一、训练 yolo3-tiny 网络模型（使用 VOC2007 数据集）进行反量化验证

使用 tflite 量化，训练后 mAP 降低了 0.05%

二、阅读论文 It's All In the Teacher: Zero-Shot Quantization Brought Closer to the Teacher

首先是 zero-shot 量化方法的概述，主要采用生成性方法：

早期的 Zero-shot 量化方法：侧重于如何通过使用权重均衡、偏差校正或范围调整等方案来构建良好的量化函数 Quant()；

CVPR2020 的论文 ZeroQ 使用知识蒸馏的合成样本，应用于全精度模型。

ECCV2020 的论文 GDFQ：采用 GAN 的思想，生成器 G 和量化模型 Q 使用以下损失函数联合训练；

（无论是生成模型还是知识蒸馏产生的合成样本的分布可能不同于原始数据的分布。在这种情况下，它们可以被视为一种对抗性样本，因此，量化网络产生了巨大的泛化差距。）生成式方法通过对抗样本生成方式，能够合成接近真实分布的数据（作为量化的数据基础），并进一步通过优化方式求解量化参数、微调权重参数，以实现有效量化。

zero-shot 量化的损失函数：通常为交叉熵（CE）和全精度网络输出的 KL 散度的组合

论文主要是通过对这两种损失函数进行 loss 平面分析，提出了不包含交叉熵的损失函数 AIT，并使用提出的梯度补偿方法对 KL 散度的梯度进行处理，使量化的学生模型能够类似于全精度的教师模型。

（1）梯度余弦相似性

KD：真实样本　　ZQ：合成样本

使用两个梯度的余弦相似性作为衡量指标，只有当余弦相似度大于零，即它们形成锐角时，才应将这两种损失一起使用。

结论：量化模型的表示能力较弱，通常很难优化多个损失项，并且损失项无法协同工作——即，CE 和 KL 梯度之间的角度相当大。

（2）泛化性能

不能一起使用，考虑从两种损失中选取一种。

（a）评估泛化性能：测量损失平面的局部曲率。Hessian 矩阵 $\mathbf{H}\left(\frac{\partial^2 L}{\partial \theta^2} \in \mathbb{R}^{n \times n}\right.$，θ是 n 个权重参数的向量。**如果优化器稳定在一个尖锐的最小值，那么与平坦的最小值相比，测试时的性能可能会发生更大的退化。（更小的局部曲率提高泛化）**

（b）Hessian 矩阵的轨迹，左：真实数据知识提取（KD），右：Zero-shot 量化（ZQ），右图差异较大。

（c）

loss 平面可视化：从 Hessian 矩阵中取每个 epoch 的最大特征向量 e，通过计算 $L(\theta(t)+k \cdot e \cdot \hat{g}(t))$，$k \in [-0.5, 0.5]$绘制 CE 和 KL 的值，ˆg 是沿 e 的平均梯度。

红色：CE　　蓝色：KL（KL 表面更平坦，尤其在接近训练结束时）

结论：观察到 CE 和 KL 在梯度空间中形成一个大角度，量化模型很难优化两个方向。此外，通过测量 Hessian 矩阵的统计数据，得出结论，KL 有一个更平坦的损失曲面，可以更好地泛

化。

（d）特征值分布：显示了损失项的局部曲率的巨大差异
当 CE 对高特征值有较长的尾部时，KL 对低特征值的尾部更为集中。

结论：KL 的损失平面通常比 CE 平坦得多，具有更好的泛化潜力。

AIT 方法

（1）仅 KL Zero-shot 量化（效果不理想）

（a)测量一个 epoch 内的平均的梯度余弦距离，与前一个 epoch 的余弦距离进行比较。(epoch 间的余弦相似性)。与真实数据蒸馏（KD）的梯度相比，Zero-shot 量化（ZQ）中 KL 的梯度指向一致的方向（余弦相似性大），表明其尚未达到最小值。

（b、c）计算量化前后发生变化的权重参数个数。通过舍入阈值的权重参数的平均数量（量化值与前一步相比发生变化的参数）。
发现：量化值中通过舍入阈值的部分非常小。
（假设这是来自限制整数值更新的量化训练过程。在训练期间，量化网络在内部存储其全部精度值。将参数量化为反向传播的前向传递，并将梯度应用于内部全精度值。当经过几次训练后梯度值变小时，参数的变化通常不足以超过阈值，只有少数层在不断变化，阻止模型向损失面中的较低点移动。）

（2）梯度补偿（GI）

目标：动态操纵每个层 l 的梯度 gl，以确保一定数量的参数在其整数值中更新。

对于随机梯度下降，考虑步骤 k 中参数 $\theta_{l,k}$ 的更新规则，其中学习率为 η：

$$\theta_{l,k+1} = \theta_{l,k} - \eta \cdot g_{l,k}. \quad (7)$$

对于梯度补偿，修改规则如下：参数 $\theta_{l,k}$，量化参数 $\theta_{l,k}^q$，来自层 l 的相应梯度 $g_{l,k}$

$$\theta_{l,k+1} = \theta_{l,k} - \eta \cdot g'_{l,k}, \quad (8)$$
$$g'_{l,k} = \kappa_l \cdot g_{l,k}, \quad (9)$$
$$\kappa_l = \arg\min_{\kappa_l} \|\Delta\theta^q_{l,k} - T\|, \quad (10)$$
$$\Delta\theta^q_{l,k} = \sum \mathbb{I}(\theta^q_{l,k} \neq \theta^q_{l,k+1}), \quad (11)$$
$$T = \rho \cdot dim(\theta_l), \quad (12)$$

ρ ∈ [0,1]：超过量化阈值的预定比例，$\mathbb{I}()$：指示符函数，dim(θ l)： θ l 中元素的个数，

目标：找到能够保证量化层 $\Delta\theta^q_{l,k}$ 上参数更新次数超过一定比率 T 的 κ l

求解：（两步启发式方法）
从 1.0 开始，κ l 加倍，直到 $\Delta\theta^q_{l,k} > T$，为了满足等式 10，通过二进制搜索在 κ /2 和 κ 之间调整 κ l。为了计算效率高，搜索步骤总数限制为五步。为了确保训练的早期稳定性，为 GI 方法添加了一个预热阶段。在预热阶段，κ 的最大值被限制为 128，以获得更精确的解决方案。当生成器需要单独预热时，GI 预热阶段在生成器预热结束后开始。与学习速率指数衰

减调度类似，我们将指数衰减应用于 ρ 。

三、GAN 代码学习

（1）GAN

模块：

生成模型（Generative Model）：输入一行正态分布随机数生成相应的输出，目的：生成让判别模型无法判断真伪的输出。

判别模型（Discriminative Model）：对生成模型的输出进行判别，判断是否为真，目的：判断出真伪。

1、Generator：

输入：一行正态分布随机数（N 维向量）

输出：生成的 MNIST 手写体图片（(28,28,2)的图片）

2、Discriminator：

输入：(28,28,1)的图片

输出：0，1 之间的数（1 为真，0 为假）

3、训练过程：

（1）随机选取 batch_size 个真实的图片；

（2）随机生成 batch_size 个 N 维向量，传入到 Generator 中生成 batch_size 个虚假图片；

（3）真实图片的 label 为 1，虚假图片的 label 为 0，将真实图片和虚假图片当作训练集传入到 Discriminator 中进行训练；

（4）将虚假图片的 Discriminator 预测结果与 1 的对比作为 loss 对 Generator 进行训练（与 1 对比的意思是，如果 Discriminator 将虚假图片判断为 1，说明这个生成的图片很“真实”）。

（2）DCGAN 代码

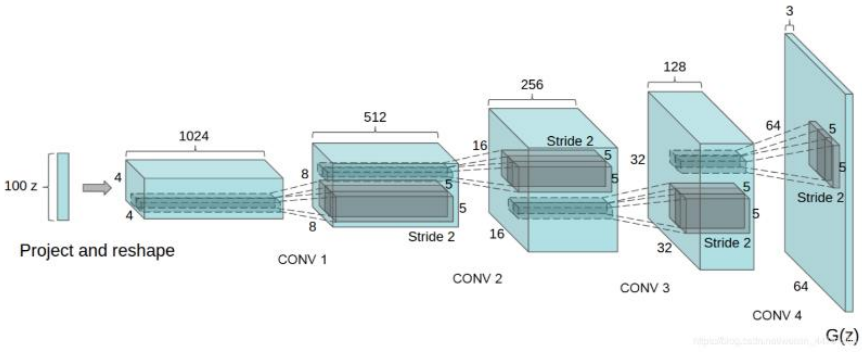
深度卷积对抗生成网络：在 GAN 的基础上增加深度卷积网络

1、Generator

输入：正态分布随机数

输出：生成的图片

网络结构：输入-全连接到 16384（4*4*1024）-reshape 到（4，4，1024）-（8，8，512）-（16，16，256）-（32，32，128）-（64，64，3）- tanh - 生成 image



（使用反卷积，**在正常卷积下，我们的特征层的高宽会不断被压缩；在反卷积下，我们的特征层的高宽会不断变大。**）

2、Discriminator：
输入：图片
输出：判断结果（0，1）的数，1 为真，0 为假

3、训练：
先训练判别器，再训练生成器
（1）训练判别器
输入：真图片+标签、假图片+标签

1、随机选取 batch_size 个真实的图片。
2、随机生成 batch_size 个 N 维向量，传入到 Generator 中生成 batch_size 个虚假图片。
3、真实图片的 label 为 1，虚假图片的 label 为 0，将真实图片和虚假图片当作训练集传入到 Discriminator 中进行训练。

（2）训练生成器
1、随机生成 batch_size 个 N 维向量，传入到 Generator 中生成 batch_size 个虚假图片。
2、将虚假图片的 Discriminator 预测结果与 1 的对比作为 loss 对 Generator 进行训练（与 1 对比的意思是，让生成器根据判别器判别的结果进行训练）。

（3）CGAN 代码
给生成结果贴上标签
（CGAN 一种带条件约束的 GAN，在生成模型（D）和判别模型（G）的建模中均引入条件变量 y（conditional variable y）。使用额外信息 y 对模型增加条件，可以指导数据生成过程。这些条件变量 y 可以基于多种信息，例如类别标签，用于图像修复的部分数据，来自不同模态（modality）的数据。）

1、Generator
生成一个 N 维的正态分布随机数，再利用 Embedding 层将正整数（索引值）转换为 N 维的稠密向量，并将这个稠密向量与 N 维的正态分布随机数相乘，从而获得一个有标签的随机

数。

2、Discriminator
输入：（28，28，1）的图片
输出：
（1）一个 0 到 1 之间的数，1 代表判断这个图片是真的，0 代表判断这个图片是假的。
（2）一个向量，用于判断这张图片属于什么类。

3、训练

1、随机选取 batch_size 个真实的图片和它的标签。

2、随机生成 batch_size 个 N 维向量和其对应的标签 label，利用 Embedding 层进行组合，传入到 Generator 中生成 batch_size 个虚假图片。

3、Discriminator 的 loss 函数由两部分组成，一部分是真伪的判断结果与真实情况的对比，一部分是图片所属标签的判断结果。

4、Generator 的 loss 函数也由两部分组成，一部分是生成的图片是否被 Discriminator 判断为 1，另一部分是生成的图片是否被分成了正确的类。

（4）ACGAN 代码
给生成结果贴上标签、使用卷积神经网络=DCGAN + CGAN

（5）LSGAN 代码
提高生成图片的质量：最小二乘 GAN：将生成模型和判别模型的 loss 函数由交叉熵更改为均方差 mse

（6）CoGAN 代码
耦合生成式对抗网络
通过同一个输入，生成不同内容

思路：

1、建立两个生成模型，两个判别模型。

2、两个生成模型的特征提取部分有一定的重合，在最后生成图片的部分分开，以生成不同类型的图片。

3、两个判别模型的特征提取部分有一定的重合，在最后判别真伪的部分分开，以判别不同类型的图片。

1、生成器
一共存在两个生成模型，两个生成模型的特征提取部分有一定的重合，在最后生成图片的部分分开，以生成不同类型的图片。
即：权值部分有一定的共享。

2、判别器
一共存在两个判别模型，两个生成模型的特征提取部分有一定的重合，在最后判别真伪的部分分开，以判别不同类型的图片。