Advice for applying machine learning

1. Deciding what to try next

Debugging a learning algorithm

You've implemented regularized linear regression on housing prices

$$J(\vec{\mathrm{w}},b) = \frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{\mathrm{w}},b}(\vec{\mathrm{x}}^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

But it makes unacceptably large errors in predictions. What do you try next?

- Get more training examples
- Try smaller sets of features
- Try getting additional features
- Try adding polynomial features ($x_1^2, x_2^2, x_1 x_2, etc$)
- Try decreasing λ
- Try increasing λ

Machine learning diagnostic

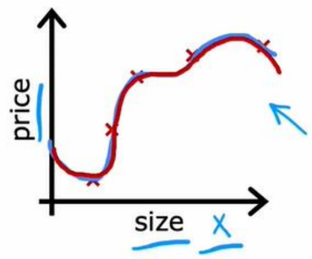
Diagnostic:

A test that you run to gain insight into what is/isn't working with a learning algorithm, to gain guidance into improving its performance.

Diagnostics can take time to implement but doing so can be a very good use of your time.

2. Evaluating a model

Evaluating your model



Model fits the training data well but will fail to generalize to new examples not in the training set.

$x = x_1$

$w_4 x^4$

$$f_{\vec{\mathrm{w}},b}(\vec{\mathrm{x}}) = w_1 x + w_2 x^2 + \cdots + w_n x^n + b$$

just x

- x_1 = size in feet²
- x_2 = no. of bedrooms
- x_3 = no. of floors
- x_4 = age of home in years

$f(\vec{x})$

Evaluating your model

Dataset:

	size	price
70%	2104	400
	1600	330
	2400	369
	1416	232
	3000	540
	1985	300
	1534	315
30%	1427	199
	1380	212
	1494	243

training set → $(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)})$, $\vdots$, $(x^{(m_{train})}, y^{(m_{train})})$

m_{train} = no. training examples $= 7$

test set → $(x_{test}^{(1)}, y_{test}^{(1)})$, $\vdots$, $(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

m_{test} = no. test examples $= 3$

Train/test procedure for linear regression (with squared error cost)

Fit parameters by minimizing cost function $J(\vec{w}, b)$

$$\rightarrow J(\vec{w}, b) = \left[\frac{1}{2m_{train}} \sum_{i=1}^{m_{train}} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m_{train}} \sum_{j=1}^{n} w_j^2 \right]$$

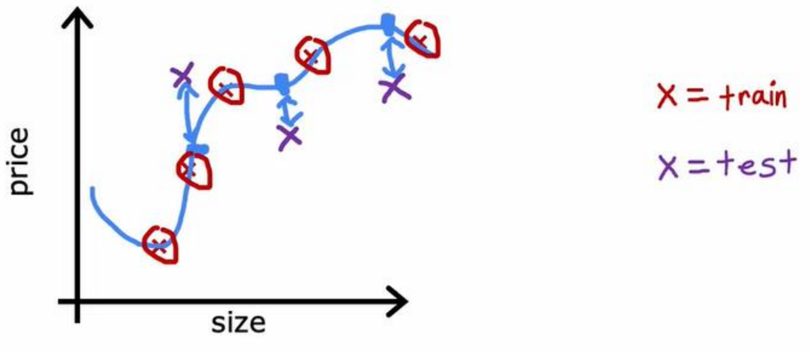
Compute test error:

$$J_{test}(\vec{w}, b) = \frac{1}{2m_{test}} \left[\sum_{i=1}^{m_{test}} \left(f_{\vec{w},b}\left(\vec{x}_{test}^{(i)}\right) - y_{test}^{(i)} \right)^2 \right] \quad \cancel{\sum_{j=1}^{n} w_j^2}$$

Compute training error:

$$J_{train}(\vec{w}, b) = \frac{1}{2m_{train}} \left[\sum_{i=1}^{m_{train}} \left(f_{\vec{w},b}\left(\vec{x}_{train}^{(i)}\right) - y_{train}^{(i)} \right)^2 \right]$$

Train/test procedure for linear regression (with squared error cost)



$J_{train}(\vec{w}, b)$ will be low

$J_{test}(\vec{w}, b)$ will be high ←

Train/test procedure for classification problem

0/1

Fit parameters by minimizing $J(\vec{w}, b)$ to find $\vec{w}, b$

E.g.,

$$J(\vec{w},b) = -\frac{1}{m_{train}} \sum_{i=1}^{m_{train}} \left[y^{(i)} \log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1-y^{(i)}) \log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right) \right] + \frac{\lambda}{2m_{train}} \sum_{j=1}^{n} w_j^2$$

Compute test error:

$$J_{test}(\vec{w},b) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} \left[y_{test}^{(i)} \log\left(f_{\vec{w},b}\left(\vec{x}_{test}^{(i)}\right)\right) + \left(1 - y_{test}^{(i)}\right) \log\left(1 - f_{\vec{w},b}\left(\vec{x}_{test}^{(i)}\right)\right) \right]$$

Compute train error:

$$J_{train}(\vec{w},b) = -\frac{1}{m_{train}} \sum_{i=1}^{m_{train}} \left[y_{train}^{(i)} \log\left(f_{\vec{w},b}\left(\vec{x}_{train}^{(i)}\right)\right) + \left(1 - y_{train}^{(i)}\right) \log\left(1 - f_{\vec{w},b}\left(\vec{x}_{train}^{(i)}\right)\right) \right]$$

Train/test procedure for classification problem

fraction of the test set and the fraction of the train set that the algorithm has misclassified.

$$\hat{y} = \begin{cases} 1 \text{ if } f_{\vec{w},b}(\vec{x}^{(i)}) \geq 0.5 \\ 0 \text{ if } f_{\vec{w},b}(\vec{x}^{(i)}) < 0.5 \end{cases}$$

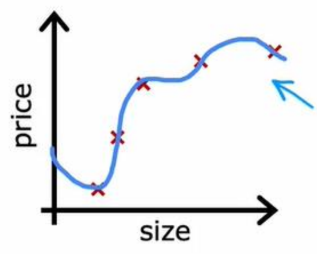
count $\hat{y} \neq y$

$J_{test}(\vec{w}, b)$ is the fraction of the test set that has been misclassified.

$J_{train}(\vec{w}, b)$ is the fraction of the train set that has been misclassified.

3. Model selection and training/cross validation/test sets

Model selection (choosing a model)



$x = x_1$

$f_{\vec{w},b}(\vec{x}) = w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$

Once parameters $\vec{w}, b$ are fit to the training set, the training error $J_{train}(\vec{w}, b)$ is likely lower than the actual generalization error.

$J_{test}(\vec{w}, b)$ is better estimate of how well the model will generalize to new data compared to $J_{train}(\vec{w}, b)$.

Model selection (choosing a model)

d=1 1. $f_{\vec{w},b}(\vec{x}) = w_1x + b$ $\rightarrow w^{<1>}, b^{<1>} \rightarrow J_{test}(w^{<1>}, b^{<1>})$

d=2 2. $f_{\vec{w},b}(\vec{x}) = w_1x + w_2x^2 + b$ $\rightarrow w^{<2>}, b^{<2>} \rightarrow J_{test}(w^{<2>}, b^{<2>})$

d=3 3. $f_{\vec{w},b}(\vec{x}) = w_1x + w_2x^2 + w_3x^3 + b$ $\rightarrow w^{<3>}, b^{<3>} \rightarrow J_{test}(w^{<3>}, b^{<3>})$

⋮

d=10 10. $f_{\vec{w},b}(\vec{x}) = w_1x + w_2x^2 + \cdots + w_{10}x^{10} + b$ $\rightarrow J_{test}(w^{<10>}, b^{<10>})$

Choose $w_1x + \cdots + w_5x^5 + b$ d=5 $J_{test}(w^{<5>}, b^{<5>})$

How well does the model perform? Report test set error $J_{test}(w^{<5>}, b^{<5>})$?
The problem: $J_{test}(w^{<5>}, b^{<5>})$ is likely to be an optimistic estimate of generalization error (ie. $J_{test}(w^{<5>}, b^{<5>}) <$ generalization error). Because an extra parameter d (degree of polynomial) was chosen using the test set.

w, b are overly optimistic estimate of generalization error on training data.

Training/cross validation/test set

validation set
development set
$\rightarrow$ dev set

size	price
2104	400
1600	330
2400	369
1416	232
3000	540
1985	300
1534	315
1427	199
1380	212
1494	243

training set 60% $\rightarrow$ $(x^{(1)}, y^{(1)})$ ⋮ $(x^{(m_{train})}, y^{(m_{train})})$ $m_{train} = 6$

cross validation 20% $\rightarrow$ $(x_{cv}^{(1)}, y_{cv}^{(1)})$ ⋮ $(x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$ $m_{cv} = 2$

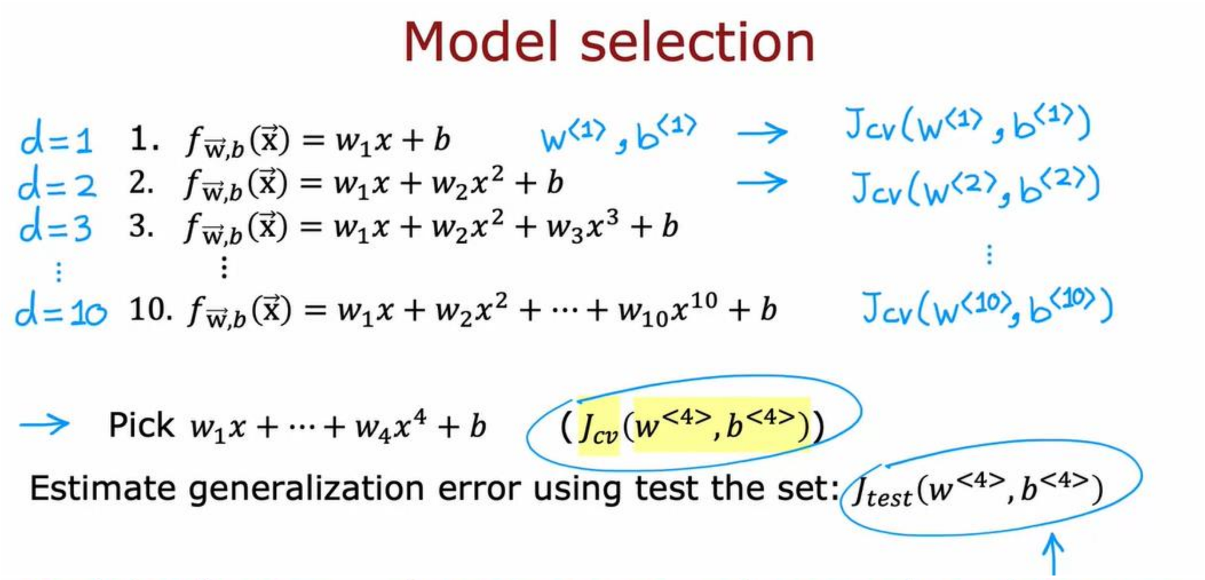
test set 20% $\rightarrow$ $(x_{test}^{(1)}, y_{test}^{(1)})$ ⋮ $(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$ $m_{test} = 2$

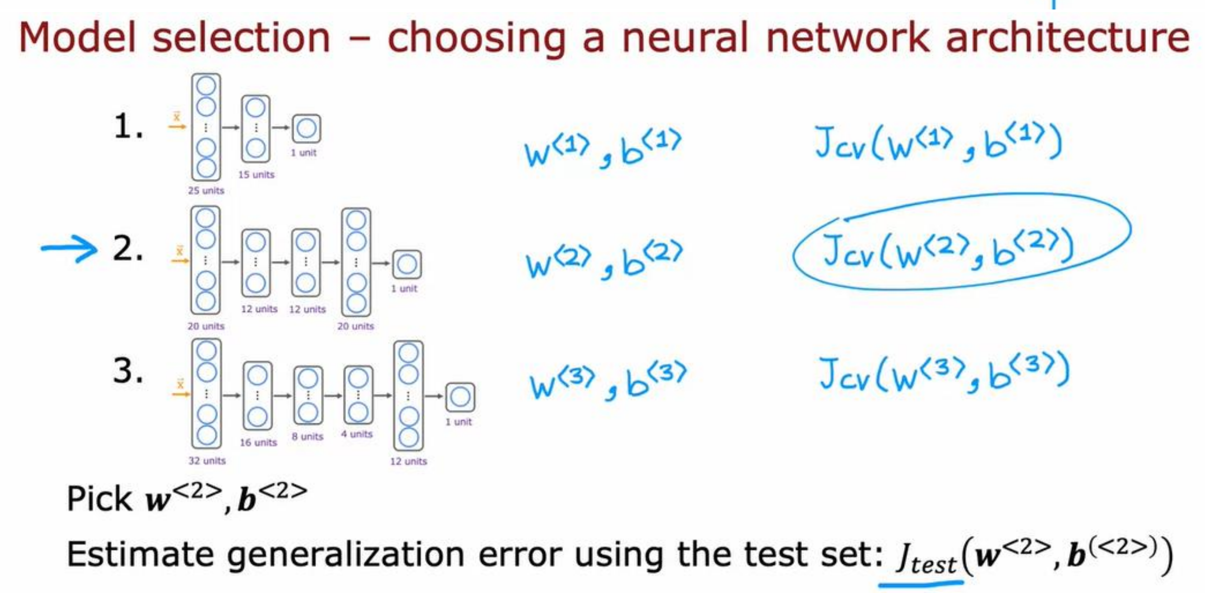
Training/cross validation/test set

Training error: $$J_{train}(\vec{w}, b) = \frac{1}{2m_{train}}\left[\sum_{i=1}^{m_{train}} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2\right]$$

Cross validation error: $$J_{cv}(\vec{w}, b) = \frac{1}{2m_{cv}}\left[\sum_{i=1}^{m_{cv}} \left(f_{\vec{w},b}\left(\vec{x}_{cv}^{(i)}\right) - y_{cv}^{(i)}\right)^2\right]$$ (validation error, dev error)

Test error: $$J_{test}(\vec{w}, b) = \frac{1}{2m_{test}}\left[\sum_{i=1}^{m_{test}} \left(f_{\vec{w},b}\left(\vec{x}_{test}^{(i)}\right) - y_{test}^{(i)}\right)^2\right]$$





4. Practice quiz

1.

In the context of machine learning, what is a diagnostic?

- ○ This refers to the process of measuring how well a learning algorithm does on a test set (data that the algorithm was not trained on).
- ○ An application of machine learning to medical applications, with the goal of diagnosing patients' conditions.
- ◉ A test that you run to gain insight into what is/isn't working with a learning algorithm.
- ○ A process by which we quickly try as many different ways to improve an algorithm as possible, so as to see what works.

Correct
Yes! A diagnostic is a test that you run to gain insight into what is/isn't working with a learning algorithm, to gain guidance into improving its performance.

2.

True/False? It is always true that the better an algorithm does on the training set, the better it will do on generalizing to new data.

- (•) False
- ( ) True

Correct
Actually, if a model overfits the training set, it may not generalize well to new data.

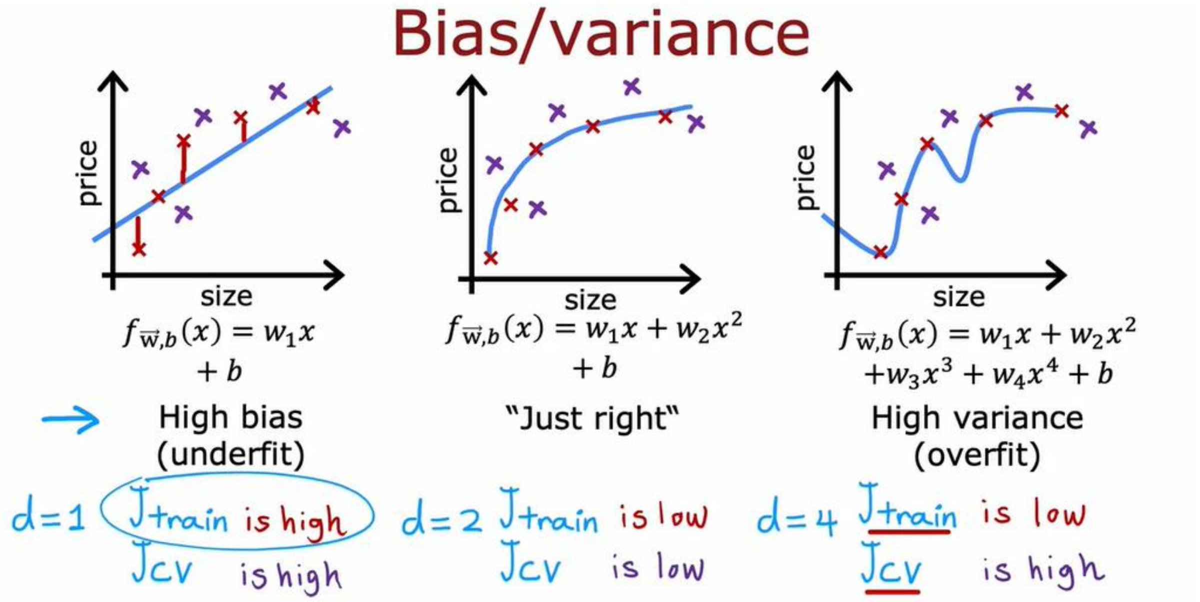
For a classification task; suppose you train three different models using three different neural network architectures. Which data do you use to evaluate the three models in order to choose the best one?

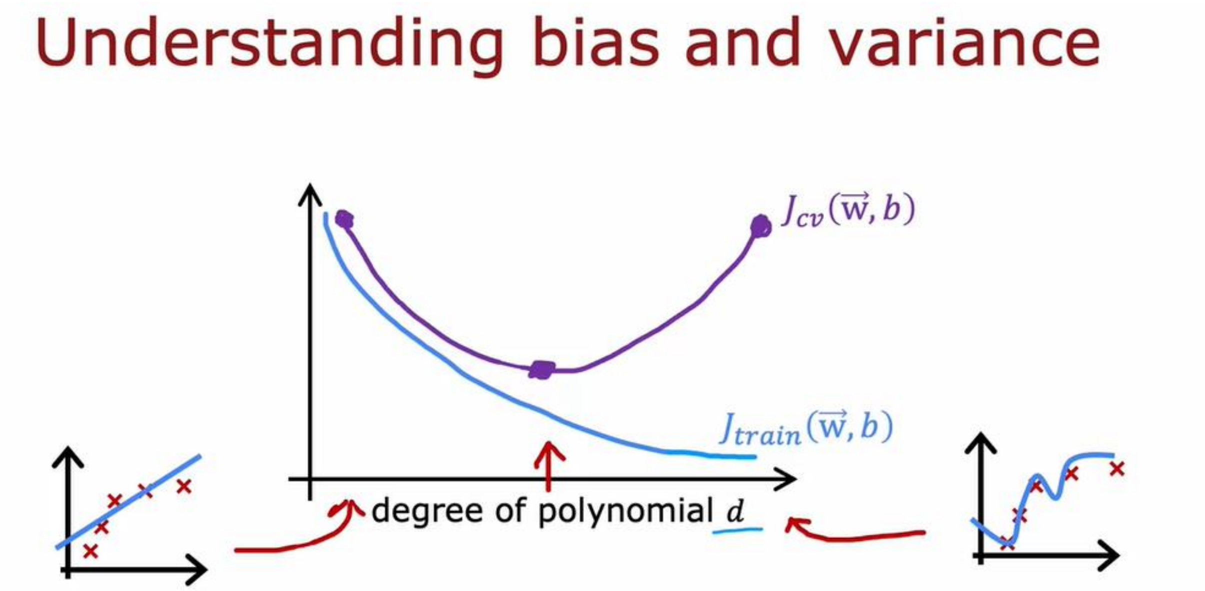
- ( ) All the data -- training, cross validation and test sets put together.
- ( ) The training set
- ( ) The test set
- (•) The cross validation set

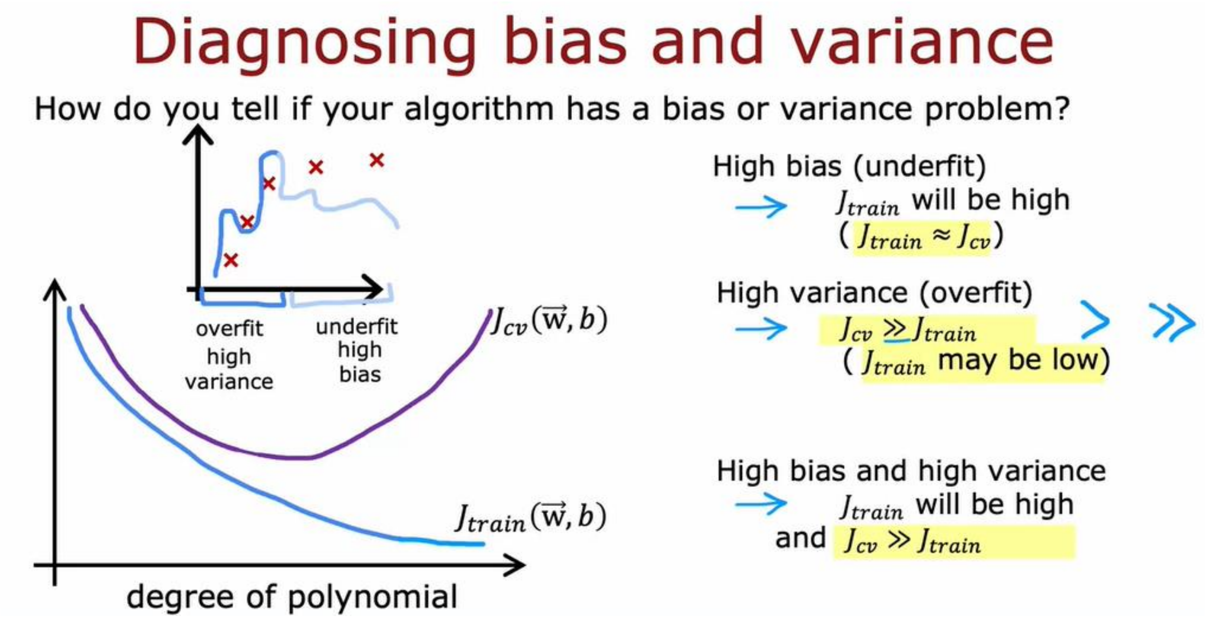
Correct
Correct. Use the cross validation set to calculate the cross validation error on all three models in order to compare which of the three models is best.

Bias and variance

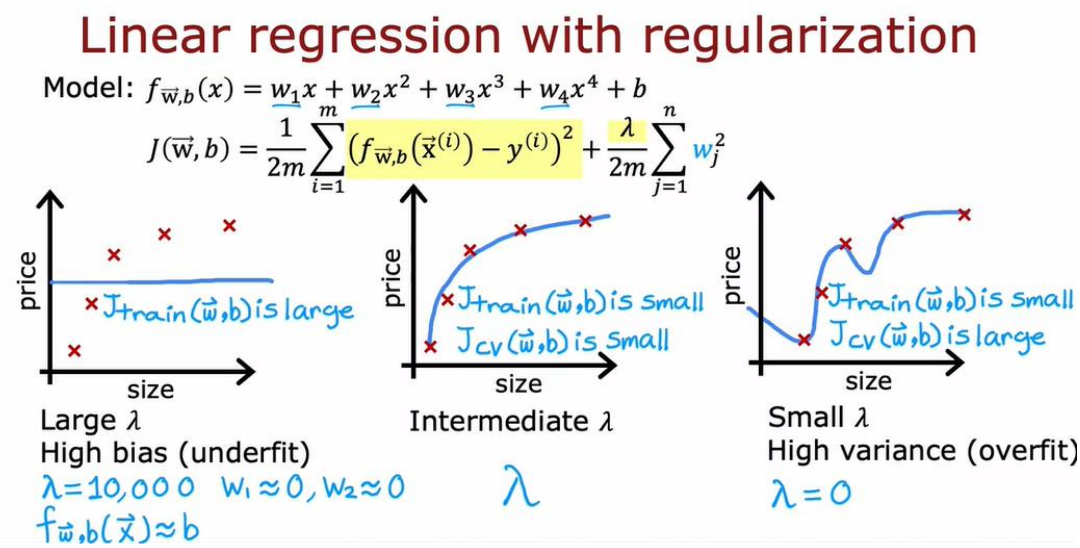
1. Diagnosing bias and variance

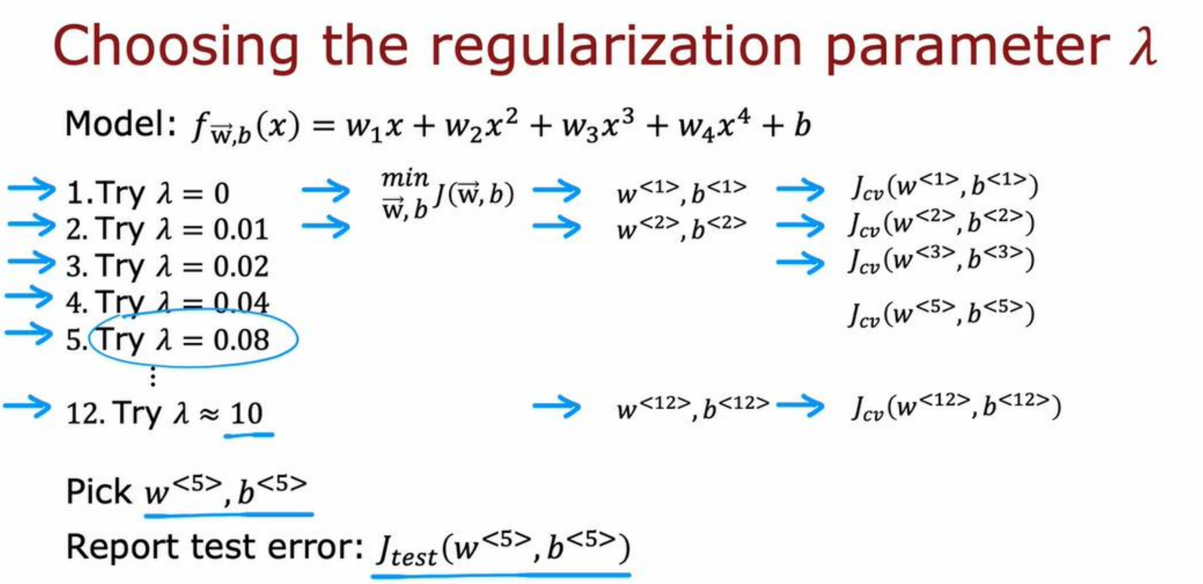


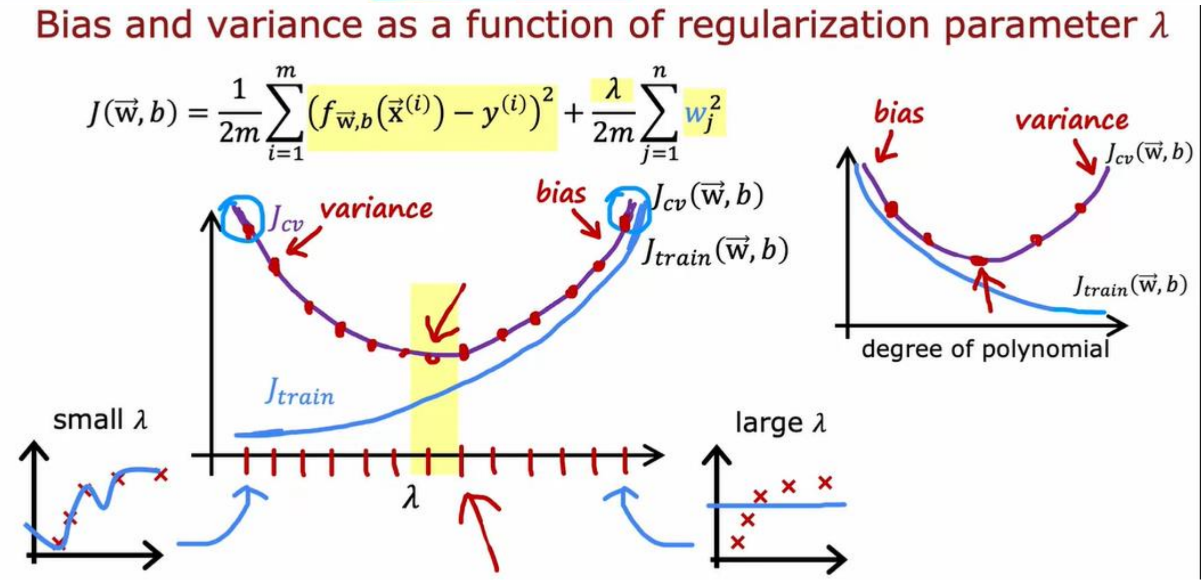




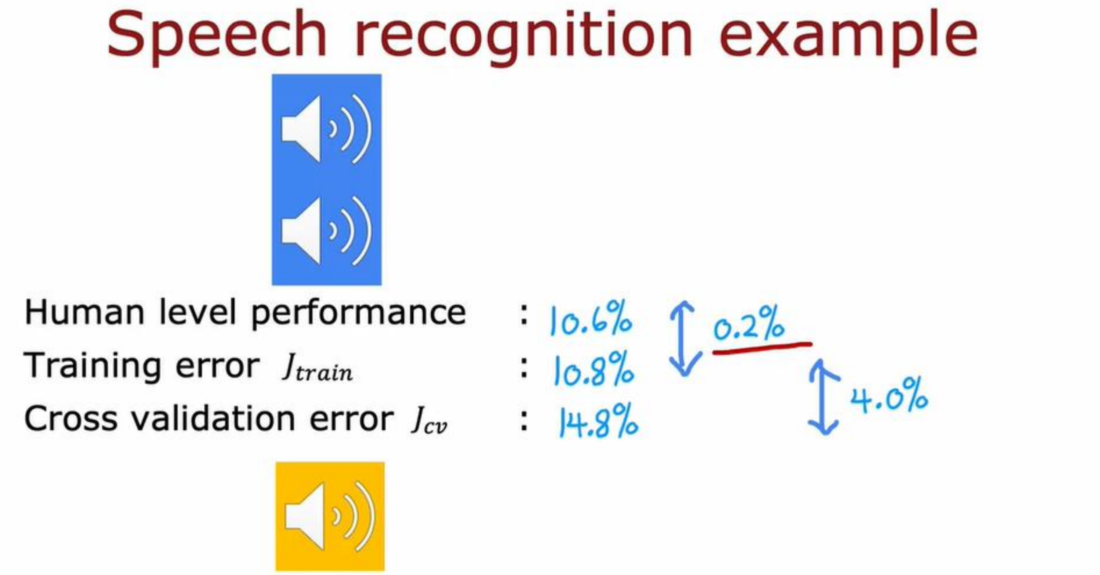
2. Regularization and bias/variance







3. Establishing a baseline level of performance



Establishing a baseline level of performance

What is the level of error you can reasonably hope to get to?

- Human level performance
- Competing algorithms performance
- Guess based on experience

Bias/variance examples

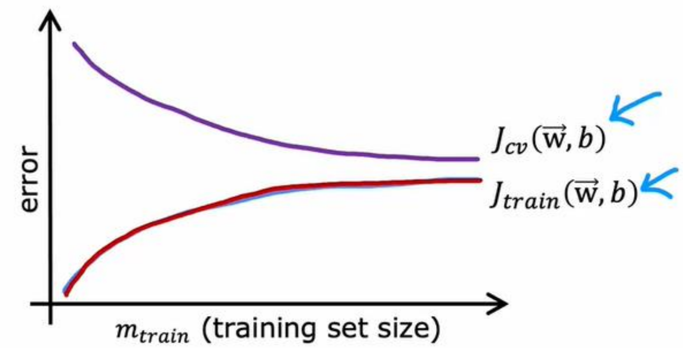
Baseline performance	10.6%	10.6%	10.6%
	0.2%	4.4%	4.4%
Training error (J_{train})	10.8%	15.0%	15.0%
	4.0%	0.5%	4.7%
Cross validation error (J_{cv})	14.8%	15.5%	19.7%
	high variance	high bias	high bias high variance

4. Learning curves

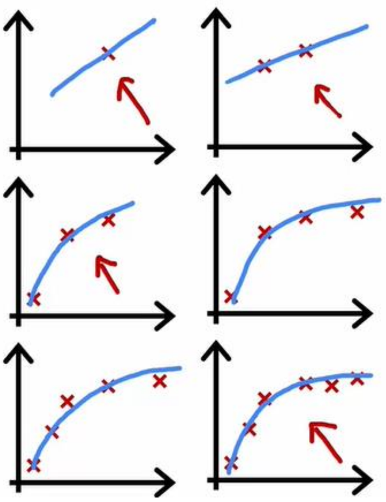
Learning curves

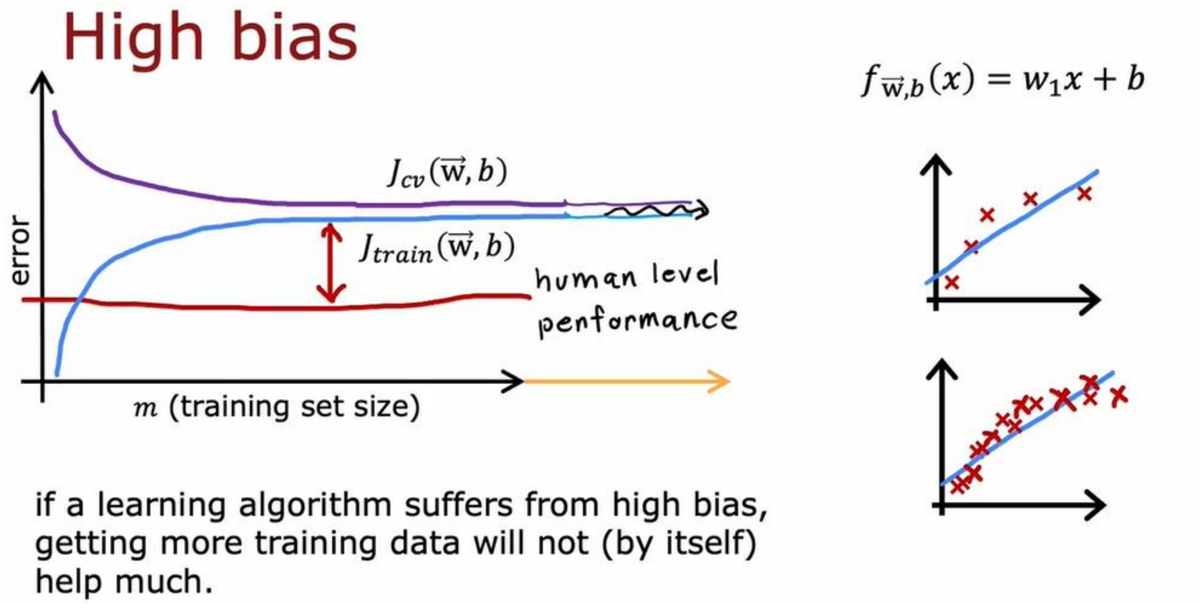
J_{train} = training error

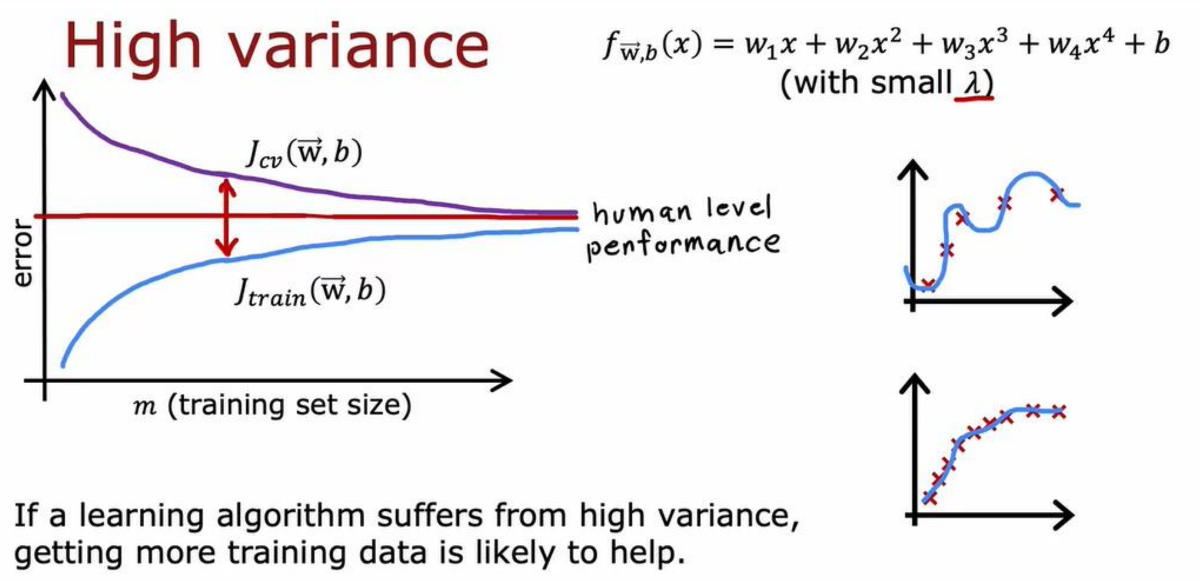
J_{cv} = cross validation error



$f_{\vec{w},b}(x) = w_1 x + w_2 x^2 + b$







5. Deciding what next revisited

Debugging a learning algorithm

You've implemented regularized linear regression on housing prices

$$J(\vec{w}, b) = \frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

But it makes unacceptably large errors in predictions. What do you try next?

- → Get more training examples — fixes high variance
- → Try smaller sets of features $x, x^2, \cancel{x^3}, \cancel{x^4}, \cancel{x^5} \ldots$ — fixes high variance
- → Try getting additional features — fixes high bias
- → Try adding polynomial features $(x_1^2, x_2^2, x_1x_2, etc)$ — fixes high bias
- → Try decreasing λ — fixes high bias
- → Try increasing λ — fixes high variance

The bias variance tradeoff

$f_{\vec{w},b}(x) = w_1 x + b$

$f_{\vec{w},b}(x) = w_1 x + w_2 x^2 + b$

$f_{\vec{w},b}(x) = w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$

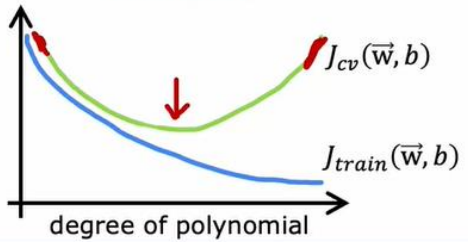
Simple model

High bias

Complex model

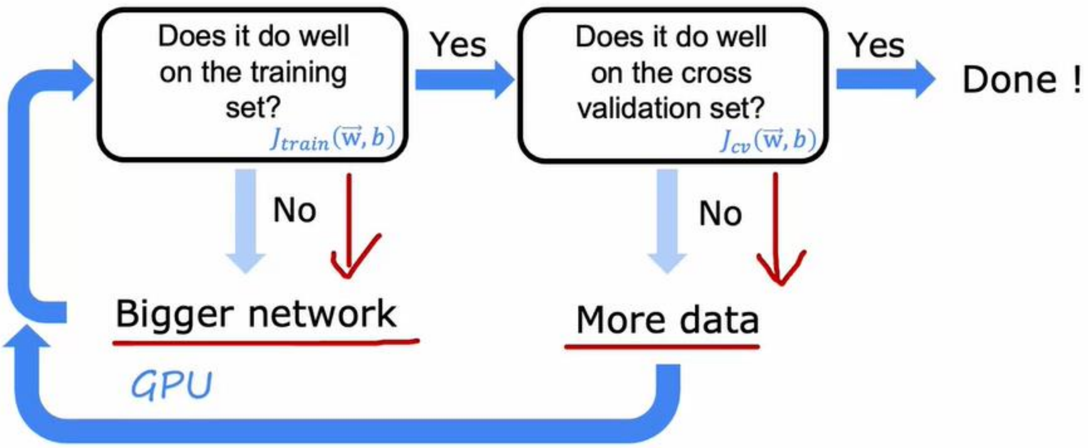
High variance

tradeoff

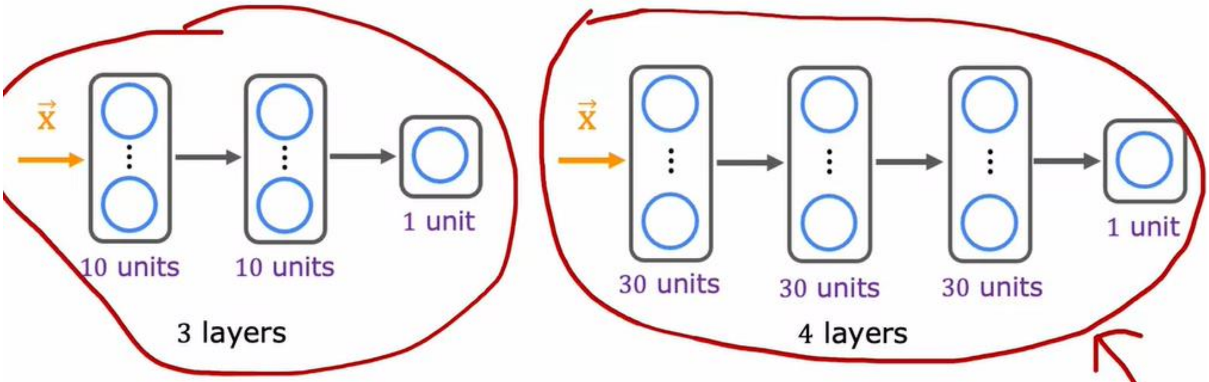


Neural networks and bias variance

Large neural networks are low bias machines



Neural networks and regularization



A large neural network will usually do as well or better than a smaller one so long as regularization is chosen appropriately.

Neural network regularization

$$\underline{J(\mathbf{W},\mathbf{B})} = \underline{\frac{1}{m}\sum_{i=1}^{m} L(f(\vec{\mathrm{x}}^{(i)}), y^{(i)})} + \underbrace{\frac{\lambda}{2m}\sum_{\textit{all weights } \mathbf{W}}(w^2)}$$

b

Unregularized MNIST model

```
layer_1 = Dense(units=25, activation="relu")
layer_2 = Dense(units=15, activation="relu")
layer_3 = Dense(units=1, activation="sigmoid")
model = Sequential([layer_1, layer_2, layer_3])
```

λ

Regularized MNIST model

```
layer_1 = Dense(units=25, activation="relu", kernel_regularizer=L2(0.01))
layer_2 = Dense(units=15, activation="relu", kernel_regularizer=L2(0.01))
layer_3 = Dense(units=1, activation="sigmoid", kernel_regularizer=L2(0.01))
model = Sequential([layer_1, layer_2, layer_3])
```

6. Practice quiz

If the model's cross validation error J_{cv} is much higher than the training error J_{train}, this is an indication that the model has…

- (•) high variance
- ( ) Low variance
- ( ) high bias
- ( ) Low bias

Correct
When $J_{cv} >> J_{train}$ (whether J_{train} is also high or not, this is a sign that the model is overfitting to the training data and performing much worse on new examples.

Which of these is the best way to determine whether your model has high bias (has underfit the training data)?

- ( ) See if the cross validation error is high compared to the baseline level of performance
- (•) Compare the training error to the baseline level of performance
- ( ) See if the training error is high (above 15% or so)
- ( ) Compare the training error to the cross validation error.

Correct
Correct. If comparing your model's training error to a baseline level of performance (such as human level performance, or performance of other well-established models), if your model's training error is much higher, then this is a sign that the model has high bias (has underfit).

You find that your algorithm has high bias. Which of these seem like good options for improving the algorithm's performance? Hint: two of these are correct.

- [x] Decrease the regularization parameter λ (lambda)

Correct
Correct. Decreasing regularization can help the model better fit the training data.

- [x] Collect additional features or add polynomial features

Correct
Correct. More features could potentially help the model better fit the training examples.

- [ ] Remove examples from the training set
- [ ] Collect more training examples

You find that your algorithm has a training error of 2%, and a cross validation error of 20% (much higher than the training error). Based on the conclusion you would draw about whether the algorithm has a high bias or high variance problem, which of these seem like good options for improving the algorithm's performance? Hint: two of these are correct.

- [x] Increase the regularization parameter λ

Correct
Yes, the model appears to have high variance (overfit), and increasing regularization would help reduce high variance.

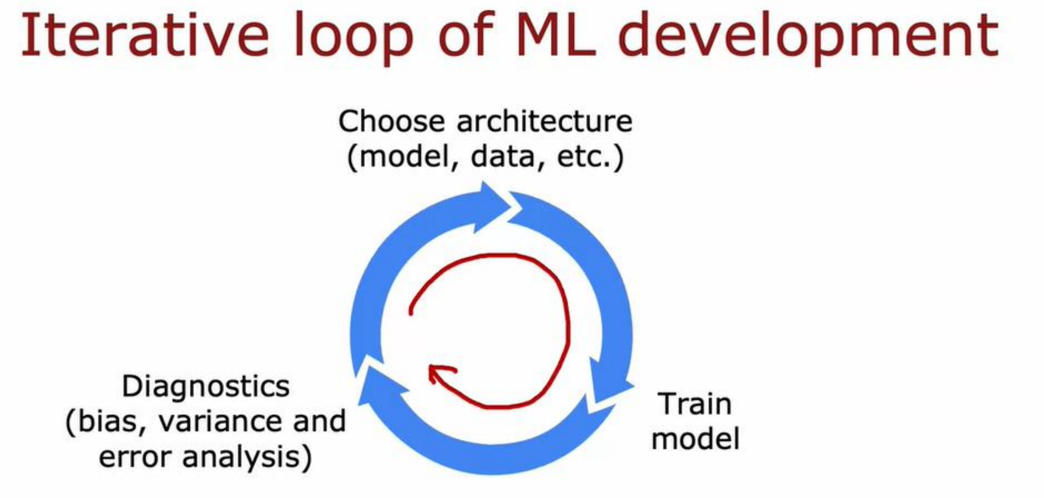
- [x] Collect more training data

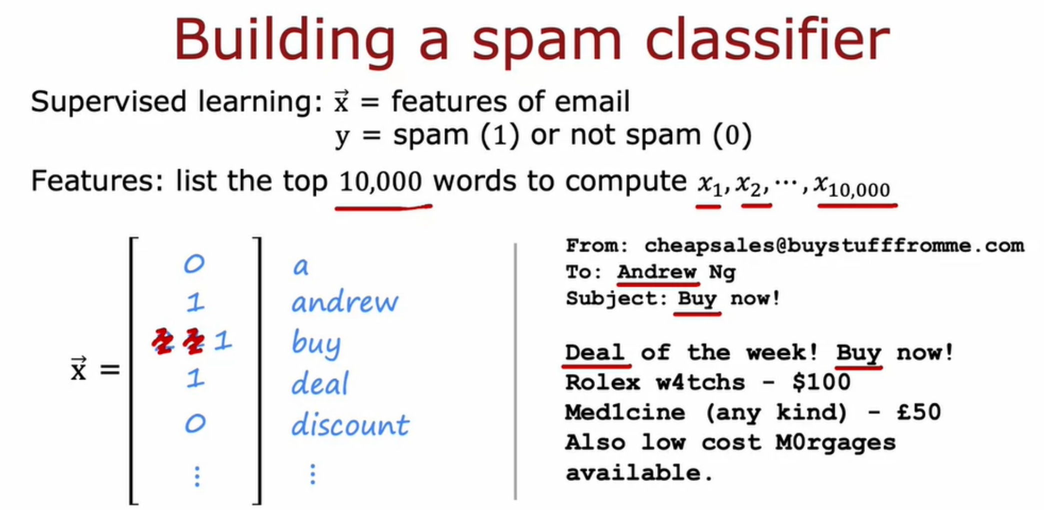
Correct
Yes, the model appears to have high variance (overfit), and collecting more training examples would help reduce high variance.

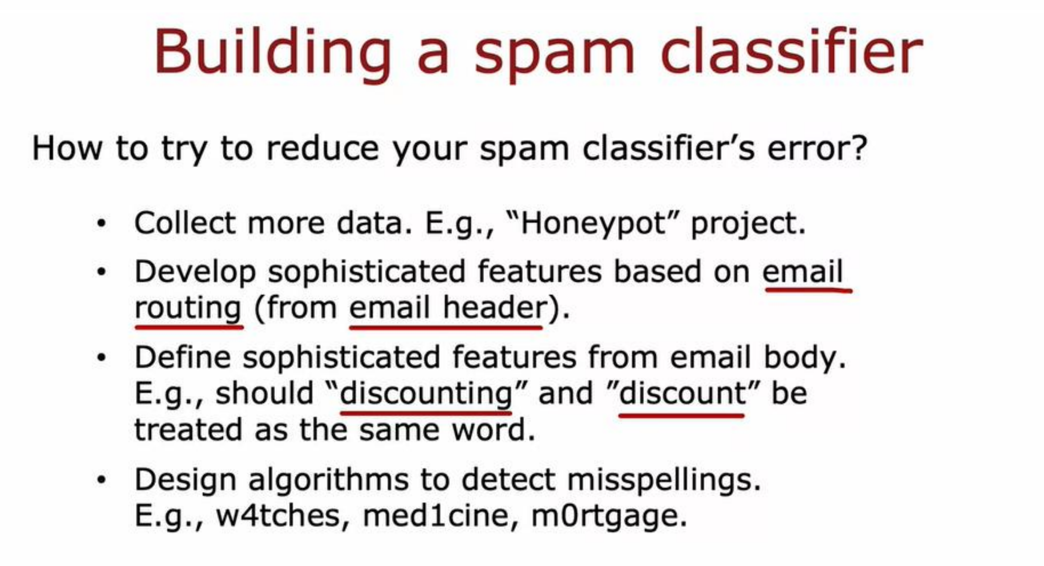
- [ ] Decrease the regularization parameter λ
- [ ] Reduce the training set size

Machine learning development process

1. Iterative loop of ML development







2. Error analysis

Error analysis

m_{cv} = ~~500~~ 5000 examples in cross validation set.

Algorithm misclassifies ~~100~~ 1000 of them.

Manually examine 100 examples and categorize them based on common traits.

- Pharma: 21 more data features
- Deliberate misspellings (w4tches, med1cine): 3
- Unusual email routing: 7
- Steal passwords (phishing): 18 more data features
- Spam message in embedded image: 5

3. Adding data

Adding data

Add more data of everything. E.g., "Honeypot" project.

Add more data of the types where error analysis has indicated it might help.

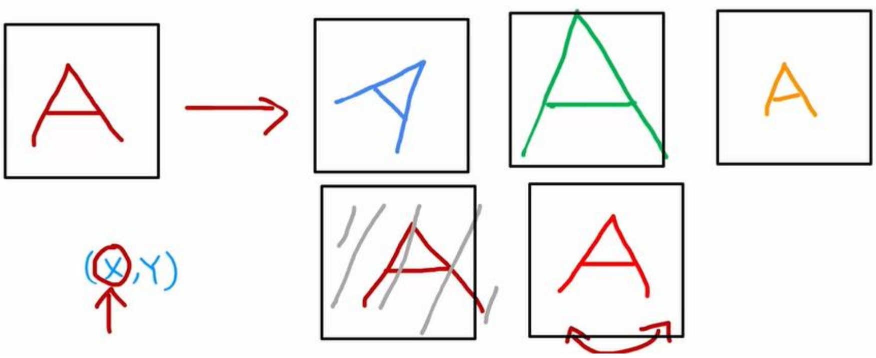
Pharma spam

E.g., Go to unlabeled data and find more examples of Pharma related spam.

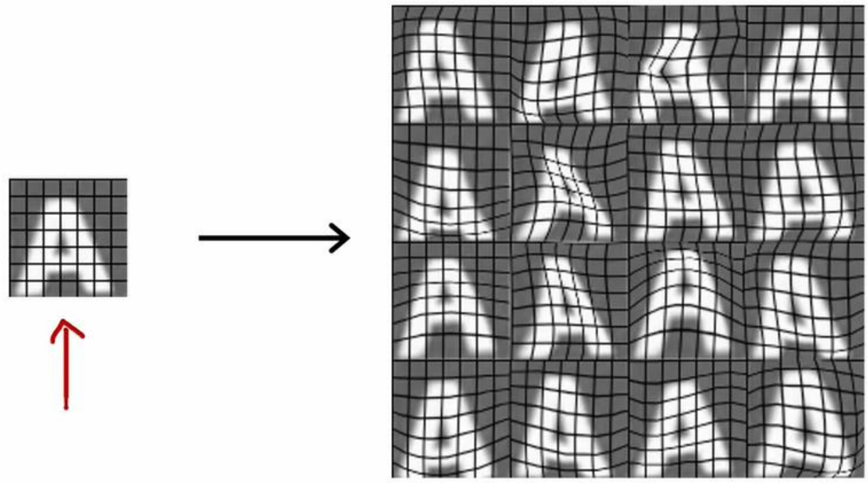
Beyond getting brand new training examples (x,y), another technique: Data augmentation

Data augmentation

Augmentation: modifying an existing training example to create a new training example.



Data augmentation by introducing distortions



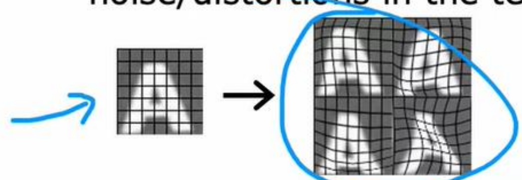
Data augmentation for speech

Speech recognition example

- Original audio (voice search: "What is today's weather?")
- + Noisy background: Crowd
- + Noisy background: Car
- + Audio on bad cellphone connection

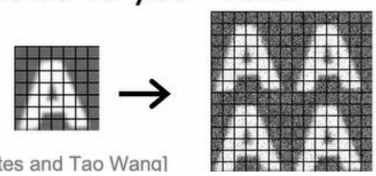
Data augmentation by introducing distortions

Distortion introduced should be representation of the type of noise/distortions in the test set.



Audio:
Background noise,
bad cellphone connection

Usually does not help to add purely random/meaningless noise to your data.



x_i=intensity (brightness) of pixel i
$x_i \leftarrow x_i$ + random noise

[Adam Coates and Tao Wang]

Artificial data synthesis for photo OCR

[http://www.publicdomainpictures.net/view-image.php?image=5745&picture=times-square]

Artificial data synthesis for photo OCR

Real data

Synthetic data

[Adam Coates and Tao Wang]

Engineering the data used by your system

Conventional model-centric approach:

AI = Code + Data

(algorithm/model)

work on this

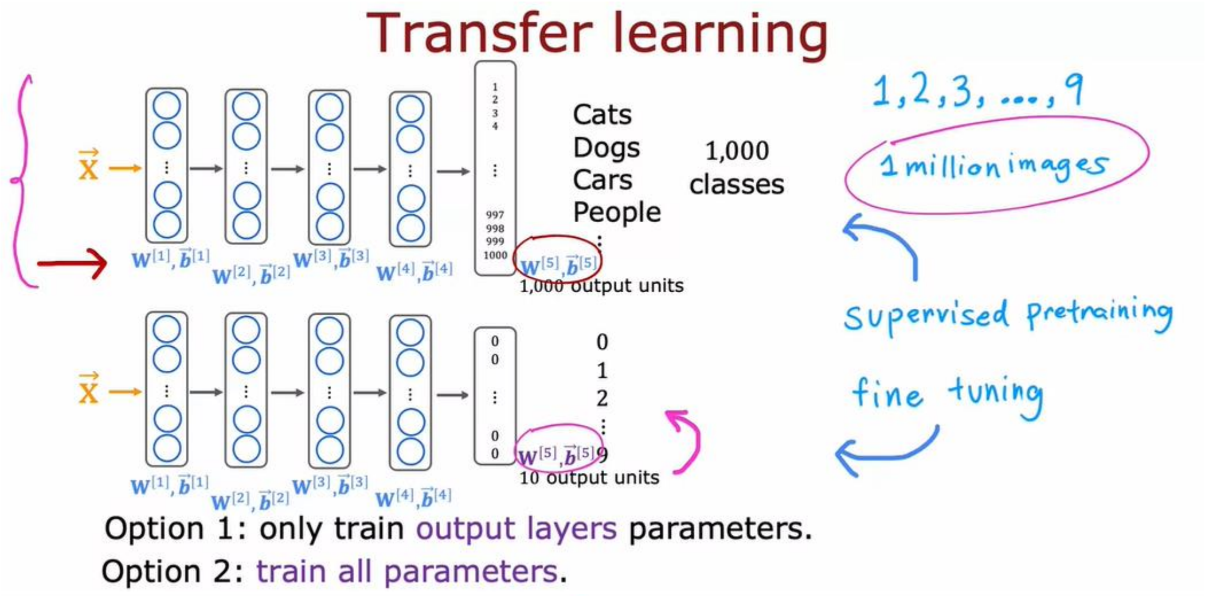
Data-centric approach:

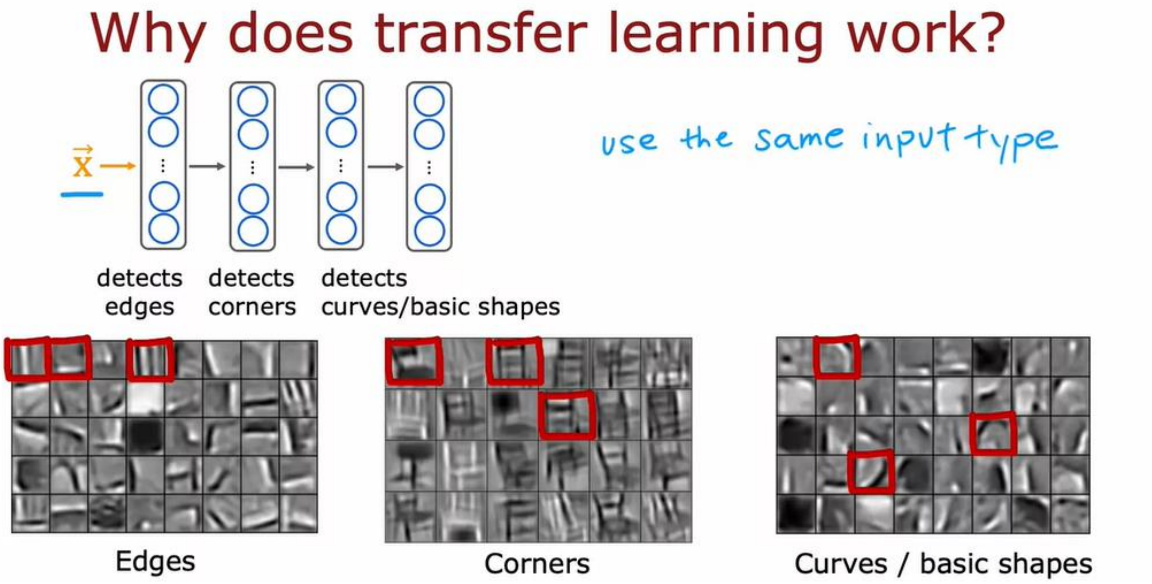
AI = Code + Data

(algorithm/model)

work on this

4. Transfer learning: using data from a different task





Transfer learning summary

1. Download neural network parameters pretrained on a large dataset with same input type (e.g., images, audio, text) as your application (or train your own).

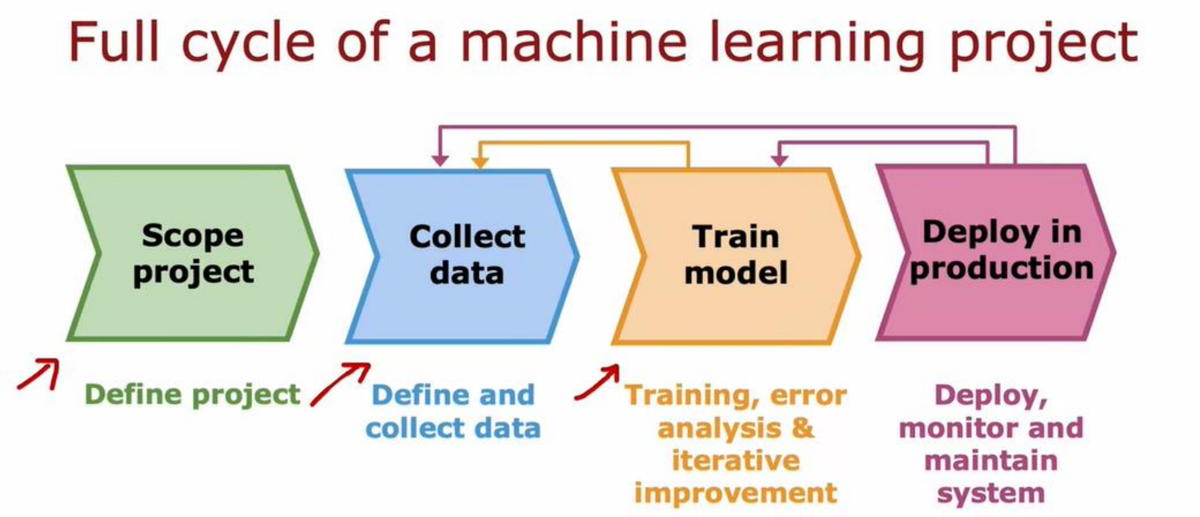
1 million images

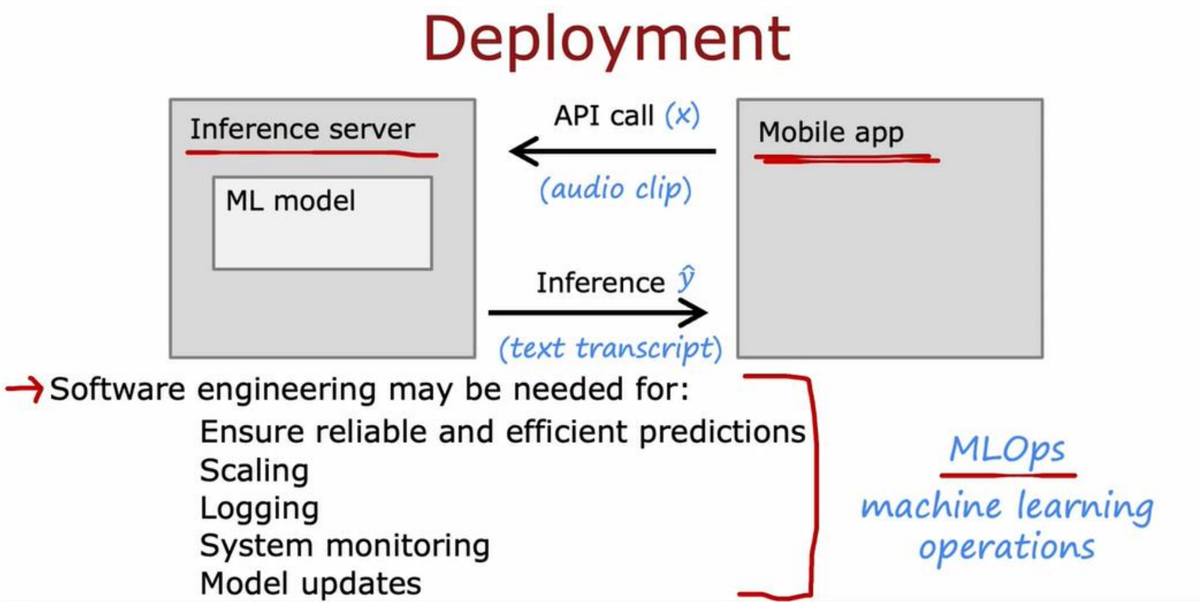
2. Further train (fine tune) the network on your own data.

1000 images

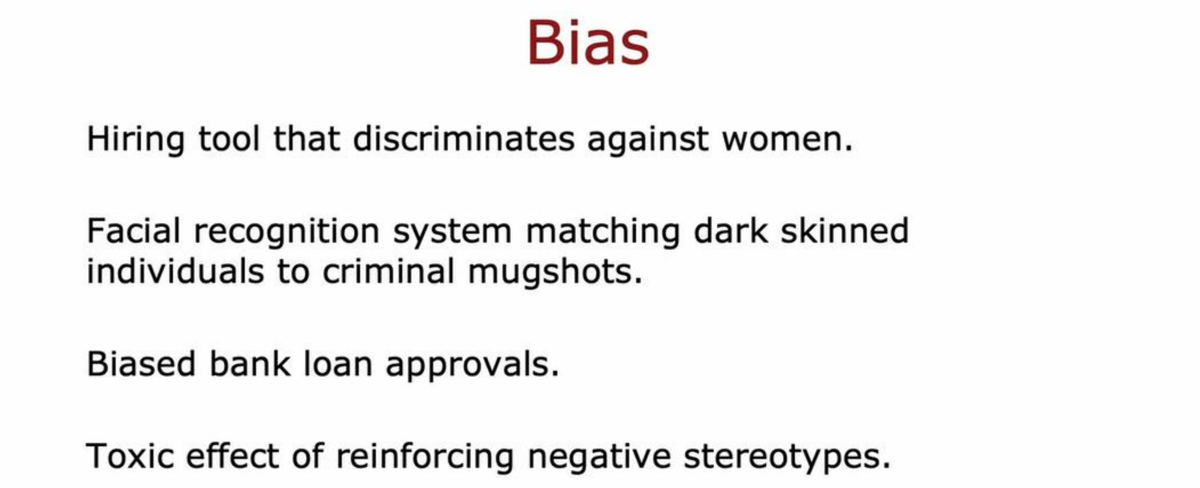
50 images

5. Full cycle of a machine learning project





6. Fairness, bias, and ethics



Adverse use cases

Deepfakes

Spreading toxic/incendiary speech through optimizing for engagement.

Generating fake content for commercial or political purposes.

Using ML to build harmful products, commit fraud etc.

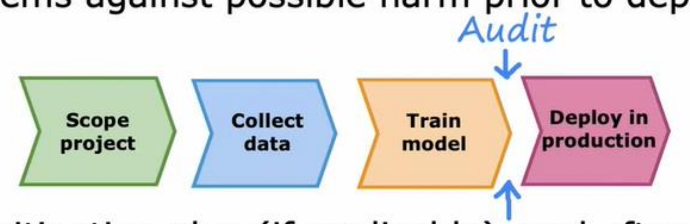
Spam vs anti-spam : fraud vs anti-fraud.

Guidelines

Get a diverse team to brainstorm things that might go wrong, with emphasis on possible harm to vulnerable groups.

Carry out literature search on standards/guidelines for your industry.

Audit systems against possible harm prior to deployment.



Develop mitigation plan (if applicable), and after deployment, monitor for possible harm.

7. Practice quiz

Which of these is a way to do error analysis?

- ○ Calculating the test error J_{test}
- ○ Calculating the training error J_{train}
- ◉ Manually examine a sample of the training examples that the model misclassified in order to identify common traits and trends.
- ○ Collecting additional training data in order to help the algorithm do better.

Correct

Correct. By identifying similar types of errors, you can collect more data that are similar to these misclassified examples in order to train the model to improve on these types of examples.

We sometimes take an existing training example and modify it (for example, by rotating an image slightly) to create a new example with the same label. What is this process called?

- ○ Machine learning diagnostic
- ◉ Data augmentation
- ○ Bias/variance analysis
- ○ Error analysis

Correct
Yes! Modifying existing data (such as images, or audio) is called data augmentation.

What are two possible ways to perform transfer learning? Hint: two of the four choices are correct.

- [x] You can choose to train just the output layers' parameters and leave the other parameters of the model fixed.

Correct
Correct. The earlier layers of the model may be reusable as is, because they are identifying low level features that are relevant to your task.

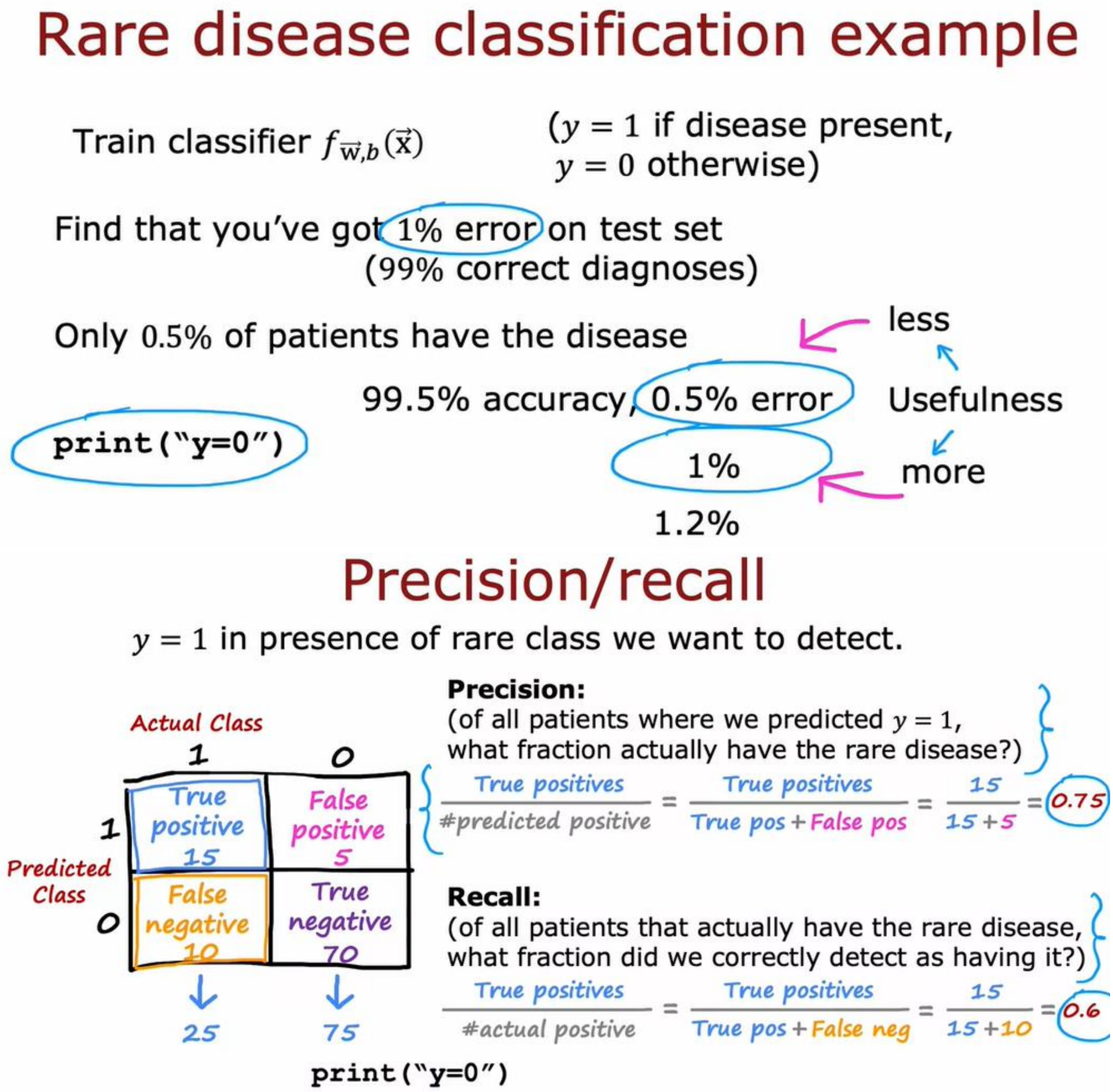
- [ ] Download a pre-trained model and use it for prediction without modifying or re-training it.
- [x] You can choose to train all parameters of the model, including the output layers, as well as the earlier layers.

Correct
Correct. It may help to train all the layers of the model on your own training set. This may take more time compared to if you just trained the parameters of the output layers.

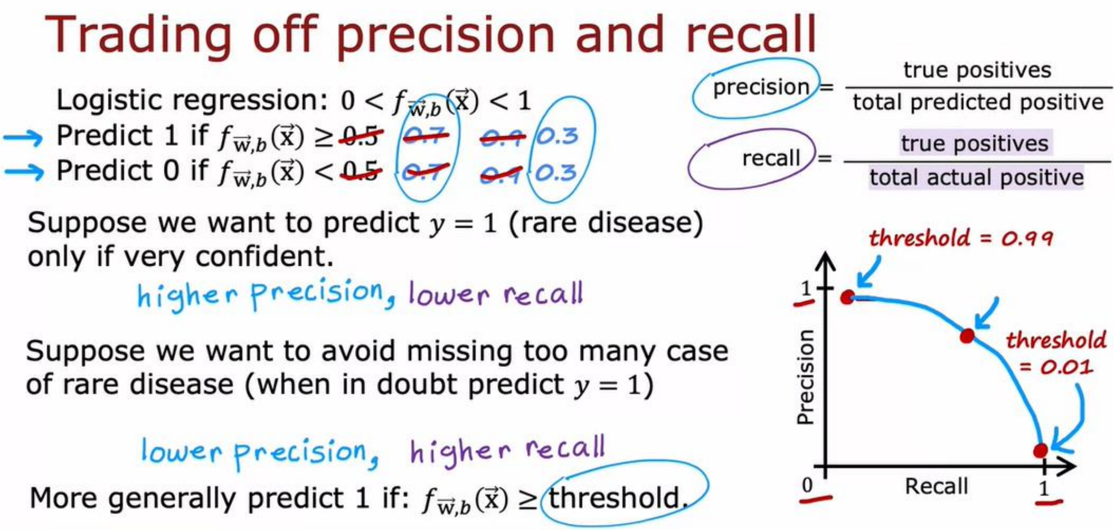
- [ ] Given a dataset, pre-train and then further fine tune a neural network on the same dataset.

Skewed Dataset

1. Error metrics for skewed dataset



2. Trading off precision and recall



F1 score

How to compare precision/recall numbers?

	Precision (P)	Recall (R)	~~Average~~	F_1 score
Algorithm 1	0.5	0.4	~~0.45~~	0.444
Algorithm 2	0.7	0.1	~~0.4~~	0.175
Algorithm 3	0.02	1.0	~~0.501~~	0.0392

```
print("y=1")
```

Harmonic mean

$$\cancel{\text{Average} = \frac{P+R}{2}}$$

$$F_1\,\text{score} = \frac{1}{\frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)} = 2\,\frac{PR}{P+R}$$