Machine Learning - Week 10

赵燕

目录

Part I

Application Example:Photo OCR

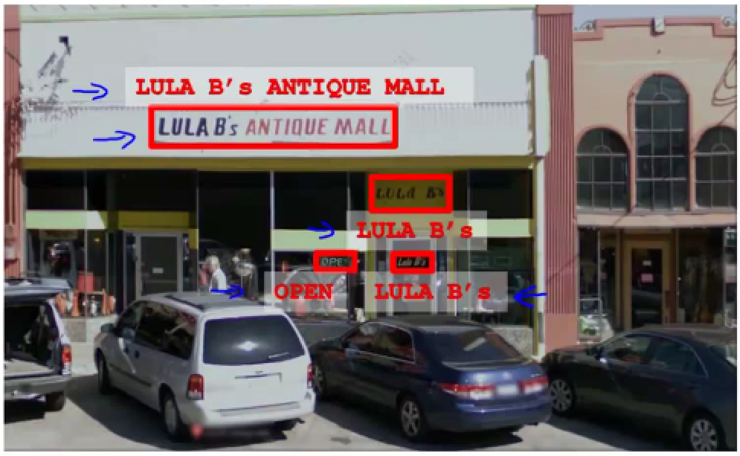
应用实例：图片文字识别

1 Photo OCR

1.1 Problem Description and Pipeline

问题描述和流程图

The Photo OCR problem



图像文字识别应用所作的事是,从一张给定的图片中识别文字。这比从一份扫描文档中识别文字要复杂的多。

为了完成这样的工作,需要采取如下步骤:

1. 文字侦测(Text detection)——将图片上的文字与其他环境对象分离开来

2. 字符切分(Character segmentation)——将文字分割成一个个单一的字符

3. 字符分类(Character classification)——确定每一个字符是什么 可以用任务流程图来表达这个问题,每一项任务可以由一个单独的小队来负责解决:

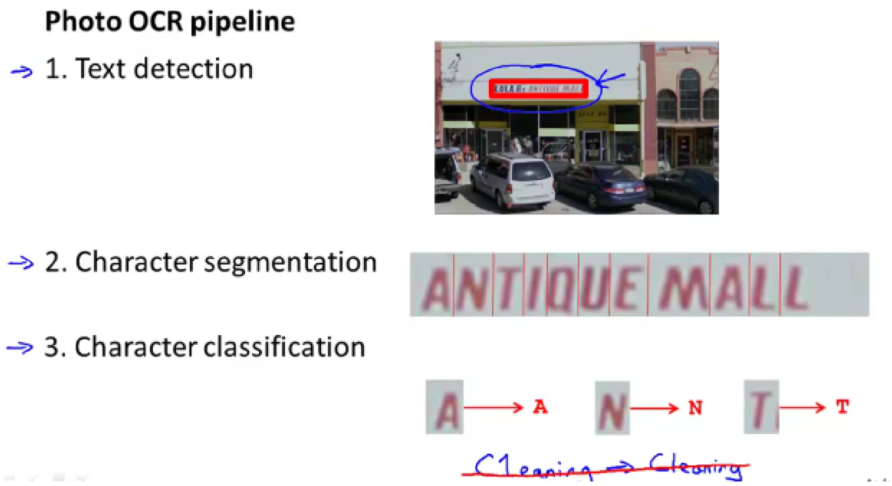


Photo OCR pipeline

1. Text detection

2. Character segmentation

3. Character classification

把这样的一个系统称为机器学习流水线（Machine Learning pipeline）

下面是照片OCR的流水线:

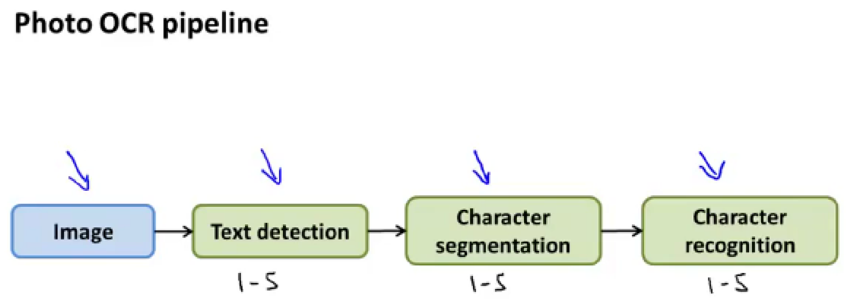
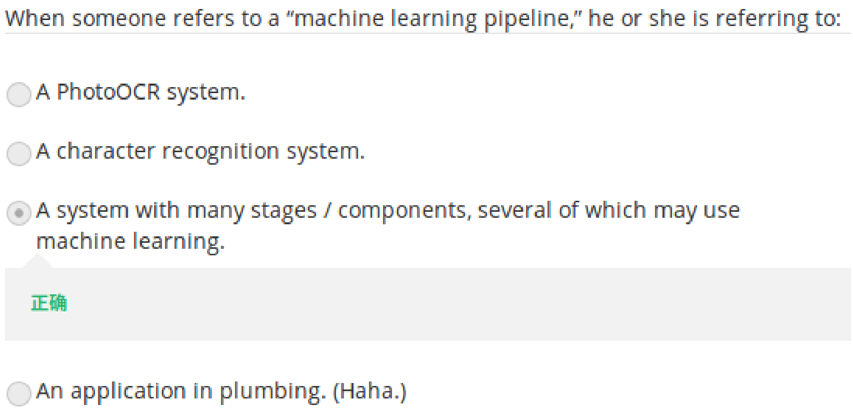


Photo OCR pipeline



When someone refers to a “machine learning pipeline,” he or she is referring to:

○ A PhotoOCR system.

○ A character recognition system.

● A system with many stages / components, several of which may use machine learning.

正确

○ An application in plumbing. (Haha.)

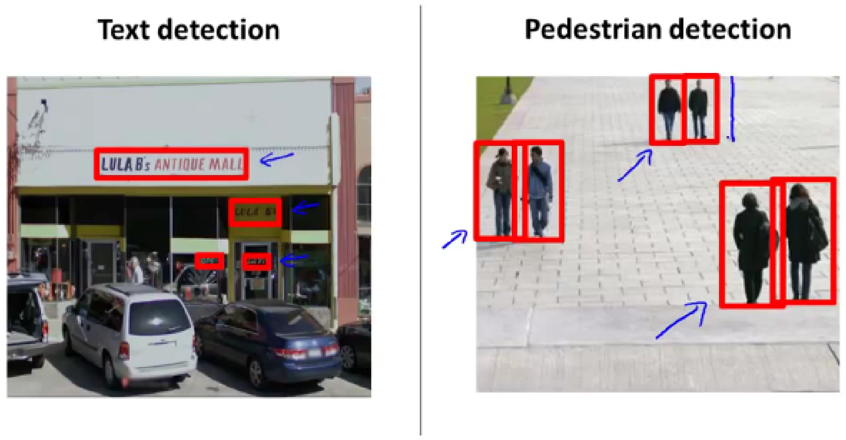
1.2 Sliding Windows

滑动窗口

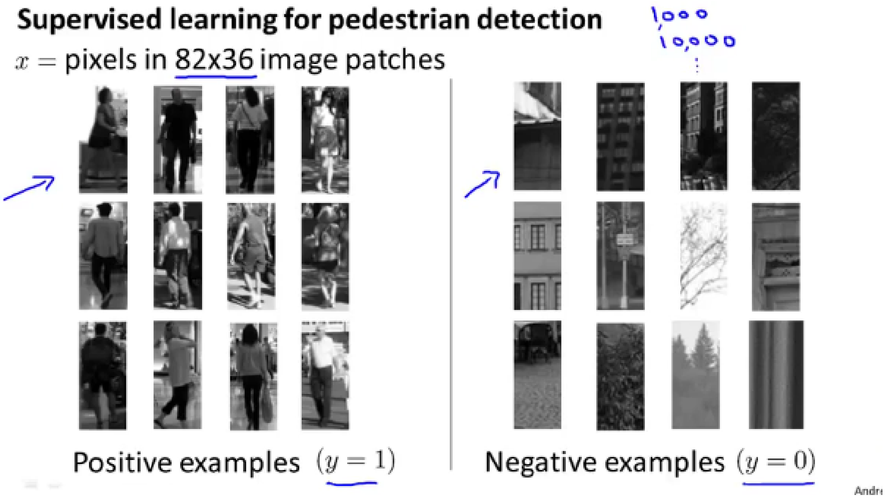
滑动窗口是一项用来从图像中抽取对象的技术。假使我们需要在一张图片中识别行人,首先要做的是用许多固定尺寸的图片来训练一个能够准确识别行人的模型。然后我们用之前训练识别行人的模型时所采用的图片尺寸在我们要进行行 人识别的图片上进行剪裁,然后将剪裁得到的切片交给模型,让模型判断是否为行人,然后在图片上滑动剪裁区域重新进行剪裁,将新剪裁的切片也交给模型进行判断,如此循环直至将图片全部检测完。

一旦完成后,我们按比例放大剪裁的区域,再以新的尺寸对图片进行剪裁,将新剪裁的切片按比例缩小至模型所采纳的尺寸,交给模型进行判断,如此循环。

滑动窗口技术也被用于文字识别,首先训练模型能够区分字符与非字符,然后,运用滑动窗口技术识别字符,一旦完成了字符的识别,我们将识别得出的区域进行一些扩展,然后将重叠的区域进行合并。接着我们以宽高比作为过滤条件,过滤掉高度比宽度更大的区域(认为单词的长度通常比高度要大)。下图中绿色的区域是经过这些步骤后被认为是文字的区域,而红色的区域是被忽略的。


Text detection

Pedestrian detection

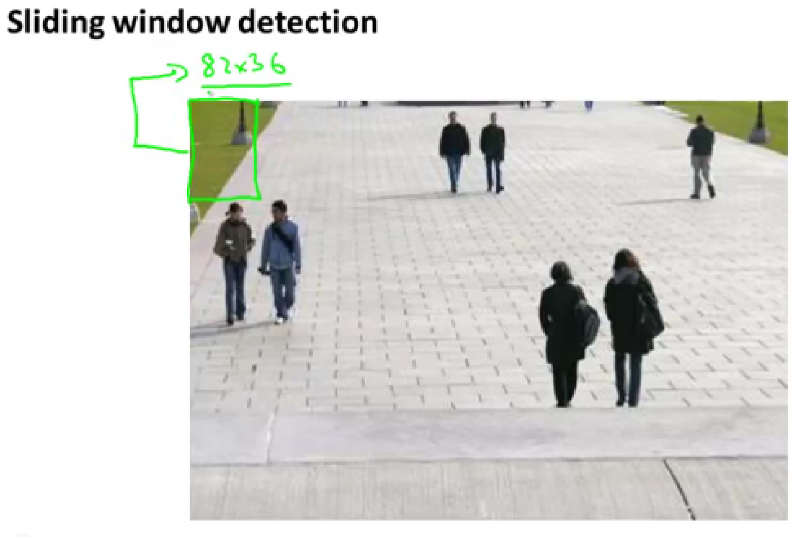

Supervised learning for pedestrian detection

x = pixels in 82x36 image patches

Positive examples $(y = 1)$

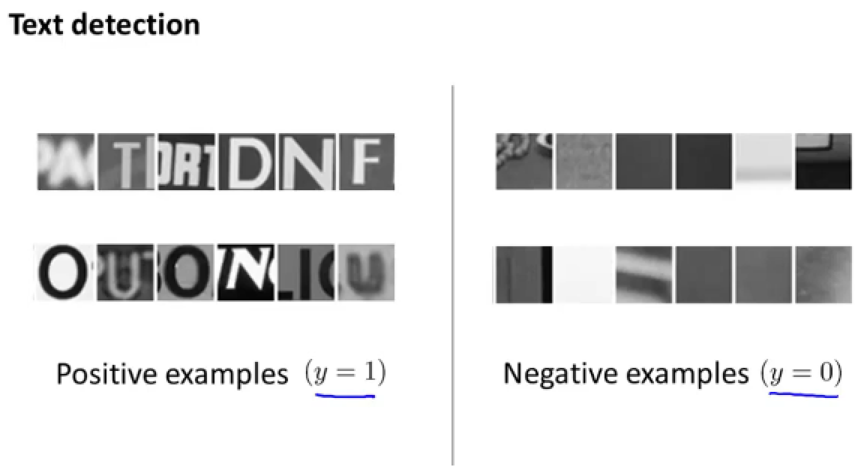
Negative examples $(y = 0)$

步长和步幅参数，是每次移动的距离。



Sliding window detection

文字识别:

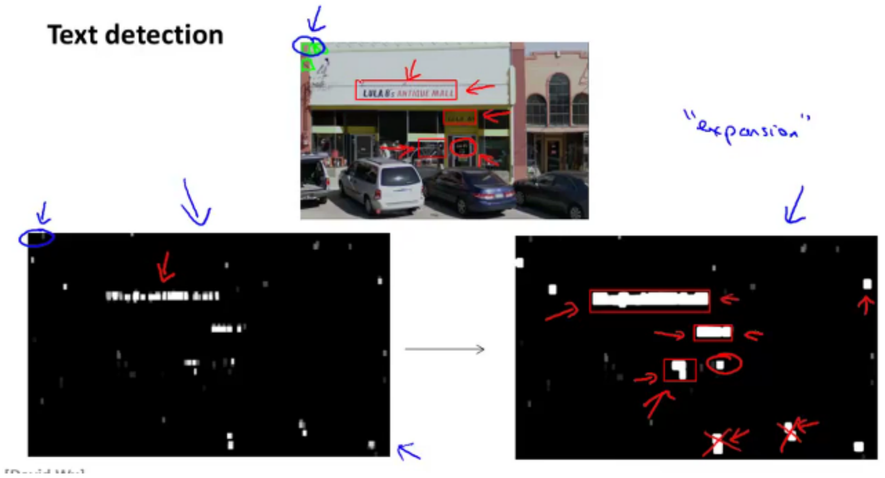


Text detection

Positive examples $(y = 1)$

Negative examples $(y = 0)$

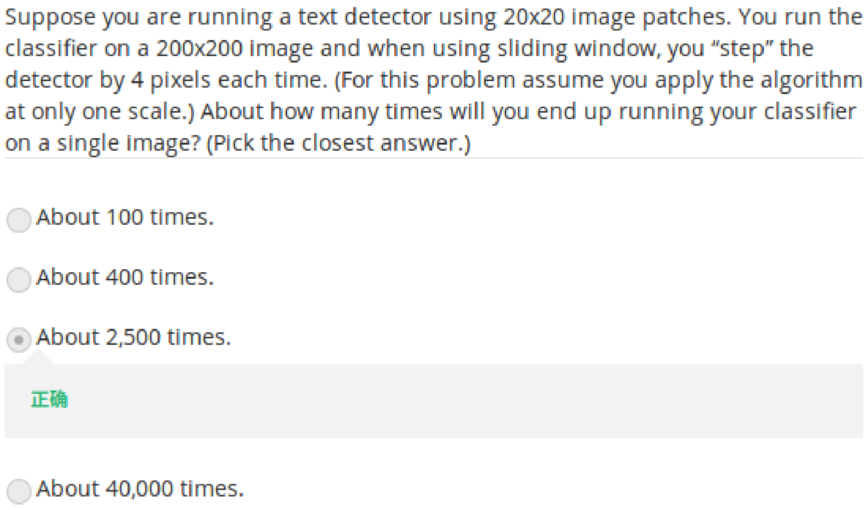
滑动窗应用于文字识别:



Text detection

实际上想做的是在图像中有文字的各区域都画上矩形窗,所以我们还需要完成一步, 我们取出分类器的输出, 然后输入到一个 ,被称为"展开器"(expansion operator)的东西 ,展开器的作用就是 它会取过这张图片 ,对每

一个白色的小点 都扩展为一块白色的区域,从数学上说 ,这一步实现就是看右边这幅图 ,我们得到右边这幅图的做法就是 ,对于每一个像素 ,我们都考察一下, 它是不是在左边这幅图中的某个白色像素的范围之内 ,所以比如说 ,如果某一个像素点 ,在最左边那幅图中, 白色像素点的五或十个像素范围中 ,那么我们将把右边那幅图的相同像素设为白色 ,因此 这样做的效果就是我们把左边图中的所有的白色小点,都扩展了一下, 让它们都变大了一些 ,根据它们周围临近像素是不是白色的 ,如果是的话 ,我们把它们也变成了白色, 这样我们就快完成了,我们现在可以根据 右边的这张图锁定那些连接部分 ,也就是这些连续的白色区域 ,然后围绕着它们画个框就行了 ,具体来讲, 如果我们分析这些白色区域 ,比如说这个, 这个, 这个 ,等等 ,我们可以简单地凭直觉来判断 哪些区域是比较奇怪的 ,因为我们知道有文字的区域, 应该不是很高的, 而是比较宽的, 所以我们忽略那些 ,又高又瘦的白块 ,比如这个和这个, 我们抛弃这些, 因为它们太瘦长了 ,然后对剩下的那些 从比例上来看,比较像正常的文字区域的 ,这些白块 画上矩形窗, 也就是说 ,们围着这些 ,白块画上矩形边界 比如这里 LULA B's ANTIQUE MALL 商标 还有 LULA B's 和那边小的"正在营业" 的牌子。



Suppose you are running a text detector using 20x20 image patches. You run the classifier on a 200x200 image and when using sliding window, you "step" the detector by 4 pixels each time. (For this problem assume you apply the algorithm at only one scale.) About how many times will you end up running your classifier on a single image? (Pick the closest answer.)

○ About 100 times.

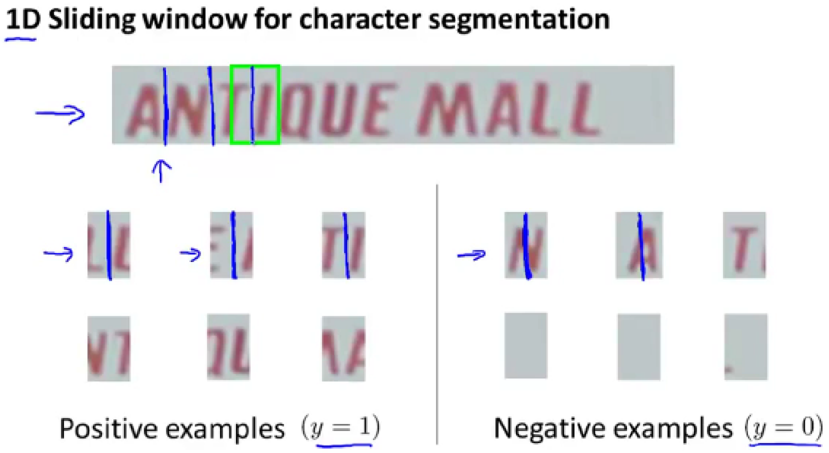
○ About 400 times.

● About 2,500 times.

正确

○ About 40,000 times.

字符分割:



1D Sliding window for character segmentation

照片OCR流水线:

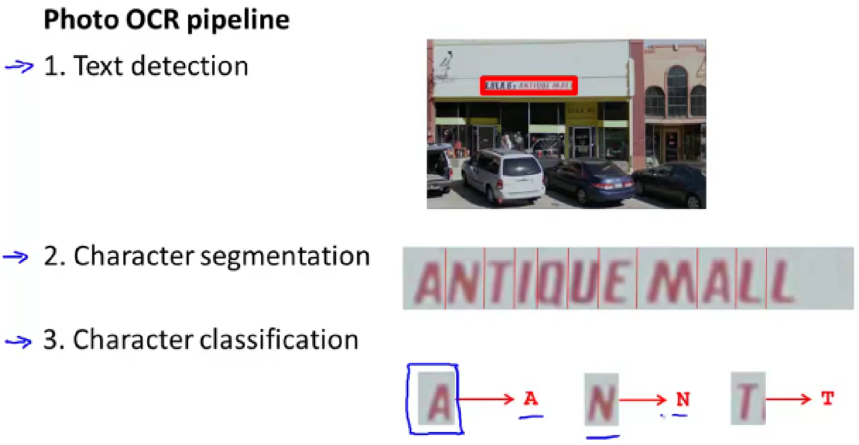


Photo OCR pipeline

1. Text detection

2. Character segmentation

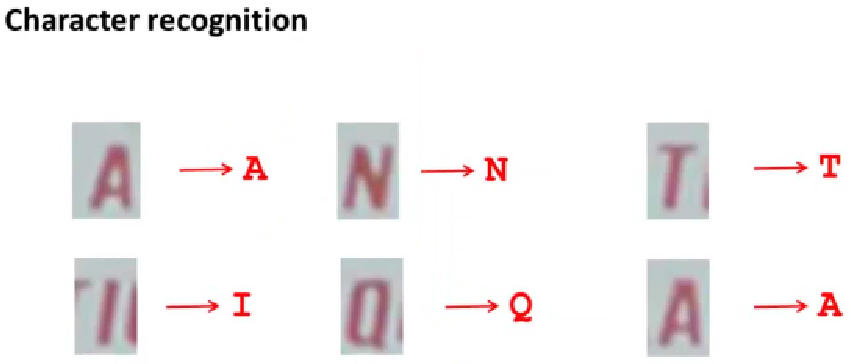
3. Character classification

在文字检测中我们使用滑动窗来检测文字,同时我们还用了一个 ,一维滑动窗来进行 ,字符的分割来将图像分割为独立的字符 ,流水线的最后一步 ,是字符分类 ,这一步 ,根据之前你已经学到的监督学习算法, 你应该比较清楚 ,应该怎么做了 ,你可以使用一种 ,标准的监督学习算法 ,比如神经网络,或者其他方法 ,输入这样的图像 ,然后将图像按字母分类 ,化为26个字母 ,A到Z中的一个, 或者我们也可以有36种字符 ,如果你算上数字字符的话 ,这就是一个多元分类问题, 你应该把这个带有字符的图像 作为输入, 然后确定这个图像中出现了什么字符。

1.3 Getting Lots of Data and Artificial Data

获取大量数据和人工数据

如果我们的模型是低方差的,那么获得更多的数据用于训练模型,是能够有更好的效果的。问题在于,我们怎样获得数据,数据不总是可以直接获得的,我们有可能需要人工地创造一些数据。



Character recognition

以我们的文字识别应用为例,我们可以字体网站下载各种字体,然后利用这些不同的字体配上各种不同的随机背景图片创造出一些用于训练的实例,这让我们能够获得一个无限大的训练集。这是从零开始创造实例。

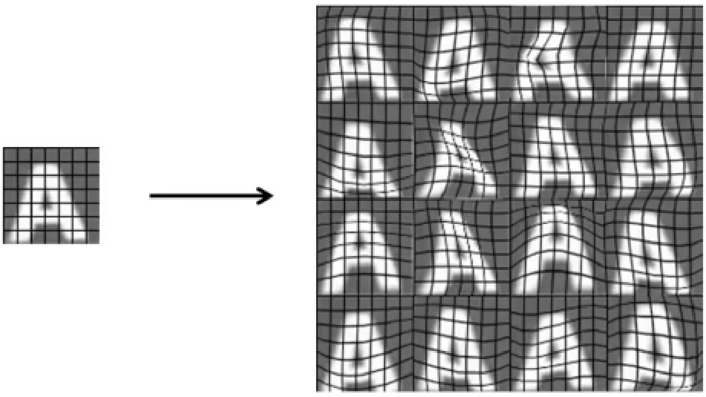
Artificial data synthesis for photo OCR

Real data

Synthetic data

另一种方法是,利用已有的数据,然后对其进行修改,例如将已有的字符图片进行一些扭曲、旋转、模糊处理。只要我们认为实际数据有可能和经过这样处理后的数据类似,我们便可以用这样的方法来创造大量的数据。

Synthesizing data by introducing distortions



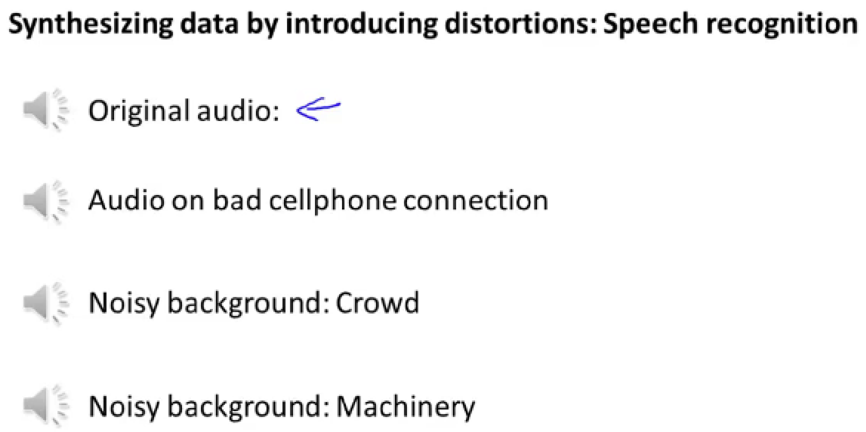
有关获得更多数据的几种方法:

1. 人工数据合成

2. 手动收集、标记数据

3. 众包

语音识别:



Synthesizing data by introducing distortions: Speech recognition

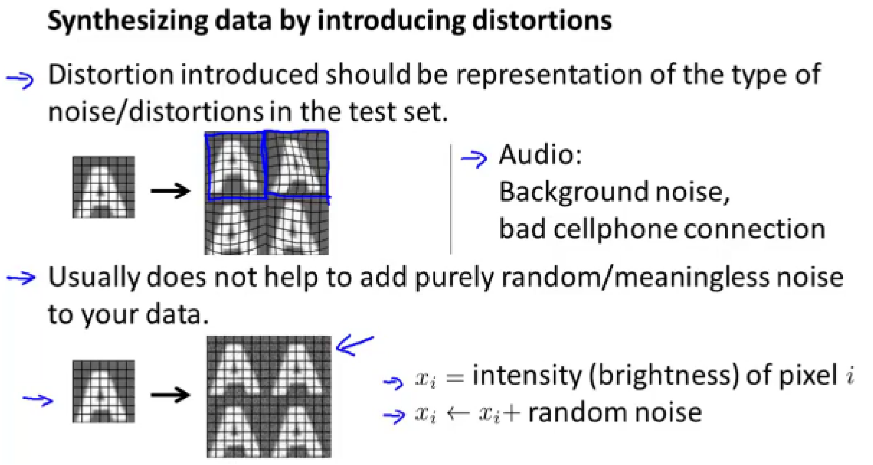
Original audio:

Audio on bad cellphone connection

Noisy background: Crowd

Noisy background: Machinery

对人工合成数据中引入变形的方法提一些注意事项：



Synthesizing data by introducing distortions

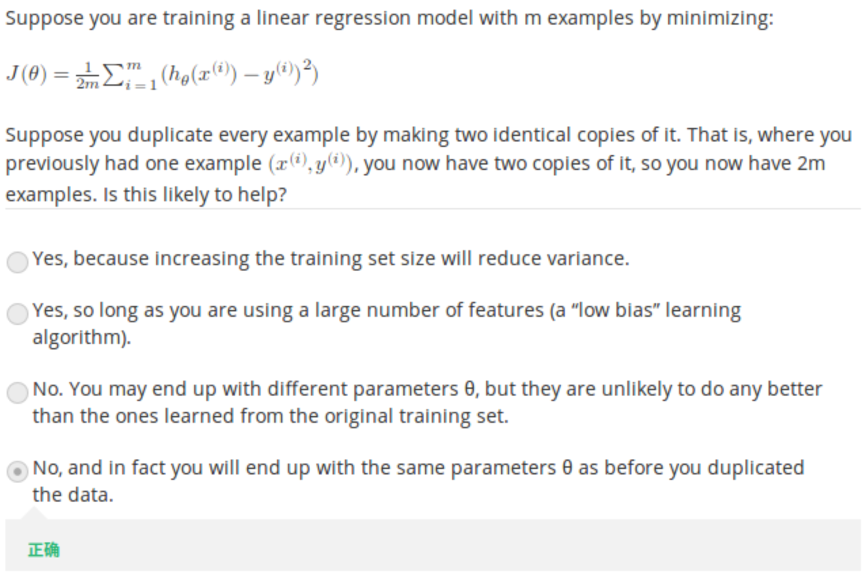
Distortion introduced should be representation of the type of noise/distortions in the test set.

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



Suppose you are training a linear regression model with m examples by minimizing:

$J(\theta)=\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2)$

Suppose you duplicate every example by making two identical copies of it. That is, where you previously had one example $(x^{(i)},y^{(i)})$, you now have two copies of it, so you now have 2m examples. Is this likely to help?

○ Yes, because increasing the training set size will reduce variance.

○ Yes, so long as you are using a large number of features (a "low bias" learning algorithm).

○ No. You may end up with different parameters θ, but they are unlikely to do any better than the ones learned from the original training set.

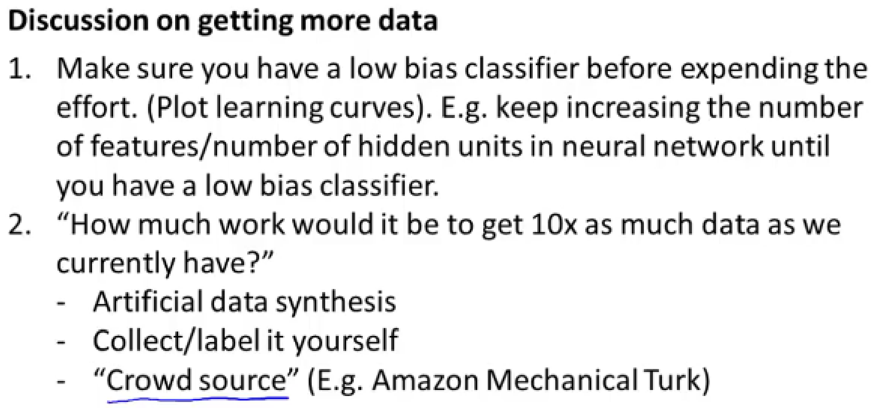
● No, and in fact you will end up with the same parameters θ as before you duplicated the data.

正确

人工数据合成获得大量数据的方法，同往常一样，在花费大量精力考虑如何产生大量人工训练样本之前，通常最好应该先保证你已经有了一个低偏差的分类器，这样得到大量的数据才真的会起作用 ，标准的方法是 ，画出学习曲线 ，然后确保你已经有了一个低偏差或者高方差的分类器 ，或者如果你没有 ，得到一个

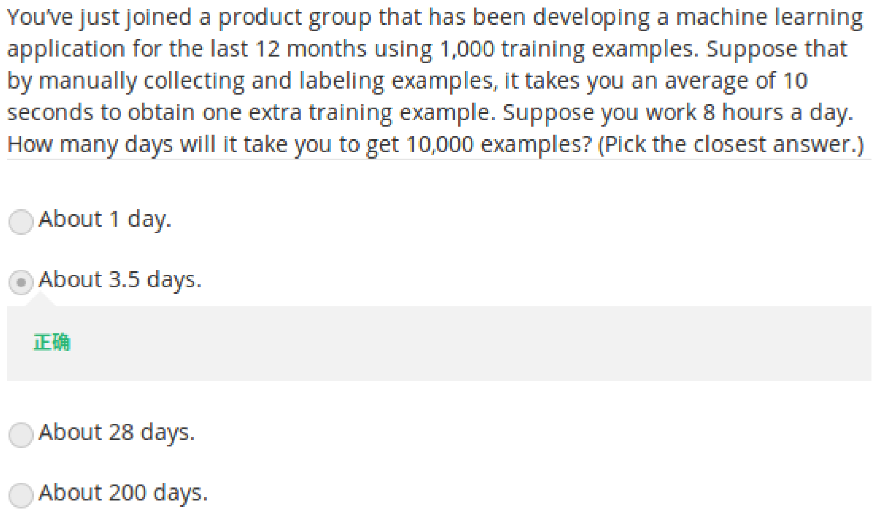
低偏差的分类器， 你还可以尝试另一种方法， 那就是增大分类器的特征数 ， 或者在神经网络中， 增大隐藏层单元数 直到你得到一个偏差比较小的分类器 ， 直到这时， 你再来考虑建立大量的人工训练集 ， 所以你一定要避免的是 ， 花了几个星期的时间， 或者几个月的工夫 ， 考虑好了怎么样 ， 能获得比较好的人工合成数据，然后才意识到即使获得了大量的训练数据自己的学习算法的表现依然没有提高多少；

人工数据合成，包括包括从零开始生成新数据，比如使用随机的字体等等，也包括第二种思路 ，那就是在现有的样本中引入一些噪声或变形，来扩大现有的训练样本， 有很多获取大量数据的方法，是你自己收集或者标记数据 ，第三点也是最后一点， 另一种很好的办法是 我们称之为”众包” (crowd sourcing) 的办法 ，现在已经有一些网站或者一些服务机构能让你通过网络 雇一些人替你完成标记大量训练数据的工作，通常都很廉价， 因此这种众包的方法或者叫众包的数据标记，很明显这种方法就像学术文献一样它也是很复杂的 ，同时取决于标记人的可靠性 ，也许世界上有 ，成百上千的标记人用很低的收入帮你为数据加上标签就像我刚才说的 这也是一种选择而已 可能“亚马逊土耳其机器人”（Amazon Mechanical Turk） ，就是当前最流行的一个众包选择 ，要让它工作 ，要想获得比较高质量的标签 ，通常还不是一件容易的事 ，需要花一定的功夫， 但这也是一种可供选择的方法 。



Discussion on getting more data

1. Make sure you have a low bias classifier before expending the effort. (Plot learning curves). E.g. keep increasing the number of features/number of hidden units in neural network until you have a low bias classifier.
2. "How much work would it be to get 10x as much data as we currently have?"
   - Artificial data synthesis
   - Collect/label it yourself
   - "Crowd source" (E.g. Amazon Mechanical Turk)



You've just joined a product group that has been developing a machine learning application for the last 12 months using 1,000 training examples. Suppose that by manually collecting and labeling examples, it takes you an average of 10 seconds to obtain one extra training example. Suppose you work 8 hours a day. How many days will it take you to get 10,000 examples? (Pick the closest answer.)

○ About 1 day.

● About 3.5 days.

正确

○ About 28 days.

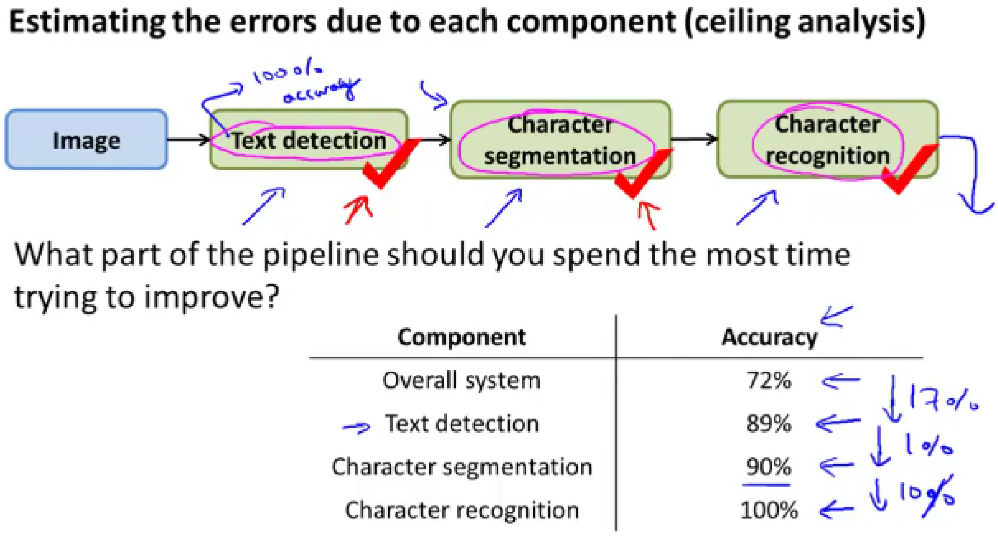
○ About 200 days.

1.4 Ceiling Analysis:What Part of the Pipeline to Work on Next

上限分析：执行流水线的哪一部分

在机器学习的应用中,我们通常需要通过几个步骤才能进行最终的预测,我们如何能够知道哪一部分最值得我们花时间和精力去改善呢?这个问题可以通过上限分析来回答。

这里打钩的地方我想做的是遍历我的测试集，直接向它公布“标准答案”， 为这个流程中的文字检测部分，直接提供正确的标签 这样好像我就会有一个非常棒的文字检测系统，能很好地检测我的测试样本，然后我们要做的是继续运行完接下来的几个模块， 也就是字符分割和字符识别， 然后使用跟之前一样的， 评价量度指标来测量整个系统的总体准确度，这样用准确的文字检测结果， 系统的表现应该会有提升， 假如说， 准确率 提高到89%， 然后我们继续进行接着执行流水线中的下一模块， 字符分割同前面一样，我还是去找出我的测试集， 然后现在我不仅用标准的文字检测结果，我还同时用标准的字符分割结果，所以还是遍历测试样本人工地给出正确的字符分割结果然后看看这样做以后效果怎样变化，假如我们这样做以后整个系统准确率提高到90%，注意跟前面一样，这里说的准确率是指整个系统的准确率，所以无论最后一个模块，字符识别模块给出的最终输出是什么， 无论整个流水线的， 最后输出结果是什么， 我们都是测出的整个系统的准确率， 最后我们还是执行最后一个模块 字符识别， 同样也是人工给出这一模块的正确标签， 这样做以后我应该理所当然得到100%准确率。



Estimating the errors due to each component (ceiling analysis)

What part of the pipeline should you spend the most time trying to improve?

Component	Accuracy
Overall system	72%
Text detection	89%
Character segmentation	90%
Character recognition	100%

流程图中每一部分的输出都是下一部分的输入,上限分析中,我们选取一部分,手工提供 100%正确的输出结果,然后看应用的整体效果提升了多少。假使我们的例子中总体效果 为 72%的正确率。

如果我们令文字侦测部分输出的结果 100%正确,发现系统的总体效果从 72%提高到了89%。这意味着我们很可能会希望投入时间精力来提高我们的文字侦测部分。

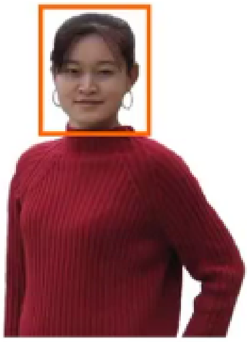
接着我们手动选择数据,让字符切分输出的结果 100%正确,发现系统的总体效果只提升了 1%,这意味着,我们的字符切分部分可能已经足够好了。

最后我们手工选择数据,让字符分类输出的结果 100%正确,系统的总体效果又提升了10%,这意味着我们可能也会应该投入更多的时间和精力来提高应用的总体表现。

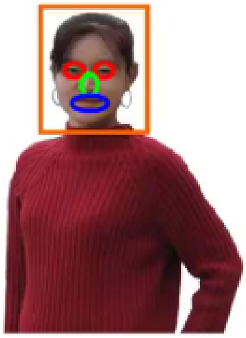
人脸识别的示例:

图片预处理：去除背景：

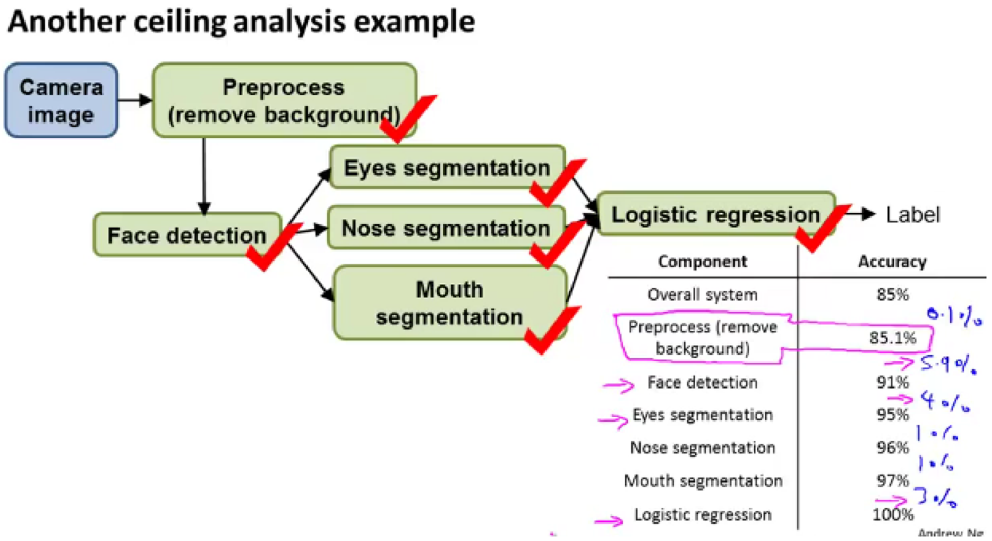
面部识别：



关键特征识别：



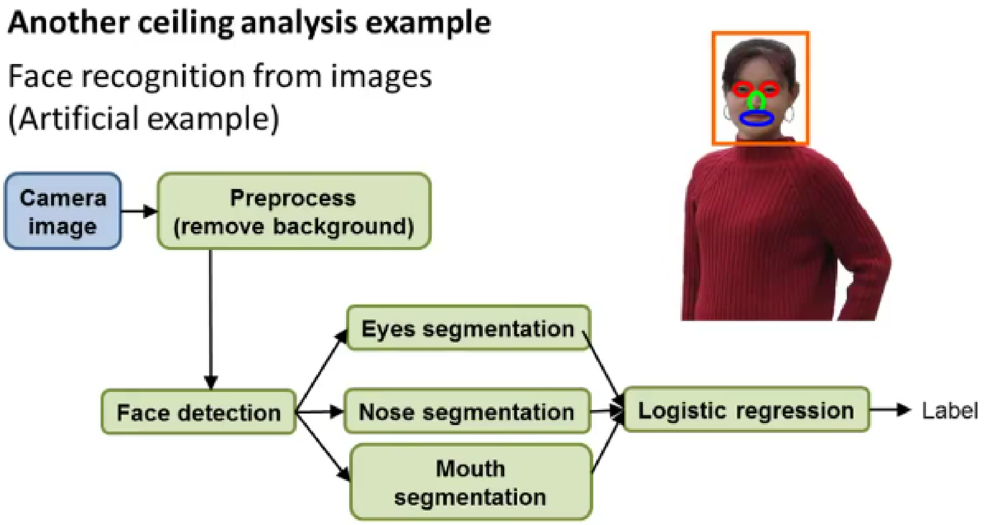
这样找出了眼睛、鼻子、嘴巴所有这些都是非常有用的特征,然后这些特征可以被输入给某个,逻辑回归的分类器,然后这个分类器的任务,就是给出最终的标签,找出我们认为能辨别出这个人是谁的最终的标签。



Another ceiling analysis example

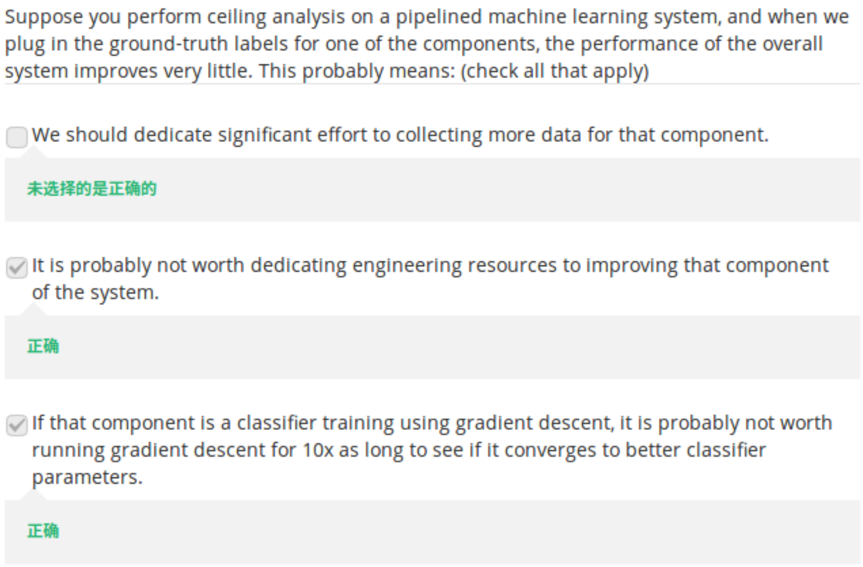
Component	Accuracy	
Overall system	85%	
Preprocess (remove background)	85.1%	0.1%
Face detection	91%	5.9%
Eyes segmentation	95%	4%
Nose segmentation	96%	1%
Mouth segmentation	97%	1%
Logistic regression	100%	3%

Andrew Ng



Another ceiling analysis example

Face recognition from images
(Artificial example)



Suppose you perform ceiling analysis on a pipelined machine learning system, and when we plug in the ground-truth labels for one of the components, the performance of the overall system improves very little. This probably means: (check all that apply)

□ We should dedicate significant effort to collecting more data for that component.

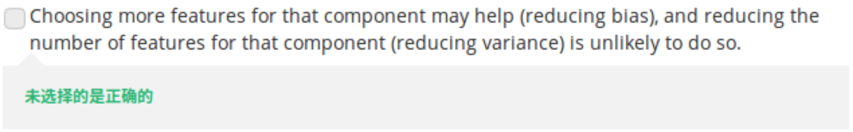
未选择的是正确的

☑ It is probably not worth dedicating engineering resources to improving that component of the system.

正确

☑ If that component is a classifier training using gradient descent, it is probably not worth running gradient descent for 10x as long to see if it converges to better classifier parameters.

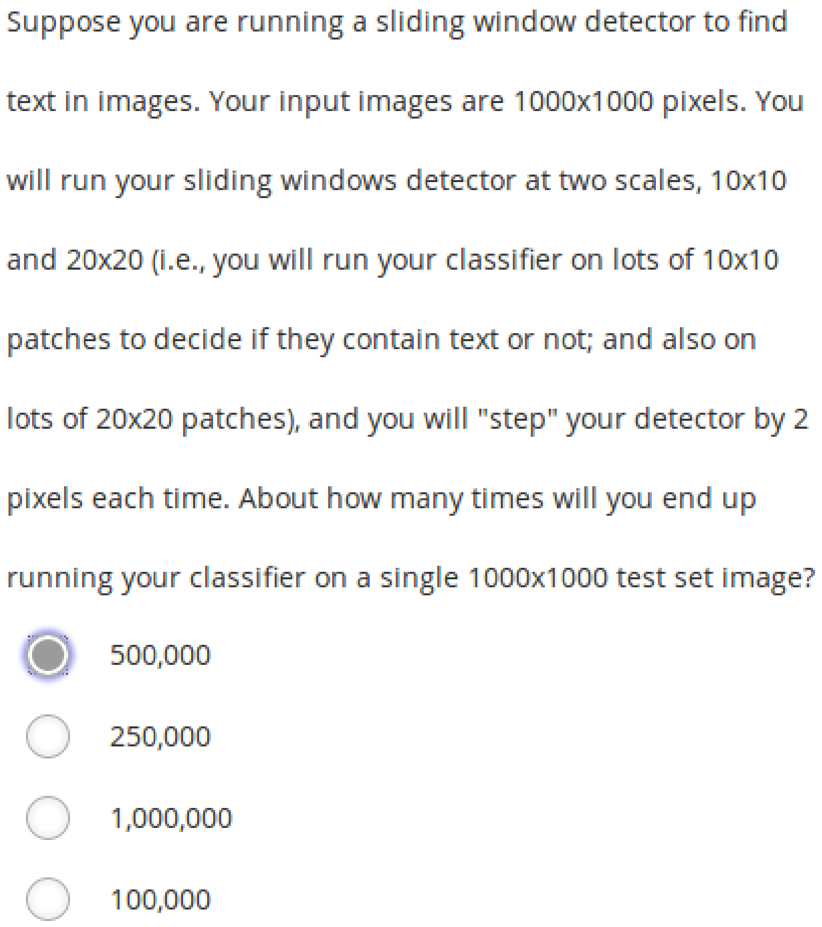
正确



□ Choosing more features for that component may help (reducing bias), and reducing the number of features for that component (reducing variance) is unlikely to do so.

未选择的是正确的

1。



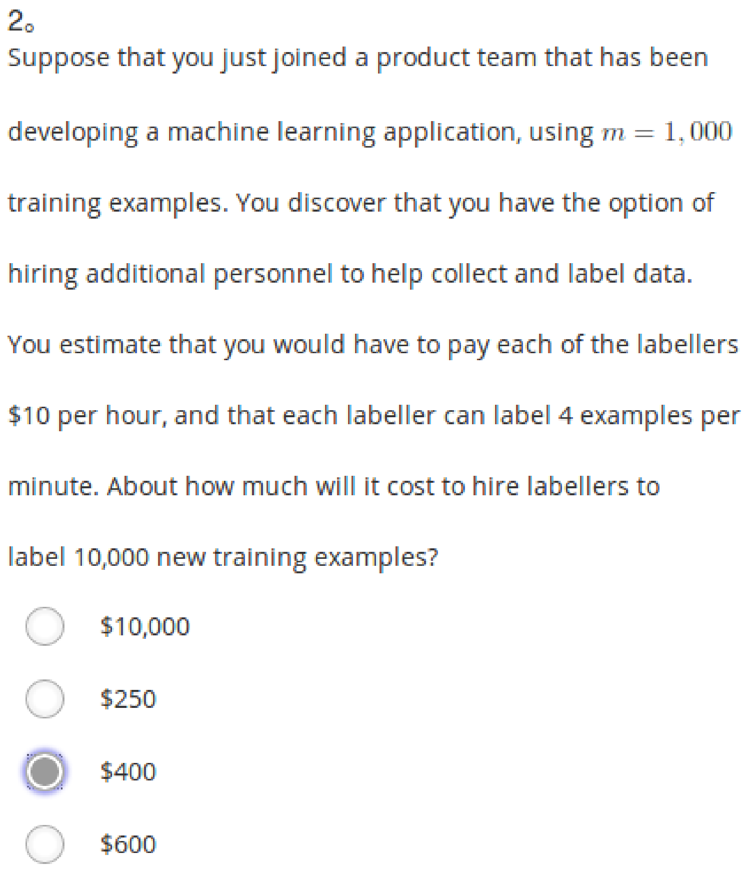
Suppose you are running a sliding window detector to find text in images. Your input images are 1000x1000 pixels. You will run your sliding windows detector at two scales, 10x10 and 20x20 (i.e., you will run your classifier on lots of 10x10 patches to decide if they contain text or not; and also on lots of 20x20 patches), and you will "step" your detector by 2 pixels each time. About how many times will you end up running your classifier on a single 1000x1000 test set image?

● 500,000

○ 250,000

○ 1,000,000

○ 100,000



2.

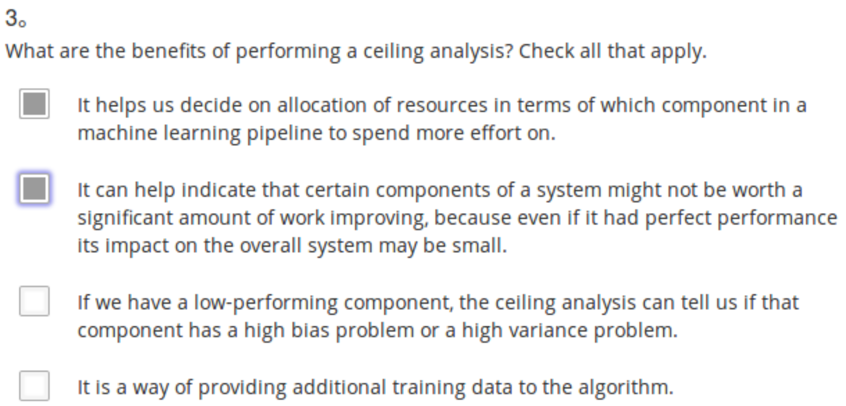
Suppose that you just joined a product team that has been developing a machine learning application, using $m = 1,000$ training examples. You discover that you have the option of hiring additional personnel to help collect and label data. You estimate that you would have to pay each of the labellers $10 per hour, and that each labeller can label 4 examples per minute. About how much will it cost to hire labellers to label 10,000 new training examples?

○ $10,000

○ $250

● $400

○ $600



3.

What are the benefits of performing a ceiling analysis? Check all that apply.

☑ It helps us decide on allocation of resources in terms of which component in a machine learning pipeline to spend more effort on.

☑ It can help indicate that certain components of a system might not be worth a significant amount of work improving, because even if it had perfect performance its impact on the overall system may be small.

□ If we have a low-performing component, the ceiling analysis can tell us if that component has a high bias problem or a high variance problem.

□ It is a way of providing additional training data to the algorithm.

4。

Suppose you are building an object classifier, that takes as input an image, and recognizes that image as either containing a car ($y = 1$) or not ($y = 0$). For example, here are a positive example and a negative example:

Positive example ($y = 1$)

Negative example ($y = 0$)

After carefully analyzing the performance of your algorithm, you conclude that you need more positive ($y = 1$) training examples. Which of the following might be a good way to get additional positive examples?

○ Mirror your training images across the vertical axis (so that a left-facing car now becomes a right-facing one).

● Take a few images from your training set, and add random, gaussian noise to every pixel.

○ Take a training example and set a random subset of its pixel to 0 to generate a new example.

○ Select two car images and average them to make a third example.

After carefully analyzing the performance of your algorithm, you conclude that you need more positive ($y = 1$) training examples. Which of the following might be a good way to get additional positive examples?

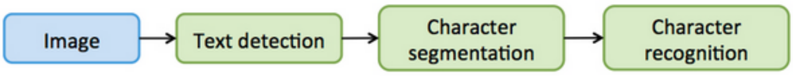
● Apply translations, distortions, and rotations to the images already in your training set.

○ Select two car images and average them to make a third example.

○ Take a few images from your training set, and add random, gaussian noise to every pixel.

○ Make two copies of each image in the training set; this immediately doubles your training set size.

5。
Suppose you have a PhotoOCR system, where you have the following pipeline:



You have decided to perform a ceiling analysis on this system, and find the following:

Component	Accuracy
Overall System	70%
Text Detection	72%
Character Segmentation	82%
Character Recognition	100%

Which of the following statements are true?

☑ The potential benefit to having a significantly improved text detection system is small, and thus it may not be worth significant effort trying to improve it.

☑ If we conclude that the character recognition's errors are mostly due to the character recognition system having high variance, then it may be worth significant effort obtaining additional training data for character recognition.

□ We should dedicate significant effort to collecting additional training data for the text detection system.

□ The most promising component to work on is the text detection system, since it has the lowest performance (72%) and thus the biggest potential gain.

☑ If the text detection system was trained using gradient descent, running gradient descent for more iterations is unlikely to help much.

☑ If we conclude that the character recognition's errors are mostly due to the character recognition system having high variance, then it may be worth significant effort obtaining additional training data for character recognition.

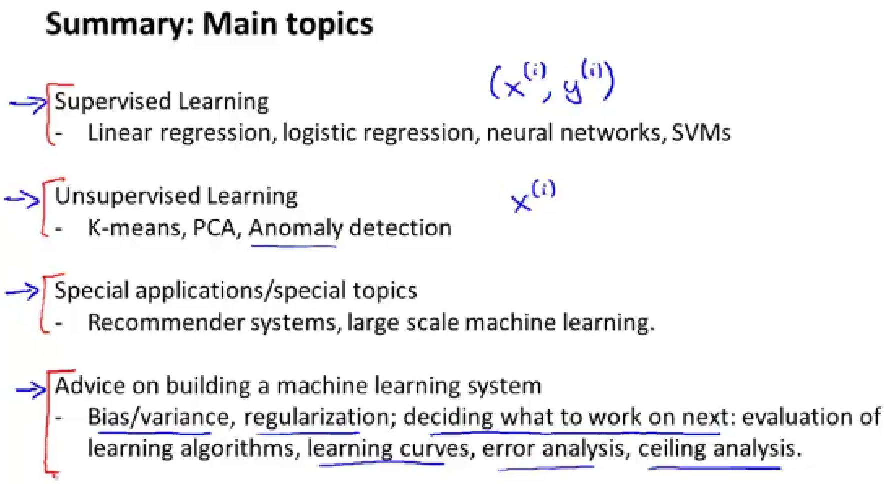
□ We should dedicate significant effort to collecting additional training data for the text detection system.

□ The least promising component to work on is the character recognition system, since it is already obtaining 100% accuracy.

Part II
Conclusion

总结

2 Summary and Thank You



Summary: Main topics

Supervised Learning $(x^{(i)}, y^{(i)})$
- Linear regression, logistic regression, neural networks, SVMs

Unsupervised Learning $x^{(i)}$
- K-means, PCA, Anomaly detection

Special applications/special topics
- Recommender systems, large scale machine learning.

Advice on building a machine learning system
- Bias/variance, regularization; deciding what to work on next: evaluation of learning algorithms, learning curves, error analysis, ceiling analysis.

在这门课中,我们花了大量的时间介绍了诸如线性回归、逻辑回归、神经网络、支持向量机等等一些监督学习算法,这类算法 具有带标签的数据和样本,比如 $x^{(i)}$ 、$y^{(i)}$。

然后我们也花了很多时间介绍无监督学习。例如 K-均值聚类、用于降维的主成分分析,以及当你只有一系列无标签数据$x^{(i)}$时的异常检测算法。

当然,有时带标签的数据,也可以用于异常检测算法的评估。此外,我们也花时间讨论了一些特别的应用或者特别的话题,比如说推荐系统。以及大规模机器学习系统,包括并行系统和映射化简方法,还有其他一些特别的应用。比如,用于计算机视觉技术的滑动窗口分类算法。

最后,我们还提到了很多关于构建机器学习系统的实用建议。这包括了怎样理解某个机器学习算法是否正常工作的原因,所以我们谈到了偏差和方差的问题,也谈到了解决方差问题的正则化,同时我们也讨论了怎样决定接下来怎么做的问题,也就是说当你在开发一个机器学习系统时,什么工作才是接下来应该优先考虑的问题。

因此我们讨论了学习算法的评价法。介绍了评价矩阵,比如:查准率、召回率以及 F1 分数,还有评价学习算法比较实用的训练集、交叉验证集和测试集。我们也介绍了学习算法的调试,以及如何确保学习算法的正常运行,于是我们介绍了一些诊断法,比如学习曲线,同时也讨论了误差分析、上限分析等等内容。

所有这些工具都能有效地指引你决定接下来应该怎样做,让你把宝贵的时间用在刀刃上。现在你已经掌握了很多机器学习的工具,包括监督学习算法和无监督学习算法等等。

但除了这些以外,我更希望你现在不仅仅只是认识这些工具,更重要的是掌握怎样有效地利用这些工具来建立强大的机器学习系统。所以,以上就是这门课的全部内容。如果你跟着我们的课程一路走来,到现在,你应该已经感觉到自己已经成为机器学习方面的专家了吧?

我们都知道,机器学习是一门对科技、工业产生深远影响的重要学科,而现在,你已经完全具备了应用这些机器学习工具来创造伟大成就的能力。我希望你们中的很多人都能在相应 的领域,应用所学的机器学习工具,构建出完美的机器学习系统,开发出无与伦比的产品和应用。并且我也希望你们通过应用机器学习,不仅仅改变自己的生活,有朝一日,还要让更多的人生活得更加美好!