ is\ non-pornographic \\ 10 & if\ x^{(i)}\ is\ pornographic \end{cases}$$

The function becomes:

$$Error: \frac{1}{\sum w^{(i)}} \sum_{i=1}^{m_{dev}} w^{(i)} \mathcal{L}\{(\hat{y}^{(i)} \neq y^{(i)}\}$$

Guideline

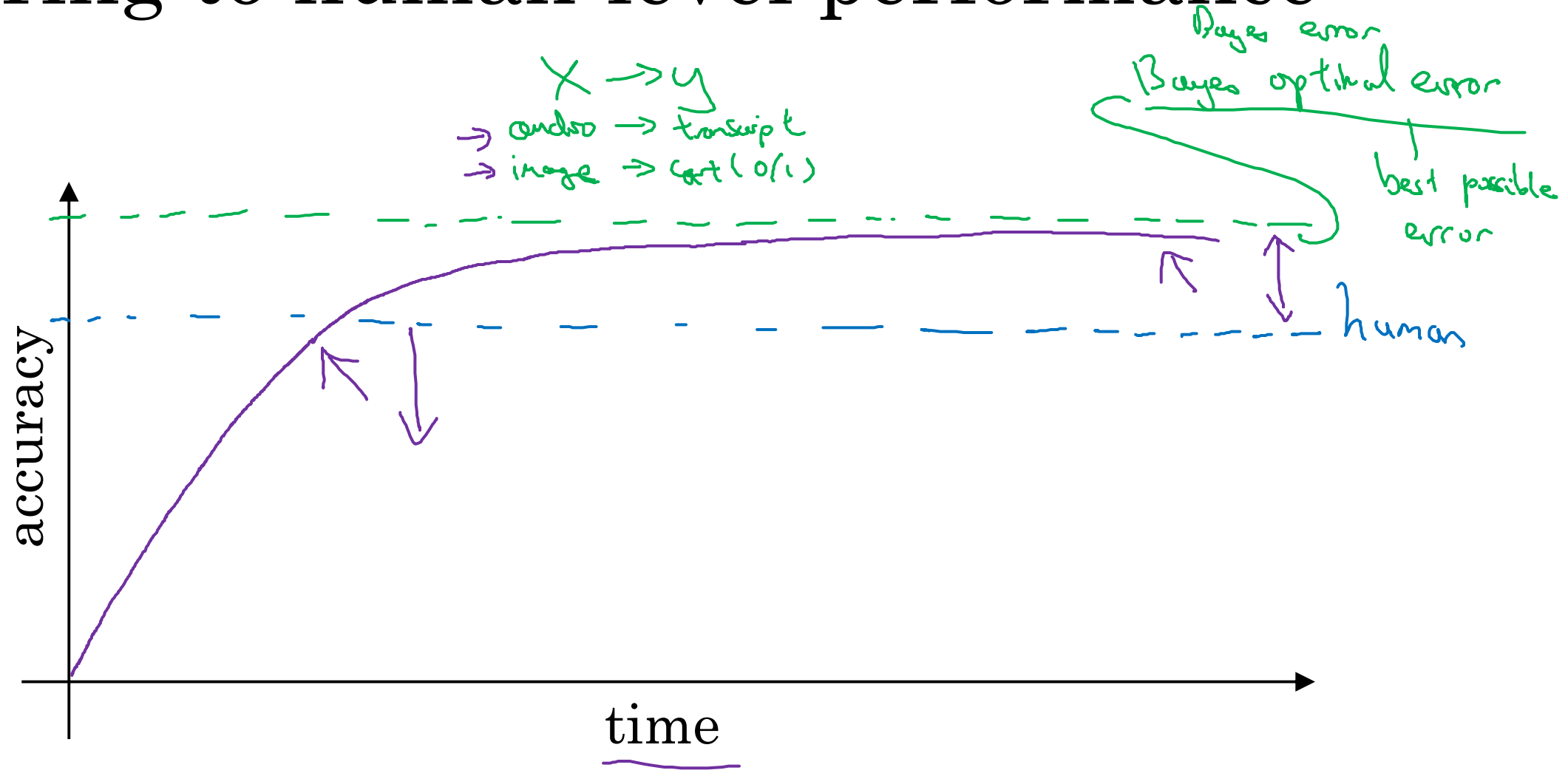
1. Define correctly an evaluation metric that helps better rank order classifiers
2. Optimize the evaluation metric

deeplearning.ai

Comparing to human-level performance

Why human-level performance?

Comparing to human-level performance



Why compare to human-level performance

Humans are quite good at a lot of tasks. So long as ML is worse than humans, you can:

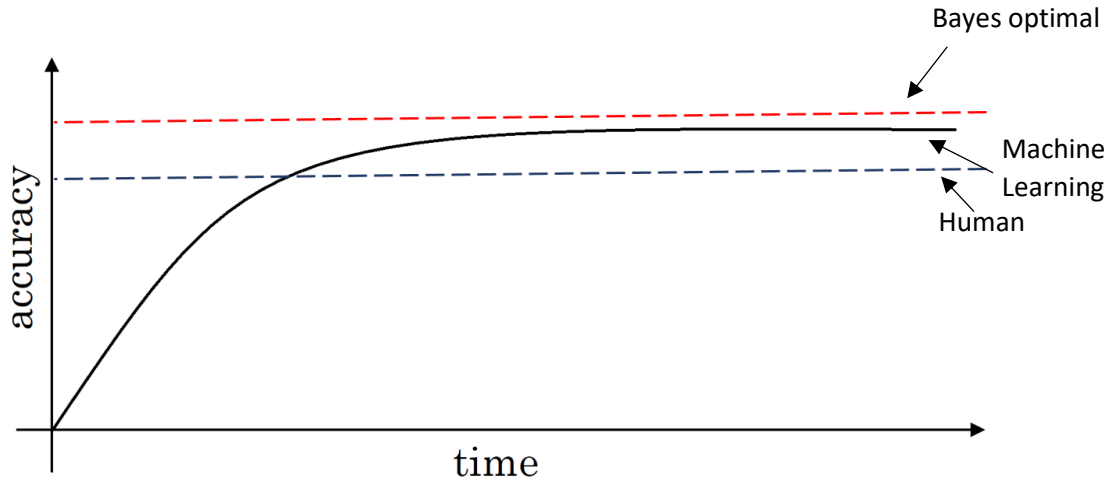
- Get labeled data from humans. (x, y)
- Gain insight from manual error analysis: Why did a person get this right?
- Better analysis of bias/variance.

Why human-level performance?

Today, machine learning algorithms can compete with human-level performance since they are more productive and more feasible in a lot of application. Also, the workflow of designing and building a machine learning system, is much more efficient than before.

Moreover, some of the tasks that humans do are close to ‘’perfection’’, which is why machine learning tries to mimic human-level performance.

The graph below shows the performance of humans and machine learning over time.



The

Machine learning progresses slowly when it surpasses human-level performance. One of the reason is that human-level performance can be close to Bayes optimal error, especially for natural perception problem.

Bayes optimal error is defined as the best possible error. In other words, it means that any functions mapping from x to y can’t surpass a certain level of accuracy.

Also, when the performance of machine learning is worse than the performance of humans, you can improve it with different tools. They are harder to use once its surpasses human-level performance.

These tools are:

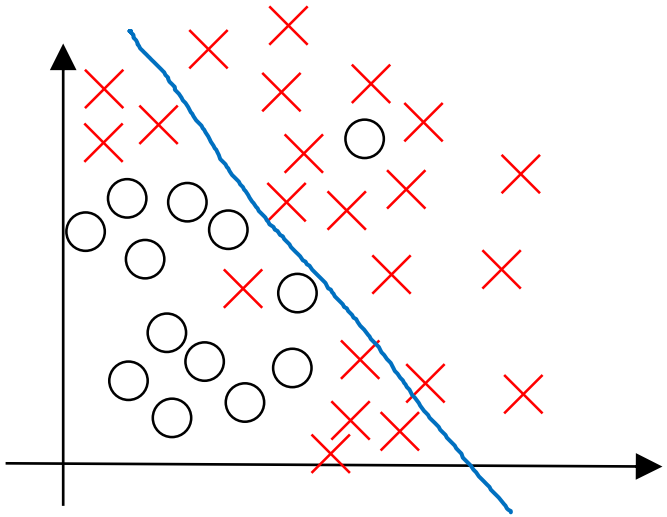
- Get labeled data from humans
- Gain insight from manual error analysis: Why did a person get this right?
- Better analysis of bias/variance.

deeplearning.ai

Comparing to human-level performance

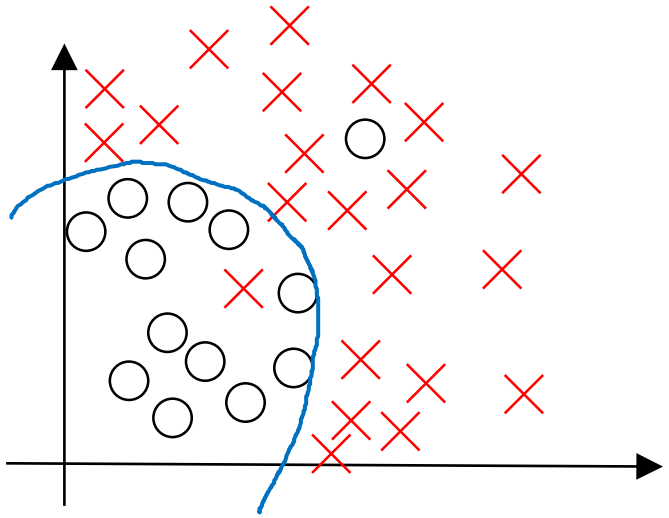
Avoidable bias

Bias and Variance

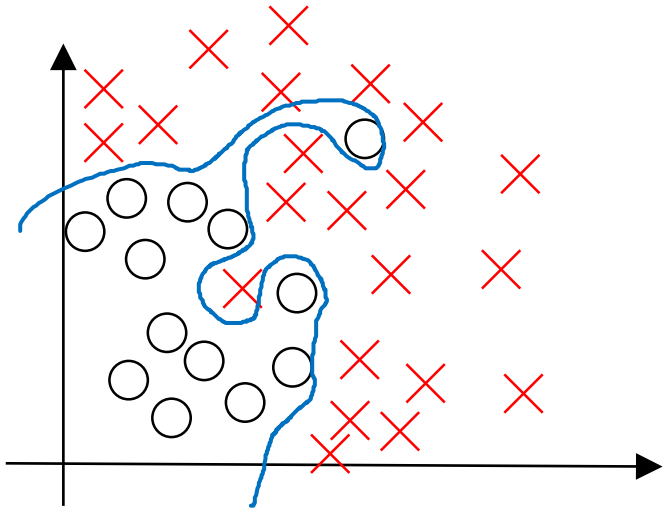


high bias

Underfitting



"just right"



high variance

overfitting.

Bias and Variance

Cat classification

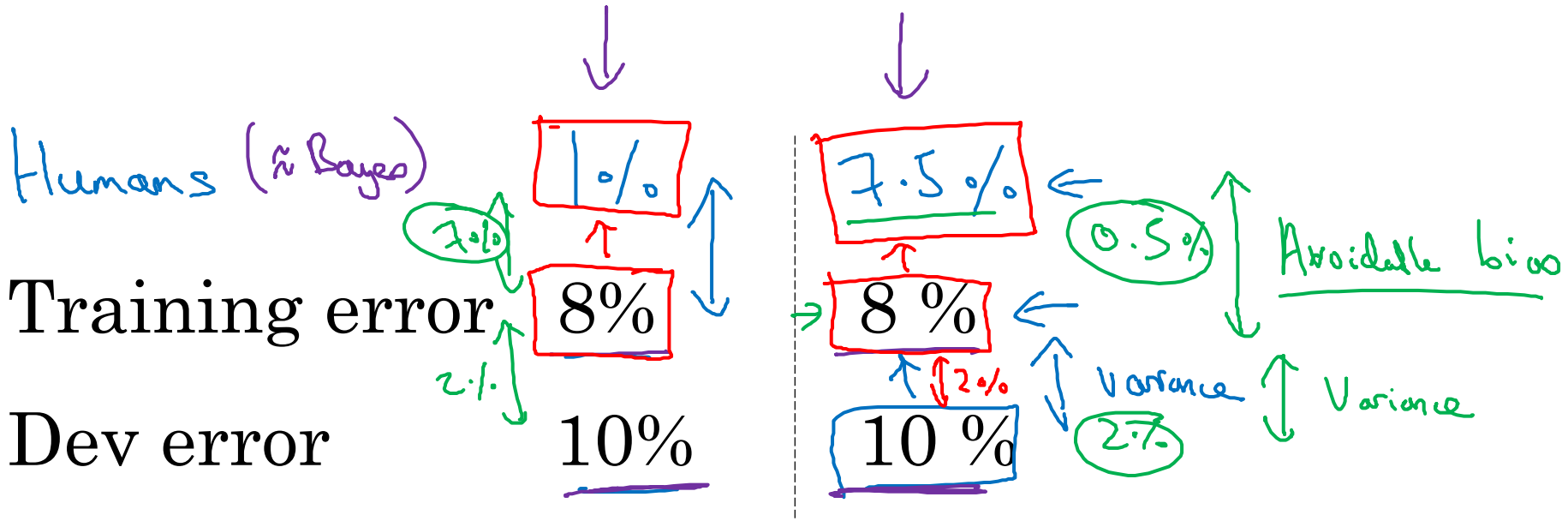
Human-level ≈ 0%

Training set error: 1% | 15% | 15% | 0.5%

Dev set error: 11% | 16% | 30% | 1%

high variance | high bias | high bias, high variance | low bias, low variance

Cat classification example



Human-level error as a proxy for Bayes error.

Avoidable bias

By knowing what the human-level performance is, it is possible to tell when a training set is performing well or not.

Example: Cat vs Non-Cat

	Classification error (%)	
	Scenario A	Scenario B
Humans	1	7.5
Training error	8	8
Development error	10	10

In this case, the human level error as a proxy for Bayes error since humans are good to identify images. If you want to improve the performance of the training set but you can't do better than the Bayes error otherwise the training set is overfitting. By knowing the Bayes error, it is easier to focus on whether bias or variance avoidance tactics will improve the performance of the model.

Scenario A

There is a 7% gap between the performance of the training set and the human level error. It means that the algorithm isn't fitting well with the training set since the target is around 1%. To resolve the issue, we use bias reduction technique such as training a bigger neural network or running the training set longer.

Scenario B

The training set is doing good since there is only a 0.5% difference with the human level error. The difference between the training set and the human level error is called avoidable bias. The focus here is to reduce the variance since the difference between the training error and the development error is 2%. To resolve the issue, we use variance reduction technique such as regularization or have a bigger training set.

deeplearning.ai

Comparing to human-level performance

Understanding human-level performance

Human-level error as a proxy for Bayes error

Medical image classification example:

Suppose:

(a) Typical human ..... 3 % error

(b) Typical doctor ..... 1 % error

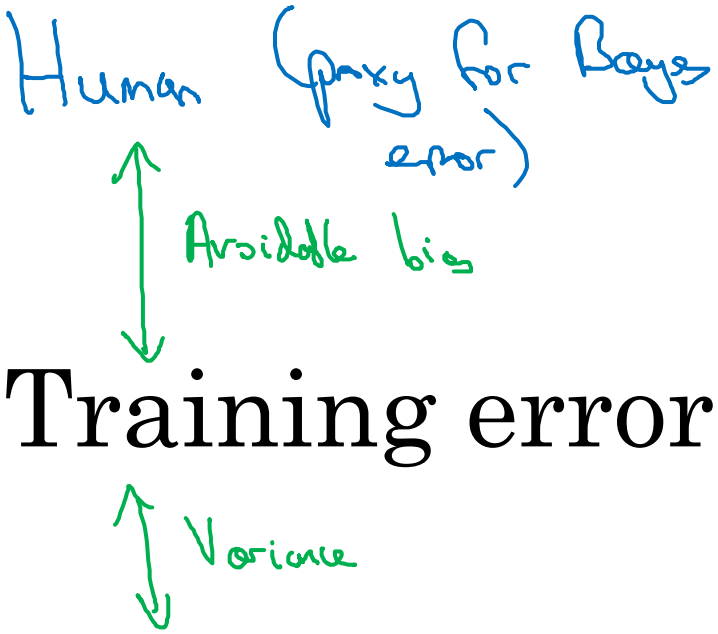
(c) Experienced doctor ..... 0.7 % error

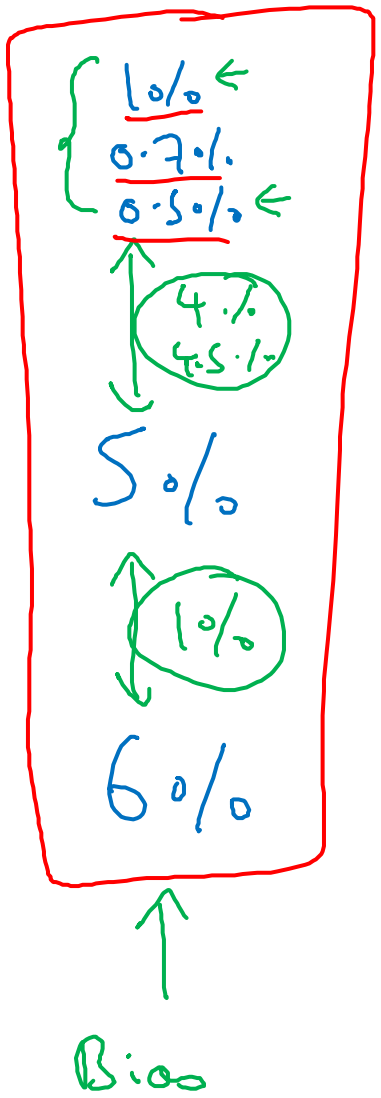
(d) Team of experienced doctors .. 0.5 % error

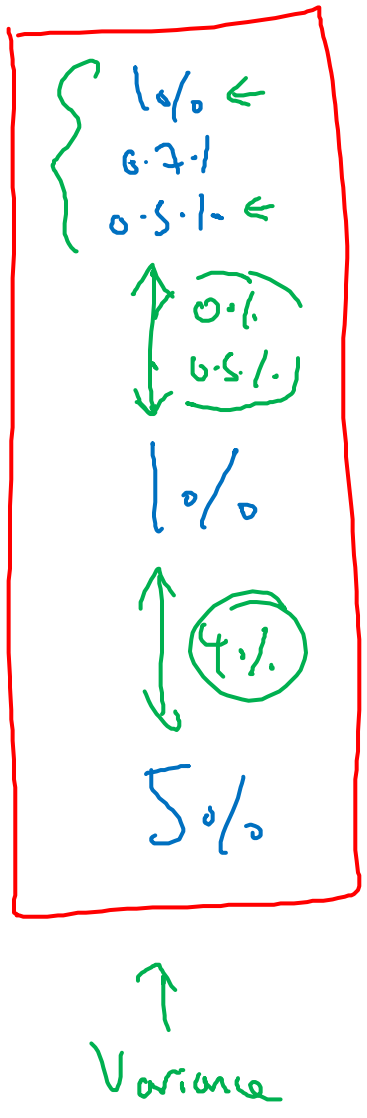
Bayes error $\leq$ 0.5%

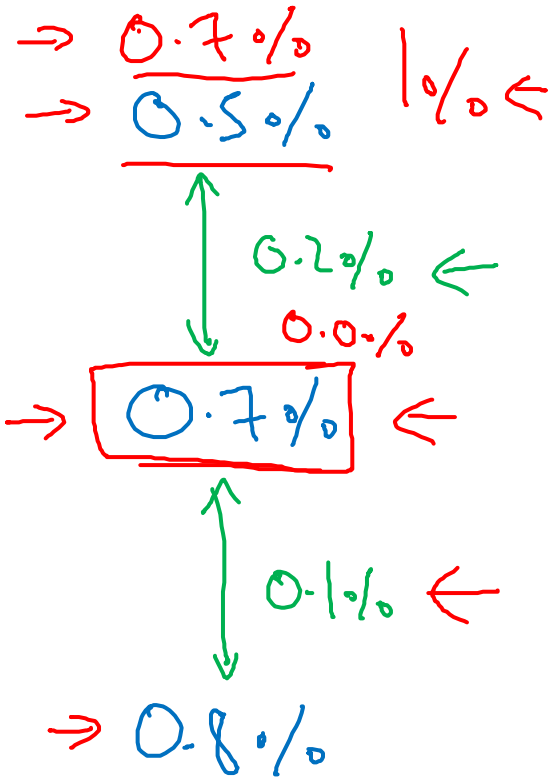
What is "human-level" error?

Error analysis example









Summary of bias/variance with human-level performance

Human-level error

(proxy for Bayes error)

Training error

Dev error

"Avoidable bias"

"Variance"

Understanding human-level performance

Human-level error gives an estimate of Bayes error.

Example 1: Medical image classification

This is an example of a medical image classification in which the input is a radiology image and the output is a diagnosis classification decision.

	Classification error (%)
Typical human	3.0
Typical doctor	1.0
Experienced doctor	0.7
Team of experienced doctors	0.5

The definition of human-level error depends on the purpose of the analysis, in this case, by definition the Bayes error is lower or equal to 0.5%.

Example 2: Error analysis

	Classification error (%)		
	Scenario A	Scenario B	Scenario C
Human (proxy for Bayes error)	1	1	0.5
	0.7	0.7	
	0.5	0.5	
Training error	5	1	0.7
Development error	6	5	0.8

Scenario A

In this case, the choice of human-level performance doesn't have an impact. The avoidable bias is between 4%-4.5% and the variance is 1%. Therefore, the focus should be on bias reduction technique.

Scenario B

In this case, the choice of human-level performance doesn't have an impact. The avoidable bias is between 0%-0.5% and the variance is 4%. Therefore, the focus should be on variance reduction technique.

Scenario C

In this case, the estimate for Bayes error has to be 0.5% since you can't go lower than the human-level performance otherwise the training set is overfitting. Also, the avoidable bias is 0.2% and the variance is 0.1%. Therefore, the focus should be on bias reduction technique.

Summary of bias/variance with human-level performance

- Human - level error – proxy for Bayes error
- If the difference between human-level error and the training error is bigger than the difference between the training error and the development error. The focus should be on bias reduction technique
- If the difference between training error and the development error is bigger than the difference between the human-level error and the training error. The focus should be on variance reduction technique

deeplearning.ai

Comparing to human-level performance

Surpassing human-level performance

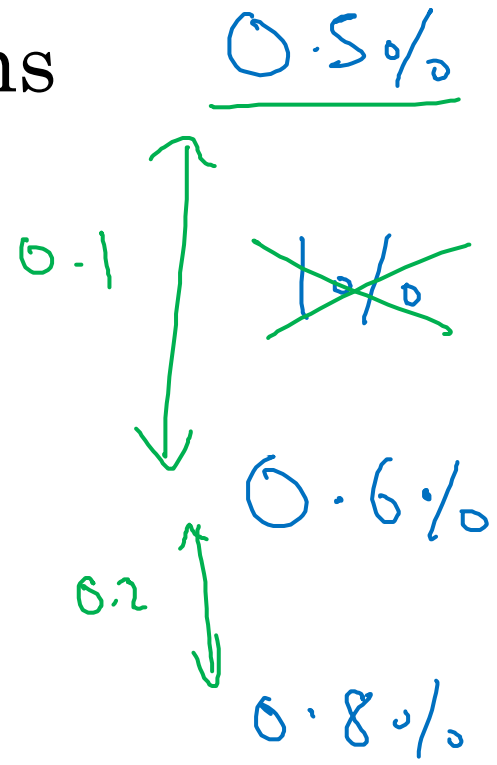
Surpassing human-level performance

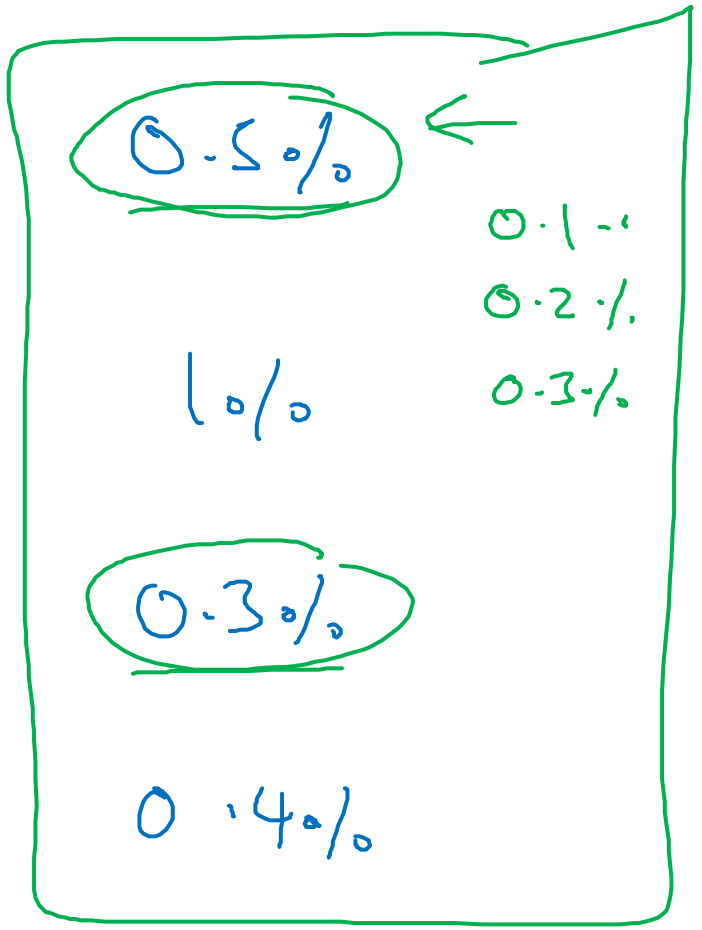
Team of humans

One human

Training error

Dev error





What is avoidable bias?

Problems where ML significantly surpasses human-level performance

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

Structural data

Not natural perception

Lots of data

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

Surpassing human-level performance

Example1: Classification task

	Classification error (%)	
	Scenario A	Scenario B
Team of humans	0.5	0.5
One human	1.0	1
Training error	0.6	0.3
Development error	0.8	0.4

Scenario A

In this case, the Bayes error is 0.5%, therefore the available bias is 0.1% et the variance is 0.2%.

Scenario B

In this case, there is not enough information to know if bias reduction or variance reduction has to be done on the algorithm. It doesn't mean that the model cannot be improve, it means that the conventional ways to know if bias reduction or variance reduction are not working in this case.

There are many problems where machine learning significantly surpasses human-level performance, especially with structured data:

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

deeplearning.ai

Comparing to human-level performance

Improving your model performance

The two fundamental assumptions of supervised learning

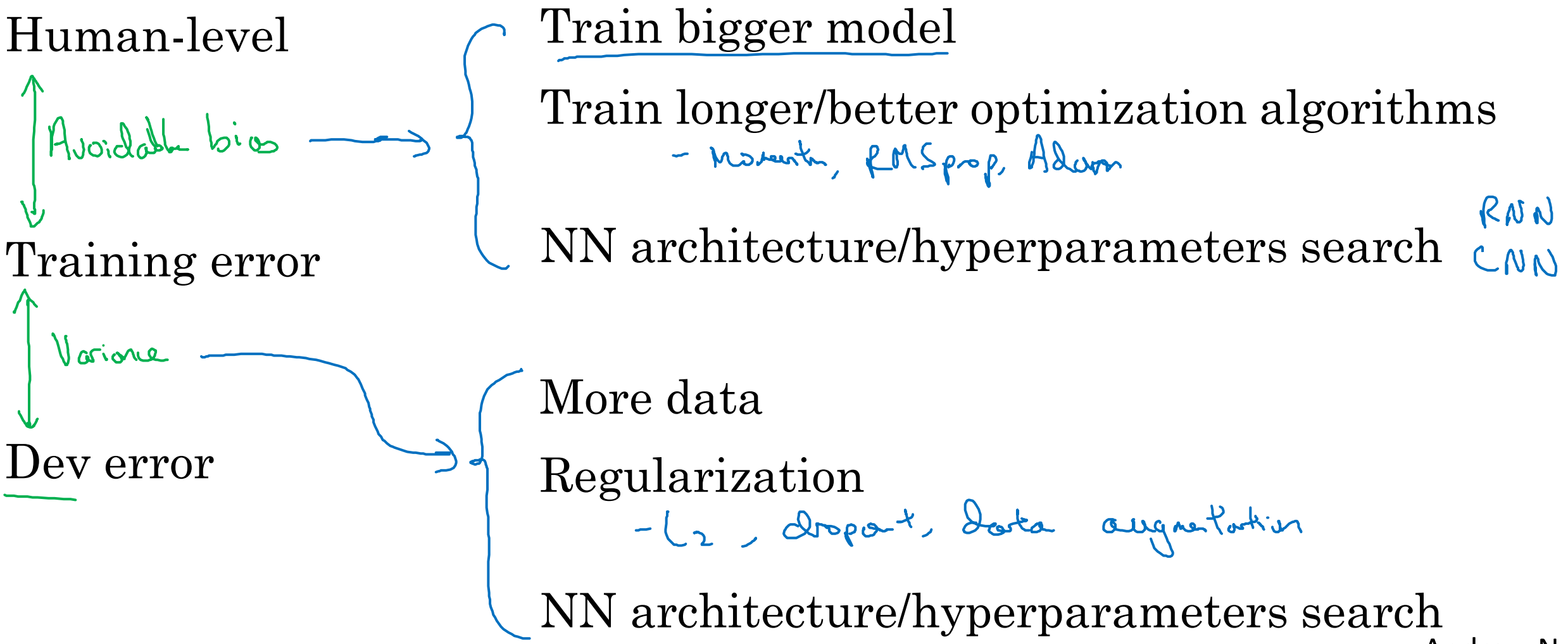
1. You can fit the training set pretty well.

~ Avoidable bias

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

~ Variance

Reducing (avoidable) bias and variance



Improving your model performance

The two fundamental assumptions of supervised learning

There are 2 fundamental assumptions of supervised learning. The first one is to have a low avoidable bias which means that the training set fits well. The second one is to have a low or acceptable variance which means that the training set performance generalizes well to the development set and test set.

If the difference between human-level error and the training error is bigger than the difference between the training error and the development error, the focus should be on bias reduction technique which are training a bigger model, training longer or change the neural networks architecture or try various hyperparameters search.

If the difference between training error and the development error is bigger than the difference between the human-level error and the training error, the focus should be on variance reduction technique which are bigger data set, regularization or change the neural networks architecture or try various hyperparameters search.

Summary