利用CNN实现无需联网的图像识别

李永会 百度多模交互搜索部 资深工程师

个人介绍

- 2014年加入百度
- 多模交互搜索部
- 图像&语音搜索客户端负责人
- 专注ARM平台架构
- 深度学习移动端落地
- 计算机视觉移动应用

李永会

拍照搜索

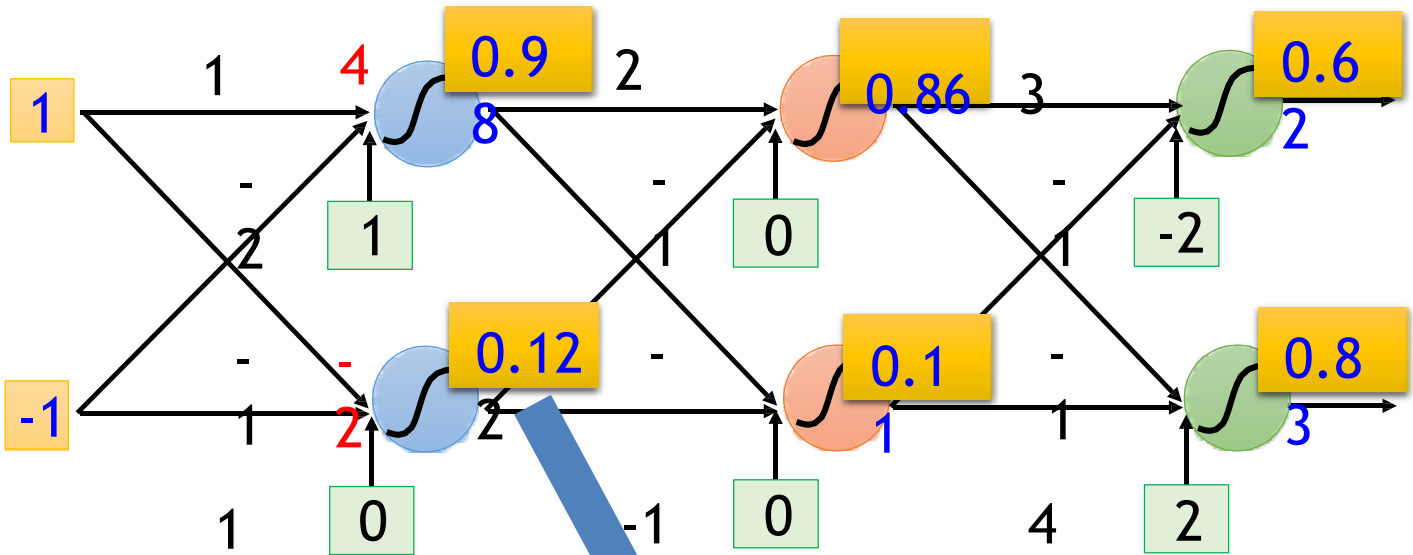
机器学习

$f($  $=$ “你好”

$f($  $)=$ “猫”

$f($ “嗨！” $)=$ “你好”

全链接前向传播



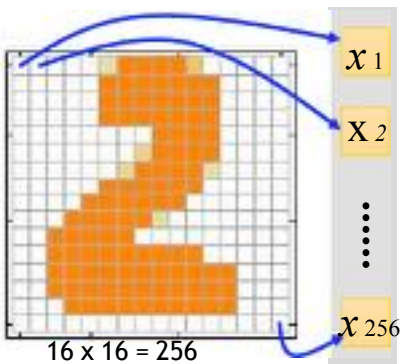
“猫”

$$S(x) = \frac{1}{1 + e^{-x}}$$

识别过程

$f($  $) =$ “2”

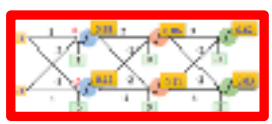
Input



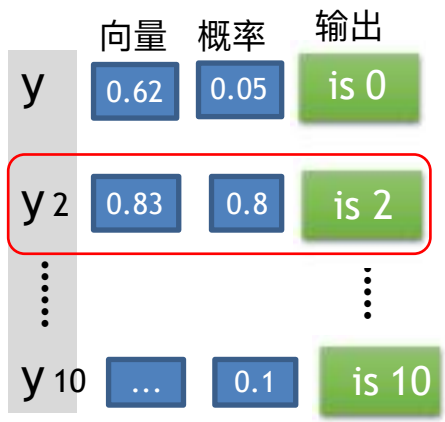
16 x 16 = 256

Ink → 1

No ink → 0

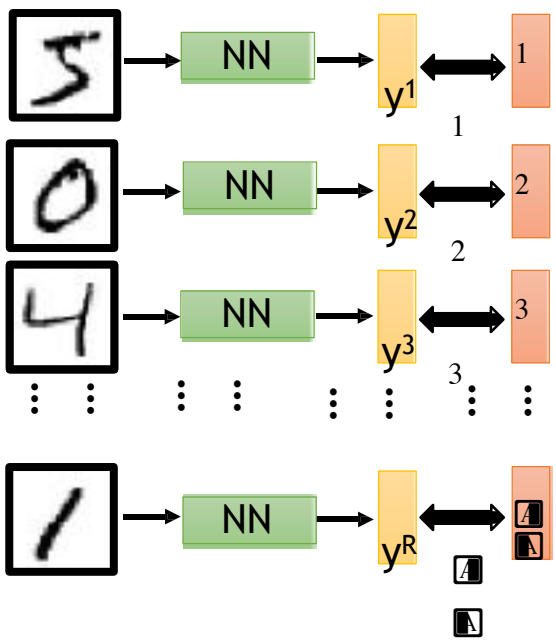


Output

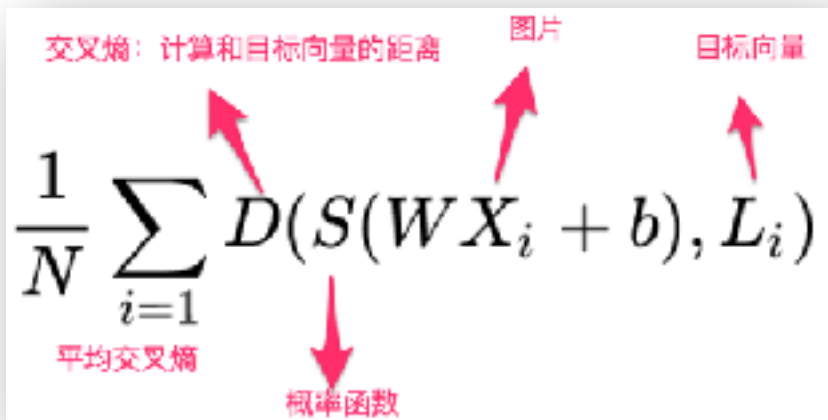


The image is “2”

训练过程


NN
NN
NN
NN
y1
y2
y3
yR
1
2
3
1
2
3

$f($  $) =$ “2”


交叉熵：计算和目标向量的距离
图片
目标向量
$\frac{1}{N}\sum_{i=1}^{} D(S(WX_i+b), L_i)$
平均交叉熵
概率函数

移动端和Server分工

- 客户端训练 + 客户端识别
- 服务端训练 + 识别
- 服务端训练 + 客户端识别

移动端只适合识别过程

Model File

PC Server 训练模型文件

移动端加载模型进行识别

卷积和池化

原图

1	-1	-1
-1	1	-1
-1	-1	1

-1	1	-1
-1	1	-1
-1	1	-1

两个卷积核

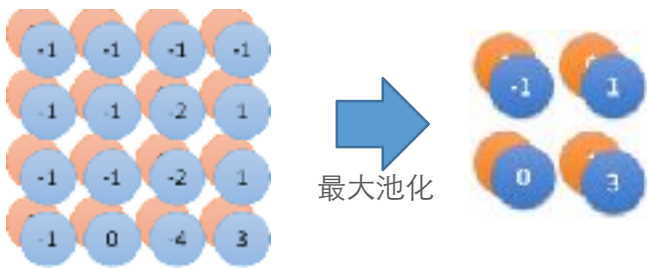
Only 9 x 2 = 18 parameters

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	1	1	0	0
0	1	1	0	0

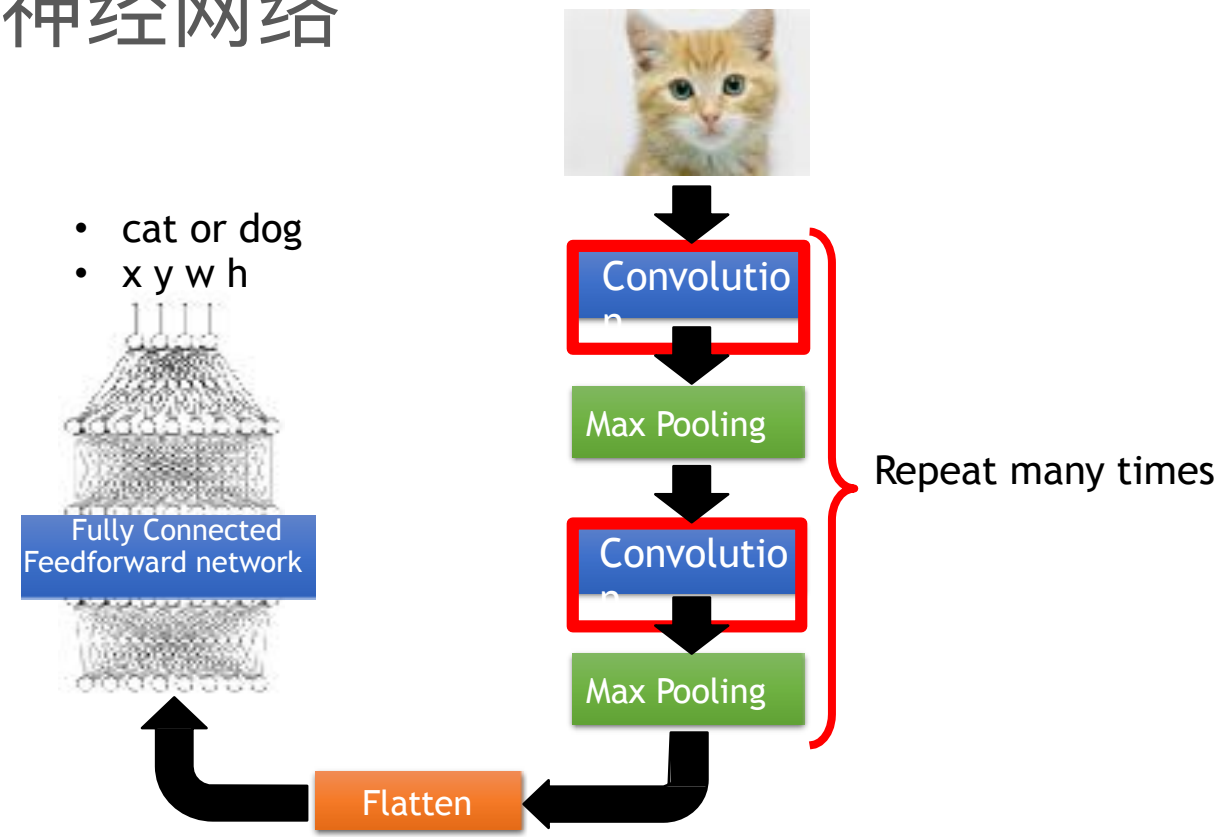
Image

4		

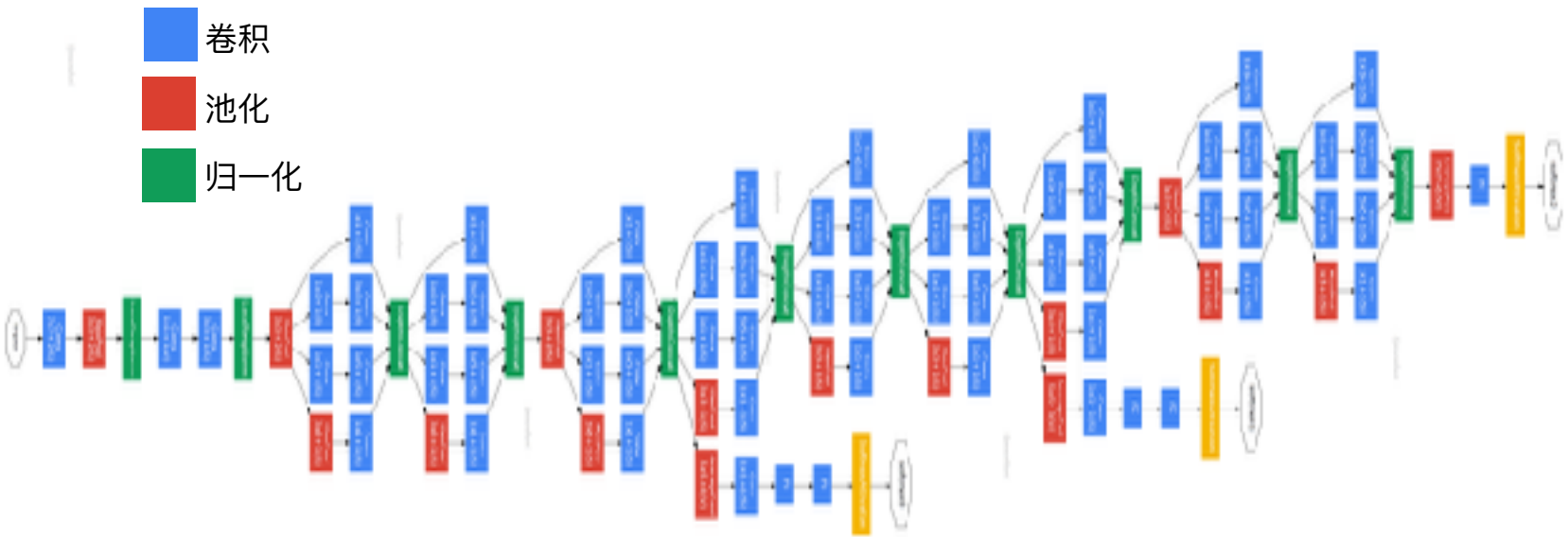
Convolved Feature



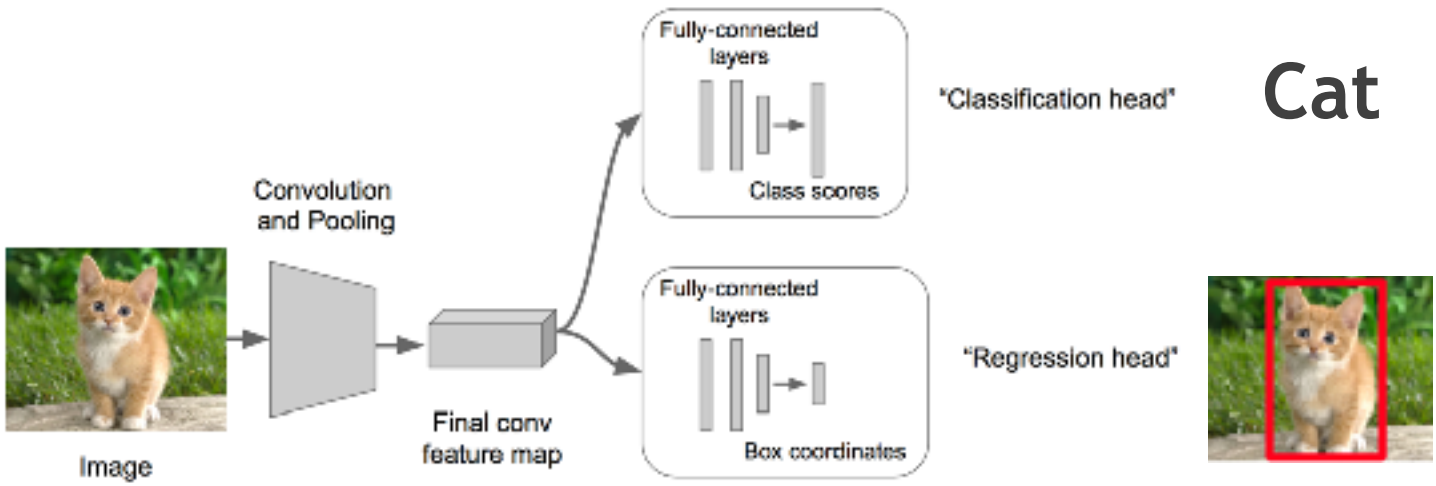
卷积神经网络


cat or dog
x y w h
Fully Connected Feedforward network
Convolution
Max Pooling
Convolution
Max Pooling
Repeat many times
Flatten

GoogLeNet v1



分类和框选 - 权值共享


Fully-connected layers
Class scores
"Classification head"
Cat
Convolution and Pooling
Fully-connected layers
Box coordinates
"Regression head"
Final conv feature map
Image

IOS落地难点

- 内存：服务端限制不严格 - 移动端的内存有限
- 耗电量：服务端不限制 - 移动端严格限制
- 图搜插件增量大小：手百移动端不能超过100K
- 模型大小：常规模型体积 500M起移动端不能超过10M
- 加密问题：服务端平台无需考虑模型泄露问题

为什么选择移植Caffe

- 可读性
- 通用性
- 图像领域应用已久
- 移植成功案例

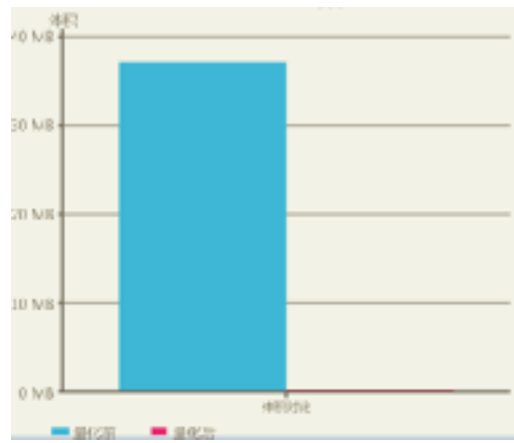
精简caffe

- blas : openblas切到cblas
- glog : handwriting
- gflag : simplification
- protobuf : handwriting
- Backpropagation : cut

手写 + 精简

- Blob
- InnerProductLayer
- ReLU
- MaxPooling
- AveragePooling
- CrossChannelLRN
- ConvolutionLayer
- Concat
- Net

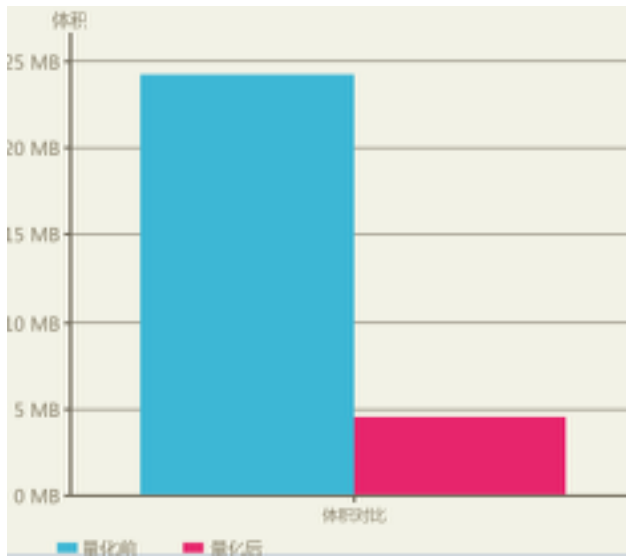
drop + 精简



37MB -> 100k

模型量化

- 粗分类+单框选，模型体积24.2MB
- GooLeNet v1模型的参数为float 32bit，图搜以8bit来存储参数。再次减小为6.4MB
- 压缩后6.4 -> 4.5MB
- 模型体积缩小后内存也因此减小



24.2 -> 4.5MB

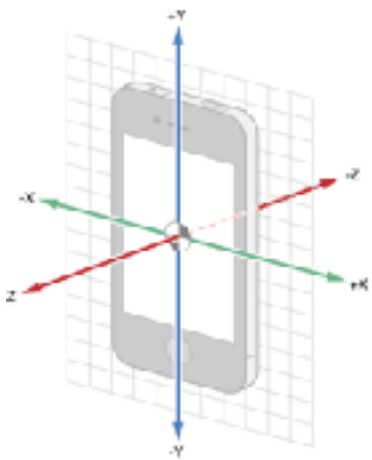
精度问题

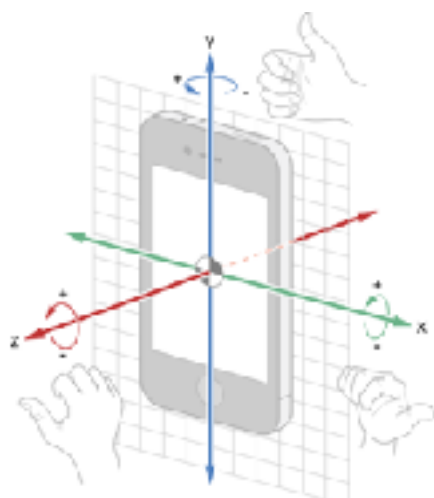
- 量化过程中将float直接强转为uint会导致参数整体偏小，结果也会偏小，所以在转换过程中，需要加入随机分桶操作。
- 例如：对于参数2.3，让其70%的概率转换成2，30%的概率转换成3。

耗电量

- 在移动客户端运行神经网络耗电量巨大，采用以下策略：
  - 用户手机达到稳定后一段时间开始识别
  - 通过选取合适的识别间隔

陀螺仪&加速计





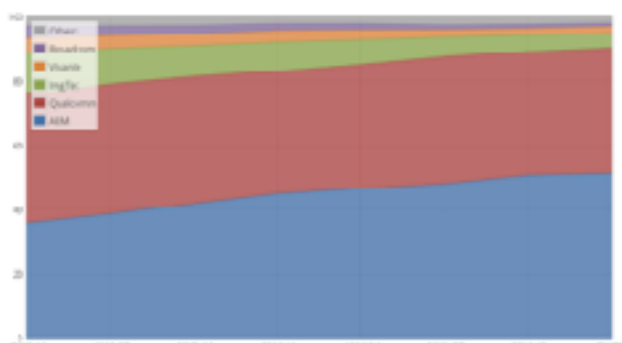
IOS最终效果

- 模型4.5MB
- iphone 6s速度170ms
- 耗电低level
- 准确率90%

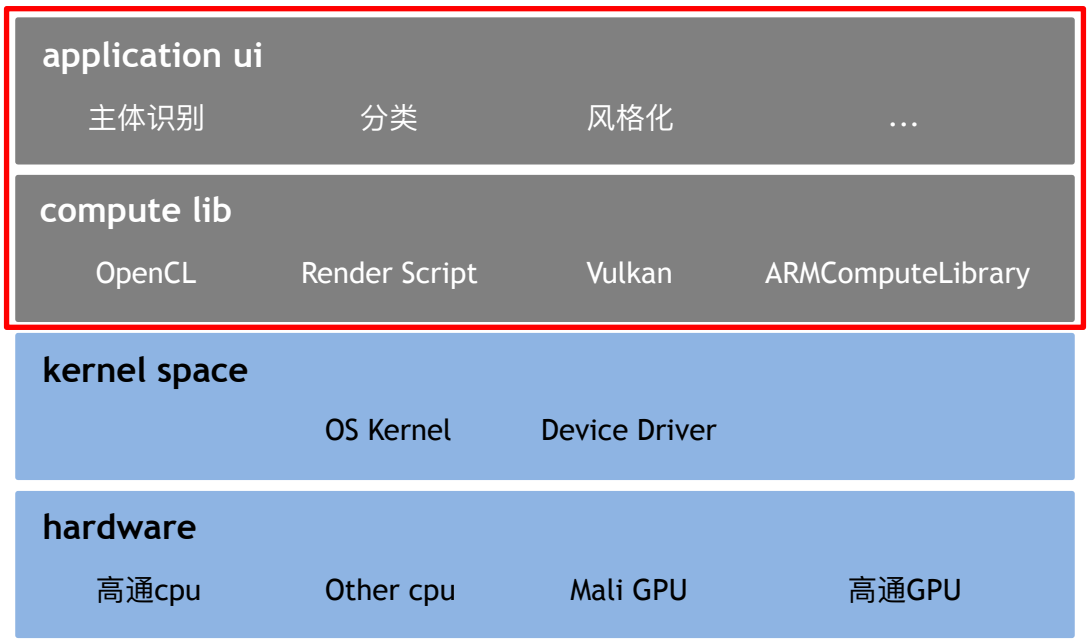
Android硬件

- CPU： 高通 & (三星、联发科、华为)
- GPU： Mali GPU
- CPU门槛1：骁龙600以上
- GPU门槛2：Mali T820 4核以上

- CPU 98.1% 是ARMv7
- GPU ARM: 51.3% Qualcomm: 39.2%



Android深度学习软件现状

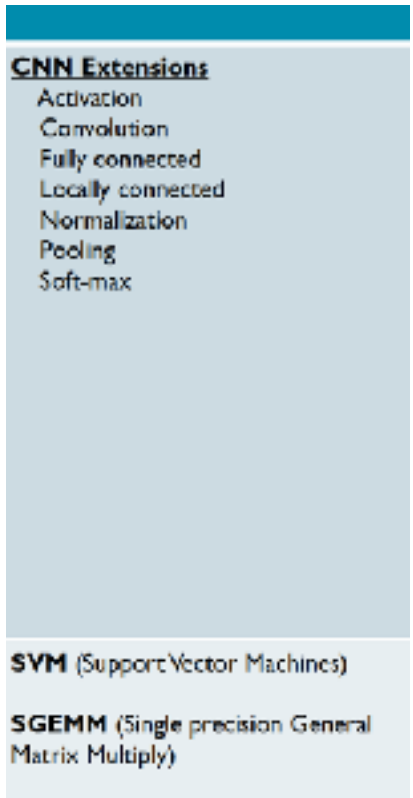


Android 运算库

- Render Script - 坑多，速度不稳定
- Open CL - 表现较好
- Vulkan - 计算前景不明朗，支持版本太少
- ARMComputeLibrary - 未来看好

与ARM协作

- ARMComputeLibrary 3月底开源
- 4月初和arm团队深入沟通
- 相互提供建议
- 5月初由于数据结构和网络现阶段支持不足够，百度独立启动研发，并启动基于gpu加速方案

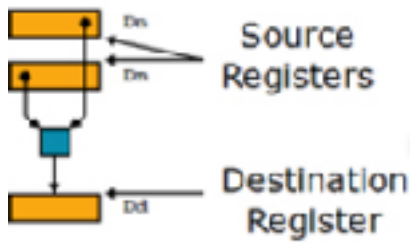


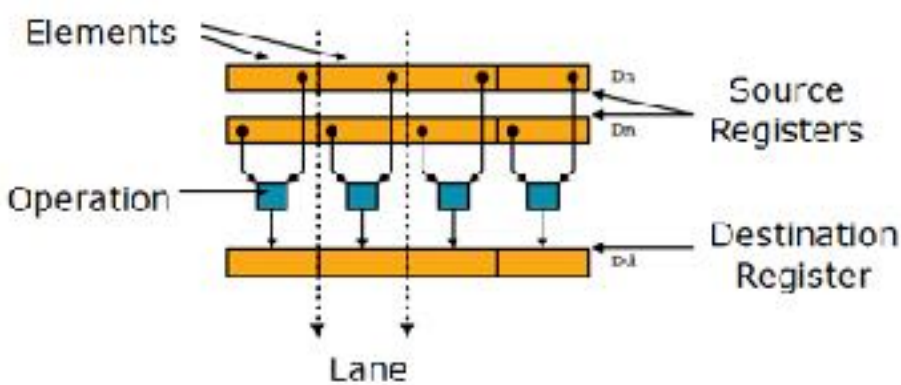
Tricks - CPU Affinity

- CPU Affinity
- biglittle

Tricks - NEON

- 卷积
- 池化
- LocalRespNorm





Tricks - 汇编

- Assembly文件
  纯汇编文件，后缀为”.S”或”.s”。注意对寄存器数据的保存
- inline assembly内联汇编
  在C/C++代码中嵌入汇编，调用简单，无需手动存储寄存器

Tricks

- loop unrolling
- static

MobileNet应用

Depthwise Separable Convolution由两部分组成：

- depthwise convolutions
- pointwise convolutions (simple 1×1 convolution)

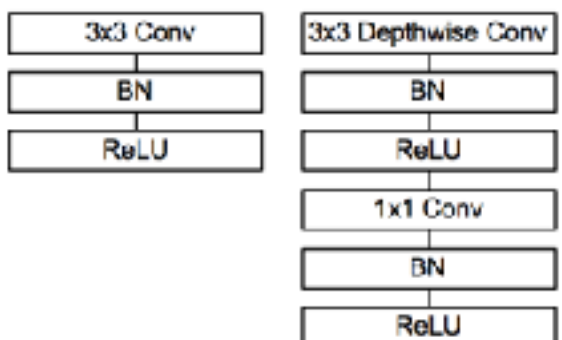


Figure 3. Left: Standard convolutional layer with batchnorm and ReLU. Right: Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.

Moblie Deep Learning

- 移动端深度学习框架
- 简单方便部署
- 分别基本不同网络实现

业界深度学习移动端应用情况

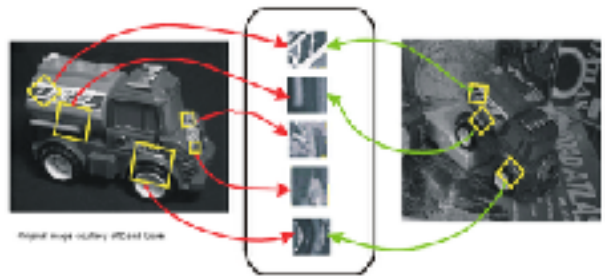
- 微软识花
- 形色
- 淘宝扫立拍
- 百度图搜:
  - 模型压缩
  - 运行速度
  - 准确率90%

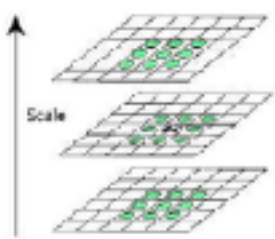
SIFT & CNN

- 拍照搜索技术思想
  - 相同图 SIFT
  - 相似图 CNN

SIFT特征

- SIFT特征点的提取
  - 纹理：特征点与周围点的梯度变化
  - 位置：局部映射，应对位置变化
  - 尺度：尺度扩展，应对大小变化
  - 方向：考虑方向，应对旋转变化
- SIFT的优缺点
  - 优点：不依赖于数据（视觉外观的抽象）
  - 缺点：柔和(渐变)的图像/边缘平滑
  - 缺点：视觉相似，而不是语义相似

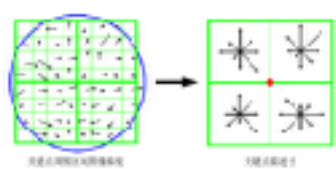




8+2*9=26个区域

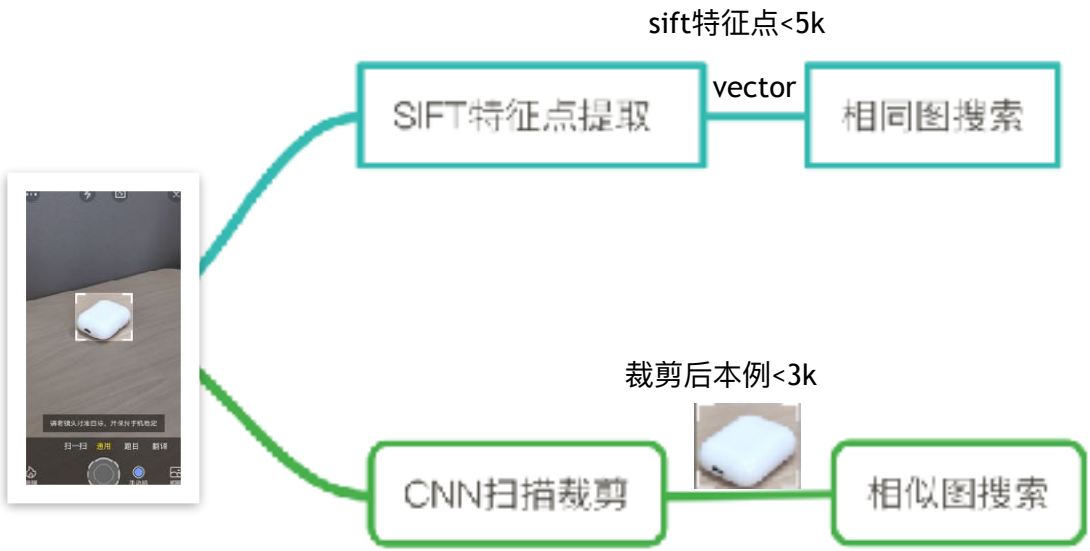
点进行局部极值

检测



生成特征描述符16个2*2，每个2*2区域生成8维的向量方向

扫描式搜索



多模相关技术

- 移动端
- Deep Learning
- Augmented Reality
- 计算机视觉
- Kotlin

多模交互搜索招人

liyonghui@baidu.com

18612188389

THANKS!