学习报告

06117113吴中行

**本周任务**

熟悉并掌握神经网络量化库QKeras

目录

[一、 背景 1](#_Toc35462366)

[二、 量化概念 2](#_Toc35462367)

[三、 量化原理 4](#_Toc35462368)

[四、 量化方法 5](#_Toc35462369)

[五、 总结 6](#_Toc35462370)

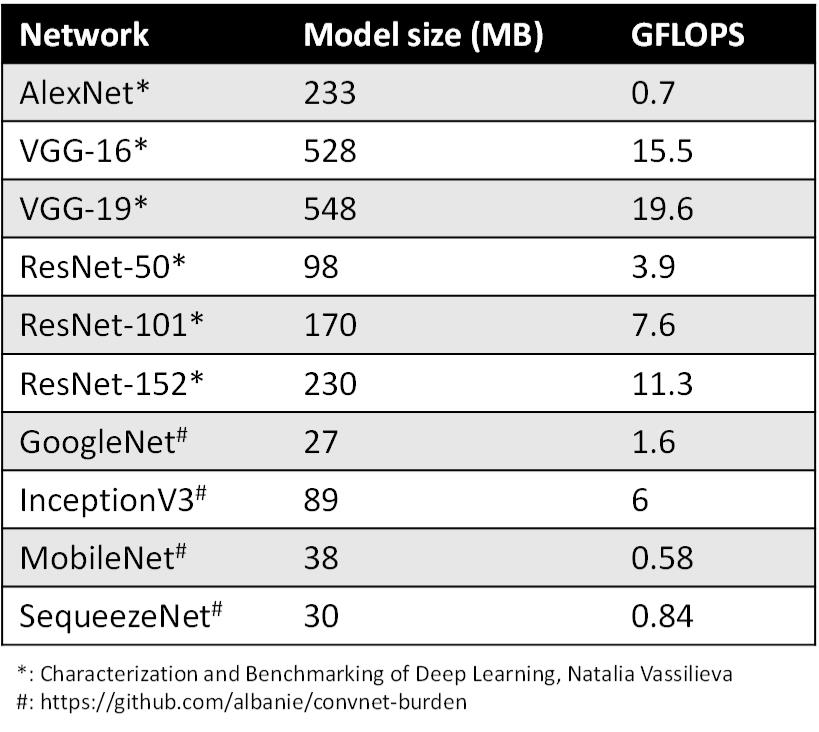
[六、 Qkeras的安装和使用 6](#_Toc35462371)

[1) 安装 6](#_Toc35462372)

[2) 运行mnist例程 7](#_Toc35462373)

## 背景

随着模型预测越来越准确，网络越来越深，神经网络消耗的内存大小成为问题，尤其是在移动设备上。通常情况下，目前（2019年初）的手机一般配备 4GB 内存来支持多个应用程序的同时运行。而三个模型运行一次通常就要占用1GB内存。除了内存问题，为了实时处理数据，还会有内存带宽的问题。



一般模型的内部的计算都采用了浮点数计算，浮点数的计算会消耗比较大的计算资源（空间和cpu/gpu时间）,如果在不影响模型准确率的情况下，模型内部可以采用其他简单数值类型进行计算的话，计算速度会提高很多，消耗的计算资源会大大减小，尤其是对于移动设备来说，这点尤其重要。

**量化就是将神经网络的浮点算法转换为定点。**对神经网络的权值进行量化，使网络模型大小变小，运行速度更快，且准确率与原来相近。这可以在移动手机上实现网络的实时运行，对云计算的部署也有帮助。

## 量化概念

简单来说，就是把网络权值从高精度转化成低精度（**32位浮点数 float32 转化成 8位定点数 int8** 或**二值化为 1 bit**），但模型准确率等指标与原来相近，模型大小变小，运行速度加快。

**FP32 🡪 INT8**

量化可以看作是噪声的一种来源，所以量化后的模型效果与原来相近。

* 优点
  1. 模型变小，运行速度变快。
  2. int8 只需 float32 内存带宽的25％，可以更好使用缓存并避免 RAM 访问出现瓶颈。
  3. 每个时钟周期执行更多的 SIMD 操作。
  4. 如有加速8位计算的 DSP 芯片则更快。
* 缺点

1. 模型准确率稍差。

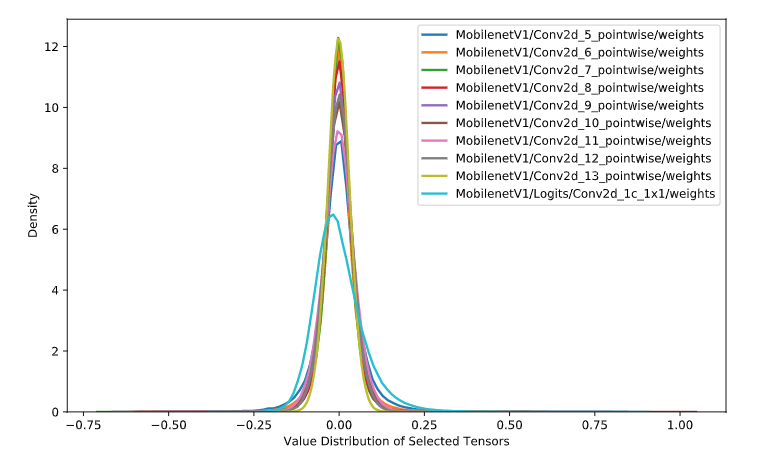
## 量化原理

神经网络由浮点运算构成。

FP32和INT8 的值域是和

而取值数量大约分别为和。因此，将网络从 FP32 转换为 INT8 并不像数据类型转换截断那样简单。

幸运的是，神经网络权重的值分布范围很窄，非常接近零。图八给出了 MobileNetV1 中十层（拥有最多值的层）的权重分布。**同时，采用普通的均匀量化方法，会导致精度下降很多。**



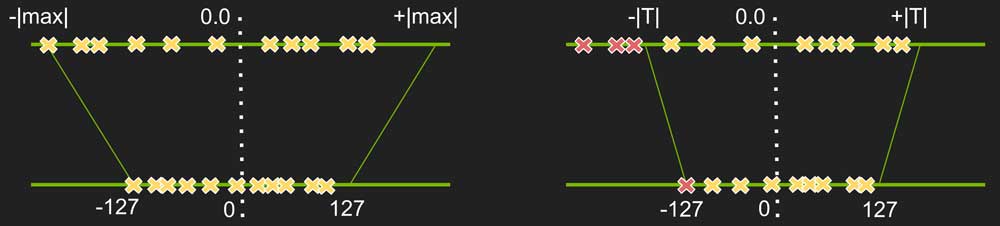
图八：十层 MobileNetV1 的权重分布。

当值落在 (−1,1) 中，量化浮点使用类似的方法将 FP32 映射到 INT8，其中表示 FP32 权重，表示量化的 INT8 权重,是映射因子（缩放因子）。

大多数情况下量化选用无符号整数，那么 INT8 值域为 [0,255] 。𝑧𝑒𝑟𝑜\_𝑝𝑜𝑖𝑛𝑡 在这种情况下更有意义。具体而言，如下面的 等式6-9 所示，量化浮点值可以分为两个步骤：

1. 通过在权重张量（Tensor）中找到 min 和 max 值从而确定和。

下图展示了可以调整 *min*/*max* 来选择一个值域，使得值域的值更准确地量化，而范围外的值映射到定点的 *min*/*max*）。例如，当从原始值范围 [−1,1] 中选定和 ，[−0.9,0.8] 中的值将更准确地映射到 [0,255] 中，而 [−1,−0.9] 和 [0.8,1] 中的值分别映射为 0 和 255。



1. 将权重张量的每个值从 FP32 转换为 INT8 。

注意，当浮点运算结果不等于整数时，需要额外的舍入步骤。例如将 FP32 值域 [−1,1] 映射到 INT8 值域 [0,255]，有，而。

## 量化方法

先训练模型，再进行量化，测试时使用量化后的模型。

• 训练

一般使用 float32 来训练模型效果较好（特别是反向传播和梯度需要浮点来表示）

• 量化

1. 加入量化和去量化操作（如一种量化操作为根据该层权值的最大值和最小值映射到 8位区间）如下图 1 变成图 2

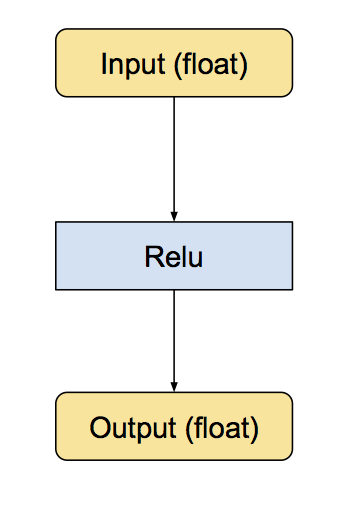
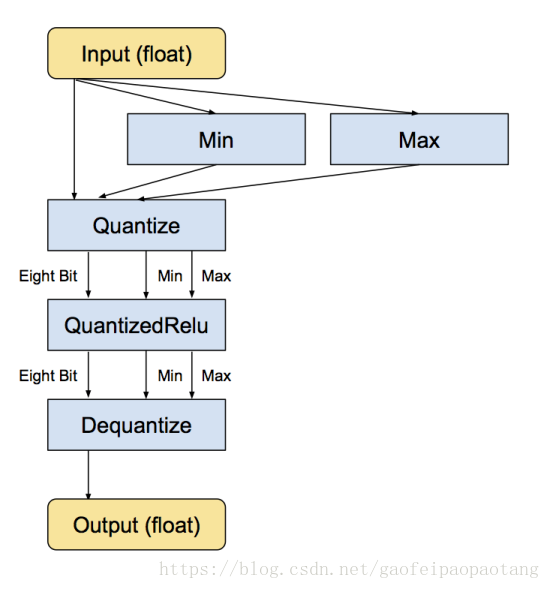


图 1 图2

②把相应的运算转化为量化的运算（实现 8位版本的卷积、矩阵乘法等）

1. 删除相邻的 去量化-量化 操作

如下图

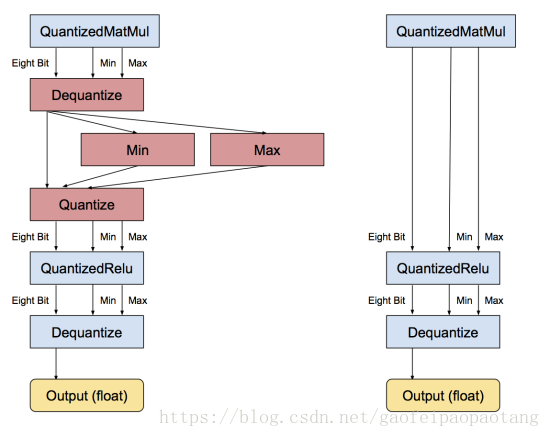


图 3

• 测试

使用量化后的模型来预测

## 总结

**量化理论有效的解释：**神经网络被过度参数化，进而包含足够的冗余信息，裁剪这些冗余信息不会导致明显的准确度下降。

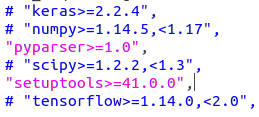
**相关证据：**对于给定的量化方法，FP32 网络和 INT8 网络之间的准确度差距对于大型网络来说较小，因为大型网络过度参数化的程度更高。

## Qkeras的安装和使用

### 安装

因为目前github上Qkeras更新到0.6.0，我安装的时候遇到了部分包缺失的问题，所以安装了Qkeras 0.5.0。

环境要求：



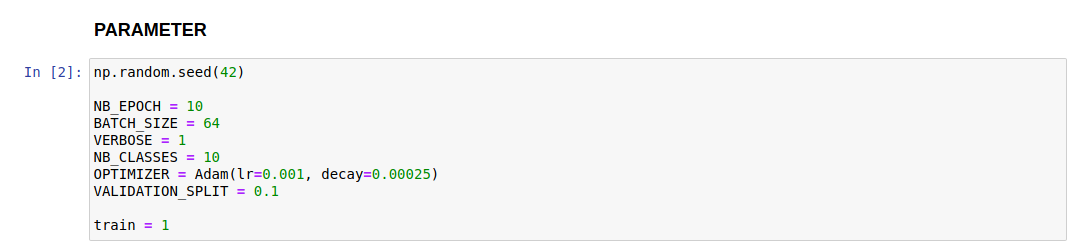
配完环境直接在setup.py目录运行pip install .

### 运行mnist例程

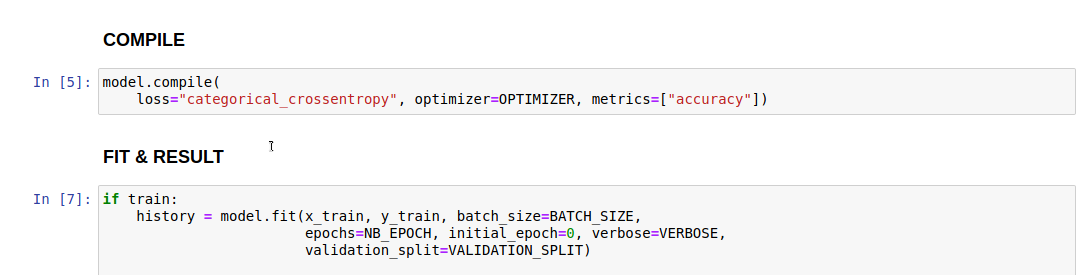
建立模型：



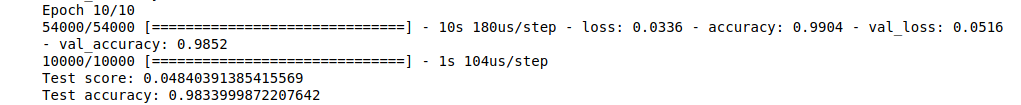
选择参数：



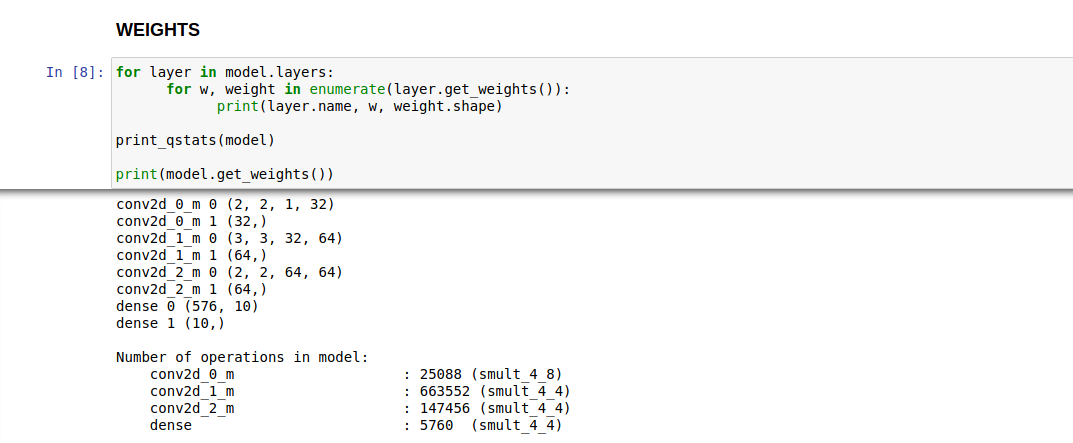
训练：

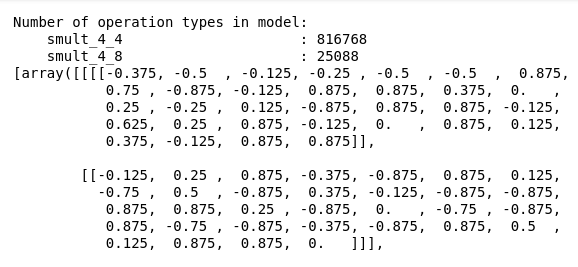


训练结果：



参数（权重）：





可以看到，量化之后，精度还是可以达到0.98