利用人工智能解决移动端性能测试痛点

刘永刚 高级测试开发工程师

email: liuyonggang@360.cn



个人简介



刘永刚 奇虎360高级测开工程师

2017年加入奇虎360,现在主要从事手机卫士的功能自动化、性能测试及相关测试工具的开发工作。

曾就职于"LG数字电视研究所"、"乐视致新",近10年的Android App开发、测试相关工作,积累了较丰富的移动端测试经验。

分享目的



本次分享通过App的启动时间为例,结合当前较火爆的人工智能技术对比传统的测试技术来展示人工智能在测试领域的优势。

详细介绍人工智能的核心概念、构造模型、训练模型、模型预测等知识点,协助相关童鞋快速理解人工智能的概念,写出自己的人工智能作品。本次分享实则抛砖引玉,希望更多的测试人去学习人工智能,使之在测试领域发挥更大的作用。

性能测试项





常规启动时间测试方法



am start 埋点log

不准确不能对比竞品

问题

手动录屏计算

效率低成本高

相机拍摄计算

覆盖机型少

基于人工智能的启动时间测试



基于人工智能的启动时间测试过程:









人工

人工 + 智能

智能

机器

学习

监督

7

无监督

 从数据中提取知识,它是统计学、人工智能和计算机 科学交叉的研究领域,又叫"预测分析"或"统计学 习"。

人工将成对的已知输入和输出提供给算法,算法会找 出一种方法,根据给定的输入给出预期输出。

没有已知输出,只有输入数据并需要从这些数据中提取知识。

学习



样本

• 每一对人工提供的数据就是一个"样本"。

特征值

• 样本的属性称之为"特征值"。

标签

• 样本的已知输出(即类别)。

模型

各机器学习库中包含很多算法用来从人工提供的数据对中提取特征值,每一个算法就是一个模型(每一个算法都有相关的类实现),模型使用前必须先实例化模型对象。



训练

• 让模型进行学习的过程。

固化

将训练完后的模型对象保存到本地文件中,下次使用时直接加载即可。

分类

- 也叫预测,使用训练的模型对没有训练过的输入数据进行判定,对其输出(即类别)做出决策。
- 训练完毕的模型对象如果能够对未被学习的输入做 出正确的判断则称该模型的泛化能力强,反之则是 泛化能力弱。

泛化

机器学习框架



Scikit-Learn

开源项目,免费使用和分发,轻松获取其源代码,包含许多最先进的机器学习算法(http://scikit-learn.org/stable/documentation)。

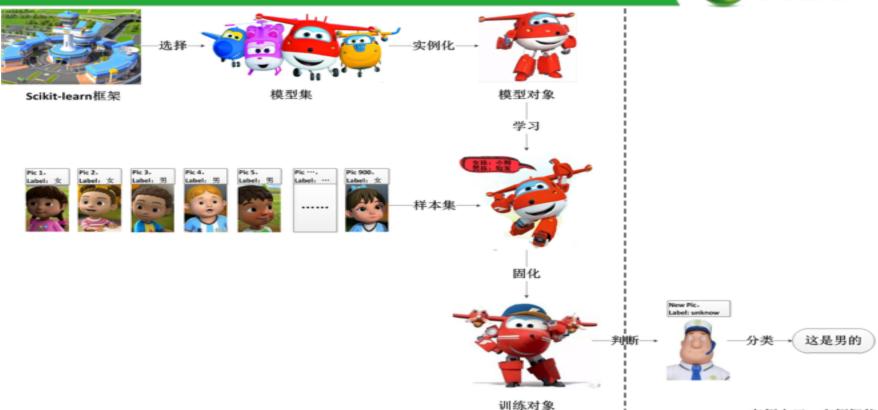
Tensor Flow

• 另一个项目, 其他略。

各概念关系图

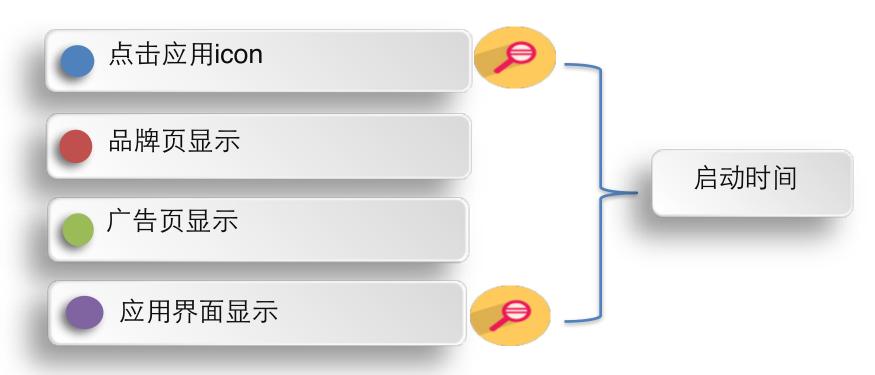


左侧人工+右侧智能



应用启动需求分析



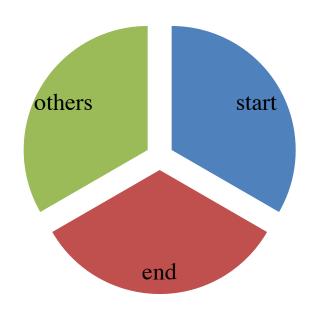


应用启动需求分析



三分类问题

场景图片	标签
App icon置灰	start
App home界面	end
其他界面	others



样本收集



100

												y www	.360.CN	
Activitie	s 🔚 Files	*					三 10	8:39 •					-	44 ⋒ →
(25)	c > +	☆ Home	Desktop	Demo start								C	ર ≔ ≣	000
	***	***	***	***		7.	2*1	241	201	2**	241	201	200	
	193fec711_	193fec711_	193fec711_	193fec711_			193fec712_				193fec713_			
?	104.jpg	105.jpg	106.jpg	107.jpg	104.jpg	105.jpg	106.jpg	107.jpg	104.jpg	105.jpg	106.jpg	107.jpg	104.jpg	
	100	- C. T.	45276	* :		**		* :						
>_	193fec714_ 105.jpg	193fec714_ 106.jpg	193fec714_ 107.jpg			612QKCQ92 29FQ1_093. jpg						612QKCQ92 29FQ1_099. jpg		
												#	#	
				612QKCQ92 29FQ1_104. jpg									864acaf01_ 073.jpg	
	864acaf01	864acaf01	864acaf01	864acaf01	864acaf01	864acaf01	864acaf01	864acaf01	864acaf01	864acaf01	864acaf01	864acaf01	864acaf01	
	074.jpg	075.jpg	076.jpg	077.jpg	078.jpg	079.jpg	080.jpg	081.jpg	082.jpg	083.jpg	084.jpg	085.jpg	086.jpg	
	#		#											
	864acaf01_ 087.jpg	864acaf01_ 088.jpg	864acaf01_ 089.jpg	a2e8ce8d1_ 063.jpg	a2e8ce8d1_ 064.jpg	a2e8ce8d1_ 065.jpg	a2e8ce8d1_ 066.jpg	a2e8ce8d1_ 067.jpg	a2e8ce8d1_ 068.jpg	a2e8ce8d1_ 069.jpg	a2e8ce8d1_ 070.jpg	a2e8ce8d1_ 071.jpg	a2e8ce8d1_ 072.jpg	
::::	a2e8ce8d1_ 073.jpg	a2e8ce8d1_ 074.jpg	a2e8ce8d1_ 075.jpg	a2e8ce8d1_ 076.jpg	a2e8ce8d1_ 077.jpg	a2e8ce8d1_ 078.jpg	a2e8ce8d1_ 079.jpg	a2e8ce8d1_ 080.jpg	a2e8ce8d1_ 081.jpg	a2e8ce8d1_ 082.jpg	a2e8ce8d1_ 083.jpg	a2e8ce8d1_ 084.jpg	a2e8ce8d1_ 085.jpg	

样本收集



						www.360.Ch							ļ			
Activitie	es 🔚 Files 🕆	•			1		三 10	3:39 •					- A	41 B +		
	< > 4	☆ Home	Desktop	Demo end	end							•	ર = ≣	000		
• • • • • • • • • • • • • • • • • • •	193fec71_ 428.jpg	193fec71_ 429.jpg	193fec71_ 434.jpg	193fec71_ 440.jpg	193fec71_ 459.jpg	29FQ_921. jpg		29FQ_926. JPg	612QKCQ92 29FQ_929. jpg		612QKCQ92 29FQ_935. jpg	612QKCQ92 29FQ_939. jpg	29FQ_947. Jpg			
<u>-</u>	612QKCQ92 29FQ_949. jpg	612QKCQ92 29FQ_953. jpg	612QKCQ92 29FQ_959. jpg	612QKCQ92 29FQ_961. jpg	612QKCQ92 29FQ_964. jpg	612QKCQ92 29FQ_969. jpg	612QKCQ92 29FQ_972. jpg	612QKCQ92 29FQ_975. jpg	612QKCQ92 29FQ_980. jpg	612QKCQ92 29FQ_986. jpg	612QKCQ92 29FQ_987. jpg	612QKCQ92 29FQ_990. jpg	612QKCQ92 29FQ_994. jpg			
	612QKCQ92 29FQ_999. jpg	612QKCQ92 29FQ_1002. jpg		612QKCQ92 29FQ_1012. jpg		612QKCQ92 29FQ_1018. Jpg					612QKCQ92 29FQ_1033. jpg					
	612QKCQ92 29FQ_1045. jpg				612QKCQ92 29FQ_1062. jpg	612QKCQ92 29FQ_1068. jpg	612QKCQ92 29FQ_1074. jpg		612QKCQ92 29FQ_1085. jpg		612QKCQ92 29FQ_1095. jpg					
	612QKCQ92 29FQ_1111. jpg	851QFDSF2 28UZ_672. jpg	851QFDSF2 28UZ_673. jpg	851QFDSF2 28UZ_676. jpg	851QFDSF2 28UZ_680. jpg	851QFDSF2 28UZ_685. jpg	851QFDSF2 28UZ_689. jpg	851QFDSF2 28UZ_693. jpg	851QFDSF2 28UZ_698. jpg	851QFDSF2 28UZ_702. jpg	851QFDSF2 28UZ_708. jpg	851QFDSF2 28UZ_711. jpg				
:::																

样本收集



												_ ~~~	.360.01	
Activitie	s 🔚 Files	•					三 10	3:40 •					.ž.	41 B →
	c > 4	☆ Home	Desktop	Demo othe	-othe	ers						•	ર ≔ ≣	000
	612QKCQ92 29FQ_001. jpg	612QKCQ92 29FQ_006. Jpg	612QKCQ92 29FQ_011. jpg	612QKCQ92 29FQ_020. Jpg	612QKCQ92 29FQ_024. Jpg	612QKCQ92 29FQ_026. Jpg	612QKCQ92 29FQ_029. Jpg	612QKCQ92 29FQ_033. Jpg	612QKCQ92 29FQ_037. Jpg	612QKCQ92 29FQ_040. Jpg	612QKCQ92 29FQ_047. Jpg	612QKCQ92 29FQ_054. Jpg	612QKCQ92 29FQ_060. Jpg	
	612QKCQ92 29FQ_065. jpg	612QKCQ92 29FQ_069. Jpg	612QKCQ92 29FQ_073. Jpg	612QKCQ92 29FQ_078. Jpg	612QKCQ92 29FQ_082. Jpg	612QKCQ92 29FQ_087. Jpg	612QKCQ92 29FQ_092. Jpg	612QKCQ92 29FQ_097. Jpg	612QKCQ92 29FQ_101. jpg	612QKCQ92 29FQ_104. jpg	612QKCQ92 29FQ_108. jpg	612QKCQ92 29FQ_114. jpg	612QKCQ92 29FQ_120. Jpg	
	612QKCQ92 29FQ_122. Jpg	612QKCQ92 29FQ_123. Jpg	612QKCQ92 29FQ_124. jpg	612QKCQ92 29FQ_125. Jpg	612QKCQ92 29FQ_127. jpg	612QKCQ92 29FQ_129. jpg	612QKCQ92 29FQ_130. Jpg	612QKCQ92 29FQ_131. jpg	612QKCQ92 29FQ_132. jpg	612QKCQ92 29FQ_133. jpg	612QKCQ92 29FQ_134. jpg	612QKCQ92 29FQ_135. jpg	612QKCQ92 29FQ_136. jpg	
	612QKCQ92 29FQ_137. jpg	612QKCQ92 29FQ_138. jpg	612QKCQ92 29FQ_139. jpg	612QKCQ92 29FQ_140. jpg	612QKCQ92 29FQ_141. jpg	612QKCQ92 29FQ_148. jpg	612QKCQ92 29FQ_162. jpg	612QKCQ92 29FQ_163. jpg	612QKCQ92 29FQ_166. jpg	612QKCQ92 29FQ_167. jpg	612QKCQ92 29FQ_169. jpg	612QKCQ92 29FQ_173. jpg	612QKCQ92 29FQ_178. jpg	
	612QKCQ92 29FQ_184. jpg	612QKCQ92 29FQ_189. jpg	612QKCQ92 29FQ_193. jpg	612QKCQ92 29FQ_195. jpg	612QKCQ92 29FQ_199. jpg	612QKCQ92 29FQ_205. JPg	612QKCQ92 29FQ_209. JPg	612QKCQ92 29FQ_213. jpg	612QKCQ92 29FQ_218. jpg	612QKCQ92 29FQ_221. jpg	612QKCQ92 29FQ_227. jpg	612QKCQ92 29FQ_231. Jpg	612QKCQ92 29FQ_234. jpg	
:::	Coe	(Ge	(0)	(0							8	€.	•	

模型训练、固化



```
def training model():
   tr imq是一个列表,其中每一项数据值都是从样本集的一个图片转成的数组数据;
   tr label是一个列表,其中每一项数据值都是对应着tr img数据值的标签
   tr imq, tr label = get train image datas()
   print ("tr img={}, length={}".format(tr img, tr img. len ()))
   print ("tr label={}, length={}".format(tr label,
tr label. len ()))
   # 实例化模型对象
 linear svcClf = svm.LinearSVC()
   # 提供学习样本给对象学习,也可以交过训练模型
 linear svcClf.fit(tr img, tr label)
   # 固化模型
 joblib.dump(linear svcClf, "linear svcClf train model.m")
```

测试过程录制、拆帧



adb shell screenrecord /sdcard/filename.mp4

ffmpeg -i filename.mp4 -s w*h -r X icname.jpg

模型预测



```
def learning by modle(model):
    # 加载已固化模型
 clf = joblib.load(model)
    # tst img是一个列表,其中每个数据值是拆帧视频得到的图片所转换成的数组
 tst img = get test image()
   # 对数据执行分类工作
 predicts = clf.predict(tst img)
    for index, result in enumerate(predicts):
       if result == "start":
           start id = index + 1
           break
    for index1, result1 in enumerate(predicts):
       if result1 == "end":
           end id = index1 + 1
           break
    return start id, end id
```

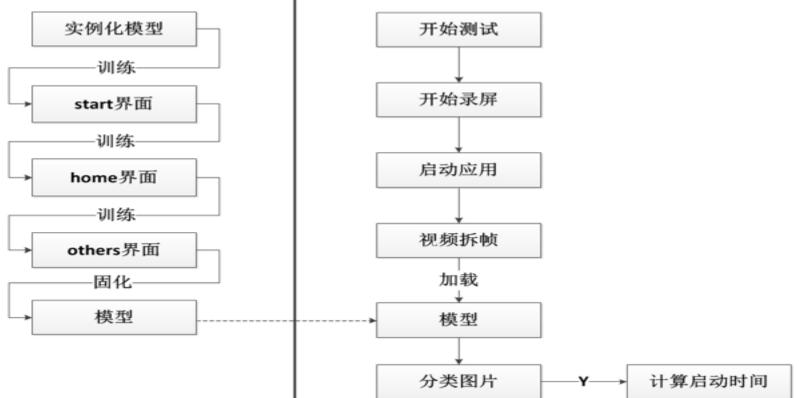
启动时间计算



启动时间=(end_id - start_id)*(1000/X)

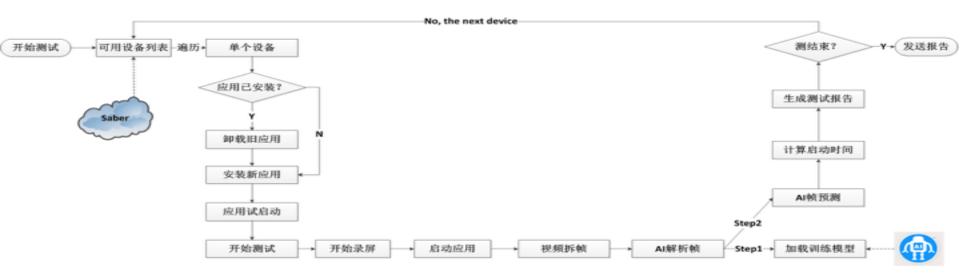
核心逻辑





基于ai启动时间测试框架





报告展示





单个手机启动时间平均值

单个手机启动时间最大值

单个手机启动时间最小值

按品牌统计启动时间

按rom版本统计启动时间

谢谢!



