热身赛

<https://competition.huaweicloud.com/information/1000035894/introduction>

编程题目

<https://developer.huaweicloud.com/hero/forum.php?mod=viewthread&tid=45502&extra=page%3D1>

免费的云服务器

排名

<https://competition.huaweicloud.com/information/1000035895/ranking>

吴恩达 视频

<https://github.com/JudasDie/deeplearning.ai>

吴恩达 笔记

<https://kyonhuang.top/Andrew-Ng-Deep-Learning-notes/#/>

<https://zhuanlan.zhihu.com/p/34726093>

吴恩达 代码

<https://github.com/JudasDie/deeplearning.ai>

吴恩达 课后作业

<https://blog.csdn.net/u013733326/article/details/79862336>

吴恩达 课程

<https://mooc.study.163.com/learn/2001281002?tid=2001392029&_trace_c_p_k2_=338681a0769f45478f00fc701d60ca05#/learn/content?type=detail&id=2001701010&cid=2001700035>

从2.5 开始看

不要用loadtxt

逻辑回归梯度下降

Python的广播就是矩阵的自适应

二分类的时候可以使用激活函数

一般可以使用tanh激活函数

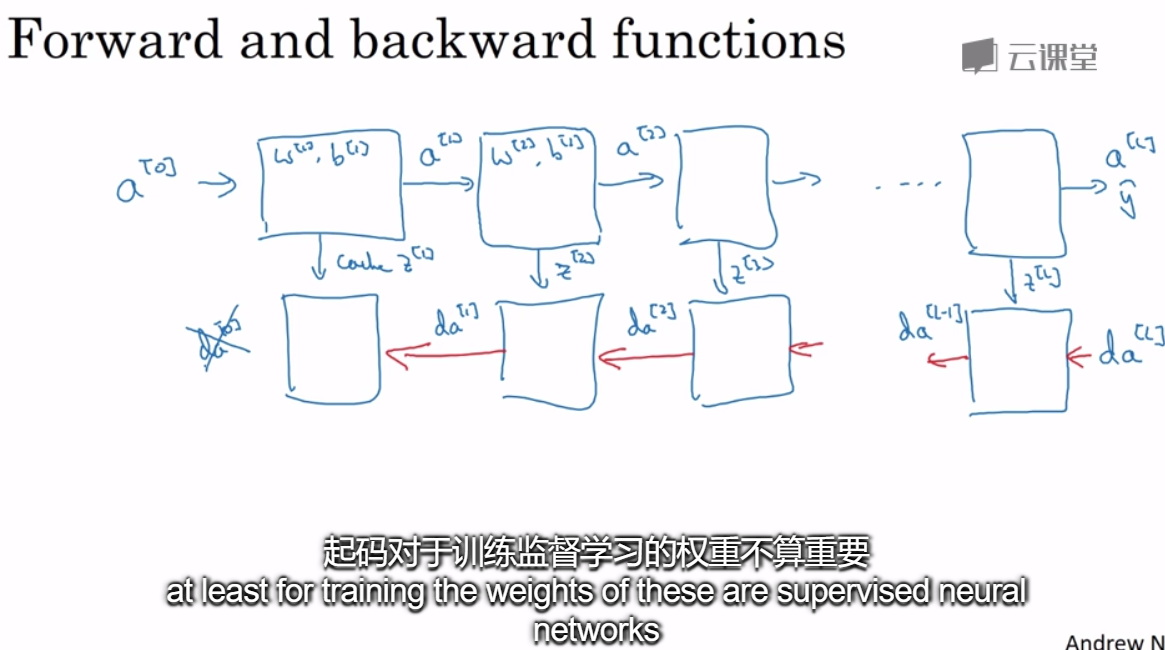
ReLu 激活函数 max(0,z) 默认选择的激活函数

线性激活函数一般没有用

刚开始的元素初始化很重要，w矩阵随机0.01 b可以为0

前向传播反向传播logical回归

深度 先认识简单的在把



大佬对于进入deep learning 的学习的建议是 读一小部分论文发现他们都错的，你直觉上对的部分然后，开始更新所有的产品线

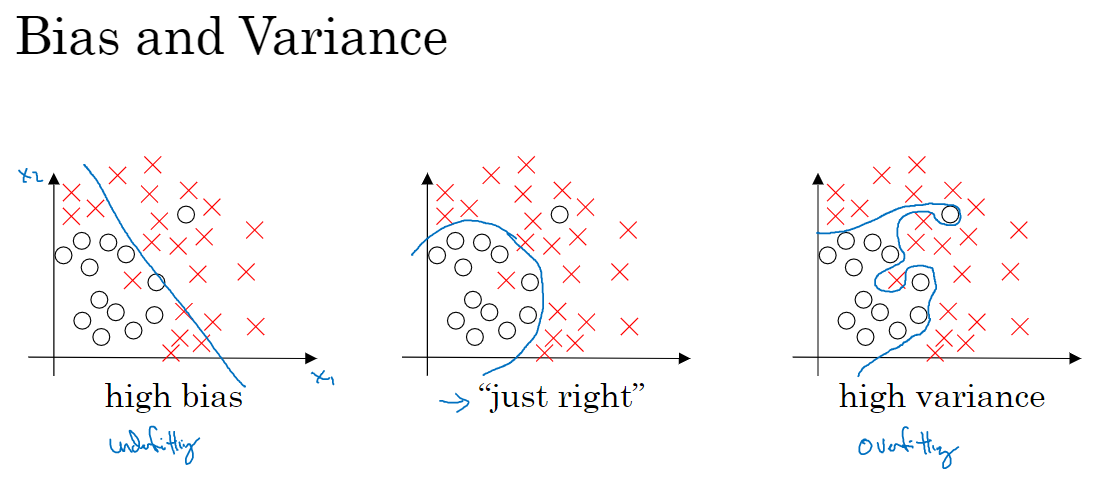
找一个和你导师也有兴趣的方向

训练集 验证集 测试集 60 20 20 %

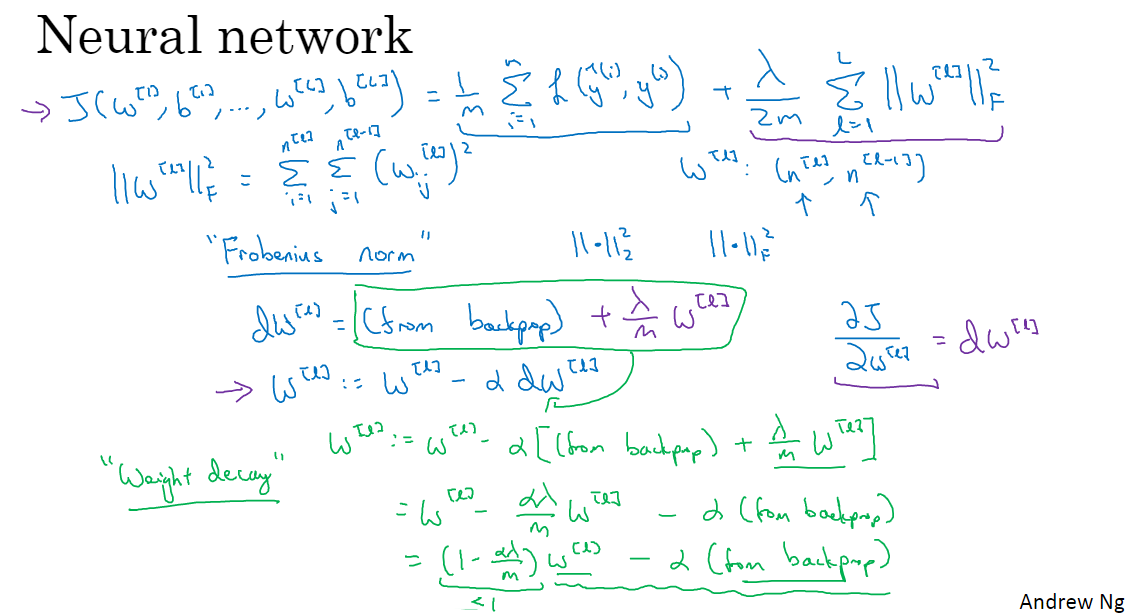
训练集错误 1% 15% 15% 0.5%

测试集错误 11% 16% 30% 1%

High variance high bias A&B good



正则化可以防止过度拟合



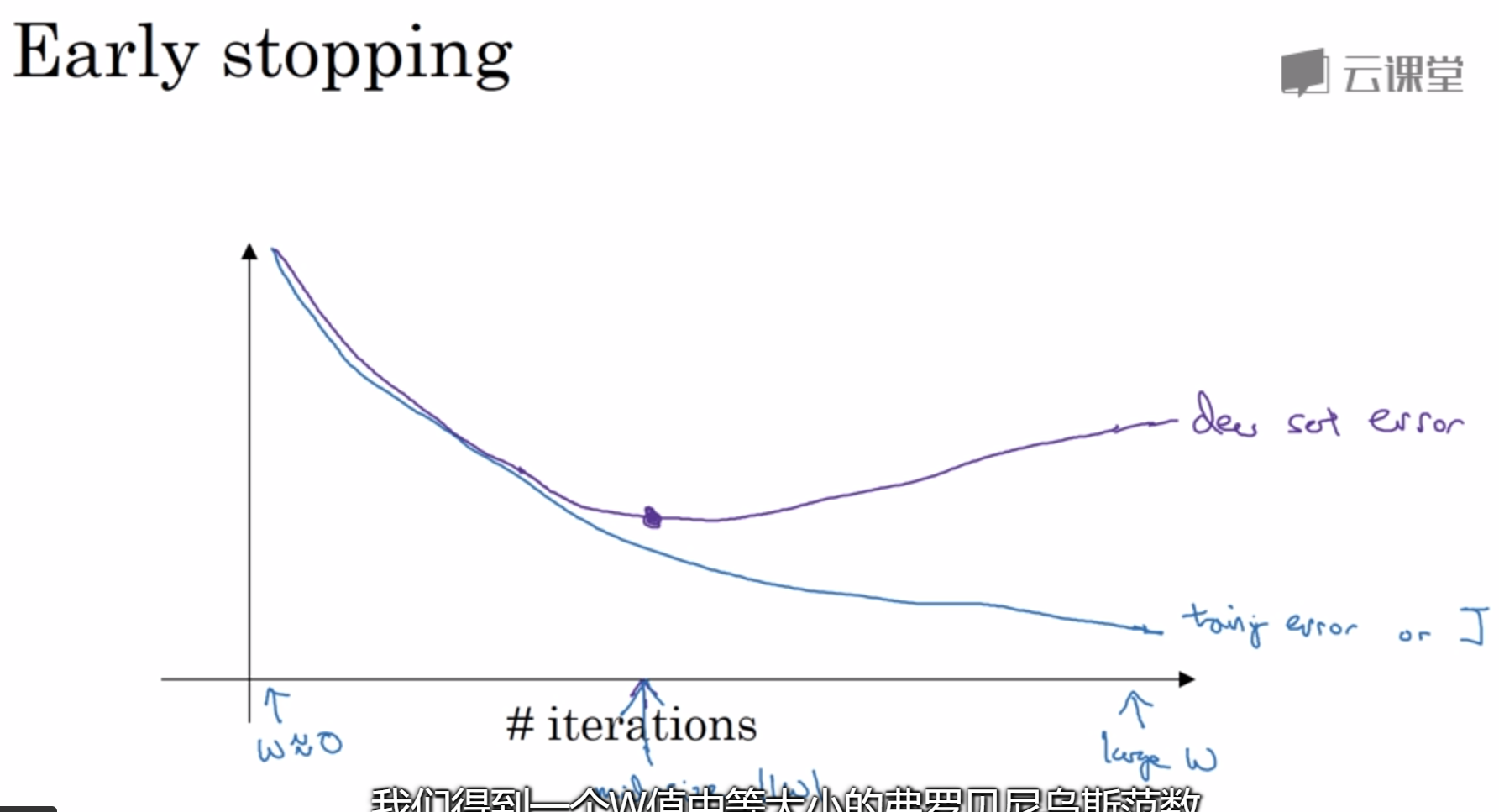
越大，结果可能效果越差因为z越小导致处于线性的阶段无法表达很复杂的形状

正则化后会随着迭代次数然后减少

Dropoout 正则化

每个节点保留和消除的概率都是0.5

停止时机

**

归一化可以让整个迭代区间像一个碗，而非归一化则可以让整个迭代区间像一个狭长的碗。

梯度消失和梯度爆炸，因为w被计算了很多次。

估算导数的值

Mini-batch 梯度下降法

处理完整个样本之前让梯度下降

随机梯度法 不会收敛会在最小值附近波动

指数加权平均

Momentum 下降法 速度分析法~

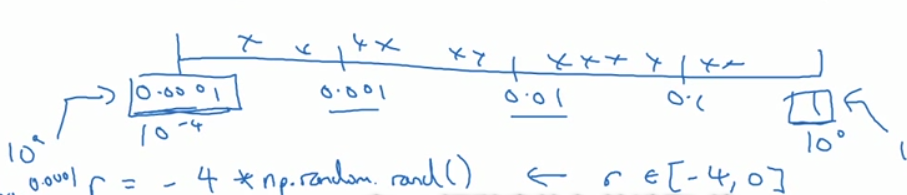
RMSprop表示的是一个一个维度上更新快一些另一个维度上更新慢一些 消除梯度下降中的摆动

Adam算法

学习率衰减

参数选取 不要网格选取，随机选取然后有一个粗糙到精细的过程

均匀选取参数

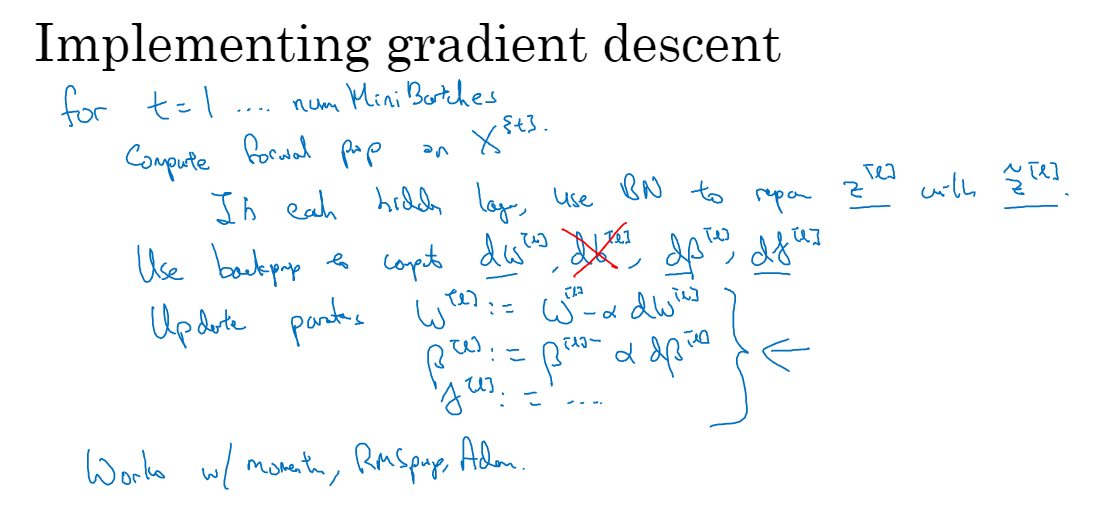


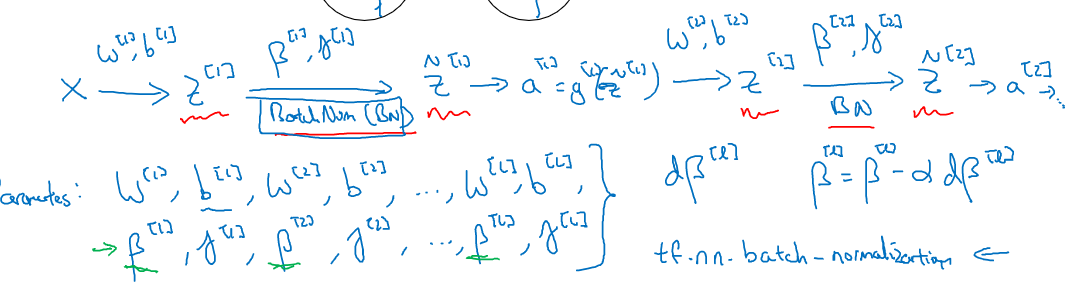
实践有两种方式

没有资源 经常更改模型

有资源 多条模型并发

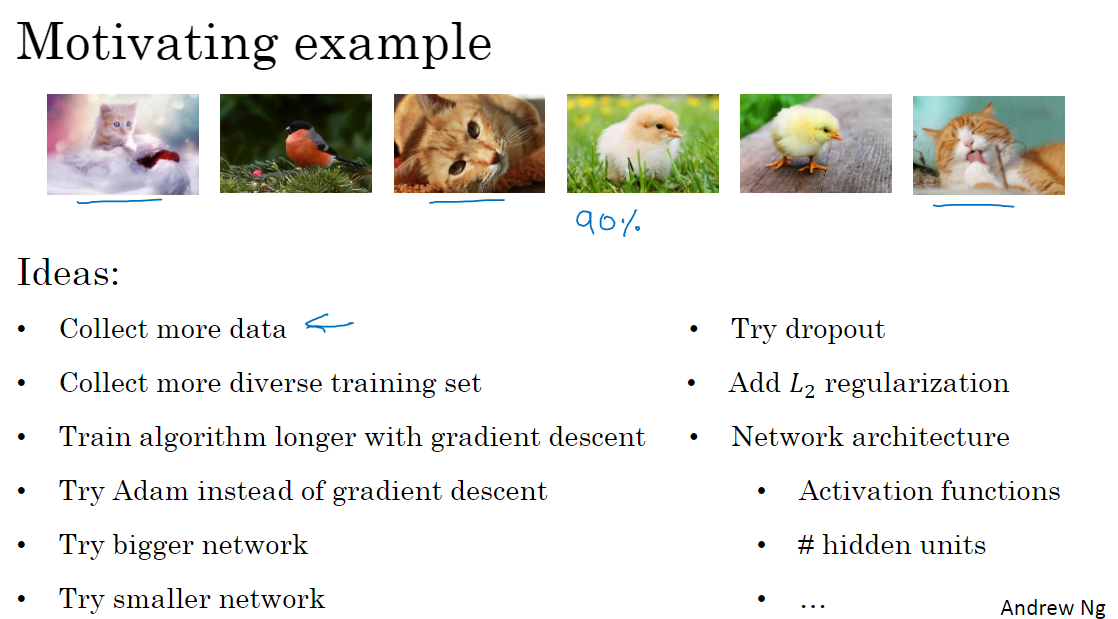
Batch 归一化，隐藏层应用归一化



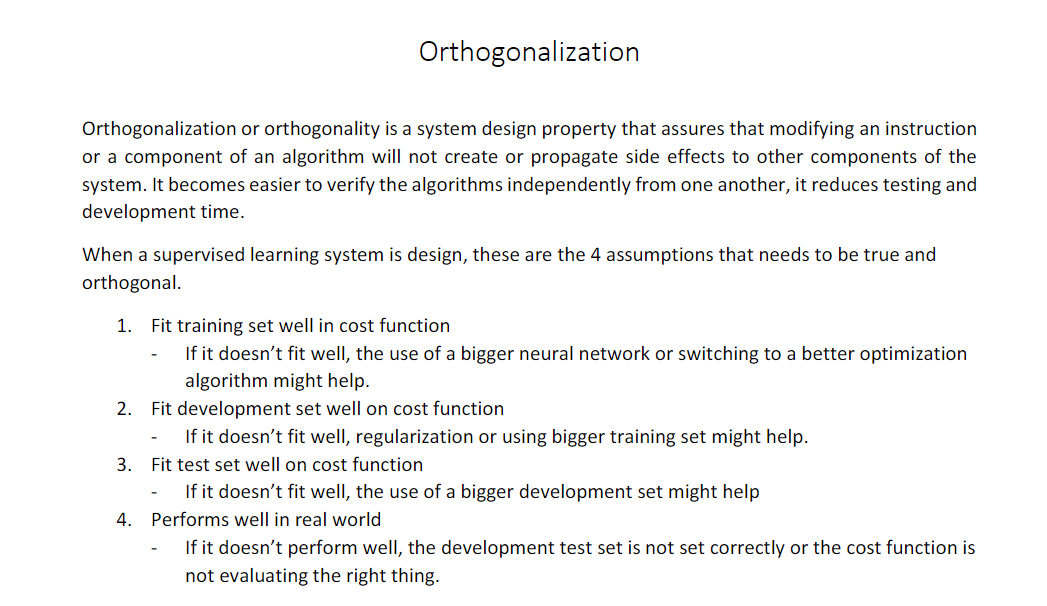


Softmax 从多个东西中找到某一个东西

Hard max 最大概率为1



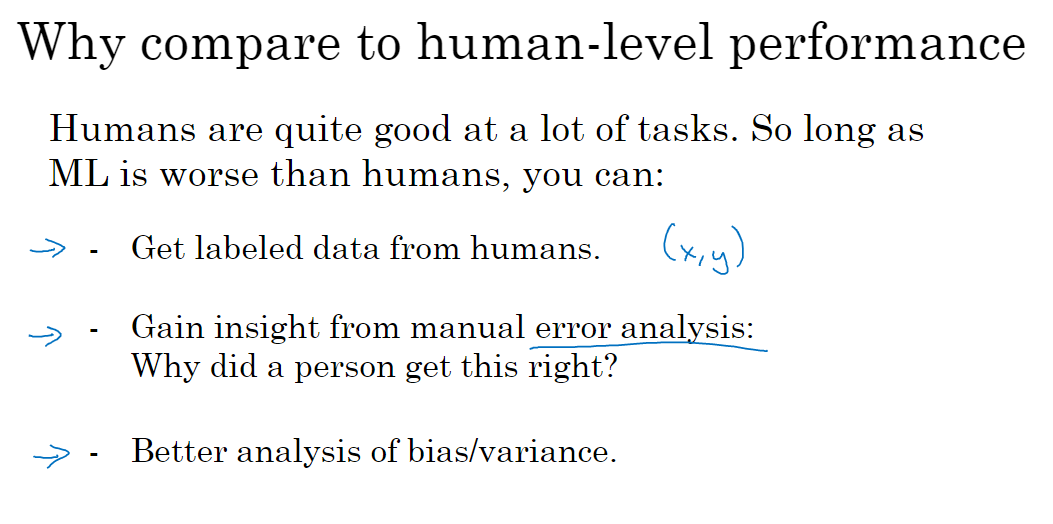
正交化 只调整一个

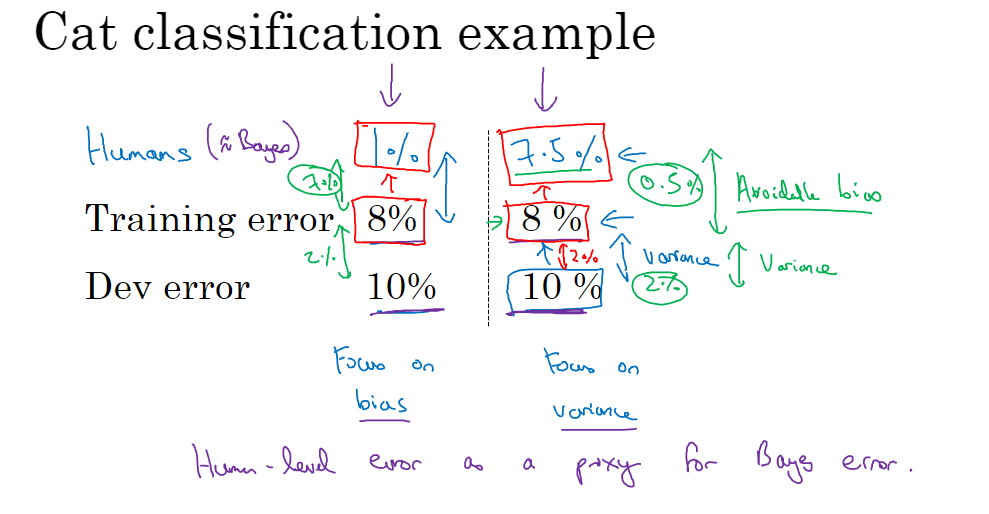


查全率 所有的猫识别出来多少

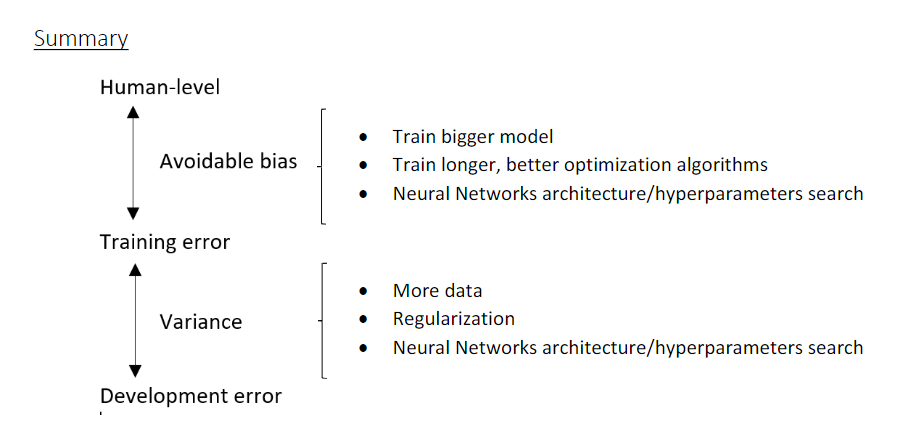
查真率 识别正确的猫是多少

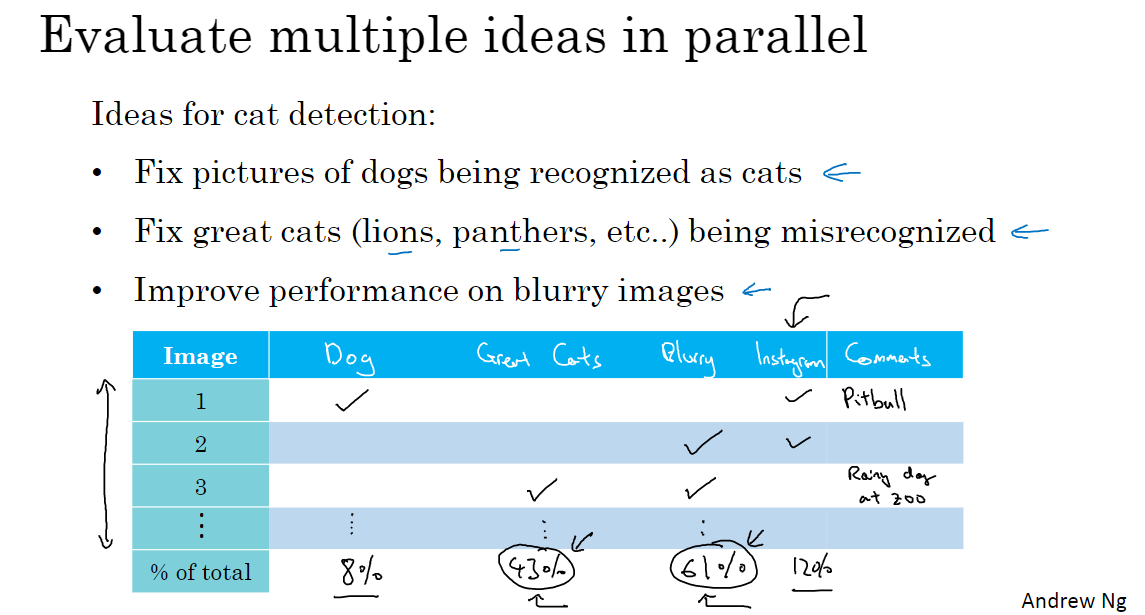
当算法能力未超越人类的时候



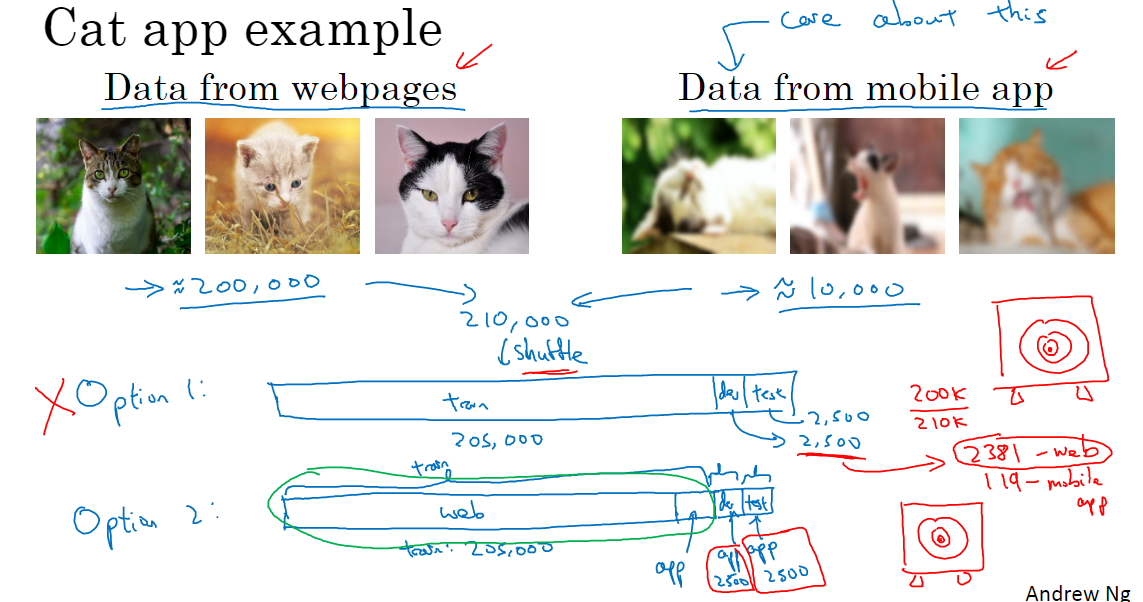


人类误差的上线 减少可避免误差还是减少 方差？

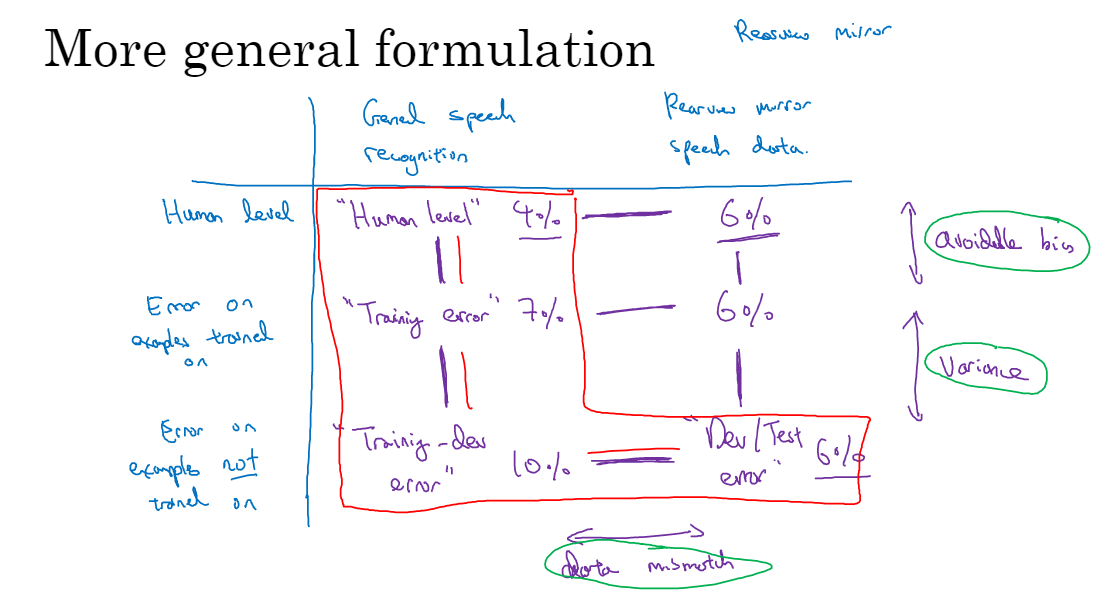




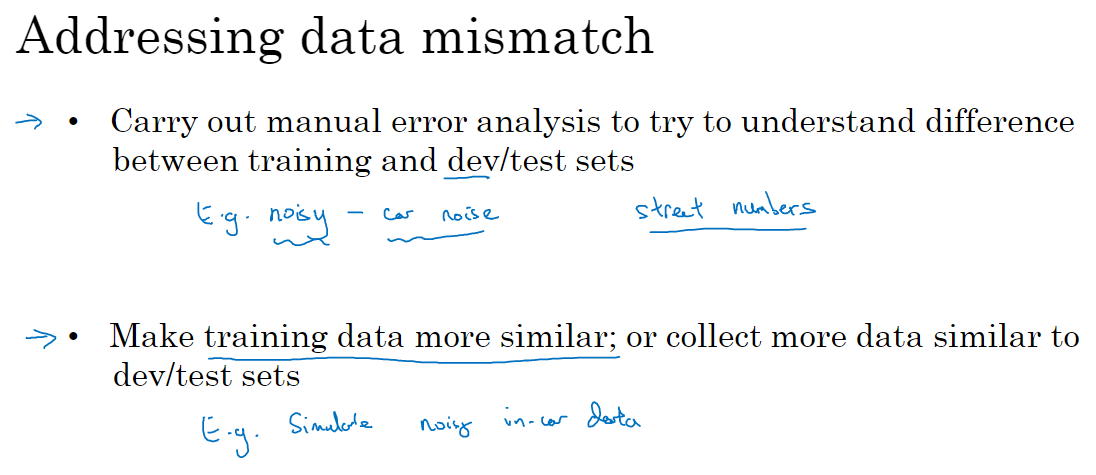
修正你的错误通过统计你的错误类型。

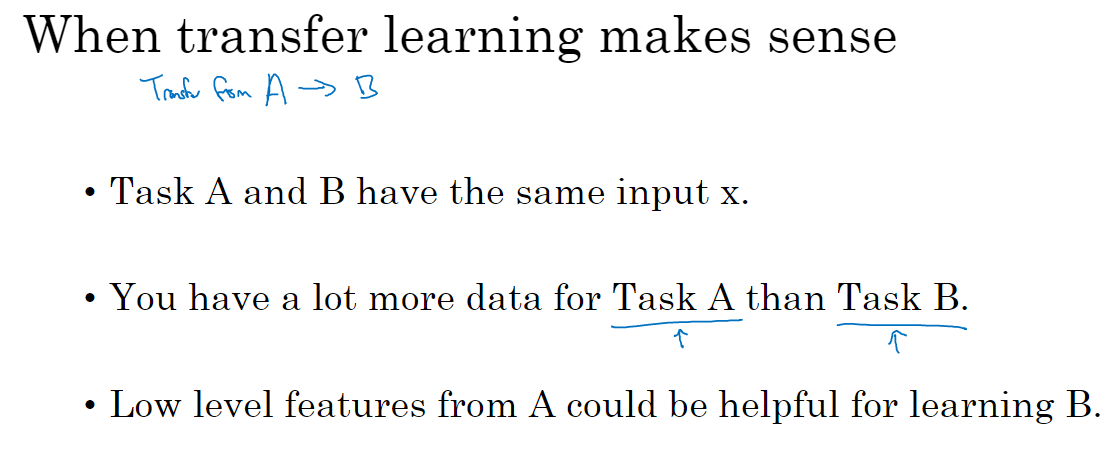


最好把特别的数据放在dev和test集合中

  
遇到数据不匹配的问题

让训练集和dev集和更加接近





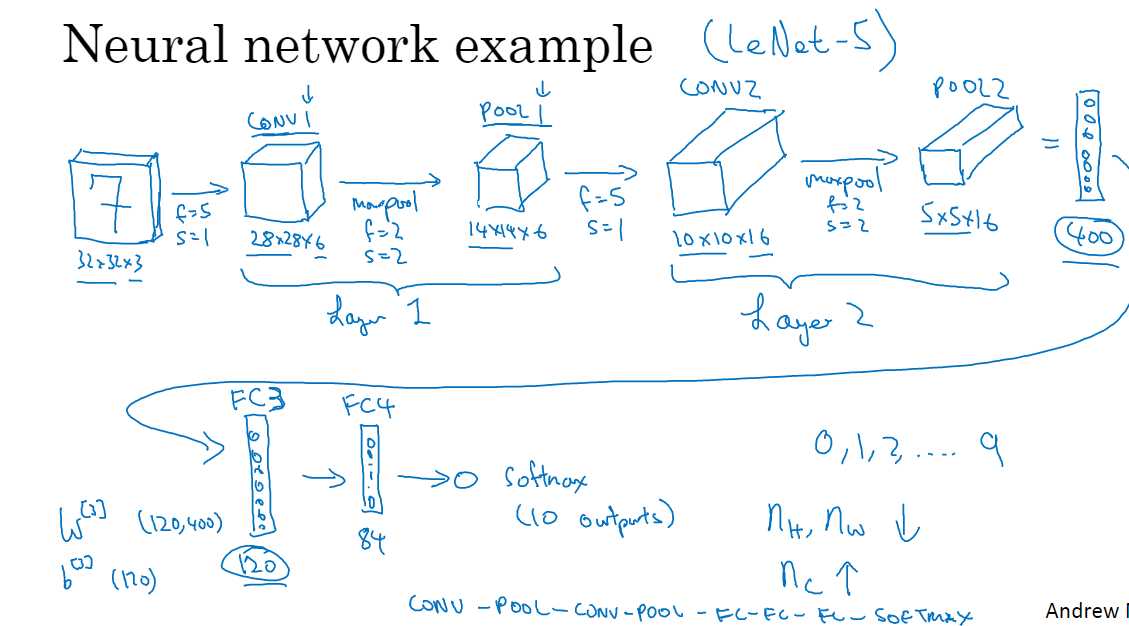
迁移学习

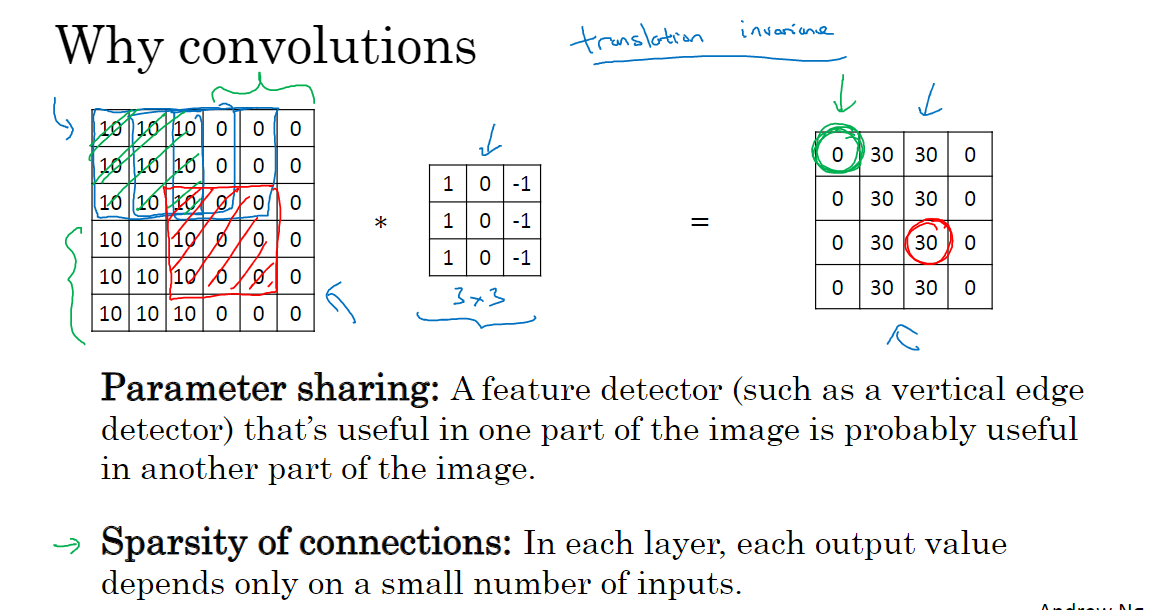
多任务学习就是对于停车问题你要同时检测好几个东西

端到端学习 比如翻译 不用经过MFCC之类的的东西

Cs231n

<https://zhuanlan.zhihu.com/p/125764650> 知乎问答

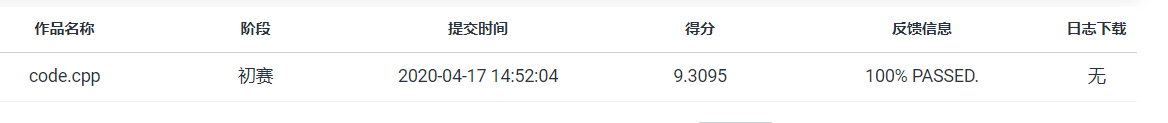




减少维度的方式

代码分析

hw57308748 谢津的代码排名



tarjan 复杂度O（n）的啊

不啊，我这边试了SCC，1.7到了1.4

我之前递归写了，又改成非递归…变成负优化

类似记忆化搜索的方案做baseline，就被各种叫大佬，还是很惶恐的。

具体来说，就是提前做深度为2的搜索（保存最后一层结点的入边），在第6层时直接根据现有结果进行判断，不进入第七层。

按照这个Naive的思路来进行优化，我们可以得到如下的解决方案：

（在线直接提交的Score为2.7左右）