**p59待写代码的时候实验**

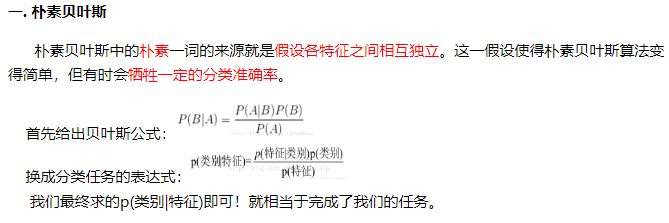
**knowledge**

专家系统=知识库+推理机、

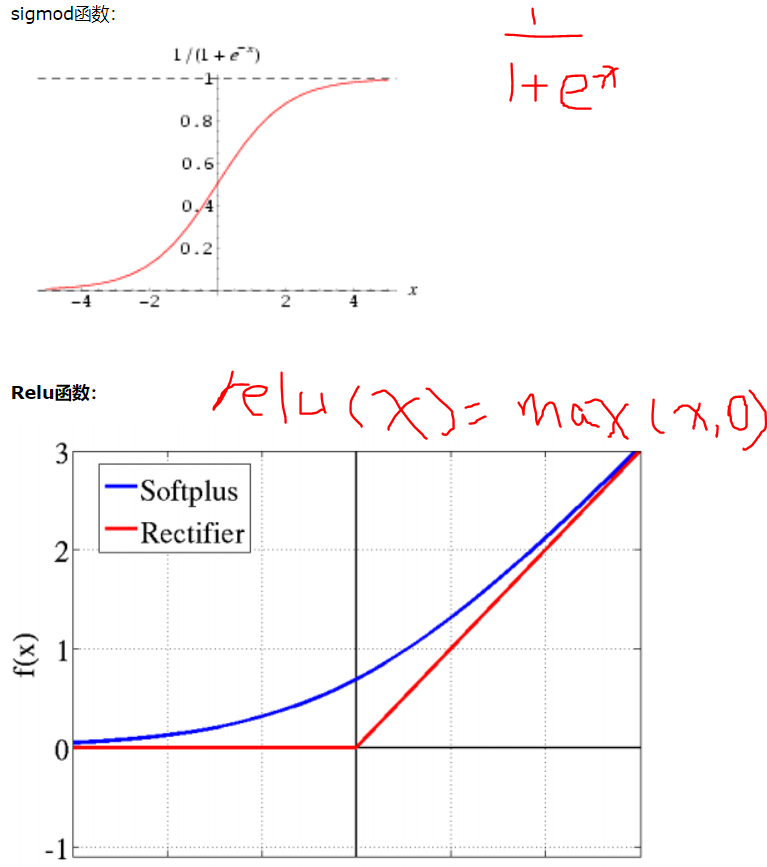
其它机器学习：概率建模（朴素贝叶斯，logistic 回归（属于分类算法，不是回归算法））

核方法（支持向量器SVM）,决策树（随机森林，梯度提升机）

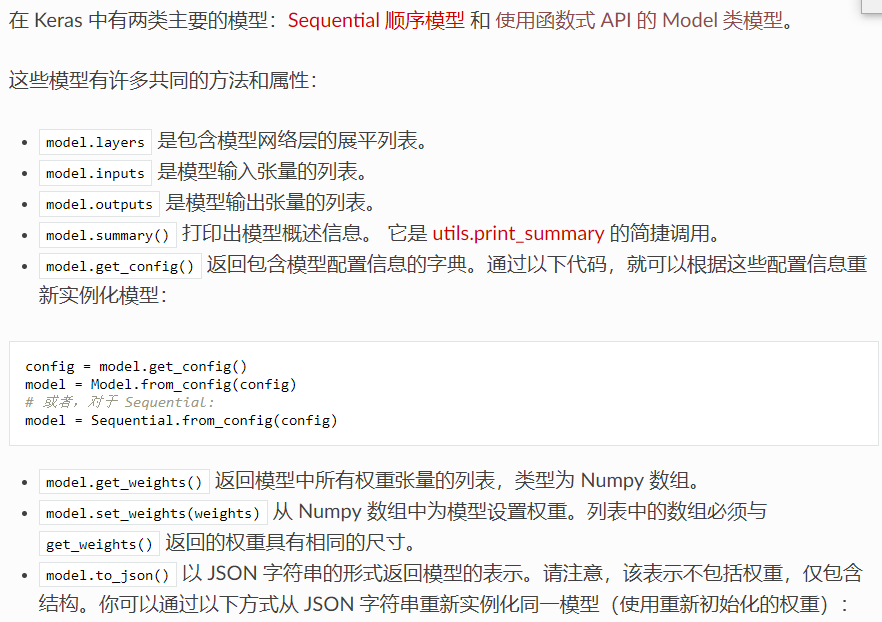
svm决策树处理结构化数据，DL感知问题



-------------------------------------------------------------------------------



-------------------------------------------------------------------------------



-----------------------------------------------------------------------------

第三章的实例：

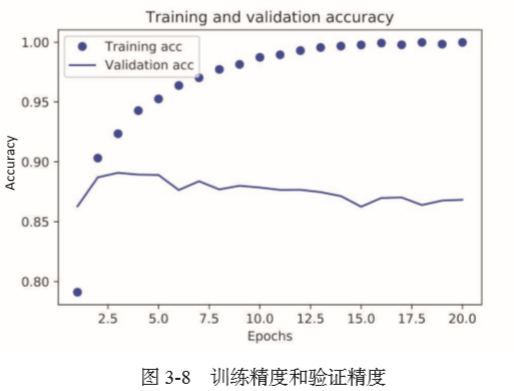
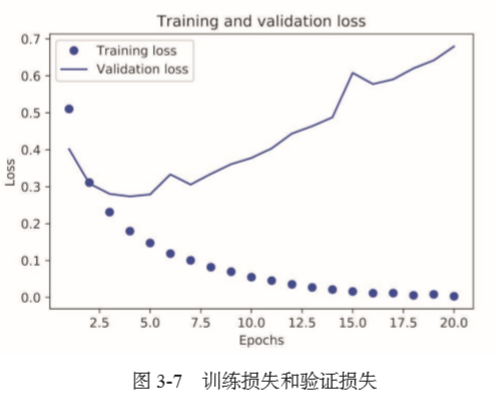
1..电影评论，二分类问题、

总结：二分类问题（两类输出），最后一层，一个单元用sigmoid(0~1)

用 binary\_crossentropy 损失函数

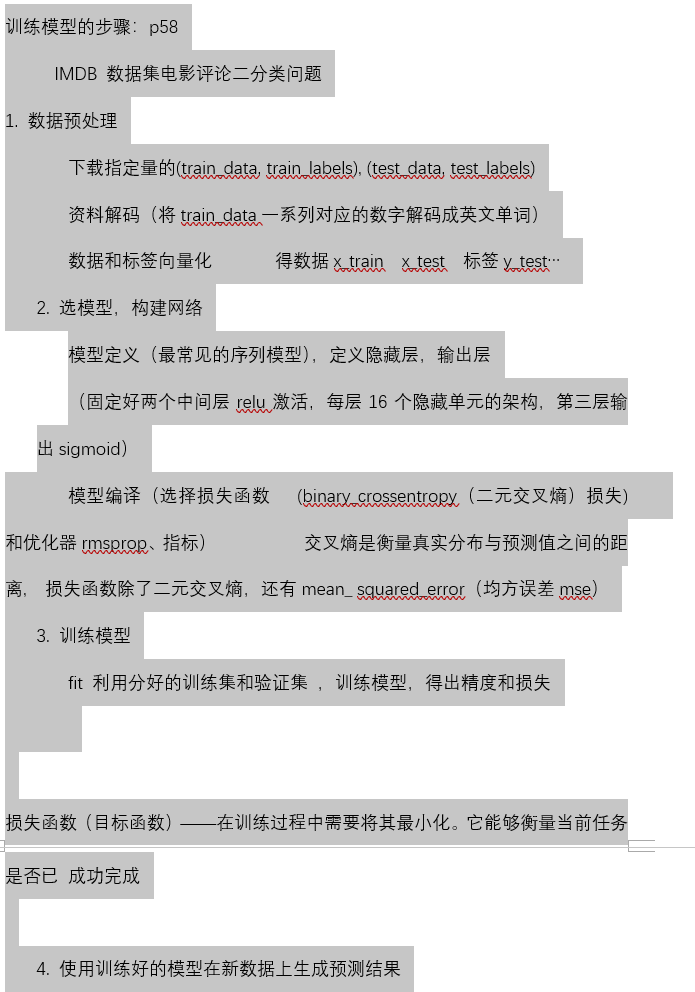
rmsprop 优化器

分析：



左点 训练损失 不断降低， 右点 训练精度 不断上升 --达到预期

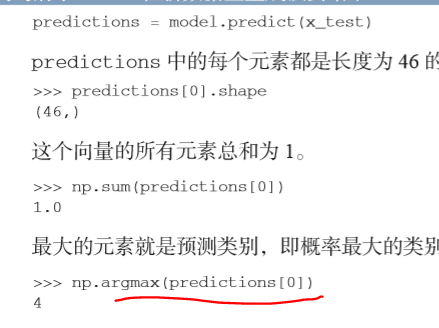
左线 验证损失（最低） 和右线 验证精度 （最高）在第四轮达到最佳值 --过拟合，将fit训练中的 epochs = 20 训练轮次 改成4



2.新闻分类，多分类问题

步骤大致同上，区别是

最后的预测不是为正的概率，而是max最大的



最后一层sigmoid -> softmax激活函数

预处理对标签编码:

one-hot（分类编码）-> categorical\_ crossentropy 作为损失函数

将标签编码为整数，使用 sparse\_categorical\_crossentropy损失函数

3.标量回归问题

连续房价回归预测

连续的数据若范围波动大要做预处理：标准化

网络的最后一层不用激活函数，若用sigmoid，只能0~1范围输出，这里最后一层纯线性，范围任意！

这里不再像上面两个分类问题一样，留出10000的验证集

损失函数：mse

监控新指标：而不是精度

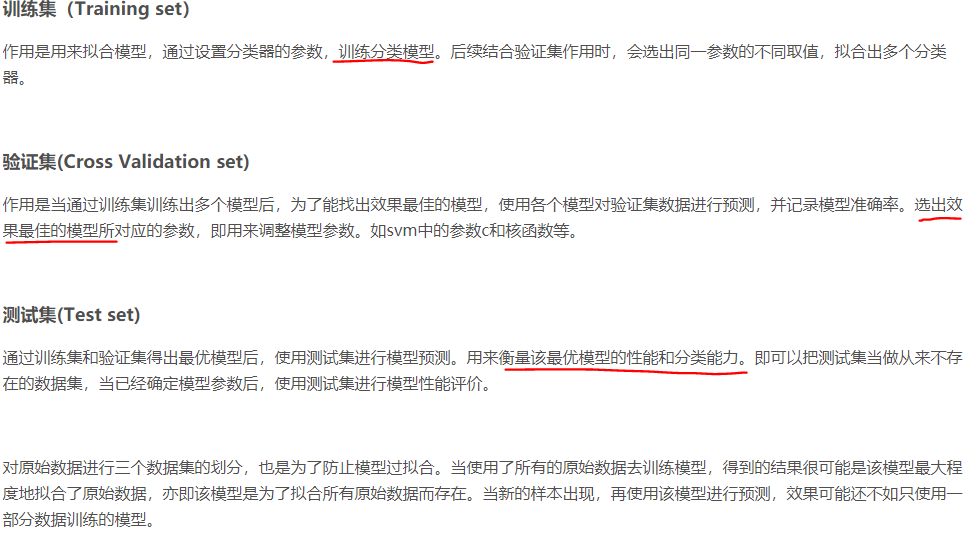
可用数据少，避免过拟合就减少隐藏层，并且K折验证（K-1训练，剩余一个验证）

分类



回归





---------------------------------------------------------------------

机器学习四个分支： --------------难点：解决过拟合，目的：泛化模型

·根本问题：优化和泛化之间的对立

监督学习（二分类，多分类，标量回归）

无监督学习：降维和聚类

自监督

强化学习

-------------------------------------------------------------------------------

降低过拟合的方法：（正则化）

获取更多的训练数据

减小网络大小

添加权重正则化（奥卡姆剃刀（Occam’s razor）原理：如果一件事情有两种解释，那么最可能正 确的解释就是最简单的那个，即假设更少的那个）。强制让模型权重去最小的值。L1 正则化（L1 regularization）：添加的成本与权重系数的绝对值成正比。

L2 正则化（L2 regularization）：添加的成本与权重系数的平方（权重的 L2 范数）成正比。 神经网络的 L2 正则化也叫权重衰减（0.001 \* weight\_ coefficient\_value）

dropout正则化。训练过程中随机将该层的一些输出特征舍 弃（设置为0）。

-------------------------------------------------------------------------------

评估方法： 复杂度依次递增

留出验证集（数据很大时采用）

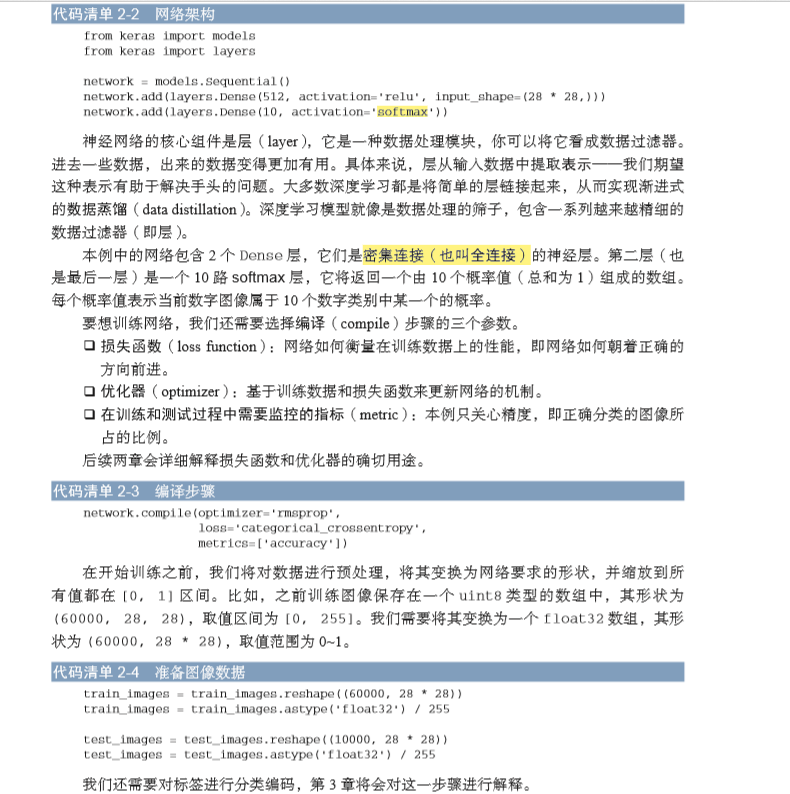
K 折交叉验证。

 重复的 K 折验

-------------------------------------------------------------------------------

卷积神经网络

密集连接网络和convent对比



**Q&A1**

1. 假设空间：备选的、预先定义好的可能性空间。有限、错误

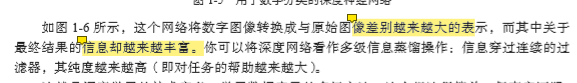
假设空间是很大的，我们最开始的权重，就像是假设空间里面的一个随机假设，然后更新这些值，就相当于我们在假设空间里调整我们的假设，让它更合理，更针对我们的任务

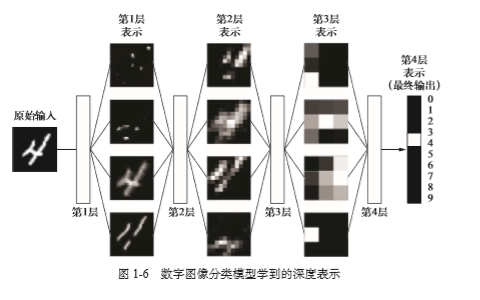
解空间里面找解的意思，解空间里面有很多解

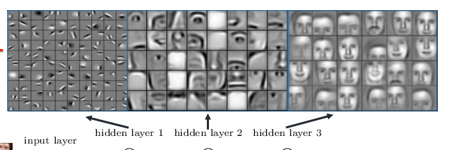
但是这个假设空间到底能怎样，是跟你的网络结构，有关系。

假设空间里，我们都不一定能得到最优

2. 信息量越大代表什么



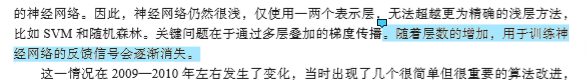




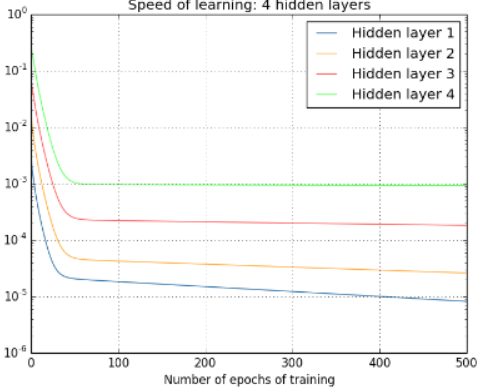
不同层级的网络，会对信息给出不同级别的抽象

一般来说，越底层，抽出来的信息，就越具有一些通用性，图像领域是这样，或者看elmo那种，不同层的结果，所蕴含的信息也都是不一样的

关于“层数”的概念，第六页。层数越深，越抽象，越通用，适用对象也不同，传达信息不同

3. 反馈信号逐渐消失，是指信息越来越丰富，纯度越来越高，反馈变少？

梯度弥散，不是过拟合

靠近输出层（layer4）的hidden layer 梯度大，参数更新快，所以很快就会收敛；

而靠近输入层的hidden layer 梯度小，参数更新慢，几乎就和初始状态一样，随机分布。

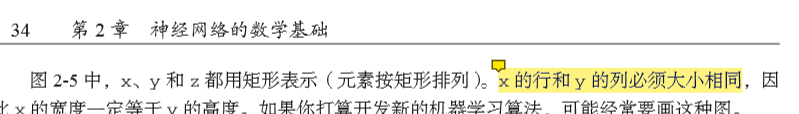
在上面的四层隐藏层网络结构中，第一层比第四层慢了接近100倍！！

这种现象就是梯度弥散（vanishing gradient problem）

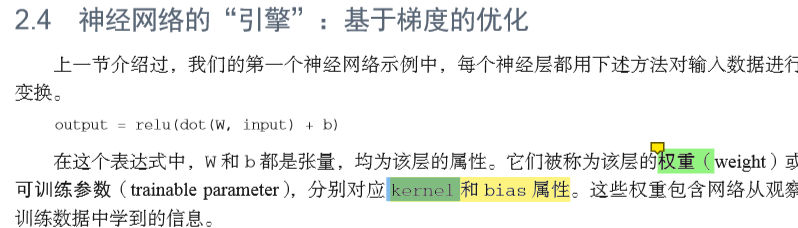
根据sigmoid的特点，它会将+∞～-∞之间的输入压缩到0～1之间。当input的值更新时，output会有很小的更新。

又因为上一层的输出将作为后一层的输入，而输出经过sigmoid后更新速率会逐步衰减，直到输出层只会有微乎其微的更新，收敛过快。

4.莫得问题



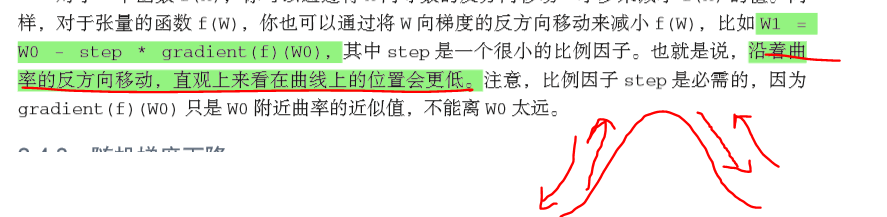
5. p36



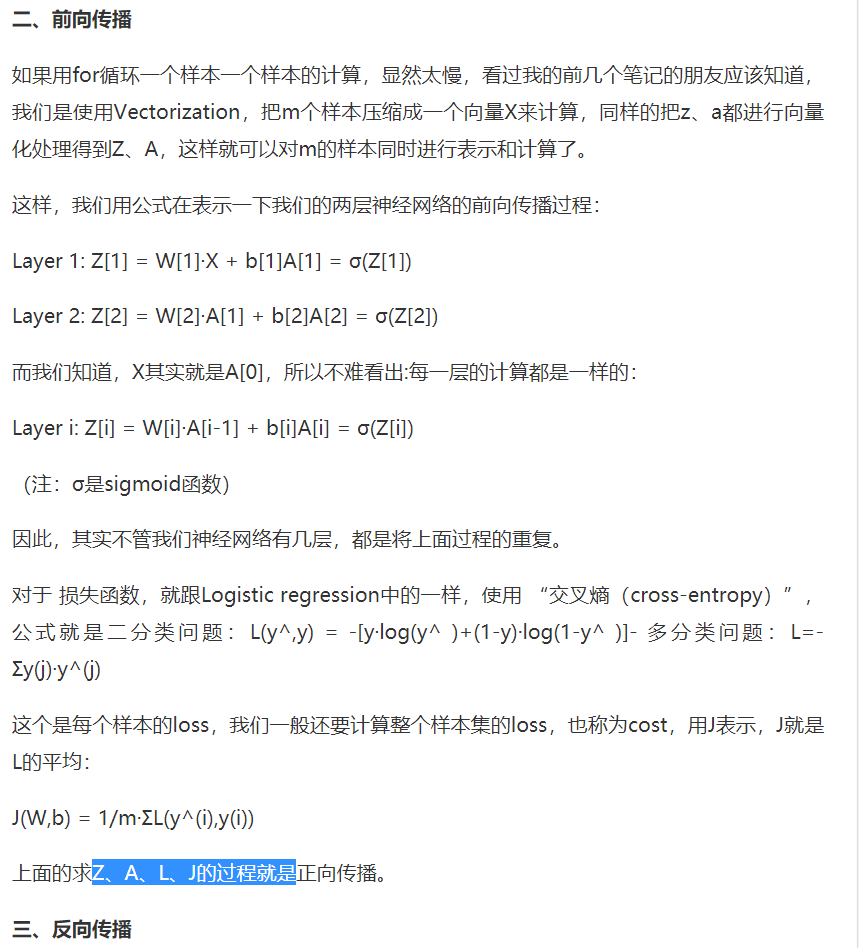
应 kernel 和 bias 属性 是什么,bias 好像能理解

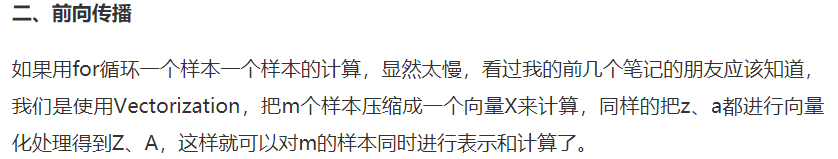
w、b 都是权重，wx+b w---kernel, b---bias

6. p38 反方向的反方向，反正就是下降就对了



7. 反向传播好像还不是正向传播的逆过程，，就这样一个合并过程，就是正向了？怎么提现正向？



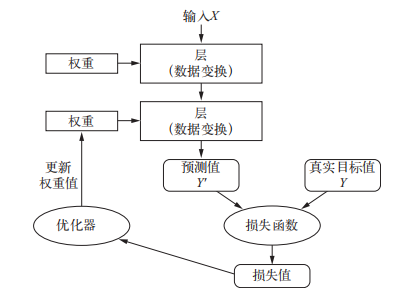


反向传播和正向传播，发现两者好像还不是互逆关系

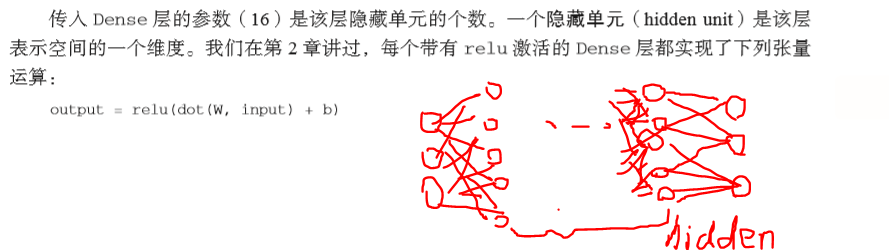
不是 昨天开会提到的 单向和双向的意思

这里是训练的时候的正向和反向，表示正向运算和反向计算梯度，用于更新参数

昨天的那个双向，是网络结构。是层的双向就是wx+b这样计算，根据你定义的层结构

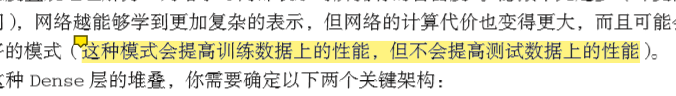
 从输入，到损失值的过程叫正向

8.



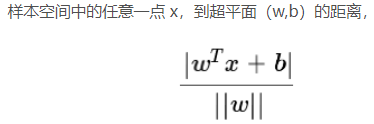
中间的一个隐藏层的小圈圈 是 一个 隐藏单元

9. 过拟合



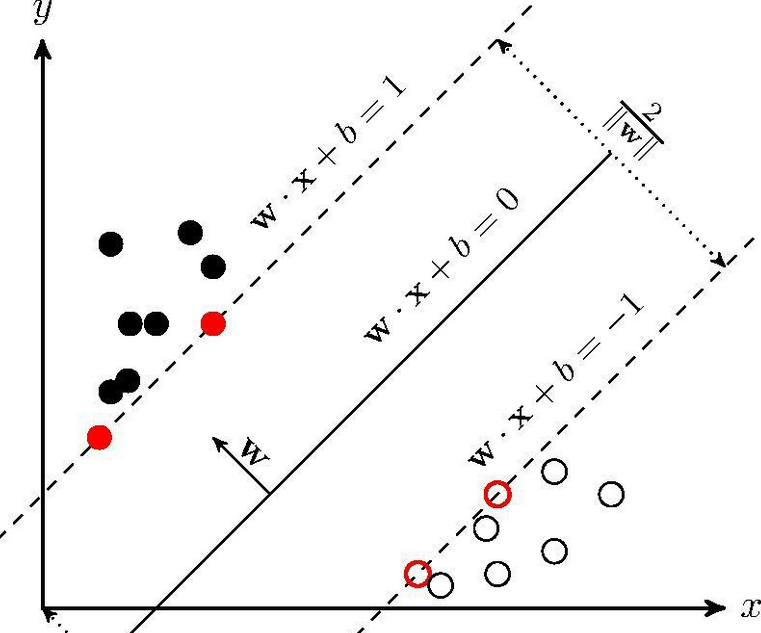
**Q&A2**

1. p13 . 



为什么距离最大而不是最小呢？那我想一种极端情况



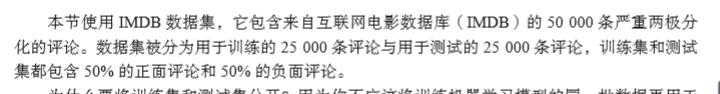
在能分对大多数的情况下，距离最大化

SVM支持向量机，让这条线，在两个类别数据的中间

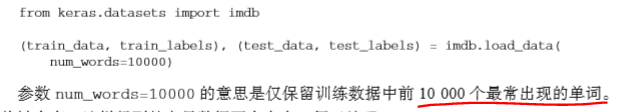
2.

p51数据集的关系

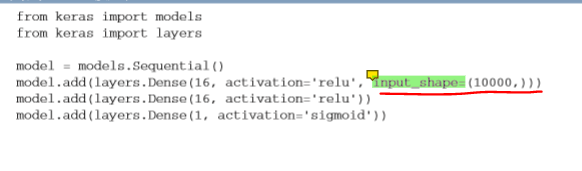
共50000条数据，但是25000对半分，和后面的10000条前，10000条后怎么区分



下载数据时，涉及到10000个词的有关信息



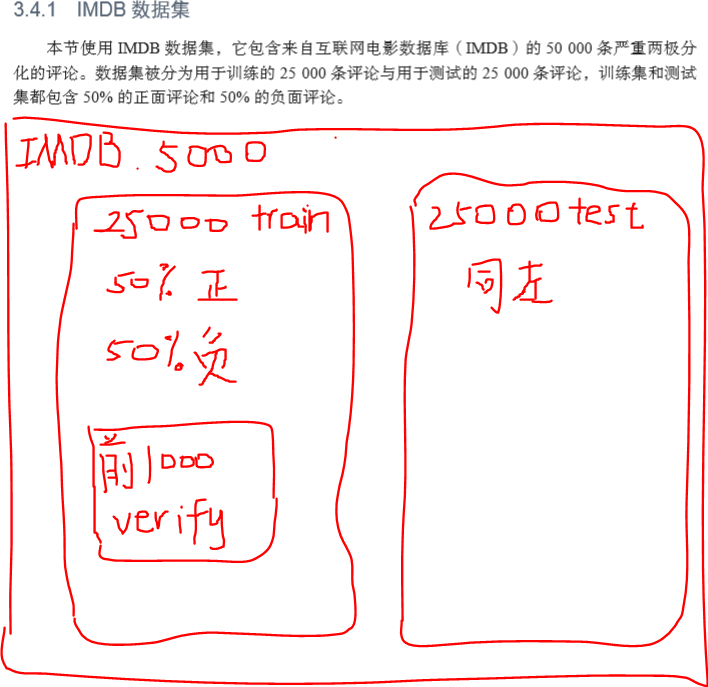
说明涉及到10000个词，留出10000个词的空间



训练集的25000条中留出10000条验证



总结上述过程，数据集的关系



3. p58

重新训练一个模型，没有验证集，而上面的没有测试集



不是完整的，也能做

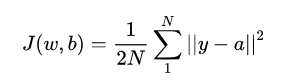
4.

p58从头开始训练一个模型，model.evaluate返回准确度和损失，损失，讲道理应该有些大，但这里却是百分数，是不是经过sigmoid 的结果，

还是evaluate的？不会是evaluate的，它只返回损失和精度

结果损失函数长什么样子，？做差求和，很大！

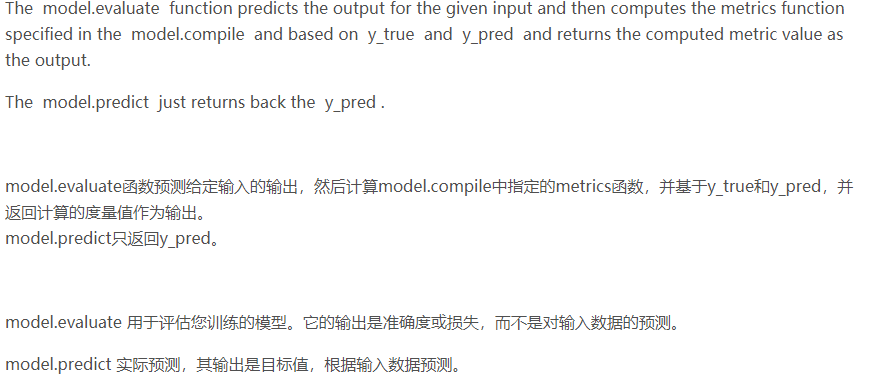
损失函数的形式：

,损失不一定<1,但是越小越好，0代表拟合的很好

**精度=真正例/（真正例+假正例）**

5.

p59 model.predict(x\_test) 为什么返回的是对结果预测为正的确信度？而不是下面截屏的语句里面 的 预测值的目标值比如0 /1



因为这里的sigmoid函数，把所有的数值，映射到0-1之间

predict就是输出0~1的概率值

这里进行判断的时候，是拿阈值进行计算的，小于阈值，分为1类，大于等于阈值，分为1类

阈值，我们是可以自己调整的

阈值是不作为参数参与模型训练的

训练的时候，它默认好像是取0.5？好像是，但是最终出结果的时候，我们可以自己设定

训练的时候，我们也可以自己写评估矩阵，然后设定阈值

理解后，为什么不是直接预测01，而是直接暑促概率，不是很直观？

6. p64 0.9损失精度这么高？ 嗯呢，有可能超过1的呢

7. 但在这个例子中，完全随机的精度约为19%

统计功效的概念

这个统计功效

就是说，你的模型，比瞎猜要好

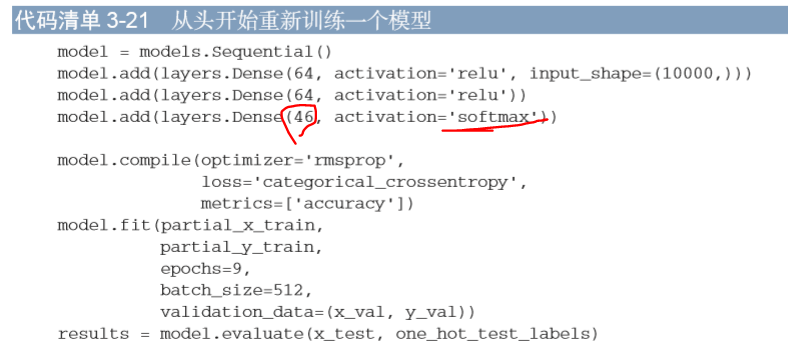
瞎猜，就是纯随机基准，比如二分类，随机猜就是0.5的正确率，只要模型比0.5好，那就说明这个模型可以

或者可以更细化，比如数据里面正负数据不均衡，这个瞎猜的概率可能就要调整，并非绝对的数值

46个类别，但是还可能数据不均衡，随机猜出来就是19%，也很正常

这个19%就是46个类别的数据不均衡

8. p64路透社新闻主题多分类数据集





46类 所以46个单元，但是上面的2类怎么一个单元呢？

这里有讲究

对于类别大于2的单标签分类，就用softmax

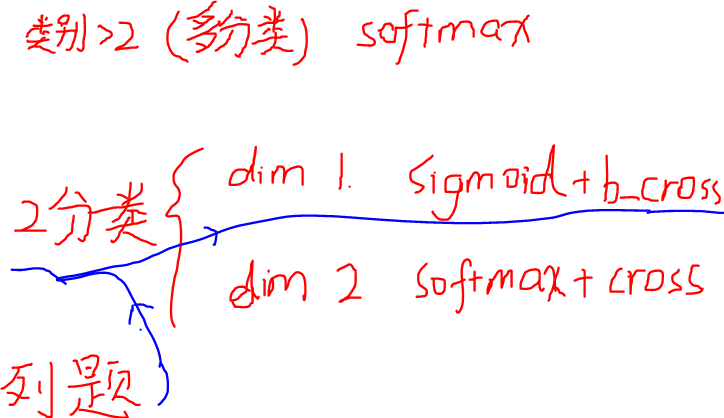
对于2分类任务，如果输出维度是1，那么要是用sigmoid激活函数，对应的损失函数就是binary\_crossentropy

同样，你也可以写输出维度是2，但是激活函数需要softmax，损失函数要用多分类的crossentropy

为啥2分类可以使用sigmoid的那个呢，是因为，我们可以控制阈值，softmax的预测，我们一般不会设置阈值，都是直接argmax

大概就是，二分类可以写成多分类的样子，但是还可以有自己独特的方式

自己独特的方式，可以更细腻一点



固定搭配

9. p65 - 66 减小维度，控制变量法，训练的批量和次数怎么能变呢

可能就是为了体现效果吧

我就是觉得这个epochs 和batch\_size对精度的影响很大

觉得应该要一样才能对比

是的，确实如此

不过就看效果而言，都还好

大的范围，理论上不会波动特别大的的吧，至少我接触的任务，波动不会太大

10

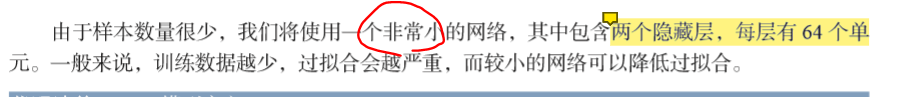
p68 非常小，对比前面分类的网络，不小啊、还是说，这里在同等阶级的回归问题里面，网络算小的

是因为样本数量非常小，包括你前面看到的网络，都特别小

你想你自己的电脑，训练起来都没问题，而且速度也还行，内存也吃不了多少，大的模型，那天我讲的那个bert，我把batch调到128，文本长度110，差不多要吃到32G

11

p68实例化模型是什么意思



3.6.4 前面也可以用

相当于，创建对象

类的实例化

这个就是面向对象的思想吧，封装便于多次构建模型

12

p78为什么在验证集上调整好参数后 都要从头在训练集上开始训练模型

为啥书上的几个例子，都是调整好参数，也训练好了后，

还要重新训练模型呢？

13.

什么叫模型性能变化很大

就是可能分K折，就是k个模型对吧

k个模型的性能差异，可能会很大，比如有的是0.9，有的是0.6

相当于每次取一块，作为测试集，取其余部分，作为训练集

然后得训练模型啊

那么每次划分，都需要训练模型

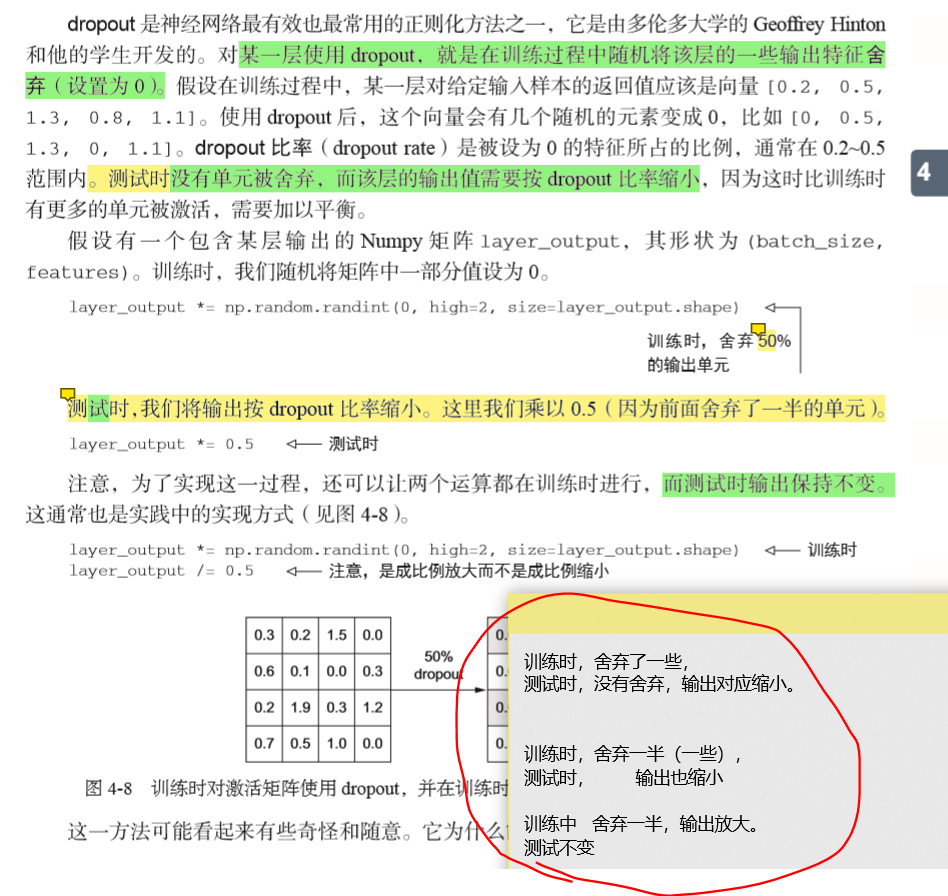
为什么模型性能变化大，就用K折验证好呢？

因为K折更能衡量整个数据集的效果

如果你不用K折，你运气好，选到了正确率炒鸡高的部分，作为验证集，你还洋洋得意，但是真实的结果，可能会给你惨痛一击

14.p87

因为之前出现了过拟合，它在找过拟合的点



Lunatic:

测试时候，不会使用dropout的

Lunatic:

所以，我觉得你理解的没啥问题