

yxwkaifa

知识改变命运

博客园 首页 新随笔 联系 订阅 管理

< 2018年10月 >

日	一	二	三	四	五	六
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

搜索

找找看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

友情链接

i云app使用介绍

国民重疾险上市——复星联合康乐e生

买“尊享e生”之前，一定要搞明白这个问题

最新评论

1. Re:嵌入式系统开发步骤

书面上的东西是很重要但是我感觉对于一大部分朋友来说还是不够直观，因此找到一些不错的视频资料简单易懂一些

--华清粉丝1号

2. Re:Mozilla5.0的含义

笑尿

--Pale Life

3. Re:做高通平台安卓驱动感言

刚开始做高通 顶起

--duantao7584567

4. Re:.net web 开发平台- 表单设计器 一(web版)

老大 能不能发一份源码给我 谢谢

351597500@qq.com

--王涌军

5. Re:Basic脚本解释器移植到STM32

可以考虑移植我的一个项目 MY-BASIC。

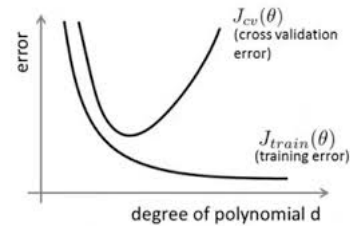
--扑来树袋熊

## 正则化方法：L1和L2 regularization、数据集扩增、dropout

本文是《[Neural networks and deep learning](#)》概览 中第三章的一部分。讲机器学习/深度学习算法中经常使用的正则化方法。（本文会不断补充）

## 正则化方法：防止过拟合，提高泛化能力

在训练数据不够多时，或者overtraining时，经常会导致overfitting（过拟合）。其直观的表现例如下图所看到的。随着训练过程的进行，模型复杂度添加，在training data上的error渐渐减小。可是在验证集上的error却反而渐渐增大——由于训练出来的网络过拟合了训练集，对训练集外的数据却不work。



为了防止overfitting。能够用的方法有非常多，下文就将以此展开。有一个概念须要先说明，在机器学习算法中，我们经常将原始数据集分为三部分：training data、validation data、testing data。这个validation data是什么？它事实上就是用来避免过拟合的。在训练过程中。我们通经常使用它来确定一些超参数（比方依据validation data上的accuracy来确定early stopping的epoch大小、依据validation data确定learning rate等等）。那为啥不直接在testing data上做这些呢？由于假设在testing data做这些，那么随着训练的进行，我们的网络实际上就是在一点一点地overfitting我们的testing data，导致最后得到的testing accuracy没有不论什么参考意义。因此，training data的作用是计算梯度更新权重，validation data如上所述。testing data则给出一个accuracy以推断网络的好坏。

避免过拟合的方法有非常多：early stopping、数据集扩增（Data augmentation）、正则化（Regularization）包含L1、L2（L2 regularization也叫weight decay），dropout。

## L2 regularization（权重衰减）

L2正则化就是在代价函数后面再加上一个正则化项：

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2;$$

C0代表原始的代价函数，后面那一项就是L2正则化项。它是这样来的：全部参数w的平方的和，除以训练集的样本大小n。

λ就是正则项系数，权衡正则项与C0项的比重。另外另一个系数1/2，1/2经常会看到，主要是为了后面求导的结果方便，后面那一项求导会产生一个2。与1/2相乘刚好凑整。

L2正则化项是怎么避免overfitting的呢？我们推导一下看看，先求导：

$$\begin{aligned} \frac{\partial C}{\partial w} &= \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} w \\ \frac{\partial C}{\partial b} &= \frac{\partial C_0}{\partial b} \end{aligned}$$

能够发现L2正则化项对b的更新没有影响，可是对于w的更新有影响：

$$\begin{aligned} w &\rightarrow w - \eta \frac{\partial C_0}{\partial w} - \frac{\eta \lambda}{n} w \\ &= \left(1 - \frac{\eta \lambda}{n}\right) w - \eta \frac{\partial C_0}{\partial w} \end{aligned}$$

在不使用L2正则化时。求导结果中w前系数为1，如今w前面系数为  $1 - \eta \lambda / n$ ，由于η、λ、n都是正的。所以  $1 - \eta \lambda / n$  小于1，它的效果是减小w。这也就是权重衰减（weight decay）的由来。

当然考虑到后面的导数项，w终于的值可能增大也可能减小。

另外。须要提一下，对于基于mini-batch的随机梯度下降，w和b更新的公式跟上面给出的有点不同：



Swiss. Embedded. Computing.

NXP

i.MX 6ULL



Colibri  
计算机模块

高性能，超高效

板载 WiFi/蓝牙

工业级温度 -30° to +85°C

知道更多>



阅读排行榜

- 汇报措辞：你懂得如何向领导汇报吗（审阅、审批、审阅、批示、查阅）？(51541)
- Mybatis分页插件PageHelper正确的用法（网上有2篇不够科学的文章）(16295)
- 最简单也最难——怎样获取到Android控件的高度(12151)
- Microsoft Visual C++ Runtime Library Runtime Error的解决的方法(10682)
- Nessus漏洞扫描教程之配置Nessus(8167)
- 正则化方法：L1和L2 regularization、数据集扩增、dropout(8079)
- 不可错过的手机APP常见8种界面导航样式(5864)
- 彩色图像--伪彩处理 灰度图转伪彩色图像(5728)
- 【VBA研究】怎样将单元格数据赋给数组(5365)
- sql server 2008如何导入mdf，ldf文件(5130)
- Matlab的parfor并行编程(5075)
- 漫谈机器学习经典算法—特征提取与特征选择(4837)
- HashMap的存储结构及原理(4307)
- 主流芯片解决方案Ambarella的高清网络摄像机、德州仪器和控制海思(4242)
- Android音频底层调试-基于tinyalsa(4029)
- W5500问题集锦(二)(3911)
- 使用IntelliJ Idea生成可执行文件jar，开关exe文件步骤(3863)
- 再谈spring jdbc 连接池断开重连设置(3813)
- 【PLSQL】package包的使用(3765)
- ListView嵌套GridView显示不完整的解决方案(3746)
- Mozilla 5.0的含义(3481)
- java项目导出为一个可执行文件jar包(3438)

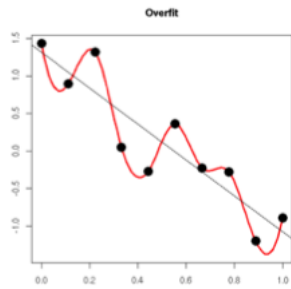
$$w \rightarrow \left(1 - \frac{\eta \lambda}{n}\right) w - \frac{\eta}{m} \sum_x \frac{\partial C_x}{\partial w}$$
$$b \rightarrow b - \frac{\eta}{m} \sum_x \frac{\partial C_x}{\partial b}$$

对照上面w的更新公式。能够发现后面那一项变了，变成全部导数加和，乘以η再除以m，m是一个mini-batch中样本的个数。

到眼下为止，我们仅仅是解释了L2正则化项有让w“变小”的效果，可是还没解释为什么w“变小”能够防止overfitting？一个所谓“显而易见”的解释就是：更小的权值w，从某种意义上说，表示网络的复杂度更低，对数据的拟合刚刚好（这个法则也叫做奥卡姆剃刀），而在实际应用中，也验证了这一点。L2正则化的效果往往好于未经正则化的效果。当然，对于非常多人（包含我）来说，这个解释似乎不那么显而易见，所以这里加入一个略微数学一点的解释（引自知乎）：

过拟合的时候，拟合函数的系数往往非常大，为什么？例如下图所看到的，过拟合。就是拟合函数须要顾忌每个点。终于形成的拟合函数波动非常大。在某些非常小的区间里，函数值的变化非常剧烈。

这就意味着函数在某些小区间里的导数值（绝对值）非常大，由于自变量值可大可小，所以仅仅有系数足够大，才干保证导数值非常大。



而正则化是通过约束参数的范数使其不要太大，所以能够在一定程度上降低过拟合情况。

L1 regularization

在原始的代价函数后面加上一个L1正则化项。即全部权重w的绝对值的和。乘以λ/n（这里不像L2正则化项那样，须要再乘以1/2。详细原因上面已经说过。）

$$C = C_0 + \frac{\lambda}{n} \sum_w |w|.$$

相同先计算导数：

$$\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} \operatorname{sgn}(w),$$

上式中sgn(w)表示w的符号。那么权重w的更新规则为：

$$w \rightarrow w' = w - \frac{\eta \lambda}{n} \operatorname{sgn}(w) - \eta \frac{\partial C_0}{\partial w},$$

比原始的更新规则多出了η \* λ \* sgn(w)/n这一项。

当w为正时，更新后的w变小。

当w为负时。更新后的w变大——因此它的效果就是让w往0靠。使网络中的权重尽可能为0，也就相当于减小了网络复杂度，防止过拟合。

23. 关于java的JIT知识(3320)
24. Caused by:  
java.lang.ClassNotFoundException: org.apache.commons.lang3.Str
25. 国内三大PT ( Private Tracker ) 站分析(3251)
26. R语言和数据分析十大: 购物篮分析(3211)
27. 自己动手写CPU之第五阶段 ( 3 ) ——MIPS指令集中的逻辑、移位与空指令(3166)
28. 项目启动会应该注意的几点 (3144)
29. 【FAQ】SpringMVC实现集合参数 ( Could not instantiate bean class [java.util.List] ) (3144)
30. js 使用for循环遍历数组(3111)

#### 评论排行榜

1. .net web 开发平台- 表单设计器——(web版)(3)
2. PCB设计资料: 看到最后才知道是福利(2)
3. 做高通平台安卓驱动感言(2)
4. jquery+正则表达式验证邮箱格式的样例(1)
5. redis的分布式解决方式--codis(1)
6. 嵌入式系统开发步骤(1)
7. Basic脚本解释器移植到STM32(1)
8. Mozilla5.0的含义(1)
9. 西门子PLC学习笔记二- ( 工作记录 ) (1)
10. Akka FSM 源代码分析(1)
11. 深度学习论文笔记-Deep Learning Face Representation from Predicting 10,000 Classes(1)
12. 彩票APP将演绎“快鱼吃慢鱼”的发展轨迹(1)
13. 高速排序算法(1)
14. Android Fragment 简单实例 (1)
15. Java SE学习之printf 日期转换符(1)

#### 推荐排行榜

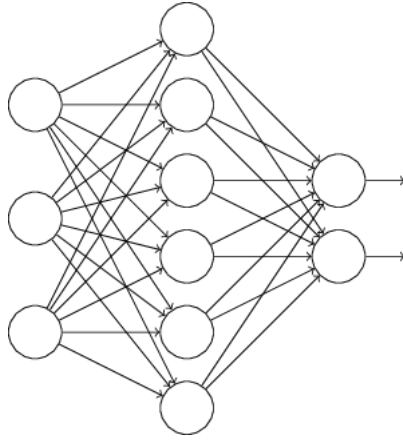
1. 模拟登陆CSDN——就是这么简单 (2)
2. redis的分布式解决方式--codis(2)
3. HashMap的存储结构及原理(2)
4. 最简单也最难——怎样获取到Android控件的高度(1)
5. linux date -d 的一些使用方法 (1)
6. Nginx负载均衡: 分布式/热备Web Server的搭建(1)
7. kafka web console安装(1)
8. hdu 4268 Alice and Bob(multiset|段树)(1)
9. Mozilla5.0的含义(1)
10. 通达OA 公共文件柜二次开发添加管理信息 ( 图文 ) (1)
11. Java SE学习之printf 日期转换符(1)

另外, 上面没有提到一个问题, 当 $w$ 为0时怎么办? 当 $w$ 等于0时,  $|w|$ 是不可导的。所以我们仅仅能依照原始的未经正则化的方法去更新 $w$ , 这就相当于去掉 $\eta * \lambda * \text{sgn}(w)/n$ 这一项, 所以我们能够规定 $\text{sgn}(0)=0$ , 这样就把 $w=0$ 的情况也统一进来了。

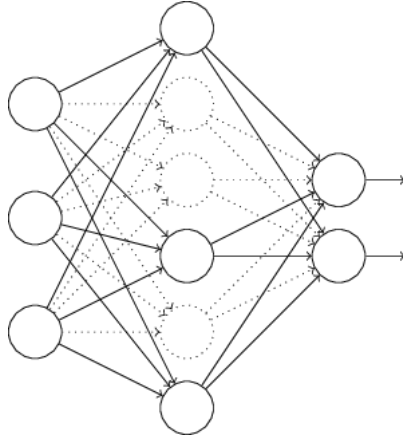
( 在编程的时候, 令 $\text{sgn}(0)=0, \text{sgn}(w>0)=1, \text{sgn}(w<0)=-1$  )

## Dropout

L1、L2正则化是通过改动代价函数来实现的, 而Dropout则是通过改动神经网络本身来实现的, 它是在训练网络时用的一种技巧 ( trike )。它的流程例如以下:



假设我们要训练上图这个网络, 在训练开始时, 我们随机地“删除”一半的隐层单元, 视它们为不存在, 得到例如以下的网络:



保持输入输出层不变, 依照BP算法更新上图神经网络中的权值 ( 虚线连接的单元不更新, 由于它们被“暂时删除”了 )。

以上就是一次迭代的过程, 在第二次迭代中, 也用相同的方法, 仅仅只是这次删除的那一半隐层单元, 跟上一次删除掉的肯定是不一样的。由于我们每一次迭代都是“随机”地去删掉一半。

第三次、第四次……都是这样, 直至训练结束。

以上就是Dropout, 它为什么有助于防止过拟合呢? 能够简单地这样解释。运用了dropout的训练过程, 相当于训练了非常多个仅仅有半数隐层单元的神经网络 ( 后面简称为“半数网络” ), 每个这种半数网络, 都能够给出一个分类结果, 这些结果有的是正确的, 有的是错误的。

随着训练的进行, 大部分半数网络都能够给出正确的分类结果。那么少数的错误分类结果就不会对最终结果造成大的影响。

更加深入地理解。能够看看Hinton和Alex两牛2012的论文《ImageNet Classification with Deep Convolutional Neural Networks》

## 数据集扩增 ( data augmentation )

“有时候不是由于算法好赢了。而是由于拥有很多其它的数据才赢了。”

不记得原话是哪位大牛说的了, hinton? 从中可见训练数据有多么重要, 特别是在深度学习方法中。很多其它的训练数据。意味着能够用更深的网络, 训练出更好的模型。

- 12. HDU1007 Quoit Design 【分治】(1)
- 13. 数据集成之主数据管理（一）基础概念篇(1)
- 14. Struts2中属性驱动与模型驱动(1)
- 15. 关于java的JIT知识(1)
- 16. PCB设计资料：看到最后才知道是福利(1)
- 17. 西门子PLC学习笔记二-（工作记录）(1)
- 18. linux学习 建立静态库，动态库，写简单的makefile(1)
- 19. 几种常见模式识别算法整理和总结(1)
- 20. cocos2d-x 3.0游戏实例学习笔记 《跑酷》完结篇--源代码放送(1)
- 21. JavaFX横幅类游戏开发 教训 游戏贴图(1)
- 22. 从零开始学习OpenCL开发（一）架构(1)
- 23. .net web 开发平台- 表单设计器 一(web版)(1)
- 24. Linux TC(Traffic Control)框架原理解析(1)

既然如此，收集很多其它的数据不即可啦？假设能够收集很多其它能够用的数据，当然好。可是非常多时候，收集很多其它的数据意味着须要耗费很多其它的人力物力。有弄过人工标注的同学就知道。效率特别低，简直是粗活。

所以。能够在原始数据上做些改动，得到很多其它的数据，以图片数据集举例，能够做各种变换，如：

- 将原始图片旋转一个小角度
- 加入随机噪声
- 一些有弹性的畸变（elastic distortions）。论文《Best practices for convolutional neural networks applied to visual document analysis》对MNIST做了各种变种扩增。
- 截取（crop）原始图片的一部分。

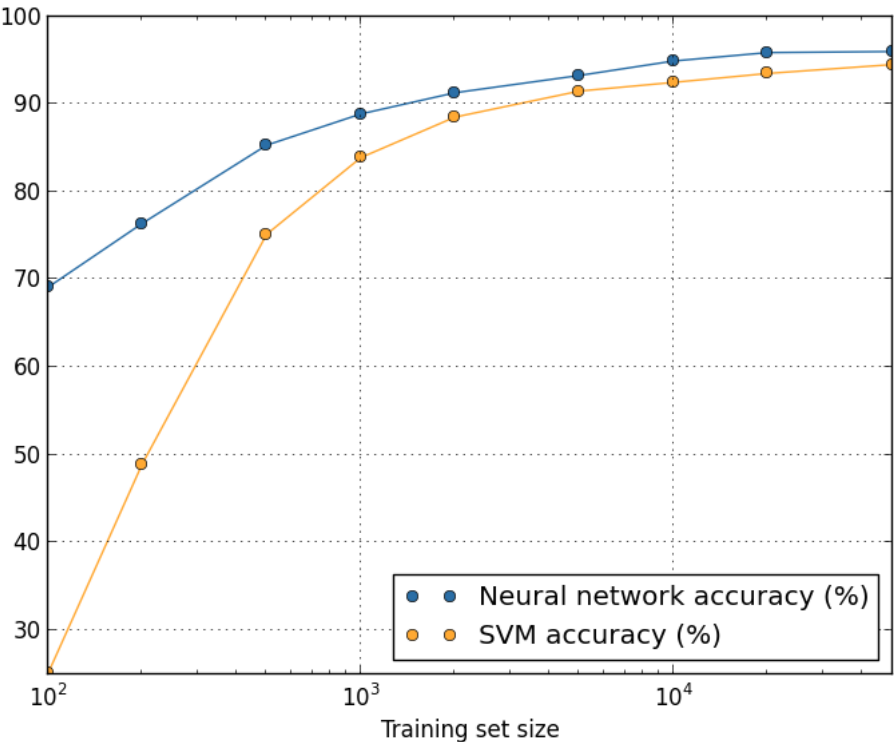
比方DeepID中，从一副人脸图中，截取出了100个小patch作为训练数据，极大地添加了数据集。

感兴趣的能够看《Deep learning face representation from predicting 10,000 classes》。

很多其它数据意味着什么？

用50000个MNIST的样本训练SVM得出的accuracy94.48%，用5000个MNIST的样本训练NN得出accuracy为93.24%，所以很多其它的数据能够使算法表现得更好。

在机器学习中，算法本身并不能决出胜负。不能武断地说这些算法谁优谁劣，由于数据对算法性能的影响非常大。



转载请注明出处：<http://blog.csdn.net/u012162613/article/details/44261657>

【众安尊享e生】 - 国民百万医疗保险，投保详解及案例分析,每年最低112元——马云杀手锏

好文要顶

关注我

收藏该文

yxwkaifa  
关注 - 0  
粉丝 - 29

+加关注

« 上一篇：[Mina入门实例](#)  
» 下一篇：[\(各个公司面试原题\)在线做了一套CC++综合测试题，也来测一下你的水平吧\(二\)](#)

posted @ 2016-03-12 14:04 yxwkaifa 阅读(8081) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】超50万VC++源码：大型组态工控、电力仿真CAD与GIS源码库！
- 【推荐】腾讯云新注册用户域名抢购1元起



**最新IT新闻:**

- 以色列将区块链用于证券系统信息安全管理
  - 苹果史上最大人员收购案诞生：6亿美元交易，300名Dialog工程师加入
  - 自愈材料能利用二氧化碳“生长”
  - 百度华为在5G MEC领域达成战略合作
  - 波音777X客机装备折叠翼尖 收放过程仅需20秒
- » 更多新闻...



**最新知识库文章:**

- 为什么说 Java 程序员必须掌握 Spring Boot ?
  - 在学习中，有一个比掌握知识更重要的能力
  - 如何招到一个靠谱的程序员
  - 一个故事看懂“区块链”
  - 被踢出去的用户
- » 更多知识库文章...

Copyright ©2018 yxwkaifa