深度学习笔记

kenny.1.liu@nokia-sbell.com

Contents

[深度学习笔记 1](#_Toc508917554)

[一元线性回归（linear regression） 2](#_Toc508917555)

[逻辑回归 3](#_Toc508917556)

[正则化 3](#_Toc508917557)

[浅层神经网络概览 4](#_Toc508917558)

## 一元线性回归（linear regression）

### 定义

一种回归模型，通过将输入特征进行线性组合，以连续值作为输出。其数学model为：

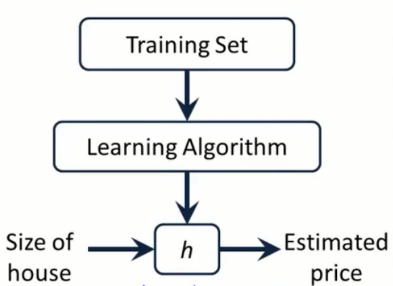
m : Number of training examples

x : “input”variable/features

y : “output”varable/“target”variable

(x, y): one training example

:第i个training example



### 训练深度学习网络的方式

#### 监督式学习（Supervised Learning）

监督学习是使用已知的“正确答案”的示例来训练网络的。想象一下，我们可以训练一个网络，让其从照片库中（其中包含你父母的照片）识别出你父母的照片。以下就是我们在这个假设场景中所要采取的步骤。

步骤1：数据集的创建和分类

首先，我们要浏览你的照片（数据集），确定所有有你父母的照片，并对其进行标注，从而开始此过程。然后我们将把整堆照片分成两堆。我们将使用第一堆来训练网络（训练数据），而通过第二堆来查看模型在选择我们父母照片操作上的准确程度（验证数据）。

等到数据集准备就绪后，我们就会将照片提供给模型。在数学上，我们的目标就是在深度网络中找到一个函数，这个函数的输入是一张照片，而当你的父母不在照片中时，其输出为0，否则输出为1。

此步骤通常称为分类任务（categorization task）。在这种情况下，我们进行的通常是一个结果为yes or no的训练，但事实是，监督学习也可以用于输出一组值，而不仅仅是0或1。例如，我们可以训练一个网络，用它来输出一个人偿还信用卡贷款的概率，那么在这种情况下，输出值就是0到100之间的任意值。这些任务我们称之为回归。

步骤2：训练

为了继续该过程，模型可通过以下规则（激活函数）对每张照片进行预测，从而决定是否点亮工作中的特定节点。这个模型每次从左到右在一个层上操作——现在我们将更复杂的网络忽略掉。当网络为网络中的每个节点计算好这一点后，我们将到达亮起（或未亮起）的最右边的节点（输出节点）。

既然我们已经知道有你父母的照片是哪些图片，那么我们就可以告诉模型它的预测是对还是错。然后我们会将这些信息反馈（feed back）给网络。

该算法使用的这种反馈，就是一个量化“真实答案与模型预测有多少偏差”的函数的结果。这个函数被称为成本函数（cost function），也称为目标函数（objective function），效用函数（utility function）或适应度函数（fitness function）。然后，该函数的结果用于修改一个称为反向传播（backpropagation）过程中节点之间的连接强度和偏差，因为信息从结果节点“向后”传播。

我们会为每个图片都重复一遍此操作，而在每种情况下，算法都在尽量最小化成本函数。

其实，我们有多种数学技术可以用来验证这个模型是正确还是错误的，但我们常用的是一个非常常见的方法，我们称之为梯度下降（gradient descent）。Algobeans上有一个 “门外汉”理论可以很好地解释它是如何工作的。迈克尔尼尔森（Michael Nielsen）用数学知识完善了这个方法，其中包括微积分和线性代数。

步骤3：验证

一旦我们处理了第一个堆栈中的所有照片，我们就应该准备去测试该模型。我们应充分利用好第二堆照片，并使用它们来验证训练有素的模型是否可以准确地挑选出含有你父母在内的照片。

我们通常会通过调整和模型相关的各种事物（超参数）来重复步骤2和3，诸如里面有多少个节点，有多少层，哪些数学函数用于决定节点是否亮起，如何在反向传播阶段积极有效地训练权值，等等。而你可以通过浏览Quora上的相关介绍来理解这一点，它会给你一个很好的解释。

步骤4：使用

最后，一旦你有了一个准确的模型，你就可以将该模型部署到你的应用程序中。你可以将模型定义为API调用，例如ParentsInPicture(photo)，并且你可以从软件中调用该方法，从而导致模型进行推理并给出相应的结果。

稍后我们将详细介绍一下这个确切的过程，编写一个识别名片的iPhone应用程序。

得到一个标注好的数据集可能会很难（也就是很昂贵），所以你需要确保预测的价值能够证明获得标记数据的成本是值得的，并且我们首先要对模型进行训练。例如，获得可能患有癌症的人的标签X射线是非常昂贵的，但是获得产生少量假阳性和少量假阴性的准确模型的值，这种可能性显然是非常高的。

#### 无监督学习（Unsupervised Learning）

待述

#### 半监督学习（Semi-supervised Learning）

待述

#### 强化学习

待述

为什么叫回归（regression）

高尔顿与回归分析的起源  
“回归”是由英国著名生物学家兼统计学家高尔顿(Francis Galton,1822～1911.生物学家达尔文的表弟)在研究人类遗传问题时提出来的。为了研究父代与子代身高的关系，高尔顿搜集了1078对父亲及其儿子的身高数据。他发现这些数据的散点图大致呈直线状态，也就是说，总的趋势是父亲的身高增加时，儿子的身高也倾向于增加。但是，高尔顿对试验数据进行了深入的分析，发现了一个很有趣的现象—回归效应。因为当父亲高于平均身高时，他们的儿子身高比他更高的概率要小于比他更矮的概率；父亲矮于平均身高时，他们的儿子身高比他更矮的概率要小于比他更高的概率。它反映了一个规律，即这两种身高父亲的儿子的身高，有向他们父辈的平均身高回归的趋势。对于这个一般结论的解释是:大自然具有一种约束力，使人类身高的分布相对稳定而不产生两极分化，这就是所谓的回归效应。  
1855年， 高尔顿发表《遗传的身高向平均数方向的回归》一文，他和他的学生卡尔•皮尔逊Karl·Pearson通过观察1078对夫妇的身高数据，以每对夫妇的平均身高作为自变量，取他们的一个成年儿子的身高作为因变量，分析儿子身高与父母身高之间的关系，发现父母的身高可以预测子女的身高，两者近乎一条直线。当父母越高或越矮时，子女的身高会比一般儿童高或矮，他将儿子与父母身高的这种现象拟合出一种线形关系，分析出儿子的身高y与父亲的身高x大致可归结为一下关系：  
y=33.73+0.516\*x (单位为英寸)  
根据换算公式1英寸=0.0254米， 1米=39.37英寸。单位换算成米后：  
Y= 0.8567+0.516\*X (单位为米);  
假如父母辈的平均身高为1.75米，则预测子女的身高为1.7597米。  
这种趋势及回归方程表明父母身高每增加一个单位时，其成年儿子的身高平均增加0.516个单位。这就是回归一词最初在遗传学上的含义。  
有趣的是，通过观察，高尔顿还注意到，尽管这是一种拟合较好的线形关系，但仍然存在例外现象:矮个父母所 生的儿子比其父要高，身材较高的父母所生子女的身高却回降到多数人的平均身高。换句话说，当父母身高走向极端，子女的身高不会象父母身高那样极端化，其身高要比父母们的身高更接近平均身高，即有“回归”到平均数去的趋势，这就是统计学上最初出现“回归”时的涵义，高尔顿把这一现象叫做“向平均数方向的回归” (regression toward mediocrity)。虽然这是一种特殊情况，与线形关系拟合的一般规则无关，但“线形回归”的术语却因此沿用下来，作为根据一种变量(父母身高)预测另一种变量(子女身高)或多种变量关系的描述方法。  
回归的现代意义：  
它要比其原始意义广泛的多。具体地说，回归分析的内容包括：  
• 确定响应变量与预报变量间的回归模型,即变量间相关关系的数学表达式(通常称为经验公式);  
• 根据样本估计并检验回归模型及未知参数；  
• 从众多的预报变量中，判断哪些变量对响应变量的影响是显著的，哪些是不显著的；  
• 根据预报变量的已知值或给定值来估计或预测响应变量的平均值并给出预测精度或根据响应变量的给定值来估计预报变量的值，即所谓的预报与控制问题。

## 逻辑回归

单个训练样本的成本函数（cost function）:

example:

m个训练样本的成本函数（cost function）：

梯度（Gradient）:

## 正则化

## 浅层神经网络概览



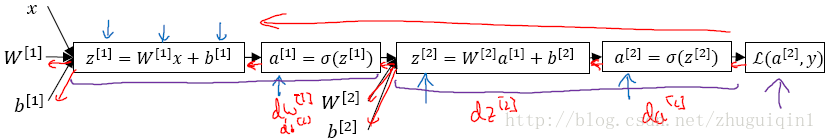
1. 浅层神经网络分成三层结构，输入层，隐藏层，以及输出层，隐藏层之所以叫隐藏层是因为其参数是不可见的。
2. 每个节点的完成的的功能如图所示。
3. 

下图是神经网络的正向和反向的计算图，

1. 上标[1]表示神经网络的第一层，上标[2]表示神经网络的第二层。

2. 正向方向上，第一层和第二层都包含有z和a的计算，浅层神经网路包含了两次的逻辑回顾计算，最终得到lost函数的值。

3. 对于反向上，每层都可以计算出倒数，da[2]dz[2]dw[2] db[2] 以及 da[1]dz[1]dw[1] db[1]



【 dasfa a[2]  w[2] ，b[2]并且w[2]是一个4x3的矩阵) b[2] （4x1的向量）

x1

**b[1]= z[1]= a[1] =**

**z[1] = x + b[1] = =**

**a[1] = =**

a[1] =

**z[1] = x + b[1]**

**a[1] =**

**z[2] = a[1] + b[2]**

**a[2] = =**

b[1] =

A =