深度学习：homework4

张驰　　前沿交叉学科研究院

2019 年 5 月 14 日

1　知识蒸馏背景

在大规模的机器学习应用中，我们可以训练一个高精度的大模型 (教师模型)，并将其“知识”转移到一个更小的适合部署的模型 (学生模型)。这个过程被称为蒸馏。一个明显的蒸馏方法是用大模型产生的类概率训练小模型。

- 本次实验中，我们将使用第一次作业的 baseline 模型作为教师模型。
- 我们将蒸馏一个量化神经网络，该网络根据 baseline 的模型，使用离散的值来表示权重和特征
- 本实验中，我们将设置量化权重为 2bit，量化特征为 1bit。

2　利用 gt 训练量化模型

2.1　实验准备

实验一中，我们先查看不利用教师模型，只利用量化神经网络训练的 baseline 结果如何。对应的，要将损失函数中的教师模型的预测值 preds 修改为 ground truth。

```
loss = tf.losses.softmax_cross_entropy(label_onehot, logits/args.temperature) + loss_reg
```

2.2　实验分析

实验运行结果，图 1 是未使用教师模型信息的量化模型的测试结果:

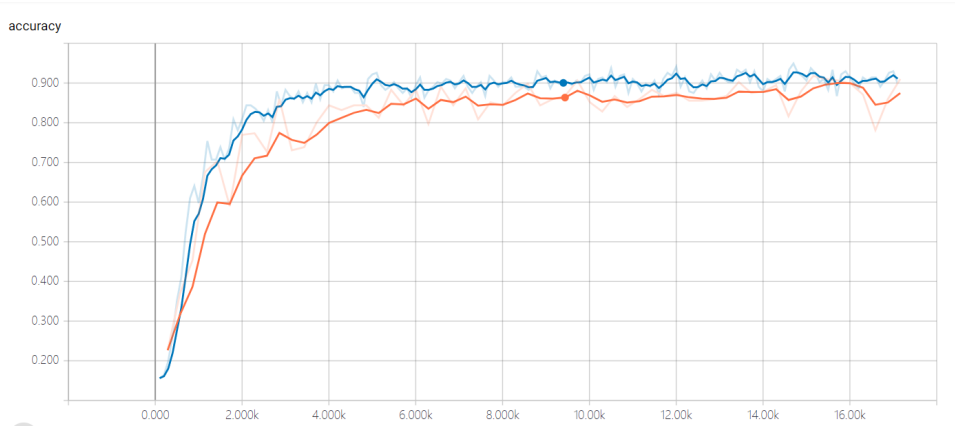


图 1: 未使用教师模型信息的量化模型的测试结果

通过与教师模型对比，如表 1 所示:

表 1: 量化模型与 baseline 对比差异

Name	Checkpoint	test-accuracy
baseline	3.43M	94.3%
q1	1.13M	90.0%

分析结果，可能是量化模型能够缩小参数搜索空间，将值量化到指定比特位数的地址空间中，因此降低了存储内存，但是对应的丧失了精度。这里根据论文再尝试理解量化操作的过程。由图 2，图 3 中的表达式以及对照给定的 baseline 代码，可以得出如图 4 所示的量化过程图分析。

$$\textbf{Forward: } r_o = \frac{1}{2^k - 1}\,\mathrm{round}((2^k - 1)r_i)$$

$$\textbf{Backward: } \frac{\partial c}{\partial r_i} = \frac{\partial c}{\partial r_o}.$$

图 2: 量化取整公式

$$\textbf{Forward: } r_o = f_\omega^k(r_i) = 2\,\mathrm{quantize_k}\left(\frac{\tanh(r_i)}{2\max(|\tanh(r_i)|)} + \frac{1}{2}\right) - 1.$$

图 3: 量化过程计算公式

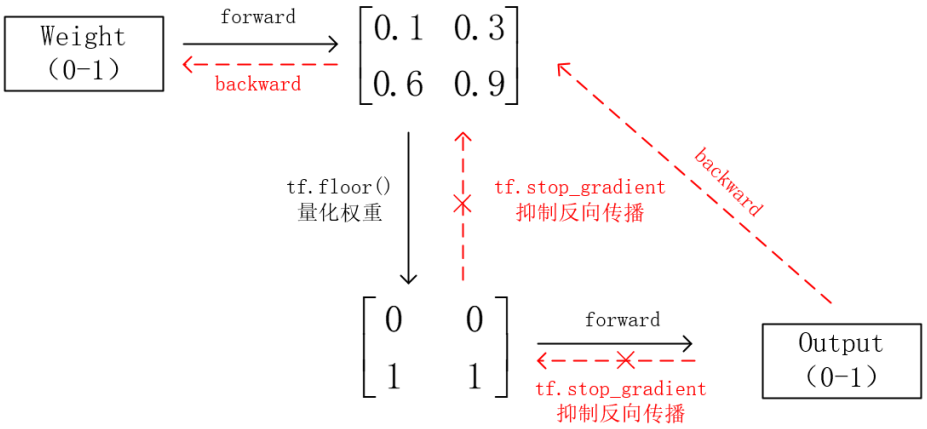


图 4: 权重量化过程

结合 baseline 代码分析图 4 如下：

- 1. 将输入的权重按照图 3 所示的公式进行处理，经过 tanh 函数将输入值限制在 [-1,1] 之间。
- 2. 再利用 tf.clip_by_value 方法将输入值限制在 [0,1] 之间。
- 3. tf.floor() 方法将输入权重量化到 2bit，4 个值空间中，再将量化后的值压缩到 [0,1] 之间。
- 4. 最后再乘以系数 2，将量化结果重新缩放回 [-1,1] 之间。
- 5. 这里利用 tf.stop_gradient 方法抑制了对权重量化结果的反向传播过程。由图 2 可知，对原始权重的求导与对量化的结果求导是等价的，因此这里我们利用了对原始权重的反向传播来更新量化的结果。但注意，我们只是对原始权重的反向传播过程后，再进行量化过程，本质上量化过程是不需要反向传播的，因此我们必须要设置对量化结果进行 tf.stop_gradient 抑制。
- 6. 经过量化后，量化权重值与输入权重值大小没有区别，只是值的范围被量化到 4 个值之间。

3 利用教师模型的分类概率训练量化模型

3.1 实验准备

使用温度 T>1 的 softmax(如公式 1 所示) 以及教师模型生成的目标分类概率来构建'soft' 交叉熵损失函数。温度 T 值越大，分类的概率分布越'soft'，教师模型的目标概率也使用相同的温度 T 的 softmax 方法生成。

$$q_i = \frac{exp\,(z_j/T)}{\sum_j exp\,(z_j/T)} \tag{1}$$

z_i, T, q_i 分别表示第 i 个类的 logit，温度，和第 i 个类的概率。对应的修改代码为：

```
loss = tf.losses.softmax_cross_entropy(target_label_onehot, logits/args.temperature) + loss_reg
```

这里存在 2 个问题需要解释，

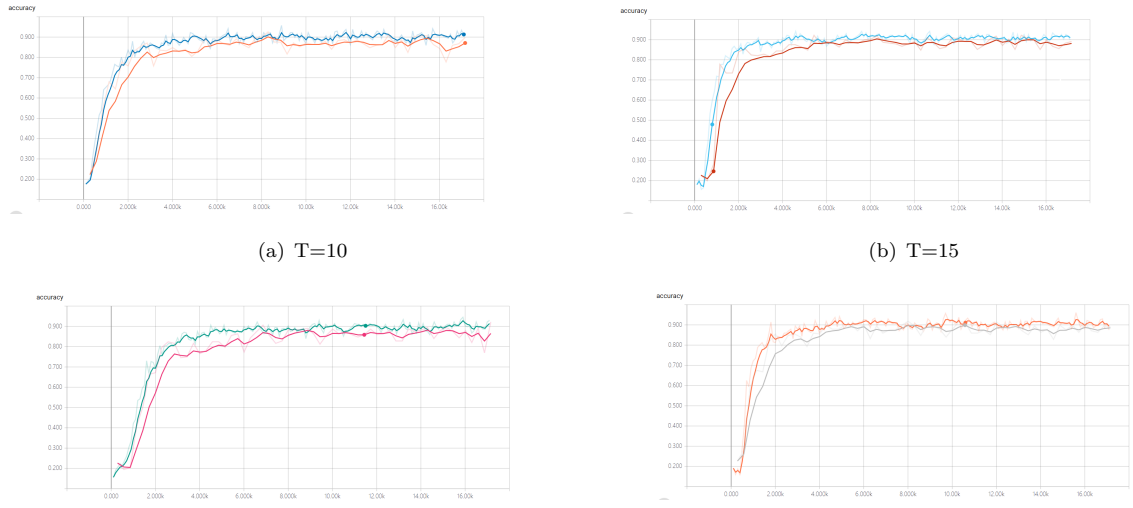
- 1. 明明 true label（hard target）是完全正确的，为什么还要教师模型的 soft target 呢？
- 2. 为什么需要增加这个温度 T？

查阅资料可知 hard target 包含的信息量（信息熵）很低，soft target 包含的信息量大，拥有不同类之间关系的信息（比如同时分类驴和马的时候，尽管某张图片是马，但是 soft target 就不会像 hard target 那样只有马的 index 处的值为 1，其余为 0，而是在驴的部分也会有概率。这样的好处是，这个图像可能更像驴，而不会去像汽车或者狗之类的，而这样的 soft 信息存在于概率中，以及 label 之间的高低相似性都存在于 soft target 中。

但是如果 soft targe 是像这样的信息 [0.98 0.01 0.01]，就意义不大了，所以需要在 softmax 中增加温度参数 T，加入温度参数 T 后，概率之间的差异变小了，例如 [0.98,0.01,0.0] 把 soft target 软化（整体除以一个数值后再 softmax），就会变成 [0.8,0.1,0.1]，这样就保证充分利用信息。Temperature 数值越大，分布越缓和；而 Temperature 数值减小，容易放大错误分类的概率，引入不必要的噪声。因此保证温度 T>1。

3.2 实验分析

将温度分别设置为 T=10,15,50,100 时，利用 Tensorboard 查看对应的训练和测试精度如图 5 所示，对应的实验结果如表 2 所示：



(a) T=10　(b) T=15

(c) T=50　(d) T=100

图 5: 不同温度下利用教师模型分类概率训练得到的测试结果

表 2: 不同温度的 soft 损失最优结果

Temperture	test−accuracy
T=10	90.0%
T=15	90.4%
T=50	88.1%
T=100	89.9%

根据实验结果，大致可以判断出，加入不同的温度值后，模型的测试精度随着温度 T 的上升呈趋势性先增大，后减少，再增大的过程。以本实验中选择的具体值为例，在 T=15 的时候效果最好，之后到 T=50 时下降到一个较低的值，当 T=100 时又回到 baseline 水平，可以得出温度的选择对测试结果有一定的影响。

4 在 soft 交叉熵基础上加入 hard 交叉熵

4.1 实验准备

在上一个问题的实验基础上增加 hard 交叉熵损失函数，则损失函数如公式 2 所示:

$$Loss = \text{softmax-cross-entropy}(z_i, gt) + c * \text{softmax-cross-entropy}(z_i/T, preds_i) \tag{2}$$

该过程的流程可简化为图 6 所示的流程：

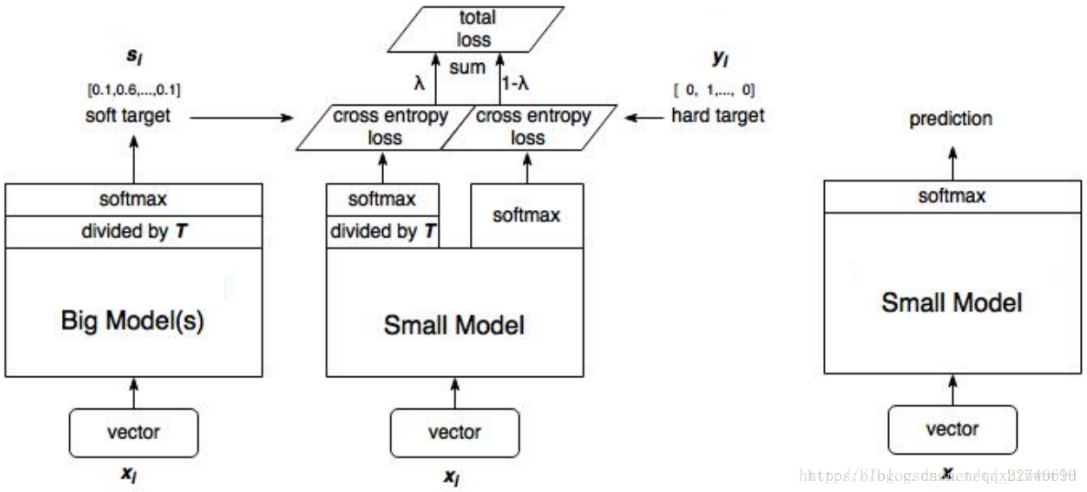


图 6: 蒸馏模型示意图

根据查阅资料显示，total loss 设计为 soft 与 hard 所对应的交叉熵的加权平均，其中 soft 交叉熵的加权系数越大，表明迁移诱导越依赖教师网络的贡献，这对训练初期阶段是很有必要的，有助于让

学生网络更轻松的鉴别简单样本，但训练后期需要适当减小软目标的比重，让真实标注帮助鉴别困难样本。

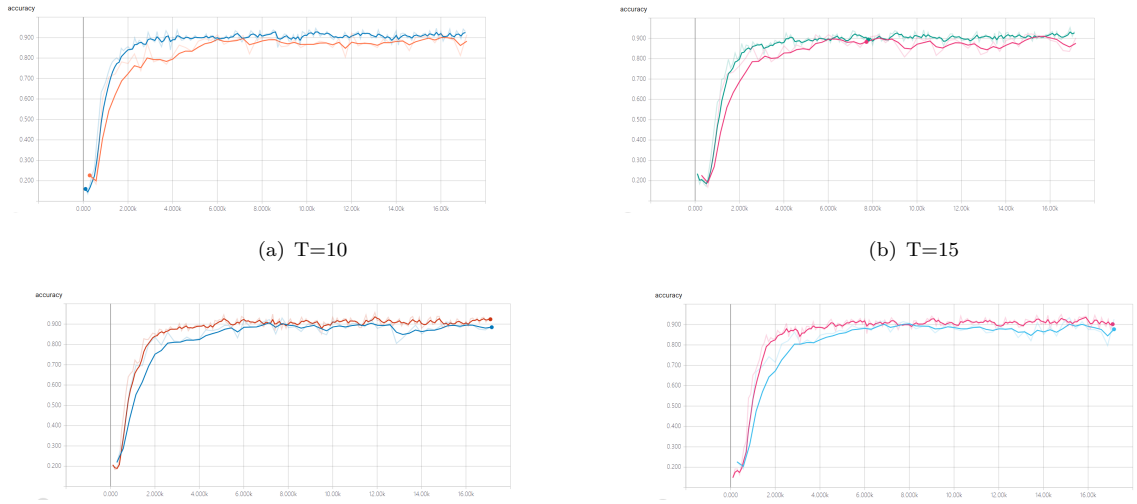
但在本实验中，soft 和 hard 的 loss 值都在一个量级，但是梯度却不一样，注意这里设置 $c = T^2$ 保证了 soft 和 hard 的反向传播影响相等 (根据反向传播公式推导可得)，同时又改变了 soft 和 hard 的损失函数权重占比，但是这里更关注的是梯度的比例，对权重的占比没有太多要求。

因此对应的修改代码的损失函数如下：

```
loss = tf.losses.softmax_cross_entropy(target_label_onehot, logits/args.temperature) +
    loss_reg
```

4.2 实验分析

同实验 2 参数设置相同，将温度分别设置为 T=10,15,50,100 时，利用 Tensorboard 查看对应的训练和测试精度如图 6 所示，对应的实验结果如表 3 所示：



(a) T=10　　(b) T=15

(c) T=50　　(d) T=100

图 7: 不同温度下 soft 加上 hard 交叉熵方法得到的测试结果

表 3: 不同温度的 soft 加上 hard 交叉熵方法最优结果

Temperture	test−accuracy
T=10	90.0%
T=15	91.0%
T=50	90.1%
T=100	90.3%

从实验结果可以看出,soft 加上 hard 交叉熵方法使测试结果相对于单独使用 soft 方法普遍有细微提

高，而且测试精度随温度变化的趋势与实验 2 中的趋势相同，在 T=15 时效果最佳，相较于 baseline 方法有 1% 的提高。

5 利用 gt 构造 soft 和 hard 交叉熵

5.1 实验准备

实验 2 中探索利用教师模型的预测分类概率训练 soft 交叉熵，实验 3 中在实验 2 的基础上加入以 gt 为目标训练的 hard 交叉熵，实验 4 则是探索直接利用 gt 作为 soft 和 hard 交叉熵的训练目标，方法相同，其具体表达式如图 3 所示：

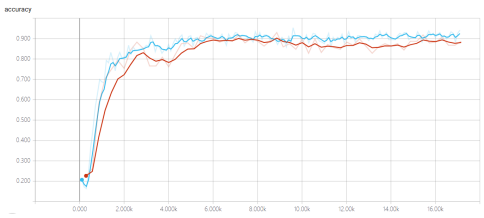
$$Loss = \text{softmax-cross-entropy}(z_i, gt) + c * \text{softmax-cross-entropy}(z_i/T, gt) \tag{3}$$

对应的修改的代码部分如下所示：

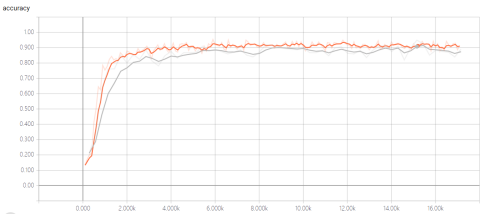
```
loss = args.temperature ** 2 * tf.losses.softmax_cross_entropy(label_onehot, logits / args.
                                temperature)+ tf.losses.softmax_cross_entropy(
                                label_onehot, logits) + loss_reg
```

5.2 实验分析

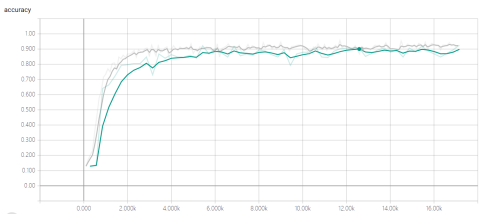
同实验 2 参数设置相同，将温度分别设置为 T=10,15,50,100 时，利用 Tensorboard 查看对应的训练和测试精度如图 7 所示，对应的实验结果如表 4 所示：



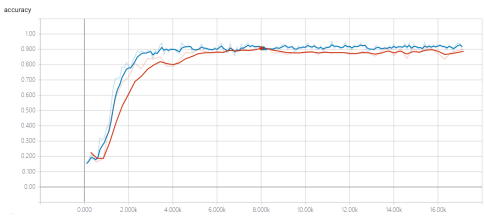
(a) T=10



(b) T=15



(c) T=50



(d) T=100

图 8: 不同温度下以 gt 为目标的 soft 加上 hard 交叉熵方法训练得到的测试结果

表 4: 不同温度以 gt 为目标的 soft 加上 hard 交叉熵方法最优结果

Temperture	test−accuracy
T=10	90.2%
T=15	91.4%
T=50	90.0%
T=100	90.6%

从实验结果可以得出，直接使用 gt 为目标来训练 soft 和 hard 交叉熵损失函数，测试结果又有细微的提高，且随温度的变化趋势依然与实验 2，实验 3 中相同，在 T=15 时效果最佳。

综上 4 个实验可以得出以下两个结论：

1. 量化方法相较于 baseline 减少了模型的参数存储内存，加快的训练速度，但是相应的损失了精度，是用精度换空间和速度的方法。

2. 结合知识蒸馏的方案，我们尝试了不同温度 T 下利用教师模型的预测分类概率的 soft 交叉) 训练量化模型，利用教师模型预测分类概率 soft 交叉熵和 ground truth 的 hard 交叉熵训练量化模型，最后利用 ground truth 为目标训练的 soft 交叉熵和 hard 交叉熵来训练量化模型，得出了在温度 T=10,15,50,100 中，T=15 时 soft 交叉熵对量化模型有明显的提高效果，即可以通过知识蒸馏方法提高量化模型的测试精度。

参考文献

[1] Hinton G , Vinyals O , Dean J . Distilling the Knowledge in a Neural Network[J]. Computer Science, 2015, 14(7):38-39.

[2] Zhou S , Wu Y , Ni Z , et al. DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients[J]. 2016.

[3] https://www.zhihu.com/question/50519680/answer/136406661

[4] https://blog.csdn.net/xbinworld/article/details/83063726