深度学习在短信拦截中的实践

团队：手机卫士数据挖掘部
讲师：郭祥

团队和个人介绍

手机卫士数据挖掘部

- 骚扰号码识别
- 垃圾短信拦截
- 反诈骗
- 手机卫士用户画像
- 手机卫士商业化

郭祥

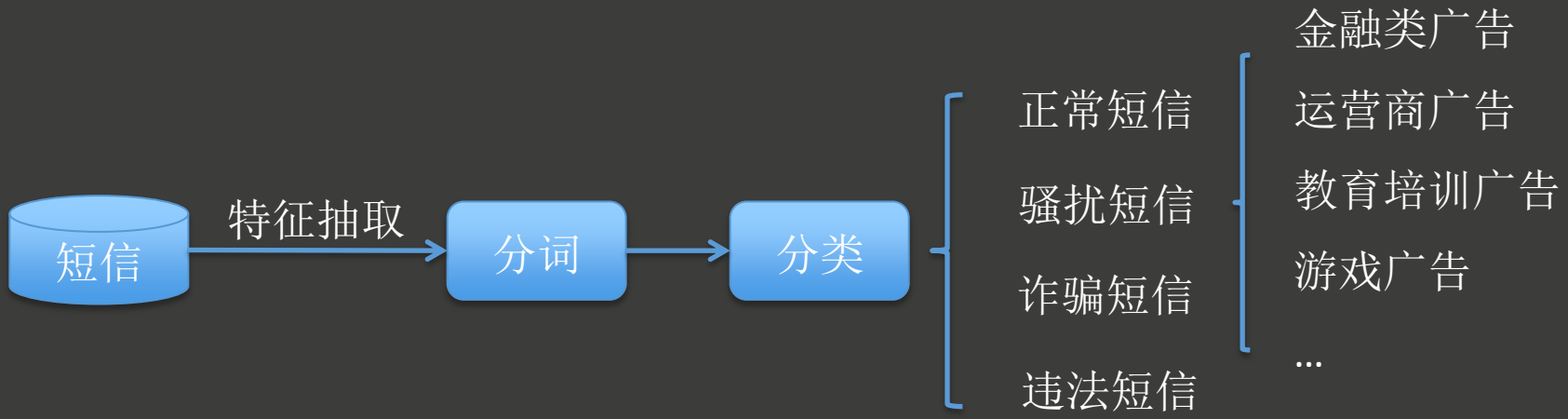
西安电子科技大学
负责垃圾短信拦截算法的优化和实现

主要内容

- 短信拦截应用场景
- 云端算法实现以及上线
- 移动端算法实现以及落地

短信拦截场景

- 尊敬的建行用户:您账户已满1万积分，可兑换5%的现金，请登入手机网 wap.evcsh.cc 查询兑换【建设银行】
- 专业办理全国各类证件，车牌，刻章，微信手机同号187xxxxxxxx
- 【XX百货】梦洁会员节，感恩钜惠，全场2折起，天丝四件套1399，送一条被芯及一对枕芯，各种超值大礼等着您，赶快行动吧！详询86xxxxxx10



短信拦截场景

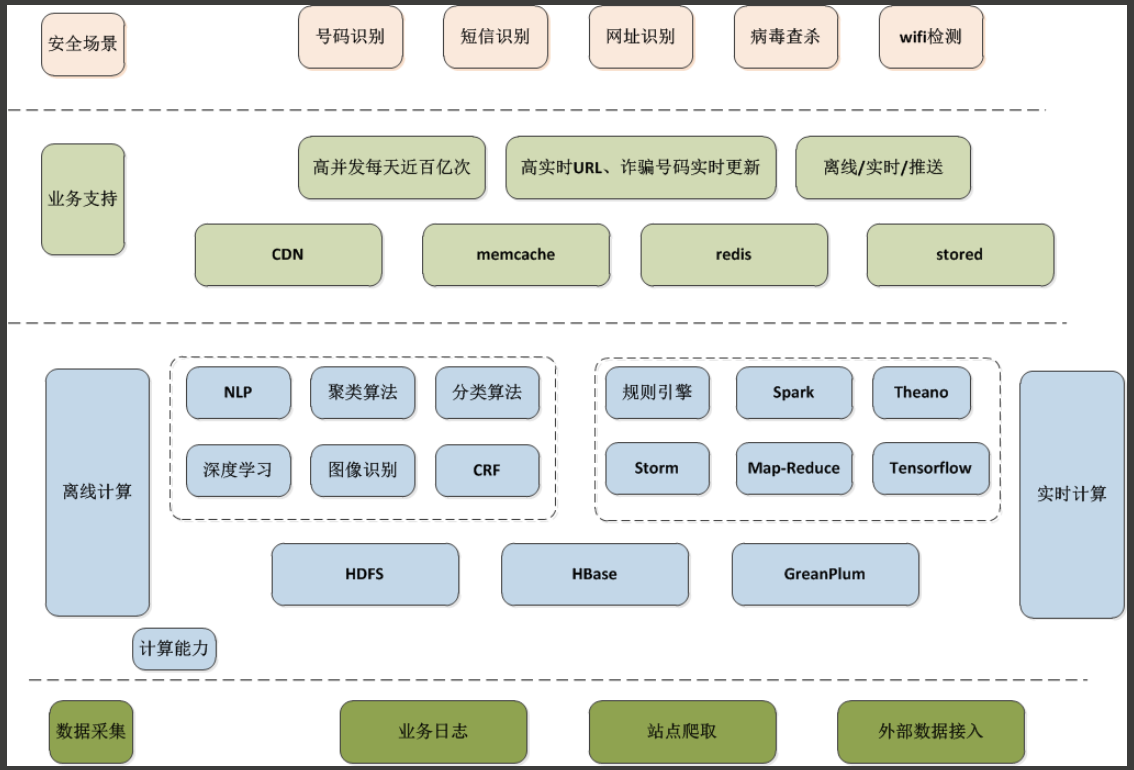
断网下，本地引擎特征识别：

- 客户端短信特征提取，AI模型
- 伪基站识别引擎

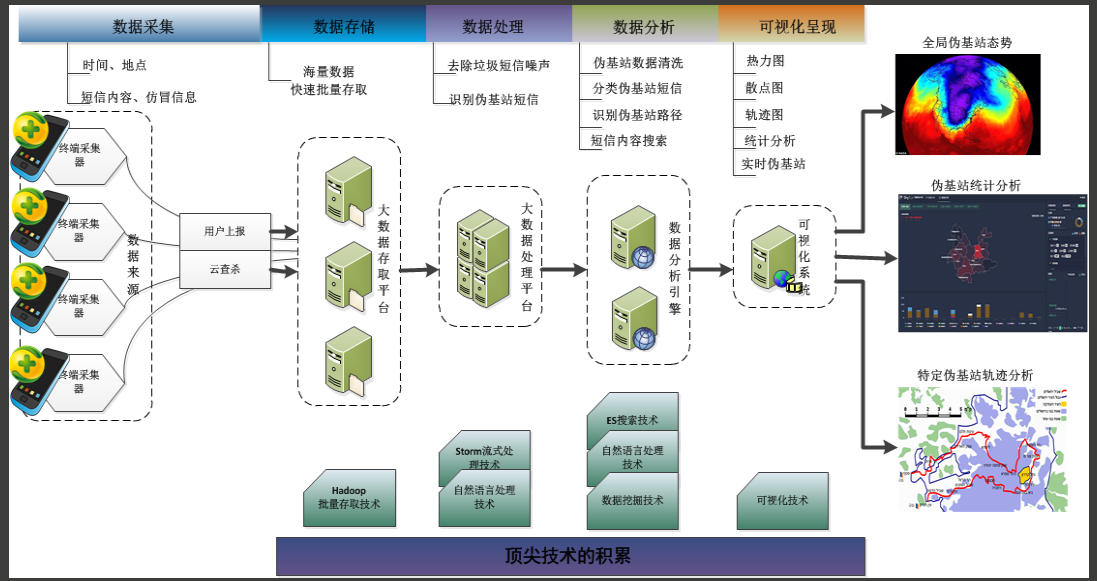
联网下，利用海量数据建立更为强大的拦截能力：

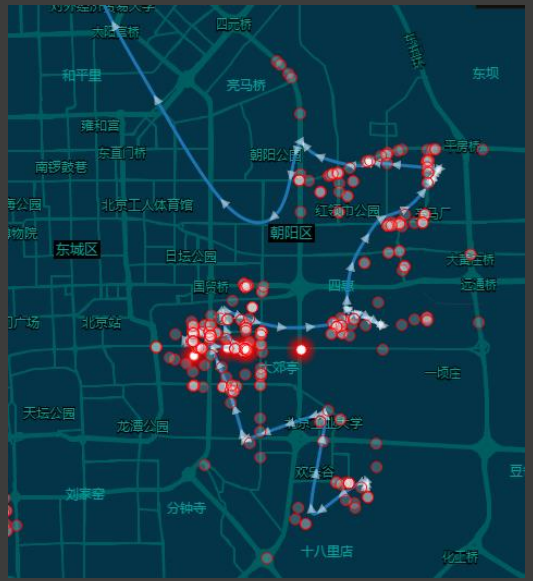
- 短信特征分析平台
- 黑网址鉴定分析平台
- 黑号码鉴定分析平台

安全架构


安全场景
号码识别
短信识别
网址识别
病毒查杀
wifi检测
业务支持
高并发每天近百亿次
高实时URL、诈骗号码实时更新
离线/实时/推送
CDN
memcache
redis
stored
离线计算
NLP
聚类算法
分类算法
深度学习
图像识别
CRF
规则引擎
Spark
Theano
Storm
Map-Reduce
Tensorflow
实时计算
HDFS
HBase
GreanPlum
计算能力
数据采集
业务日志
站点爬取
外部数据接入

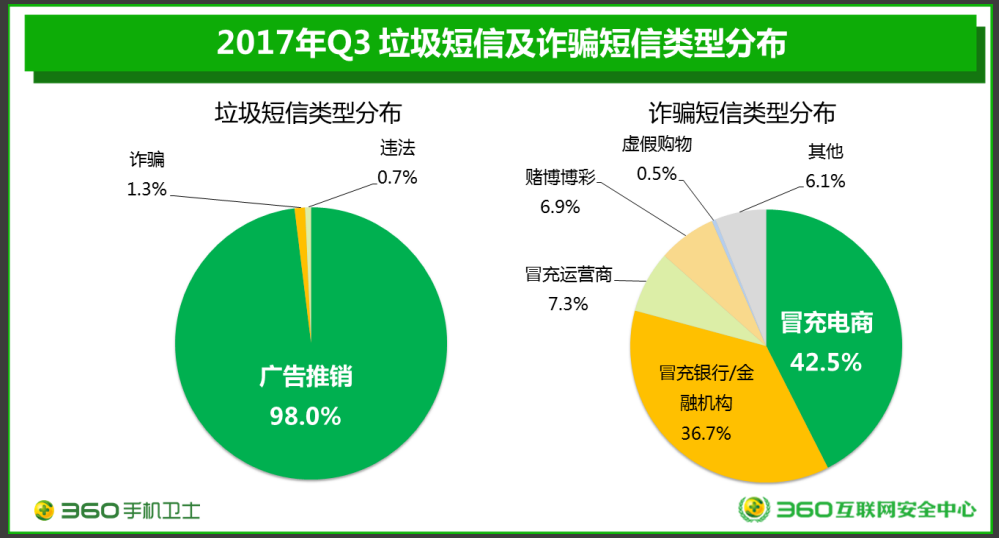
伪基站轨迹


数据采集
数据存储
数据处理
数据分析
可视化呈现
时间、地点
短信内容、伪冒信息
海量数据
快速批量存取
去除垃圾短信噪声
识别伪基站短信
伪基站数据清洗
分类伪基站短信
识别伪基站路径
短信内容搜索
热力图
散点图
轨迹图
统计分析
实时伪基站
终端采集器
终端采集器
终端采集器
终端采集器
数据来源
用户上报
云查杀
大数据存取平台
大数据处理平台
数据分析引擎
可视化系统
全局伪基站态势
伪基站统计分析
特定伪基站轨迹分析
Hadoop批量存取技术
Storm流式处理技术
自然语言处理技术
ES搜索技术
自然语言处理技术
数据挖掘技术
可视化技术
顶尖技术的积累


和平里
亮马桥
东坝
朝阳公园
南锣鼓巷
北京工人体育馆
朝阳区
东城区
日坛公园
北京站
十八里店

拦截能力

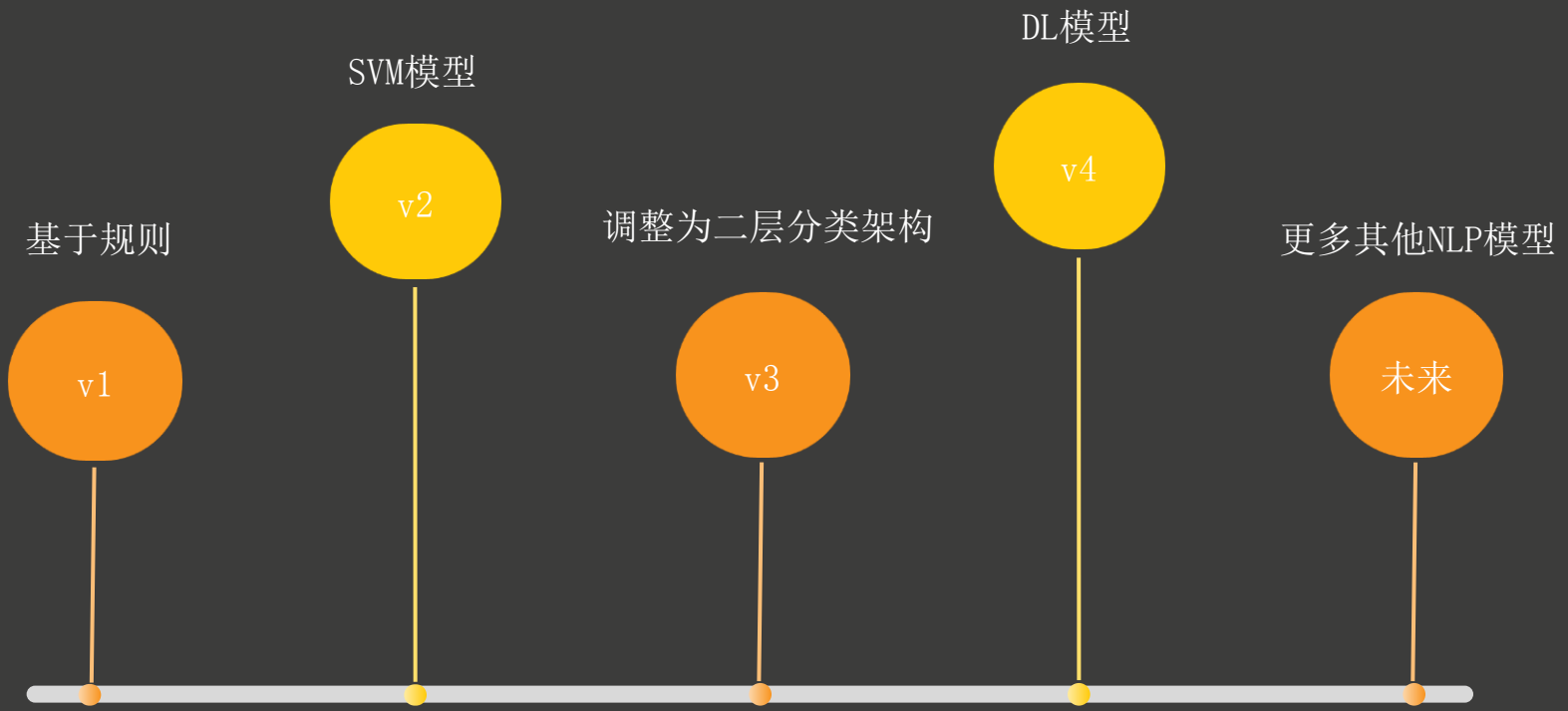
- 拦截垃圾短信约20.6亿条
- 拦截诈骗短信近3000万条
- 平均每天拦截垃圾短信2238.1万条



主要内容

- 短信拦截应用场景
- 云端算法实现以及上线
- 移动端算法实现以及落地

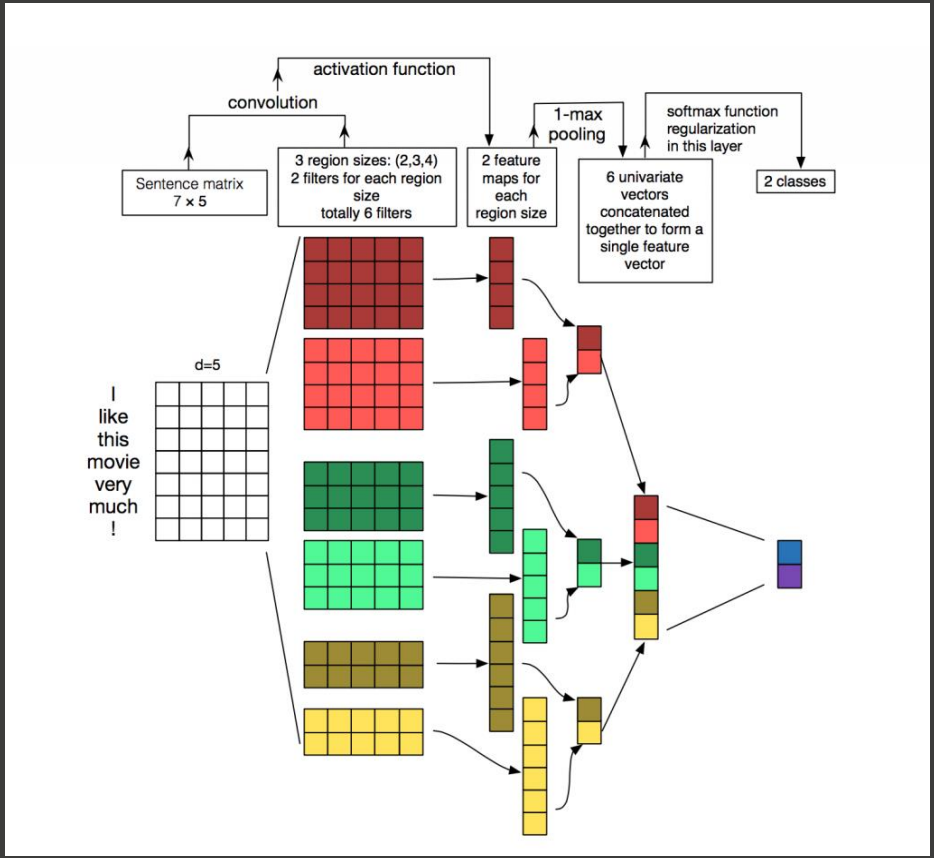
垃圾短信拦截算法演变



TEXT_CNN

1. 词嵌入层
   word2vec,glove,随机初始化
2. 卷积层
   隐式的表示n-gram
3. 池化层
   提取主要特征信息
4. 全连接层以及softmax层
5. 交叉熵损失以及L2正则化

$$Loss = -\frac{1}{m} * \sum_{i=1}^{m} y_i^{'} \log f(x_i) + \lambda \sum_{k=1}^{L} sum(\|W_k\|^2)$$



WORD2VEC

两种训练架构:

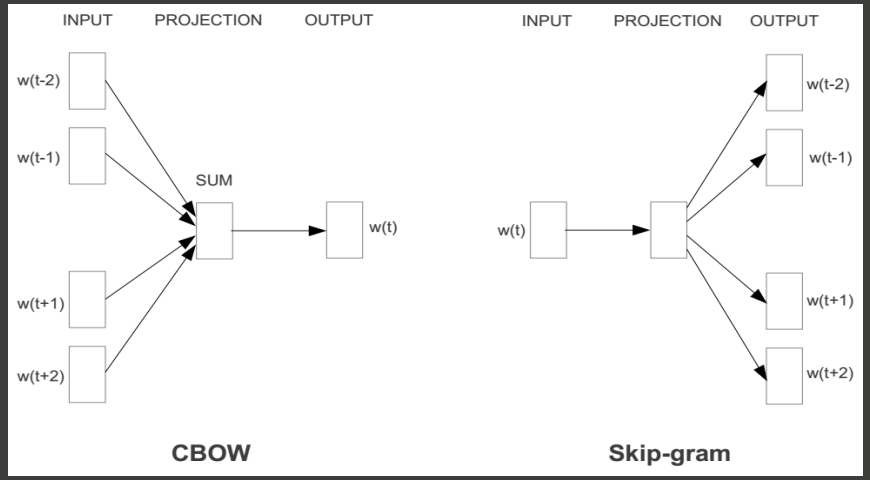
- CBOW : 根据上下文预测当前词
- SG : 根据当前词预测上下文

最大化对数似然:

- CBOW : $L = \sum_{w \in C} \log p(w | Context(w))$
- SG : $L = \sum_{w \in C} \log p(Context(w) | w)$

两种优化方法:

- hierarchical softmax
- negative sampling



Word Analogies

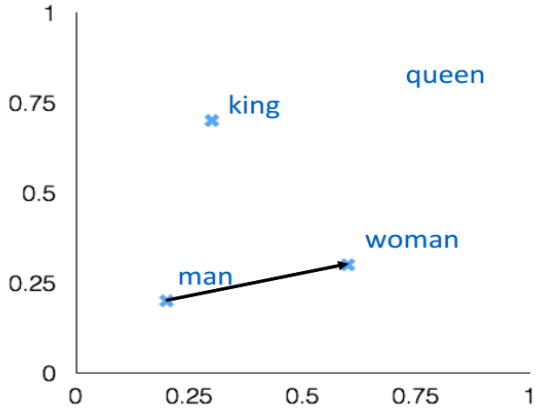
Test for linear relationships, examined by Mikolov et al. (2014)

a:b :: c:?

$$d = \arg\max_x \frac{(w_b - w_a + w_c)^T w_x}{||w_b - w_a + w_c||}$$

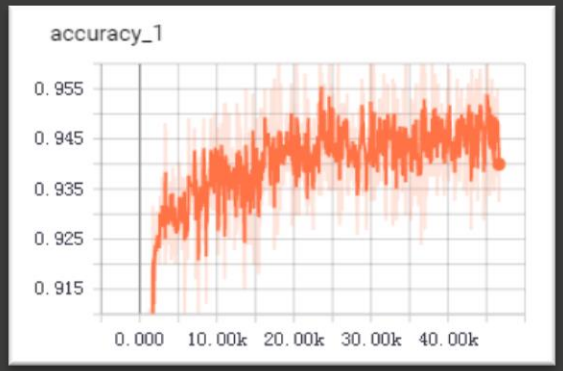
man:woman :: king:?

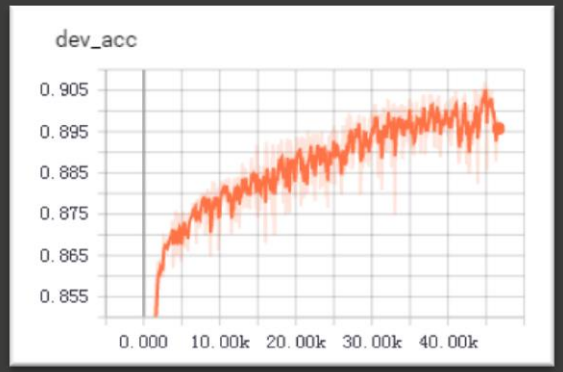
+	king	[0.30 0.70]
-	man	[0.20 0.20]
+	woman	[0.60 0.30]
	queen	[0.70 0.80]



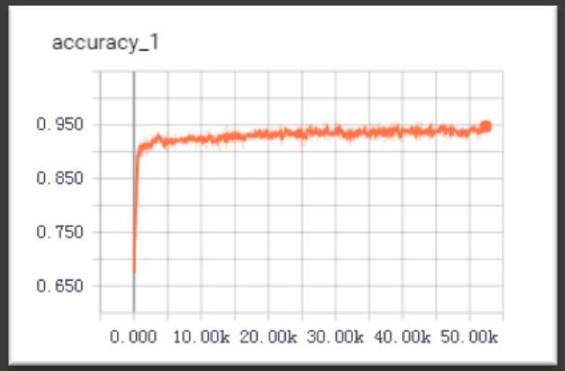
使用word2vec做预训练

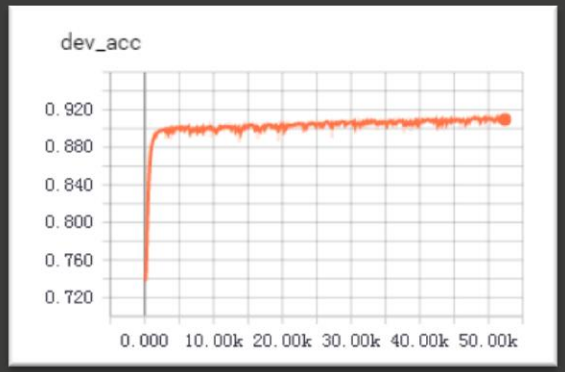
No pretrain





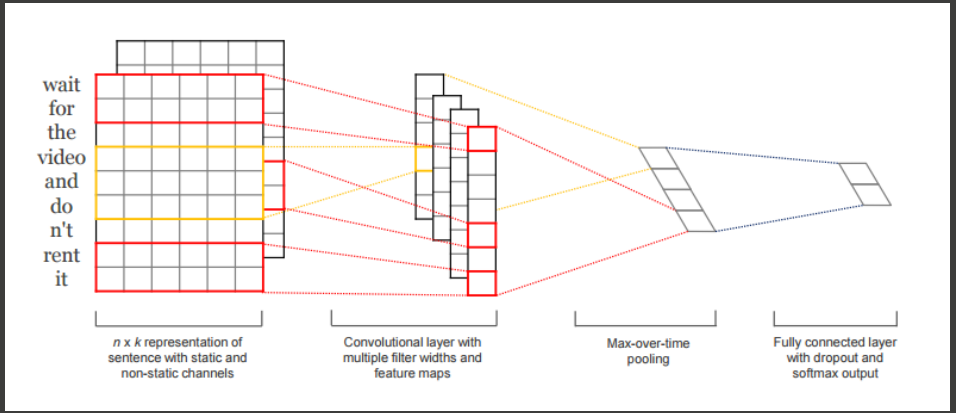
pretrain





多通道分类架构

- 可训练的词向量
- 不可训练的词向量
- 拼音对应的向量
- ...

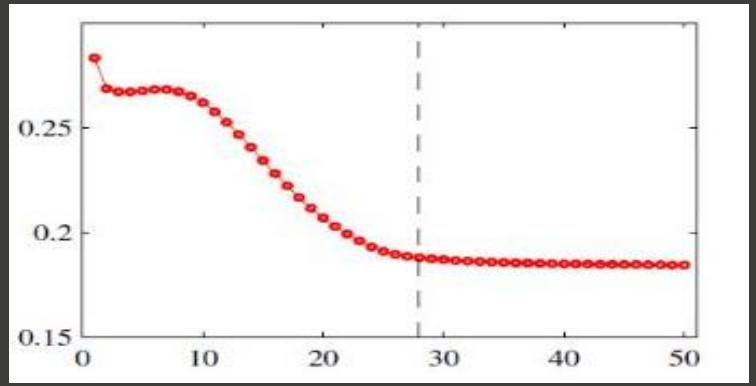


样本不均衡以及过拟合

问题1：样本不均衡

解决方案：

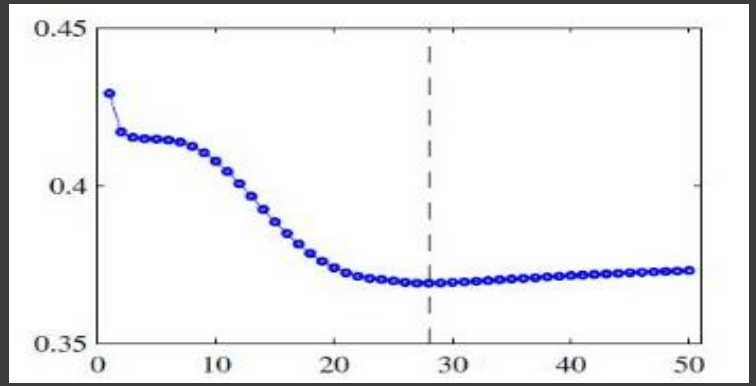
- 加权loss
- 过采样/欠采样



问题2：过拟合

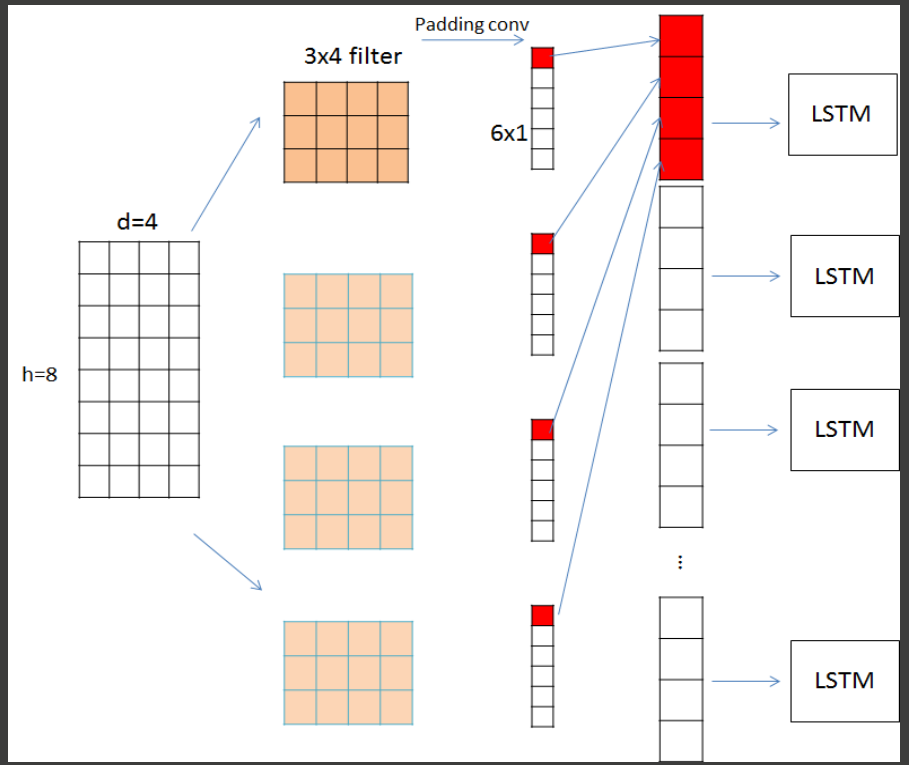
解决方案：

- early stop
- dropout
- L2/L1



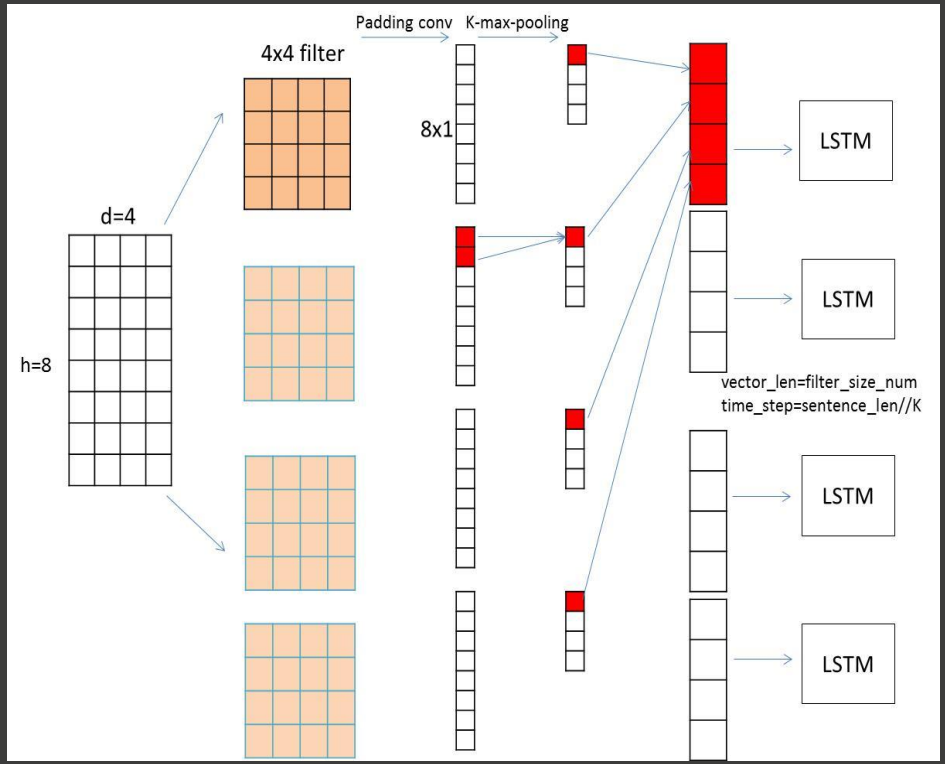
C_LSTM

1. 词嵌入层
2. 卷积层
   高维特征抽取
3. 时序特征组合层
4. LSTM层
   作用于高维时序特征
5. 全连接层以及softmax层
6. 交叉熵损失以及L2正则化



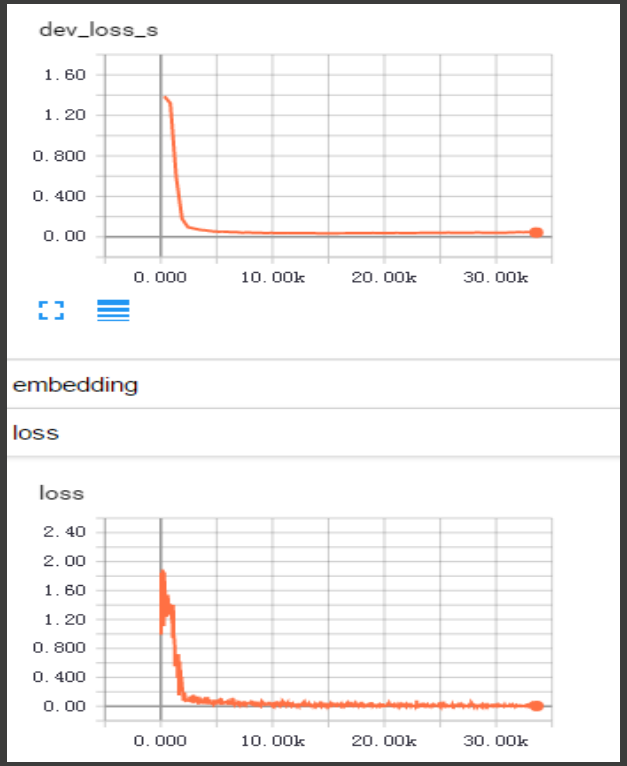
C_LSTM

1. 词嵌入层
2. 卷积层
   高维特征抽取
3. K-maxpooling
   减少LSTM时序维度
4. 时序特征组合层
5. LSTM层
   作用于高维时序特征
6. 全连接层以及softmax层
7. 交叉熵损失以及L2正则化

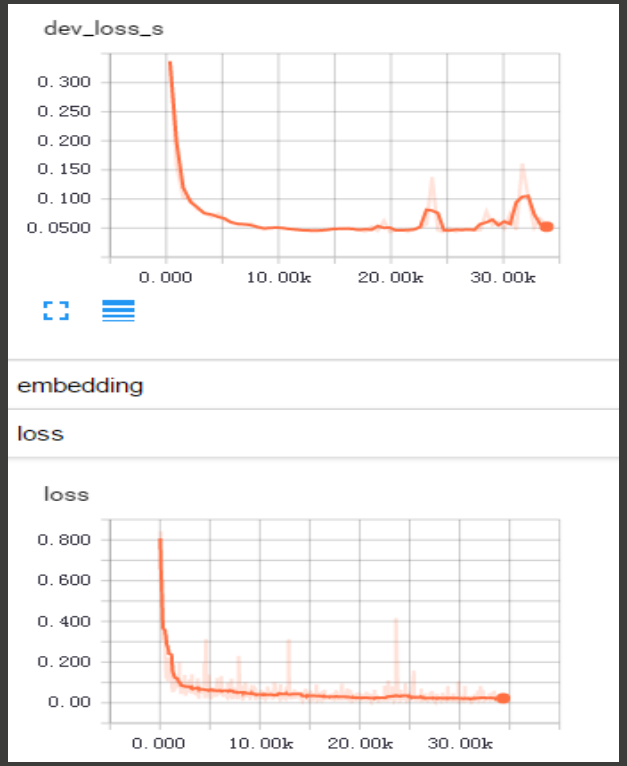


C_LSTM VS TEXT_CNN

C_LSTM



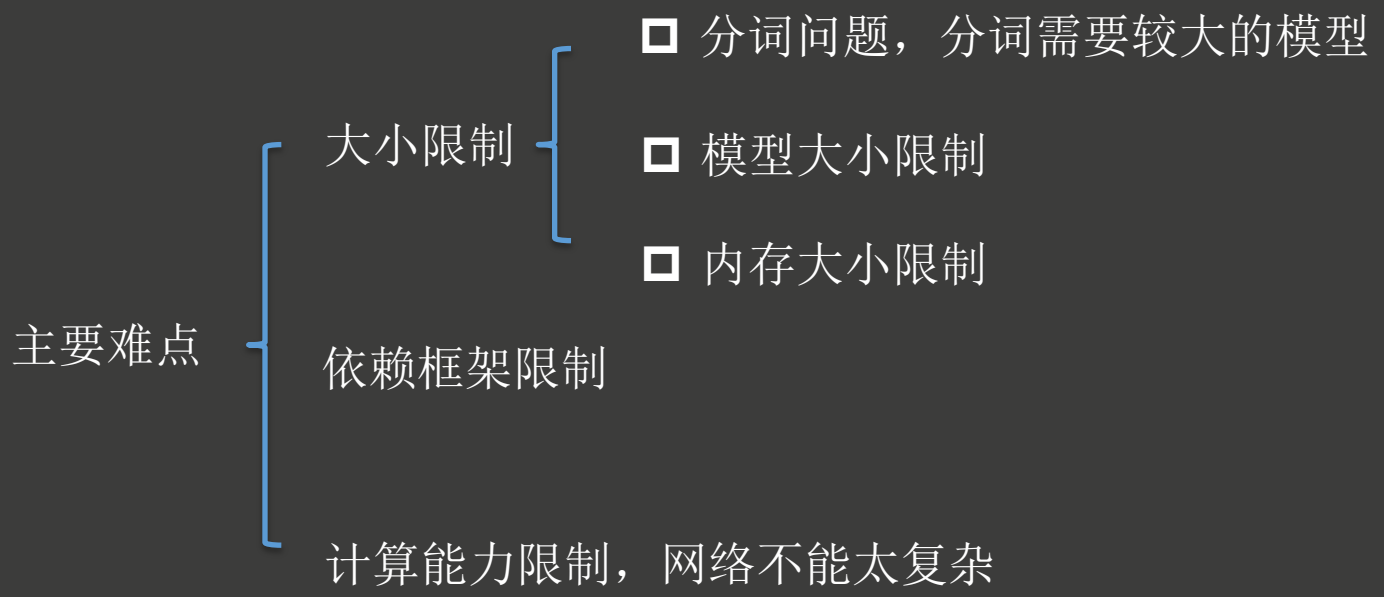
TEXT_CNN



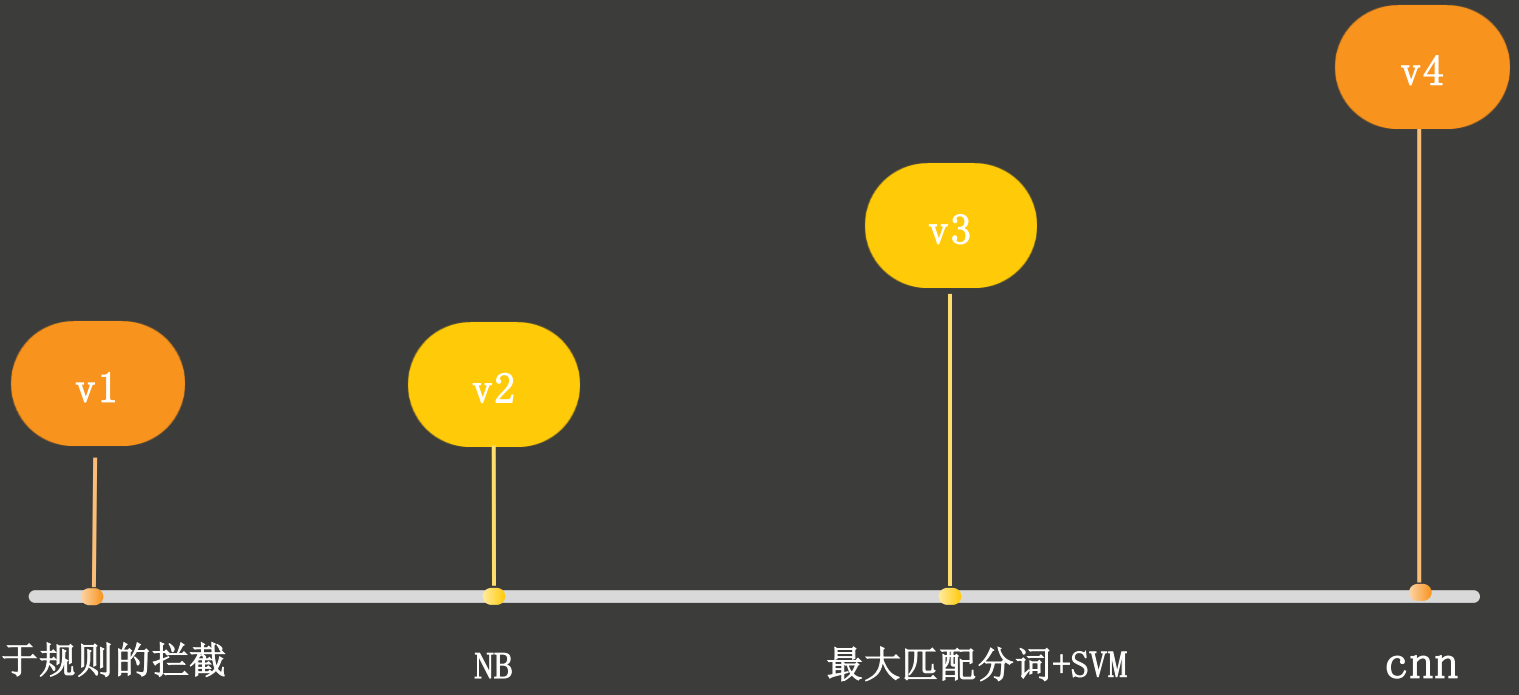
主要内容

- 短信拦截应用场景
- 云端算法实现以及上线
- 移动端算法实现以及落地

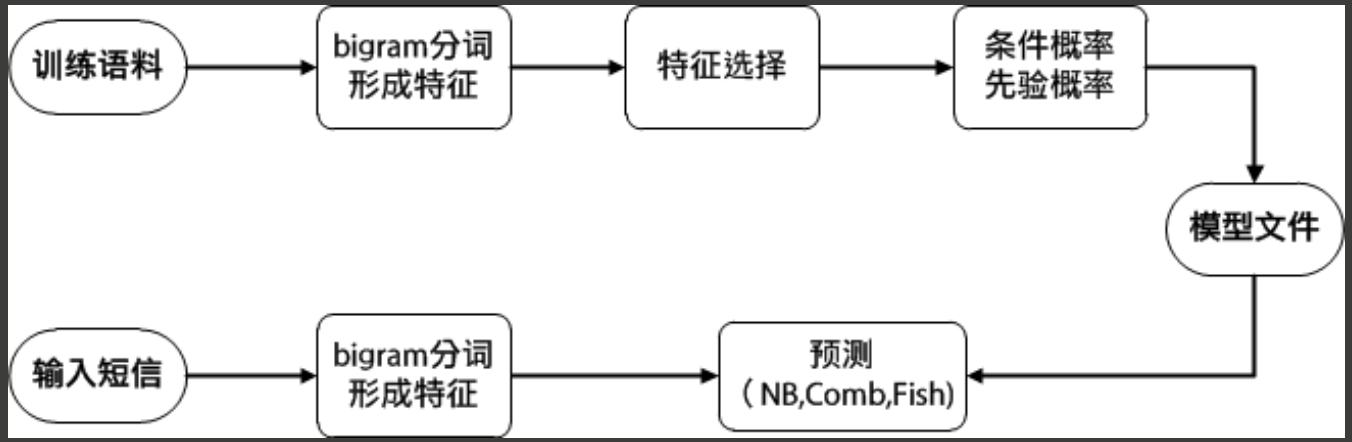
深度学习模型在移动端实现的难点



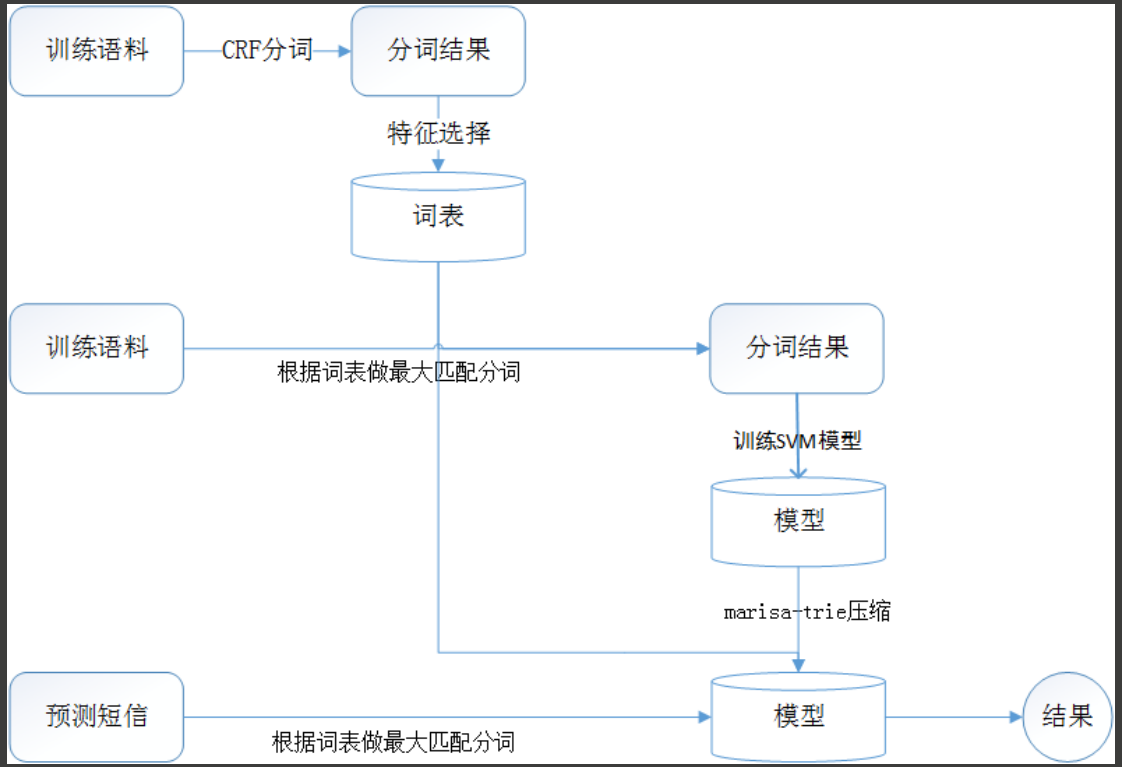
客户端短信拦截算法演变



bigram分词+NB


训练语料
bigram分词
形成特征
特征选择
条件概率
先验概率
模型文件
输入短信
bigram分词
形成特征
预测
（NB,Comb,Fish）

基于词典的最大匹配分词+SVM



CNN算法

- 压缩模型
- 压缩依赖库
- 验证准确率
- android端以及iOS端落地

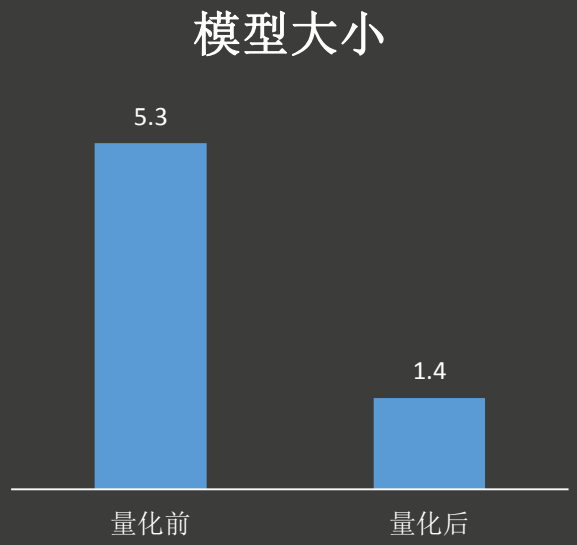
压缩模型

1.特征降维

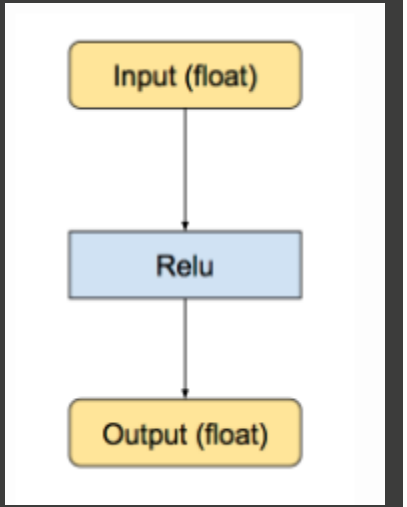
- 抽取关键特征
- 降维后，省去大量词向量空间
- 降维前模型大小109M，降维后模型大小5.3M

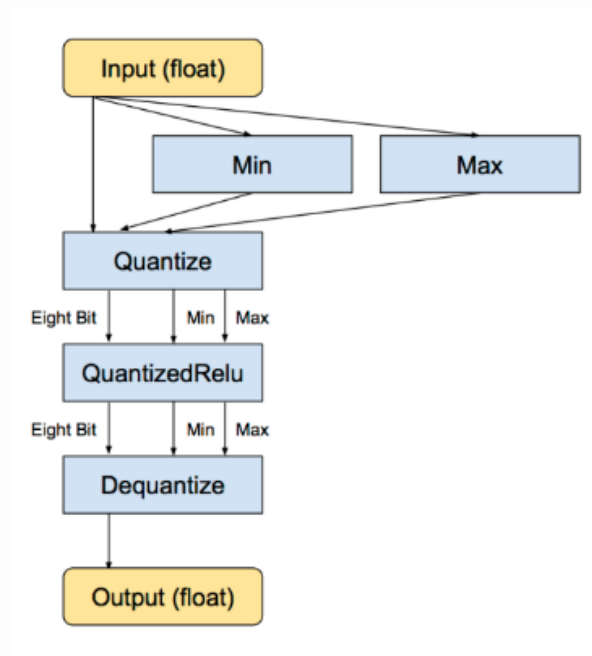
2.量化模型

将模型中的参数8bit定点化量化



量化模型





为什么量化是可能的？

神经网络本身是抗噪和鲁棒的

进行推断时，减低一定的精度，不会产生太大影响。

如何量化和量化意义？

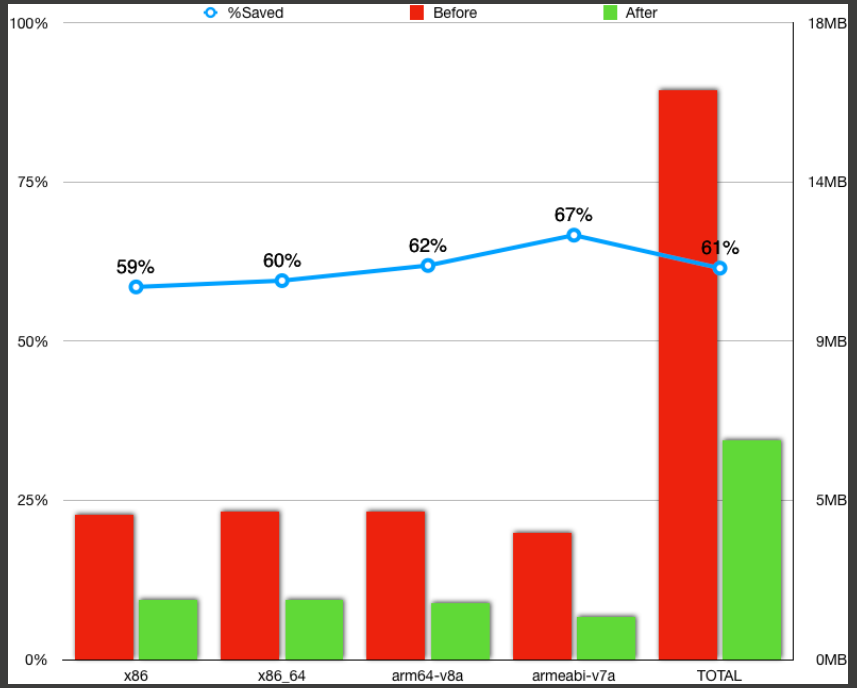
神经网络参数每层数值在一定的范围内，如[-6.0,4.0]，

有大量论文研究表明确认最大值和最小值后每层数据使用8bit定点化量化可以满足推断计算

量化可以减少约3/4空间大小

压缩依赖库

- 提取模型中的算子列表
- 生成只包含算子列表的SO



验证准确率

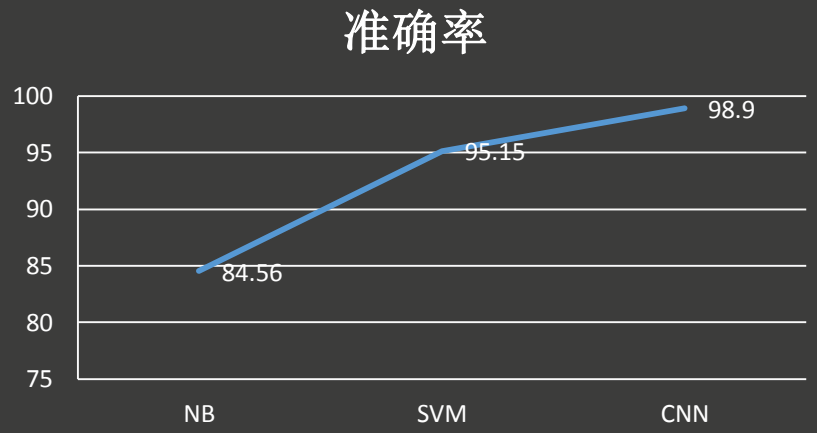
精度损失情况

测试短信：
会计培训！家财万贯，不如一技在手！会计是招工单位必备人才！艺不压身！早学早受益！报名xxxxxx微信号：lxxxxxx316

模型压缩前：
PC上python预测该短信结果：
[0.05079584 0.94151241 0.0041042 0.00358752]
Android上预测结果：
[0.050795842 0.9415124 0.0041042017 0.0035875132]

模型压缩后：
PC上用python预测：
[0.06486285 0.92589331 0.0052266 0.00401725]
Android上预测结果：
[0.05891386 0.93307567 0.0043378514 0.0036725344]

准确率提升

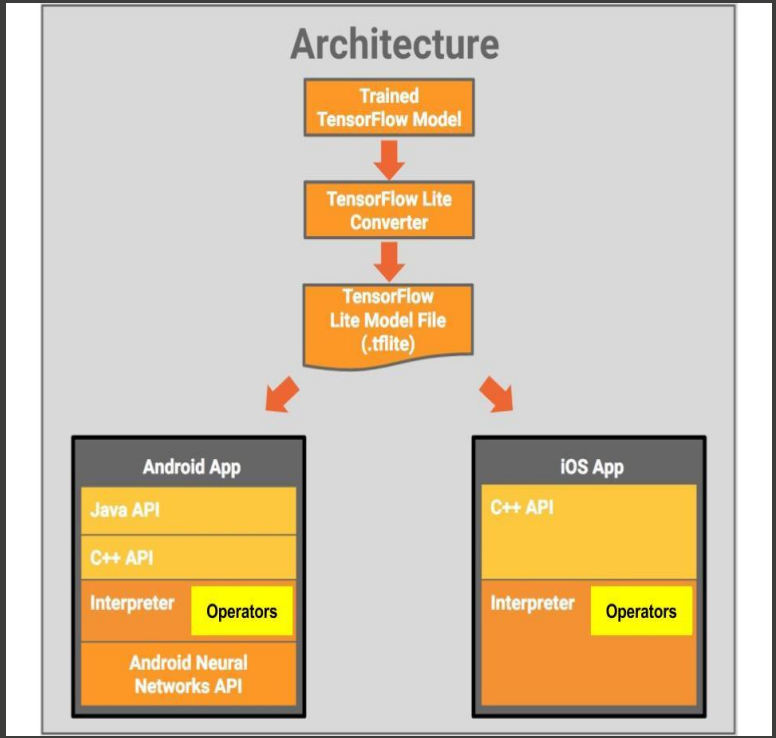


类型	精确率	召回率	f1 score
白	0.98395119	0.99272871	0.98832046
广告	0.99068715	0.97777632	0.9841894
违法	0.99686289	0.99911223	0.99798629
诈骗	0.99926534	0.9973115	0.99828746

Android端落地

Tensorflow Lite

- embedding_lookup函数中的gather算子，Lite暂时不支持
- 需要将embedding_lookup放到模型外面来做
- Lite依赖的so大小为1M左右



iOS端落地

- 依赖Tensorflow Lite库预测
- CoreMltools : Keras model → CoreML model
- Tf-CoreMl : TF model → CoreML model

防骚扰大师

手机卫士iOS版

谢 谢!

技术交流（干货）：奇卓社（360移动技术微信公众号）