

Avazu DSP机器学习 技术简介

www.avazuinc.com

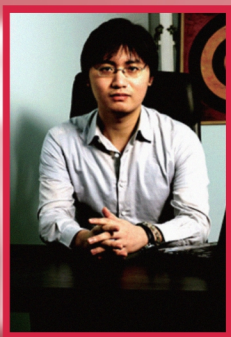
官方 facebook 页：
www.facebook.com/avazuinc



微信二维码



微博二维码



A stylized, handwritten signature in black ink, consisting of several fluid, connected strokes.

CEO 寄语

Avazu (艾维昂动) 自2009年成立开始就从事程序化广告业务, 并于09年10月在全球第一家广告交易平台Rightmedia上进行了实时广告交易。我们自研的各项程序化广告产品在帮助广告商最大化ROI的同时也大大提升了媒体的库存变现效率。在这5年的时间里, 我们从技术、产品到运营都具有较大的积累, 并且能够满足客户投放广告的同时让其按照实际带来的效果付费(新增用户, 新增销售等)。让我们坚持下来埋头打磨产品的动力是因为我们一直坚信, 技术导向和平台化的商业模式才是未来可扩展性最强的, 所以在这几年内花费大量的人力, 物力和财力对广告系统的各个领域进行研究和打磨, 包括但不局限于分布式架构, 大数据处理, 模型识别和机器学习。今天的Paper主要目的是用最快速的方式来介绍我们将要发布的移动DSP (Avazu Mobile DSP) 产品最常用的两个机器学习算法以及它们在广告领域的应用。

目录



前言	01
广告优化	02
机器学习	02
逻辑回归	03
概述	03
训练方法	04
稀疏数据	05
特征编码与个性化	05
实验和分析	06
深度学习	07
概述	07
训练方法	08
广告数据	10
实验和分析	10
小结	11

前言

Avazu（艾维邑动）是一家集PC和移动互联网广告全球投放，全球专业移动游戏运营及发行的技术型公司。为了给客户提供最好的广告效果，公司自主研发的DSP平台使用了最前沿的机器学习算法，下面就来介绍一下相关的广告优化原理和机器学习算法。

广告优化

众所周知，广告点击率（CTR）和转化率（CR）代表了广告投放的效果，如何提高CTR和CR是每个广告主都十分关心的问题。Avazu通过机器学习算法，自动地为实时流量预估CTR，广告主只需简单的将优化目标设置为期望CTR，DSP投放引擎即可为广告主购买相应的优质流量，完成这一任务。如广告主设置优化目标为CPC，则投放引擎通过将CPC转化为期望CPM（ $CPM = CTR * 1000 * CPC$ ），购买对应价格的流量。所以机器学习预测越准，广告优化效果就越好。

机器学习

机器学习是一类从数据中自动分析获得规律，并利用规律对未知数据进行预测的算法。由于互联网行业的数据规模已超过人工分析能力之所及，机器学习技术几乎成为每家互联网公司的标配，在搜索排序，商品排序，点击率预估，反作弊，实时竞价等各种领域有着广泛的应用。Avazu的机器学习平台最主要的算法包括逻辑回归、随机森林、深度学习，这里我们介绍逻辑回归和深度学习。

逻辑回归

概述

逻辑回归 (Logistic Regression) 是线性模型的一种, 历史悠久, 广泛应用于各种分类任务, 尤其在互联网广告行业中, 已成为点击率预估的基准方法。

二分类逻辑回归的预测公式形如 $p(y^j=1|x^j)=1/(1+\exp(-wx^j))$

$p(y^j=-1|x^j)=\exp(-wx^j)/(1+\exp(-wx^j))$

其中 x_i 为数据特征, w 为模型参数, y_i 为分类目标, $p(y^j=1|x^j)$ 即预测 y 为 1 的概率, 在广告点击率预估中即为点击的概率。

对模型预测的 y 值和真实 y 值建立的损失函数 E 为 LogLoss, 形如

$$E = \frac{1}{n} \sum \log(1 + \exp(-y^j * wx^j))$$

最小化损失函数可使预测值与真实值差异最小。为控制模型过拟合, 可以为损失函数加上正则项, 常见的正则项包括 w 的 2 范数或 1 范数, 即 $E = \frac{1}{n} \sum \log(1 + \exp(-y^j * wx^j)) + \lambda \|w\|_2^2$

其中 $\|w\|_2^2$ 为 w 的 2 范数, λ 为超参数, 需要通过交叉验证 (cross-validation) 选取。

训练方法

我们可以通过一些最优化（Optimization）算法来最小化损失函数，常见的最优化方法有随机梯度下降（SGD）和L-BFGS等。

SGD是一种在线（online）算法，其特点是每次更新模型只使用少量随机数据（通常为一个样本），因此训练速度很快。SGD的更新方法形如

$$w^{t+1} = w^t - \eta^t g^t$$

其中 g^t 为损失函数 E 对 w^t 的梯度， η^t 为梯度下降的步长。对于步长 η^t 有很多研究，一种常见的设计为

$$\eta^t = \eta^0 / (1 + \lambda \eta^0 t)$$

一些更为现代的方法包括各种自适应步长、模型平均、Nesterov的一阶加速方法等。

由于SGD的随机性，其收敛速度和质量不是最好，可能需要迭代几十轮（遍历整个数据集的次数）。

L-BFGS则是一种batch算法，与online算法相反，每次更新模型都使用整个数据集，因此训练速度较慢，但模型收敛稳定，预测质量好于SGD。

对于广告行业来说，由于数据具有冗余性（即在整个数据集中，相同的记录会出现多次），所以以L-BFGS为代表的batch算法在一次迭代中做了很多重复计算，因此SGD在大数据集上更受欢迎。也可以在迭代的前几轮使用SGD，以较小的代价求得一个较好的解，再用L-BFGS继续训练，得到更好更稳定的解。

在大数据背景下，算法通常需要并行化。一种常见的策略是将数据随机分成多份，每份数据各自训练独立的模型，最后将多个模型的参数按一定的加权办法融合平均。其他并行/分布式训练策略还有很多，在此不一一赘述。

稀疏数据

互联网广告行业的数据通常被称为“稀疏数据”，即一行记录只包含少量特征。举例来说，如果我们有 n 个广告商，那么整个模型关于广告商的特征有 n 个，而一次曝光只含有一个广告商，则该曝光该广告商特征取值1，其他广告商特征取值0。这种特征处理方式被称为1-hot编码。对于0值我们不做记录也不做处理，因此称为稀疏数据。

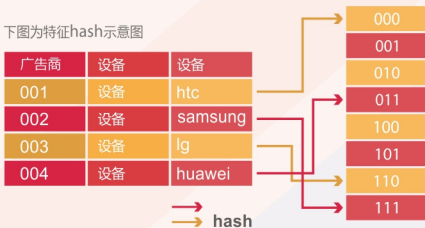
特征编码与个性化

如前所述，我们对所有的特征都进行1-hot编码，由于每个特征的编码集大小动态变化（比如加入了新的广告商），使得模型训练颇为不便，所以我们使用hash方法将特征映射到固定大小的编码集上。比如说，对设备这个特征，以二元组 \langle “设备”，设备ID \rangle 进行hash，得到值1024，则将映射特征1024的取值为1。同时，我们还可以组成特征三元组，如 \langle 广告商ID，“设备”，设备ID \rangle ，hash以后的特征对于每

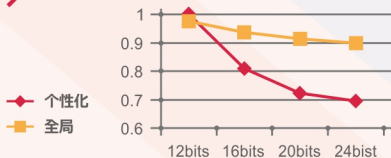
个广告商不尽相同，则训练出来的模型可视为广告商个性化投放模型。

使用hash方法给特征编码，相对于使用字典编码，速度非常快，并且特征和模型参数也是固定的。

下图为特征hash示意图



实验和分析



上图为使用一个月的Avazu日志数据训练个性化逻辑回归（以三元组hash编码的形式实现，图中x坐标轴为hash bit数）与非个性化逻辑回归的CTR预估误差（相对值）对比。我们可以看到，个性化模型大大提高了CTR预估的准确性。

深度学习



概述

深度学习（Deep Learning）是神经网络的一种，即层数很多的神经网络，其概念由Hinton于2006年提出，成为近年来神经网络复兴的标志。

深度学习的成功，最初来自于微软研究院在语音识别上的突破，而后由Hinton带领的小组在图像识别任务上取得了惊人成果。

深度学习的这股热潮，在国内机器学习业界几乎无人不谈，目前几大互联网公司均在尝试，其成果以百度深度学习研究院最为突出，但应用于广告领域，国内外尚未见大量成功案例的报道。

训练方法

传统的神经网络通常只有3层（输入层、隐层、输出层），因多层神经网络训练时会遇到所谓梯度消散（vanishing gradient）的问题，故并不成功，直到2006年，Hinton提出了逐层预训练（layer-wise pre-training）的方法才得以改善。后在2010年由James Martens提出一种Hessian-free方法使得深度学习不再需要预训练，大为简化。

如今的深度学习算法，回到了80年代发明的反向传播算法，以超大的数据量，亿万级别的参数，超长的训练时间来弥补算法的不足。下面就简述一下反向传播算法。

假设每一层的网络，为如下形式

$$Y_n = W_n X_{n-1}$$

$$X_n = F(Y_n)$$

其中X为输入层，Y为输出层，n和n-1为层的标号，F为激励函数，W则是模型的参数。

激励函数的目的是使得神经网络非线性化，否则多层的线性变换会退化为单层线性变换，也就失去意义了，常见的激励函数包括sigmoid、tanh等。

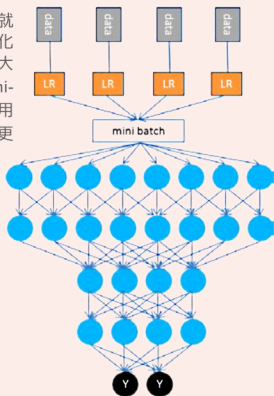
由此，我们可以使用前向传播算法，由下一层的输入X求出上一层的输出Y，而由Y的激励函数F求出更上一层的输入X，直至最上层的Y。

现在定义某种形式的损失函数E，对其求偏导，根据链式求导法则，有

$$\frac{\partial E^p}{\partial Y_n} = F'(Y_n) \frac{\partial E^p}{\partial X_n}$$
$$\frac{\partial E^p}{\partial W_n} = X_{n-1} \frac{\partial E^p}{\partial Y_n}$$
$$\frac{\partial E^p}{\partial X_{n-1}} = W_n^T \frac{\partial E^p}{\partial Y_n}$$

现在，我们可以使用反向传播算法，由上一层输入X的偏导求出该层输出Y的偏导，而由Y的偏导求出该层W和下一层输入X的偏导，直至最下层的W。

算出了每一层的梯度以后，就可以使用梯度下降之类的优化算法更新模型参数W。针对大数据，深度学习通常采用mini-batch的更新方法，即每次使用128或256个样本的梯度信息更新模型。



广告数据

实际上深度学习并不能直接应用于广告领域建模，因为如前所述，广告数据是稀疏数据，而深度学习主要是矩阵运算，是针对稠密数据的算法。所以我们对第一层采用了预训练的方法。具体如下：

如上图所示，第一层依然使用逻辑回归（LR）的并行训练算法，每个线程使用SGD训练自己的数据。模型接近收敛以后不做加权平均，而作为深度学习下一层的输入，再使用前向传播-反向传播算法训练深度神经网络。

显然，逻辑回归的输出是一个稠密向量（如果有16个线程训练，那就会输出一个16维的向量），取其mini-batch适合于训练深度学习模型。

实验和分析

我们再次使用一个月的日志数据分别训练了逻辑回归模型和深度学习模型（都使用个性化hash），并欣喜的发现，深度学习对点击率预估的错误率（以RMSE衡量）相对于逻辑回归降低了6%之多。

Campaign Name	Impressions	Clicks	CTR
0627 A/B Test_02 (Goal CTR=2%)	22,939	518	2.26%
0627 A/B Test_02 (NO GOAL)	75,673	594	0.78%
0627 A/B Test_03 (Goal CTR=2%)	33,478	832	2.49%
0627 A/B Test_03 (NO GOAL)	35,272	213	0.60%

上图为设定CTR目标前后的实际投放测试效果对比，可见使用深度学习后，DSP投放引擎自动选择了符合目标的优质流量。

对于深度学习的提升作用，我们认为可能出于以下原因：

1. 逻辑回归每个线程只训练自己的数据，而深度学习则看到了所有的数据。
2. 逻辑回归最终模型加权方式是人工指定的，而深度学习则使用了一个复杂的深度神经网络来进行融合。

深度学习的缺点在于需要保留n份逻辑回归模型，其预测耗时相比逻辑回归增加了许多，所幸逻辑回归本身计算速度非常快。

小结

Avazu DSP的机器学习技术，将广告投放的优化工作简化为算法自动优化。在个性化逻辑回归的基础上，新开发的深度学习算法又有大幅提升。那么，深度学习在广告行业是否还有更合适的训练方法，能否取得更好的效果？Avazu还将继续探索。

