模型压缩

知识蒸馏 教师-学生模型 模型压缩

模型压缩方法:

- 参数修剪与共享: 去除冗余和不重要的项。
- 低秩分解: 使用矩阵分解估计深度学习模型的信息参数。
- 迁移/压缩卷积滤波器: 设计特殊的结构卷积滤波器降低存储和计算复杂度。
- 知识蒸馏: 训练更紧凑的网络重现一个更大网络的输出。

知识蒸馏

知识蒸馏是一种常见的模型压缩方法，在teacher-student框架中，将复杂、学习能力强的网络学到的特征表示知识蒸馏出来，传递给参数量小、学习能力弱的网络。蒸馏可以提供student在one-shot label上学不到的soft label信息，及student小网络学不到而teacher网络可以学到的特征表示知识，所以一般可以提高student网络的精度。

知识蒸馏带来的预期收益

- 模型优化，通过知识蒸馏带来模型效果提升
- 模型压缩

Hinton的知识蒸馏论文

在SoftMax中加入T参数的作用: 网上举的比较形象的例子，每次负重进行登山，虽然过程很辛苦，但是当有一天取下负重，正常登山的时候就会变得轻松。因此在知识蒸馏中在SoftMax中加入T参数也是类似的目的，通过加入T参数使得SoftMax的输出变得比较平缓，因此训练的难度变大。

交叉熵导数推导(推导过程以真实label进行推导):
假设神经网络对每个类的输出值为z_i，其对应的softmax输出值为p_i，真实的label值为y_i，则

$$p_i = \frac{e^{z_i}}{\sum_k e^{z_k}}$$

交叉熵计算公式为:

$$C = -\sum_i y_i \ln p_i$$

则计算交叉熵在z_i上的偏导数，根据链式计算法则，计算公式如下:

$$\frac{\partial C}{\partial z_i} = \sum_j \left(\frac{\partial C_j}{\partial p_j} \frac{\partial p_j}{\partial z_i}\right)$$

我们分步骤进行求偏导，先求$\frac{\partial C}{\partial p_j}$，根据乘法求导法则可得到

$$\frac{\partial C_j}{\partial p_j} = \frac{\partial(-y_j ln p_j)}{\partial p_j} = -1 * (0 * ln p_j + y_j * \frac{1}{p_j}) = -\frac{y_j}{p_j}$$

之后计算$\frac{\partial p_j}{\partial z_i}$，对该部分的求导计算要分为两个部分，即$i = j$和$i \neq j$两种情况进行求导，具体的计算过程如下所示:

- 当$i = j$时，具体的计算过程为

$$\frac{\partial p_i}{\partial z_i} = \frac{(\partial \frac{e^{z_i}}{\sum_k e^{z_k}})}{\partial z_i} = \frac{\sum_k e^{z_i} e^{z_k} - (e^{z_i})^2}{(\sum_k e^{z_k})^2} = \frac{e^{z_i}}{\sum_k e^{z_k}} (1 - \frac{z_i}{\sum_k e^{z_k}}) = p_i(1 - p_i)$$

- 当$i \neq j$时，具体的计算过程为

$$\frac{\partial p_j}{\partial z_i} = \frac{\partial \frac{e^{z_j}}{\sum_k e^{z_k}}}{\partial z_i} = \frac{-e^{z_j} e^{z_i}}{(\sum_k e^{z_k})^2} = -\frac{e^{z_j}}{\sum_k e^{z_k}} \frac{e^{z_i}}{\sum_k e^{z_k}} = -p_j p_i$$

  最终的导数推导为

$$\frac{\partial C}{\partial z_i} = \sum_{i \neq j} (-\frac{y_j}{pj} * (-p_j p_i)) + p_i(1 - pi) * -\frac{y_i}{p_i} = \sum_{i \neq j} (y_j p_i) + y_i * (p_i - 1) = \sum_j y_j p_i - y_i = p_i - y_i$$

对Hinton论文2.1节中的推导解释，在2.1节中Hinton针对一种特殊情况进行了推导

$$\frac{\partial C}{\partial z_i} = \frac{1}{T}(q_i - p_i) = \frac{1}{t}(\frac{e^{\frac{z_i}{T}}}{\sum_j e^{\frac{z_j}{T}}} - \frac{e^{\frac{v_i}{T}}}{\sum_j e^{\frac{v_j}{T}}})$$

在论文中提到的特殊情况为，T值相比logits数量级要高，则可以转换为

$$\frac{\partial C}{\partial z_i} \simeq \frac{1}{T}(\frac{1 + \frac{z_i}{T}}{N + \sum_j \frac{z_j}{T}} - \frac{1 + \frac{v_i}{T}}{N + \sum_j \frac{v_j}{T}})$$

这里推导中比较奇怪的一步就是为什么T值相比logits数量级大就可以近似转换为下面这个公式，可以通过下式可以得出

$$\lim \frac{z_i}{T} \to \infty e^{\frac{e_i}{T}} = 1 + \frac{z_i}{T}$$

之后文中假设所有Logits之和为0，从而进一步简化为

$$\frac{\partial C}{\partial z_i} \simeq \frac{1}{NT^2}(z_i - v_i)$$

这一步转化并不困难，只要把$\sum_j z_j = \sum_j v_j = 0$带入公式就可以求出了。

参考文章: https://blog.csdn.net/qian99/article/details/78046329

- Soft Target(软目标)
  Soft Target是指加入T参数的目标，要尽量接近于大网络加入T后的分布概率。
- Hard target(硬目标)
  Hard目标是正常网络训练的目标，要尽量接近训练的Label。
- 两个目标函数
  学生网络会有两个Loss，分别对应硬目标和软目标计算出的交叉熵，两者加和作为学生网络的Loss。

- 具体蒸馏训练
  教师网络的Loss: 对SoftMax(T)的输出与原始Label求Loss
  学生网络的Loss: 1) 对SoftMax(T)的输出与教师的Softmax(T)输出求Loss1 2) 对SoftMax(T)的输出与原始的Label求Loss2 3) Loss= Loss1+Loss2

模型蒸馏

论文合集

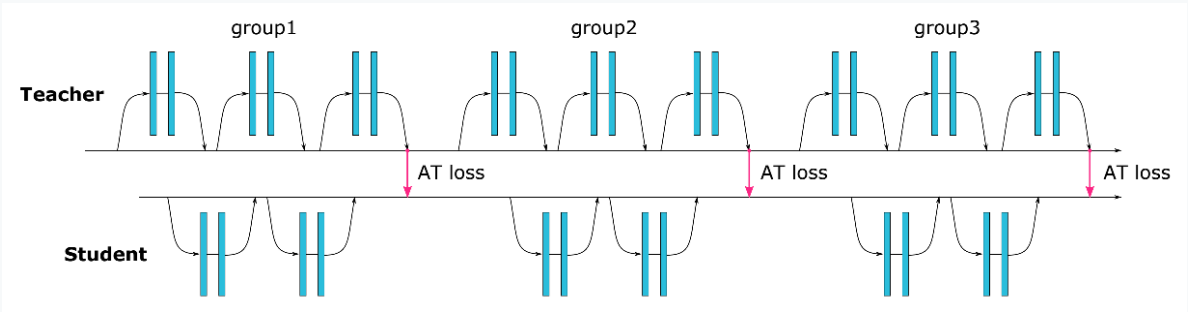
- Hinton论文: Distilling the Knowledge In a Neural Network

Distilling the Knowledge in a Neural Network.pdf
104.1 KB

- Attention Transfer: Paying More Attention To Attention: Imporving the Performance of Convolutional Neural Networks via Attention Transfer

PAYING MORE ATTENTION TO ATTENTION- IMPROVING THE PERFORMANCE OF CONVOLUTIONAL NEURAL NETWORKS VIA ATTENTION TRANSFER.pdf
1.2 MB

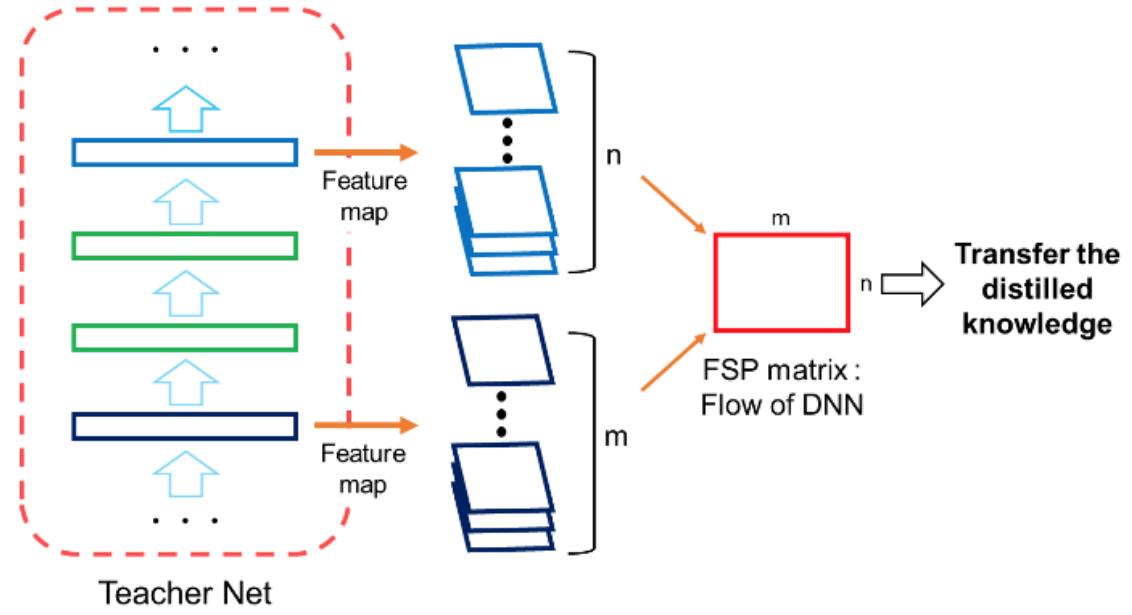
提出利用教师-学生模型，不仅仅通过教师的输出作为Soft Target，同时让学生学习教师中间层的输出。



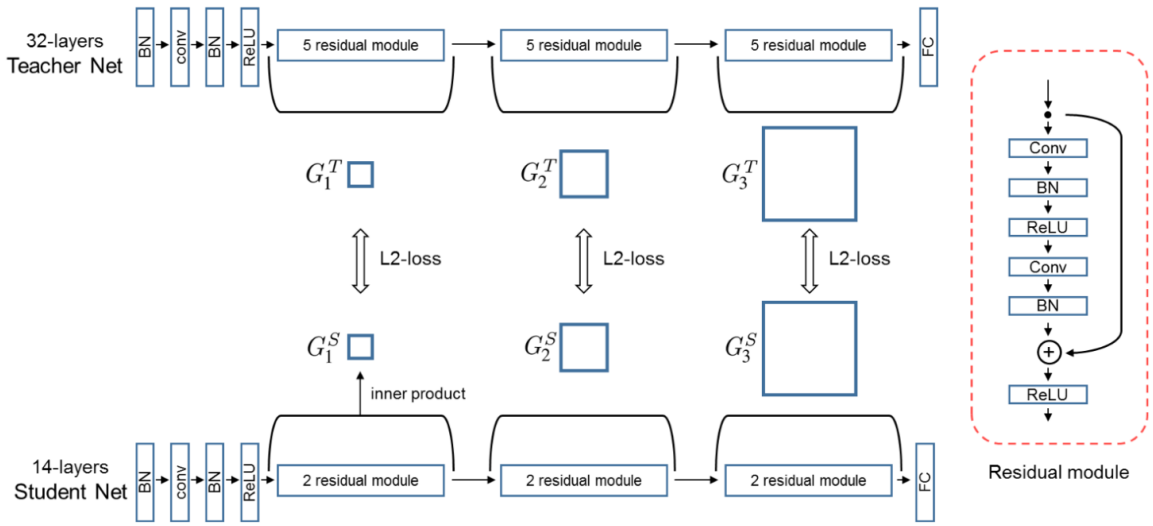
- A Gift from Knowledge Distillation:
  Fast Optimization, Network Minimization and Transfer Learning

A Gift from Knowledge Distillation- Fast Optimization, Network Minimization and Transfer Learning.pdf
551.5 KB

韩国人发的文章，相比PAY MORE ATTENTION…更进一步，不再学习教师中间层的输出，而是让学生学习教师中间层输出层与层之间的关系。

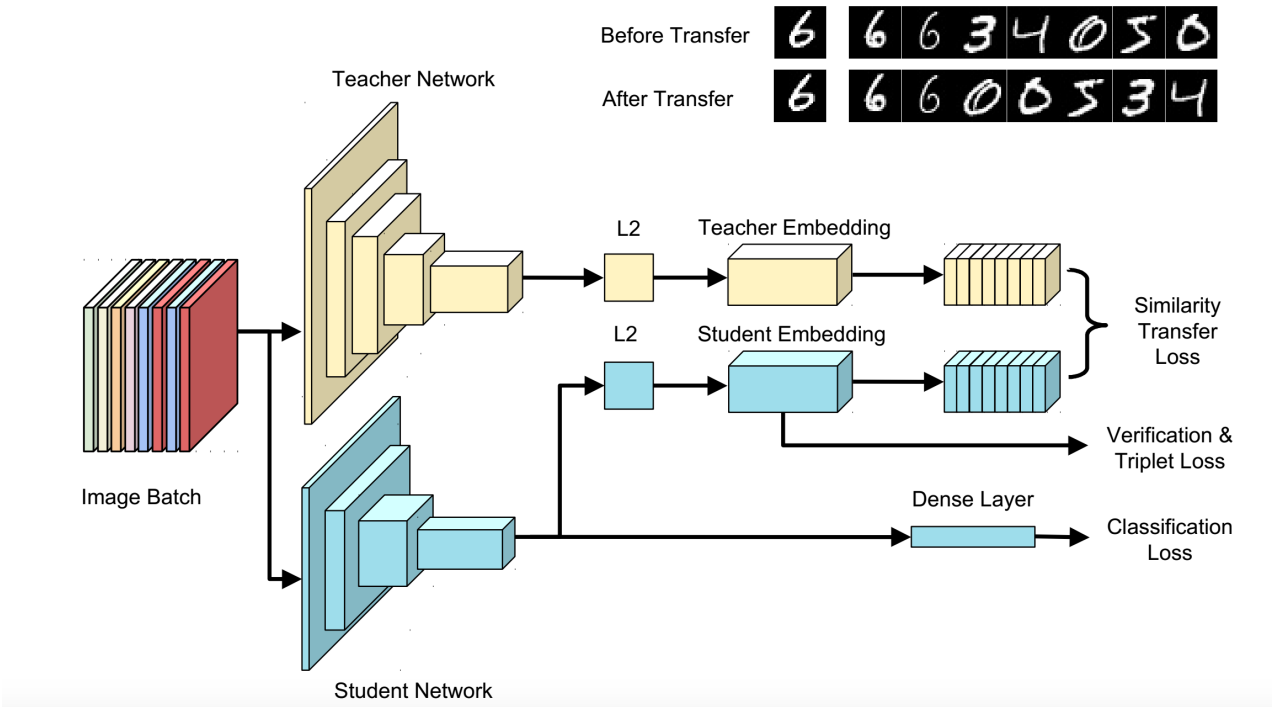


$$L_{FSP}(W_t, W_s) = \frac{1}{N}\sum_{x}\sum_{i=1}^{n}\lambda_i \times \|(G_i^T(x; W_t) - G_i^S(x; W_s)\|_2^2,$$



- DarkRank: Accelerating Deep Metric Learning via Cross Sample Similarities Transfer

DarkRank- Accelerating Deep Metric Learning via Cross Sample Similarities Transfer.pdf
464.2 KB



- Like What You Like: Knowledge Distill via Neuron Selectivity Transfer

Like What You Like- Knowledge Distill via Neuron Selectivity Transfer.pdf
2.3 MB

- Training Shallow and Thin Networks for Acceleration via Knowledge Distillation with Conditional Adversarial Networks

Training Shallow and Thin Networks for Acceleration via Knowledge Distillation with Conditional Adversarial Networks.pdf
575.6 KB

- Deep Mutual Learning

Deep Mutual Learning.pdf
389 KB

- Born Again Neural Networks

Born-Again Neural Networks.pdf
546.9 KB

再生网络

再生网络中对Dark Knowledege的解释

在文章中首先指出在Hinton的论文中指出是隐藏在错误响应中的逻辑分布的暗知识带来知识蒸馏的收益。另外一种解释是通过比较在蒸馏和通常的监督学习中输出节点在正确类别上相应的梯度滑动，知识蒸馏类似于重要性，权重与教师对正确预测的权重相对应。

学生逻辑值z_j和教师逻辑值t_j在单样本交叉熵的梯度为

$$\frac{\partial L_i}{\partial z_i} = q_i - p_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} - \frac{e^{t_i}}{\sum_{j=1}^{n} e^{t_j}}$$

在一个Batch上学生的反向传播的梯度均值可以表示为

$$\sum_{s=1}^{b}\sum_{i=1}^{n}\frac{\partial L_{i,s}}{\partial z_{i,s}} = \sum_{s=1}^{b}\sum_{i=1}^{n}(q_{i,s} - q_{i,s}) = \sum_{s=1}^{b}(q_{*,s} - p_{*,s}) + \sum_{s=1}^{b}\sum_{i=1}^{n-1}(q_{i,s} - p_{i,s})$$

上面式子中第二部分表示来自所有错误输出的信息。第一部分表示来自正确选择的梯度。第一部分可以进一步写作

$$\frac{1}{b}\sum_{s=1}^{b}(q_{*,s} - p_{*,s}y_{*,s})$$

因为$y_{*,s}$是训练样本中所标注的Label值，所以在该式中该值都是1，因此可以改写做该式。通过该式可以将$p_{*,s}$看做是为真实Label加了一个权重，该权重是教师模型输出的概率值，因此可以看做是教师学习模型的置信度(因为在样本标注中，我们都是直接给出一些离散值，比如0，1，但是对于同为1的样本它们的置信度也是不同的，所以通过教师模型的输出将这样的信息加了进来，并且如果教师模型比较理想，这种置信度是比较可信的)，文中进一步将该式做转换，转换为对每个样本梯度加权重的形式

$$\sum_{s=1}^{b}\frac{w_s}{\sum_{u=1}^{b} w_u}(q_{*,s} - y_{*,s}) = \sum_{s=1}^{b}\frac{p_{*,s}}{\sum_{u=1}^{b} p_{*,u}}(q_{*,s} - y_{*,s})$$

因此可以看到对整个梯度的贡献，就与教师模型的输出值产生的权重产生了影响。

最后继续提出疑问:

1. 暗知识的成功是归功于教师输出的非最大值还是简单的产出一种重要权重。

个人感想

教师-学生模型之所以有效的一个直观理解，为什么教师模型需要复杂网络，学生模型则不需要特别复杂的网络，我们直观理解，对于一个新的知识，从头开始学习(From Scratch)难度是非常大的，所以需要网络足够复杂才能够学习出这些知识(暗知识)，但是一旦这些知识被学习出来后，教师教给学生的难度就大大降低了，因此学生此时就不再需要特别复杂的网络进行学习就可以学习到教师已经学习到的网络。

所以这样来想，如果一个教师学习到的暗知识教给学生会对学生有帮助，那么多个教师模型做Ensemble的话，学生网络就可以学习到更多的知识，从而对学生模型带来帮助。

参考文章

- 模型压缩和加速的方法: https://zhuanlan.zhihu.com/p/36051603
- 知识蒸馏资料合集: https://github.com/dkozlov/awesome-knowledge-distillation